ECE521 Tutorial 11

Topic Review

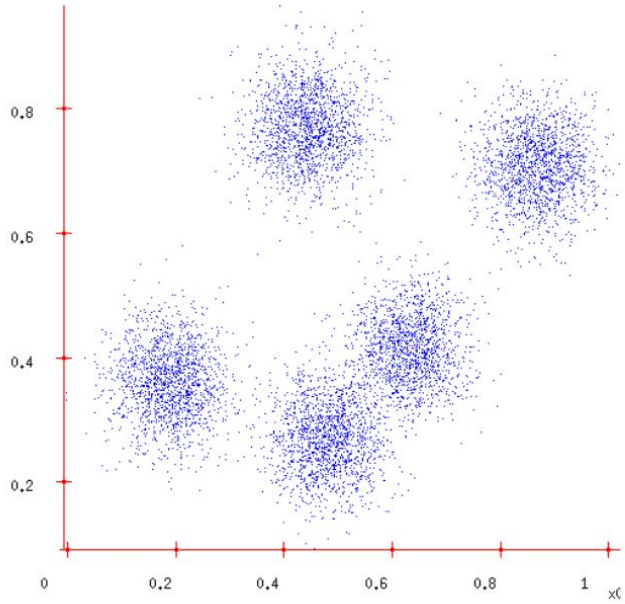
ECE521 Winter 2016

Credits to Alireza Makhzani, Alex Schwing, Rich Zemel and TAs for slides

Outline

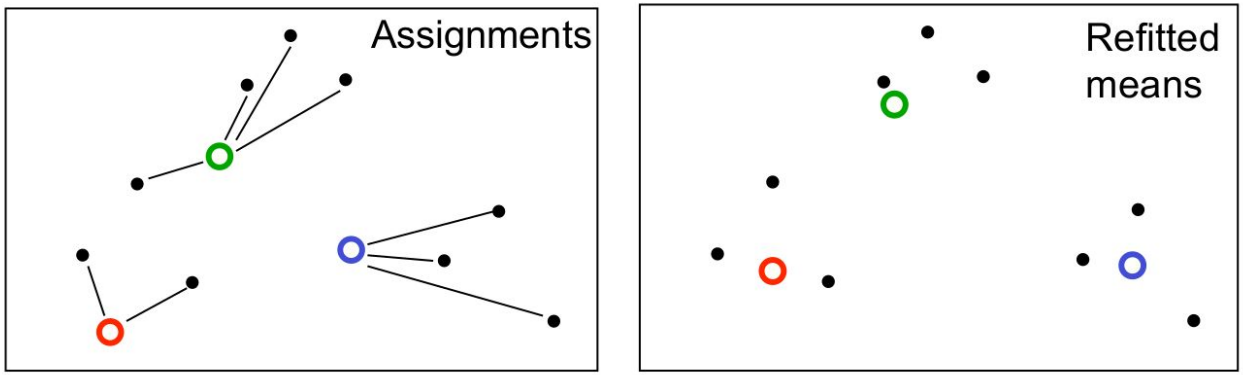
k-Means

- Technique for clustering
- Unsupervised pattern and grouping discovery
- Class prediction
- Outlier detection



k-Means

- Assume the data lives in a Euclidean space.
- Assume we want k classes/patterns
- Initialization: randomly located cluster centers
- The algorithm alternates between two steps:
  - Assignment step: Assign each datapoint to the closest cluster.
  - Refitting step: Move each cluster center to the center of gravity of the data assigned to it.



k-Means

- Define an iterative procedure to minimize:

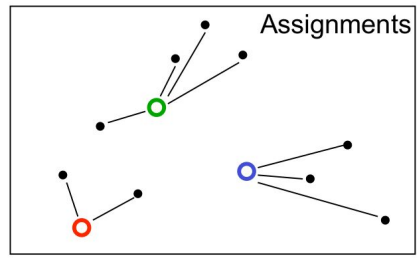
$$J = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} ||\mathbf{x}_n - \boldsymbol{\mu}_k||^2.$$

- Given $\boldsymbol{\mu}_k$, minimize J with respect to r_{nk} (akin to the **E-step** in EM):

$$r_{nk} = \begin{cases} 1 & \text{if } k = \arg\min_j ||\mathbf{x}_n - \boldsymbol{\mu}_j||^2 \\ 0 & \text{otherwise} \end{cases}$$

← Hard assignments of points to clusters

which simply says assign the n^{th} data point $\mathbf{x}_n$ to its closest cluster centre

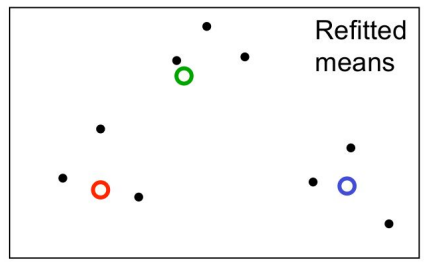


- Given r_{nk}, minimize J with respect to $\boldsymbol{\mu}_k$ (akin to the **M-step**):

$$\boldsymbol{\mu}_k = \frac{\sum_n r_{nk} \mathbf{x}_n}{\sum_n r_{nk}}.$$

← Number of points assigned to cluster k.

Set $\boldsymbol{\mu}_k$ equal to the mean of all the data points assigned to cluster k

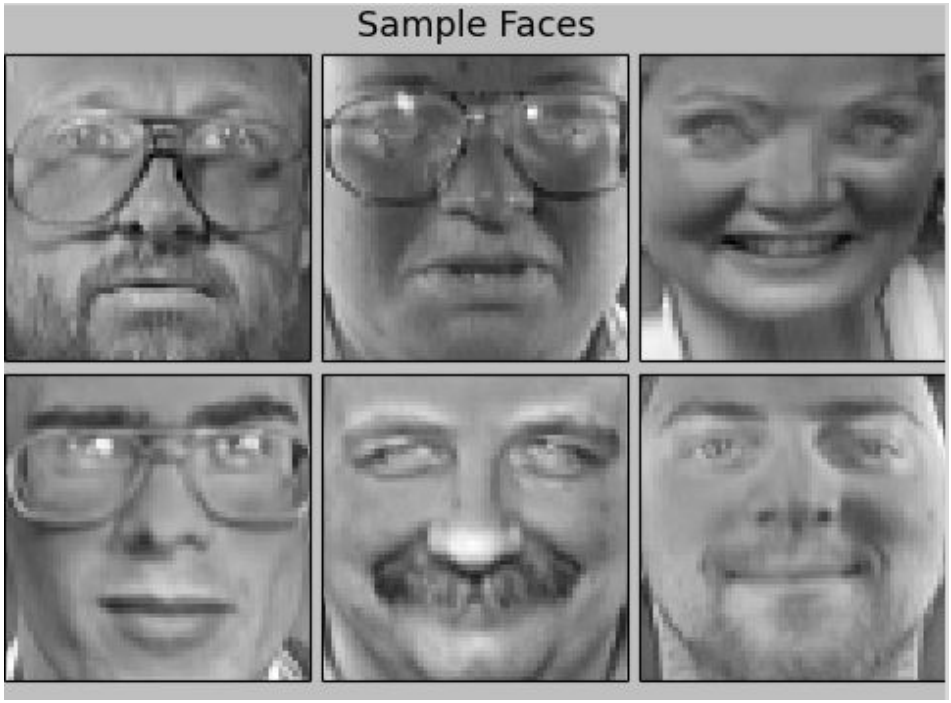


PCA

- Dimensionality Reduction
- Visualization
- Compression

Olivetti Faces Dataset

- Gray scale images
- 64x64
- 10 subjects
- 40 images per subject
- 400 images each
- Problem: 4096-dim feature vector for only 400 datapoints
- Would like: 200 dimensional feature space (each image described by 200 numbers)



PCA

- Algorithm: to find M components underlying D-dimensional data
  1. Select the top M eigenvectors of C (data covariance matrix):

$$C = \frac{1}{N}\sum_{n=1}^{N}(\mathbf{x}^{(n)} - \bar{\mathbf{x}})(\mathbf{x}^{(n)} - \bar{\mathbf{x}})^T = U\Sigma U^T \approx U\Sigma_{1:M}U_{1:M}^T$$

     where U: orthogonal, columns = unit-length eigenvectors

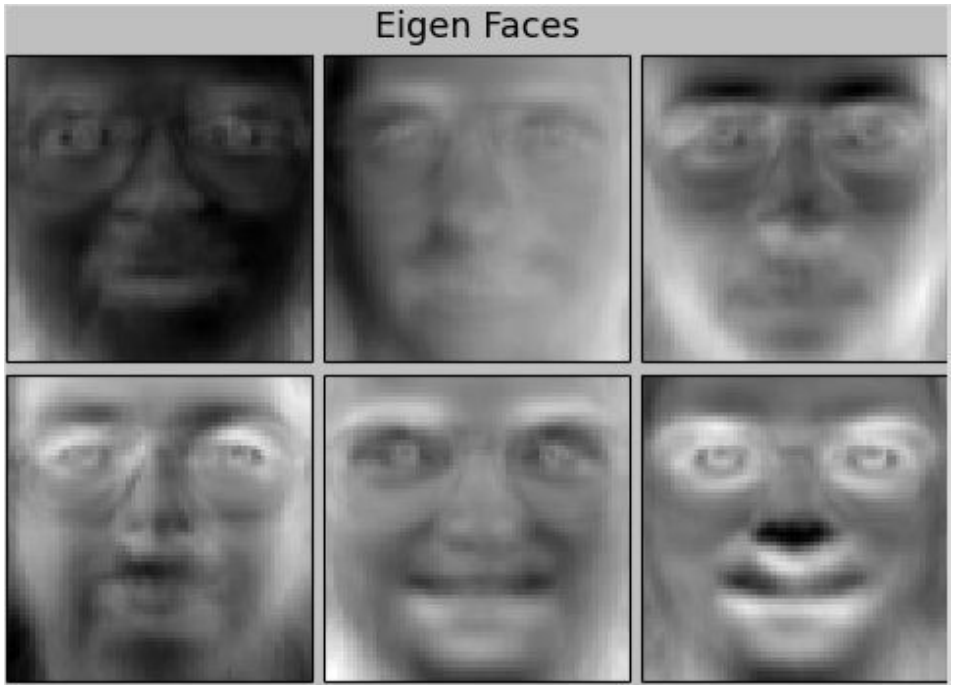
$$U^T U = UU^T = 1$$

     and Σ: matrix with eigenvalues in diagonal = variance in direction of eigenvector
  2. Project each input vector x into this subspace, e.g.,

$$z_j = \mathbf{u}_j^T\mathbf{x}; \qquad \mathbf{z} = U_{1:M}^T\mathbf{x}$$

Olivetti Faces Dataset - PCA

- First six principal components (eigen faces) u_0, … u_5
- u_j is column j of matrix U



Bayes Theorem

- The **posterior** probability of θ, given our observation (x) is proportional to the **likelihood** times the **prior** probability of θ.

$$P(\theta \mid x) = \frac{P(x \mid \theta) P(\theta)}{P(x)}$$

Bayesian Modelling

$$P(\theta|\mathcal{D}, m) = \frac{P(\mathcal{D}|\theta, m)P(\theta|m)}{P(\mathcal{D}|m)}$$

$P(\mathcal{D}\|\theta, m)$	likelihood of parameters θ in model m
$P(\theta\|m)$	prior probability of θ
$P(\theta\|\mathcal{D}, m)$	posterior of θ given data $\mathcal{D}$

Prediction:

$$P(x|\mathcal{D}, m) = \int P(x|\theta, \mathcal{D}, m)P(\theta|\mathcal{D}, m)d\theta$$

Model Comparison:

$$P(m|\mathcal{D}) = \frac{P(\mathcal{D}|m)P(m)}{P(\mathcal{D})}$$

$$P(\mathcal{D}|m) = \int P(\mathcal{D}|\theta, m)P(\theta|m)\, d\theta$$

Conjugacy

If the posterior distributions $p(\theta|x)$ are in the same family as the prior probability distribution $p(\theta)$, the prior and posterior are then called **conjugate distributions**, and the prior is called a **conjugate prior** for the likelihood.

Binomial Data, Beta Prior

Suppose the prior distribution for θ is Beta(α1,α2) and the conditional distribution of X given θ is Bin(n, p). Then

$$P(x|\theta) = \binom{n}{x}\theta^x(1-\theta)^{(n-x)}$$

$$P(\theta) = \frac{\Gamma(\alpha_1+\alpha_2)}{\Gamma(\alpha_1)+\Gamma(\alpha_2)}\theta^{\alpha_1-1}(1-\theta)^{(\alpha_2-1)}$$

Binomial Data, Beta Prior (cont.)

We now calculate the posterior:

$$\text{posterior} \propto \text{likelihood} \times \text{prior}.$$

$$P(\theta|x) \propto P(x|\theta)P(\theta)$$

$$= \binom{n}{x} \frac{\Gamma(\alpha_1 + \alpha_2)}{\Gamma(\alpha_1) + \Gamma(\alpha_2)} \theta^{x+\alpha_1-1}(1-\theta)^{(n-x+\alpha_2-1)}$$

Binomial Data, Beta Prior (cont.)

Given x, $\binom{n}{x} \frac{\Gamma(\alpha_1 + \alpha_2)}{\Gamma(\alpha_1) + \Gamma(\alpha_2)}$ is a constant. Therefore,

$$P(\theta|x) \propto \theta^{x+\alpha_1-1}(1-\theta)^{(n-x+\alpha_2-1)}$$

We now recognize it as another Beta distribution with parameter (x+α1) and (n-x+α2): Beta(x+α1,n-x+α2).

Same family as the prior distribution: conjugate prior!

Conjugate Priors

- Bernoulli, model parameter p: Beta distribution with α, β
- Binomial, model parameter p: Beta distribution with α, β
- Bernoulli, model parameter λ: Gamma distribution with k, θ
- Gaussian, variance known, model parameter μ: Gaussian distribution with μ_0, σ_0^2
- Gaussian, precision known, model parameter μ: Gaussian distribution with μ_0, τ_0^2
- Gaussian, mean known, model parameter σ^2: Inverse gamma distribution with α, β
- MV Gaussian, covariance known, model parameter μ: MV Gaussian distribution with $\mu_{\mathbf{0}}, \Sigma_0$
- MV Gaussian, mean known, model parameter $\mathbf{\Sigma}$: Inverse-Wishart distribution with ν, Ψ

Prior, posterior and likelihood functions

- When designing machine learning models with the parameters θ , we assume uncertainties in the parameters:
  - Prior: $P(\theta)$ (the model of engineering knowledge)
  - Likelihood: $P(\text{data} \mid \theta)$ (the model of data)
- Learning is about finding a “good” set of parameters under our modelling assumption
  - One common approach is Maximum-likelihood estimation (MLE)

$$\theta^* = \underset{\theta}{\text{argmin}} \left[-\log P(\text{data} \mid \theta) \right]$$

Prior, posterior and likelihood functions

- When designing machine learning models with the parameters θ , we assume uncertainties in the parameters:
  - Prior: $P(\theta)$ (the model of engineering knowledge)
  - Likelihood: $P(\text{data} \mid \theta)$ (the model of data)
- Learning is about finding a “good” set of parameters under our modelling assumption
  - One common approach is Maximum-likelihood estimation (MLE)

$$\theta^* = \underset{\theta}{\text{argmax}} \log P(\text{data} \mid \theta)$$

Prior, posterior and likelihood functions

- When designing machine learning models with the parameters θ , we assume uncertainties in the parameters:
  - Prior: $P(\theta)$ (the model of engineering knowledge)
  - Likelihood: $P(\text{data} \mid \theta)$ (the model of data)
- Learning is about finding a “good” set of parameters under our modelling assumption
  - Another approach is Maximum a posteriori estimation (MAP)

$$P(\theta \mid \text{data}) = \frac{P(\text{data} \mid \theta) P(\theta)}{P(\text{data})}$$
$$\propto P(\text{data} \mid \theta) P(\theta)$$

Prior, posterior and likelihood functions

- When designing machine learning models with the parameters θ , we assume uncertainties in the parameters:
  - Prior: $P(\theta)$ (the model of engineering knowledge)
  - Likelihood: $P(\text{data} \mid \theta)$ (the model of data)
- Learning is about finding a “good” set of parameters under our modelling assumption
  - Another approach is Maximum a posteriori estimation (MAP)

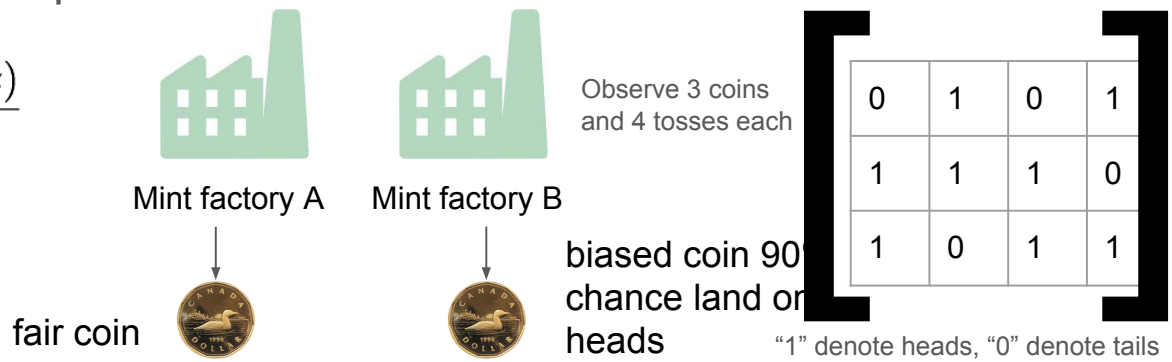
$$\begin{aligned}\theta^* &= \underset{\theta}{\text{argmax}} \log P(\theta \mid \text{data}) \\ &= \underset{\theta}{\text{argmax}} \left[\log P(\text{data} \mid \theta) + \log P(\theta)\right]\end{aligned}$$

Prior, posterior and likelihood functions

- Build more complicated probabilistic models by introducing latent random variables z
  - Prior: $P(z)$ (the model of the world)
  - Likelihood: $P(\text{data} \mid z)$(the model of data)
- Inference is to compute the posterior distributions of the latent RVs

$$P(z \mid \text{data}) = \frac{P(\text{data} \mid z)P(z)}{P(\text{data})}$$

Examples:

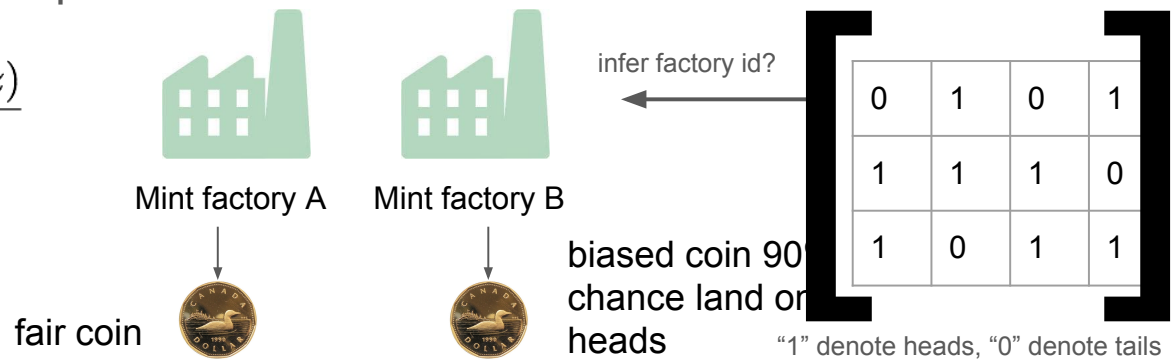


Prior, posterior and likelihood functions

- Build more complicated probabilistic models by introducing latent random variables z
  - Prior: $P(z)$ (the model of the world)
  - Likelihood: $P(\text{data} \mid z)$(the model of data)
- Inference is to compute the posterior distributions of the latent RVs

$$P(z \mid \text{data}) = \frac{P(\text{data} \mid z)P(z)}{P(\text{data})}$$

Examples:



Prior, posterior and likelihood functions

- Mixtures of Bernoullis model
  - z = {1, 2}, denote the factory identity
  - Prior: $P(z = 1) = P(z = 2) = 0.5$
  - Likelihood: $P(x = \text{heads} \mid z = 1) = 0.5$
    $P(x = \text{heads} \mid z = 2) = 0.9$

Mint factory A

Mint factory B

fair coin

biased coin 90% chance land on heads

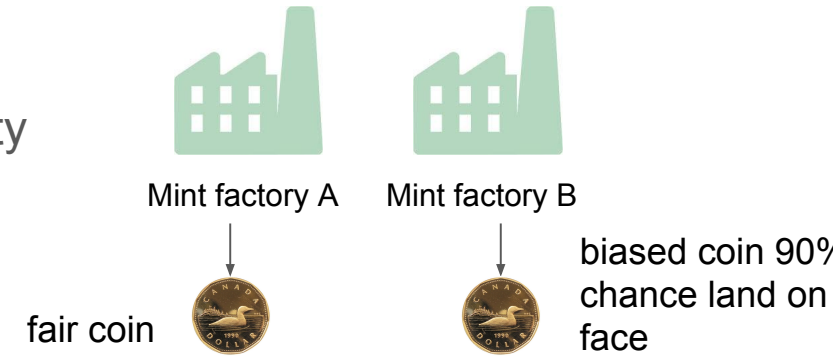
Posterior: $P(z \mid \text{data}) = \dfrac{P(\text{data} \mid z)P(z)}{P(\text{data})}$

"1" denote heads, "0" denote tails

0	1	0	1
1	1	1	0
1	0	1	1

Prior, posterior and likelihood functions

- Mixtures of Bernoullis model
  - z = {1, 2}, denote the factory identity
  - Prior: $P(z=1) = P(z=2) = 0.5$
  - Likelihood: $P(x = \text{heads} \,|\, z=1) = 0.5$
    $P(x = \text{heads} \,|\, z=2) = 0.9$

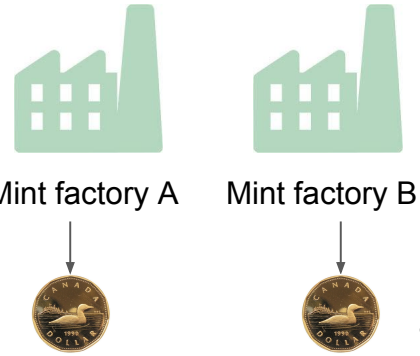


Posterior:
$$P(z=1 \,|\, x_1 = [0,1,0,1]) = \frac{P(x_1 = [0,1,0,1] \,|\, z=1)P(z=1)}{P(x_1 = [0,1,0,1])}$$
$$\propto P(x=0 \,|\, z=1)^2 P(x=1 \,|\, z=1)^2 P(z=1)$$
$$= 0.5^4 * 0.5 = 0.0315$$

0	1	0	1
1	1	1	0
1	0	1	1

Prior, posterior and likelihood functions

- Mixtures of Bernoullis model
  - z = {1, 2}, denote the factory identity
  - Prior: $P(z = 1) = P(z = 2) = 0.5$
  - Likelihood: $P(x = \text{heads} \,|\, z = 1) = 0.5$
    $P(x = \text{heads} \,|\, z = 2) = 0.9$



$$P(z = 1 \,|\, x_1 = [0, 1, 0, 1]) \propto 0.0315$$

Posterior:

$$P(z = 2 \,|\, x_1 = [0, 1, 0, 1]) = \frac{P(x_1 = [0, 1, 0, 1] \,|\, z = 2) P(z = 2)}{P(x_1 = [0, 1, 0, 1])}$$
$$\propto P(x = 0 \,|\, z = 2)^2 P(x = 1 \,|\, z = 2)^2 P(z = 2)$$
$$= 0.9^2 * 0.1^2 * 0.5 = 0.00405$$

0	1	0	1
1	1	1	0
1	0	1	1

Prior, posterior and likelihood functions

- Mixtures of Bernoullis model
  - z = {1, 2}, denote the factory identity
  - Prior: $P(z = 1) = P(z = 2) = 0.5$
  - Likelihood: $P(x = \text{heads} \,|\, z = 1) = 0.5$
    $P(x = \text{heads} \,|\, z = 2) = 0.9$

Mint factory A Mint factory B

fair coin

biased coin 90% chance land on face

Posterior:

$$P(z = 1 \,|\, x_1 = [0, 1, 0, 1]) \propto 0.0315$$

$$P(z = 2 \,|\, x_1 = [0, 1, 0, 1]) \propto 0.00405$$

0	1	0	1
1	1	1	0
1	0	1	1

Prior, posterior and likelihood functions

- Mixtures of Bernoullis model
  - z = {1, 2}, denote the factory identity
  - Prior: $P(z = 1) = P(z = 2) = 0.5$
  - Likelihood: $P(x = \text{heads} \,|\, z = 1) = 0.5$
    $P(x = \text{heads} \,|\, z = 2) = 0.9$

Mint factory A

Mint factory B

fair coin

biased coin 90% chance land on face

Posterior:

$$P(z = 1 \,|\, x_1 = [0, 1, 0, 1]) \propto 0.0315$$

$$P(z = 2 \,|\, x_1 = [0, 1, 0, 1]) \propto 0.00405$$

$$P(z = 1 \,|\, x_1 = [0, 1, 0, 1]) = \frac{0.0315}{0.0315 + 0.00405} = 0.9$$

0	1	0	1
1	1	1	0
1	0	1	1

Expectation maximization

- Consider the Maximum-likelihood Estimation (MLE) approach to learn model parameters $\theta = \{\theta_{\text{prior}}, \theta_{\text{likelihood}}\}$:
  - Prior: $P(z \mid \theta_{\text{prior}})$ (the model of the world)
  - Likelihood: $P(x \mid z, \theta_{\text{likelihood}})$ (the model of data)

Expectation maximization

- Consider the Maximum-likelihood Estimation (MLE) approach to learn model parameters $\theta = \{\theta_{\text{prior}}, \theta_{\text{likelihood}}\}$:
  - Prior: $P(z \mid \theta_{\text{prior}})$ (the model of the world), e.g. mint example
    $$P(z = 1) = \theta_{\text{prior}}$$
    $$P(z = 2) = 1 - \theta_{\text{prior}}$$
  - Likelihood: $P(x \mid z, \theta_{\text{likelihood}})$ (the model of data)
    $$P(x = \text{heads} \mid z = 1) = \theta_{\text{likelihood}1}$$
    $$P(x = \text{heads} \mid z = 2) = \theta_{\text{likelihood}2}$$

Expectation maximization

- Consider the Maximum-likelihood Estimation (MLE) approach to learn model parameters $\theta = \{\theta_{\text{prior}}, \theta_{\text{likelihood}}\}$:
  - Prior: $P(z \mid \theta_{\text{prior}})$ (the model of the world)
  - Likelihood: $P(x \mid z, \theta_{\text{likelihood}})$ (the model of data)
- Define data likelihood or marginal likelihood as:

$$P(x \mid \theta) = \sum_{z} P(z \mid \theta_{\text{prior}}) P(x \mid z, \theta_{\text{likelihood}})$$

Expectation maximization

- Consider the Maximum-likelihood Estimation (MLE) approach to learn model parameters $\theta = \{\theta_{\text{prior}}, \theta_{\text{likelihood}}\}$:
  - Prior: $P(z \mid \theta_{\text{prior}})$ (the model of the world)
  - Likelihood: $P(x \mid z, \theta_{\text{likelihood}})$ (the model of data)
- Define data likelihood or marginal likelihood as:

$$P(x \mid \theta) = \sum_z P(z \mid \theta_{\text{prior}}) P(x \mid z, \theta_{\text{likelihood}})$$

- MLE of the model parameter:

$$\theta^* = \underset{\theta}{\text{argmax}} \log P(x \mid \theta) = \underset{\theta}{\text{argmax}} \log \sum_z P(z \mid \theta) P(x \mid z, \theta)$$

Learning: Expectation maximization

- Main idea: maximize a “tight” lower bound will also improve the marginal log likelihood

$$\log P(x \mid \theta) = \log \sum_z P(z \mid \theta) P(x \mid z, \theta)$$

Learning: Expectation maximization

- Main idea: maximize a “tight” lower bound will also improve the marginal log likelihood

$$\log P(x \mid \theta) = \log \sum_z P(z \mid \theta) P(x \mid z, \theta)$$

$$\log \sum_z P(z \mid \theta) P(x \mid z, \theta) = \log \sum_z Q(z) P(z \mid \theta) P(x \mid z, \theta) / Q(z)$$

Learning: Expectation maximization

- Main idea: maximize a “tight” lower bound will also improve the marginal log likelihood

$$\log P(x \mid \theta) = \log \sum_z P(z \mid \theta) P(x \mid z, \theta)$$

$$\log \sum_z P(z \mid \theta) P(x \mid z, \theta) = \log \sum_z Q(z) P(z \mid \theta) P(x \mid z, \theta) / Q(z)$$

$$= \log \mathbb{E}_{Q(z)} \left[P(z \mid \theta) P(x \mid z, \theta) / Q(z) \right]$$

Learning: Expectation maximization

- Main idea: maximize a “tight” lower bound will also improve the marginal log likelihood

$$\log P(x \mid \theta) = \log \sum_z P(z \mid \theta) P(x \mid z, \theta)$$

$$\log \sum_z P(z \mid \theta) P(x \mid z, \theta) = \log \sum_z Q(z) P(z \mid \theta) P(x \mid z, \theta) / Q(z)$$

$$= \log \mathbb{E}_{Q(z)} \left[P(z \mid \theta) P(x \mid z, \theta) / Q(z) \right]$$

$$\geq \mathbb{E}_{Q(z)} \left[\log \frac{P(z \mid \theta) P(x \mid z, \theta)}{Q(z)} \right]$$

Jensen’s Inequality

$\mathbb{E}[f(X)] \geq f(\mathbb{E}[X])$

f() is log that is concave

Learning: Expectation maximization

- Main idea: maximize a “tight” lower bound will also improve the marginal log likelihood

$$\log P(x \mid \theta) = \log \sum_z P(z \mid \theta) P(x \mid z, \theta)$$

$$\begin{aligned}
\log \sum_z P(z \mid \theta) P(x \mid z, \theta) &= \log \sum_z Q(z) P(z \mid \theta) P(x \mid z, \theta) / Q(z) \\
&= \log \mathbb{E}_{Q(z)} \left[P(z \mid \theta) P(x \mid z, \theta) / Q(z) \right] \\
&\geq \mathbb{E}_{Q(z)} \left[\log \frac{P(z \mid \theta) P(x \mid z, \theta)}{Q(z)} \right] \\
&= \sum_z Q(z) \log \frac{P(z \mid \theta) P(x \mid z, \theta)}{Q(z)}
\end{aligned}$$

Learning: Expectation maximization

- Main idea: maximize a “tight” lower bound will also improve the marginal log likelihood

$$\log \sum_z P(z \mid \theta) P(x \mid z, \theta) = \log \sum_z Q(z) P(z \mid \theta) P(x \mid z, \theta) / Q(z)$$

lower bound:

$$\geq \sum_z Q(z) \log \frac{P(z \mid \theta) P(x \mid z, \theta)}{Q(z)}$$

- First, we ensure the lower bound is tight to the marginal log likelihood
  - Find the Q distribution for which the equality holds in the Jensen’s Inequality:

tighten the lower bound:

$$Q(z) \propto P(z \mid \theta) P(x \mid z, \theta)$$

$$\text{i.e.} Q(z) = P(z \mid x, \theta)$$

- Second, optimize the parameters in the lower bound

optimize the lower bound:

$$\theta^* = \underset{\theta}{\operatorname{argmax}} \sum_z P(z \mid x) \log P(z \mid \theta) P(x \mid z, \theta)$$

Learning: Expectation maximization

- Main idea: maximize a "tight" lower bound will also improve the marginal log likelihood

$$\log \sum_z P(z \mid \theta) P(x \mid z, \theta) = \log \sum_z Q(z) P(z \mid \theta) P(x \mid z, \theta) / Q(z)$$

lower bound:

$$\geq \sum_z Q(z) \log \frac{P(z \mid \theta) P(x \mid z, \theta)}{Q(z)}$$

- First, we ensure the lower bound is tight to the marginal log likelihood
  - Find the Q distribution for which the equality holds in the Jensen's Inequality:

tighten the lower bound:

$$Q(z) \propto P(z \mid \theta) P(x \mid z, \theta)$$

$$\text{i.e.} Q(z) = P(z \mid x, \theta)$$

- Second, optimize the parameters in the lower bound

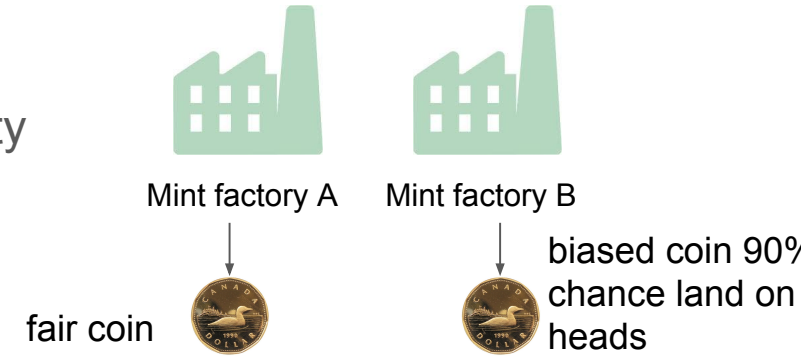
optimize the lower bound:

$$\theta^* = \underset{\theta}{\operatorname{argmax}} \sum_z P(z \mid x) \log P(z \mid \theta) P(x \mid z, \theta)$$

repeat till convergence

Learning: Expectation maximization

- Mixtures of Bernoullis model
  - z = {1, 2}, denote the factory identity
  - Prior: $P(z=1)=\theta_z, P(z=2)=1-\theta_z$
  - Likelihood: $P(x=\text{heads}\,|\,z=1)=\theta_1$
    $P(x=\text{heads}\,|\,z=2)=\theta_2$



Learn new parameters:

$$\theta_z^*, \theta_1^*, \theta_2^* = \underset{\theta_z,\theta_1,\theta_2}{\text{argmax}} \sum_n \sum_z P(z_n \,|\, x_n) \log P(z_n \,|\, \theta_z) P(x_n \,|\, z_n, \theta_1, \theta_2)$$

$$= \underset{\theta_z,\theta_1,\theta_2}{\text{argmax}} \sum_n \sum_z P(z_n \,|\, x_n) [\log P(z_n \,|\, \theta_z) + \log P(x_n \,|\, z_n, \theta_1, \theta_2)]$$

3 unknown coins, each tossed 4 times

“1” denote heads, “0” denote tail

x_1	0	1	0	1
x_2	1	1	1	0
x_3	1	0	1	1

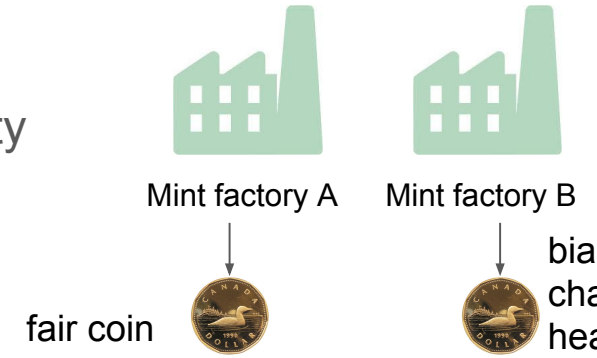
Learning: Expectation maximization

- Mixtures of Bernoullis model
  - z = {1, 2}, denote the factory identity
  - Prior: $P(z=1)=\theta_z, P(z=2)=1-\theta_z$
  - Likelihood: $P(x=\text{heads}\,|\,z=1)=\theta_1$
    $P(x=\text{heads}\,|\,z=2)=\theta_2$

$$\theta_z^*,\theta_1^*,\theta_2^* = \underset{\theta_z,\theta_1,\theta_2}{\operatorname{argmax}} \sum_n \sum_z P(z_n\,|\,x_n)\log P(z_n\,|\,\theta_z)P(x_n\,|\,z_n,\theta_1,\theta_2)$$

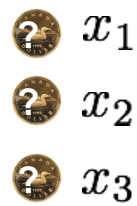
Denote:

$$\mathcal{F} = \sum_n \sum_z P(z_n\,|\,x_n)[\log P(z_n\,|\,\theta_z) + \log P(x_n\,|\,z_n,\theta_1,\theta_2)]$$



"1" denote heads, "0" denote tail

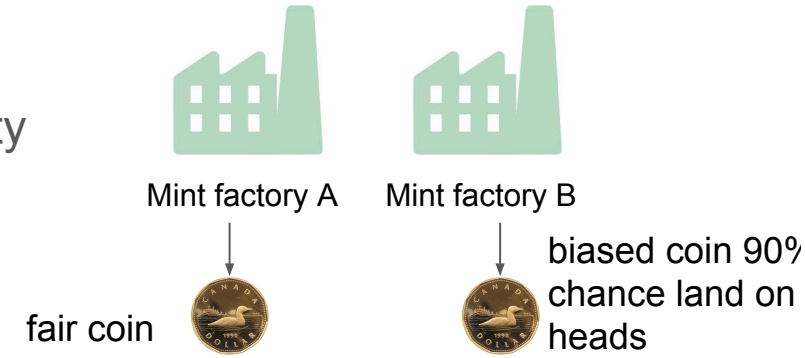
3 unknown coins, each tossed 4 times



0	1	0	1
1	1	1	0
1	0	1	1

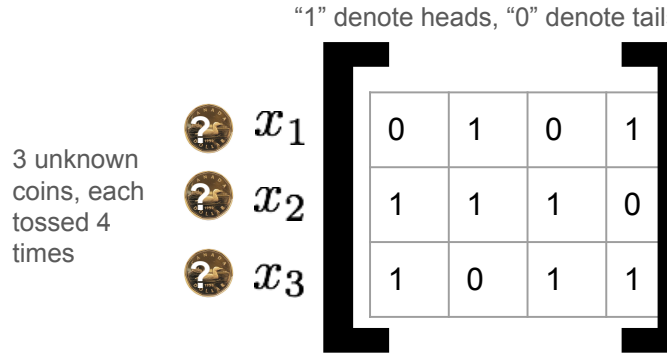
Learning: Expectation maximization

- Mixtures of Bernoullis model
  - z = {1, 2}, denote the factory identity
  - Prior: $P(z=1)=\theta_z, P(z=2)=1-\theta_z$
  - Likelihood: $P(x=\text{heads}\,|\,z=1)=\theta_1$
    $P(x=\text{heads}\,|\,z=2)=\theta_2$



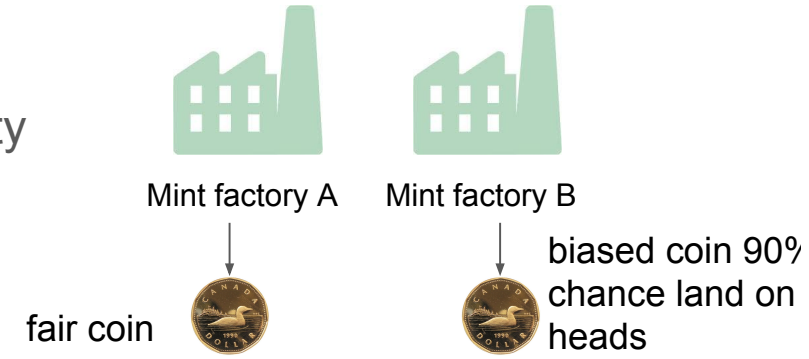
$$\mathcal{F}=\sum_n\sum_z P(z_n\,|\,x_n)[\log P(z_n\,|\,\theta_z)+\log P(x_n\,|\,z_n,\theta_1,\theta_2)]$$

Rewrite: $P(z_n|\theta_z)=\theta_z^{\{z=1\}}(1-\theta_z)^{\{z=2\}}$



Learning: Expectation maximization

- Mixtures of Bernoullis model
  - z = {1, 2}, denote the factory identity
  - Prior: $P(z=1)=\theta_z, P(z=2)=1-\theta_z$
  - Likelihood: $P(x=\text{heads}\,|\,z=1)=\theta_1$
    $P(x=\text{heads}\,|\,z=2)=\theta_2$



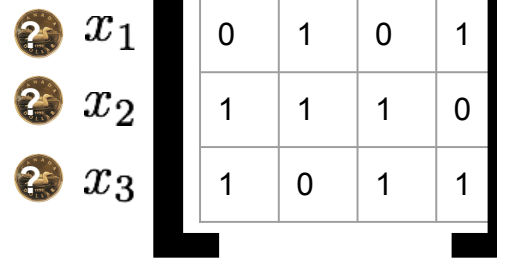
$$\mathcal{F}=\sum_n\sum_z P(z_n\,|\,x_n)[\log P(z_n\,|\,\theta_z)+\log P(x_n\,|\,z_n,\theta_1,\theta_2)]$$

Rewrite: $P(z_n|\theta_z)=\theta_z^{\{z=1\}}(1-\theta_z)^{\{z=2\}}$

$$\log P(z_n|\theta_z)=\{z=1\}\log\theta_z+\{z=2\}\log(1-\theta_z)$$

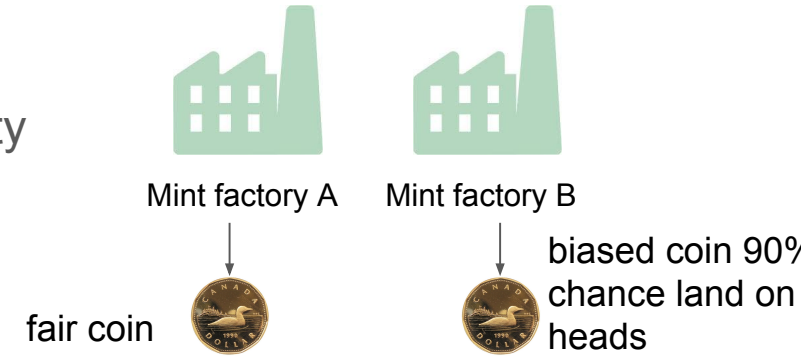
"1" denote heads, "0" denote tail

3 unknown coins, each tossed 4 times



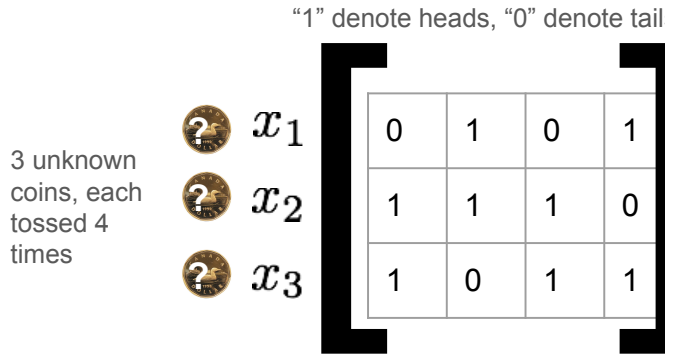
Learning: Expectation maximization

- Mixtures of Bernoullis model
  - z = {1, 2}, denote the factory identity
  - Prior: $P(z=1)=\theta_z, P(z=2)=1-\theta_z$
  - Likelihood: $P(x=\text{heads}\,|\,z=1)=\theta_1$
    $P(x=\text{heads}\,|\,z=2)=\theta_2$



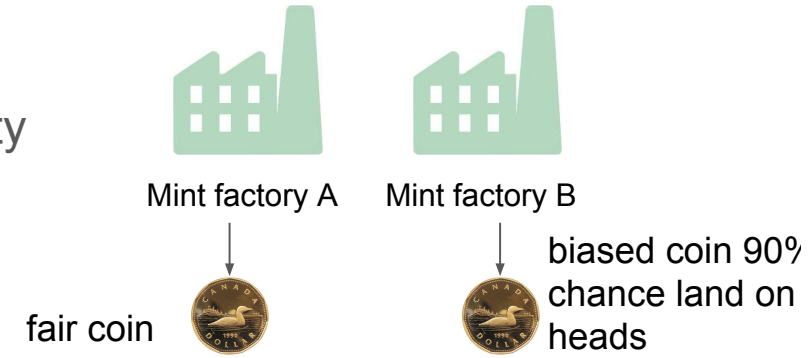
$$\mathcal{F}=\sum_n\sum_z P(z_n\,|\,x_n)[\log P(z_n\,|\,\theta_z)+\log P(x_n\,|\,z_n,\theta_1,\theta_2)]$$

Rewrite: $$\frac{\partial\mathcal{F}}{\partial\theta_z}=\sum_n\sum_z P(z_n\,|\,x_n)\left[\frac{\partial}{\partial\theta_z}\log P(z_n\,|\,\theta_z)\right]$$



Learning: Expectation maximization

- Mixtures of Bernoullis model
  - z = {1, 2}, denote the factory identity
  - Prior: $P(z=1)=\theta_z, P(z=2)=1-\theta_z$
  - Likelihood: $P(x=\text{heads}\,|\,z=1)=\theta_1$
    $P(x=\text{heads}\,|\,z=2)=\theta_2$



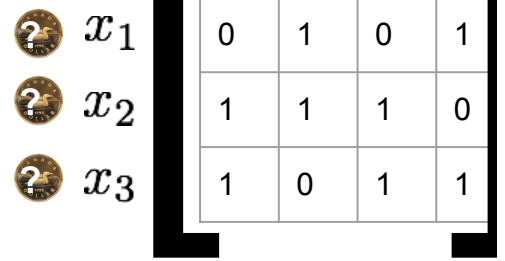
$$\mathcal{F}=\sum_n\sum_z P(z_n\,|\,x_n)[\log P(z_n\,|\,\theta_z)+\log P(x_n\,|\,z_n,\theta_1,\theta_2)]$$

Rewrite: $$\frac{\partial\mathcal{F}}{\partial\theta_z}=\sum_n\sum_z P(z_n\,|\,x_n)\left[\frac{\partial}{\partial\theta_z}\log P(z_n\,|\,\theta_z)\right]$$

$$=0$$ set derivative to zero to solve for optimals

"1" denote heads, "0" denote tail

3 unknown coins, each tossed 4 times



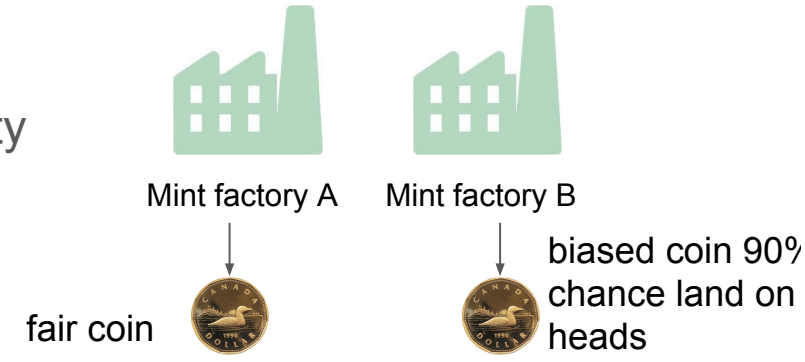
Learning: Expectation maximization

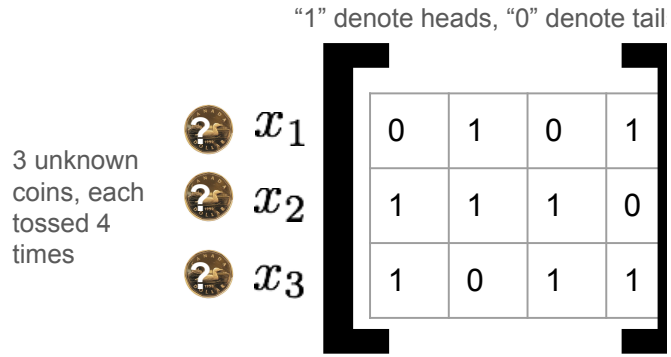
- Mixtures of Bernoullis model
  - z = {1, 2}, denote the factory identity
  - Prior: $P(z=1)=\theta_z, P(z=2)=1-\theta_z$
  - Likelihood: $P(x=\text{heads}\,|\,z=1)=\theta_1$
    $P(x=\text{heads}\,|\,z=2)=\theta_2$

$$\frac{\partial \mathcal{F}}{\partial \theta_z} = \sum_n \sum_z P(z_n \,|\, x_n) \left[\frac{\partial}{\partial \theta_z} \log P(z_n \,|\, \theta_z) \right]$$

$$= 0$$

substitute $\log P(z_n|\theta_z) = \{z=1\} \log \theta_z + \{z=2\} \log(1-\theta_z)$

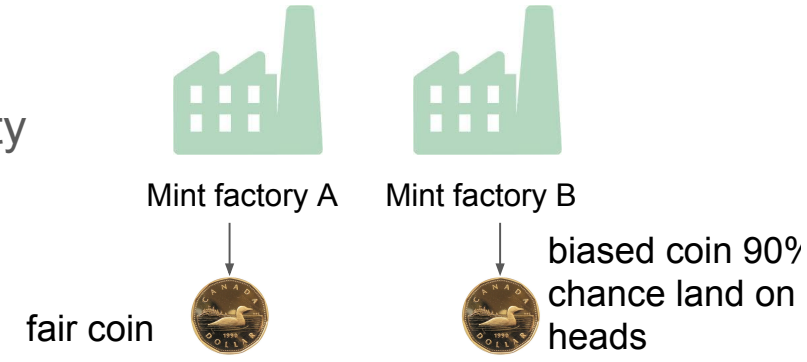


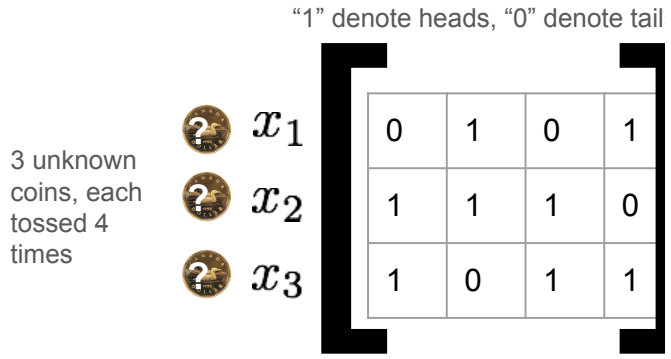


Learning: Expectation maximization

- Mixtures of Bernoullis model
  - z = {1, 2}, denote the factory identity
  - Prior: $P(z=1)=\theta_z, P(z=2)=1-\theta_z$
  - Likelihood: $P(x=\text{heads}\,|\,z=1)=\theta_1$
    $P(x=\text{heads}\,|\,z=2)=\theta_2$

$$\theta_z = \frac{\sum_n P(z_n=1|x_n)}{\sum_n P(z_n=1|x_n)+\sum_n P(z_n=2|x_n)}$$

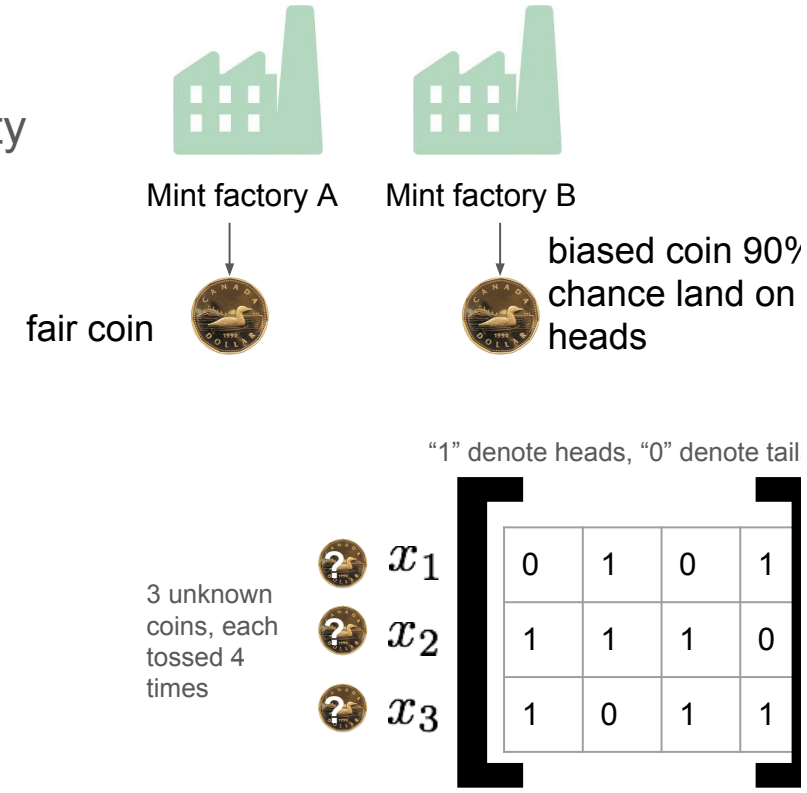




Learning: Expectation maximization

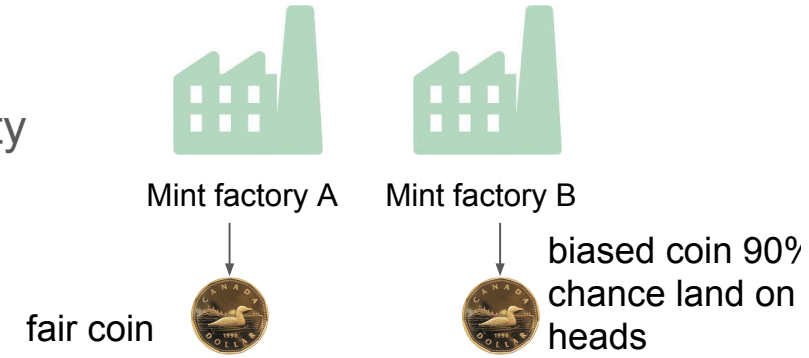
- Mixtures of Bernoullis model
  - z = {1, 2}, denote the factory identity
  - Prior: $P(z=1)=\theta_z, P(z=2)=1-\theta_z$
  - Likelihood: $P(x=\text{heads}\,|\,z=1)=\theta_1$
    $P(x=\text{heads}\,|\,z=2)=\theta_2$

$$\theta_z = \frac{\sum_n P(z_n=1|x_n)}{\sum_n P(z_n=1|x_n)+\sum_n P(z_n=2|x_n)} = \frac{\sum_n P(z_n=1|x_n)}{N}$$



Learning: Expectation maximization

- Mixtures of Bernoullis model
  - z = {1, 2}, denote the factory identity
  - Prior: $P(z=1)=\theta_z, P(z=2)=1-\theta_z$
  - Likelihood: $P(x=\text{heads}\,|\,z=1)=\theta_1$
    $P(x=\text{heads}\,|\,z=2)=\theta_2$



$$\theta_z = \frac{\sum_n P(z_n=1|x_n)}{\sum_n P(z_n=1|x_n)+\sum_n P(z_n=2|x_n)} = \frac{\sum_n P(z_n=1|x_n)}{N}$$

$$P(z_1=1\,|\,x_2=[0,1,0,1]) = \frac{0.0315}{0.0315+0.00405} = 0.9$$

$$P(z_2=1\,|\,x_2=[1,1,1,0]) = \frac{0.0315}{0.0315+0.03645} = 0.459$$

$$P(z_3=1\,|\,x_2=[1,0,1,1]) = \frac{0.0315}{0.0315+0.03645} = 0.459$$

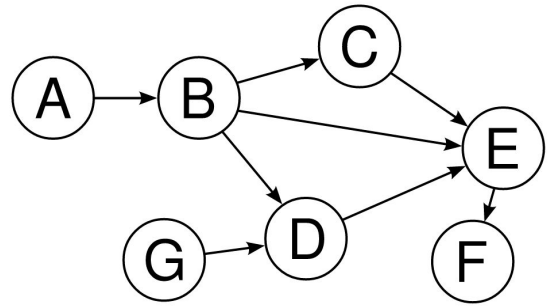
"1" denote heads, "0" denote tail

3 unknown coins, each tossed 4 times

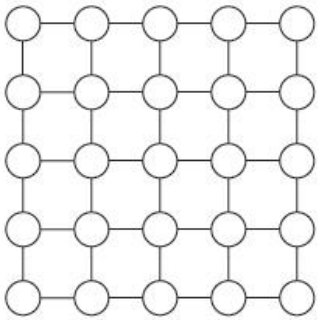
0	1	0	1
1	1	1	0
1	0	1	1

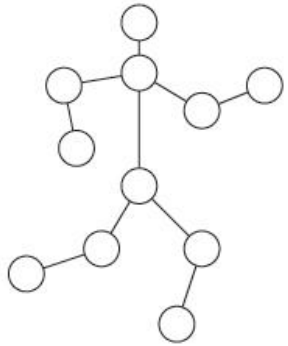
Graphical models

- Bayesian networks (i.e. BN, BayesNet), directed-acyclic-graph (DAG)

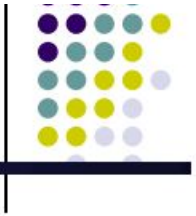


- Markov random fields, undirected graph





Bayesian Network:



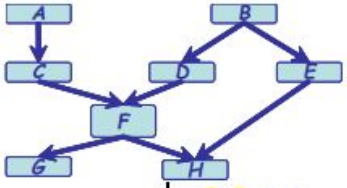
- A BN is a directed graph whose nodes represent the random variables and whose edges represent direct influence of one variable on another.

- It is a data structure that provides the skeleton for representing **a joint distribution** compactly in a **factorized** way;

- It offers a compact representation for **a set of conditional independence assumptions** about a distribution;

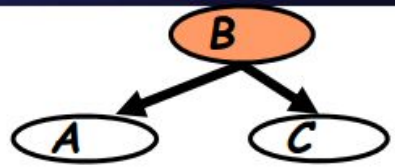
- We can view the graph as encoding a generative sampling process executed by nature, where the value for each variable is selected by nature using a distribution that depends only on its parents. In other words, each variable is a stochastic function of its parents.

Local Structures & Independencies



- Common parent
  - Fixing B decouples A and C

    "given the level of gene B, the levels of A and C are independent"



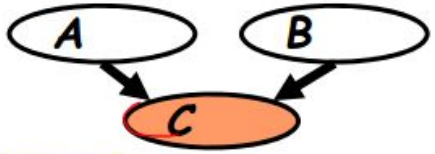
- Cascade
  - Knowing B decouples A and C

    "given the level of gene B, the level gene A provides no extra prediction value for the level of gene C"

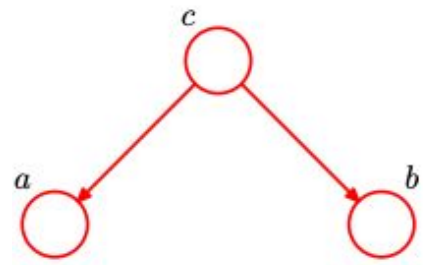
- V-structure
  - Knowing C couples A and B

    because A can "explain away" B w.r.t. C

    "If A correlates to C, then chance for B to also correlate to B will decrease"



Common parent



According to the graphical model, we can decompose the joint probability over the 3 variables as:

$$p(a, b, c) = p(a|c)p(b|c)p(c)$$

In general, we have: $p(a, b) = \sum_c p(a|c)p(b|c)p(c)$

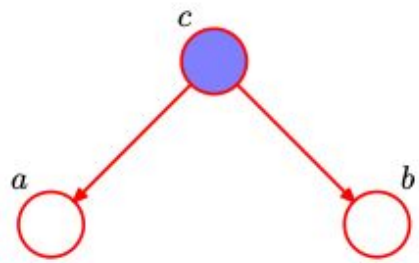
This does not in general decompose into: $p(a, b) = p(a)p(b)$

So a and b are not independent.

Common parent

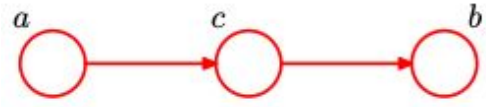
… But if we observe c:

$$p(a, b|c) = \frac{p(a, b, c)}{p(c)} = p(a|c)p(b|c)$$



So a and b are **conditionally** independent given c

Cascade



According to the graphical model we can decompose the joint as:

$$p(a, b, c) = p(a)p(c|a)p(b|c)$$

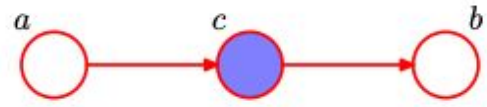
In general, we have:

$$p(a, b) = p(a) \sum_c p(c|a)p(b|c) = p(a)p(b|a)$$

Which does not in general factorize as: $p(a, b) = p(a)p(b)$

So a and b are not independent

Cascade

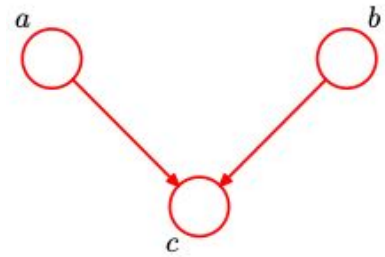


But if we condition on c...

$$p(a, b|c) = \frac{p(a, b, c)}{p(c)} = \frac{p(a)p(c|a)p(b|c)}{p(c)} = p(a|c)p(b|c)$$

a and b are **conditionally independent** given c

V-structure



According to the graphical model we can decompose the joint as:

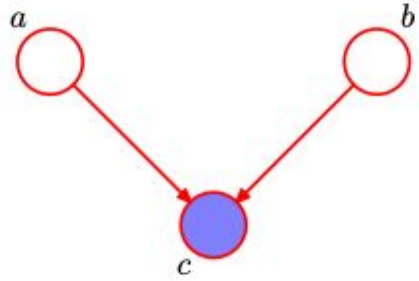
$$p(a, b, c) = p(a)p(b)p(c|a, b)$$

In general, we have:

$$p(a, b) = \sum_c p(a, b, c) = \sum_c p(a)p(b)p(c|a, b) = p(a)p(b)$$

So a and b are independent!

V-structure



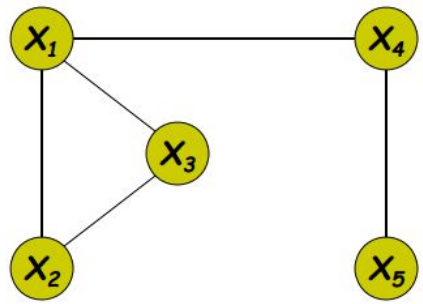
… but if we condition on c:

$$p(a,b|c) = \frac{p(a,b,c)}{p(c)} = \frac{p(a)p(b)p(c|a,b)}{p(c)}$$

which does does not in general factorize into $p(a)p(b)$

Therefore a and b are not conditionally independent given c.

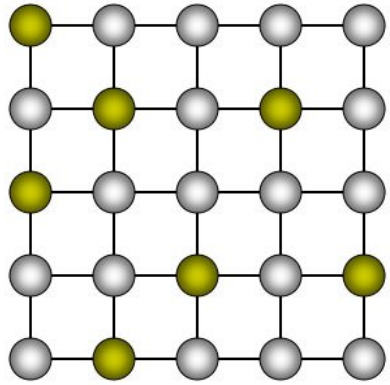
Undirected graphical models (UGM)



- Pairwise (non-causal) relationships
- Can write down model, and score specific configurations of the graph, but no explicit way to generate samples
- Contingency constrains on node configurations

Canonical example

- The grid model



- Naturally arises in image processing, lattice physics, etc.
- Each node may represent a single "pixel", or an atom
  - The states of adjacent or nearby nodes are "coupled" due to pattern continuity or electro-magnetic force, etc.
  - Most likely joint-configurations usually correspond to a "low-energy" state

Representation

- Defn: an undirected graphical model represents a distribution $P(X_1,\ldots,X_n)$ defined by an undirected graph H, and a set of positive ***potential functions*** y_c associated with the cliques of H, s.t.

$$P(x_1,\ldots,x_n)=\frac{1}{Z}\prod_{c\in C}\psi_c(\mathbf{x}_c)$$

where Z is known as the partition function:

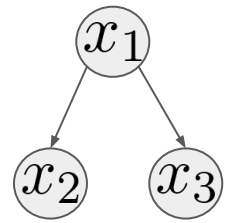
$$Z=\sum_{x_1,\ldots,x_n}\prod_{c\in C}\psi_c(\mathbf{x}_c)$$

- Also known as Markov Random Fields, Markov networks …
- The ***potential function*** can be understood as an contingency function of its arguments assigning "pre-probabilistic" score of their joint configuration.

Factor Graphs

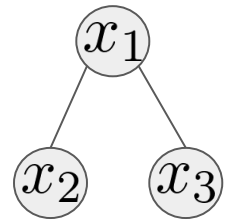
- Both directed and undirected graphical models express a joint probability distribution in a factorized way. For example:
- Directed:

$$p(x_1, x_2, x_3) = p(x_1)p(x_2|x_1)p(x_3|x_1)$$



- Undirected:

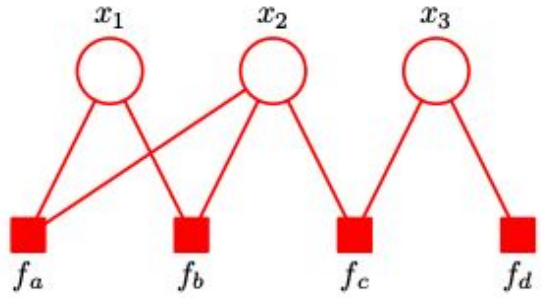
$$p(x_1, x_2, x_3) = \frac{1}{Z}\psi(x_1, x_2)\psi(x_1, x_3)$$



Factor Graphs

Let us write the joint distribution over a set of variables in the form of a product of factors (with x_s denoting a subset of variables):

$$p(x) = \prod_s f_s(x_s)$$



Factor graphs have nodes for variables as before (circles) and also for factors (squares). This can be used to represent either a directed or undirected PGM.

Example factor graphs for directed GM

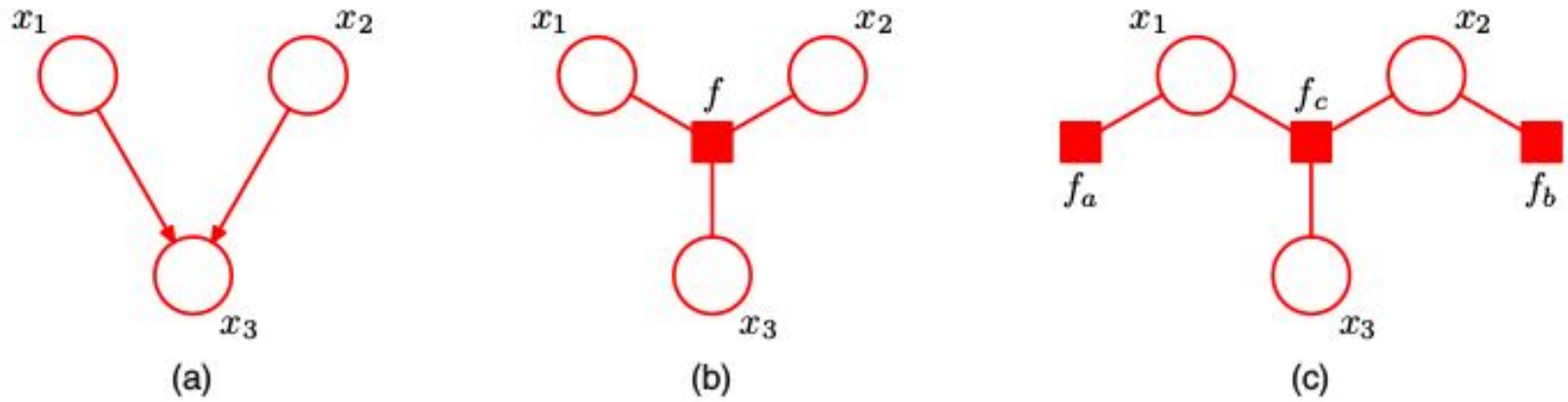


Figure 8.42 (a) A directed graph with the factorization $p(x_1)p(x_2)p(x_3|x_1, x_2)$. (b) A factor graph representing the same distribution as the directed graph, whose factor satisfies $f(x_1, x_2, x_3) = p(x_1)p(x_2)p(x_3|x_1, x_2)$. (c) A different factor graph representing the same distribution with factors $f_a(x_1) = p(x_1)$, $f_b(x_2) = p(x_2)$ and $f_c(x_1, x_2, x_3) = p(x_3|x_1, x_2)$.

Example factor graphs for undirected GM

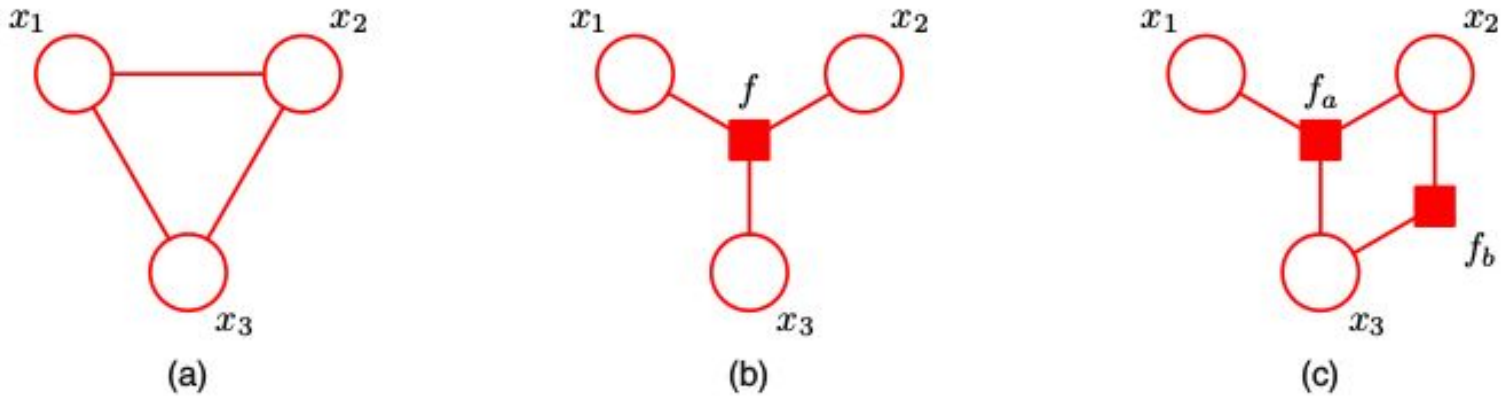
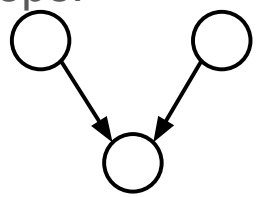


Figure 8.41 (a) An undirected graph with a single clique potential $\psi(x_1, x_2, x_3)$. (b) A factor graph with factor $f(x_1, x_2, x_3) = \psi(x_1, x_2, x_3)$ representing the same distribution as the undirected graph. (c) A different factor graph representing the same distribution, whose factors satisfy $f_a(x_1, x_2, x_3) f_b(x_1, x_2) = \psi(x_1, x_2, x_3)$.

BNs $\Longleftrightarrow$ factor graph

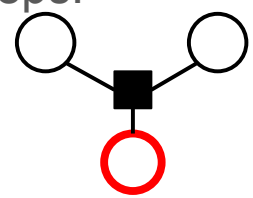
- Converting Bayesian Networks to factor graph takes the following steps:
  - Consider all the parents of a child node
  - “Pinch” all the edges from its parents to the child into one factor
  - Create an additional edge from the factor to the child node
  - Move on the the next child node
  - Last step is to add all the priors as individual “dongles” to the corresponding variables



- Let the original BN have N variables and E edges.
  The converted factor graph will have N+E edges in total

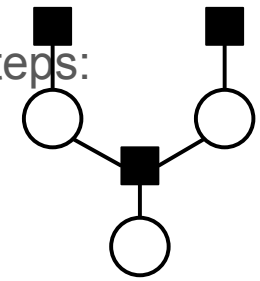
BNs $\Longleftrightarrow$ factor graph

- Converting Bayesian Networks to factor graph takes the following steps:
  - Consider all the parents of a child node
  - “Pinch” all the edges from its parents to the child into one factor
  - Create an additional edge from the factor to the child node
  - Move on the the next child node
  - Last step is to add all the priors as individual “dongles” to the corresponding variables



- Let the original BN have N variables and E edges.
  The converted factor graph will have N+E edges in total

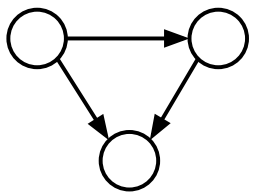
BNs $\Longleftrightarrow$ factor graph

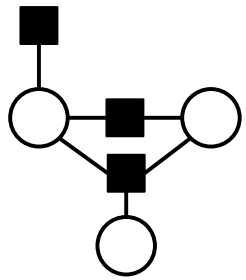


- Converting Bayesian Networks to factor graph takes the following steps:
  - Consider all the parents of a child node
  - “Pinch” all the edges from its parents to the child into one factor
  - Create an additional edge from the factor to the child node
  - Move on the the next child node
  - Last step is to add all the priors as individual “dongles” to the corresponding variables
- Let the original BN have N variables and E edges.
  The converted factor graph will have N+E edges in total

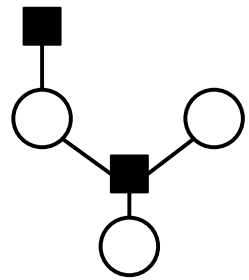
BNs $\Longleftrightarrow$ factor graph

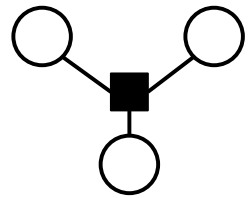
- With this approach you may get factor graphs like the following:





which can be simplified to:

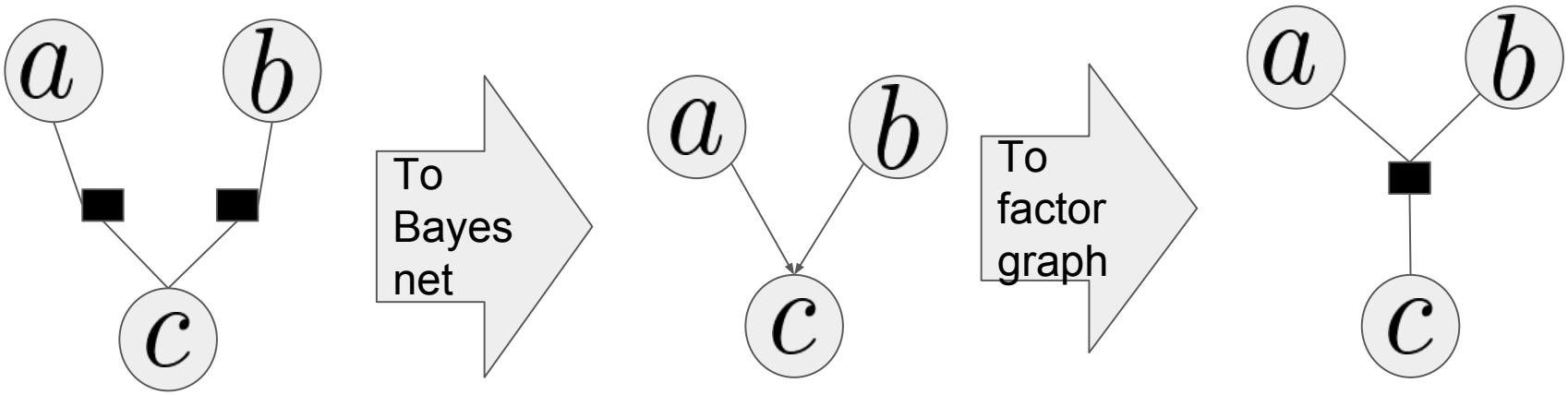




BNs $\Longleftrightarrow$ factor graph

- Convert FG back to BN by just reserving the “pinching” on each factor node
- Then put back the direction on the edge according to the conditional probabilities

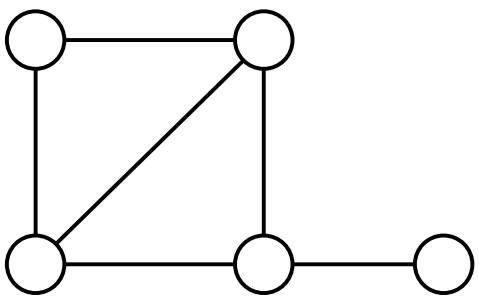
BNs $\Longleftrightarrow$ factor graph



Notice that we don’t get the same factor graph back...

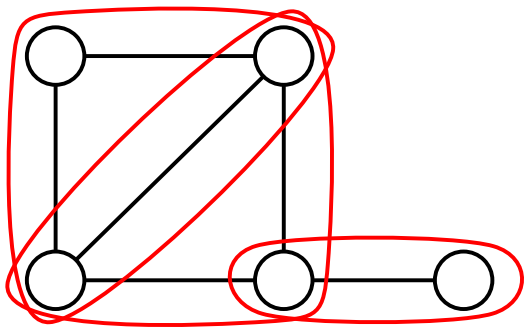
MRF $\Longleftrightarrow$ factor graph

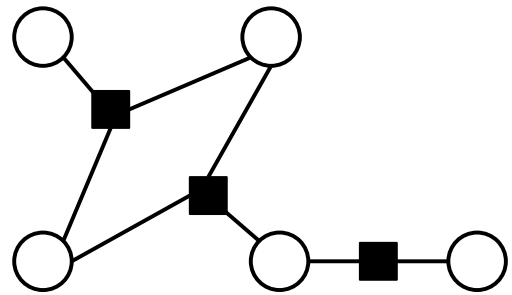
- Converting Markov Random Fields to factor graph takes the following steps:
  - Consider all the maximum cliques of the MRF
  - Create a factor node for each of the maximum cliques
  - Connect all the nodes of the maximum clique to the new factor nodes



MRF $\Longleftrightarrow$ factor graph

- Converting Markov Random Fields to factor graph takes the following steps:
  - Consider all the maximum cliques of the MRF
  - Create a factor node for each of the maximum cliques
  - Connect all the nodes of the maximum clique to the new factor nodes





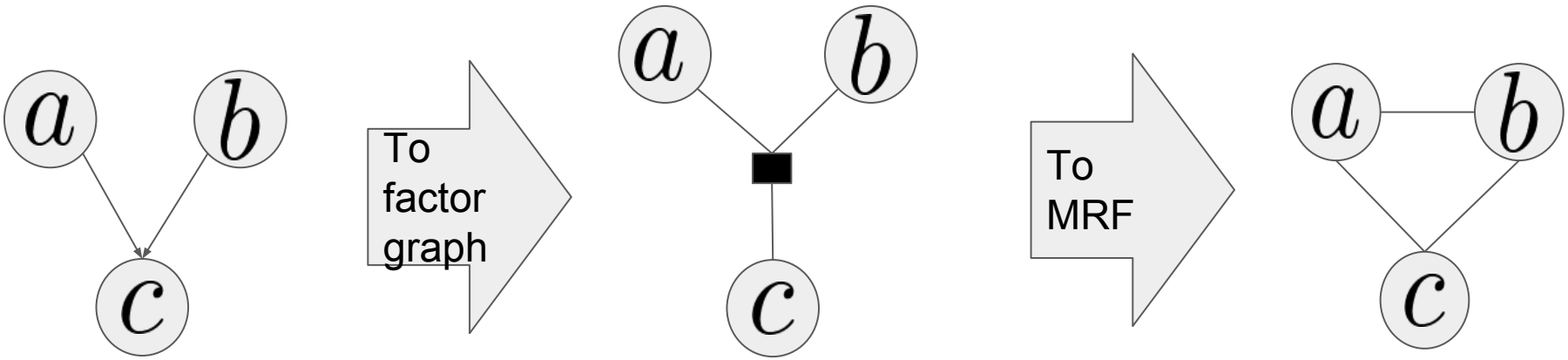
MRF $\Longleftrightarrow$ factor graph

- Convert FG back to MRF is easy
- For each factor, create all pairwise connections of the variables in the factor

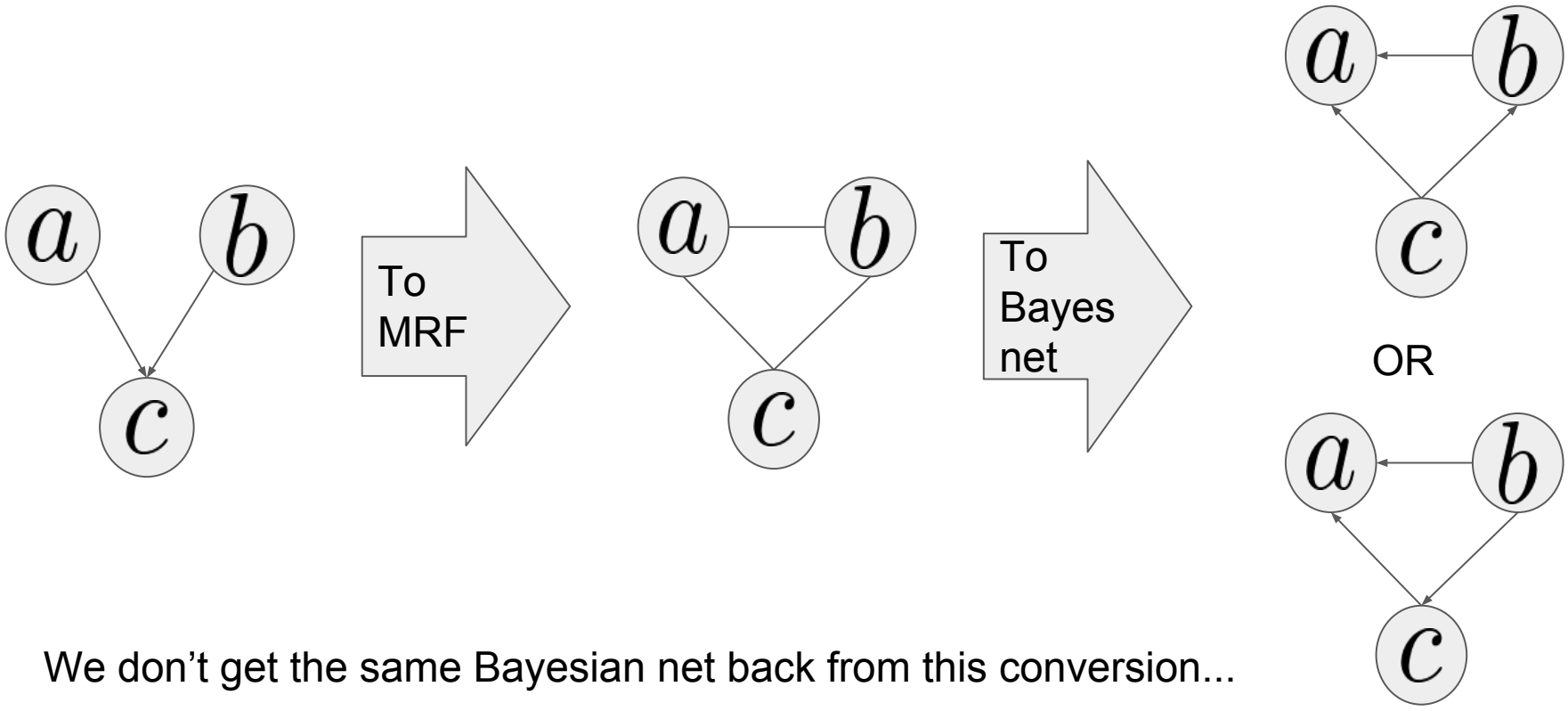
BNs $\Longleftrightarrow$ MRF

Algorithm:

- Create the factor graph for the Bayesian network
- Then remove the factors but add edges between any two nodes that share a factor

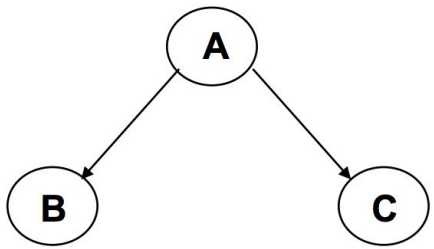


BNs $\Longleftrightarrow$ MRF



We don’t get the same Bayesian net back from this conversion...

Posterior inference example



Conditionally independent effects:
p(A,B,C) = p(B|A)p(C|A)p(A)

B and C are conditionally independent Given A

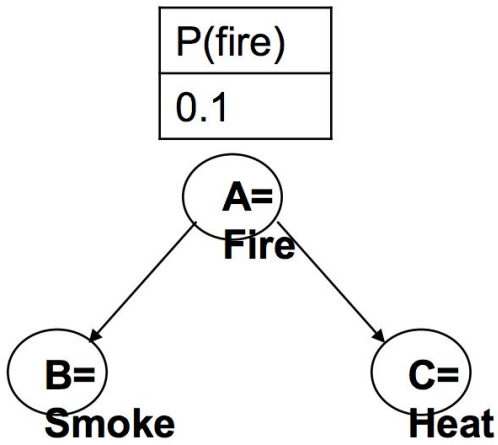
E.g., A is a disease, and we model B and C as conditionally independent symptoms given A

E.g., A is Fire, B is Heat, C is Smoke. "Where there's Smoke, there's Fire."

If we see Smoke, we can infer Fire.

If we see Smoke, observing Heat tells us very little additional information.

Posterior inference example



Fire	P(Smoke)
t	.90
f	.001

Fire	P(Heat)
t	.99
f	.0001

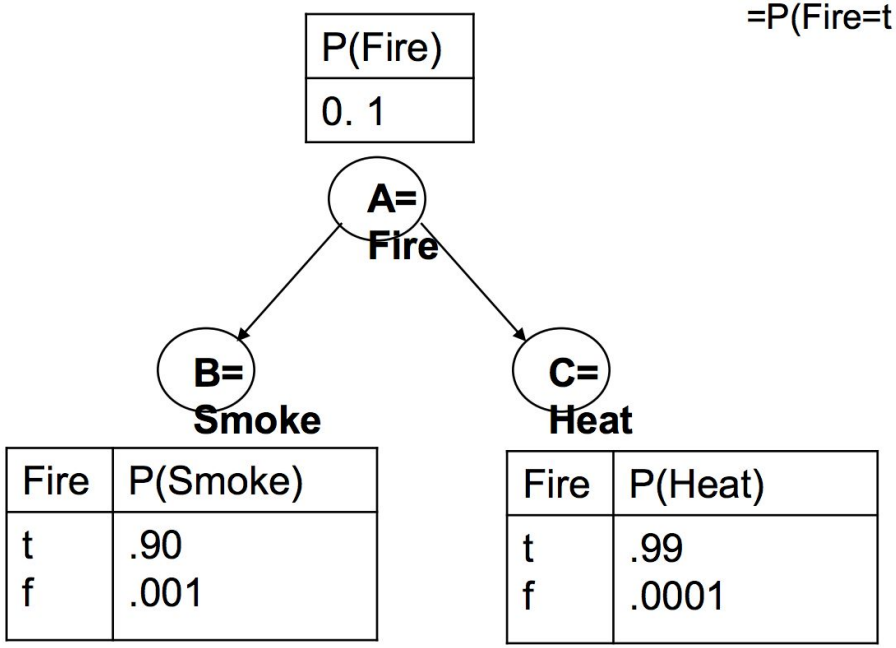
Conditionally independent effects:
P(A,B,C) = P(B|A)P(C|A)P(A)

Smoke and Heat are conditionally independent given Fire.

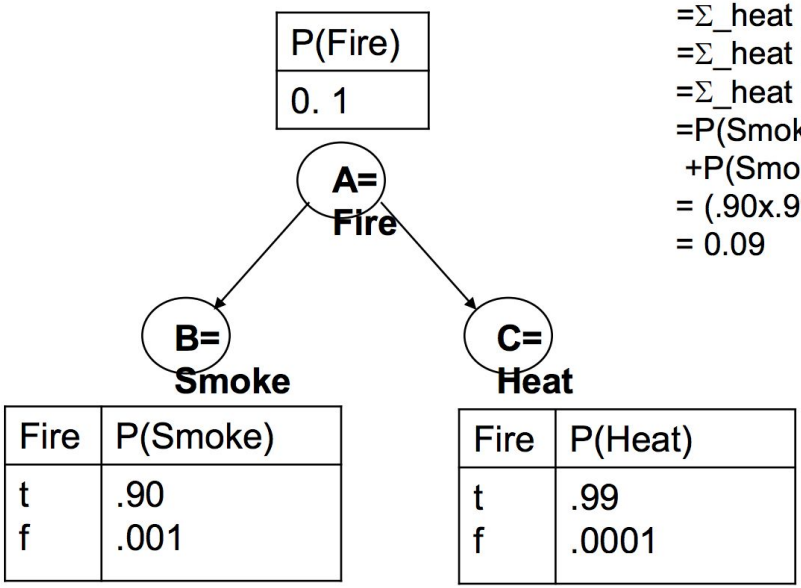
If we see B=Smoke, observing C=Heat tells us very little additional information.

Posterior inference example

What is P(Fire=t | Smoke=t)?
P(Fire=t | Smoke=t)
=P(Fire=t & Smoke=t) / P(Smoke=t)



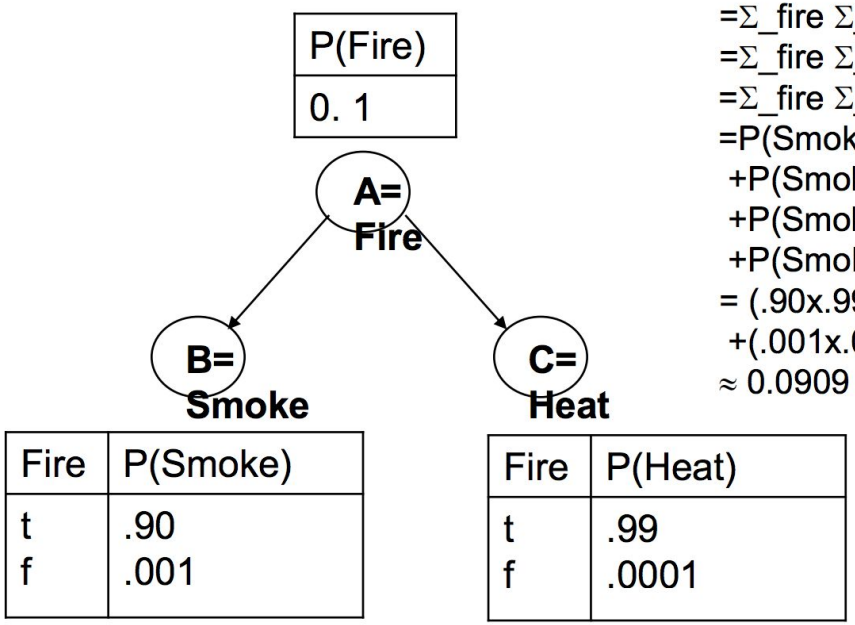
Posterior inference example



What is P(Fire=t & Smoke=t)?

P(Fire=t & Smoke=t)
=Σ_heat P(Fire=t&Smoke=t&heat)
=Σ_heat P(Smoke=t&heat|Fire=t)P(Fire=t)
=Σ_heat P(Smoke=t|Fire=t) P(heat|Fire=t)P(Fire=t)
=P(Smoke=t|Fire=t) P(heat=t|Fire=t)P(Fire=t)
+P(Smoke=t|Fire=t)P(heat=f|Fire=t)P(Fire=t)
= (.90x.99x.1)+(.90x.01x.1)
= 0.09

Posterior inference example



What is P(Smoke=t)?

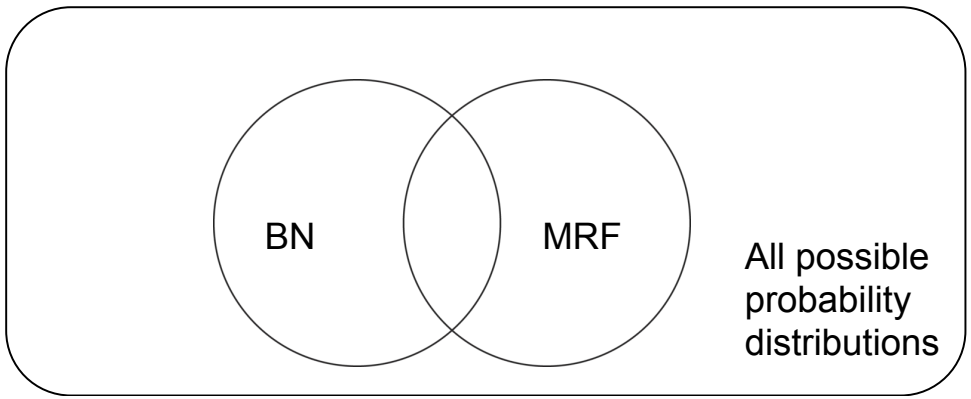
P(Smoke=t)
=Σ_fire Σ_heat P(Smoke=t&fire&heat)
=Σ_fire Σ_heat P(Smoke=t&heat|fire)P(fire)
=Σ_fire Σ_heat P(Smoke=t|fire) P(heat|fire)P(fire)
=P(Smoke=t|fire=t) P(heat=t|fire=t)P(fire=t)
+P(Smoke=t|fire=t)P(heat=f|fire=t)P(fire=t)
+P(Smoke=t|fire=f) P(heat=t|fire=f)P(fire=f)
+P(Smoke=t|fire=f)P(heat=f|fire=f)P(fire=f)
= (.90x.99x.1)+(.90x.01x.1)
+(.001x.0001x.9)+(.001x.9999x.9)
≈ 0.0909

Graphical models

- A graphical model expresses two properties about a joint distribution:
  - Conditional independence
  - Factorization
- Neither BNs nor MRFs can represent all the possible conditional independence and factorization properties

Graphical models

- A graphical model expresses two properties about a joint distribution:
  - Conditional independence
  - Factorization
- Neither BNs nor MRFs can represent all the possible conditional independence and factorization properties



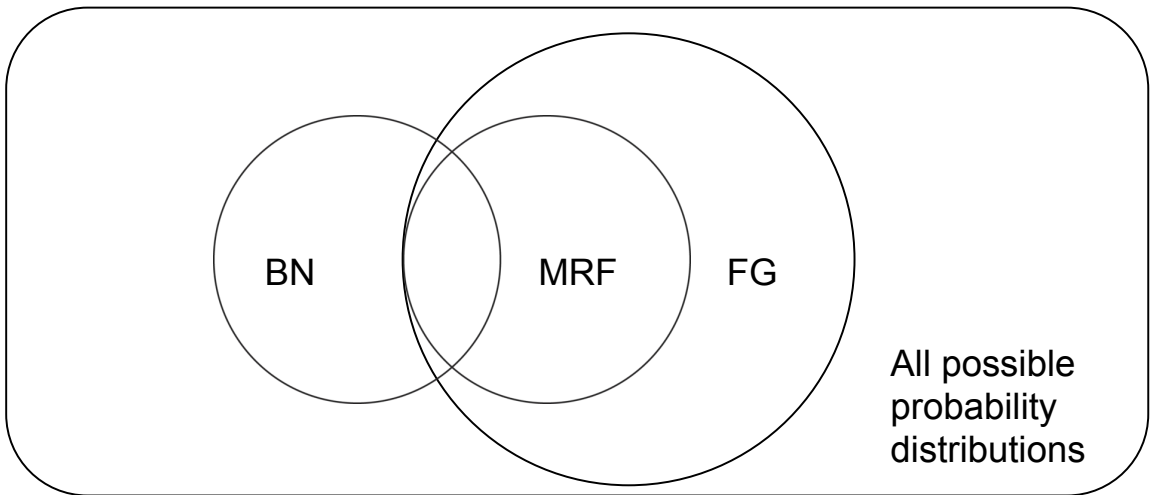
Graphical models

- BN and MRFs are not one-to-one mapping for both conditional independence and factorization properties

- i.e. if we convert the graph representations back and forth, we would obtain a different graph from what we started

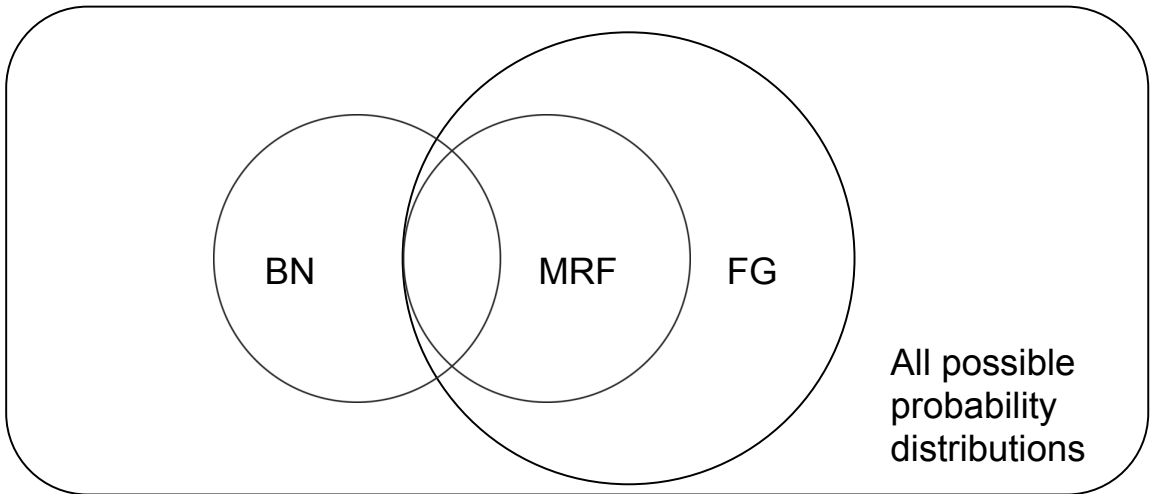
Graphical models

- Factor graph is meant to unify both BN and MRF
- But...



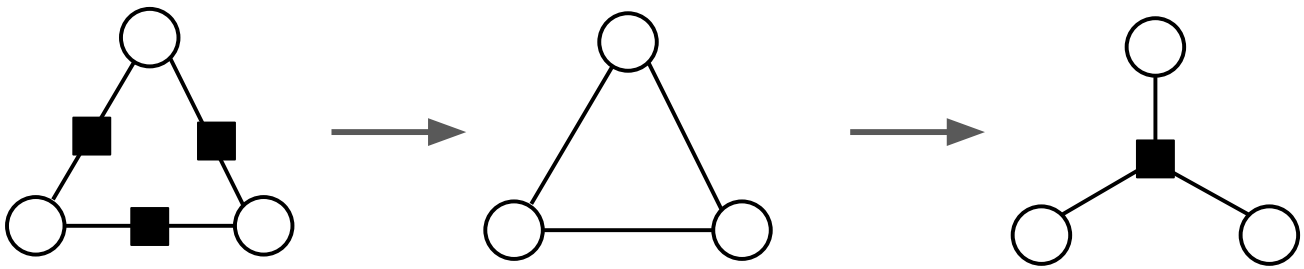
Graphical models

- Factor graph has one-to-one mapping to MRF **ONLY** in terms of conditional independence properties
- The factorization properties does not carry over during conversion



Graphical models

- Factor graph has one-to-one mapping to MRF **ONLY** in terms of conditional independence properties
- The factorization properties does not carry over during conversion



a FG expresses pair-wise factorization

Convert the FG on the left to a MRF

Convert the MRF back to a FG, we lost the factorization property

Hidden Markov Models

- Parameters – stationary/homogeneous markov model (independent of time t)

Initial probabilities

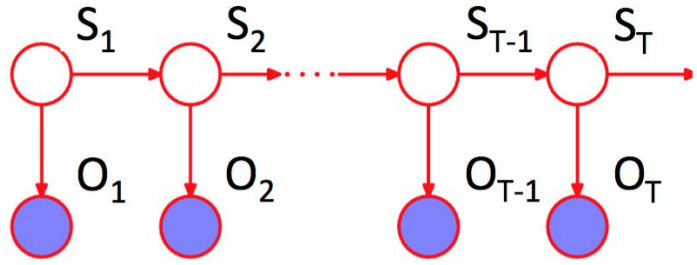
$p(S_1 = i) = \pi_i$

Transition probabilities

$p(S_t = j | S_{t-1} = i) = p_{ij}$

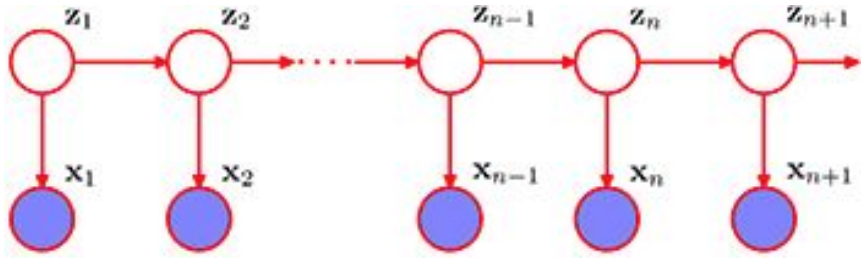
Emission probabilities

$p(O_t = y | S_t = i) = q_i^y$



$$p(\{S_t\}_{t=1}^T, \{O_t\}_{t=1}^T) = p(S_1) \prod_{t=2}^{T} p(S_t|S_{t-1}) \prod_{t=1}^{T} p(O_t|S_t)$$

Example of Hidden Markov Model



- States Z: L/H (atmospheric pressure).
- Observations X: R/D .
- Transition probabilities:

t|t-1	L	H
L	0.3	0.2
H	0.7	0.8

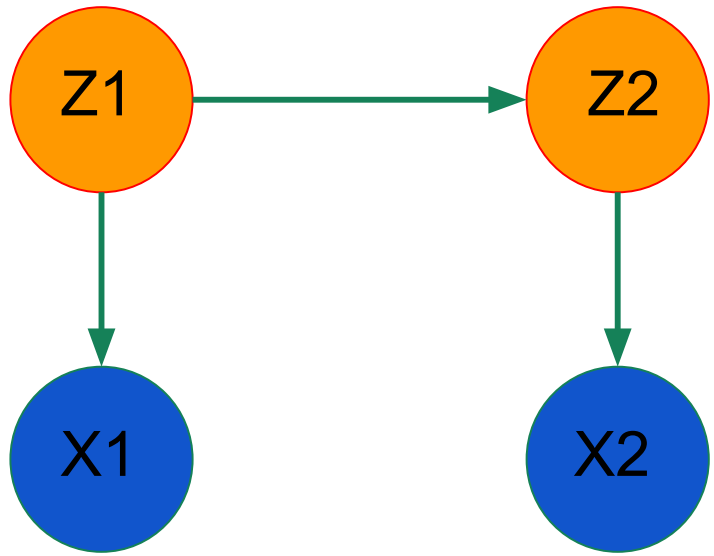
Observation probabilities:

X|Z	L	H
R	0.6	0.4
D	0.4	0.6

- Initial probabilities: say P(Z_1=L)=0.4 , P(Z_1=H)=0.6 .

Example of Hidden Markov Model

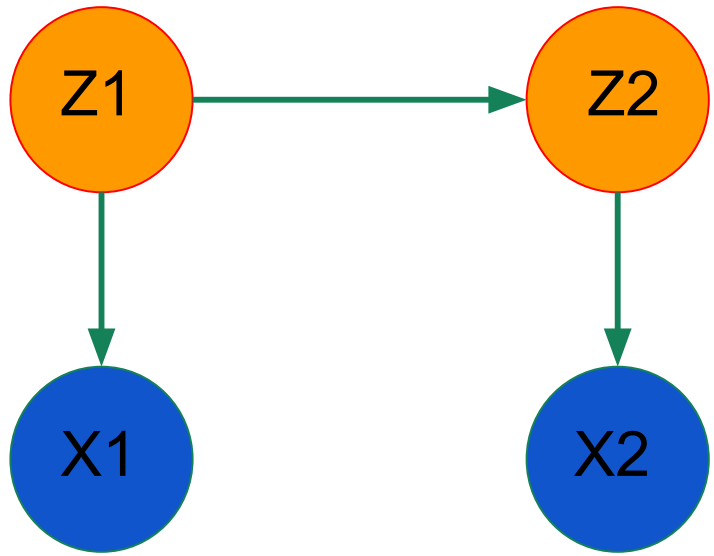
- Ex1:
- What is P(X1=D,X2=R)?



Example of Hidden Markov Model

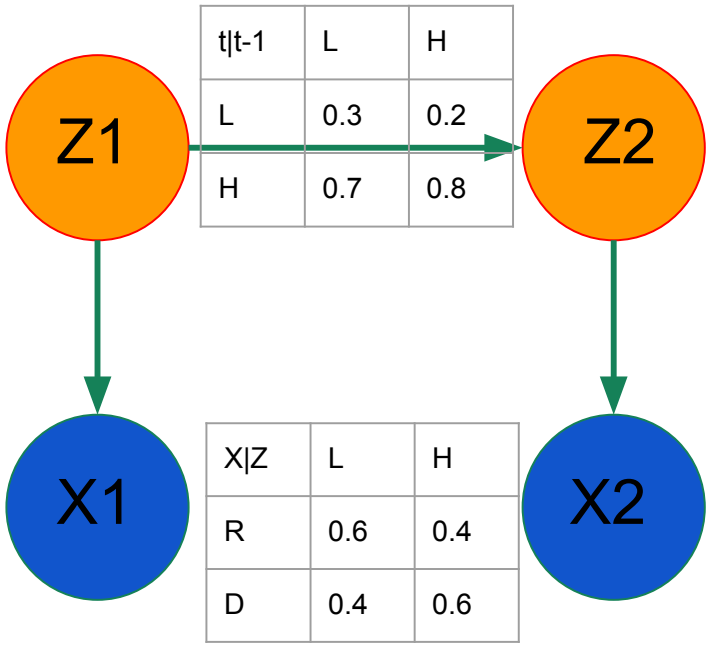
- Ex1:
- What is P(X1=D,X2=R)?

- P(X1=D,X2=R) =
  P(X1=D,X2=R,Z1=L,Z2=L)+
  P(X1=D,X2=R,Z1=L,Z2=H)+
  P(X1=D,X2=R,Z1=H,Z2=L)+
  P(X1=D,X2=R,Z1=H,Z2=H)



Example of Hidden Markov Model

- First term:
  P(X1=D,X2=R,Z1=L,Z2=L)
  = P(X1=D,X2=R| Z1=L,Z2=L)*P(Z1=L,Z2=L)
  = P(X1=D| Z1=L)*
  P(X2=R| Z2=L)*P(Z1=L)*P(Z2=L|Z1=L)
  = 0.4*0.6*0.4*0.3 = 0.0288



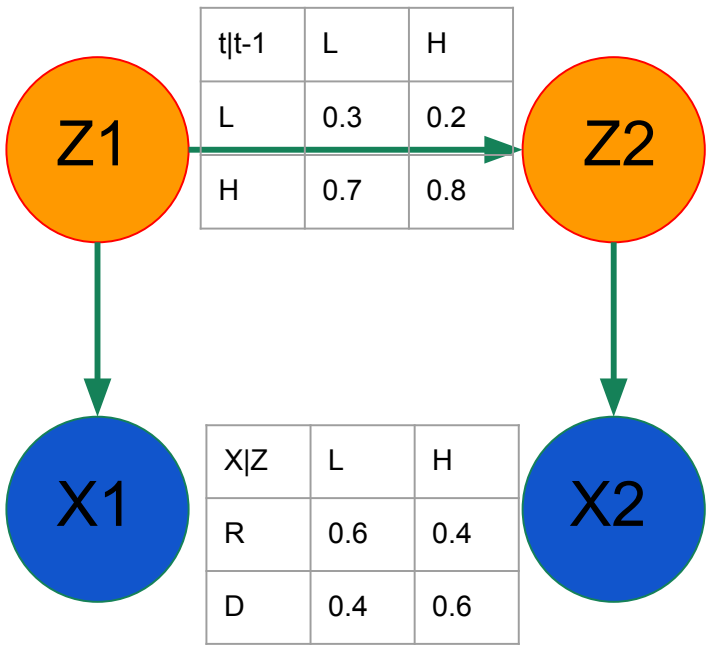
t|t-1	L	H
L	0.3	0.2
H	0.7	0.8

X|Z	L	H
R	0.6	0.4
D	0.4	0.6

Example of Hidden Markov Model

- Ex1:
- What is P(X1=D,X2=R)?

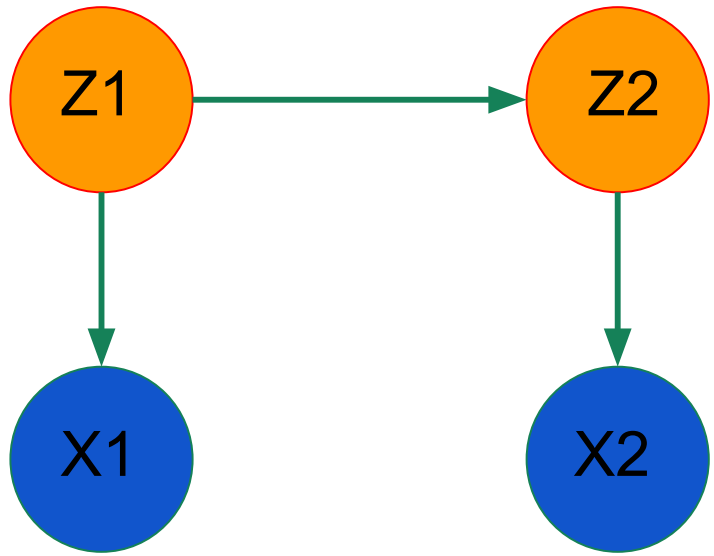
- P(X1=D,X2=R) =
  P(X1=D,X2=R,Z1=L,Z2=L)+
  P(X1=D,X2=R,Z1=L,Z2=H)+
  P(X1=D,X2=R,Z1=H,Z2=L)+
  P(X1=D,X2=R,Z1=H,Z2=H)
  = 0.4*0.6*0.4*0.3+
  0.4*0.4*0.4*0.7+
  0.6*0.6*0.6*0.2+
  0.6*0.4*0.6*0.8 = 0.232



Example of Hidden Markov Model

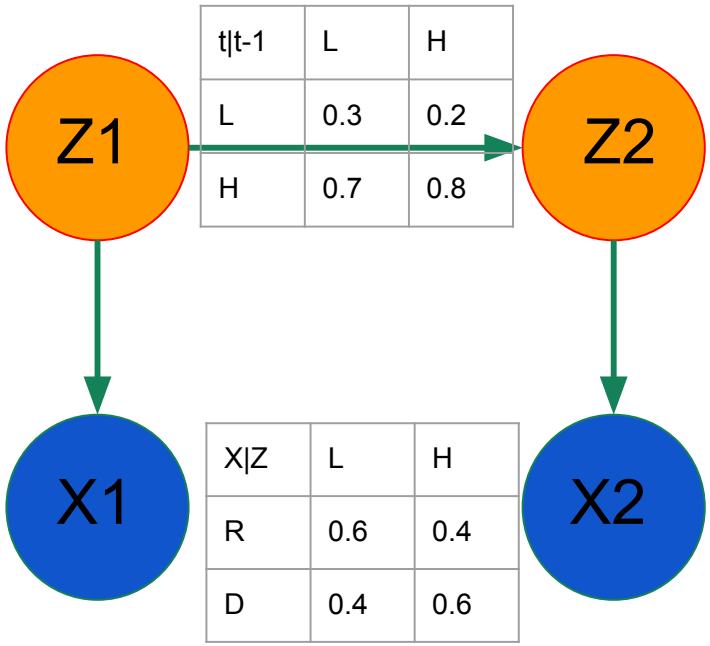
- Ex2:
- What is P(Z1=H|X1=D,X2=R)?

- P(Z1=H|X1=D,X2=R)=
  P(Z1=H,X1=D,X2=R)/P(X1=D,X2=R)



Example of Hidden Markov Model

- P(Z1=H,X1=D,X2=R)=
  P(Z1=H,Z2=L,X1=D,X2=R)+
  P(Z1=H,Z2=H,X1=D,X2=R)
  =0.6*0.6*0.6*0.2+
  0.6*0.4*0.6*0.8
  =0.1584



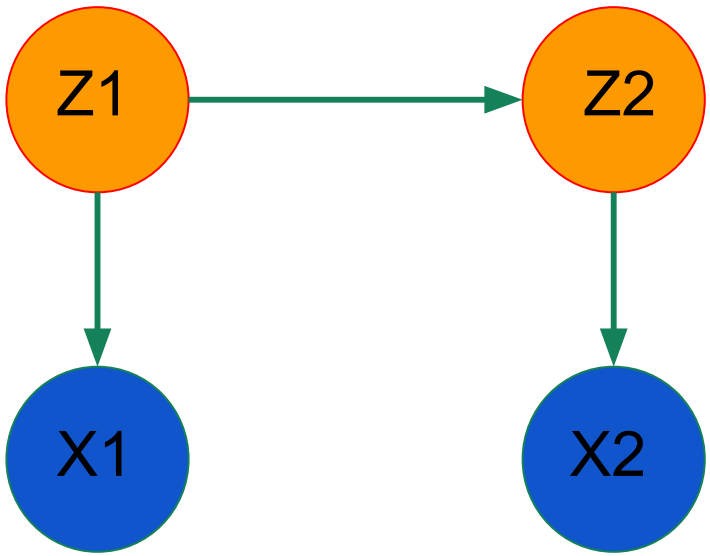
t|t-1	L	H
L	0.3	0.2
H	0.7	0.8

X|Z	L	H
R	0.6	0.4
D	0.4	0.6

Example of Hidden Markov Model

- Ex2:
- What is P(Z1=H|X1=D,X2=R)?

- P(Z1=H|X1=D,X2=R)=
  P(Z1=H,X1=D,X2=R)/P(X1=D,X2=R)
  =0.1584/0.232=0.683

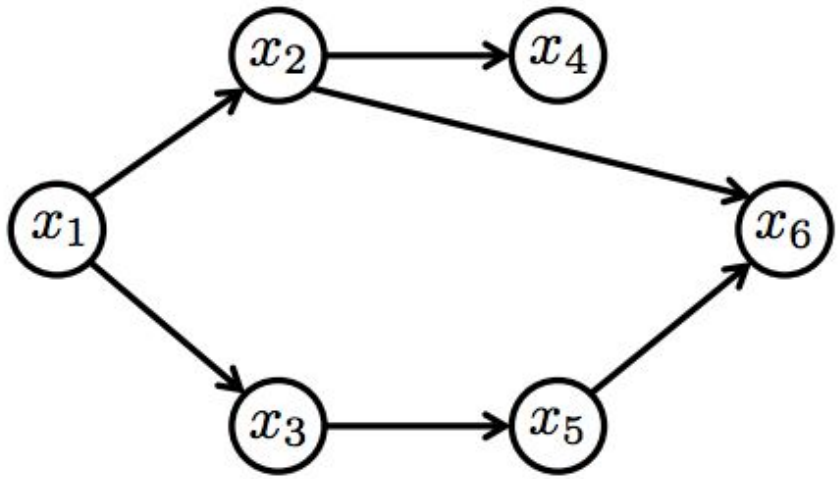


Introduction to A4

1, Graphical Models

$$p(x) = p(x_1)p(x_2 \mid x_1)p(x_3 \mid x_1)p(x_4 \mid x_2)p(x_5 \mid x_3)p(x_6 \mid x_2, x_5)$$

Draw Bayes-net:



Introduction to A4

Factor graph:

$$p(x) = \frac{1}{Z} \prod_{f \in \mathcal{F}} \psi_f(x_f) \qquad Z = \sum_x \prod_{f \in \mathcal{F}} \psi_f(x_f)$$

$Z > 0 \longrightarrow$ normalization constant (partition function)

$\psi_f(x_f) \geq 0 \longrightarrow$ arbitrary non-negative *potential function*

$\mathcal{F} \longrightarrow$ set of hyperedges linking subsets of nodes $f \subseteq \mathcal{V}$

$\mathcal{V} \longrightarrow$ set of N nodes or vertices, $\{1, 2, \ldots, N\}$

Introduction to A4

1, Graphical Models

$$p(x) = p(x_1)p(x_2 \mid x_1)p(x_3 \mid x_1)p(x_4 \mid x_2)p(x_5 \mid x_3)p(x_6 \mid x_2, x_5)$$

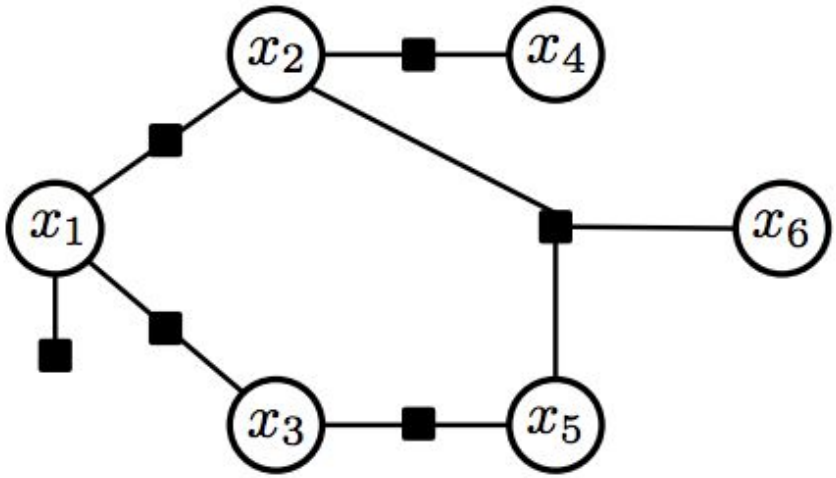
Draw factor-graph: $p(x) \propto \psi_1(x_1)\psi_2(x_2, x_1)\psi_3(x_3, x_1)\psi_4(x_4, x_2)\psi_5(x_5, x_3)\psi_6(x_6, x_2, x_5)$

Introduction to A4

1, Graphical Models

$$p(x) = p(x_1)p(x_2 \mid x_1)p(x_3 \mid x_1)p(x_4 \mid x_2)p(x_5 \mid x_3)p(x_6 \mid x_2, x_5)$$

Draw factor-graph: $p(x) \propto \psi_1(x_1)\psi_2(x_2, x_1)\psi_3(x_3, x_1)\psi_4(x_4, x_2)\psi_5(x_5, x_3)\psi_6(x_6, x_2, x_5)$

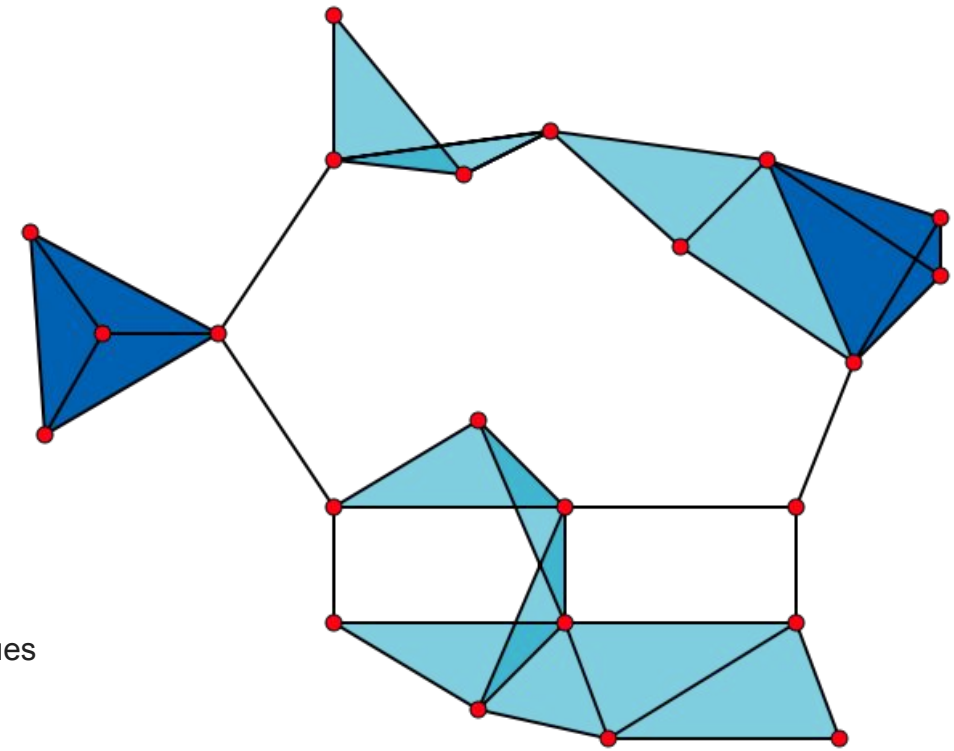


Introduction to A4

1, Graphical Models

Clique: A subset of vertices of an undirected graph such that every two distinct vertices in the clique are adjacent.

Maximal Clique: A clique that cannot be extended by including one more adjacent vertex



The 11 light blue 3-cliques (triangles) form maximal cliques

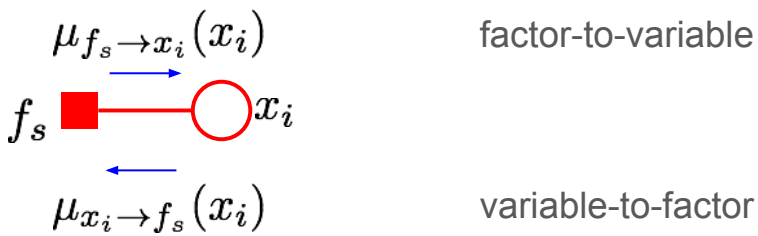
The 2 dark blue 4-cliques form maximal cliques

Sum Product Algorithm

- The sum-product algorithm is used to compute probabilities for a **subset** of the variables of a joint distribution, e.g. P(a, b, c, d)
  - Marginal distributions, e.g. P(b)
  - Joint distributions of a subset of variables, e.g. P(a,b)
  - Conditional distributions (often the posterior distributions), e.g. P(a,c | d)=P(a,c,d) / P(d)

Message notations

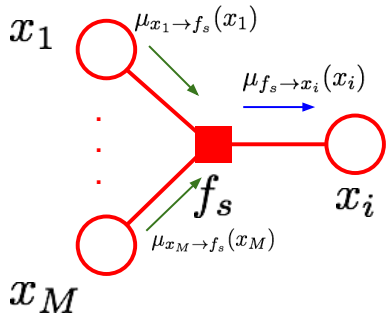
- We use function μ to denote messages.
- On each edge of the factor graph, there are two messages traveling in opposite directions
- We use subscript to denote the origin and the destination of these messages, e.g.:



The sum product algorithm on factor graph

- Two rules in the sum-product algorithm:
  - Factor-to-variable messages:

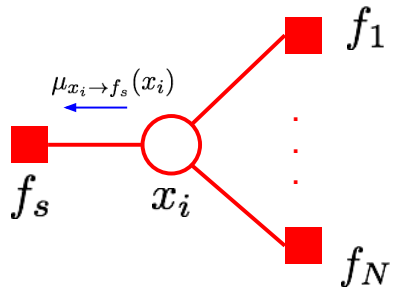
$$\mu_{f_s \to x_i}(x_i) = \sum_{Ne(f_s)\backslash x_i} f_s(x_i, x_1, \cdots, x_M) \underbrace{\prod_{x_m \in Ne(f_s)\backslash x_i} \mu_{x_m \to f_s}(x_m)}_{\text{Incoming messages}}$$



The sum product algorithm on factor graph

- Two rules in the sum-product algorithm:
  - Variable-to-factor messages:

$$\mu_{x_i \to f_s}(x_i) = \prod_{f_n \in Ne(x_i) \setminus f_s} \mu_{f_n \to x_i}(x_i)$$



The sum product algorithm on factor graph

- Two rules in the sum-product algorithm:
  - Factor-to-variable messages:

$$\mu_{f_s \to x_i}(x_i) = \sum_{Ne(f_s)\setminus x_i} f_s(x_i, x_1, \cdots, x_M) \prod_{x_m \in Ne(f_s)\setminus x_i} \mu_{x_m \to f_s}(x_m)$$

  - Variable-to-factor messages:

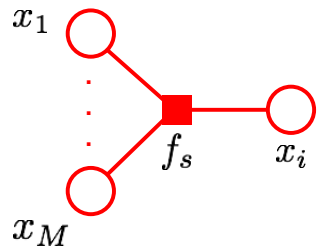
$$\mu_{x_i \to f_s}(x_i) = \prod_{f_n \in Ne(x_i)\setminus f_s} \mu_{f_n \to x_i}(x_i)$$

The sum product algorithm on factor graph

- How to start the sum-product algorithm:
  - Choose a node in the factor graph as the root node
  - Compute all the leaf-to-root messages
  - Compute all the root-to-leaf messages
- Initial conditions:
  - Starting from a factor leaf/root node, the initial factor-to-variable message is the factor itself
  - Starting from a variable leaf/root node, the initial variable-to-factor message is a vector of ones

Compute probabilities with messages

- How to convert messages to actual probabilities:



Compute unnormalized probabilities using messages (the sum-product algorithm):

unnormalized marginal probabilities:

$$g(x_i) = \prod_{f_n \in Ne(x_i)} \mu_{f_n \to x_i}(x_i)$$

normalization constant:

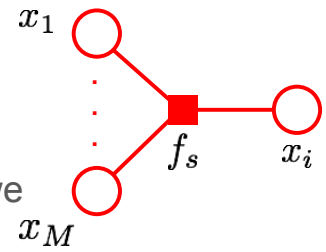
$$Z = \sum_{x_i} g(x_i)$$

marginal probabilities:

$$P(x_i) = \frac{1}{Z} g(x_i)$$

Compute probabilities with messages

- How to convert messages to actual probabilities:



Let $\{x_i\} \cup X_s \subseteq \{x_i, x_1, \cdots, x_M\}$ be a subset of the variables we wish to compute joint distribution over

Compute unnormalized probabilities using messages (the sum-product algorithm):

unnormalized joint probabilities:

$$g(x_i, X_s) = f_s(x_i, X_s) \prod_{x_m \in \{x_i, X_s\}} \prod_{f_n \in Ne(x_m) \setminus f_s} \mu_{f_n \to x_m}(x_m)$$

When all the variables in $\{x_i\} \cup X_s$ are connected to the same factor. We make the computation efficient

normalization constant:

$$Z = \sum_{x_i} g(x_i)$$

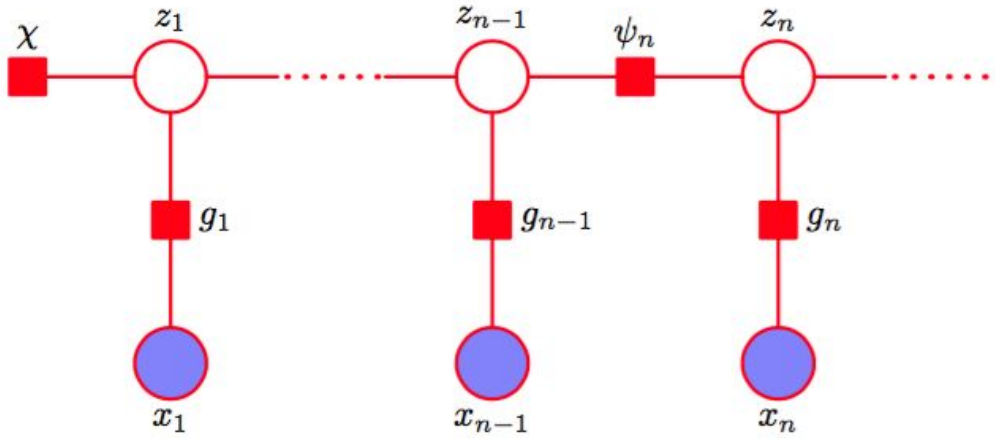
marginal probabilities:

$$P(x_i, x_1, \cdots, x_k) = \frac{1}{Z} g(x_i, x_1, \cdots, x_k)$$

Equivalence

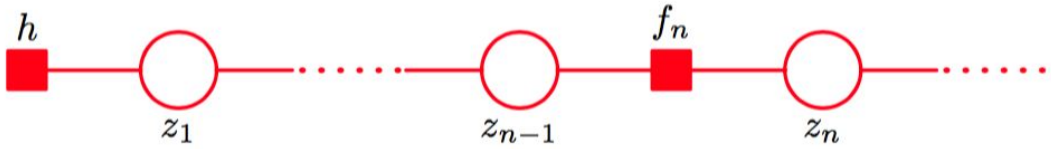
Sum-product algorithm is equivalent to forward-backward algorithm in the context of Hidden Markov Models

Factor graph of HMM:



Forward-Backward Algorithm

For the purpose of inference, we can simplify the factor graph as below:

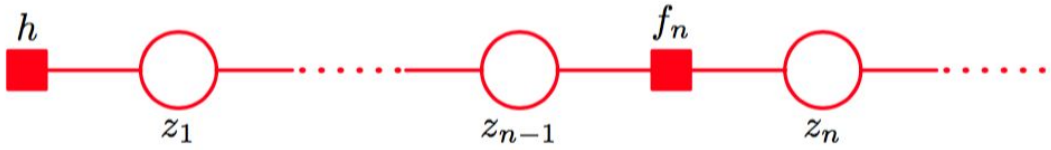


Factors are:

$$h(\mathbf{z}_1) = p(\mathbf{z}_1)p(\mathbf{x}_1|\mathbf{z}_1)$$
$$f_n(\mathbf{z}_{n-1}, \mathbf{z}_n) = p(\mathbf{z}_n|\mathbf{z}_{n-1})p(\mathbf{x}_n|\mathbf{z}_n)$$

Forward-Backward Algorithm

For the purpose of inference, we can simplify the factor graph as below:



Factors are:

$$h(\mathbf{z}_1) = p(\mathbf{z}_1)p(\mathbf{x}_1|\mathbf{z}_1)$$
$$f_n(\mathbf{z}_{n-1}, \mathbf{z}_n) = p(\mathbf{z}_n|\mathbf{z}_{n-1})p(\mathbf{x}_n|\mathbf{z}_n)$$

Messages are:

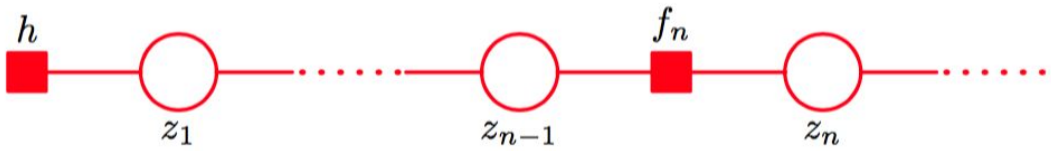
$$\mu_{\mathbf{z}_{n-1}\to f_n}(\mathbf{z}_{n-1}) = \mu_{f_{n-1}\to \mathbf{z}_{n-1}}(\mathbf{z}_{n-1})$$

Quiz: Why?

$$\mu_{f_n\to \mathbf{z}_n}(\mathbf{z}_n) = \sum_{\mathbf{z}_{n-1}} f_n(\mathbf{z}_{n-1}, \mathbf{z}_n)\mu_{\mathbf{z}_{n-1}\to f_n}(\mathbf{z}_{n-1})$$

Forward-Backward Algorithm

For the purpose of inference, we can simplify the factor graph as below:



Rewrite message:

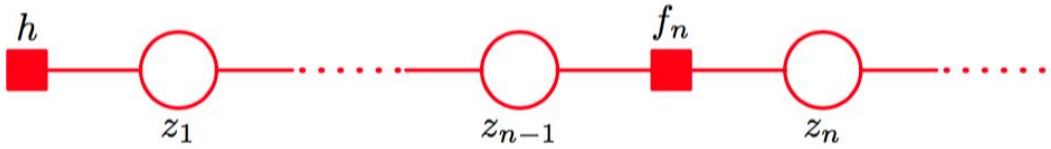
$$\mu_{\mathbf{z}_{n-1}\to f_n}(\mathbf{z}_{n-1}) = \mu_{f_{n-1}\to \mathbf{z}_{n-1}}(\mathbf{z}_{n-1})$$

$$\mu_{f_n\to \mathbf{z}_n}(\mathbf{z}_n) = \sum_{\mathbf{z}_{n-1}} f_n(\mathbf{z}_{n-1},\mathbf{z}_n)\mu_{\mathbf{z}_{n-1}\to f_n}(\mathbf{z}_{n-1})$$

$$\mu_{f_n\to \mathbf{z}_n}(\mathbf{z}_n) = \sum_{\mathbf{z}_{n-1}} f_n(\mathbf{z}_{n-1},\mathbf{z}_n)\mu_{f_{n-1}\to \mathbf{z}_{n-1}}(\mathbf{z}_{n-1})$$

Forward-Backward Algorithm

For the purpose of inference, we can simplify the factor graph as below:



Rewrite message:

$$\mu_{f_n \to \mathbf{z}_n}(\mathbf{z}_n) = \sum_{\mathbf{z}_{n-1}} f_n(\mathbf{z}_{n-1}, \mathbf{z}_n) \mu_{f_{n-1} \to \mathbf{z}_{n-1}}(\mathbf{z}_{n-1})$$

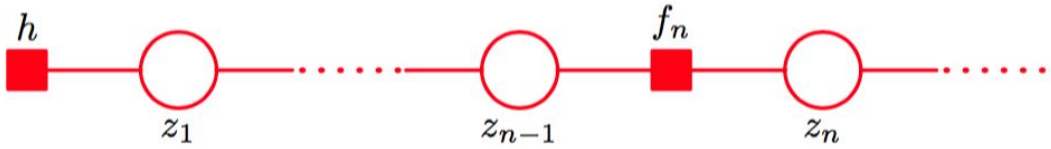
$$\alpha(\mathbf{z}_n) = \mu_{f_n \to \mathbf{z}_n}(\mathbf{z}_n)$$

Replace factor:

$$\alpha(\mathbf{z}_n) = p(\mathbf{x}_n|\mathbf{z}_n) \sum_{\mathbf{z}_{n-1}} \alpha(\mathbf{z}_{n-1}) p(\mathbf{z}_n|\mathbf{z}_{n-1})$$

Forward-Backward Algorithm

For the purpose of inference, we can simplify the factor graph as below:



Rewrite message:

$$\mu_{z_{n+1} \to f_{n+1}}(\mathbf{z}_{n+1}) = \mu_{f_{n+2} \to z_{n+1}}(\mathbf{z}_{n+1})$$

$$\mu_{f_{n+1} \to z_n}(\mathbf{z}_n) = \sum_{\mathbf{z}_{n+1}} f_{n+1}(\mathbf{z}_n, \mathbf{z}_{n+1}) \mu_{f_{n+2} \to z_{n+1}}(\mathbf{z}_{n+1})$$

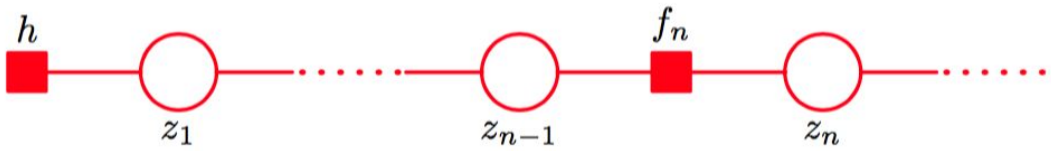
$$\beta(\mathbf{z}_n) = \mu_{f_{n+1} \to \mathbf{z}_n}(\mathbf{z}_n)$$

Replace factor:

$$\beta(\mathbf{z}_n) = \sum_{\mathbf{z}_{n+1}} \beta(\mathbf{z}_{n+1}) p(\mathbf{x}_{n+1} | \mathbf{z}_{n+1}) p(\mathbf{z}_{n+1} | \mathbf{z}_n)$$

Forward-Backward Algorithm

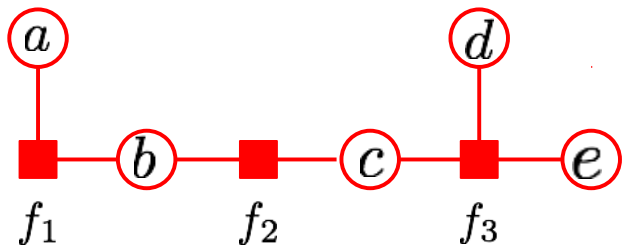
For the purpose of inference, we can simplify the factor graph as below:



Obtain Posterior: $p(\mathbf{z}_n, \mathbf{X}) = \mu_{f_n \to \mathbf{z}_n}(\mathbf{z}_n)\mu_{f_{n+1} \to \mathbf{z}_n}(\mathbf{z}_n) = \alpha(\mathbf{z}_n)\beta(\mathbf{z}_n)$

$$\gamma(\mathbf{z}_n) = \frac{p(\mathbf{z}_n, \mathbf{X})}{p(\mathbf{X})} = \frac{\alpha(\mathbf{z}_n)\beta(\mathbf{z}_n)}{p(\mathbf{X})}$$

Example of the sum product algorithm



A set of binary random variables: $a, b, c, d, e \in \{0, 1\}$

$$f_1(a, b) = \begin{bmatrix} f_1(0,0), f_1(1,0) \\ f_1(0,1), f_1(1,1) \end{bmatrix} = \begin{bmatrix} 1, 2 \\ 1, 2 \end{bmatrix}$$

$$f_2(b, c) = \begin{bmatrix} 3, 2 \\ 1, 0.5 \end{bmatrix}$$

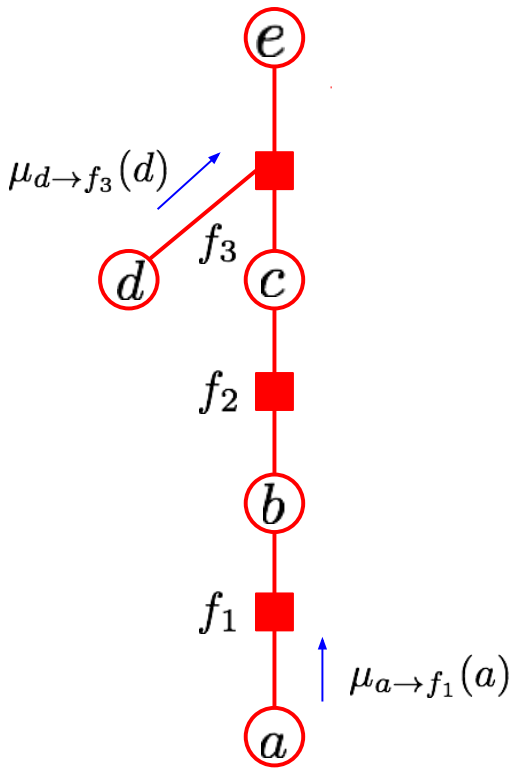
$$f_3(c = 0, d, e) = \begin{bmatrix} 1, 2 \\ 2, 1 \end{bmatrix}$$

$$f_3(c = 1, d, e) = \begin{bmatrix} 1, 2 \\ 2, 1 \end{bmatrix}$$

The sum product algorithm on factor graph

- How to start the sum-product algorithm:
  - **Choose a node in the factor graph as the root node**
  - Compute all the leaf-to-root messages
  - Compute all the root-to-leaf messages
- Initial conditions:
  - Starting from a factor leaf/root node, the initial factor-to-variable message is the factor itself
  - Starting from a variable leaf/root node, the initial variable-to-factor message is a vector of ones

Example of the sum product algorithm



$$\mu_{a\to f_1}(a) = [1,1]^T$$
$$\mu_{d\to f_3}(d) = [1,1]^T$$

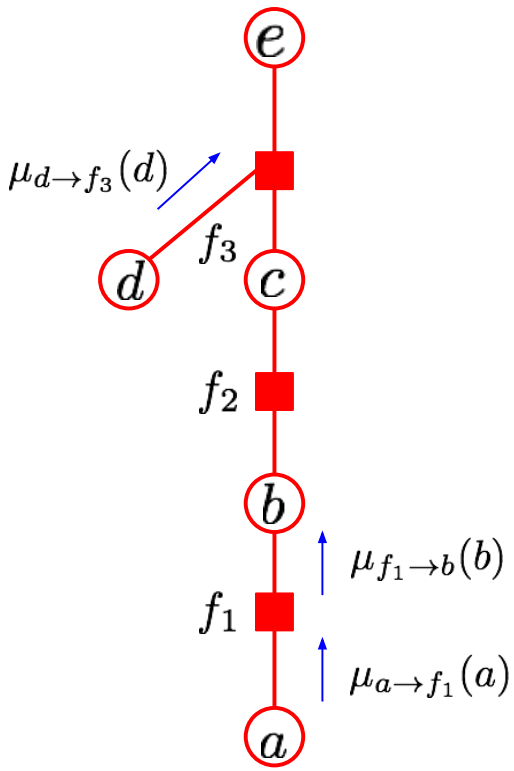
$$f_1(a,b) = \begin{bmatrix} f_1(0,0), f_1(1,0) \\ f_1(0,1), f_1(1,1) \end{bmatrix} = \begin{bmatrix} 1,2 \\ 1,2 \end{bmatrix}$$

$$f_2(b,c) = \begin{bmatrix} 3,2 \\ 1,0.5 \end{bmatrix}$$

$$f_3(c=0,d,e) = \begin{bmatrix} 1,2 \\ 2,1 \end{bmatrix}$$

$$f_3(c=1,d,e) = \begin{bmatrix} 1,2 \\ 2,1 \end{bmatrix}$$

Example of the sum product algorithm



$$\mu_{a \to f_1}(a) = [1,1]^T$$
$$\mu_{d \to f_3}(d) = [1,1]^T$$

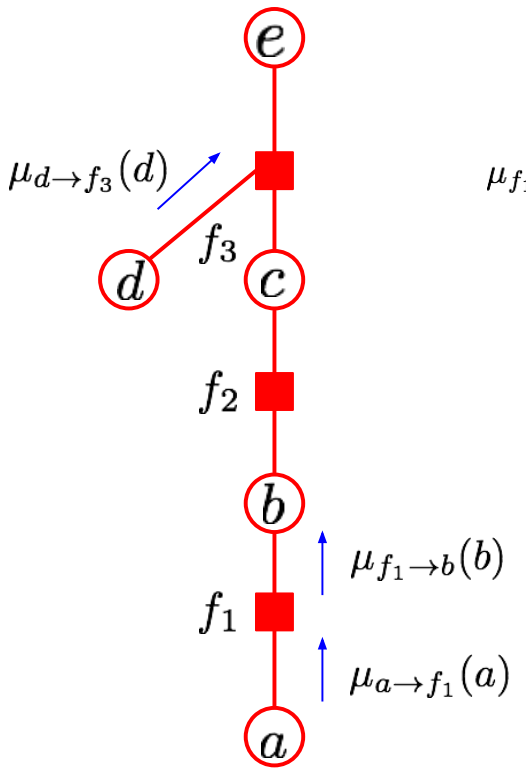
$$f_1(a,b) = \begin{bmatrix} f_1(0,0), f_1(1,0) \\ f_1(0,1), f_1(1,1) \end{bmatrix} = \begin{bmatrix} 1,2 \\ 1,2 \end{bmatrix}$$

$$f_2(b,c) = \begin{bmatrix} 3,2 \\ 1,0.5 \end{bmatrix}$$

$$f_3(c=0,d,e) = \begin{bmatrix} 1,2 \\ 2,1 \end{bmatrix}$$

$$f_3(c=1,d,e) = \begin{bmatrix} 1,2 \\ 2,1 \end{bmatrix}$$

Example of the sum product algorithm



$$\mu_{a\to f_1}(a) = [1,1]^T$$
$$\mu_{d\to f_3}(d) = [1,1]^T$$

$$\mu_{f_1\to b}(b) = \sum_a f_1(a,b)\mu_{a\to f_1(a)}(a)$$

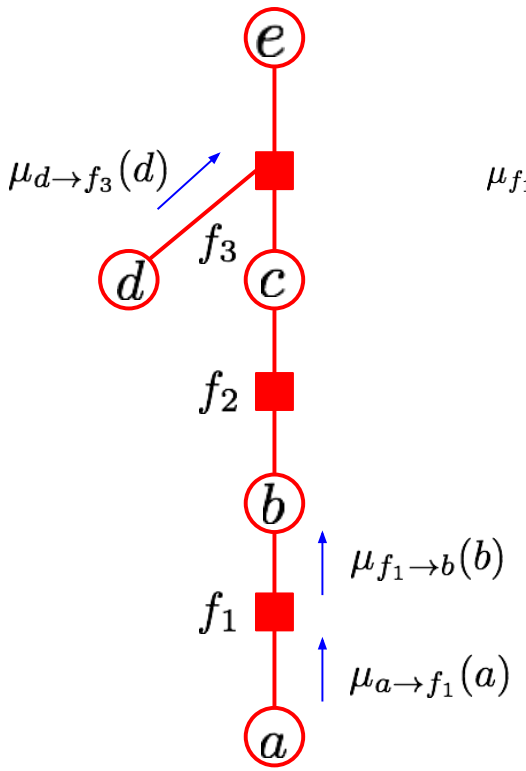
$$f_1(a,b) = \begin{bmatrix} f_1(0,0), f_1(1,0) \\ f_1(0,1), f_1(1,1) \end{bmatrix} = \begin{bmatrix} 1,2 \\ 1,2 \end{bmatrix}$$

$$f_2(b,c) = \begin{bmatrix} 3,2 \\ 1,0.5 \end{bmatrix}$$

$$f_3(c=0,d,e) = \begin{bmatrix} 1,2 \\ 2,1 \end{bmatrix}$$

$$f_3(c=1,d,e) = \begin{bmatrix} 1,2 \\ 2,1 \end{bmatrix}$$

Example of the sum product algorithm



$$\mu_{a \to f_1}(a) = [1,1]^T$$
$$\mu_{d \to f_3}(d) = [1,1]^T$$

$$\mu_{f_1 \to b}(b) = \sum_a f_1(a,b)\mu_{a \to f_1(a)}(a)$$
$$= \begin{bmatrix} 1,2 \\ 1,2 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 3 \\ 3 \end{bmatrix}$$

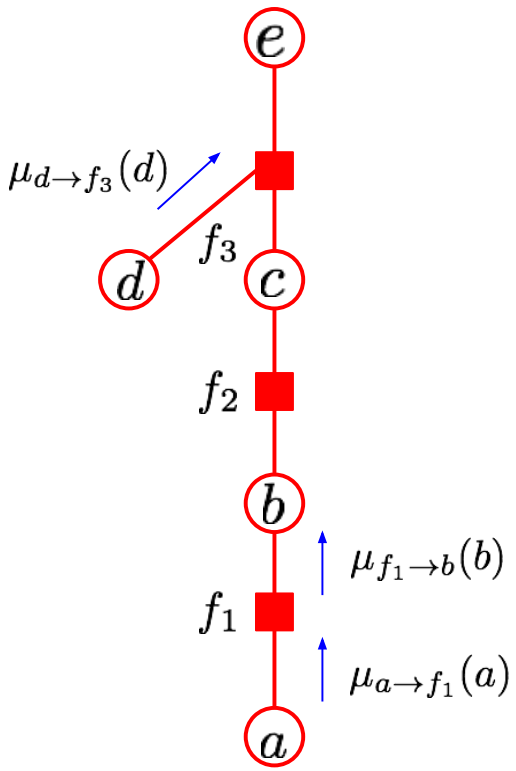
$$f_1(a,b) = \begin{bmatrix} f_1(0,0), f_1(1,0) \\ f_1(0,1), f_1(1,1) \end{bmatrix} = \begin{bmatrix} 1,2 \\ 1,2 \end{bmatrix}$$

$$f_2(b,c) = \begin{bmatrix} 3,2 \\ 1,0.5 \end{bmatrix}$$

$$f_3(c=0,d,e) = \begin{bmatrix} 1,2 \\ 2,1 \end{bmatrix}$$

$$f_3(c=1,d,e) = \begin{bmatrix} 1,2 \\ 2,1 \end{bmatrix}$$

Example of the sum product algorithm



$$\mu_{a\to f_1}(a) = [1,1]^T$$
$$\mu_{d\to f_3}(d) = [1,1]^T$$
$$\mu_{f_1\to b}(b) = [3,3]^T$$

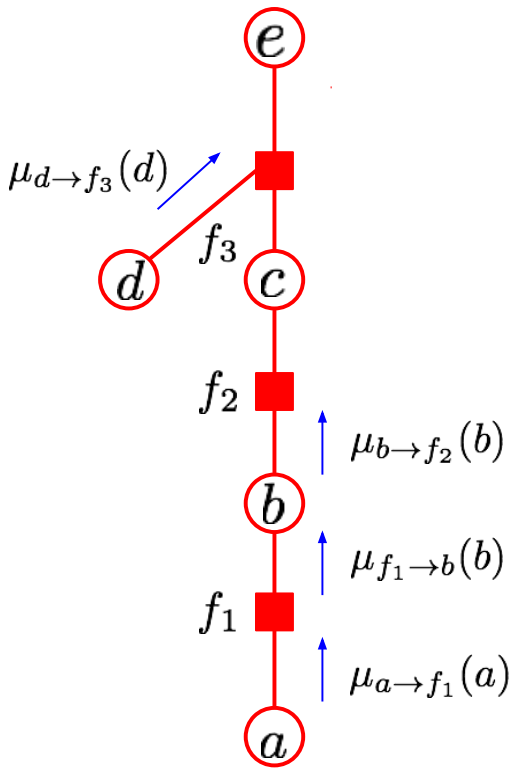
$$f_1(a,b) = \begin{bmatrix} f_1(0,0), f_1(1,0) \\ f_1(0,1), f_1(1,1) \end{bmatrix} = \begin{bmatrix} 1,2 \\ 1,2 \end{bmatrix}$$

$$f_2(b,c) = \begin{bmatrix} 3,2 \\ 1,0.5 \end{bmatrix}$$

$$f_3(c=0,d,e) = \begin{bmatrix} 1,2 \\ 2,1 \end{bmatrix}$$

$$f_3(c=1,d,e) = \begin{bmatrix} 1,2 \\ 2,1 \end{bmatrix}$$

Example of the sum product algorithm



$$\mu_{a\to f_1}(a) = [1,1]^T$$
$$\mu_{d\to f_3}(d) = [1,1]^T$$
$$\mu_{f_1\to b}(b) = [3,3]^T$$

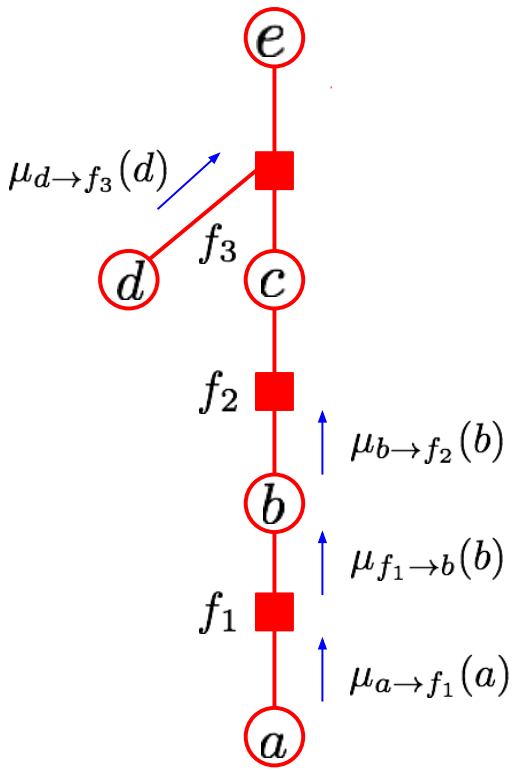
$$f_1(a,b) = \begin{bmatrix} f_1(0,0), f_1(1,0) \\ f_1(0,1), f_1(1,1) \end{bmatrix} = \begin{bmatrix} 1,2 \\ 1,2 \end{bmatrix}$$

$$f_2(b,c) = \begin{bmatrix} 3,2 \\ 1,0.5 \end{bmatrix}$$

$$f_3(c=0,d,e) = \begin{bmatrix} 1,2 \\ 2,1 \end{bmatrix}$$

$$f_3(c=1,d,e) = \begin{bmatrix} 1,2 \\ 2,1 \end{bmatrix}$$

Example of the sum product algorithm



$$\mu_{a\to f_1}(a) = [1,1]^T$$
$$\mu_{d\to f_3}(d) = [1,1]^T$$
$$\mu_{f_1\to b}(b) = [3,3]^T$$

$$\mu_{b\to f2}(b) = \mu_{f_1\to b}(b)$$

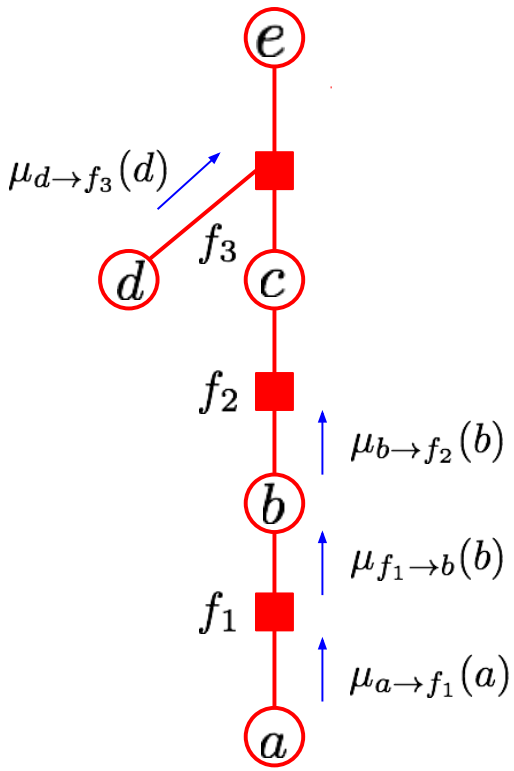
$$f_1(a,b) = \begin{bmatrix} f_1(0,0), f_1(1,0) \\ f_1(0,1), f_1(1,1) \end{bmatrix} = \begin{bmatrix} 1,2 \\ 1,2 \end{bmatrix}$$

$$f_2(b,c) = \begin{bmatrix} 3,2 \\ 1,0.5 \end{bmatrix}$$

$$f_3(c=0,d,e) = \begin{bmatrix} 1,2 \\ 2,1 \end{bmatrix}$$

$$f_3(c=1,d,e) = \begin{bmatrix} 1,2 \\ 2,1 \end{bmatrix}$$

Example of the sum product algorithm



$$\mu_{a\to f_1}(a) = [1,1]^T$$
$$\mu_{d\to f_3}(d) = [1,1]^T$$
$$\mu_{f_1\to b}(b) = [3,3]^T$$

$$\mu_{b\to f2}(b) = \mu_{f_1\to b}(b) = [3,3]^T$$

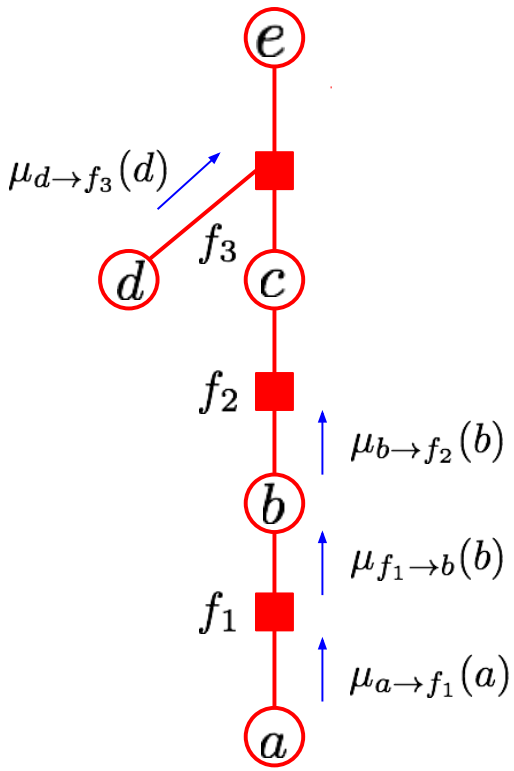
$$f_1(a,b) = \begin{bmatrix} f_1(0,0), f_1(1,0) \\ f_1(0,1), f_1(1,1) \end{bmatrix} = \begin{bmatrix} 1,2 \\ 1,2 \end{bmatrix}$$

$$f_2(b,c) = \begin{bmatrix} 3,2 \\ 1,0.5 \end{bmatrix}$$

$$f_3(c=0,d,e) = \begin{bmatrix} 1,2 \\ 2,1 \end{bmatrix}$$

$$f_3(c=1,d,e) = \begin{bmatrix} 1,2 \\ 2,1 \end{bmatrix}$$

Example of the sum product algorithm



$$\mu_{a\to f_1}(a) = [1,1]^T$$
$$\mu_{d\to f_3}(d) = [1,1]^T$$
$$\mu_{f_1\to b}(b) = [3,3]^T$$
$$\mu_{b\to f2}(b) = [3,3]^T$$

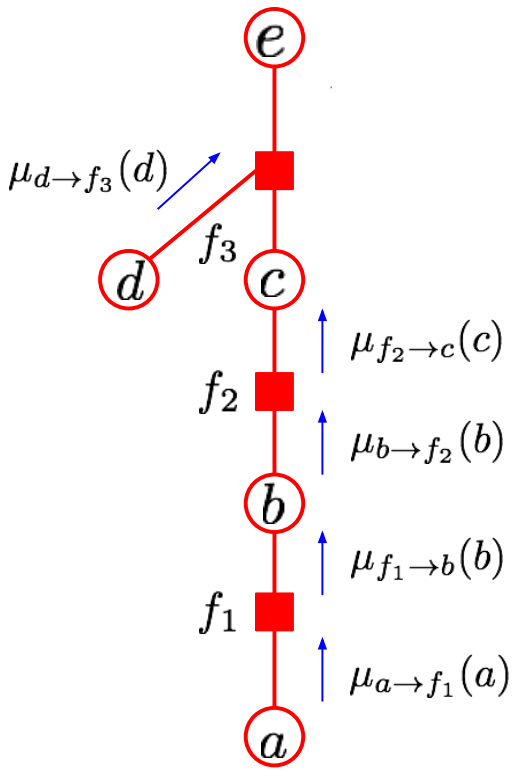
$$f_1(a,b) = \begin{bmatrix} f_1(0,0), f_1(1,0) \\ f_1(0,1), f_1(1,1) \end{bmatrix} = \begin{bmatrix} 1,2 \\ 1,2 \end{bmatrix}$$

$$f_2(b,c) = \begin{bmatrix} 3,2 \\ 1,0.5 \end{bmatrix}$$

$$f_3(c=0,d,e) = \begin{bmatrix} 1,2 \\ 2,1 \end{bmatrix}$$

$$f_3(c=1,d,e) = \begin{bmatrix} 1,2 \\ 2,1 \end{bmatrix}$$

Example of the sum product algorithm



$$\mu_{a\to f_1}(a) = [1,1]^T$$
$$\mu_{d\to f_3}(d) = [1,1]^T$$
$$\mu_{f_1\to b}(b) = [3,3]^T$$
$$\mu_{b\to f2}(b) = [3,3]^T$$

$$\mu_{f_2\to c}(c) = \sum_b f_2(b,c)\mu_{b\to f_2}(b)$$
$$= \begin{bmatrix} 3,2 \\ 1,0.5 \end{bmatrix} \begin{bmatrix} 3 \\ 3 \end{bmatrix} = \begin{bmatrix} 15 \\ 4.5 \end{bmatrix}$$

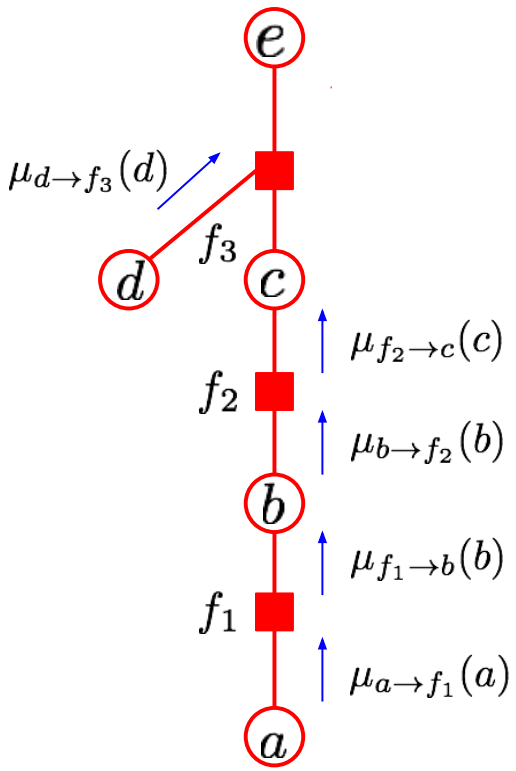
$$f_1(a,b) = \begin{bmatrix} f_1(0,0), f_1(1,0) \\ f_1(0,1), f_1(1,1) \end{bmatrix} = \begin{bmatrix} 1,2 \\ 1,2 \end{bmatrix}$$

$$f_2(b,c) = \begin{bmatrix} 3,2 \\ 1,0.5 \end{bmatrix}$$

$$f_3(c=0,d,e) = \begin{bmatrix} 1,2 \\ 2,1 \end{bmatrix}$$

$$f_3(c=1,d,e) = \begin{bmatrix} 1,2 \\ 2,1 \end{bmatrix}$$

Example of the sum product algorithm



$$\mu_{a\to f_1}(a) = [1,1]^T$$
$$\mu_{d\to f_3}(d) = [1,1]^T$$
$$\mu_{f_1\to b}(b) = [3,3]^T$$
$$\mu_{b\to f2}(b) = [3,3]^T$$
$$\mu_{f_2\to c}(c) = [15,4.5]^T$$

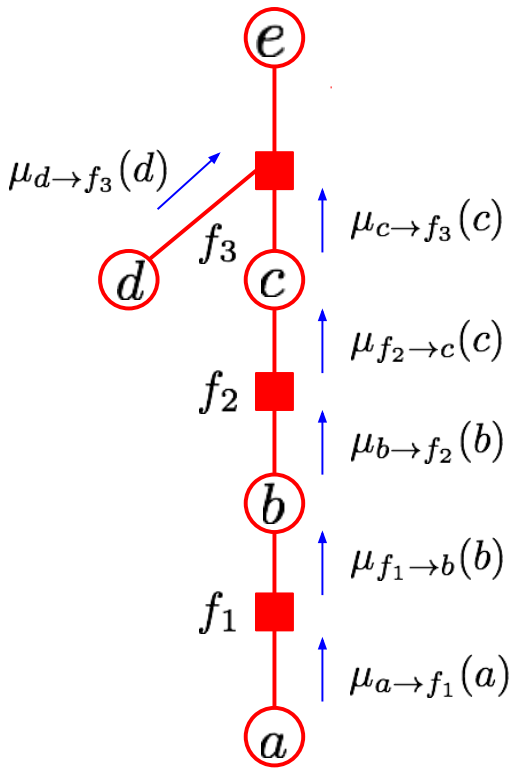
$$f_1(a,b) = \begin{bmatrix} f_1(0,0), f_1(1,0) \\ f_1(0,1), f_1(1,1) \end{bmatrix} = \begin{bmatrix} 1,2 \\ 1,2 \end{bmatrix}$$

$$f_2(b,c) = \begin{bmatrix} 3,2 \\ 1,0.5 \end{bmatrix}$$

$$f_3(c=0,d,e) = \begin{bmatrix} 1,2 \\ 2,1 \end{bmatrix}$$

$$f_3(c=1,d,e) = \begin{bmatrix} 1,2 \\ 2,1 \end{bmatrix}$$

Example of the sum product algorithm



$$\mu_{a\to f_1}(a) = [1,1]^T$$
$$\mu_{d\to f_3}(d) = [1,1]^T$$
$$\mu_{f_1\to b}(b) = [3,3]^T$$
$$\mu_{b\to f2}(b) = [3,3]^T$$
$$\mu_{f_2\to c}(c) = [15,4.5]^T$$
$$\mu_{c\to f_3}(c) = [15,4.5]^T$$

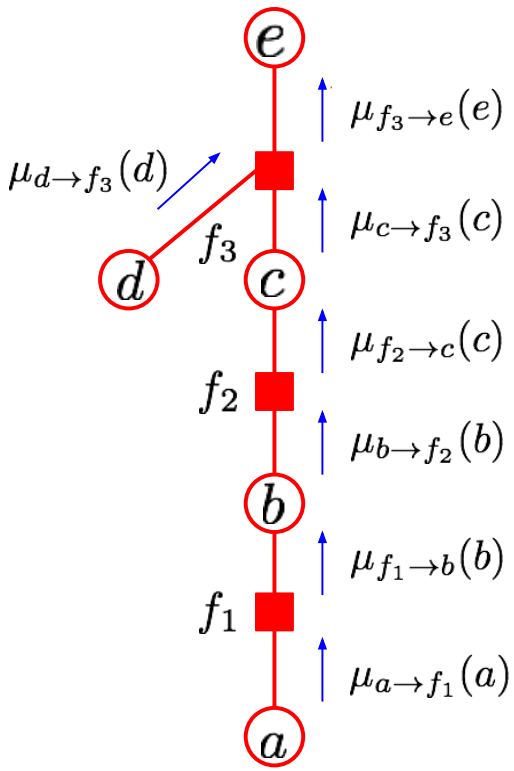
$$f_1(a,b) = \begin{bmatrix} f_1(0,0), f_1(1,0) \\ f_1(0,1), f_1(1,1) \end{bmatrix} = \begin{bmatrix} 1,2 \\ 1,2 \end{bmatrix}$$

$$f_2(b,c) = \begin{bmatrix} 3,2 \\ 1,0.5 \end{bmatrix}$$

$$f_3(c=0,d,e) = \begin{bmatrix} 1,2 \\ 2,1 \end{bmatrix}$$

$$f_3(c=1,d,e) = \begin{bmatrix} 1,2 \\ 2,1 \end{bmatrix}$$

Example of the sum product algorithm



$$\mu_{a \to f_1}(a) = [1,1]^T$$
$$\mu_{d \to f_3}(d) = [1,1]^T$$
$$\mu_{f_1 \to b}(b) = [3,3]^T$$
$$\mu_{b \to f2}(b) = [3,3]^T$$
$$\mu_{f_2 \to c}(c) = [15,4.5]^T$$
$$\mu_{c \to f_3}(c) = [15,4.5]^T$$

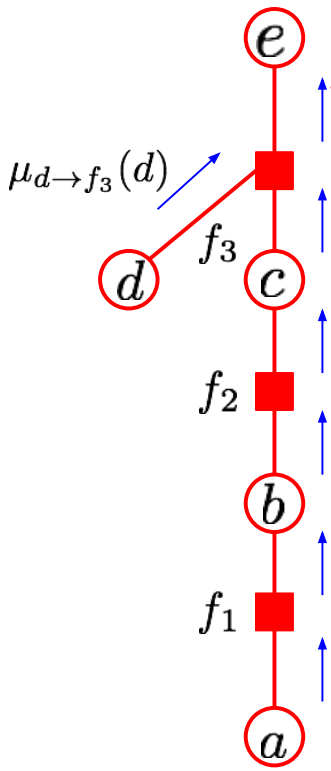
$$f_1(a,b) = \begin{bmatrix} f_1(0,0), f_1(1,0) \\ f_1(0,1), f_1(1,1) \end{bmatrix} = \begin{bmatrix} 1,2 \\ 1,2 \end{bmatrix}$$

$$f_2(b,c) = \begin{bmatrix} 3,2 \\ 1,0.5 \end{bmatrix}$$

$$f_3(c=0,d,e) = \begin{bmatrix} 1,2 \\ 2,1 \end{bmatrix}$$

$$f_3(c=1,d,e) = \begin{bmatrix} 1,2 \\ 2,1 \end{bmatrix}$$

Example of the sum product algorithm



$$\mu_{a\to f_1}(a) = [1,1]^T$$
$$\mu_{d\to f_3}(d) = [1,1]^T$$
$$\mu_{f_1\to b}(b) = [3,3]^T$$
$$\mu_{b\to f2}(b) = [3,3]^T$$
$$\mu_{f_2\to c}(c) = [15,4.5]^T$$
$$\mu_{c\to f_3}(c) = [15,4.5]^T$$

$$\mu_{f_3\to e}(e) = \sum_{c,d} f_3(c,d,e)\mu_{c\to f_3}(c)\mu_{d\to f_3}(d)$$
$$= 15 \begin{bmatrix}1,2\\2,1\end{bmatrix}\begin{bmatrix}1\\1\end{bmatrix} + 4.5 \begin{bmatrix}1,2\\2,1\end{bmatrix}\begin{bmatrix}1\\1\end{bmatrix}$$
$$= \begin{bmatrix}58.5\\58.5\end{bmatrix}$$

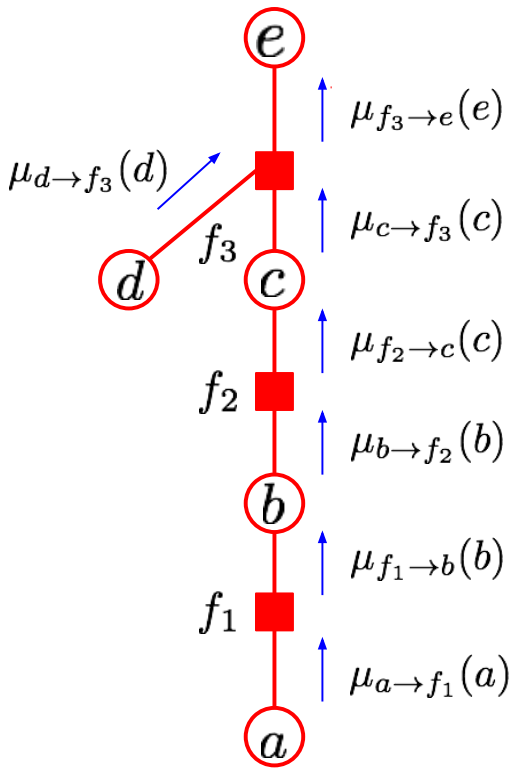
$$f_1(a,b) = \begin{bmatrix}f_1(0,0), f_1(1,0)\\f_1(0,1), f_1(1,1)\end{bmatrix} = \begin{bmatrix}1,2\\1,2\end{bmatrix}$$

$$f_2(b,c) = \begin{bmatrix}3,2\\1,0.5\end{bmatrix}$$

$$f_3(c=0,d,e) = \begin{bmatrix}1,2\\2,1\end{bmatrix}$$

$$f_3(c=1,d,e) = \begin{bmatrix}1,2\\2,1\end{bmatrix}$$

Example of the sum product algorithm



$$\mu_{a\to f_1}(a) = [1,1]^T$$
$$\mu_{d\to f_3}(d) = [1,1]^T$$
$$\mu_{f_1\to b}(b) = [3,3]^T$$
$$\mu_{b\to f2}(b) = [3,3]^T$$
$$\mu_{f_2\to c}(c) = [15,4.5]^T$$
$$\mu_{c\to f_3}(c) = [15,4.5]^T$$
$$\mu_{f_3\to e}(e) = [58.5,58.5]^T$$

$$f_1(a,b) = \begin{bmatrix} f_1(0,0), f_1(1,0) \\ f_1(0,1), f_1(1,1) \end{bmatrix} = \begin{bmatrix} 1,2 \\ 1,2 \end{bmatrix}$$

$$f_2(b,c) = \begin{bmatrix} 3,2 \\ 1,0.5 \end{bmatrix}$$

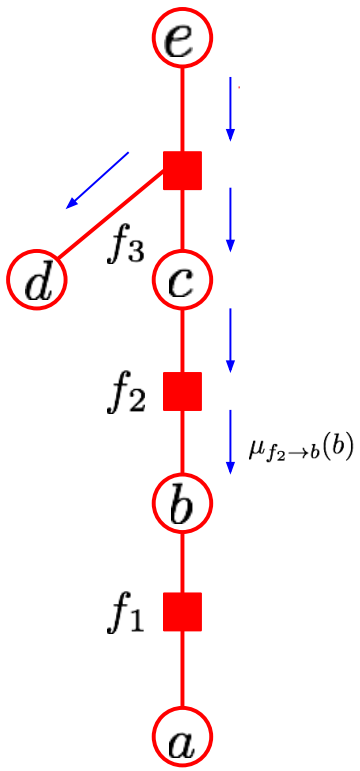
$$f_3(c=0,d,e) = \begin{bmatrix} 1,2 \\ 2,1 \end{bmatrix}$$

$$f_3(c=1,d,e) = \begin{bmatrix} 1,2 \\ 2,1 \end{bmatrix}$$

The sum product algorithm on factor graph

- How to start the sum-product algorithm:
  - Choose a node in the factor graph as the root node
  - Compute all the leaf-to-root messages
  - **Compute all the root-to-leaf messages**
- Initial conditions:
  - Starting from a factor leaf/root node, the initial factor-to-variable message is the factor itself
  - Starting from a variable leaf/root node, the initial variable-to-factor message is a vector of ones

Example of the sum product algorithm



$$\mu_{a\to f_1}(a) = [1,1]^T$$
$$\mu_{d\to f_3}(d) = [1,1]^T$$
$$\mu_{f_1\to b}(b) = [3,3]^T$$
$$\mu_{b\to f2}(b) = [3,3]^T$$
$$\mu_{f_2\to c}(c) = [15,4.5]^T$$
$$\mu_{c\to f_3}(c) = [15,4.5]^T$$
$$\mu_{f_3\to e}(e) = [58.5,58.5]^T$$
$$\mu_{e\to f_3}(e) = [1,1]^T$$
$$\mu_{f_3\to d}(e) = [58.5,58.5]^T$$
$$\mu_{f_3\to c}(c) = [6,6]^T$$
$$\mu_{c\to f_2}(c) = [6,6]^T$$

$$\mu_{f_2\to b}(b) = \begin{bmatrix}3,2\\1,0.5\end{bmatrix}^T \begin{bmatrix}6\\6\end{bmatrix}$$
$$= [24,15]^T$$

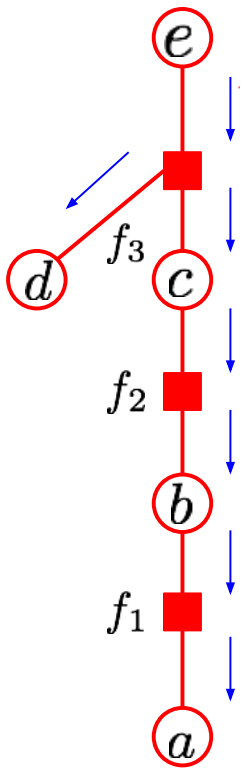
$$f_1(a,b) = \begin{bmatrix}f_1(0,0), f_1(1,0)\\f_1(0,1), f_1(1,1)\end{bmatrix} = \begin{bmatrix}1,2\\1,2\end{bmatrix}$$

$$f_2(b,c) = \begin{bmatrix}3,2\\1,0.5\end{bmatrix}$$

$$f_3(c=0,d,e) = \begin{bmatrix}1,2\\2,1\end{bmatrix}$$

$$f_3(c=1,d,e) = \begin{bmatrix}1,2\\2,1\end{bmatrix}$$

Example of the sum product algorithm



$$\mu_{a\to f_1}(a) = [1,1]^T$$
$$\mu_{d\to f_3}(d) = [1,1]^T$$
$$\mu_{f_1\to b}(b) = [3,3]^T$$
$$\mu_{b\to f2}(b) = [3,3]^T$$
$$\mu_{f_2\to c}(c) = [15,4.5]^T$$
$$\mu_{c\to f_3}(c) = [15,4.5]^T$$
$$\mu_{f_3\to e}(e) = [58.5,58.5]^T$$
$$\mu_{e\to f_3}(e) = [1,1]^T$$
$$\mu_{f_3\to d}(e) = [58.5,58.5]^T$$
$$\mu_{f_3\to c}(c) = [6,6]^T$$
$$\mu_{c\to f_2}(c) = [6,6]^T$$
$$\mu_{f_2\to b}(b) = [24,15]^T$$
$$\mu_{b\to f_1}(b) = [24,15]^T$$
$$\mu_{f_1\to a}(a) = [39,78]^T$$

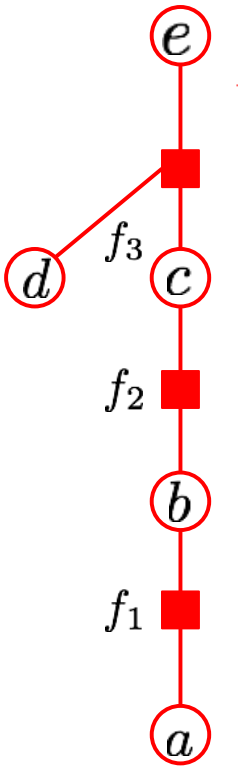
$$f_1(a,b) = \begin{bmatrix} f_1(0,0), f_1(1,0) \\ f_1(0,1), f_1(1,1) \end{bmatrix} = \begin{bmatrix} 1,2 \\ 1,2 \end{bmatrix}$$

$$f_2(b,c) = \begin{bmatrix} 3,2 \\ 1,0.5 \end{bmatrix}$$

$$f_3(c=0,d,e) = \begin{bmatrix} 1,2 \\ 2,1 \end{bmatrix}$$

$$f_3(c=1,d,e) = \begin{bmatrix} 1,2 \\ 2,1 \end{bmatrix}$$

Example of the sum product algorithm



$$\mu_{a\to f_1}(a) = [1,1]^T$$
$$\mu_{d\to f_3}(d) = [1,1]^T$$
$$\mu_{f_1\to b}(b) = [3,3]^T$$
$$\mu_{b\to f2}(b) = [3,3]^T$$
$$\mu_{f_2\to c}(c) = [15,4.5]^T$$
$$\mu_{c\to f_3}(c) = [15,4.5]^T$$
$$\mu_{f_3\to e}(e) = [58.5,58.5]^T$$
$$\mu_{e\to f_3}(e) = [1,1]^T$$
$$\mu_{f_3\to d}(e) = [58.5,58.5]^T$$
$$\mu_{f_3\to c}(c) = [6,6]^T$$
$$\mu_{c\to f_2}(c) = [6,6]^T$$
$$\mu_{f_2\to b}(b) = [24,15]^T$$
$$\mu_{b\to f_1}(b) = [24,15]^T$$
$$\mu_{f_1\to a}(a) = [39,78]^T$$

$$f_1(a,b) = \begin{bmatrix} f_1(0,0), f_1(1,0) \\ f_1(0,1), f_1(1,1) \end{bmatrix} = \begin{bmatrix} 1,2 \\ 1,2 \end{bmatrix}$$

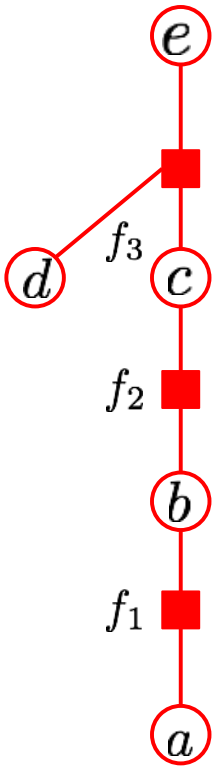
$$f_2(b,c) = \begin{bmatrix} 3,2 \\ 1,0.5 \end{bmatrix}$$

$$f_3(c=0,d,e) = \begin{bmatrix} 1,2 \\ 2,1 \end{bmatrix}$$

$$f_3(c=1,d,e) = \begin{bmatrix} 1,2 \\ 2,1 \end{bmatrix}$$

How do we know we did it right?

Example of the sum product algorithm



We can verify the correctness by computing normalization constant Z

$$Z = \sum_{x_i} g(x_i)$$

$$\mu_{a \to f_1}(a) = [1,1]^T \qquad f_1(a,b) = \begin{bmatrix} f_1(0,0), f_1(1,0) \\ f_1(0,1), f_1(1,1) \end{bmatrix} = \begin{bmatrix} 1,2 \\ 1,2 \end{bmatrix}$$

$$\mu_{d \to f_3}(d) = [1,1]^T$$

$$\mu_{f_1 \to b}(b) = [3,3]^T$$

$$\mu_{b \to f2}(b) = [3,3]^T \qquad f_2(b,c) = \begin{bmatrix} 3,2 \\ 1,0.5 \end{bmatrix}$$

$$\mu_{f_2 \to c}(c) = [15,4.5]^T$$

$$\mu_{c \to f_3}(c) = [15,4.5]^T \qquad f_3(c=0,d,e) = \begin{bmatrix} 1,2 \\ 2,1 \end{bmatrix}$$

$$\mu_{f_3 \to e}(e) = [58.5,58.5]^T$$

$$\mu_{e \to f_3}(e) = [1,1]^T \qquad f_3(c=1,d,e) = \begin{bmatrix} 1,2 \\ 2,1 \end{bmatrix}$$

$$\mu_{f_3 \to d}(e) = [58.5,58.5]^T$$

$$\mu_{f_3 \to c}(c) = [6,6]^T$$

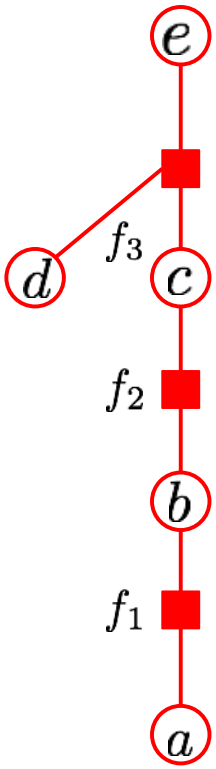
$$\mu_{c \to f_2}(c) = [6,6]^T$$

$$\mu_{f_2 \to b}(b) = [24,15]^T$$

$$\mu_{b \to f_1}(b) = [24,15]^T$$

$$\mu_{f_1 \to a}(a) = [39,78]^T$$

Example of the sum product algorithm



We can verify the correctness by computing normalization constant Z

$$Z = \sum_{x_i} g(x_i)$$

$$Z = \sum_{e} g(e) = \sum_{e} \mu_{f_3 \to e}(e) = 117$$

$$\mu_{a \to f_1}(a) = [1,1]^T$$
$$\mu_{d \to f_3}(d) = [1,1]^T$$
$$\mu_{f_1 \to b}(b) = [3,3]^T$$
$$\mu_{b \to f2}(b) = [3,3]^T$$
$$\mu_{f_2 \to c}(c) = [15, 4.5]^T$$
$$\mu_{c \to f_3}(c) = [15, 4.5]^T$$
$$\mu_{f_3 \to e}(e) = [58.5, 58.5]^T$$
$$\mu_{e \to f_3}(e) = [1,1]^T$$
$$\mu_{f_3 \to d}(e) = [58.5, 58.5]^T$$
$$\mu_{f_3 \to c}(c) = [6,6]^T$$
$$\mu_{c \to f_2}(c) = [6,6]^T$$
$$\mu_{f_2 \to b}(b) = [24, 15]^T$$
$$\mu_{b \to f_1}(b) = [24, 15]^T$$
$$\mu_{f_1 \to a}(a) = [39, 78]^T$$

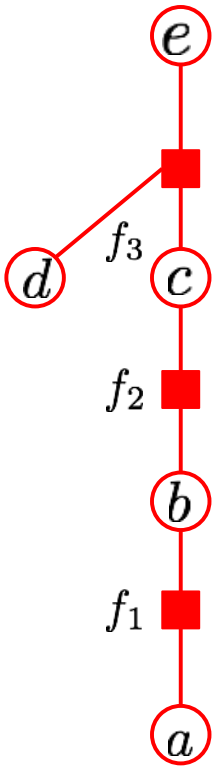
$$f_1(a,b) = \begin{bmatrix} f_1(0,0), f_1(1,0) \\ f_1(0,1), f_1(1,1) \end{bmatrix} = \begin{bmatrix} 1,2 \\ 1,2 \end{bmatrix}$$

$$f_2(b,c) = \begin{bmatrix} 3,2 \\ 1,0.5 \end{bmatrix}$$

$$f_3(c=0,d,e) = \begin{bmatrix} 1,2 \\ 2,1 \end{bmatrix}$$

$$f_3(c=1,d,e) = \begin{bmatrix} 1,2 \\ 2,1 \end{bmatrix}$$

Example of the sum product algorithm



We can verify the correctness by computing normalization constant Z

$$Z = \sum_{x_i} g(x_i)$$

$$Z = \sum_{e} g(e) = \sum_{e} \mu_{f_3 \to e}(e) = 117$$

$$Z = \sum_{b} g(b)$$

$$= \sum_{b} \mu_{f_1 \to b}(b) \mu_{f_2 \to b}(b)$$

$$= 39 \times 3 = 117$$

$\mu_{a \to f_1}(a) = [1,1]^T$

$\mu_{d \to f_3}(d) = [1,1]^T$

$\mu_{f_1 \to b}(b) = [3,3]^T$

$\mu_{b \to f2}(b) = [3,3]^T$

$\mu_{f_2 \to c}(c) = [15, 4.5]^T$

$\mu_{c \to f_3}(c) = [15, 4.5]^T$

$\mu_{f_3 \to e}(e) = [58.5, 58.5]^T$

$\mu_{e \to f_3}(e) = [1,1]^T$

$\mu_{f_3 \to d}(e) = [58.5, 58.5]^T$

$\mu_{f_3 \to c}(c) = [6,6]^T$

$\mu_{c \to f_2}(c) = [6,6]^T$

$\mu_{f_2 \to b}(b) = [24, 15]^T$

$\mu_{b \to f_1}(b) = [24, 15]^T$

$\mu_{f_1 \to a}(a) = [39, 78]^T$

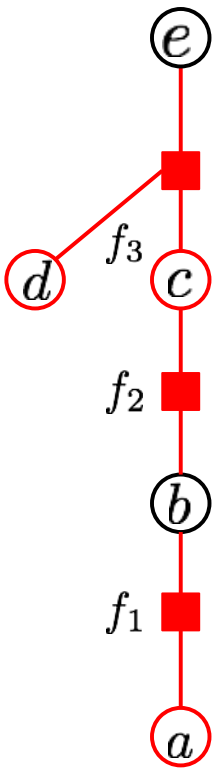
$$f_1(a,b) = \begin{bmatrix} f_1(0,0), f_1(1,0) \\ f_1(0,1), f_1(1,1) \end{bmatrix} = \begin{bmatrix} 1,2 \\ 1,2 \end{bmatrix}$$

$$f_2(b,c) = \begin{bmatrix} 3,2 \\ 1,0.5 \end{bmatrix}$$

$$f_3(c=0,d,e) = \begin{bmatrix} 1,2 \\ 2,1 \end{bmatrix}$$

$$f_3(c=1,d,e) = \begin{bmatrix} 1,2 \\ 2,1 \end{bmatrix}$$

Example of the sum product algorithm



Naive computation:

$$Z = \sum_{a,b,c,d,e} P(a,b,c,d,e)$$

$$\mu_{a \to f_1}(a) = [1,1]^T \qquad f_1(a,b) = \begin{bmatrix} f_1(0,0), f_1(1,0) \\ f_1(0,1), f_1(1,1) \end{bmatrix} = \begin{bmatrix} 1,2 \\ 1,2 \end{bmatrix}$$

$$\mu_{d \to f_3}(d) = [1,1]^T$$

$$\mu_{f_1 \to b}(b) = [3,3]^T$$

$$\mu_{b \to f2}(b) = [3,3]^T \qquad f_2(b,c) = \begin{bmatrix} 3,2 \\ 1,0.5 \end{bmatrix}$$

$$\mu_{f_2 \to c}(c) = [15,4.5]^T$$

$$\mu_{c \to f_3}(c) = [15,4.5]^T \qquad f_3(c=0,d,e) = \begin{bmatrix} 1,2 \\ 2,1 \end{bmatrix}$$

$$\mu_{f_3 \to e}(e) = [58.5,58.5]^T$$

$$\mu_{e \to f_3}(e) = [1,1]^T \qquad f_3(c=1,d,e) = \begin{bmatrix} 1,2 \\ 2,1 \end{bmatrix}$$

$$\mu_{f_3 \to d}(e) = [58.5,58.5]^T$$

$$\mu_{f_3 \to c}(c) = [6,6]^T$$

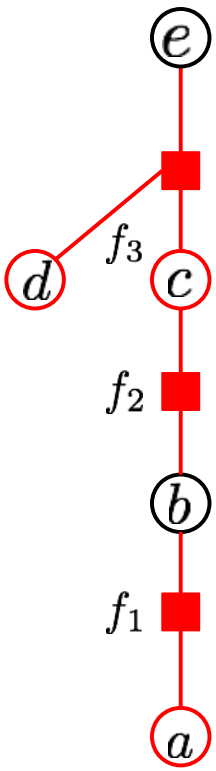
$$\mu_{c \to f_2}(c) = [6,6]^T$$

$$\mu_{f_2 \to b}(b) = [24,15]^T$$

$$\mu_{b \to f_1}(b) = [24,15]^T$$

$$\mu_{f_1 \to a}(a) = [39,78]^T$$

Example of the sum product algorithm



Naive computation:

$$Z = \sum_{a,b,c,d,e} P(a,b,c,d,e)$$

O(2^5)

$$\mu_{a\to f_1}(a) = [1,1]^T \qquad f_1(a,b) = \begin{bmatrix} f_1(0,0), f_1(1,0) \\ f_1(0,1), f_1(1,1) \end{bmatrix} = \begin{bmatrix} 1,2 \\ 1,2 \end{bmatrix}$$

$$\mu_{d\to f_3}(d) = [1,1]^T$$

$$\mu_{f_1\to b}(b) = [3,3]^T$$

$$\mu_{b\to f2}(b) = [3,3]^T \qquad f_2(b,c) = \begin{bmatrix} 3,2 \\ 1,0.5 \end{bmatrix}$$

$$\mu_{f_2\to c}(c) = [15,4.5]^T$$

$$\mu_{c\to f_3}(c) = [15,4.5]^T \qquad f_3(c=0,d,e) = \begin{bmatrix} 1,2 \\ 2,1 \end{bmatrix}$$

$$\mu_{f_3\to e}(e) = [58.5,58.5]^T$$

$$\mu_{e\to f_3}(e) = [1,1]^T \qquad f_3(c=1,d,e) = \begin{bmatrix} 1,2 \\ 2,1 \end{bmatrix}$$

$$\mu_{f_3\to d}(e) = [58.5,58.5]^T$$

$$\mu_{f_3\to c}(c) = [6,6]^T$$

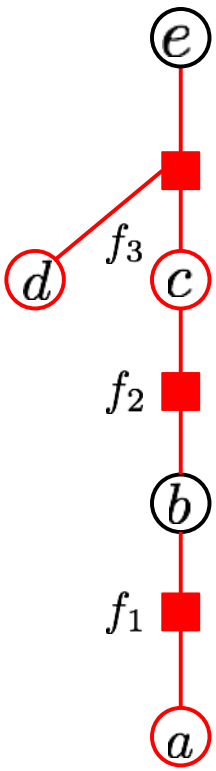
$$\mu_{c\to f_2}(c) = [6,6]^T$$

$$\mu_{f_2\to b}(b) = [24,15]^T$$

$$\mu_{b\to f_1}(b) = [24,15]^T$$

$$\mu_{f_1\to a}(a) = [39,78]^T$$

Example of the sum product algorithm



Naive computation:

$$Z = \sum_{a,b,c,d,e} P(a,b,c,d,e)$$

O(2^5)

Message-passing:

$$Z = \sum_{e} g(e)$$

O(2^3)
Dominated by the largest factor

$$\mu_{a \to f_1}(a) = [1,1]^T$$
$$\mu_{d \to f_3}(d) = [1,1]^T$$
$$\mu_{f_1 \to b}(b) = [3,3]^T$$
$$\mu_{b \to f2}(b) = [3,3]^T$$
$$\mu_{f_2 \to c}(c) = [15, 4.5]^T$$
$$\mu_{c \to f_3}(c) = [15, 4.5]^T$$
$$\mu_{f_3 \to e}(e) = [58.5, 58.5]^T$$
$$\mu_{e \to f_3}(e) = [1,1]^T$$
$$\mu_{f_3 \to d}(e) = [58.5, 58.5]^T$$
$$\mu_{f_3 \to c}(c) = [6,6]^T$$
$$\mu_{c \to f_2}(c) = [6,6]^T$$
$$\mu_{f_2 \to b}(b) = [24, 15]^T$$
$$\mu_{b \to f_1}(b) = [24, 15]^T$$
$$\mu_{f_1 \to a}(a) = [39, 78]^T$$

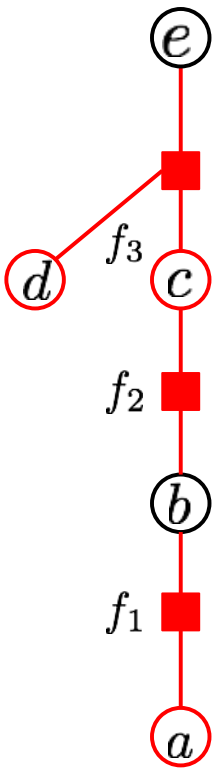
$$f_1(a,b) = \begin{bmatrix} f_1(0,0), f_1(1,0) \\ f_1(0,1), f_1(1,1) \end{bmatrix} = \begin{bmatrix} 1,2 \\ 1,2 \end{bmatrix}$$

$$f_2(b,c) = \begin{bmatrix} 3,2 \\ 1,0.5 \end{bmatrix}$$

$$f_3(c=0,d,e) = \begin{bmatrix} 1,2 \\ 2,1 \end{bmatrix}$$

$$f_3(c=1,d,e) = \begin{bmatrix} 1,2 \\ 2,1 \end{bmatrix}$$

Example of the sum product algorithm



Naive computation:

$$Z = \sum_{a,b,c,d,e} P(a,b,c,d,e)$$

O(2^5)

Message-passing:

$$Z = \sum_{e} g(e)$$

O(2^3)
Dominated by the largest factor

$$\mu_{a \to f_1}(a) = [1,1]^T$$
$$\mu_{d \to f_3}(d) = [1,1]^T$$
$$\mu_{f_1 \to b}(b) = [3,3]^T$$
$$\mu_{b \to f2}(b) = [3,3]^T$$
$$\mu_{f_2 \to c}(c) = [15,4.5]^T$$
$$\mu_{c \to f_3}(c) = [15,4.5]^T$$
$$\mu_{f_3 \to e}(e) = [58.5,58.5]^T$$
$$\mu_{e \to f_3}(e) = [1,1]^T$$
$$\mu_{f_3 \to d}(e) = [58.5,58.5]^T$$
$$\mu_{f_3 \to c}(c) = [6,6]^T$$
$$\mu_{c \to f_2}(c) = [6,6]^T$$
$$\mu_{f_2 \to b}(b) = [24,15]^T$$
$$\mu_{b \to f_1}(b) = [24,15]^T$$
$$\mu_{f_1 \to a}(a) = [39,78]^T$$

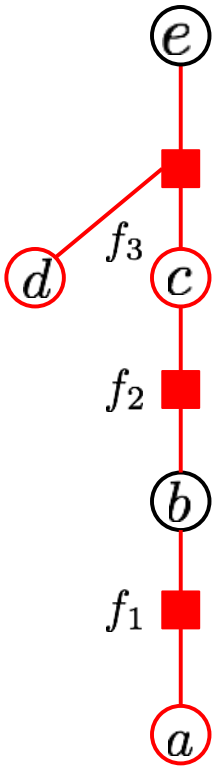
$$f_1(a,b) = \begin{bmatrix} f_1(0,0), f_1(1,0) \\ f_1(0,1), f_1(1,1) \end{bmatrix} = \begin{bmatrix} 1,2 \\ 1,2 \end{bmatrix}$$

$$f_2(b,c) = \begin{bmatrix} 3,2 \\ 1,0.5 \end{bmatrix}$$

$$f_3(c=0,d,e) = \begin{bmatrix} 1,2 \\ 2,1 \end{bmatrix}$$

$$f_3(c=1,d,e) = \begin{bmatrix} 1,2 \\ 2,1 \end{bmatrix}$$

Example of the sum product algorithm



How do we compute P(e, b)?

$$\mu_{a\to f_1}(a) = [1,1]^T \qquad f_1(a,b) = \begin{bmatrix} f_1(0,0), f_1(1,0) \\ f_1(0,1), f_1(1,1) \end{bmatrix} = \begin{bmatrix} 1,2 \\ 1,2 \end{bmatrix}$$

$$\mu_{d\to f_3}(d) = [1,1]^T$$

$$\mu_{f_1\to b}(b) = [3,3]^T$$

$$\mu_{b\to f2}(b) = [3,3]^T \qquad f_2(b,c) = \begin{bmatrix} 3,2 \\ 1,0.5 \end{bmatrix}$$

$$\mu_{f_2\to c}(c) = [15,4.5]^T$$

$$\mu_{c\to f_3}(c) = [15,4.5]^T \qquad f_3(c=0,d,e) = \begin{bmatrix} 1,2 \\ 2,1 \end{bmatrix}$$

$$\mu_{f_3\to e}(e) = [58.5,58.5]^T$$

$$\mu_{e\to f_3}(e) = [1,1]^T \qquad f_3(c=1,d,e) = \begin{bmatrix} 1,2 \\ 2,1 \end{bmatrix}$$

$$\mu_{f_3\to d}(e) = [58.5,58.5]^T$$

$$\mu_{f_3\to c}(c) = [6,6]^T$$

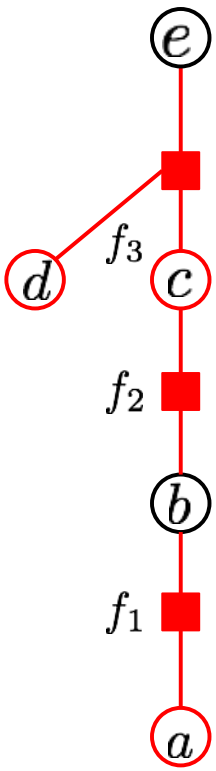
$$\mu_{c\to f_2}(c) = [6,6]^T$$

$$\mu_{f_2\to b}(b) = [24,15]^T$$

$$\mu_{b\to f_1}(b) = [24,15]^T$$

$$\mu_{f_1\to a}(a) = [39,78]^T$$

Example of the sum product algorithm



How do we compute P(e, b)?

We can eliminate the summation over b or e during message-passing

$$\mu_{a\to f_1}(a) = [1,1]^T \qquad f_1(a,b) = \begin{bmatrix} f_1(0,0), f_1(1,0) \\ f_1(0,1), f_1(1,1) \end{bmatrix} = \begin{bmatrix} 1,2 \\ 1,2 \end{bmatrix}$$

$$\mu_{d\to f_3}(d) = [1,1]^T$$

$$\mu_{f_1\to b}(b) = [3,3]^T$$

$$\mu_{b\to f2}(b) = [3,3]^T \qquad f_2(b,c) = \begin{bmatrix} 3,2 \\ 1,0.5 \end{bmatrix}$$

$$\mu_{f_2\to c}(c) = [15,4.5]^T$$

$$\mu_{c\to f_3}(c) = [15,4.5]^T \qquad f_3(c=0,d,e) = \begin{bmatrix} 1,2 \\ 2,1 \end{bmatrix}$$

$$\mu_{f_3\to e}(e) = [58.5,58.5]^T$$

$$\mu_{e\to f_3}(e) = [1,1]^T \qquad f_3(c=1,d,e) = \begin{bmatrix} 1,2 \\ 2,1 \end{bmatrix}$$

$$\mu_{f_3\to d}(e) = [58.5,58.5]^T$$

$$\mu_{f_3\to c}(c) = [6,6]^T$$

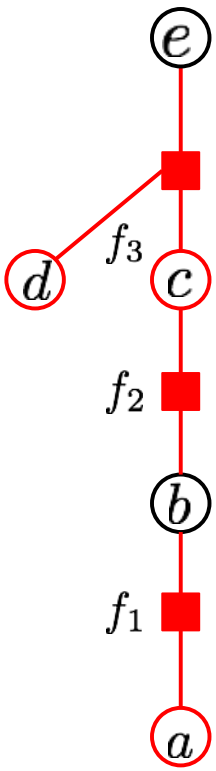
$$\mu_{c\to f_2}(c) = [6,6]^T$$

$$\mu_{f_2\to b}(b) = [24,15]^T$$

$$\mu_{b\to f_1}(b) = [24,15]^T$$

$$\mu_{f_1\to a}(a) = [39,78]^T$$

Example of the sum product algorithm



How do we compute P(e, b)?

We can eliminate the summation over b or e during message-passing

Assume choose e as an anchor and eliminate summation over b:

$$g(b,e) = \mu_{f_3 \to e}(b,e)$$

Or we can choose b as an anchor and eliminate summation over c:

$$g(b,e) = \mu_{f_2 \to b}(b,e)\mu_{f_1 \to b}(b)$$

$$\mu_{a \to f_1}(a) = [1,1]^T$$
$$\mu_{d \to f_3}(d) = [1,1]^T$$
$$\mu_{f_1 \to b}(b) = [3,3]^T$$
$$\mu_{b \to f2}(b) = [3,3]^T$$
$$\mu_{f_2 \to c}(c) = [15, 4.5]^T$$
$$\mu_{c \to f_3}(c) = [15, 4.5]^T$$
$$\mu_{f_3 \to e}(e) = [58.5, 58.5]^T$$
$$\mu_{e \to f_3}(e) = [1,1]^T$$
$$\mu_{f_3 \to d}(e) = [58.5, 58.5]^T$$
$$\mu_{f_3 \to c}(c) = [6,6]^T$$
$$\mu_{c \to f_2}(c) = [6,6]^T$$
$$\mu_{f_2 \to b}(b) = [24, 15]^T$$
$$\mu_{b \to f_1}(b) = [24, 15]^T$$
$$\mu_{f_1 \to a}(a) = [39, 78]^T$$

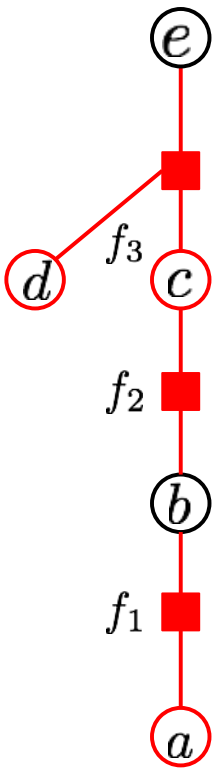
$$f_1(a,b) = \begin{bmatrix} f_1(0,0), f_1(1,0) \\ f_1(0,1), f_1(1,1) \end{bmatrix} = \begin{bmatrix} 1,2 \\ 1,2 \end{bmatrix}$$

$$f_2(b,c) = \begin{bmatrix} 3,2 \\ 1,0.5 \end{bmatrix}$$

$$f_3(c=0,d,e) = \begin{bmatrix} 1,2 \\ 2,1 \end{bmatrix}$$

$$f_3(c=1,d,e) = \begin{bmatrix} 1,2 \\ 2,1 \end{bmatrix}$$

Example of the sum product algorithm



How do we compute P(e, b)?

Assume choose e as an anchor and eliminate summation over b:

$$g(b,e) = \mu_{f_3 \to e}(b,e)$$

Need to re-compute the messages to carry b along

$$\mu_{a \to f_1}(a) = [1,1]^T \qquad f_1(a,b) = \begin{bmatrix} f_1(0,0), f_1(1,0) \\ f_1(0,1), f_1(1,1) \end{bmatrix} = \begin{bmatrix} 1,2 \\ 1,2 \end{bmatrix}$$

$$\mu_{d \to f_3}(d) = [1,1]^T$$

$$\mu_{f_1 \to b}(b) = [3,3]^T$$

$$\mu_{b \to f2}(b) = [3,3]^T \qquad f_2(b,c) = \begin{bmatrix} 3,2 \\ 1,0.5 \end{bmatrix}$$

$$\mu_{f_2 \to c}(c) = [15,4.5]^T$$

$$\mu_{c \to f_3}(c) = [15,4.5]^T \qquad f_3(c=0,d,e) = \begin{bmatrix} 1,2 \\ 2,1 \end{bmatrix}$$

$$\mu_{f_3 \to e}(e) = [58.5,58.5]^T$$

$$\mu_{e \to f_3}(e) = [1,1]^T \qquad f_3(c=1,d,e) = \begin{bmatrix} 1,2 \\ 2,1 \end{bmatrix}$$

$$\mu_{f_3 \to d}(e) = [58.5,58.5]^T$$

$$\mu_{f_3 \to c}(c) = [6,6]^T$$

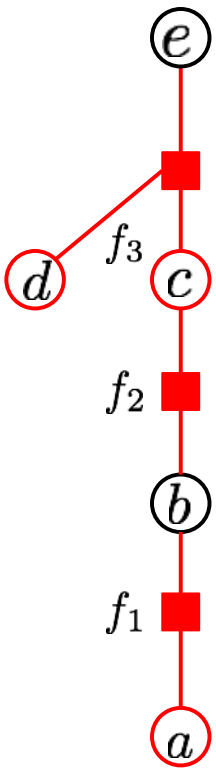
$$\mu_{c \to f_2}(c) = [6,6]^T$$

$$\mu_{f_2 \to b}(b) = [24,15]^T$$

$$\mu_{b \to f_1}(b) = [24,15]^T$$

$$\mu_{f_1 \to a}(a) = [39,78]^T$$

Example of the sum product algorithm



How do we compute P(e, b)?

Assume choose e as an anchor and eliminate summation over b:

$$g(b,e) = \mu_{f_3 \to e}(b,e)$$

Need to re-compute the messages to carry b along

$$\mu_{a \to f_1}(a) = [1,1]^T$$
$$\mu_{d \to f_3}(d) = [1,1]^T$$
$$\mu_{f_1 \to b}(b) = [3,3]^T$$
$$\mu_{b \to f2}(b) = [3,3]^T$$
$$\mu_{f_2 \to c}(c) = [15, 4.5]^T$$
$$\mu_{c \to f_3}(c) = [15, 4.5]^T$$
$$\mu_{f_3 \to e}(e) = [58.5, 58.5]^T$$
$$\mu_{e \to f_3}(e) = [1,1]^T$$
$$\mu_{f_3 \to d}(e) = [58.5, 58.5]^T$$
$$\mu_{f_3 \to c}(c) = [6,6]^T$$
$$\mu_{c \to f_2}(c) = [6,6]^T$$
$$\mu_{f_2 \to b}(b) = [24, 15]^T$$
$$\mu_{b \to f_1}(b) = [24, 15]^T$$
$$\mu_{f_1 \to a}(a) = [39, 78]^T$$

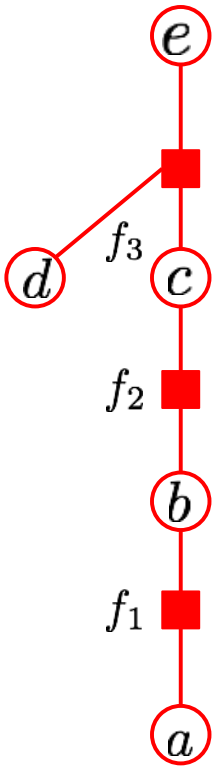
$$f_1(a,b) = \begin{bmatrix} f_1(0,0), f_1(1,0) \\ f_1(0,1), f_1(1,1) \end{bmatrix} = \begin{bmatrix} 1,2 \\ 1,2 \end{bmatrix}$$

$$f_2(b,c) = \begin{bmatrix} 3,2 \\ 1,0.5 \end{bmatrix}$$

$$f_3(c=0,d,e) = \begin{bmatrix} 1,2 \\ 2,1 \end{bmatrix}$$

$$f_3(c=1,d,e) = \begin{bmatrix} 1,2 \\ 2,1 \end{bmatrix}$$

Example of the sum product algorithm



How do we compute P(e, b)?

New messages carrying b:

$$\mu_{a\to f_1}(a) = [1,1]^T \qquad f_1(a,b) = \begin{bmatrix} f_1(0,0), f_1(1,0) \\ f_1(0,1), f_1(1,1) \end{bmatrix} = \begin{bmatrix} 1,2 \\ 1,2 \end{bmatrix}$$

$$\mu_{d\to f_3}(d) = [1,1]^T$$

$$\mu_{f_1\to b}(b) = [3,3]^T$$

$$\mu_{b\to f2}(b) = [3,3]^T \qquad f_2(b,c) = \begin{bmatrix} 3,2 \\ 1,0.5 \end{bmatrix}$$

$$\mu_{f_2\to c}(b,c) \quad \mu_{f_2\to c}(c) = [15,4.5]^T$$

$$\mu_{c\to f_3}(b,c) \quad \mu_{c\to f_3}(c) = [15,4.5]^T \qquad f_3(c=0,d,e) = \begin{bmatrix} 1,2 \\ 2,1 \end{bmatrix}$$

$$\mu_{f_3\to e}(b,e) \quad \mu_{f_3\to e}(e) = [58.5,58.5]^T$$

$$\mu_{e\to f_3}(e) = [1,1]^T \qquad f_3(c=1,d,e) = \begin{bmatrix} 1,2 \\ 2,1 \end{bmatrix}$$

$$\mu_{f_3\to d}(e) = [58.5,58.5]^T$$

$$\mu_{f_3\to c}(c) = [6,6]^T$$

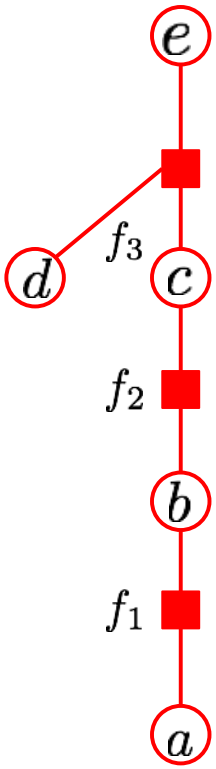
$$\mu_{c\to f_2}(c) = [6,6]^T$$

$$\mu_{f_2\to b}(b) = [24,15]^T$$

$$\mu_{b\to f_1}(b) = [24,15]^T$$

$$\mu_{f_1\to a}(a) = [39,78]^T$$

Example of the sum product algorithm



How do we compute P(e, b)?

New messages carrying b:

$$g(b,e) = \mu_{f_3 \to e}(b,e)$$

$$P(b,e) = \frac{1}{Z}\mu_{f_3 \to e}(b,e)$$

$$\mu_{a \to f_1}(a) = [1,1]^T \qquad f_1(a,b) = \begin{bmatrix} f_1(0,0), f_1(1,0) \\ f_1(0,1), f_1(1,1) \end{bmatrix} = \begin{bmatrix} 1,2 \\ 1,2 \end{bmatrix}$$

$$\mu_{d \to f_3}(d) = [1,1]^T$$

$$\mu_{f_1 \to b}(b) = [3,3]^T$$

$$\mu_{b \to f2}(b) = [3,3]^T \qquad f_2(b,c) = \begin{bmatrix} 3,2 \\ 1,0.5 \end{bmatrix}$$

$$\mu_{f_2 \to c}(b,c) \quad \mu_{f_2 \to c}(c) = [15,4.5]^T$$

$$\mu_{c \to f_3}(b,c) \quad \mu_{c \to f_3}(c) = [15,4.5]^T \qquad f_3(c=0,d,e) = \begin{bmatrix} 1,2 \\ 2,1 \end{bmatrix}$$

$$\mu_{f_3 \to e}(b,e) \quad \mu_{f_3 \to e}(e) = [58.5,58.5]^T$$

$$\mu_{e \to f_3}(e) = [1,1]^T \qquad f_3(c=1,d,e) = \begin{bmatrix} 1,2 \\ 2,1 \end{bmatrix}$$

$$\mu_{f_3 \to d}(e) = [58.5,58.5]^T$$

$$\mu_{f_3 \to c}(c) = [6,6]^T$$

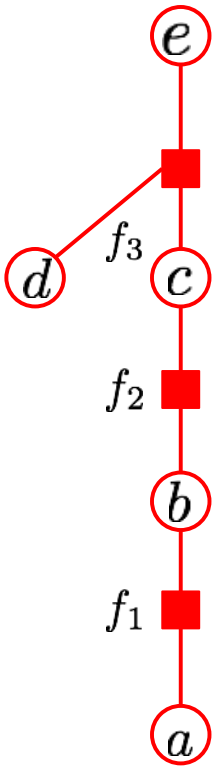
$$\mu_{c \to f_2}(c) = [6,6]^T$$

$$\mu_{f_2 \to b}(b) = [24,15]^T$$

$$\mu_{b \to f_1}(b) = [24,15]^T$$

$$\mu_{f_1 \to a}(a) = [39,78]^T$$

Example of the sum product algorithm



How do we compute P(e, b)?

New messages carrying b:

$$\mu_{f_2\to c}(b,c) = f_2(b,c)\mu_{b\to f_2}(b) = \begin{bmatrix} 3\times 3 & 2\times 3 \\ 1\times 3 & 0.5\times 3 \end{bmatrix} = \begin{bmatrix} 9 & 6 \\ 3 & 1.5 \end{bmatrix}$$

$$\mu_{c\to f_3}(b,c) = \mu_{f_2\to c}(b,c) = \begin{bmatrix} 9 & 6 \\ 3 & 1.5 \end{bmatrix}$$

$$\mu_{a\to f_1}(a) = [1,1]^T \qquad f_1(a,b) = \begin{bmatrix} f_1(0,0), f_1(1,0) \\ f_1(0,1), f_1(1,1) \end{bmatrix} = \begin{bmatrix} 1,2 \\ 1,2 \end{bmatrix}$$

$$\mu_{d\to f_3}(d) = [1,1]^T$$

$$\mu_{f_1\to b}(b) = [3,3]^T$$

$$\mu_{b\to f2}(b) = [3,3]^T \qquad f_2(b,c) = \begin{bmatrix} 3,2 \\ 1,0.5 \end{bmatrix}$$

$$\mu_{f_2\to c}(b,c) \quad \mu_{f_2\to c}(c) = [15,4.5]^T$$

$$\mu_{c\to f_3}(b,c) \quad \mu_{c\to f_3}(c) = [15,4.5]^T \qquad f_3(c=0,d,e) = \begin{bmatrix} 1,2 \\ 2,1 \end{bmatrix}$$

$$\mu_{f_3\to e}(b,e) \quad \mu_{f_3\to e}(e) = [58.5,58.5]^T$$

$$\mu_{e\to f_3}(e) = [1,1]^T \qquad f_3(c=1,d,e) = \begin{bmatrix} 1,2 \\ 2,1 \end{bmatrix}$$

$$\mu_{f_3\to d}(e) = [58.5,58.5]^T$$

$$\mu_{f_3\to c}(c) = [6,6]^T$$

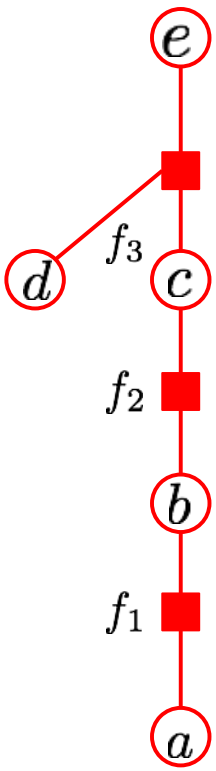
$$\mu_{c\to f_2}(c) = [6,6]^T$$

$$\mu_{f_2\to b}(b) = [24,15]^T$$

$$\mu_{b\to f_1}(b) = [24,15]^T$$

$$\mu_{f_1\to a}(a) = [39,78]^T$$

Example of the sum product algorithm



How do we compute P(e, b)?

New messages carrying b:

$$\mu_{c\to f_3}(b,c) = \mu_{f_2\to 3}(b,c) = \begin{bmatrix} 9 & 6 \\ 3 & 1.5 \end{bmatrix}$$

$$\mu_{f_3\to e}(b,e) = \sum_{c,d} f_3(c,d,e)\mu_{c\to f_3}(b,c)\mu_{d\to f_3}(d)$$

$$\mu_{f_3\to e}(b=0,e) = 9\begin{bmatrix} 1 & 2 \\ 2 & 1 \end{bmatrix}\begin{bmatrix} 1 \\ 1 \end{bmatrix} + 3\begin{bmatrix} 1 & 2 \\ 2 & 1 \end{bmatrix}\begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 36 \\ 36 \end{bmatrix}$$

$$\mu_{f_3\to e}(b=1,e) = 6\begin{bmatrix} 1 & 2 \\ 2 & 1 \end{bmatrix}\begin{bmatrix} 1 \\ 1 \end{bmatrix} + 1.5\begin{bmatrix} 1 & 2 \\ 2 & 1 \end{bmatrix}\begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 22.5 \\ 22.5 \end{bmatrix}$$

$$g(b,e) = \mu_{f_3\to e}(b,e) = \begin{bmatrix} 36 & 22.5 \\ 36 & 22.5 \end{bmatrix}$$

$$\mu_{a\to f_1}(a) = [1,1]^T \qquad f_1(a,b) = \begin{bmatrix} f_1(0,0), f_1(1,0) \\ f_1(0,1), f_1(1,1) \end{bmatrix} = \begin{bmatrix} 1,2 \\ 1,2 \end{bmatrix}$$

$$\mu_{d\to f_3}(d) = [1,1]^T$$

$$\mu_{f_1\to b}(b) = [3,3]^T$$

$$\mu_{b\to f2}(b) = [3,3]^T \qquad f_2(b,c) = \begin{bmatrix} 3,2 \\ 1,0.5 \end{bmatrix}$$

$\mu_{f_2\to c}(b,c)$ $\quad \mu_{f_2\to c}(c) = [15,4.5]^T$

$\mu_{c\to f_3}(b,c)$ $\quad \mu_{c\to f_3}(c) = [15,4.5]^T \qquad f_3(c=0,d,e) = \begin{bmatrix} 1,2 \\ 2,1 \end{bmatrix}$

$\mu_{f_3\to e}(b,e)$ $\quad \mu_{f_3\to e}(e) = [58.5,58.5]^T$

$$\mu_{e\to f_3}(e) = [1,1]^T \qquad f_3(c=1,d,e) = \begin{bmatrix} 1,2 \\ 2,1 \end{bmatrix}$$

$$\mu_{f_3\to d}(e) = [58.5,58.5]^T$$

$$\mu_{f_3\to c}(c) = [6,6]^T$$

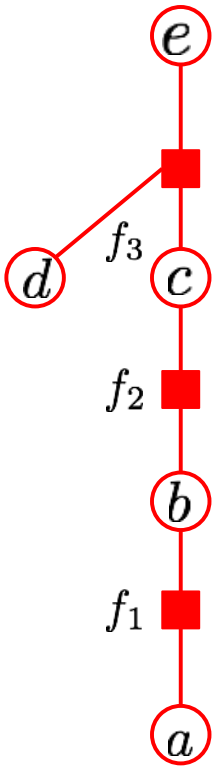
$$\mu_{c\to f_2}(c) = [6,6]^T$$

$$\mu_{f_2\to b}(b) = [24,15]^T$$

$$\mu_{b\to f_1}(b) = [24,15]^T$$

$$\mu_{f_1\to a}(a) = [39,78]^T$$

Example of the sum product algorithm



How do we compute P(e, b)?

New messages carrying b:

$$g(b,e) = \mu_{f_3 \to e}(b,e) = \begin{bmatrix} 36 & 22.5 \\ 36 & 22.5 \end{bmatrix}$$

$$P(b,e) = \frac{1}{Z} g(b,e) = \frac{1}{117} \begin{bmatrix} 36 & 22.5 \\ 36 & 22.5 \end{bmatrix}$$

$$\mu_{a \to f_1}(a) = [1,1]^T \qquad f_1(a,b) = \begin{bmatrix} f_1(0,0), f_1(1,0) \\ f_1(0,1), f_1(1,1) \end{bmatrix} = \begin{bmatrix} 1,2 \\ 1,2 \end{bmatrix}$$

$$\mu_{d \to f_3}(d) = [1,1]^T$$

$$\mu_{f_1 \to b}(b) = [3,3]^T$$

$$\mu_{b \to f2}(b) = [3,3]^T \qquad f_2(b,c) = \begin{bmatrix} 3,2 \\ 1,0.5 \end{bmatrix}$$

$$\mu_{f_2 \to c}(b,c) \quad \mu_{f_2 \to c}(c) = [15,4.5]^T$$

$$\mu_{c \to f_3}(b,c) \quad \mu_{c \to f_3}(c) = [15,4.5]^T \qquad f_3(c=0,d,e) = \begin{bmatrix} 1,2 \\ 2,1 \end{bmatrix}$$

$$\mu_{f_3 \to e}(b,e) \quad \mu_{f_3 \to e}(e) = [58.5,58.5]^T$$

$$\mu_{e \to f_3}(e) = [1,1]^T \qquad f_3(c=1,d,e) = \begin{bmatrix} 1,2 \\ 2,1 \end{bmatrix}$$

$$\mu_{f_3 \to d}(e) = [58.5,58.5]^T$$

$$\mu_{f_3 \to c}(c) = [6,6]^T$$

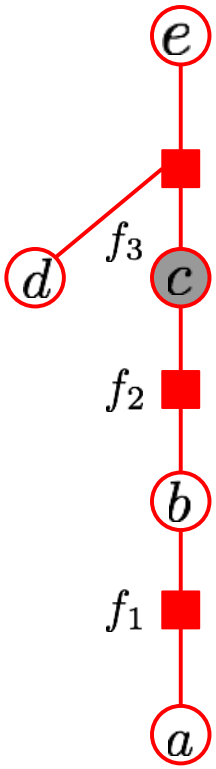
$$\mu_{c \to f_2}(c) = [6,6]^T$$

$$\mu_{f_2 \to b}(b) = [24,15]^T$$

$$\mu_{b \to f_1}(b) = [24,15]^T$$

$$\mu_{f_1 \to a}(a) = [39,78]^T$$

Example of the sum product algorithm



What if we observed c as c =[0,1]?

$$\mu_{a\to f_1}(a) = [1,1]^T \qquad f_1(a,b) = \begin{bmatrix} f_1(0,0), f_1(1,0) \\ f_1(0,1), f_1(1,1) \end{bmatrix} = \begin{bmatrix} 1,2 \\ 1,2 \end{bmatrix}$$

$$\mu_{d\to f_3}(d) = [1,1]^T$$

$$\mu_{f_1\to b}(b) = [3,3]^T$$

$$\mu_{b\to f2}(b) = [3,3]^T \qquad f_2(b,c) = \begin{bmatrix} 3,2 \\ 1,0.5 \end{bmatrix}$$

$$\mu_{f_2\to c}(c) = [15,4.5]^T$$

$$\mu_{c\to f_3}(c) = [15,4.5]^T \qquad f_3(c=0,d,e) = \begin{bmatrix} 1,2 \\ 2,1 \end{bmatrix}$$

$$\mu_{f_3\to e}(e) = [58.5,58.5]^T$$

$$\mu_{e\to f_3}(e) = [1,1]^T \qquad f_3(c=1,d,e) = \begin{bmatrix} 1,2 \\ 2,1 \end{bmatrix}$$

$$\mu_{f_3\to d}(e) = [58.5,58.5]^T$$

$$\mu_{f_3\to c}(c) = [6,6]^T$$

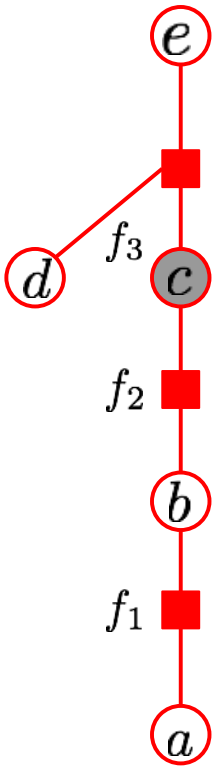
$$\mu_{c\to f_2}(c) = [6,6]^T$$

$$\mu_{f_2\to b}(b) = [24,15]^T$$

$$\mu_{b\to f_1}(b) = [24,15]^T$$

$$\mu_{f_1\to a}(a) = [39,78]^T$$

Example of the sum product algorithm



What if we observed c as c =[0,1]?

It changes the outgoing messages from c

$$\mu_{a\to f_1}(a) = [1,1]^T \qquad f_1(a,b) = \begin{bmatrix} f_1(0,0), f_1(1,0) \\ f_1(0,1), f_1(1,1) \end{bmatrix} = \begin{bmatrix} 1,2 \\ 1,2 \end{bmatrix}$$

$$\mu_{d\to f_3}(d) = [1,1]^T$$

$$\mu_{f_1\to b}(b) = [3,3]^T$$

$$\mu_{b\to f2}(b) = [3,3]^T \qquad f_2(b,c) = \begin{bmatrix} 3,2 \\ 1,0.5 \end{bmatrix}$$

$$\mu_{f_2\to c}(c) = [15,4.5]^T$$

$$\mu_{c\to f_3}(c) = [15,4.5]^T \qquad f_3(c=0,d,e) = \begin{bmatrix} 1,2 \\ 2,1 \end{bmatrix}$$

$$\mu_{f_3\to e}(e) = [58.5,58.5]^T$$

$$\mu_{e\to f_3}(e) = [1,1]^T \qquad f_3(c=1,d,e) = \begin{bmatrix} 1,2 \\ 2,1 \end{bmatrix}$$

$$\mu_{f_3\to d}(e) = [58.5,58.5]^T$$

$$\mu_{f_3\to c}(c) = [6,6]^T$$

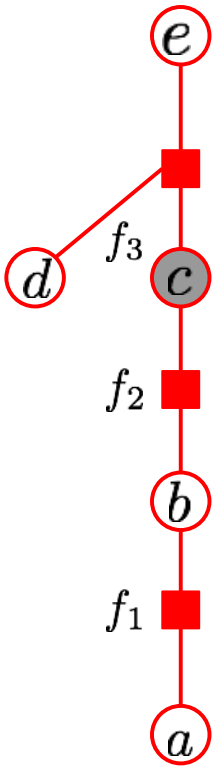
$$\mu_{c\to f_2}(c) = [6,6]^T$$

$$\mu_{f_2\to b}(b) = [24,15]^T$$

$$\mu_{b\to f_1}(b) = [24,15]^T$$

$$\mu_{f_1\to a}(a) = [39,78]^T$$

Example of the sum product algorithm



What if we observed c as c =[0,1]?

It changes the outgoing messages from c
And their derivative messages

$$\mu_{a\to f_1}(a) = [1,1]^T$$
$$\mu_{d\to f_3}(d) = [1,1]^T$$
$$\mu_{f_1\to b}(b) = [3,3]^T$$
$$\mu_{b\to f2}(b) = [3,3]^T$$
$$\mu_{f_2\to c}(c) = [15,4.5]^T$$
$$\mu_{c\to f_3}(c) = [15,4.5]^T$$
$$\mu_{f_3\to e}(c) = [58.5,58.5]^T$$
$$\mu_{e\to f_3}(e) = [1,1]^T$$
$$\mu_{f_3\to d}(e) = [58.5,58.5]^T$$
$$\mu_{f_3\to c}(c) = [6,6]^T$$
$$\mu_{c\to f_2}(c) = [6,6]^T$$
$$\mu_{f_2\to b}(b) = [24,15]^T$$
$$\mu_{b\to f_1}(b) = [24,15]^T$$
$$\mu_{f_1\to a}(a) = [39,78]^T$$

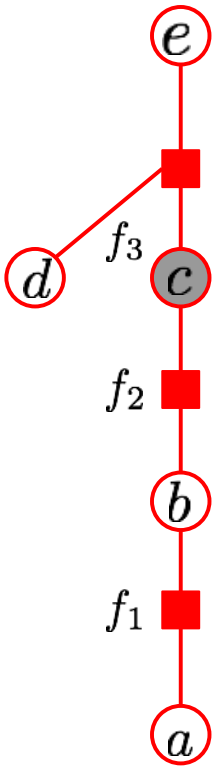
$$f_1(a,b) = \begin{bmatrix} f_1(0,0), f_1(1,0) \\ f_1(0,1), f_1(1,1) \end{bmatrix} = \begin{bmatrix} 1,2 \\ 1,2 \end{bmatrix}$$

$$f_2(b,c) = \begin{bmatrix} 3,2 \\ 1,0.5 \end{bmatrix}$$

$$f_3(c=0,d,e) = \begin{bmatrix} 1,2 \\ 2,1 \end{bmatrix}$$

$$f_3(c=1,d,e) = \begin{bmatrix} 1,2 \\ 2,1 \end{bmatrix}$$

Example of the sum product algorithm



What if we observed c as c =[0,1]?

It changes the outgoing messages from c
And their derivative messages

$$\mu_{a\to f_1}(a) = [1,1]^T$$
$$\mu_{d\to f_3}(d) = [1,1]^T$$
$$\mu_{f_1\to b}(b) = [3,3]^T$$
$$\mu_{b\to f2}(b) = [3,3]^T$$
$$\mu_{f_2\to c}(c) = [15,4.5]^T$$
$$\mu_{c\to f_3}(c) = [0,1]^T$$
$$\cancel{\mu_{f_3\to e}(c) = [58.5,58.5]^T}$$
$$\mu_{e\to f_3}(e) = [1,1]^T$$
$$\cancel{\mu_{f_3\to d}(e) = [58.5,58.5]^T}$$
$$\mu_{f_3\to c}(c) = [6,6]^T$$
$$\mu_{c\to f_2}(c) = [0,1]^T$$
$$\cancel{\mu_{f_2\to b}(b) = [24,15]^T}$$
$$\cancel{\mu_{b\to f_1}(b) = [24,15]^T}$$
$$\cancel{\mu_{f_1\to a}(a) = [39,78]^T}$$

$$f_1(a,b) = \begin{bmatrix} f_1(0,0), f_1(1,0) \\ f_1(0,1), f_1(1,1) \end{bmatrix} = \begin{bmatrix} 1,2 \\ 1,2 \end{bmatrix}$$

$$f_2(b,c) = \begin{bmatrix} 3,2 \\ 1,0.5 \end{bmatrix}$$

$$f_3(c=0,d,e) = \begin{bmatrix} 1,2 \\ 2,1 \end{bmatrix}$$

$$f_3(c=1,d,e) = \begin{bmatrix} 1,2 \\ 2,1 \end{bmatrix}$$

Topics to review

- k-NN, linear regression, logistic regression, probability distributions, MLE/MAP
- Bayes Theorem, SGD, techinques to improve SGD, variants of neural nets, backpropagation
- k-means, PCA, mixture models (Gaussians, Bernoullis), Naive Bayes, EM algorithm
- Graphical models, BN/MRF/FG conversions, HMM, sum-product/max-sum algorithm, junction tree, (loopy) belief propagation