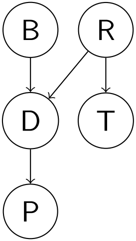
This time...

- Bayesian Net
- Belief Propagation Algorithm
- LDPC/IRA Codes

Bayesian Net

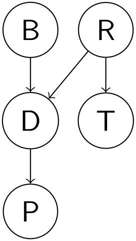
- Relationship of variables depicted by a directed graph with no loop
- Given a variable's parents, the variable is conditionally independent of any non-descendants
- Reduce model complexity
- Facilitate easier inference



Burlgar and racoon

Burlgar: B; Racoon: R; Dog barked: D; Police called: P; Trash can fell: T

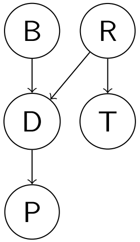
$$p(p,d,b,t,r) = p(p|d,b,t,r)p(d|b,t,r)p(b|t,r)p(t|r)p(r)$$



Burlgar and racoon

Burlgar: B; Racoon: R; Dog barked: D; Police called: P; Trash can fell: T

$$
\begin{aligned}
p(p,d,b,t,r) &= p(p|d,b,t,r)p(d|b,t,r)p(b|t,r)p(t|r)p(r) \\
&= \underbrace{p(p|d,\cancel{b},\cancel{t},\cancel{r})}_{2\ parameters}p(d|b,\cancel{t},r)p(b|\cancel{t},\cancel{r})p(t|r)p(r)
\end{aligned}
$$



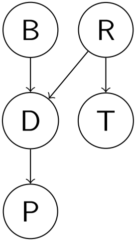
Burlgar and racoon

Burlgar: B; Racoon: R; Dog barked: D; Police called: P; Trash can fell: T

$$p(p,d,b,t,r) = p(p|d,b,t,r)p(d|b,t,r)p(b|t,r)p(t|r)p(r)$$
$$= \underbrace{p(p|d,\cancel{b},\cancel{t},\cancel{r})}_{2\ parameters}p(d|b,\cancel{t},r)p(b|\cancel{t},\cancel{r})p(t|r)p(r)$$

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

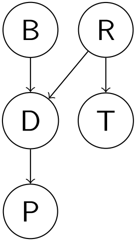
Burlgar: B; Racoon: R; Dog barked: D; Police called: P; Trash can fell: T

$$\begin{aligned} p(p,d,b,t,r) =& p(p|d,b,t,r)p(d|b,t,r)p(b|t,r)p(t|r)p(r) \\ =& \underbrace{p(p|d,\cancel{b},\cancel{t},\cancel{r})}_{2 \ parameters} p(d|b,\cancel{t},r)p(b|\cancel{t},\cancel{r})p(t|r)p(r) \end{aligned}$$

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

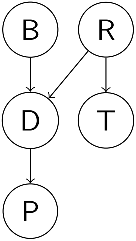
T	R	$p(t\|r)$
t	$\neg r$	0.05
t	r	0.7
$\neg t$	$\neg r$	0.95
$\neg t$	r	0.3

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



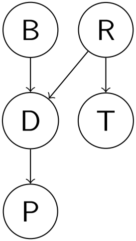
Comparison of # parameters

- # parameters of complete model: $2^5 - 1 = 31$



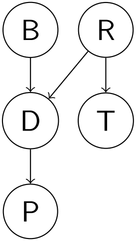
Comparison of # parameters

- # parameters of complete model: $2^5 - 1 = 31$
- # parameters of Bayesian net:



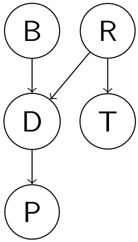
Comparison of # parameters

- # parameters of complete model: $2^5 - 1 = 31$
- # parameters of Bayesian net:
  - $p(p|d)$: 2



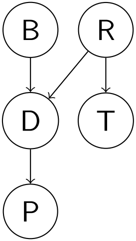
Comparison of # parameters

- # parameters of complete model: $2^5 - 1 = 31$
- # parameters of Bayesian net:
  - $p(p|d)$: 2
  - $p(d|b,r)$: 4



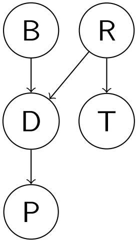
Comparison of # parameters

- # parameters of complete model: $2^5 - 1 = 31$
- # parameters of Bayesian net:
    - $p(p|d)$: 2
    - $p(d|b,r)$: 4
    - $p(b)$: 1



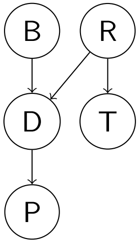
Comparison of # parameters

- # 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
$\cdots$				

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
$\cdots$				

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
$\cdots$				

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
$\cdots$					

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
$\cdots$					

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
$\cdots$					

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
$\cdots$					

Burglar and racoon

Question: What is the probability of a burglar visit if police was called but trash can stayed untouched?

$$p(b|\neg t, p) = 0.058099 + 0.0045868 \approx 0.0626$$

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
$\cdots$					

Belief Propagation Algorithm

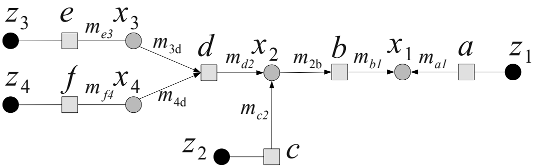
- It is also known to be the sum-product algorithm
- The goal of belief propagation is to efficiently compute the marginal distribution out of the joint distribution of multiple variables. This is essential for inferring the outcome of a particular variable with insufficient information
- The belief 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

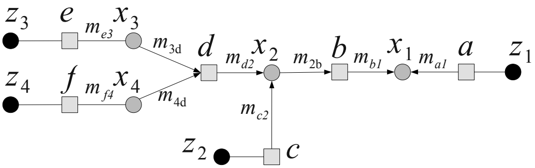
An Example

- 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



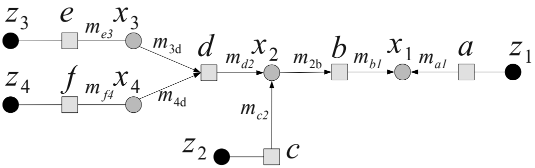
An Example

- 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

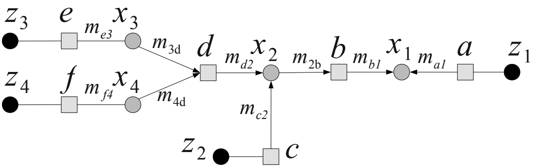
- A factor graph example is shown below. We have 8 *discrete* random variables, x_1^4 and z_1^4, depicted by 8 variable nodes
- Among the variable nodes, random variables x_1^4 (indicated by light circles) are unknown and variables z_1^4 (indicated by dark circles) are observed with known outcomes $\tilde{z}_1^4$
- 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

$$p(x^4, z^4) = p(x^4)p(z_1|x_1)p(z_2|x_2)p(z_3|x_3)p(z_4|x_4)$$

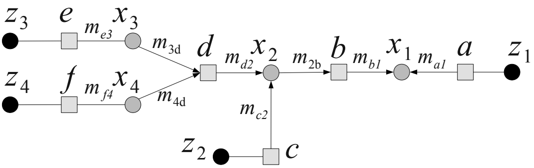
- Note that each factor function corresponds to a factor node in the factor graph.
- The arguments of the factor function correspond to the variable nodes that the factor node connects to.



- 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

$$\begin{aligned} p(x^4, z^4) &= p(x^4)p(z_1|x_1)p(z_2|x_2)p(z_3|x_3)p(z_4|x_4) \\ &= \underbrace{p(x_1, x_2)}_{f_b(x_1,x_2)}\underbrace{p(x_3, x_4|x_2)}_{f_d(x_2,x_3,x_4)}\underbrace{p(z_3|x_3)}_{f_e(x_3,z_3)}\underbrace{p(z_1|x_1)}_{f_a(x_1,z_1)}\underbrace{p(z_4|x_4)}_{f_f(x_4,z_4)}\underbrace{p(z_2|x_2)}_{f_c(x_2,z_2)} \end{aligned}$$

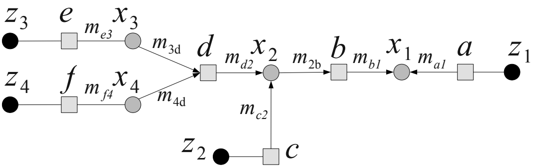
- Note that each factor function corresponds to a factor node in the factor graph.
- The arguments of the factor function correspond to the variable nodes that the factor node connects to.



- 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

$$p(x_1, \tilde{z}^4) = \sum_{x_2^4} p(x^4, \tilde{z}^4)$$

$$= \sum_{x_2^4} f_a(x_1, \tilde{z}_1) f_b(x_1, x_2) f_c(x_2, \tilde{z}_2) f_d(x_2, x_3, x_4) f_e(x_3, \tilde{z}_3) f_f(x_4, \tilde{z}_4)$$

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

$$p(x_1, \tilde{z}^4) = \sum_{x_2^4} p(x^4, \tilde{z}^4)$$

$$= \sum_{x_2^4} f_a(x_1, \tilde{z}_1) f_b(x_1, x_2) f_c(x_2, \tilde{z}_2) f_d(x_2, x_3, x_4) f_e(x_3, \tilde{z}_3) f_f(x_4, \tilde{z}_4)$$

$$= \underbrace{f_a(x_1, \tilde{z}_1)}_{m_{a1}} \underbrace{\sum_{x_2} f_b(x_1, x_2) \underbrace{\underbrace{f_c(x_2, \tilde{z}_2)}_{m_{c2}} \underbrace{\sum_{x_3, x_4} f_d(x_2, x_3, x_4) \underbrace{f_e(x_3, \tilde{z}_3)}_{m_{3d}} \underbrace{f_f(x_4, \tilde{z}_4)}_{m_{4d}}}_{m_{d2}}}_{m_{2b}}}_{m_{b1}}$$

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

$$m_{a1}(x_1) \leftarrow f_a(x_1, \tilde{z}_1) = \sum_{z_1} f_a(x_1, z_1) \underbrace{p(z_1)}_{m_{1a}},$$

$$m_{c2}(x_2) \leftarrow f_c(x_2, \tilde{z}_2) = \sum_{z_2} f_c(x_2, z_2) \underbrace{p(z_2)}_{m_{2c}},$$

$$m_{e3}(x_3) \leftarrow f_e(x_3, \tilde{z}_3) = \sum_{z_3} f_e(x_3, z_3) \underbrace{p(z_3)}_{m_{3e}},$$

$$m_{f4}(x_4) \leftarrow f_f(x_4, \tilde{z}_4) = \sum_{z_4} f_f(x_4, z_4) \underbrace{p(z_4)}_{m_{4f}},$$

where $p(z_i) = \begin{cases} 1, & z_i = \tilde{z}_i \\ 0, & \text{otherwise} \end{cases}$

$$p(x_1,\tilde{z}^4)=\underbrace{f_a(x_1,\tilde{z}_1)}_{m_{a1}}\underbrace{\sum_{x_2} f_b(x_1,x_2)\underbrace{\underbrace{f_c(x_2,\tilde{z}_2)}_{m_{c2}}\underbrace{\sum_{x_3,x_4} f_d(x_2,x_3,x_4)\underbrace{f_e(x_3,\tilde{z}_3)}_{m_{3d}}\underbrace{f_f(x_4,\tilde{z}_4)}_{m_{4d}}}_{m_{d2}}}_{m_{2b}}}_{m_{b1}} \quad (1)$$

$$m_{3d}(x_3) \leftarrow m_{e3}(x_3) = f_e(x_3, \tilde{z}_3),$$
$$m_{4d}(x_4) \leftarrow m_{f4}(x_4) = f_f(x_4, \tilde{z}_4),$$

$$p(x_1, \tilde{z}^4) = \underbrace{f_a(x_1, \tilde{z}_1)}_{m_{a1}} \underbrace{\sum_{x_2} f_b(x_1, x_2) \underbrace{\underbrace{f_c(x_2, \tilde{z}_2)}_{m_{c2}} \underbrace{\sum_{x_3, x_4} f_d(x_2, x_3, x_4) \underbrace{f_e(x_3, \tilde{z}_3)}_{\color{red}{m_{3d}}} \underbrace{f_f(x_4, \tilde{z}_4)}_{\color{red}{m_{4d}}}}_{m_{d2}}}_{m_{2b}}}_{m_{b1}} \quad (1)$$

$$m_{3d}(x_3) \leftarrow m_{e3}(x_3) = f_e(x_3, \tilde{z}_3),$$
$$m_{4d}(x_4) \leftarrow m_{f4}(x_4) = f_f(x_4, \tilde{z}_4),$$
$$m_{d2}(x_2) \leftarrow \sum_{x_3, x_4} f_d(x_2, x_3, x_4) m_{3d}(x_3) m_{4d}(x_4),$$

$$p(x_1, \tilde{z}^4) = \underbrace{f_a(x_1, \tilde{z}_1)}_{m_{a1}} \underbrace{\sum_{x_2} f_b(x_1, x_2) \underbrace{\underbrace{f_c(x_2, \tilde{z}_2)}_{m_{c2}} \underbrace{\sum_{x_3, x_4} f_d(x_2, x_3, x_4) \underbrace{f_e(x_3, \tilde{z}_3)}_{m_{3d}} \underbrace{f_f(x_4, \tilde{z}_4)}_{m_{4d}}}_{\color{red}{m_{d2}}}}_{m_{2b}}}_{m_{b1}} \quad (1)$$

$$
\begin{aligned}
m_{3d}(x_3) &\leftarrow m_{e3}(x_3) = f_e(x_3, \tilde{z}_3), \\
m_{4d}(x_4) &\leftarrow m_{f4}(x_4) = f_f(x_4, \tilde{z}_4), \\
m_{d2}(x_2) &\leftarrow \sum_{x_3, x_4} f_d(x_2, x_3, x_4) m_{3d}(x_3) m_{4d}(x_4), \\
m_{2b}(x_2) &\leftarrow m_{c2}(x_2) m_{d2}(x_2),
\end{aligned}
$$

$$
p(x_1, \tilde{z}^4) = \underbrace{f_a(x_1, \tilde{z}_1)}_{m_{a1}} \underbrace{\sum_{x_2} f_b(x_1, x_2) \underbrace{\underbrace{f_c(x_2, \tilde{z}_2)}_{m_{c2}} \underbrace{\sum_{x_3, x_4} f_d(x_2, x_3, x_4) \underbrace{f_e(x_3, \tilde{z}_3)}_{m_{3d}} \underbrace{f_f(x_4, \tilde{z}_4)}_{m_{4d}}}_{m_{d2}}}_{m_{2b}}}_{m_{b1}} \tag{1}
$$

$$
\begin{aligned}
m_{3d}(x_3) &\leftarrow m_{e3}(x_3) = f_e(x_3, \tilde{z}_3), \\
m_{4d}(x_4) &\leftarrow m_{f4}(x_4) = f_f(x_4, \tilde{z}_4), \\
m_{d2}(x_2) &\leftarrow \sum_{x_3, x_4} f_d(x_2, x_3, x_4) m_{3d}(x_3) m_{4d}(x_4), \\
m_{2b}(x_2) &\leftarrow m_{c2}(x_2) m_{d2}(x_2), \\
m_{b1}(x_1) &\leftarrow \sum_{x_2} f_b(x_1, x_2) m_{2b}(x_2),
\end{aligned}
$$

$$
p(x_1, \tilde{z}^4) = \underbrace{f_a(x_1, \tilde{z}_1)}_{m_{a1}} \underbrace{\sum_{x_2} f_b(x_1, x_2) \underbrace{\underbrace{f_c(x_2, \tilde{z}_2)}_{m_{c2}} \underbrace{\sum_{x_3, x_4} f_d(x_2, x_3, x_4) \underbrace{f_e(x_3, \tilde{z}_3)}_{m_{3d}} \underbrace{f_f(x_4, \tilde{z}_4)}_{m_{4d}}}_{m_{d2}}}_{m_{2b}}}_{\color{red}{m_{b1}}} \quad (1)
$$

$$
\begin{aligned}
m_{3d}(x_3) &\leftarrow m_{e3}(x_3) = f_e(x_3, \tilde{z}_3), \\
m_{4d}(x_4) &\leftarrow m_{f4}(x_4) = f_f(x_4, \tilde{z}_4), \\
m_{d2}(x_2) &\leftarrow \sum_{x_3, x_4} f_d(x_2, x_3, x_4) m_{3d}(x_3) m_{4d}(x_4), \\
m_{2b}(x_2) &\leftarrow m_{c2}(x_2) m_{d2}(x_2), \\
m_{b1}(x_1) &\leftarrow \sum_{x_2} f_b(x_1, x_2) m_{2b}(x_2), \\
p(x_1, \tilde{z}^4) &\leftarrow m_{a1}(x_1) m_{b1}(x_1),
\end{aligned}
$$

$$
p(x_1, \tilde{z}^4) = \underbrace{f_a(x_1, \tilde{z}_1)}_{m_{a1}} \underbrace{\sum_{x_2} f_b(x_1, x_2) \underbrace{\underbrace{f_c(x_2, \tilde{z}_2)}_{m_{c2}} \underbrace{\sum_{x_3, x_4} f_d(x_2, x_3, x_4) \underbrace{f_e(x_3, \tilde{z}_3)}_{m_{3d}} \underbrace{f_f(x_4, \tilde{z}_4)}_{m_{4d}}}_{m_{d2}}}_{m_{2b}}}_{m_{b1}} \quad (1)
$$

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**:

$$m_{ia}(x_i) \leftarrow \prod_{b \in N(i) \backslash a} m_{bi}(x_i),$$

$$m_{ai}(x_i) \leftarrow \sum_{\mathbf{x}_a} f_a(\mathbf{x}_a) \prod_{j \in N(a) \backslash i} m_{ja}(x_j) \qquad (\text{"sum-product"})$$

- **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**:

$$m_{ia}(x_i) \leftarrow \prod_{b \in N(i) \backslash a} m_{bi}(x_i),$$

$$m_{ai}(x_i) \leftarrow \sum_{\mathbf{x}_a} f_a(\mathbf{x}_a) \prod_{j \in N(a) \backslash i} m_{ja}(x_j) \qquad (\text{“sum-product”})$$

- **Belief update**:

$$\beta_i(x_i) \leftarrow \prod_{a \in N(i)} m_{ai}(x_i)$$

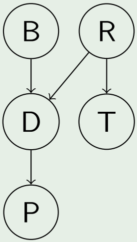
- **Stopping criteria**: repeat message update and/or belief update until the algorithm stops when maximum number of iterations is reached or some other conditions are satisfied.

Remark

- We have not assumed the precise phyical 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

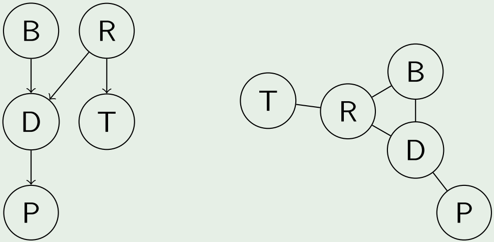
Burglar and racoon revisit

Question: What is the probability of a burglar visit if police was called but trash can stayed untouched?



Burglar and racoon revisit

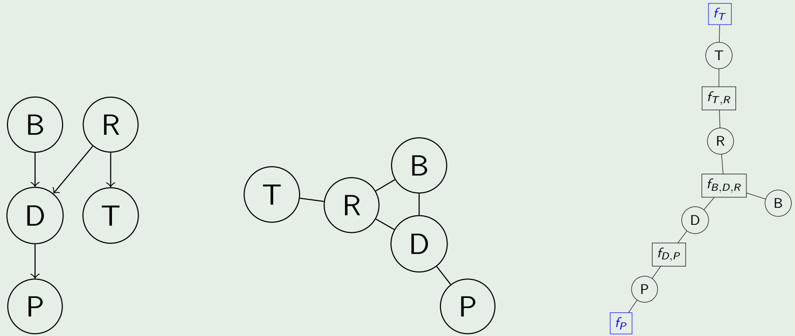
Question: What is the probability of a burglar visit if police was called but trash can stayed untouched?



Moralization...

Burglar and racoon revisit

Question: What is the probability of a burglar visit if police was called but trash can stayed untouched?



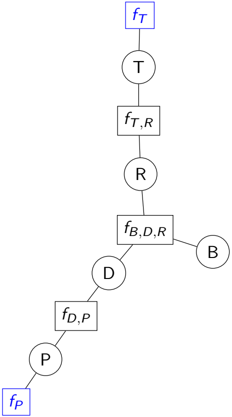
Convert to factor graph..

Using belief propagation...

$$\begin{cases} f_P(p) & = 1 \\ f_P(\neg p) & = 0 \end{cases}$$

$$\begin{cases} f_T(t) & = 0 \\ f_T(\neg t) & = 1 \end{cases}$$

$$\begin{aligned} f_{B,D,R}(b,d,r) &= p(b,d,r) \\ f_{T,R}(t,r) &= p(t|r) \\ f_{D,P}(d,p) &= p(p|d) \end{aligned}$$



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

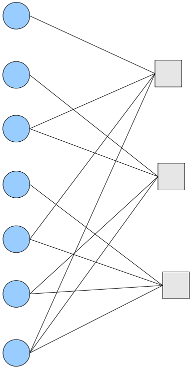
- As its name suggests, LDPC codes refer to codes that with sparse (low-density) parity check matrices. In other words, there are only few ones in a parity check matrix and the rest are all zeros
- We learn from the proof of Channel Coding Theorem that random code is asymptotically optimum. This suggests 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!

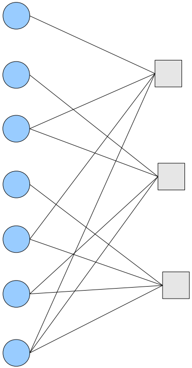
Tanner Graph

- An LDPC code can be represented using a Tanner graph as shown on the right



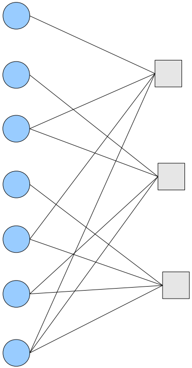
Tanner Graph

- 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



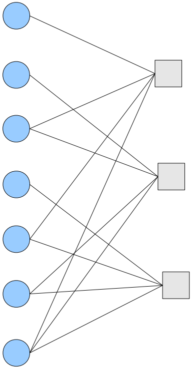
Tanner Graph

- 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



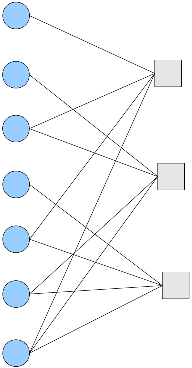
Tanner Graph

- 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 $x_1, x_2, \cdots, x_N$ is a codeword only if all checks are zero



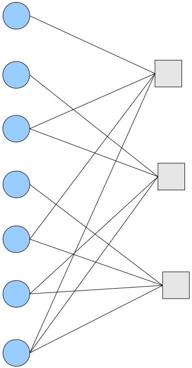
Tanner Graph

- 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 $x_1, x_2, \cdots, x_N$ 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



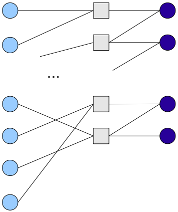
Tanner Graph

- 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 $x_1, x_2, \cdots, x_N$ 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



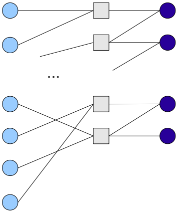
IRA Codes

- Irregular repeated accumulate (IRA) code a type of systematic LDPC code, i.e., each codeword can be partitioned into message bits and syndrome bits



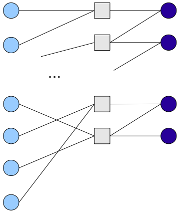
IRA Codes

- 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



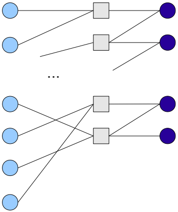
IRA Codes

- 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



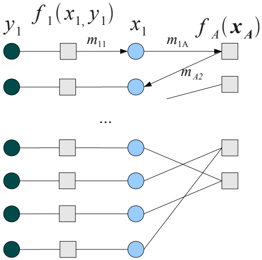
IRA Codes

- 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 conecting to the check + *last syndrome bit*. All (dark blue) syndrome bits can be assigned in similar token



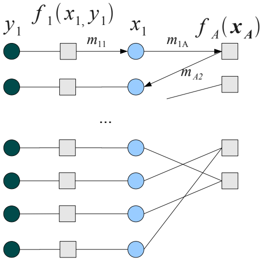
LDPC Decoding

- $x_1, \cdots, x_N$ (light blue): transmitted bits
- $y_1, \cdots, y_N$ (dark grey): received bits



LDPC Decoding

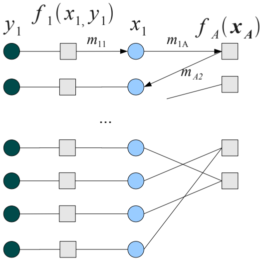
- $x_1, \cdots, x_N$ (light blue): transmitted bits
- $y_1, \cdots, y_N$ (dark grey): received bits
- $p(x^N, y^N) = \prod_i \underbrace{p(y_i|x_i)}_{f_i(x_i,y_i)} \underbrace{p(x^N)}_{\prod_A f_A(\mathbf{x}_A)}$



LDPC Decoding

- $x_1, \cdots, x_N$ (light blue): transmitted bits
- $y_1, \cdots, y_N$ (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

$$l_{ai} \triangleq \frac{m_{ai}(0)}{m_{ai}(1)}, \qquad L_{ai} \triangleq \log l_{ai} \tag{2}$$

and

$$l_{ia} \triangleq \frac{m_{ia}(0)}{m_{ia}(1)}, \qquad L_{ia} \triangleq \log l_{ia} \tag{3}$$

for any variable node i and factor node a.

- Then,

$$L_{ia} \leftarrow \sum_{b \in N(i) \backslash i} L_{ai}. \tag{4}$$

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

$$m_{a1}(1) \leftarrow m_{2a}(1)m_{3a}(0) + m_{2a}(0)m_{3a}(1) \qquad (5)$$
$$m_{a1}(0) \leftarrow m_{2a}(0)m_{3a}(0) + m_{2a}(1)m_{3a}(1) \qquad (6)$$

- Substitute in the likelihood ratios and log-likelihood ratios, we have

$$l_{a1} \triangleq \frac{m_{a1}(0)}{m_{a1}(1)} \leftarrow \frac{1 + l_{2a}l_{3a}}{l_{2a} + l_{3a}} \qquad (7)$$

and

$$e^{L_{a1}} = l_{a1} \leftarrow \frac{1 + e^{L_{2a}}e^{L_{3a}}}{e^{L_{2a}} + e^{L_{3a}}}. \qquad (8)$$

- Note that

$$
\begin{aligned}
\tanh\left(\frac{L_{a1}}{2}\right) &= \frac{e^{\frac{L_{a1}}{2}} - e^{-\frac{L_{a1}}{2}}}{e^{\frac{L_{a1}}{2}} + e^{-\frac{L_{a1}}{2}}} = \frac{e^{L_{a1}} - 1}{e^{L_{a1}} + 1} && (9) \\
&\leftarrow \frac{1 + e^{L_{2a}} e^{L_{3a}} - e^{L_{2a}} - e^{L_{3a}}}{1 + e^{L_{2a}} e^{L_{3a}} + e^{L_{2a}} + e^{L_{3a}}} && (10) \\
&= \frac{(e^{L_{2a}} - 1)(e^{L_{3a}} - 1)}{(e^{L_{2a}} + 1)(e^{L_{3a}} + 1)} && (11) \\
&= \tanh\left(\frac{L_{2a}}{2}\right) \tanh\left(\frac{L_{3a}}{2}\right). && (12)
\end{aligned}
$$

- When we have more than 3 variable nodes connecting to the check node a, it is easy to show using induction that

$$
\tanh\left(\frac{L_{ai}}{2}\right) \leftarrow \prod_{j \in N(a) \backslash i} \tanh\left(\frac{L_{ja}}{2}\right). \qquad (13)
$$