

## Review

- Univariate Normal:  $\mathcal{N}(x; \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$
- Multivariate Normal:  $\mathcal{N}(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{\det(2\pi\boldsymbol{\Sigma})} e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})}$

## Remark

*Note that  $\mathcal{N}(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \mathcal{N}(\boldsymbol{\mu}; \mathbf{x}, \boldsymbol{\Sigma})$ . It is trivial but quite useful*

# Symmetric matrices

Lemma

$$(M^T)^{-1} = (M^{-1})^T$$

# Symmetric matrices

## Lemma

$$(M^T)^{-1} = (M^{-1})^T$$

## Proof.

$$(M^{-1})^T M^T = (M M^{-1})^T = I \Rightarrow (M^{-1})^T \text{ is inverse of } M^T \quad \square$$

## Lemma

*If  $M$  is symmetric, so is  $M^{-1}$*

# Symmetric matrices

## Lemma

$$(M^T)^{-1} = (M^{-1})^T$$

## Proof.

$$(M^{-1})^T M^T = (M M^{-1})^T = I \Rightarrow (M^{-1})^T \text{ is inverse of } M^T \quad \square$$

## Lemma

*If  $M$  is symmetric, so is  $M^{-1}$*

## Proof.

$$(M^{-1})^T = (M^T)^{-1} = M^{-1} \quad \square$$

# Hermitian matrices

- An extension of transpose operation to complex matrices is the hermitian transpose operation, which is simply the transpose and conjugate of a matrix (vector)
- We denote the hermitian transpose of  $M$  as  $M^\dagger \triangleq \overline{M}^T$ , when  $\overline{M}$  is the complex conjugate of  $M$
- A matrix is Hermitian if  $M^\dagger = M$ . **Note that a real symmetric matrix is Hermitian**

# Eigenvalues of Hermitian matrices

## Lemma

*If  $M$  is Hermitian ( $M^\dagger = M$ ), all eigenvalues are real*

# Eigenvalues of Hermitian matrices

## Lemma

*If  $M$  is Hermitian ( $M^\dagger = M$ ), all eigenvalues are real*

## Proof.

$$\overline{\lambda(x^\dagger x)} = (\lambda x)^\dagger x = (Mx)^\dagger x = x^\dagger M^\dagger x = x^\dagger Mx = x^\dagger (\lambda x) = \lambda(x^\dagger x) \quad \square$$

## Lemma

*If  $M$  is Hermitian, eigenvectors of different eigenvalues are orthogonal*

# Eigenvalues of Hermitian matrices

## Lemma

If  $M$  is Hermitian ( $M^\dagger = M$ ), all eigenvalues are real

## Proof.

$$\overline{\lambda(x^\dagger x)} = (\lambda x)^\dagger x = (Mx)^\dagger x = x^\dagger M^\dagger x = x^\dagger Mx = x^\dagger (\lambda x) = \lambda(x^\dagger x) \quad \square$$

## Lemma

If  $M$  is Hermitian, eigenvectors of different eigenvalues are orthogonal

## Proof.

$$\begin{aligned} \lambda_1 x_1^\dagger x_2 &= (Mx_1)^\dagger x_2 = x_1^\dagger Mx_2 = \lambda_2 x_1^\dagger x_2 \\ \Rightarrow \lambda_1 \neq \lambda_2 &\Rightarrow x_1^\dagger x_2 = 0 \end{aligned}$$

□



# Hermitian matrices are diagonalizable

## Lemma

*Hermitian matrices are diagonalizable*

## Proof.

We will sketch the proof by construction. For any  $n$ -d Hermitian matrix  $M$ , consider an eigenvalue  $\lambda$  and corresponding eigenvector  $u$ , without loss of generality, let's also normalize  $u$  such that  $\|u\| = 1$ . Consider the subspace orthogonal to  $u$ ,  $U^\perp$ , and let  $v_1, \dots, v_{n-1}$  be arbitrary orthonormal basis of  $U^\perp$ . Note that for any  $k$ ,  $Av_k$  will be orthogonal to  $u$  since

$$u^\dagger Mv_k = u^\dagger M^\dagger v_k = (Mu)^\dagger v_k = \lambda u^\dagger v_k = 0.$$

Thus,  $(u, v_1, \dots, v_{n-1})^\dagger M (u, v_1, \dots, v_{n-1}) = \begin{pmatrix} \lambda & 0 \\ 0 & M' \end{pmatrix}$ . Moreover,  $M'$  is also a Hermitian matrix with one less dimension. We can apply the same process on  $M'$  and “diagonalize” one more row/column. That is,

$\begin{pmatrix} 1 & 0 \\ 0 & P' \end{pmatrix}^\dagger P^\dagger M P \begin{pmatrix} 1 & 0 \\ 0 & P' \end{pmatrix} = \begin{pmatrix} \lambda & 0 & \dots \\ 0 & \lambda' & \\ & & M'' \end{pmatrix}$ . We can repeat this until the entire  $M$  is diagonalized □

# Hermitian matrices are diagonalizable

## Remark

*A Hermitian matrix is diagonalized by its eigenvectors and the diagonalized matrix is composed of the corresponding eigenvalues. That is,*

$$(v_1, \dots, v_n)^\dagger \underbrace{M(v_1, \dots, v_n)}_V = \begin{pmatrix} \lambda_1 & 0 & \dots \\ 0 & \lambda_2 & \\ \vdots & & \ddots \end{pmatrix}.$$

*Moreover,  $V$  is unitary (orthogonal), i.e.,  $V^\dagger V = I$  and thus  $V^{-1} = V^\dagger$*

## Remark

*The reverse is obviously true. If a matrix can be diagonalized by a unitary matrix into a real diagonal matrix, the matrix is Hermitian*

## Remark

*Recall that real-symmetric matrices are Hermitian, thus can be diagonalized by its eigenvectors also*

# Positive definite matrices

## Definition (Positive definite)

For a Hermitian matrix  $M$ , it is positive definite iff  $\forall x, x^\dagger Mx > 0$

## Definition (Positive semi-definite)

For a Hermitian matrix  $M$ , it is positive semi-definite iff  $\forall x, x^\dagger Mx \geq 0$

## Remark

*$M$  is positive definite (semi-definite) iff all its eigenvalue is larger (larger or equal to) 0*

# Positive definite matrices

## Definition (Positive definite)

For a Hermitian matrix  $M$ , it is positive definite iff  $\forall x, x^\dagger Mx > 0$

## Definition (Positive semi-definite)

For a Hermitian matrix  $M$ , it is positive semi-definite iff  $\forall x, x^\dagger Mx \geq 0$

## Remark

*$M$  is positive definite (semi-definite) iff all its eigenvalue is larger (larger or equal to) 0*

## Proof.

$\Rightarrow$ : assume positive definite but some eigenvalue  $< 0$ , WLOG, let  $\lambda_1 < 0$ , then  $v_1^\dagger Mv_1 = \lambda_1 < 0$  contradicts that  $M$  is positive definite

$\Leftarrow$ : If  $\forall k, \lambda_k > 0$ , for any  $x$ ,

$$x^\dagger Mx = (V^\dagger x)^\dagger \begin{pmatrix} \lambda_1 & 0 \\ & \ddots \\ 0 & \ddots \end{pmatrix} V^\dagger x = \sum_i \lambda_i (V^\dagger x)_i^2 > 0$$

□

# Some probability basic

- Probability mass function (pmf) for discrete random variable (r.v.)  $X$ 
  - $p(x) \geq 0$
  - $p(x) \leq 1$
  - $\sum_x p(x) = 1$

# Some probability basic

- Probability mass function (pmf) for discrete random variable (r.v.)  $X$ 
  - $p(x) \geq 0$
  - $p(x) \leq 1$
  - $\sum_x p(x) = 1$
- Probability density function (pdf) for continuous r.v.  $X$ 
  - $p(x) \geq 0$
  - $p(x)$  can be larger than 1
  - $Pr(a \leq X \leq b) = \int_a^b p(x)$  (Area between  $p(x)$  and  $x$ -axis)
  - $\int_x p(x) = 1$

# Some probability basic

- Probability mass function (pmf) for discrete random variable (r.v.)  $X$ 
  - $p(x) \geq 0$
  - $p(x) \leq 1$
  - $\sum_x p(x) = 1$
- Probability density function (pdf) for continuous r.v.  $X$ 
  - $p(x) \geq 0$
  - $p(x)$  can be larger than 1
  - $Pr(a \leq X \leq b) = \int_a^b p(x)$  (Area between  $p(x)$  and  $x$ -axis)
  - $\int_x p(x) = 1$
- Marginalization:  $\sum_x p(x, y) = p(y)$

# Some probability basic

- Probability mass function (pmf) for discrete random variable (r.v.)  $X$ 
  - $p(x) \geq 0$
  - $p(x) \leq 1$
  - $\sum_x p(x) = 1$
- Probability density function (pdf) for continuous r.v.  $X$ 
  - $p(x) \geq 0$
  - $p(x)$  can be larger than 1
  - $Pr(a \leq X \leq b) = \int_a^b p(x)$  (Area between  $p(x)$  and  $x$ -axis)
  - $\int_x p(x) = 1$
- Marginalization:  $\sum_x p(x, y) = p(y)$
- Conditional probability:  $p(x|y) = \frac{p(x,y)}{p(y)}$ 
  - N.B.  $\sum_x p(x|y) = 1$  but  $\sum_y p(x|y) \neq 1$



# Some probability basic

- Probability mass function (pmf) for discrete random variable (r.v.)  $X$ 
  - $p(x) \geq 0$
  - $p(x) \leq 1$
  - $\sum_x p(x) = 1$
- Probability density function (pdf) for continuous r.v.  $X$ 
  - $p(x) \geq 0$
  - $p(x)$  can be larger than 1
  - $Pr(a \leq X \leq b) = \int_a^b p(x)$  (Area between  $p(x)$  and  $x$ -axis)
  - $\int_x p(x) = 1$
- Marginalization:  $\sum_x p(x, y) = p(y)$
- Conditional probability:  $p(x|y) = \frac{p(x,y)}{p(y)}$ 
  - N.B.  $\sum_x p(x|y) = 1$  but  $\sum_y p(x|y) \neq 1$
- Chain rule:  $p(x, y, z) = p(x)p(y|x)p(z|x, y)$

# Some probability basic

- Probability mass function (pmf) for discrete random variable (r.v.)  $X$ 
  - $p(x) \geq 0$
  - $p(x) \leq 1$
  - $\sum_x p(x) = 1$
- Probability density function (pdf) for continuous r.v.  $X$ 
  - $p(x) \geq 0$
  - $p(x)$  can be larger than 1
  - $Pr(a \leq X \leq b) = \int_a^b p(x)$  (Area between  $p(x)$  and  $x$ -axis)
  - $\int_x p(x) = 1$
- Marginalization:  $\sum_x p(x, y) = p(y)$
- Conditional probability:  $p(x|y) = \frac{p(x,y)}{p(y)}$ 
  - N.B.  $\sum_x p(x|y) = 1$  but  $\sum_y p(x|y) \neq 1$
- Chain rule:  $p(x, y, z) = p(x)p(y|x)p(z|x, y)$
- Independence:  $p(x, y) = p(x)p(y)$ ,  $X \perp Y$

# Some probability basic

- Probability mass function (pmf) for discrete random variable (r.v.)  $X$ 
  - $p(x) \geq 0$
  - $p(x) \leq 1$
  - $\sum_x p(x) = 1$
- Probability density function (pdf) for continuous r.v.  $X$ 
  - $p(x) \geq 0$
  - $p(x)$  can be larger than 1
  - $Pr(a \leq X \leq b) = \int_a^b p(x)$  (Area between  $p(x)$  and  $x$ -axis)
  - $\int_x p(x) = 1$
- Marginalization:  $\sum_x p(x, y) = p(y)$
- Conditional probability:  $p(x|y) = \frac{p(x,y)}{p(y)}$ 
  - N.B.  $\sum_x p(x|y) = 1$  but  $\sum_y p(x|y) \neq 1$
- Chain rule:  $p(x, y, z) = p(x)p(y|x)p(z|x, y)$
- Independence:  $p(x, y) = p(x)p(y)$ ,  $X \perp Y$
- Markov property and conditional independence:  
 $p(x, y|z) = p(x|z)p(y|z)$ ,  $X \perp Y|Z$ ,  $X \leftrightarrow Z \leftrightarrow Y$

# Some probability basic

- Probability mass function (pmf) for discrete random variable (r.v.)  $X$ 
  - $p(x) \geq 0$
  - $p(x) \leq 1$
  - $\sum_x p(x) = 1$
- Probability density function (pdf) for continuous r.v.  $X$ 
  - $p(x) \geq 0$
  - $p(x)$  can be larger than 1
  - $Pr(a \leq X \leq b) = \int_a^b p(x)$  (Area between  $p(x)$  and  $x$ -axis)
  - $\int_x p(x) = 1$
- Marginalization:  $\sum_x p(x, y) = p(y)$
- Conditional probability:  $p(x|y) = \frac{p(x,y)}{p(y)}$ 
  - N.B.  $\sum_x p(x|y) = 1$  but  $\sum_y p(x|y) \neq 1$
- Chain rule:  $p(x, y, z) = p(x)p(y|x)p(z|x, y)$
- Independence:  $p(x, y) = p(x)p(y)$ ,  $X \perp\!\!\!\perp Y$
- Markov property and conditional independence:  
 $p(x, y|z) = p(x|z)p(y|z)$ ,  $X \perp\!\!\!\perp Y|Z$ ,  $X \leftrightarrow Z \leftrightarrow Y$
- Inference: ML, MAP, Bayesian

# Independence but not conditional independence

Consider flipping two coins with outcomes store as  $X$  and  $Y$ , say 1 represents a head and 0 represents a tail

- In general the two outcomes should be independent (maybe unless if you are some professional/magical gambler), so we have  $X \perp\!\!\!\perp Y$
- Now, let  $Z = X \oplus Y$ , where  $\oplus$  is the exclusive or operation ( $1 \oplus 0 = 0 \oplus 1 = 1$  and  $1 \oplus 1 = 0 \oplus 0 = 0$ )
  - Even though  $X \perp\!\!\!\perp Y$ ,  $X \not\perp\!\!\!\perp Y|Z$
  - Actually given  $Z$ ,  $X$  “depends” very much on  $Y$  since from  $X = Y \oplus Z$ , we can find out  $X$  precisely given  $Y$
  - We can also check the condition  $X \perp\!\!\!\perp Y|Z$  by comparing the probability  $p(x|z, y)$  with  $p(x|z)$ 
    - For example,  $p_{X|Z}(0|0) = 0.5 \neq 1 = p_{X|Z,Y}(0|0,0)$ . Thus  $X \not\perp\!\!\!\perp Y|Z$  cannot be true

## Review

- Univariate Normal:  $\mathcal{N}(x; \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$
- Multivariate Normal:  $\mathcal{N}(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{\det(2\pi\boldsymbol{\Sigma})} e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})}$
- Covariance matrices are Hermitian and thus can be diagonalized by its eigenvectors. Covariance matrices are positive semi-definite (eigenvalues  $\geq 0$ )
- Independence:  $p(x, y) = p(x)p(y)$ ,  $X \perp\!\!\!\perp Y$
- Markov property and conditional independence:  
 $p(x, y|z) = p(x|z)p(y|z)$ ,  $X \perp\!\!\!\perp Y|Z$ ,  $X \leftrightarrow Z \leftrightarrow Y$

## Remark

*Note that  $\mathcal{N}(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \mathcal{N}(\boldsymbol{\mu}; \mathbf{x}, \boldsymbol{\Sigma})$ . It is trivial but quite useful*

# Inference

$o$ : (Observed) evidence,  $\theta$ : Parameter,  $x$ : prediction

## Maximum Likelihood (ML)

$$\hat{x} = \arg \max_x p(x|\hat{\theta}), \hat{\theta} = \arg \max_{\theta} p(o|\theta)$$

## Maximum A Posteriori (MAP)

$$\hat{x} = \arg \max_x p(x|\hat{\theta}), \hat{\theta} = \arg \max_{\theta} p(\theta|o)$$

## Bayesian

$$\hat{x} = \sum_x x \underbrace{\sum_{\theta} p(x|\theta)p(\theta|o)}_{p(x|o)}$$

where  $p(\theta|o) = \frac{p(o|\theta)p(\theta)}{p(o)} \propto p(o|\theta) \underbrace{p(\theta)}_{\text{prior}}$

# Covariance matrices

## Definition (Covariance matrices)

Recall that for a vector random variable  $\mathbf{X} = [X_1, X_2, \dots, X_n]^T$ , the covariance matrix  $\Sigma \triangleq E[(\mathbf{X} - \boldsymbol{\mu})(\mathbf{X} - \boldsymbol{\mu})^T]$

## Remark

*Covariance matrices are always positive semi-definite since  $\forall u$ ,  $u^T \Sigma u = E[u^T (\mathbf{X} - \boldsymbol{\mu})(\mathbf{X} - \boldsymbol{\mu})^T u] = E[\|(\mathbf{X} - \boldsymbol{\mu})^T u\|^2] \geq 0$*

## Remark

*In general, we usually would like to assume  $\Sigma$  to be strictly positive definite. Because otherwise it means that some of its eigenvalues are zero and so in some dimension, there is actually no variation and is just constant along that dimension. Representing those dimension as random variable is troublesome since "1/ $\sigma^2$ " which occurs often will become infinite. Instead we can always simply strip away those dimensions to avoid complications*



# Eigenvectors and eigenvalues of covariance matrices

- WLOG, let's assume  $\mathbf{X} = [X_1, X_2, \dots, X_n]^T$  is zero mean. So the covariance matrix  $\Sigma_X = E[\mathbf{X}\mathbf{X}^T]$

# Eigenvectors and eigenvalues of covariance matrices

- WLOG, let's assume  $\mathbf{X} = [X_1, X_2, \dots, X_n]^T$  is zero mean. So the covariance matrix  $\Sigma_X = E[\mathbf{X}\mathbf{X}^T]$
- Covariance matrices are real symmetric (hence Hermitian) and so can be diagonalized by its eigenvectors. That is,
  - $P^T \Sigma_X P = D$ , where  $P = [u_1, u_2, \dots, u_n]$  with  $u_k$  being eigenvectors of  $\Sigma$  and  $D$  is a diagonal matrix with eigenvalues  $\lambda_1, \lambda_2, \dots, \lambda_n$  as the diagonal elements

# Eigenvectors and eigenvalues of covariance matrices

- WLOG, let's assume  $\mathbf{X} = [X_1, X_2, \dots, X_n]^T$  is zero mean. So the covariance matrix  $\Sigma_X = E[\mathbf{X}\mathbf{X}^T]$
- Covariance matrices are real symmetric (hence Hermitian) and so can be diagonalized by its eigenvectors. That is,
  - $P^T \Sigma_X P = D$ , where  $P = [u_1, u_2, \dots, u_n]$  with  $u_k$  being eigenvectors of  $\Sigma$  and  $D$  is a diagonal matrix with eigenvalues  $\lambda_1, \lambda_2, \dots, \lambda_n$  as the diagonal elements
- Let  $\mathbf{Y} = P^T \mathbf{X}$ , note that the covariance matrix of  $\mathbf{Y}$

$$\Sigma_Y = E[\mathbf{Y}\mathbf{Y}^T] = E[P^T \mathbf{X}\mathbf{X}^T P] = P^T E[\mathbf{X}\mathbf{X}^T] P = P^T \Sigma_X P = D$$

is diagonalized

# Eigenvectors and eigenvalues of covariance matrices

- WLOG, let's assume  $\mathbf{X} = [X_1, X_2, \dots, X_n]^T$  is zero mean. So the covariance matrix  $\Sigma_X = E[\mathbf{X}\mathbf{X}^T]$
- Covariance matrices are real symmetric (hence Hermitian) and so can be diagonalized by its eigenvectors. That is,
  - $P^T \Sigma_X P = D$ , where  $P = [u_1, u_2, \dots, u_n]$  with  $u_k$  being eigenvectors of  $\Sigma$  and  $D$  is a diagonal matrix with eigenvalues  $\lambda_1, \lambda_2, \dots, \lambda_n$  as the diagonal elements
- Let  $\mathbf{Y} = P^T \mathbf{X}$ , note that the covariance matrix of  $\mathbf{Y}$

$$\Sigma_Y = E[\mathbf{Y}\mathbf{Y}^T] = E[P^T \mathbf{X}\mathbf{X}^T P] = P^T E[\mathbf{X}\mathbf{X}^T] P = P^T \Sigma_X P = D$$

is diagonalized

- So the variance of  $Y_k$  is simply  $\lambda_k$
- $E[Y_i Y_j] = 0$  for  $i \neq j$ . That is,  $Y_i \perp Y_j$  for  $i \neq j$
- Note that  $\mathbf{Y} = P^T \mathbf{X}$  is just principal component analysis (PCA)

# Principal component analysis (PCA)

- Recall that  $\Sigma = E[\mathbf{X}\mathbf{X}^T]$  (assume  $\mathbf{X}$  is zero-mean) and  $\mathbf{Y} = P^T\mathbf{X}$  with  $E[\mathbf{Y}\mathbf{Y}^T] = P^T\Sigma P = D$
- Assume that the diagonal of  $D$  (note that those are eigenvalues) are arranged in descending order that  $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n$

---

$${}^1tr(AB) = \sum_i \sum_j a_{i,j} b_{j,i} = \sum_j \sum_i b_{j,i} a_{i,j} = tr(BA)$$

# Principal component analysis (PCA)

- Recall that  $\Sigma = E[\mathbf{X}\mathbf{X}^T]$  (assume  $\mathbf{X}$  is zero-mean) and  $\mathbf{Y} = P^T\mathbf{X}$  with  $E[\mathbf{Y}\mathbf{Y}^T] = P^T\Sigma P = D$
- Assume that the diagonal of  $D$  (note that those are eigenvalues) are arranged in descending order that  $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n$ 
  - Generate an approximate  $\hat{\mathbf{Y}}$  of  $\mathbf{Y}$  by setting all components except first  $k$  as 0

---

$${}^1tr(AB) = \sum_i \sum_j a_{i,j} b_{j,i} = \sum_j \sum_i b_{j,i} a_{i,j} = tr(BA)$$

# Principal component analysis (PCA)

- Recall that  $\Sigma = E[\mathbf{X}\mathbf{X}^T]$  (assume  $\mathbf{X}$  is zero-mean) and  $\mathbf{Y} = P^T\mathbf{X}$  with  $E[\mathbf{Y}\mathbf{Y}^T] = P^T\Sigma P = D$
- Assume that the diagonal of  $D$  (note that those are eigenvalues) are arranged in descending order that  $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n$ 
  - Generate an approximate  $\hat{\mathbf{Y}}$  of  $\mathbf{Y}$  by setting all components except first  $k$  as 0
  - The mean square error (mse) of<sup>1</sup>  $\hat{\mathbf{Y}} = E[(\mathbf{Y} - \hat{\mathbf{Y}})^T(\mathbf{Y} - \hat{\mathbf{Y}})]$

---

<sup>1</sup> $tr(AB) = \sum_i \sum_j a_{i,j} b_{j,i} = \sum_j \sum_i b_{j,i} a_{i,j} = tr(BA)$

# Principal component analysis (PCA)

- Recall that  $\Sigma = E[\mathbf{X}\mathbf{X}^T]$  (assume  $\mathbf{X}$  is zero-mean) and  $\mathbf{Y} = P^T\mathbf{X}$  with  $E[\mathbf{Y}\mathbf{Y}^T] = P^T\Sigma P = D$
- Assume that the diagonal of  $D$  (note that those are eigenvalues) are arranged in descending order that  $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n$ 
  - Generate an approximate  $\hat{\mathbf{Y}}$  of  $\mathbf{Y}$  by setting all components except first  $k$  as 0
  - The mean square error (mse) of<sup>1</sup>  $\hat{\mathbf{Y}} = E[(\mathbf{Y} - \hat{\mathbf{Y}})^T(\mathbf{Y} - \hat{\mathbf{Y}})]$   
 $= \text{tr}(E[(\mathbf{Y} - \hat{\mathbf{Y}})^T(\mathbf{Y} - \hat{\mathbf{Y}})])$

---

<sup>1</sup> $\text{tr}(AB) = \sum_i \sum_j a_{i,j} b_{j,i} = \sum_j \sum_i b_{j,i} a_{i,j} = \text{tr}(BA)$



# Principal component analysis (PCA)

- Recall that  $\Sigma = E[\mathbf{X}\mathbf{X}^T]$  (assume  $\mathbf{X}$  is zero-mean) and  $\mathbf{Y} = P^T\mathbf{X}$  with  $E[\mathbf{Y}\mathbf{Y}^T] = P^T\Sigma P = D$
- Assume that the diagonal of  $D$  (note that those are eigenvalues) are arranged in descending order that  $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n$ 
  - Generate an approximate  $\hat{\mathbf{Y}}$  of  $\mathbf{Y}$  by setting all components except first  $k$  as 0
  - The mean square error (mse) of<sup>1</sup>  $\hat{\mathbf{Y}} = E[(\mathbf{Y} - \hat{\mathbf{Y}})^T(\mathbf{Y} - \hat{\mathbf{Y}})]$   
 $= tr(E[(\mathbf{Y} - \hat{\mathbf{Y}})^T(\mathbf{Y} - \hat{\mathbf{Y}})]) = E[tr((\mathbf{Y} - \hat{\mathbf{Y}})^T(\mathbf{Y} - \hat{\mathbf{Y}}))]$

---

<sup>1</sup> $tr(AB) = \sum_i \sum_j a_{i,j} b_{j,i} = \sum_j \sum_i b_{j,i} a_{i,j} = tr(BA)$

# Principal component analysis (PCA)

- Recall that  $\Sigma = E[\mathbf{X}\mathbf{X}^T]$  (assume  $\mathbf{X}$  is zero-mean) and  $\mathbf{Y} = P^T\mathbf{X}$  with  $E[\mathbf{Y}\mathbf{Y}^T] = P^T\Sigma P = D$
- Assume that the diagonal of  $D$  (note that those are eigenvalues) are arranged in descending order that  $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n$ 
  - Generate an approximate  $\hat{\mathbf{Y}}$  of  $\mathbf{Y}$  by setting all components except first  $k$  as 0
  - The mean square error (mse) of<sup>1</sup>  $\hat{\mathbf{Y}} = E[(\mathbf{Y} - \hat{\mathbf{Y}})^T(\mathbf{Y} - \hat{\mathbf{Y}})]$   
 $= \text{tr}(E[(\mathbf{Y} - \hat{\mathbf{Y}})^T(\mathbf{Y} - \hat{\mathbf{Y}})]) = E[\text{tr}((\mathbf{Y} - \hat{\mathbf{Y}})^T(\mathbf{Y} - \hat{\mathbf{Y}}))]$   
 $= E[\text{tr}((\mathbf{Y} - \hat{\mathbf{Y}})(\mathbf{Y} - \hat{\mathbf{Y}})^T)] = \text{tr}(E[(\mathbf{Y} - \hat{\mathbf{Y}})(\mathbf{Y} - \hat{\mathbf{Y}})^T])$

---

<sup>1</sup> $\text{tr}(AB) = \sum_i \sum_j a_{i,j} b_{j,i} = \sum_j \sum_i b_{j,i} a_{i,j} = \text{tr}(BA)$

# Principal component analysis (PCA)

- Recall that  $\Sigma = E[\mathbf{X}\mathbf{X}^T]$  (assume  $\mathbf{X}$  is zero-mean) and  $\mathbf{Y} = P^T\mathbf{X}$  with  $E[\mathbf{Y}\mathbf{Y}^T] = P^T\Sigma P = D$
- Assume that the diagonal of  $D$  (note that those are eigenvalues) are arranged in descending order that  $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n$ 
  - Generate an approximate  $\hat{\mathbf{Y}}$  of  $\mathbf{Y}$  by setting all components except first  $k$  as 0
  - The mean square error (mse) of<sup>1</sup>  $\hat{\mathbf{Y}} = E[(\mathbf{Y} - \hat{\mathbf{Y}})^T(\mathbf{Y} - \hat{\mathbf{Y}})]$   
 $= \text{tr}(E[(\mathbf{Y} - \hat{\mathbf{Y}})^T(\mathbf{Y} - \hat{\mathbf{Y}})]) = E[\text{tr}((\mathbf{Y} - \hat{\mathbf{Y}})^T(\mathbf{Y} - \hat{\mathbf{Y}}))]$   
 $= E[\text{tr}((\mathbf{Y} - \hat{\mathbf{Y}})(\mathbf{Y} - \hat{\mathbf{Y}})^T)] = \text{tr}(E[(\mathbf{Y} - \hat{\mathbf{Y}})(\mathbf{Y} - \hat{\mathbf{Y}})^T]) = \sum_{i=k+1}^n \lambda_i$
  - Similarly, if we “reconstruct”  $\mathbf{X}$  as  $\hat{\mathbf{X}} = P\hat{\mathbf{Y}}$ . The mse of  $\hat{\mathbf{X}} = E[(\mathbf{X} - \hat{\mathbf{X}})^T(\mathbf{X} - \hat{\mathbf{X}})] = \text{tr}(E[(\mathbf{X} - \hat{\mathbf{X}})(\mathbf{X} - \hat{\mathbf{X}})^T])$

---

<sup>1</sup> $\text{tr}(AB) = \sum_i \sum_j a_{i,j} b_{j,i} = \sum_j \sum_i b_{j,i} a_{i,j} = \text{tr}(BA)$

# Principal component analysis (PCA)

- Recall that  $\Sigma = E[\mathbf{X}\mathbf{X}^T]$  (assume  $\mathbf{X}$  is zero-mean) and  $\mathbf{Y} = P^T\mathbf{X}$  with  $E[\mathbf{Y}\mathbf{Y}^T] = P^T\Sigma P = D$
- Assume that the diagonal of  $D$  (note that those are eigenvalues) are arranged in descending order that  $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n$ 
  - Generate an approximate  $\hat{\mathbf{Y}}$  of  $\mathbf{Y}$  by setting all components except first  $k$  as 0
  - The mean square error (mse) of<sup>1</sup>  $\hat{\mathbf{Y}} = E[(\mathbf{Y} - \hat{\mathbf{Y}})^T(\mathbf{Y} - \hat{\mathbf{Y}})]$   
 $= \text{tr}(E[(\mathbf{Y} - \hat{\mathbf{Y}})^T(\mathbf{Y} - \hat{\mathbf{Y}})]) = E[\text{tr}((\mathbf{Y} - \hat{\mathbf{Y}})^T(\mathbf{Y} - \hat{\mathbf{Y}}))]$   
 $= E[\text{tr}((\mathbf{Y} - \hat{\mathbf{Y}})(\mathbf{Y} - \hat{\mathbf{Y}})^T)] = \text{tr}(E[(\mathbf{Y} - \hat{\mathbf{Y}})(\mathbf{Y} - \hat{\mathbf{Y}})^T]) = \sum_{i=k+1}^n \lambda_i$
  - Similarly, if we “reconstruct”  $\mathbf{X}$  as  $\hat{\mathbf{X}} = P\hat{\mathbf{Y}}$ . The mse of  $\hat{\mathbf{X}} = E[(\mathbf{X} - \hat{\mathbf{X}})^T(\mathbf{X} - \hat{\mathbf{X}})] = \text{tr}(E[(\mathbf{X} - \hat{\mathbf{X}})(\mathbf{X} - \hat{\mathbf{X}})^T]) = \text{tr}(PE[(\mathbf{Y} - \hat{\mathbf{Y}})(\mathbf{Y} - \hat{\mathbf{Y}})^T]P^T)$

---

<sup>1</sup> $\text{tr}(AB) = \sum_i \sum_j a_{i,j} b_{j,i} = \sum_j \sum_i b_{j,i} a_{i,j} = \text{tr}(BA)$

# Principal component analysis (PCA)

- Recall that  $\Sigma = E[\mathbf{X}\mathbf{X}^T]$  (assume  $\mathbf{X}$  is zero-mean) and  $\mathbf{Y} = P^T\mathbf{X}$  with  $E[\mathbf{Y}\mathbf{Y}^T] = P^T\Sigma P = D$
- Assume that the diagonal of  $D$  (note that those are eigenvalues) are arranged in descending order that  $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n$ 
  - Generate an approximate  $\hat{\mathbf{Y}}$  of  $\mathbf{Y}$  by setting all components except first  $k$  as 0
  - The mean square error (mse) of<sup>1</sup>  $\hat{\mathbf{Y}} = E[(\mathbf{Y} - \hat{\mathbf{Y}})^T(\mathbf{Y} - \hat{\mathbf{Y}})]$   
 $= \text{tr}(E[(\mathbf{Y} - \hat{\mathbf{Y}})^T(\mathbf{Y} - \hat{\mathbf{Y}})]) = E[\text{tr}((\mathbf{Y} - \hat{\mathbf{Y}})^T(\mathbf{Y} - \hat{\mathbf{Y}}))]$   
 $= E[\text{tr}((\mathbf{Y} - \hat{\mathbf{Y}})(\mathbf{Y} - \hat{\mathbf{Y}})^T)] = \text{tr}(E[(\mathbf{Y} - \hat{\mathbf{Y}})(\mathbf{Y} - \hat{\mathbf{Y}})^T]) = \sum_{i=k+1}^n \lambda_i$
  - Similarly, if we “reconstruct”  $\mathbf{X}$  as  $\hat{\mathbf{X}} = P\hat{\mathbf{Y}}$ . The mse of  $\hat{\mathbf{X}} = E[(\mathbf{X} - \hat{\mathbf{X}})^T(\mathbf{X} - \hat{\mathbf{X}})] = \text{tr}(E[(\mathbf{X} - \hat{\mathbf{X}})(\mathbf{X} - \hat{\mathbf{X}})^T]) = \text{tr}(PE[(\mathbf{Y} - \hat{\mathbf{Y}})(\mathbf{Y} - \hat{\mathbf{Y}})^T]P^T) = \text{tr}(P^TPE[(\mathbf{Y} - \hat{\mathbf{Y}})(\mathbf{Y} - \hat{\mathbf{Y}})^T])$

---

<sup>1</sup> $\text{tr}(AB) = \sum_i \sum_j a_{i,j} b_{j,i} = \sum_j \sum_i b_{j,i} a_{i,j} = \text{tr}(BA)$

# Principal component analysis (PCA)

- Recall that  $\Sigma = E[\mathbf{X}\mathbf{X}^T]$  (assume  $\mathbf{X}$  is zero-mean) and  $\mathbf{Y} = P^T\mathbf{X}$  with  $E[\mathbf{Y}\mathbf{Y}^T] = P^T\Sigma P = D$
- Assume that the diagonal of  $D$  (note that those are eigenvalues) are arranged in descending order that  $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n$ 
  - Generate an approximate  $\hat{\mathbf{Y}}$  of  $\mathbf{Y}$  by setting all components except first  $k$  as 0
  - The mean square error (mse) of<sup>1</sup>  $\hat{\mathbf{Y}} = E[(\mathbf{Y} - \hat{\mathbf{Y}})^T(\mathbf{Y} - \hat{\mathbf{Y}})]$   
 $= \text{tr}(E[(\mathbf{Y} - \hat{\mathbf{Y}})^T(\mathbf{Y} - \hat{\mathbf{Y}})]) = E[\text{tr}((\mathbf{Y} - \hat{\mathbf{Y}})^T(\mathbf{Y} - \hat{\mathbf{Y}}))]$   
 $= E[\text{tr}((\mathbf{Y} - \hat{\mathbf{Y}})(\mathbf{Y} - \hat{\mathbf{Y}})^T)] = \text{tr}(E[(\mathbf{Y} - \hat{\mathbf{Y}})(\mathbf{Y} - \hat{\mathbf{Y}})^T]) = \sum_{i=k+1}^n \lambda_i$
  - Similarly, if we “reconstruct”  $\mathbf{X}$  as  $\hat{\mathbf{X}} = P\hat{\mathbf{Y}}$ . The mse of  $\hat{\mathbf{X}} = E[(\mathbf{X} - \hat{\mathbf{X}})^T(\mathbf{X} - \hat{\mathbf{X}})] = \text{tr}(E[(\mathbf{X} - \hat{\mathbf{X}})(\mathbf{X} - \hat{\mathbf{X}})^T]) = \text{tr}(PE[(\mathbf{Y} - \hat{\mathbf{Y}})(\mathbf{Y} - \hat{\mathbf{Y}})]P^T) = \text{tr}(P^TPE[(\mathbf{Y} - \hat{\mathbf{Y}})(\mathbf{Y} - \hat{\mathbf{Y}})]) = \sum_{i=k+1}^n \lambda_i$

---

<sup>1</sup> $\text{tr}(AB) = \sum_i \sum_j a_{i,j} b_{j,i} = \sum_j \sum_i b_{j,i} a_{i,j} = \text{tr}(BA)$

# Principal component analysis (PCA)

- Recall that  $\Sigma = E[\mathbf{X}\mathbf{X}^T]$  (assume  $\mathbf{X}$  is zero-mean) and  $\mathbf{Y} = P^T\mathbf{X}$  with  $E[\mathbf{Y}\mathbf{Y}^T] = P^T\Sigma P = D$
- Assume that the diagonal of  $D$  (note that those are eigenvalues) are arranged in descending order that  $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n$ 
  - Generate an approximate  $\hat{\mathbf{Y}}$  of  $\mathbf{Y}$  by setting all components except first  $k$  as 0
  - The mean square error (mse) of<sup>1</sup>  $\hat{\mathbf{Y}} = E[(\mathbf{Y} - \hat{\mathbf{Y}})^T(\mathbf{Y} - \hat{\mathbf{Y}})]$   
 $= \text{tr}(E[(\mathbf{Y} - \hat{\mathbf{Y}})^T(\mathbf{Y} - \hat{\mathbf{Y}})]) = E[\text{tr}((\mathbf{Y} - \hat{\mathbf{Y}})^T(\mathbf{Y} - \hat{\mathbf{Y}}))]$   
 $= E[\text{tr}((\mathbf{Y} - \hat{\mathbf{Y}})(\mathbf{Y} - \hat{\mathbf{Y}})^T)] = \text{tr}(E[(\mathbf{Y} - \hat{\mathbf{Y}})(\mathbf{Y} - \hat{\mathbf{Y}})^T]) = \sum_{i=k+1}^n \lambda_i$
  - Similarly, if we “reconstruct”  $\mathbf{X}$  as  $\hat{\mathbf{X}} = P\hat{\mathbf{Y}}$ . The mse of  $\hat{\mathbf{X}} = E[(\mathbf{X} - \hat{\mathbf{X}})^T(\mathbf{X} - \hat{\mathbf{X}})] = \text{tr}(E[(\mathbf{X} - \hat{\mathbf{X}})(\mathbf{X} - \hat{\mathbf{X}})^T]) = \text{tr}(PE[(\mathbf{Y} - \hat{\mathbf{Y}})(\mathbf{Y} - \hat{\mathbf{Y}})^T]P^T) = \text{tr}(P^TPE[(\mathbf{Y} - \hat{\mathbf{Y}})(\mathbf{Y} - \hat{\mathbf{Y}})^T]) = \sum_{i=k+1}^n \lambda_i$
  - Note that the eigenvectors of  $\Sigma$  (columns of  $P$ ) are known as the principal components

---

<sup>1</sup> $\text{tr}(AB) = \sum_i \sum_j a_{i,j} b_{j,i} = \sum_j \sum_i b_{j,i} a_{i,j} = \text{tr}(BA)$

# Practical PCA

In practice, we typically are given a dataset with samples of  $\mathbf{X}$  instead of the distribution or covariance matrix of  $\mathbf{X}$ . Denote the data as  $\mathcal{X}$  with each row is a data point and a total of  $m$  data points. Thus  $\mathcal{X}$  is an  $m$  by  $n$  matrix

---

<sup>2</sup>I used the matlab notations for *ones*( $\cdot$ ) and *mean*( $\cdot$ ) here

<sup>3</sup>Note that  $\hat{\Sigma}$  won't be full rank and positive definite as one would hope 



# Practical PCA

In practice, we typically are given a dataset with samples of  $\mathbf{X}$  instead of the distribution or covariance matrix of  $\mathbf{X}$ . Denote the data as  $\mathcal{X}$  with each row is a data point and a total of  $m$  data points. Thus  $\mathcal{X}$  is an  $m$  by  $n$  matrix

- Data are rarely zero-mean to begin with, but we can easily preprocess it by subtracting the mean. That is<sup>2</sup>  $\mathcal{X} \leftarrow \mathcal{X} - \mathbf{ones}(m, 1)\mathit{mean}(\mathcal{X})$

---

<sup>2</sup>I used the matlab notations for  $\mathit{ones}(\cdot)$  and  $\mathit{mean}(\cdot)$  here

<sup>3</sup>Note that  $\hat{\Sigma}$  won't be full rank and positive definite as one would hope 

# Practical PCA

In practice, we typically are given a dataset with samples of  $\mathbf{X}$  instead of the distribution or covariance matrix of  $\mathbf{X}$ . Denote the data as  $\mathcal{X}$  with each row is a data point and a total of  $m$  data points. Thus  $\mathcal{X}$  is an  $m$  by  $n$  matrix

- Data are rarely zero-mean to begin with, but we can easily preprocess it by subtracting the mean. That is<sup>2</sup>  $\mathcal{X} \leftarrow \mathcal{X} - \text{ones}(m, 1)\text{mean}(\mathcal{X})$
- Note that  $\hat{\Sigma} \approx \frac{1}{m}\mathcal{X}^T\mathcal{X}$ . We could directly compute the eigenvectors and eigenvalues of  $\hat{\Sigma}$  as discussed previously. But in many cases,  $m < n$  making  $\hat{\Sigma}$  a bad approximate<sup>3</sup>

---

<sup>2</sup>I used the matlab notations for  $\text{ones}(\cdot)$  and  $\text{mean}(\cdot)$  here

<sup>3</sup>Note that  $\hat{\Sigma}$  won't be full rank and positive definite as one would hope 

# Practical PCA

In practice, we typically are given a dataset with samples of  $\mathbf{X}$  instead of the distribution or covariance matrix of  $\mathbf{X}$ . Denote the data as  $\mathcal{X}$  with each row is a data point and a total of  $m$  data points. Thus  $\mathcal{X}$  is an  $m$  by  $n$  matrix

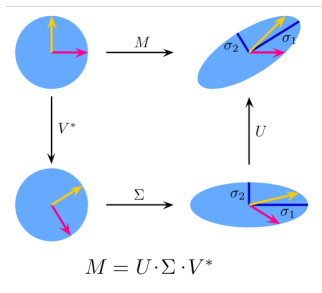
- Data are rarely zero-mean to begin with, but we can easily preprocess it by subtracting the mean. That is<sup>2</sup>  $\mathcal{X} \leftarrow \mathcal{X} - \text{ones}(m, 1)\text{mean}(\mathcal{X})$
- Note that  $\hat{\Sigma} \approx \frac{1}{m}\mathcal{X}^T\mathcal{X}$ . We could directly compute the eigenvectors and eigenvalues of  $\hat{\Sigma}$  as discussed previously. But in many cases,  $m < n$  making  $\hat{\Sigma}$  a bad approximate<sup>3</sup>
  - A more common approach is to decompose  $\mathcal{X}$  with singular value decomposition (SVD) instead

---

<sup>2</sup>I used the matlab notations for  $\text{ones}(\cdot)$  and  $\text{mean}(\cdot)$  here

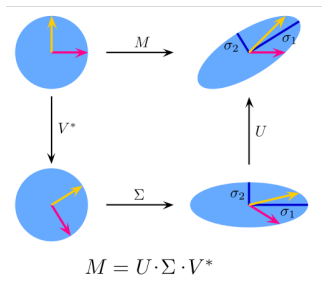
<sup>3</sup>Note that  $\hat{\Sigma}$  won't be full rank and positive definite as one would hope 

# Singular value decomposition (SVD)



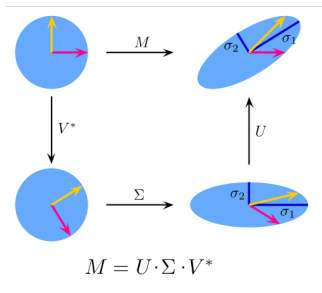
- Every matrix  $M$  can be decomposed as  $M = UDV^\dagger$ , where  $D$  is diagonal and  $U, V$  are unitary. The diagonal terms in  $\Sigma$  are known to be the **singular values**

# Singular value decomposition (SVD)



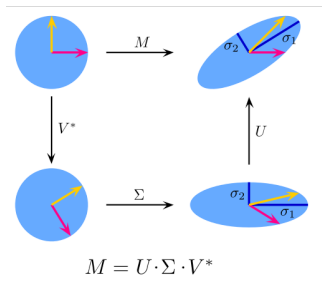
- Every matrix  $M$  can be decomposed as  $M = UDV^\dagger$ , where  $D$  is diagonal and  $U, V$  are unitary. The diagonal terms in  $\Sigma$  are known to be the **singular values**
- For real matrix  $M$ , we can write  $M = UDV^T$  instead.  $U, V$  are now “real unitary” or orthogonal

# Singular value decomposition (SVD)



- Every matrix  $M$  can be decomposed as  $M = UDV^\dagger$ , where  $D$  is diagonal and  $U, V$  are unitary. The diagonal terms in  $\Sigma$  are known to be the **singular values**
- For real matrix  $M$ , we can write  $M = UDV^T$  instead.  $U, V$  are now “real unitary” or orthogonal
  - Note that  $M^T M = VD^T U^T U D V^T = VD^2 V^T$ . Therefore,  $V$  are really eigenvectors of  $M^T M$  with eigenvalues equal to the square of the singular values

# Singular value decomposition (SVD)



- Every matrix  $M$  can be decomposed as  $M = UDV^\dagger$ , where  $D$  is diagonal and  $U, V$  are unitary. The diagonal terms in  $\Sigma$  are known to be the **singular values**
- For real matrix  $M$ , we can write  $M = UDV^T$  instead.  $U, V$  are now “real unitary” or orthogonal
  - Note that  $M^T M = VD^T U^T U D V^T = VD^2 V^T$ . Therefore,  $V$  are really eigenvectors of  $M^T M$  with eigenvalues equal to the square of the singular values
  - Similar, we have  $MM^T = UD^2 U^T$

# PCA with SVD

So from previous slides, instead of first estimating the covariance matrix and then diagonalize it. We should directly decompose the data  $\mathcal{X}$  with SVD instead. The process is summarized below



# PCA with SVD

So from previous slides, instead of first estimating the covariance matrix and then diagonalize it. We should directly decompose the data  $\mathcal{X}$  with SVD instead. The process is summarized below

- Estimate mean from data and subtract mean from that

# PCA with SVD

So from previous slides, instead of first estimating the covariance matrix and then diagonalize it. We should directly decompose the data  $\mathcal{X}$  with SVD instead. The process is summarized below

- Estimate mean from data and subtract mean from that
- Decomposed the mean subtracted data with SVD. We get

$$\mathcal{X} = UDV^T$$

# PCA with SVD

So from previous slides, instead of first estimating the covariance matrix and then diagonalize it. We should directly decompose the data  $\mathcal{X}$  with SVD instead. The process is summarized below

- Estimate mean from data and subtract mean from that
- Decomposed the mean subtracted data with SVD. We get  $\mathcal{X} = UDV^T$
- Note that column of  $V$  are now the principal components, and we can transform a data column as  $V^T x$ . The entire data set can be transformed as  $\mathcal{Y} = \mathcal{X}V$

# PCA with SVD

So from previous slides, instead of first estimating the covariance matrix and then diagonalize it. We should directly decompose the data  $\mathcal{X}$  with SVD instead. The process is summarized below

- Estimate mean from data and subtract mean from that
- Decomposed the mean subtracted data with SVD. We get  $\mathcal{X} = UDV^T$
- Note that column of  $V$  are now the principal components, and we can transform a data column as  $V^T x$ . The entire data set can be transformed as  $\mathcal{Y} = \mathcal{X}V$ 
  - The first few columns of  $\mathcal{Y}$  will contain most “information” regarding the original  $\mathcal{X}$

# PCA with SVD

So from previous slides, instead of first estimating the covariance matrix and then diagonalize it. We should directly decompose the data  $\mathcal{X}$  with SVD instead. The process is summarized below

- Estimate mean from data and subtract mean from that
- Decomposed the mean subtracted data with SVD. We get  $\mathcal{X} = UDV^T$
- Note that column of  $V$  are now the principal components, and we can transform a data column as  $V^T x$ . The entire data set can be transformed as  $\mathcal{Y} = \mathcal{X}V$ 
  - The first few columns of  $\mathcal{Y}$  will contain most “information” regarding the original  $\mathcal{X}$
  - For example, they can be taken as features for recognition or one can omit other columns besides the first few for “compression” as discussed earlier

# Marginalization of normal distribution

- Consider  $\mathbf{Z} \sim \mathcal{N}(\boldsymbol{\mu}_{\mathbf{Z}}, \boldsymbol{\Sigma}_{\mathbf{Z}})$  and let say  $\mathbf{X}$  is a segment of  $\mathbf{Z}$ . That is,  $\mathbf{Z} = \begin{pmatrix} \mathbf{X} \\ \mathbf{Y} \end{pmatrix}$  for some  $\mathbf{Y}$ . Then how should  $\mathbf{X}$  behave?

# Marginalization of normal distribution

- Consider  $\mathbf{Z} \sim \mathcal{N}(\boldsymbol{\mu}_{\mathbf{Z}}, \boldsymbol{\Sigma}_{\mathbf{Z}})$  and let say  $\mathbf{X}$  is a segment of  $\mathbf{Z}$ . That is,  $\mathbf{Z} = \begin{pmatrix} \mathbf{X} \\ \mathbf{Y} \end{pmatrix}$  for some  $\mathbf{Y}$ . Then how should  $\mathbf{X}$  behave?
- We can find the pdf of  $\mathbf{X}$  by just marginalizing that of  $\mathbf{Z}$ . That is

$$\begin{aligned} p(\mathbf{x}) &= \int p(\mathbf{x}, \mathbf{y}) d\mathbf{y} \\ &= \frac{1}{\sqrt{\det(2\pi\boldsymbol{\Sigma})}} \int \exp\left(-\frac{1}{2} \begin{pmatrix} \mathbf{x} - \boldsymbol{\mu}_{\mathbf{X}} \\ \mathbf{y} - \boldsymbol{\mu}_{\mathbf{Y}} \end{pmatrix}^T \boldsymbol{\Sigma}^{-1} \begin{pmatrix} \mathbf{x} - \boldsymbol{\mu}_{\mathbf{X}} \\ \mathbf{y} - \boldsymbol{\mu}_{\mathbf{Y}} \end{pmatrix}\right) d\mathbf{y} \end{aligned}$$

# Marginalization of normal distribution

- Denote  $\Sigma^{-1}$  as  $\Lambda$  (also known as the precision matrix). And partition both  $\Sigma$  and  $\Lambda$  into  $\Sigma = \begin{pmatrix} \Sigma_{XX} & \Sigma_{XY} \\ \Sigma_{YX} & \Sigma_{YY} \end{pmatrix}$  and  $\Lambda = \begin{pmatrix} \Lambda_{XX} & \Lambda_{XY} \\ \Lambda_{YX} & \Lambda_{YY} \end{pmatrix}$



# Marginalization of normal distribution

- Denote  $\Sigma^{-1}$  as  $\Lambda$  (also known as the precision matrix). And partition both  $\Sigma$  and  $\Lambda$  into  $\Sigma = \begin{pmatrix} \Sigma_{\mathbf{X}\mathbf{X}} & \Sigma_{\mathbf{X}\mathbf{Y}} \\ \Sigma_{\mathbf{Y}\mathbf{X}} & \Sigma_{\mathbf{Y}\mathbf{Y}} \end{pmatrix}$  and  $\Lambda = \begin{pmatrix} \Lambda_{\mathbf{X}\mathbf{X}} & \Lambda_{\mathbf{X}\mathbf{Y}} \\ \Lambda_{\mathbf{Y}\mathbf{X}} & \Lambda_{\mathbf{Y}\mathbf{Y}} \end{pmatrix}$
- Then we have

$$\begin{aligned}
 p(\mathbf{x}) &= \frac{1}{\sqrt{\det(2\pi\Sigma)}} \int \exp \left( -\frac{1}{2} \left[ (\mathbf{x} - \boldsymbol{\mu}_{\mathbf{X}})^T \Lambda_{\mathbf{X}\mathbf{X}} (\mathbf{x} - \boldsymbol{\mu}_{\mathbf{X}}) \right. \right. \\
 &\quad \left. \left. + (\mathbf{y} - \boldsymbol{\mu}_{\mathbf{Y}})^T \Lambda_{\mathbf{Y}\mathbf{X}} (\mathbf{x} - \boldsymbol{\mu}_{\mathbf{X}}) + (\mathbf{x} - \boldsymbol{\mu}_{\mathbf{X}})^T \Lambda_{\mathbf{X}\mathbf{Y}} (\mathbf{y} - \boldsymbol{\mu}_{\mathbf{Y}}) \right. \right. \\
 &\quad \left. \left. + (\mathbf{y} - \boldsymbol{\mu}_{\mathbf{Y}})^T \Lambda_{\mathbf{Y}\mathbf{Y}} (\mathbf{y} - \boldsymbol{\mu}_{\mathbf{Y}}) \right] \right) d\mathbf{y} \\
 &= \frac{e^{-\frac{(\mathbf{x} - \boldsymbol{\mu}_{\mathbf{X}})^T \Lambda_{\mathbf{X}\mathbf{X}} (\mathbf{x} - \boldsymbol{\mu}_{\mathbf{X}})}{2}}}{\sqrt{\det(2\pi\Sigma)}} \int \exp \left( -\frac{1}{2} \left[ (\mathbf{y} - \boldsymbol{\mu}_{\mathbf{Y}})^T \Lambda_{\mathbf{Y}\mathbf{X}} (\mathbf{x} - \boldsymbol{\mu}_{\mathbf{X}}) \right. \right. \\
 &\quad \left. \left. + (\mathbf{x} - \boldsymbol{\mu}_{\mathbf{X}})^T \Lambda_{\mathbf{X}\mathbf{Y}} (\mathbf{y} - \boldsymbol{\mu}_{\mathbf{Y}}) + (\mathbf{y} - \boldsymbol{\mu}_{\mathbf{Y}})^T \Lambda_{\mathbf{Y}\mathbf{Y}} (\mathbf{y} - \boldsymbol{\mu}_{\mathbf{Y}}) \right] \right) d\mathbf{y}
 \end{aligned}$$

# Marginalization of normal distribution

To proceed, let's apply the completing square trick on

$$(\mathbf{y} - \boldsymbol{\mu}_Y)^T \boldsymbol{\Lambda}_{YX} (\mathbf{x} - \boldsymbol{\mu}_X) + (\mathbf{x} - \boldsymbol{\mu}_X)^T \boldsymbol{\Lambda}_{XY} (\mathbf{y} - \boldsymbol{\mu}_Y) + (\mathbf{y} - \boldsymbol{\mu}_Y)^T \boldsymbol{\Lambda}_{YY} (\mathbf{y} - \boldsymbol{\mu}_Y).$$

For the ease of exposition, let us denote  $\tilde{\mathbf{x}}$  as  $\mathbf{x} - \boldsymbol{\mu}_X$  and  $\tilde{\mathbf{y}}$  as  $\mathbf{y} - \boldsymbol{\mu}_Y$ . We have

# Marginalization of normal distribution

To proceed, let's apply the completing square trick on

$$(\mathbf{y} - \boldsymbol{\mu}_Y)^T \Lambda_{YX} (\mathbf{x} - \boldsymbol{\mu}_X) + (\mathbf{x} - \boldsymbol{\mu}_X)^T \Lambda_{XY} (\mathbf{y} - \boldsymbol{\mu}_Y) + (\mathbf{y} - \boldsymbol{\mu}_Y)^T \Lambda_{YY} (\mathbf{y} - \boldsymbol{\mu}_Y).$$

For the ease of exposition, let us denote  $\tilde{\mathbf{x}}$  as  $\mathbf{x} - \boldsymbol{\mu}_X$  and  $\tilde{\mathbf{y}}$  as  $\mathbf{y} - \boldsymbol{\mu}_Y$ . We have

$$\begin{aligned} & \tilde{\mathbf{y}}^T \Lambda_{YX} \tilde{\mathbf{x}} + \tilde{\mathbf{x}}^T \Lambda_{XY} \tilde{\mathbf{y}} + \tilde{\mathbf{y}}^T \Lambda_{YY} \tilde{\mathbf{y}} \\ &= (\tilde{\mathbf{y}} + \Lambda_{YY}^{-1} \Lambda_{YX} \tilde{\mathbf{x}})^T \Lambda_{YY} (\tilde{\mathbf{y}} + \Lambda_{YY}^{-1} \Lambda_{YX} \tilde{\mathbf{x}}) - \tilde{\mathbf{x}}^T \Lambda_{XY} \Lambda_{YY}^{-1} \Lambda_{YX} \tilde{\mathbf{x}}, \end{aligned}$$

where we use the fact that  $\Lambda = \Sigma^{-1}$  is symmetric and so  $\Lambda_{XY} = \Lambda_{YX}$

# Marginalization of normal distribution

$$p(\mathbf{x}) = \frac{e^{-\frac{\bar{\mathbf{x}}^T (\Lambda_{\mathbf{XX}} - \Lambda_{\mathbf{XY}} \Lambda_{\mathbf{YY}}^{-1} \Lambda_{\mathbf{YX}}) \bar{\mathbf{x}}}{2}}}{\sqrt{\det(2\pi \Sigma)}} \int e^{-\frac{(\bar{\mathbf{y}} + \Lambda_{\mathbf{YY}}^{-1} \Lambda_{\mathbf{YX}} \bar{\mathbf{x}})^T \Lambda_{\mathbf{YY}} (\bar{\mathbf{y}} + \Lambda_{\mathbf{YY}}^{-1} \Lambda_{\mathbf{YX}} \bar{\mathbf{x}})}{2}} d\mathbf{y}$$

# Marginalization of normal distribution

$$\begin{aligned}
 p(\mathbf{x}) &= \frac{e^{-\frac{\tilde{\mathbf{x}}^T (\Lambda_{\mathbf{XX}} - \Lambda_{\mathbf{XY}} \Lambda_{\mathbf{YY}}^{-1} \Lambda_{\mathbf{YX}}) \tilde{\mathbf{x}}}{2}}}{\sqrt{\det(2\pi \Sigma)}} \int e^{-\frac{(\tilde{\mathbf{y}} + \Lambda_{\mathbf{YY}}^{-1} \Lambda_{\mathbf{YX}} \tilde{\mathbf{x}})^T \Lambda_{\mathbf{YY}} (\tilde{\mathbf{y}} + \Lambda_{\mathbf{YY}}^{-1} \Lambda_{\mathbf{YX}} \tilde{\mathbf{x}})}{2}} d\mathbf{y} \\
 &= \frac{\sqrt{\det(2\pi \Lambda_{\mathbf{YY}}^{-1})}}{\sqrt{\det(2\pi \Sigma)}} \exp\left(-\frac{\tilde{\mathbf{x}}^T (\Lambda_{\mathbf{XX}} - \Lambda_{\mathbf{XY}} \Lambda_{\mathbf{YY}}^{-1} \Lambda_{\mathbf{YX}}) \tilde{\mathbf{x}}}{2}\right)
 \end{aligned}$$

# Marginalization of normal distribution

$$\begin{aligned}
 p(\mathbf{x}) &= \frac{e^{-\frac{\tilde{\mathbf{x}}^T (\Lambda_{\mathbf{XX}} - \Lambda_{\mathbf{XY}} \Lambda_{\mathbf{YY}}^{-1} \Lambda_{\mathbf{YX}}) \tilde{\mathbf{x}}}{2}}}{\sqrt{\det(2\pi \Sigma)}} \int e^{-\frac{(\tilde{\mathbf{y}} + \Lambda_{\mathbf{YY}}^{-1} \Lambda_{\mathbf{YX}} \tilde{\mathbf{x}})^T \Lambda_{\mathbf{YY}} (\tilde{\mathbf{y}} + \Lambda_{\mathbf{YY}}^{-1} \Lambda_{\mathbf{YX}} \tilde{\mathbf{x}})}{2}} d\mathbf{y} \\
 &= \frac{\sqrt{\det(2\pi \Lambda_{\mathbf{YY}}^{-1})}}{\sqrt{\det(2\pi \Sigma)}} \exp\left(-\frac{\tilde{\mathbf{x}}^T (\Lambda_{\mathbf{XX}} - \Lambda_{\mathbf{XY}} \Lambda_{\mathbf{YY}}^{-1} \Lambda_{\mathbf{YX}}) \tilde{\mathbf{x}}}{2}\right) \\
 &\stackrel{(a)}{=} \frac{\sqrt{\det(2\pi \Lambda_{\mathbf{YY}}^{-1})}}{\sqrt{\det(2\pi \Sigma)}} \exp\left(-\frac{\tilde{\mathbf{x}}^T \Sigma_{\mathbf{XX}}^{-1} \tilde{\mathbf{x}}}{2}\right)
 \end{aligned}$$

# Marginalization of normal distribution

$$\begin{aligned}
 p(\mathbf{x}) &= \frac{e^{-\frac{\tilde{\mathbf{x}}^T (\Lambda_{\mathbf{XX}} - \Lambda_{\mathbf{XY}} \Lambda_{\mathbf{YY}}^{-1} \Lambda_{\mathbf{YX}}) \tilde{\mathbf{x}}}{2}}}{\sqrt{\det(2\pi \Sigma)}} \int e^{-\frac{(\tilde{\mathbf{y}} + \Lambda_{\mathbf{YY}}^{-1} \Lambda_{\mathbf{YX}} \tilde{\mathbf{x}})^T \Lambda_{\mathbf{YY}} (\tilde{\mathbf{y}} + \Lambda_{\mathbf{YY}}^{-1} \Lambda_{\mathbf{YX}} \tilde{\mathbf{x}})}{2}} d\mathbf{y} \\
 &= \frac{\sqrt{\det(2\pi \Lambda_{\mathbf{YY}}^{-1})}}{\sqrt{\det(2\pi \Sigma)}} \exp\left(-\frac{\tilde{\mathbf{x}}^T (\Lambda_{\mathbf{XX}} - \Lambda_{\mathbf{XY}} \Lambda_{\mathbf{YY}}^{-1} \Lambda_{\mathbf{YX}}) \tilde{\mathbf{x}}}{2}\right) \\
 &\stackrel{(a)}{=} \frac{\sqrt{\det(2\pi \Lambda_{\mathbf{YY}}^{-1})}}{\sqrt{\det(2\pi \Sigma)}} \exp\left(-\frac{\tilde{\mathbf{x}}^T \Sigma_{\mathbf{XX}}^{-1} \tilde{\mathbf{x}}}{2}\right) \\
 &\stackrel{(b)}{=} \frac{1}{\sqrt{\det(2\pi \Sigma_{\mathbf{XX}})}} \exp\left(-\frac{\tilde{\mathbf{x}}^T \Sigma_{\mathbf{XX}}^{-1} \tilde{\mathbf{x}}}{2}\right)
 \end{aligned}$$

# Marginalization of normal distribution

$$\begin{aligned}
 p(\mathbf{x}) &= \frac{e^{-\frac{\tilde{\mathbf{x}}^T (\Lambda_{\mathbf{XX}} - \Lambda_{\mathbf{XY}} \Lambda_{\mathbf{YY}}^{-1} \Lambda_{\mathbf{YX}}) \tilde{\mathbf{x}}}{2}}}{\sqrt{\det(2\pi \Sigma)}} \int e^{-\frac{(\tilde{\mathbf{y}} + \Lambda_{\mathbf{YY}}^{-1} \Lambda_{\mathbf{YX}} \tilde{\mathbf{x}})^T \Lambda_{\mathbf{YY}} (\tilde{\mathbf{y}} + \Lambda_{\mathbf{YY}}^{-1} \Lambda_{\mathbf{YX}} \tilde{\mathbf{x}})}{2}} d\mathbf{y} \\
 &= \frac{\sqrt{\det(2\pi \Lambda_{\mathbf{YY}}^{-1})}}{\sqrt{\det(2\pi \Sigma)}} \exp\left(-\frac{\tilde{\mathbf{x}}^T (\Lambda_{\mathbf{XX}} - \Lambda_{\mathbf{XY}} \Lambda_{\mathbf{YY}}^{-1} \Lambda_{\mathbf{YX}}) \tilde{\mathbf{x}}}{2}\right) \\
 &\stackrel{(a)}{=} \frac{\sqrt{\det(2\pi \Lambda_{\mathbf{YY}}^{-1})}}{\sqrt{\det(2\pi \Sigma)}} \exp\left(-\frac{\tilde{\mathbf{x}}^T \Sigma_{\mathbf{XX}}^{-1} \tilde{\mathbf{x}}}{2}\right) \\
 &\stackrel{(b)}{=} \frac{1}{\sqrt{\det(2\pi \Sigma_{\mathbf{XX}})}} \exp\left(-\frac{\tilde{\mathbf{x}}^T \Sigma_{\mathbf{XX}}^{-1} \tilde{\mathbf{x}}}{2}\right) \\
 &= \frac{1}{\sqrt{\det(2\pi \Sigma_{\mathbf{XX}})}} \exp\left(-\frac{(\mathbf{x} - \boldsymbol{\mu}_{\mathbf{X}})^T \Sigma_{\mathbf{XX}}^{-1} (\mathbf{x} - \boldsymbol{\mu}_{\mathbf{X}})}{2}\right),
 \end{aligned}$$

where (a) and (b) will be shown next



$$(a) \Sigma_{XX}^{-1} = \Lambda_{XX} - \Lambda_{XY} \Lambda_{YY}^{-1} \Lambda_{YX}$$

Proof.

Since  $\Lambda = \Sigma^{-1}$ , we have  $\Sigma_{XX} \Lambda_{XY} + \Sigma_{XY} \Lambda_{YY} = 0$  and  $\Sigma_{XX} \Lambda_{XX} + \Sigma_{XY} \Lambda_{YX} = I$ . Insert an identity into the latter equation, we have  $\Sigma_{XX} \Lambda_{XX} + \Sigma_{XY} (\Lambda_{YY} \Lambda_{YY}^{-1}) \Lambda_{YX} = \Sigma_{XX} \Lambda_{XX} - (\Sigma_{XX} \Lambda_{XY}) \Lambda_{YY}^{-1} \Lambda_{YX} = \Sigma_{XX} (\Lambda_{XX} - \Lambda_{XY} \Lambda_{YY}^{-1} \Lambda_{YX}) = I$ . □

Remark

*By symmetry, we also have*

$$\Lambda_{XX}^{-1} = \Sigma_{XX} - \Sigma_{XY} \Sigma_{YY}^{-1} \Sigma_{YX}$$

$$(b') \det(\Sigma) = \det(\Sigma_{YY}) \det(\Lambda_{XX}^{-1})$$

Proof.

$$\det(\Sigma) = \det \begin{pmatrix} \Sigma_{XX} & \Sigma_{XY} \\ \Sigma_{YX} & \Sigma_{YY} \end{pmatrix}$$



$$(b') \det(\Sigma) = \det(\Sigma_{YY}) \det(\Lambda_{XX}^{-1})$$

Proof.

$$\begin{aligned} \det(\Sigma) &= \det \begin{pmatrix} \Sigma_{XX} & \Sigma_{XY} \\ \Sigma_{YX} & \Sigma_{YY} \end{pmatrix} \\ &= \det \left( \begin{pmatrix} I & 0 \\ 0 & \Sigma_{YY} \end{pmatrix} \begin{pmatrix} \Sigma_{XX} & \Sigma_{XY} \\ \Sigma_{YY}^{-1} \Sigma_{YX} & I \end{pmatrix} \right) \end{aligned}$$



$$(b') \det(\Sigma) = \det(\Sigma_{YY}) \det(\Lambda_{XX}^{-1})$$

Proof.

$$\begin{aligned} \det(\Sigma) &= \det \begin{pmatrix} \Sigma_{XX} & \Sigma_{XY} \\ \Sigma_{YX} & \Sigma_{YY} \end{pmatrix} \\ &= \det \left( \begin{pmatrix} I & 0 \\ 0 & \Sigma_{YY} \end{pmatrix} \begin{pmatrix} \Sigma_{XX} & \Sigma_{XY} \\ \Sigma_{YY}^{-1} \Sigma_{YX} & I \end{pmatrix} \right) \\ &= \det \left( \begin{pmatrix} I & 0 \\ 0 & \Sigma_{YY} \end{pmatrix} \begin{pmatrix} I & \Sigma_{XY} \\ 0 & I \end{pmatrix} \begin{pmatrix} \Sigma_{XX} - \Sigma_{XY} \Sigma_{YY}^{-1} \Sigma_{YX} & 0 \\ \Sigma_{YY}^{-1} \Sigma_{YX} & I \end{pmatrix} \right) \end{aligned}$$



$$(b') \det(\Sigma) = \det(\Sigma_{YY}) \det(\Lambda_{XX}^{-1})$$

Proof.

$$\begin{aligned} \det(\Sigma) &= \det \begin{pmatrix} \Sigma_{XX} & \Sigma_{XY} \\ \Sigma_{YX} & \Sigma_{YY} \end{pmatrix} \\ &= \det \left( \begin{pmatrix} I & 0 \\ 0 & \Sigma_{YY} \end{pmatrix} \begin{pmatrix} \Sigma_{XX} & \Sigma_{XY} \\ \Sigma_{YY}^{-1} \Sigma_{YX} & I \end{pmatrix} \right) \\ &= \det \left( \begin{pmatrix} I & 0 \\ 0 & \Sigma_{YY} \end{pmatrix} \begin{pmatrix} I & \Sigma_{XY} \\ 0 & I \end{pmatrix} \begin{pmatrix} \Sigma_{XX} - \Sigma_{XY} \Sigma_{YY}^{-1} \Sigma_{YX} & 0 \\ \Sigma_{YY}^{-1} \Sigma_{YX} & I \end{pmatrix} \right) \\ &= \det \begin{pmatrix} I & 0 \\ 0 & \Sigma_{YY} \end{pmatrix} \det \begin{pmatrix} I & \Sigma_{XY} \\ 0 & I \end{pmatrix} \det \begin{pmatrix} \Sigma_{XX} - \Sigma_{XY} \Sigma_{YY}^{-1} \Sigma_{YX} & 0 \\ \Sigma_{YY}^{-1} \Sigma_{YX} & I \end{pmatrix} \end{aligned}$$



$$(b') \det(\Sigma) = \det(\Sigma_{YY}) \det(\Lambda_{XX}^{-1})$$

Proof.

$$\begin{aligned} \det(\Sigma) &= \det \begin{pmatrix} \Sigma_{XX} & \Sigma_{XY} \\ \Sigma_{YX} & \Sigma_{YY} \end{pmatrix} \\ &= \det \left( \begin{pmatrix} I & 0 \\ 0 & \Sigma_{YY} \end{pmatrix} \begin{pmatrix} \Sigma_{XX} & \Sigma_{XY} \\ \Sigma_{YY}^{-1} \Sigma_{YX} & I \end{pmatrix} \right) \\ &= \det \left( \begin{pmatrix} I & 0 \\ 0 & \Sigma_{YY} \end{pmatrix} \begin{pmatrix} I & \Sigma_{XY} \\ 0 & I \end{pmatrix} \begin{pmatrix} \Sigma_{XX} - \Sigma_{XY} \Sigma_{YY}^{-1} \Sigma_{YX} & 0 \\ \Sigma_{YY}^{-1} \Sigma_{YX} & I \end{pmatrix} \right) \\ &= \det \begin{pmatrix} I & 0 \\ 0 & \Sigma_{YY} \end{pmatrix} \det \begin{pmatrix} I & \Sigma_{XY} \\ 0 & I \end{pmatrix} \det \begin{pmatrix} \Sigma_{XX} - \Sigma_{XY} \Sigma_{YY}^{-1} \Sigma_{YX} & 0 \\ \Sigma_{YY}^{-1} \Sigma_{YX} & I \end{pmatrix} \\ &= \det \Sigma_{YY} \det(\Sigma_{XX} - \Sigma_{XY} \Sigma_{YY}^{-1} \Sigma_{YX}) \end{aligned}$$



$$(b') \det(\Sigma) = \det(\Sigma_{YY}) \det(\Lambda_{XX}^{-1})$$

Proof.

$$\begin{aligned} \det(\Sigma) &= \det \begin{pmatrix} \Sigma_{XX} & \Sigma_{XY} \\ \Sigma_{YX} & \Sigma_{YY} \end{pmatrix} \\ &= \det \left( \begin{pmatrix} I & 0 \\ 0 & \Sigma_{YY} \end{pmatrix} \begin{pmatrix} \Sigma_{XX} & \Sigma_{XY} \\ \Sigma_{YY}^{-1} \Sigma_{YX} & I \end{pmatrix} \right) \\ &= \det \left( \begin{pmatrix} I & 0 \\ 0 & \Sigma_{YY} \end{pmatrix} \begin{pmatrix} I & \Sigma_{XY} \\ 0 & I \end{pmatrix} \begin{pmatrix} \Sigma_{XX} - \Sigma_{XY} \Sigma_{YY}^{-1} \Sigma_{YX} & 0 \\ \Sigma_{YY}^{-1} \Sigma_{YX} & I \end{pmatrix} \right) \\ &= \det \begin{pmatrix} I & 0 \\ 0 & \Sigma_{YY} \end{pmatrix} \det \begin{pmatrix} I & \Sigma_{XY} \\ 0 & I \end{pmatrix} \det \begin{pmatrix} \Sigma_{XX} - \Sigma_{XY} \Sigma_{YY}^{-1} \Sigma_{YX} & 0 \\ \Sigma_{YY}^{-1} \Sigma_{YX} & I \end{pmatrix} \\ &= \det \Sigma_{YY} \det(\Sigma_{XX} - \Sigma_{XY} \Sigma_{YY}^{-1} \Sigma_{YX}) \\ &= \det \Sigma_{YY} \det \Lambda_{XX}^{-1}, \end{aligned}$$

where the last equality is from (a) □

(b)  $\det(a\Sigma) = \det(a\Sigma_{\mathbf{Y}\mathbf{Y}}) \det(a\Lambda_{\mathbf{X}\mathbf{X}}^{-1})$  for any constant  $a$

Proof.

Note that since the width (height) of  $\Sigma$  is equal to the sum of the widths of  $\Sigma_{\mathbf{X}\mathbf{X}}$  and  $\Sigma_{\mathbf{Y}\mathbf{Y}}$ . The equation below follows immediately  $\square$

Remark

*Note that by symmetry, we also have  $\det(a\Sigma) = \det(a\Sigma_{\mathbf{X}\mathbf{X}}) \det(a\Lambda_{\mathbf{Y}\mathbf{Y}}^{-1})$  for any constant  $a$ . Take  $a = 2\pi$  and that is exactly what we need for (b)*



## Review

- ML:  $\hat{x} = \arg \max_x p(x|\hat{\theta})$ ,  $\hat{\theta} = \arg \max_{\theta} p(o|\theta)$
- MAP:  $\hat{x} = \arg \max_x p(x|\hat{\theta})$ ,  $\hat{\theta} = \arg \max_{\theta} p(\theta|o)$
- Bayesian:  $\hat{x} = \sum_{\theta} p(\theta|o) \sum_x xp(x|\theta)$
- For zero-mean  $\mathbf{X}$ ,  $\Sigma_{\mathbf{X}} = E[\mathbf{X}\mathbf{X}^T]$  and say we have  $P^T \Sigma_{\mathbf{X}} P = D$ . The transformed  $\mathbf{Y} = P^T \mathbf{X}$  are independent to each other
  - Note that the transform is just **principal component analysis**
- Marginalization of a normal distribution is still a normal distribution
- (a)  $\Sigma_{\mathbf{X}\mathbf{X}}^{-1} = \Lambda_{\mathbf{X}\mathbf{X}} - \Lambda_{\mathbf{X}\mathbf{Y}} \Lambda_{\mathbf{Y}\mathbf{Y}}^{-1} \Lambda_{\mathbf{Y}\mathbf{X}}$
- (b)  $\det(a\Sigma) = \det(a\Sigma_{\mathbf{Y}\mathbf{Y}}) \det(a\Lambda_{\mathbf{X}\mathbf{X}}^{-1})$  for any constant  $a$

# Principal component analysis (PCA)

- Recall that  $\Sigma = E[\mathbf{X}\mathbf{X}^T]$  (assume  $\mathbf{X}$  is zero-mean) and  $\mathbf{Y} = P^T\mathbf{X}$  with  $E[\mathbf{Y}\mathbf{Y}^T] = P^T\Sigma P = D$
- Assume that the diagonal of  $D$  (note that those are eigenvalues) are arranged in descending order that  $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n$

---

$${}^4 \text{tr}(AB) = \sum_i \sum_j a_{i,j} b_{j,i} = \sum_j \sum_i b_{j,i} a_{i,j} = \text{tr}(BA)$$

# Principal component analysis (PCA)

- Recall that  $\Sigma = E[\mathbf{X}\mathbf{X}^T]$  (assume  $\mathbf{X}$  is zero-mean) and  $\mathbf{Y} = P^T\mathbf{X}$  with  $E[\mathbf{Y}\mathbf{Y}^T] = P^T\Sigma P = D$
- Assume that the diagonal of  $D$  (note that those are eigenvalues) are arranged in descending order that  $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n$ 
  - Generate an approximate  $\hat{\mathbf{Y}}$  of  $\mathbf{Y}$  by setting all components except first  $k$  as 0

---

$${}^4 \text{tr}(AB) = \sum_i \sum_j a_{i,j} b_{j,i} = \sum_j \sum_i b_{j,i} a_{i,j} = \text{tr}(BA)$$

# Principal component analysis (PCA)

- Recall that  $\Sigma = E[\mathbf{X}\mathbf{X}^T]$  (assume  $\mathbf{X}$  is zero-mean) and  $\mathbf{Y} = P^T\mathbf{X}$  with  $E[\mathbf{Y}\mathbf{Y}^T] = P^T\Sigma P = D$
- Assume that the diagonal of  $D$  (note that those are eigenvalues) are arranged in descending order that  $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n$ 
  - Generate an approximate  $\hat{\mathbf{Y}}$  of  $\mathbf{Y}$  by setting all components except first  $k$  as 0
  - The mean square error (mse) of<sup>4</sup>  $\hat{\mathbf{Y}} = E[(\mathbf{Y} - \hat{\mathbf{Y}})^T(\mathbf{Y} - \hat{\mathbf{Y}})]$

---

<sup>4</sup> $tr(AB) = \sum_i \sum_j a_{i,j} b_{j,i} = \sum_j \sum_i b_{j,i} a_{i,j} = tr(BA)$

# Principal component analysis (PCA)

- Recall that  $\Sigma = E[\mathbf{X}\mathbf{X}^T]$  (assume  $\mathbf{X}$  is zero-mean) and  $\mathbf{Y} = P^T\mathbf{X}$  with  $E[\mathbf{Y}\mathbf{Y}^T] = P^T\Sigma P = D$
- Assume that the diagonal of  $D$  (note that those are eigenvalues) are arranged in descending order that  $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n$ 
  - Generate an approximate  $\hat{\mathbf{Y}}$  of  $\mathbf{Y}$  by setting all components except first  $k$  as 0
  - The mean square error (mse) of<sup>4</sup>  $\hat{\mathbf{Y}} = E[(\mathbf{Y} - \hat{\mathbf{Y}})^T(\mathbf{Y} - \hat{\mathbf{Y}})]$   
 $= \text{tr}(E[(\mathbf{Y} - \hat{\mathbf{Y}})^T(\mathbf{Y} - \hat{\mathbf{Y}})])$

---

<sup>4</sup> $\text{tr}(AB) = \sum_i \sum_j a_{i,j} b_{j,i} = \sum_j \sum_i b_{j,i} a_{i,j} = \text{tr}(BA)$

# Principal component analysis (PCA)

- Recall that  $\Sigma = E[\mathbf{X}\mathbf{X}^T]$  (assume  $\mathbf{X}$  is zero-mean) and  $\mathbf{Y} = P^T\mathbf{X}$  with  $E[\mathbf{Y}\mathbf{Y}^T] = P^T\Sigma P = D$
- Assume that the diagonal of  $D$  (note that those are eigenvalues) are arranged in descending order that  $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n$ 
  - Generate an approximate  $\hat{\mathbf{Y}}$  of  $\mathbf{Y}$  by setting all components except first  $k$  as 0
  - The mean square error (mse) of<sup>4</sup>  $\hat{\mathbf{Y}} = E[(\mathbf{Y} - \hat{\mathbf{Y}})^T(\mathbf{Y} - \hat{\mathbf{Y}})]$   
 $= tr(E[(\mathbf{Y} - \hat{\mathbf{Y}})^T(\mathbf{Y} - \hat{\mathbf{Y}})]) = E[tr((\mathbf{Y} - \hat{\mathbf{Y}})^T(\mathbf{Y} - \hat{\mathbf{Y}}))]$

---

<sup>4</sup> $tr(AB) = \sum_i \sum_j a_{i,j} b_{j,i} = \sum_j \sum_i b_{j,i} a_{i,j} = tr(BA)$

# Principal component analysis (PCA)

- Recall that  $\Sigma = E[\mathbf{X}\mathbf{X}^T]$  (assume  $\mathbf{X}$  is zero-mean) and  $\mathbf{Y} = P^T\mathbf{X}$  with  $E[\mathbf{Y}\mathbf{Y}^T] = P^T\Sigma P = D$
- Assume that the diagonal of  $D$  (note that those are eigenvalues) are arranged in descending order that  $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n$ 
  - Generate an approximate  $\hat{\mathbf{Y}}$  of  $\mathbf{Y}$  by setting all components except first  $k$  as 0
  - The mean square error (mse) of<sup>4</sup>  $\hat{\mathbf{Y}} = E[(\mathbf{Y} - \hat{\mathbf{Y}})^T(\mathbf{Y} - \hat{\mathbf{Y}})]$   
 $= \text{tr}(E[(\mathbf{Y} - \hat{\mathbf{Y}})^T(\mathbf{Y} - \hat{\mathbf{Y}})]) = E[\text{tr}((\mathbf{Y} - \hat{\mathbf{Y}})^T(\mathbf{Y} - \hat{\mathbf{Y}}))]$   
 $= E[\text{tr}((\mathbf{Y} - \hat{\mathbf{Y}})(\mathbf{Y} - \hat{\mathbf{Y}})^T)] = \text{tr}(E[(\mathbf{Y} - \hat{\mathbf{Y}})(\mathbf{Y} - \hat{\mathbf{Y}})^T])$

---

<sup>4</sup> $\text{tr}(AB) = \sum_i \sum_j a_{i,j} b_{j,i} = \sum_j \sum_i b_{j,i} a_{i,j} = \text{tr}(BA)$

# Principal component analysis (PCA)

- Recall that  $\Sigma = E[\mathbf{X}\mathbf{X}^T]$  (assume  $\mathbf{X}$  is zero-mean) and  $\mathbf{Y} = P^T\mathbf{X}$  with  $E[\mathbf{Y}\mathbf{Y}^T] = P^T\Sigma P = D$
- Assume that the diagonal of  $D$  (note that those are eigenvalues) are arranged in descending order that  $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n$ 
  - Generate an approximate  $\hat{\mathbf{Y}}$  of  $\mathbf{Y}$  by setting all components except first  $k$  as 0
  - The mean square error (mse) of<sup>4</sup>  $\hat{\mathbf{Y}} = E[(\mathbf{Y} - \hat{\mathbf{Y}})^T(\mathbf{Y} - \hat{\mathbf{Y}})]$   
 $= \text{tr}(E[(\mathbf{Y} - \hat{\mathbf{Y}})^T(\mathbf{Y} - \hat{\mathbf{Y}})]) = E[\text{tr}((\mathbf{Y} - \hat{\mathbf{Y}})^T(\mathbf{Y} - \hat{\mathbf{Y}}))]$   
 $= E[\text{tr}((\mathbf{Y} - \hat{\mathbf{Y}})(\mathbf{Y} - \hat{\mathbf{Y}})^T)] = \text{tr}(E[(\mathbf{Y} - \hat{\mathbf{Y}})(\mathbf{Y} - \hat{\mathbf{Y}})^T]) = \sum_{i=k+1}^n \lambda_i$
  - Similarly, if we “reconstruct”  $\mathbf{X}$  as  $\hat{\mathbf{X}} = P\hat{\mathbf{Y}}$ . The mse of  $\hat{\mathbf{X}} = E[(\mathbf{X} - \hat{\mathbf{X}})^T(\mathbf{X} - \hat{\mathbf{X}})] = \text{tr}(E[(\mathbf{X} - \hat{\mathbf{X}})(\mathbf{X} - \hat{\mathbf{X}})^T])$

<sup>4</sup> $\text{tr}(AB) = \sum_i \sum_j a_{i,j} b_{j,i} = \sum_j \sum_i b_{j,i} a_{i,j} = \text{tr}(BA)$



# Principal component analysis (PCA)

- Recall that  $\Sigma = E[\mathbf{X}\mathbf{X}^T]$  (assume  $\mathbf{X}$  is zero-mean) and  $\mathbf{Y} = P^T\mathbf{X}$  with  $E[\mathbf{Y}\mathbf{Y}^T] = P^T\Sigma P = D$
- Assume that the diagonal of  $D$  (note that those are eigenvalues) are arranged in descending order that  $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n$ 
  - Generate an approximate  $\hat{\mathbf{Y}}$  of  $\mathbf{Y}$  by setting all components except first  $k$  as 0
  - The mean square error (mse) of<sup>4</sup>  $\hat{\mathbf{Y}} = E[(\mathbf{Y} - \hat{\mathbf{Y}})^T(\mathbf{Y} - \hat{\mathbf{Y}})]$   
 $= \text{tr}(E[(\mathbf{Y} - \hat{\mathbf{Y}})^T(\mathbf{Y} - \hat{\mathbf{Y}})]) = E[\text{tr}((\mathbf{Y} - \hat{\mathbf{Y}})^T(\mathbf{Y} - \hat{\mathbf{Y}}))]$   
 $= E[\text{tr}((\mathbf{Y} - \hat{\mathbf{Y}})(\mathbf{Y} - \hat{\mathbf{Y}})^T)] = \text{tr}(E[(\mathbf{Y} - \hat{\mathbf{Y}})(\mathbf{Y} - \hat{\mathbf{Y}})^T]) = \sum_{i=k+1}^n \lambda_i$
  - Similarly, if we “reconstruct”  $\mathbf{X}$  as  $\hat{\mathbf{X}} = P\hat{\mathbf{Y}}$ . The mse of  $\hat{\mathbf{X}} = E[(\mathbf{X} - \hat{\mathbf{X}})^T(\mathbf{X} - \hat{\mathbf{X}})] = \text{tr}(E[(\mathbf{X} - \hat{\mathbf{X}})(\mathbf{X} - \hat{\mathbf{X}})^T]) = \text{tr}(PE[(\mathbf{Y} - \hat{\mathbf{Y}})(\mathbf{Y} - \hat{\mathbf{Y}})^T]P^T)$

<sup>4</sup> $\text{tr}(AB) = \sum_i \sum_j a_{i,j} b_{j,i} = \sum_j \sum_i b_{j,i} a_{i,j} = \text{tr}(BA)$

# Principal component analysis (PCA)

- Recall that  $\Sigma = E[\mathbf{X}\mathbf{X}^T]$  (assume  $\mathbf{X}$  is zero-mean) and  $\mathbf{Y} = P^T\mathbf{X}$  with  $E[\mathbf{Y}\mathbf{Y}^T] = P^T\Sigma P = D$
- Assume that the diagonal of  $D$  (note that those are eigenvalues) are arranged in descending order that  $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n$ 
  - Generate an approximate  $\hat{\mathbf{Y}}$  of  $\mathbf{Y}$  by setting all components except first  $k$  as 0
  - The mean square error (mse) of<sup>4</sup>  $\hat{\mathbf{Y}} = E[(\mathbf{Y} - \hat{\mathbf{Y}})^T(\mathbf{Y} - \hat{\mathbf{Y}})]$   
 $= \text{tr}(E[(\mathbf{Y} - \hat{\mathbf{Y}})^T(\mathbf{Y} - \hat{\mathbf{Y}})]) = E[\text{tr}((\mathbf{Y} - \hat{\mathbf{Y}})^T(\mathbf{Y} - \hat{\mathbf{Y}}))]$   
 $= E[\text{tr}((\mathbf{Y} - \hat{\mathbf{Y}})(\mathbf{Y} - \hat{\mathbf{Y}})^T)] = \text{tr}(E[(\mathbf{Y} - \hat{\mathbf{Y}})(\mathbf{Y} - \hat{\mathbf{Y}})^T]) = \sum_{i=k+1}^n \lambda_i$
  - Similarly, if we “reconstruct”  $\mathbf{X}$  as  $\hat{\mathbf{X}} = P\hat{\mathbf{Y}}$ . The mse of  $\hat{\mathbf{X}} = E[(\mathbf{X} - \hat{\mathbf{X}})^T(\mathbf{X} - \hat{\mathbf{X}})] = \text{tr}(E[(\mathbf{X} - \hat{\mathbf{X}})(\mathbf{X} - \hat{\mathbf{X}})^T]) = \text{tr}(PE[(\mathbf{Y} - \hat{\mathbf{Y}})(\mathbf{Y} - \hat{\mathbf{Y}})^T]P^T) = \text{tr}(P^TPE[(\mathbf{Y} - \hat{\mathbf{Y}})(\mathbf{Y} - \hat{\mathbf{Y}})^T])$

<sup>4</sup> $\text{tr}(AB) = \sum_i \sum_j a_{i,j} b_{j,i} = \sum_j \sum_i b_{j,i} a_{i,j} = \text{tr}(BA)$

# Principal component analysis (PCA)

- Recall that  $\Sigma = E[\mathbf{X}\mathbf{X}^T]$  (assume  $\mathbf{X}$  is zero-mean) and  $\mathbf{Y} = P^T\mathbf{X}$  with  $E[\mathbf{Y}\mathbf{Y}^T] = P^T\Sigma P = D$
- Assume that the diagonal of  $D$  (note that those are eigenvalues) are arranged in descending order that  $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n$ 
  - Generate an approximate  $\hat{\mathbf{Y}}$  of  $\mathbf{Y}$  by setting all components except first  $k$  as 0
  - The mean square error (mse) of<sup>4</sup>  $\hat{\mathbf{Y}} = E[(\mathbf{Y} - \hat{\mathbf{Y}})^T(\mathbf{Y} - \hat{\mathbf{Y}})]$   
 $= \text{tr}(E[(\mathbf{Y} - \hat{\mathbf{Y}})^T(\mathbf{Y} - \hat{\mathbf{Y}})]) = E[\text{tr}((\mathbf{Y} - \hat{\mathbf{Y}})^T(\mathbf{Y} - \hat{\mathbf{Y}}))]$   
 $= E[\text{tr}((\mathbf{Y} - \hat{\mathbf{Y}})(\mathbf{Y} - \hat{\mathbf{Y}})^T)] = \text{tr}(E[(\mathbf{Y} - \hat{\mathbf{Y}})(\mathbf{Y} - \hat{\mathbf{Y}})^T]) = \sum_{i=k+1}^n \lambda_i$
  - Similarly, if we “reconstruct”  $\mathbf{X}$  as  $\hat{\mathbf{X}} = P\hat{\mathbf{Y}}$ . The mse of  $\hat{\mathbf{X}} = E[(\mathbf{X} - \hat{\mathbf{X}})^T(\mathbf{X} - \hat{\mathbf{X}})] = \text{tr}(E[(\mathbf{X} - \hat{\mathbf{X}})(\mathbf{X} - \hat{\mathbf{X}})^T]) = \text{tr}(PE[(\mathbf{Y} - \hat{\mathbf{Y}})(\mathbf{Y} - \hat{\mathbf{Y}})^T]P^T) = \text{tr}(P^TPE[(\mathbf{Y} - \hat{\mathbf{Y}})(\mathbf{Y} - \hat{\mathbf{Y}})^T]) = \sum_{i=k+1}^n \lambda_i$

<sup>4</sup> $\text{tr}(AB) = \sum_i \sum_j a_{i,j} b_{j,i} = \sum_j \sum_i b_{j,i} a_{i,j} = \text{tr}(BA)$

# Principal component analysis (PCA)

- Recall that  $\Sigma = E[\mathbf{X}\mathbf{X}^T]$  (assume  $\mathbf{X}$  is zero-mean) and  $\mathbf{Y} = P^T\mathbf{X}$  with  $E[\mathbf{Y}\mathbf{Y}^T] = P^T\Sigma P = D$
- Assume that the diagonal of  $D$  (note that those are eigenvalues) are arranged in descending order that  $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n$ 
  - Generate an approximate  $\hat{\mathbf{Y}}$  of  $\mathbf{Y}$  by setting all components except first  $k$  as 0
  - The mean square error (mse) of<sup>4</sup>  $\hat{\mathbf{Y}} = E[(\mathbf{Y} - \hat{\mathbf{Y}})^T(\mathbf{Y} - \hat{\mathbf{Y}})]$   
 $= \text{tr}(E[(\mathbf{Y} - \hat{\mathbf{Y}})^T(\mathbf{Y} - \hat{\mathbf{Y}})]) = E[\text{tr}((\mathbf{Y} - \hat{\mathbf{Y}})^T(\mathbf{Y} - \hat{\mathbf{Y}}))]$   
 $= E[\text{tr}((\mathbf{Y} - \hat{\mathbf{Y}})(\mathbf{Y} - \hat{\mathbf{Y}})^T)] = \text{tr}(E[(\mathbf{Y} - \hat{\mathbf{Y}})(\mathbf{Y} - \hat{\mathbf{Y}})^T]) = \sum_{i=k+1}^n \lambda_i$
  - Similarly, if we “reconstruct”  $\mathbf{X}$  as  $\hat{\mathbf{X}} = P\hat{\mathbf{Y}}$ . The mse of  $\hat{\mathbf{X}} = E[(\mathbf{X} - \hat{\mathbf{X}})^T(\mathbf{X} - \hat{\mathbf{X}})] = \text{tr}(E[(\mathbf{X} - \hat{\mathbf{X}})(\mathbf{X} - \hat{\mathbf{X}})^T]) = \text{tr}(PE[(\mathbf{Y} - \hat{\mathbf{Y}})(\mathbf{Y} - \hat{\mathbf{Y}})^T]P^T) = \text{tr}(P^TPE[(\mathbf{Y} - \hat{\mathbf{Y}})(\mathbf{Y} - \hat{\mathbf{Y}})^T]) = \sum_{i=k+1}^n \lambda_i$
  - Note that the eigenvectors of  $\Sigma$  (columns of  $P$ ) are known as the principal components

<sup>4</sup> $\text{tr}(AB) = \sum_i \sum_j a_{i,j} b_{j,i} = \sum_j \sum_i b_{j,i} a_{i,j} = \text{tr}(BA)$

# Practical PCA

In practice, we typically are given a dataset with samples of  $\mathbf{X}$  instead of the distribution or covariance matrix of  $\mathbf{X}$ . Denote the data as  $\mathcal{X}$  with each row is a data point and a total of  $m$  data points. Thus  $\mathcal{X}$  is an  $m$  by  $n$  matrix

---

<sup>5</sup>I used the matlab notations for *ones*( $\cdot$ ) and *mean*( $\cdot$ ) here

<sup>6</sup>Note that  $\hat{\Sigma}$  won't be full rank and positive definite as one would hope 

# Practical PCA

In practice, we typically are given a dataset with samples of  $\mathbf{X}$  instead of the distribution or covariance matrix of  $\mathbf{X}$ . Denote the data as  $\mathcal{X}$  with each row is a data point and a total of  $m$  data points. Thus  $\mathcal{X}$  is an  $m$  by  $n$  matrix

- Data are rarely zero-mean to begin with, but we can easily preprocess it by subtracting the mean. That is<sup>5</sup>  $\mathcal{X} \leftarrow \mathcal{X} - \text{ones}(m, 1)\text{mean}(\mathcal{X})$

---

<sup>5</sup>I used the matlab notations for  $\text{ones}(\cdot)$  and  $\text{mean}(\cdot)$  here

<sup>6</sup>Note that  $\hat{\Sigma}$  won't be full rank and positive definite as one would hope 

# Practical PCA

In practice, we typically are given a dataset with samples of  $\mathbf{X}$  instead of the distribution or covariance matrix of  $\mathbf{X}$ . Denote the data as  $\mathcal{X}$  with each row is a data point and a total of  $m$  data points. Thus  $\mathcal{X}$  is an  $m$  by  $n$  matrix

- Data are rarely zero-mean to begin with, but we can easily preprocess it by subtracting the mean. That is<sup>5</sup>  $\mathcal{X} \leftarrow \mathcal{X} - \text{ones}(m, 1)\text{mean}(\mathcal{X})$
- Note that  $\hat{\Sigma} \approx \frac{1}{m}\mathcal{X}^T\mathcal{X}$ . We could directly compute the eigenvectors and eigenvalues of  $\hat{\Sigma}$  as discussed previously. But in many cases,  $m < n$  making  $\hat{\Sigma}$  a bad approximate<sup>6</sup>

---

<sup>5</sup>I used the matlab notations for  $\text{ones}(\cdot)$  and  $\text{mean}(\cdot)$  here

<sup>6</sup>Note that  $\hat{\Sigma}$  won't be full rank and positive definite as one would hope 

# Practical PCA

In practice, we typically are given a dataset with samples of  $\mathbf{X}$  instead of the distribution or covariance matrix of  $\mathbf{X}$ . Denote the data as  $\mathcal{X}$  with each row is a data point and a total of  $m$  data points. Thus  $\mathcal{X}$  is an  $m$  by  $n$  matrix

- Data are rarely zero-mean to begin with, but we can easily preprocess it by subtracting the mean. That is<sup>5</sup>  $\mathcal{X} \leftarrow \mathcal{X} - \text{ones}(m, 1)\text{mean}(\mathcal{X})$
- Note that  $\hat{\Sigma} \approx \frac{1}{m}\mathcal{X}^T\mathcal{X}$ . We could directly compute the eigenvectors and eigenvalues of  $\hat{\Sigma}$  as discussed previously. But in many cases,  $m < n$  making  $\hat{\Sigma}$  a bad approximate<sup>6</sup>
  - A more common approach is to decompose  $\mathcal{X}$  with singular value decomposition (SVD) instead

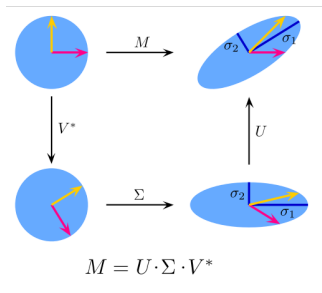
---

<sup>5</sup>I used the matlab notations for  $\text{ones}(\cdot)$  and  $\text{mean}(\cdot)$  here

<sup>6</sup>Note that  $\hat{\Sigma}$  won't be full rank and positive definite as one would hope 

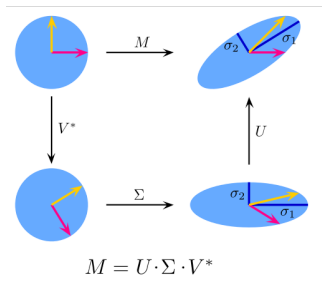


# Singular value decomposition (SVD)



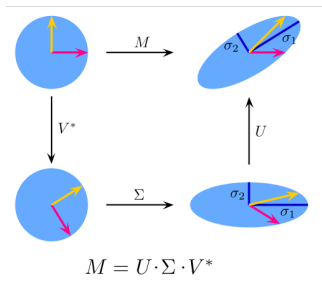
- Every matrix  $M$  can be decomposed as  $M = UDV^\dagger$ , where  $D$  is diagonal and  $U, V$  are unitary. The diagonal terms in  $\Sigma$  are known to be the **singular values**

# Singular value decomposition (SVD)



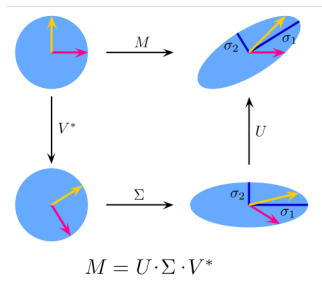
- Every matrix  $M$  can be decomposed as  $M = UDV^\dagger$ , where  $D$  is diagonal and  $U, V$  are unitary. The diagonal terms in  $\Sigma$  are known to be the **singular values**
- For real matrix  $M$ , we can write  $M = UDV^T$  instead.  $U, V$  are now “real unitary” or orthogonal

# Singular value decomposition (SVD)



- Every matrix  $M$  can be decomposed as  $M = UDV^\dagger$ , where  $D$  is diagonal and  $U, V$  are unitary. The diagonal terms in  $\Sigma$  are known to be the **singular values**
- For real matrix  $M$ , we can write  $M = UDV^T$  instead.  $U, V$  are now “real unitary” or orthogonal
  - Note that  $M^T M = VD^T U^T U D V^T = VD^2 V^T$ . Therefore,  $V$  are really eigenvectors of  $M^T M$  with eigenvalues equal to the square of the singular values

# Singular value decomposition (SVD)



- Every matrix  $M$  can be decomposed as  $M = UDV^\dagger$ , where  $D$  is diagonal and  $U, V$  are unitary. The diagonal terms in  $\Sigma$  are known to be the **singular values**
- For real matrix  $M$ , we can write  $M = UDV^T$  instead.  $U, V$  are now “real unitary” or orthogonal
  - Note that  $M^T M = VD^T U^T U D V^T = VD^2 V^T$ . Therefore,  $V$  are really eigenvectors of  $M^T M$  with eigenvalues equal to the square of the singular values
  - Similar, we have  $MM^T = UD^2 U^T$

# PCA with SVD

So from previous slides, instead of first estimating the covariance matrix and then diagonalize it. We should directly decompose the data  $\mathcal{X}$  with SVD instead. The process is summarized below

# PCA with SVD

So from previous slides, instead of first estimating the covariance matrix and then diagonalize it. We should directly decompose the data  $\mathcal{X}$  with SVD instead. The process is summarized below

- Estimate mean from data and subtract mean from that

# PCA with SVD

So from previous slides, instead of first estimating the covariance matrix and then diagonalize it. We should directly decompose the data  $\mathcal{X}$  with SVD instead. The process is summarized below

- Estimate mean from data and subtract mean from that
- Decomposed the mean subtracted data with SVD. We get

$$\mathcal{X} = UDV^T$$

# PCA with SVD

So from previous slides, instead of first estimating the covariance matrix and then diagonalize it. We should directly decompose the data  $\mathcal{X}$  with SVD instead. The process is summarized below

- Estimate mean from data and subtract mean from that
- Decomposed the mean subtracted data with SVD. We get  $\mathcal{X} = UDV^T$
- Note that column of  $V$  are now the principal components, and we can transform a data column as  $V^T x$ . The entire data set can be transformed as  $\mathcal{Y} = \mathcal{X}V$



# PCA with SVD

So from previous slides, instead of first estimating the covariance matrix and then diagonalize it. We should directly decompose the data  $\mathcal{X}$  with SVD instead. The process is summarized below

- Estimate mean from data and subtract mean from that
- Decomposed the mean subtracted data with SVD. We get  $\mathcal{X} = UDV^T$
- Note that column of  $V$  are now the principal components, and we can transform a data column as  $V^T x$ . The entire data set can be transformed as  $\mathcal{Y} = \mathcal{X}V$ 
  - The first few columns of  $\mathcal{Y}$  will contain most “information” regarding the original  $\mathcal{X}$

# PCA with SVD

So from previous slides, instead of first estimating the covariance matrix and then diagonalize it. We should directly decompose the data  $\mathcal{X}$  with SVD instead. The process is summarized below

- Estimate mean from data and subtract mean from that
- Decomposed the mean subtracted data with SVD. We get  $\mathcal{X} = UDV^T$
- Note that column of  $V$  are now the principal components, and we can transform a data column as  $V^T x$ . The entire data set can be transformed as  $\mathcal{Y} = \mathcal{X}V$ 
  - The first few columns of  $\mathcal{Y}$  will contain most “information” regarding the original  $\mathcal{X}$
  - For example, they can be taken as features for recognition or one can omit other columns besides the first few for “compression” as discussed earlier

# Review

- ML:  $\hat{x} = \arg \max_x p(x|\hat{\theta}), \hat{\theta} = \arg \max_{\theta} p(o|\theta)$
- MAP:  $\hat{x} = \arg \max_x p(x|\hat{\theta}), \hat{\theta} = \arg \max_{\theta} p(\theta|o)$
- Bayesian:  $\hat{x} = \sum_{\theta} p(\theta|o) \sum_x xp(x|\theta)$
- For zero-mean  $\mathbf{X}$ ,  $\Sigma_{\mathbf{X}} = E[\mathbf{X}\mathbf{X}^T]$  and say we have  $P^T \Sigma_{\mathbf{X}} P = D$ . The transformed  $\mathbf{Y} = P^T \mathbf{X}$  are independent to each other
  - Note that the transform is just principal component analysis
- Marginalization of a normal distribution is still a normal distribution
- (a)  $\Sigma_{\mathbf{X}\mathbf{X}}^{-1} = \Lambda_{\mathbf{X}\mathbf{X}} - \Lambda_{\mathbf{X}\mathbf{Y}} \Lambda_{\mathbf{Y}\mathbf{Y}}^{-1} \Lambda_{\mathbf{Y}\mathbf{X}}$
- (b)  $\det(a\Sigma) = \det(a\Sigma_{\mathbf{Y}\mathbf{Y}}) \det(a\Lambda_{\mathbf{X}\mathbf{X}}^{-1})$  for any constant  $a$

# Conditioning of normal distribution

- Consider the same  $\mathbf{Z} \sim \mathcal{N}(\boldsymbol{\mu}_{\mathbf{Z}}, \boldsymbol{\Sigma}_{\mathbf{Z}})$  and  $\mathbf{Z} = \begin{pmatrix} \mathbf{X} \\ \mathbf{Y} \end{pmatrix}$ . What will  $\mathbf{X}$  be like if  $\mathbf{Y}$  is observed to be  $\mathbf{y}$ ?

# Conditioning of normal distribution

- Consider the same  $\mathbf{Z} \sim \mathcal{N}(\boldsymbol{\mu}_{\mathbf{Z}}, \boldsymbol{\Sigma}_{\mathbf{Z}})$  and  $\mathbf{Z} = \begin{pmatrix} \mathbf{X} \\ \mathbf{Y} \end{pmatrix}$ . What will  $\mathbf{X}$  be like if  $\mathbf{Y}$  is observed to be  $\mathbf{y}$ ?
- Basically, we want to find  $p(\mathbf{x}|\mathbf{y}) = p(\mathbf{x}, \mathbf{y})/p(\mathbf{y})$

# Conditioning of normal distribution

- Consider the same  $\mathbf{Z} \sim \mathcal{N}(\boldsymbol{\mu}_Z, \Sigma_Z)$  and  $\mathbf{Z} = \begin{pmatrix} \mathbf{X} \\ \mathbf{Y} \end{pmatrix}$ . What will  $\mathbf{X}$  be like if  $\mathbf{Y}$  is observed to be  $\mathbf{y}$ ?
- Basically, we want to find  $p(\mathbf{x}|\mathbf{y}) = p(\mathbf{x}, \mathbf{y})/p(\mathbf{y})$
- From previous result, we have  $p(\mathbf{y}) = \mathcal{N}(\mathbf{y}; \boldsymbol{\mu}_Y, \Sigma_{YY})$ . Therefore,

$$\begin{aligned}
 p(\mathbf{x}|\mathbf{y}) &\propto \exp\left(-\frac{1}{2}\left[\begin{pmatrix} \tilde{\mathbf{x}} \\ \tilde{\mathbf{y}} \end{pmatrix}^T \Sigma^{-1} \begin{pmatrix} \tilde{\mathbf{x}} \\ \tilde{\mathbf{y}} \end{pmatrix} - \tilde{\mathbf{y}}^T \Sigma_{YY}^{-1} \tilde{\mathbf{y}}\right]\right) \\
 &\propto \exp\left(-\frac{1}{2}[\tilde{\mathbf{x}}^T \Lambda_{XX} \tilde{\mathbf{x}} + \tilde{\mathbf{x}}^T \Lambda_{XY} \tilde{\mathbf{y}} + \tilde{\mathbf{y}}^T \Lambda_{YX} \tilde{\mathbf{x}}]\right),
 \end{aligned}$$

where we use  $\tilde{\mathbf{x}}$  and  $\tilde{\mathbf{y}}$  as shorthands of  $\mathbf{x} - \boldsymbol{\mu}_X$  and  $\mathbf{y} - \boldsymbol{\mu}_Y$  as before

# Conditioning of normal distribution

- Completing the square for  $\tilde{\mathbf{x}}$ , we have

$$\begin{aligned}
 p(\mathbf{x}|\mathbf{y}) &\propto \exp\left(-\frac{1}{2}(\tilde{\mathbf{x}} + \Lambda_{\mathbf{XX}}^{-1}\Lambda_{\mathbf{XY}}\tilde{\mathbf{y}})^T \Lambda_{\mathbf{XX}}(\tilde{\mathbf{x}} + \Lambda_{\mathbf{XX}}^{-1}\Lambda_{\mathbf{XY}}\tilde{\mathbf{y}})\right) \\
 &= \exp\left(-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_{\mathbf{X}} + \Lambda_{\mathbf{XX}}^{-1}\Lambda_{\mathbf{XY}}(\mathbf{y} - \boldsymbol{\mu}_{\mathbf{Y}}))^T \Lambda_{\mathbf{XX}}\right. \\
 &\quad \left. (\mathbf{x} - \boldsymbol{\mu}_{\mathbf{X}} + \Lambda_{\mathbf{XX}}^{-1}\Lambda_{\mathbf{XY}}(\mathbf{y} - \boldsymbol{\mu}_{\mathbf{Y}}))\right)
 \end{aligned}$$

# Conditioning of normal distribution

- Completing the square for  $\tilde{\mathbf{x}}$ , we have

$$\begin{aligned} p(\mathbf{x}|\mathbf{y}) &\propto \exp\left(-\frac{1}{2}(\tilde{\mathbf{x}} + \Lambda_{\mathbf{XX}}^{-1}\Lambda_{\mathbf{XY}}\tilde{\mathbf{y}})^T \Lambda_{\mathbf{XX}}(\tilde{\mathbf{x}} + \Lambda_{\mathbf{XX}}^{-1}\Lambda_{\mathbf{XY}}\tilde{\mathbf{y}})\right) \\ &= \exp\left(-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_{\mathbf{X}} + \Lambda_{\mathbf{XX}}^{-1}\Lambda_{\mathbf{XY}}(\mathbf{y} - \boldsymbol{\mu}_{\mathbf{Y}}))^T \Lambda_{\mathbf{XX}}\right. \\ &\quad \left. (\mathbf{x} - \boldsymbol{\mu}_{\mathbf{X}} + \Lambda_{\mathbf{XX}}^{-1}\Lambda_{\mathbf{XY}}(\mathbf{y} - \boldsymbol{\mu}_{\mathbf{Y}}))\right) \end{aligned}$$

- Therefore  $\mathbf{X}|\mathbf{y}$  is Gaussian distributed with mean  $\boldsymbol{\mu}_{\mathbf{X}} - \Lambda_{\mathbf{XX}}^{-1}\Lambda_{\mathbf{XY}}(\mathbf{y} - \boldsymbol{\mu}_{\mathbf{Y}})$  and covariance  $\Lambda_{\mathbf{XX}}^{-1}$



# Conditioning of normal distribution

- Completing the square for  $\tilde{\mathbf{x}}$ , we have

$$\begin{aligned} p(\mathbf{x}|\mathbf{y}) &\propto \exp\left(-\frac{1}{2}(\tilde{\mathbf{x}} + \Lambda_{\mathbf{XX}}^{-1}\Lambda_{\mathbf{XY}}\tilde{\mathbf{y}})^T \Lambda_{\mathbf{XX}}(\tilde{\mathbf{x}} + \Lambda_{\mathbf{XX}}^{-1}\Lambda_{\mathbf{XY}}\tilde{\mathbf{y}})\right) \\ &= \exp\left(-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_{\mathbf{X}} + \Lambda_{\mathbf{XX}}^{-1}\Lambda_{\mathbf{XY}}(\mathbf{y} - \boldsymbol{\mu}_{\mathbf{Y}}))^T \Lambda_{\mathbf{XX}}\right. \\ &\quad \left. (\mathbf{x} - \boldsymbol{\mu}_{\mathbf{X}} + \Lambda_{\mathbf{XX}}^{-1}\Lambda_{\mathbf{XY}}(\mathbf{y} - \boldsymbol{\mu}_{\mathbf{Y}}))\right) \end{aligned}$$

- Therefore  $\mathbf{X}|\mathbf{y}$  is Gaussian distributed with mean  $\boldsymbol{\mu}_{\mathbf{X}} - \Lambda_{\mathbf{XX}}^{-1}\Lambda_{\mathbf{XY}}(\mathbf{y} - \boldsymbol{\mu}_{\mathbf{Y}})$  and covariance  $\Lambda_{\mathbf{XX}}^{-1}$
- Note that since  $\Lambda_{\mathbf{XX}}\Sigma_{\mathbf{XY}} + \Lambda_{\mathbf{XY}}\Sigma_{\mathbf{YY}} = 0 \Rightarrow \Lambda_{\mathbf{XX}}^{-1}\Lambda_{\mathbf{XY}} = -\Sigma_{\mathbf{XY}}\Sigma_{\mathbf{YY}}^{-1}$  and from (a), we have

$$\mathbf{X}|\mathbf{y} \sim \mathcal{N}(\boldsymbol{\mu}_{\mathbf{X}} + \Sigma_{\mathbf{XY}}\Sigma_{\mathbf{YY}}^{-1}(\mathbf{y} - \boldsymbol{\mu}_{\mathbf{Y}}), \Sigma_{\mathbf{XX}} - \Sigma_{\mathbf{XY}}\Sigma_{\mathbf{YY}}^{-1}\Sigma_{\mathbf{YX}})$$

# Interpretation of conditioning

$$\mathbf{X}|\mathbf{y} \sim \mathcal{N}(\boldsymbol{\mu}_X + \boldsymbol{\Sigma}_{XY}\boldsymbol{\Sigma}_{YY}^{-1}(\mathbf{y} - \boldsymbol{\mu}_Y), \boldsymbol{\Sigma}_{XX} - \boldsymbol{\Sigma}_{XY}\boldsymbol{\Sigma}_{YY}^{-1}\boldsymbol{\Sigma}_{YX})$$

- When the observation of  $\mathbf{Y}$  is exactly the mean, the conditioned mean does not change

# Interpretation of conditioning

$$\mathbf{X}|\mathbf{y} \sim \mathcal{N}(\boldsymbol{\mu}_{\mathbf{X}} + \boldsymbol{\Sigma}_{\mathbf{X}\mathbf{Y}}\boldsymbol{\Sigma}_{\mathbf{Y}\mathbf{Y}}^{-1}(\mathbf{y} - \boldsymbol{\mu}_{\mathbf{Y}}), \boldsymbol{\Sigma}_{\mathbf{X}\mathbf{X}} - \boldsymbol{\Sigma}_{\mathbf{X}\mathbf{Y}}\boldsymbol{\Sigma}_{\mathbf{Y}\mathbf{Y}}^{-1}\boldsymbol{\Sigma}_{\mathbf{Y}\mathbf{X}})$$

- When the observation of  $\mathbf{Y}$  is exactly the mean, the conditioned mean does not change
- Otherwise, it needs to be modified and the size of the adjustment decreases with  $\boldsymbol{\Sigma}_{\mathbf{Y}\mathbf{Y}}$ , the variance of  $\mathbf{Y}$  for the 1-D case.
  - The observation is less reliable with the increase of  $\boldsymbol{\Sigma}_{\mathbf{Y}\mathbf{Y}}$ . The adjustment is finally scaled by  $\boldsymbol{\Sigma}_{\mathbf{X}\mathbf{Y}}$ , which translates the variation of  $\mathbf{Y}$  to the variation of  $\mathbf{X}$

# Interpretation of conditioning

$$\mathbf{X}|\mathbf{y} \sim \mathcal{N}(\boldsymbol{\mu}_{\mathbf{X}} + \boldsymbol{\Sigma}_{\mathbf{X}\mathbf{Y}}\boldsymbol{\Sigma}_{\mathbf{Y}\mathbf{Y}}^{-1}(\mathbf{y} - \boldsymbol{\mu}_{\mathbf{Y}}), \boldsymbol{\Sigma}_{\mathbf{X}\mathbf{X}} - \boldsymbol{\Sigma}_{\mathbf{X}\mathbf{Y}}\boldsymbol{\Sigma}_{\mathbf{Y}\mathbf{Y}}^{-1}\boldsymbol{\Sigma}_{\mathbf{Y}\mathbf{X}})$$

- When the observation of  $\mathbf{Y}$  is exactly the mean, the conditioned mean does not change
- Otherwise, it needs to be modified and the size of the adjustment decreases with  $\boldsymbol{\Sigma}_{\mathbf{Y}\mathbf{Y}}$ , the variance of  $\mathbf{Y}$  for the 1-D case.
  - The observation is less reliable with the increase of  $\boldsymbol{\Sigma}_{\mathbf{Y}\mathbf{Y}}$ . The adjustment is finally scaled by  $\boldsymbol{\Sigma}_{\mathbf{X}\mathbf{Y}}$ , which translates the variation of  $\mathbf{Y}$  to the variation of  $\mathbf{X}$
  - In particular, if  $\mathbf{X}$  and  $\mathbf{Y}$  are negatively correlated, the sign of the adjustment will be reversed

# Interpretation of conditioning

$$\mathbf{X}|\mathbf{y} \sim \mathcal{N}(\boldsymbol{\mu}_{\mathbf{X}} + \boldsymbol{\Sigma}_{\mathbf{X}\mathbf{Y}}\boldsymbol{\Sigma}_{\mathbf{Y}\mathbf{Y}}^{-1}(\mathbf{y} - \boldsymbol{\mu}_{\mathbf{Y}}), \boldsymbol{\Sigma}_{\mathbf{X}\mathbf{X}} - \boldsymbol{\Sigma}_{\mathbf{X}\mathbf{Y}}\boldsymbol{\Sigma}_{\mathbf{Y}\mathbf{Y}}^{-1}\boldsymbol{\Sigma}_{\mathbf{Y}\mathbf{X}})$$

- When the observation of  $\mathbf{Y}$  is exactly the mean, the conditioned mean does not change
- Otherwise, it needs to be modified and the size of the adjustment decreases with  $\boldsymbol{\Sigma}_{\mathbf{Y}\mathbf{Y}}$ , the variance of  $\mathbf{Y}$  for the 1-D case.
  - The observation is less reliable with the increase of  $\boldsymbol{\Sigma}_{\mathbf{Y}\mathbf{Y}}$ . The adjustment is finally scaled by  $\boldsymbol{\Sigma}_{\mathbf{X}\mathbf{Y}}$ , which translates the variation of  $\mathbf{Y}$  to the variation of  $\mathbf{X}$
  - In particular, if  $\mathbf{X}$  and  $\mathbf{Y}$  are negatively correlated, the sign of the adjustment will be reversed
- As for the variance of the conditioned variable, it always decreases and the decrease is larger if  $\boldsymbol{\Sigma}_{\mathbf{Y}\mathbf{Y}}$  is smaller and  $\boldsymbol{\Sigma}_{\mathbf{X}\mathbf{Y}}$  is larger ( $\mathbf{X}$  and  $\mathbf{Y}$  are more correlated)

$X \perp\!\!\!\perp Y|Z$  if  $\rho_{XZ}\rho_{YZ} = \rho_{XY}$

### Corollary

Given multivariate Gaussian variables  $X, Y$  and  $Z$ , we have  $X$  and  $Y$  are conditionally independent given  $Z$  if  $\rho_{XZ}\rho_{YZ} = \rho_{XY}$ , where  $\rho_{XZ} = \frac{E[(X-E(X))(Z-E(Z))]}{\sqrt{E[(X-E(X))^2]E[(Z-E(Z))^2]}}$  is the correlation coefficient between  $X$  and  $Z$ . Similarly,  $\rho_{YZ}$  and  $\rho_{XY}$  are the correlation coefficients between  $Y$  and  $Z$ , and  $X$  and  $Y$ , respectively.

$$X \perp\!\!\!\perp Y|Z \text{ if } \rho_{XZ}\rho_{YZ} = \rho_{XY}$$

Proof.

- Without loss of generality, we can assume the variables with mean 0 and variance 1. Thus,  $\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} \sim \mathcal{N}(\mathbf{0}, \Sigma)$ , where  $\Sigma = \begin{pmatrix} 1 & \rho_{XY} & \rho_{XZ} \\ \rho_{XY} & 1 & \rho_{YZ} \\ \rho_{XZ} & \rho_{YZ} & 1 \end{pmatrix}$

$X \perp\!\!\!\perp Y|Z$  if  $\rho_{XZ}\rho_{YZ} = \rho_{XY}$

Proof.

- Without loss of generality, we can assume the variables with mean 0 and variance 1. Thus,  $\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} \sim \mathcal{N}(\mathbf{0}, \Sigma)$ , where  $\Sigma = \begin{pmatrix} 1 & \rho_{XY} & \rho_{XZ} \\ \rho_{XY} & 1 & \rho_{YZ} \\ \rho_{XZ} & \rho_{YZ} & 1 \end{pmatrix}$
- Then from the conditioning result, we have

$$\begin{aligned} \Sigma \begin{pmatrix} X \\ Y \end{pmatrix} | Z &= \begin{pmatrix} 1 & \rho_{XY} \\ \rho_{XY} & 1 \end{pmatrix} - (\rho_{XZ} \quad \rho_{YZ}) \sigma_{YY}^{-1} \begin{pmatrix} \rho_{XZ} \\ \rho_{YZ} \end{pmatrix} \\ &= \begin{pmatrix} 1 - \rho_{XZ}^2 & \rho_{XY} - \rho_{XZ}\rho_{YZ} \\ \rho_{XY} - \rho_{XZ}\rho_{YZ} & 1 - \rho_{YZ}^2 \end{pmatrix} \end{aligned}$$



$X \perp\!\!\!\perp Y|Z$  if  $\rho_{XZ}\rho_{YZ} = \rho_{XY}$

Proof.

- Without loss of generality, we can assume the variables with mean 0 and variance 1. Thus,  $\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} \sim \mathcal{N}(\mathbf{0}, \Sigma)$ , where  $\Sigma = \begin{pmatrix} 1 & \rho_{XY} & \rho_{XZ} \\ \rho_{XY} & 1 & \rho_{YZ} \\ \rho_{XZ} & \rho_{YZ} & 1 \end{pmatrix}$

- Then from the conditioning result, we have

$$\begin{aligned} \Sigma \begin{pmatrix} X \\ Y \end{pmatrix} | Z &= \begin{pmatrix} 1 & \rho_{XY} \\ \rho_{XY} & 1 \end{pmatrix} - (\rho_{XZ} \quad \rho_{YZ}) \sigma_{YY}^{-1} \begin{pmatrix} \rho_{XZ} \\ \rho_{YZ} \end{pmatrix} \\ &= \begin{pmatrix} 1 - \rho_{XZ}^2 & \rho_{XY} - \rho_{XZ}\rho_{YZ} \\ \rho_{XY} - \rho_{XZ}\rho_{YZ} & 1 - \rho_{YZ}^2 \end{pmatrix} \end{aligned}$$

- Therefore,  $X$  and  $Y$  are uncorrelated given  $Z$  when  $\sigma_{XY|Z} = \rho_{XY} - \rho_{XZ}\rho_{YZ} = 0$  or  $\rho_{XY} = \rho_{XZ}\rho_{YZ}$ . Since for Gaussian variables, uncorrelatedness implies independence. This concludes the proof. □

# Product of normal distributions

- Assume that we try to recover some vector parameter  $\mathbf{x}$ , which is subject to multivariate Gaussian noise

# Product of normal distributions

- Assume that we try to recover some vector parameter  $\mathbf{x}$ , which is subject to multivariate Gaussian noise
- Say we made two measurements  $\mathbf{y}_1$  and  $\mathbf{y}_2$ , where  $\mathbf{Y}_1 \sim \mathcal{N}(\mathbf{x}, \Sigma_{\mathbf{Y}_1})$  and  $\mathbf{Y}_2 \sim \mathcal{N}(\mathbf{x}, \Sigma_{\mathbf{Y}_2})$ . Note that even though both measurements have mean  $\mathbf{x}$ , they have different covariance
  - This variation, for instance, can be due to environment change between the two measurements

# Product of normal distributions

- Assume that we try to recover some vector parameter  $\mathbf{x}$ , which is subject to multivariate Gaussian noise
- Say we made two measurements  $\mathbf{y}_1$  and  $\mathbf{y}_2$ , where  $\mathbf{Y}_1 \sim \mathcal{N}(\mathbf{x}, \Sigma_{\mathbf{Y}_1})$  and  $\mathbf{Y}_2 \sim \mathcal{N}(\mathbf{x}, \Sigma_{\mathbf{Y}_2})$ . Note that even though both measurements have mean  $\mathbf{x}$ , they have different covariance
  - This variation, for instance, can be due to environment change between the two measurements
- Now, if we want to compute the overall likelihood,  $p(\mathbf{y}_1, \mathbf{y}_2 | \mathbf{x})$ . Assuming that  $\mathbf{Y}_1$  and  $\mathbf{Y}_2$  are conditionally independent given  $\mathbf{X}$ , we have

$$\begin{aligned} p(\mathbf{y}_1, \mathbf{y}_2 | \mathbf{x}) &= p(\mathbf{y}_1 | \mathbf{x}) p(\mathbf{y}_2 | \mathbf{x}) \\ &= \mathcal{N}(\mathbf{y}_1; \mathbf{x}, \Sigma_{\mathbf{Y}_1}) \mathcal{N}(\mathbf{y}_2; \mathbf{x}, \Sigma_{\mathbf{Y}_2}). \end{aligned}$$

# Product of normal distributions

- Assume that we try to recover some vector parameter  $\mathbf{x}$ , which is subject to multivariate Gaussian noise
- Say we made two measurements  $\mathbf{y}_1$  and  $\mathbf{y}_2$ , where  $\mathbf{Y}_1 \sim \mathcal{N}(\mathbf{x}, \Sigma_{\mathbf{Y}_1})$  and  $\mathbf{Y}_2 \sim \mathcal{N}(\mathbf{x}, \Sigma_{\mathbf{Y}_2})$ . Note that even though both measurements have mean  $\mathbf{x}$ , they have different covariance
  - This variation, for instance, can be due to environment change between the two measurements
- Now, if we want to compute the overall likelihood,  $p(\mathbf{y}_1, \mathbf{y}_2 | \mathbf{x})$ . Assuming that  $\mathbf{Y}_1$  and  $\mathbf{Y}_2$  are conditionally independent given  $\mathbf{X}$ , we have

$$\begin{aligned} p(\mathbf{y}_1, \mathbf{y}_2 | \mathbf{x}) &= p(\mathbf{y}_1 | \mathbf{x}) p(\mathbf{y}_2 | \mathbf{x}) \\ &= \mathcal{N}(\mathbf{y}_1; \mathbf{x}, \Sigma_{\mathbf{Y}_1}) \mathcal{N}(\mathbf{y}_2; \mathbf{x}, \Sigma_{\mathbf{Y}_2}). \end{aligned}$$

- Essentially, we just need to compute the product of two Gaussian pdfs. Such computation is very useful and it occurs often when one needs to perform inference

# Product of normal distributions

As in previous cases, the product turns out to be normal also. However, unlike them, **the product is not a pdf and so it does not normalize to 1**. So we have to compute both the scaling factor and the exponent explicitly. Let us start with the exponent.

$$\mathcal{N}(\mathbf{y}_1; \mathbf{x}, \Sigma_{\mathbf{Y}_1})\mathcal{N}(\mathbf{y}_2; \mathbf{x}, \Sigma_{\mathbf{Y}_2})$$

# Product of normal distributions

As in previous cases, the product turns out to be normal also. However, unlike them, **the product is not a pdf and so it does not normalize to 1**. So we have to compute both the scaling factor and the exponent explicitly. Let us start with the exponent.

$$\begin{aligned} & \mathcal{N}(\mathbf{y}_1; \mathbf{x}, \Sigma_{\mathbf{Y}_1}) \mathcal{N}(\mathbf{y}_2; \mathbf{x}, \Sigma_{\mathbf{Y}_2}) \\ & \propto \exp \left( -\frac{1}{2} [(\mathbf{x} - \mathbf{y}_1)^T \Lambda_{\mathbf{Y}_1} (\mathbf{x} - \mathbf{y}_1) + (\mathbf{x} - \mathbf{y}_2)^T \Lambda_{\mathbf{Y}_2} (\mathbf{x} - \mathbf{y}_2)] \right) \end{aligned}$$

# Product of normal distributions

As in previous cases, the product turns out to be normal also. However, unlike them, **the product is not a pdf and so it does not normalize to 1**. So we have to compute both the scaling factor and the exponent explicitly. Let us start with the exponent.

$$\begin{aligned} & \mathcal{N}(\mathbf{y}_1; \mathbf{x}, \Sigma_{\mathbf{Y}_1}) \mathcal{N}(\mathbf{y}_2; \mathbf{x}, \Sigma_{\mathbf{Y}_2}) \\ & \propto \exp \left( -\frac{1}{2} [(\mathbf{x} - \mathbf{y}_1)^T \Lambda_{\mathbf{Y}_1} (\mathbf{x} - \mathbf{y}_1) + (\mathbf{x} - \mathbf{y}_2)^T \Lambda_{\mathbf{Y}_2} (\mathbf{x} - \mathbf{y}_2)] \right) \\ & \propto \exp \left( -\frac{1}{2} [\mathbf{x}^T (\Lambda_{\mathbf{Y}_1} + \Lambda_{\mathbf{Y}_2}) \mathbf{x} - (\mathbf{y}_2^T \Lambda_{\mathbf{Y}_2} + \mathbf{y}_1^T \Lambda_{\mathbf{Y}_1}) \mathbf{x} - \mathbf{x}^T (\Lambda_{\mathbf{Y}_2} \mathbf{y}_2 + \Lambda_{\mathbf{Y}_1} \mathbf{y}_1)] \right) \end{aligned}$$



# Product of normal distributions

As in previous cases, the product turns out to be normal also. However, unlike them, **the product is not a pdf and so it does not normalize to 1**. So we have to compute both the scaling factor and the exponent explicitly. Let us start with the exponent.

$$\begin{aligned}
 & \mathcal{N}(\mathbf{y}_1; \mathbf{x}, \Sigma_{\mathbf{Y}_1}) \mathcal{N}(\mathbf{y}_2; \mathbf{x}, \Sigma_{\mathbf{Y}_2}) \\
 & \propto \exp \left( -\frac{1}{2} [(\mathbf{x} - \mathbf{y}_1)^T \Lambda_{\mathbf{Y}_1} (\mathbf{x} - \mathbf{y}_1) + (\mathbf{x} - \mathbf{y}_2)^T \Lambda_{\mathbf{Y}_2} (\mathbf{x} - \mathbf{y}_2)] \right) \\
 & \propto \exp \left( -\frac{1}{2} [\mathbf{x}^T (\Lambda_{\mathbf{Y}_1} + \Lambda_{\mathbf{Y}_2}) \mathbf{x} - (\mathbf{y}_2^T \Lambda_{\mathbf{Y}_2} + \mathbf{y}_1^T \Lambda_{\mathbf{Y}_1}) \mathbf{x} - \mathbf{x}^T (\Lambda_{\mathbf{Y}_2} \mathbf{y}_2 + \Lambda_{\mathbf{Y}_1} \mathbf{y}_1)] \right) \\
 & \propto e^{-\frac{1}{2} [(\mathbf{x} - (\Lambda_{\mathbf{Y}_1} + \Lambda_{\mathbf{Y}_2})^{-1} (\Lambda_{\mathbf{Y}_2} \mathbf{y}_2 + \Lambda_{\mathbf{Y}_1} \mathbf{y}_1))^T (\Lambda_{\mathbf{Y}_1} + \Lambda_{\mathbf{Y}_2}) (\mathbf{x} - (\Lambda_{\mathbf{Y}_1} + \Lambda_{\mathbf{Y}_2})^{-1} (\Lambda_{\mathbf{Y}_2} \mathbf{y}_2 + \Lambda_{\mathbf{Y}_1} \mathbf{y}_1))] }
 \end{aligned}$$

# Product of normal distributions

As in previous cases, the product turns out to be normal also. However, unlike them, **the product is not a pdf and so it does not normalize to 1**. So we have to compute both the scaling factor and the exponent explicitly. Let us start with the exponent.

$$\begin{aligned}
 & \mathcal{N}(\mathbf{y}_1; \mathbf{x}, \Sigma_{\mathbf{Y}_1}) \mathcal{N}(\mathbf{y}_2; \mathbf{x}, \Sigma_{\mathbf{Y}_2}) \\
 & \propto \exp \left( -\frac{1}{2} [(\mathbf{x} - \mathbf{y}_1)^T \Lambda_{\mathbf{Y}_1} (\mathbf{x} - \mathbf{y}_1) + (\mathbf{x} - \mathbf{y}_2)^T \Lambda_{\mathbf{Y}_2} (\mathbf{x} - \mathbf{y}_2)] \right) \\
 & \propto \exp \left( -\frac{1}{2} [\mathbf{x}^T (\Lambda_{\mathbf{Y}_1} + \Lambda_{\mathbf{Y}_2}) \mathbf{x} - (\mathbf{y}_2^T \Lambda_{\mathbf{Y}_2} + \mathbf{y}_1^T \Lambda_{\mathbf{Y}_1}) \mathbf{x} - \mathbf{x}^T (\Lambda_{\mathbf{Y}_2} \mathbf{y}_2 + \Lambda_{\mathbf{Y}_1} \mathbf{y}_1)] \right) \\
 & \propto e^{-\frac{1}{2} [(\mathbf{x} - (\Lambda_{\mathbf{Y}_1} + \Lambda_{\mathbf{Y}_2})^{-1} (\Lambda_{\mathbf{Y}_2} \mathbf{y}_2 + \Lambda_{\mathbf{Y}_1} \mathbf{y}_1))^T (\Lambda_{\mathbf{Y}_1} + \Lambda_{\mathbf{Y}_2}) (\mathbf{x} - (\Lambda_{\mathbf{Y}_1} + \Lambda_{\mathbf{Y}_2})^{-1} (\Lambda_{\mathbf{Y}_2} \mathbf{y}_2 + \Lambda_{\mathbf{Y}_1} \mathbf{y}_1))]} \\
 & \propto \mathcal{N}(\mathbf{x}; (\Lambda_{\mathbf{Y}_1} + \Lambda_{\mathbf{Y}_2})^{-1} (\Lambda_{\mathbf{Y}_2} \mathbf{y}_2 + \Lambda_{\mathbf{Y}_1} \mathbf{y}_1), (\Lambda_{\mathbf{Y}_2} + \Lambda_{\mathbf{Y}_1})^{-1})
 \end{aligned}$$

Therefore,

$$\begin{aligned}
 & \mathcal{N}(\mathbf{y}_1; \mathbf{x}, \Sigma_{\mathbf{Y}_1}) \mathcal{N}(\mathbf{y}_2; \mathbf{x}, \Sigma_{\mathbf{Y}_2}) \\
 & = K(\mathbf{y}_1, \mathbf{y}_2, \Sigma_{\mathbf{Y}_1}, \Sigma_{\mathbf{Y}_2}) \mathcal{N}(\mathbf{x}; (\Lambda_{\mathbf{Y}_1} + \Lambda_{\mathbf{Y}_2})^{-1} (\Lambda_{\mathbf{Y}_2} \mathbf{y}_2 + \Lambda_{\mathbf{Y}_1} \mathbf{y}_1), (\Lambda_{\mathbf{Y}_2} + \Lambda_{\mathbf{Y}_1})^{-1})
 \end{aligned}$$

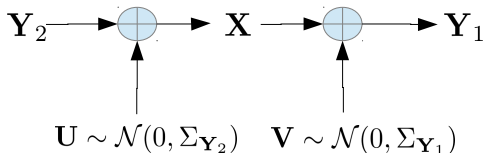
for some scaling factor  $K(\mathbf{y}_1, \mathbf{y}_2, \Sigma_{\mathbf{Y}_1}, \Sigma_{\mathbf{Y}_2})$  independent of  $\mathbf{x}$ .

# Product of normal distributions

- One can compute the scaling factor  $K(\mathbf{y}_1, \mathbf{y}_2, \Sigma_{\mathbf{Y}_1}, \Sigma_{\mathbf{Y}_2})$  directly

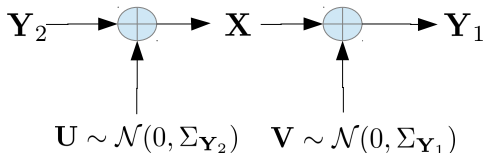
# Product of normal distributions

- One can compute the scaling factor  $K(\mathbf{y}_1, \mathbf{y}_2, \Sigma_{\mathbf{Y}_1}, \Sigma_{\mathbf{Y}_2})$  directly
- However, it is much easier to take advantage for the following setup when  $\mathbf{Y}_1 \perp\!\!\!\perp \mathbf{Y}_2 | \mathbf{X}$  as shown below



# Product of normal distributions

- One can compute the scaling factor  $K(\mathbf{y}_1, \mathbf{y}_2, \Sigma_{\mathbf{Y}_1}, \Sigma_{\mathbf{Y}_2})$  directly
- However, it is much easier to take advantage for the following setup when  $\mathbf{Y}_1 \perp\!\!\!\perp \mathbf{Y}_2 | \mathbf{X}$  as shown below



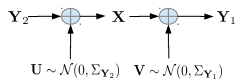
- Since  $\mathcal{N}(\mathbf{y}_2; \mathbf{x}, \Sigma_{\mathbf{Y}_2}) = \mathcal{N}(\mathbf{x}; \mathbf{y}_2, \Sigma_{\mathbf{Y}_2})$  and  $\mathbf{Y}_1 \perp\!\!\!\perp \mathbf{Y}_2 | \mathbf{X}$ , we have

$$\mathcal{N}(\mathbf{y}_1; \mathbf{x}, \Sigma_{\mathbf{Y}_1}) \mathcal{N}(\mathbf{y}_2; \mathbf{x}, \Sigma_{\mathbf{Y}_2}) = \underbrace{\mathcal{N}(\mathbf{y}_1; \mathbf{x}, \Sigma_{\mathbf{Y}_1})}_{p(\mathbf{y}_1 | \mathbf{x}) = p(\mathbf{y}_1 | \mathbf{x}, \mathbf{y}_2)} \underbrace{\mathcal{N}(\mathbf{x}; \mathbf{y}_2, \Sigma_{\mathbf{Y}_2})}_{p(\mathbf{x} | \mathbf{y}_2)} = p(\mathbf{y}_1, \mathbf{x} | \mathbf{y}_2)$$

# Product of normal distributions

- Then, marginalizing  $\mathbf{x}$  out from  $p(\mathbf{y}_1, \mathbf{x}|\mathbf{y}_2)$ , we have

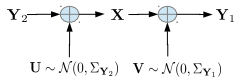
$p(\mathbf{y}_1|\mathbf{y}_2) = \int p(\mathbf{y}_1, \mathbf{x}|\mathbf{y}_2) d\mathbf{x}$ . However, from the figure,



$$\int p(\mathbf{y}_1, \mathbf{x}|\mathbf{y}_2) d\mathbf{x} = p(\mathbf{y}_1|\mathbf{y}_2) = \mathcal{N}(\mathbf{y}_1; \mathbf{y}_2, \Sigma_{Y_2} + \Sigma_{Y_1})$$

# Product of normal distributions

- Then, marginalizing  $\mathbf{x}$  out from  $p(\mathbf{y}_1, \mathbf{x}|\mathbf{y}_2)$ , we have



$p(\mathbf{y}_1|\mathbf{y}_2) = \int p(\mathbf{y}_1, \mathbf{x}|\mathbf{y}_2)d\mathbf{x}$ . However, from the figure,

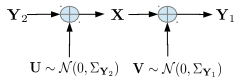
$$\int p(\mathbf{y}_1, \mathbf{x}|\mathbf{y}_2)d\mathbf{x} = p(\mathbf{y}_1|\mathbf{y}_2) = \mathcal{N}(\mathbf{y}_1; \mathbf{y}_2, \Sigma_{\mathbf{Y}_2} + \Sigma_{\mathbf{Y}_1})$$

- On the other hand,

$$\begin{aligned} \int p(\mathbf{y}_1, \mathbf{x}|\mathbf{y}_2)d\mathbf{x} &= \int \mathcal{N}(\mathbf{y}_1; \mathbf{x}, \Sigma_{\mathbf{Y}_1})\mathcal{N}(\mathbf{y}_2; \mathbf{x}, \Sigma_{\mathbf{Y}_2})d\mathbf{x} \\ &= \int K(\mathbf{y}_1, \mathbf{y}_2, \Sigma_{\mathbf{Y}_1}, \Sigma_{\mathbf{Y}_2})\mathcal{N}(\mathbf{x}; (\Lambda_{\mathbf{Y}_1} + \Lambda_{\mathbf{Y}_2})^{-1}(\Lambda_{\mathbf{Y}_2}\mathbf{y}_2 + \Lambda_{\mathbf{Y}_1}\mathbf{y}_1), (\Lambda_{\mathbf{Y}_2} + \Lambda_{\mathbf{Y}_1})^{-1})d\mathbf{x} \\ &= K(\mathbf{y}_1, \mathbf{y}_2, \Sigma_{\mathbf{Y}_1}, \Sigma_{\mathbf{Y}_2}). \end{aligned}$$

# Product of normal distributions

- Then, marginalizing  $\mathbf{x}$  out from  $p(\mathbf{y}_1, \mathbf{x}|\mathbf{y}_2)$ , we have



$p(\mathbf{y}_1|\mathbf{y}_2) = \int p(\mathbf{y}_1, \mathbf{x}|\mathbf{y}_2)d\mathbf{x}$ . However, from the figure,

$$\int p(\mathbf{y}_1, \mathbf{x}|\mathbf{y}_2)d\mathbf{x} = p(\mathbf{y}_1|\mathbf{y}_2) = \mathcal{N}(\mathbf{y}_1; \mathbf{y}_2, \Sigma_{\mathbf{Y}_2} + \Sigma_{\mathbf{Y}_1})$$

- On the other hand,

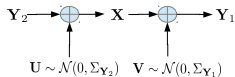
$$\begin{aligned} \int p(\mathbf{y}_1, \mathbf{x}|\mathbf{y}_2)d\mathbf{x} &= \int \mathcal{N}(\mathbf{y}_1; \mathbf{x}, \Sigma_{\mathbf{Y}_1})\mathcal{N}(\mathbf{y}_2; \mathbf{x}, \Sigma_{\mathbf{Y}_2})d\mathbf{x} \\ &= \int K(\mathbf{y}_1, \mathbf{y}_2, \Sigma_{\mathbf{Y}_1}, \Sigma_{\mathbf{Y}_2})\mathcal{N}(\mathbf{x}; (\Lambda_{\mathbf{Y}_1} + \Lambda_{\mathbf{Y}_2})^{-1}(\Lambda_{\mathbf{Y}_2}\mathbf{y}_2 + \Lambda_{\mathbf{Y}_1}\mathbf{y}_1), (\Lambda_{\mathbf{Y}_2} + \Lambda_{\mathbf{Y}_1})^{-1})d\mathbf{x} \\ &= K(\mathbf{y}_1, \mathbf{y}_2, \Sigma_{\mathbf{Y}_1}, \Sigma_{\mathbf{Y}_2}). \end{aligned}$$

- Thus we have  $K(\mathbf{y}_1, \mathbf{y}_2, \Sigma_{\mathbf{Y}_1}, \Sigma_{\mathbf{Y}_2}) = \mathcal{N}(\mathbf{y}_1; \mathbf{y}_2, \Sigma_{\mathbf{Y}_2} + \Sigma_{\mathbf{Y}_1})$



# Product of normal distributions

- Then, marginalizing  $\mathbf{x}$  out from  $p(\mathbf{y}_1, \mathbf{x}|\mathbf{y}_2)$ , we have



$p(\mathbf{y}_1|\mathbf{y}_2) = \int p(\mathbf{y}_1, \mathbf{x}|\mathbf{y}_2) d\mathbf{x}$ . However, from the figure,

$$\int p(\mathbf{y}_1, \mathbf{x}|\mathbf{y}_2) d\mathbf{x} = p(\mathbf{y}_1|\mathbf{y}_2) = \mathcal{N}(\mathbf{y}_1; \mathbf{y}_2, \Sigma_{\mathbf{Y}_2} + \Sigma_{\mathbf{Y}_1})$$

- On the other hand,

$$\begin{aligned} \int p(\mathbf{y}_1, \mathbf{x}|\mathbf{y}_2) d\mathbf{x} &= \int \mathcal{N}(\mathbf{y}_1; \mathbf{x}, \Sigma_{\mathbf{Y}_1}) \mathcal{N}(\mathbf{y}_2; \mathbf{x}, \Sigma_{\mathbf{Y}_2}) d\mathbf{x} \\ &= \int K(\mathbf{y}_1, \mathbf{y}_2, \Sigma_{\mathbf{Y}_1}, \Sigma_{\mathbf{Y}_2}) \mathcal{N}(\mathbf{x}; (\Lambda_{\mathbf{Y}_1} + \Lambda_{\mathbf{Y}_2})^{-1}(\Lambda_{\mathbf{Y}_2} \mathbf{y}_2 + \Lambda_{\mathbf{Y}_1} \mathbf{y}_1), (\Lambda_{\mathbf{Y}_2} + \Lambda_{\mathbf{Y}_1})^{-1}) d\mathbf{x} \\ &= K(\mathbf{y}_1, \mathbf{y}_2, \Sigma_{\mathbf{Y}_1}, \Sigma_{\mathbf{Y}_2}). \end{aligned}$$

- Thus we have  $K(\mathbf{y}_1, \mathbf{y}_2, \Sigma_{\mathbf{Y}_1}, \Sigma_{\mathbf{Y}_2}) = \mathcal{N}(\mathbf{y}_1; \mathbf{y}_2, \Sigma_{\mathbf{Y}_2} + \Sigma_{\mathbf{Y}_1})$  and so

$$\begin{aligned} &\mathcal{N}(\mathbf{y}_1; \mathbf{x}, \Sigma_{\mathbf{Y}_1}) \mathcal{N}(\mathbf{y}_2; \mathbf{x}, \Sigma_{\mathbf{Y}_2}) \\ &= \mathcal{N}(\mathbf{y}_1; \mathbf{y}_2, \Sigma_{\mathbf{Y}_2} + \Sigma_{\mathbf{Y}_1}) \mathcal{N}(\mathbf{x}; (\Lambda_{\mathbf{Y}_1} + \Lambda_{\mathbf{Y}_2})^{-1}(\Lambda_{\mathbf{Y}_2} \mathbf{y}_2 + \Lambda_{\mathbf{Y}_1} \mathbf{y}_1), (\Lambda_{\mathbf{Y}_2} + \Lambda_{\mathbf{Y}_1})^{-1}) \end{aligned}$$

# Product of normal distributions

Let us try to interpret the product as the overall likelihood after making two observations. Consider the simpler case when  $\mathbf{X}$ ,  $\mathbf{Y}_1$  and  $\mathbf{Y}_2$  are all scalar

# Product of normal distributions

Let us try to interpret the product as the overall likelihood after making two observations. Consider the simpler case when  $\mathbf{X}$ ,  $\mathbf{Y}_1$  and  $\mathbf{Y}_2$  are all scalar

- The mean considering both observations,  $(\Lambda_{\mathbf{Y}_1} + \Lambda_{\mathbf{Y}_2})^{-1}(\Lambda_{\mathbf{Y}_2}\mathbf{y}_2 + \Lambda_{\mathbf{Y}_1}y)$ , is essentially a weighted average of observations  $\mathbf{y}_2$  and  $\mathbf{y}_1$ 
  - The weight is higher when the precision  $\Lambda_{\mathbf{Y}_2}$  or  $\Lambda_{\mathbf{Y}_1}$  is larger

# Product of normal distributions

Let us try to interpret the product as the overall likelihood after making two observations. Consider the simpler case when  $\mathbf{X}$ ,  $\mathbf{Y}_1$  and  $\mathbf{Y}_2$  are all scalar

- The mean considering both observations,  $(\Lambda_{\mathbf{Y}_1} + \Lambda_{\mathbf{Y}_2})^{-1}(\Lambda_{\mathbf{Y}_2}y_2 + \Lambda_{\mathbf{Y}_1}y_1)$ , is essentially a weighted average of observations  $y_2$  and  $y_1$ 
  - The weight is higher when the precision  $\Lambda_{\mathbf{Y}_2}$  or  $\Lambda_{\mathbf{Y}_1}$  is larger
- The overall variance  $(\Lambda_{\mathbf{Y}_2} + \Lambda_{\mathbf{Y}_1})^{-1}$  is always smaller than the individual variance  $\Sigma_{\mathbf{Y}_2}$  and  $\Sigma_{\mathbf{Y}_1}$ 
  - We are more certain with  $\mathbf{x}$  after considering both  $y_1$  and  $y_2$

# Product of normal distributions

Let us try to interpret the product as the overall likelihood after making two observations. Consider the simpler case when  $\mathbf{X}$ ,  $\mathbf{Y}_1$  and  $\mathbf{Y}_2$  are all scalar

- The mean considering both observations,  $(\Lambda_{\mathbf{Y}_1} + \Lambda_{\mathbf{Y}_2})^{-1}(\Lambda_{\mathbf{Y}_2}\mathbf{y}_2 + \Lambda_{\mathbf{Y}_1}\mathbf{y}_1)$ , is essentially a weighted average of observations  $\mathbf{y}_2$  and  $\mathbf{y}_1$ 
  - The weight is higher when the precision  $\Lambda_{\mathbf{Y}_2}$  or  $\Lambda_{\mathbf{Y}_1}$  is larger
- The overall variance  $(\Lambda_{\mathbf{Y}_2} + \Lambda_{\mathbf{Y}_1})^{-1}$  is always smaller than the individual variance  $\Sigma_{\mathbf{Y}_2}$  and  $\Sigma_{\mathbf{Y}_1}$ 
  - We are more certain with  $\mathbf{x}$  after considering both  $\mathbf{y}_1$  and  $\mathbf{y}_2$
- The scaling factor,  $\mathcal{N}(\mathbf{y}_1; \mathbf{y}_2, \Sigma_{\mathbf{Y}_2} + \Sigma_{\mathbf{Y}_1})$ , can be interpreted as how much one can believe on the overall likelihood.
  - The value is reasonable since when the two observations are far away with respect to the overall variance  $\Sigma_{\mathbf{Y}_2} + \Sigma_{\mathbf{Y}_1}$ , the likelihood will become less reliable
  - The scaling factor is especially useful when we deal with mixture of Gaussian to be discussed next

## Review

- PCA (assume zero mean)
  - Via eigen-decomposition
    - 1  $\Sigma \approx \frac{1}{m} \mathcal{X}^T \mathcal{X}$
    - 2  $P^T \Sigma P = D$
    - 3  $Y = P^T X$
  - Via SVD
    - 1  $U^T \mathcal{X} V = D$
    - 2  $Y = V^T X$
- Marginalization of a normal distribution is still a normal distribution

- Conditioning of normal distribution:

$$\mathbf{X}|\mathbf{y} \sim \mathcal{N}(\boldsymbol{\mu}_X + \Sigma_{XY} \Sigma_{YY}^{-1}(\mathbf{y} - \boldsymbol{\mu}_Y), \Sigma_{XX} - \Sigma_{XY} \Sigma_{YY}^{-1} \Sigma_{YX})$$

- Product of normal distribution:

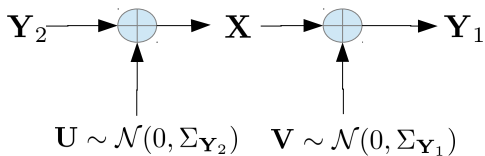
$$\mathcal{N}(\mathbf{y}_1; \mathbf{x}, \Sigma_{Y_1}) \mathcal{N}(\mathbf{y}_2; \mathbf{x}, \Sigma_{Y_2}) = \mathcal{N}(\mathbf{y}_1; \mathbf{y}_2, \Sigma_{Y_2} + \Sigma_{Y_1}) \mathcal{N}(\mathbf{x}; (\Lambda_{Y_1} + \Lambda_{Y_2})^{-1}(\Lambda_{Y_2} \mathbf{y}_2 + \Lambda_{Y_1} \mathbf{y}_1), (\Lambda_{Y_2} + \Lambda_{Y_1})^{-1})$$

# Correction: product of normal distributions

- One can compute the scaling factor  $K(\mathbf{y}_1, \mathbf{y}_2, \Sigma_{\mathbf{Y}_1}, \Sigma_{\mathbf{Y}_2})$  directly

# Correction: product of normal distributions

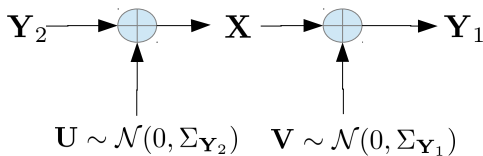
- One can compute the scaling factor  $K(\mathbf{y}_1, \mathbf{y}_2, \Sigma_{\mathbf{Y}_1}, \Sigma_{\mathbf{Y}_2})$  directly
- However, recall that  $\mathbf{Y}_1 \perp\!\!\!\perp \mathbf{Y}_2 | \mathbf{X}$ , it is model the variables as shown below





## Correction: product of normal distributions

- One can compute the scaling factor  $K(\mathbf{y}_1, \mathbf{y}_2, \Sigma_{\mathbf{Y}_1}, \Sigma_{\mathbf{Y}_2})$  directly
- However, recall that  $\mathbf{Y}_1 \perp\!\!\!\perp \mathbf{Y}_2 | \mathbf{X}$ , it is model the variables as shown below



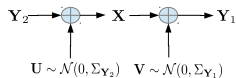
- Since  $\mathcal{N}(\mathbf{y}_2; \mathbf{x}, \Sigma_{\mathbf{Y}_2}) = \mathcal{N}(\mathbf{x}; \mathbf{y}_2, \Sigma_{\mathbf{Y}_2})$  and  $\mathbf{Y}_1 \perp\!\!\!\perp \mathbf{Y}_2 | \mathbf{X}$ , we have

$$\mathcal{N}(\mathbf{y}_1; \mathbf{x}, \Sigma_{\mathbf{Y}_1}) \mathcal{N}(\mathbf{y}_2; \mathbf{x}, \Sigma_{\mathbf{Y}_2}) = \underbrace{\mathcal{N}(\mathbf{y}_1; \mathbf{x}, \Sigma_{\mathbf{Y}_1})}_{p(\mathbf{y}_1 | \mathbf{x}) = p(\mathbf{y}_1 | \mathbf{x}, \mathbf{y}_2)} \underbrace{\mathcal{N}(\mathbf{x}; \mathbf{y}_2, \Sigma_{\mathbf{Y}_2})}_{p(\mathbf{x} | \mathbf{y}_2)} = p(\mathbf{y}_1, \mathbf{x} | \mathbf{y}_2)$$

# Correction: product of normal distributions

- Then, marginalizing  $\mathbf{x}$  out from  $p(\mathbf{y}_1, \mathbf{x}|\mathbf{y}_2)$ , we have

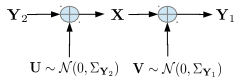
$p(\mathbf{y}_1|\mathbf{y}_2) = \int p(\mathbf{y}_1, \mathbf{x}|\mathbf{y}_2) d\mathbf{x}$ . However, from the figure,



$$\int p(\mathbf{y}_1, \mathbf{x}|\mathbf{y}_2) d\mathbf{x} = p(\mathbf{y}_1|\mathbf{y}_2) = \mathcal{N}(\mathbf{y}_1; \mathbf{y}_2, \Sigma_{Y_2} + \Sigma_{Y_1})$$

# Correction: product of normal distributions

- Then, marginalizing  $\mathbf{x}$  out from  $p(\mathbf{y}_1, \mathbf{x}|\mathbf{y}_2)$ , we have



$p(\mathbf{y}_1|\mathbf{y}_2) = \int p(\mathbf{y}_1, \mathbf{x}|\mathbf{y}_2)d\mathbf{x}$ . However, from the figure,

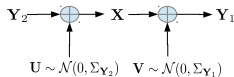
$$\int p(\mathbf{y}_1, \mathbf{x}|\mathbf{y}_2)d\mathbf{x} = p(\mathbf{y}_1|\mathbf{y}_2) = \mathcal{N}(\mathbf{y}_1; \mathbf{y}_2, \Sigma_{\mathbf{Y}_2} + \Sigma_{\mathbf{Y}_1})$$

- On the other hand,

$$\begin{aligned} \int p(\mathbf{y}_1, \mathbf{x}|\mathbf{y}_2)d\mathbf{x} &= \int \mathcal{N}(\mathbf{y}_1; \mathbf{x}, \Sigma_{\mathbf{Y}_1})\mathcal{N}(\mathbf{y}_2; \mathbf{x}, \Sigma_{\mathbf{Y}_2})d\mathbf{x} \\ &= \int K(\mathbf{y}_1, \mathbf{y}_2, \Sigma_{\mathbf{Y}_1}, \Sigma_{\mathbf{Y}_2})\mathcal{N}(\mathbf{x}; (\Lambda_{\mathbf{Y}_1} + \Lambda_{\mathbf{Y}_2})^{-1}(\Lambda_{\mathbf{Y}_2}\mathbf{y}_2 + \Lambda_{\mathbf{Y}_1}\mathbf{y}_1), (\Lambda_{\mathbf{Y}_2} + \Lambda_{\mathbf{Y}_1})^{-1})d\mathbf{x} \\ &= K(\mathbf{y}_1, \mathbf{y}_2, \Sigma_{\mathbf{Y}_1}, \Sigma_{\mathbf{Y}_2}). \end{aligned}$$

## Correction: product of normal distributions

- Then, marginalizing  $\mathbf{x}$  out from  $p(\mathbf{y}_1, \mathbf{x}|\mathbf{y}_2)$ , we have



$p(\mathbf{y}_1|\mathbf{y}_2) = \int p(\mathbf{y}_1, \mathbf{x}|\mathbf{y}_2)d\mathbf{x}$ . However, from the figure,

$$\int p(\mathbf{y}_1, \mathbf{x}|\mathbf{y}_2)d\mathbf{x} = p(\mathbf{y}_1|\mathbf{y}_2) = \mathcal{N}(\mathbf{y}_1; \mathbf{y}_2, \Sigma_{\mathbf{Y}_2} + \Sigma_{\mathbf{Y}_1})$$

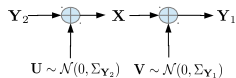
- On the other hand,

$$\begin{aligned} \int p(\mathbf{y}_1, \mathbf{x}|\mathbf{y}_2)d\mathbf{x} &= \int \mathcal{N}(\mathbf{y}_1; \mathbf{x}, \Sigma_{\mathbf{Y}_1})\mathcal{N}(\mathbf{y}_2; \mathbf{x}, \Sigma_{\mathbf{Y}_2})d\mathbf{x} \\ &= \int K(\mathbf{y}_1, \mathbf{y}_2, \Sigma_{\mathbf{Y}_1}, \Sigma_{\mathbf{Y}_2})\mathcal{N}(\mathbf{x}; (\Lambda_{\mathbf{Y}_1} + \Lambda_{\mathbf{Y}_2})^{-1}(\Lambda_{\mathbf{Y}_2}\mathbf{y}_2 + \Lambda_{\mathbf{Y}_1}\mathbf{y}_1), (\Lambda_{\mathbf{Y}_2} + \Lambda_{\mathbf{Y}_1})^{-1})d\mathbf{x} \\ &= K(\mathbf{y}_1, \mathbf{y}_2, \Sigma_{\mathbf{Y}_1}, \Sigma_{\mathbf{Y}_2}). \end{aligned}$$

- Thus we have  $K(\mathbf{y}_1, \mathbf{y}_2, \Sigma_{\mathbf{Y}_1}, \Sigma_{\mathbf{Y}_2}) = \mathcal{N}(\mathbf{y}_1; \mathbf{y}_2, \Sigma_{\mathbf{Y}_2} + \Sigma_{\mathbf{Y}_1})$

## Correction: product of normal distributions

- Then, marginalizing  $\mathbf{x}$  out from  $p(\mathbf{y}_1, \mathbf{x}|\mathbf{y}_2)$ , we have



$p(\mathbf{y}_1|\mathbf{y}_2) = \int p(\mathbf{y}_1, \mathbf{x}|\mathbf{y}_2)d\mathbf{x}$ . However, from the figure,

$$\int p(\mathbf{y}_1, \mathbf{x}|\mathbf{y}_2)d\mathbf{x} = p(\mathbf{y}_1|\mathbf{y}_2) = \mathcal{N}(\mathbf{y}_1; \mathbf{y}_2, \Sigma_{\mathbf{Y}_2} + \Sigma_{\mathbf{Y}_1})$$

- On the other hand,

$$\begin{aligned} \int p(\mathbf{y}_1, \mathbf{x}|\mathbf{y}_2)d\mathbf{x} &= \int \mathcal{N}(\mathbf{y}_1; \mathbf{x}, \Sigma_{\mathbf{Y}_1})\mathcal{N}(\mathbf{y}_2; \mathbf{x}, \Sigma_{\mathbf{Y}_2})d\mathbf{x} \\ &= \int K(\mathbf{y}_1, \mathbf{y}_2, \Sigma_{\mathbf{Y}_1}, \Sigma_{\mathbf{Y}_2})\mathcal{N}(\mathbf{x}; (\Lambda_{\mathbf{Y}_1} + \Lambda_{\mathbf{Y}_2})^{-1}(\Lambda_{\mathbf{Y}_2}\mathbf{y}_2 + \Lambda_{\mathbf{Y}_1}\mathbf{y}_1), (\Lambda_{\mathbf{Y}_2} + \Lambda_{\mathbf{Y}_1})^{-1})d\mathbf{x} \\ &= K(\mathbf{y}_1, \mathbf{y}_2, \Sigma_{\mathbf{Y}_1}, \Sigma_{\mathbf{Y}_2}). \end{aligned}$$

- Thus we have  $K(\mathbf{y}_1, \mathbf{y}_2, \Sigma_{\mathbf{Y}_1}, \Sigma_{\mathbf{Y}_2}) = \mathcal{N}(\mathbf{y}_1; \mathbf{y}_2, \Sigma_{\mathbf{Y}_2} + \Sigma_{\mathbf{Y}_1})$  and so

$$\begin{aligned} &\mathcal{N}(\mathbf{y}_1; \mathbf{x}, \Sigma_{\mathbf{Y}_1})\mathcal{N}(\mathbf{y}_2; \mathbf{x}, \Sigma_{\mathbf{Y}_2}) \\ &= \mathcal{N}(\mathbf{y}_1; \mathbf{y}_2, \Sigma_{\mathbf{Y}_2} + \Sigma_{\mathbf{Y}_1})\mathcal{N}(\mathbf{x}; (\Lambda_{\mathbf{Y}_1} + \Lambda_{\mathbf{Y}_2})^{-1}(\Lambda_{\mathbf{Y}_2}\mathbf{y}_2 + \Lambda_{\mathbf{Y}_1}\mathbf{y}_1), (\Lambda_{\mathbf{Y}_2} + \Lambda_{\mathbf{Y}_1})^{-1}) \end{aligned}$$

# Division of normal distributions

- To compute  $\frac{\mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)}{\mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2)}$ , note that from the product formula earlier

$$\begin{aligned} & \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2) \mathcal{N}(\mathbf{x}; (\boldsymbol{\Lambda}_1 - \boldsymbol{\Lambda}_2)^{-1}(\boldsymbol{\Lambda}_1 \boldsymbol{\mu}_1 - \boldsymbol{\Lambda}_2 \boldsymbol{\mu}_2), (\boldsymbol{\Lambda}_1 - \boldsymbol{\Lambda}_2)^{-1}) \\ &= \mathcal{N}(\boldsymbol{\mu}_2; (\boldsymbol{\Lambda}_1 - \boldsymbol{\Lambda}_2)^{-1}(\boldsymbol{\Lambda}_1 \boldsymbol{\mu}_1 - \boldsymbol{\Lambda}_2 \boldsymbol{\mu}_2), \boldsymbol{\Lambda}_2^{-1} + (\boldsymbol{\Lambda}_1 - \boldsymbol{\Lambda}_2)^{-1}) \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1) \end{aligned}$$

# Division of normal distributions

- To compute  $\frac{\mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)}{\mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2)}$ , note that from the product formula earlier

$$\begin{aligned} & \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2) \mathcal{N}(\mathbf{x}; (\boldsymbol{\Lambda}_1 - \boldsymbol{\Lambda}_2)^{-1}(\boldsymbol{\Lambda}_1 \boldsymbol{\mu}_1 - \boldsymbol{\Lambda}_2 \boldsymbol{\mu}_2), (\boldsymbol{\Lambda}_1 - \boldsymbol{\Lambda}_2)^{-1}) \\ &= \mathcal{N}(\boldsymbol{\mu}_2; (\boldsymbol{\Lambda}_1 - \boldsymbol{\Lambda}_2)^{-1}(\boldsymbol{\Lambda}_1 \boldsymbol{\mu}_1 - \boldsymbol{\Lambda}_2 \boldsymbol{\mu}_2), \boldsymbol{\Lambda}_2^{-1} + (\boldsymbol{\Lambda}_1 - \boldsymbol{\Lambda}_2)^{-1}) \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1) \end{aligned}$$

- Therefore,

$$\begin{aligned} \frac{\mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)}{\mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2)} &= \frac{\mathcal{N}(\mathbf{x}; (\boldsymbol{\Lambda}_1 - \boldsymbol{\Lambda}_2)^{-1}(\boldsymbol{\Lambda}_1 \boldsymbol{\mu}_1 - \boldsymbol{\Lambda}_2 \boldsymbol{\mu}_2), (\boldsymbol{\Lambda}_1 - \boldsymbol{\Lambda}_2)^{-1})}{\mathcal{N}(\boldsymbol{\mu}_2; (\boldsymbol{\Lambda}_1 - \boldsymbol{\Lambda}_2)^{-1}(\boldsymbol{\Lambda}_1 \boldsymbol{\mu}_1 - \boldsymbol{\Lambda}_2 \boldsymbol{\mu}_2), \boldsymbol{\Lambda}_2^{-1} + (\boldsymbol{\Lambda}_1 - \boldsymbol{\Lambda}_2)^{-1})} \\ &= \frac{\mathcal{N}(\mathbf{x}; \boldsymbol{\mu}, (\boldsymbol{\Lambda}_1 - \boldsymbol{\Lambda}_2)^{-1})}{\mathcal{N}(\boldsymbol{\mu}_2; \boldsymbol{\mu}, \boldsymbol{\Lambda}_2^{-1} + (\boldsymbol{\Lambda}_1 - \boldsymbol{\Lambda}_2)^{-1})}, \end{aligned}$$

where  $\boldsymbol{\mu} = (\boldsymbol{\Lambda}_1 - \boldsymbol{\Lambda}_2)^{-1}(\boldsymbol{\Lambda}_1 \boldsymbol{\mu}_1 - \boldsymbol{\Lambda}_2 \boldsymbol{\mu}_2)$

# Division of normal distributions

- To compute  $\frac{\mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)}{\mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2)}$ , note that from the product formula earlier

$$\begin{aligned} & \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2) \mathcal{N}(\mathbf{x}; (\boldsymbol{\Lambda}_1 - \boldsymbol{\Lambda}_2)^{-1}(\boldsymbol{\Lambda}_1 \boldsymbol{\mu}_1 - \boldsymbol{\Lambda}_2 \boldsymbol{\mu}_2), (\boldsymbol{\Lambda}_1 - \boldsymbol{\Lambda}_2)^{-1}) \\ &= \mathcal{N}(\boldsymbol{\mu}_2; (\boldsymbol{\Lambda}_1 - \boldsymbol{\Lambda}_2)^{-1}(\boldsymbol{\Lambda}_1 \boldsymbol{\mu}_1 - \boldsymbol{\Lambda}_2 \boldsymbol{\mu}_2), \boldsymbol{\Lambda}_2^{-1} + (\boldsymbol{\Lambda}_1 - \boldsymbol{\Lambda}_2)^{-1}) \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1) \end{aligned}$$

- Therefore,

$$\begin{aligned} \frac{\mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)}{\mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2)} &= \frac{\mathcal{N}(\mathbf{x}; (\boldsymbol{\Lambda}_1 - \boldsymbol{\Lambda}_2)^{-1}(\boldsymbol{\Lambda}_1 \boldsymbol{\mu}_1 - \boldsymbol{\Lambda}_2 \boldsymbol{\mu}_2), (\boldsymbol{\Lambda}_1 - \boldsymbol{\Lambda}_2)^{-1})}{\mathcal{N}(\boldsymbol{\mu}_2; (\boldsymbol{\Lambda}_1 - \boldsymbol{\Lambda}_2)^{-1}(\boldsymbol{\Lambda}_1 \boldsymbol{\mu}_1 - \boldsymbol{\Lambda}_2 \boldsymbol{\mu}_2), \boldsymbol{\Lambda}_2^{-1} + (\boldsymbol{\Lambda}_1 - \boldsymbol{\Lambda}_2)^{-1})} \\ &= \frac{\mathcal{N}(\mathbf{x}; \boldsymbol{\mu}, (\boldsymbol{\Lambda}_1 - \boldsymbol{\Lambda}_2)^{-1})}{\mathcal{N}(\boldsymbol{\mu}_2; \boldsymbol{\mu}, \boldsymbol{\Lambda}_2^{-1} + (\boldsymbol{\Lambda}_1 - \boldsymbol{\Lambda}_2)^{-1})}, \end{aligned}$$

where  $\boldsymbol{\mu} = (\boldsymbol{\Lambda}_1 - \boldsymbol{\Lambda}_2)^{-1}(\boldsymbol{\Lambda}_1 \boldsymbol{\mu}_1 - \boldsymbol{\Lambda}_2 \boldsymbol{\mu}_2)$

- Note that the final pdf will be Gaussian-like if  $\boldsymbol{\Lambda}_1 \succeq \boldsymbol{\Lambda}_2$ . Otherwise, one can still write out the pdf using the precision matrix. But the covariance matrix will not be defined (Try plot some pdfs out yourselves)



# Mixture of Gaussians

Consider an electrical system that outputs signal of different statistics when it is on and off

- When the system is on, the output signal  $S$  behaves like  $\mathcal{N}(5, 1)$ .  
When the system is off,  $S$  behaves like  $\mathcal{N}(0, 1)$

# Mixture of Gaussians

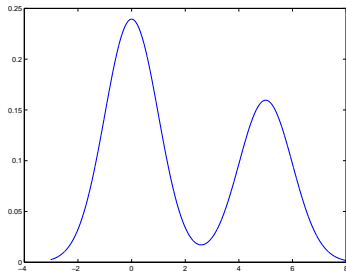
Consider an electrical system that outputs signal of different statistics when it is on and off

- When the system is on, the output signal  $S$  behaves like  $\mathcal{N}(5, 1)$ .  
When the system is off,  $S$  behaves like  $\mathcal{N}(0, 1)$
- If someone measuring the signal does not know the status of the system but only knows that the system is on 40% of the time, then to the observer, the signal  $S$  behaves like a mixture of Gaussians

# Mixture of Gaussians

Consider an electrical system that outputs signal of different statistics when it is on and off

- When the system is on, the output signal  $S$  behaves like  $\mathcal{N}(5, 1)$ . When the system is off,  $S$  behaves like  $\mathcal{N}(0, 1)$
- If someone measuring the signal does not know the status of the system but only knows that the system is on 40% of the time, then to the observer, the signal  $S$  behaves like a mixture of Gaussians
- The pdf of  $S$  will be  $0.4\mathcal{N}(s; 5, 1) + 0.6\mathcal{N}(s; 0, 1)$  as shown below



# Mixture of Gaussians

- A main limitation of normal distribution is that it is unimodal

# Mixture of Gaussians

- A main limitation of normal distribution is that it is unimodal
- Mixture of Gaussian distribution allows multimodal and can virtually model any pdfs. But there is a computational cost for this gain

# Mixture of Gaussians

- A main limitation of normal distribution is that it is unimodal
- Mixture of Gaussian distribution allows multimodal and can virtually model any pdfs. But there is a computational cost for this gain
- Let us illustrate this with the following example:
  - Consider two mixtures of Gaussian likelihood of  $x$  given two observations  $y_1$  and  $y_2$  as follows:

$$p(y_1|x) = 0.6\mathcal{N}(x; 0, 1) + 0.4\mathcal{N}(x; 5, 1);$$

$$p(y_2|x) = 0.5\mathcal{N}(x; -2, 1) + 0.5\mathcal{N}(x; 4, 1).$$

What is the overall likelihood,  $p(y_1, y_2|x)$ ?

# Mixture of Gaussians

- A main limitation of normal distribution is that it is unimodal
- Mixture of Gaussian distribution allows multimodal and can virtually model any pdfs. But there is a computational cost for this gain
- Let us illustrate this with the following example:
  - Consider two mixtures of Gaussian likelihood of  $x$  given two observations  $y_1$  and  $y_2$  as follows:

$$p(y_1|x) = 0.6\mathcal{N}(x; 0, 1) + 0.4\mathcal{N}(x; 5, 1);$$

$$p(y_2|x) = 0.5\mathcal{N}(x; -2, 1) + 0.5\mathcal{N}(x; 4, 1).$$

What is the overall likelihood,  $p(y_1, y_2|x)$ ?

- As usual, it is reasonable to assume the observations to be conditionally independent given  $x$ . Then,

$$\begin{aligned} p(y_1, y_2|x) &= p(y_1|x)p(y_2|x) \\ &= (0.6\mathcal{N}(x; 0, 1) + 0.4\mathcal{N}(x; 5, 1))(0.5\mathcal{N}(x; -2, 1) + 0.5\mathcal{N}(x; 4, 1)) \\ &= 0.3\mathcal{N}(x; 0, 1)\mathcal{N}(x; -2, 1) + 0.2\mathcal{N}(x; 5, 1)\mathcal{N}(x; -2, 1) \\ &\quad + 0.3\mathcal{N}(x; 0, 1)\mathcal{N}(x; 4, 1) + 0.2\mathcal{N}(x; 5, 1)\mathcal{N}(x; 4, 1) \end{aligned}$$

# Explosion of Gaussians

- The last step involves computing products of Gaussians but we have learned it in previous sections. Using the previous result,

$$p(y_1, y_2|x) = 0.3\mathcal{N}(-2; 0, 2)\mathcal{N}(x; -1, 0.5) + 0.2\mathcal{N}(-2; 5, 2)\mathcal{N}(x; 1.5, 0.5) \\ + 0.3\mathcal{N}(4; 0, 2)\mathcal{N}(x; 2, 0.5) + 0.2\mathcal{N}(4; 5, 2)\mathcal{N}(x; 4.5, 0.5).$$



# Explosion of Gaussians

- The last step involves computing products of Gaussians but we have learned it in previous sections. Using the previous result,

$$p(y_1, y_2|x) = 0.3\mathcal{N}(-2; 0, 2)\mathcal{N}(x; -1, 0.5) + 0.2\mathcal{N}(-2; 5, 2)\mathcal{N}(x; 1.5, 0.5) \\ + 0.3\mathcal{N}(4; 0, 2)\mathcal{N}(x; 2, 0.5) + 0.2\mathcal{N}(4; 5, 2)\mathcal{N}(x; 4.5, 0.5).$$

So we have the overall likelihood is a mixture of four Gaussians

# Explosion of Gaussians

- The last step involves computing products of Gaussians but we have learned it in previous sections. Using the previous result,

$$p(y_1, y_2|x) = 0.3\mathcal{N}(-2; 0, 2)\mathcal{N}(x; -1, 0.5) + 0.2\mathcal{N}(-2; 5, 2)\mathcal{N}(x; 1.5, 0.5) \\ + 0.3\mathcal{N}(4; 0, 2)\mathcal{N}(x; 2, 0.5) + 0.2\mathcal{N}(4; 5, 2)\mathcal{N}(x; 4.5, 0.5).$$

So we have the overall likelihood is a mixture of four Gaussians

- Let's repeat our discussion but with  $n$  observations instead. The overall likelihood will be a mixture of  $2^n$  Gaussians!
  - Therefore, the computation will quickly become intractable as the number of observations increases

# Explosion of Gaussians

- The last step involves computing products of Gaussians but we have learned it in previous sections. Using the previous result,

$$p(y_1, y_2|x) = 0.3\mathcal{N}(-2; 0, 2)\mathcal{N}(x; -1, 0.5) + 0.2\mathcal{N}(-2; 5, 2)\mathcal{N}(x; 1.5, 0.5) \\ + 0.3\mathcal{N}(4; 0, 2)\mathcal{N}(x; 2, 0.5) + 0.2\mathcal{N}(4; 5, 2)\mathcal{N}(x; 4.5, 0.5).$$

So we have the overall likelihood is a mixture of four Gaussians

- Let's repeat our discussion but with  $n$  observations instead. The overall likelihood will be a mixture of  $2^n$  Gaussians!
  - Therefore, the computation will quickly become intractable as the number of observations increases
  - Fortunately, in reality, some of the Gaussians in the mixture tend to have a very small weight

# Reduce number of components in Gaussian mixtures

- For instance, in our previous numerical example, if we continue our numerical computation for the two observation example, we have

$$p(y_1, y_2|x) = 0.4163\mathcal{N}(x; -1, 0.5) + 3.5234 \times 10^{-6}\mathcal{N}(x; 1.5, 0.5) \\ + 0.0202\mathcal{N}(x; 2, 0.5) + 0.5734\mathcal{N}(x; 4.5, 0.5).$$

# Reduce number of components in Gaussian mixtures

- For instance, in our previous numerical example, if we continue our numerical computation for the two observation example, we have

$$p(y_1, y_2 | x) = 0.4163\mathcal{N}(x; -1, 0.5) + 3.5234 \times 10^{-6}\mathcal{N}(x; 1.5, 0.5) \\ + 0.0202\mathcal{N}(x; 2, 0.5) + 0.5734\mathcal{N}(x; 4.5, 0.5).$$

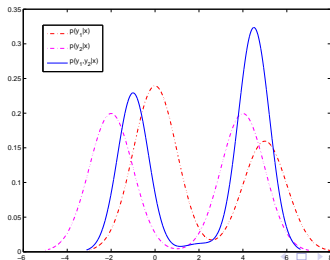
- We can see that the weight for the component at mean 1.5 is very small. And the component at mean 2 has a rather small weight also.

# Reduce number of components in Gaussian mixtures

- For instance, in our previous numerical example, if we continue our numerical computation for the two observation example, we have

$$p(y_1, y_2|x) = 0.4163\mathcal{N}(x; -1, 0.5) + 3.5234 \times 10^{-6}\mathcal{N}(x; 1.5, 0.5) \\ + 0.0202\mathcal{N}(x; 2, 0.5) + 0.5734\mathcal{N}(x; 4.5, 0.5).$$

- We can see that the weight for the component at mean 1.5 is very small. And the component at mean 2 has a rather small weight also.
- Even with the four Gaussian components, the overall likelihood is essentially just a bimodal distribution as shown in the figure below



# Reduce number of components in Gaussian mixtures

- Therefore, we may approximate  $p(y_1, y_2|x)$  with only two of its original component as  $0.4163/(0.4163 + 0.5734)\mathcal{N}(x; -1, 0.5) + 0.5734/(0.4163 + 0.5734)\mathcal{N}(x; 4.5, 0.5) = 0.4206\mathcal{N}(x; -1, 0.5) + 0.5794\mathcal{N}(x; 4.5, 0.5)$

# Reduce number of components in Gaussian mixtures

- Therefore, we may approximate  $p(y_1, y_2|x)$  with only two of its original component as  $0.4163/(0.4163 + 0.5734)\mathcal{N}(x; -1, 0.5) + 0.5734/(0.4163 + 0.5734)\mathcal{N}(x; 4.5, 0.5) = 0.4206\mathcal{N}(x; -1, 0.5) + 0.5794\mathcal{N}(x; 4.5, 0.5)$
- However, it is not always a good approximation strategy just to dump away the small components in a Gaussian mixture



# Another example

Consider

$$p(x) = 0.1\mathcal{N}(x; -0.2, 1) + 0.1\mathcal{N}(x; -0.1, 1) + 0.1\mathcal{N}(x; 0, 1) + 0.1\mathcal{N}(x; 0.1, 1) \\ + 0.1\mathcal{N}(x; 0.2, 1) + 0.5\mathcal{N}(x; 5, 1).$$

# Another example

Consider

$$p(x) = 0.1\mathcal{N}(x; -0.2, 1) + 0.1\mathcal{N}(x; -0.1, 1) + 0.1\mathcal{N}(x; 0, 1) + 0.1\mathcal{N}(x; 0.1, 1) \\ + 0.1\mathcal{N}(x; 0.2, 1) + 0.5\mathcal{N}(x; 5, 1).$$

- Let say we want to reduce  $p(x)$  to only a mixture of two Gaussians. It is tempting to just dumping four smallest one and renormalized the weight. For example, if we choose to remove the first four components, we have

$$\hat{p}(x) = 1/6\mathcal{N}(x; 0.2, 1) + 5/6\mathcal{N}(x; 5, 1)$$

# Another example

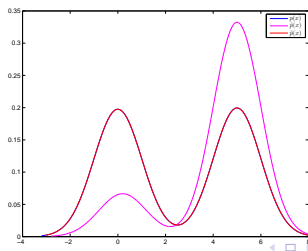
Consider

$$p(x) = 0.1\mathcal{N}(x; -0.2, 1) + 0.1\mathcal{N}(x; -0.1, 1) + 0.1\mathcal{N}(x; 0, 1) + 0.1\mathcal{N}(x; 0.1, 1) \\ + 0.1\mathcal{N}(x; 0.2, 1) + 0.5\mathcal{N}(x; 5, 1).$$

- Let say we want to reduce  $p(x)$  to only a mixture of two Gaussians. It is tempting to just dumping four smallest one and renormalized the weight. For example, if we choose to remove the first four components, we have

$$\hat{p}(x) = 1/6\mathcal{N}(x; 0.2, 1) + 5/6\mathcal{N}(x; 5, 1)$$

- The approximation  $\hat{p}(x)$  is significantly different from  $p(x)$  as shown below



# Merging components

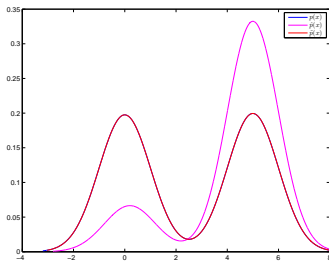
- The problem is that while the first five components are all relatively small compared to the last one, they are all quite similar and their combined contribution is comparable to the latter

# Merging components

- The problem is that while the first five components are all relatively small compared to the last one, they are all quite similar and their combined contribution is comparable to the latter
- Actually the first five components are so similar that their combined contribution can be accurately modeled as one Gaussian

# Merging components

- The problem is that while the first five components are all relatively small compared to the last one, they are all quite similar and their combined contribution is comparable to the latter
- Actually the first five components are so similar that their combined contribution can be accurately modeled as one Gaussian
- So rather than discarding the components, one can get a much more accurate approximation by merging them. The approximation is illustrated as  $\tilde{p}(x)$  in the figure below



# Merging components

To successfully obtain such approximation  $\tilde{p}(x)$ , we have to answer two questions:

- which components to merge?
- how to merge them?

# Which Components to Merge?

It is reasonable to pick similar components to merge. The question is how do we gauge the similarity between two components.



# Which Components to Merge?

It is reasonable to pick similar components to merge. The question is how do we will gauge the similarity between two components.

- Consider two pdfs  $p(\mathbf{x})$  and  $q(\mathbf{x})$ , note that we can define an inner product of  $p(\mathbf{x})$  and  $q(\mathbf{x})$  by

$$\langle p(\mathbf{x}), q(\mathbf{x}) \rangle = \int p(\mathbf{x})q(\mathbf{x})d\mathbf{x}$$

# Which Components to Merge?

It is reasonable to pick similar components to merge. The question is how do we gauge the similarity between two components.

- Consider two pdfs  $p(\mathbf{x})$  and  $q(\mathbf{x})$ , note that we can define an inner product of  $p(\mathbf{x})$  and  $q(\mathbf{x})$  by

$$\langle p(\mathbf{x}), q(\mathbf{x}) \rangle = \int p(\mathbf{x})q(\mathbf{x})d\mathbf{x}$$

- Note that the inner product is well defined and  $\langle p(\mathbf{x}), p(\mathbf{x}) \rangle \geq 0$

# Which Components to Merge?

It is reasonable to pick similar components to merge. The question is how do we will gauge the similarity between two components.

- Consider two pdfs  $p(\mathbf{x})$  and  $q(\mathbf{x})$ , note that we can define an inner product of  $p(\mathbf{x})$  and  $q(\mathbf{x})$  by

$$\langle p(\mathbf{x}), q(\mathbf{x}) \rangle = \int p(\mathbf{x})q(\mathbf{x})d\mathbf{x}$$

- Note that the inner product is well defined and  $\langle p(\mathbf{x}), p(\mathbf{x}) \rangle \geq 0$
- By Cauchy-Schwartz inequality,

$$\frac{\langle p(\mathbf{x}), q(\mathbf{x}) \rangle}{\sqrt{\langle p(\mathbf{x}), p(\mathbf{x}) \rangle \langle q(\mathbf{x}), q(\mathbf{x}) \rangle}} = \frac{\int p(\mathbf{x})q(\mathbf{x})d\mathbf{x}}{\sqrt{\int p(\mathbf{x})^2d\mathbf{x} \int q(\mathbf{x})^2d\mathbf{x}}} \leq 1$$

# Which Components to Merge?

It is reasonable to pick similar components to merge. The question is how do we will gauge the similarity between two components.

- Consider two pdfs  $p(\mathbf{x})$  and  $q(\mathbf{x})$ , note that we can define an inner product of  $p(\mathbf{x})$  and  $q(\mathbf{x})$  by

$$\langle p(\mathbf{x}), q(\mathbf{x}) \rangle = \int p(\mathbf{x})q(\mathbf{x})d\mathbf{x}$$

- Note that the inner product is well defined and  $\langle p(\mathbf{x}), p(\mathbf{x}) \rangle \geq 0$
- By Cauchy-Schwartz inequality,

$$\frac{\langle p(\mathbf{x}), q(\mathbf{x}) \rangle}{\sqrt{\langle p(\mathbf{x}), p(\mathbf{x}) \rangle \langle q(\mathbf{x}), q(\mathbf{x}) \rangle}} = \frac{\int p(\mathbf{x})q(\mathbf{x})d\mathbf{x}}{\sqrt{\int p(\mathbf{x})^2d\mathbf{x} \int q(\mathbf{x})^2d\mathbf{x}}} \leq 1$$

- The inner product maximizes ( $= 1$ ) when  $p(\mathbf{x}) = q(\mathbf{x})$ . This suggests a very reasonable similarity measure between two pdfs

# Similarity measure

- Let's define

$$\text{Sim}(p(\mathbf{x}), q(\mathbf{x})) \triangleq \frac{\int p(\mathbf{x})q(\mathbf{x})d\mathbf{x}}{\sqrt{\int p(\mathbf{x})^2d\mathbf{x} \int q(\mathbf{x})^2d\mathbf{x}}}$$

# Similarity measure

- Let's define

$$\text{Sim}(p(\mathbf{x}), q(\mathbf{x})) \triangleq \frac{\int p(\mathbf{x})q(\mathbf{x})d\mathbf{x}}{\sqrt{\int p(\mathbf{x})^2d\mathbf{x} \int q(\mathbf{x})^2d\mathbf{x}}}$$

- In particular, if  $p(\mathbf{x}) = \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_p, \Sigma_p)$  and  $q(\mathbf{x}) = \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_q, \Sigma_q)$ , we have (please verify)

$$\text{Sim}(\mathcal{N}(\boldsymbol{\mu}_p, \Sigma_p), \mathcal{N}(\boldsymbol{\mu}_q, \Sigma_q)) = \frac{\mathcal{N}(\boldsymbol{\mu}_p; \boldsymbol{\mu}_q, \Sigma_p + \Sigma_q)}{\sqrt{\mathcal{N}(0; 0, 2\Sigma_p)\mathcal{N}(0; 0, 2\Sigma_q)}},$$

which can be computed very easily and is equal to one only when means and covariances are the same

# How to Merge Components?

Say we have  $n$  components  $\mathcal{N}(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1), \mathcal{N}(\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2), \dots, \mathcal{N}(\boldsymbol{\mu}_n, \boldsymbol{\Sigma}_n)$  with weights  $w_1, w_2, \dots, w_n$ . What should the combined component be like?

# How to Merge Components?

Say we have  $n$  components  $\mathcal{N}(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1), \mathcal{N}(\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2), \dots, \mathcal{N}(\boldsymbol{\mu}_n, \boldsymbol{\Sigma}_n)$  with weights  $w_1, w_2, \dots, w_n$ . What should the combined component be like?

- Combined component weight should equal to total weight  $\sum_{i=1}^n w_i$



# How to Merge Components?

Say we have  $n$  components  $\mathcal{N}(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1), \mathcal{N}(\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2), \dots, \mathcal{N}(\boldsymbol{\mu}_n, \boldsymbol{\Sigma}_n)$  with weights  $w_1, w_2, \dots, w_n$ . What should the combined component be like?

- Combined component weight should equal to total weight  $\sum_{i=1}^n w_i$
- Combined mean will simply be  $\sum_{i=1}^n \hat{w}_i \boldsymbol{\mu}_i$ , where  $\hat{w}_i = \frac{w_i}{\sum_{i=1}^n w_i}$

# How to Merge Components?

Say we have  $n$  components  $\mathcal{N}(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1), \mathcal{N}(\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2), \dots, \mathcal{N}(\boldsymbol{\mu}_n, \boldsymbol{\Sigma}_n)$  with weights  $w_1, w_2, \dots, w_n$ . What should the combined component be like?

- Combined component weight should equal to total weight  $\sum_{i=1}^n w_i$
- Combined mean will simply be  $\sum_{i=1}^n \hat{w}_i \boldsymbol{\mu}_i$ , where  $\hat{w}_i = \frac{w_i}{\sum_{i=1}^n w_i}$
- For combined covariance, it may be tempting to approximate it as  $\sum_{i=1}^n \hat{w}_i \boldsymbol{\Sigma}_i$ .

# How to Merge Components?

Say we have  $n$  components  $\mathcal{N}(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1), \mathcal{N}(\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2), \dots, \mathcal{N}(\boldsymbol{\mu}_n, \boldsymbol{\Sigma}_n)$  with weights  $w_1, w_2, \dots, w_n$ . What should the combined component be like?

- Combined component weight should equal to total weight  $\sum_{i=1}^n w_i$
- Combined mean will simply be  $\sum_{i=1}^n \hat{w}_i \boldsymbol{\mu}_i$ , where  $\hat{w}_i = \frac{w_i}{\sum_{i=1}^n w_i}$
- For combined covariance, it may be tempting to approximate it as  $\sum_{i=1}^n \hat{w}_i \boldsymbol{\Sigma}_i$ .
  - However, it is an underestimate

# How to Merge Components?

Say we have  $n$  components  $\mathcal{N}(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1), \mathcal{N}(\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2), \dots, \mathcal{N}(\boldsymbol{\mu}_n, \boldsymbol{\Sigma}_n)$  with weights  $w_1, w_2, \dots, w_n$ . What should the combined component be like?

- Combined component weight should equal to total weight  $\sum_{i=1}^n w_i$
- Combined mean will simply be  $\sum_{i=1}^n \hat{w}_i \boldsymbol{\mu}_i$ , where  $\hat{w}_i = \frac{w_i}{\sum_{i=1}^n w_i}$
- For combined covariance, it may be tempting to approximate it as  $\sum_{i=1}^n \hat{w}_i \boldsymbol{\Sigma}_i$ .
  - However, it is an underestimate
  - Because the weighted sum only counted the contribution of variation among each component, it did not take into account the variation due to different means across components.

# How to Merge Components?

Say we have  $n$  components  $\mathcal{N}(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1), \mathcal{N}(\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2), \dots, \mathcal{N}(\boldsymbol{\mu}_n, \boldsymbol{\Sigma}_n)$  with weights  $w_1, w_2, \dots, w_n$ . What should the combined component be like?

- Combined component weight should equal to total weight  $\sum_{i=1}^n w_i$
- Combined mean will simply be  $\sum_{i=1}^n \hat{w}_i \boldsymbol{\mu}_i$ , where  $\hat{w}_i = \frac{w_i}{\sum_{i=1}^n w_i}$
- For combined covariance, it may be tempting to approximate it as  $\sum_{i=1}^n \hat{w}_i \boldsymbol{\Sigma}_i$ .
  - However, it is an underestimate
  - Because the weighted sum only counted the contribution of variation among each component, it did not take into account the variation due to different means across components.
  - Instead, let's denote  $\mathbf{X}$  as the variable sampled from the mixture. That is,  $\mathbf{X} \sim \mathcal{N}(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$  with probability  $\hat{w}_i$ . Then, we have (please verify)

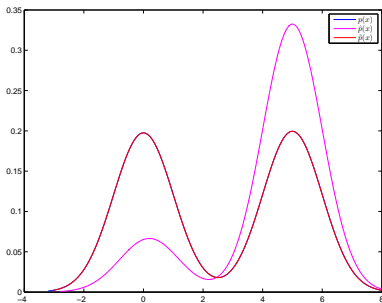
$$\begin{aligned} \boldsymbol{\Sigma} &= E[\mathbf{X}\mathbf{X}^T] - E[\mathbf{X}]E[\mathbf{X}]^T \\ &= \sum_{i=1}^n \hat{w}_i (\boldsymbol{\Sigma}_i + \boldsymbol{\mu}_i \boldsymbol{\mu}_i^T) - \sum_{i=1}^n \sum_{j=1}^n \hat{w}_i \hat{w}_j \boldsymbol{\mu}_i \boldsymbol{\mu}_j^T. \end{aligned}$$

Now, go back to our previous numerical example

- Recall that  $p(x) = 0.1\mathcal{N}(x; -0.2, 1) + 0.1\mathcal{N}(x; -0.1, 1) + 0.1\mathcal{N}(x; 0, 1) + 0.1\mathcal{N}(x; 0.1, 1) + 0.1\mathcal{N}(x; 0.2, 1) + 0.5\mathcal{N}(x; 5, 1)$

Now, go back to our previous numerical example

- Recall that  $p(x) = 0.1\mathcal{N}(x; -0.2, 1) + 0.1\mathcal{N}(x; -0.1, 1) + 0.1\mathcal{N}(x; 0, 1) + 0.1\mathcal{N}(x; 0.1, 1) + 0.1\mathcal{N}(x; 0.2, 1) + 0.5\mathcal{N}(x; 5, 1)$
- If we merge the five smallest components (one can easily check that they are also more similar to each other than to the last component), we have  $\tilde{p}(x) = 0.5\mathcal{N}(x; 0, 1.02) + 0.5\mathcal{N}(x; 5, 1)$  as shown again below. The approximate pdf is virtually indistinguishable from the original



# Review multivariate normal

- Marginalization of a normal distribution is still a normal distribution
- Conditioning of normal distribution:

$$\mathbf{X}|\mathbf{y} \sim \mathcal{N}(\boldsymbol{\mu}_X + \Sigma_{XY}\Sigma_{YY}^{-1}(\mathbf{y} - \boldsymbol{\mu}_Y), \Sigma_{XX} - \Sigma_{XY}\Sigma_{YY}^{-1}\Sigma_{YX})$$

- Product of normal distribution:

$$\begin{aligned} \mathcal{N}(\mathbf{y}_1; \mathbf{x}, \Sigma_{Y_1})\mathcal{N}(\mathbf{y}_2; \mathbf{x}, \Sigma_{Y_2}) = \\ \mathcal{N}(\mathbf{y}_1; \mathbf{y}_2, \Sigma_{Y_2} + \Sigma_{Y_1})\mathcal{N}(\mathbf{x}; (\Lambda_{Y_1} + \Lambda_{Y_2})^{-1}(\Lambda_{Y_2}\mathbf{y}_2 + \Lambda_{Y_1}\mathbf{y}_1), (\Lambda_{Y_2} + \Lambda_{Y_1})^{-1}) \end{aligned}$$

- Division of normal distribution:

$$\frac{\mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_1, \Sigma_1)}{\mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_2, \Sigma_2)} = \frac{\mathcal{N}(\mathbf{x}; \boldsymbol{\mu}, (\Lambda_1 - \Lambda_2)^{-1})}{\mathcal{N}(\boldsymbol{\mu}_2; \boldsymbol{\mu}, \Lambda_2^{-1} + (\Lambda_1 - \Lambda_2)^{-1})},$$

where  $\boldsymbol{\mu} = (\Lambda_1 - \Lambda_2)^{-1}(\Lambda_1\boldsymbol{\mu}_1 - \Lambda_2\boldsymbol{\mu}_2)$

- Similarity measure

$$\text{Sim}(\mathcal{N}(\boldsymbol{\mu}_p, \Sigma_p), \mathcal{N}(\boldsymbol{\mu}_q, \Sigma_q)) = \frac{\mathcal{N}(\boldsymbol{\mu}_p; \boldsymbol{\mu}_q, \Sigma_p + \Sigma_q)}{\sqrt{\mathcal{N}(0; 0, 2\Sigma_p)\mathcal{N}(0; 0, 2\Sigma_q)}},$$



# Bernoulli distribution

- Consider someone flips a biased coin. The probability of the outcome is described by the Bernoulli distribution. Denote  $X = 1$  for a head and  $X = 0$  for a tail. Let  $Pr(X = 1) = p$ .

# Bernoulli distribution

- Consider someone flips a biased coin. The probability of the outcome is described by the Bernoulli distribution. Denote  $X = 1$  for a head and  $X = 0$  for a tail. Let  $Pr(X = 1) = p$ . Then the Bernoulli distribution is simply

$$Bern(x|p) = \begin{cases} p, & x = 1 \\ 1 - p, & x = 0 \end{cases}$$

# Bernoulli distribution

- Consider someone flips a biased coin. The probability of the outcome is described by the Bernoulli distribution. Denote  $X = 1$  for a head and  $X = 0$  for a tail. Let  $Pr(X = 1) = p$ . Then the Bernoulli distribution is simply

$$Bern(x|p) = \begin{cases} p, & x = 1 \\ 1 - p, & x = 0 \end{cases}$$

- More concisely, we can write it as

$$Bern(x|p) = p^x(1 - p)^{1-x},$$

# Bernoulli distribution

- Consider someone flips a biased coin. The probability of the outcome is described by the Bernoulli distribution. Denote  $X = 1$  for a head and  $X = 0$  for a tail. Let  $Pr(X = 1) = p$ . Then the Bernoulli distribution is simply

$$\text{Bern}(x|p) = \begin{cases} p, & x = 1 \\ 1 - p, & x = 0 \end{cases}$$

- More concisely, we can write it as

$$\text{Bern}(x|p) = p^x(1 - p)^{1-x},$$

- The mean and variance are

$$E[X] = p \cdot 1 + (1 - p) \cdot 0 = p$$

$$\text{Var}[X] = p \cdot (1 - p)^2 + (1 - p) \cdot p^2 = p(1 - p)$$

# Binomial distribution ( $N$ trials)

- Repeat the experiment for  $N$  times, the probability of the outcome will now be described by the binomial distribution. Note that  $x$  is now the number of obtained heads, we have

# Binomial distribution ( $N$ trials)

- Repeat the experiment for  $N$  times, the probability of the outcome will now be described by the binomial distribution. Note that  $x$  is now the number of obtained heads, we have

$$\text{Bin}(x|p, N) = \binom{N}{x} p^x (1-p)^{N-x},$$

- Mean and variances are given by
  - $E[X] = \sum_{x=0}^N \text{Bin}(x|p)x$

# Binomial distribution ( $N$ trials)

- Repeat the experiment for  $N$  times, the probability of the outcome will now be described by the binomial distribution. Note that  $x$  is now the number of obtained heads, we have

$$\text{Bin}(x|p, N) = \binom{N}{x} p^x (1-p)^{N-x},$$

- Mean and variances are given by

- $E[X] = \sum_{x=0}^N \text{Bin}(x|p) x = \sum_{x=1}^N \frac{N!}{(x-1)!(N-x)!} p^x (1-p)^{N-x}$

# Binomial distribution ( $N$ trials)

- Repeat the experiment for  $N$  times, the probability of the outcome will now be described by the binomial distribution. Note that  $x$  is now the number of obtained heads, we have

$$\text{Bin}(x|p, N) = \binom{N}{x} p^x (1-p)^{N-x},$$

- Mean and variances are given by

$$\begin{aligned} E[X] &= \sum_{x=0}^N \text{Bin}(x|p) x = \sum_{x=1}^N \frac{N!}{(x-1)!(N-x)!} p^x (1-p)^{N-x} \\ &= Np \sum_{x=1}^N \frac{(N-1)!}{(x-1)!(N-x)!} p^{x-1} (1-p)^{N-x} \end{aligned}$$



# Binomial distribution ( $N$ trials)

- Repeat the experiment for  $N$  times, the probability of the outcome will now be described by the binomial distribution. Note that  $x$  is now the number of obtained heads, we have

$$\text{Bin}(x|p, N) = \binom{N}{x} p^x (1-p)^{N-x},$$

- Mean and variances are given by

$$\begin{aligned} E[X] &= \sum_{x=0}^N \text{Bin}(x|p) x = \sum_{x=1}^N \frac{N!}{(x-1)!(N-x)!} p^x (1-p)^{N-x} \\ &= Np \sum_{x=1}^N \frac{(N-1)!}{(x-1)!(N-x)!} p^{x-1} (1-p)^{N-x} = Np \sum_{x=0}^{N-1} \text{Bin}(x|p, N-1) \end{aligned}$$

# Binomial distribution ( $N$ trials)

- Repeat the experiment for  $N$  times, the probability of the outcome will now be described by the binomial distribution. Note that  $x$  is now the number of obtained heads, we have

$$\text{Bin}(x|p, N) = \binom{N}{x} p^x (1-p)^{N-x},$$

- Mean and variances are given by

$$\begin{aligned} E[X] &= \sum_{x=0}^N \text{Bin}(x|p) x = \sum_{x=1}^N \frac{N!}{(x-1)!(N-x)!} p^x (1-p)^{N-x} \\ &= Np \sum_{x=1}^N \frac{(N-1)!}{(x-1)!(N-x)!} p^{x-1} (1-p)^{N-x} = Np \sum_{x=0}^{N-1} \text{Bin}(x|p, N-1) \\ &= Np \end{aligned}$$

# Binomial distribution ( $N$ trials)

- Repeat the experiment for  $N$  times, the probability of the outcome will now be described by the binomial distribution. Note that  $x$  is now the number of obtained heads, we have

$$\text{Bin}(x|p, N) = \binom{N}{x} p^x (1-p)^{N-x},$$

- Mean and variances are given by

- $$E[X] = \sum_{x=0}^N \text{Bin}(x|p) x = \sum_{x=1}^N \frac{N!}{(x-1)!(N-x)!} p^x (1-p)^{N-x}$$

$$= Np \sum_{x=1}^N \frac{(N-1)!}{(x-1)!(N-x)!} p^{x-1} (1-p)^{N-x} = Np \sum_{x=0}^{N-1} \text{Bin}(x|p, N-1)$$

$$= Np$$
- Similar,  $E[X(X-1)] = \sum_{x=2}^N \frac{N!}{(x-2)!(N-x)!} p^x (1-p)^{N-x}$

# Binomial distribution ( $N$ trials)

- Repeat the experiment for  $N$  times, the probability of the outcome will now be described by the binomial distribution. Note that  $x$  is now the number of obtained heads, we have

$$\text{Bin}(x|p, N) = \binom{N}{x} p^x (1-p)^{N-x},$$

- Mean and variances are given by

- $$E[X] = \sum_{x=0}^N \text{Bin}(x|p) x = \sum_{x=1}^N \frac{N!}{(x-1)!(N-x)!} p^x (1-p)^{N-x}$$

$$= Np \sum_{x=1}^N \frac{(N-1)!}{(x-1)!(N-x)!} p^{x-1} (1-p)^{N-x} = Np \sum_{x=0}^{N-1} \text{Bin}(x|p, N-1)$$

$$= Np$$
- $$\text{Similar, } E[X(X-1)] = \sum_{x=2}^N \frac{N!}{(x-2)!(N-x)!} p^x (1-p)^{N-x}$$

$$= N(N-1)p^2 \sum_{x=0}^{N-2} \text{Bin}(x|p, N-2) = N(N-1)p^2$$

# Binomial distribution ( $N$ trials)

- Repeat the experiment for  $N$  times, the probability of the outcome will now be described by the binomial distribution. Note that  $x$  is now the number of obtained heads, we have

$$\text{Bin}(x|p, N) = \binom{N}{x} p^x (1-p)^{N-x},$$

- Mean and variances are given by

- $$E[X] = \sum_{x=0}^N \text{Bin}(x|p) x = \sum_{x=1}^N \frac{N!}{(x-1)!(N-x)!} p^x (1-p)^{N-x}$$

$$= Np \sum_{x=1}^N \frac{(N-1)!}{(x-1)!(N-x)!} p^{x-1} (1-p)^{N-x} = Np \sum_{x=0}^{N-1} \text{Bin}(x|p, N-1)$$

$$= Np$$
- $$\text{Similar, } E[X(X-1)] = \sum_{x=2}^N \frac{N!}{(x-2)!(N-x)!} p^x (1-p)^{N-x}$$

$$= N(N-1)p^2 \sum_{x=0}^{N-2} \text{Bin}(x|p, N-2) = N(N-1)p^2$$
- Therefore,  $\text{Var}[X] = E[X^2] - E[X]^2$

# Binomial distribution ( $N$ trials)

- Repeat the experiment for  $N$  times, the probability of the outcome will now be described by the binomial distribution. Note that  $x$  is now the number of obtained heads, we have

$$\text{Bin}(x|p, N) = \binom{N}{x} p^x (1-p)^{N-x},$$

- Mean and variances are given by

- $$E[X] = \sum_{x=0}^N \text{Bin}(x|p) x = \sum_{x=1}^N \frac{N!}{(x-1)!(N-x)!} p^x (1-p)^{N-x}$$

$$= Np \sum_{x=1}^N \frac{(N-1)!}{(x-1)!(N-x)!} p^{x-1} (1-p)^{N-x} = Np \sum_{x=0}^{N-1} \text{Bin}(x|p, N-1)$$

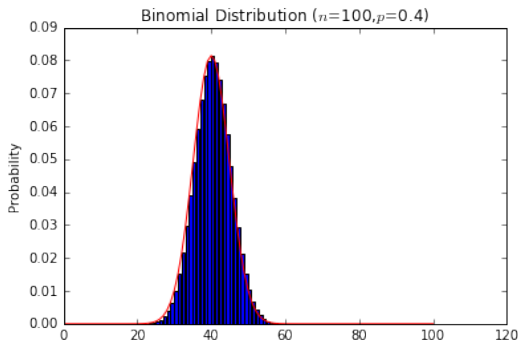
$$= Np$$
- $$\text{Similar, } E[X(X-1)] = \sum_{x=2}^N \frac{N!}{(x-2)!(N-x)!} p^x (1-p)^{N-x}$$

$$= N(N-1)p^2 \sum_{x=0}^{N-2} \text{Bin}(x|p, N-2) = N(N-1)p^2$$
- $$\text{Therefore, } \text{Var}[X] = E[X^2] - E[X]^2 = E[X(X-1)] + E[X] - E[X]^2 =$$

$$N(N-1)p^2 + Np - (Np)^2 = Np(1-p)$$

# Binomial distribution

As shown below, the binomial distribution can be model well with a normal distribution  $\mathcal{N}(Np, Np(1 - p))$  for large  $N$



The binomial distribution is shown in blue and an approximation by normal distribution is shown in red

# Conjugate prior

- Note that both Bernoulli and binomial distributions have the form  $p^u(1 - p)^v$



# Conjugate prior

- Note that both Bernoulli and binomial distributions have the form  $p^u(1-p)^v$
- To estimate  $p$ , recall that the ML estimator will try to compute

$$\hat{p} = \arg \max_p p(u, v|p) = \arg \max_p p^u(1-p)^v$$

# Conjugate prior

- Note that both Bernoulli and binomial distributions have the form  $p^u(1-p)^v$
- To estimate  $p$ , recall that the ML estimator will try to compute

$$\hat{p} = \arg \max_p p(u, v|p) = \arg \max_p p^u(1-p)^v$$

- Now if we would like to use the MAP estimator instead, we need to introduce a prior  $p(p)$  and solve instead

$$\hat{p} = \arg \max_p p(u, v|p)p(p) = \arg \max_p p^u(1-p)^v p(p)$$

# Conjugate prior

- Note that both Bernoulli and binomial distributions have the form  $p^u(1-p)^v$
- To estimate  $p$ , recall that the ML estimator will try to compute

$$\hat{p} = \arg \max_p p(u, v|p) = \arg \max_p p^u(1-p)^v$$

- Now if we would like to use the MAP estimator instead, we need to introduce a prior  $p(p)$  and solve instead

$$\hat{p} = \arg \max_p p(u, v|p)p(p) = \arg \max_p p^u(1-p)^v p(p)$$

- It is very difficult to determine the prior unanimously. Actually it can be controversial just to determine the form of it

# Conjugate prior

- Note that both Bernoulli and binomial distributions have the form  $p^u(1-p)^v$
- To estimate  $p$ , recall that the ML estimator will try to compute

$$\hat{p} = \arg \max_p p(u, v|p) = \arg \max_p p^u(1-p)^v$$

- Now if we would like to use the MAP estimator instead, we need to introduce a prior  $p(p)$  and solve instead

$$\hat{p} = \arg \max_p p(u, v|p)p(p) = \arg \max_p p^u(1-p)^v p(p)$$

- It is very difficult to determine the prior unanimously. Actually it can be controversial just to determine the form of it
- However, if we select  $p(p)$  of a form  $p(p) \propto p^a(1-p)^b$ , then the resulting posterior distribution with the same form as before. This choice is often chosen for practical purposes, and a prior with same “form” as its likelihood (and thus posterior) is known as the

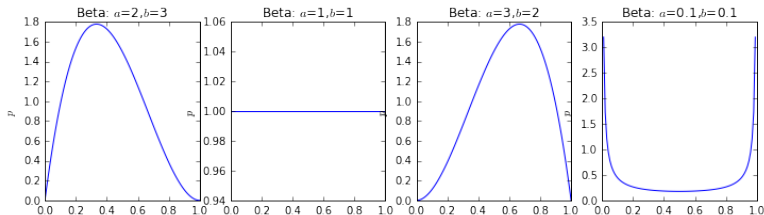
conjugate prior

# Beta distribution

- The conjugate prior of both Bernoulli and binomial distributions is the beta distribution. Its pdf is given by

$$\text{Beta}(x|a, b) = \frac{x^{a-1}(1-x)^{b-1}}{B(a, b)},$$

where  $X \in [0, 1]$  and  $B(a, b) = \frac{\Gamma(a)\Gamma(b)}{\Gamma(a+b)}$

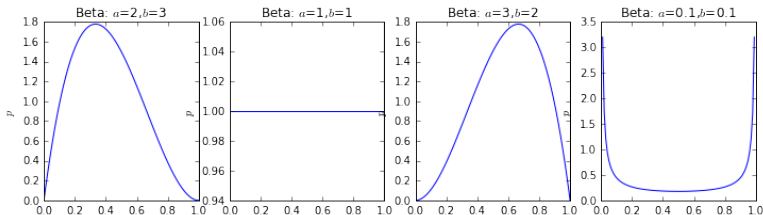


# Beta distribution

- The conjugate prior of both Bernoulli and binomial distributions is the beta distribution. Its pdf is given by

$$\text{Beta}(x|a, b) = \frac{x^{a-1}(1-x)^{b-1}}{B(a, b)},$$

where  $X \in [0, 1]$  and  $B(a, b) = \frac{\Gamma(a)\Gamma(b)}{\Gamma(a+b)}$



- Note that with  $a = b = 1$ ,  $\text{Beta}(x|1, 1) = 1$ . It is the same as no prior

# Gamma function

Note that  $\Gamma(z) = \int_0^{\infty} x^{z-1} e^{-x} dx$

- $\Gamma(1) = \int_0^{\infty} e^{-x} dx = -e^{-x} \Big|_0^{\infty} = 1$

# Gamma function

Note that  $\Gamma(z) = \int_0^{\infty} x^{z-1} e^{-x} dx$

- $\Gamma(1) = \int_0^{\infty} e^{-x} dx = -e^{-x} \Big|_0^{\infty} = 1$
- For  $z > 1$ , we have  $\Gamma(z) = (z-1)\Gamma(z-1)$



# Gamma function

Note that  $\Gamma(z) = \int_0^{\infty} x^{z-1} e^{-x} dx$

- $\Gamma(1) = \int_0^{\infty} e^{-x} dx = -e^{-x} \Big|_0^{\infty} = 1$
- For  $z > 1$ , we have  $\Gamma(z) = (z-1)\Gamma(z-1)$

Proof.

$$\Gamma(z) = \int_0^{\infty} x^{z-1} e^{-x} dx$$

# Gamma function

Note that  $\Gamma(z) = \int_0^{\infty} x^{z-1} e^{-x} dx$

- $\Gamma(1) = \int_0^{\infty} e^{-x} dx = -e^{-x} \Big|_0^{\infty} = 1$
- For  $z > 1$ , we have  $\Gamma(z) = (z-1)\Gamma(z-1)$

Proof.

$$\Gamma(z) = \int_0^{\infty} x^{z-1} e^{-x} dx = - \int_0^{\infty} x^{z-1} de^{-x}$$

# Gamma function

Note that  $\Gamma(z) = \int_0^{\infty} x^{z-1} e^{-x} dx$

- $\Gamma(1) = \int_0^{\infty} e^{-x} dx = -e^{-x} \Big|_0^{\infty} = 1$
- For  $z > 1$ , we have  $\Gamma(z) = (z-1)\Gamma(z-1)$

Proof.

$$\begin{aligned}\Gamma(z) &= \int_0^{\infty} x^{z-1} e^{-x} dx = - \int_0^{\infty} x^{z-1} de^{-x} \\ &= -x^{z-1} e^{-x} \Big|_0^{\infty} + (z-1) \int_0^{\infty} x^{z-2} e^{-x} dx\end{aligned}$$

# Gamma function

Note that  $\Gamma(z) = \int_0^{\infty} x^{z-1} e^{-x} dx$

- $\Gamma(1) = \int_0^{\infty} e^{-x} dx = -e^{-x} \Big|_0^{\infty} = 1$
- For  $z > 1$ , we have  $\Gamma(z) = (z-1)\Gamma(z-1)$

Proof.

$$\begin{aligned}\Gamma(z) &= \int_0^{\infty} x^{z-1} e^{-x} dx = - \int_0^{\infty} x^{z-1} de^{-x} \\ &= -x^{z-1} e^{-x} \Big|_0^{\infty} + (z-1) \int_0^{\infty} x^{z-2} e^{-x} dx \\ &= (z-1) \int_0^{\infty} x^{z-2} e^{-x} dx = (z-1)\Gamma(z-1)\end{aligned}$$



# Gamma function

Note that  $\Gamma(z) = \int_0^{\infty} x^{z-1} e^{-x} dx$

- $\Gamma(1) = \int_0^{\infty} e^{-x} dx = -e^{-x} \Big|_0^{\infty} = 1$
- For  $z > 1$ , we have  $\Gamma(z) = (z-1)\Gamma(z-1)$

Proof.

$$\begin{aligned}\Gamma(z) &= \int_0^{\infty} x^{z-1} e^{-x} dx = - \int_0^{\infty} x^{z-1} de^{-x} \\ &= -x^{z-1} e^{-x} \Big|_0^{\infty} + (z-1) \int_0^{\infty} x^{z-2} e^{-x} dx \\ &= (z-1) \int_0^{\infty} x^{z-2} e^{-x} dx = (z-1)\Gamma(z-1)\end{aligned}$$

□

- Therefore, for integer  $z > 1$ ,  $\Gamma(z) = (z-1)!$

# Mode of beta distribution

The mode is the peak of a distribution. Recall that

$Beta(x|a, b) = \frac{x^{a-1}(1-x)^{b-1}}{B(a, b)}$ . Set

$$\frac{\partial Beta(x|a, b)}{\partial x} = \frac{(a-1)x^{a-2}(1-x)^{b-1} - (b-1)x^{a-1}(1-x)^{b-2}}{B(a, b)} = 0,$$

we have  $(a-1)(1-x) = (b-1)x \Rightarrow x = \frac{a-1}{a+b-2}$

# Mean and variance of Beta distribution

Note that  $\int_{x=0}^1 p(x|a, b) = 1 \Rightarrow \int_{x=0}^1 x^{a-1}(1-x)^{b-1} = B(a, b) = \frac{\Gamma(a)\Gamma(b)}{\Gamma(a+b)}$ .  
This gives us a handy trick to manipulate beta distribution

# Mean and variance of Beta distribution

Note that  $\int_{x=0}^1 p(x|a, b) = 1 \Rightarrow \int_{x=0}^1 x^{a-1}(1-x)^{b-1} = B(a, b) = \frac{\Gamma(a)\Gamma(b)}{\Gamma(a+b)}$ .  
This gives us a handy trick to manipulate beta distribution

$$\begin{aligned} E[X] &= \int_{x=0}^1 x \text{Beta}(x|a, b) dx = \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)} \int_{x=0}^1 x^a (1-x)^{b-1} dx \\ &= \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)} \frac{\Gamma(a+1)\Gamma(b)}{\Gamma(a+b+1)} = \frac{a}{a+b} \end{aligned}$$



# Mean and variance of Beta distribution

Note that  $\int_{x=0}^1 p(x|a, b) = 1 \Rightarrow \int_{x=0}^1 x^{a-1}(1-x)^{b-1} = B(a, b) = \frac{\Gamma(a)\Gamma(b)}{\Gamma(a+b)}$ .  
This gives us a handy trick to manipulate beta distribution

$$\begin{aligned} E[X] &= \int_{x=0}^1 x \text{Beta}(x|a, b) dx = \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)} \int_{x=0}^1 x^a(1-x)^{b-1} dx \\ &= \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)} \frac{\Gamma(a+1)\Gamma(b)}{\Gamma(a+b+1)} = \frac{a}{a+b} \end{aligned}$$

Similarly,  $E[X^2] = \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)} \int_{x=0}^1 x^{a+1}(1-x)^{b-1} dx$

# Mean and variance of Beta distribution

Note that  $\int_{x=0}^1 p(x|a, b) = 1 \Rightarrow \int_{x=0}^1 x^{a-1}(1-x)^{b-1} = B(a, b) = \frac{\Gamma(a)\Gamma(b)}{\Gamma(a+b)}$ .

This gives us a handy trick to manipulate beta distribution

$$\begin{aligned} E[X] &= \int_{x=0}^1 x \text{Beta}(x|a, b) dx = \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)} \int_{x=0}^1 x^a (1-x)^{b-1} dx \\ &= \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)} \frac{\Gamma(a+1)\Gamma(b)}{\Gamma(a+b+1)} = \frac{a}{a+b} \end{aligned}$$

Similarly,  $E[X^2] = \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)} \int_{x=0}^1 x^{a+1}(1-x)^{b-1} dx = \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)} \frac{\Gamma(a+2)\Gamma(b)}{\Gamma(a+b+2)} = \frac{a(a+1)}{(a+b)(a+b+1)}$ .

# Mean and variance of Beta distribution

Note that  $\int_{x=0}^1 p(x|a, b) = 1 \Rightarrow \int_{x=0}^1 x^{a-1}(1-x)^{b-1} = B(a, b) = \frac{\Gamma(a)\Gamma(b)}{\Gamma(a+b)}$ .  
 This gives us a handy trick to manipulate beta distribution

$$\begin{aligned} E[X] &= \int_{x=0}^1 x \text{Beta}(x|a, b) dx = \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)} \int_{x=0}^1 x^a(1-x)^{b-1} dx \\ &= \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)} \frac{\Gamma(a+1)\Gamma(b)}{\Gamma(a+b+1)} = \frac{a}{a+b} \end{aligned}$$

Similarly,  $E[X^2] = \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)} \int_{x=0}^1 x^{a+1}(1-x)^{b-1} dx = \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)} \frac{\Gamma(a+2)\Gamma(b)}{\Gamma(a+b+2)} = \frac{a(a+1)}{(a+b)(a+b+1)}$ . Thus,

$$\begin{aligned} \text{Var}[X] &= E[X^2] - E[X]^2 = \frac{a(a+1)}{(a+b)(a+b+1)} - \frac{a^2}{(a+b)^2} \\ &= \frac{a(a+1)(a+b) - a^2(a+b+1)}{(a+b)^2(a+b+1)} = \frac{ab}{(a+b)^2(a+b+1)} \end{aligned}$$

# Review multivariate normal

- Marginalization of a normal distribution is still a normal distribution
- Conditioning of normal distribution:

$$\mathbf{X}|\mathbf{y} \sim \mathcal{N}(\boldsymbol{\mu}_{\mathbf{X}} + \boldsymbol{\Sigma}_{\mathbf{X}\mathbf{Y}}\boldsymbol{\Sigma}_{\mathbf{Y}\mathbf{Y}}^{-1}(\mathbf{y} - \boldsymbol{\mu}_{\mathbf{Y}}), \boldsymbol{\Sigma}_{\mathbf{X}\mathbf{X}} - \boldsymbol{\Sigma}_{\mathbf{X}\mathbf{Y}}\boldsymbol{\Sigma}_{\mathbf{Y}\mathbf{Y}}^{-1}\boldsymbol{\Sigma}_{\mathbf{Y}\mathbf{X}})$$

- Product of normal distribution:

$$\mathcal{N}(\mathbf{y}_1; \mathbf{x}, \boldsymbol{\Sigma}_{\mathbf{Y}_1})\mathcal{N}(\mathbf{y}_2; \mathbf{x}, \boldsymbol{\Sigma}_{\mathbf{Y}_2}) =$$

$$\mathcal{N}(\mathbf{y}_1; \mathbf{y}_2, \boldsymbol{\Sigma}_{\mathbf{Y}_2} + \boldsymbol{\Sigma}_{\mathbf{Y}_1})\mathcal{N}(\mathbf{x}; (\boldsymbol{\Lambda}_{\mathbf{Y}_1} + \boldsymbol{\Lambda}_{\mathbf{Y}_2})^{-1}(\boldsymbol{\Lambda}_{\mathbf{Y}_2}\mathbf{y}_2 + \boldsymbol{\Lambda}_{\mathbf{Y}_1}\mathbf{y}_1), (\boldsymbol{\Lambda}_{\mathbf{Y}_2} + \boldsymbol{\Lambda}_{\mathbf{Y}_1})^{-1})$$

- Mixture of Gaussian

- Merge components:

$$w \leftarrow \sum_i w_i, \quad \hat{w}_i = \frac{w_i}{\sum_j w_j}, \quad \boldsymbol{\mu}_i \leftarrow \sum_i w_i \boldsymbol{\mu}_i,$$

$$\boldsymbol{\Sigma} \leftarrow \sum_{i=1}^n \hat{w}_i (\boldsymbol{\Sigma}_i + \boldsymbol{\mu}_i \boldsymbol{\mu}_i^T) - \sum_{i=1}^n \sum_{j=1}^n \hat{w}_i \hat{w}_j \boldsymbol{\mu}_i \boldsymbol{\mu}_j$$

- Similarity measure

$$\text{Sim}(\mathcal{N}(\boldsymbol{\mu}_p, \boldsymbol{\Sigma}_p), \mathcal{N}(\boldsymbol{\mu}_q, \boldsymbol{\Sigma}_q)) = \frac{\mathcal{N}(\boldsymbol{\mu}_p; \boldsymbol{\mu}_q, \boldsymbol{\Sigma}_p + \boldsymbol{\Sigma}_q)}{\sqrt{\mathcal{N}(0; 0, 2\boldsymbol{\Sigma}_p)\mathcal{N}(0; 0, 2\boldsymbol{\Sigma}_q)}}$$

# More from last week...

- Bernoulli pdf:  $Bern(x|p) = p^x(1-p)^{1-x}$
- Binomial pdf:  $Bin(x|p, N) \propto p^x(1-p)^{N-x}$
- Beta pdf:  $Beta(x|a, b) = \frac{x^{a-1}(1-x)^{b-1}}{B(a, b)}$ , where  $B(a, b) = \frac{\Gamma(a)\Gamma(b)}{\Gamma(a+b)}$
- Gamma function  $\Gamma(z)$ 
  - $\Gamma(z) = (z-1)\Gamma(z-1)$
  - $\Gamma(n) = (n-1)!$  if  $n$  is an integer  $\geq 1$
- Conjugate prior: a prior with same “form” as its posterior distribution
  - Beta distribution is conjugate prior of Bernoulli and binomial distributions

# Summary of Beta distribution

- Pdf:

$$\text{Beta}(x|a, b) = \frac{x^{a-1}(1-x)^{b-1}}{B(a, b)}$$

with  $B(a, b) = \frac{\Gamma(a)\Gamma(b)}{\Gamma(a+b)}$

- Mean:

$$\frac{a}{a+b}$$

- Variance:

$$\frac{ab}{(a+b)^2(a+b+1)}$$

- Mode:

$$\frac{a-1}{a+b-2}$$

# Posterior estimate of probability $p$

Consider the coin flipping example again. Let say the prior probability<sup>7</sup> of the coin is beta distributed with parameters  $a$  and  $b$ . And we flip the coin once to get outcome  $x$ .

---

<sup>7</sup>Note that this can be very confusing at the beginning. Beware that we are talking about the distribution of the **probability** of some outcome

# Posterior estimate of probability $p$

Consider the coin flipping example again. Let say the prior probability<sup>7</sup> of the coin is beta distributed with parameters  $a$  and  $b$ . And we flip the coin once to get outcome  $x$ . Upon observing  $x$ , we can estimate  $p$  by

$$p(p|x, a, b)$$

---

<sup>7</sup>Note that this can be very confusing at the beginning. Beware that we are talking about the distribution of the **probability** of some outcome



# Posterior estimate of probability $p$

Consider the coin flipping example again. Let say the prior probability<sup>7</sup> of the coin is beta distributed with parameters  $a$  and  $b$ . And we flip the coin once to get outcome  $x$ . Upon observing  $x$ , we can estimate  $p$  by

$$p(p|x, a, b) = \text{Const} \cdot \text{Beta}(p|a, b) \text{Bern}(x|p)$$

---

<sup>7</sup>Note that this can be very confusing at the beginning. Beware that we are talking about the distribution of the **probability** of some outcome


# Posterior estimate of probability $p$

Consider the coin flipping example again. Let say the prior probability<sup>7</sup> of the coin is beta distributed with parameters  $a$  and  $b$ . And we flip the coin once to get outcome  $x$ . Upon observing  $x$ , we can estimate  $p$  by

$$\begin{aligned} p(p|x, a, b) &= \text{Const1} \cdot \text{Beta}(p|a, b) \text{Bern}(x|p) \\ &= \text{Const2} \cdot p^{a-1+x} (1-p)^{b-1+1-x} \\ &= \text{Beta}(p|\tilde{a}, \tilde{b}) \end{aligned}$$

So the posterior probability distribution is also beta distributed and the parameters just changed to  $\tilde{a} \leftarrow a + x$  and  $\tilde{b} \leftarrow b + 1 - x$

---

<sup>7</sup>Note that this can be very confusing at the beginning. Beware that we are talking about the distribution of the **probability** of some outcome 

# Posterior estimate of probability $p$

Let say we continue our example and we flip the coin by  $N$  times and obtain  $x$  head. So instead of the Bernoulli likelihood, we have a binomial likelihood. Like the last slide, we have the same beta prior with parameters  $a$  and  $b$ .

# Posterior estimate of probability $p$

Let say we continue our example and we flip the coin by  $N$  times and obtain  $x$  head. So instead of the Bernoulli likelihood, we have a binomial likelihood. Like the last slide, we have the same beta prior with parameters  $a$  and  $b$ . After the experiment  $x$ , we can update the distribution of our estimated  $p$  by

$$\begin{aligned} p(p|x, a, b) &= \text{Const1} \cdot \text{Beta}(p|a, b) \text{Bin}(x|p, N) \\ &= \text{Const2} \cdot p^{a-1+x} (1-p)^{b-1+N-x} \\ &= \text{Beta}(p|\tilde{a}, \tilde{b}) \end{aligned}$$

# Posterior estimate of probability $p$

Let say we continue our example and we flip the coin by  $N$  times and obtain  $x$  head. So instead of the Bernoulli likelihood, we have a binomial likelihood. Like the last slide, we have the same beta prior with parameters  $a$  and  $b$ . After the experiment  $x$ , we can update the distribution of our estimated  $p$  by

$$\begin{aligned} p(p|x, a, b) &= \text{Const1} \cdot \text{Beta}(p|a, b) \text{Bin}(x|p, N) \\ &= \text{Const2} \cdot p^{a-1+x} (1-p)^{b-1+N-x} \\ &= \text{Beta}(p|\tilde{a}, \tilde{b}) \end{aligned}$$

Again, the posterior distribution is still beta but with parameters updated to  $\tilde{a} \leftarrow a + x$  and  $\tilde{b} \leftarrow b + N - x$

# Prior and regularization

- One major reason of introducing prior is for the sake of “regularizing” the answer
- Another coin example
  - Fall back to high school, assume that we flip a coin for 10 times and got 3 heads. We want to estimate the chance of getting heads

# Prior and regularization

- One major reason of introducing prior is for the sake of “regularizing” the answer
- Another coin example
  - Fall back to high school, assume that we flip a coin for 10 times and got 3 heads. We want to estimate the chance of getting heads
  - 3/10, right?
  - And if I asked you chance of getting another head in the future, you will say the chance of getting another head is 3/10

# Prior and regularization

- One major reason of introducing prior is for the sake of “regularizing” the answer
- Another coin example
  - Fall back to high school, assume that we flip a coin for 10 times and got 3 heads. We want to estimate the chance of getting heads
  - 3/10, right?
  - And if I asked you chance of getting another head in the future, you will say the chance of getting another head is 3/10
  - Now, if I actually flip the coin for 10 times and got no head, what do you expect the chance of getting a head next time?



# Prior and regularization

- One major reason of introducing prior is for the sake of “regularizing” the answer
- Another coin example
  - Fall back to high school, assume that we flip a coin for 10 times and got 3 heads. We want to estimate the chance of getting heads
  - 3/10, right?
  - And if I asked you chance of getting another head in the future, you will say the chance of getting another head is 3/10
  - Now, if I actually flip the coin for 10 times and got no head, what do you expect the chance of getting a head next time?
  - 0? Okay, the estimate is a bit extreme. We know that it is very difficult to make a coin that always gives a tail

# Prior and regularization

- One major reason of introducing prior is for the sake of “regularizing” the answer
- Another coin example
  - Fall back to high school, assume that we flip a coin for 10 times and got 3 heads. We want to estimate the chance of getting heads
  - $3/10$ , right?
  - And if I asked you chance of getting another head in the future, you will say the chance of getting another head is  $3/10$
  - Now, if I actually flip the coin for 10 times and got no head, what do you expect the chance of getting a head next time?
  - 0? Okay, the estimate is a bit extreme. We know that it is very difficult to make a coin that always gives a tail
  - How about we first assumed that we actually flipped two times and got 1 head before we did experiment? We will estimate  $1/12$  instead of  $0/10$

# Prior and regularization

- We can verify that this is exactly what we got for a Beta prior with  $a = 2$  and  $b = 2$ .

# Prior and regularization

- We can verify that this is exactly what we got for a Beta prior with  $a = 2$  and  $b = 2$ . Note that the posterior distribution is

$$Beta(p|2, 2)Bin(x = 0|p, N = 10) \sim Beta(0 + a, 10 + b) = Beta(2, 12)$$

Now, what is the MAP estimate?

# Prior and regularization

- We can verify that this is exactly what we got for a Beta prior with  $a = 2$  and  $b = 2$ . Note that the posterior distribution is

$$Beta(p|2, 2)Bin(x = 0|p, N = 10) \sim Beta(0 + a, 10 + b) = Beta(2, 12)$$

Now, what is the MAP estimate? It should be the  $p$  that maximize the posterior probability. That is the mode of  $Beta(2, 12)$ .

# Prior and regularization

- We can verify that this is exactly what we got for a Beta prior with  $a = 2$  and  $b = 2$ . Note that the posterior distribution is

$$Beta(p|2, 2)Bin(x = 0|p, N = 10) \sim Beta(0 + a, 10 + b) = Beta(2, 12)$$

Now, what is the MAP estimate? It should be the  $p$  that maximize the posterior probability. That is the mode of  $Beta(2, 12)$ . Thus,

$$p_{Head}^{(MAP)} = \frac{a - 1}{a + b - 2} = \frac{1}{12}$$

# Prior and regularization

- We can verify that this is exactly what we got for a Beta prior with  $a = 2$  and  $b = 2$ . Note that the posterior distribution is

$$Beta(p|2, 2)Bin(x = 0|p, N = 10) \sim Beta(0 + a, 10 + b) = Beta(2, 12)$$

Now, what is the MAP estimate? It should be the  $p$  that maximize the posterior probability. That is the mode of  $Beta(2, 12)$ . Thus,

$$p_{Head}^{(MAP)} = \frac{a - 1}{a + b - 2} = \frac{1}{12}$$

- Recall that  $Beta(1, 1) = 1$  and so likelihood function is equivalent to  $Beta(p|1, 1)Bin(0|p, 10) \sim Beta(1, 11)$ .

# Prior and regularization

- We can verify that this is exactly what we got for a Beta prior with  $a = 2$  and  $b = 2$ . Note that the posterior distribution is

$$Beta(p|2, 2)Bin(x = 0|p, N = 10) \sim Beta(0 + a, 10 + b) = Beta(2, 12)$$

Now, what is the MAP estimate? It should be the  $p$  that maximize the posterior probability. That is the mode of  $Beta(2, 12)$ . Thus,

$$p_{Head}^{(MAP)} = \frac{a - 1}{a + b - 2} = \frac{1}{12}$$

- Recall that  $Beta(1, 1) = 1$  and so likelihood function is equivalent to  $Beta(p|1, 1)Bin(0|p, 10) \sim Beta(1, 11)$ . Thus the ML estimate is the mode of  $Beta(1, 11) \Rightarrow p_{Head}^{(ML)} = \frac{1-1}{1+11-2} = \frac{0}{10} = 0$ 
  - This indeed is the same as our high school naïve estimate



# Bayesian estimation and regularization

- Now let's consider the Bayesian estimate. Even for the case with no prior (equivalently an uniform prior or Beta prior with  $a = 1$  and  $b = 1$ ), recall that the “posterior distribution” is  $Beta(1, 11)$

# Bayesian estimation and regularization

- Now let's consider the Bayesian estimate. Even for the case with no prior (equivalently an uniform prior or Beta prior with  $a = 1$  and  $b = 1$ ), recall that the “posterior distribution” is  $Beta(1, 11)$
- The Bayesian estimate should be the average  $p$  summing all possibility of  $p$ ,

# Bayesian estimation and regularization

- Now let's consider the Bayesian estimate. Even for the case with no prior (equivalently an uniform prior or Beta prior with  $a = 1$  and  $b = 1$ ), recall that the “posterior distribution” is  $Beta(1, 11)$
- The Bayesian estimate should be the average  $p$  summing all possibility of  $p$ , which is essentially just,  $\int pBeta(p|1, 11)dp = E[p]$ , i.e., the mean.

# Bayesian estimation and regularization

- Now let's consider the Bayesian estimate. Even for the case with no prior (equivalently an uniform prior or Beta prior with  $a = 1$  and  $b = 1$ ), recall that the “posterior distribution” is  $Beta(1, 11)$
- The Bayesian estimate should be the average  $p$  summing all possibility of  $p$ , which is essentially just,  $\int pBeta(p|1, 11)dp = E[p]$ , i.e., the mean. Thus

$$p_{Head}^{(Bayesian)} = \frac{a}{a + b} = \frac{1}{11}$$

# Bayesian estimation and regularization

- Now let's consider the Bayesian estimate. Even for the case with no prior (equivalently an uniform prior or Beta prior with  $a = 1$  and  $b = 1$ ), recall that the “posterior distribution” is  $Beta(1, 11)$
- The Bayesian estimate should be the average  $p$  summing all possibility of  $p$ , which is essentially just,  $\int pBeta(p|1, 11)dp = E[p]$ , i.e., the mean. Thus

$$p_{Head}^{(Bayesian)} = \frac{a}{a + b} = \frac{1}{11}$$

- Note that Bayesian estimation is “self-regularized” (i.e., giving less extreme results) since it inherently averages out all possible cases

# Multinomial distribution

- Binomial distribution models the probability of a binary outcome. For a random event with discrete but non-binary (more than two) outcomes, we can model the event with a multinomial distribution

# Multinomial distribution

- Binomial distribution models the probability of a binary outcome. For a random event with discrete but non-binary (more than two) outcomes, we can model the event with a multinomial distribution
- Let say the probability of each possible outcome  $i$  is  $p_i$ . And we have conducted  $N$  different experiments, let say  $x_i$  is the number of times we obtain outcome  $i$ . Then the probability of such even is given by

# Multinomial distribution

- Binomial distribution models the probability of a binary outcome. For a random event with discrete but non-binary (more than two) outcomes, we can model the event with a multinomial distribution
- Let say the probability of each possible outcome  $i$  is  $p_i$ . And we have conducted  $N$  different experiments, let say  $x_i$  is the number of times we obtain outcome  $i$ . Then the probability of such even is given by

$$\text{Mult}(x_1, \dots, x_n | p_1, \dots, p_n) = \binom{N}{x_1 x_2 \dots x_n} p_1^{x_1} p_2^{x_2} \dots p_n^{x_n},$$



# Multinomial distribution

- Binomial distribution models the probability of a binary outcome. For a random event with discrete but non-binary (more than two) outcomes, we can model the event with a multinomial distribution
- Let say the probability of each possible outcome  $i$  is  $p_i$ . And we have conducted  $N$  different experiments, let say  $x_i$  is the number of times we obtain outcome  $i$ . Then the probability of such even is given by

$$\text{Mult}(x_1, \dots, x_n | p_1, \dots, p_n) = \binom{N}{x_1 x_2 \dots x_n} p_1^{x_1} p_2^{x_2} \dots p_n^{x_n},$$

- Just make sure we are in the same pace. Note that  $p_1 + p_2 + \dots + p_n = 1$  and  $x_1 + x_2 + \dots + x_n = N$

# Dirichlet distribution

- Note that the conjugate prior of multinomial distribution should take the form  $x_1^{\alpha_1-1} x_2^{\alpha_2-1} \dots x_n^{\alpha_n-1}$

# Dirichlet distribution

- Note that the conjugate prior of multinomial distribution should take the form  $x_1^{\alpha_1-1} x_2^{\alpha_2-1} \dots x_n^{\alpha_n-1}$
- It turns out that the distribution is the so-called Dirichlet distribution. Its pdf is given by

$$\begin{aligned} & Dir(x_1, \dots, x_n | \alpha_1, \dots, \alpha_n) \\ &= \frac{\Gamma(\alpha_1 + \dots + \alpha_n)}{\Gamma(\alpha_1)\Gamma(\alpha_2)\dots\Gamma(\alpha_n)} x_1^{\alpha_1-1} x_2^{\alpha_2-1} \dots x_n^{\alpha_n-1} \end{aligned}$$

# Dirichlet distribution

- Note that the conjugate prior of multinomial distribution should take the form  $x_1^{\alpha_1-1} x_2^{\alpha_2-1} \dots x_n^{\alpha_n-1}$
- It turns out that the distribution is the so-called Dirichlet distribution. Its pdf is given by

$$\begin{aligned} & Dir(x_1, \dots, x_n | \alpha_1, \dots, \alpha_n) \\ &= \frac{\Gamma(\alpha_1 + \dots + \alpha_n)}{\Gamma(\alpha_1)\Gamma(\alpha_2)\dots\Gamma(\alpha_n)} x_1^{\alpha_1-1} x_2^{\alpha_2-1} \dots x_n^{\alpha_n-1} \end{aligned}$$

- As usual since pdf should be normalized to 1, we have

$$\int x_1^{\alpha_1-1} x_2^{\alpha_2-1} \dots x_n^{\alpha_n-1} = \frac{\Gamma(\alpha_1)\Gamma(\alpha_2)\dots\Gamma(\alpha_n)}{\Gamma(\alpha_1 + \dots + \alpha_n)}$$

# Mean, mode, variance of Dirichlet distribution

- Mean:

$$\begin{aligned} E[X_1] &= \frac{\Gamma(\alpha_1 + \cdots + \alpha_n)}{\Gamma(\alpha_1) \cdots \Gamma(\alpha_n)} \int x_1^{\alpha_1} x_2^{\alpha_2 - 1} \cdots x_n^{\alpha_n - 1} \\ &= \frac{\Gamma(\alpha_1 + \cdots + \alpha_n)}{\Gamma(\alpha_1) \cdots \Gamma(\alpha_n)} \frac{\Gamma(\alpha_1 + 1) \cdots \Gamma(\alpha_n)}{\Gamma(\alpha_1 + \cdots + \alpha_n + 1)} = \frac{\alpha_1}{\alpha_1 + \cdots + \alpha_n} \end{aligned}$$

# Mean, mode, variance of Dirichlet distribution

- Mean:

$$\begin{aligned}
 E[X_1] &= \frac{\Gamma(\alpha_1 + \cdots + \alpha_n)}{\Gamma(\alpha_1) \cdots \Gamma(\alpha_n)} \int x_1^{\alpha_1} x_2^{\alpha_2-1} \cdots x_n^{\alpha_n-1} \\
 &= \frac{\Gamma(\alpha_1 + \cdots + \alpha_n)}{\Gamma(\alpha_1) \cdots \Gamma(\alpha_n)} \frac{\Gamma(\alpha_1 + 1) \cdots \Gamma(\alpha_n)}{\Gamma(\alpha_1 + \cdots + \alpha_n + 1)} = \frac{\alpha_1}{\alpha_1 + \cdots + \alpha_n}
 \end{aligned}$$

- Similarly,  $E[X_1^2] = \frac{\Gamma(\alpha_1 + \cdots + \alpha_n)}{\Gamma(\alpha_1) \cdots \Gamma(\alpha_n)} \int x_1^{\alpha_1+1} x_2^{\alpha_2-1} \cdots x_n^{\alpha_n-1} = \frac{\Gamma(\alpha_1 + \cdots + \alpha_n)}{\Gamma(\alpha_1) \cdots \Gamma(\alpha_n)} \frac{\Gamma(\alpha_1+2) \cdots \Gamma(\alpha_n)}{\Gamma(\alpha_1 + \cdots + \alpha_n + 2)} = \frac{(\alpha_1+1)\alpha_1}{(\alpha_1 + \cdots + \alpha_n + 1)(\alpha_1 + \cdots + \alpha_n)}$ .

# Mean, mode, variance of Dirichlet distribution

- Mean:

$$\begin{aligned}
 E[X_1] &= \frac{\Gamma(\alpha_1 + \cdots + \alpha_n)}{\Gamma(\alpha_1) \cdots \Gamma(\alpha_n)} \int x_1^{\alpha_1} x_2^{\alpha_2-1} \cdots x_n^{\alpha_n-1} \\
 &= \frac{\Gamma(\alpha_1 + \cdots + \alpha_n)}{\Gamma(\alpha_1) \cdots \Gamma(\alpha_n)} \frac{\Gamma(\alpha_1 + 1) \cdots \Gamma(\alpha_n)}{\Gamma(\alpha_1 + \cdots + \alpha_n + 1)} = \frac{\alpha_1}{\alpha_1 + \cdots + \alpha_n}
 \end{aligned}$$

- Similarly,  $E[X_1^2] = \frac{\Gamma(\alpha_1 + \cdots + \alpha_n)}{\Gamma(\alpha_1) \cdots \Gamma(\alpha_n)} \int x_1^{\alpha_1+1} x_2^{\alpha_2-1} \cdots x_n^{\alpha_n-1} = \frac{\Gamma(\alpha_1 + \cdots + \alpha_n)}{\Gamma(\alpha_1) \cdots \Gamma(\alpha_n)} \frac{\Gamma(\alpha_1 + 2) \cdots \Gamma(\alpha_n)}{\Gamma(\alpha_1 + \cdots + \alpha_n + 2)} = \frac{(\alpha_1 + 1)\alpha_1}{(\alpha_1 + \cdots + \alpha_n + 1)(\alpha_1 + \cdots + \alpha_n)}$ . Thus,  $Var(X_1) = E[X_1^2] - E[X_1]^2 = \frac{(\alpha_1 + 1)\alpha_1}{(\alpha_1 + \cdots + \alpha_n + 1)(\alpha_1 + \cdots + \alpha_n)} - \frac{\alpha_1^2}{(\alpha_1 + \cdots + \alpha_n)^2} = \frac{\alpha_1(\alpha_0 - \alpha_1)}{\alpha_0^2(\alpha_0 + 1)}$ , where  $\alpha_0 = \alpha_1 + \cdots + \alpha_n$

# Mean, mode, variance of Dirichlet distribution

- Mean:

$$\begin{aligned}
 E[X_1] &= \frac{\Gamma(\alpha_1 + \cdots + \alpha_n)}{\Gamma(\alpha_1) \cdots \Gamma(\alpha_n)} \int x_1^{\alpha_1} x_2^{\alpha_2-1} \cdots x_n^{\alpha_n-1} \\
 &= \frac{\Gamma(\alpha_1 + \cdots + \alpha_n)}{\Gamma(\alpha_1) \cdots \Gamma(\alpha_n)} \frac{\Gamma(\alpha_1 + 1) \cdots \Gamma(\alpha_n)}{\Gamma(\alpha_1 + \cdots + \alpha_n + 1)} = \frac{\alpha_1}{\alpha_1 + \cdots + \alpha_n}
 \end{aligned}$$

- Similarly,  $E[X_1^2] = \frac{\Gamma(\alpha_1 + \cdots + \alpha_n)}{\Gamma(\alpha_1) \cdots \Gamma(\alpha_n)} \int x_1^{\alpha_1+1} x_2^{\alpha_2-1} \cdots x_n^{\alpha_n-1} = \frac{\Gamma(\alpha_1 + \cdots + \alpha_n)}{\Gamma(\alpha_1) \cdots \Gamma(\alpha_n)} \frac{\Gamma(\alpha_1+2) \cdots \Gamma(\alpha_n)}{\Gamma(\alpha_1 + \cdots + \alpha_n + 2)} = \frac{(\alpha_1+1)\alpha_1}{(\alpha_1 + \cdots + \alpha_n + 1)(\alpha_1 + \cdots + \alpha_n)}$ . Thus,

$$\begin{aligned}
 \text{Var}(X_1) &= E[X_1^2] - E[X_1]^2 = \frac{(\alpha_1+1)\alpha_1}{(\alpha_1 + \cdots + \alpha_n + 1)(\alpha_1 + \cdots + \alpha_n)} - \frac{\alpha_1^2}{(\alpha_1 + \cdots + \alpha_n)^2} = \\
 &= \frac{\alpha_1(\alpha_0 - \alpha_1)}{\alpha_0^2(\alpha_0 + 1)}, \text{ where } \alpha_0 = \alpha_1 + \cdots + \alpha_n
 \end{aligned}$$

- Mode: one can show that the mode of  $Dir(\alpha_1, \cdots, \alpha_n)$  is

$$\frac{\alpha_i - 1}{\alpha_1 + \cdots + \alpha_n - n}.$$

We will not show it now but will leave as an **exercise**



# Summary of Dirichlet distribution

- Pdf:

$$Dir(\mathbf{x}|\boldsymbol{\alpha}) = \frac{\Gamma(\alpha_1 + \cdots + \alpha_n)}{\Gamma(\alpha_1)\Gamma(\alpha_2)\cdots\Gamma(\alpha_n)} x_1^{\alpha_1-1} x_2^{\alpha_2-1} \cdots x_n^{\alpha_n-1}$$

- Mean:

$$\frac{\alpha_j}{\alpha_1 + \cdots + \alpha_n}$$

- Variance:

$$\frac{\alpha_j(\alpha_0 - \alpha_j)}{\alpha_0^2(\alpha_0 + 1)}$$

- Mode:

$$\frac{\alpha_j - 1}{\alpha_1 + \cdots + \alpha_n - n}$$

# Posterior probability given Multinomial likelihood and Dirichlet prior

Upon observing  $x_1, \dots, x_n$ , the posterior distribution of  $p_1, \dots, p_n$  becomes

$$p(p_1, \dots, p_n | x_1, \dots, x_n, \alpha_1, \dots, \alpha_n)$$

# Posterior probability given Multinomial likelihood and Dirichlet prior

Upon observing  $x_1, \dots, x_n$ , the posterior distribution of  $p_1, \dots, p_n$  becomes

$$\begin{aligned} & p(p_1, \dots, p_n | x_1, \dots, x_n, \alpha_1, \dots, \alpha_n) \\ &= \text{Const}1 \cdot \text{Dir}(p_1, \dots, p_n | \alpha_1, \dots, \alpha_n) \text{Mult}(x_1, \dots, x_n | p_1, \dots, p_n) \end{aligned}$$

# Posterior probability given Multinomial likelihood and Dirichlet prior

Upon observing  $x_1, \dots, x_n$ , the posterior distribution of  $p_1, \dots, p_n$  becomes

$$\begin{aligned}
 & p(p_1, \dots, p_n | x_1, \dots, x_n, \alpha_1, \dots, \alpha_n) \\
 &= \text{Const1} \cdot \text{Dir}(p_1, \dots, p_n | \alpha_1, \dots, \alpha_n) \text{Mult}(x_1, \dots, x_n | p_1, \dots, p_n) \\
 &= \text{Const2} \cdot p_1^{x_1 + \alpha_1} \dots p_n^{x_n + \alpha_n} \\
 &= \text{Dir}(p_1, \dots, p_n | \tilde{\alpha}_1, \dots, \tilde{\alpha}_n)
 \end{aligned}$$

So the posterior distribution is Dirichlet with parameters updated to  $\tilde{\alpha}_1 \leftarrow x_1 + \alpha_1, \dots, \tilde{\alpha}_n \leftarrow x_n + \alpha_n$

# Poisson distribution

Poisson distribution describes the number of arrival  $K$  within some period. For example, one can use Poisson distribution to model the arrival process (Poisson process) of customers into a store.

# Poisson distribution

Poisson distribution describes the number of arrival  $K$  within some period. For example, one can use Poisson distribution to model the arrival process (Poisson process) of customers into a store. Its pdf is given by

$$Poisson(k|\lambda T) = \frac{e^{-\lambda T} (\lambda T)^k}{k!},$$

where  $k$  is a non-negative integer,  $\lambda$  is rate of arrival and  $T$  is the length of the observed period.

# Poisson distribution

Poisson distribution describes the number of arrival  $K$  within some period. For example, one can use Poisson distribution to model the arrival process (Poisson process) of customers into a store. Its pdf is given by

$$Poisson(k|\lambda T) = \frac{e^{-\lambda T}(\lambda T)^k}{k!},$$

where  $k$  is a non-negative integer,  $\lambda$  is rate of arrival and  $T$  is the length of the observed period. It is easy to check that (**please verify**)

$$Mean = \lambda T$$

$$Variance = \lambda T$$

N.B. the parameters  $\lambda T$  comes as a group and so we can consider it as a single parameter

# Poisson process

Poisson process is probably the simplest random process to model event arrivals. It is based on two simple assumptions



# Poisson process

Poisson process is probably the simplest random process to model event arrivals. It is based on two simple assumptions

- 1 Arrival rate is invariant over time
  - That is,  $\lambda$  is a constant that does not change with time

# Poisson process

Poisson process is probably the simplest random process to model event arrivals. It is based on two simple assumptions

- 1 Arrival rate is invariant over time
  - That is,  $\lambda$  is a constant that does not change with time
- 2 Each arrival is independent of the other

# Poisson process

Poisson process is probably the simplest random process to model event arrivals. It is based on two simple assumptions

- 1 Arrival rate is invariant over time
  - That is,  $\lambda$  is a constant that does not change with time
- 2 Each arrival is independent of the other
  - For example, even though we just have one customer coming in, the probability that the next customer to come in immediately should not decrease

# Poisson process

Poisson process is probably the simplest random process to model event arrivals. It is based on two simple assumptions

- 1 Arrival rate is invariant over time
  - That is,  $\lambda$  is a constant that does not change with time
- 2 Each arrival is independent of the other
  - For example, even though we just have one customer coming in, the probability that the next customer to come in immediately should not decrease
  - It makes sense to model say customers to a department store

# Poisson process

Poisson process is probably the simplest random process to model event arrivals. It is based on two simple assumptions

- 1 Arrival rate is invariant over time
  - That is,  $\lambda$  is a constant that does not change with time
- 2 Each arrival is independent of the other
  - For example, even though we just have one customer coming in, the probability that the next customer to come in immediately should not decrease
  - It makes sense to model say customers to a department store
  - It can be less perfect to model the times my car broke down. The events are likely to be related

# Poisson process and Poisson distribution

- Consider a period  $T$  and let's the arrival rate be  $\lambda$  as before. Let's partition  $T$  into  $N$  different very short intervals of length  $\Delta$ .

# Poisson process and Poisson distribution

- Consider a period  $T$  and let's the arrival rate be  $\lambda$  as before. Let's partition  $T$  into  $N$  different very short intervals of length  $\Delta$ . Hence,  $T = N\Delta$ . We will also assume  $N \rightarrow \infty$  and thus  $\Delta \rightarrow 0$ .

# Poisson process and Poisson distribution

- Consider a period  $T$  and let's the arrival rate be  $\lambda$  as before. Let's partition  $T$  into  $N$  different very short intervals of length  $\Delta$ . Hence,  $T = N\Delta$ . We will also assume  $N \rightarrow \infty$  and thus  $\Delta \rightarrow 0$ . The probability of getting an arrival in any interval  $\Delta$  is thus  $\lambda\Delta$ .



# Poisson process and Poisson distribution

- Consider a period  $T$  and let's the arrival rate be  $\lambda$  as before. Let's partition  $T$  into  $N$  different very short intervals of length  $\Delta$ . Hence,  $T = N\Delta$ . We will also assume  $N \rightarrow \infty$  and thus  $\Delta \rightarrow 0$ . The probability of getting an arrival in any interval  $\Delta$  is thus  $\lambda\Delta$ . Moreover, since  $\Delta \rightarrow 0$ , the probability of getting two arrivals  $\propto \Delta^2$  and is negligible compared to  $\lambda\Delta$

# Poisson process and Poisson distribution

- Consider a period  $T$  and let's the arrival rate be  $\lambda$  as before. Let's partition  $T$  into  $N$  different very short intervals of length  $\Delta$ . Hence,  $T = N\Delta$ . We will also assume  $N \rightarrow \infty$  and thus  $\Delta \rightarrow 0$ . The probability of getting an arrival in any interval  $\Delta$  is thus  $\lambda\Delta$ . Moreover, since  $\Delta \rightarrow 0$ , the probability of getting two arrivals  $\propto \Delta^2$  and is negligible compared to  $\lambda\Delta$
- Then, the probability of  $k$  arrivals  
 $Pr(k \text{ arrivals in } T)$

# Poisson process and Poisson distribution

- Consider a period  $T$  and let's the arrival rate be  $\lambda$  as before. Let's partition  $T$  into  $N$  different very short intervals of length  $\Delta$ . Hence,  $T = N\Delta$ . We will also assume  $N \rightarrow \infty$  and thus  $\Delta \rightarrow 0$ . The probability of getting an arrival in any interval  $\Delta$  is thus  $\lambda\Delta$ . Moreover, since  $\Delta \rightarrow 0$ , the probability of getting two arrivals  $\propto \Delta^2$  and is negligible compared to  $\lambda\Delta$
- Then, the probability of  $k$  arrivals  
$$\Pr(k \text{ arrivals in } T) = \binom{N}{k} (\lambda\Delta)^k (1 - \lambda\Delta)^{N-k}$$

# Poisson process and Poisson distribution

- Consider a period  $T$  and let's the arrival rate be  $\lambda$  as before. Let's partition  $T$  into  $N$  different very short intervals of length  $\Delta$ . Hence,  $T = N\Delta$ . We will also assume  $N \rightarrow \infty$  and thus  $\Delta \rightarrow 0$ . The probability of getting an arrival in any interval  $\Delta$  is thus  $\lambda\Delta$ . Moreover, since  $\Delta \rightarrow 0$ , the probability of getting two arrivals  $\propto \Delta^2$  and is negligible compared to  $\lambda\Delta$

- Then, the probability of  $k$  arrivals

$$\begin{aligned} Pr(k \text{ arrivals in } T) &= \binom{N}{k} (\lambda\Delta)^k (1 - \lambda\Delta)^{N-k} \\ &= \frac{N(N-1)\dots(N-k+1)}{k!} (\lambda\Delta)^k (1 - \lambda\Delta)^{N-k} \end{aligned}$$

# Poisson process and Poisson distribution

- Consider a period  $T$  and let's the arrival rate be  $\lambda$  as before. Let's partition  $T$  into  $N$  different very short intervals of length  $\Delta$ . Hence,  $T = N\Delta$ . We will also assume  $N \rightarrow \infty$  and thus  $\Delta \rightarrow 0$ . The probability of getting an arrival in any interval  $\Delta$  is thus  $\lambda\Delta$ . Moreover, since  $\Delta \rightarrow 0$ , the probability of getting two arrivals  $\propto \Delta^2$  and is negligible compared to  $\lambda\Delta$

- Then, the probability of  $k$  arrivals

$$\begin{aligned} \Pr(k \text{ arrivals in } T) &= \binom{N}{k} (\lambda\Delta)^k (1 - \lambda\Delta)^{N-k} \\ &= \frac{N(N-1)\dots(N-k+1)}{k!} (\lambda\Delta)^k (1 - \lambda\Delta)^{N-k} \approx \frac{N^k}{k!} \lambda^k \frac{T^k}{N^k} (1 - \lambda\Delta)^{N-k} \end{aligned}$$

# Poisson process and Poisson distribution

- Consider a period  $T$  and let's the arrival rate be  $\lambda$  as before. Let's partition  $T$  into  $N$  different very short intervals of length  $\Delta$ . Hence,  $T = N\Delta$ . We will also assume  $N \rightarrow \infty$  and thus  $\Delta \rightarrow 0$ . The probability of getting an arrival in any interval  $\Delta$  is thus  $\lambda\Delta$ . Moreover, since  $\Delta \rightarrow 0$ , the probability of getting two arrivals  $\propto \Delta^2$  and is negligible compared to  $\lambda\Delta$

- Then, the probability of  $k$  arrivals

$$\begin{aligned}
 \Pr(k \text{ arrivals in } T) &= \binom{N}{k} (\lambda\Delta)^k (1 - \lambda\Delta)^{N-k} \\
 &= \frac{N(N-1)\dots(N-k+1)}{k!} (\lambda\Delta)^k (1 - \lambda\Delta)^{N-k} \approx \frac{N^k}{k!} \lambda^k \frac{T^k}{N^k} (1 - \lambda\Delta)^{N-k} \\
 &= \frac{(\lambda T)^k}{k!} \left(1 - \frac{\lambda T}{N}\right)^{N-k}
 \end{aligned}$$

# Poisson process and Poisson distribution

- Consider a period  $T$  and let's the arrival rate be  $\lambda$  as before. Let's partition  $T$  into  $N$  different very short intervals of length  $\Delta$ . Hence,  $T = N\Delta$ . We will also assume  $N \rightarrow \infty$  and thus  $\Delta \rightarrow 0$ . The probability of getting an arrival in any interval  $\Delta$  is thus  $\lambda\Delta$ . Moreover, since  $\Delta \rightarrow 0$ , the probability of getting two arrivals  $\propto \Delta^2$  and is negligible compared to  $\lambda\Delta$

- Then, the probability of  $k$  arrivals

$$\begin{aligned}
 \Pr(k \text{ arrivals in } T) &= \binom{N}{k} (\lambda\Delta)^k (1 - \lambda\Delta)^{N-k} \\
 &= \frac{N(N-1)\dots(N-k+1)}{k!} (\lambda\Delta)^k (1 - \lambda\Delta)^{N-k} \approx \frac{N^k}{k!} \lambda^k \frac{T^k}{N^k} (1 - \lambda\Delta)^{N-k} \\
 &= \frac{(\lambda T)^k}{k!} \left(1 - \frac{\lambda T}{N}\right)^{N-k} \approx \frac{(\lambda T)^k}{k!} \left(1 - \frac{\lambda T}{N}\right)^N
 \end{aligned}$$

# Poisson process and Poisson distribution

- Consider a period  $T$  and let's the arrival rate be  $\lambda$  as before. Let's partition  $T$  into  $N$  different very short intervals of length  $\Delta$ . Hence,  $T = N\Delta$ . We will also assume  $N \rightarrow \infty$  and thus  $\Delta \rightarrow 0$ . The probability of getting an arrival in any interval  $\Delta$  is thus  $\lambda\Delta$ . Moreover, since  $\Delta \rightarrow 0$ , the probability of getting two arrivals  $\propto \Delta^2$  and is negligible compared to  $\lambda\Delta$

- Then, the probability of  $k$  arrivals

$$\begin{aligned}
 \Pr(k \text{ arrivals in } T) &= \binom{N}{k} (\lambda\Delta)^k (1 - \lambda\Delta)^{N-k} \\
 &= \frac{N(N-1)\dots(N-k+1)}{k!} (\lambda\Delta)^k (1 - \lambda\Delta)^{N-k} \approx \frac{N^k}{k!} \lambda^k \frac{T^k}{N^k} (1 - \lambda\Delta)^{N-k} \\
 &= \frac{(\lambda T)^k}{k!} \left(1 - \frac{\lambda T}{N}\right)^{N-k} \approx \frac{(\lambda T)^k}{k!} \left(1 - \frac{\lambda T}{N}\right)^N = \frac{(\lambda T)^k}{k!} \exp(-\lambda T),
 \end{aligned}$$

where we use  $(1 + a/N)^N = \exp(a)$  for the last equality



# Poisson process and Poisson distribution

- Consider a period  $T$  and let's the arrival rate be  $\lambda$  as before. Let's partition  $T$  into  $N$  different very short intervals of length  $\Delta$ . Hence,  $T = N\Delta$ . We will also assume  $N \rightarrow \infty$  and thus  $\Delta \rightarrow 0$ . The probability of getting an arrival in any interval  $\Delta$  is thus  $\lambda\Delta$ . Moreover, since  $\Delta \rightarrow 0$ , the probability of getting two arrivals  $\propto \Delta^2$  and is negligible compared to  $\lambda\Delta$

- Then, the probability of  $k$  arrivals

$$\begin{aligned}
 Pr(k \text{ arrivals in } T) &= \binom{N}{k} (\lambda\Delta)^k (1 - \lambda\Delta)^{N-k} \\
 &= \frac{N(N-1)\dots(N-k+1)}{k!} (\lambda\Delta)^k (1 - \lambda\Delta)^{N-k} \approx \frac{N^k}{k!} \lambda^k \frac{T^k}{N^k} (1 - \lambda\Delta)^{N-k} \\
 &= \frac{(\lambda T)^k}{k!} \left(1 - \frac{\lambda T}{N}\right)^{N-k} \approx \frac{(\lambda T)^k}{k!} \left(1 - \frac{\lambda T}{N}\right)^N = \frac{(\lambda T)^k}{k!} \exp(-\lambda T),
 \end{aligned}$$

where we use  $(1 + a/N)^N = \exp(a)$  for the last equality

Note that indeed  $Pr(k \text{ arrivals in } T) = \text{Poisson}(k|\lambda T)$

# Interarrival time of Poisson process

Using the similar analysis, we can also easily evaluate the distribution of interarrival time, the time that the next event will happen given that an event just happened. Let  $t = n\Delta$  and use the same notation as before

# Interarrival time of Poisson process

Using the similar analysis, we can also easily evaluate the distribution of interarrival time, the time that the next event will happen given that an event just happened. Let  $t = n\Delta$  and use the same notation as before

- Note that  $t > 0$  and  $\Delta \rightarrow 0$  and so  $n \rightarrow \infty$ .

# Interarrival time of Poisson process

Using the similar analysis, we can also easily evaluate the distribution of interarrival time, the time that the next event will happen given that an event just happened. Let  $t = n\Delta$  and use the same notation as before

- Note that  $t > 0$  and  $\Delta \rightarrow 0$  and so  $n \rightarrow \infty$ . Now,

$$Pr(\text{next event happened within in time } [t, t + \Delta])$$

$$= Pr(\text{next event happened within in time } [n\Delta, (n + 1)\Delta])$$

# Interarrival time of Poisson process

Using the similar analysis, we can also easily evaluate the distribution of interarrival time, the time that the next event will happen given that an event just happened. Let  $t = n\Delta$  and use the same notation as before

- Note that  $t > 0$  and  $\Delta \rightarrow 0$  and so  $n \rightarrow \infty$ . Now,  
 $Pr(\text{next event happened within in time } [t, t + \Delta])$   
 $= Pr(\text{next event happened within in time } [n\Delta, (n + 1)\Delta])$   
 $= Pr(\text{no event in first } n \text{ intervals})Pr(\text{event happened in } n + 1 \text{ interval})$

# Interarrival time of Poisson process

Using the similar analysis, we can also easily evaluate the distribution of interarrival time, the time that the next event will happen given that an event just happened. Let  $t = n\Delta$  and use the same notation as before

- Note that  $t > 0$  and  $\Delta \rightarrow 0$  and so  $n \rightarrow \infty$ . Now,  
$$\begin{aligned} & Pr(\text{next event happened within in time } [t, t + \Delta]) \\ &= Pr(\text{next event happened within in time } [n\Delta, (n + 1)\Delta]) \\ &= Pr(\text{no event in first } n \text{ intervals})Pr(\text{event happened in } n + 1 \text{ interval}) \\ &= (1 - \lambda\Delta)^n(\lambda\Delta) \end{aligned}$$

# Interarrival time of Poisson process

Using the similar analysis, we can also easily evaluate the distribution of interarrival time, the time that the next event will happen given that an event just happened. Let  $t = n\Delta$  and use the same notation as before

- Note that  $t > 0$  and  $\Delta \rightarrow 0$  and so  $n \rightarrow \infty$ . Now,
 
$$\begin{aligned} & Pr(\text{next event happened within in time } [t, t + \Delta]) \\ &= Pr(\text{next event happened within in time } [n\Delta, (n + 1)\Delta]) \\ &= Pr(\text{no event in first } n \text{ intervals})Pr(\text{event happened in } n + 1 \text{ interval}) \\ &= (1 - \lambda\Delta)^n(\lambda\Delta) \end{aligned}$$
- Let  $f_T(t)$  be the pdf of the interval time. Then,
 
$$f_T(t) = \frac{(1 - \lambda\Delta)^n(\lambda\Delta)}{\Delta}$$

# Interarrival time of Poisson process

Using the similar analysis, we can also easily evaluate the distribution of interarrival time, the time that the next event will happen given that an event just happened. Let  $t = n\Delta$  and use the same notation as before

- Note that  $t > 0$  and  $\Delta \rightarrow 0$  and so  $n \rightarrow \infty$ . Now,
 
$$\begin{aligned} & Pr(\text{next event happened within in time } [t, t + \Delta]) \\ &= Pr(\text{next event happened within in time } [n\Delta, (n + 1)\Delta]) \\ &= Pr(\text{no event in first } n \text{ intervals})Pr(\text{event happened in } n + 1 \text{ interval}) \\ &= (1 - \lambda\Delta)^n(\lambda\Delta) \end{aligned}$$
- Let  $f_T(t)$  be the pdf of the interval time. Then,
 
$$f_T(t) = \frac{(1 - \lambda\Delta)^n(\lambda\Delta)}{\Delta} = \lambda(1 - \lambda\frac{t}{n})^n$$



# Interarrival time of Poisson process

Using the similar analysis, we can also easily evaluate the distribution of interarrival time, the time that the next event will happen given that an event just happened. Let  $t = n\Delta$  and use the same notation as before

- Note that  $t > 0$  and  $\Delta \rightarrow 0$  and so  $n \rightarrow \infty$ . Now,
 
$$\begin{aligned} & Pr(\text{next event happened within in time } [t, t + \Delta]) \\ &= Pr(\text{next event happened within in time } [n\Delta, (n + 1)\Delta]) \\ &= Pr(\text{no event in first } n \text{ intervals})Pr(\text{event happened in } n + 1 \text{ interval}) \\ &= (1 - \lambda\Delta)^n(\lambda\Delta) \end{aligned}$$
- Let  $f_T(t)$  be the pdf of the interval time. Then,
 
$$f_T(t) = \frac{(1-\lambda\Delta)^n(\lambda\Delta)}{\Delta} = \lambda(1 - \lambda\frac{t}{n})^n = \lambda \exp(-\lambda t),$$
 where we use  $(1 + a/n)^n = \exp(a)$  again for  $n \rightarrow \infty$

# Interarrival time of Poisson process

Using the similar analysis, we can also easily evaluate the distribution of interarrival time, the time that the next event will happen given that an event just happened. Let  $t = n\Delta$  and use the same notation as before

- Note that  $t > 0$  and  $\Delta \rightarrow 0$  and so  $n \rightarrow \infty$ . Now,
 
$$\begin{aligned} & Pr(\text{next event happened within in time } [t, t + \Delta]) \\ &= Pr(\text{next event happened within in time } [n\Delta, (n + 1)\Delta]) \\ &= Pr(\text{no event in first } n \text{ intervals})Pr(\text{event happened in } n + 1 \text{ interval}) \\ &= (1 - \lambda\Delta)^n(\lambda\Delta) \end{aligned}$$
- Let  $f_T(t)$  be the pdf of the interval time. Then,
 
$$f_T(t) = \frac{(1-\lambda\Delta)^n(\lambda\Delta)}{\Delta} = \lambda(1 - \lambda\frac{t}{n})^n = \lambda \exp(-\lambda t),$$
 where we use  $(1 + a/n)^n = \exp(a)$  again for  $n \rightarrow \infty$

## Exponential distribution

$f_T(t) = \lambda \exp(-\lambda t) \triangleq \text{Exp}(t|\lambda)$  is the pdf of the exponential distribution with parameter  $\lambda$ . It is easy to verify that (as exercise)

- $E[T] = 1/\lambda$
- $\text{Var}(T) = 1/\lambda^2$

# Normal distribution revisit

For a univariate normal random variable, the pdf is given by

$$\begin{aligned} \text{Norm}(x|\mu, \sigma^2) &= \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) \\ &= \sqrt{\frac{\lambda}{2\pi}} \exp\left(-\frac{\lambda(x-\mu)^2}{2}\right) \end{aligned}$$

with

$$E[X|\mu, \sigma^2] = \mu,$$

$$E[(X - \mu)^2|\mu, \sigma^2] = \sigma^2,$$

Recall that  $\lambda = \frac{1}{\sigma^2}$  is the precision parameter that simplifies computations in many cases

# Conjugate prior of normal distribution for fixed $\sigma^2$

Consider  $\sigma^2$  fixed and  $\mu$  as the model parameter, then the posterior probability is given by

$$p(\mu|x; \sigma^2) \propto p(\mu, x; \sigma^2)$$

# Conjugate prior of normal distribution for fixed $\sigma^2$

Consider  $\sigma^2$  fixed and  $\mu$  as the model parameter, then the posterior probability is given by

$$\begin{aligned} p(\mu|x; \sigma^2) &\propto p(\mu, x; \sigma^2) \\ &= p(\mu) \text{Norm}(x|\mu; \sigma^2) \end{aligned}$$

# Conjugate prior of normal distribution for fixed $\sigma_2$

Consider  $\sigma^2$  fixed and  $\mu$  as the model parameter, then the posterior probability is given by

$$\begin{aligned} p(\mu|x; \sigma^2) &\propto p(\mu, x; \sigma^2) \\ &= p(\mu) \text{Norm}(x|\mu; \sigma^2) \\ &\propto p(\mu) \exp\left(-\frac{(x - \mu)^2}{2\sigma^2}\right) \end{aligned}$$

# Conjugate prior of normal distribution for fixed $\sigma^2$

Consider  $\sigma^2$  fixed and  $\mu$  as the model parameter, then the posterior probability is given by

$$\begin{aligned} p(\mu|x; \sigma^2) &\propto p(\mu, x; \sigma^2) \\ &= p(\mu) \text{Norm}(x|\mu; \sigma^2) \\ &\propto p(\mu) \exp\left(-\frac{(x - \mu)^2}{2\sigma^2}\right) \end{aligned}$$

It is apparent that the posterior will keep the same form if  $p(\mu)$  is also normal. Therefore, normal distribution is the conjugate prior of itself for fixed variance

# Posterior distribution of normal variable for fixed $\sigma^2$

Given prior  $p(\mu) = \text{Norm}(\mu|\mu_0, \sigma_0^2)$  and likelihood  $\text{Norm}(x|\mu; \sigma^2)$ . Let's find the posterior probability,

$$\begin{aligned} & p(\mu|x; \sigma^2, \mu_0, \sigma_0^2) \\ &= \text{Const} \cdot \text{Norm}(\mu|\mu_0, \sigma_0^2) \text{Norm}(x|\mu; \sigma^2) \end{aligned}$$



Posterior distribution of normal variable for fixed  $\sigma^2$ 

Given prior  $p(\mu) = \text{Norm}(\mu|\mu_0, \sigma_0^2)$  and likelihood  $\text{Norm}(x|\mu; \sigma^2)$ . Let's find the posterior probability,

$$\begin{aligned} & p(\mu|x; \sigma^2, \mu_0, \sigma_0^2) \\ &= \text{Const} \cdot \text{Norm}(\mu|\mu_0, \sigma_0^2) \text{Norm}(x|\mu; \sigma^2) \\ &= \text{Const2} \cdot \exp\left(-\frac{(x-\mu)^2}{2\sigma^2} - \frac{(\mu-\mu_0)^2}{2\sigma_0^2}\right) \\ &= \text{Norm}(\mu; \tilde{\mu}, \tilde{\sigma}^2), \end{aligned}$$

# Posterior distribution of normal variable for fixed $\sigma^2$

Given prior  $p(\mu) = \text{Norm}(\mu|\mu_0, \sigma_0^2)$  and likelihood  $\text{Norm}(x|\mu; \sigma^2)$ . Let's find the posterior probability,

$$\begin{aligned}
 & p(\mu|x; \sigma^2, \mu_0, \sigma_0^2) \\
 &= \text{Const} \cdot \text{Norm}(\mu|\mu_0, \sigma_0^2) \text{Norm}(x|\mu; \sigma^2) \\
 &= \text{Const2} \cdot \exp\left(-\frac{(x-\mu)^2}{2\sigma^2} - \frac{(\mu-\mu_0)^2}{2\sigma_0^2}\right) \\
 &= \text{Norm}(\mu; \tilde{\mu}, \tilde{\sigma}^2),
 \end{aligned}$$

where  $\tilde{\mu} = \frac{\sigma_0^2 x + \mu_0 \sigma^2}{\sigma_0^2 + \sigma^2}$  and  $\tilde{\sigma}^2 = \frac{\sigma_0^2 \sigma^2}{\sigma_0^2 + \sigma^2}$ . Alternatively,  $\tilde{\lambda} = \lambda_0 + \lambda$  and  $\tilde{\mu} = \frac{\lambda}{\tilde{\lambda}} x + \frac{\lambda_0}{\tilde{\lambda}} \mu_0$ . Note that we have already come across the more general expression when we studied product of multivariate normal distribution

# Conjugate prior of normal distribution for fixed $\mu$

Consider  $\mu$  fixed and  $\lambda$  as the model parameter

$$p(x|\lambda; \mu) \propto p(x, \lambda; \mu) = p(\lambda) \text{Norm}(x|\lambda; \mu)$$

# Conjugate prior of normal distribution for fixed $\mu$

Consider  $\mu$  fixed and  $\lambda$  as the model parameter

$$\begin{aligned} p(x|\lambda; \mu) &\propto p(x, \lambda; \mu) = p(\lambda) \text{Norm}(x|\lambda; \mu) \\ &\propto p(\lambda) \sqrt{\lambda} \exp\left(-\frac{\lambda(x - \mu)^2}{2}\right) \end{aligned}$$

# Conjugate prior of normal distribution for fixed $\mu$

Consider  $\mu$  fixed and  $\lambda$  as the model parameter

$$\begin{aligned} p(x|\lambda; \mu) &\propto p(x, \lambda; \mu) = p(\lambda) \text{Norm}(x|\lambda; \mu) \\ &\propto p(\lambda) \sqrt{\lambda} \exp\left(-\frac{\lambda(x - \mu)^2}{2}\right) \end{aligned}$$

More generally, when we have  $N$  observations from the same source,

$$\begin{aligned} p(x_1, \dots, x_N, \lambda; \mu) &= p(\lambda) \prod_{i=1}^N \text{Norm}(x_i|\lambda; \mu) \\ &\propto p(\lambda) \lambda^{\frac{N}{2}} \exp\left(-\lambda \sum_{i=1}^N \frac{(x_i - \mu)^2}{2}\right) \end{aligned}$$

# Conjugate prior of normal distribution for fixed $\mu$

Consider  $\mu$  fixed and  $\lambda$  as the model parameter

$$\begin{aligned} p(x|\lambda; \mu) &\propto p(x, \lambda; \mu) = p(\lambda) \text{Norm}(x|\lambda; \mu) \\ &\propto p(\lambda) \sqrt{\lambda} \exp\left(-\frac{\lambda(x - \mu)^2}{2}\right) \end{aligned}$$

More generally, when we have  $N$  observations from the same source,

$$\begin{aligned} p(x_1, \dots, x_N, \lambda; \mu) &= p(\lambda) \prod_{i=1}^N \text{Norm}(x_i|\lambda; \mu) \\ &\propto p(\lambda) \lambda^{\frac{N}{2}} \exp\left(-\lambda \sum_{i=1}^N \frac{(x_i - \mu)^2}{2}\right) \end{aligned}$$

From inspection, the conjugate prior should have a form  $\lambda^a \exp(-b\lambda)$

# Gamma distribution

The distribution with the desired form described in previous slide turns out to be the Gamma distribution. Its pdf, mean, and variance (please verify the mean and variance) are given by

$$\text{Gamma}(\lambda|a, b) = \frac{1}{\Gamma(a)} b^a \lambda^{a-1} \exp(-b\lambda)$$

$$E[\lambda] = \frac{a}{b}$$

$$\text{Var}[\lambda] = \frac{a}{b^2},$$

where  $a, b > 0$  and  $\lambda \geq 0$

# Gamma distribution

The distribution with the desired form described in previous slide turns out to be the Gamma distribution. Its pdf, mean, and variance (please verify the mean and variance) are given by

$$\text{Gamma}(\lambda|a, b) = \frac{1}{\Gamma(a)} b^a \lambda^{a-1} \exp(-b\lambda)$$

$$E[\lambda] = \frac{a}{b}$$

$$\text{Var}[\lambda] = \frac{a}{b^2},$$

where  $a, b > 0$  and  $\lambda \geq 0$

N.B. when  $a = 1$ , Gamma reduces to the exponential distribution. When  $a$  is integer, it reduces to Erlang distribution



# Posterior distribution of normal variable for fixed $\mu$

Posterior probability given Normal likelihood (fixed mean) and Gamma prior

$$p(\lambda|x, a, b; \mu) = \text{Const1} \cdot \text{Gamma}(\lambda|a, b) \text{Norm}(x|\lambda; \mu)$$

# Posterior distribution of normal variable for fixed $\mu$

Posterior probability given Normal likelihood (fixed mean) and Gamma prior

$$\begin{aligned}
 p(\lambda|x, a, b; \mu) &= \text{Const1} \cdot \text{Gamma}(\lambda|a, b) \text{Norm}(x|\lambda; \mu) \\
 &= \text{Const2} \cdot \lambda^{a-1} \exp(-b\lambda) \sqrt{\lambda} \exp\left(-\lambda \frac{(x-\mu)^2}{2}\right) \\
 &= \text{Gamma}\left(\lambda; \tilde{a}, \tilde{b}\right),
 \end{aligned}$$

where  $\tilde{a} \leftarrow a + \frac{1}{2}$  and  $\tilde{b} \leftarrow b + \frac{(x-\mu)^2}{2}$

# Conjugate prior summary

Distribution	Likelihood $p(\mathbf{x} \theta)$	Prior $p(\theta)$	Distribution
Bernoulli	$(1 - \theta)^{(1-x)}\theta^x$	$\propto (1 - \theta)^{(a-1)}\theta^{(b-1)}$	Beta
Binomial	$\propto (1 - \theta)^{(N-x)}\theta^x$	$\propto (1 - \theta)^{(a-1)}\theta^{(b-1)}$	Beta
Multinomial	$\propto \theta_1^{x_1}\theta_2^{x_2}\theta_3^{x_3}$	$\propto \theta_1^{\alpha_1-1}\theta_2^{\alpha_2-1}\theta_3^{\alpha_3-1}$	Dirichlet
Normal (fixed $\sigma^2$ )	$\propto \exp\left(-\frac{(x-\theta)^2}{2\sigma^2}\right)$	$\propto \exp\left(-\frac{(\theta-\mu_0)^2}{2\sigma_0^2}\right)$	Normal
Normal (fixed $\mu$ )	$\propto \sqrt{\theta} \exp\left(-\frac{\theta(x-\mu)^2}{2}\right)$	$\propto \theta^{a-1} \exp(-b\theta)$	Gamma
Poisson	$\propto \theta^x \exp(-\theta)$	$\propto \theta^{a-1} \exp(-b\theta)$	Gamma

# An example

- Simple economy:  $m$  prosumers,  $n$  different goods<sup>8</sup>
- Each individual: production  $\mathbf{p}_i \in \mathbb{R}_n$ , consumption  $\mathbf{c}_i \in \mathbb{R}_n$
- Expense of producing “ $\mathbf{p}$ ” for agent  $i = e_i(\mathbf{p})$
- Utility (happiness) of consuming “ $\mathbf{c}$ ” units for agent  $i = u_i(\mathbf{c})$
- Maximize happiness

$$\max_{\mathbf{p}_i, \mathbf{c}_i} \sum_i (u_i(\mathbf{c}_i) - e_i(\mathbf{p}_i)) \quad \text{s.t.} \quad \sum_i \mathbf{c}_i = \sum_i \mathbf{p}_i$$

<sup>8</sup>Example borrowed from the first lecture of Prof Gordon's CMU CS 10-725

# Walrasian equilibrium

$$\max_{\mathbf{p}_i, \mathbf{c}_i} \sum_i (u_i(\mathbf{c}_i) - e_i(\mathbf{p}_i)) \quad s.t. \quad \sum_i \mathbf{c}_i = \sum_i \mathbf{p}_i$$

- Idea: introduce price  $\lambda_j$  to each good  $j$ . Let the market decide
  - Price  $\lambda_j \uparrow$ : consumption of good  $j \downarrow$ , production of good  $j \uparrow$
  - Price  $\lambda_j \downarrow$ : consumption of good  $j \uparrow$ , production of good  $j \downarrow$
  - Can adjust price until consumption = production for each good

# Algorithm: tâtonnement

Assume that the appropriate prices are found, we can ignore the equality constraint, then the problem becomes

$$\max_{\mathbf{p}_i, \mathbf{c}_i} \sum_i (u_i(\mathbf{c}_i) - e_i(\mathbf{p}_i)) \quad \Rightarrow \quad \sum_i \max_{\mathbf{p}_i, \mathbf{c}_i} (u_i(\mathbf{c}_i) - e_i(\mathbf{p}_i))$$

So we can simply optimize production and consumption of each individual independently

---

## Algorithm 1 tâtonnement

---

- 1: **procedure** FINDBESTPRICES
  - 2:      $\lambda \leftarrow [0, 0, \dots, 0]$
  - 3:     **for**  $k = 1, 2, \dots$  **do**
  - 4:         Each individual solves for its  $c_i$  and  $p_i$  for the given  $\lambda$
  - 5:          $\lambda \leftarrow \lambda + \delta_k \sum_i (c_i - p_i)$
-

# Lagrange multiplier

## Problem

$$\begin{aligned} \max_{\mathbf{x}} f(\mathbf{x}) \\ g(\mathbf{x}) = 0 \end{aligned}$$

Consider  $L(\mathbf{x}, \lambda) = f(\mathbf{x}) - \lambda g(\mathbf{x})$  and let  $\tilde{f}(\mathbf{x}) = \min_{\lambda} L(\mathbf{x}, \lambda)$ .

# Lagrange multiplier

## Problem

$$\begin{aligned} \max_{\mathbf{x}} f(\mathbf{x}) \\ g(\mathbf{x}) = 0 \end{aligned}$$

Consider  $L(\mathbf{x}, \lambda) = f(\mathbf{x}) - \lambda g(\mathbf{x})$  and let  $\tilde{f}(\mathbf{x}) = \min_{\lambda} L(\mathbf{x}, \lambda)$ . Note that

$$\tilde{f}(\mathbf{x}) = \begin{cases} f(\mathbf{x}) & \text{if } g(\mathbf{x}) = 0 \\ -\infty & \text{otherwise} \end{cases}$$



# Lagrange multiplier

## Problem

$$\begin{aligned} \max_{\mathbf{x}} f(\mathbf{x}) \\ g(\mathbf{x}) = 0 \end{aligned}$$

Consider  $L(\mathbf{x}, \lambda) = f(\mathbf{x}) - \lambda g(\mathbf{x})$  and let  $\tilde{f}(\mathbf{x}) = \min_{\lambda} L(\mathbf{x}, \lambda)$ . Note that

$$\tilde{f}(\mathbf{x}) = \begin{cases} f(\mathbf{x}) & \text{if } g(\mathbf{x}) = 0 \\ -\infty & \text{otherwise} \end{cases}$$

Therefore, the problem is identical to  $\max_{\mathbf{x}} \tilde{f}(\mathbf{x})$  or

$$\max_{\mathbf{x}} \min_{\lambda} (f(\mathbf{x}) - \lambda g(\mathbf{x})),$$

where  $\lambda$  is known to be the Lagrange multiplier.

# Lagrange multiplier (con't)

Assume the optimum is a saddle point,

$$\max_{\mathbf{x}} \min_{\lambda} (f(\mathbf{x}) - \lambda g(\mathbf{x})) = \min_{\lambda} \max_{\mathbf{x}} (f(\mathbf{x}) - \lambda g(\mathbf{x})),$$

the R.H.S. implies

$$\nabla f(\mathbf{x}) = \lambda \nabla g(\mathbf{x})$$

# Inequality constraint

## Problem

$$\begin{aligned} \max_{\mathbf{x}} f(\mathbf{x}) \\ g(\mathbf{x}) \leq 0 \end{aligned}$$

Consider  $\tilde{f}(\mathbf{x}) = \min_{\lambda \geq 0} (f(\mathbf{x}) - \lambda g(\mathbf{x}))$ ,

# Inequality constraint

## Problem

$$\begin{aligned} \max_{\mathbf{x}} \quad & f(\mathbf{x}) \\ g(\mathbf{x}) \leq & 0 \end{aligned}$$

Consider  $\tilde{f}(\mathbf{x}) = \min_{\lambda \geq 0} (f(\mathbf{x}) - \lambda g(\mathbf{x}))$ , note that

$$\tilde{f}(\mathbf{x}) = \begin{cases} f(\mathbf{x}) & \text{if } g(\mathbf{x}) \leq 0 \\ -\infty & \text{otherwise} \end{cases}$$

# Inequality constraint

## Problem

$$\begin{aligned} \max_{\mathbf{x}} \quad & f(\mathbf{x}) \\ g(\mathbf{x}) \leq & 0 \end{aligned}$$

Consider  $\tilde{f}(\mathbf{x}) = \min_{\lambda \geq 0} (f(\mathbf{x}) - \lambda g(\mathbf{x}))$ , note that

$$\tilde{f}(\mathbf{x}) = \begin{cases} f(\mathbf{x}) & \text{if } g(\mathbf{x}) \leq 0 \\ -\infty & \text{otherwise} \end{cases}$$

Therefore, we can rewrite the problem as

$$\max_{\mathbf{x}} \min_{\lambda \geq 0} (f(\mathbf{x}) - \lambda g(\mathbf{x}))$$

# Inequality constraint (con't)

Assume

$$\max_{\mathbf{x}} \min_{\lambda \geq 0} (f(\mathbf{x}) - \lambda g(\mathbf{x})) = \min_{\lambda \geq 0} \max_{\mathbf{x}} (f(\mathbf{x}) - \lambda g(\mathbf{x}))$$

The R.H.S. implies

$$\nabla f(\mathbf{x}) = \lambda \nabla g(\mathbf{x})$$

# Inequality constraint (con't)

Assume

$$\max_{\mathbf{x}} \min_{\lambda \geq 0} (f(\mathbf{x}) - \lambda g(\mathbf{x})) = \min_{\lambda \geq 0} \max_{\mathbf{x}} (f(\mathbf{x}) - \lambda g(\mathbf{x}))$$

The R.H.S. implies

$$\nabla f(\mathbf{x}) = \lambda \nabla g(\mathbf{x})$$

Moreover, at the optimum point  $(\mathbf{x}^*, \lambda^*)$ , we should have the so-called “complementary slackness” condition

$$\lambda^* g(\mathbf{x}^*) = 0$$

since

$$\max_{\substack{\mathbf{x} \\ g(\mathbf{x}) \leq 0}} f(\mathbf{x}) \equiv \max_{\mathbf{x}} \min_{\lambda \geq 0} (f(\mathbf{x}) - \lambda g(\mathbf{x}))$$

# Karush-Kuhn-Tucker conditions

## Problem

$$\begin{aligned} \max_{\mathbf{x}} f(\mathbf{x}) \\ g(\mathbf{x}) \leq 0, \quad h(\mathbf{x}) = 0 \end{aligned}$$

## Conditions

$$\begin{aligned} \nabla f(\mathbf{x}^*) - \mu^* \nabla g(\mathbf{x}^*) - \lambda^* \nabla h(\mathbf{x}^*) &= 0 \\ g(\mathbf{x}^*) &\leq 0 \\ h(\mathbf{x}^*) &= 0 \\ \mu^* &\geq 0 \\ \mu^* g(\mathbf{x}^*) &= 0 \end{aligned}$$



# Overview of source coding

- The objective of “source coding” is to compress some source

# Overview of source coding

- The objective of “source coding” is to compress some source
- We can think of compression as “coding”. Meaning that we replace each input by a corresponding coded sequence. So encoding is just a mapping/function process

# Overview of source coding

- The objective of “source coding” is to compress some source
- We can think of compression as “coding”. Meaning that we replace each input by a corresponding coded sequence. So encoding is just a mapping/function process
- Without loss of generality, we can use binary domain for our coded sequence. So for each input message, it is converted to a sequence of 1s and 0s

# Overview of source coding

- The objective of “source coding” is to compress some source
- We can think of compression as “coding”. Meaning that we replace each input by a corresponding coded sequence. So encoding is just a mapping/function process
- Without loss of generality, we can use binary domain for our coded sequence. So for each input message, it is converted to a sequence of 1s and 0s
- Consider encoding (compressing) a sequence  $x_1, x_2, \dots$  one symbol at a time, resulting  $c(x_1), c(x_2), \dots$

# Overview of source coding

- The objective of “source coding” is to compress some source
- We can think of compression as “coding”. Meaning that we replace each input by a corresponding coded sequence. So encoding is just a mapping/function process
- Without loss of generality, we can use binary domain for our coded sequence. So for each input message, it is converted to a sequence of 1s and 0s
- Consider encoding (compressing) a sequence  $x_1, x_2, \dots$  one symbol at a time, resulting  $c(x_1), c(x_2), \dots$
- Denote the lengths of  $x_1, x_2, \dots$  as  $l(x_1), l(x_2), \dots$ , one of the major goal is to have  $E[l(X)]$  to be as small as possible

# Overview of source coding

- The objective of “source coding” is to compress some source
- We can think of compression as “coding”. Meaning that we replace each input by a corresponding coded sequence. So encoding is just a mapping/function process
- Without loss of generality, we can use binary domain for our coded sequence. So for each input message, it is converted to a sequence of 1s and 0s
- Consider encoding (compressing) a sequence  $x_1, x_2, \dots$  one symbol at a time, resulting  $c(x_1), c(x_2), \dots$
- Denote the lengths of  $x_1, x_2, \dots$  as  $l(x_1), l(x_2), \dots$ , one of the major goal is to have  $E[l(X)]$  to be as small as possible
- However, we want to make sure that we can losslessly decode the message also!

# Uniquely decodable code

- To ensure that we can recover message without loss, we must make sure that no message share the same codeword

# Uniquely decodable code

- To ensure that we can recover message without loss, we must make sure that no message share the same codeword
- We say a code is “singular” (broken) if  $c(x_1) = c(x_2)$  for some different  $x_1$  and  $x_2$



# Uniquely decodable code

- To ensure that we can recover message without loss, we must make sure that no message share the same codeword
- We say a code is “singular” (broken) if  $c(x_1) = c(x_2)$  for some different  $x_1$  and  $x_2$
- Even when a code is not “singular”, we still cannot guarantee that we can always recover the original message losslessly, consider 4 different possible input symbols  $a, b, c, d$  and an encoding map  $c(\cdot)$  :
  - $a \mapsto 0, b \mapsto 1, c \mapsto 10, d \mapsto 11$
  - What should be the message for 1110?

# Uniquely decodable code

- To ensure that we can recover message without loss, we must make sure that no message share the same codeword
- We say a code is “singular” (broken) if  $c(x_1) = c(x_2)$  for some different  $x_1$  and  $x_2$
- Even when a code is not “singular”, we still cannot guarantee that we can always recover the original message losslessly, consider 4 different possible input symbols  $a, b, c, d$  and an encoding map  $c(\cdot)$  :
  - $a \mapsto 0, b \mapsto 1, c \mapsto 10, d \mapsto 11$
  - What should be the message for 1110?
    - *dba? Or bbba?*

# Uniquely decodable code

- To ensure that we can recover message without loss, we must make sure that no message share the same codeword
- We say a code is “singular” (broken) if  $c(x_1) = c(x_2)$  for some different  $x_1$  and  $x_2$
- Even when a code is not “singular”, we still cannot guarantee that we can always recover the original message losslessly, consider 4 different possible input symbols  $a, b, c, d$  and an encoding map  $c(\cdot)$  :
  - $a \mapsto 0, b \mapsto 1, c \mapsto 10, d \mapsto 11$
  - What should be the message for 1110?
    - *dba? Or bbba?*
- So it is not sufficient to just have  $c(\cdot)$  to map to different output for each input. Let's overload the notation  $c(\cdot)$  a little bit and for any message sequence  $\mathbf{x} = x_1, x_2, \dots, x_n$ , encode sequence  $x_1, x_2, \dots, x_n$  to  $c(\mathbf{x}) = c(x_1, x_2, \dots, x_n) = c(x_1)c(x_2) \cdots c(x_n)$

# Uniquely decodable code

- To ensure that we can recover message without loss, we must make sure that no message share the same codeword
- We say a code is “singular” (broken) if  $c(x_1) = c(x_2)$  for some different  $x_1$  and  $x_2$
- Even when a code is not “singular”, we still cannot guarantee that we can always recover the original message losslessly, consider 4 different possible input symbols  $a, b, c, d$  and an encoding map  $c(\cdot)$  :
  - $a \mapsto 0, b \mapsto 1, c \mapsto 10, d \mapsto 11$
  - What should be the message for 1110?
    - *dba? Or bbba?*
- So it is not sufficient to just have  $c(\cdot)$  to map to different output for each input. Let's overload the notation  $c(\cdot)$  a little bit and for any message sequence  $\mathbf{x} = x_1, x_2, \dots, x_n$ , encode sequence  $x_1, x_2, \dots, x_n$  to  $c(\mathbf{x}) = c(x_1, x_2, \dots, x_n) = c(x_1)c(x_2) \dots c(x_n)$ 
  - We say  $c(\mathbf{x})$  is **uniquely decodable** if all input sequences map to different outputs

# Prefix-free code

- For practical purpose, we would like to be able to decode a symbol “once it is available”. Consider a code with map
  - $a \mapsto 10, b \mapsto 00, c \mapsto 11, d \mapsto 110$

# Prefix-free code

- For practical purpose, we would like to be able to decode a symbol “once it is available”. Consider a code with map
  - $a \mapsto 10, b \mapsto 00, c \mapsto 11, d \mapsto 110$
  - One can show that it is uniquely decodable. However, consider an input sequence  $cbbb \mapsto 11000000$

# Prefix-free code

- For practical purpose, we would like to be able to decode a symbol “once it is available”. Consider a code with map
  - $a \mapsto 10, b \mapsto 00, c \mapsto 11, d \mapsto 110$
  - One can show that it is uniquely decodable. However, consider an input sequence  $cbbb \mapsto 11000000$
  - When the decoder read the first 3 bits, it is not able to determine if the first input symbol is  $c$  or  $d$

# Prefix-free code

- For practical purpose, we would like to be able to decode a symbol “once it is available”. Consider a code with map
  - $a \mapsto 10, b \mapsto 00, c \mapsto 11, d \mapsto 110$
  - One can show that it is uniquely decodable. However, consider an input sequence  $cbbb \mapsto 11000000$
  - When the decoder read the first 3 bits, it is not able to determine if the first input symbol is  $c$  or  $d$
  - Actually, it will be until the decoder read the last bit that it will be able to confirm that the first input symbol is  $c$ . It is definitely not something very desirable



# Prefix-free code

- For practical purpose, we would like to be able to decode a symbol “once it is available”. Consider a code with map
  - $a \mapsto 10, b \mapsto 00, c \mapsto 11, d \mapsto 110$
  - One can show that it is uniquely decodable. However, consider an input sequence  $cbbb \mapsto 11000000$
  - When the decoder read the first 3 bits, it is not able to determine if the first input symbol is  $c$  or  $d$
  - Actually, it will be until the decoder read the last bit that it will be able to confirm that the first input symbol is  $c$ . It is definitely not something very desirable
- Instead, for a mapping  $a \mapsto 1, b \mapsto 01, c \mapsto 001, d \mapsto 0001$ , I will argue that we can always decode a symbol “once it is available”

# Prefix-free code

- For practical purpose, we would like to be able to decode a symbol “once it is available”. Consider a code with map
  - $a \mapsto 10, b \mapsto 00, c \mapsto 11, d \mapsto 110$
  - One can show that it is uniquely decodable. However, consider an input sequence  $cbbb \mapsto 11000000$
  - When the decoder read the first 3 bits, it is not able to determine if the first input symbol is  $c$  or  $d$
  - Actually, it will be until the decoder read the last bit that it will be able to confirm that the first input symbol is  $c$ . It is definitely not something very desirable
- Instead, for a mapping  $a \mapsto 1, b \mapsto 01, c \mapsto 001, d \mapsto 0001$ , I will argue that we can always decode a symbol “once it is available”
  - Note that the catch is that there is no codeword being the “prefix” of another codeword
  - We call such code a prefix-free code or an instantaneous code

# Kraft's Inequality

Let  $l_1, l_2, \dots, l_K$  satisfy  $\sum_{k=1}^K 2^{-l_k} \leq 1$ . Then, there exists a uniquely decodable code for symbols  $x_1, x_2, \dots, x_K$  such that  $l(x_1) = l_1$ ,  $l(x_2) = l_2, \dots, l(x_K) = l_K$ .

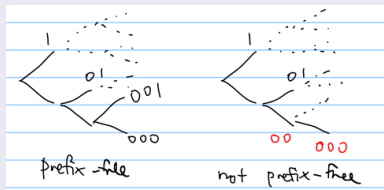
# Kraft's Inequality

Let  $l_1, l_2, \dots, l_K$  satisfy  $\sum_{k=1}^K 2^{-l_k} \leq 1$ . Then, there exists a uniquely decodable code for symbols  $x_1, x_2, \dots, x_K$  such that  $l(x_1) = l_1$ ,  $l(x_2) = l_2, \dots, l(x_K) = l_K$ .

## Intuition

Consider # “descendants” of each codeword at the “ $l_{max}$ ”-level, then for prefix-free code, we have

$$\sum_{k=1}^K 2^{l_{max}-l_k} \leq 2^{l_{max}}$$



# Kraft's Inequality

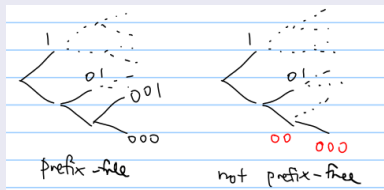
Let  $l_1, l_2, \dots, l_K$  satisfy  $\sum_{k=1}^K 2^{-l_k} \leq 1$ . Then, there exists a uniquely decodable code for symbols  $x_1, x_2, \dots, x_K$  such that  $l(x_1) = l_1$ ,  $l(x_2) = l_2, \dots, l(x_K) = l_K$ .

## Intuition

Consider # “descendants” of each codeword at the “ $l_{max}$ ”-level, then for prefix-free code, we have

$$\sum_{k=1}^K 2^{l_{max}-l_k} \leq 2^{l_{max}}$$

$$\Rightarrow \sum_{k=1}^K 2^{-l_k} \leq 1$$



# Forward Proof

Given  $l_1, l_2, \dots, l_K$  satisfy  $\sum_{k=1}^K 2^{-l_k} \leq 1$ , we can assign nodes on a tree as previous slides. More precisely,

- Assign  $i$ -th node as a node at level  $l_i$ , then cross out all its descendants
- Repeat the procedure for  $i$  from 1 to  $K$
- We know that there are sufficient tree nodes to be assigned since the Kraft's inequality is satisfied

The corresponding code is apparently prefix-free and thus is uniquely decodable

# Converse Proof

Consider message from coding  $k$  symbols  $\mathbf{x} = x_1, x_2, \dots, x_k$

$$\begin{aligned} \left( \sum_{\mathbf{x} \in \mathcal{X}} 2^{-l(\mathbf{x})} \right)^k &= \left( \sum_{x_1 \in \mathcal{X}} 2^{-l(x_1)} \right) \left( \sum_{x_2 \in \mathcal{X}} 2^{-l(x_2)} \right) \dots \left( \sum_{x_k \in \mathcal{X}} 2^{-l(x_k)} \right) \\ &= \sum_{x_1, x_2, \dots, x_k \in \mathcal{X}^k} 2^{-l(x_1) + l(x_2) + \dots + l(x_k)} \\ &= \sum_{\mathbf{x} \in \mathcal{X}^k} 2^{-l(\mathbf{x})} \end{aligned}$$

# Converse Proof

Consider message from coding  $k$  symbols  $\mathbf{x} = x_1, x_2, \dots, x_k$

$$\begin{aligned}
 \left( \sum_{x \in \mathcal{X}} 2^{-l(x)} \right)^k &= \left( \sum_{x_1 \in \mathcal{X}} 2^{-l(x_1)} \right) \left( \sum_{x_2 \in \mathcal{X}} 2^{-l(x_2)} \right) \dots \left( \sum_{x_k \in \mathcal{X}} 2^{-l(x_k)} \right) \\
 &= \sum_{x_1, x_2, \dots, x_k \in \mathcal{X}^k} 2^{-l(x_1) + l(x_2) + \dots + l(x_k)} \\
 &= \sum_{\mathbf{x} \in \mathcal{X}^k} 2^{-l(\mathbf{x})} = \sum_{m=1}^{kl_{\max}} a(m) 2^{-m},
 \end{aligned}$$

where  $a(m)$  is the number of codeword with length  $m$ . However, for the code to be uniquely decodable,  $a(m) \leq 2^m$ , where  $2^m$  is the number of available codewords with length  $m$ .



# Converse Proof

Consider message from coding  $k$  symbols  $\mathbf{x} = x_1, x_2, \dots, x_k$

$$\begin{aligned} \left( \sum_{x \in \mathcal{X}} 2^{-l(x)} \right)^k &= \left( \sum_{x_1 \in \mathcal{X}} 2^{-l(x_1)} \right) \left( \sum_{x_2 \in \mathcal{X}} 2^{-l(x_2)} \right) \dots \left( \sum_{x_k \in \mathcal{X}} 2^{-l(x_k)} \right) \\ &= \sum_{x_1, x_2, \dots, x_k \in \mathcal{X}^k} 2^{-l(x_1) + l(x_2) + \dots + l(x_k)} \\ &= \sum_{x \in \mathcal{X}^k} 2^{-l(x)} = \sum_{m=1}^{kl_{\max}} a(m) 2^{-m}, \end{aligned}$$

where  $a(m)$  is the number of codeword with length  $m$ . However, for the code to be uniquely decodable,  $a(m) \leq 2^m$ , where  $2^m$  is the number of available codewords with length  $m$ . Therefore,

$$\sum_{x \in \mathcal{X}} 2^{-l(x)} \leq (kl_{\max})^{1/k}$$

# Converse Proof

Consider message from coding  $k$  symbols  $\mathbf{x} = x_1, x_2, \dots, x_k$

$$\begin{aligned} \left( \sum_{x \in \mathcal{X}} 2^{-l(x)} \right)^k &= \left( \sum_{x_1 \in \mathcal{X}} 2^{-l(x_1)} \right) \left( \sum_{x_2 \in \mathcal{X}} 2^{-l(x_2)} \right) \dots \left( \sum_{x_k \in \mathcal{X}} 2^{-l(x_k)} \right) \\ &= \sum_{x_1, x_2, \dots, x_k \in \mathcal{X}^k} 2^{-l(x_1) + l(x_2) + \dots + l(x_k)} \\ &= \sum_{x \in \mathcal{X}^k} 2^{-l(x)} = \sum_{m=1}^{kl_{\max}} a(m) 2^{-m}, \end{aligned}$$

where  $a(m)$  is the number of codeword with length  $m$ . However, for the code to be uniquely decodable,  $a(m) \leq 2^m$ , where  $2^m$  is the number of available codewords with length  $m$ . Therefore,

$$\sum_{x \in \mathcal{X}} 2^{-l(x)} \leq (kl_{\max})^{1/k} \approx 1 \text{ as } k \rightarrow \infty$$

# Minimum rate required to compress a source

$$\begin{aligned} & \min_{l_1, l_2, \dots, l_K} \sum_{k=1}^K p_k l_k \text{ subject to } \sum_{k=1}^K 2^{-l_k} \leq 1 \text{ and } l_1, \dots, l_K \geq 0 \\ & \equiv \max_{l_1, l_2, \dots, l_K} - \sum_{k=1}^K p_k l_k \text{ subject to } \sum_{k=1}^K 2^{-l_k} - 1 \leq 0 \text{ and } -l_1, \dots, -l_K \leq 0 \end{aligned}$$

## KKT conditions

$$-\nabla \left( \sum_{k=1}^K p_k l_k \right) - \mu_0 \nabla \left( \sum_{k=1}^K 2^{-l_k} - 1 \right) + \sum_{k=1}^K \mu_k \nabla l_k = 0$$

# Minimum rate required to compress a source

$$\min_{l_1, l_2, \dots, l_K} \sum_{k=1}^K p_k l_k \text{ subject to } \sum_{k=1}^K 2^{-l_k} \leq 1 \text{ and } l_1, \dots, l_K \geq 0$$

$$\equiv \max_{l_1, l_2, \dots, l_K} - \sum_{k=1}^K p_k l_k \text{ subject to } \sum_{k=1}^K 2^{-l_k} - 1 \leq 0 \text{ and } -l_1, \dots, -l_K \leq 0$$

## KKT conditions

$$-\nabla \left( \sum_{k=1}^K p_k l_k \right) - \mu_0 \nabla \left( \sum_{k=1}^K 2^{-l_k} - 1 \right) + \sum_{k=1}^K \mu_k \nabla l_k = 0$$

$$\sum_{k=1}^K 2^{-l_k} - 1 \leq 0, \quad l_1, \dots, l_K \geq 0, \quad \mu_0, \mu_1, \dots, \mu_K \geq 0$$

# Minimum rate required to compress a source

$$\begin{aligned} & \min_{l_1, l_2, \dots, l_K} \sum_{k=1}^K p_k l_k \text{ subject to } \sum_{k=1}^K 2^{-l_k} \leq 1 \text{ and } l_1, \dots, l_K \geq 0 \\ & \equiv \max_{l_1, l_2, \dots, l_K} - \sum_{k=1}^K p_k l_k \text{ subject to } \sum_{k=1}^K 2^{-l_k} - 1 \leq 0 \text{ and } -l_1, \dots, -l_K \leq 0 \end{aligned}$$

## KKT conditions

$$-\nabla \left( \sum_{k=1}^K p_k l_k \right) - \mu_0 \nabla \left( \sum_{k=1}^K 2^{-l_k} - 1 \right) + \sum_{k=1}^K \mu_k \nabla l_k = 0$$

$$\sum_{k=1}^K 2^{-l_k} - 1 \leq 0, \quad l_1, \dots, l_K \geq 0, \quad \mu_0, \mu_1, \dots, \mu_K \geq 0$$

$$\mu_0 \left( \sum_{k=1}^K 2^{-l_k} - 1 \right) = 0, \quad \mu_k l_k = 0$$

# Minimum rate required to compress a source

Since we expect  $I_k > 0$ ,  $\mu_k = 0$ .

# Minimum rate required to compress a source

Since we expect  $l_k > 0$ ,  $\mu_k = 0$ . Expand the first equation, we get

$$-p_j + \mu_0 2^{-l_j} \log 2 = 0 \Rightarrow 2^{-l_j} = \frac{p_j}{\mu_0 \log 2}$$

# Minimum rate required to compress a source

Since we expect  $l_k > 0$ ,  $\mu_k = 0$ . Expand the first equation, we get

$$-p_j + \mu_0 2^{-l_j} \log 2 = 0 \Rightarrow 2^{-l_j} = \frac{p_j}{\mu_0 \log 2}$$

And by  $\sum_{k=1}^K 2^{-l_k} \leq 1$ , we have

$$\sum_{k=1}^K \frac{p_j}{\mu_0 \log 2} = \frac{1}{\mu_0 \log 2} \leq 1 \Rightarrow \mu_0 \geq \frac{1}{\log 2}$$



# Minimum rate required to compress a source

Since we expect  $l_k > 0$ ,  $\mu_k = 0$ . Expand the first equation, we get

$$-p_j + \mu_0 2^{-l_j} \log 2 = 0 \Rightarrow 2^{-l_j} = \frac{p_j}{\mu_0 \log 2}$$

And by  $\sum_{k=1}^K 2^{-l_k} \leq 1$ , we have

$$\sum_{k=1}^K \frac{p_j}{\mu_0 \log 2} = \frac{1}{\mu_0 \log 2} \leq 1 \Rightarrow \mu_0 \geq \frac{1}{\log 2}$$

Note that as  $\mu_0 \downarrow$ ,  $\frac{p_j}{\mu_0 \log 2} \uparrow$  and  $l_j \downarrow$ .

# Minimum rate required to compress a source

Since we expect  $l_k > 0$ ,  $\mu_k = 0$ . Expand the first equation, we get

$$-p_j + \mu_0 2^{-l_j} \log 2 = 0 \Rightarrow 2^{-l_j} = \frac{p_j}{\mu_0 \log 2}$$

And by  $\sum_{k=1}^K 2^{-l_k} \leq 1$ , we have

$$\sum_{k=1}^K \frac{p_k}{\mu_0 \log 2} = \frac{1}{\mu_0 \log 2} \leq 1 \Rightarrow \mu_0 \geq \frac{1}{\log 2}$$

Note that as  $\mu_0 \downarrow$ ,  $\frac{p_j}{\mu_0 \log 2} \uparrow$  and  $l_j \downarrow$ . Therefore, if we want to decrease code rate, we should reduce  $\mu_0$  as much as possible. Thus, take  $\mu_0 = \frac{1}{\log 2}$ . Then  $2^{-l_j} = p_j \Rightarrow l_j = -\log_2 p_j$ . Thus, the minimum rate becomes

$$\sum_{k=1}^K p_k l_k = -\sum_{k=1}^K p_k \log_2 p_k \triangleq H(p_1, \dots, p_K)$$

# Review

- Kraft's inequality:  $\sum_{k=1}^K 2^{-l_k} \leq 1$

# Review

- Kraft's inequality:  $\sum_{k=1}^K 2^{-l_k} \leq 1$ 
  - We showed that given a code "length-profile", we can always find a prefix-free code if the profile satisfies Kraft's inequality

# Review

- Kraft's inequality:  $\sum_{k=1}^K 2^{-l_k} \leq 1$ 
  - We showed that given a code "length-profile", we can always find a prefix-free code if the profile satisfies Kraft's inequality
  - Conversely, if Kraft's inequality is not satisfied, any code with that profile is not uniquely decodable  $\Rightarrow$  trash

# Review

- Kraft's inequality:  $\sum_{k=1}^K 2^{-l_k} \leq 1$ 
  - We showed that given a code "length-profile", we can always find a prefix-free code if the profile satisfies Kraft's inequality
  - Conversely, if Kraft's inequality is not satisfied, any code with that profile is not uniquely decodable  $\Rightarrow$  trash
- Converse proof of Source Coding Theorem: if we minimize the expected code length subject to the Kraft's inequality, the minimum "code rate" is equal to the entropy of the source.

# Review

- Kraft's inequality:  $\sum_{k=1}^K 2^{-l_k} \leq 1$ 
  - We showed that given a code "length-profile", we can always find a prefix-free code if the profile satisfies Kraft's inequality
  - Conversely, if Kraft's inequality is not satisfied, any code with that profile is not uniquely decodable  $\Rightarrow$  trash
- Converse proof of Source Coding Theorem: if we minimize the expected code length subject to the Kraft's inequality, the minimum "code rate" is equal to the entropy of the source.
  - In other words, we cannot compress a source losslessly below its entropy

# Review

- Kraft's inequality:  $\sum_{k=1}^K 2^{-l_k} \leq 1$ 
  - We showed that given a code "length-profile", we can always find a prefix-free code if the profile satisfies Kraft's inequality
  - Conversely, if Kraft's inequality is not satisfied, any code with that profile is not uniquely decodable  $\Rightarrow$  trash
- Converse proof of Source Coding Theorem: if we minimize the expected code length subject to the Kraft's inequality, the minimum "code rate" is equal to the entropy of the source.
  - In other words, we cannot compress a source losslessly below its entropy
- Forward proof of Source Coding Theorem: the obvious question now is can we compress any source arbitrary close to its entropy?



# Review

- Kraft's inequality:  $\sum_{k=1}^K 2^{-l_k} \leq 1$ 
  - We showed that given a code "length-profile", we can always find a prefix-free code if the profile satisfies Kraft's inequality
  - Conversely, if Kraft's inequality is not satisfied, any code with that profile is not uniquely decodable  $\Rightarrow$  trash
- Converse proof of Source Coding Theorem: if we minimize the expected code length subject to the Kraft's inequality, the minimum "code rate" is equal to the entropy of the source.
  - In other words, we cannot compress a source losslessly below its entropy
- Forward proof of Source Coding Theorem: the obvious question now is can we compress any source arbitrary close to its entropy?
  - Absolutely! And we will show it today

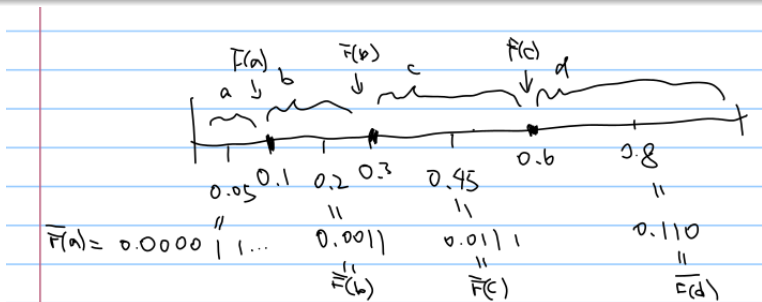
# Shannon-Fano-Elias code

## Key idea

Each codeword corresponds to an interval of  $[0, 1]$

## Example

110 corresponds to  $[0.110, 0.1101] = [0.11, 0.111) = [0.75, 0.875)$



# Shannon-Fano-Elias code

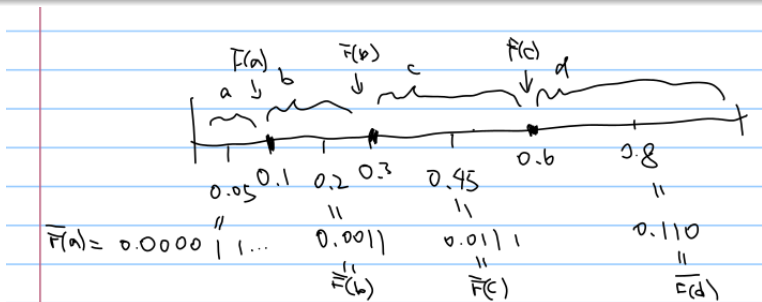
## Key idea

Each codeword corresponds to an interval of  $[0, 1]$

## Example

110 corresponds to  $[0.110, 0.1101] = [0.11, 0.111) = [0.75, 0.875)$

011 corresponds to  $[0.011, 0.0111] = [0.011, 0.1) = [0.375, 0.5)$



# Observations

## Remark (Observation 1)

*Let  $l(x) = |c(x)|$  be the length of the SFE codeword, and let  $u(x)$  be the corresponding interval. Then, the length of the interval  $|u(x)| = 2^{-l(x)}$*

# Observations

## Remark (Observation 1)

*Let  $l(x) = |c(x)|$  be the length of the SFE codeword, and let  $u(x)$  be the corresponding interval. Then, the length of the interval  $|u(x)| = 2^{-l(x)}$*

## Remark (Observation 2)

*If  $u(x_1)$  and  $u(x_2)$  do not overlap, then  $c(x_1)$  and  $c(x_2)$  cannot be prefix of one another*

# Observations

## Remark (Observation 1)

Let  $l(x) = |c(x)|$  be the length of the SFE codeword, and let  $u(x)$  be the corresponding interval. Then, the length of the interval  $|u(x)| = 2^{-l(x)}$

## Remark (Observation 2)

If  $u(x_1)$  and  $u(x_2)$  do not overlap, then  $c(x_1)$  and  $c(x_2)$  cannot be prefix of one another

## Proof of Observation 2.

WLOG, assume  $c(x_1)$  is a prefix of  $c(x_2)$ , the lower boundary of  $u(x_1)$  is below the lower boundary of  $u(x_2)$  and yet the upper boundary of  $u(x_1)$  is above the upper boundary of  $u(x_2)$ . Thus,  $u(x_2) \subseteq u(x_1)$  and hence  $u(x_1)$  and  $u(x_2)$  overlap each other □

# Example

Consider a source that

$$p(x_1) = 0.25, p(x_2) = 0.25, p(x_3) = 0.2, p(x_4) = 0.15, p(x_5) = 0.15$$

$x$	$p(x)$	$F(x)$	$\overline{F}(x)$	$\overline{F}(x)$ in Binary	$l(x) = \left\lceil \log \frac{1}{p(x)} \right\rceil + 1$	Codeword
1	0.25	0.25	0.125	0.001	3	001
2	0.25	0.5	0.375	0.011	3	011
3	0.2	0.7	0.6	0.10011	4	1001
4	0.15	0.85	0.775	0.1100011	4	1100
5	0.15	1.0	0.925	0.1110110	4	1110

# Property

- The length of the codeword of  $x$  is  $\lceil \log_2 \frac{1}{p(x)} \rceil + 1$ . This ensures that the “code interval” of each codeword does not overlap



# Property

- The length of the codeword of  $x$  is  $\lceil \log_2 \frac{1}{p(x)} \rceil + 1$ . This ensures that the “code interval” of each codeword does not overlap
- Recall from observation 1, SFE code is prefix-free  $\rightarrow$  uniquely decodable

# Property

- The length of the codeword of  $x$  is  $\lceil \log_2 \frac{1}{p(x)} \rceil + 1$ . This ensures that the “code interval” of each codeword does not overlap
- Recall from observation 1, SFE code is prefix-free  $\rightarrow$  uniquely decodable
  - If a codeword is prefix of another (say 10 and 1010), the corresponding intervals must overlap each other (consider  $[0.10, 0.11)$  and  $[0.101, 0.11)$ )

# Property

- The length of the codeword of  $x$  is  $\lceil \log_2 \frac{1}{p(x)} \rceil + 1$ . This ensures that the “code interval” of each codeword does not overlap
- Recall from observation 1, SFE code is prefix-free  $\rightarrow$  uniquely decodable
  - If a codeword is prefix of another (say 10 and 1010), the corresponding intervals must overlap each other (consider  $[0.10, 0.11)$  and  $[0.101, 0.11)$ )
  - Since no codeword can overlap in SFE, no code word can be prefix of another

# Property

- The length of the codeword of  $x$  is  $\lceil \log_2 \frac{1}{p(x)} \rceil + 1$ . This ensures that the “code interval” of each codeword does not overlap
- Recall from observation 1, SFE code is prefix-free  $\rightarrow$  uniquely decodable
  - If a codeword is prefix of another (say 10 and 1010), the corresponding intervals must overlap each other (consider  $[0.10, 0.11)$  and  $[0.101, 0.11)$ )
  - Since no codeword can overlap in SFE, no code word can be prefix of another
- Average code rate is upper bounded by  $H(X) + 2$

$$\begin{aligned} \sum_{x \in \mathcal{X}} p(x) l(x) &= \sum_{x \in \mathcal{X}} p(x) \left( \left\lceil \log_2 \frac{1}{p(x)} \right\rceil + 1 \right) \\ &\leq \sum_{x \in \mathcal{X}} p(x) \left( \log_2 \frac{1}{p(x)} + 2 \right) = H(X) + 2 \end{aligned}$$

# “Symbol grouping” trick

- Let's consider two symbols as a super-symbol and compress the pair at each time with SFE code
- The code rate is bounded by  $H(X_S) + 2$ , where

# “Symbol grouping” trick

- Let's consider two symbols as a super-symbol and compress the pair at each time with SFE code
- The code rate is bounded by  $H(X_S) + 2$ , where

$$H(X_S) = - \sum_{x_1, x_2 \in \mathcal{X}^2} p(x_1, x_2) \log_2 p(x_1, x_2)$$

# “Symbol grouping” trick

- Let's consider two symbols as a super-symbol and compress the pair at each time with SFE code
- The code rate is bounded by  $H(X_S) + 2$ , where

$$\begin{aligned} H(X_S) &= - \sum_{x_1, x_2 \in \mathcal{X}^2} p(x_1, x_2) \log_2 p(x_1, x_2) \\ &= - \sum_{x_1, x_2 \in \mathcal{X}^2} p(x_1, x_2) \log_2 (p(x_1)p(x_2)) \end{aligned}$$

# “Symbol grouping” trick

- Let's consider two symbols as a super-symbol and compress the pair at each time with SFE code
- The code rate is bounded by  $H(X_S) + 2$ , where

$$\begin{aligned}
 H(X_S) &= - \sum_{x_1, x_2 \in \mathcal{X}^2} p(x_1, x_2) \log_2 p(x_1, x_2) \\
 &= - \sum_{x_1, x_2 \in \mathcal{X}^2} p(x_1, x_2) \log_2 (p(x_1)p(x_2)) \\
 &= - \sum_{x_1, x_2 \in \mathcal{X}^2} p(x_1, x_2) \log_2 p(x_1) - \sum_{x_1, x_2 \in \mathcal{X}^2} p(x_1, x_2) \log_2 p(x_2)
 \end{aligned}$$



# “Symbol grouping” trick

- Let's consider two symbols as a super-symbol and compress the pair at each time with SFE code
- The code rate is bounded by  $H(X_S) + 2$ , where

$$\begin{aligned}
 H(X_S) &= - \sum_{x_1, x_2 \in \mathcal{X}^2} p(x_1, x_2) \log_2 p(x_1, x_2) \\
 &= - \sum_{x_1, x_2 \in \mathcal{X}^2} p(x_1, x_2) \log_2 (p(x_1)p(x_2)) \\
 &= - \sum_{x_1, x_2 \in \mathcal{X}^2} p(x_1, x_2) \log_2 p(x_1) - \sum_{x_1, x_2 \in \mathcal{X}^2} p(x_1, x_2) \log_2 p(x_2) \\
 &= - \sum_{x_1 \in \mathcal{X}} p(x_1) \log_2 p(x_1) - \sum_{x_2 \in \mathcal{X}} p(x_2) \log_2 p(x_2) \\
 &= 2H(X)
 \end{aligned}$$

Therefore, the code rate per original symbol is upper bounded by

$$\frac{1}{2} (H(X_S) + 2) = H(X) + 1$$

# Forward proof of Source Coding Theorem

In theory, we can group as many symbols as we want (we want do it in practice, why?), say we group  $N$  symbols at a time and compress it using SFE code.

# Forward proof of Source Coding Theorem

In theory, we can group as many symbols as we want (we want do it in practice, why?), say we group  $N$  symbols at a time and compress it using SFE code. The code rate per original symbol is upper bounded by

$$\frac{1}{N} (H(X_S) + 2) = \frac{1}{N} (NH(X) + 2) = H(X) + \frac{2}{N}$$

# Forward proof of Source Coding Theorem

In theory, we can group as many symbols as we want (we want do it in practice, why?), say we group  $N$  symbols at a time and compress it using SFE code. The code rate per original symbol is upper bounded by

$$\frac{1}{N} (H(X_S) + 2) = \frac{1}{N} (NH(X) + 2) = H(X) + \frac{2}{N}$$

Therefore as long as a given rate  $R > H(X)$ , we can always find a large enough  $N$  such that the code rate using the “grouping trick” and SFE code is below  $R$ . This concludes the forward proof

# Entropy for discrete random variable

## Von Neuman to Shannon

"You should call it entropy for two reasons: first because that is what the formula is in statistical mechanics but second and more important, as nobody knows what entropy is, whenever you use the term you will always be at an advantage!" -John von Neuman

$$H(X) = - \sum_{x \in \mathcal{X}} p(x) \log p(x) = E[-\log p(X)]$$

# Entropy for discrete random variable

## Von Neuman to Shannon

"You should call it entropy for two reasons: first because that is what the formula is in statistical mechanics but second and more important, as nobody knows what entropy is, whenever you use the term you will always be at an advantage!" -John von Neuman

$$H(X) = - \sum_{x \in \mathcal{X}} p(x) \log p(x) = E[-\log p(X)]$$

- From the expression, it suggests that there is  $\log \frac{1}{p(x)}$  bits of information for the outcome  $x$

# Entropy for discrete random variable

## Von Neuman to Shannon

"You should call it entropy for two reasons: first because that is what the formula is in statistical mechanics but second and more important, as nobody knows what entropy is, whenever you use the term you will always be at an advantage!" -John von Neuman

$$H(X) = - \sum_{x \in \mathcal{X}} p(x) \log p(x) = E[-\log p(X)]$$

- From the expression, it suggests that there is  $\log \frac{1}{p(x)}$  bits of information for the outcome  $x$
- This actually comes with no surprise! Consider a uniform random variable with 4 outcomes, each outcome will have probability  $1/4 = 0.25$  of happening it. And to represent each outcome, we need  $\log 4 = \log \frac{1}{0.25}$  bits

# Entropy for discrete random variable

## Von Neuman to Shannon

"You should call it entropy for two reasons: first because that is what the formula is in statistical mechanics but second and more important, as nobody knows what entropy is, whenever you use the term you will always be at an advantage!" -John von Neuman

$$H(X) = - \sum_{x \in \mathcal{X}} p(x) \log p(x) = E[-\log p(X)]$$

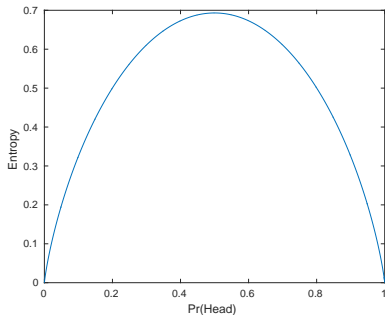
- From the expression, it suggests that there is  $\log \frac{1}{p(x)}$  bits of information for the outcome  $x$
- This actually comes with no surprise! Consider a uniform random variable with 4 outcomes, each outcome will have probability  $1/4 = 0.25$  of happening it. And to represent each outcome, we need  $\log 4 = \log \frac{1}{0.25}$  bits
- A less likely event has "more" information and requires more bits to store.  $H(X)$  is just the average number of bits required



# Biased coin with $Pr(\text{Head}) = p$

$$\begin{aligned} H(X) &= -Pr(\text{Head}) \log Pr(\text{Head}) - Pr(\text{Tail}) \log Pr(\text{Tail}) \\ &= -p \log p - (1 - p) \log(1 - p) \end{aligned}$$

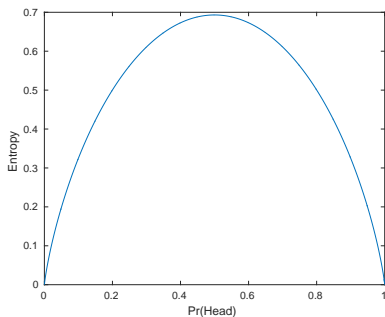
- Entropy is largest (=1) when  $p = 0.5$
- Entropy is 0 when  $p = 0$  or  $p = 1$



# Biased coin with $Pr(\text{Head}) = p$

$$\begin{aligned} H(X) &= -Pr(\text{Head}) \log Pr(\text{Head}) - Pr(\text{Tail}) \log Pr(\text{Tail}) \\ &= -p \log p - (1 - p) \log(1 - p) \end{aligned}$$

- Entropy is largest (=1) when  $p = 0.5$
- Entropy is 0 when  $p = 0$  or  $p = 1$
- Entropy can be interpreted as *the average uncertainty of the outcome or the amount of information “gained” after the outcome is revealed*



# Differential entropy

$$H(X) = - \sum_{x \in \mathcal{X}} p(x) \log p(x) = E[-\log p(X)]$$

The definition makes little sense for a continuous  $X$ . Since the probability of an outcome  $x$  is always 0, we may define instead the differential entropy for  $X$  as

$$h(X) = - \int_{x \in \mathcal{X}} p(x) \log p(x) dx$$

where  $p(x)$  is now the pdf rather than the pmf

# Differential entropy

$$H(X) = - \sum_{x \in \mathcal{X}} p(x) \log p(x) = E[-\log p(X)]$$

The definition makes little sense for a continuous  $X$ . Since the probability of an outcome  $x$  is always 0, we may define instead the differential entropy for  $X$  as

$$h(X) = - \int_{x \in \mathcal{X}} p(x) \log p(x) dx = E[-\log p(x)],$$

where  $p(x)$  is now the pdf rather than the pmf

# Differential entropy of common distributions

## Uniform Distribution

$$\text{If } p(X) = \begin{cases} 1/a & 0 \leq x \leq a \\ 0 & \text{otherwise} \end{cases}$$

$$h(X) = - \int_{x=0}^a \frac{1}{a} \log \frac{1}{a} dx = \log a$$

# Differential entropy of common distributions

## Uniform Distribution

$$\text{If } p(X) = \begin{cases} 1/a & 0 \leq x \leq a \\ 0 & \text{otherwise} \end{cases}$$

$$h(X) = - \int_{x=0}^a \frac{1}{a} \log \frac{1}{a} dx = \log a$$

## Exponential distribution

For exponentially distributed  $T \sim \text{Exp}(\lambda)$ ,

$$h(T) = E[-\log p(T)] = E[-\log(\lambda \exp(-\lambda T))]$$

# Differential entropy of common distributions

## Uniform Distribution

$$\text{If } p(X) = \begin{cases} 1/a & 0 \leq x \leq a \\ 0 & \text{otherwise} \end{cases}$$

$$h(X) = - \int_{x=0}^a \frac{1}{a} \log \frac{1}{a} dx = \log a$$

## Exponential distribution

For exponentially distributed  $T \sim \text{Exp}(\lambda)$ ,

$$\begin{aligned} h(T) &= E[-\log p(T)] = E[-\log(\lambda \exp(-\lambda T))] \\ &= E[\lambda T - \log \lambda] \end{aligned}$$

# Differential entropy of common distributions

## Uniform Distribution

$$\text{If } p(X) = \begin{cases} 1/a & 0 \leq x \leq a \\ 0 & \text{otherwise} \end{cases}$$

$$h(X) = - \int_{x=0}^a \frac{1}{a} \log \frac{1}{a} dx = \log a$$

## Exponential distribution

For exponentially distributed  $T \sim \text{Exp}(\lambda)$ ,

$$\begin{aligned} h(T) &= E[-\log p(T)] = E[-\log(\lambda \exp(-\lambda T))] \\ &= E[\lambda T - \log \lambda] \\ &= 1 - \log \lambda \end{aligned}$$



# Differential entropy of common distributions

## Univariate Normal distribution

For univariate normally distributed  $X \sim \mathcal{N}(\mu, \sigma^2)$ ,

$$h(X) = E[-\log p(X)]$$

# Differential entropy of common distributions

## Univariate Normal distribution

For univariate normally distributed  $X \sim \mathcal{N}(\mu, \sigma^2)$ ,

$$h(X) = E[-\log p(X)] = E \left[ -\log \left( \frac{1}{\sqrt{2\pi\sigma^2}} \exp \frac{-(X - \mu)^2}{2\sigma^2} \right) \right]$$

# Differential entropy of common distributions

## Univariate Normal distribution

For univariate normally distributed  $X \sim \mathcal{N}(\mu, \sigma^2)$ ,

$$\begin{aligned} h(X) &= E[-\log p(X)] = E\left[-\log\left(\frac{1}{\sqrt{2\pi\sigma^2}} \exp\frac{-(X-\mu)^2}{2\sigma^2}\right)\right] \\ &= E\left[\log\sqrt{2\pi\sigma^2} + \frac{(X-\mu)^2}{2\sigma^2} \log e\right] \end{aligned}$$

# Differential entropy of common distributions

## Univariate Normal distribution

For univariate normally distributed  $X \sim \mathcal{N}(\mu, \sigma^2)$ ,

$$\begin{aligned}h(X) &= E[-\log p(X)] = E\left[-\log\left(\frac{1}{\sqrt{2\pi\sigma^2}} \exp\frac{-(X-\mu)^2}{2\sigma^2}\right)\right] \\&= E\left[\log\sqrt{2\pi\sigma^2} + \frac{(X-\mu)^2}{2\sigma^2} \log e\right] \\&= \log\sqrt{2\pi\sigma^2} + \frac{1}{2} \log e \\&= \log\sqrt{2\pi e\sigma^2}\end{aligned}$$

N.B.  $h(X)$  only depends on  $\sigma^2$  and is independent of  $\mu$  as one would expect

# Multivariate Normal distribution

For  $N$ -dim multivariate normal distributed  $\mathbf{X} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ ,

$$\begin{aligned} h(\mathbf{X}) &= E[-\log p(\mathbf{X})] \\ &= -E \left[ \log \left( \frac{1}{\sqrt{\det(2\pi\boldsymbol{\Sigma})}} \exp \left( -\frac{1}{2}(\mathbf{X} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\mathbf{X} - \boldsymbol{\mu}) \right) \right) \right] \end{aligned}$$

# Multivariate Normal distribution

For  $N$ -dim multivariate normal distributed  $\mathbf{X} \sim \mathcal{N}(\boldsymbol{\mu}, \Sigma)$ ,

$$\begin{aligned}
 h(\mathbf{X}) &= E[-\log p(\mathbf{X})] \\
 &= -E \left[ \log \left( \frac{1}{\sqrt{\det(2\pi\Sigma)}} \exp \left( -\frac{1}{2}(\mathbf{X} - \boldsymbol{\mu})^T \Sigma^{-1}(\mathbf{X} - \boldsymbol{\mu}) \right) \right) \right] \\
 &= \log \sqrt{\det(2\pi\Sigma)} + \frac{\log e}{2} E \left[ \sum_{i,j} (X_i - \mu_i) [\Sigma^{-1}]_{i,j} (X_j - \mu_j) \right]
 \end{aligned}$$

# Multivariate Normal distribution

For  $N$ -dim multivariate normal distributed  $\mathbf{X} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ ,

$$\begin{aligned}
 h(\mathbf{X}) &= E[-\log p(\mathbf{X})] \\
 &= -E \left[ \log \left( \frac{1}{\sqrt{\det(2\pi\boldsymbol{\Sigma})}} \exp \left( -\frac{1}{2}(\mathbf{X} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\mathbf{X} - \boldsymbol{\mu}) \right) \right) \right] \\
 &= \log \sqrt{\det(2\pi\boldsymbol{\Sigma})} + \frac{\log e}{2} E \left[ \sum_{i,j} (X_i - \mu_i) [\boldsymbol{\Sigma}^{-1}]_{i,j} (X_j - \mu_j) \right] \\
 &= \log \sqrt{\det(2\pi\boldsymbol{\Sigma})} + \frac{\log e}{2} \sum_{i,j} [\boldsymbol{\Sigma}^{-1}]_{i,j} E [(X_j - \mu_j)(X_i - \mu_i)]
 \end{aligned}$$

# Multivariate Normal distribution

For  $N$ -dim multivariate normal distributed  $\mathbf{X} \sim \mathcal{N}(\boldsymbol{\mu}, \Sigma)$ ,

$$\begin{aligned}
 h(\mathbf{X}) &= E[-\log p(\mathbf{X})] \\
 &= -E \left[ \log \left( \frac{1}{\sqrt{\det(2\pi\Sigma)}} \exp \left( -\frac{1}{2}(\mathbf{X} - \boldsymbol{\mu})^T \Sigma^{-1}(\mathbf{X} - \boldsymbol{\mu}) \right) \right) \right] \\
 &= \log \sqrt{\det(2\pi\Sigma)} + \frac{\log e}{2} E \left[ \sum_{i,j} (X_i - \mu_i) [\Sigma^{-1}]_{i,j} (X_j - \mu_j) \right] \\
 &= \log \sqrt{\det(2\pi\Sigma)} + \frac{\log e}{2} \sum_{i,j} [\Sigma^{-1}]_{i,j} E [(X_j - \mu_j)(X_i - \mu_i)] \\
 &= \log \sqrt{\det(2\pi\Sigma)} + \frac{\log e}{2} \sum_{i,j} [\Sigma^{-1}]_{i,j} \Sigma_{j,i}
 \end{aligned}$$



# Multivariate Normal distribution

For  $N$ -dim multivariate normal distributed  $\mathbf{X} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ ,

$$\begin{aligned}
 h(\mathbf{X}) &= E[-\log p(\mathbf{X})] \\
 &= -E \left[ \log \left( \frac{1}{\sqrt{\det(2\pi\boldsymbol{\Sigma})}} \exp \left( -\frac{1}{2}(\mathbf{X} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\mathbf{X} - \boldsymbol{\mu}) \right) \right) \right] \\
 &= \log \sqrt{\det(2\pi\boldsymbol{\Sigma})} + \frac{\log e}{2} E \left[ \sum_{i,j} (X_i - \mu_i) [\boldsymbol{\Sigma}^{-1}]_{i,j} (X_j - \mu_j) \right] \\
 &= \log \sqrt{\det(2\pi\boldsymbol{\Sigma})} + \frac{\log e}{2} \sum_{i,j} [\boldsymbol{\Sigma}^{-1}]_{i,j} E [(X_j - \mu_j)(X_i - \mu_i)] \\
 &= \log \sqrt{\det(2\pi\boldsymbol{\Sigma})} + \frac{\log e}{2} \sum_{i,j} [\boldsymbol{\Sigma}^{-1}]_{i,j} \boldsymbol{\Sigma}_{j,i} \\
 &= \log \sqrt{\det(2\pi\boldsymbol{\Sigma})} + \frac{N \log e}{2} = \log \sqrt{e^N \det(2\pi\boldsymbol{\Sigma})}
 \end{aligned}$$

# Multivariate Normal distribution

For  $N$ -dim multivariate normal distributed  $\mathbf{X} \sim \mathcal{N}(\boldsymbol{\mu}, \Sigma)$ ,

$$\begin{aligned}
 h(\mathbf{X}) &= E[-\log p(\mathbf{X})] \\
 &= -E \left[ \log \left( \frac{1}{\sqrt{\det(2\pi\Sigma)}} \exp \left( -\frac{1}{2}(\mathbf{X} - \boldsymbol{\mu})^T \Sigma^{-1}(\mathbf{X} - \boldsymbol{\mu}) \right) \right) \right] \\
 &= \log \sqrt{\det(2\pi\Sigma)} + \frac{\log e}{2} E \left[ \sum_{i,j} (X_i - \mu_i) [\Sigma^{-1}]_{i,j} (X_j - \mu_j) \right] \\
 &= \log \sqrt{\det(2\pi\Sigma)} + \frac{\log e}{2} \sum_{i,j} [\Sigma^{-1}]_{i,j} E [(X_j - \mu_j)(X_i - \mu_i)] \\
 &= \log \sqrt{\det(2\pi\Sigma)} + \frac{\log e}{2} \sum_{i,j} [\Sigma^{-1}]_{i,j} \Sigma_{j,i} \\
 &= \log \sqrt{\det(2\pi\Sigma)} + \frac{N \log e}{2} = \log \sqrt{e^N \det(2\pi\Sigma)} = \log \sqrt{\det(2\pi e\Sigma)}
 \end{aligned}$$

# Differential entropy and entropy

How differential entropy is related to its discrete counterpart?

- Consider a continuous random variable  $X$
- Let  $X^\Delta$  is a “quantized” version of it with quantization stepsize of  $\Delta$

$$H(X^\Delta) = \sum -p_{X^\Delta}(x^\Delta) \log p_{X^\Delta}(x^\Delta)$$

# Differential entropy and entropy

How differential entropy is related to its discrete counterpart?

- Consider a continuous random variable  $X$
- Let  $X^\Delta$  is a “quantized” version of it with quantization stepsize of  $\Delta$

$$H(X^\Delta) = \sum -p_{X^\Delta}(x^\Delta) \log p_{X^\Delta}(x^\Delta) \approx \sum -p_X(x^\Delta) \Delta \log(p_X(x^\Delta) \Delta)$$

# Differential entropy and entropy

How differential entropy is related to its discrete counterpart?

- Consider a continuous random variable  $X$
- Let  $X^\Delta$  is a “quantized” version of it with quantization stepsize of  $\Delta$

$$\begin{aligned} H(X^\Delta) &= \sum -p_{X^\Delta}(x^\Delta) \log p_{X^\Delta}(x^\Delta) \approx \sum -p_X(x^\Delta) \Delta \log(p_X(x^\Delta) \Delta) \\ &\approx \int -p_X(x) \log(p_X(x) \Delta) dx \end{aligned}$$

# Differential entropy and entropy

How differential entropy is related to its discrete counterpart?

- Consider a continuous random variable  $X$
- Let  $X^\Delta$  is a “quantized” version of it with quantization stepsize of  $\Delta$

$$\begin{aligned} H(X^\Delta) &= \sum -p_{X^\Delta}(x^\Delta) \log p_{X^\Delta}(x^\Delta) \approx \sum -p_X(x^\Delta) \Delta \log(p_X(x^\Delta) \Delta) \\ &\approx \int -p_X(x) \log(p_X(x) \Delta) dx \\ &= \int -p_X(x) \log p_X(x) - \int p_X(x) \log \Delta dx \\ &= h(X) - \log \Delta \end{aligned}$$

# Example

Consider the processing time of a packet follow an exponential distribution with an average of 1 ms. If we want to store the time with the precision of 0.01 ms, about how many bits are needed to store the result?

# Example

Consider the processing time of a packet follow an exponential distribution with an average of 1 ms. If we want to store the time with the precision of 0.01 ms, about how many bits are needed to store the result?

## Answer

- The processing time  $T$  follows an exponential distribution with parameter  $\lambda = 1/1 = 1ms^{-1}$



# Example

Consider the processing time of a packet follow an exponential distribution with an average of 1 ms. If we want to store the time with the precision of 0.01 ms, about how many bits are needed to store the result?

## Answer

- The processing time  $T$  follows an exponential distribution with parameter  $\lambda = 1/1 = 1\text{ms}^{-1}$
- The corresponding differential entropy  $h(T) = 1 - \log(\lambda) = 1$

# Example

Consider the processing time of a packet follow an exponential distribution with an average of 1 ms. If we want to store the time with the precision of 0.01 ms, about how many bits are needed to store the result?

## Answer

- The processing time  $T$  follows an exponential distribution with parameter  $\lambda = 1/1 = 1ms^{-1}$
- The corresponding differential entropy  $h(T) = 1 - \log(\lambda) = 1$
- If we want to store with precision of 0.01 ms, we need  $h(T) - \log 0.01 \approx 7.64bits$

# Lower bound of entropy

$$H(X) \geq 0$$

Since  $p(X) \leq 1$ ,  $-\log p(X) \geq 0$ , therefore

$$H(X) = E[-\log p(X)] \geq 0$$

After all,  $H(X)$  represents the required bits to compress the source  $X$

# Lower bound of entropy

$$H(X) \geq 0$$

Since  $p(X) \leq 1$ ,  $-\log p(X) \geq 0$ , therefore

$$H(X) = E[-\log p(X)] \geq 0$$

After all,  $H(X)$  represents the required bits to compress the source  $X$

## Caveat

It does NOT need to be true for differential entropy. It is possible that

$$h(X) < 0$$

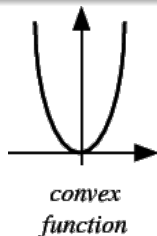
For example, for a uniformly distributed  $X$  from 0 to 0.5,

$$h(X) = \log 0.5 = -1$$

# Jensen's Inequality

For a convex (bowl-shape) function  $f$

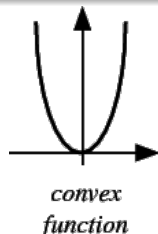
$$E[f(X)] \geq f(E[X])$$



# Jensen's Inequality

For a convex (bowl-shape) function  $f$

$$E[f(X)] \geq f(E[X])$$



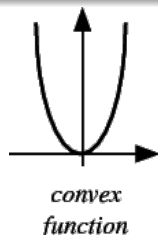
Let us consider  $X$  with only two outcomes  $x_1$  and  $x_2$  with probabilities  $p$  and  $1 - p$ . Easy to see that

$$E[f(X)] = pf(x_1) + (1 - p)f(x_2) \geq f(px_1 + (1 - p)x_2) = f(E[X])$$

# Jensen's Inequality

For a convex (bowl-shape) function  $f$

$$E[f(X)] \geq f(E[X])$$



Let us consider  $X$  with only two outcomes  $x_1$  and  $x_2$  with probabilities  $p$  and  $1 - p$ . Easy to see that

$$E[f(X)] = pf(x_1) + (1 - p)f(x_2) \geq f(px_1 + (1 - p)x_2) = f(E[X])$$

Result can be extended to variables with more than two outcomes easily

# Upper bound of entropy

$$H(X) \leq \log |\mathcal{X}|$$

$$H(X) = E[-\log p(X)] = E \left[ \log \frac{1}{p(X)} \right]$$



# Upper bound of entropy

$$H(X) \leq \log |\mathcal{X}|$$

$$\begin{aligned} H(X) &= E[-\log p(X)] = E\left[\log \frac{1}{p(X)}\right] \\ &\leq \log E\left[\frac{1}{p(X)}\right] \quad (\text{by Jensen's inequality}) \end{aligned}$$

# Upper bound of entropy

$$H(X) \leq \log |\mathcal{X}|$$

$$\begin{aligned} H(X) &= E[-\log p(X)] = E\left[\log \frac{1}{p(X)}\right] \\ &\leq \log E\left[\frac{1}{p(X)}\right] \quad (\text{by Jensen's inequality}) \\ &= \log \sum_{x \in \mathcal{X}} p(x) \frac{1}{p(x)} = \log |\mathcal{X}| \end{aligned}$$

N.B. The upper bound is attained when the distribution is uniform

# Upper bound of entropy

$$H(X) \leq \log |\mathcal{X}|$$

$$\begin{aligned} H(X) &= E[-\log p(X)] = E\left[\log \frac{1}{p(X)}\right] \\ &\leq \log E\left[\frac{1}{p(X)}\right] \quad (\text{by Jensen's inequality}) \\ &= \log \sum_{x \in \mathcal{X}} p(x) \frac{1}{p(x)} = \log |\mathcal{X}| \end{aligned}$$

N.B. The upper bound is attained when the distribution is uniform

## Examples

You should know this bound long alone. Think of the maximum number of bits needed:

- to store the outcome of flipping a coin:  $\log 2 = 1$  bit
- to store the outcome of throwing a dice:  $\log 6 \leq 3$  bits

# Review

- Source coding theorem: For an independent and identically distributed (i.i.d.) discrete memoryless source (DMS)  $X$ , we can always compress it with no less than  $H(X)$  bits per input symbol, where  $H(X) = -\sum_{x \in \mathcal{X}} p(x) \log p(x) = E[-\log p(X)]$

# Review

- Source coding theorem: For an independent and identically distributed (i.i.d.) discrete memoryless source (DMS)  $X$ , we can always compress it with no less than  $H(X)$  bits per input symbol, where  $H(X) = -\sum_{x \in \mathcal{X}} p(x) \log p(x) = E[-\log p(X)]$
- Jensen's inequality: For a convex (bowl-shape) function  $f$   $E[f(X)] \geq f(E[X])$ . Similarly  $E[g(X)] \leq g(E[X])$  for a concave  $g$

# Review

- Source coding theorem: For an independent and identically distributed (i.i.d.) discrete memoryless source (DMS)  $X$ , we can always compress it with no less than  $H(X)$  bits per input symbol, where  $H(X) = -\sum_{x \in \mathcal{X}} p(x) \log p(x) = E[-\log p(X)]$
- Jensen's inequality: For a convex (bowl-shape) function  $f$   $E[f(X)] \geq f(E[X])$ . Similarly  $E[g(X)] \leq g(E[X])$  for a concave  $g$
- For continuous random variable  $X$ , the differential entropy is given by  $h(X) = -\int_{x \in \mathcal{X}} p(x) \log p(x) dx = E[-\log p(x)]$

## Review

- Source coding theorem: For an independent and identically distributed (i.i.d.) discrete memoryless source (DMS)  $X$ , we can always compress it with no less than  $H(X)$  bits per input symbol, where  $H(X) = -\sum_{x \in \mathcal{X}} p(x) \log p(x) = E[-\log p(X)]$
- Jensen's inequality: For a convex (bowl-shape) function  $f$   $E[f(X)] \geq f(E[X])$ . Similarly  $E[g(X)] \leq g(E[X])$  for a concave  $g$
- For continuous random variable  $X$ , the differential entropy is given by  $h(X) = -\int_{x \in \mathcal{X}} p(x) \log p(x) dx = E[-\log p(x)]$
- For a quantized version of continuous  $X$ ,  $H(X_\Delta) = h(X) - \log \Delta$

## Review

- Source coding theorem: For an independent and identically distributed (i.i.d.) discrete memoryless source (DMS)  $X$ , we can always compress it with no less than  $H(X)$  bits per input symbol, where  $H(X) = -\sum_{x \in \mathcal{X}} p(x) \log p(x) = E[-\log p(X)]$
- Jensen's inequality: For a convex (bowl-shape) function  $f$   $E[f(X)] \geq f(E[X])$ . Similarly  $E[g(X)] \leq g(E[X])$  for a concave  $g$
- For continuous random variable  $X$ , the differential entropy is given by  $h(X) = -\int_{x \in \mathcal{X}} p(x) \log p(x) dx = E[-\log p(x)]$
- For a quantized version of continuous  $X$ ,  $H(X_\Delta) = h(X) - \log \Delta$
- For multivariate normal  $\mathbf{X} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ ,

$$h(\mathbf{X}) = \log \sqrt{\det(2\pi e \boldsymbol{\Sigma})}$$



# Upper bound of differential entropy

$$h(X) \leq \log E \left[ \frac{1}{p(X)} \right] = \log \int_{x \in \mathcal{X}} p(x) \frac{1}{p(x)} dx = \log |\mathcal{X}|$$

- The expression still makes sense but it is not useful usually since the sampling space can be unbounded  $|\mathcal{X}| = \infty$  (for example, normally distributed  $X$ )

# Upper bound of differential entropy

$$h(X) \leq \log E \left[ \frac{1}{p(X)} \right] = \log \int_{x \in \mathcal{X}} p(x) \frac{1}{p(x)} dx = \log |\mathcal{X}|$$

- The expression still makes sense but it is not useful usually since the sampling space can be unbounded  $|\mathcal{X}| = \infty$  (for example, normally distributed  $X$ )
- Thus it makes much more sense to consider upper bound of a differential entropy constrained on the variance of the variable (**why not constrained on mean?**)

# Upper bound of differential entropy

$$h(X) \leq \log E \left[ \frac{1}{p(X)} \right] = \log \int_{x \in \mathcal{X}} p(x) \frac{1}{p(x)} dx = \log |\mathcal{X}|$$

- The expression still makes sense but it is not useful usually since the sampling space can be unbounded  $|\mathcal{X}| = \infty$  (for example, normally distributed  $X$ )
- Thus it makes much more sense to consider upper bound of a differential entropy constrained on the variance of the variable (**why not constrained on mean?**)
- It turns out that for a fixed variance  $\sigma^2$ , the variable will have largest differential entropy if it is normally distributed (will show later). Thus

$$h(X) \leq \log \sqrt{2\pi e \sigma^2}$$

# Joint entropy

For multivariate random variable, we can extend the definition of entropy naturally as follows:

## Entropy

$$H(X, Y) = E[-\log p(X, Y)]$$

and

$$H(X_1, X_2, \dots, X_N) = E[-\log p(X_1, \dots, X_N)]$$

# Joint entropy

For multivariate random variable, we can extend the definition of entropy naturally as follows:

## Entropy

$$H(X, Y) = E[-\log p(X, Y)]$$

and

$$H(X_1, X_2, \dots, X_N) = E[-\log p(X_1, \dots, X_N)]$$

## Differential entropy

$$h(X, Y) = E[-\log p(X, Y)]$$

and

$$h(X_1, X_2, \dots, X_N) = E[-\log p(X_1, \dots, X_N)]$$

# Conditional entropy

$$\begin{aligned} H(X, Y) &= E[-\log p(X, Y)] = E[-\log p(X) - \log p(Y|X)] \\ &= H(X) + \underbrace{E[-\log p(Y|X)]}_{H(Y|X)} \end{aligned}$$

## Entropy

$$H(Y|X) \triangleq H(X, Y) - H(X)$$

# Conditional entropy

$$\begin{aligned} H(X, Y) &= E[-\log p(X, Y)] = E[-\log p(X) - \log p(Y|X)] \\ &= H(X) + \underbrace{E[-\log p(Y|X)]}_{H(Y|X)} \end{aligned}$$

## Entropy

$$H(Y|X) \triangleq H(X, Y) - H(X)$$

## Differential entropy

$$h(Y|X) \triangleq h(X, Y) - h(X)$$

# Conditional entropy

$$\begin{aligned} H(X, Y) &= E[-\log p(X, Y)] = E[-\log p(X) - \log p(Y|X)] \\ &= H(X) + \underbrace{E[-\log p(Y|X)]}_{H(Y|X)} \end{aligned}$$

## Entropy

$$H(Y|X) \triangleq H(X, Y) - H(X)$$

## Differential entropy

$$h(Y|X) \triangleq h(X, Y) - h(X)$$

## Interpretation

Total Info. of  $X$  and  $Y$  = Info. of  $X$  + Info. of  $Y$  knowing  $X$



# Expanding conditional entropy

$$H(Y|X) = E[-\log p(Y|X)]$$

# Expanding conditional entropy

$$\begin{aligned} H(Y|X) &= E[-\log p(Y|X)] \\ &= \sum_{x,y} -p(x,y) \log p(y|x) \end{aligned}$$

# Expanding conditional entropy

$$\begin{aligned} H(Y|X) &= E[-\log p(Y|X)] \\ &= \sum_{x,y} -p(x,y) \log p(y|x) \\ &= \sum_x p(x) \sum_y -p(y|x) \log p(y|x) \end{aligned}$$

# Expanding conditional entropy

$$\begin{aligned} H(Y|X) &= E[-\log p(Y|X)] \\ &= \sum_{x,y} -p(x,y) \log p(y|x) \\ &= \sum_x p(x) \sum_y -p(y|x) \log p(y|x) \\ &= \sum_x p(x) H(Y|x) \end{aligned}$$

# Expanding conditional entropy

$$\begin{aligned} H(Y|X) &= E[-\log p(Y|X)] \\ &= \sum_{x,y} -p(x,y) \log p(y|x) \\ &= \sum_x p(x) \sum_y -p(y|x) \log p(y|x) \\ &= \sum_x p(x) H(Y|x) \end{aligned}$$

The conditional entropy  $H(Y|X)$  is essentially the average of  $H(Y|x)$  over all possible value of  $x$

# Chain rule

## Entropy

$$H(X_1, X_2, \dots, X_N) = H(X_1) + H(X_2|X_1) + H(X_3|X_1, X_2) + \dots \\ + H(X_N|X_1, X_2, \dots, X_{N-1}).$$

# Chain rule

## Entropy

$$H(X_1, X_2, \dots, X_N) = H(X_1) + H(X_2|X_1) + H(X_3|X_1, X_2) + \dots \\ + H(X_N|X_1, X_2, \dots, X_{N-1}).$$

## Differential entropy

$$h(X_1, X_2, \dots, X_N) = h(X_1) + h(X_2|X_1) + h(X_3|X_1, X_2) + \dots \\ + h(X_N|X_1, X_2, \dots, X_{N-1}).$$

# Example

$$\Pr(\text{Rain, With umbrella}) = 0.2$$

$$\Pr(\text{Rain, No umbrella}) = 0.1$$

$$\Pr(\text{Sunny, With umbrella}) = 0.2$$

$$\Pr(\text{Sunny, No umbrella}) = 0.5$$

$$W \in \{\text{Rain, Sunny}\}$$

$$U \in \{\text{With umbrella, No umbrella}\}$$

## Entropies

$$H(W, U) = -0.2 \log 0.2 - 0.1 \log 0.1 - 0.2 \log 0.2 - 0.5 \log 0.5 = 1.76 \text{ bits}$$

$$H(W) = -0.3 \log 0.3 - 0.7 \log 0.7 = 0.88 \text{ bits}$$

$$H(U) = -0.4 \log 0.4 - 0.6 \log 0.6 = 0.97 \text{ bits}$$

$$H(W|U) = H(W, U) - H(U) = 0.79 \text{ bits}$$

$$H(U|W) = H(W, U) - H(W) = 0.88 \text{ bits}$$



# Definition

It is often useful to gauge the difference between two distributions. KL-divergence is also known to be relative entropy. It is a way to measure the difference between two distributions. For two distributions of  $X$ ,  $p(x)$  and  $q(x)$ ,

$$KL(p(x)||q(x)) \triangleq \sum_{x \in \mathcal{X}} p(x) \log_2 \frac{p(x)}{q(x)}.$$

# Definition

It is often useful to gauge the difference between two distributions. KL-divergence is also known to be relative entropy. It is a way to measure the difference between two distributions. For two distributions of  $X$ ,  $p(x)$  and  $q(x)$ ,

$$KL(p(x)||q(x)) \triangleq \sum_{x \in \mathcal{X}} p(x) \log_2 \frac{p(x)}{q(x)}.$$

- N.B. If  $p(x) = q(x)$  for all  $x$ ,  $KL(p(x)||q(x)) = 0$  as desired

# Definition

It is often useful to gauge the difference between two distributions. KL-divergence is also known to be relative entropy. It is a way to measure the difference between two distributions. For two distributions of  $X$ ,  $p(x)$  and  $q(x)$ ,

$$KL(p(x)||q(x)) \triangleq \sum_{x \in \mathcal{X}} p(x) \log_2 \frac{p(x)}{q(x)}.$$

- N.B. If  $p(x) = q(x)$  for all  $x$ ,  $KL(p(x)||q(x)) = 0$  as desired
- N.B.  $KL(p(x)||q(x)) \neq KL(q(x)||p(x))$  in general

# KL-divergence is non-negative

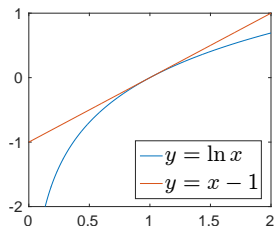
$$\begin{aligned} KL(p(x)||q(x)) &= \sum_{x \in \mathcal{X}} p(x) \log_2 \frac{p(x)}{q(x)} \\ &= - \sum_{x \in \mathcal{X}} p(x) \log_2 \frac{q(x)}{p(x)} \end{aligned}$$

# KL-divergence is non-negative

$$\begin{aligned} KL(p(x)||q(x)) &= \sum_{x \in \mathcal{X}} p(x) \log_2 \frac{p(x)}{q(x)} \\ &= - \sum_{x \in \mathcal{X}} p(x) \log_2 \frac{q(x)}{p(x)} \\ &= - \sum_{x \in \mathcal{X}} \frac{p(x)}{\ln 2} \ln \frac{q(x)}{p(x)} \end{aligned}$$

# KL-divergence is non-negative

$$\begin{aligned}
 KL(p(x)||q(x)) &= \sum_{x \in \mathcal{X}} p(x) \log_2 \frac{p(x)}{q(x)} \\
 &= - \sum_{x \in \mathcal{X}} p(x) \log_2 \frac{q(x)}{p(x)} \\
 &= - \sum_{x \in \mathcal{X}} \frac{p(x)}{\ln 2} \ln \frac{q(x)}{p(x)}
 \end{aligned}$$

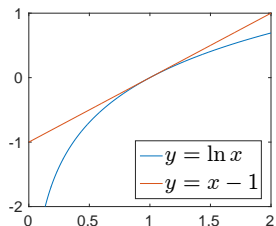


## Fact

For any real  $x$ ,  $\ln(x) \leq x - 1$ . Moreover, the equality only holds when  $x = 1$ .

# KL-divergence is non-negative

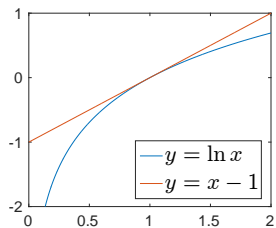
$$\begin{aligned}
 KL(p(x)||q(x)) &= \sum_{x \in \mathcal{X}} p(x) \log_2 \frac{p(x)}{q(x)} \\
 &= - \sum_{x \in \mathcal{X}} p(x) \log_2 \frac{q(x)}{p(x)} \\
 &= - \sum_{x \in \mathcal{X}} \frac{p(x)}{\ln 2} \ln \frac{q(x)}{p(x)} \\
 &\geq - \sum_{x \in \mathcal{X}} \frac{p(x)}{\ln 2} \left( \frac{q(x)}{p(x)} - 1 \right)
 \end{aligned}$$



## Fact

For any real  $x$ ,  $\ln(x) \leq x - 1$ . Moreover, the equality only holds when  $x = 1$ .

# KL-divergence is non-negative



$$\begin{aligned}
 KL(p(x)||q(x)) &= \sum_{x \in \mathcal{X}} p(x) \log_2 \frac{p(x)}{q(x)} \\
 &= - \sum_{x \in \mathcal{X}} p(x) \log_2 \frac{q(x)}{p(x)} \\
 &= - \sum_{x \in \mathcal{X}} \frac{p(x)}{\ln 2} \ln \frac{q(x)}{p(x)} \\
 &\geq - \sum_{x \in \mathcal{X}} \frac{p(x)}{\ln 2} \left( \frac{q(x)}{p(x)} - 1 \right) \\
 &= \frac{1}{\ln 2} \left( \sum_{x \in \mathcal{X}} p(x) - \sum_{x \in \mathcal{X}} q(x) \right) = 0
 \end{aligned}$$

## Fact

For any real  $x$ ,  $\ln(x) \leq x - 1$ . Moreover, the equality only holds when  $x = 1$



# Continuous variables

We can define KL-divergence for continuous variables in a similar manner

$$\begin{aligned} KL(p(x)||q(x)) &\triangleq \int_{x \in \mathcal{X}} p(x) \log_2 \frac{p(x)}{q(x)} dx \\ &= - \int_{x \in \mathcal{X}} p(x) \log_2 \frac{q(x)}{p(x)} dx \\ &= - \int_{x \in \mathcal{X}} \frac{p(x)}{\ln 2} \ln \frac{q(x)}{p(x)} dx \end{aligned}$$

# Continuous variables

We can define KL-divergence for continuous variables in a similar manner

$$\begin{aligned} KL(p(x)||q(x)) &\triangleq \int_{x \in \mathcal{X}} p(x) \log_2 \frac{p(x)}{q(x)} dx \\ &= - \int_{x \in \mathcal{X}} p(x) \log_2 \frac{q(x)}{p(x)} dx \\ &= - \int_{x \in \mathcal{X}} \frac{p(x)}{\ln 2} \ln \frac{q(x)}{p(x)} dx \\ &\geq - \int_{x \in \mathcal{X}} \frac{p(x)}{\ln 2} \left( \frac{q(x)}{p(x)} - 1 \right) dx \end{aligned}$$

# Continuous variables

We can define KL-divergence for continuous variables in a similar manner

$$\begin{aligned} KL(p(x)||q(x)) &\triangleq \int_{x \in \mathcal{X}} p(x) \log_2 \frac{p(x)}{q(x)} dx \\ &= - \int_{x \in \mathcal{X}} p(x) \log_2 \frac{q(x)}{p(x)} dx \\ &= - \int_{x \in \mathcal{X}} \frac{p(x)}{\ln 2} \ln \frac{q(x)}{p(x)} dx \\ &\geq - \int_{x \in \mathcal{X}} \frac{p(x)}{\ln 2} \left( \frac{q(x)}{p(x)} - 1 \right) dx \\ &= - \frac{1}{\ln 2} \left( \int_{x \in \mathcal{X}} q(x) dx - \int_{x \in \mathcal{X}} p(x) dx \right) = 0 \end{aligned}$$

# Normal distribution has highest entropy

For fixed variance (covariance matrix), normal distribution has highest entropy

# Normal distribution has highest entropy

For fixed variance (covariance matrix), normal distribution has highest entropy

## Proof

Let's consider the multivariate case with a fixed covariance matrix  $\Sigma$ , the univariate (scalar) case is a special case thus automatically taken care of.

# Normal distribution has highest entropy

For fixed variance (covariance matrix), normal distribution has highest entropy

## Proof

Let's consider the multivariate case with a fixed covariance matrix  $\Sigma$ , the univariate (scalar) case is a special case thus automatically taken care of. Without loss of generality, let's consider zero mean. Denote  $\mathcal{N}(\mathbf{x}; \mathbf{0}, \Sigma) = \phi(\mathbf{x})$ .

# Normal distribution has highest entropy

For fixed variance (covariance matrix), normal distribution has highest entropy

## Proof

Let's consider the multivariate case with a fixed covariance matrix  $\Sigma$ , the univariate (scalar) case is a special case thus automatically taken care of.

Without loss of generality, let's consider zero mean. Denote

$\mathcal{N}(\mathbf{x}; \mathbf{0}, \Sigma) = \phi(\mathbf{x})$ . For any other distribution  $f(\mathbf{x})$  with the same covariance matrix  $\Sigma$ , first note that  $\int_{\mathbf{x}} f(\mathbf{x}) \log \phi(\mathbf{x}) d\mathbf{x} = \int_{\mathbf{x}} \phi(\mathbf{x}) \log \phi(\mathbf{x}) d\mathbf{x}$  (to be show in the next slide).

# Normal distribution has highest entropy

For fixed variance (covariance matrix), normal distribution has highest entropy

## Proof

Let's consider the multivariate case with a fixed covariance matrix  $\Sigma$ , the univariate (scalar) case is a special case thus automatically taken care of. Without loss of generality, let's consider zero mean. Denote  $\mathcal{N}(\mathbf{x}; \mathbf{0}, \Sigma) = \phi(\mathbf{x})$ . For any other distribution  $f(\mathbf{x})$  with the same covariance matrix  $\Sigma$ , first note that  $\int_{\mathbf{x}} f(\mathbf{x}) \log \phi(\mathbf{x}) d\mathbf{x} = \int_{\mathbf{x}} \phi(\mathbf{x}) \log \phi(\mathbf{x}) d\mathbf{x}$  (to be show in the next slide). Then,

$$0 \leq KL(f \parallel \phi) = \int_{\mathbf{x}} f(\mathbf{x}) \log \frac{f(\mathbf{x})}{\phi(\mathbf{x})} d\mathbf{x}$$



# Normal distribution has highest entropy

For fixed variance (covariance matrix), normal distribution has highest entropy

## Proof

Let's consider the multivariate case with a fixed covariance matrix  $\Sigma$ , the univariate (scalar) case is a special case thus automatically taken care of.

Without loss of generality, let's consider zero mean. Denote

$\mathcal{N}(\mathbf{x}; \mathbf{0}, \Sigma) = \phi(\mathbf{x})$ . For any other distribution  $f(\mathbf{x})$  with the same covariance matrix  $\Sigma$ , first note that  $\int_{\mathbf{x}} f(\mathbf{x}) \log \phi(\mathbf{x}) d\mathbf{x} = \int_{\mathbf{x}} \phi(\mathbf{x}) \log \phi(\mathbf{x}) d\mathbf{x}$  (to be show in the next slide). Then,

$$0 \leq KL(f \parallel \phi) = \int_{\mathbf{x}} f(\mathbf{x}) \log \frac{f(\mathbf{x})}{\phi(\mathbf{x})} d\mathbf{x} = -h(f) - \int_{\mathbf{x}} f(\mathbf{x}) \log \phi(\mathbf{x}) d\mathbf{x}$$

# Normal distribution has highest entropy

For fixed variance (covariance matrix), normal distribution has highest entropy

## Proof

Let's consider the multivariate case with a fixed covariance matrix  $\Sigma$ , the univariate (scalar) case is a special case thus automatically taken care of.

Without loss of generality, let's consider zero mean. Denote

$\mathcal{N}(\mathbf{x}; \mathbf{0}, \Sigma) = \phi(\mathbf{x})$ . For any other distribution  $f(\mathbf{x})$  with the same covariance matrix  $\Sigma$ , first note that  $\int_{\mathbf{x}} f(\mathbf{x}) \log \phi(\mathbf{x}) d\mathbf{x} = \int_{\mathbf{x}} \phi(\mathbf{x}) \log \phi(\mathbf{x}) d\mathbf{x}$  (to be show in the next slide). Then,

$$\begin{aligned} 0 \leq KL(f \parallel \phi) &= \int_{\mathbf{x}} f(\mathbf{x}) \log \frac{f(\mathbf{x})}{\phi(\mathbf{x})} d\mathbf{x} = -h(f) - \int_{\mathbf{x}} f(\mathbf{x}) \log \phi(\mathbf{x}) d\mathbf{x} \\ &= -h(f) - \int_{\mathbf{x}} \phi(\mathbf{x}) \log \phi(\mathbf{x}) d\mathbf{x} = -h(f) + h(\phi) \end{aligned}$$

$$\int_{\mathbf{x}} f(\mathbf{x}) \log \phi(\mathbf{x}) d\mathbf{x} = \int_{\mathbf{x}} \phi(\mathbf{x}) \log \phi(\mathbf{x}) d\mathbf{x}$$

$$\int_{\mathbf{x}} \phi(\mathbf{x}) \log \phi(\mathbf{x}) d\mathbf{x} = \int_{\mathbf{x}} \phi(\mathbf{x}) \left[ -\log \sqrt{\det(2\pi\Sigma)} - \frac{1}{2} \mathbf{x}^T \Sigma^{-1} \mathbf{x} \right] d\mathbf{x}$$

$$\int_{\mathbf{x}} f(\mathbf{x}) \log \phi(\mathbf{x}) d\mathbf{x} = \int_{\mathbf{x}} \phi(\mathbf{x}) \log \phi(\mathbf{x}) d\mathbf{x}$$

$$\begin{aligned} \int_{\mathbf{x}} \phi(\mathbf{x}) \log \phi(\mathbf{x}) d\mathbf{x} &= \int_{\mathbf{x}} \phi(\mathbf{x}) \left[ -\log \sqrt{\det(2\pi\Sigma)} - \frac{1}{2} \mathbf{x}^T \Sigma^{-1} \mathbf{x} \right] d\mathbf{x} \\ &= \int_{\mathbf{x}} \phi(\mathbf{x}) \left[ -\log \sqrt{\det(2\pi\Sigma)} - \frac{1}{2} \sum_{i,j} x_i [\Sigma^{-1}]_{i,j} x_j \right] d\mathbf{x} \end{aligned}$$

$$\int_{\mathbf{x}} f(\mathbf{x}) \log \phi(\mathbf{x}) d\mathbf{x} = \int_{\mathbf{x}} \phi(\mathbf{x}) \log \phi(\mathbf{x}) d\mathbf{x}$$

$$\begin{aligned} \int_{\mathbf{x}} \phi(\mathbf{x}) \log \phi(\mathbf{x}) d\mathbf{x} &= \int_{\mathbf{x}} \phi(\mathbf{x}) \left[ -\log \sqrt{\det(2\pi\Sigma)} - \frac{1}{2} \mathbf{x}^T \Sigma^{-1} \mathbf{x} \right] d\mathbf{x} \\ &= \int_{\mathbf{x}} \phi(\mathbf{x}) \left[ -\log \sqrt{\det(2\pi\Sigma)} - \frac{1}{2} \sum_{i,j} x_i [\Sigma^{-1}]_{i,j} x_j \right] d\mathbf{x} \\ &= \int_{\mathbf{x}} \phi(\mathbf{x}) \left[ -\log \sqrt{\det(2\pi\Sigma)} - \frac{1}{2} \sum_{i,j} [\Sigma^{-1}]_{i,j} x_i x_j \right] d\mathbf{x} \end{aligned}$$

$$\int_{\mathbf{x}} f(\mathbf{x}) \log \phi(\mathbf{x}) d\mathbf{x} = \int_{\mathbf{x}} \phi(\mathbf{x}) \log \phi(\mathbf{x}) d\mathbf{x}$$

$$\begin{aligned} \int_{\mathbf{x}} \phi(\mathbf{x}) \log \phi(\mathbf{x}) d\mathbf{x} &= \int_{\mathbf{x}} \phi(\mathbf{x}) \left[ -\log \sqrt{\det(2\pi\Sigma)} - \frac{1}{2} \mathbf{x}^T \Sigma^{-1} \mathbf{x} \right] d\mathbf{x} \\ &= \int_{\mathbf{x}} \phi(\mathbf{x}) \left[ -\log \sqrt{\det(2\pi\Sigma)} - \frac{1}{2} \sum_{i,j} x_i [\Sigma^{-1}]_{i,j} x_j \right] d\mathbf{x} \\ &= \int_{\mathbf{x}} \phi(\mathbf{x}) \left[ -\log \sqrt{\det(2\pi\Sigma)} - \frac{1}{2} \sum_{i,j} [\Sigma^{-1}]_{i,j} x_i x_j \right] d\mathbf{x} \\ &= \int_{\mathbf{x}} f(\mathbf{x}) \left[ -\log \sqrt{\det(2\pi\Sigma)} - \frac{1}{2} \sum_{i,j} [\Sigma^{-1}]_{i,j} x_i x_j \right] d\mathbf{x} \end{aligned}$$

$$\int_{\mathbf{x}} f(\mathbf{x}) \log \phi(\mathbf{x}) d\mathbf{x} = \int_{\mathbf{x}} \phi(\mathbf{x}) \log \phi(\mathbf{x}) d\mathbf{x}$$

$$\begin{aligned} \int_{\mathbf{x}} \phi(\mathbf{x}) \log \phi(\mathbf{x}) d\mathbf{x} &= \int_{\mathbf{x}} \phi(\mathbf{x}) \left[ -\log \sqrt{\det(2\pi\Sigma)} - \frac{1}{2} \mathbf{x}^T \Sigma^{-1} \mathbf{x} \right] d\mathbf{x} \\ &= \int_{\mathbf{x}} \phi(\mathbf{x}) \left[ -\log \sqrt{\det(2\pi\Sigma)} - \frac{1}{2} \sum_{i,j} x_i [\Sigma^{-1}]_{ij} x_j \right] d\mathbf{x} \\ &= \int_{\mathbf{x}} \phi(\mathbf{x}) \left[ -\log \sqrt{\det(2\pi\Sigma)} - \frac{1}{2} \sum_{i,j} [\Sigma^{-1}]_{ij} x_i x_j \right] d\mathbf{x} \\ &= \int_{\mathbf{x}} f(\mathbf{x}) \left[ -\log \sqrt{\det(2\pi\Sigma)} - \frac{1}{2} \sum_{i,j} [\Sigma^{-1}]_{ij} x_i x_j \right] d\mathbf{x} \\ &= \int_{\mathbf{x}} f(\mathbf{x}) \log \phi(\mathbf{x}) d\mathbf{x} \end{aligned}$$

# Application: Cross-entropy and cross-entropy error

In machine learning, it is often needed to assess the quality of a trained system. Consider the example of classifying an the political affiliation of an individual

computed	targets	correct?
0.3 0.3 0.4	0 0 1 (democrat)	yes
0.3 0.4 0.3	0 1 0 (republican)	yes
0.1 0.2 0.7	1 0 0 (other)	no

computed	targets	correct?
0.1 0.2 0.7	0 0 1 (democrat)	yes
0.1 0.7 0.2	0 1 0 (republican)	yes
0.3 0.4 0.3	1 0 0 (other)	no

In a first glance, both examples appear to work equally well (or bad). Both have one classification error. However, a closer look will suggest the prediction of LHS is worse than RHS (why?)

(<https://jamesmccaffrey.wordpress.com/2013/11/05/why-you-should-use-cross-entropy-error-instead-of-classification-error-or-mean-squared-error-for-neural-network-classifier-training/>)



# Application: Cross-entropy and cross-entropy error

In machine learning, it is often needed to assess the quality of a trained system. Consider the example of classifying an the political affiliation of an individual

computed	targets	correct?
0.3 0.3 0.4	0 0 1 (democrat)	yes
0.3 0.4 0.3	0 1 0 (republican)	yes
0.1 0.2 0.7	1 0 0 (other)	no

computed	targets	correct?
0.1 0.2 0.7	0 0 1 (democrat)	yes
0.1 0.7 0.2	0 1 0 (republican)	yes
0.3 0.4 0.3	1 0 0 (other)	no

In a first glance, both examples appear to work equally well (or bad). Both have one classification error. However, a closer look will suggest the prediction of LHS is worse than RHS (why?) For a better assessment, we can treat both the computed result and the target result as distribution and compare them with KL-divergence. Namely

$$\begin{aligned}
 KL(p_{target} || p_{computed}) &= \sum_{group} p_{target}(group) \log \frac{p_{target}(group)}{p_{computed}(group)} \\
 &= -H(p_{target}) - \underbrace{\sum_{group} p_{target}(group) \log p_{computed}(group)}_{cross\ entropy}
 \end{aligned}$$

(<https://jamesmccaffrey.wordpress.com/2013/11/05/why-you-should-use-cross-entropy-error-instead-of-classification-error-or-mean-squared-error-for-neural-network-classifier-training/>)

# Application: Cross-entropy and cross-entropy error

$$\begin{aligned} \text{Cross entropy}(p\|q) &\triangleq \sum_x p(x) \log \frac{1}{q(x)} = E_p[-\log q(X)] \\ &= H(p) + KL(p\|q) \end{aligned}$$

# Application: Cross-entropy and cross-entropy error

$$\begin{aligned} \text{Cross entropy}(p\|q) &\triangleq \sum_x p(x) \log \frac{1}{q(x)} = E_p[-\log q(X)] \\ &= H(p) + KL(p\|q) \end{aligned}$$

- To compute KL-divergence, one needs to find  $H(p_{target})$ , which is independent of the machine learning system and thus does not reflect the performance of the system

# Application: Cross-entropy and cross-entropy error

$$\begin{aligned} \text{Cross entropy}(p\|q) &\triangleq \sum_x p(x) \log \frac{1}{q(x)} = E_p[-\log q(X)] \\ &= H(p) + KL(p\|q) \end{aligned}$$

- To compute KL-divergence, one needs to find  $H(p_{target})$ , which is independent of the machine learning system and thus does not reflect the performance of the system
- Thus in practice, cross-entropy is commonly used instead of KL-divergence to measure the performance of a machine learning system

# Example: Text processing

- In text processing, it is common that one may need to measure the similarity between two documents  $D_1$  and  $D_2$ .

# Example: Text processing

- In text processing, it is common that one may need to measure the similarity between two documents  $D_1$  and  $D_2$ .
- How to represent documents? One may use the “bag of words”. That is, to convert document into a vector of numbers. Each number is the count of a corresponding word

# Example: Text processing

- In text processing, it is common that one may need to measure the similarity between two documents  $D_1$  and  $D_2$ .
- How to represent documents? One may use the “bag of words”. That is, to convert document into a vector of numbers. Each number is the count of a corresponding word
- One can then compares two documents using cross entropy

$$\text{Cross entropy}(p_1 \| p_2) = \sum_w p_1(w) \log \frac{1}{p_2(w)},$$

where  $p_1$  and  $p_2$  are the word distributions of documents  $D_1$  and  $D_2$ , respectively

# TF-IDF and cross entropy

It may be also interesting of comparing word distribution of a document to the word distribution across all documents That is, let  $q$  be the word distribution across all documents,

$$\begin{aligned}
 \text{Cross entropy}(p_1 \| q) &= \sum_w p_1(w) \log \frac{1}{q(w)} \\
 &= \sum_w \underbrace{\frac{\# w \text{ in } D_1}{\text{total } \# \text{ words in } D_1}}_{TF-IDF(w)} \log \frac{\text{total } \# \text{ docs}}{\# \text{ doc with } w},
 \end{aligned}$$

where  $TF-IDF(w)$ , short for term frequency-inverse document frequency, can reflect how important of the word  $w$  to the target document and can be used in search engine



# Definition

As  $H(X)$  is equivalent to the information revealed by  $X$  and  $H(X|Y)$  the remaining information of  $X$  knowing  $Y$ , we expect that  $H(X) - H(X|Y)$  is the information of  $X$  shared by  $Y \Rightarrow$  “mutual information”

$$I(X; Y) \triangleq H(X) - H(X|Y)$$

# Definition

As  $H(X)$  is equivalent to the information revealed by  $X$  and  $H(X|Y)$  the remaining information of  $X$  knowing  $Y$ , we expect that  $H(X) - H(X|Y)$  is the information of  $X$  shared by  $Y \Rightarrow$  “mutual information”

$$I(X; Y) \triangleq H(X) - H(X|Y)$$

Similarly, we can define the “conditional mutual information” shared between  $X$  and  $Y$  given  $Z$  as

$$I(X; Y|Z) \triangleq H(X|Z) - H(X|Y, Z)$$

# Property of mutual information

$$I(X; Y) = I(Y; X) \geq 0$$

The definition is symmetric and non-negative as desired.

$$I(X; Y) = H(X) - H(X|Y) = E[-\log p(X)] - E[-\log p(X|Y)]$$

# Property of mutual information

$$I(X; Y) = I(Y; X) \geq 0$$

The definition is symmetric and non-negative as desired.

$$\begin{aligned} I(X; Y) &= H(X) - H(X|Y) = E[-\log p(X)] - E[-\log p(X|Y)] \\ &= - \sum_x p(x) \log p(x) + \sum_{x,y} p(x,y) \log p(x|y) \end{aligned}$$

# Property of mutual information

$$I(X; Y) = I(Y; X) \geq 0$$

The definition is symmetric and non-negative as desired.

$$I(X; Y) = H(X) - H(X|Y) = E[-\log p(X)] - E[-\log p(X|Y)]$$

$$= - \sum_x p(x) \log p(x) + \sum_{x,y} p(x,y) \log p(x|y)$$

$$= - \sum_{x,y} p(x,y) \log p(x) + \sum_{x,y} p(x,y) \log p(x|y) = \sum_{x,y} p(x,y) \log \frac{p(x|y)}{p(x)}$$

# Property of mutual information

$$I(X; Y) = I(Y; X) \geq 0$$

The definition is symmetric and non-negative as desired.

$$\begin{aligned} I(X; Y) &= H(X) - H(X|Y) = E[-\log p(X)] - E[-\log p(X|Y)] \\ &= -\sum_x p(x) \log p(x) + \sum_{x,y} p(x,y) \log p(x|y) \\ &= -\sum_{x,y} p(x,y) \log p(x) + \sum_{x,y} p(x,y) \log p(x|y) = \sum_{x,y} p(x,y) \log \frac{p(x|y)}{p(x)} \\ &= \sum_{x,y} p(x,y) \log \frac{p(x,y)}{p(x)p(y)} \end{aligned}$$

# Property of mutual information

$$I(X; Y) = I(Y; X) \geq 0$$

The definition is symmetric and non-negative as desired.

$$\begin{aligned}
 I(X; Y) &= H(X) - H(X|Y) = E[-\log p(X)] - E[-\log p(X|Y)] \\
 &= - \sum_x p(x) \log p(x) + \sum_{x,y} p(x,y) \log p(x|y) \\
 &= - \sum_{x,y} p(x,y) \log p(x) + \sum_{x,y} p(x,y) \log p(x|y) = \sum_{x,y} p(x,y) \log \frac{p(x|y)}{p(x)} \\
 &= \sum_{x,y} p(x,y) \log \frac{p(x,y)}{p(x)p(y)} = KL(p(x,y) \| p(x)p(y)) \geq 0
 \end{aligned}$$

# Property of conditional mutual information

$$I(X; Y|Z) = I(Y; X|Z) \geq 0$$

The definition is symmetric and non-negative as desired.

$$I(X; Y|Z) = H(X|Z) - H(X|Y, Z) = E[-\log p(X|Z)] - E[-\log p(X|Y, Z)]$$



# Property of conditional mutual information

$$I(X; Y|Z) = I(Y; X|Z) \geq 0$$

The definition is symmetric and non-negative as desired.

$$\begin{aligned} I(X; Y|Z) &= H(X|Z) - H(X|Y, Z) = E[-\log p(X|Z)] - E[-\log p(X|Y, Z)] \\ &= - \sum_{x,z} p(x, z) \log p(x|z) + \sum_{x,y,z} p(x, y, z) \log p(x|y, z) \end{aligned}$$

# Property of conditional mutual information

$$I(X; Y|Z) = I(Y; X|Z) \geq 0$$

The definition is symmetric and non-negative as desired.

$$\begin{aligned} I(X; Y|Z) &= H(X|Z) - H(X|Y, Z) = E[-\log p(X|Z)] - E[-\log p(X|Y, Z)] \\ &= - \sum_{x,z} p(x, z) \log p(x|z) + \sum_{x,y,z} p(x, y, z) \log p(x|y, z) \\ &= - \sum_{x,y,z} p(x, y, z) \log p(x|z) + \sum_{x,y,z} p(x, y, z) \log p(x|y, z) \\ &= \sum_{x,y,z} p(x, y, z) \log \frac{p(x|y, z)}{p(x|z)} \end{aligned}$$

# Property of conditional mutual information

$$I(X; Y|Z) = I(Y; X|Z) \geq 0$$

The definition is symmetric and non-negative as desired.

$$\begin{aligned}
 I(X; Y|Z) &= H(X|Z) - H(X|Y, Z) = E[-\log p(X|Z)] - E[-\log p(X|Y, Z)] \\
 &= - \sum_{x,z} p(x, z) \log p(x|z) + \sum_{x,y,z} p(x, y, z) \log p(x|y, z) \\
 &= - \sum_{x,y,z} p(x, y, z) \log p(x|z) + \sum_{x,y,z} p(x, y, z) \log p(x|y, z) \\
 &= \sum_{x,y,z} p(x, y, z) \log \frac{p(x|y, z)}{p(x|z)} \\
 &= \sum_z p(z) \sum_{x,y} p(x, y|z) \log \frac{p(x, y|z)}{p(x|z)p(y|z)} \\
 &= \sum_z p(z) KL(p(x, y|z) \| p(x|z)p(y|z)) \geq 0
 \end{aligned}$$

# Independence and mutual information

$$I(X; Y) = 0 \Leftrightarrow X \perp Y$$

$$I(X; Y) = KL(p(x, y) \| p(x)p(y)) = 0$$

implies  $p(x, y) = p(x)p(y)$ . Therefore  $X \perp Y$

# Independence and mutual information

$$I(X; Y) = 0 \Leftrightarrow X \perp Y$$

$$I(X; Y) = KL(p(x, y) \| p(x)p(y)) = 0$$

implies  $p(x, y) = p(x)p(y)$ . Therefore  $X \perp Y$

$$I(X; Y|Z) = 0 \Leftrightarrow X \perp Y|Z$$

$$I(X; Y|Z) = \sum_z p(z) KL(p(x, y|z) \| p(x|z)p(y|z)) = 0$$

implies  $p(x, y|z) = p(x|z)p(y|z)$  for all  $z$  s.t.  $p(z) > 0$ . Therefore  $X \perp Y|Z$

## Remark

This is just as what we expect. If there is no share information between  $X$  and  $Y$ , they should be independent!

# Chain rule for mutual information

$$I(X_1, X_2, \dots, X_N | Y)$$

# Chain rule for mutual information

$$\begin{aligned} & I(X_1, X_2, \dots, X_N | Y) \\ &= H(X_1, X_2, \dots, X_N) - H(X_1, X_2, \dots, X_N | Y) \end{aligned}$$

# Chain rule for mutual information

$$\begin{aligned} & I(X_1, X_2, \dots, X_N | Y) \\ &= H(X_1, X_2, \dots, X_N) - H(X_1, X_2, \dots, X_N | Y) \\ &= \sum_{i=1}^N H(X_i | X^{i-1}) - H(X_i | X^{i-1}, Y) \end{aligned}$$

N.B.  $X^N = X_1, X_2, \dots, X_N$



# Chain rule for mutual information

$$\begin{aligned} & I(X_1, X_2, \dots, X_N | Y) \\ &= H(X_1, X_2, \dots, X_N) - H(X_1, X_2, \dots, X_N | Y) \\ &= \sum_{i=1}^N H(X_i | X^{i-1}) - H(X_i | X^{i-1}, Y) \\ &= \sum_{i=1}^N I(X_i; Y | X^{i-1}) \end{aligned}$$

N.B.  $X^N = X_1, X_2, \dots, X_N$

# Mutual information for continuous variables

For continuous  $X, Y, Z$ , we can define  $I(X; Y) = h(X) - h(X|Y)$  and  $I(X; Y|Z) = h(X) - h(X|Y, Z)$

Then, the followings still hold true

# Mutual information for continuous variables

For continuous  $X, Y, Z$ , we can define  $I(X; Y) = h(X) - h(X|Y)$  and  $I(X; Y|Z) = h(X) - h(X|Y, Z)$

Then, the followings still hold true

- $I(X; Y) = KL(p(x, y) \| p(x)p(y)) = I(Y; X) \geq 0$

# Mutual information for continuous variables

For continuous  $X, Y, Z$ , we can define  $I(X; Y) = h(X) - h(X|Y)$  and  $I(X; Y|Z) = h(X) - h(X|Y, Z)$

Then, the followings still hold true

- $I(X; Y) = KL(p(x, y) \| p(x)p(y)) = I(Y; X) \geq 0$
- $I(X; Y|Z) = \int_z p(z) KL(p(x, y|z) \| p(x|z)p(y|z)) dz = I(Y; X|Z) \geq 0$

# Mutual information for continuous variables

For continuous  $X, Y, Z$ , we can define  $I(X; Y) = h(X) - h(X|Y)$  and  $I(X; Y|Z) = h(X) - h(X|Y, Z)$

Then, the followings still hold true

- $I(X; Y) = KL(p(x, y) \| p(x)p(y)) = I(Y; X) \geq 0$
- $I(X; Y|Z) = \int_z p(z) KL(p(x, y|z) \| p(x|z)p(y|z)) dz = I(Y; X|Z) \geq 0$
- $I(X; Y) = 0 \Leftrightarrow X \perp Y$

# Mutual information for continuous variables

For continuous  $X, Y, Z$ , we can define  $I(X; Y) = h(X) - h(X|Y)$  and  $I(X; Y|Z) = h(X) - h(X|Y, Z)$

Then, the followings still hold true

- $I(X; Y) = KL(p(x, y) \| p(x)p(y)) = I(Y; X) \geq 0$
- $I(X; Y|Z) = \int_z p(z) KL(p(x, y|z) \| p(x|z)p(y|z)) dz = I(Y; X|Z) \geq 0$
- $I(X; Y) = 0 \Leftrightarrow X \perp Y$
- $I(X; Y|Z) = 0 \Leftrightarrow X \perp Y|Z$

# Mutual information for continuous variables

For continuous  $X, Y, Z$ , we can define  $I(X; Y) = h(X) - h(X|Y)$  and  $I(X; Y|Z) = h(X) - h(X|Y, Z)$

Then, the followings still hold true

- $I(X; Y) = KL(p(x, y) \| p(x)p(y)) = I(Y; X) \geq 0$
- $I(X; Y|Z) = \int_z p(z) KL(p(x, y|z) \| p(x|z)p(y|z)) dz = I(Y; X|Z) \geq 0$
- $I(X; Y) = 0 \Leftrightarrow X \perp Y$
- $I(X; Y|Z) = 0 \Leftrightarrow X \perp Y|Z$
- $I(X_1, X_2, \dots, X_N|Y) = \sum_{i=1}^N I(X_i; Y|X^{i-1})$

# Conditioning reduces entropy

Given more information, the residual information (uncertainty) should decrease.



# Conditioning reduces entropy

Given more information, the residual information (uncertainty) should decrease. More precisely,

$$H(X) \geq H(X|Y) \quad H(X|Y) \geq H(X|Y, Z)$$

This is obvious from our previous discussion since

$$H(X) - H(X|Y) = I(X; Y) \geq 0 \text{ and}$$

$$H(X|Y) - H(X|Y, Z) = I(X; Z|Y) \geq 0$$

# Conditioning reduces entropy

Given more information, the residual information (uncertainty) should decrease. More precisely,

$$H(X) \geq H(X|Y) \quad H(X|Y) \geq H(X|Y, Z)$$

This is obvious from our previous discussion since

$$H(X) - H(X|Y) = I(X; Y) \geq 0 \text{ and} \\ H(X|Y) - H(X|Y, Z) = I(X; Z|Y) \geq 0$$

Of course, we also have

$$h(X) \geq h(X|Y) \quad h(X|Y) \geq h(X|Y, Z)$$

$$\text{since } h(X) - h(X|Y) = I(X; Y) \geq 0 \text{ and} \\ h(X|Y) - h(X|Y, Z) = I(X; Z|Y) \geq 0$$

# Data processing inequality

If random variables  $X, Y, Z$  satisfy  $X \leftrightarrow Y \leftrightarrow Z$ , then

$$I(X; Y) \geq I(X; Z).$$

Proof

$$I(X; Y) = I(X; Y, Z) - I(X; Z|Y)$$

# Data processing inequality

If random variables  $X, Y, Z$  satisfy  $X \leftrightarrow Y \leftrightarrow Z$ , then

$$I(X; Y) \geq I(X; Z).$$

Proof

$$\begin{aligned} I(X; Y) &= I(X; Y, Z) - I(X; Z|Y) \\ &= I(X; Y, Z) \quad (\text{since } X \leftrightarrow Y \leftrightarrow Z) \end{aligned}$$

# Data processing inequality

If random variables  $X, Y, Z$  satisfy  $X \leftrightarrow Y \leftrightarrow Z$ , then

$$I(X; Y) \geq I(X; Z).$$

Proof

$$\begin{aligned} I(X; Y) &= I(X; Y, Z) - I(X; Z|Y) \\ &= I(X; Y, Z) \quad (\text{since } X \leftrightarrow Y \leftrightarrow Z) \\ &= I(X; Z) + I(X; Y|Z) \\ &\geq I(X; Z) \end{aligned}$$

# Application: perfect secrecy

## Example (A simple cryptography example)

- Say you have a very personal letter that you don't want to let anyone else except some special someone to read

# Application: perfect secrecy

## Example (A simple cryptography example)

- Say you have a very personal letter that you don't want to let anyone else except some special someone to read
- You will first encrypt the letter to some code. To decrypt the message, you will need some key and you will also pass it to your special someone.

# Application: perfect secrecy

## Example (A simple cryptography example)

- Say you have a very personal letter that you don't want to let anyone else except some special someone to read
- You will first encrypt the letter to some code. To decrypt the message, you will need some key and you will also pass it to your special someone. Translate to the cryptography language/symbols
  - Letter: plaintext message  $M$
  - Code: ciphertext  $C$
  - Key: key  $K$



# Application: perfect secrecy

## Example (A simple cryptography example)

- Say you have a very personal letter that you don't want to let anyone else except some special someone to read
- You will first encrypt the letter to some code. To decrypt the message, you will need some key and you will also pass it to your special someone. Translate to the cryptography language/symbols
  - Letter: plaintext message  $M$
  - Code: ciphertext  $C$
  - Key: key  $K$

## Remark

*Shannon's result: to ensure perfect secrecy, we can show that*  
 $H(M) \leq H(K)$

# Application: perfect secrecy

Recall that  $M, C, K$  be plaintext message, ciphertext, and key, respectively

## Assumption

*We will assume here that we have a **non-probabilistic** encryption scheme. In other words, each plaintext message maps to a unique ciphertext given a fixed key. So there is no ambiguity during decoding. Therefore,*

$$H(M|C, K) = 0$$

# Application: perfect secrecy

Recall that  $M, C, K$  be plaintext message, ciphertext, and key, respectively

## Assumption

*We will assume here that we have a **non-probabilistic** encryption scheme. In other words, each plaintext message maps to a unique ciphertext given a fixed key. So there is no ambiguity during decoding. Therefore,*

$$H(M|C, K) = 0$$

## Remark (Independence)

*For perfect secrecy, one should not be able to deduce anything regarding the message from the ciphertext. Therefore,  $C$  and  $M$  should be independent.*

# Application: perfect secrecy

Recall that  $M, C, K$  be plaintext message, ciphertext, and key, respectively

## Assumption

*We will assume here that we have a **non-probabilistic** encryption scheme. In other words, each plaintext message maps to a unique ciphertext given a fixed key. So there is no ambiguity during decoding. Therefore,*

$$H(M|C, K) = 0$$

## Remark (Independence)

*For perfect secrecy, one should not be able to deduce anything regarding the message from the ciphertext. Therefore,  $C$  and  $M$  should be independent. Thus,*

$$I(C; M) = 0 \Rightarrow H(M) = H(M|C) + I(C; M) = H(M|C)$$

# Application: perfect secrecy

## Lemma (Entropy bound)

For any **non-probabilistic** encryption scheme,  $H(M|C) \leq H(K|C)$

# Application: perfect secrecy

## Lemma (Entropy bound)

For any **non-probabilistic** encryption scheme,  $H(M|C) \leq H(K|C)$

## Proof.

Recall that for non-probabilistic encryption scheme,  $H(M|K, C) = 0 \Rightarrow H(M|C) \leq H(M, K|C)$

# Application: perfect secrecy

## Lemma (Entropy bound)

For any **non-probabilistic** encryption scheme,  $H(M|C) \leq H(K|C)$

## Proof.

Recall that for non-probabilistic encryption scheme,  $H(M|K, C) = 0 \Rightarrow H(M|C) \leq H(M, K|C) = H(K|C) + H(M|K, C) = H(K|C)$  □

# Application: perfect secrecy

## Lemma (Entropy bound)

For any **non-probabilistic** encryption scheme,  $H(M|C) \leq H(K|C)$

## Proof.

Recall that for non-probabilistic encryption scheme,  $H(M|K, C) = 0 \Rightarrow H(M|C) \leq H(M, K|C) = H(K|C) + H(M|K, C) = H(K|C)$   $\square$

## Corollary (Entropy bound)

For any non-probabilistic encryption scheme,  $H(M|C) \leq H(K)$



# Application: perfect secrecy

## Lemma (Entropy bound)

For any **non-probabilistic** encryption scheme,  $H(M|C) \leq H(K|C)$

## Proof.

Recall that for non-probabilistic encryption scheme,  $H(M|K, C) = 0 \Rightarrow H(M|C) \leq H(M, K|C) = H(K|C) + H(M|K, C) = H(K|C)$   $\square$

## Corollary (Entropy bound)

For any non-probabilistic encryption scheme,  $H(M|C) \leq H(K)$

## Theorem (Perfect secrecy)

We have perfect secrecy if  $H(M) \leq H(K)$

# Application: perfect secrecy

## Lemma (Entropy bound)

For any **non-probabilistic** encryption scheme,  $H(M|C) \leq H(K|C)$

## Proof.

Recall that for non-probabilistic encryption scheme,  $H(M|K, C) = 0 \Rightarrow H(M|C) \leq H(M, K|C) = H(K|C) + H(M|K, C) = H(K|C)$   $\square$

## Corollary (Entropy bound)

For any non-probabilistic encryption scheme,  $H(M|C) \leq H(K)$

## Theorem (Perfect secrecy)

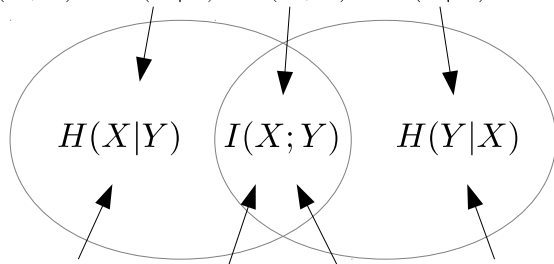
We have perfect secrecy if  $H(M) \leq H(K)$

## Proof.

Combine Corollary (Entropy bound) and Remark (Independence)  $\square$

# Summary

$$H(X, Y) = H(X|Y) + I(X; Y) + H(Y|X)$$

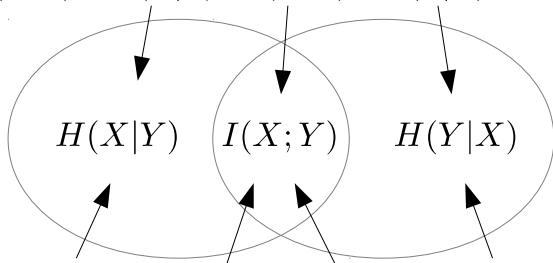


$$H(X) = H(X|Y) + I(X; Y)$$

$$I(X; Y) + H(Y|X) = H(Y)$$

## Review

$$H(X, Y) = H(X|Y) + I(X; Y) + H(Y|X)$$



$$H(X) = H(X|Y) + I(X; Y)$$

$$I(X; Y) + H(Y|X) = H(Y)$$

# Review

- Conditioning reduces entropy

# Review

- Conditioning reduces entropy
- Chain rules:
  - $H(X, Y, Z)$

# Review

- Conditioning reduces entropy
- Chain rules:
  - $H(X, Y, Z) = H(Z) + H(Y|X) + H(Z|X, Y)$
  - $H(X, Y, U|V)$

# Review

- Conditioning reduces entropy
- Chain rules:
  - $H(X, Y, Z) = H(Z) + H(Y|X) + H(Z|X, Y)$
  - $H(X, Y, U|V) = H(X|V) + H(Y|X, V) + H(U|Y, X, V)$
  - $I(X, Y, Z; U)$



# Review

- Conditioning reduces entropy
- Chain rules:
  - $H(X, Y, Z) = H(Z) + H(Y|X) + H(Z|X, Y)$
  - $H(X, Y, U|V) = H(X|V) + H(Y|X, V) + H(U|Y, X, V)$
  - $I(X, Y, Z; U) = I(X; U) + I(Y; U|X) + I(Z; U|X, Y)$
  - $I(X, Y, Z; U|V)$

# Review

- Conditioning reduces entropy
- Chain rules:
  - $H(X, Y, Z) = H(Z) + H(Y|X) + H(Z|X, Y)$
  - $H(X, Y, U|V) = H(X|V) + H(Y|X, V) + H(U|Y, X, V)$
  - $I(X, Y, Z; U) = I(X; U) + I(Y; U|X) + I(Z; U|X, Y)$
  - $I(X, Y, Z; U|V) = I(X; U|V) + I(Y; U|V, X) + I(Z; U|V, X, Y)$
- Data processing inequality: if  $X \perp Y|Z$ ,

# Review

- Conditioning reduces entropy
- Chain rules:
  - $H(X, Y, Z) = H(Z) + H(Y|X) + H(Z|X, Y)$
  - $H(X, Y, U|V) = H(X|V) + H(Y|X, V) + H(U|Y, X, V)$
  - $I(X, Y, Z; U) = I(X; U) + I(Y; U|X) + I(Z; U|X, Y)$
  - $I(X, Y, Z; U|V) = I(X; U|V) + I(Y; U|V, X) + I(Z; U|V, X, Y)$
- Data processing inequality: if  $X \perp Y|Z$ ,  $I(X; Y) \geq I(X; Z)$
- Independence and mutual information:
  - $X \perp Y \Leftrightarrow$

## Review

- Conditioning reduces entropy
- Chain rules:
  - $H(X, Y, Z) = H(Z) + H(Y|X) + H(Z|X, Y)$
  - $H(X, Y, U|V) = H(X|V) + H(Y|X, V) + H(U|Y, X, V)$
  - $I(X, Y, Z; U) = I(X; U) + I(Y; U|X) + I(Z; U|X, Y)$
  - $I(X, Y, Z; U|V) = I(X; U|V) + I(Y; U|V, X) + I(Z; U|V, X, Y)$
- Data processing inequality: if  $X \perp Y|Z$ ,  $I(X; Y) \geq I(X; Z)$
- Independence and mutual information:
  - $X \perp Y \Leftrightarrow I(X; Y) = 0$
  - $X \perp Y|Z \Leftrightarrow$

## Review

- Conditioning reduces entropy
- Chain rules:
  - $H(X, Y, Z) = H(Z) + H(Y|X) + H(Z|X, Y)$
  - $H(X, Y, U|V) = H(X|V) + H(Y|X, V) + H(U|Y, X, V)$
  - $I(X, Y, Z; U) = I(X; U) + I(Y; U|X) + I(Z; U|X, Y)$
  - $I(X, Y, Z; U|V) = I(X; U|V) + I(Y; U|V, X) + I(Z; U|V, X, Y)$
- Data processing inequality: if  $X \perp Y|Z$ ,  $I(X; Y) \geq I(X; Z)$
- Independence and mutual information:
  - $X \perp Y \Leftrightarrow I(X; Y) = 0$
  - $X \perp Y|Z \Leftrightarrow I(X; Y|Z) = 0$
- KL-divergence:  $KL(p||q) \triangleq$

## Review

- Conditioning reduces entropy
- Chain rules:
  - $H(X, Y, Z) = H(Z) + H(Y|X) + H(Z|X, Y)$
  - $H(X, Y, U|V) = H(X|V) + H(Y|X, V) + H(U|Y, X, V)$
  - $I(X, Y, Z; U) = I(X; U) + I(Y; U|X) + I(Z; U|X, Y)$
  - $I(X, Y, Z; U|V) = I(X; U|V) + I(Y; U|V, X) + I(Z; U|V, X, Y)$
- Data processing inequality: if  $X \perp Y|Z$ ,  $I(X; Y) \geq I(X; Z)$
- Independence and mutual information:
  - $X \perp Y \Leftrightarrow I(X; Y) = 0$
  - $X \perp Y|Z \Leftrightarrow I(X; Y|Z) = 0$
- KL-divergence:  $KL(p||q) \triangleq \sum_x p(x) \log \frac{p(x)}{q(x)}$

# This time

- Identification/Decision trees
- Random forests
- Law of Large Number
- Asymptotic equipartition (AEP) and typical sequences

# Vampire database

## Romanian Data Base

Vampire?	Shadow?	Garlic?	Complexion?	Accent?
No	?	Yes	Pale	None
No	Yes	Yes	Ruddy	None
Yes	?	No	Ruddy	None
Yes	No	No	Average	Heavy
Yes	?	No	Average	Odd
No	Yes	No	Pale	Heavy
No	Yes	No	Average	Heavy
No	?	Yes	Ruddy	Odd

([https://www.youtube.com/watch?v=SXBG3RGr\\_Rc](https://www.youtube.com/watch?v=SXBG3RGr_Rc))



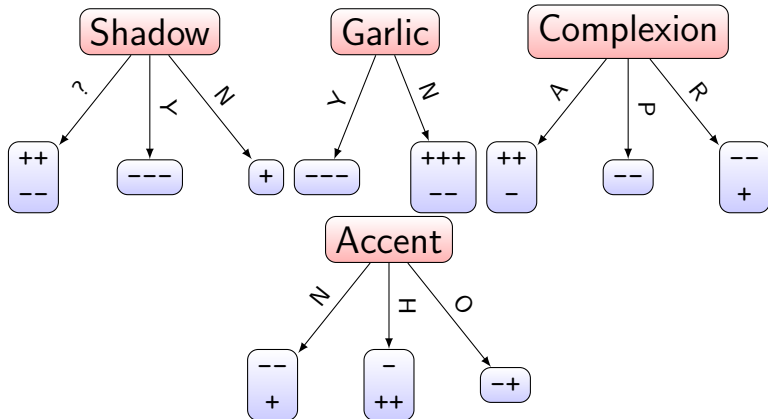
# Identifying vampire

Goal: Design a set of tests to identify vampires

## Potential difficulties

- Non-numerical data
- Some information may not matter
- Some may matter only sometimes
- Tests may be costly  $\Rightarrow$  conduct as few as possible

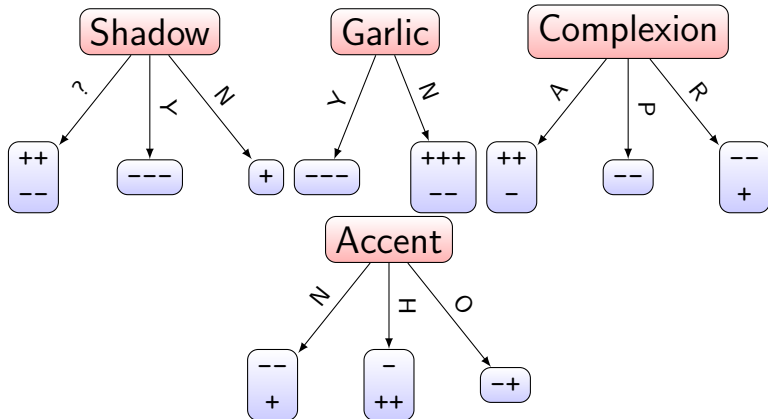
# Test trees



+ : Vampire

- : Not vampire

# Test trees

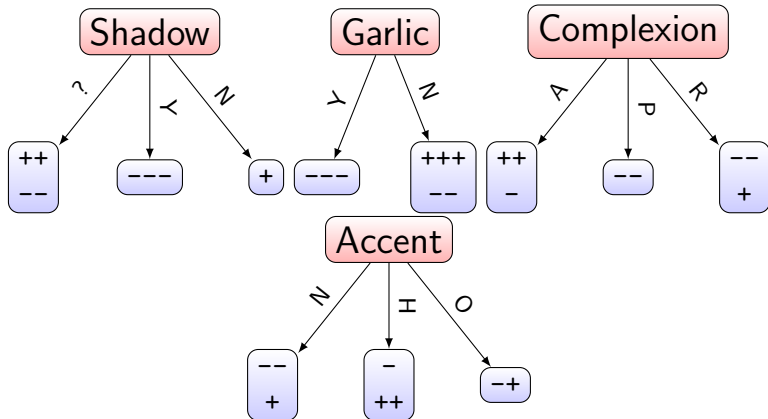


+ : Vampire

- : Not vampire

How to pick a good test?

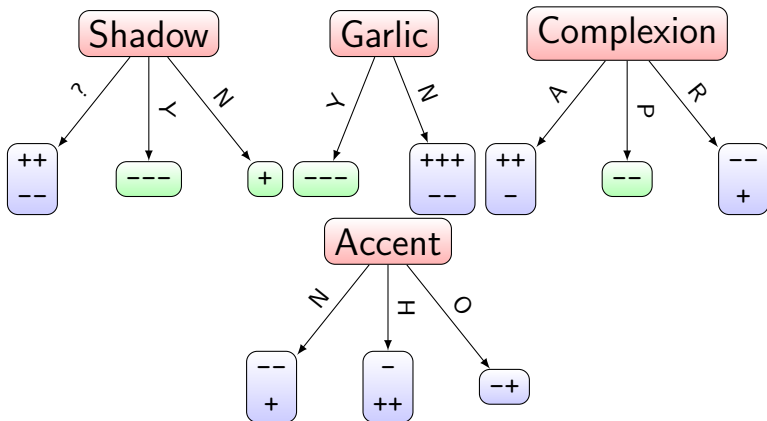
# Test trees



$+$  : Vampire       $-$  : Not vampire

How to pick a good test? Pick test that identifies most vampires (and non-vampires)!

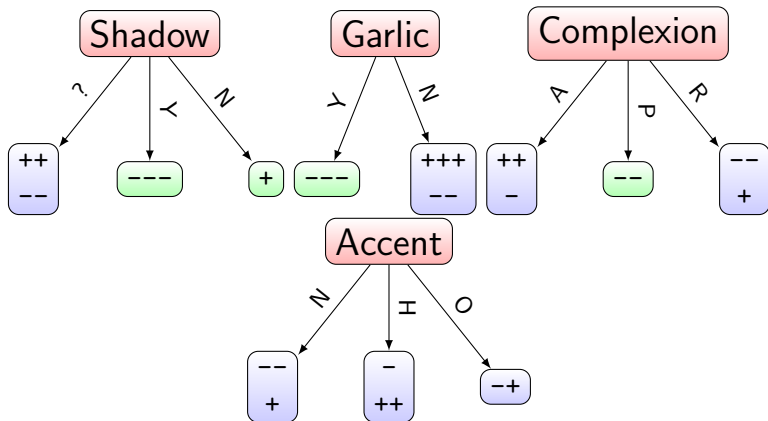
# Sizes of homogeneous sets



+ : Vampire

- : Not vampire

# Sizes of homogeneous sets



$+$  : Vampire

$-$  : Not vampire

Shadow: 4

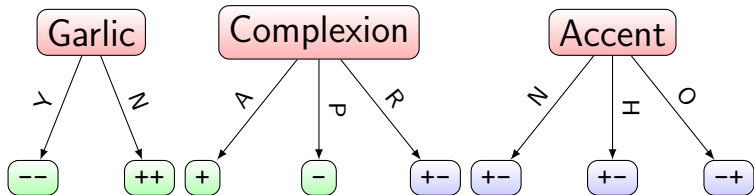
Garlic: 3

Complexion: 2

Accent: 0

# Picking second test

Let say we pick “shadow” as the first test after all. Then, for the remaining unclassified individuals,

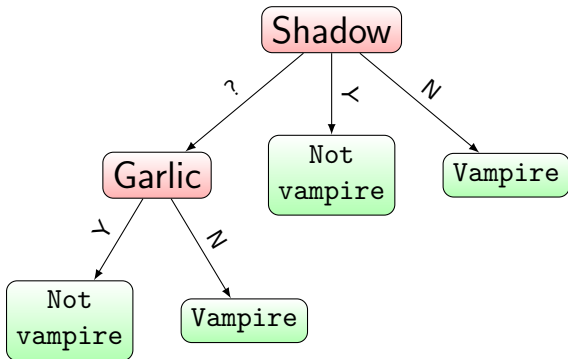


Garlic: 4

Complexion: 2

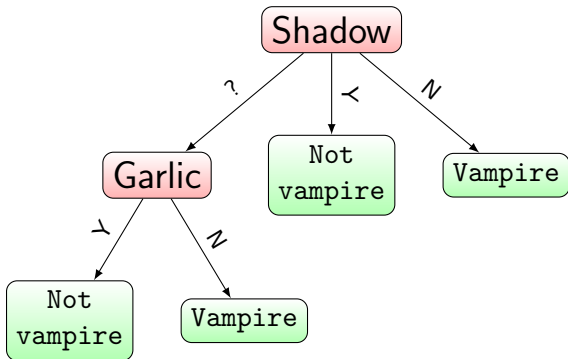
Accent: 0

# Combined tests





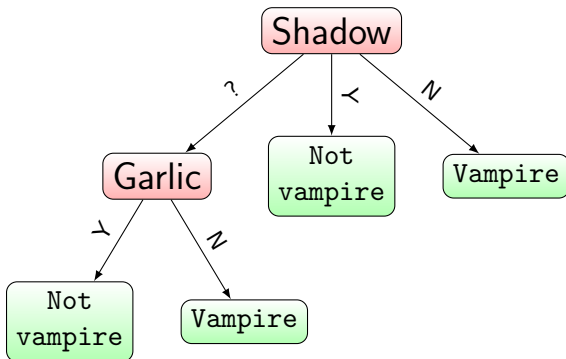
# Combined tests



## Problem

When our database size increases, none of the test likely to completely separate vampire from non-vampire. All tests will score 0 then.

# Combined tests



## Problem

When our database size increases, none of the test likely to completely separate vampire from non-vampire. All tests will score 0 then.

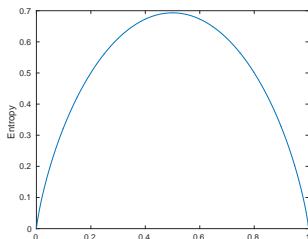
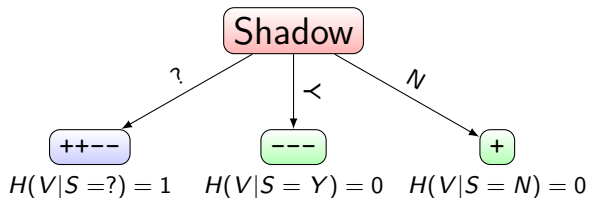
**Entropy comes to the rescue!**

# Conditional entropy as a measure of test efficiency

Consider the database is randomly sampled from a distribution. A set is

- Very homogeneous  $\approx$  high certainty
- Not so homogenous  $\approx$  high randomness

These can be measured with its entropy

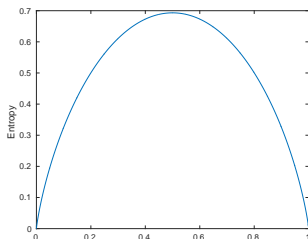
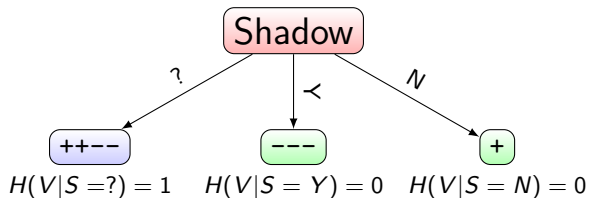


# Conditional entropy as a measure of test efficiency

Consider the database is randomly sampled from a distribution. A set is

- Very homogeneous  $\approx$  high certainty
- Not so homogenous  $\approx$  high randomness

These can be measured with its entropy



Remaining uncertainty given the test:

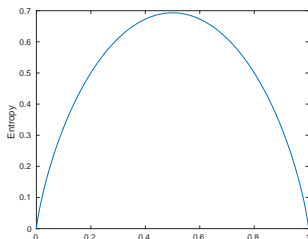
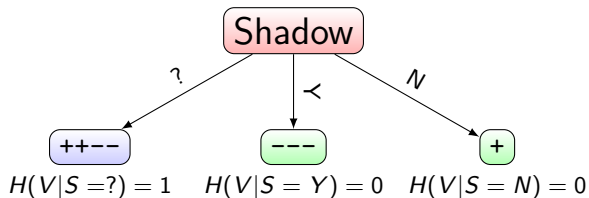
$$\frac{4}{8} H(V|S=?)$$

# Conditional entropy as a measure of test efficiency

Consider the database is randomly sampled from a distribution. A set is

- Very homogeneous  $\approx$  high certainty
- Not so homogenous  $\approx$  high randomness

These can be measured with its entropy



Remaining uncertainty given the test:

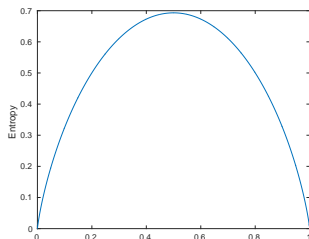
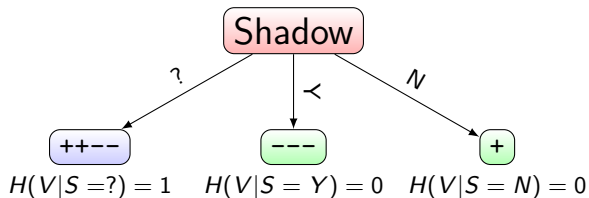
$$\frac{4}{8}H(V|S=?) + \frac{3}{8}H(V|S=Y)$$

# Conditional entropy as a measure of test efficiency

Consider the database is randomly sampled from a distribution. A set is

- Very homogeneous  $\approx$  high certainty
- Not so homogenous  $\approx$  high randomness

These can be measured with its entropy



Remaining uncertainty given the test:

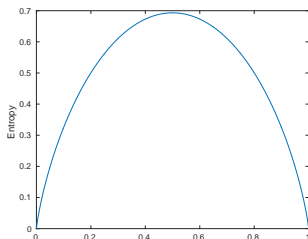
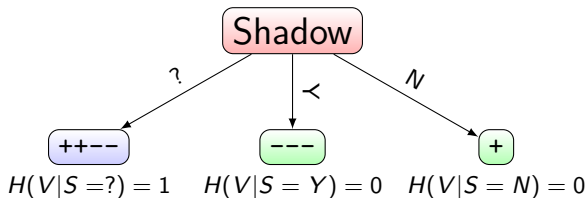
$$\frac{4}{8}H(V|S=?) + \frac{3}{8}H(V|S=Y) + \frac{1}{8}H(V|S=N) = 0.5$$

# Conditional entropy as a measure of test efficiency

Consider the database is randomly sampled from a distribution. A set is

- Very homogeneous  $\approx$  high certainty
- Not so homogenous  $\approx$  high randomness

These can be measured with its entropy



Remaining uncertainty given the test:

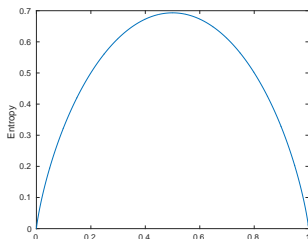
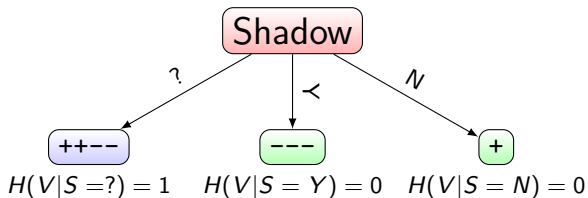
$$\begin{aligned}
 & \frac{4}{8}H(V|S = ?) + \frac{3}{8}H(V|S = Y) + \frac{1}{8}H(V|S = N) = 0.5 \\
 & = Pr(S = ?)H(V|S = ?) + Pr(S = Y)H(V|S = Y) + Pr(S = N)H(V|S = N)
 \end{aligned}$$

# Conditional entropy as a measure of test efficiency

Consider the database is randomly sampled from a distribution. A set is

- Very homogeneous  $\approx$  high certainty
- Not so homogenous  $\approx$  high randomness

These can be measured with its entropy

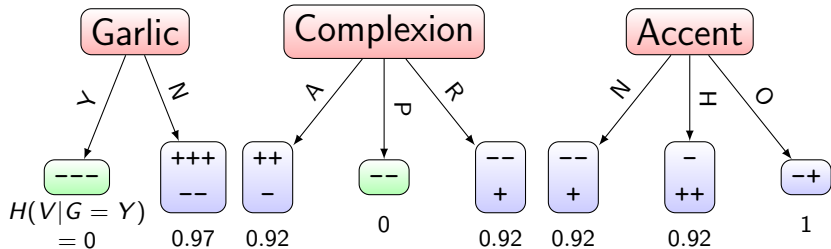


Remaining uncertainty given the test:

$$\begin{aligned}
 & \frac{4}{8}H(V|S = ?) + \frac{3}{8}H(V|S = Y) + \frac{1}{8}H(V|S = N) = 0.5 \\
 & = Pr(S = ?)H(V|S = ?) + Pr(S = Y)H(V|S = Y) + Pr(S = N)H(V|S = N) \\
 & = H(V|S)
 \end{aligned}$$

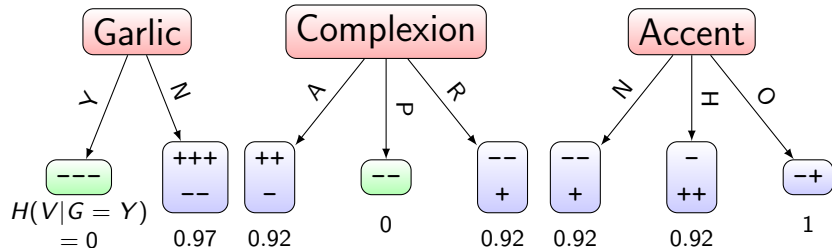


# Remaining uncertainty



$$H(V|S) = 0.5$$

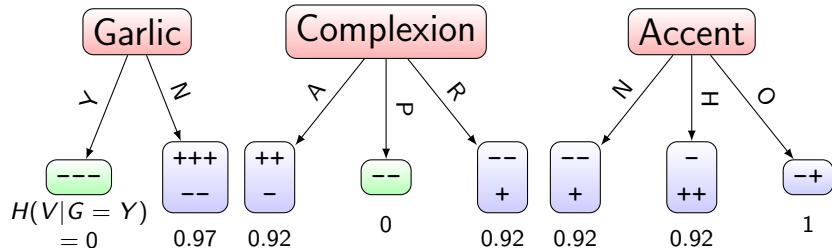
# Remaining uncertainty



$$H(V|S) = 0.5$$

$$H(V|G) = \frac{3}{8} \cdot 0 + \frac{5}{8} \cdot 0.97 = 0.61$$

# Remaining uncertainty

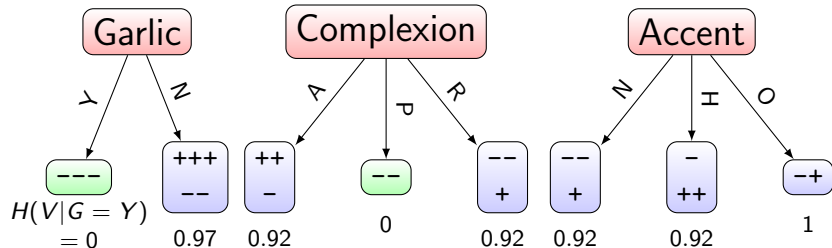


$$H(V|S) = 0.5$$

$$H(V|G) = \frac{3}{8} \cdot 0 + \frac{5}{8} \cdot 0.97 = 0.61$$

$$H(V|C) = \frac{3}{8} \cdot 0.92 + \frac{2}{8} \cdot 0 + \frac{3}{8} \cdot 0.92 = 0.69$$

# Remaining uncertainty



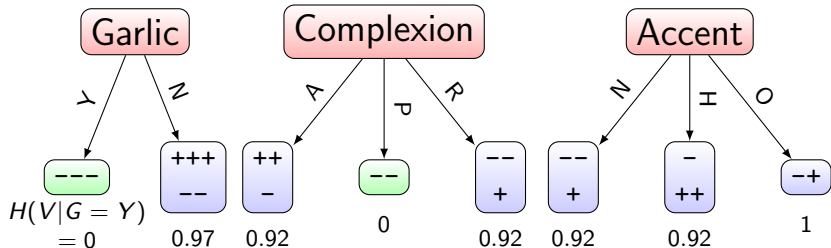
$$H(V|S) = 0.5$$

$$H(V|G) = \frac{3}{8} \cdot 0 + \frac{5}{8} \cdot 0.97 = 0.61$$

$$H(V|C) = \frac{3}{8} \cdot 0.92 + \frac{2}{8} \cdot 0 + \frac{3}{8} \cdot 0.92 = 0.69$$

$$H(V|A) = \frac{3}{8} \cdot 0.92 + \frac{3}{8} \cdot 0.92 + \frac{2}{8} \cdot 1 = 0.94$$

# Remaining uncertainty



$$H(V|S) = 0.5$$

$$H(V|G) = \frac{3}{8} \cdot 0 + \frac{5}{8} \cdot 0.97 = 0.61$$

$$H(V|C) = \frac{3}{8} \cdot 0.92 + \frac{2}{8} \cdot 0 + \frac{3}{8} \cdot 0.92 = 0.69$$

$$H(V|A) = \frac{3}{8} \cdot 0.92 + \frac{3}{8} \cdot 0.92 + \frac{2}{8} \cdot 1 = 0.94$$

$H(V|S)$  is maximum. Thus should pick test  $S$  first

# Potential extensions

- The test does not need to return discrete result. Let  $X$  be the test outcome. It can be continuous as well

# Potential extensions

- The test does not need to return discrete result. Let  $X$  be the test outcome. It can be continuous as well
  - We should just pick  $i$  such that  $H(V|X_i)$  to be as small as possible

# Potential extensions

- The test does not need to return discrete result. Let  $X$  be the test outcome. It can be continuous as well
  - We should just pick  $i$  such that  $H(V|X_i)$  to be as small as possible
  - It is equivalent of saying  $I(V; X_i) = H(V) - H(V|X_i)$  is as large as possible. This is intuitive because we want to pick the information that is most relevant (sharing most information with) to  $V$



# Potential extensions

- The test does not need to return discrete result. Let  $X$  be the test outcome. It can be continuous as well
  - We should just pick  $i$  such that  $H(V|X_i)$  to be as small as possible
  - It is equivalent of saying  $I(V; X_i) = H(V) - H(V|X_i)$  is as large as possible. This is intuitive because we want to pick the information that is most relevant (sharing most information with) to  $V$
- Build a number of trees instead of a single tree  $\Rightarrow$  random forests

# Random forests

- Pick random subset of training samples
- Train on each random subset but limited to a subset of features/attributes
- Given a test sample
  - Classify sample using each of the trees
  - Make final decision based on majority vote

# Law of Large Number (LLN)

If we randomly sample  $x_1, x_2, \dots, x_N$  from an i.i.d. (identical and independently distributed) source, the average of  $f(x_i)$  will approach the expected value as  $N \rightarrow \infty$ . That is,

$$\frac{1}{N} \sum_{i=1}^N f(x_i) = E[f(X)] \quad \text{as } N \rightarrow \infty$$

# Law of Large Number (LLN)

If we randomly sample  $x_1, x_2, \dots, x_N$  from an i.i.d. (identical and independently distributed) source, the average of  $f(x_i)$  will approach the expected value as  $N \rightarrow \infty$ . That is,

$$\frac{1}{N} \sum_{i=1}^N f(x_i) = E[f(X)] \quad \text{as } N \rightarrow \infty$$

## Example

This is precisely how poll supposes to work! Pollster randomly draws sample from a portion of the population but will expect the prediction matches the outcome

# Proof of LLN

The LLN is a rather strong result. We will only show a weak version here

$$\Pr \left( \left| \frac{1}{N} \sum_{i=1}^N f(X_i) - E[f(X)] \right| \geq a \right) \leq \frac{\text{Var}(f(X))}{Na^2} \propto \frac{1}{N}$$

# Proof of LLN

The LLN is a rather strong result. We will only show a weak version here

$$\Pr \left( \left| \frac{1}{N} \sum_{i=1}^N f(X_i) - E[f(X)] \right| \geq a \right) \leq \frac{\text{Var}(f(X))}{Na^2} \propto \frac{1}{N}$$

## Markov's Inequality

$$\Pr(X \geq b) \leq \frac{E[X]}{b} \quad \text{if } X \geq 0$$

# Proof of LLN

The LLN is a rather strong result. We will only show a weak version here

$$\Pr \left( \left| \frac{1}{N} \sum_{i=1}^N f(X_i) - E[f(X)] \right| \geq a \right) \leq \frac{\text{Var}(f(X))}{Na^2} \propto \frac{1}{N}$$

## Markov's Inequality

$$\Pr(X \geq b) \leq \frac{E[X]}{b} \quad \text{if } X \geq 0$$

Proof:

$$X = I(X \geq b) \cdot X + I(X < b) \cdot X \geq I(X \geq b) \cdot b$$

# Proof of LLN

The LLN is a rather strong result. We will only show a weak version here

$$\Pr \left( \left| \frac{1}{N} \sum_{i=1}^N f(X_i) - E[f(X)] \right| \geq a \right) \leq \frac{\text{Var}(f(X))}{Na^2} \propto \frac{1}{N}$$

## Markov's Inequality

$$\Pr(X \geq b) \leq \frac{E[X]}{b} \quad \text{if } X \geq 0$$

Proof:

$$X = I(X \geq b) \cdot X + I(X < b) \cdot X \geq I(X \geq b) \cdot b \Rightarrow E[X] \geq \Pr(X \geq b) \cdot b$$



# Proof of LLN

## Markov's Inequality

$$\Pr(X \geq b) \leq \frac{E[X]}{b} \quad \text{if } X \geq 0$$

## Chebyshev's Inequality

$$\Pr(|Y - E[Y]| \geq a) \leq \frac{\text{Var}(Y)}{a^2}$$

# Proof of LLN

## Markov's Inequality

$$\Pr(X \geq b) \leq \frac{E[X]}{b} \quad \text{if } X \geq 0$$

## Chebyshev's Inequality

$$\Pr(|Y - E[Y]| \geq a) \leq \frac{\text{Var}(Y)}{a^2}$$

Proof: Take  $X = |Y - E[Y]|^2$  and  $b = a^2$ , by Markov's Inequality

# Proof of LLN

## Markov's Inequality

$$\Pr(X \geq b) \leq \frac{E[X]}{b} \quad \text{if } X \geq 0$$

## Chebyshev's Inequality

$$\Pr(|Y - E[Y]| \geq a) \leq \frac{\text{Var}(Y)}{a^2}$$

Proof: Take  $X = |Y - E[Y]|^2$  and  $b = a^2$ , by Markov's Inequality

$$\begin{aligned} \Pr(|Y - E[Y]| \geq a) &= \Pr(|Y - E[Y]|^2 \geq a^2) \\ &\leq \frac{E[|Y - E[Y]|^2]}{a^2} \end{aligned}$$

# Proof of LLN

## Markov's Inequality

$$\Pr(X \geq b) \leq \frac{E[X]}{b} \quad \text{if } X \geq 0$$

## Chebyshev's Inequality

$$\Pr(|Y - E[Y]| \geq a) \leq \frac{\text{Var}(Y)}{a^2}$$

Proof: Take  $X = |Y - E[Y]|^2$  and  $b = a^2$ , by Markov's Inequality

$$\begin{aligned} \Pr(|Y - E[Y]| \geq a) &= \Pr(|Y - E[Y]|^2 \geq a^2) \\ &\leq \frac{E[|Y - E[Y]|^2]}{a^2} = \frac{\text{Var}(Y)}{a^2} \end{aligned}$$

# Proof of LLN

## Chebyshev's Inequality

$$\Pr(|Y - E[Y]| \geq a) \leq \frac{\text{Var}(Y)}{a^2}$$

## Proof of weak LLN

Let  $Z_N = \frac{1}{N} \sum_{i=1}^N f(X_i)$ , apparently  $E[Z_N] = E[f(X)]$  and

$$\text{Var}(Z_N) = \frac{1}{N^2} \sum_{i=1}^N \text{Var}(f(X)) = \frac{\text{Var}(f(X))}{N}$$

By Chebyshev's Inequality,

# Proof of LLN

## Chebyshev's Inequality

$$\Pr(|Y - E[Y]| \geq a) \leq \frac{\text{Var}(Y)}{a^2}$$

## Proof of weak LLN

Let  $Z_N = \frac{1}{N} \sum_{i=1}^N f(X_i)$ , apparently  $E[Z_N] = E[f(X)]$  and

$$\text{Var}(Z_N) = \frac{1}{N^2} \sum_{i=1}^N \text{Var}(f(X)) = \frac{\text{Var}(f(X))}{N}$$

By Chebyshev's Inequality,

$$\begin{aligned} & \Pr\left(\left|\frac{1}{N} \sum_{i=1}^N f(X_i) - E[f(X)]\right| \geq a\right) \\ &= \Pr(|Z_N - E[Z_N]| \geq a) \leq \frac{\text{Var}(Z_N)}{a^2} \end{aligned}$$

# Proof of LLN

## Chebyshev's Inequality

$$\Pr(|Y - E[Y]| \geq a) \leq \frac{\text{Var}(Y)}{a^2}$$

## Proof of weak LLN

Let  $Z_N = \frac{1}{N} \sum_{i=1}^N f(X_i)$ , apparently  $E[Z_N] = E[f(X)]$  and

$$\text{Var}(Z_N) = \frac{1}{N^2} \sum_{i=1}^N \text{Var}(f(X)) = \frac{\text{Var}(f(X))}{N}$$

By Chebyshev's Inequality,

$$\begin{aligned} & \Pr\left(\left|\frac{1}{N} \sum_{i=1}^N f(X_i) - E[f(X)]\right| \geq a\right) \\ &= \Pr(|Z_N - E[Z_N]| \geq a) \leq \frac{\text{Var}(Z_N)}{a^2} = \frac{\text{Var}(f(X))}{Na^2} \end{aligned}$$

# Main idea

Consider a sequence of symbols  $x_1, x_2, \dots, x_N$  sampled from a DMS and consider the sample average of the log-probabilities of each sampled symbols

$$\frac{1}{N} \sum_{i=1}^N \log \frac{1}{p(x_i)} \rightarrow E \left[ \log \frac{1}{p(X)} \right]$$

by LLN.



# Main idea

Consider a sequence of symbols  $x_1, x_2, \dots, x_N$  sampled from a DMS and consider the sample average of the log-probabilities of each sampled symbols

$$\frac{1}{N} \sum_{i=1}^N \log \frac{1}{p(x_i)} \rightarrow E \left[ \log \frac{1}{p(X)} \right] = H(X)$$

by LLN.

# Main idea

Consider a sequence of symbols  $x_1, x_2, \dots, x_N$  sampled from a DMS and consider the sample average of the log-probabilities of each sampled symbols

$$\frac{1}{N} \sum_{i=1}^N \log \frac{1}{p(x_i)} \rightarrow E \left[ \log \frac{1}{p(X)} \right] = H(X)$$

by LLN. But for the LHS,

$$\frac{1}{N} \sum_{i=1}^N \log \frac{1}{p(x_i)} = \frac{1}{N} \log \frac{1}{\prod_{i=1}^N p(x_i)} = -\frac{1}{N} \log p(x^N),$$

where  $x^N = x_1, x_2, \dots, x_N$

# Main idea

Consider a sequence of symbols  $x_1, x_2, \dots, x_N$  sampled from a DMS and consider the sample average of the log-probabilities of each sampled symbols

$$\frac{1}{N} \sum_{i=1}^N \log \frac{1}{p(x_i)} \rightarrow E \left[ \log \frac{1}{p(X)} \right] = H(X)$$

by LLN. But for the LHS,

$$\frac{1}{N} \sum_{i=1}^N \log \frac{1}{p(x_i)} = \frac{1}{N} \log \frac{1}{\prod_{i=1}^N p(x_i)} = -\frac{1}{N} \log p(x^N),$$

where  $x^N = x_1, x_2, \dots, x_N$

Rearranging the terms, this implies that for any sequence sampled from the source, the probability of the sampled sequence  $p(x^N) \rightarrow 2^{-NH(X)}$ !

# Set of typical sequences

Let's name the sequence  $x^N$  with  $p(x^N) \sim 2^{-NH(X)}$  typical and define the set of typical sequences

$$\mathcal{A}_\epsilon^N(X) = \{x^N | 2^{-N(H(X)+\epsilon)} \leq p(x^N) \leq 2^{-N(H(X)-\epsilon)}\}$$

# Set of typical sequences

Let's name the sequence  $x^N$  with  $p(x^N) \sim 2^{-NH(X)}$  typical and define the set of typical sequences

$$\mathcal{A}_\epsilon^N(X) = \{x^N | 2^{-N(H(X)+\epsilon)} \leq p(x^N) \leq 2^{-N(H(X)-\epsilon)}\}$$

- For any  $\epsilon > 0$ , we can find a sufficiently large  $N$  such that any sampled sequence from the source is typical

# Set of typical sequences

Let's name the sequence  $x^N$  with  $p(x^N) \sim 2^{-NH(X)}$  typical and define the set of typical sequences

$$\mathcal{A}_\epsilon^N(X) = \{x^N | 2^{-N(H(X)+\epsilon)} \leq p(x^N) \leq 2^{-N(H(X)-\epsilon)}\}$$

- For any  $\epsilon > 0$ , we can find a sufficiently large  $N$  such that any sampled sequence from the source is typical
- Since all typical sequences have probability  $\sim 2^{-NH(X)}$  and they fill up the entire probability space (everything is typical), there should be approximately  $\frac{1}{2^{-NH(X)}} = 2^{NH(X)}$  typical sequences

# Precise bounds on the size of typical set

$$(1 - \delta)2^{N(H(X) - \epsilon)} \leq |\mathcal{A}_\epsilon^N(X)| \leq 2^{N(H(X) + \epsilon)}$$

$$1 \geq \Pr(X^N \in \mathcal{A}_\epsilon^N(X))$$

# Precise bounds on the size of typical set

$$(1 - \delta)2^{N(H(X) - \epsilon)} \leq |\mathcal{A}_\epsilon^N(X)| \leq 2^{N(H(X) + \epsilon)}$$

$$1 \geq \Pr(X^N \in \mathcal{A}_\epsilon^N(X)) = \sum_{x^N \in \mathcal{A}_\epsilon^N(X)} p(x^N)$$



# Precise bounds on the size of typical set

$$(1 - \delta)2^{N(H(X) - \epsilon)} \leq |\mathcal{A}_\epsilon^N(X)| \leq 2^{N(H(X) + \epsilon)}$$

$$1 \geq \Pr(X^N \in \mathcal{A}_\epsilon^N(X)) = \sum_{x^N \in \mathcal{A}_\epsilon^N(X)} p(x^N) \geq \sum_{x^N \in \mathcal{A}_\epsilon^N(X)} 2^{-N(H(X) + \epsilon)}$$

# Precise bounds on the size of typical set

$$(1 - \delta)2^{N(H(X)-\epsilon)} \leq |\mathcal{A}_\epsilon^N(X)| \leq 2^{N(H(X)+\epsilon)}$$

$$\begin{aligned} 1 &\geq \Pr(X^N \in \mathcal{A}_\epsilon^N(X)) = \sum_{x^N \in \mathcal{A}_\epsilon^N(X)} p(x^N) \geq \sum_{x^N \in \mathcal{A}_\epsilon^N(X)} 2^{-N(H(X)+\epsilon)} \\ &= |\mathcal{A}_\epsilon^N(X)| 2^{-N(H(X)+\epsilon)} \end{aligned}$$

# Precise bounds on the size of typical set

$$(1 - \delta)2^{N(H(X)-\epsilon)} \leq |\mathcal{A}_\epsilon^N(X)| \leq 2^{N(H(X)+\epsilon)}$$

$$\begin{aligned} 1 &\geq \Pr(X^N \in \mathcal{A}_\epsilon^N(X)) = \sum_{x^N \in \mathcal{A}_\epsilon^N(X)} p(x^N) \geq \sum_{x^N \in \mathcal{A}_\epsilon^N(X)} 2^{-N(H(X)+\epsilon)} \\ &= |\mathcal{A}_\epsilon^N(X)| 2^{-N(H(X)+\epsilon)} \end{aligned}$$

For a sufficiently large  $N$ , we have

$$1 - \delta \leq \Pr(X^N \in \mathcal{A}_\epsilon^N(X))$$

# Precise bounds on the size of typical set

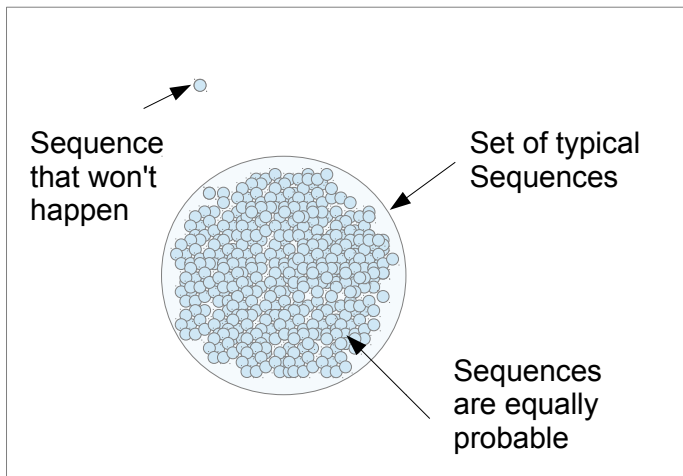
$$(1 - \delta)2^{N(H(X)-\epsilon)} \leq |\mathcal{A}_\epsilon^N(X)| \leq 2^{N(H(X)+\epsilon)}$$

$$\begin{aligned} 1 &\geq \Pr(X^N \in \mathcal{A}_\epsilon^N(X)) = \sum_{x^N \in \mathcal{A}_\epsilon^N(X)} p(x^N) \geq \sum_{x^N \in \mathcal{A}_\epsilon^N(X)} 2^{-N(H(X)+\epsilon)} \\ &= |\mathcal{A}_\epsilon^N(X)| 2^{-N(H(X)+\epsilon)} \end{aligned}$$

For a sufficiently large  $N$ , we have

$$\begin{aligned} 1 - \delta &\leq \Pr(X^N \in \mathcal{A}_\epsilon^N(X)) = \sum_{x^N \in \mathcal{A}_\epsilon^N(X)} p(x^N) \leq \sum_{x^N \in \mathcal{A}_\epsilon^N(X)} 2^{-N(H(X)-\epsilon)} \\ &= |\mathcal{A}_\epsilon^N(X)| 2^{-N(H(X)-\epsilon)} \end{aligned}$$

## AEP



Asymptotic equipartition refers to the fact that the probability space is equally partitioned by the typical sequences

# AEP and compression limit

Consider coin flipping again, let say  $Pr(\text{Head}) = 0.3$  and  $N = 1000$

# AEP and compression limit

Consider coin flipping again, let say  $Pr(\text{Head}) = 0.3$  and  $N = 1000$

- The typical sequences will be those with approximately 300 heads and 700 tails

# AEP and compression limit

Consider coin flipping again, let say  $Pr(\text{Head}) = 0.3$  and  $N = 1000$

- The typical sequences will be those with approximately 300 heads and 700 tails
- AEP (LLN) tells us that it is almost impossible to get, say, a sequence of 100 heads and 900 tails



# AEP and compression limit

Consider coin flipping again, let say  $Pr(\text{Head}) = 0.3$  and  $N = 1000$

- The typical sequences will be those with approximately 300 heads and 700 tails
- AEP (LLN) tells us that it is almost impossible to get, say, a sequence of 100 heads and 900 tails
- AEP also tells us that the number of typical sequences are approximately  $2^{NH(X)}$

# AEP and compression limit

Consider coin flipping again, let say  $Pr(\text{Head}) = 0.3$  and  $N = 1000$

- The typical sequences will be those with approximately 300 heads and 700 tails
- AEP (LLN) tells us that it is almost impossible to get, say, a sequence of 100 heads and 900 tails
- AEP also tells us that the number of typical sequences are approximately  $2^{NH(X)}$
- Therefore, we can simply assign index to all the typical sequences and ignore the rest. Then we only need  $\log 2^{NH(X)} = NH(X)$  to store a sequence of  $N$  symbols. And on average, we need  $H(X)$  bits per symbol as before!

# Previously...

- Identification/Decision trees
- Random forests
- Law of Large Number
- Asymptotic equipartition (AEP) and typical sequences

# This time

- Joint typical sequences
- Covering and Packing Lemmas
- Channel coding setup
- Channel coding rate
- Channel capacity
- Channel Coding Theorem

# Jointly typical sequences

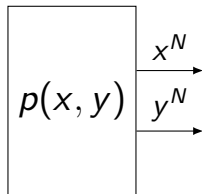
For a pair of sequences  $x^N$  and  $y^N$ , we say that they are jointly typical if

$$2^{-N(H(X,Y)+\epsilon)} \leq p(x^N, y^N) \leq 2^{-N(H(X,Y)-\epsilon)}$$

and  $x^N$  and  $y^N$  themselves are typical

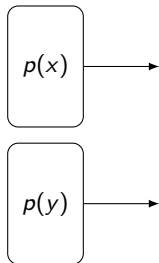
As in the single sequence case,

- Any sequence pair drawing from a joint source  $p(x, y)$  is essentially jointly typical
- There are  $\sim 2^{NH(X,Y)}$  jointly typical sequences



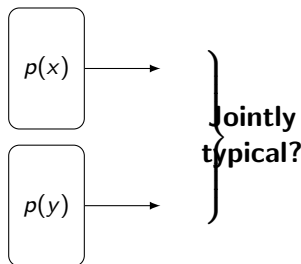
# Joint typicality of independent sequences

- Given sequences  $X^N$  and  $Y^N$  independently drawn from discrete memoryless sources  $p(x)$  and  $p(y)$



# Joint typicality of independent sequences

- Given sequences  $X^N$  and  $Y^N$  independently drawn from discrete memoryless sources  $p(x)$  and  $p(y)$
- What is the probability that  $X^N$  and  $Y^N$  are jointly typical?

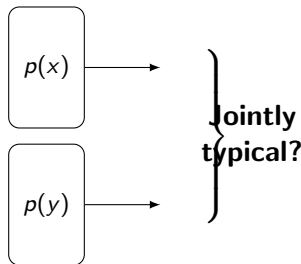


# Joint typicality of independent sequences

- Given sequences  $X^N$  and  $Y^N$  independently drawn from discrete memoryless sources  $p(x)$  and  $p(y)$
- What is the probability that  $X^N$  and  $Y^N$  are jointly typical?

$$Pr((X^N, Y^N) \in \mathcal{A}_\epsilon^{(N)})$$

$$= \sum_{\{(x^N, y^N) | (x^N, y^N) \in \mathcal{A}_\epsilon^{(N)}\}} p(x^N, y^N)$$

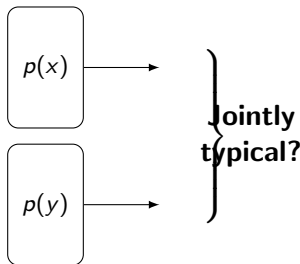




# Joint typicality of independent sequences

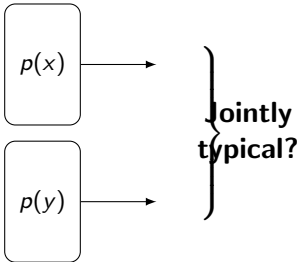
- Given sequences  $X^N$  and  $Y^N$  independently drawn from discrete memoryless sources  $p(x)$  and  $p(y)$
- What is the probability that  $X^N$  and  $Y^N$  are jointly typical?

$$\begin{aligned}
 & Pr((X^N, Y^N) \in \mathcal{A}_\epsilon^{(N)}) \\
 = & \sum_{\{(x^N, y^N) | (x^N, y^N) \in \mathcal{A}_\epsilon^{(N)}\}} p(x^N, y^N) \\
 = & \sum_{\{(x^N, y^N) | (x^N, y^N) \in \mathcal{A}_\epsilon^{(N)}\}} p(x^N) p(y^N)
 \end{aligned}$$



# Joint typicality of independent sequences

- Given sequences  $X^N$  and  $Y^N$  independently drawn from discrete memoryless sources  $p(x)$  and  $p(y)$
- What is the probability that  $X^N$  and  $Y^N$  are jointly typical?

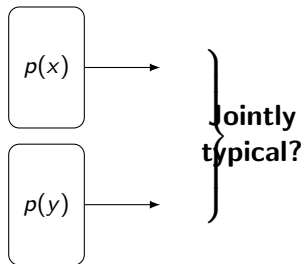
$$\begin{aligned}
 & Pr((X^N, Y^N) \in \mathcal{A}_\epsilon^{(N)}) \\
 = & \sum_{\{(x^N, y^N) | (x^N, y^N) \in \mathcal{A}_\epsilon^{(N)}\}} p(x^N, y^N) \\
 = & \sum_{\{(x^N, y^N) | (x^N, y^N) \in \mathcal{A}_\epsilon^{(N)}\}} p(x^N) p(y^N) \\
 \leq & \sum_{\{(x^N, y^N) | (x^N, y^N) \in \mathcal{A}_\epsilon^{(N)}\}} 2^{-N(H(X)-\epsilon)} 2^{-N(H(Y)-\epsilon)}
 \end{aligned}$$


The diagram illustrates two independent discrete memoryless sources,  $p(x)$  and  $p(y)$ , each represented by a rounded rectangular box. Arrows from both boxes point towards a large right-facing curly bracket on the right side of the slide. The text "Jointly typical?" is written vertically inside this bracket, indicating the question of whether the outputs from these two sources are jointly typical.

# Joint typicality of independent sequences

- Given sequences  $X^N$  and  $Y^N$  independently drawn from discrete memoryless sources  $p(x)$  and  $p(y)$
- What is the probability that  $X^N$  and  $Y^N$  are jointly typical?

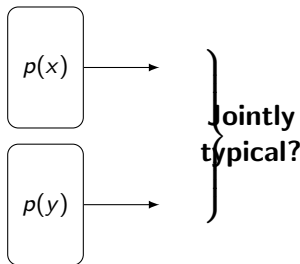
$$\begin{aligned}
 & Pr((X^N, Y^N) \in \mathcal{A}_\epsilon^{(N)}) \\
 = & \sum_{\{(x^N, y^N) | (x^N, y^N) \in \mathcal{A}_\epsilon^{(N)}\}} p(x^N, y^N) \\
 = & \sum_{\{(x^N, y^N) | (x^N, y^N) \in \mathcal{A}_\epsilon^{(N)}\}} p(x^N) p(y^N) \\
 \leq & \sum_{\{(x^N, y^N) | (x^N, y^N) \in \mathcal{A}_\epsilon^{(N)}\}} 2^{-N(H(X)-\epsilon)} 2^{-N(H(Y)-\epsilon)} \\
 \leq & 2^{-N(I(X; Y)-3\epsilon)}
 \end{aligned}$$



# Joint typicality of independent sequences

- Given sequences  $X^N$  and  $Y^N$  independently drawn from discrete memoryless sources  $p(x)$  and  $p(y)$
- What is the probability that  $X^N$  and  $Y^N$  are jointly typical?

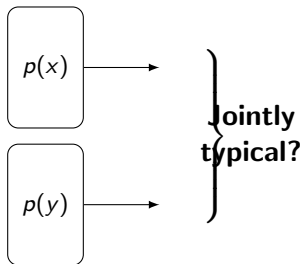
$$\begin{aligned}
 & Pr((X^N, Y^N) \in \mathcal{A}_\epsilon^{(N)}) \\
 = & \sum_{\{(x^N, y^N) | (x^N, y^N) \in \mathcal{A}_\epsilon^{(N)}\}} p(x^N, y^N) \\
 = & \sum_{\{(x^N, y^N) | (x^N, y^N) \in \mathcal{A}_\epsilon^{(N)}\}} p(x^N) p(y^N)
 \end{aligned}$$



# Joint typicality of independent sequences

- Given sequences  $X^N$  and  $Y^N$  independently drawn from discrete memoryless sources  $p(x)$  and  $p(y)$
- What is the probability that  $X^N$  and  $Y^N$  are jointly typical?

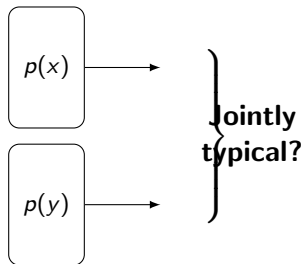
$$\begin{aligned}
 & Pr((X^N, Y^N) \in \mathcal{A}_\epsilon^{(N)}) \\
 = & \sum_{\{(x^N, y^N) | (x^N, y^N) \in \mathcal{A}_\epsilon^{(N)}\}} p(x^N, y^N) \\
 = & \sum_{\{(x^N, y^N) | (x^N, y^N) \in \mathcal{A}_\epsilon^{(N)}\}} p(x^N) p(y^N) \\
 \geq & \sum_{\{(x^N, y^N) | (x^N, y^N) \in \mathcal{A}_\epsilon^{(N)}\}} 2^{-N(H(X)+\epsilon)} 2^{-N(H(Y)+\epsilon)}
 \end{aligned}$$



# Joint typicality of independent sequences

- Given sequences  $X^N$  and  $Y^N$  independently drawn from discrete memoryless sources  $p(x)$  and  $p(y)$
- What is the probability that  $X^N$  and  $Y^N$  are jointly typical?

$$\begin{aligned}
 & Pr((X^N, Y^N) \in \mathcal{A}_\epsilon^{(N)}) \\
 = & \sum_{\{(x^N, y^N) | (x^N, y^N) \in \mathcal{A}_\epsilon^{(N)}\}} p(x^N, y^N) \\
 = & \sum_{\{(x^N, y^N) | (x^N, y^N) \in \mathcal{A}_\epsilon^{(N)}\}} p(x^N)p(y^N) \\
 \geq & \sum_{\{(x^N, y^N) | (x^N, y^N) \in \mathcal{A}_\epsilon^{(N)}\}} 2^{-N(H(X)+\epsilon)} 2^{-N(H(Y)+\epsilon)} \\
 \geq & (1 - \delta) 2^{-N(I(X; Y) + 3\epsilon)}
 \end{aligned}$$



# Packing lemma

- Instead of drawing one  $Y^N$  sequences, let us draw  $M$  of them

# Packing lemma

- Instead of drawing one  $Y^N$  sequences, let us draw  $M$  of them
- The probability of any of the sequence to be jointly typical with  $X^N$  is bounded by

$$Pr(\text{Any one of } M Y^N \text{ jointly typical with } X^N)$$



# Packing lemma

- Instead of drawing one  $Y^N$  sequences, let us draw  $M$  of them
- The probability of any of the sequence to be jointly typical with  $X^N$  is bounded by

$$\begin{aligned} & Pr(\text{Any one of } M \text{ } Y^N \text{ jointly typical with } X^N) \\ & \leq M Pr((X^N, Y^N) \in \mathcal{A}_\epsilon^N(X, Y)) \\ & \leq M 2^{-N(I(X; Y) - 3\epsilon)} \end{aligned}$$

# Packing lemma

- Instead of drawing one  $Y^N$  sequences, let us draw  $M$  of them
- The probability of any of the sequence to be jointly typical with  $X^N$  is bounded by

$$\begin{aligned}
 & Pr(\text{Any one of } M \text{ } Y^N \text{ jointly typical with } X^N) \\
 & \leq M Pr((X^N, Y^N) \in \mathcal{A}_\epsilon^N(X, Y)) \\
 & \leq M 2^{-N(I(X; Y) - 3\epsilon)} \\
 & \leq 2^{-N(I(X; Y) - R - 3\epsilon)}
 \end{aligned}$$

where  $2^{NR} = M$

# Packing lemma

- Instead of drawing one  $Y^N$  sequences, let us draw  $M$  of them
- The probability of any of the sequence to be jointly typical with  $X^N$  is bounded by

$$\begin{aligned}
 & Pr(\text{Any one of } M \text{ } Y^N \text{ jointly typical with } X^N) \\
 & \leq M Pr((X^N, Y^N) \in \mathcal{A}_\epsilon^N(X, Y)) \\
 & \leq M 2^{-N(I(X; Y) - 3\epsilon)} \\
 & \leq 2^{-N(I(X; Y) - R - 3\epsilon)} \rightarrow 0 \text{ as } N \rightarrow \infty \text{ and } I(X; Y) - 3\epsilon > R,
 \end{aligned}$$

where  $2^{NR} = M$

# Packing lemma

- Instead of drawing one  $Y^N$  sequences, let us draw  $M$  of them
- The probability of any of the sequence to be jointly typical with  $X^N$  is bounded by

$$\begin{aligned}
 & Pr(\text{Any one of } M \ Y^N \text{ jointly typical with } X^N) \\
 & \leq M Pr((X^N, Y^N) \in \mathcal{A}_\epsilon^N(X, Y)) \\
 & \leq M 2^{-N(I(X; Y) - 3\epsilon)} \\
 & \leq 2^{-N(I(X; Y) - R - 3\epsilon)} \rightarrow 0 \text{ as } N \rightarrow \infty \text{ and } I(X; Y) - 3\epsilon > R,
 \end{aligned}$$

where  $2^{NR} = M$

Since  $\epsilon$  can be made arbitrarily small as  $N$  increases, as long as  $I(X; Y) > R$ , we can find a sufficiently large  $N$  so that we can “pack” the  $M$   $Y^N$  with  $X^N$  and none of the  $Y^N$  will be jointly typical with  $X^N$

# Covering lemma

- Again, draw  $M(= 2^{NR})$   $Y^N$  sequences
- Under what condition that *at least one*  $Y^N$  jointly typical with  $X^N$ ?

# Covering lemma

- Again, draw  $M(= 2^{NR})$   $Y^N$  sequences
- Under what condition that *at least one*  $Y^N$  jointly typical with  $X^N$ ?

$$Pr((X^N(m), Y^N) \notin \mathcal{A}_\epsilon^{(N)}(X, Y) \text{ for all } m)$$

# Covering lemma

- Again, draw  $M(= 2^{NR})$   $Y^N$  sequences
- Under what condition that *at least one*  $Y^N$  jointly typical with  $X^N$ ?

$$Pr((X^N(m), Y^N) \notin \mathcal{A}_\epsilon^{(N)}(X, Y) \text{ for all } m)$$

$$= \prod_{m=1}^M Pr((X^N(m), Y^N) \notin \mathcal{A}_\epsilon^{(N)}(Y, X))$$

# Covering lemma

- Again, draw  $M(= 2^{NR})$   $Y^N$  sequences
- Under what condition that *at least one*  $Y^N$  jointly typical with  $X^N$ ?

$$\begin{aligned}
 & Pr((X^N(m), Y^N) \notin \mathcal{A}_\epsilon^{(N)}(X, Y) \text{ for all } m) \\
 &= \prod_{m=1}^M Pr((X^N(m), Y^N) \notin \mathcal{A}_\epsilon^{(N)}(Y, X)) \\
 &= \prod_{m=1}^M [1 - Pr((X^N(m), Y^N) \in \mathcal{A}_\epsilon^{(N)}(Y, X))]
 \end{aligned}$$



# Covering lemma

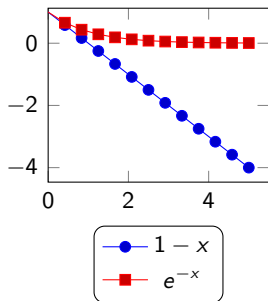
- Again, draw  $M(= 2^{NR})$   $Y^N$  sequences
- Under what condition that *at least one*  $Y^N$  jointly typical with  $X^N$ ?

$$\begin{aligned}
 & Pr((X^N(m), Y^N) \notin \mathcal{A}_\epsilon^{(N)}(X, Y) \text{ for all } m) \\
 &= \prod_{m=1}^M Pr((X^N(m), Y^N) \notin \mathcal{A}_\epsilon^{(N)}(Y, X)) \\
 &= \prod_{m=1}^M [1 - Pr((X^N(m), Y^N) \in \mathcal{A}_\epsilon^{(N)}(Y, X))] \\
 &\leq (1 - (1 - \delta)2^{-N(I(Y;X)+3\epsilon)})^M
 \end{aligned}$$

# Covering lemma

- Again, draw  $M(= 2^{NR})$   $Y^N$  sequences
- Under what condition that *at least one*  $Y^N$  jointly typical with  $X^N$ ?

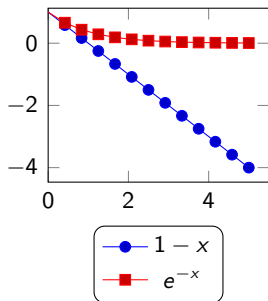
$$\begin{aligned}
 & Pr((X^N(m), Y^N) \notin \mathcal{A}_\epsilon^{(N)}(X, Y) \text{ for all } m) \\
 &= \prod_{m=1}^M Pr((X^N(m), Y^N) \notin \mathcal{A}_\epsilon^{(N)}(Y, X)) \\
 &= \prod_{m=1}^M [1 - Pr((X^N(m), Y^N) \in \mathcal{A}_\epsilon^{(N)}(Y, X))] \\
 &\leq (1 - (1 - \delta)2^{-N(I(Y;X)+3\epsilon)})^M \\
 &\leq \exp(-M(1 - \delta)2^{-N(I(Y;X)+3\epsilon)})
 \end{aligned}$$



# Covering lemma

- Again, draw  $M(= 2^{NR})$   $Y^N$  sequences
- Under what condition that *at least one*  $Y^N$  jointly typical with  $X^N$ ?

$$\begin{aligned}
 & Pr((X^N(m), Y^N) \notin \mathcal{A}_\epsilon^{(N)}(X, Y) \text{ for all } m) \\
 &= \prod_{m=1}^M Pr((X^N(m), Y^N) \notin \mathcal{A}_\epsilon^{(N)}(Y, X)) \\
 &= \prod_{m=1}^M \left[ 1 - Pr((X^N(m), Y^N) \in \mathcal{A}_\epsilon^{(N)}(Y, X)) \right] \\
 &\leq (1 - (1 - \delta)2^{-N(I(Y;X)+3\epsilon)})^M \\
 &\leq \exp(-M(1 - \delta)2^{-N(I(Y;X)+3\epsilon)}) \\
 &\leq \exp(-(1 - \delta)2^{-N(I(Y;X)-R+3\epsilon)}) \rightarrow 0 \text{ as } N \rightarrow \infty \text{ and } R > I(X; Y) + 3\epsilon
 \end{aligned}$$



# Summary of packing lemma and covering lemma

## Packing Lemma

We can “pack”  $M = 2^{NR}$  (with  $R < I(X; Y)$ )  $x^N$  together without being jointly typical with  $y^N$

## Covering Lemma

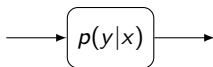
We can “cover” with  $M = 2^{NR}$  (with  $R > I(X; Y)$ )  $x^N$  such that at least one  $x^N$  being jointly typical with  $y^N$

## Remark

- Packing lemma is useful in the proof of channel coding theorem
- Covering lemma is useful in the proof of rate-distortion theorem

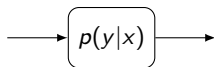
We will look into the above applications later in this course

# Channel coding setup



- As the name suggests, the output of a discrete memoryless channel (DMS) only depends on the current input (thus no memoryless). And both its input  $X$  and output  $Y$  are characterized by the conditional probability  $p(y|x)$

# Channel coding setup



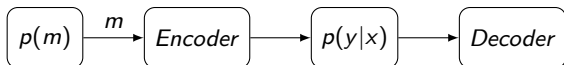
- As the name suggests, the output of a discrete memoryless channel (DMS) only depends on the current input (thus no memoryless). And both its input  $X$  and output  $Y$  are characterized by the conditional probability  $p(y|x)$
- Given an input sequence  $x^N = x_1, \dots, x_N$ , the probability of getting an output sequence  $y^N = y_1, \dots, y_N$  is  $p(y^N|x^N) = \prod_{i=1}^N p(y_i|x_i)$

# Channel coding setup



- As the name suggests, the output of a discrete memoryless channel (DMS) only depends on the current input (thus no memoryless). And both its input  $X$  and output  $Y$  are characterized by the conditional probability  $p(y|x)$
- Given an input sequence  $x^N = x_1, \dots, x_N$ , the probability of getting an output sequence  $y^N = y_1, \dots, y_N$  is  $p(y^N|x^N) = \prod_{i=1}^N p(y_i|x_i)$
- Given a message  $m$  (say generated from a distribution  $p(m)$ )

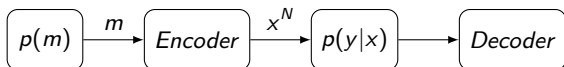
# Channel coding setup



- As the name suggests, the output of a discrete memoryless channel (DMS) only depends on the current input (thus no memoryless). And both its input  $X$  and output  $Y$  are characterized by the conditional probability  $p(y|x)$
- Given an input sequence  $x^N = x_1, \dots, x_N$ , the probability of getting an output sequence  $y^N = y_1, \dots, y_N$  is  $p(y^N|x^N) = \prod_{i=1}^N p(y_i|x_i)$
- Given a message  $m$  (say generated from a distribution  $p(m)$ )
  - We will have an encoder decoder pair

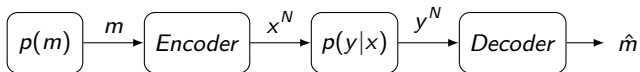


# Channel coding setup



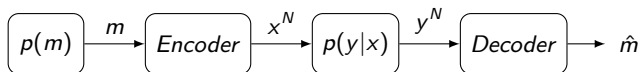
- As the name suggests, the output of a discrete memoryless channel (DMS) only depends on the current input (thus no memoryless). And both its input  $X$  and output  $Y$  are characterized by the conditional probability  $p(y|x)$
- Given an input sequence  $x^N = x_1, \dots, x_N$ , the probability of getting an output sequence  $y^N = y_1, \dots, y_N$  is  $p(y^N|x^N) = \prod_{i=1}^N p(y_i|x_i)$
- Given a message  $m$  (say generated from a distribution  $p(m)$ )
  - We will have an encoder decoder pair
  - The encoder will convert  $m$  to  $x^N$  suitable for transmission

# Channel coding setup



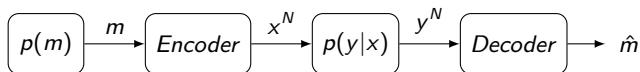
- As the name suggests, the output of a discrete memoryless channel (DMS) only depends on the current input (thus no memoryless). And both its input  $X$  and output  $Y$  are characterized by the conditional probability  $p(y|x)$
- Given an input sequence  $x^N = x_1, \dots, x_N$ , the probability of getting an output sequence  $y^N = y_1, \dots, y_N$  is  $p(y^N|x^N) = \prod_{i=1}^N p(y_i|x_i)$
- Given a message  $m$  (say generated from a distribution  $p(m)$ )
  - We will have an encoder decoder pair
  - The encoder will convert  $m$  to  $x^N$  suitable for transmission
  - Decoder will try to extract the message from the channel output  $y^N$

# Channel coding rate



The channel coding rate is defined as number of bits of message can be sent per channel use

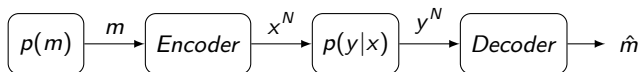
# Channel coding rate



The channel coding rate is defined as number of bits of message can be sent per channel use

- Since there is  $H(M)$  bits of information for each message  $M$  sent

# Channel coding rate



The channel coding rate is defined as number of bits of message can be sent per channel use

- Since there is  $H(M)$  bits of information for each message  $M$  sent
- $R = \frac{H(M)}{N}$

# Channel capacity

- By Shannon's channel coding theorem, the capacity of the channel (will be shown later) is given by

$$C = \max_{p(x)} I(X; Y)$$

# Channel capacity

- By Shannon's channel coding theorem, the capacity of the channel (will be shown later) is given by

$$C = \max_{p(x)} I(X; Y)$$

- This means that as long as the rate  $R$  is less than the capacity  $C$ , we can find encoder-decoder pair such that the decoding error ( $Pr(\hat{M} \neq M)$ ) can be made arbitrarily small

# Channel capacity

- By Shannon's channel coding theorem, the capacity of the channel (will be shown later) is given by

$$C = \max_{p(x)} I(X; Y)$$

- This means that as long as the rate  $R$  is less than the capacity  $C$ , we can find encoder-decoder pair such that the decoding error ( $Pr(\hat{M} \neq M)$ ) can be made arbitrarily small
- On the other hand, if  $R$  is larger than the capacity  $C$ , no matter how we try, it is impossible to reconstruct  $m$  error free



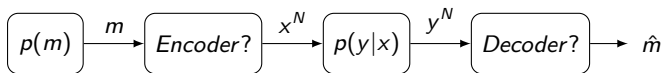
# Channel capacity

- By Shannon's channel coding theorem, the capacity of the channel (will be shown later) is given by

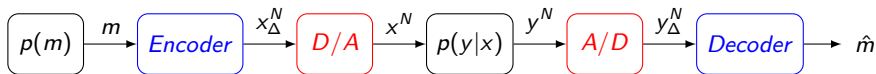
$$C = \max_{p(x)} I(X; Y)$$

- This means that as long as the rate  $R$  is less than the capacity  $C$ , we can find encoder-decoder pair such that the decoding error ( $Pr(\hat{M} \neq M)$ ) can be made arbitrarily small
- On the other hand, if  $R$  is larger than the capacity  $C$ , no matter how we try, it is impossible to reconstruct  $m$  error free
- An intuitive interpretation is that the amount of information can be passed through a channel is just mutual information between the input and output. And since we can pick the statistics of our input, we may make our choice wisely and maximize the mutual information. And the maximum that we can attain is the capacity

# Continuous channel

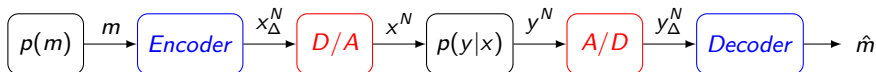


# Continuous channel



- For continuous channel, we can create a “pseudo” discrete channel using A/D and D/A converters

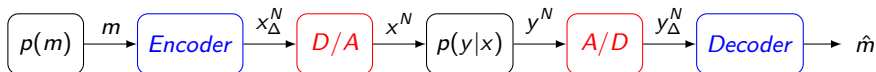
# Continuous channel



- For continuous channel, we can create a “pseudo” discrete channel using A/D and D/A converters
- The maximum information that can pass through the channel will then be

$$\begin{aligned}
 C_{\Delta} &= \max_{p(x)} I(X_{\Delta}; Y_{\Delta}) = \max_{p(x)} H(Y_{\Delta}) - H(Y_{\Delta}|X_{\Delta}) \\
 &\approx \max_{p(x)} h(Y) - \log \Delta - h(Y|X_{\Delta}) + \log \Delta \\
 &\approx \max_{p(x)} h(Y) - h(Y|X) = \max_{p(x)} I(X; Y)
 \end{aligned}$$

# Continuous channel



- For continuous channel, we can create a “pseudo” discrete channel using A/D and D/A converters
- The maximum information that can pass through the channel will then be

$$\begin{aligned}
 C_{\Delta} &= \max_{p(x)} I(X_{\Delta}; Y_{\Delta}) = \max_{p(x)} H(Y_{\Delta}) - H(Y_{\Delta}|X_{\Delta}) \\
 &\approx \max_{p(x)} h(Y) - \log \Delta - h(Y|X_{\Delta}) + \log \Delta \\
 &\approx \max_{p(x)} h(Y) - h(Y|X) = \max_{p(x)} I(X; Y)
 \end{aligned}$$

- As  $\Delta \rightarrow 0$ ,  $C = \max_{p(x)} I(X; Y)$ . So expression is completely the same as the discrete case

# Example: Binary symmetric channel

- Both input and output are binary (say take value 0 or 1)

# Example: Binary symmetric channel

- Both input and output are binary (say take value 0 or 1)
- The channel is symmetric in the sense that

$$p_{Y|X}(1|0) = p_{Y|X}(0|1) = p$$

and

$$p_{Y|X}(0|0) = p_{Y|X}(1|1) = 1 - p,$$

where  $p$  is known to be the cross-over probability

# Example: Binary symmetric channel

- Both input and output are binary (say take value 0 or 1)
- The channel is symmetric in the sense that

$$p_{Y|X}(1|0) = p_{Y|X}(0|1) = p$$

and

$$p_{Y|X}(0|0) = p_{Y|X}(1|1) = 1 - p,$$

where  $p$  is known to be the cross-over probability

- Capacity is given by

$$C = \max_{p(x)} I(X; Y)$$



# Example: Binary symmetric channel

- Both input and output are binary (say take value 0 or 1)
- The channel is symmetric in the sense that

$$p_{Y|X}(1|0) = p_{Y|X}(0|1) = p$$

and

$$p_{Y|X}(0|0) = p_{Y|X}(1|1) = 1 - p,$$

where  $p$  is known to be the cross-over probability

- Capacity is given by

$$\begin{aligned} C &= \max_{p(x)} I(X; Y) \\ &= \max_{p(x)} H(Y) - H(Y|X) \end{aligned}$$

# Example: Binary symmetric channel

- Both input and output are binary (say take value 0 or 1)
- The channel is symmetric in the sense that

$$p_{Y|X}(1|0) = p_{Y|X}(0|1) = p$$

and

$$p_{Y|X}(0|0) = p_{Y|X}(1|1) = 1 - p,$$

where  $p$  is known to be the cross-over probability

- Capacity is given by

$$\begin{aligned} C &= \max_{p(x)} I(X; Y) \\ &= \max_{p(x)} H(Y) - H(Y|X) \\ &= \max_{p(x)} H(Y) - H(p) \end{aligned}$$

# Example: Binary symmetric channel

- Both input and output are binary (say take value 0 or 1)
- The channel is symmetric in the sense that

$$p_{Y|X}(1|0) = p_{Y|X}(0|1) = p$$

and

$$p_{Y|X}(0|0) = p_{Y|X}(1|1) = 1 - p,$$

where  $p$  is known to be the cross-over probability

- Capacity is given by

$$\begin{aligned} C &= \max_{p(x)} I(X; Y) \\ &= \max_{p(x)} H(Y) - H(Y|X) \\ &= \max_{p(x)} H(Y) - H(p) = 1 - H(p) \end{aligned}$$

# Example: Gaussian channel

The channel output  $Y = X + Z$ , where  $Z$  is a **zero-mean** Gaussian noise (independent of the input  $X$ )

$$C = \max_{p(x)} I(X; Y)$$

# Example: Gaussian channel

The channel output  $Y = X + Z$ , where  $Z$  is a **zero-mean** Gaussian noise (independent of the input  $X$ )

$$\begin{aligned} C &= \max_{p(x)} I(X; Y) \\ &= \max_{p(x)} h(Y) - h(Y|X) \end{aligned}$$

# Example: Gaussian channel

The channel output  $Y = X + Z$ , where  $Z$  is a **zero-mean** Gaussian noise (independent of the input  $X$ )

$$\begin{aligned} C &= \max_{p(x)} I(X; Y) \\ &= \max_{p(x)} h(Y) - h(Y|X) = \max_{p(x)} h(Y) - h(X + Z|X) \end{aligned}$$

# Example: Gaussian channel

The channel output  $Y = X + Z$ , where  $Z$  is a **zero-mean** Gaussian noise (independent of the input  $X$ )

$$\begin{aligned} C &= \max_{p(x)} I(X; Y) \\ &= \max_{p(x)} h(Y) - h(Y|X) = \max_{p(x)} h(Y) - h(X + Z|X) \\ &= \max_{p(x)} h(Y) - h(Z|X) \end{aligned}$$

# Example: Gaussian channel

The channel output  $Y = X + Z$ , where  $Z$  is a **zero-mean** Gaussian noise (independent of the input  $X$ )

$$\begin{aligned} C &= \max_{p(x)} I(X; Y) \\ &= \max_{p(x)} h(Y) - h(Y|X) = \max_{p(x)} h(Y) - h(X + Z|X) \\ &= \max_{p(x)} h(Y) - h(Z|X) = \max_{p(x)} h(Y) - h(Z) \end{aligned}$$



# Example: Gaussian channel

The channel output  $Y = X + Z$ , where  $Z$  is a **zero-mean** Gaussian noise (independent of the input  $X$ )

$$\begin{aligned} C &= \max_{p(x)} I(X; Y) \\ &= \max_{p(x)} h(Y) - h(Y|X) = \max_{p(x)} h(Y) - h(X + Z|X) \\ &= \max_{p(x)} h(Y) - h(Z|X) = \max_{p(x)} h(Y) - h(Z) \\ &= \max_{p(x)} h(Y) - \frac{1}{2} \log 2\pi e\sigma_Z^2 \end{aligned}$$

# Example: Gaussian channel

The channel output  $Y = X + Z$ , where  $Z$  is a **zero-mean** Gaussian noise (independent of the input  $X$ )

$$\begin{aligned} C &= \max_{p(x)} I(X; Y) \\ &= \max_{p(x)} h(Y) - h(Y|X) = \max_{p(x)} h(Y) - h(X + Z|X) \\ &= \max_{p(x)} h(Y) - h(Z|X) = \max_{p(x)} h(Y) - h(Z) \\ &= \max_{p(x)} h(Y) - \frac{1}{2} \log 2\pi e \sigma_Z^2 = \frac{1}{2} \log 2\pi e \sigma_Y^2 - \frac{1}{2} \log 2\pi e \sigma_Z^2 \end{aligned}$$

# Example: Gaussian channel

The channel output  $Y = X + Z$ , where  $Z$  is a **zero-mean** Gaussian noise (independent of the input  $X$ )

$$\begin{aligned} C &= \max_{p(x)} I(X; Y) \\ &= \max_{p(x)} h(Y) - h(Y|X) = \max_{p(x)} h(Y) - h(X + Z|X) \\ &= \max_{p(x)} h(Y) - h(Z|X) = \max_{p(x)} h(Y) - h(Z) \\ &= \max_{p(x)} h(Y) - \frac{1}{2} \log 2\pi e \sigma_Z^2 = \frac{1}{2} \log 2\pi e \sigma_Y^2 - \frac{1}{2} \log 2\pi e \sigma_Z^2 \\ &= \frac{1}{2} \log \frac{\sigma_X^2 + \sigma_Z^2}{\sigma_Z^2} = \frac{1}{2} \log \left( 1 + \frac{\sigma_X^2}{\sigma_Z^2} \right) \end{aligned}$$

# Example: Gaussian channel

The channel output  $Y = X + Z$ , where  $Z$  is a **zero-mean** Gaussian noise (independent of the input  $X$ )

$$\begin{aligned}
 C &= \max_{p(x)} I(X; Y) \\
 &= \max_{p(x)} h(Y) - h(Y|X) = \max_{p(x)} h(Y) - h(X + Z|X) \\
 &= \max_{p(x)} h(Y) - h(Z|X) = \max_{p(x)} h(Y) - h(Z) \\
 &= \max_{p(x)} h(Y) - \frac{1}{2} \log 2\pi e \sigma_Z^2 = \frac{1}{2} \log 2\pi e \sigma_Y^2 - \frac{1}{2} \log 2\pi e \sigma_Z^2 \\
 &= \frac{1}{2} \log \frac{\sigma_X^2 + \sigma_Z^2}{\sigma_Z^2} = \frac{1}{2} \log \left( 1 + \frac{\sigma_X^2}{\sigma_Z^2} \right) = \frac{1}{2} \log(1 + SNR),
 \end{aligned}$$

where  $SNR$  is the signal to noise ratio

# Example: Additive White Gaussian Noise (AWGN) channel

Consider an AWGN channel with bandwidth  $W$  and two-sided power spectrum density of  $N_0/2$

# Example: Additive White Gaussian Noise (AWGN) channel

Consider an AWGN channel with bandwidth  $W$  and two-sided power spectrum density of  $N_0/2$

- From the result of Nyquist and Shannon, a signal of bandwidth  $W$  will need to at least  $2W$  samples per second to be fully reconstructed

# Example: Additive White Gaussian Noise (AWGN) channel

Consider an AWGN channel with bandwidth  $W$  and two-sided power spectrum density of  $N_0/2$

- From the result of Nyquist and Shannon, a signal of bandwidth  $W$  will need to at least  $2W$  samples per second to be fully reconstructed
- Per each second,  $2W$  samples needed to recover the signal

# Example: Additive White Gaussian Noise (AWGN) channel

Consider an AWGN channel with bandwidth  $W$  and two-sided power spectrum density of  $N_0/2$

- From the result of Nyquist and Shannon, a signal of bandwidth  $W$  will need to at least  $2W$  samples per second to be fully reconstructed
- Per each second,  $2W$  samples needed to recover the signal
- Per each second,  $2W$  degrees of freedom exists  $\Rightarrow 2W$  parallel Gaussian channel per second



# Example: Additive White Gaussian Noise (AWGN) channel

Consider an AWGN channel with bandwidth  $W$  and two-sided power spectrum density of  $N_0/2$

- From the result of Nyquist and Shannon, a signal of bandwidth  $W$  will need to at least  $2W$  samples per second to be fully reconstructed
- Per each second,  $2W$  samples needed to recover the signal
- Per each second,  $2W$  degrees of freedom exists  $\Rightarrow 2W$  parallel Gaussian channel per second
- Given  $N_0$ ,  $SNR = \frac{\sigma_X^2}{WN_0} = \frac{P}{WN_0}$

# Example: Additive White Gaussian Noise (AWGN) channel

Consider an AWGN channel with bandwidth  $W$  and two-sided power spectrum density of  $N_0/2$

- From the result of Nyquist and Shannon, a signal of bandwidth  $W$  will need to at least  $2W$  samples per second to be fully reconstructed
- Per each second,  $2W$  samples needed to recover the signal
- Per each second,  $2W$  degrees of freedom exists  $\Rightarrow 2W$  parallel Gaussian channel per second
- Given  $N_0$ ,  $SNR = \frac{\sigma_x^2}{WN_0} = \frac{P}{WN_0}$

$$C = 2W \frac{1}{2} \log(1 + SNR) = W \log \left( 1 + \frac{P}{WN_0} \right)$$

# Codebook construction

## Forward statement

If the code rate  $R < C = \max_{p(x)} I(X; Y)$ , according to the Channel Coding Theorem, we should be able to find a code with encoding mapping  $\mathbf{c} : m \in \{1, 2, \dots, 2^{NR}\} \rightarrow \{0, 1\}^N$  and the error probability of transmitting any message  $m \in \{1, 2, \dots, 2^{NR}\}$ ,  $p_e(m)$ , is arbitrarily small

# Codebook construction

## Forward statement

If the code rate  $R < C = \max_{p(x)} I(X; Y)$ , according to the Channel Coding Theorem, we should be able to find a code with encoding mapping  $\mathbf{c} : m \in \{1, 2, \dots, 2^{NR}\} \rightarrow \{0, 1\}^N$  and the error probability of transmitting any message  $m \in \{1, 2, \dots, 2^{NR}\}$ ,  $p_e(m)$ , is arbitrarily small

- The main tool of the proof is **random coding**

# Codebook construction

## Forward statement

If the code rate  $R < C = \max_{p(x)} I(X; Y)$ , according to the Channel Coding Theorem, we should be able to find a code with encoding mapping  $\mathbf{c} : m \in \{1, 2, \dots, 2^{NR}\} \rightarrow \{0, 1\}^N$  and the error probability of transmitting any message  $m \in \{1, 2, \dots, 2^{NR}\}$ ,  $p_e(m)$ , is arbitrarily small

- The main tool of the proof is **random coding**
- Let  $p^*(x) = \arg \max_{p(x)} I(X; Y)$ . Generate codewords from the DMS  $p^*(x)$  by sampling  $2^n$  length- $n$  sequences from the source:

$$\mathbf{c}(1) = (x_1(1), x_2(1), \dots, x_N(1))$$

$$\mathbf{c}(2) = (x_1(2), x_2(2), \dots, x_N(2))$$

...

$$\mathbf{c}(2^{NR}) = (x_1(2^{NR}), x_2(2^{NR}), \dots, x_N(2^{NR}))$$

# Encoding and decoding

The encoding and decoding procedures will be as follows.

# Encoding and decoding

The encoding and decoding procedures will be as follows.

## Encoding

For input message  $m$ , output  $\mathbf{c}(m) = (x_1(m), x_2(m), \dots, x_N(m))$

# Encoding and decoding

The encoding and decoding procedures will be as follows.

## Encoding

For input message  $m$ , output  $\mathbf{c}(m) = (x_1(m), x_2(m), \dots, x_N(m))$

## Decoding

Upon receiving sequence  $\mathbf{y} = (y_1, y_2, \dots, y_N)$ , pick the sequence  $\mathbf{c}(m)$  from  $\{\mathbf{c}(1), \dots, \mathbf{c}(2^{NR})\}$  such that  $(\mathbf{c}(m), \mathbf{y})$  are jointly typical. That is  $p_{X^N, Y^N}(\mathbf{c}(m), \mathbf{y}) \sim 2^{-nH(X, Y)}$ . If no such  $\mathbf{c}(m)$  exists or more than one such sequence exist, announce error. Otherwise output the decoded message as  $m$



# Average performance

Without loss of generality, let us assume  $M = 1$ , decoding error occurs when:

# Average performance

Without loss of generality, let us assume  $M = 1$ , decoding error occurs when:

$$\textcircled{1} P_1 = Pr(\mathbf{C}(1), \mathbf{Y}) \notin A_\epsilon^N(X, Y)$$

# Average performance

Without loss of generality, let us assume  $M = 1$ , decoding error occurs when:

- 1  $P_1 = Pr(\mathbf{C}(1), \mathbf{Y}) \notin A_\epsilon^N(X, Y)$
- 2  $P_2 : \exists M' \neq 1$  and  $(\mathbf{c}(M'), \mathbf{Y}) \in A_\epsilon^N(X, Y)$

Thus  $P(\text{error}) = P(\text{error}|M = 1) \leq P_1 + P_2$

# Average performance

Without loss of generality, let us assume  $M = 1$ , decoding error occurs when:

- ①  $P_1 = Pr(\mathbf{C}(1), \mathbf{Y}) \notin A_\epsilon^N(X, Y)$
- ②  $P_2 : \exists M' \neq 1$  and  $(\mathbf{c}(M'), \mathbf{Y}) \in A_\epsilon^N(X, Y)$

Thus  $P(\text{error}) = P(\text{error}|M = 1) \leq P_1 + P_2$

- ① Since  $(\mathbf{C}(1), \mathbf{Y})$  is coming out of the joint source  $X, Y$ ,  $P_1 \rightarrow 0$  as  $n \rightarrow \infty$

# Average performance

Without loss of generality, let us assume  $M = 1$ , decoding error occurs when:

- ①  $P_1 = Pr(\mathbf{C}(1), \mathbf{Y}) \notin A_\epsilon^N(X, Y)$
- ②  $P_2 : \exists M' \neq 1$  and  $(\mathbf{c}(M'), \mathbf{Y}) \in A_\epsilon^N(X, Y)$

Thus  $P(\text{error}) = P(\text{error}|M = 1) \leq P_1 + P_2$

- ① Since  $(\mathbf{C}(1), \mathbf{Y})$  is coming out of the joint source  $X, Y$ ,  $P_1 \rightarrow 0$  as  $n \rightarrow \infty$
- ② Note that  $\mathbf{C}(M')$  and  $\mathbf{Y}$  are independent and thus by Packing lemma,

$$P_2 \leq 2^{-N(I(X;Y)-R-3\epsilon)} \quad (1)$$

# Average performance

Without loss of generality, let us assume  $M = 1$ , decoding error occurs when:

- ①  $P_1 = Pr(\mathbf{C}(1), \mathbf{Y}) \notin A_\epsilon^N(X, Y)$
- ②  $P_2 : \exists M' \neq 1$  and  $(\mathbf{c}(M'), \mathbf{Y}) \in A_\epsilon^N(X, Y)$

Thus  $P(\text{error}) = P(\text{error} | M = 1) \leq P_1 + P_2$

- ① Since  $(\mathbf{C}(1), \mathbf{Y})$  is coming out of the joint source  $X, Y$ ,  $P_1 \rightarrow 0$  as  $n \rightarrow \infty$
- ② Note that  $\mathbf{C}(M')$  and  $\mathbf{Y}$  are independent and thus by Packing lemma,

$$P_2 \leq 2^{-N(I(X;Y) - R - 3\epsilon)} \quad (1)$$

Since  $\epsilon$  can be made arbitrarily small as  $N$  increase, as long as  $I(X; Y) - 3\epsilon > R$ , we can make  $P_2$  arbitrarily small also given a sufficiently large  $N$

# A bit more caveat

- We show that the average error over all random codes can be made arbitrarily small

# A bit more caveat

- We show that the average error over all random codes can be made arbitrarily small
- But we really *want* is to find a code  $\mathbf{c}^*(\cdot)$  and ensure that  $Pr(\text{error}|\mathbf{c}^*, m) \rightarrow 0$  no matter what message  $m$  is sent



# A bit more caveat

- We show that the average error over all random codes can be made arbitrarily small
- But we really *want* is to find a code  $\mathbf{c}^*(\cdot)$  and ensure that  $Pr(\text{error}|\mathbf{c}^*, m) \rightarrow 0$  no matter what message  $m$  is sent
- Let say for a finite  $N$ , the average error is  $\delta$ . Then, we should be able to find a code  $\mathbf{c}^*$  such that it has average error at least equal to  $\delta$

# A bit more caveat

- We show that the average error over all random codes can be made arbitrarily small
- But we really *want* is to find a code  $\mathbf{c}^*(\cdot)$  and ensure that  $Pr(error|\mathbf{c}^*, m) \rightarrow 0$  no matter what message  $m$  is sent
- Let say for a finite  $N$ , the average error is  $\delta$ . Then, we should be able to find a code  $\mathbf{c}^*$  such that it has average error at least equal to  $\delta$
- Without loss of generality and for simplicity, assume that all messages are equally likely  $Pr(error|\mathbf{c}^*) = \frac{1}{2^{NR}} \sum_m Pr(error|\mathbf{c}^*, m) \leq \delta$

# A bit more caveat

- We show that the average error over all random codes can be made arbitrarily small
- But we really *want* is to find a code  $\mathbf{c}^*(\cdot)$  and ensure that  $Pr(error|\mathbf{c}^*, m) \rightarrow 0$  no matter what message  $m$  is sent
- Let say for a finite  $N$ , the average error is  $\delta$ . Then, we should be able to find a code  $\mathbf{c}^*$  such that it has average error at least equal to  $\delta$
- Without loss of generality and for simplicity, assume that all messages are equally likely  $Pr(error|\mathbf{c}^*) = \frac{1}{2^{NR}} \sum_m Pr(error|\mathbf{c}^*, m) \leq \delta$
- If we discard the worse half of the codewords, for any remaining message  $m$ , we have  $Pr(error|\mathbf{c}^*, m) \leq 2Pr(error|\mathbf{c}^*) \leq 2\delta \rightarrow 0$  as  $N \rightarrow \infty$

# A bit more caveat

- We show that the average error over all random codes can be made arbitrarily small
- But we really *want* is to find a code  $\mathbf{c}^*(\cdot)$  and ensure that  $Pr(error|\mathbf{c}^*, m) \rightarrow 0$  no matter what message  $m$  is sent
- Let say for a finite  $N$ , the average error is  $\delta$ . Then, we should be able to find a code  $\mathbf{c}^*$  such that it has average error at least equal to  $\delta$
- Without loss of generality and for simplicity, assume that all messages are equally likely  $Pr(error|\mathbf{c}^*) = \frac{1}{2^{NR}} \sum_m Pr(error|\mathbf{c}^*, m) \leq \delta$
- If we discard the worse half of the codewords, for any remaining message  $m$ , we have  $Pr(error|\mathbf{c}^*, m) \leq 2Pr(error|\mathbf{c}^*) \leq 2\delta \rightarrow 0$  as  $N \rightarrow \infty$
- Even though the rate reduces from  $R$  to  $R - \frac{1}{N}$  (number of messages from  $2^{NR} \rightarrow 2^{NR-1}$ ). But we can still make the final rate arbitrarily close to the capacity as  $N \rightarrow \infty$

# Previously...

- Joint typical sequences
- Covering and Packing Lemmas
- Channel Coding Theorem
- Capacity of Gaussian channel
- Capacity of additive white Gaussian channel
- Forward proof of Channel Coding Theorem

# This time

- Converse Proof of Channel Coding Theorem
- Non-white Gaussian Channel
- Rate-distortion problems
- Rate-distortion Theorem

# Converse proof

We want to say that whenever the code rate is larger than the capacity, the probability of error will be non-zero

# Converse proof

We want to say that whenever the code rate is larger than the capacity, the probability of error will be non-zero

Equivalently...

As long as the probability of error is 0, the rate of the code  $R$  has to be larger than the capacity



# Converse proof

We want to say that whenever the code rate is larger than the capacity, the probability of error will be non-zero

Equivalently...

As long as the probability of error is 0, the rate of the code  $R$  has to be larger than the capacity

To continue the converse proof, we will need to introduce a simple result from Fano

# Fano's inequality

## Fano's inequality

Denote  $Pr(\text{error}) = P_e = Pr(M \neq \hat{M})$ , then  $H(M|Y^N) \leq 1 + P_e H(M)$

Intuitively, if  $P_e \rightarrow 0$ , on average we will know  $M$  for certain given  $y$  and

thus  $\frac{1}{N} H(M|Y^N) \rightarrow 0$

# Fano's inequality

## Fano's inequality

Denote  $Pr(\text{error}) = P_e = Pr(M \neq \hat{M})$ , then  $H(M|Y^N) \leq 1 + P_e H(M)$

Intuitively, if  $P_e \rightarrow 0$ , on average we will know  $M$  for certain given  $y$  and thus  $\frac{1}{N} H(M|Y^N) \rightarrow 0$

Proof: Let  $E = I(M \neq \hat{M})$ , then

# Fano's inequality

## Fano's inequality

Denote  $Pr(\text{error}) = P_e = Pr(M \neq \hat{M})$ , then  $H(M|Y^N) \leq 1 + P_e H(M)$   
Intuitively, if  $P_e \rightarrow 0$ , on average we will know  $M$  for certain given  $y$  and thus  $\frac{1}{N} H(M|Y^N) \rightarrow 0$

Proof: Let  $E = I(M \neq \hat{M})$ , then

$$H(M|Y^N) = H(M, E|Y^N) - H(E|Y^N, M)$$

# Fano's inequality

## Fano's inequality

Denote  $Pr(\text{error}) = P_e = Pr(M \neq \hat{M})$ , then  $H(M|Y^N) \leq 1 + P_e H(M)$   
Intuitively, if  $P_e \rightarrow 0$ , on average we will know  $M$  for certain given  $y$  and thus  $\frac{1}{N} H(M|Y^N) \rightarrow 0$

Proof: Let  $E = I(M \neq \hat{M})$ , then

$$\begin{aligned} H(M|Y^N) &= H(M, E|Y^N) - H(E|Y^N, M) \\ &= H(M, E|Y^N) = H(E|Y^N) + H(M|Y^N, E) \end{aligned}$$

# Fano's inequality

## Fano's inequality

Denote  $Pr(\text{error}) = P_e = Pr(M \neq \hat{M})$ , then  $H(M|Y^N) \leq 1 + P_e H(M)$   
 Intuitively, if  $P_e \rightarrow 0$ , on average we will know  $M$  for certain given  $y$  and thus  $\frac{1}{N} H(M|Y^N) \rightarrow 0$

Proof: Let  $E = I(M \neq \hat{M})$ , then

$$\begin{aligned} H(M|Y^N) &= H(M, E|Y^N) - H(E|Y^N, M) \\ &= H(M, E|Y^N) = H(E|Y^N) + H(M|Y^N, E) \\ &\leq H(E) + H(M|Y^N, E) \end{aligned}$$

# Fano's inequality

## Fano's inequality

Denote  $Pr(\text{error}) = P_e = Pr(M \neq \hat{M})$ , then  $H(M|Y^N) \leq 1 + P_e H(M)$   
 Intuitively, if  $P_e \rightarrow 0$ , on average we will know  $M$  for certain given  $y$  and thus  $\frac{1}{N} H(M|Y^N) \rightarrow 0$

Proof: Let  $E = I(M \neq \hat{M})$ , then

$$\begin{aligned}
 H(M|Y^N) &= H(M, E|Y^N) - H(E|Y^N, M) \\
 &= H(M, E|Y^N) = H(E|Y^N) + H(M|Y^N, E) \\
 &\leq H(E) + H(M|Y^N, E) \\
 &\leq 1 + P(E = 0)H(M|Y^N, E = 0) + P(E = 1)H(M|Y^N, E = 1)
 \end{aligned}$$

# Fano's inequality

## Fano's inequality

Denote  $Pr(\text{error}) = P_e = Pr(M \neq \hat{M})$ , then  $H(M|Y^N) \leq 1 + P_e H(M)$   
 Intuitively, if  $P_e \rightarrow 0$ , on average we will know  $M$  for certain given  $y$  and thus  $\frac{1}{N} H(M|Y^N) \rightarrow 0$

Proof: Let  $E = I(M \neq \hat{M})$ , then

$$\begin{aligned}
 H(M|Y^N) &= H(M, E|Y^N) - H(E|Y^N, M) \\
 &= H(M, E|Y^N) = H(E|Y^N) + H(M|Y^N, E) \\
 &\leq H(E) + H(M|Y^N, E) \\
 &\leq 1 + P(E=0)H(M|Y^N, E=0) + P(E=1)H(M|Y^N, E=1) \\
 &\leq 1 + 0 + P_e H(M|Y^N, E=1) \stackrel{(d)}{\leq} 1 + P_e H(M)
 \end{aligned}$$



# Converse proof

$$R = \frac{H(M)}{N} = \frac{1}{N} \left[ I(M; Y^N) + H(M|Y^N) \right]$$

# Converse proof

$$R = \frac{H(M)}{N} = \frac{1}{N} \left[ I(M; Y^N) + H(M|Y^N) \right] \leq \frac{1}{N} \left[ I(X^N; Y^N) + H(M|Y^N) \right]$$

# Converse proof

$$\begin{aligned} R &= \frac{H(M)}{N} = \frac{1}{N} \left[ I(M; Y^N) + H(M|Y^N) \right] \leq \frac{1}{N} \left[ I(X^N; Y^N) + H(M|Y^N) \right] \\ &= \frac{1}{N} \left[ H(Y^N) - H(Y^N|X^N) + H(M|Y^N) \right] \end{aligned}$$

## Converse proof

$$\begin{aligned} R &= \frac{H(M)}{N} = \frac{1}{N} \left[ I(M; Y^N) + H(M|Y^N) \right] \leq \frac{1}{N} \left[ I(X^N; Y^N) + H(M|Y^N) \right] \\ &= \frac{1}{N} \left[ H(Y^N) - H(Y^N|X^N) + H(M|Y^N) \right] \\ &= \frac{1}{N} \left[ H(Y^N) - \sum_i H(Y_i|X^N, Y^{i-1}) + H(M|Y^N) \right] \end{aligned}$$

# Converse proof

$$\begin{aligned}
 R &= \frac{H(M)}{N} = \frac{1}{N} \left[ I(M; Y^N) + H(M|Y^N) \right] \leq \frac{1}{N} \left[ I(X^N; Y^N) + H(M|Y^N) \right] \\
 &= \frac{1}{N} \left[ H(Y^N) - H(Y^N|X^N) + H(M|Y^N) \right] \\
 &= \frac{1}{N} \left[ H(Y^N) - \sum_i H(Y_i|X^N, Y^{i-1}) + H(M|Y^N) \right] \\
 &= \frac{1}{N} \left[ H(Y^N) - \sum_i H(Y_i|X_i) + H(M|Y^N) \right]
 \end{aligned}$$

# Converse proof

$$\begin{aligned}
 R &= \frac{H(M)}{N} = \frac{1}{N} \left[ I(M; Y^N) + H(M|Y^N) \right] \leq \frac{1}{N} \left[ I(X^N; Y^N) + H(M|Y^N) \right] \\
 &= \frac{1}{N} \left[ H(Y^N) - H(Y^N|X^N) + H(M|Y^N) \right] \\
 &= \frac{1}{N} \left[ H(Y^N) - \sum_i H(Y_i|X^N, Y^{i-1}) + H(M|Y^N) \right] \\
 &= \frac{1}{N} \left[ H(Y^N) - \sum_i H(Y_i|X_i) + H(M|Y^N) \right] \\
 &\leq \frac{1}{N} \left[ \sum_i H(Y_i) - \sum_i H(Y_i|X_i) + H(M|Y^N) \right]
 \end{aligned}$$

# Converse proof

$$\begin{aligned}
 R &= \frac{H(M)}{N} = \frac{1}{N} \left[ I(M; Y^N) + H(M|Y^N) \right] \leq \frac{1}{N} \left[ I(X^N; Y^N) + H(M|Y^N) \right] \\
 &= \frac{1}{N} \left[ H(Y^N) - H(Y^N|X^N) + H(M|Y^N) \right] \\
 &= \frac{1}{N} \left[ H(Y^N) - \sum_i H(Y_i|X^N, Y^{i-1}) + H(M|Y^N) \right] \\
 &= \frac{1}{N} \left[ H(Y^N) - \sum_i H(Y_i|X_i) + H(M|Y^N) \right] \\
 &\leq \frac{1}{N} \left[ \sum_i H(Y_i) - \sum_i H(Y_i|X_i) + H(M|Y^N) \right] \\
 &= \frac{1}{N} \left[ \sum_i I(X_i; Y_i) + H(M|Y^N) \right] = I(X; Y) + \frac{H(M|Y^N)}{N}
 \end{aligned}$$

# Converse proof

$$\begin{aligned}
 R &= \frac{H(M)}{N} = \frac{1}{N} \left[ I(M; Y^N) + H(M|Y^N) \right] \leq \frac{1}{N} \left[ I(X^N; Y^N) + H(M|Y^N) \right] \\
 &= \frac{1}{N} \left[ H(Y^N) - H(Y^N|X^N) + H(M|Y^N) \right] \\
 &= \frac{1}{N} \left[ H(Y^N) - \sum_i H(Y_i|X^N, Y^{i-1}) + H(M|Y^N) \right] \\
 &= \frac{1}{N} \left[ H(Y^N) - \sum_i H(Y_i|X_i) + H(M|Y^N) \right] \\
 &\leq \frac{1}{N} \left[ \sum_i H(Y_i) - \sum_i H(Y_i|X_i) + H(M|Y^N) \right] \\
 &= \frac{1}{N} \left[ \sum_i I(X_i; Y_i) + H(M|Y^N) \right] = I(X; Y) + \frac{H(M|Y^N)}{N} \rightarrow I(X; Y)
 \end{aligned}$$

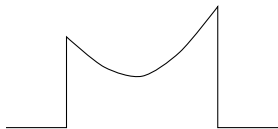
as  $N \rightarrow \infty$  by Fano's inequality



# Color channels

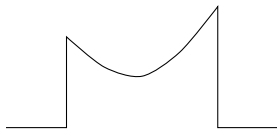
- We look into capacity of white Gaussian channel last time

# Color channels



- We look into capacity of white Gaussian channel last time
- But sometimes noise power can be different for different band, consequently, “color” channels

# Color channels



- We look into capacity of white Gaussian channel last time
- But sometimes noise power can be different for different band, consequently, “color” channels
- Intuitively, we should assign different amount of power to different band. Hence, we have an allocation problem

# Color channels



- We look into capacity of white Gaussian channel last time
- But sometimes noise power can be different for different band, consequently, “color” channels
- Intuitively, we should assign different amount of power to different band. Hence, we have an allocation problem
- Without loss of generality, let’s consider the discrete approximation, parallel Gaussian channel

# Parallel Gaussian channels

- Consider that we have  $K$  parallel channels ( $K$  bands) and the corresponding noise powers are  $\sigma_1^2, \sigma_2^2, \dots, \sigma_K^2$

# Parallel Gaussian channels

- Consider that we have  $K$  parallel channels ( $K$  bands) and the corresponding noise powers are  $\sigma_1^2, \sigma_2^2, \dots, \sigma_K^2$
- And say, we can allocate a total of  $P$  power to all channels. The powers assigned to the channels are  $P_1, P_2, \dots, P_K$ . So we need 
$$\sum_{i=1}^K P_i \leq P$$

# Parallel Gaussian channels

- Consider that we have  $K$  parallel channels ( $K$  bands) and the corresponding noise powers are  $\sigma_1^2, \sigma_2^2, \dots, \sigma_K^2$
- And say, we can allocate a total of  $P$  power to all channels. The powers assigned to the channels are  $P_1, P_2, \dots, P_K$ . So we need  $\sum_{i=1}^K P_i \leq P$
- Therefore, for the  $k$ -th channel, we can transmit  $\frac{1}{2} \log \left( 1 + \frac{P_k}{\sigma_k^2} \right)$  bits per channel use

# Parallel Gaussian channels

- Consider that we have  $K$  parallel channels ( $K$  bands) and the corresponding noise powers are  $\sigma_1^2, \sigma_2^2, \dots, \sigma_K^2$
- And say, we can allocate a total of  $P$  power to all channels. The powers assigned to the channels are  $P_1, P_2, \dots, P_K$ . So we need  $\sum_{i=1}^K P_i \leq P$
- Therefore, for the  $k$ -th channel, we can transmit  $\frac{1}{2} \log \left( 1 + \frac{P_k}{\sigma_k^2} \right)$  bits per channel use
- So our goal is to assign  $P_1, P_2, \dots, P_K \geq 0$  ( $\sum_{k=1}^K P_k \leq P$ ) such that the total capacity

$$\sum_{k=1}^K \frac{1}{2} \log \left( 1 + \frac{P_k}{\sigma_k^2} \right)$$

is maximize



# KKT conditions

Let's list all the KKT conditions for the optimization problem

$$\max \sum_{k=1}^K \frac{1}{2} \log \left( 1 + \frac{P_k}{\sigma_k^2} \right) \quad \text{such that}$$

$$P_1, \dots, P_K \geq 0, \quad \sum_{k=1}^K P_k \leq P$$

$$\frac{\partial}{\partial P_i} \left[ \sum_{k=1}^K \frac{1}{2} \log \left( 1 + \frac{P_k}{\sigma_k^2} \right) + \sum_{k=1}^K \lambda_k P_k - \mu \left( \sum_{k=1}^K P_k - P \right) \right] = 0$$

# KKT conditions

Let's list all the KKT conditions for the optimization problem

$$\max \sum_{k=1}^K \frac{1}{2} \log \left( 1 + \frac{P_k}{\sigma_k^2} \right) \quad \text{such that}$$

$$P_1, \dots, P_K \geq 0, \quad \sum_{k=1}^K P_k \leq P$$

$$\frac{\partial}{\partial P_i} \left[ \sum_{k=1}^K \frac{1}{2} \log \left( 1 + \frac{P_k}{\sigma_k^2} \right) + \sum_{k=1}^K \lambda_k P_k - \mu \left( \sum_{k=1}^K P_k - P \right) \right] = 0$$

$$\mu, \lambda_1, \dots, \lambda_K \geq 0$$

# KKT conditions

Let's list all the KKT conditions for the optimization problem

$$\max \sum_{k=1}^K \frac{1}{2} \log \left( 1 + \frac{P_k}{\sigma_k^2} \right) \quad \text{such that}$$

$$P_1, \dots, P_K \geq 0, \quad \sum_{k=1}^K P_k \leq P$$

$$\frac{\partial}{\partial P_i} \left[ \sum_{k=1}^K \frac{1}{2} \log \left( 1 + \frac{P_k}{\sigma_k^2} \right) + \sum_{k=1}^K \lambda_k P_k - \mu \left( \sum_{k=1}^K P_k - P \right) \right] = 0$$

$$\mu, \lambda_1, \dots, \lambda_K \geq 0, P_1, \dots, P_K \geq 0, \sum_{k=1}^K P_k \leq P$$

# KKT conditions

Let's list all the KKT conditions for the optimization problem

$$\max \sum_{k=1}^K \frac{1}{2} \log \left( 1 + \frac{P_k}{\sigma_k^2} \right) \quad \text{such that}$$

$$P_1, \dots, P_K \geq 0, \quad \sum_{k=1}^K P_k \leq P$$

$$\frac{\partial}{\partial P_i} \left[ \sum_{k=1}^K \frac{1}{2} \log \left( 1 + \frac{P_k}{\sigma_k^2} \right) + \sum_{k=1}^K \lambda_k P_k - \mu \left( \sum_{k=1}^K P_k - P \right) \right] = 0$$

$$\mu, \lambda_1, \dots, \lambda_K \geq 0, P_1, \dots, P_K \geq 0, \sum_{k=1}^K P_k \leq P$$

$$\mu \left( \sum_{k=1}^K P_k - P \right) = 0, \quad \lambda_k P_k = 0, \forall k$$

# Capacity of parallel channels

$$\frac{\partial}{\partial P_i} \left[ \sum_{k=1}^K \frac{1}{2} \log \left( 1 + \frac{P_k}{\sigma_k^2} \right) + \sum_{k=1}^K \lambda_k P_k - \mu \left( \sum_{k=1}^K P_k - P \right) \right] = 0$$

# Capacity of parallel channels

$$\frac{\partial}{\partial P_i} \left[ \sum_{k=1}^K \frac{1}{2} \log \left( 1 + \frac{P_k}{\sigma_k^2} \right) + \sum_{k=1}^K \lambda_k P_k - \mu \left( \sum_{k=1}^K P_k - P \right) \right] = 0$$
$$\Rightarrow \frac{1}{2} \frac{1}{P_i + \sigma_i^2} = \mu - \lambda_i$$

# Capacity of parallel channels

$$\frac{\partial}{\partial P_i} \left[ \sum_{k=1}^K \frac{1}{2} \log \left( 1 + \frac{P_k}{\sigma_k^2} \right) + \sum_{k=1}^K \lambda_k P_k - \mu \left( \sum_{k=1}^K P_k - P \right) \right] = 0$$
$$\Rightarrow \frac{1}{2} \frac{1}{P_i + \sigma_i^2} = \mu - \lambda_i \Rightarrow P_i + \sigma_i^2 = \frac{1}{2(\mu - \lambda_i)}$$

# Capacity of parallel channels

$$\frac{\partial}{\partial P_i} \left[ \sum_{k=1}^K \frac{1}{2} \log \left( 1 + \frac{P_k}{\sigma_k^2} \right) + \sum_{k=1}^K \lambda_k P_k - \mu \left( \sum_{k=1}^K P_k - P \right) \right] = 0$$

$$\Rightarrow \frac{1}{2} \frac{1}{P_i + \sigma_i^2} = \mu - \lambda_i \Rightarrow P_i + \sigma_i^2 = \frac{1}{2(\mu - \lambda_i)}$$

Since  $\lambda_i P_i = 0$ , for  $P_i > 0$ , we have  $\lambda_i = 0$  and thus

$$P_i + \sigma_i^2 = \frac{1}{2\mu}$$



## Capacity of parallel channels

$$\frac{\partial}{\partial P_i} \left[ \sum_{k=1}^K \frac{1}{2} \log \left( 1 + \frac{P_k}{\sigma_k^2} \right) + \sum_{k=1}^K \lambda_k P_k - \mu \left( \sum_{k=1}^K P_k - P \right) \right] = 0$$

$$\Rightarrow \frac{1}{2} \frac{1}{P_i + \sigma_i^2} = \mu - \lambda_i \Rightarrow P_i + \sigma_i^2 = \frac{1}{2(\mu - \lambda_i)}$$

Since  $\lambda_i P_i = 0$ , for  $P_i > 0$ , we have  $\lambda_i = 0$  and thus

$$P_i + \sigma_i^2 = \frac{1}{2\mu}$$

This suggests that  $\mu > 0$  and thus  $\sum_{k=1}^K P_k = P$

## Capacity of parallel channels

$$\frac{\partial}{\partial P_i} \left[ \sum_{k=1}^K \frac{1}{2} \log \left( 1 + \frac{P_k}{\sigma_k^2} \right) + \sum_{k=1}^K \lambda_k P_k - \mu \left( \sum_{k=1}^K P_k - P \right) \right] = 0$$

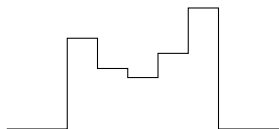
$$\Rightarrow \frac{1}{2} \frac{1}{P_i + \sigma_i^2} = \mu - \lambda_i \Rightarrow P_i + \sigma_i^2 = \frac{1}{2(\mu - \lambda_i)}$$

Since  $\lambda_i P_i = 0$ , for  $P_i > 0$ , we have  $\lambda_i = 0$  and thus

$$P_i + \sigma_i^2 = \frac{1}{2\mu} = \text{constant}$$

This suggests that  $\mu > 0$  and thus  $\sum_{k=1}^K P_k = P$

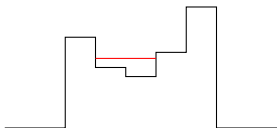
# Water-filling interpretation



From  $P_i + \sigma_i^2 = \text{const}$ , power can be allocated intuitively as filling water to a pond (hence “water-filling”)

## Example

# Water-filling interpretation

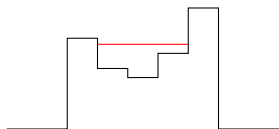


From  $P_i + \sigma_i^2 = \text{const}$ , power can be allocated intuitively as filling water to a pond (hence “water-filling”)

## Example

- $P_1 = 0, P_2 = 0.3, P_3 = 0.6, P_4 = 0, P_5 = 0$

# Water-filling interpretation

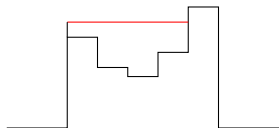


From  $P_i + \sigma_i^2 = \text{const}$ , power can be allocated intuitively as filling water to a pond (hence “water-filling”)

## Example

- $P_1 = 0, P_2 = 0.3, P_3 = 0.6, P_4 = 0, P_5 = 0$
- $P_1 = 0, P_2 = 0.8, P_3 = 1.1, P_4 = 0.3, P_5 = 0$

# Water-filling interpretation



From  $P_i + \sigma_i^2 = \text{const}$ , power can be allocated intuitively as filling water to a pond (hence “water-filling”)

## Example

- $P_1 = 0, P_2 = 0.3, P_3 = 0.6, P_4 = 0, P_5 = 0$
- $P_1 = 0, P_2 = 0.8, P_3 = 1.1, P_4 = 0.3, P_5 = 0$
- $P_1 = 0.5, P_2 = 1.5, P_3 = 1.8, P_4 = 1, P_5 = 0$

# Water-filling interpretation

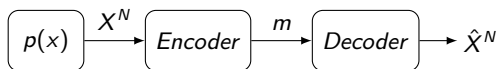


From  $P_i + \sigma_i^2 = \text{const}$ , power can be allocated intuitively as filling water to a pond (hence “water-filling”)

## Example

- $P_1 = 0, P_2 = 0.3, P_3 = 0.6, P_4 = 0, P_5 = 0$
- $P_1 = 0, P_2 = 0.8, P_3 = 1.1, P_4 = 0.3, P_5 = 0$
- $P_1 = 0.5, P_2 = 1.5, P_3 = 1.8, P_4 = 1, P_5 = 0$

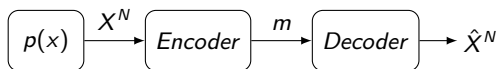
# Rate-distortion problem



- We know that  $H(X)$  bits are needed on average to represent each sample of a source  $X$

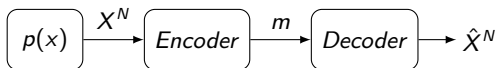


# Rate-distortion problem



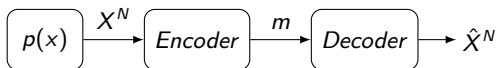
- We know that  $H(X)$  bits are needed on average to represent each sample of a source  $X$
- If  $X$  is continuous, there is no way to recover  $X$  precisely

# Rate-distortion problem



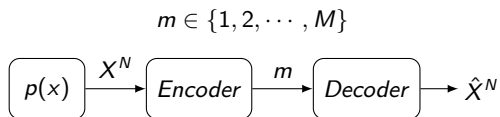
- We know that  $H(X)$  bits are needed on average to represent each sample of a source  $X$
- If  $X$  is continuous, there is no way to recover  $X$  precisely
- Let say we are satisfied as long as we can recover  $X$  up to certain fidelity, how many bits are needed per sample?

# Rate-distortion problem

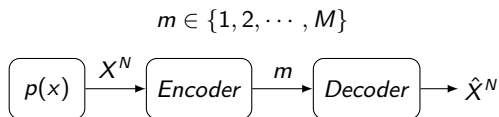


- We know that  $H(X)$  bits are needed on average to represent each sample of a source  $X$
- If  $X$  is continuous, there is no way to recover  $X$  precisely
- Let say we are satisfied as long as we can recover  $X$  up to certain fidelity, how many bits are needed per sample?
- There is an apparent rate (bits per sample) and distortion (fidelity) trade-off. We expect that needed rate is smaller if we allow a lower fidelity (higher distortion). What we are really interested in is a rate-distortion function

# Rate-distortion function

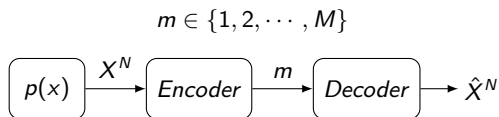


# Rate-distortion function



$$R = \frac{\log M}{N}, \quad D = E[d(\hat{X}^N, X^N)] = \frac{1}{N} \sum_{i=1}^N d(\hat{X}_i, X_i)$$

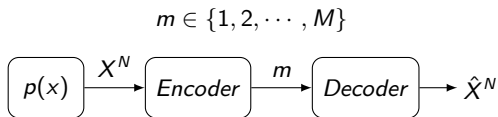
# Rate-distortion function



$$R = \frac{\log M}{N}, \quad D = E[d(\hat{X}^N, X^N)] = \frac{1}{N} \sum_{i=1}^N d(\hat{X}_i, X_i)$$

- Maybe you can guess at this point. For given  $X$  and  $\hat{X}$ , the required rate is simply  $I(X; \hat{X})$

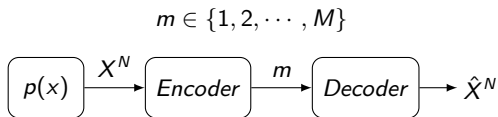
# Rate-distortion function



$$R = \frac{\log M}{N}, \quad D = E[d(\hat{X}^N, X^N)] = \frac{1}{N} \sum_{i=1}^N d(\hat{X}_i, X_i)$$

- Maybe you can guess at this point. For given  $X$  and  $\hat{X}$ , the required rate is simply  $I(X; \hat{X})$
- How is it related to the distortion though?

# Rate-distortion function

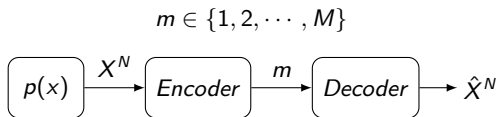


$$R = \frac{\log M}{N}, \quad D = E[d(\hat{X}^N, X^N)] = \frac{1}{N} \sum_{i=1}^N d(\hat{X}_i, X_i)$$

- Maybe you can guess at this point. For given  $X$  and  $\hat{X}$ , the required rate is simply  $I(X; \hat{X})$
- How is it related to the distortion though?
- Note that we have a freedom to pick  $p(\hat{x}|x)$  such that  $E[d(\hat{X}^N, X^N)]$  (less than or) equal to the desired  $\mathcal{D}$



# Rate-distortion function



$$R = \frac{\log M}{N}, \quad D = E[d(\hat{X}^N, X^N)] = \frac{1}{N} \sum_{i=1}^N d(\hat{X}_i, X_i)$$

- Maybe you can guess at this point. For given  $X$  and  $\hat{X}$ , the required rate is simply  $I(X; \hat{X})$
- How is it related to the distortion though?
- Note that we have a freedom to pick  $p(\hat{x}|x)$  such that  $E[d(\hat{X}^N, X^N)]$  (less than or) equal to the desired  $\mathcal{D}$
- Therefore given  $\mathcal{D}$ , the rate-distortion function is simply

$$R(\mathcal{D}) = \min_{p(\hat{x}|x)} I(\hat{X}; X)$$

such that  $E[d(\hat{X}^N, X^N)] \leq \mathcal{D}$

# Binary symmetric source

- Let's try to compress outcome from a fair coin toss

# Binary symmetric source

- Let's try to compress outcome from a fair coin toss
- We know that we need 1 bit to compress the outcome losslessly, what if we have only 0.5 bit per sample?

# Binary symmetric source

- Let's try to compress outcome from a fair coin toss
- We know that we need 1 bit to compress the outcome losslessly, what if we have only 0.5 bit per sample?
- In this case, we can't losslessly recover the outcome. But how good will we do?

# Binary symmetric source

- Let's try to compress outcome from a fair coin toss
- We know that we need 1 bit to compress the outcome losslessly, what if we have only 0.5 bit per sample?
- In this case, we can't losslessly recover the outcome. But how good will we do?
- We need to introduce a distortion measure first. Note that we have two types of errors: taking head as tail and taking tail as head. A natural measure will just weights both error equally

$$d(X = H, \hat{X} = T) = d(X = T, \hat{X} = H) = 1$$

$$d(X = H, \hat{X} = H) = d(X = T, \hat{X} = T) = 0$$

# Binary symmetric source

- Let's try to compress outcome from a fair coin toss
- We know that we need 1 bit to compress the outcome losslessly, what if we have only 0.5 bit per sample?
- In this case, we can't losslessly recover the outcome. But how good will we do?
- We need to introduce a distortion measure first. Note that we have two types of errors: taking head as tail and taking tail as head. A natural measure will just weights both error equally

$$d(X = H, \hat{X} = T) = d(X = T, \hat{X} = H) = 1$$

$$d(X = H, \hat{X} = H) = d(X = T, \hat{X} = T) = 0$$

- If rate is  $> 1$  bit, we know that distortion is 0. How about rate is 0, what distortion suppose to be?

# Binary symmetric source

- Let's try to compress outcome from a fair coin toss
- We know that we need 1 bit to compress the outcome losslessly, what if we have only 0.5 bit per sample?
- In this case, we can't losslessly recover the outcome. But how good will we do?
- We need to introduce a distortion measure first. Note that we have two types of errors: taking head as tail and taking tail as head. A natural measure will just weights both error equally

$$d(X = H, \hat{X} = T) = d(X = T, \hat{X} = H) = 1$$

$$d(X = H, \hat{X} = H) = d(X = T, \hat{X} = T) = 0$$

- If rate is  $> 1$  bit, we know that distortion is 0. How about rate is 0, what distortion suppose to be?
- If decoders know nothing, the best bet will be just always decode head (or tail). Then  $D = E[d(X, H)] = 0.5$

# Binary symmetric source

For  $0 < D < 0.5$ , denote  $Z$  as the prediction error such that  $X = \hat{X} + Z$ .



# Binary symmetric source

For  $0 < D < 0.5$ , denote  $Z$  as the prediction error such that  $X = \hat{X} + Z$ .  
Note that

$$Pr(Z = 1) = D$$

# Binary symmetric source

For  $0 < D < 0.5$ , denote  $Z$  as the prediction error such that  $X = \hat{X} + Z$ . Note that

$$Pr(Z = 1) = D$$

$$R = \min_{p(\hat{x}|x)} I(\hat{X}; X) = \min_{p(\hat{x}|x)} H(X) - H(X|\hat{X})$$

# Binary symmetric source

For  $0 < D < 0.5$ , denote  $Z$  as the prediction error such that  $X = \hat{X} + Z$ .  
Note that

$$Pr(Z = 1) = D$$

$$\begin{aligned} R &= \min_{p(\hat{x}|x)} I(\hat{X}; X) = \min_{p(\hat{x}|x)} H(X) - H(X|\hat{X}) \\ &= \min_{p(\hat{x}|x)} H(X) - H(\hat{X} + Z|\hat{X}) \end{aligned}$$

# Binary symmetric source

For  $0 < D < 0.5$ , denote  $Z$  as the prediction error such that  $X = \hat{X} + Z$ .  
Note that

$$Pr(Z = 1) = D$$

$$\begin{aligned} R &= \min_{p(\hat{x}|x)} I(\hat{X}; X) = \min_{p(\hat{x}|x)} H(X) - H(X|\hat{X}) \\ &= \min_{p(\hat{x}|x)} H(X) - H(\hat{X} + Z|\hat{X}) \\ &= \min_{p(\hat{x}|x)} H(X) - H(Z|\hat{X}) \end{aligned}$$

# Binary symmetric source

For  $0 < D < 0.5$ , denote  $Z$  as the prediction error such that  $X = \hat{X} + Z$ .  
Note that

$$Pr(Z = 1) = D$$

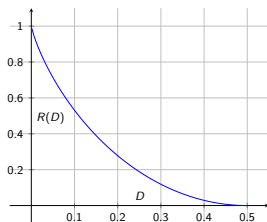
$$\begin{aligned} R &= \min_{p(\hat{x}|x)} I(\hat{X}; X) = \min_{p(\hat{x}|x)} H(X) - H(X|\hat{X}) \\ &= \min_{p(\hat{x}|x)} H(X) - H(\hat{X} + Z|\hat{X}) \\ &= \min_{p(\hat{x}|x)} H(X) - H(Z|\hat{X}) \\ &= \min_{p(\hat{x}|x)} H(X) - H(Z) \end{aligned}$$

# Binary symmetric source

For  $0 < D < 0.5$ , denote  $Z$  as the prediction error such that  $X = \hat{X} + Z$ . Note that

$$Pr(Z = 1) = D$$

$$\begin{aligned} R &= \min_{p(\hat{x}|x)} I(\hat{X}; X) = \min_{p(\hat{x}|x)} H(X) - H(X|\hat{X}) \\ &= \min_{p(\hat{x}|x)} H(X) - H(\hat{X} + Z|\hat{X}) \\ &= \min_{p(\hat{x}|x)} H(X) - H(Z|\hat{X}) \\ &= \min_{p(\hat{x}|x)} H(X) - H(Z) \\ &= 1 - H(D) \end{aligned}$$



# Previously...

- Converse Proof of Channel Coding Theorem
- Non-white Gaussian Channel
- Rate-distortion problems

# This time

- Proof of the Rate-distortion Theorem



# Gaussian source

- Consider  $X \sim \mathcal{N}(0, \sigma_X^2)$ . To determine the rate-distortion function, we need first to decide the distortion measure. An intuitive will be just the square error. That is,

$$d(\hat{X}, X) = (\hat{X} - X)^2$$

# Gaussian source

- Consider  $X \sim \mathcal{N}(0, \sigma_X^2)$ . To determine the rate-distortion function, we need first to decide the distortion measure. An intuitive will be just the square error. That is,

$$d(\hat{X}, X) = (\hat{X} - X)^2$$

- Given  $E[d(\hat{X}, X)] = D$ , what is the minimum rate required?

# Gaussian source

- Consider  $X \sim \mathcal{N}(0, \sigma_X^2)$ . To determine the rate-distortion function, we need first to decide the distortion measure. An intuitive will be just the square error. That is,

$$d(\hat{X}, X) = (\hat{X} - X)^2$$

- Given  $E[d(\hat{X}, X)] = D$ , what is the minimum rate required?
- Like before, let us denote  $Z = X - \hat{X}$  as the prediction error. Note that  $\text{Var}(Z) = D$

# Gaussian source

- Consider  $X \sim \mathcal{N}(0, \sigma_X^2)$ . To determine the rate-distortion function, we need first to decide the distortion measure. An intuitive will be just the square error. That is,

$$d(\hat{X}, X) = (\hat{X} - X)^2$$

- Given  $E[d(\hat{X}, X)] = D$ , what is the minimum rate required?
- Like before, let us denote  $Z = X - \hat{X}$  as the prediction error. Note that  $\text{Var}(Z) = D$

$$R(D) = \min_{p(\hat{x}|x)} I(\hat{X}; X) = \min_{p(\hat{x}|x)} h(X) - h(X|\hat{X})$$

# Gaussian source

- Consider  $X \sim \mathcal{N}(0, \sigma_X^2)$ . To determine the rate-distortion function, we need first to decide the distortion measure. An intuitive will be just the square error. That is,

$$d(\hat{X}, X) = (\hat{X} - X)^2$$

- Given  $E[d(\hat{X}, X)] = D$ , what is the minimum rate required?
- Like before, let us denote  $Z = X - \hat{X}$  as the prediction error. Note that  $\text{Var}(Z) = D$

$$\begin{aligned} R(D) &= \min_{p(\hat{x}|x)} I(\hat{X}; X) = \min_{p(\hat{x}|x)} h(X) - h(X|\hat{X}) \\ &= \min_{p(\hat{x}|x)} h(X) - h(Z + \hat{X}|\hat{X}) \end{aligned}$$

# Gaussian source

- Consider  $X \sim \mathcal{N}(0, \sigma_X^2)$ . To determine the rate-distortion function, we need first to decide the distortion measure. An intuitive will be just the square error. That is,

$$d(\hat{X}, X) = (\hat{X} - X)^2$$

- Given  $E[d(\hat{X}, X)] = D$ , what is the minimum rate required?
- Like before, let us denote  $Z = X - \hat{X}$  as the prediction error. Note that  $\text{Var}(Z) = D$

$$\begin{aligned} R(D) &= \min_{p(\hat{x}|x)} I(\hat{X}; X) = \min_{p(\hat{x}|x)} h(X) - h(X|\hat{X}) \\ &= \min_{p(\hat{x}|x)} h(X) - h(Z + \hat{X}|\hat{X}) \\ &= \min_{p(\hat{x}|x)} h(X) - h(Z|\hat{X}) \end{aligned}$$

# Gaussian source

- Consider  $X \sim \mathcal{N}(0, \sigma_X^2)$ . To determine the rate-distortion function, we need first to decide the distortion measure. An intuitive will be just the square error. That is,

$$d(\hat{X}, X) = (\hat{X} - X)^2$$

- Given  $E[d(\hat{X}, X)] = D$ , what is the minimum rate required?
- Like before, let us denote  $Z = X - \hat{X}$  as the prediction error. Note that  $\text{Var}(Z) = D$

$$\begin{aligned} R(D) &= \min_{p(\hat{x}|x)} I(\hat{X}; X) = \min_{p(\hat{x}|x)} h(X) - h(X|\hat{X}) \\ &= \min_{p(\hat{x}|x)} h(X) - h(Z + \hat{X}|\hat{X}) \\ &= \min_{p(\hat{x}|x)} h(X) - h(Z|\hat{X}) \\ &= \min_{p(\hat{x}|x)} h(X) - h(Z) \end{aligned}$$

# Gaussian source

- Consider  $X \sim \mathcal{N}(0, \sigma_X^2)$ . To determine the rate-distortion function, we need first to decide the distortion measure. An intuitive will be just the square error. That is,

$$d(\hat{X}, X) = (\hat{X} - X)^2$$

- Given  $E[d(\hat{X}, X)] = D$ , what is the minimum rate required?
- Like before, let us denote  $Z = X - \hat{X}$  as the prediction error. Note that  $\text{Var}(Z) = D$

$$\begin{aligned} R(D) &= \min_{p(\hat{x}|x)} I(\hat{X}; X) = \min_{p(\hat{x}|x)} h(X) - h(X|\hat{X}) \\ &= \min_{p(\hat{x}|x)} h(X) - h(Z + \hat{X}|\hat{X}) \\ &= \min_{p(\hat{x}|x)} h(X) - h(Z|\hat{X}) \\ &= \min_{p(\hat{x}|x)} h(X) - h(Z) \\ &= \frac{1}{2} \log \frac{\sigma_X^2}{D} \end{aligned}$$



# Forward proof

## Forward statement

Given distortion constraint  $\mathcal{D}$ , we can find scheme such that the require rate is no bigger than

$$R(\mathcal{D}) = \min_{p(\hat{x}|x)} I(X; \hat{X}),$$

where the  $\hat{X}$  introduced by  $p(\hat{x}|x)$  should satisfy  $E[d(X, \hat{X})] \leq \mathcal{D}$

# Forward proof

## Forward statement

Given distortion constraint  $\mathcal{D}$ , we can find scheme such that the require rate is no bigger than

$$R(\mathcal{D}) = \min_{p(\hat{x}|x)} I(X; \hat{X}),$$

where the  $\hat{X}$  introduced by  $p(\hat{x}|x)$  should satisfy  $E[d(X, \hat{X})] \leq \mathcal{D}$

## Code book construction

Let say  $p^*(\hat{x}|x)$  is the distribution that achieve the rate-distortion optimiation problem. Randomly construct  $2^{NR}$  codewords as follows

# Forward proof

## Forward statement

Given distortion constraint  $\mathcal{D}$ , we can find scheme such that the require rate is no bigger than

$$R(\mathcal{D}) = \min_{p(\hat{x}|x)} I(X; \hat{X}),$$

where the  $\hat{X}$  introduced by  $p(\hat{x}|x)$  should satisfy  $E[d(X, \hat{X})] \leq \mathcal{D}$

## Code book construction

Let say  $p^*(\hat{x}|x)$  is the distribution that achieve the rate-distortion optimiation problem. Randomly construct  $2^{NR}$  codewords as follows

- Sample  $X$  from the source and pass  $X$  into  $p^*(\hat{x}|x)$  to obtain  $\hat{X}$

# Forward proof

## Forward statement

Given distortion constraint  $\mathcal{D}$ , we can find scheme such that the require rate is no bigger than

$$R(\mathcal{D}) = \min_{p(\hat{x}|x)} I(X; \hat{X}),$$

where the  $\hat{X}$  introduced by  $p(\hat{x}|x)$  should satisfy  $E[d(X, \hat{X})] \leq \mathcal{D}$

## Code book construction

Let say  $p^*(\hat{x}|x)$  is the distribution that achieve the rate-distortion optimiation problem. Randomly construct  $2^{NR}$  codewords as follows

- Sample  $X$  from the source and pass  $X$  into  $p^*(\hat{x}|x)$  to obtain  $\hat{X}$
- Repeat this  $N$  time to get a length- $N$  codeword
- Store the  $i$ -th codeword as  $\mathbf{C}(i)$

# Forward proof

## Forward statement

Given distortion constraint  $\mathcal{D}$ , we can find scheme such that the require rate is no bigger than

$$R(\mathcal{D}) = \min_{p(\hat{x}|x)} I(X; \hat{X}),$$

where the  $\hat{X}$  introduced by  $p(\hat{x}|x)$  should satisfy  $E[d(X, \hat{X})] \leq \mathcal{D}$

## Code book construction

Let say  $p^*(\hat{x}|x)$  is the distribution that achieve the rate-distortion optimiation problem. Randomly construct  $2^{NR}$  codewords as follows

- Sample  $X$  from the source and pass  $X$  into  $p^*(\hat{x}|x)$  to obtain  $\hat{X}$
- Repeat this  $N$  time to get a length- $N$  codeword
- Store the  $i$ -th codeword as  $\mathbf{C}(i)$

Note that the code rate is  $\frac{\log 2^{NR}}{N} = R$  as desired

# Covering lemma and distortion typical sequences

We say joint typical sequences  $x^N$  and  $\hat{x}^N$  are distortion typical  $((x^N, \hat{x}^N) \in \mathcal{A}_{d,\epsilon}^N)$  if  $|d(x^N, \hat{x}^N) - E[d(X, \hat{X})]| \leq \epsilon$

# Covering lemma and distortion typical sequences

We say joint typical sequences  $x^N$  and  $\hat{x}^N$  are distortion typical  $((x^N, \hat{x}^N) \in \mathcal{A}_{d,\epsilon}^N)$  if  $|d(x^N, \hat{x}^N) - E[d(X, \hat{X})]| \leq \epsilon$

- By LLN, every pair of sequences sampled from the joint source will virtually be distortion typical

# Covering lemma and distortion typical sequences

We say joint typical sequences  $x^N$  and  $\hat{x}^N$  are distortion typical  $((x^N, \hat{x}^N) \in \mathcal{A}_{d,\epsilon}^N)$  if  $|d(x^N, \hat{x}^N) - E[d(X, \hat{X})]| \leq \epsilon$

- By LLN, every pair of sequences sampled from the joint source will virtually be distortion typical
- Consequently,  $(1 - \delta)2^{N(H(X, \hat{X}) - \epsilon)} \leq |\mathcal{A}_{d,\epsilon}^N| \leq 2^{N(H(X, \hat{X}) + \epsilon)}$  as before



# Covering lemma and distortion typical sequences

We say joint typical sequences  $x^N$  and  $\hat{x}^N$  are distortion typical  $((x^N, \hat{x}^N) \in \mathcal{A}_{d,\epsilon}^N)$  if  $|d(x^N, \hat{x}^N) - E[d(X, \hat{X})]| \leq \epsilon$

- By LLN, every pair of sequences sampled from the joint source will virtually be distortion typical
- Consequently,  $(1 - \delta)2^{N(H(X, \hat{X}) - \epsilon)} \leq |\mathcal{A}_{d,\epsilon}^N| \leq 2^{N(H(X, \hat{X}) + \epsilon)}$  as before
- For two independently drawn sequences  $\hat{X}^N$  and  $X^N$ , the probability for them to be distortion typical will be just the same as before. In particular,  $(1 - \delta)2^{-N(I(X; \hat{X}) - 3\epsilon)} \leq Pr((X^N, \hat{X}^N) \in \mathcal{A}_{d,\epsilon}^N(X, \hat{X}))$

# Covering lemma for distortion typical sequences

# Covering lemma for distortion typical sequences

$$\Pr((X^N, \hat{X}^N(m)) \notin \mathcal{A}_{d,\epsilon}^{(N)}(X, \hat{X}) \text{ for all } m)$$

# Covering lemma for distortion typical sequences

$$\begin{aligned} & Pr((X^N, \hat{X}^N(m)) \notin \mathcal{A}_{d,\epsilon}^{(N)}(X, \hat{X}) \text{ for all } m) \\ &= \prod_{m=1}^M Pr((X^N, \hat{X}^N(m)) \notin \mathcal{A}_{d,\epsilon}^{(N)}(\hat{X}, X)) \end{aligned}$$

# Covering lemma for distortion typical sequences

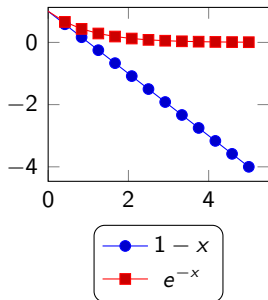
$$\begin{aligned} & Pr((X^N, \hat{X}^N(m)) \notin \mathcal{A}_{d,\epsilon}^{(N)}(X, \hat{X}) \text{ for all } m) \\ &= \prod_{m=1}^M Pr((X^N, \hat{X}^N(m)) \notin \mathcal{A}_{d,\epsilon}^{(N)}(\hat{X}, X)) \\ &= \prod_{m=1}^M \left[ 1 - Pr((X^N, \hat{X}^N(m)) \in \mathcal{A}_{d,\epsilon}^{(N)}(\hat{X}, X)) \right] \end{aligned}$$

# Covering lemma for distortion typical sequences

$$\begin{aligned}
 & Pr((X^N, \hat{X}^N(m)) \notin \mathcal{A}_{d,\epsilon}^{(N)}(X, \hat{X}) \text{ for all } m) \\
 &= \prod_{m=1}^M Pr((X^N, \hat{X}^N(m)) \notin \mathcal{A}_{d,\epsilon}^{(N)}(\hat{X}, X)) \\
 &= \prod_{m=1}^M [1 - Pr((X^N, \hat{X}^N(m)) \in \mathcal{A}_{d,\epsilon}^{(N)}(\hat{X}, X))] \\
 &\leq (1 - (1 - \delta)2^{-N(I(\hat{X};X)+3\epsilon)})^M
 \end{aligned}$$

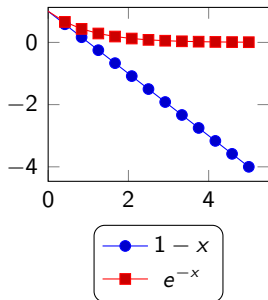
## Covering lemma for distortion typical sequences

$$\begin{aligned}
& Pr((X^N, \hat{X}^N(m)) \notin \mathcal{A}_{d,\epsilon}^{(N)}(X, \hat{X}) \text{ for all } m) \\
&= \prod_{m=1}^M Pr((X^N, \hat{X}^N(m)) \notin \mathcal{A}_{d,\epsilon}^{(N)}(\hat{X}, X)) \\
&= \prod_{m=1}^M [1 - Pr((X^N, \hat{X}^N(m)) \in \mathcal{A}_{d,\epsilon}^{(N)}(\hat{X}, X))] \\
&\leq (1 - (1 - \delta)2^{-N(I(\hat{X};X)+3\epsilon)})^M \\
&\leq \exp(-M(1 - \delta)2^{-N(I(\hat{X};X)+3\epsilon)})
\end{aligned}$$



## Covering lemma for distortion typical sequences

$$\begin{aligned}
& Pr((X^N, \hat{X}^N(m)) \notin \mathcal{A}_{d,\epsilon}^{(N)}(X, \hat{X}) \text{ for all } m) \\
&= \prod_{m=1}^M Pr((X^N, \hat{X}^N(m)) \notin \mathcal{A}_{d,\epsilon}^{(N)}(\hat{X}, X)) \\
&= \prod_{m=1}^M [1 - Pr((X^N, \hat{X}^N(m)) \in \mathcal{A}_{d,\epsilon}^{(N)}(\hat{X}, X))] \\
&\leq (1 - (1 - \delta)2^{-N(I(\hat{X}; X) + 3\epsilon)})^M \\
&\leq \exp(-M(1 - \delta)2^{-N(I(\hat{X}; X) + 3\epsilon)}) \\
&\leq \exp(-(1 - \delta)2^{-N(I(\hat{X}; X) - R + 3\epsilon)}) \rightarrow 0 \text{ as } N \rightarrow \infty \text{ and } R > I(X; \hat{X}) + 3\epsilon
\end{aligned}$$





# Forward proof

## Encoding

Given input  $X^N$ , find out of the codewords the one that is jointly typical with  $X^N$ . And say, if the codeword is  $\mathbf{C}(i)$ , output index  $i$  to the decoder

# Forward proof

## Encoding

Given input  $X^N$ , find out of the codewords the one that is jointly typical with  $X^N$ . And say, if the codeword is  $\mathbf{C}(i)$ , output index  $i$  to the decoder

## Decoding

Upon receiving the index  $i$ , simply output  $\mathbf{C}(i)$

# Forward proof

## Encoding

Given input  $X^N$ , find out of the codewords the one that is jointly typical with  $X^N$ . And say, if the codeword is  $\mathbf{C}(i)$ , output index  $i$  to the decoder

## Decoding

Upon receiving the index  $i$ , simply output  $\mathbf{C}(i)$

## Performance analysis

- First of all, the only point of failure lies on encoding, that is when the encoder cannot find a codeword jointly typical with  $X^N$

# Forward proof

## Encoding

Given input  $X^N$ , find out of the codewords the one that is jointly typical with  $X^N$ . And say, if the codeword is  $\mathbf{C}(i)$ , output index  $i$  to the decoder

## Decoding

Upon receiving the index  $i$ , simply output  $\mathbf{C}(i)$

## Performance analysis

- First of all, the only point of failure lies on encoding, that is when the encoder cannot find a codeword jointly typical with  $X^N$
- By covering Lemma, encoding failure is negligible as long as  $R > I(X; \hat{X})$

# Forward proof

## Encoding

Given input  $X^N$ , find out of the codewords the one that is jointly typical with  $X^N$ . And say, if the codeword is  $\mathbf{C}(i)$ , output index  $i$  to the decoder

## Decoding

Upon receiving the index  $i$ , simply output  $\mathbf{C}(i)$

## Performance analysis

- First of all, the only point of failure lies on encoding, that is when the encoder cannot find a codeword jointly typical with  $X^N$
- By covering Lemma, encoding failure is negligible as long as  $R > I(X; \hat{X})$
- If encoding is successful,  $\mathbf{C}(i)$  and  $X^N$  should be distortion typical. Therefore,  $E[d(\mathbf{C}(i); X^N)] \sim E[d(\hat{X}, X)] \leq \mathcal{D}$  as desired

# Converse proof

## Converse statement

If rate is smaller than  $R(\mathcal{D})$ , distortion will be larger than  $\mathcal{D}$

# Converse proof

## Converse statement

If rate is smaller than  $R(\mathcal{D})$ , distortion will be larger than  $\mathcal{D}$

## Alternative statement

If distortion is less than or equal to  $\mathcal{D}$ , the rate must be larger than  $R(\mathcal{D})$

# Converse proof

## Converse statement

If rate is smaller than  $R(\mathcal{D})$ , distortion will be larger than  $\mathcal{D}$

## Alternative statement

If distortion is less than or equal to  $\mathcal{D}$ , the rate must be larger than  $R(\mathcal{D})$

In the proof, we need to use the convex property of  $R(\mathcal{D})$ . That is,

$$R(a\mathcal{D}_1 + (1 - a)\mathcal{D}_2) \geq aR(\mathcal{D}_1) + (1 - a)R(\mathcal{D}_2)$$

So we will digress a little bit to show this convex property first



# Log-sum inequality

## Log-sum inequality

For any  $a_1, \dots, a_n \geq 0$  and  $b_1, \dots, b_n \geq 0$ , we have

$$\sum_i a_i \log_2 \frac{a_i}{b_i} \geq \left( \sum_i a_i \right) \log_2 \frac{\sum_i a_i}{\sum_i b_i}.$$

# Log-sum inequality

## Log-sum inequality

For any  $a_1, \dots, a_n \geq 0$  and  $b_1, \dots, b_n \geq 0$ , we have

$$\sum_i a_i \log_2 \frac{a_i}{b_i} \geq \left( \sum_i a_i \right) \log_2 \frac{\sum_i a_i}{\sum_i b_i}.$$

## Proof

We can define two distributions  $p(x)$  and  $q(x)$  with  $p(x_i) = \frac{a_i}{\sum_i a_i}$  and  $q(x_i) = \frac{b_i}{\sum_i b_i}$ . Since  $p(x)$  and  $q(x)$  are both non-negative and sum up to 1, they are indeed valid probability mass functions.

# Log-sum inequality

## Log-sum inequality

For any  $a_1, \dots, a_n \geq 0$  and  $b_1, \dots, b_n \geq 0$ , we have

$$\sum_i a_i \log_2 \frac{a_i}{b_i} \geq \left( \sum_i a_i \right) \log_2 \frac{\sum_i a_i}{\sum_i b_i}.$$

## Proof

We can define two distributions  $p(x)$  and  $q(x)$  with  $p(x_i) = \frac{a_i}{\sum_i a_i}$  and  $q(x_i) = \frac{b_i}{\sum_i b_i}$ . Since  $p(x)$  and  $q(x)$  are both non-negative and sum up to 1, they are indeed valid probability mass functions. Then, we have

# Log-sum inequality

## Log-sum inequality

For any  $a_1, \dots, a_n \geq 0$  and  $b_1, \dots, b_n \geq 0$ , we have

$$\sum_i a_i \log_2 \frac{a_i}{b_i} \geq \left( \sum_i a_i \right) \log_2 \frac{\sum_i a_i}{\sum_i b_i}.$$

## Proof

We can define two distributions  $p(x)$  and  $q(x)$  with  $p(x_i) = \frac{a_i}{\sum_i a_i}$  and  $q(x_i) = \frac{b_i}{\sum_i b_i}$ . Since  $p(x)$  and  $q(x)$  are both non-negative and sum up to 1, they are indeed valid probability mass functions. Then, we have

$$0 \leq KL(p(x) \| q(x)) = \sum_i p(x_i) \log_2 \frac{p(x_i)}{q(x_i)}$$

# Log-sum inequality

## Log-sum inequality

For any  $a_1, \dots, a_n \geq 0$  and  $b_1, \dots, b_n \geq 0$ , we have

$$\sum_i a_i \log_2 \frac{a_i}{b_i} \geq \left( \sum_i a_i \right) \log_2 \frac{\sum_i a_i}{\sum_i b_i}.$$

## Proof

We can define two distributions  $p(x)$  and  $q(x)$  with  $p(x_i) = \frac{a_i}{\sum_i a_i}$  and  $q(x_i) = \frac{b_i}{\sum_i b_i}$ . Since  $p(x)$  and  $q(x)$  are both non-negative and sum up to 1, they are indeed valid probability mass functions. Then, we have

$$\begin{aligned} 0 \leq KL(p(x) \| q(x)) &= \sum_i p(x_i) \log_2 \frac{p(x_i)}{q(x_i)} \\ &= \sum_i \frac{a_i}{\sum_i a_i} \left( \log_2 \frac{a_i}{b_i} - \log_2 \frac{\sum_i a_i}{\sum_i b_i} \right) \end{aligned}$$

# Convexity of KL-Divergence

For any four distributions  $p_1(\cdot)$ ,  $p_2(\cdot)$ ,  $q_1(\cdot)$ , and  $q_2(\cdot)$ , we have

$$\lambda_1 KL(p_1 \| q_1) + \lambda_2 KL(p_2 \| q_2) \geq KL(\lambda_1 p_1 + \lambda_2 p_2 \| \lambda_1 q_1 + \lambda_2 q_2),$$

where  $\lambda_1, \lambda_2 \geq 0$  and  $\lambda_1 + \lambda_2 = 1$

# Convexity of KL-Divergence

For any four distributions  $p_1(\cdot)$ ,  $p_2(\cdot)$ ,  $q_1(\cdot)$ , and  $q_2(\cdot)$ , we have

$$\lambda_1 KL(p_1 \| q_1) + \lambda_2 KL(p_2 \| q_2) \geq KL(\lambda_1 p_1 + \lambda_2 p_2 \| \lambda_1 q_1 + \lambda_2 q_2),$$

where  $\lambda_1, \lambda_2 \geq 0$  and  $\lambda_1 + \lambda_2 = 1$

## Proof

$$\begin{aligned} & \lambda_1 KL(p_1 \| q_1) + \lambda_2 KL(p_2 \| q_2) \\ &= \lambda_1 \sum_{x \in \mathcal{X}} p_1(x) \log \frac{p_1(x)}{q_1(x)} + \lambda_2 \sum_{x \in \mathcal{X}} p_2(x) \log \frac{p_2(x)}{q_2(x)} \end{aligned}$$

# Convexity of KL-Divergence

For any four distributions  $p_1(\cdot)$ ,  $p_2(\cdot)$ ,  $q_1(\cdot)$ , and  $q_2(\cdot)$ , we have

$$\lambda_1 KL(p_1 \| q_1) + \lambda_2 KL(p_2 \| q_2) \geq KL(\lambda_1 p_1 + \lambda_2 p_2 \| \lambda_1 q_1 + \lambda_2 q_2),$$

where  $\lambda_1, \lambda_2 \geq 0$  and  $\lambda_1 + \lambda_2 = 1$

## Proof

$$\begin{aligned} & \lambda_1 KL(p_1 \| q_1) + \lambda_2 KL(p_2 \| q_2) \\ &= \lambda_1 \sum_{x \in \mathcal{X}} p_1(x) \log \frac{p_1(x)}{q_1(x)} + \lambda_2 \sum_{x \in \mathcal{X}} p_2(x) \log \frac{p_2(x)}{q_2(x)} \\ &= \sum_{x \in \mathcal{X}} \lambda_1 p_1(x) \log \frac{\lambda_1 p_1(x)}{\lambda_1 q_1(x)} + \lambda_2 p_2(x) \log \frac{\lambda_2 p_2(x)}{\lambda_2 q_2(x)} \end{aligned}$$



# Convexity of KL-Divergence

For any four distributions  $p_1(\cdot)$ ,  $p_2(\cdot)$ ,  $q_1(\cdot)$ , and  $q_2(\cdot)$ , we have

$$\lambda_1 KL(p_1 \| q_1) + \lambda_2 KL(p_2 \| q_2) \geq KL(\lambda_1 p_1 + \lambda_2 p_2 \| \lambda_1 q_1 + \lambda_2 q_2),$$

where  $\lambda_1, \lambda_2 \geq 0$  and  $\lambda_1 + \lambda_2 = 1$

## Proof

$$\begin{aligned} & \lambda_1 KL(p_1 \| q_1) + \lambda_2 KL(p_2 \| q_2) \\ &= \lambda_1 \sum_{x \in \mathcal{X}} p_1(x) \log \frac{p_1(x)}{q_1(x)} + \lambda_2 \sum_{x \in \mathcal{X}} p_2(x) \log \frac{p_2(x)}{q_2(x)} \\ &= \sum_{x \in \mathcal{X}} \lambda_1 p_1(x) \log \frac{\lambda_1 p_1(x)}{\lambda_1 q_1(x)} + \lambda_2 p_2(x) \log \frac{\lambda_2 p_2(x)}{\lambda_2 q_2(x)} \\ &\geq \sum_{x \in \mathcal{X}} (\lambda_1 p_1(x) + \lambda_2 p_2(x)) \log \frac{\lambda_1 p_1(x) + \lambda_2 p_2(x)}{\lambda_1 q_1(x) + \lambda_2 q_2(x)} \quad (\text{by log-sum inequality}) \end{aligned}$$

# Convexity of KL-Divergence

For any four distributions  $p_1(\cdot)$ ,  $p_2(\cdot)$ ,  $q_1(\cdot)$ , and  $q_2(\cdot)$ , we have

$$\lambda_1 KL(p_1 \| q_1) + \lambda_2 KL(p_2 \| q_2) \geq KL(\lambda_1 p_1 + \lambda_2 p_2 \| \lambda_1 q_1 + \lambda_2 q_2),$$

where  $\lambda_1, \lambda_2 \geq 0$  and  $\lambda_1 + \lambda_2 = 1$

## Proof

$$\begin{aligned} & \lambda_1 KL(p_1 \| q_1) + \lambda_2 KL(p_2 \| q_2) \\ &= \lambda_1 \sum_{x \in \mathcal{X}} p_1(x) \log \frac{p_1(x)}{q_1(x)} + \lambda_2 \sum_{x \in \mathcal{X}} p_2(x) \log \frac{p_2(x)}{q_2(x)} \\ &= \sum_{x \in \mathcal{X}} \lambda_1 p_1(x) \log \frac{\lambda_1 p_1(x)}{\lambda_1 q_1(x)} + \lambda_2 p_2(x) \log \frac{\lambda_2 p_2(x)}{\lambda_2 q_2(x)} \\ &\geq \sum_{x \in \mathcal{X}} (\lambda_1 p_1(x) + \lambda_2 p_2(x)) \log \frac{\lambda_1 p_1(x) + \lambda_2 p_2(x)}{\lambda_1 q_1(x) + \lambda_2 q_2(x)} \quad (\text{by log-sum inequality}) \\ &= KL(\lambda_1 p_1 + \lambda_2 p_2 \| \lambda_1 q_1 + \lambda_2 q_2) \end{aligned}$$

# Convexity of $I(X; Y)$ with respect to $p(y|x)$

For any random variables  $X$  and  $Y$ ,  $I(X; Y)$  is a convex function of  $p(y|x)$  for a fixed  $p(x)$

# Convexity of $I(X; Y)$ with respect to $p(y|x)$

For any random variables  $X$  and  $Y$ ,  $I(X; Y)$  is a convex function of  $p(y|x)$  for a fixed  $p(x)$

## Remark

$I(X; Y)$  is concave with respect to  $p(x)$  for fixed  $p(y|x)$  though. A proof is given in Cover and Thomas and will be omitted here

# Convexity of $I(X; Y)$ with respect to $p(y|x)$

For any random variables  $X$  and  $Y$ ,  $I(X; Y)$  is a convex function of  $p(y|x)$  for a fixed  $p(x)$

## Remark

$I(X; Y)$  is concave with respect to  $p(x)$  for fixed  $p(y|x)$  though. A proof is given in Cover and Thomas and will be omitted here

## Proof

Let us write

$$\begin{aligned} I(X; Y) &= KL(p(x, y) \| p(x)p(y)) \\ &= KL\left(p(x)p(y|x) \left\| p(x) \sum_x p(x)p(y|x)\right.\right) \triangleq f(p(y|x)) \end{aligned}$$

# Convexity of $I(X; Y)$ with respect to $p(y|x)$

For any random variables  $X$  and  $Y$ ,  $I(X; Y)$  is a convex function of  $p(y|x)$  for a fixed  $p(x)$

## Remark

$I(X; Y)$  is concave with respect to  $p(x)$  for fixed  $p(y|x)$  though. A proof is given in Cover and Thomas and will be omitted here

## Proof

Let us write

$$\begin{aligned} I(X; Y) &= KL(p(x, y) \| p(x)p(y)) \\ &= KL\left(p(x)p(y|x) \left\| p(x) \sum_x p(x)p(y|x)\right.\right) \triangleq f(p(y|x)) \end{aligned}$$

We want to show

$$\lambda f(p_1(y|x)) + (1 - \lambda)f(p_2(y|x)) \geq f(\lambda p_1(y|x) + (1 - \lambda)p_2(y|x))$$

# Proof

Continue from previous slide, we have

$$\begin{aligned} & \lambda f(p_1(y|x)) + (1 - \lambda)f(p_2(y|x)) \\ &= \lambda KL\left(p(x)p_1(y|x) \parallel p(x) \sum_x p(x)p_1(y|x)\right) \\ & \quad + (1 - \lambda)KL\left(p(x)p_2(y|x) \parallel p(x) \sum_x p(x)p_2(y|x)\right) \end{aligned}$$

## Proof

Continue from previous slide, we have

$$\begin{aligned}
 & \lambda f(p_1(y|x)) + (1 - \lambda)f(p_2(y|x)) \\
 = & \lambda KL\left(p(x)p_1(y|x) \parallel p(x) \sum_x p(x)p_1(y|x)\right) \\
 & + (1 - \lambda)KL\left(p(x)p_2(y|x) \parallel p(x) \sum_x p(x)p_2(y|x)\right) \\
 \geq & KL\left(\lambda p(x)p_1(y|x) + (1 - \lambda)p(x)p_2(y|x) \parallel \lambda p(x) \sum_x p(x)p_1(y|x) \right. \\
 & \left. + (1 - \lambda)p(x) \sum_x p(x)p_2(y|x)\right)
 \end{aligned}$$



## Proof

Continue from previous slide, we have

$$\begin{aligned}
 & \lambda f(p_1(y|x)) + (1 - \lambda)f(p_2(y|x)) \\
 = & \lambda KL\left(p(x)p_1(y|x) \parallel p(x) \sum_x p(x)p_1(y|x)\right) \\
 & + (1 - \lambda)KL\left(p(x)p_2(y|x) \parallel p(x) \sum_x p(x)p_2(y|x)\right) \\
 \geq & KL\left(\lambda p(x)p_1(y|x) + (1 - \lambda)p(x)p_2(y|x) \parallel \lambda p(x) \sum_x p(x)p_1(y|x) \right. \\
 & \left. + (1 - \lambda)p(x) \sum_x p(x)p_2(y|x)\right) \\
 = & KL\left(p(x)[\lambda p_1(y|x) + (1 - \lambda)p_2(y|x)] \parallel p(x) \sum_x p(x)[\lambda p_1(y|x) + (1 - \lambda)p_2(y|x)]\right)
 \end{aligned}$$

## Proof

Continue from previous slide, we have

$$\begin{aligned}
 & \lambda f(p_1(y|x)) + (1 - \lambda)f(p_2(y|x)) \\
 = & \lambda KL\left(p(x)p_1(y|x) \parallel p(x) \sum_x p(x)p_1(y|x)\right) \\
 & + (1 - \lambda)KL\left(p(x)p_2(y|x) \parallel p(x) \sum_x p(x)p_2(y|x)\right) \\
 \geq & KL\left(\lambda p(x)p_1(y|x) + (1 - \lambda)p(x)p_2(y|x) \parallel \lambda p(x) \sum_x p(x)p_1(y|x) \right. \\
 & \left. + (1 - \lambda)p(x) \sum_x p(x)p_2(y|x)\right) \\
 = & KL\left(p(x)[\lambda p_1(y|x) + (1 - \lambda)p_2(y|x)] \parallel p(x) \sum_x p(x)[\lambda p_1(y|x) + (1 - \lambda)p_2(y|x)]\right) \\
 = & f(\lambda p_1(y|x) + (1 - \lambda)p_2(y|x))
 \end{aligned}$$

# Convexity of $R(\mathcal{D})$

Recall that  $R(\mathcal{D}) = \min_{p(\hat{X}|X)} I(\hat{X}; X)$  with  $E[d(X, \hat{X})] \leq \mathcal{D}$

We want to show that

$$R(\lambda\mathcal{D}_1 + (1 - \lambda)\mathcal{D}_2) \leq \lambda R(\mathcal{D}_1) + (1 - \lambda)R(\mathcal{D}_2)$$

# Convexity of $R(\mathcal{D})$

Recall that  $R(\mathcal{D}) = \min_{p(\hat{x}|x)} I(\hat{X}; X)$  with  $E[d(X, \hat{X})] \leq \mathcal{D}$

We want to show that

$$R(\lambda\mathcal{D}_1 + (1 - \lambda)\mathcal{D}_2) \leq \lambda R(\mathcal{D}_1) + (1 - \lambda)R(\mathcal{D}_2)$$

## Proof

Let  $p_1^*(\hat{x}|x)$  and  $p_2^*(\hat{x}|x)$  be the distributions that optimize  $R(\mathcal{D}_1)$  and  $R(\mathcal{D}_2)$ . Let's try to time share between the two distributions.

# Convexity of $R(\mathcal{D})$

Recall that  $R(\mathcal{D}) = \min_{p(\hat{x}|x)} I(\hat{X}; X)$  with  $E[d(X, \hat{X})] \leq \mathcal{D}$

We want to show that

$$R(\lambda\mathcal{D}_1 + (1 - \lambda)\mathcal{D}_2) \leq \lambda R(\mathcal{D}_1) + (1 - \lambda)R(\mathcal{D}_2)$$

## Proof

Let  $p_1^*(\hat{x}|x)$  and  $p_2^*(\hat{x}|x)$  be the distributions that optimize  $R(\mathcal{D}_1)$  and  $R(\mathcal{D}_2)$ . Let's try to time share between the two distributions. That is, using  $p_1^*(\hat{x}|x)$  with  $\lambda$  fraction of time and  $p_2^*(\hat{x}|x)$  with  $(1 - \lambda)$  fraction of time.

# Convexity of $R(\mathcal{D})$

Recall that  $R(\mathcal{D}) = \min_{p(\hat{x}|x)} I(\hat{X}; X)$  with  $E[d(X, \hat{X})] \leq \mathcal{D}$

We want to show that

$$R(\lambda\mathcal{D}_1 + (1 - \lambda)\mathcal{D}_2) \leq \lambda R(\mathcal{D}_1) + (1 - \lambda)R(\mathcal{D}_2)$$

## Proof

Let  $p_1^*(\hat{x}|x)$  and  $p_2^*(\hat{x}|x)$  be the distributions that optimize  $R(\mathcal{D}_1)$  and  $R(\mathcal{D}_2)$ . Let's try to time share between the two distributions. That is, using  $p_1^*(\hat{x}|x)$  with  $\lambda$  fraction of time and  $p_2^*(\hat{x}|x)$  with  $(1 - \lambda)$  fraction of time. The resulting distortion will be  $\lambda\mathcal{D}_1 + (1 - \lambda)\mathcal{D}_2$ .

# Convexity of $R(\mathcal{D})$

Recall that  $R(\mathcal{D}) = \min_{p(\hat{x}|x)} I(\hat{X}; X)$  with  $E[d(X, \hat{X})] \leq \mathcal{D}$

We want to show that

$$R(\lambda\mathcal{D}_1 + (1 - \lambda)\mathcal{D}_2) \leq \lambda R(\mathcal{D}_1) + (1 - \lambda)R(\mathcal{D}_2)$$

## Proof

Let  $p_1^*(\hat{x}|x)$  and  $p_2^*(\hat{x}|x)$  be the distributions that optimize  $R(\mathcal{D}_1)$  and  $R(\mathcal{D}_2)$ . Let's try to time share between the two distributions. That is, using  $p_1^*(\hat{x}|x)$  with  $\lambda$  fraction of time and  $p_2^*(\hat{x}|x)$  with  $(1 - \lambda)$  fraction of time. The resulting distortion will be  $\lambda\mathcal{D}_1 + (1 - \lambda)\mathcal{D}_2$ . Therefore,

$$\lambda R(\mathcal{D}_1) + (1 - \lambda)R(\mathcal{D}_2) = \lambda I(\hat{X}_1; X) + (1 - \lambda)I(\hat{X}_2; X)$$

# Convexity of $R(\mathcal{D})$

Recall that  $R(\mathcal{D}) = \min_{p(\hat{x}|x)} I(\hat{X}; X)$  with  $E[d(X, \hat{X})] \leq \mathcal{D}$

We want to show that

$$R(\lambda\mathcal{D}_1 + (1 - \lambda)\mathcal{D}_2) \leq \lambda R(\mathcal{D}_1) + (1 - \lambda)R(\mathcal{D}_2)$$

## Proof

Let  $p_1^*(\hat{x}|x)$  and  $p_2^*(\hat{x}|x)$  be the distributions that optimize  $R(\mathcal{D}_1)$  and  $R(\mathcal{D}_2)$ . Let's try to time share between the two distributions. That is, using  $p_1^*(\hat{x}|x)$  with  $\lambda$  fraction of time and  $p_2^*(\hat{x}|x)$  with  $(1 - \lambda)$  fraction of time. The resulting distortion will be  $\lambda\mathcal{D}_1 + (1 - \lambda)\mathcal{D}_2$ . Therefore,

$$\begin{aligned} \lambda R(\mathcal{D}_1) + (1 - \lambda)R(\mathcal{D}_2) &= \lambda I(\hat{X}_1; X) + (1 - \lambda)I(\hat{X}_2; X) \\ &= \lambda f(p_1^*(\hat{x}|x)) + (1 - \lambda)f(p_2^*(\hat{x}|x)) \end{aligned}$$



# Convexity of $R(\mathcal{D})$

Recall that  $R(\mathcal{D}) = \min_{p(\hat{x}|x)} I(\hat{X}; X)$  with  $E[d(X, \hat{X})] \leq \mathcal{D}$

We want to show that

$$R(\lambda\mathcal{D}_1 + (1 - \lambda)\mathcal{D}_2) \leq \lambda R(\mathcal{D}_1) + (1 - \lambda)R(\mathcal{D}_2)$$

## Proof

Let  $p_1^*(\hat{x}|x)$  and  $p_2^*(\hat{x}|x)$  be the distributions that optimize  $R(\mathcal{D}_1)$  and  $R(\mathcal{D}_2)$ . Let's try to time share between the two distributions. That is, using  $p_1^*(\hat{x}|x)$  with  $\lambda$  fraction of time and  $p_2^*(\hat{x}|x)$  with  $(1 - \lambda)$  fraction of time. The resulting distortion will be  $\lambda\mathcal{D}_1 + (1 - \lambda)\mathcal{D}_2$ . Therefore,

$$\begin{aligned} \lambda R(\mathcal{D}_1) + (1 - \lambda)R(\mathcal{D}_2) &= \lambda I(\hat{X}_1; X) + (1 - \lambda)I(\hat{X}_2; X) \\ &= \lambda f(p_1^*(\hat{x}|x)) + (1 - \lambda)f(p_2^*(\hat{x}|x)) \geq f(\lambda p_1^*(\hat{x}|x) + (1 - \lambda)p_2^*(\hat{x}|x)) \end{aligned}$$

# Convexity of $R(\mathcal{D})$

Recall that  $R(\mathcal{D}) = \min_{p(\hat{x}|x)} I(\hat{X}; X)$  with  $E[d(X, \hat{X})] \leq \mathcal{D}$

We want to show that

$$R(\lambda\mathcal{D}_1 + (1 - \lambda)\mathcal{D}_2) \leq \lambda R(\mathcal{D}_1) + (1 - \lambda)R(\mathcal{D}_2)$$

## Proof

Let  $p_1^*(\hat{x}|x)$  and  $p_2^*(\hat{x}|x)$  be the distributions that optimize  $R(\mathcal{D}_1)$  and  $R(\mathcal{D}_2)$ . Let's try to time share between the two distributions. That is, using  $p_1^*(\hat{x}|x)$  with  $\lambda$  fraction of time and  $p_2^*(\hat{x}|x)$  with  $(1 - \lambda)$  fraction of time. The resulting distortion will be  $\lambda\mathcal{D}_1 + (1 - \lambda)\mathcal{D}_2$ . Therefore,

$$\begin{aligned} \lambda R(\mathcal{D}_1) + (1 - \lambda)R(\mathcal{D}_2) &= \lambda I(\hat{X}_1; X) + (1 - \lambda)I(\hat{X}_2; X) \\ &= \lambda f(p_1^*(\hat{x}|x)) + (1 - \lambda)f(p_2^*(\hat{x}|x)) \geq f(\lambda p_1^*(\hat{x}|x) + (1 - \lambda)p_2^*(\hat{x}|x)) \\ &= I(\tilde{X}; X) \end{aligned}$$

# Convexity of $R(\mathcal{D})$

Recall that  $R(\mathcal{D}) = \min_{p(\hat{x}|x)} I(\hat{X}; X)$  with  $E[d(X, \hat{X})] \leq \mathcal{D}$

We want to show that

$$R(\lambda\mathcal{D}_1 + (1 - \lambda)\mathcal{D}_2) \leq \lambda R(\mathcal{D}_1) + (1 - \lambda)R(\mathcal{D}_2)$$

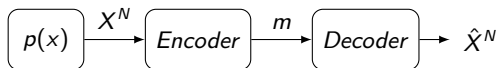
## Proof

Let  $p_1^*(\hat{x}|x)$  and  $p_2^*(\hat{x}|x)$  be the distributions that optimize  $R(\mathcal{D}_1)$  and  $R(\mathcal{D}_2)$ . Let's try to time share between the two distributions. That is, using  $p_1^*(\hat{x}|x)$  with  $\lambda$  fraction of time and  $p_2^*(\hat{x}|x)$  with  $(1 - \lambda)$  fraction of time. The resulting distortion will be  $\lambda\mathcal{D}_1 + (1 - \lambda)\mathcal{D}_2$ . Therefore,

$$\begin{aligned} \lambda R(\mathcal{D}_1) + (1 - \lambda)R(\mathcal{D}_2) &= \lambda I(\hat{X}_1; X) + (1 - \lambda)I(\hat{X}_2; X) \\ &= \lambda f(p_1^*(\hat{x}|x)) + (1 - \lambda)f(p_2^*(\hat{x}|x)) \geq f(\lambda p_1^*(\hat{x}|x) + (1 - \lambda)p_2^*(\hat{x}|x)) \\ &= I(\tilde{X}; X) \geq R(\lambda\mathcal{D}_1 + (1 - \lambda)\mathcal{D}_2), \end{aligned}$$

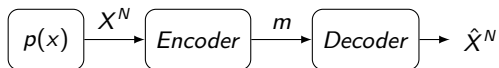
where  $\tilde{X} = \begin{cases} \hat{X}_1 & \text{with } \lambda \text{ fraction of time} \\ \hat{X}_2 & \text{with } (1 - \lambda) \text{ fraction of time} \end{cases}$

# Converse proof



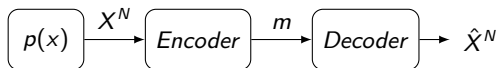
$H(M)$

# Converse proof



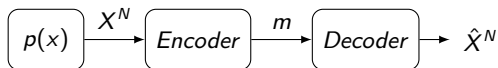
$$H(M) \geq H(M) - H(M|X^N) = I(M; X^N)$$

## Converse proof



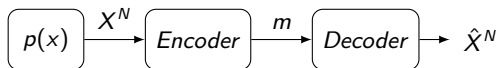
$$\begin{aligned} H(M) &\geq H(M) - H(M|X^N) = I(M; X^N) \geq I(\hat{X}^N; X^N) \\ &= H(X^N) - H(X^N|\hat{X}^N) \end{aligned}$$

# Converse proof



$$\begin{aligned}
 H(M) &\geq H(M) - H(M|X^N) = I(M; X^N) \geq I(\hat{X}^N; X^N) \\
 &= H(X^N) - H(X^N|\hat{X}^N) = \sum_{i=1}^N H(X_i) - \sum_{i=1}^N H(X_i|\hat{X}^N, X^{i-1})
 \end{aligned}$$

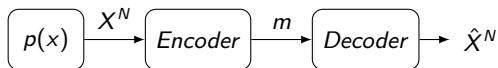
# Converse proof



$$\begin{aligned}
 H(M) &\geq H(M) - H(M|X^N) = I(M; X^N) \geq I(\hat{X}^N; X^N) \\
 &= H(X^N) - H(X^N|\hat{X}^N) = \sum_{i=1}^N H(X_i) - \sum_{i=1}^N H(X_i|\hat{X}^N, X^{i-1}) \\
 &\geq \sum_{i=1}^N H(X_i) - \sum_{i=1}^N H(X_i|\hat{X}_i) = \sum_{i=1}^N I(X_i; \hat{X}_i)
 \end{aligned}$$

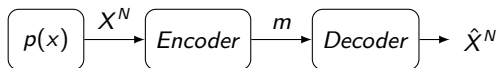


# Converse proof



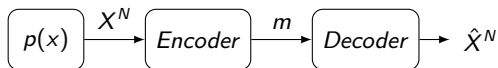
$$\begin{aligned}
 H(M) &\geq H(M) - H(M|X^N) = I(M; X^N) \geq I(\hat{X}^N; X^N) \\
 &= H(X^N) - H(X^N|\hat{X}^N) = \sum_{i=1}^N H(X_i) - \sum_{i=1}^N H(X_i|\hat{X}^N, X^{i-1}) \\
 &\geq \sum_{i=1}^N H(X_i) - \sum_{i=1}^N H(X_i|\hat{X}_i) = \sum_{i=1}^N I(X_i; \hat{X}_i) \\
 &\geq \sum_{i=1}^N R(E[d(X_i, \hat{X}_i)]) = N \left( \frac{1}{N} \sum_{i=1}^N R(E[d(X_i, \hat{X}_i)]) \right)
 \end{aligned}$$

## Converse proof



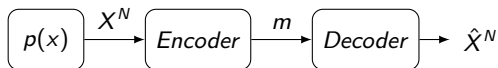
$$\begin{aligned}
 H(M) &\geq H(M) - H(M|X^N) = I(M; X^N) \geq I(\hat{X}^N; X^N) \\
 &= H(X^N) - H(X^N|\hat{X}^N) = \sum_{i=1}^N H(X_i) - \sum_{i=1}^N H(X_i|\hat{X}^N, X^{i-1}) \\
 &\geq \sum_{i=1}^N H(X_i) - \sum_{i=1}^N H(X_i|\hat{X}_i) = \sum_{i=1}^N I(X_i; \hat{X}_i) \\
 &\geq \sum_{i=1}^N R(E[d(X_i, \hat{X}_i)]) = N \left( \frac{1}{N} \sum_{i=1}^N R(E[d(X_i, \hat{X}_i)]) \right) \\
 &\geq NR \left( \frac{1}{N} \sum_{i=1}^N E[d(X_i, \hat{X}_i)] \right)
 \end{aligned}$$

## Converse proof



$$\begin{aligned}
 H(M) &\geq H(M) - H(M|X^N) = I(M; X^N) \geq I(\hat{X}^N; X^N) \\
 &= H(X^N) - H(X^N|\hat{X}^N) = \sum_{i=1}^N H(X_i) - \sum_{i=1}^N H(X_i|\hat{X}^N, X^{i-1}) \\
 &\geq \sum_{i=1}^N H(X_i) - \sum_{i=1}^N H(X_i|\hat{X}_i) = \sum_{i=1}^N I(X_i; \hat{X}_i) \\
 &\geq \sum_{i=1}^N R(E[d(X_i, \hat{X}_i)]) = N \left( \frac{1}{N} \sum_{i=1}^N R(E[d(X_i, \hat{X}_i)]) \right) \\
 &\geq NR \left( \frac{1}{N} \sum_{i=1}^N E[d(X_i, \hat{X}_i)] \right) = NR \left( E \left[ \frac{1}{N} \sum_{i=1}^N d(X_i, \hat{X}_i) \right] \right)
 \end{aligned}$$

# Converse proof



$$\begin{aligned}
 H(M) &\geq H(M) - H(M|X^N) = I(M; X^N) \geq I(\hat{X}^N; X^N) \\
 &= H(X^N) - H(X^N|\hat{X}^N) = \sum_{i=1}^N H(X_i) - \sum_{i=1}^N H(X_i|\hat{X}^N, X^{i-1}) \\
 &\geq \sum_{i=1}^N H(X_i) - \sum_{i=1}^N H(X_i|\hat{X}_i) = \sum_{i=1}^N I(X_i; \hat{X}_i) \\
 &\geq \sum_{i=1}^N R(E[d(X_i, \hat{X}_i)]) = N \left( \frac{1}{N} \sum_{i=1}^N R(E[d(X_i, \hat{X}_i)]) \right) \\
 &\geq NR \left( \frac{1}{N} \sum_{i=1}^N E[d(X_i; \hat{X}_i)] \right) = NR \left( E \left[ \frac{1}{N} \sum_{i=1}^N d(X_i; \hat{X}_i) \right] \right) \\
 &= NR(E[d(X^N; \hat{X}^N)]) \geq NR(D)
 \end{aligned}$$

# Previously...

- Forward and converse proof of the rate-distortion theorem

# This time

- Method of types
- Universal source coding
- Large deviation theory

# Project presentation

- Start as usual class time (12/12)
- Please prepare  $\sim 30$  minutes presentation. Explain your problem statement. Focus on your approach and result
  - Take a format similar to a conference presentation
- Expect  $\sim 5$  minutes Q/A
- Grading
  - Presentation: clarity, structure, references, etc. (10/40)
  - Technical: correctness, depth, novelty, etc. (15/40)
  - Evaluation and results: sound evaluation metric, thoroughness in analysis and experimentation (if any), results and performance (15/40)
- Expectation
  - National conference quality (4/4), reserach day quality (3/4), research meeting quality (2/4), just show up (1/4)

# Motivation

- In previous lectures, we have introduced LLN and typical sequences. In a sense that every sequences drawn from a discrete memoryless source is typical



# Motivation

- In previous lectures, we have introduced LLN and typical sequences. In a sense that every sequences drawn from a discrete memoryless source is typical
- Take coin tossing as example again, if  $Pr(\text{Head}) = 0.6$ , and we throw the coin 1000 times. We expect that almost all drawn sequences with have about 600 heads. And the rest have negligible probability

# Motivation

- In previous lectures, we have introduced LLN and typical sequences. In a sense that every sequences drawn from a discrete memoryless source is typical
- Take coin tossing as example again, if  $Pr(\text{Head}) = 0.6$ , and we throw the coin 1000 times. We expect that almost all drawn sequences with have about 600 heads. And the rest have negligible probability
- However, sometimes we are interested in the probability of getting say 400 heads, even though we know that the probability is negligible

# Motivation

- In previous lectures, we have introduced LLN and typical sequences. In a sense that every sequences drawn from a discrete memoryless source is typical
- Take coin tossing as example again, if  $Pr(\text{Head}) = 0.6$ , and we throw the coin 1000 times. We expect that almost all drawn sequences with have about 600 heads. And the rest have negligible probability
- However, sometimes we are interested in the probability of getting say 400 heads, even though we know that the probability is negligible  $\rightarrow$  method of types

# Motivation

By the end of the class, we will be able to solve the following nontrivial puzzle

- Tom throws a unbiased dice for 10,000 times and adds all values

# Motivation

By the end of the class, we will be able to solve the following nontrivial puzzle

- Tom throws a unbiased dice for 10,000 times and adds all values
- For whatever reason, he is not happy until the sum is at least 40,000. If not, he will just throw the dice again for 10,000

# Motivation

By the end of the class, we will be able to solve the following nontrivial puzzle

- Tom throws a unbiased dice for 10,000 times and adds all values
- For whatever reason, he is not happy until the sum is at least 40,000. If not, he will just throw the dice again for 10,000
- Now, by the time he eventually got a sequence with sum at least 40,000, *approximately how many ones in the sequence?*

# Type class

Continue with the coin-tossing example

- Recall that the probability of getting a particular sequence with 600 heads is

$$0.6^{600}0.4^{400}$$

# Type class

Continue with the coin-tossing example

- Recall that the probability of getting a particular sequence with 600 heads is

$$0.6^{600}0.4^{400} = 2^{-1000(-0.6 \log 0.6 - 0.4 \log 0.4)}$$



# Type class

Continue with the coin-tossing example

- Recall that the probability of getting a particular sequence with 600 heads is

$$0.6^{600}0.4^{400} = 2^{-1000(-0.6 \log 0.6 - 0.4 \log 0.4)} = 2^{-NH(X)}$$

# Type class

Continue with the coin-tossing example

- Recall that the probability of getting a particular sequence with 600 heads is

$$0.6^{600}0.4^{400} = 2^{-1000(-0.6 \log 0.6 - 0.4 \log 0.4)} = 2^{-NH(X)}$$

- How about the probability of getting a particular sequence with 400 heads? It is

$$0.6^{400}0.4^{600} = 2^{-1000(-0.4 \log 0.6 - 0.6 \log 0.4)}$$

# Type class

Continue with the coin-tossing example

- Recall that the probability of getting a particular sequence with 600 heads is

$$0.6^{600}0.4^{400} = 2^{-1000(-0.6 \log 0.6 - 0.4 \log 0.4)} = 2^{-NH(X)}$$

- How about the probability of getting a particular sequence with 400 heads? It is

$$\begin{aligned} 0.6^{400}0.4^{600} &= 2^{-1000(-0.4 \log 0.6 - 0.6 \log 0.4)} \\ &= 2^{-1000(-0.4 \log 0.4 - 0.6 \log 0.6 + 0.4 \log \frac{0.4}{0.6} + 0.6 \log \frac{0.6}{0.4})} \end{aligned}$$

# Type class

Continue with the coin-tossing example

- Recall that the probability of getting a particular sequence with 600 heads is

$$0.6^{600}0.4^{400} = 2^{-1000(-0.6 \log 0.6 - 0.4 \log 0.4)} = 2^{-NH(X)}$$

- How about the probability of getting a particular sequence with 400 heads? It is

$$\begin{aligned} 0.6^{400}0.4^{600} &= 2^{-1000(-0.4 \log 0.6 - 0.6 \log 0.4)} \\ &= 2^{-1000(-0.4 \log 0.4 - 0.6 \log 0.6 + 0.4 \log \frac{0.4}{0.6} + 0.6 \log \frac{0.6}{0.4})} \\ &= 2^{-N(H(X) + KL((0.4, 0.6) || (0.6, 0.4)))} \end{aligned}$$

# Type class

Continue with the coin-tossing example

- Recall that the probability of getting a particular sequence with 600 heads is

$$0.6^{600}0.4^{400} = 2^{-1000(-0.6 \log 0.6 - 0.4 \log 0.4)} = 2^{-NH(X)}$$

- How about the probability of getting a particular sequence with 400 heads? It is

$$\begin{aligned} 0.6^{400}0.4^{600} &= 2^{-1000(-0.4 \log 0.6 - 0.6 \log 0.4)} \\ &= 2^{-1000(-0.4 \log 0.4 - 0.6 \log 0.6 + 0.4 \log \frac{0.4}{0.6} + 0.6 \log \frac{0.6}{0.4})} \\ &= 2^{-N(H(X) + KL((0.4, 0.6) || (0.6, 0.4)))} \end{aligned}$$

- Every sequence with 400 heads has the same probability. And in general, sequences with the same fraction of outcomes have same probability and we can put them into the same **(type) class**

# Type class

- For convenience, let us denote the number of  $a$  in the sequence  $x^N$  as  $\mathcal{N}(a|x^N)$

# Type class

- For convenience, let us denote the number of  $a$  in the sequence  $x^N$  as  $\mathcal{N}(a|x^N)$
- Then for any valid distribution of  $X$ ,  $p(x)$ , we will define a type class  $T(p_X)$  as the set containing all sequences such that  $\frac{\mathcal{N}(a|x^N)}{N} \approx p(a)$ ,  $\forall a \in \mathcal{X}$

# Type class

- For convenience, let us denote the number of  $a$  in the sequence  $x^N$  as  $\mathcal{N}(a|x^N)$
- Then for any valid distribution of  $X$ ,  $p(x)$ , we will define a type class  $T(p_X)$  as the set containing all sequences such that  $\frac{\mathcal{N}(a|x^N)}{N} \approx p(a)$ ,  $\forall a \in \mathcal{X}$
- Let us reserve  $q(x)$  as the true distribution of  $x$  (i.e.,  $q(\text{Head}) = 0.6$  and  $q(\text{Tail}) = 0.4$ ). And in general, we expect all sequences drawn from the source should belong to  $T(q)$  asymptotically



# Type class

- For convenience, let us denote the number of  $a$  in the sequence  $x^N$  as  $\mathcal{N}(a|x^N)$
- Then for any valid distribution of  $X$ ,  $p(x)$ , we will define a type class  $T(p_X)$  as the set containing all sequences such that  $\frac{\mathcal{N}(a|x^N)}{N} \approx p(a)$ ,  $\forall a \in \mathcal{X}$
- Let us reserve  $q(x)$  as the true distribution of  $x$  (i.e.,  $q(\text{Head}) = 0.6$  and  $q(\text{Tail}) = 0.4$ ). And in general, we expect all sequences drawn from the source should belong to  $T(q)$  asymptotically
- Let's also refer  $p_{x^N}$  as the empirical distribution of  $x^N$ . That is  $p_{x^N}(a) = \frac{\mathcal{N}(a|x^N)}{N}$ . So  $T(p_{x^N})$  is the type class containing  $x^N$

# Example

Let  $\mathcal{X} \in \{1, 2, 3\}$  and  $x^N = 11321$

- $p_{x^N}(1) = \frac{3}{5}$ ,

# Example

Let  $\mathcal{X} \in \{1, 2, 3\}$  and  $x^N = 11321$

- $p_{x^N}(1) = \frac{3}{5}$ ,  $p_{x^N}(2) = \frac{1}{5}$ ,  $p_{x^N}(3) = \frac{1}{5}$

# Example

Let  $\mathcal{X} \in \{1, 2, 3\}$  and  $x^N = 11321$

- $p_{x^N}(1) = \frac{3}{5}$ ,  $p_{x^N}(2) = \frac{1}{5}$ ,  $p_{x^N}(3) = \frac{1}{5}$
- $T(p_{x^N}) = \{11123, 11132, 11231, 11321, \dots\}$  containing all sequences with three 1's, one 2, and one 3

# Example

Let  $\mathcal{X} \in \{1, 2, 3\}$  and  $x^N = 11321$

- $p_{x^N}(1) = \frac{3}{5}$ ,  $p_{x^N}(2) = \frac{1}{5}$ ,  $p_{x^N}(3) = \frac{1}{5}$
- $T(p_{x^N}) = \{11123, 11132, 11231, 11321, \dots\}$  containing all sequences with three 1's, one 2, and one 3
- $|T(p_{x^N})| = \frac{5!}{3!1!1!} = 20$ .

# Example

Let  $\mathcal{X} \in \{1, 2, 3\}$  and  $x^N = 11321$

- $p_{x^N}(1) = \frac{3}{5}$ ,  $p_{x^N}(2) = \frac{1}{5}$ ,  $p_{x^N}(3) = \frac{1}{5}$
- $T(p_{x^N}) = \{11123, 11132, 11231, 11321, \dots\}$  containing all sequences with three 1's, one 2, and one 3
- $|T(p_{x^N})| = \frac{5!}{3!1!1!} = 20$ . In general,

$$|T(p)| = \frac{N!}{(Np(x_1))!(Np(x_2))!(Np(x_3))! \dots}$$

# Example

Let  $\mathcal{X} \in \{1, 2, 3\}$  and  $x^N = 11321$

- $p_{x^N}(1) = \frac{3}{5}$ ,  $p_{x^N}(2) = \frac{1}{5}$ ,  $p_{x^N}(3) = \frac{1}{5}$
- $T(p_{x^N}) = \{11123, 11132, 11231, 11321, \dots\}$  containing all sequences with three 1's, one 2, and one 3
- $|T(p_{x^N})| = \frac{5!}{3!1!1!} = 20$ . In general,

$$|T(p)| = \frac{N!}{(Np(x_1))!(Np(x_2))!(Np(x_3))! \dots}$$

Actually we don't care too much what  $|T(p)|$  is exactly. We will provide bounds for  $|T(p)|$  as we come back later on

# Example

Let  $\mathcal{X} \in \{1, 2, 3\}$  and  $x^N = 11321$

- $p_{x^N}(1) = \frac{3}{5}$ ,  $p_{x^N}(2) = \frac{1}{5}$ ,  $p_{x^N}(3) = \frac{1}{5}$
- $T(p_{x^N}) = \{11123, 11132, 11231, 11321, \dots\}$  containing all sequences with three 1's, one 2, and one 3
- $|T(p_{x^N})| = \frac{5!}{3!1!1!} = 20$ . In general,

$$|T(p)| = \frac{N!}{(Np(x_1))!(Np(x_2))!(Np(x_3))! \dots}$$

Actually we don't care too much what  $|T(p)|$  is exactly. We will provide bounds for  $|T(p)|$  as we come back later on

- And for any sequence  $\mathbf{y}$  in  $T(p_{x^N})$ ,  $p(\mathbf{y}) = q(1)^3 q(2) q(3)$ , where  $q(\cdot)$  is the true distribution



# Type sequence probability

Even though we have seen that in the coin toss example, let's restate it more formally.

## Theorem 1

If  $x^N \in \mathcal{T}(p)$  and  $q(\cdot)$  is the true distribution of  $X$ , the probability of getting  $x^N$  from sampling  $q(\cdot)$  for  $N$  times, as denoted as  $q^N(x^N)$ , is given by

$$2^{-N(H(p)+KL(p||q))}$$

# Type sequence probability

Even though we have seen that in the coin toss example, let's restate it more formally.

## Theorem 1

If  $x^N \in \mathcal{T}(p)$  and  $q(\cdot)$  is the true distribution of  $X$ , the probability of getting  $x^N$  from sampling  $q(\cdot)$  for  $N$  times, as denoted as  $q^N(x^N)$ , is given by

$$2^{-N(H(p)+KL(p||q))}$$

## Proof

$$q^N(x^N) = \prod_{i=1}^N q(x_i) = 2^{\sum_{i=1}^N \log q(x_i)}$$

# Type sequence probability

Even though we have seen that in the coin toss example, let's restate it more formally.

## Theorem 1

If  $x^N \in \mathcal{T}(p)$  and  $q(\cdot)$  is the true distribution of  $X$ , the probability of getting  $x^N$  from sampling  $q(\cdot)$  for  $N$  times, as denoted as  $q^N(x^N)$ , is given by

$$2^{-N(H(p)+KL(p||q))}$$

## Proof

$$q^N(x^N) = \prod_{i=1}^N q(x_i) = 2^{\sum_{i=1}^N \log q(x_i)} = 2^{\sum_{a \in \mathcal{X}} \mathcal{N}(a|x^N) \log q(a)}$$

# Type sequence probability

Even though we have seen that in the coin toss example, let's restate it more formally.

## Theorem 1

If  $x^N \in \mathcal{T}(p)$  and  $q(\cdot)$  is the true distribution of  $X$ , the probability of getting  $x^N$  from sampling  $q(\cdot)$  for  $N$  times, as denoted as  $q^N(x^N)$ , is given by

$$2^{-N(H(p)+KL(p||q))}$$

## Proof

$$\begin{aligned} q^N(x^N) &= \prod_{i=1}^N q(x_i) = 2^{\sum_{i=1}^N \log q(x_i)} = 2^{\sum_{a \in \mathcal{X}} \mathcal{N}(a|x^N) \log q(a)} \\ &= 2^{-N \sum_{a \in \mathcal{X}} -p_{x^N}(a) \log q(a)} \end{aligned}$$

# Type sequence probability

Even though we have seen that in the coin toss example, let's restate it more formally.

## Theorem 1

If  $x^N \in \mathcal{T}(p)$  and  $q(\cdot)$  is the true distribution of  $X$ , the probability of getting  $x^N$  from sampling  $q(\cdot)$  for  $N$  times, as denoted as  $q^N(x^N)$ , is given by

$$2^{-N(H(p)+KL(p||q))}$$

## Proof

$$\begin{aligned} q^N(x^N) &= \prod_{i=1}^N q(x_i) = 2^{\sum_{i=1}^N \log q(x_i)} = 2^{\sum_{a \in \mathcal{X}} \mathcal{N}(a|x^N) \log q(a)} \\ &= 2^{-N \sum_{a \in \mathcal{X}} p_{x^N}(a) \log q(a)} = 2^{-N \left( - \sum_{a \in \mathcal{X}} p(a) \log p(a) - \sum_{a \in \mathcal{X}} p(a) \log \frac{p(a)}{q(a)} \right)} \end{aligned}$$

# Type sequence probability

Even though we have seen that in the coin toss example, let's restate it more formally.

## Theorem 1

If  $x^N \in \mathcal{T}(p)$  and  $q(\cdot)$  is the true distribution of  $X$ , the probability of getting  $x^N$  from sampling  $q(\cdot)$  for  $N$  times, as denoted as  $q^N(x^N)$ , is given by

$$2^{-N(H(p)+KL(p||q))}$$

## Proof

$$\begin{aligned} q^N(x^N) &= \prod_{i=1}^N q(x_i) = 2^{\sum_{i=1}^N \log q(x_i)} = 2^{\sum_{a \in \mathcal{X}} \mathcal{N}(a|x^N) \log q(a)} \\ &= 2^{-N \sum_{a \in \mathcal{X}} p_{x^N}(a) \log q(a)} = 2^{-N \left( - \sum_{a \in \mathcal{X}} p(a) \log p(a) - \sum_{a \in \mathcal{X}} p(a) \log \frac{p(a)}{q(a)} \right)} \\ &= 2^{-N(H(p)+KL(p||q))} \end{aligned}$$

# Probability of a sequence in the “typical” class

If  $x^N \in T(q)$ , where  $q(\cdot)$  is the true distribution of  $X$ , then

$$q^N(x^N) = 2^{-NH(q)} = 2^{-NH(X)}$$

# Probability of a sequence in the “typical” class

If  $x^N \in T(q)$ , where  $q(\cdot)$  is the true distribution of  $X$ , then

$$q^N(x^N) = 2^{-NH(q)} = 2^{-NH(X)}$$

## Remarks

- Note that the probability is exactly equal to  $2^{-NH(X)}$



# Probability of a sequence in the “typical” class

If  $x^N \in T(q)$ , where  $q(\cdot)$  is the true distribution of  $X$ , then

$$q^N(x^N) = 2^{-NH(q)} = 2^{-NH(X)}$$

## Remarks

- Note that the probability is exactly equal to  $2^{-NH(X)}$
- Recall that this is the probability of a typical sequence supposed to be. Therefore, any  $x^N$  in  $T(q)$  is a typical sequence ( $T(q) \subset A_\epsilon^N(X)$ )

# Set of all empirical distribution $\mathcal{P}_N(X)$

Denote  $\mathcal{P}_N(X)$  as the set of all empirical distribution of  $X$  in a length- $N$  sequence

# Set of all empirical distribution $\mathcal{P}_N(X)$

Denote  $\mathcal{P}_N(X)$  as the set of all empirical distribution of  $X$  in a length- $N$  sequence

## Example

If  $X \in \{0, 1\}$ ,

$$\mathcal{P}_N(X) = \left\{ (p_X(0), p_X(1)) : \left( \frac{0}{N}, \frac{N}{N} \right), \left( \frac{1}{N}, \frac{N-1}{N} \right), \dots, \left( \frac{N}{N}, \frac{0}{N} \right) \right\}$$

Note that  $|\mathcal{P}_N(X)| = N + 1$

# Set of all empirical distribution $\mathcal{P}_N(X)$

Denote  $\mathcal{P}_N(X)$  as the set of all empirical distribution of  $X$  in a length- $N$  sequence

## Example

If  $X \in \{0, 1\}$ ,

$$\mathcal{P}_N(X) = \left\{ (p_X(0), p_X(1)) : \left( \frac{0}{N}, \frac{N}{N} \right), \left( \frac{1}{N}, \frac{N-1}{N} \right), \dots, \left( \frac{N}{N}, \frac{0}{N} \right) \right\}$$

Note that  $|\mathcal{P}_N(X)| = N + 1$

- Since a type is uniquely characterized by a distribution of  $X$  in a length- $N$  sequence

# Set of all empirical distribution $\mathcal{P}_N(X)$

Denote  $\mathcal{P}_N(X)$  as the set of all empirical distribution of  $X$  in a length- $N$  sequence

## Example

If  $X \in \{0, 1\}$ ,

$$\mathcal{P}_N(X) = \left\{ (p_X(0), p_X(1)) : \left( \frac{0}{N}, \frac{N}{N} \right), \left( \frac{1}{N}, \frac{N-1}{N} \right), \dots, \left( \frac{N}{N}, \frac{0}{N} \right) \right\}$$

Note that  $|\mathcal{P}_N(X)| = N + 1$

- Since a type is uniquely characterized by a distribution of  $X$  in a length- $N$  sequence
- Each element  $p$  of  $\mathcal{P}_N(X)$  corresponds a type  $T(p)$

# Set of all empirical distribution $\mathcal{P}_N(X)$

Denote  $\mathcal{P}_N(X)$  as the set of all empirical distribution of  $X$  in a length- $N$  sequence

## Example

If  $X \in \{0, 1\}$ ,

$$\mathcal{P}_N(X) = \left\{ (p_X(0), p_X(1)) : \left( \frac{0}{N}, \frac{N}{N} \right), \left( \frac{1}{N}, \frac{N-1}{N} \right), \dots, \left( \frac{N}{N}, \frac{0}{N} \right) \right\}$$

Note that  $|\mathcal{P}_N(X)| = N + 1$

- Since a type is uniquely characterized by a distribution of  $X$  in a length- $N$  sequence
- Each element  $p$  of  $\mathcal{P}_N(X)$  corresponds a type  $T(p)$
- Number of types is  $|\mathcal{P}_N(X)|$

# Number of types

It is not too difficult to count the exact number of types. But in practice, we don't quite bother with it as long as we know that the number is relatively "small"

## Theorem 2

$$|\mathcal{P}_N(X)| \leq (N + 1)^{|X|}$$

# Number of types

It is not too difficult to count the exact number of types. But in practice, we don't quite bother with it as long as we know that the number is relatively "small"

## Theorem 2

$$|\mathcal{P}_N(X)| \leq (N + 1)^{|X|}$$

## Proof

Note that each type is specified by the empirical probability of each outcome of  $X$ . And the possible values of the empirical probabilities are  $\frac{0}{N}, \frac{1}{N}, \dots, \frac{N}{N}$  ( $N + 1$  of them).



# Number of types

It is not too difficult to count the exact number of types. But in practice, we don't quite bother with it as long as we know that the number is relatively "small"

## Theorem 2

$$|\mathcal{P}_N(X)| \leq (N + 1)^{|\mathcal{X}|}$$

## Proof

Note that each type is specified by the empirical probability of each outcome of  $X$ . And the possible values of the empirical probabilities are  $\frac{0}{N}, \frac{1}{N}, \dots, \frac{N}{N}$  ( $N + 1$  of them). Since there are  $|\mathcal{X}|$  elements, the number of types is bounded by  $(N + 1)^{|\mathcal{X}|}$

# Size of a type class

Recall that  $|T(p)| = \frac{N!}{(Np(x_1))!(Np(x_2))!(Np(x_3))! \dots}$  but the following bounds are much more useful in practice

## Theorem 3

$$\frac{1}{(N+1)^{|\mathcal{X}|}} 2^{NH(p)} \leq |T(p)| \leq 2^{NH(p)}$$

# Size of a type class

Recall that  $|T(p)| = \frac{N!}{(Np(x_1))!(Np(x_2))!(Np(x_3))! \dots}$  but the following bounds are much more useful in practice

## Theorem 3

$$\frac{1}{(N+1)^{|\mathcal{X}|}} 2^{NH(p)} \leq |T(p)| \leq 2^{NH(p)}$$

## Proof

Let's assume  $p(\cdot)$  is the actual distribution of  $X$  here

$$1 \geq \sum_{x^N \in T(p)} p^N(x^N)$$

# Size of a type class

Recall that  $|T(p)| = \frac{N!}{(Np(x_1))!(Np(x_2))!(Np(x_3))! \dots}$  but the following bounds are much more useful in practice

## Theorem 3

$$\frac{1}{(N+1)^{|\mathcal{X}|}} 2^{NH(p)} \leq |T(p)| \leq 2^{NH(p)}$$

## Proof

Let's assume  $p(\cdot)$  is the actual distribution of  $X$  here

$$1 \geq \sum_{x^N \in T(p)} p^N(x^N) = \sum_{x^N \in T(p)} 2^{-NH(p)} = |T(p)| 2^{-NH(p)}$$

# Size of a type class

Recall that  $|T(p)| = \frac{N!}{(Np(x_1))!(Np(x_2))!(Np(x_3))! \dots}$  but the following bounds are much more useful in practice

## Theorem 3

$$\frac{1}{(N+1)^{|\mathcal{X}|}} 2^{NH(p)} \leq |T(p)| \leq 2^{NH(p)}$$

## Proof

Let's assume  $p(\cdot)$  is the actual distribution of  $X$  here

$$1 \geq \sum_{x^N \in T(p)} p^N(x^N) = \sum_{x^N \in T(p)} 2^{-NH(p)} = |T(p)| 2^{-NH(p)}$$

$$1 = \sum_{\hat{p} \in \mathcal{P}_N} Pr(T(\hat{p}))$$

# Size of a type class

Recall that  $|T(p)| = \frac{N!}{(Np(x_1))!(Np(x_2))!(Np(x_3))! \dots}$  but the following bounds are much more useful in practice

## Theorem 3

$$\frac{1}{(N+1)^{|\mathcal{X}|}} 2^{NH(p)} \leq |T(p)| \leq 2^{NH(p)}$$

## Proof

Let's assume  $p(\cdot)$  is the actual distribution of  $X$  here

$$1 \geq \sum_{x^N \in T(p)} p^N(x^N) = \sum_{x^N \in T(p)} 2^{-NH(p)} = |T(p)| 2^{-NH(p)}$$

$$1 = \sum_{\hat{p} \in \mathcal{P}_N} \Pr(T(\hat{p})) \leq \sum_{\hat{p} \in \mathcal{P}_N} \max_{\tilde{p}} \Pr(T(\tilde{p}))$$

# Size of a type class

Recall that  $|T(p)| = \frac{N!}{(Np(x_1))!(Np(x_2))!(Np(x_3))! \dots}$  but the following bounds are much more useful in practice

## Theorem 3

$$\frac{1}{(N+1)^{|\mathcal{X}|}} 2^{NH(p)} \leq |T(p)| \leq 2^{NH(p)}$$

## Proof

Let's assume  $p(\cdot)$  is the actual distribution of  $X$  here

$$1 \geq \sum_{x^N \in T(p)} p^N(x^N) = \sum_{x^N \in T(p)} 2^{-NH(p)} = |T(p)| 2^{-NH(p)}$$

$$1 = \sum_{\hat{p} \in \mathcal{P}_N} Pr(T(\hat{p})) \leq \sum_{\hat{p} \in \mathcal{P}_N} \max_{\tilde{p}} Pr(T(\tilde{p})) = \sum_{\hat{p} \in \mathcal{P}_N} Pr(T(p))$$

# Size of a type class

Recall that  $|T(p)| = \frac{N!}{(Np(x_1))!(Np(x_2))!(Np(x_3))! \dots}$  but the following bounds are much more useful in practice

## Theorem 3

$$\frac{1}{(N+1)^{|\mathcal{X}|}} 2^{NH(p)} \leq |T(p)| \leq 2^{NH(p)}$$

## Proof

Let's assume  $p(\cdot)$  is the actual distribution of  $X$  here

$$1 \geq \sum_{x^N \in T(p)} p^N(x^N) = \sum_{x^N \in T(p)} 2^{-NH(p)} = |T(p)| 2^{-NH(p)}$$

$$1 = \sum_{\hat{p} \in \mathcal{P}_N} Pr(T(\hat{p})) \leq \sum_{\hat{p} \in \mathcal{P}_N} \max_{\tilde{p}} Pr(T(\tilde{p})) = \sum_{\hat{p} \in \mathcal{P}_N} Pr(T(p)) \leq (N+1)^{|\mathcal{X}|} Pr(T(p))$$



# Size of a type class

Recall that  $|T(p)| = \frac{N!}{(Np(x_1))!(Np(x_2))!(Np(x_3))! \dots}$  but the following bounds are much more useful in practice

## Theorem 3

$$\frac{1}{(N+1)^{|\mathcal{X}|}} 2^{NH(p)} \leq |T(p)| \leq 2^{NH(p)}$$

## Proof

Let's assume  $p(\cdot)$  is the actual distribution of  $X$  here

$$1 \geq \sum_{x^N \in T(p)} p^N(x^N) = \sum_{x^N \in T(p)} 2^{-NH(p)} = |T(p)| 2^{-NH(p)}$$

$$\begin{aligned} 1 &= \sum_{\hat{p} \in \mathcal{P}_N} Pr(T(\hat{p})) \leq \sum_{\hat{p} \in \mathcal{P}_N} \max_{\tilde{p}} Pr(T(\tilde{p})) = \sum_{\hat{p} \in \mathcal{P}_N} Pr(T(p)) \leq (N+1)^{|\mathcal{X}|} Pr(T(p)) \\ &= (N+1)^{|\mathcal{X}|} |T(p)| 2^{-NH(p)} \end{aligned}$$

# Probability of a type class

## Theorem 4

Let the true distribution of  $X$  is  $q(\cdot)$ , then

$$\frac{2^{-N(KL(p||q))}}{(N+1)^{|\mathcal{X}|}} \leq Pr(T(p)) \leq 2^{-N(KL(p||q))}$$

# Probability of a type class

## Theorem 4

Let the true distribution of  $X$  is  $q(\cdot)$ , then

$$\frac{2^{-N(KL(p||q))}}{(N+1)^{|\mathcal{X}|}} \leq \Pr(T(p)) \leq 2^{-N(KL(p||q))}$$

## Proof

From Theorem 1, each sequence in  $T(p)$  has probability  $2^{-N(H(p)+KL(p||q))}$  and since  $\frac{1}{(N+1)^{|\mathcal{X}|}} 2^{NH(p)} \leq |T(p)| \leq 2^{NH(p)}$  from Theorem 3,

$$\frac{1}{(N+1)^{|\mathcal{X}|}} 2^{NH(p)} 2^{-N(H(p)+KL(p||q))} \leq \Pr(T(p)) \leq 2^{NH(p)} 2^{-N(H(p)+KL(p||q))}$$

# Summary of type

- Type class  $T(p)$  contains all sequences with empirical distribution of  $p$ . That is,

$$T(p) = \left\{ x^N : \frac{\mathcal{N}(a|x^N)}{N} = p(a) \right\}$$

# Summary of type

- Type class  $T(p)$  contains all sequences with empirical distribution of  $p$ . That is,

$$T(p) = \left\{ x^N : \frac{\mathcal{N}(a|x^N)}{N} = p(a) \right\}$$

- All sequences in the type class  $T(p)$  has the same probability ( $q(\cdot)$  is the true distribution)

$$q^N(x^N) = 2^{-N(H(p)+KL(p||q))}$$

# Summary of type

- Type class  $T(p)$  contains all sequences with empirical distribution of  $p$ . That is,

$$T(p) = \left\{ x^N : \frac{\mathcal{N}(a|x^N)}{N} = p(a) \right\}$$

- All sequences in the type class  $T(p)$  has the same probability ( $q(\cdot)$  is the true distribution)

$$q^N(x^N) = 2^{-N(H(p)+KL(p||q))}$$

- There are about  $2^{NH(p)}$  sequences in  $T(p)$

$$\frac{1}{(N+1)^{|\mathcal{X}|}} 2^{NH(p)} \leq |T(p)| \leq 2^{NH(p)}$$

# Summary of type

- Type class  $T(p)$  contains all sequences with empirical distribution of  $p$ . That is,

$$T(p) = \left\{ x^N : \frac{\mathcal{N}(a|x^N)}{N} = p(a) \right\}$$

- All sequences in the type class  $T(p)$  has the same probability ( $q(\cdot)$  is the true distribution)

$$q^N(x^N) = 2^{-N(H(p)+KL(p||q))}$$

- There are about  $2^{NH(p)}$  sequences in  $T(p)$

$$\frac{1}{(N+1)^{|\mathcal{X}|}} 2^{NH(p)} \leq |T(p)| \leq 2^{NH(p)}$$

- Probability of getting a sequence in  $T(p)$  is about  $2^{-N(KL(p||q))}$ . More precisely,

$$\frac{2^{-N(KL(p||q))}}{(N+1)^{|\mathcal{X}|}} \leq Pr(T(p)) \leq 2^{-N(KL(p||q))}$$

# Summary of type

- Type class  $T(p)$  contains all sequences with empirical distribution of  $p$ . That is,

$$T(p) = \left\{ x^N : \frac{\mathcal{N}(a|x^N)}{N} = p(a) \right\}$$

- All sequences in the type class  $T(p)$  has the same probability ( $q(\cdot)$  is the true distribution)

$$q^N(x^N) = 2^{-N(H(p)+KL(p||q))}$$

- There are about  $2^{NH(p)}$  sequences in  $T(p)$

$$\frac{1}{(N+1)^{|\mathcal{X}|}} 2^{NH(p)} \leq |T(p)| \leq 2^{NH(p)}$$

- Probability of getting a sequence in  $T(p)$  is about  $2^{-N(KL(p||q))}$ . More precisely,

$$\frac{2^{-N(KL(p||q))}}{(N+1)^{|\mathcal{X}|}} \leq Pr(T(p)) \leq 2^{-N(KL(p||q))}$$

- There are  $(N+1)^{|\mathcal{X}|}$  types



# Rationale

- For the compression scheme (such as Huffmann coding) that we discussed earlier in this class, one needs to know the source distribution ahead to design the encoder and decoder

# Rationale

- For the compression scheme (such as Huffmann coding) that we discussed earlier in this class, one needs to know the source distribution ahead to design the encoder and decoder
- Question: Is it possible to construct compression scheme without knowing the source distibution and still performs as good?

# Rationale

- For the compression scheme (such as Huffman coding) that we discussed earlier in this class, one needs to know the source distribution ahead to design the encoder and decoder
- Question: Is it possible to construct compression scheme without knowing the source distribution and still performs as good?
- Answer: Yes. At least theoretically  $\rightarrow$  universal source coding

# Theory of universal source coding

Given any source  $Q$  with  $H(Q) < R$ , there exists a length- $N$  universal code of rate  $R$  such that the source can be decoded losslessly as  $N \rightarrow \infty$

# Theory of universal source coding

Given any source  $Q$  with  $H(Q) < R$ , there exists a length- $N$  universal code of rate  $R$  such that the source can be decoded losslessly as  $N \rightarrow \infty$

## Proof

Let  $R_N = R - |\mathcal{X}| \frac{\log(N+1)}{N}$ , and consider the set of sequences  $A = \{x^N : H(p_{x^N}) < R_N\}$  as the code book.

# Theory of universal source coding

Given any source  $Q$  with  $H(Q) < R$ , there exists a length- $N$  universal code of rate  $R$  such that the source can be decoded losslessly as  $N \rightarrow \infty$

## Proof

Let  $R_N = R - |\mathcal{X}| \frac{\log(N+1)}{N}$ , and consider the set of sequences  $A = \{x^N : H(p_{x^N}) < R_N\}$  as the code book. Note that the rate is  $< R$  as

$$|A| = \sum_{p: H(p) < R_N} |T(p)|$$

# Theory of universal source coding

Given any source  $Q$  with  $H(Q) < R$ , there exists a length- $N$  universal code of rate  $R$  such that the source can be decoded losslessly as  $N \rightarrow \infty$

## Proof

Let  $R_N = R - |\mathcal{X}| \frac{\log(N+1)}{N}$ , and consider the set of sequences  $A = \{x^N : H(p_{x^N}) < R_N\}$  as the code book. Note that the rate is  $< R$  as

$$|A| = \sum_{p: H(p) < R_N} |T(p)| \leq \sum_{p: H(p) < R_N} 2^{NH(p)}$$

# Theory of universal source coding

Given any source  $Q$  with  $H(Q) < R$ , there exists a length- $N$  universal code of rate  $R$  such that the source can be decoded losslessly as  $N \rightarrow \infty$

## Proof

Let  $R_N = R - |\mathcal{X}| \frac{\log(N+1)}{N}$ , and consider the set of sequences  $A = \{x^N : H(p_{x^N}) < R_N\}$  as the code book. Note that the rate is  $< R$  as

$$|A| = \sum_{p: H(p) < R_N} |T(p)| \leq \sum_{p: H(p) < R_N} 2^{NH(p)} < \sum_{p: H(p) < R_N} 2^{NR_N}$$



# Theory of universal source coding

Given any source  $Q$  with  $H(Q) < R$ , there exists a length- $N$  universal code of rate  $R$  such that the source can be decoded losslessly as  $N \rightarrow \infty$

## Proof

Let  $R_N = R - |\mathcal{X}| \frac{\log(N+1)}{N}$ , and consider the set of sequences  $A = \{x^N : H(p_{x^N}) < R_N\}$  as the code book. Note that the rate is  $< R$  as

$$\begin{aligned} |A| &= \sum_{p: H(p) < R_N} |T(p)| \leq \sum_{p: H(p) < R_N} 2^{NH(p)} < \sum_{p: H(p) < R_N} 2^{NR_N} \\ &\leq (N+1)^{|\mathcal{X}|} 2^{NR_N} \end{aligned}$$

# Theory of universal source coding

Given any source  $Q$  with  $H(Q) < R$ , there exists a length- $N$  universal code of rate  $R$  such that the source can be decoded losslessly as  $N \rightarrow \infty$

## Proof

Let  $R_N = R - |\mathcal{X}| \frac{\log(N+1)}{N}$ , and consider the set of sequences  $A = \{x^N : H(p_{x^N}) < R_N\}$  as the code book. Note that the rate is  $< R$  as

$$\begin{aligned} |A| &= \sum_{p: H(p) < R_N} |T(p)| \leq \sum_{p: H(p) < R_N} 2^{NH(p)} < \sum_{p: H(p) < R_N} 2^{NR_N} \\ &\leq (N+1)^{|\mathcal{X}|} 2^{NR_N} = 2^{N\left(R_N + |\mathcal{X}| \frac{\log(N+1)}{N}\right)} = 2^{NR} \end{aligned}$$

# Theory of universal source coding

Given any source  $Q$  with  $H(Q) < R$ , there exists a length- $N$  universal code of rate  $R$  such that the source can be decoded losslessly as  $N \rightarrow \infty$

## Proof

Let  $R_N = R - |\mathcal{X}| \frac{\log(N+1)}{N}$ , and consider the set of sequences  $A = \{x^N : H(p_{x^N}) < R_N\}$  as the code book. Note that the rate is  $< R$  as

$$\begin{aligned} |A| &= \sum_{p: H(p) < R_N} |T(p)| \leq \sum_{p: H(p) < R_N} 2^{NH(p)} < \sum_{p: H(p) < R_N} 2^{NR_N} \\ &\leq (N+1)^{|\mathcal{X}|} 2^{NR_N} = 2^{N\left(R_N + |\mathcal{X}| \frac{\log(N+1)}{N}\right)} = 2^{NR} \end{aligned}$$

- Encoder: given input, check if input is in  $A$ , output index if so. Otherwise, declare failure
- Decoder: simply map index back to the sequence

# Theory of universal source coding

## Proof (con't)

Note that the probability of error  $P_e$  is given by

$$P_e = \sum_{p:H(p)>R_N} Pr(T(p))$$

# Theory of universal source coding

## Proof (con't)

Note that the probability of error  $P_e$  is given by

$$P_e = \sum_{p:H(p)>R_N} Pr(T(p)) \leq \sum_{p:H(p)>R_N} \max_{\tilde{p}:H(\tilde{p})>R_N} Pr(T(\tilde{p}))$$

# Theory of universal source coding

## Proof (con't)

Note that the probability of error  $P_e$  is given by

$$\begin{aligned}
 P_e &= \sum_{p:H(p)>R_N} Pr(T(p)) \leq \sum_{p:H(p)>R_N} \max_{\tilde{p}:H(\tilde{p})>R_N} Pr(T(\tilde{p})) \\
 &\leq (1 + N)^{|\mathcal{X}|} 2^{-N \left( \min_{\tilde{p}:H(\tilde{p})>R_N} KL(\tilde{p}||q) \right)}
 \end{aligned}$$

# Theory of universal source coding

## Proof (con't)

Note that the probability of error  $P_e$  is given by

$$\begin{aligned}
 P_e &= \sum_{p:H(p)>R_N} Pr(T(p)) \leq \sum_{p:H(p)>R_N} \max_{\tilde{p}:H(\tilde{p})>R_N} Pr(T(\tilde{p})) \\
 &\leq (1+N)^{|\mathcal{X}|} 2^{-N \left( \min_{\tilde{p}:H(\tilde{p})>R_N} KL(\tilde{p}||q) \right)}
 \end{aligned}$$

- If  $H(q) < R$ , as  $R_N \rightarrow R$  as  $N$  increases, we can find some  $N_0$  such that  $H(q) < R_N$  for all  $N \geq N_0$

# Theory of universal source coding

## Proof (con't)

Note that the probability of error  $P_e$  is given by

$$\begin{aligned}
 P_e &= \sum_{p: H(p) > R_N} Pr(T(p)) \leq \sum_{p: H(p) > R_N} \max_{\tilde{p}: H(\tilde{p}) > R_N} Pr(T(\tilde{p})) \\
 &\leq (1 + N)^{|\mathcal{X}|} 2^{-N \left( \min_{\tilde{p}: H(\tilde{p}) > R_N} KL(\tilde{p}||q) \right)}
 \end{aligned}$$

- If  $H(q) < R$ , as  $R_N \rightarrow R$  as  $N$  increases, we can find some  $N_0$  such that  $H(q) < R_N$  for all  $N \geq N_0$
- Therefore, any  $p$  in  $\{p : H(p) > R_N\}$  cannot be the same as  $q$



# Theory of universal source coding

## Proof (con't)

Note that the probability of error  $P_e$  is given by

$$\begin{aligned}
 P_e &= \sum_{p:H(p)>R_N} Pr(T(p)) \leq \sum_{p:H(p)>R_N} \max_{\tilde{p}:H(\tilde{p})>R_N} Pr(T(\tilde{p})) \\
 &\leq (1+N)^{|\mathcal{X}|} 2^{-N \left( \min_{\tilde{p}:H(\tilde{p})>R_N} KL(\tilde{p}||q) \right)}
 \end{aligned}$$

- If  $H(q) < R$ , as  $R_N \rightarrow R$  as  $N$  increases, we can find some  $N_0$  such that  $H(q) < R_N$  for all  $N \geq N_0$
- Therefore, any  $p$  in  $\{p : H(p) > R_N\}$  cannot be the same as  $q$
- $\Rightarrow \min_{\tilde{p}:H(\tilde{p})>R_N} KL(\tilde{p}||q) > 0$  for  $N \geq N_0$

# Theory of universal source coding

## Proof (con't)

Note that the probability of error  $P_e$  is given by

$$\begin{aligned}
 P_e &= \sum_{p:H(p)>R_N} Pr(T(p)) \leq \sum_{p:H(p)>R_N} \max_{\tilde{p}:H(\tilde{p})>R_N} Pr(T(\tilde{p})) \\
 &\leq (1+N)^{|\mathcal{X}|} 2^{-N \left( \min_{\tilde{p}:H(\tilde{p})>R_N} KL(\tilde{p}||q) \right)}
 \end{aligned}$$

- If  $H(q) < R$ , as  $R_N \rightarrow R$  as  $N$  increases, we can find some  $N_0$  such that  $H(q) < R_N$  for all  $N \geq N_0$
- Therefore, any  $p$  in  $\{p : H(p) > R_N\}$  cannot be the same as  $q$
- $\Rightarrow \min_{\tilde{p}:H(\tilde{p})>R_N} KL(\tilde{p}||q) > 0$  for  $N \geq N_0$
- Hence,  $P_e \rightarrow 0$  as  $N \rightarrow \infty$

# Lempel-Ziv coding

- Its variants are widely used by compression tools almost everywhere (zip, pkzip, tiff, etc.)

# Lempel-Ziv coding

- Its variants are widely used by compression tools almost everywhere (zip, pkzip, tiff, etc.)
- Main ideas
  - Construct a dictionary including all previously seen segments

# Lempel-Ziv coding

- Its variants are widely used by compression tools almost everywhere (zip, pkzip, tiff, etc.)
- Main ideas
  - Construct a dictionary including all previously seen segments
  - Bits needed to send a new segment can be reduced taking advantage known segment in the dictionary

# Lempel-Ziv coding

- Its variants are widely used by compression tools almost everywhere (zip, pkzip, tiff, etc.)
- Main ideas
  - Construct a dictionary including all previously seen segments
  - Bits needed to send a new segment can be reduced taking advantage known segment in the dictionary
- Example: let's compress 10110111011110111
  - First parse segment into segments that haven't seen before  $\Rightarrow$

# Lempel-Ziv coding

- Its variants are widely used by compression tools almost everywhere (zip, pkzip, tiff, etc.)
- Main ideas
  - Construct a dictionary including all previously seen segments
  - Bits needed to send a new segment can be reduced taking advantage known segment in the dictionary
- Example: let's compress 10110111011110111
  - First parse segment into segments that haven't seen before  $\Rightarrow$   
1  
1

# Lempel-Ziv coding

- Its variants are widely used by compression tools almost everywhere (zip, pkzip, tiff, etc.)
- Main ideas
  - Construct a dictionary including all previously seen segments
  - Bits needed to send a new segment can be reduced taking advantage known segment in the dictionary
- Example: let's compress 10110111011110111
  - First parse segment into segments that haven't seen before  $\Rightarrow$   
$$\begin{array}{l} 1 \quad 2 \\ 1, 0 \end{array}$$



# Lempel-Ziv coding

- Its variants are widely used by compression tools almost everywhere (zip, pkzip, tiff, etc.)
- Main ideas
  - Construct a dictionary including all previously seen segments
  - Bits needed to send a new segment can be reduced taking advantage known segment in the dictionary
- Example: let's compress 10110111011110111
  - First parse segment into segments that haven't seen before  $\Rightarrow$   
1<sup>1</sup> 2<sup>2</sup> 3<sup>3</sup>  
1, 0, 11

# Lempel-Ziv coding

- Its variants are widely used by compression tools almost everywhere (zip, pkzip, tiff, etc.)
- Main ideas
  - Construct a dictionary including all previously seen segments
  - Bits needed to send a new segment can be reduced taking advantage known segment in the dictionary
- Example: let's compress 10110111011110111
  - First parse segment into segments that haven't seen before  $\Rightarrow$   
1 2 3 4  
1, 0, 11, 01

# Lempel-Ziv coding

- Its variants are widely used by compression tools almost everywhere (zip, pkzip, tiff, etc.)
- Main ideas
  - Construct a dictionary including all previously seen segments
  - Bits needed to send a new segment can be reduced taking advantage known segment in the dictionary
- Example: let's compress 10110111011110111
  - First parse segment into segments that haven't seen before  $\Rightarrow$   

$$\begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & \\ 1, & 0, & 11, & 01, & 110 & \end{array}$$

# Lempel-Ziv coding

- Its variants are widely used by compression tools almost everywhere (zip, pkzip, tiff, etc.)
- Main ideas
  - Construct a dictionary including all previously seen segments
  - Bits needed to send a new segment can be reduced taking advantage known segment in the dictionary
- Example: let's compress 10110111011110111
  - First parse segment into segments that haven't seen before  $\Rightarrow$   
1 2 3 4 5 6  
1, 0, 11, 01, 110, 111

# Lempel-Ziv coding

- Its variants are widely used by compression tools almost everywhere (zip, pkzip, tiff, etc.)
- Main ideas
  - Construct a dictionary including all previously seen segments
  - Bits needed to send a new segment can be reduced taking advantage known segment in the dictionary
- Example: let's compress 10110111011110111
  - First parse segment into segments that haven't seen before  $\Rightarrow$   
 $\begin{array}{cccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 1, & 0, & 11, & 01, & 110, & 111, & 10 \end{array}$

# Lempel-Ziv coding

- Its variants are widely used by compression tools almost everywhere (zip, pkzip, tiff, etc.)
- Main ideas
  - Construct a dictionary including all previously seen segments
  - Bits needed to send a new segment can be reduced taking advantage known segment in the dictionary
- Example: let's compress 10110111011110111
  - First parse segment into segments that haven't seen before  $\Rightarrow$ 

$$\begin{array}{cccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 1, & 0, & 11, & 01, & 110, & 111, & 10, & 111 \end{array}$$
  - Encode each segment into representation containing a pair of numbers:

# Lempel-Ziv coding

- Its variants are widely used by compression tools almost everywhere (zip, pkzip, tiff, etc.)
- Main ideas
  - Construct a dictionary including all previously seen segments
  - Bits needed to send a new segment can be reduced taking advantage known segment in the dictionary
- Example: let's compress 10110111011110111
  - First parse segment into segments that haven't seen before  $\Rightarrow$ 

$$\begin{array}{cccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 1, & 0, & 11, & 01, & 110, & 111, & 10, & 111 \end{array}$$
  - Encode each segment into representation containing a pair of numbers:
    - 1) index of segment (excluding the last bit) in the dictionary;

# Lempel-Ziv coding

- Its variants are widely used by compression tools almost everywhere (zip, pkzip, tiff, etc.)
- Main ideas
  - Construct a dictionary including all previously seen segments
  - Bits needed to send a new segment can be reduced taking advantage known segment in the dictionary
- Example: let's compress 10110111011110111
  - First parse segment into segments that haven't seen before  $\Rightarrow$   

$$\begin{array}{cccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 1, & 0, & 11, & 01, & 110, & 111, & 10, & 111 \end{array}$$
  - Encode each segment into representation containing a pair of numbers: 1) index of segment (excluding the last bit) in the dictionary; 2) the last bit



# Lempel-Ziv coding

- Its variants are widely used by compression tools almost everywhere (zip, pkzip, tiff, etc.)
- Main ideas
  - Construct a dictionary including all previously seen segments
  - Bits needed to send a new segment can be reduced taking advantage known segment in the dictionary
- Example: let's compress 10110111011110111
  - First parse segment into segments that haven't seen before  $\Rightarrow$ 

$$\begin{array}{cccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 1, & 0, & 11, & 01, & 110, & 111, & 10, & 111 \end{array}$$
  - Encode each segment into representation containing a pair of numbers: 1) index of segment (excluding the last bit) in the dictionary; 2) the last bit  $\Rightarrow (0, 1), (0, 0), (1, 1), (2, 1), (3, 0), (3, 1), (1, 0), (6, \emptyset)$

# Lempel-Ziv coding

- Its variants are widely used by compression tools almost everywhere (zip, pkzip, tiff, etc.)
- Main ideas
  - Construct a dictionary including all previously seen segments
  - Bits needed to send a new segment can be reduced taking advantage known segment in the dictionary
- Example: let's compress 10110111011110111
  - First parse segment into segments that haven't seen before  $\Rightarrow$ 

$$\begin{array}{cccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 1, & 0, & 11, & 01, & 110, & 111, & 10, & 111 \end{array}$$
  - Encode each segment into representation containing a pair of numbers: 1) index of segment (excluding the last bit) in the dictionary; 2) the last bit  $\Rightarrow (0, 1), (0, 0), (1, 1), (2, 1), (3, 0), (3, 1), (1, 0), (6, \emptyset)$
  - Encode representation to bit stream. Note that as the dictionary grows, number of bits needed to store the index increases  $\Rightarrow$   
**0100011101011100110010110**

# Lempel-Ziv decoding

- Decode bitstream back to representation

0100011101011001110010110  $\Rightarrow$

(0, 1), (0, 0), (1, 1), (2, 1), (3, 0), (3, 1), (1, 0), (6,  $\emptyset$ )

- Build dictionary and decode

# Lempel-Ziv decoding

- Decode bitstream back to representation

0100011101011001110010110  $\Rightarrow$

(0, 1), (0, 0), (1, 1), (2, 1), (3, 0), (3, 1), (1, 0), (6,  $\emptyset$ )

- Build dictionary and decode

1

1

$\Rightarrow$  1

# Lempel-Ziv decoding

- Decode bitstream back to representation

0100011101011001110010110  $\Rightarrow$

(0, 1), (0, 0), (1, 1), (2, 1), (3, 0), (3, 1), (1, 0), (6,  $\emptyset$ )

- Build dictionary and decode

1	2
1	0

$\Rightarrow$  10

# Lempel-Ziv decoding

- Decode bitstream back to representation

0100011101011001110010110  $\Rightarrow$

(0, 1), (0, 0), (1, 1), (2, 1), (3, 0), (3, 1), (1, 0), (6,  $\emptyset$ )

- Build dictionary and decode

1	2	3
1	0	11

$\Rightarrow$  1011

# Lempel-Ziv decoding

- Decode bitstream back to representation

0100011101011001110010110  $\Rightarrow$

(0, 1), (0, 0), (1, 1), (2, 1), (3, 0), (3, 1), (1, 0), (6,  $\emptyset$ )

- Build dictionary and decode

1	2	3	4
1	0	11	01

$\Rightarrow$  101101

# Lempel-Ziv decoding

- Decode bitstream back to representation

0100011101011001110010110  $\Rightarrow$

(0, 1), (0, 0), (1, 1), (2, 1), (3, 0), (3, 1), (1, 0), (6,  $\emptyset$ )

- Build dictionary and decode

1	2	3	4	5
1	0	11	01	110

$\Rightarrow$  101101110



# Lempel-Ziv decoding

- Decode bitstream back to representation

0100011101011001110010110  $\Rightarrow$

(0, 1), (0, 0), (1, 1), (2, 1), (3, 0), (3, 1), (1, 0), (6,  $\emptyset$ )

- Build dictionary and decode

1	2	3	4	5	6
1	0	11	01	110	111

$\Rightarrow$  101101110111

# Lempel-Ziv decoding

- Decode bitstream back to representation

0100011101011001110010110  $\Rightarrow$

(0, 1), (0, 0), (1, 1), (2, 1), (3, 0), (3, 1), (1, 0), (6,  $\emptyset$ )

- Build dictionary and decode

1	2	3	4	5	6	7
1	0	11	01	110	111	10

$\Rightarrow$  10110111011110

# Lempel-Ziv decoding

- Decode bitstream back to representation

0100011101011001110010110  $\Rightarrow$

(0, 1), (0, 0), (1, 1), (2, 1), (3, 0), (3, 1), (1, 0), (6,  $\emptyset$ )

- Build dictionary and decode

1	2	3	4	5	6	7	8
1	0	11	01	110	111	10	111

$\Rightarrow$  10110111011110111

# Motivation

- Let's revisit some coin tossing example. Say if a coin is fair, and we toss it for 1000 times, we know that we will almost always get 500 heads. So getting, say, 400 heads has negligible probability

# Motivation

- Let's revisit some coin tossing example. Say if a coin is fair, and we toss it for 1000 times, we know that we will almost always get 500 heads. So getting, say, 400 heads has negligible probability
- However, if we insist finding the probability of getting 400 heads, from discussion up to now, we know that it is just

$$Pr(T((0.4, 0.6))) \approx 2^{-1000(KL((0.4, 0.6)|| (0.5, 0.5)))}$$

# Motivation

- Let's revisit some coin tossing example. Say if a coin is fair, and we toss it for 1000 times, we know that we will almost always get 500 heads. So getting, say, 400 heads has negligible probability
- However, if we insist finding the probability of getting 400 heads, from discussion up to now, we know that it is just

$$Pr(T((0.4, 0.6))) \approx 2^{-1000(KL((0.4, 0.6)||0.5, 0.5))}$$

- Now, what if we are interested in the probability of a more general case? Say what is the probability of getting  $> 300$  and  $< 400$  heads?

# Sanov's Theorem

Let  $\mathcal{E} = \{p : 0.3 \leq p(\text{Head}) \leq 0.4\}$  and  $q(\cdot) = (0.5, 0.5)$  is the true distribution, then

$$Pr(\mathcal{E}) = Pr(\mathcal{E} \cap \mathcal{P}_{1000})$$

# Sanov's Theorem

Let  $\mathcal{E} = \{p : 0.3 \leq p(\text{Head}) \leq 0.4\}$  and  $q(\cdot) = (0.5, 0.5)$  is the true distribution, then

$$Pr(\mathcal{E}) = Pr(\mathcal{E} \cap \mathcal{P}_{1000}) = \sum_{p \in \mathcal{E} \cap \mathcal{P}_{1000}} Pr(T(p))$$



# Sanov's Theorem

Let  $\mathcal{E} = \{p : 0.3 \leq p(\text{Head}) \leq 0.4\}$  and  $q(\cdot) = (0.5, 0.5)$  is the true distribution, then

$$\Pr(\mathcal{E}) = \Pr(\mathcal{E} \cap \mathcal{P}_{1000}) = \sum_{p \in \mathcal{E} \cap \mathcal{P}_{1000}} \Pr(T(p)) \approx \sum_{p \in \mathcal{E} \cap \mathcal{P}_{1000}} 2^{-1000(KL(p||q))}$$

# Sanov's Theorem

Let  $\mathcal{E} = \{p : 0.3 \leq p(\text{Head}) \leq 0.4\}$  and  $q(\cdot) = (0.5, 0.5)$  is the true distribution, then

$$\begin{aligned} Pr(\mathcal{E}) &= Pr(\mathcal{E} \cap \mathcal{P}_{1000}) = \sum_{p \in \mathcal{E} \cap \mathcal{P}_{1000}} Pr(T(p)) \approx \sum_{p \in \mathcal{E} \cap \mathcal{P}_{1000}} 2^{-1000(KL(p||q))} \\ &= 2^{-1000(KL((0.4,0.6)|| (0.5,0.5)))} + 2^{-1000(KL((0.399,0.601)|| (0.5,0.5)))} + \\ &\quad 2^{-1000(KL((0.398,0.602)|| (0.5,0.5)))} + \dots + 2^{-1000(KL((0.3,0.7)|| (0.5,0.5)))} \end{aligned}$$

# Sanov's Theorem

Let  $\mathcal{E} = \{p : 0.3 \leq p(\text{Head}) \leq 0.4\}$  and  $q(\cdot) = (0.5, 0.5)$  is the true distribution, then

$$\begin{aligned}
 \Pr(\mathcal{E}) &= \Pr(\mathcal{E} \cap \mathcal{P}_{1000}) = \sum_{p \in \mathcal{E} \cap \mathcal{P}_{1000}} \Pr(T(p)) \approx \sum_{p \in \mathcal{E} \cap \mathcal{P}_{1000}} 2^{-1000(KL(p||q))} \\
 &= 2^{-1000(KL((0.4,0.6)|| (0.5,0.5)))} + 2^{-1000(KL((0.399,0.601)|| (0.5,0.5)))} + \\
 &\quad 2^{-1000(KL((0.398,0.602)|| (0.5,0.5)))} + \dots + 2^{-1000(KL((0.3,0.7)|| (0.5,0.5)))} \\
 &\leq |\mathcal{P}_{1000}| 2^{-1000(KL((0.4,0.6)|| (0.5,0.5)))}
 \end{aligned}$$

# Sanov's Theorem

Let  $\mathcal{E} = \{p : 0.3 \leq p(\text{Head}) \leq 0.4\}$  and  $q(\cdot) = (0.5, 0.5)$  is the true distribution, then

$$\begin{aligned} Pr(\mathcal{E}) &= Pr(\mathcal{E} \cap \mathcal{P}_{1000}) = \sum_{p \in \mathcal{E} \cap \mathcal{P}_{1000}} Pr(T(p)) \approx \sum_{p \in \mathcal{E} \cap \mathcal{P}_{1000}} 2^{-1000(KL(p||q))} \\ &= 2^{-1000(KL((0.4,0.6)|| (0.5,0.5)))} + 2^{-1000(KL((0.399,0.601)|| (0.5,0.5)))} + \\ &\quad 2^{-1000(KL((0.398,0.602)|| (0.5,0.5)))} + \dots + 2^{-1000(KL((0.3,0.7)|| (0.5,0.5)))} \\ &\leq |\mathcal{P}_{1000}| 2^{-1000(KL((0.4,0.6)|| (0.5,0.5)))} \end{aligned}$$

## Sanov's Theorem

Let  $X_1, X_2, \dots, X_N$  be i.i.d.  $\sim q(\cdot)$  and  $\mathcal{E}$  be a set of distribution. Then

$$Pr(\mathcal{E}) = Pr(\mathcal{E} \cap \mathcal{P}_N) \leq (N+1)^{|\mathcal{X}|} 2^{-N(KL(p^*||q))},$$

where  $p^* = \arg \min_{p \in \mathcal{E}} KL(p||q)$ .

# Sanov's Theorem

Let  $\mathcal{E} = \{p : 0.3 \leq p(\text{Head}) \leq 0.4\}$  and  $q(\cdot) = (0.5, 0.5)$  is the true distribution, then

$$\begin{aligned} Pr(\mathcal{E}) &= Pr(\mathcal{E} \cap \mathcal{P}_{1000}) = \sum_{p \in \mathcal{E} \cap \mathcal{P}_{1000}} Pr(T(p)) \approx \sum_{p \in \mathcal{E} \cap \mathcal{P}_{1000}} 2^{-1000(KL(p||q))} \\ &= 2^{-1000(KL((0.4,0.6)|| (0.5,0.5)))} + 2^{-1000(KL((0.399,0.601)|| (0.5,0.5)))} + \\ &\quad 2^{-1000(KL((0.398,0.602)|| (0.5,0.5)))} + \dots + 2^{-1000(KL((0.3,0.7)|| (0.5,0.5)))} \\ &\leq |\mathcal{P}_{1000}| 2^{-1000(KL((0.4,0.6)|| (0.5,0.5)))} \end{aligned}$$

## Sanov's Theorem

Let  $X_1, X_2, \dots, X_N$  be i.i.d.  $\sim q(\cdot)$  and  $\mathcal{E}$  be a set of distribution. Then

$$Pr(\mathcal{E}) = Pr(\mathcal{E} \cap \mathcal{P}_N) \leq (N+1)^{|\mathcal{X}|} 2^{-N(KL(p^*||q))},$$

where  $p^* = \arg \min_{p \in \mathcal{E}} KL(p||q)$ . Moreover, given a rather weak condition (closure of interior of  $\mathcal{E}$  is  $\mathcal{E}$  itself), we have

$$\frac{1}{N} \log Pr(\mathcal{E}) \rightarrow -KL(p^*||q)$$

# Conditional limit theorem

- The first part of Sanov's Theorem is easy to show as similar to the example. However, the second half will need some more math background (mostly mathematical analysis) to understand the proof and so we will skip it here

# Conditional limit theorem

- The first part of Sanov's Theorem is easy to show as similar to the example. However, the second half will need some more math background (mostly mathematical analysis) to understand the proof and so we will skip it here
- The latter part of Sanov's Theorem suggests that the probability of getting  $\mathcal{E}$  is the same as the probability of getting  $T(\rho^*)$

# Conditional limit theorem

- The first part of Sanov's Theorem is easy to show as similar to the example. However, the second half will need some more math background (mostly mathematical analysis) to understand the proof and so we will skip it here
- The latter part of Sanov's Theorem suggests that the probability of getting  $\mathcal{E}$  is the same as the probability of getting  $T(\rho^*)$
- It turns out that we can claim something stronger. We will state the theorem below without proof



# Conditional limit theorem

- The first part of Sanov's Theorem is easy to show as similar to the example. However, the second half will need some more math background (mostly mathematical analysis) to understand the proof and so we will skip it here
- The latter part of Sanov's Theorem suggests that the probability of getting  $\mathcal{E}$  is the same as the probability of getting  $T(\rho^*)$
- It turns out that we can claim something stronger. We will state the theorem below without proof

## Conditional limit theorem

Let  $\mathcal{E}$  be a closed convex subset of  $\mathcal{P}$  (the set of all distributions) and  $q(\cdot)$  be the true distribution which is  $\notin \mathcal{E}$ .

# Conditional limit theorem

- The first part of Sanov's Theorem is easy to show as similar to the example. However, the second half will need some more math background (mostly mathematical analysis) to understand the proof and so we will skip it here
- The latter part of Sanov's Theorem suggests that the probability of getting  $\mathcal{E}$  is the same as the probability of getting  $T(p^*)$
- It turns out that we can claim something stronger. We will state the theorem below without proof

## Conditional limit theorem

Let  $\mathcal{E}$  be a closed convex subset of  $\mathcal{P}$  (the set of all distributions) and  $q(\cdot)$  be the true distribution which is  $\notin \mathcal{E}$ . If  $x_1, x_2, \dots, x_N$  are drawn from  $q(\cdot)$  and we know that  $p_{x_N} \in \mathcal{E}$ , then

$$\frac{\mathcal{N}(a|x_N)}{N} \rightarrow p^*(a)$$

in probability as  $N \rightarrow \infty$

# Examples

## Coin toss

- Let's go back to our previous example. If we throw a fair coin 1000 times and some one tells you that there are 300 to 400 heads, recall
$$\mathcal{E} = \{0.3 \leq p(\text{Head}) \leq 0.4\}$$

# Examples

## Coin toss

- Let's go back to our previous example. If we throw a fair coin 1000 times and some one tells you that there are 300 to 400 heads, recall

$$\mathcal{E} = \{0.3 \leq p(\text{Head}) \leq 0.4\}$$

- Since apparently,

$$p^* = \arg \min_{p \in \mathcal{E}} KL(p || (0.5, 0.5)) = (0.4, 0.6)$$

# Examples

## Coin toss

- Let's go back to our previous example. If we throw a fair coin 1000 times and some one tells you that there are 300 to 400 heads, recall

$$\mathcal{E} = \{0.3 \leq p(\text{Head}) \leq 0.4\}$$

- Since apparently,

$$p^* = \arg \min_{p \in \mathcal{E}} KL(p || (0.5, 0.5)) = (0.4, 0.6)$$

- By conditional limit theorem, knowing the the number of head is within the range, the coin behaves like a biased coin with  $p(\text{Head}) = 0.4$

# Examples

## Coin toss

- Let's go back to our previous example. If we throw a fair coin 1000 times and some one tells you that there are 300 to 400 heads, recall

$$\mathcal{E} = \{0.3 \leq p(\text{Head}) \leq 0.4\}$$

- Since apparently,

$$p^* = \arg \min_{p \in \mathcal{E}} KL(p || (0.5, 0.5)) = (0.4, 0.6)$$

- By conditional limit theorem, knowing the the number of head is within the range, the coin behaves like a biased coin with  $p(\text{Head}) = 0.4$
- A best bet would be there are 400 heads

# Examples

## Lower bounds

- Let say  $x_1, x_2, \dots, x_N$  are drawn from  $q(\cdot)$ . And we have  $K$  functions  $g_1(\cdot), g_2(\cdot), \dots, g_K(\cdot)$  such that for  $k = 1, \dots, K$ ,

$$\frac{1}{N} \sum_{i=1}^N g_k(x_i) \geq \alpha_k$$

# Examples

## Lower bounds

- Let say  $x_1, x_2, \dots, x_N$  are drawn from  $q(\cdot)$ . And we have  $K$  functions  $g_1(\cdot), g_2(\cdot), \dots, g_K(\cdot)$  such that for  $k = 1, \dots, K$ ,

$$\frac{1}{N} \sum_{i=1}^N g_k(x_i) \geq \alpha_k$$

- Let  $\mathcal{E} = \{p : \sum_a p(a)g_k(a) \geq \alpha_k, k = 1, \dots, K\}$



# Examples

## Lower bounds

- Let say  $x_1, x_2, \dots, x_N$  are drawn from  $q(\cdot)$ . And we have  $K$  functions  $g_1(\cdot), g_2(\cdot), \dots, g_K(\cdot)$  such that for  $k = 1, \dots, K$ ,

$$\frac{1}{N} \sum_{i=1}^N g_k(x_i) \geq \alpha_k$$

- Let  $\mathcal{E} = \{p : \sum_a p(a)g_k(a) \geq \alpha_k, k = 1, \dots, K\}$

- From conditional limit theorem,  $\frac{\mathcal{N}(a|x^N)}{N} \rightarrow p^*(a)$ , where  
$$p^* = \arg \min_{p \in \mathcal{E}} KL(p||q)$$

# Examples

## Lower bounds

- Let say  $x_1, x_2, \dots, x_N$  are drawn from  $q(\cdot)$ . And we have  $K$  functions  $g_1(\cdot), g_2(\cdot), \dots, g_K(\cdot)$  such that for  $k = 1, \dots, K$ ,

$$\frac{1}{N} \sum_{i=1}^N g_k(x_i) \geq \alpha_k$$

- Let  $\mathcal{E} = \{p : \sum_a p(a)g_k(a) \geq \alpha_k, k = 1, \dots, K\}$

- From conditional limit theorem,  $\frac{\mathcal{N}(a|x^N)}{N} \rightarrow p^*(a)$ , where  

$$p^* = \arg \min_{p \in \mathcal{E}} KL(p||q)$$

- This is a simple constrained optimization problem and can be solved with KKT conditions. If you go through the conditions, you will find that

$$p^*(x) \propto q(x) 2^{\sum_{k=1}^K \lambda_k g_k(x)},$$

with  $\lambda_k(\sum_a p(a)g_k(a) - \alpha_k) = 0$ ,  $\lambda_k \geq 0$ , and  $\sum_a p(a)g_k(a) \geq \alpha_k$

# Examples

I think this example below gives a nice demonstration that the technique we have learned today can solve some amazing puzzle!

# Examples

I think this example below gives a nice demonstration that the technique we have learned today can solve some amazing puzzle!

## Fair dice

A fair dice is thrown 10,000 times and the sum of all outcomes is larger than 40,000, out of the 10,000 throw, how many ones do you think there are?

# Fair dice

- From the result of previous example, let  $g_1(x) = x$  and  $\alpha_1 = 4$ , we expect

$$p^*(i) = \frac{2^{\lambda i}}{\sum_{j=1}^6 2^{\lambda j}}$$

for some  $\lambda$

# Fair dice

- From the result of previous example, let  $g_1(x) = x$  and  $\alpha_1 = 4$ , we expect

$$p^*(i) = \frac{2^{\lambda i}}{\sum_{j=1}^6 2^{\lambda j}}$$

for some  $\lambda$

- $\lambda \neq 0$  since  $\sum_a p(a)g_1(a) = 3.5 < 4 = \alpha_1$  if so

# Fair dice

- From the result of previous example, let  $g_1(x) = x$  and  $\alpha_1 = 4$ , we expect

$$p^*(i) = \frac{2^{\lambda i}}{\sum_{j=1}^6 2^{\lambda j}}$$

for some  $\lambda$

- $\lambda \neq 0$  since  $\sum_a p(a)g_1(a) = 3.5 < 4 = \alpha_1$  if so
- Since  $\lambda \neq 0$ , by the complementary slackness constraint  $\lambda_k(\sum_a p(a)g_k(a) - \alpha_k) = 0$ ,

$$\sum_a p(a)g_1(a) = \alpha_1 = 4$$

# Fair dice

- From the result of previous example, let  $g_1(x) = x$  and  $\alpha_1 = 4$ , we expect

$$p^*(i) = \frac{2^{\lambda i}}{\sum_{j=1}^6 2^{\lambda j}}$$

for some  $\lambda$

- $\lambda \neq 0$  since  $\sum_a p(a)g_1(a) = 3.5 < 4 = \alpha_1$  if so
- Since  $\lambda \neq 0$ , by the complementary slackness constraint  $\lambda_k(\sum_a p(a)g_k(a) - \alpha_k) = 0$ ,

$$\sum_a p(a)g_1(a) = \alpha_1 = 4$$

- This gives us  $\lambda = 0.2519$ , and thus  $p^* = (0.103, 0.123, 0.146, 0.174, 0.207, 0.247)$



# Fair dice

- From the result of previous example, let  $g_1(x) = x$  and  $\alpha_1 = 4$ , we expect

$$p^*(i) = \frac{2^{\lambda i}}{\sum_{j=1}^6 2^{\lambda j}}$$

for some  $\lambda$

- $\lambda \neq 0$  since  $\sum_a p(a)g_1(a) = 3.5 < 4 = \alpha_1$  if so
- Since  $\lambda \neq 0$ , by the complementary slackness constraint  $\lambda_k(\sum_a p(a)g_k(a) - \alpha_k) = 0$ ,

$$\sum_a p(a)g_1(a) = \alpha_1 = 4$$

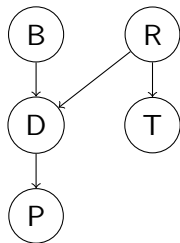
- This gives us  $\lambda = 0.2519$ , and thus  $p^* = (0.103, 0.123, 0.146, 0.174, 0.207, 0.247)$
- # ones  $\approx 0.103 \times 10000 = 1030$

# This time...

- Bayesian Net
- Belief Propagation Algorithm
- LDPC/IRA Codes

# Bayesian Net

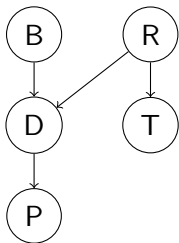
- Relationship of variables depicted by a directed graph with no loop
- Given a variable's parents, the variable is conditionally independent of any non-descendants
- Reduce model complexity
- Facilitate easier inference



# Burlgar and racoon

Burlgar: B; Racoon: R; Dog barked: D; Police called: P; Trash can fell: T

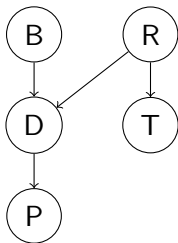
$$p(p, d, b, t, r) = p(p|d, b, t, r)p(d|b, t, r)p(b|t, r)p(t|r)p(r)$$



# Burlgar and racoon

Burlgar: B; Racoon: R; Dog barked: D; Police called: P; Trash can fell: T

$$\begin{aligned}
 p(p, d, b, t, r) &= p(p|d, b, t, r)p(d|b, t, r)p(b|t, r)p(t|r)p(r) \\
 &= \underbrace{p(p|d, b, t, r)}_{2 \text{ parameters}}p(d|b, t, r)p(b|t, r)p(t|r)p(r)
 \end{aligned}$$



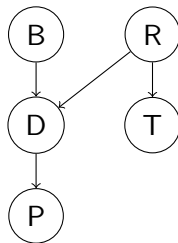
# Burlgar and racoon

Burlgar: B; Racoon: R; Dog barked: D; Police called: P; Trash can fell: T

$$\begin{aligned}
 p(p, d, b, t, r) &= p(p|d, b, t, r)p(d|b, t, r)p(b|t, r)p(t|r)p(r) \\
 &= \underbrace{p(p|d, \bar{b}, \bar{t}, \bar{r})}_{2 \text{ parameters}}p(d|b, \bar{t}, r)p(b|\bar{t}, \bar{r})p(t|r)p(r)
 \end{aligned}$$

$P$	$D$	$p(p d)$
$p$	$\neg d$	0.01
$p$	$d$	0.4
$\neg p$	$\neg d$	0.99
$\neg p$	$d$	0.6

$T$	$R$	$p(t r)$
$t$	$\neg r$	0.05
$t$	$r$	0.7
$\neg t$	$\neg r$	0.95
$\neg t$	$r$	0.3



## Burlgar and racoon

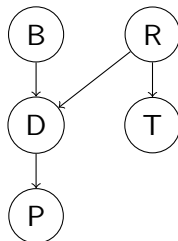
Burlgar: B; Racoon: R; Dog barked: D; Police called: P; Trash can fell: T

$$\begin{aligned}
 p(p, d, b, t, r) &= p(p|d, b, t, r)p(d|b, t, r)p(b|t, r)p(t|r)p(r) \\
 &= \underbrace{p(p|d, \bar{b}, \bar{t}, r)}_{2 \text{ parameters}} p(d|b, \bar{t}, r)p(b|\bar{t}, r)p(t|r)p(r)
 \end{aligned}$$

$P$	$D$	$p(p d)$
$p$	$\neg d$	0.01
$p$	$d$	0.4
$\neg p$	$\neg d$	0.99
$\neg p$	$d$	0.6

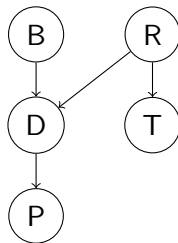
$T$	$R$	$p(t r)$
$t$	$\neg r$	0.05
$t$	$r$	0.7
$\neg t$	$\neg r$	0.95
$\neg t$	$r$	0.3

$D$	$B$	$R$	$p(d b, r)$
$d$	$\neg b$	$\neg r$	0.1
$d$	$\neg b$	$r$	0.5
$d$	$b$	$\neg r$	1
$d$	$b$	$r$	1
$\neg d$	$\neg b$	$\neg r$	0.9
$\neg d$	$\neg b$	$r$	0.5
$\neg d$	$b$	$\neg r$	0
$\neg d$	$b$	$r$	0



# Comparison of # parameters

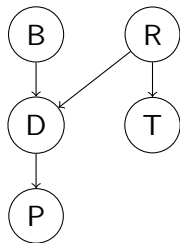
- # parameters of complete model:  $2^5 - 1 = 31$





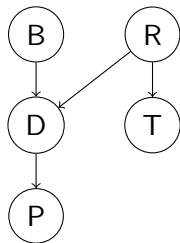
# Comparison of # parameters

- # parameters of complete model:  $2^5 - 1 = 31$
- # parameters of Bayesian net:



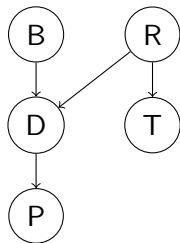
# Comparison of # parameters

- # parameters of complete model:  $2^5 - 1 = 31$
- # parameters of Bayesian net:
  - $p(p|d)$ : 2



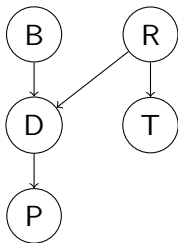
# Comparison of # parameters

- # parameters of complete model:  $2^5 - 1 = 31$
- # parameters of Bayesian net:
  - $p(p|d)$ : 2
  - $p(d|b, r)$ : 4



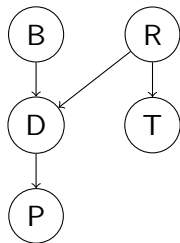
# Comparison of # parameters

- # parameters of complete model:  $2^5 - 1 = 31$
- # parameters of Bayesian net:
  - $p(p|d)$ : 2
  - $p(d|b, r)$ : 4
  - $p(b)$ : 1



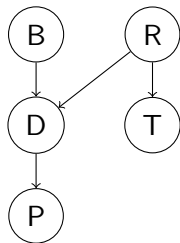
# Comparison of # parameters

- # parameters of complete model:  $2^5 - 1 = 31$
- # parameters of Bayesian net:
  - $p(p|d)$ : 2
  - $p(d|b, r)$ : 4
  - $p(b)$ : 1
  - $p(t|r)$ : 2



# Comparison of # parameters

- # parameters of complete model:  $2^5 - 1 = 31$
- # parameters of Bayesian net:
  - $p(p|d)$ : 2
  - $p(d|b, r)$ : 4
  - $p(b)$ : 1
  - $p(t|r)$ : 2
  - $p(r)$ : 1
  - Total:  $2 + 4 + 1 + 2 + 1 = 10$
- The model size reduces to less than  $\frac{1}{3}$ !



# Burglar and racoon

Question: What is the probability of a burglar visit if police was called but trash can stayed untouched?

Let  $p(r) = 0.2$  and  $p(b) = 0.01$

$D$	$B$	$R$	$p(d b,r)$
$d$	$\neg b$	$\neg r$	0.1
$d$	$\neg b$	$r$	0.5
$d$	$b$	$\neg r$	1
$d$	$b$	$r$	1
$\neg d$	$\neg b$	$\neg r$	0.9
$\neg d$	$\neg b$	$r$	0.5
$\neg d$	$b$	$\neg r$	0
$\neg d$	$b$	$r$	0

# Burglar and racoon

Question: What is the probability of a burglar visit if police was called but trash can stayed untouched?

Let  $p(r) = 0.2$  and  $p(b) = 0.01$

$D$	$B$	$R$	$p(d b,r)$
$d$	$\neg b$	$\neg r$	0.1
$d$	$\neg b$	$r$	0.5
$d$	$b$	$\neg r$	1
$d$	$b$	$r$	1
$\neg d$	$\neg b$	$\neg r$	0.9
$\neg d$	$\neg b$	$r$	0.5
$\neg d$	$b$	$\neg r$	0
$\neg d$	$b$	$r$	0

$\Rightarrow$

$D$	$B$	$R$	$p(d, b, r)$
$d$	$\neg b$	$\neg r$	0.0792
$d$	$\neg b$	$r$	0.099
$d$	$b$	$\neg r$	0.008
$d$	$b$	$r$	0.002
$\neg d$	$\neg b$	$\neg r$	0.7128
$\neg d$	$\neg b$	$r$	0.099
$\neg d$	$b$	$\neg r$	0
$\neg d$	$b$	$r$	0



# Burglar and racoon

Question: What is the probability of a burglar visit if police was called but trash can stayed untouched?

$P$	$D$	$p(p d)$
$p$	$\neg d$	0.01
$p$	$d$	0.4
$\neg p$	$\neg d$	0.99
$\neg p$	$d$	0.6

$P$	$D$	$B$	$R$	$p(d, b, r, p)$
$p$	$d$	$\neg b$	$\neg r$	0.0792
$p$	$d$	$\neg b$	$r$	0.099
$p$	$d$	$b$	$\neg r$	0.008
$p$	$d$	$b$	$r$	0.002
$p$	$\neg d$	$\neg b$	$\neg r$	0.7128
$p$	$\neg d$	$\neg b$	$r$	0.099
$p$	$\neg d$	$b$	$\neg r$	0
$p$	$\neg d$	$b$	$r$	0
...				

# Burglar and racoon

Question: What is the probability of a burglar visit if police was called but trash can stayed untouched?

$P$	$D$	$p(p d)$
$p$	$\neg d$	0.01
$p$	$d$	0.4
$\neg p$	$\neg d$	0.99
$\neg p$	$d$	0.6

$P$	$D$	$B$	$R$	$p(d, b, r, p)$
$p$	$d$	$\neg b$	$\neg r$	0.0792
$p$	$d$	$\neg b$	$r$	0.099
$p$	$d$	$b$	$\neg r$	0.008
$p$	$d$	$b$	$r$	0.002
$p$	$\neg d$	$\neg b$	$\neg r$	0.007128
$p$	$\neg d$	$\neg b$	$r$	0.00099
$p$	$\neg d$	$b$	$\neg r$	0
$p$	$\neg d$	$b$	$r$	0
...				

# Burglar and racoon

Question: What is the probability of a burglar visit if police was called but trash can stayed untouched?

$P$	$D$	$p(p d)$
$p$	$\neg d$	0.01
$p$	$d$	0.4
$\neg p$	$\neg d$	0.99
$\neg p$	$d$	0.6

$P$	$D$	$B$	$R$	$p(d, b, r, p)$
$p$	$d$	$\neg b$	$\neg r$	0.03168
$p$	$d$	$\neg b$	$r$	0.0396
$p$	$d$	$b$	$\neg r$	0.0032
$p$	$d$	$b$	$r$	0.0008
$p$	$\neg d$	$\neg b$	$\neg r$	0.007128
$p$	$\neg d$	$\neg b$	$r$	0.00099
$p$	$\neg d$	$b$	$\neg r$	0
$p$	$\neg d$	$b$	$r$	0
...				

# Burglar and racoon

Question: What is the probability of a burglar visit if police was called but trash can stayed untouched?

$T$	$R$	$p(t r)$
$t$	$\neg r$	0.05
$t$	$r$	0.7
$\neg t$	$\neg r$	0.95
$\neg t$	$r$	0.3

$T$	$P$	$D$	$B$	$R$	$p(d, b, r, p, t)$
$\neg t$	$p$	$d$	$\neg b$	$\neg r$	0.03168
$\neg t$	$p$	$d$	$\neg b$	$r$	0.0396
$\neg t$	$p$	$d$	$b$	$\neg r$	0.0032
$\neg t$	$p$	$d$	$b$	$r$	0.0008
$\neg t$	$p$	$\neg d$	$\neg b$	$\neg r$	0.007128
$\neg t$	$p$	$\neg d$	$\neg b$	$r$	0.00099
$\neg t$	$p$	$\neg d$	$b$	$\neg r$	0
$\neg t$	$p$	$\neg d$	$b$	$r$	0
...					

# Burglar and racoon

Question: What is the probability of a burglar visit if police was called but trash can stayed untouched?

$T$	$R$	$p(t r)$
$t$	$\neg r$	0.05
$t$	$r$	0.7
$\neg t$	$\neg r$	0.95
$\neg t$	$r$	0.3

$T$	$P$	$D$	$B$	$R$	$p(d, b, r, p, t)$
$\neg t$	$p$	$d$	$\neg b$	$\neg r$	0.030096
$\neg t$	$p$	$d$	$\neg b$	$r$	0.0396
$\neg t$	$p$	$d$	$b$	$\neg r$	0.00304
$\neg t$	$p$	$d$	$b$	$r$	0.0008
$\neg t$	$p$	$\neg d$	$\neg b$	$\neg r$	0.0067716
$\neg t$	$p$	$\neg d$	$\neg b$	$r$	0.00099
$\neg t$	$p$	$\neg d$	$b$	$\neg r$	0
$\neg t$	$p$	$\neg d$	$b$	$r$	0
...					

# Burglar and racoon

Question: What is the probability of a burglar visit if police was called but trash can stayed untouched?

$T$	$R$	$p(t r)$
$t$	$\neg r$	0.05
$t$	$r$	0.7
$\neg t$	$\neg r$	0.95
$\neg t$	$r$	0.3

$T$	$P$	$D$	$B$	$R$	$p(d, b, r, p, t)$
$\neg t$	$p$	$d$	$\neg b$	$\neg r$	0.030096
$\neg t$	$p$	$d$	$\neg b$	$r$	0.01188
$\neg t$	$p$	$d$	$b$	$\neg r$	0.00304
$\neg t$	$p$	$d$	$b$	$r$	0.00024
$\neg t$	$p$	$\neg d$	$\neg b$	$\neg r$	0.0067716
$\neg t$	$p$	$\neg d$	$\neg b$	$r$	0.000297
$\neg t$	$p$	$\neg d$	$b$	$\neg r$	0
$\neg t$	$p$	$\neg d$	$b$	$r$	0
...					

# Burglar and racoon

Question: What is the probability of a burglar visit if police was called but trash can stayed untouched?

Normalize...

$T$	$P$	$D$	$B$	$R$	$p(d, b, r, p)$
$\neg t$	$p$	$d$	$\neg b$	$\neg r$	0.030096
$\neg t$	$p$	$d$	$\neg b$	$r$	0.01188
$\neg t$	$p$	$d$	$b$	$\neg r$	0.00304
$\neg t$	$p$	$d$	$b$	$r$	0.00024
$\neg t$	$p$	$\neg d$	$\neg b$	$\neg r$	0.0067716
$\neg t$	$p$	$\neg d$	$\neg b$	$r$	0.000297
$\neg t$	$p$	$\neg d$	$b$	$\neg r$	0
$\neg t$	$p$	$\neg d$	$b$	$r$	0
...					

# Burglar and racoon

Question: What is the probability of a burglar visit if police was called but trash can stayed untouched?

Normalize...

$T$	$P$	$D$	$B$	$R$	$p(d, b, r, p)$
$\neg t$	$p$	$d$	$\neg b$	$\neg r$	0.57518
$\neg t$	$p$	$d$	$\neg b$	$r$	0.22704
$\neg t$	$p$	$d$	$b$	$\neg r$	0.058099
$\neg t$	$p$	$d$	$b$	$r$	0.0045868
$\neg t$	$p$	$\neg d$	$\neg b$	$\neg r$	0.12942
$\neg t$	$p$	$\neg d$	$\neg b$	$r$	0.0056761
$\neg t$	$p$	$\neg d$	$b$	$\neg r$	0
$\neg t$	$p$	$\neg d$	$b$	$r$	0
...					



# Burglar and racoon

Question: What is the probability of a burglar visit if police was called but trash can stayed untouched?

$$\begin{aligned}
 & p(b|\neg t, p) \\
 &= 0.058099 + 0.0045868 \\
 &\approx 0.0626
 \end{aligned}$$

$T$	$P$	$D$	$B$	$R$	$p(d, b, r, p)$
$\neg t$	$p$	$d$	$\neg b$	$\neg r$	0.57518
$\neg t$	$p$	$d$	$\neg b$	$r$	0.22704
$\neg t$	$p$	$d$	$b$	$\neg r$	0.058099
$\neg t$	$p$	$d$	$b$	$r$	0.0045868
$\neg t$	$p$	$\neg d$	$\neg b$	$\neg r$	0.12942
$\neg t$	$p$	$\neg d$	$\neg b$	$r$	0.0056761
$\neg t$	$p$	$\neg d$	$b$	$\neg r$	0
$\neg t$	$p$	$\neg d$	$b$	$r$	0
...					

# Belief Propagation Algorithm

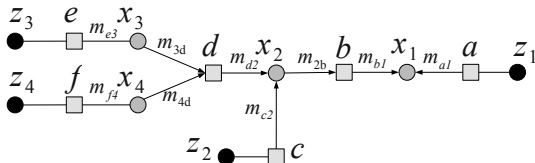
- It is also known to be the sum-product algorithm
- The goal of belief propagation is to efficiently compute the marginal distribution out of the joint distribution of multiple variables. This is essential for inferring the outcome of a particular variable with insufficient information
- The belief propagation algorithm is usually applied to problems modeled by a undirected graph (Markov random field) or a factor graph
- Rather than giving a rigorous proof of the algorithm, we will provide a simple example to illustrate the basic idea

# Factor Graph

- A factor graph is a bipartite graph describing the correlation among several random variables. It generally contains two different types of nodes in the graph: variable nodes and factor nodes
- A variable node that is usually shown as circles corresponds to a random variable
- A factor node that is usually shown as a square connects variable nodes whose corresponding variables are immediately related

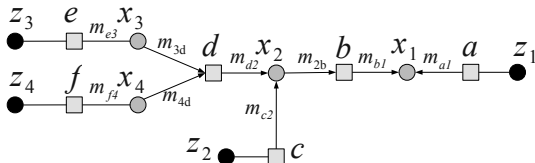
# An Example

- A factor graph example is shown below. We have 8 *discrete* random variables,  $x_1^4$  and  $z_1^4$ , depicted by 8 variable nodes



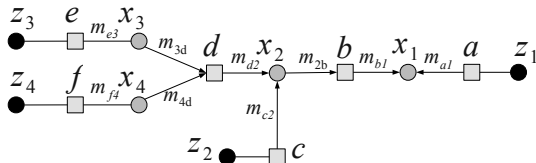
# An Example

- A factor graph example is shown below. We have 8 *discrete* random variables,  $x_1^4$  and  $z_1^4$ , depicted by 8 variable nodes
- Among the variable nodes, random variables  $x_1^4$  (indicated by light circles) are unknown and variables  $z_1^4$  (indicated by dark circles) are observed with known outcomes  $\tilde{z}_1^4$



# An Example

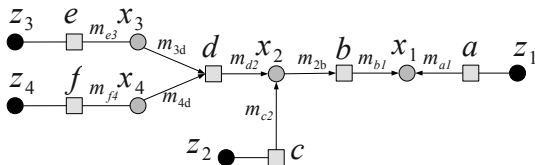
- A factor graph example is shown below. We have 8 *discrete* random variables,  $x_1^4$  and  $z_1^4$ , depicted by 8 variable nodes
- Among the variable nodes, random variables  $x_1^4$  (indicated by light circles) are unknown and variables  $z_1^4$  (indicated by dark circles) are observed with known outcomes  $\tilde{z}_1^4$
- The relationships among variables are captured entirely by the figure. For example, given  $x_1^4$ ,  $z_1$ ,  $z_2$ ,  $z_3$ , and  $z_4$  are conditional independent of each other. Moreover,  $(x_3, x_4)$  are conditional independent of  $x_1$  given  $x_2$



- The joint probability  $p(x^4, z^4)$  of all variables can be decomposed into factor functions with subsets of all variables as arguments in the following

$$p(x^4, z^4) = p(x^4)p(z_1|x_1)p(z_2|x_2)p(z_3|x_3)p(z_4|x_4)$$

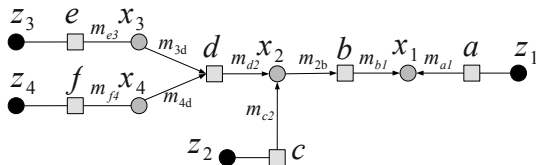
- Note that each factor function corresponds to a factor node in the factor graph.
- The arguments of the factor function correspond to the variable nodes that the factor node connects to.



- The joint probability  $p(x^4, z^4)$  of all variables can be decomposed into factor functions with subsets of all variables as arguments in the following

$$\begin{aligned}
 p(x^4, z^4) &= p(x^4)p(z_1|x_1)p(z_2|x_2)p(z_3|x_3)p(z_4|x_4) \\
 &= \underbrace{p(x_1, x_2)}_{f_b(x_1, x_2)} \underbrace{p(x_3, x_4|x_2)}_{f_d(x_2, x_3, x_4)} \underbrace{p(z_3|x_3)}_{f_e(x_3, z_3)} \underbrace{p(z_1|x_1)}_{f_a(x_1, z_1)} \underbrace{p(z_4|x_4)}_{f_f(x_4, z_4)} \underbrace{p(z_2|x_2)}_{f_c(x_2, z_2)}
 \end{aligned}$$

- Note that each factor function corresponds to a factor node in the factor graph.
- The arguments of the factor function correspond to the variable nodes that the factor node connects to.

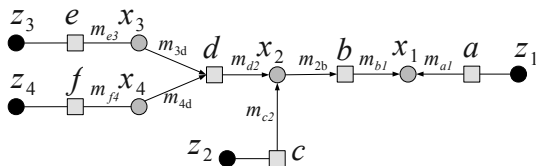




- The joint probability  $p(x^4, z^4)$  of all variables can be decomposed into factor functions with subsets of all variables as arguments in the following

$$\begin{aligned}
 p(x^4, z^4) &= p(x^4)p(z_1|x_1)p(z_2|x_2)p(z_3|x_3)p(z_4|x_4) \\
 &= \underbrace{p(x_1, x_2)}_{f_b(x_1, x_2)} \underbrace{p(x_3, x_4|x_2)}_{f_d(x_2, x_3, x_4)} \underbrace{p(z_3|x_3)}_{f_e(x_3, z_3)} \underbrace{p(z_1|x_1)}_{f_a(x_1, z_1)} \underbrace{p(z_4|x_4)}_{f_f(x_4, z_4)} \underbrace{p(z_2|x_2)}_{f_c(x_2, z_2)} \\
 &= f_b(x_1, x_2)f_d(x_2, x_3, x_4)f_e(x_3, z_3)f_a(x_1, z_1)f_f(x_4, z_4)f_c(x_2, z_2)
 \end{aligned}$$

- Note that each factor function corresponds to a factor node in the factor graph.
- The arguments of the factor function correspond to the variable nodes that the factor node connects to.



One common problem in probability inference is to estimate the value of a variable given incomplete information. For example, we may want to estimate  $x_1$  given  $z^4$  as  $\tilde{z}^4$ . The optimum estimate  $\hat{x}_1$  will satisfy

$$\hat{x}_1 = \arg \max_{x_1} p(x_1 | \tilde{z}^4) = \arg \max_{x_1} \frac{p(x_1, \tilde{z}^4)}{p(\tilde{z}^4)} = \arg \max_{x_1} p(x_1, \tilde{z}^4).$$

This requires us to compute the marginal distribution  $p(x_1, \tilde{z}^4)$  out of the joint probability  $p(x^4, \tilde{z}^4)$ . Note that

$$p(x_1, \tilde{z}^4) = \sum_{x_2^4} p(x^4, \tilde{z}^4)$$

One common problem in probability inference is to estimate the value of a variable given incomplete information. For example, we may want to estimate  $x_1$  given  $z^4$  as  $\tilde{z}^4$ . The optimum estimate  $\hat{x}_1$  will satisfy

$$\hat{x}_1 = \arg \max_{x_1} p(x_1 | \tilde{z}^4) = \arg \max_{x_1} \frac{p(x_1, \tilde{z}^4)}{p(\tilde{z}^4)} = \arg \max_{x_1} p(x_1, \tilde{z}^4).$$

This requires us to compute the marginal distribution  $p(x_1, \tilde{z}^4)$  out of the joint probability  $p(x^4, \tilde{z}^4)$ . Note that

$$\begin{aligned} p(x_1, \tilde{z}^4) &= \sum_{x_2^4} p(x^4, \tilde{z}^4) \\ &= \sum_{x_2^4} f_a(x_1, \tilde{z}_1) f_b(x_1, x_2) f_c(x_2, \tilde{z}_2) f_d(x_2, x_3, x_4) f_e(x_3, \tilde{z}_3) f_f(x_4, \tilde{z}_4) \end{aligned}$$

One common problem in probability inference is to estimate the value of a variable given incomplete information. For example, we may want to estimate  $x_1$  given  $z^4$  as  $\tilde{z}^4$ . The optimum estimate  $\hat{x}_1$  will satisfy

$$\hat{x}_1 = \arg \max_{x_1} p(x_1 | \tilde{z}^4) = \arg \max_{x_1} \frac{p(x_1, \tilde{z}^4)}{p(\tilde{z}^4)} = \arg \max_{x_1} p(x_1, \tilde{z}^4).$$

This requires us to compute the marginal distribution  $p(x_1, \tilde{z}^4)$  out of the joint probability  $p(x^4, \tilde{z}^4)$ . Note that

$$\begin{aligned} p(x_1, \tilde{z}^4) &= \sum_{x_2^4} p(x^4, \tilde{z}^4) \\ &= \sum_{x_2^4} f_a(x_1, \tilde{z}_1) f_b(x_1, x_2) f_c(x_2, \tilde{z}_2) f_d(x_2, x_3, x_4) f_e(x_3, \tilde{z}_3) f_f(x_4, \tilde{z}_4) \\ &= \underbrace{f_a(x_1, \tilde{z}_1)}_{m_{a1}} \sum_{x_2} \underbrace{f_b(x_1, x_2) f_c(x_2, \tilde{z}_2)}_{m_{c2}} \sum_{x_3, x_4} \underbrace{f_d(x_2, x_3, x_4) f_e(x_3, \tilde{z}_3) f_f(x_4, \tilde{z}_4)}_{m_{d2}} \\ &\quad \underbrace{\hspace{10em}}_{m_{2b}} \\ &\quad \underbrace{\hspace{15em}}_{m_{b1}} \end{aligned}$$

We can see from the last equation that the joint probability can be computed by combining a sequence of messages passing from a variable node  $i$  to a factor node  $a$  ( $m_{ia}$ ) and vice versa ( $m_{ai}$ ). More precisely, we can write

$$m_{a1}(x_1) \leftarrow f_a(x_1, \tilde{z}_1) = \sum_{z_1} f_a(x_1, z_1) \underbrace{p(z_1)}_{m_{1a}},$$

$$m_{c2}(x_2) \leftarrow f_c(x_2, \tilde{z}_2) = \sum_{z_2} f_c(x_2, z_2) \underbrace{p(z_2)}_{m_{2c}},$$

$$m_{e3}(x_3) \leftarrow f_e(x_3, \tilde{z}_3) = \sum_{z_3} f_e(x_3, z_3) \underbrace{p(z_3)}_{m_{3e}},$$

$$m_{f4}(x_4) \leftarrow f_f(x_4, \tilde{z}_4) = \sum_{z_4} f_f(x_4, z_4) \underbrace{p(z_4)}_{m_{4f}},$$

$$\text{where } p(z_i) = \begin{cases} 1, & z_i = \tilde{z}_i \\ 0, & \text{otherwise} \end{cases}$$

$$\begin{aligned}
 p(x_1, \tilde{z}^4) = & \underbrace{f_a(x_1, \tilde{z}_1)}_{m_{a1}} \sum_{x_2} f_b(x_1, x_2) \underbrace{f_c(x_2, \tilde{z}_2)}_{m_{c2}} \sum_{x_3, x_4} \underbrace{f_d(x_2, x_3, x_4)}_{m_{d2}} \underbrace{f_e(x_3, \tilde{z}_3)}_{m_{3d}} \underbrace{f_f(x_4, \tilde{z}_4)}_{m_{4d}} \quad (2) \\
 & \underbrace{\hspace{10em}}_{m_{2b}} \\
 & \underbrace{\hspace{15em}}_{m_{b1}}
 \end{aligned}$$

$$m_{3d}(x_3) \leftarrow m_{e3}(x_3) = f_e(x_3, \tilde{z}_3),$$

$$m_{4d}(x_4) \leftarrow m_{f4}(x_4) = f_f(x_4, \tilde{z}_4),$$

$$p(x_1, \tilde{z}^4) = \underbrace{f_a(x_1, \tilde{z}_1)}_{m_{a1}} \sum_{x_2} f_b(x_1, x_2) \underbrace{f_c(x_2, \tilde{z}_2)}_{m_{c2}} \sum_{x_3, x_4} \underbrace{f_d(x_2, x_3, x_4)}_{m_{d2}} \underbrace{f_e(x_3, \tilde{z}_3)}_{m_{3d}} \underbrace{f_f(x_4, \tilde{z}_4)}_{m_{4d}} \quad (2)$$

$$\begin{aligned}
 m_{3d}(x_3) &\leftarrow m_{e3}(x_3) = f_e(x_3, \tilde{z}_3), \\
 m_{4d}(x_4) &\leftarrow m_{f4}(x_4) = f_f(x_4, \tilde{z}_4), \\
 m_{d2}(x_2) &\leftarrow \sum_{x_3, x_4} f_d(x_2, x_3, x_4) m_{3d}(x_3) m_{4d}(x_4),
 \end{aligned}$$

$$\begin{aligned}
 p(x_1, \tilde{z}^4) &= \underbrace{f_a(x_1, \tilde{z}_1)}_{m_{a1}} \sum_{x_2} f_b(x_1, x_2) \underbrace{f_c(x_2, \tilde{z}_2)}_{m_{c2}} \underbrace{\sum_{x_3, x_4} f_d(x_2, x_3, x_4) \underbrace{f_e(x_3, \tilde{z}_3)}_{m_{3d}} \underbrace{f_f(x_4, \tilde{z}_4)}_{m_{4d}}}_{m_{d2}} \quad (2) \\
 &\quad \underbrace{\hspace{15em}}_{m_{2b}} \\
 &\quad \underbrace{\hspace{25em}}_{m_{b1}}
 \end{aligned}$$



$$\begin{aligned}
 m_{3d}(x_3) &\leftarrow m_{e3}(x_3) = f_e(x_3, \tilde{z}_3), \\
 m_{4d}(x_4) &\leftarrow m_{f4}(x_4) = f_f(x_4, \tilde{z}_4), \\
 m_{d2}(x_2) &\leftarrow \sum_{x_3, x_4} f_d(x_2, x_3, x_4) m_{3d}(x_3) m_{4d}(x_4), \\
 m_{2b}(x_2) &\leftarrow m_{c2}(x_2) m_{d2}(x_2),
 \end{aligned}$$

$$\begin{aligned}
 p(x_1, \tilde{z}^4) &= \underbrace{f_a(x_1, \tilde{z}_1)}_{m_{a1}} \sum_{x_2} f_b(x_1, x_2) \underbrace{f_c(x_2, \tilde{z}_2)}_{m_{c2}} \underbrace{\sum_{x_3, x_4} f_d(x_2, x_3, x_4) \underbrace{f_e(x_3, \tilde{z}_3)}_{m_{3d}} \underbrace{f_f(x_4, \tilde{z}_4)}_{m_{4d}}}_{m_{d2}} \quad (2) \\
 &\quad \underbrace{\hspace{15em}}_{m_{2b}} \\
 &\quad \underbrace{\hspace{20em}}_{m_{b1}}
 \end{aligned}$$

$$m_{3d}(x_3) \leftarrow m_{e3}(x_3) = f_e(x_3, \tilde{z}_3),$$

$$m_{4d}(x_4) \leftarrow m_{f4}(x_4) = f_f(x_4, \tilde{z}_4),$$

$$m_{d2}(x_2) \leftarrow \sum_{x_3, x_4} f_d(x_2, x_3, x_4) m_{3d}(x_3) m_{4d}(x_4),$$

$$m_{2b}(x_2) \leftarrow m_{c2}(x_2) m_{d2}(x_2),$$

$$m_{b1}(x_1) \leftarrow \sum_{x_2} f_b(x_1, x_2) m_{2b}(x_2),$$

$$p(x_1, \tilde{z}^4) = \underbrace{f_a(x_1, \tilde{z}_1)}_{m_{a1}} \sum_{x_2} \underbrace{f_b(x_1, x_2)}_{m_{c2}} \underbrace{f_c(x_2, \tilde{z}_2)}_{m_{c2}} \underbrace{\sum_{x_3, x_4} f_d(x_2, x_3, x_4) f_e(x_3, \tilde{z}_3) f_f(x_4, \tilde{z}_4)}_{m_{d2}} \quad (2)$$

$\underbrace{\hspace{15em}}_{m_{2b}}$   
 $\underbrace{\hspace{20em}}_{m_{b1}}$

$$m_{3d}(x_3) \leftarrow m_{e3}(x_3) = f_e(x_3, \tilde{z}_3),$$

$$m_{4d}(x_4) \leftarrow m_{f4}(x_4) = f_f(x_4, \tilde{z}_4),$$

$$m_{d2}(x_2) \leftarrow \sum_{x_3, x_4} f_d(x_2, x_3, x_4) m_{3d}(x_3) m_{4d}(x_4),$$

$$m_{2b}(x_2) \leftarrow m_{c2}(x_2) m_{d2}(x_2),$$

$$m_{b1}(x_1) \leftarrow \sum_{x_2} f_b(x_1, x_2) m_{2b}(x_2),$$

$$p(x_1, \tilde{z}^4) \leftarrow m_{a1}(x_1) m_{b1}(x_1),$$

$$p(x_1, \tilde{z}^4) = \underbrace{f_a(x_1, \tilde{z}_1)}_{m_{a1}} \sum_{x_2} \underbrace{f_b(x_1, x_2)}_{m_{c2}} \underbrace{f_c(x_2, \tilde{z}_2)}_{m_{c2}} \underbrace{\sum_{x_3, x_4} f_d(x_2, x_3, x_4)}_{m_{d2}} \underbrace{f_e(x_3, \tilde{z}_3)}_{m_{3d}} \underbrace{f_f(x_4, \tilde{z}_4)}_{m_{4d}} \quad (2)$$

# Belief propagation algorithm

- **Initialization:** For any variable node  $i$ , if the prior probability of  $x_i$  is known and equal to  $p(x_i)$ , for  $a \in N(i)$ ,
- **Message passing:**
- **Belief update:**
- **Stopping criteria:** repeat message update and/or belief update until the algorithm stops when maximum number of iterations is reached or some other conditions are satisfied.

# Belief propagation algorithm

- **Initialization:** For any variable node  $i$ , if the prior probability of  $x_i$  is known and equal to  $p(x_i)$ , for  $a \in N(i)$ ,

$$m_{ia}(x_i) \leftarrow p(x_i)$$

- **Message passing:**

- **Belief update:**

- **Stopping criteria:** repeat message update and/or belief update until the algorithm stops when maximum number of iterations is reached or some other conditions are satisfied.

# Belief propagation algorithm

- **Initialization:** For any variable node  $i$ , if the prior probability of  $x_i$  is known and equal to  $p(x_i)$ , for  $a \in N(i)$ ,

$$m_{ia}(x_i) \leftarrow p(x_i)$$

- **Message passing:**

$$m_{ia}(x_i) \leftarrow \prod_{b \in N(i) \setminus a} m_{bi}(x_i),$$

$$m_{ai}(x_i) \leftarrow \sum_{\mathbf{x}_a} f_a(\mathbf{x}_a) \prod_{j \in N(a) \setminus i} m_{ja}(x_j) \quad (\text{"sum-product"})$$

- **Belief update:**

- **Stopping criteria:** repeat message update and/or belief update until the algorithm stops when maximum number of iterations is reached or some other conditions are satisfied.

# Belief propagation algorithm

- **Initialization:** For any variable node  $i$ , if the prior probability of  $x_i$  is known and equal to  $p(x_i)$ , for  $a \in N(i)$ ,

$$m_{ia}(x_i) \leftarrow p(x_i)$$

- **Message passing:**

$$m_{ia}(x_i) \leftarrow \prod_{b \in N(i) \setminus a} m_{bi}(x_i),$$

$$m_{ai}(x_i) \leftarrow \sum_{\mathbf{x}_a} f_a(\mathbf{x}_a) \prod_{j \in N(a) \setminus i} m_{ja}(x_j) \quad (\text{"sum-product"})$$

- **Belief update:**

$$\beta_i(x_i) \leftarrow \prod_{a \in N(i)} m_{ai}(x_i)$$

- **Stopping criteria:** repeat message update and/or belief update until the algorithm stops when maximum number of iterations is reached or some other conditions are satisfied.

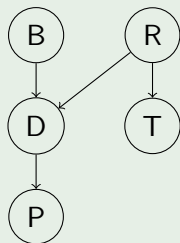
# Remark

- We have not assumed the precise physical meanings of the factor functions themselves. The only assumption we made is that the joint probability can be decomposed into the factor functions and apparently this decomposition is not unique
- The belief propagation algorithm as shown above is exact only because the corresponding graph is a tree and has no loop. If loop exists, the algorithm is not exact and generally the final belief may not even converge
- While the result is no longer exact, applying BP algorithm for general graphs (sometimes refer to as loopy BP) works well in many applications such as LDPC decoding



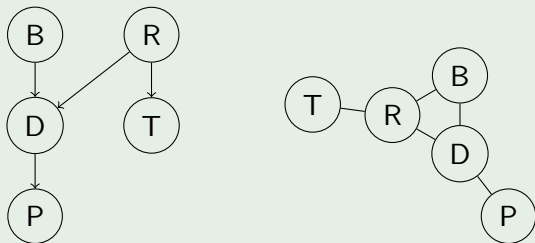
# Burglar and racoon revisit

Question: What is the probability of a burglar visit if police was called but trash can stayed untouched?



# Burglar and racoon revisit

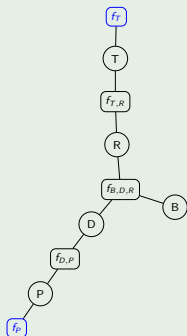
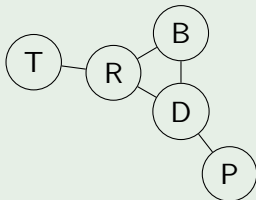
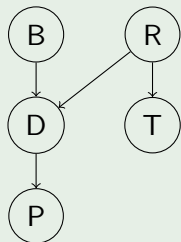
Question: What is the probability of a burglar visit if police was called but trash can stayed untouched?



Moralization...

# Burglar and racoon revisit

Question: What is the probability of a burglar visit if police was called but trash can stayed untouched?



Convert to factor graph..

## Using belief propagation...

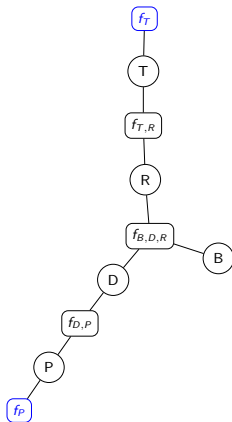
$$\begin{cases} f_P(p) &= 1 \\ f_P(\neg p) &= 0 \end{cases}$$

$$\begin{cases} f_T(t) &= 0 \\ f_T(\neg t) &= 1 \end{cases}$$

$$f_{B,D,R}(b, d, r) = p(b, d, r)$$

$$f_{T,R}(t, r) = p(t|r)$$

$$f_{D,P}(d, p) = p(p|d)$$



# Some History of LDPC Codes

- Before 1990's, the strategy for channel code has always been looking for codes that can be decoded optimally. This leads to a wide range of so-called algebraic codes. It turns out the “optimally-decodable” codes are usually poor codes
- Until early 1990's, researchers had basically agreed that the Shannon capacity was restricted to theoretical interest and could hardly be reached in practice
- The introduction of turbo codes gave a huge shock to the research community. The community were so dubious about the amazing performance of turbo codes that they did not accept the finding initially until independent researchers had verified the results
- The low-density parity-check (LDPC) codes were later rediscovered and both LDPC codes and turbo codes are based on the same philosophy differs from codes in the past. Instead of designing and using codes that can be decoded “optimally”, let us just pick some *random* codes and perform decoding “sub-optimally”

# LDPC Codes

- As its name suggests, LDPC codes refer to codes that with sparse (low-density) parity check matrices. In other words, there are only few ones in a parity check matrix and the rest are all zeros

# LDPC Codes

- As its name suggests, LDPC codes refer to codes that with sparse (low-density) parity check matrices. In other words, there are only few ones in a parity check matrix and the rest are all zeros
- We learn from the proof of Channel Coding Theorem that random code is asymptotically optimum. This suggests that if we just generate a code randomly with a very long code length. It is likely that we will get a very good code.

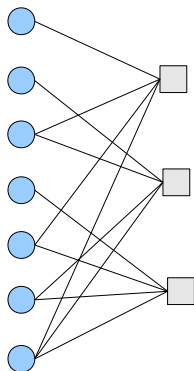
# LDPC Codes

- As its name suggests, LDPC codes refer to codes that with sparse (low-density) parity check matrices. In other words, there are only few ones in a parity check matrix and the rest are all zeros
- We learn from the proof of Channel Coding Theorem that random code is asymptotically optimum. This suggests that if we just generate a code randomly with a very long code length. It is likely that we will get a very good code.
- The problem is: how do we perform decoding? Due to the lack of structure of a random code, tricks that enable fast decoding for structured algebraic codes that were widely used before 1990's are unrealizable here
- Solution: Belief propagation!



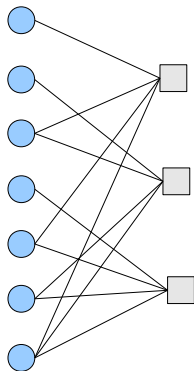
# Tanner Graph

- An LDPC code can be represented using a Tanner graph as shown on the right



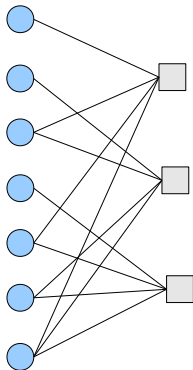
# Tanner Graph

- An LDPC code can be represented using a Tanner graph as shown on the right
- Each circle  $x_i$  represents a code bit sent to the decoder



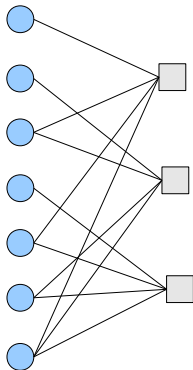
# Tanner Graph

- An LDPC code can be represented using a Tanner graph as shown on the right
- Each circle  $x_i$  represents a code bit sent to the decoder
- Each square represents a check bit with value equal to the sum of code bit connecting to it



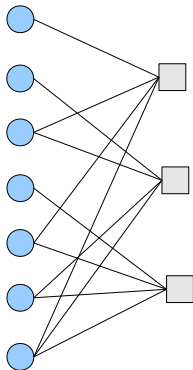
# Tanner Graph

- An LDPC code can be represented using a Tanner graph as shown on the right
- Each circle  $x_i$  represents a code bit sent to the decoder
- Each square represents a check bit with value equal to the sum of code bit connecting to it
- The vector  $x_1, x_2, \dots, x_N$  is a codeword only if all checks are zero



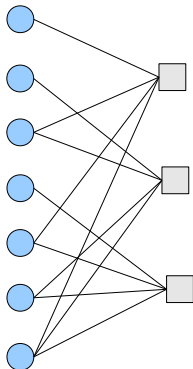
# Tanner Graph

- An LDPC code can be represented using a Tanner graph as shown on the right
- Each circle  $x_i$  represents a code bit sent to the decoder
- Each square represents a check bit with value equal to the sum of code bit connecting to it
- The vector  $x_1, x_2, \dots, x_N$  is a codeword only if all checks are zero
- By default, the mapping between a codeword to the actual message is non-trivial for an LDPC code



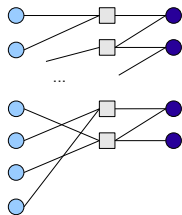
# Tanner Graph

- An LDPC code can be represented using a Tanner graph as shown on the right
- Each circle  $x_i$  represents a code bit sent to the decoder
- Each square represents a check bit with value equal to the sum of code bit connecting to it
- The vector  $x_1, x_2, \dots, x_N$  is a codeword only if all checks are zero
- By default, the mapping between a codeword to the actual message is non-trivial for an LDPC code
- It would be great if the actual message is included in the codeword. That is, some of the bits in the codeword spell out the actual message  $\Rightarrow$  IRA codes



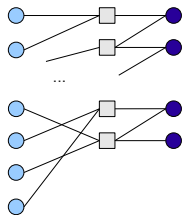
# IRA Codes

- Irregular repeated accumulate (IRA) code a type of systematic LDPC code, i.e., each codeword can be partitioned into message bits and syndrome bits



# IRA Codes

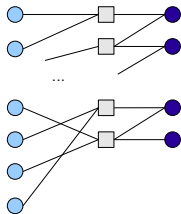
- Irregular repeated accumulate (IRA) code a type of systematic LDPC code, i.e., each codeword can be partitioned into message bits and syndrome bits
- As shown on the right, light blue circles correspond to the input message bits and the dark blue circle correspond to the syndrome bits





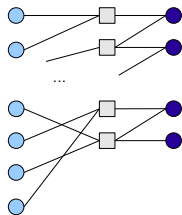
# IRA Codes

- Irregular repeated accumulate (IRA) code a type of systematic LDPC code, i.e., each codeword can be partitioned into message bits and syndrome bits
- As shown on the right, light blue circles correspond to the input message bits and the dark blue circle correspond to the syndrome bits
- To ensure the top check bit is satisfied, the top syndrome bit will be set to be the sum of message bits connecting to the check



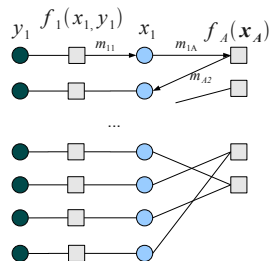
# IRA Codes

- Irregular repeated accumulate (IRA) code a type of systematic LDPC code, i.e., each codeword can be partitioned into message bits and syndrome bits
- As shown on the right, light blue circles correspond to the input message bits and the dark blue circle correspond to the syndrome bits
- To ensure the top check bit is satisfied, the top syndrome bit will be set to be the sum of message bits connecting to the check
- The computed syndrome bit will then pass to the next check and again we can ensure the next check bit is satisfied by setting that second syndrome bit as the sum of message bits connecting to the check + *last syndrome bit*. All (dark blue) syndrome bits can be assigned in similar token



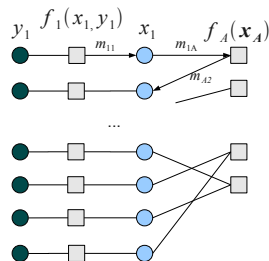
## LDPC Decoding

- $x_1, \dots, x_N$  (light blue): transmitted bits
- $y_1, \dots, y_N$  (dark grey): received bits



## LDPC Decoding

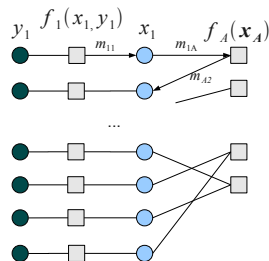
- $x_1, \dots, x_N$  (light blue): transmitted bits
- $y_1, \dots, y_N$  (dark grey): received bits
- $p(x^N, y^N) = \prod_i \underbrace{p(y_i|x_i)}_{f_i(x_i, y_i)} \underbrace{p(x^N)}_{\prod_A f_A(x_A)}$



## LDPC Decoding

- $x_1, \dots, x_N$  (light blue): transmitted bits
- $y_1, \dots, y_N$  (dark grey): received bits
- $p(x^N, y^N) = \prod_i \underbrace{p(y_i|x_i)}_{f_i(x_i, y_i)} \underbrace{p(x^N)}_{\prod_A f_A(\mathbf{x}_A)}$
- $f_i(x_i, y_i) = p(y_i|x_i)$  and

$$f_A(\mathbf{x}) = \begin{cases} 0, & \mathbf{x} \text{ contains even number of 1,} \\ 1, & \mathbf{x} \text{ contains odd number of 1.} \end{cases}$$



# Variable Node Update

- Since the unknown variables are binary, it is more convenient to represent the messages using likelihood or log-likelihood ratios. Define

$$l_{ai} \triangleq \frac{m_{ai}(0)}{m_{ai}(1)}, \quad L_{ai} \triangleq \log l_{ai} \quad (3)$$

and

$$l_{ia} \triangleq \frac{m_{ia}(0)}{m_{ia}(1)}, \quad L_{ia} \triangleq \log l_{ia} \quad (4)$$

for any variable node  $i$  and factor node  $a$ .

- Then,

$$L_{ia} \leftarrow \sum_{b \in N(i) \setminus i} L_{ai}. \quad (5)$$

# Check Node Update

- Assuming that we have three variable nodes 1,2, and 3 connecting to the check node  $a$ , then the check to variable node updates become

$$m_{a1}(1) \leftarrow m_{2a}(1)m_{3a}(0) + m_{2a}(0)m_{3a}(1) \quad (6)$$

$$m_{a1}(0) \leftarrow m_{2a}(0)m_{3a}(0) + m_{2a}(1)m_{3a}(1) \quad (7)$$

- Substitute in the likelihood ratios and log-likelihood ratios, we have

$$l_{a1} \triangleq \frac{m_{a1}(0)}{m_{a1}(1)} \leftarrow \frac{1 + l_{2a}l_{3a}}{l_{2a} + l_{3a}} \quad (8)$$

and

$$e^{L_{a1}} = l_{a1} \leftarrow \frac{1 + e^{L_{2a}}e^{L_{3a}}}{e^{L_{2a}} + e^{L_{3a}}} \quad (9)$$

- Note that

$$\tanh\left(\frac{L_{a1}}{2}\right) = \frac{e^{\frac{L_{a1}}{2}} - e^{-\frac{L_{a1}}{2}}}{e^{\frac{L_{a1}}{2}} + e^{-\frac{L_{a1}}{2}}} = \frac{e^{L_{a1}} - 1}{e^{L_{a1}} + 1} \quad (10)$$

$$\leftarrow \frac{1 + e^{L_{2a}}e^{L_{3a}} - e^{L_{2a}} - e^{L_{3a}}}{1 + e^{L_{2a}}e^{L_{3a}} + e^{L_{2a}} + e^{L_{3a}}} \quad (11)$$

$$= \frac{(e^{L_{2a}} - 1)(e^{L_{3a}} - 1)}{(e^{L_{2a}} + 1)(e^{L_{3a}} + 1)} \quad (12)$$

$$= \tanh\left(\frac{L_{2a}}{2}\right) \tanh\left(\frac{L_{3a}}{2}\right). \quad (13)$$

- When we have more than 3 variable nodes connecting to the check node  $a$ , it is easy to show using induction that

$$\tanh\left(\frac{L_{ai}}{2}\right) \leftarrow \prod_{j \in N(a) \setminus i} \tanh\left(\frac{L_{ja}}{2}\right). \quad (14)$$