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 satisfies, 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 satisfies, 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 satisfies, 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 satisfies, 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 satisfies, 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

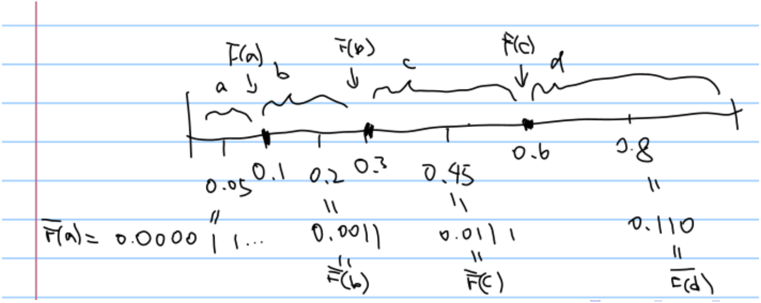
Shannon-Fano-Elias code

Key idea

Each codeword corresponds to an intervel of $[0, 1]$

Example

110 corresponds to $[0.110, 0.110\dot{1}] = [0.11, 0.111) = [0.75, 0.875)$



Shannon-Fano-Elias code

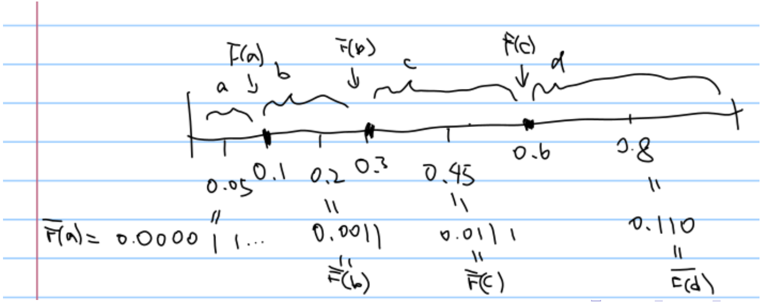
Key idea

Each codeword corresponds to an intervel of $[0, 1]$

Example

110 corresponds to $[0.110, 0.110\dot{1}] = [0.11, 0.111) = [0.75, 0.875)$

011 corresponds to $[0.011, 0.011\dot{1}] = [0.011, 0.1) = [0.375, 0.5)$



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	$0.1\overline{0011}$	4	1001
4	0.15	0.85	0.775	$0.110\overline{0011}$	4	1100
5	0.15	1.0	0.925	$0.111\overline{0110}$	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
- 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
- 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
- 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
- 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 $H(X) + 2$

$$\sum_{x \in \mathcal{X}} p(x) l(x) = \sum_{x \in \mathcal{X}} p(x) \left(\left\lceil \log_2 \frac{1}{p(x)} \right\rceil + 1 \right)$$
$$\leq \sum_{x \in \mathcal{X}} p(x) \left(\log_2 \frac{1}{p(x)} + 2 \right) = H(X) + 2$$

"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 symbol 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 symbol 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 symbol 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 Neumman 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 Neumman

$$H(X) = -\sum_{x \in \mathcal{X}} p(x) \log p(x) = E[-\log p(X)]$$

Entropy for discrete random variable

Von Neumman 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 Neumman

$$H(X) = -\sum_{x \in \mathcal{X}} p(x) \log p(x) = E[-\log p(X)]$$

- 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 Neumman 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 Neumman

$$H(X) = -\sum_{x\in\mathcal{X}} p(x)\log p(x) = E[-\log p(X)]$$

- 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 probalility $1/4 = 0.25$ of happening it. And to represent each outcome, we need $\log 4 = \log \frac{1}{0.25}$ bits

Entropy for discrete random variable

Von Neumman 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 Neumman

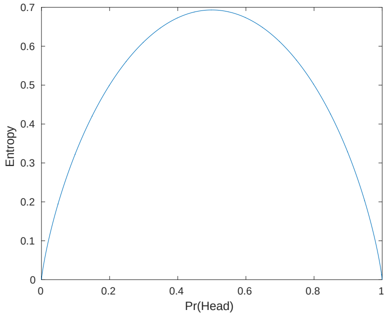
$$H(X) = -\sum_{x\in\mathcal{X}} p(x)\log p(x) = E[-\log p(X)]$$

- 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 probalility $1/4 = 0.25$ 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 $Pr(Head) = p$

$$H(X) = -Pr(Head)\log Pr(Head) - Pr(Tail)\log Pr(Tail)$$
$$= -p\log p - (1-p)\log(1-p)$$

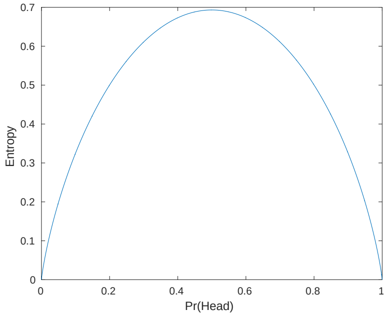
- Entropy is largest (=1) when $p = 0.5$
- Entropy is 0 when $p = 0$ or $p = 1$



Biased coin with $Pr(Head) = p$

$$H(X) = -Pr(Head)\log Pr(Head) - Pr(Tail)\log Pr(Tail)$$
$$= -p\log p - (1-p)\log(1-p)$$

- 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

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

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 $T \sim Exp(\lambda)$,

$$h(T) = E[-\log p(T)] = E\left[-\log\left(\lambda \exp(-\lambda T)\right)\right]$$

Differential entropy of common distributions

Uniform Distribution

If $p(X) = \begin{cases} 1/a & 0 \le x \le a \\ 0 & \text{otherwise} \end{cases}$

$$h(X) = -\int_{x=0}^{a} \frac{1}{a} \log \frac{1}{a} dx = \log a$$

Exponential distribution

For exponentially distributed $T \sim Exp(\lambda)$,

$$\begin{aligned} h(T) &= E[-\log p(T)] = E\left[-\log\left(\lambda \exp(-\lambda T)\right)\right] \\ &= E\left[\lambda T - \log \lambda\right] \end{aligned}$$

Differential entropy of common distributions

Uniform Distribution

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 $T \sim Exp(\lambda)$,

$$\begin{aligned} h(T) &= E[-\log p(T)] = E\left[-\log\left(\lambda \exp(-\lambda T)\right)\right] \\ &= E\left[\lambda T - \log \lambda\right] \\ &= 1 - \log \lambda \end{aligned}$$

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 $X \sim \mathcal{N}(\mu, \sigma^2)$,

$$h(X) = E[-\log p(X)] = E\left[-\log\left(\frac{1}{\sqrt{2\pi\sigma^2}}\exp\frac{-(X-\mu)^2}{2\sigma^2}\right)\right]$$

$$= E\left[\log\sqrt{2\pi\sigma^2} + \frac{(X-\mu)^2}{2\sigma^2}\log e\right]$$

Differential entropy of common distributions

Univariate Normal distribution

For univariate normally distributed $X \sim \mathcal{N}(\mu, \sigma^2)$,

$$h(X) = E[-\log p(X)] = E\left[-\log\left(\frac{1}{\sqrt{2\pi\sigma^2}}\exp\frac{-(X-\mu)^2}{2\sigma^2}\right)\right]$$
$$= E\left[\log\sqrt{2\pi\sigma^2} + \frac{(X-\mu)^2}{2\sigma^2}\log e\right]$$
$$= \log\sqrt{2\pi\sigma^2} + \frac{1}{2}\log e$$
$$= \log\sqrt{2\pi e\sigma^2}$$

N.B. $h(X)$ only depends on σ^2 and is independent of μ as one would expect

Multivariate Normal distribution

For N-dim multivariate normal distributed $\mathbf{X} \sim \mathcal{N}(\boldsymbol{\mu}, \Sigma)$,

$$\begin{aligned} h(\mathbf{X}) &= E[-\log p(\mathbf{X})] \\ &= -E\left[\log\left(\frac{1}{\sqrt{\det(2\pi\Sigma)}}\exp\left(-\frac{1}{2}(\boldsymbol{X}-\boldsymbol{\mu})^T\Sigma^{-1}(\boldsymbol{X}-\boldsymbol{\mu})\right)\right)\right] \end{aligned}$$

Multivariate Normal distribution

For N-dim multivariate normal distributed $\mathbf{X} \sim \mathcal{N}(\boldsymbol{\mu}, \Sigma)$,

$$
\begin{aligned}
h(\mathbf{X}) &= E[-\log p(\mathbf{X})] \\
&= -E\left[\log\left(\frac{1}{\sqrt{\det(2\pi\Sigma)}}\exp\left(-\frac{1}{2}(\boldsymbol{X}-\boldsymbol{\mu})^T\Sigma^{-1}(\boldsymbol{X}-\boldsymbol{\mu})\right)\right)\right] \\
&= \log\sqrt{\det(2\pi\Sigma)} + \frac{\log e}{2}E\left[\sum_{i,j}(x_i-\mu_i)\left[\Sigma^{-1}\right]_{i,j}(x_j-\mu_j)\right]
\end{aligned}
$$

Multivariate Normal distribution

For N-dim multivariate normal distributed $\mathbf{X} \sim \mathcal{N}(\boldsymbol{\mu}, \Sigma)$,

$$\begin{aligned}
h(\mathbf{X}) &= E[-\log p(\mathbf{X})] \\
&= -E\left[\log\left(\frac{1}{\sqrt{\det(2\pi\Sigma)}}\exp\left(-\frac{1}{2}(\boldsymbol{X}-\boldsymbol{\mu})^T\Sigma^{-1}(\boldsymbol{X}-\boldsymbol{\mu})\right)\right)\right] \\
&= \log\sqrt{\det(2\pi\Sigma)} + \frac{\log e}{2} E\left[\sum_{i,j}(x_i-\mu_i)\left[\Sigma^{-1}\right]_{i,j}(x_j-\mu_j)\right] \\
&= \log\sqrt{\det(2\pi\Sigma)} + \frac{\log e}{2}\sum_{i,j}\left[\Sigma^{-1}\right]_{i,j} E\left[(x_j-\mu_j)(x_i-\mu_i)\right]
\end{aligned}$$

Multivariate Normal distribution

For N-dim multivariate normal distributed $\mathbf{X} \sim \mathcal{N}(\boldsymbol{\mu}, \Sigma)$,

$$\begin{aligned}
h(\mathbf{X}) &= E[-\log p(\mathbf{X})] \\
&= -E\left[\log\left(\frac{1}{\sqrt{\det(2\pi\Sigma)}}\exp\left(-\frac{1}{2}(\boldsymbol{X}-\boldsymbol{\mu})^T\Sigma^{-1}(\boldsymbol{X}-\boldsymbol{\mu})\right)\right)\right] \\
&= \log\sqrt{\det(2\pi\Sigma)} + \frac{\log e}{2}E\left[\sum_{i,j}(x_i-\mu_i)\left[\Sigma^{-1}\right]_{i,j}(x_j-\mu_j)\right] \\
&= \log\sqrt{\det(2\pi\Sigma)} + \frac{\log e}{2}\sum_{i,j}\left[\Sigma^{-1}\right]_{i,j}E\left[(x_j-\mu_j)(x_i-\mu_i)\right] \\
&= \log\sqrt{\det(2\pi\Sigma)} + \frac{\log e}{2}\sum_{i,j}\left[\Sigma^{-1}\right]_{i,j}\Sigma_{j,i}
\end{aligned}$$

Multivariate Normal distribution

For N-dim multivariate normal distributed $\mathbf{X} \sim \mathcal{N}(\boldsymbol{\mu}, \Sigma)$,

$$
\begin{aligned}
h(\mathbf{X}) &= E[-\log p(\mathbf{X})] \\
&= -E\left[\log\left(\frac{1}{\sqrt{\det(2\pi\Sigma)}}\exp\left(-\frac{1}{2}(\boldsymbol{X}-\boldsymbol{\mu})^T\Sigma^{-1}(\boldsymbol{X}-\boldsymbol{\mu})\right)\right)\right] \\
&= \log\sqrt{\det(2\pi\Sigma)} + \frac{\log e}{2}E\left[\sum_{i,j}(x_i-\mu_i)\left[\Sigma^{-1}\right]_{i,j}(x_j-\mu_j)\right] \\
&= \log\sqrt{\det(2\pi\Sigma)} + \frac{\log e}{2}\sum_{i,j}\left[\Sigma^{-1}\right]_{i,j}E\left[(x_j-\mu_j)(x_i-\mu_i)\right] \\
&= \log\sqrt{\det(2\pi\Sigma)} + \frac{\log e}{2}\sum_{i,j}\left[\Sigma^{-1}\right]_{i,j}\Sigma_{j,i} \\
&= \log\sqrt{\det(2\pi\Sigma)} + \frac{N\log e}{2} = \log\sqrt{e^N\det(2\pi\Sigma)}
\end{aligned}
$$

Multivariate Normal distribution

For N-dim multivariate normal distributed $\mathbf{X} \sim \mathcal{N}(\boldsymbol{\mu}, \Sigma)$,

$$\begin{aligned}
h(\mathbf{X}) &= E[-\log p(\mathbf{X})] \\
&= -E\left[\log\left(\frac{1}{\sqrt{\det(2\pi\Sigma)}}\exp\left(-\frac{1}{2}(\boldsymbol{X}-\boldsymbol{\mu})^T\Sigma^{-1}(\boldsymbol{X}-\boldsymbol{\mu})\right)\right)\right] \\
&= \log\sqrt{\det(2\pi\Sigma)} + \frac{\log e}{2}E\left[\sum_{i,j}(x_i-\mu_i)\left[\Sigma^{-1}\right]_{i,j}(x_j-\mu_j)\right] \\
&= \log\sqrt{\det(2\pi\Sigma)} + \frac{\log e}{2}\sum_{i,j}\left[\Sigma^{-1}\right]_{i,j}E\left[(x_j-\mu_j)(x_i-\mu_i)\right] \\
&= \log\sqrt{\det(2\pi\Sigma)} + \frac{\log e}{2}\sum_{i,j}\left[\Sigma^{-1}\right]_{i,j}\Sigma_{j,i} \\
&= \log\sqrt{\det(2\pi\Sigma)} + \frac{N\log e}{2} = \log\sqrt{e^N\det(2\pi\Sigma)} = \log\sqrt{\det(2\pi e\Sigma)}
\end{aligned}$$

Differential entropy and entropy

How differential entropy is related to its discrete counterpart?

- Consider a continuous random variable X
- Let X^Δ is a "quantized" version of it with quantization stepsize of Δ

$$H(X^\Delta) = \sum -p_{X^\Delta}(x^\Delta) \log p_{X^\Delta}(x^\Delta)$$

Differential entropy and entropy

How differential entropy is related to its discrete counterpart?

- Consider a continuous random variable X
- Let X^{Δ} is a "quantized" version of it with quantization stepsize of Δ

$$H(X^{\Delta}) = \sum -p_{X^{\Delta}}(x^{\Delta}) \log p_{X^{\Delta}}(x^{\Delta}) \approx \sum -p_X(x^{\Delta})\Delta \log(p_X(x^{\Delta})\Delta)$$

Differential entropy and entropy

How differential entropy is related to its discrete counterpart?

- Consider a continuous random variable X
- Let X^{Δ} is a "quantized" version of it with quantization stepsize of Δ

$$H(X^{\Delta}) = \sum -p_{X^{\Delta}}(x^{\Delta}) \log p_{X^{\Delta}}(x^{\Delta}) \approx \sum -p_X(x^{\Delta})\Delta \log(p_X(x^{\Delta})\Delta)$$
$$\approx \int -p_X(x) \log(p_X(x)\Delta) dx$$

Differential entropy and entropy

How differential entropy is related to its discrete counterpart?

- Consider a continuous random variable X
- Let X^Δ is a "quantized" version of it with quantization stepsize of Δ

$$
\begin{aligned}
H(X^\Delta) &= \sum -p_{X^\Delta}(x^\Delta)\log p_{X^\Delta}(x^\Delta) \approx \sum -p_X(x^\Delta)\Delta \log(p_X(x^\Delta)\Delta) \\
&\approx \int -p_X(x)\log(p_X(x)\Delta)dx \\
&= \int -p_X(x)\log p_X(x) - \int p_X(x)\log\Delta dx \\
&= h(X) - \log\Delta
\end{aligned}
$$

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 $\lambda = 1/1 = 1ms^{-1}$
- 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 $h(T) - \log 0.01 \approx 7.64 bits$

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

$$h(X) < 0$$

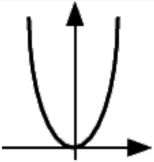
For example, for a uniformly distributed X from 0 to 0.5,

$$h(X) = \log 0.5 = -1$$

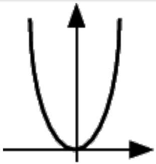
Jensen's Inequality

For a convex (bowl-shape) function f

$$E[f(X)] \geq f(E[X])$$



Jensen's Inequality



For a convex (bowl-shape) function f

$$E[f(X)] \geq f(E[X])$$

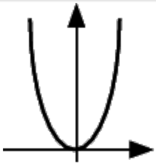
Let us consider X with only two outcomes x_1 and x_2 with probabilities p and $1-p$. Easy to see that

$$E[f(X)] = pf(x_1) + (1-p)f(x_2) \geq f(px_1 + (1-p)x_2) = f(E[X])$$

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

$H(X) \leq \log |\mathcal{X}|$

$$
\begin{aligned}
H(X) = E[-\log p(X)] &= E\left[\log \frac{1}{p(X)}\right] \\
&\leq \log E\left[\frac{1}{p(X)}\right] \qquad \text{(by Jensen's inequality)}
\end{aligned}
$$

Upper bound of entropy

$H(X) \leq \log |\mathcal{X}|$

$$
\begin{aligned}
H(X) &= E[-\log p(X)] = E\left[\log \frac{1}{p(X)}\right] \\
&\leq \log E\left[\frac{1}{p(X)}\right] \qquad \text{(by Jensen's inequality)} \\
&= \log \sum_{x \in \mathcal{X}} p(x) \frac{1}{p(x)} = \log |\mathcal{X}|
\end{aligned}
$$

N.B. The upper bound is attained when the distribution is uniform

Upper bound of entropy

$H(X) \leq \log |\mathcal{X}|$

$$
\begin{aligned}
H(X) &= E[-\log p(X)] = E\left[\log \frac{1}{p(X)}\right] \\
&\leq \log E\left[\frac{1}{p(X)}\right] \qquad \text{(by Jensen's inequality)} \\
&= \log \sum_{x \in \mathcal{X}} p(x) \frac{1}{p(x)} = \log |\mathcal{X}|
\end{aligned}
$$

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

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 σ^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

$$H(X_1, X_2, \cdots, X_N) = E[-\log p(X_1, \cdots, X_N)]$$

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

$$H(X_1, X_2, \cdots, X_N) = E[-\log p(X_1, \cdots, X_N)]$$

Differential entropy

$$h(X, Y) = E[-\log p(X, Y)]$$

and

$$h(X_1, X_2, \cdots, X_N) = E[-\log p(X_1, \cdots, X_N)]$$

Conditional entropy

$$H(X,Y) = E[-\log p(X,Y)] = E[-\log p(X) - \log p(Y|X)]$$
$$= H(X) + \underbrace{E[-\log p(Y|X)]}_{H(Y|X)}$$

Entropy

$$H(Y|X) \triangleq H(X,Y) - H(X)$$

Conditional entropy

$$H(X,Y) = E[-\log p(X,Y)] = E[-\log p(X) - \log p(Y|X)]$$
$$= H(X) + \underbrace{E[-\log p(Y|X)]}_{H(Y|X)}$$

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

$$H(X_1, X_2, \cdots, X_N) = H(X_1) + H(X_2|X_1) + H(X_3|X_1, X_2) + \cdots + H(X_N|X_1, X_2, \cdots, X_{N-1}).$$

Chain rule

Entropy

$$H(X_1, X_2, \cdots, X_N) = H(X_1) + H(X_2|X_1) + H(X_3|X_1, X_2) + \cdots + H(X_N|X_1, X_2, \cdots, X_{N-1}).$$

Differential entropy

$$h(X_1, X_2, \cdots, X_N) = h(X_1) + h(X_2|X_1) + h(X_3|X_1, X_2) + \cdots + h(X_N|X_1, X_2, \cdots, X_{N-1}).$$

Example

$$Pr(Rain, With\ umbrella) = 0.2 \qquad Pr(Rain, No\ umbrella) = 0.1$$
$$Pr(Sunny, With\ umbrella) = 0.2 \qquad Pr(Sunny, No\ umbrella) = 0.5$$

$$W \in \{Rain, Sunny\} \qquad U \in \{With\ umbrella, No\ umbrella\}$$

Entropies

$$H(W,U) = -0.2 \log 0.2 - 0.1 \log 0.1 - 0.2 \log 0.2 - 0.5 \log 0.5 = 1.76\ bits$$
$$H(W) = -0.3 \log 0.3 - 0.7 \log 0.7 = 0.88\ bits$$
$$H(U) = -0.4 \log 0.4 - 0.6 \log 0.6 = 0.97\ bits$$
$$H(W|U) = H(W,U) - H(U) = 0.79\ bits$$
$$H(U|W) = H(W,U) - H(W) = 0.88\ bits$$

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 $p(y)$,

$$KL(p(x)\|q(x)) \triangleq \sum_{x\in\mathcal{X}} p(x)\log_2 \frac{p(x)}{q(x)}.$$

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 $p(y)$,

$$KL(p(x)\|q(x)) \triangleq \sum_{x\in\mathcal{X}} p(x)\log_2\frac{p(x)}{q(x)}.$$

- 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 $p(y)$,

$$KL(p(x)\|q(x)) \triangleq \sum_{x\in\mathcal{X}} p(x)\log_2 \frac{p(x)}{q(x)}.$$

- 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

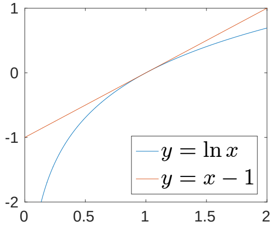
$$KL(p(x)\|q(x)) = \sum_{x\in\mathcal{X}} p(x)\log_2\frac{p(x)}{q(x)}$$
$$= -\sum_{x\in\mathcal{X}} p(x)\log_2\frac{q(x)}{p(x)}$$

KL-divergence is non-negative

$$KL(p(x)\|q(x)) = \sum_{x\in\mathcal{X}} p(x)\log_2\frac{p(x)}{q(x)}$$
$$= -\sum_{x\in\mathcal{X}} p(x)\log_2\frac{q(x)}{p(x)}$$
$$= -\sum_{x\in\mathcal{X}} \frac{p(x)}{\ln 2}\ln\frac{q(x)}{p(x)}$$

KL-divergence is non-negative

$$KL(p(x)\|q(x)) = \sum_{x\in\mathcal{X}} p(x)\log_2\frac{p(x)}{q(x)}$$
$$= -\sum_{x\in\mathcal{X}} p(x)\log_2\frac{q(x)}{p(x)}$$
$$= -\sum_{x\in\mathcal{X}} \frac{p(x)}{\ln 2}\ln\frac{q(x)}{p(x)}$$

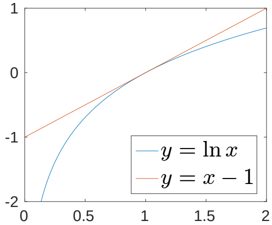


Fact

For any real x, $\ln(x) \le x - 1$. Moreover, the equality only holds when $x = 1$

KL-divergence is non-negative

$$
\begin{aligned}
KL(p(x)\|q(x)) &= \sum_{x\in\mathcal{X}} p(x)\log_2\frac{p(x)}{q(x)} \\
&= -\sum_{x\in\mathcal{X}} p(x)\log_2\frac{q(x)}{p(x)} \\
&= -\sum_{x\in\mathcal{X}} \frac{p(x)}{\ln 2}\ln\frac{q(x)}{p(x)} \\
&\geq -\sum_{x\in\mathcal{X}} \frac{p(x)}{\ln 2}\left(\frac{q(x)}{p(x)}-1\right)
\end{aligned}
$$

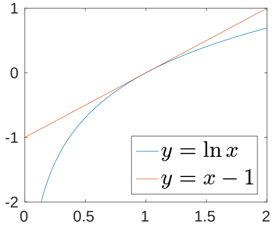


Fact

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 Σ, 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 Σ, 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 Σ, 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 Σ, first note that $\int_x f(\mathbf{x})\log\phi(\mathbf{x})dx=\int_x \phi(\mathbf{x})\log\phi(\mathbf{x})dx$ (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 Σ, 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 Σ, first note that $\int_x f(\mathbf{x})\log\phi(\mathbf{x})dx=\int_x\phi(\mathbf{x})\log\phi(\mathbf{x})dx$ (to be show in the next slide). Then,

$$0\leq KL(f\|\phi)=\int_x f(\mathbf{x})\log\frac{f(\mathbf{x})}{\phi(\mathbf{x})}dx$$

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 Σ, 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 Σ, first note that $\int_x f(\mathbf{x}) \log \phi(\mathbf{x}) dx = \int_x \phi(\mathbf{x}) \log \phi(\mathbf{x}) dx$ (to be show in the next slide). Then,

$$0 \leq KL(f \| \phi) = \int_x f(\mathbf{x}) \log \frac{f(\mathbf{x})}{\phi(\mathbf{x})} dx = -h(f) - \int_x f(\mathbf{x}) \log \phi(\mathbf{x}) dx$$

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 Σ, 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 Σ, first note that $\int_x f(\mathbf{x})\log\phi(\mathbf{x})dx=\int_x \phi(\mathbf{x})\log\phi(\mathbf{x})dx$ (to be show in the next slide). Then,

$$
\begin{aligned}
0 \leq & KL(f\|\phi)=\int_x f(\mathbf{x})\log\frac{f(\mathbf{x})}{\phi(\mathbf{x})}dx=-h(f)-\int_x f(\mathbf{x})\log\phi(\mathbf{x})dx \\
= & -h(f)-\int_x \phi(\mathbf{x})\log\phi(\mathbf{x})dx=-h(f)+h(\phi)
\end{aligned}
$$

$\int_x f(\mathbf{x}) \log \phi(\mathbf{x}) dx = \int_x \phi(\mathbf{x}) \log \phi(\mathbf{x}) dx$

$$\int_x \phi(\mathbf{x}) \log \phi(\mathbf{x}) dx = \int_x \phi(\mathbf{x}) \left[-\log \sqrt{\det(2\pi\Sigma)} - \frac{1}{2}\mathbf{x}^T \Sigma^{-1} \mathbf{x} \right] dx$$

$\int_x f(\mathbf{x}) \log \phi(\mathbf{x}) dx = \int_x \phi(\mathbf{x}) \log \phi(\mathbf{x}) dx$

$$\int_x \phi(\mathbf{x}) \log \phi(\mathbf{x}) dx = \int_x \phi(\mathbf{x}) \left[-\log \sqrt{\det(2\pi\Sigma)} - \frac{1}{2}\mathbf{x}^T \Sigma^{-1} \mathbf{x} \right] dx$$

$$= \int_x \phi(\mathbf{x}) \left[-\log \sqrt{\det(2\pi\Sigma)} - \frac{1}{2} \sum_{i,j} x_i \left[\Sigma^{-1}\right]_{i,j} x_j \right] dx$$

$\int_x f(\mathbf{x}) \log \phi(\mathbf{x}) dx = \int_x \phi(\mathbf{x}) \log \phi(\mathbf{x}) dx$

$$\int_x \phi(\mathbf{x}) \log \phi(\mathbf{x}) dx = \int_x \phi(\mathbf{x}) \left[-\log \sqrt{\det(2\pi\Sigma)} - \frac{1}{2}\mathbf{x}^T \Sigma^{-1} \mathbf{x} \right] dx$$

$$= \int_x \phi(\mathbf{x}) \left[-\log \sqrt{\det(2\pi\Sigma)} - \frac{1}{2} \sum_{i,j} x_i \left[\Sigma^{-1}\right]_{i,j} x_j \right] dx$$

$$= \int_x \phi(\mathbf{x}) \left[-\log \sqrt{\det(2\pi\Sigma)} - \frac{1}{2} \sum_{i,j} \left[\Sigma^{-1}\right]_{i,j} x_i x_j \right] dx$$

$\int_x f(\mathbf{x}) \log \phi(\mathbf{x}) dx = \int_x \phi(\mathbf{x}) \log \phi(\mathbf{x}) dx$

$$\int_x \phi(\mathbf{x}) \log \phi(\mathbf{x}) dx = \int_x \phi(\mathbf{x}) \left[-\log \sqrt{\det(2\pi\Sigma)} - \frac{1}{2}\mathbf{x}^T \Sigma^{-1} \mathbf{x} \right] dx$$

$$= \int_x \phi(\mathbf{x}) \left[-\log \sqrt{\det(2\pi\Sigma)} - \frac{1}{2} \sum_{i,j} x_i \left[\Sigma^{-1}\right]_{i,j} x_j \right] dx$$

$$= \int_x \phi(\mathbf{x}) \left[-\log \sqrt{\det(2\pi\Sigma)} - \frac{1}{2} \sum_{i,j} \left[\Sigma^{-1}\right]_{i,j} x_i x_j \right] dx$$

$$= \int_x f(\mathbf{x}) \left[-\log \sqrt{\det(2\pi\Sigma)} - \frac{1}{2} \sum_{i,j} \left[\Sigma^{-1}\right]_{i,j} x_i x_j \right] dx$$

$\int_x f(\mathbf{x}) \log \phi(\mathbf{x}) dx = \int_x \phi(\mathbf{x}) \log \phi(\mathbf{x}) dx$

$$\int_x \phi(\mathbf{x}) \log \phi(\mathbf{x}) dx = \int_x \phi(\mathbf{x}) \left[-\log \sqrt{\det(2\pi\Sigma)} - \frac{1}{2} \mathbf{x}^T \Sigma^{-1} \mathbf{x} \right] dx$$

$$= \int_x \phi(\mathbf{x}) \left[-\log \sqrt{\det(2\pi\Sigma)} - \frac{1}{2} \sum_{i,j} x_i \left[\Sigma^{-1}\right]_{i,j} x_j \right] dx$$

$$= \int_x \phi(\mathbf{x}) \left[-\log \sqrt{\det(2\pi\Sigma)} - \frac{1}{2} \sum_{i,j} \left[\Sigma^{-1}\right]_{i,j} x_i x_j \right] dx$$

$$= \int_x f(\mathbf{x}) \left[-\log \sqrt{\det(2\pi\Sigma)} - \frac{1}{2} \sum_{i,j} \left[\Sigma^{-1}\right]_{i,j} x_i x_j \right] dx$$

$$= \int_x f(\mathbf{x}) \log \phi(\mathbf{x}) dx$$

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 affliation 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 affliation 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

$$KL(p_{target} \| p_{computed}) = \sum_{group} p_{target}(group) \log \frac{p_{target}(group)}{p_{computed}(group)}$$

$$= -H(p_{target}) \underbrace{- \sum_{group} p_{target}(group) \log p_{computed}(group)}_{cross\ \ entropy}$$

(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

$$
\begin{aligned}
Cross\ \ entropy(p\|q) &\triangleq \sum_x p(x)\log\frac{1}{q(x)} = E_p[-\log q(X)] \\
&= H(p) + KL(p\|q)
\end{aligned}
$$

Application: Cross-entropy and cross-entropy error

$$Cross\ \ entropy(p\|q) \triangleq \sum_x p(x) \log \frac{1}{q(x)} = E_p[-\log q(X)]$$
$$= H(p) + KL(p\|q)$$

- 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

$$
\begin{aligned}
Cross\ \ entropy(p\|q) &\triangleq \sum_x p(x)\log\frac{1}{q(x)} = E_p[-\log q(X)] \\
&= H(p) + KL(p\|q)
\end{aligned}
$$

- 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

Application: Text processing

- In text processing, it is common that one may need to measure the similiarity between two documents D_1 and D_2.

Application: Text processing

- In text processing, it is common that one may need to measure the similiarity 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

Application: Text processing

- In text processing, it is common that one may need to measure the similiarity 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

$$Cross\ entropy(p_1 \| p_2) = \sum_w p_1(w) \log \frac{1}{p_2(w)},$$

where p_1 and p_2 are the word distributions of documents D_1 and D_2, respectively

Application: Text processing

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,

$$
\begin{aligned}
Cross\ entropy(p_1\|q) &= \sum_w p_1(w)\log\frac{1}{q(w)} \\
&= \sum_w \underbrace{\frac{\#\ w \text{ in } D_1}{\text{total } \#\text{ words in } D_1}\log\frac{\text{total }\#\text{ docs}}{\#\text{ doc with } w}}_{TF\text{-}IDF(w)},
\end{aligned}
$$

where $TF\text{-}IDF(w)$, 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.

$$
\begin{aligned}
I(X;Y) =& H(X) - H(X|Y) = E[-\log p(X)] - E[-\log p(X|Y)] \\
=& -\sum_x p(x)\log p(x) + \sum_{x,y} p(x,y)\log p(x|y) \\
=& -\sum_{x,y} p(x,y)\log p(x) + \sum_{x,y} p(x,y)\log p(x|y) = \sum_{x,y} p(x,y)\log\frac{p(x|y)}{p(x)}
\end{aligned}
$$

Property of mutual information

$I(X;Y) = I(Y;X) \geq 0$

The definition is symmetric and non-negative as desired.

$$
\begin{aligned}
I(X;Y) =& H(X) - H(X|Y) = E[-\log p(X)] - E[-\log p(X|Y)] \\
&= -\sum_{x} p(x) \log p(x) + \sum_{x,y} p(x,y) \log p(x|y) \\
&= -\sum_{x,y} p(x,y) \log p(x) + \sum_{x,y} p(x,y) \log p(x|y) = \sum_{x,y} p(x,y) \log \frac{p(x|y)}{p(x)} \\
&= \sum_{x,y} p(x,y) \log \frac{p(x,y)}{p(x)p(y)}
\end{aligned}
$$

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.

$$I(X;Y|Z) = H(X|Z) - H(X|Y,Z) = E[-\log p(X|Z)] - E[-\log p(X|Y,Z)]$$
$$= -\sum_{x,z} p(x,z)\log p(x|z) + \sum_{x,y,z} p(x,y,z)\log p(x|y,z)$$

Property of conditional mutual information

$I(X;Y|Z) = I(Y;X|Z) \geq 0$

The definition is symmetric and non-negative as desired.

$$
\begin{aligned}
I(X;Y|Z) =& H(X|Z) - H(X|Y,Z) = E[-\log p(X|Z)] - E[-\log p(X|Y,Z)] \\
&= -\sum_{x,z} p(x,z)\log p(x|z) + \sum_{x,y,z} p(x,y,z)\log p(x|y,z) \\
&= -\sum_{x,y,z} p(x,y,z)\log p(x|z) + \sum_{x,y,z} p(x,yz)\log p(x|y,z) \\
&= \sum_{x,y,z} p(x,y,z)\log \frac{p(x|y,z)}{p(x|z)}
\end{aligned}
$$

Property of conditional mutual information

$I(X;Y|Z) = I(Y;X|Z) \geq 0$

The definition is symmetric and non-negative as desired.

$$\begin{aligned}
I(X;Y|Z) =& H(X|Z) - H(X|Y,Z) = E[-\log p(X|Z)] - E[-\log p(X|Y,Z)] \\
=& -\sum_{x,z} p(x,z) \log p(x|z) + \sum_{x,y,z} p(x,y,z) \log p(x|y,z) \\
=& -\sum_{x,y,z} p(x,y,z) \log p(x|z) + \sum_{x,y,z} p(x,yz) \log p(x|y,z) \\
=& \sum_{x,y,z} p(x,y,z) \log \frac{p(x|y,z)}{p(x|z)} \\
=& \sum_{z} p(z) \sum_{x,y} p(x,y|z) \log \frac{p(x,y|z)}{p(x|z)p(y|z)} \\
=& \sum_{z} p(z) KL(p(x,y|z) \| p(x|z)p(y|z)) \geq 0
\end{aligned}$$

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 indepedent!

Chain rule for mutual information

$$I(X_1, X_2, \cdots, X_N | Y)$$

Chain rule for mutual information

$$
\begin{aligned}
&I(X_1, X_2, \cdots, X_N|Y) \\
=&H(X_1, X_2, \cdots, X_N) - H(X_1, X_2, \cdots, X_N|Y)
\end{aligned}
$$

Chain rule for mutual information

$$
\begin{aligned}
&I(X_1, X_2, \cdots, X_N|Y) \\
=&H(X_1, X_2, \cdots, X_N) - H(X_1, X_2, \cdots, X_N|Y) \\
=&\sum_{i=1}^{N} H(X_i|X^{i-1}) - H(X_i|X^{i-1}, Y)
\end{aligned}
$$

N.B. $X^N = X_1, X_2, \cdots, X_N$

Chain rule for mutual information

$$
\begin{aligned}
& I(X_1, X_2, \cdots, X_N|Y) \\
=& H(X_1, X_2, \cdots, X_N) - H(X_1, X_2, \cdots, X_N|Y) \\
=& \sum_{i=1}^{N} H(X_i|X^{i-1}) - H(X_i|X^{i-1}, Y) \\
=& \sum_{i=1}^{N} I(X_i; Y|X^{i-1})
\end{aligned}
$$

N.B. $X^N = X_1, X_2, \cdots, X_N$

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

- $I(X;Y) = KL(p(x,y)\|p(x)p(y)) = I(Y;X) \geq 0$
- $I(X;Y|Z) = \int_z p(z)KL(p(x,y|z)\|p(x|z)p(y|z))dz = I(Y;X|Z) \geq 0$
- $I(X;Y) = 0 \Leftrightarrow X \perp Y$
- $I(X;Y|Z) = 0 \Leftrightarrow X \perp Y|Z$
- $I(X_1, X_2, \cdots, X_N|Y) = \sum_{i=1}^N I(X_i; Y|X^{i-1})$

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,

$$H(X) \geq H(X|Y) \qquad H(X|Y) \geq H(X|Y,Z)$$

This is obvious from our previous discussion since
$H(X) - H(X|Y) = I(X;Y) \geq 0$ 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,

$$H(X) \geq H(X|Y) \qquad H(X|Y) \geq H(X|Y,Z)$$

This is obvious from our previous discussion since
$H(X) - H(X|Y) = I(X;Y) \geq 0$ and
$H(X|Y) - H(X|Y,Z) = I(X;Z|Y) \geq 0$

Of course, we also have

$$h(X) \geq h(X|Y) \qquad h(X|Y) \geq h(X|Y,Z)$$

since $h(X) - h(X|Y) = I(X;Y) \geq 0$ and
$h(X|Y) - h(X|Y) = I(X;Z|Y) \geq 0$

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

$$\begin{aligned} I(X;Y) &= I(X;Y,Z) - I(X;Z|Y) \\ &= I(X;Y,Z) \qquad\qquad (\text{since } X \leftrightarrow Y \leftrightarrow Z) \end{aligned}$$

Data processing inequality

If random variables X, Y, Z satisfy $X \leftrightarrow Y \leftrightarrow Z$, then

$$I(X;Y) \geq I(X;Z).$$

Proof

$$\begin{aligned}
I(X;Y) &= I(X;Y,Z) - I(X;Z|Y) \\
&= I(X;Y,Z) \qquad\qquad (\text{since } X \leftrightarrow Y \leftrightarrow Z) \\
&= I(X;Z) + I(X;Y|Z) \\
&\geq I(X;Z)
\end{aligned}$$

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, $H(M|K,C) = 0 \Rightarrow$
$H(M|C) \leq H(M,K|C) = H(K|C) + H(M|K,C) = H(K|C)$ □

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, $H(M|K,C) = 0 \Rightarrow H(M|C) \leq H(M,K|C) = H(K|C) + H(M|K,C) = H(K|C)$ □

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, $H(M|K,C) = 0 \Rightarrow$
$H(M|C) \leq H(M,K|C) = H(K|C) + H(M|K,C) = H(K|C)$ □

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) □

Summary