NPFL108 – Bayesian inference

Inference in discrete graphs

Filip Jurčíček

Institute of Formal and Applied Linguistics
Charles University in Prague
Czech Republic

Home page: http://ufal.mff.cuni.cz/~jurcicek

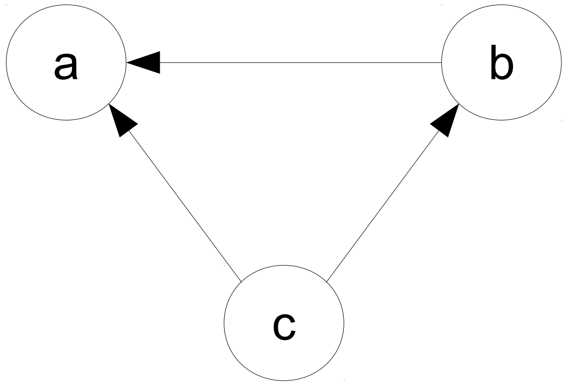
Version: 13/03/2014

Outline

- Inference in discrete graphical models
  - Variables, Conditional probabilities, Parameters, Plate notation
  - Conditional Independence
  - Markov blanket
  - Message passing, Belief propagation, Loopy belief propagation

Graphical models

- Provide simple way to visualize probabilistic models
- Give insight into properties of the model, e.g. conditional independence
- Help to understand complex inference methods



Examples of Probabilistic Graphical Models

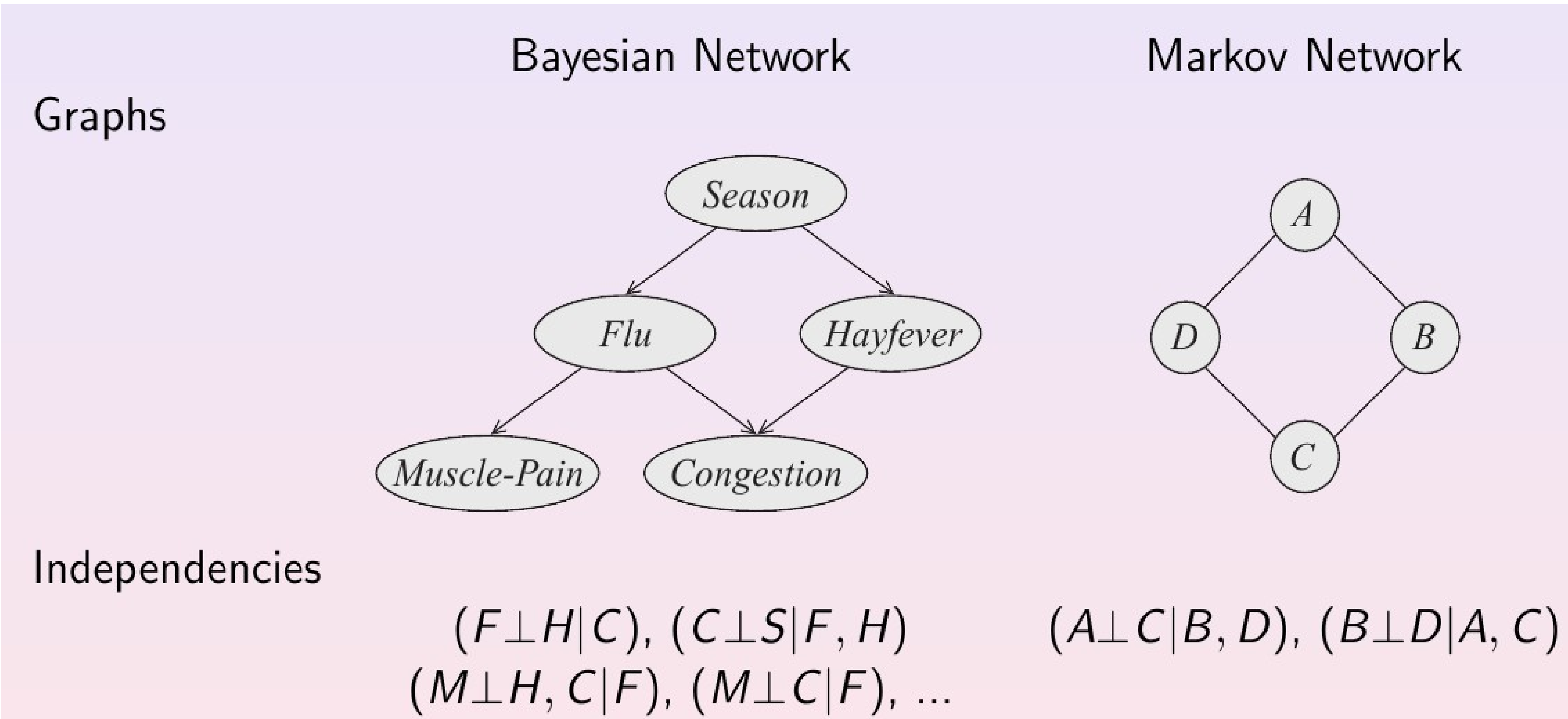


Figure source [Koller et al. 2009].

Examples of Probabilistic Graphical Models

BN Examples: Naive Bayes

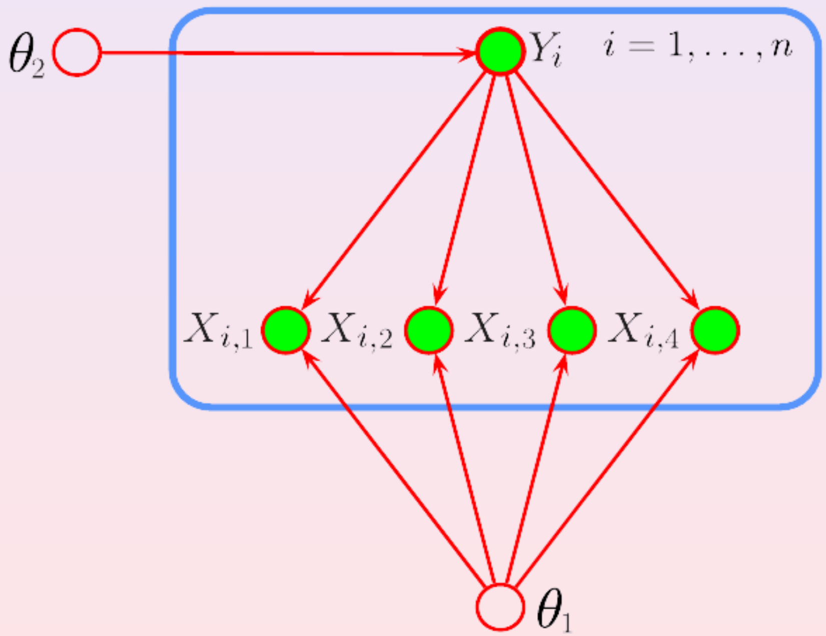


Figure source [Murphy 2012].

BN Examples: Hidden Markov Model

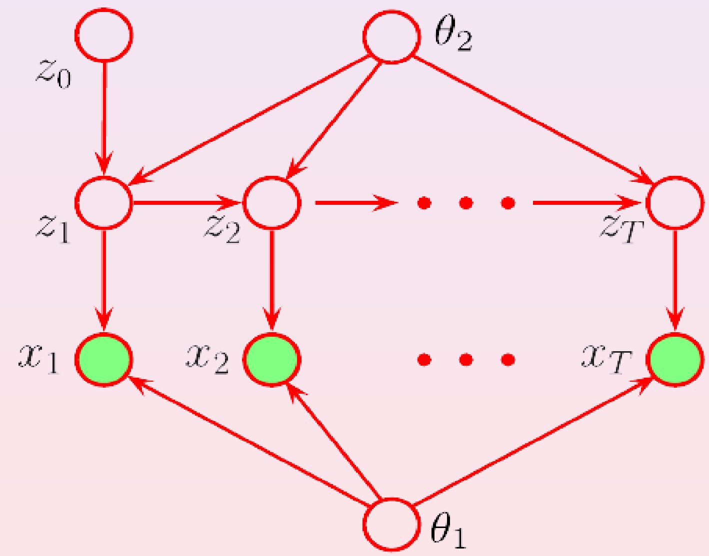


Figure source [Murphy 2012].

Plate notation 1

- Is a method for representing variables that repeat in a graphical model

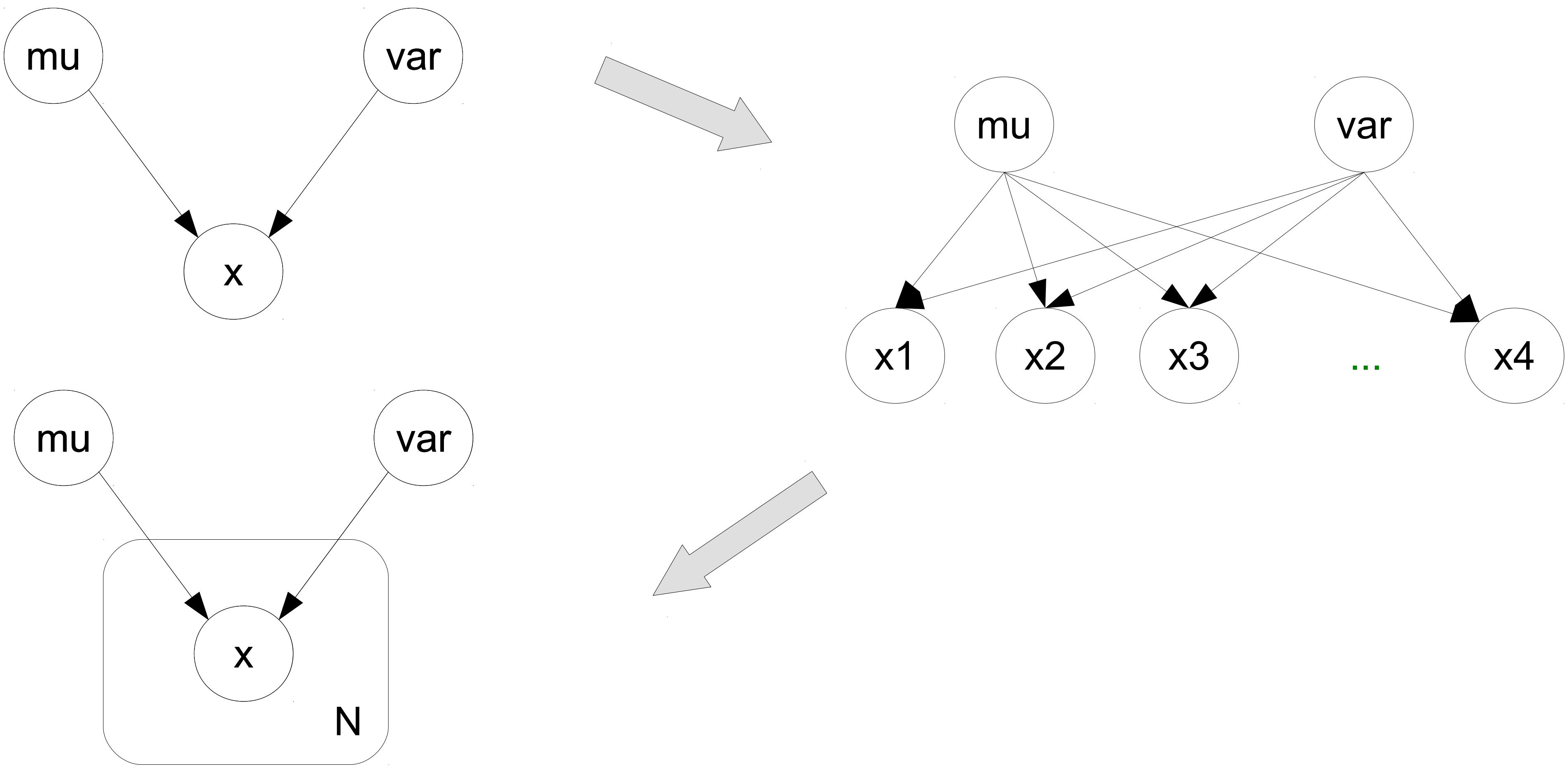
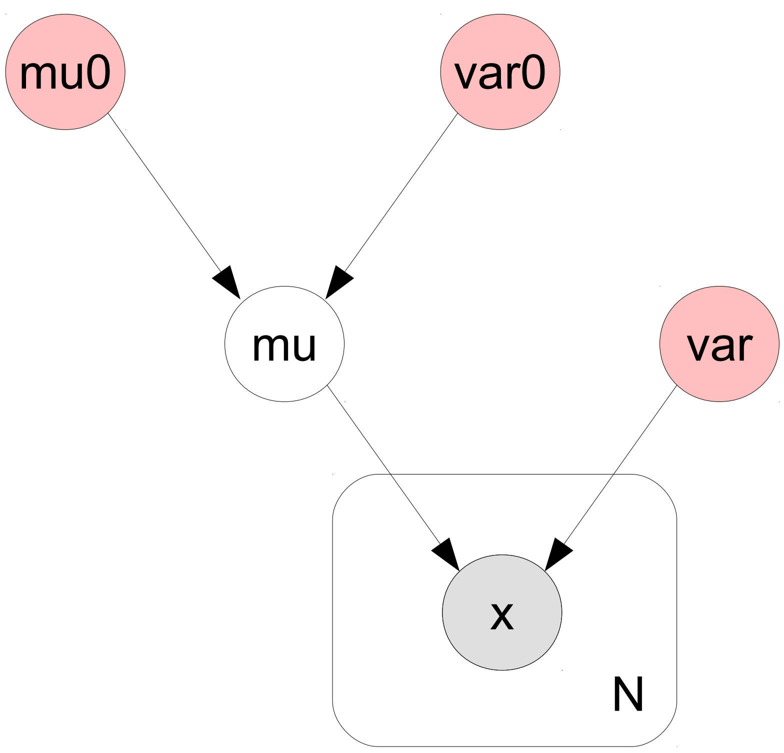


Plate notation 2

- Observed variables are marked
  - including priors (these can be denoted differently)



vs.

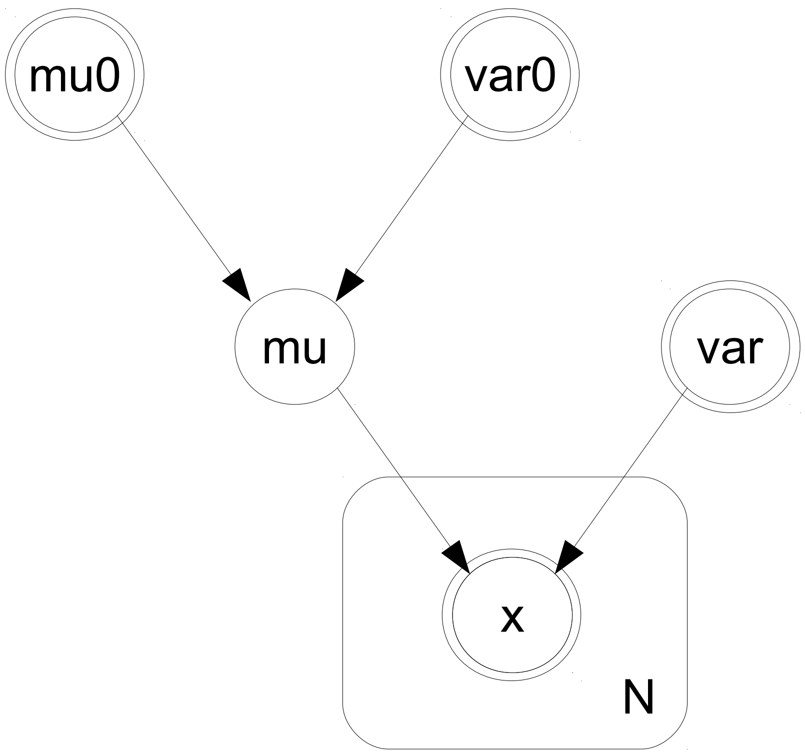
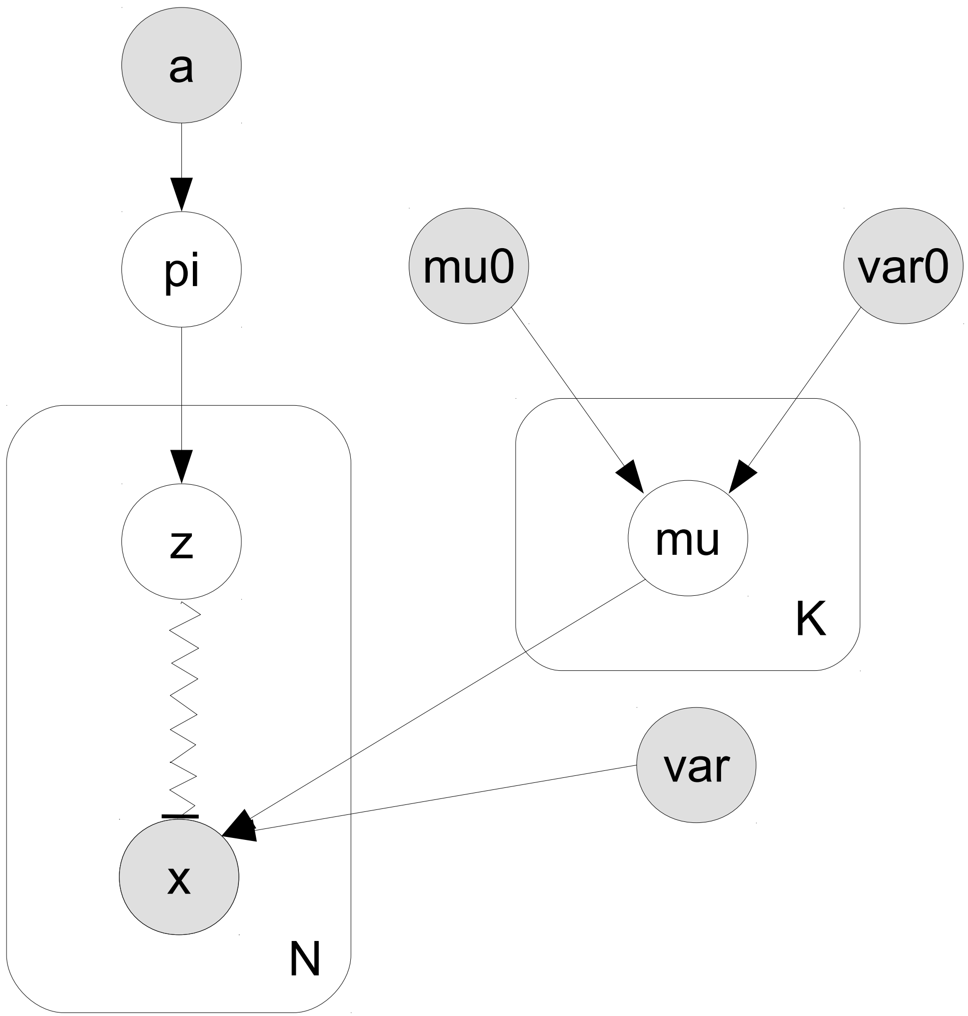
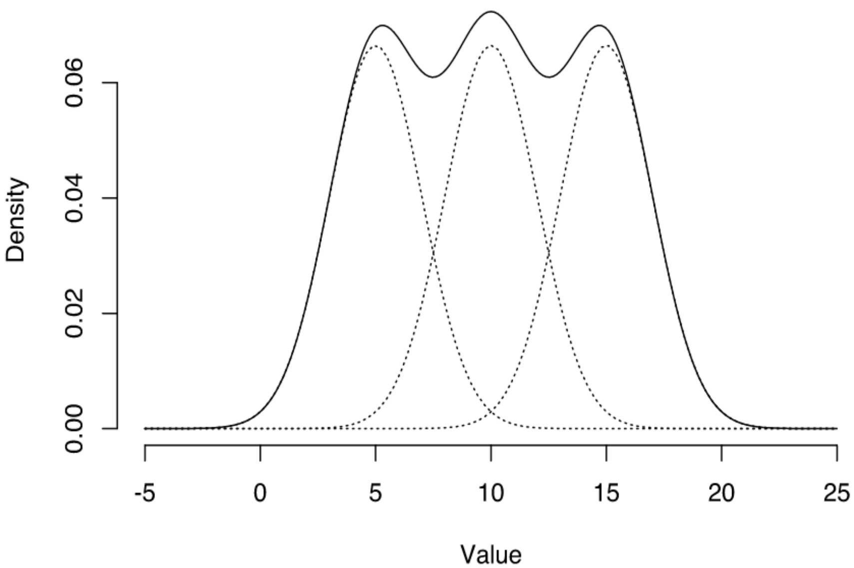


Plate notation 3 - switching

- Mixture model





Source Wikipedia, 2014

$$\pi \sim Dir(\alpha)$$

$$\mu_k \sim N(\mu_{0,} \sigma_0^2)$$

$$z_n \sim Categorical(\pi)$$

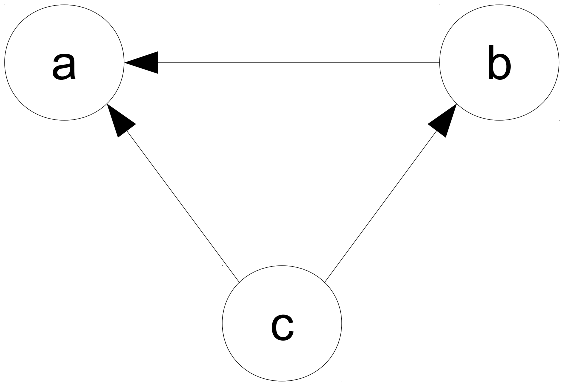
$$x_n \sim N(\mu_{z_n}, \sigma^2)$$

Bayesian Networks

- BN is a directed graphical model consisting of
  - nodes – random variable
  - links – probabilistic relationship between random var.
- Factorisation:
  - The basic idea is to represent a complex distribution by a product of simpler distribution

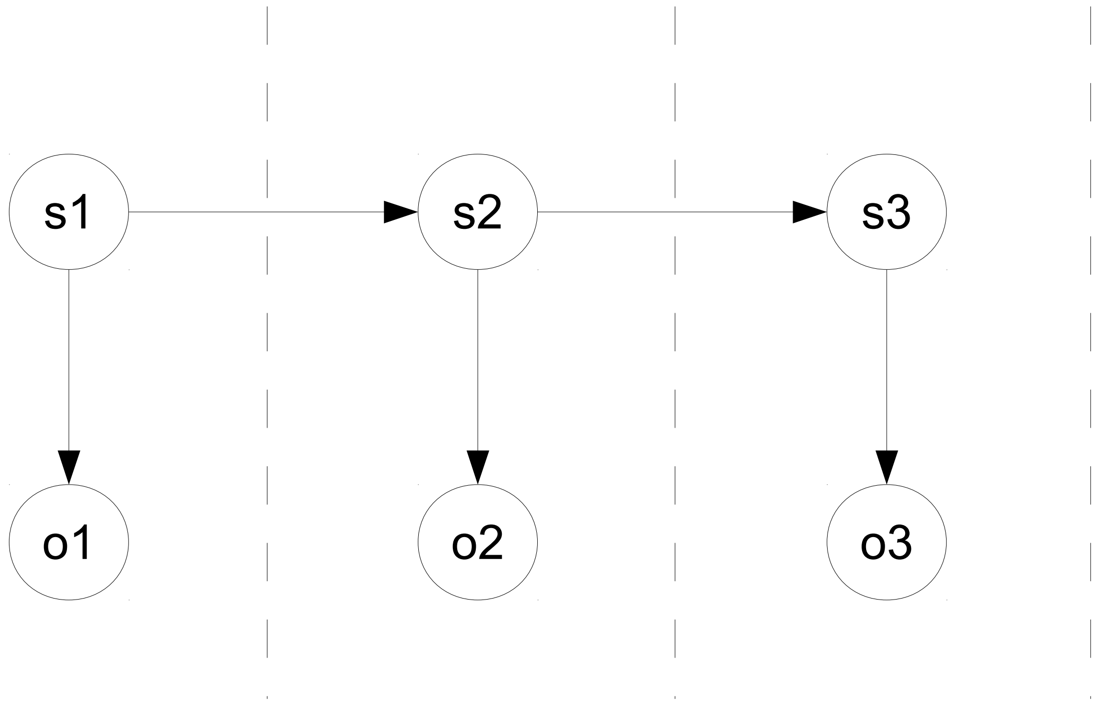
$$p(a, b, c) = p(a|b, c)\, p(b|c)\, p(c)$$

- This can be graphically represented as



Dynamic Bayesian Networks

- Like a Bayesian network
- However, it can grow.

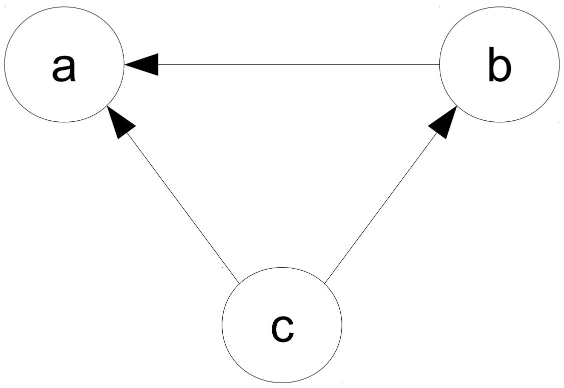


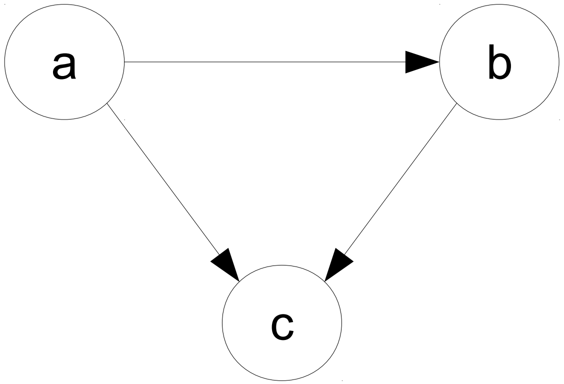
Factorization

- Factorization is not unique
  - it can have many, theoretically equivalent, forms

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

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





Conditional independence

- Independence of two random variables

$$p(a, b) = p(a)\,p(b)$$

- Conditional independence of two random variables

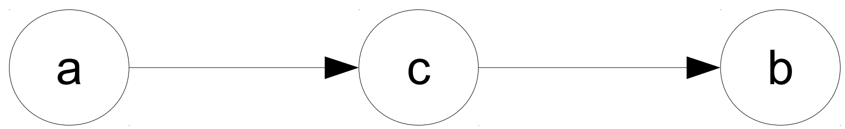
$$p(a, b|c) = p(a|c)\,p(b|c)$$

- This is also noted as

$$a \perp b | c$$

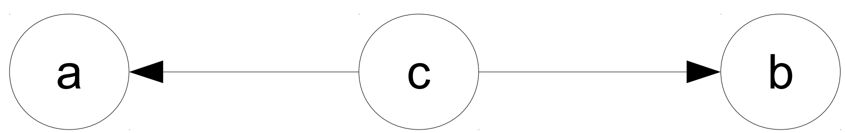
Examples

- Head to tail



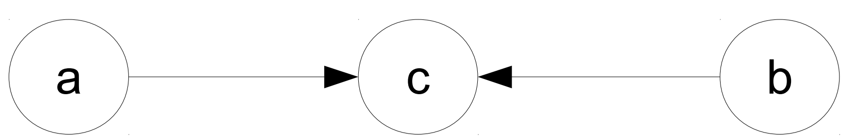
$$a \perp b | c$$

- Tail to tail



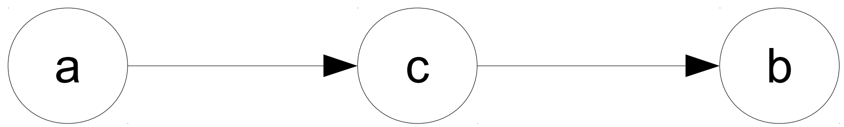
$$a \perp b | c$$

- Head to head



$$a \perp b \not| \, c$$

Head to tail



$$a \perp b \mid c$$

$$p(ab|c) \equiv \frac{P(abc)}{P(c)} = \frac{P(a)P(c|a)P(b|c)}{P(c)}$$

$$P(a|c) \leftarrow \frac{P(a,c)}{P(c)}$$

$$p(ab|c) = P(a|c)P(b|c)$$

D-separation

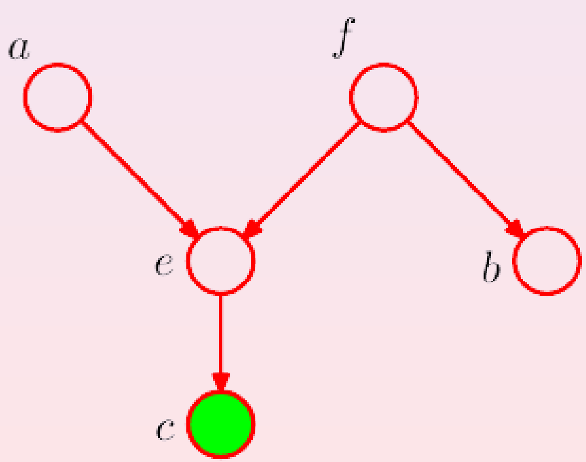
Conditional independence properties can be read directly from the graph.

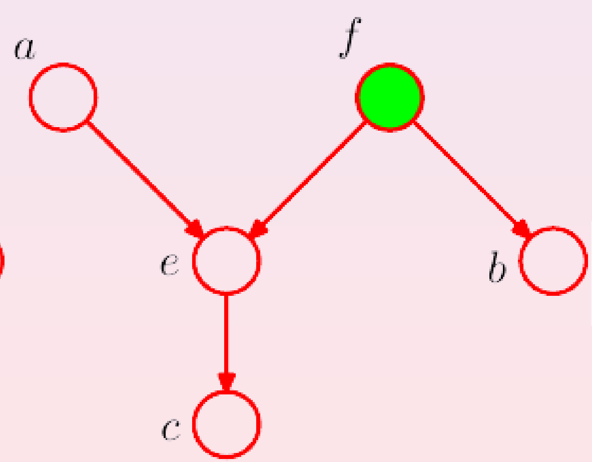
We say that the sets of nodes A, B and C satisfy $(A \perp B | C)$ when all of the possible paths from any node in A to any node in B are blocked.

A path will be blocked if it contains a node x with arrows meeting at x

1 - i) head-to-tail or ii) tail-to-tail and x is C.

2 - head-to-head and neither x, nor any of its descendants, is in C.





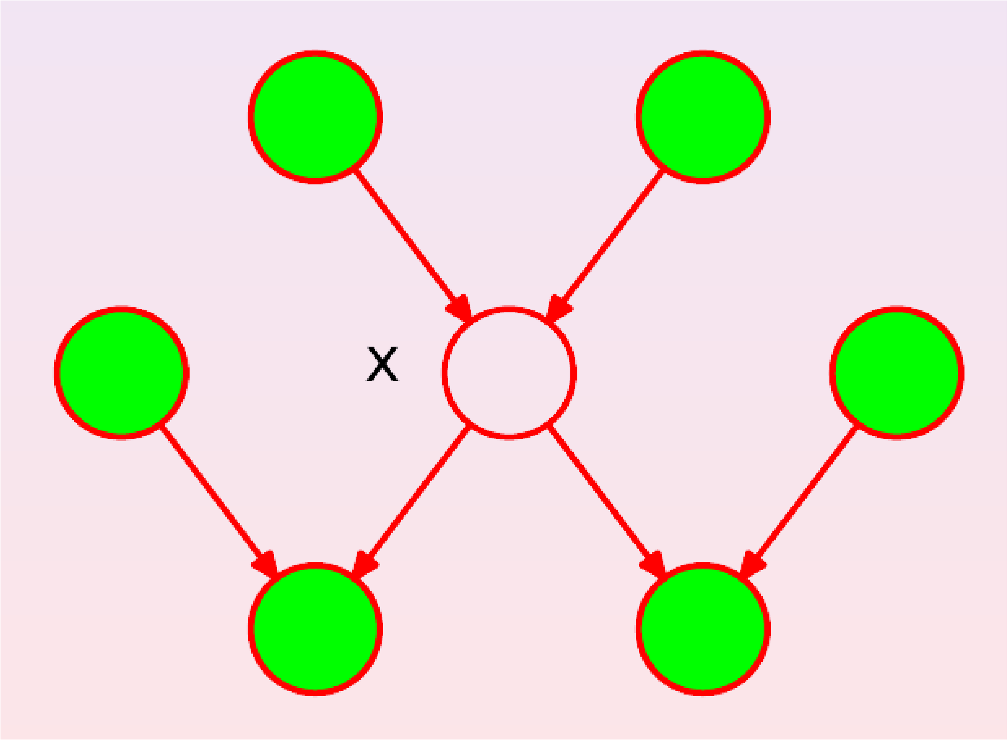
$(a \perp b | c)$ does not follow from the graph.

$a \perp b | f$ is implied by the graph.

Figure source [Bishop 2006].

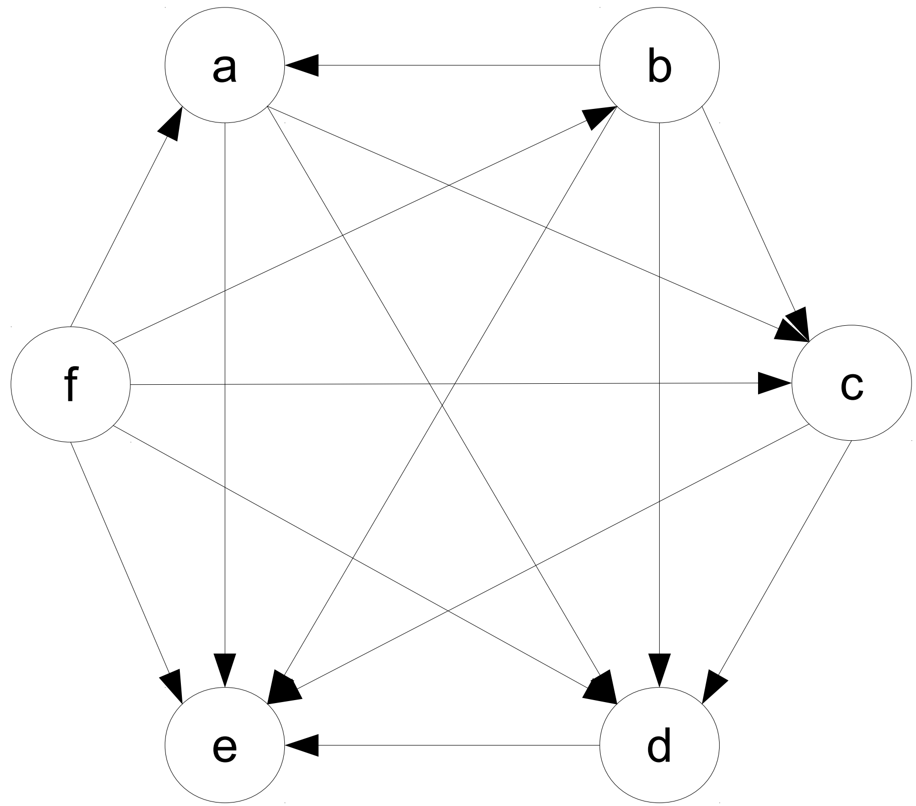
Markov blanket

- The Markov blanket of a node x is the set of nodes comprising parents, children and co-parents of x.
- It is the minimal set of nodes that isolates a node from the rest, that is, x is CI of any other node in the graph given its Markov blanket .



Source [Bishop 2006]

Fully connected networks



$$p(a,b,c,d,e,f) = p(e|a,b,c,d,f)\,p(d|a,b,c,f)\\ p(c|a,b,f)\,p(a|b,f)\,p(b|f)\,p(f)$$

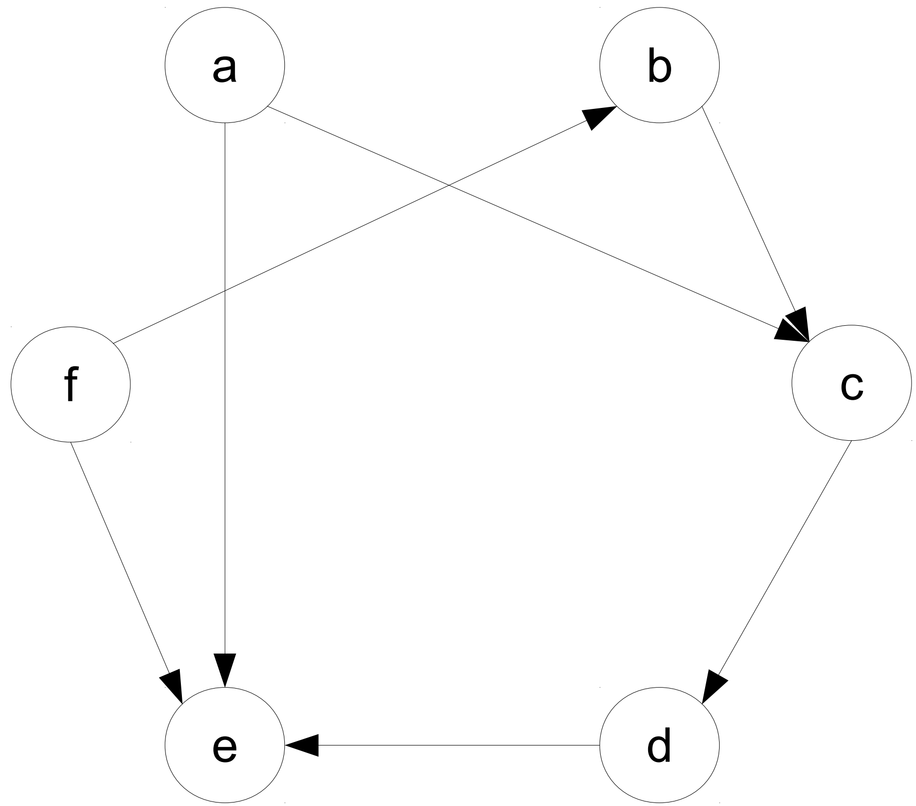
Marginalization

- Computing joint distribution of only some subset of variables

$$p(a,b,c)=\sum_d \sum_e \sum_f p(a,b,c,d,e,f)$$

- Trivial. However, it can be slow.

Partially connected networks



$$p(a,b,c,d,e,f) = p(e|a,d,f)\,p(d|c)\,p(c|a,b)$$
$$p(b|f)\,p(a)\,p(f)$$

Marginalization

- Marginalization on factored partially connected network speed the inference

$$p(a,b,c)=\sum_d \sum_e \sum_f p(e|a,d,f)\,p(d|c)\,p(c|a,b)\,p(b|f)\,p(a)\,p(f)$$

$$xy+xz \quad = \quad x(y+z)$$

- Use the fact that x distributes over +

2 multiplies + 1 addition   1 multiply + 1 addition

Marginalization on factored joint dist.

- In this case, you need
  - |d|.|e|.|f| additions
  - |d|.|e|.|f|*5 multiplications

$$p(a,b,c)=\sum_d \sum_e \sum_f p(e|a,d,f)\,p(d|c)\,p(c|a,b)\,p(b|f)\,p(a)\,p(f)$$

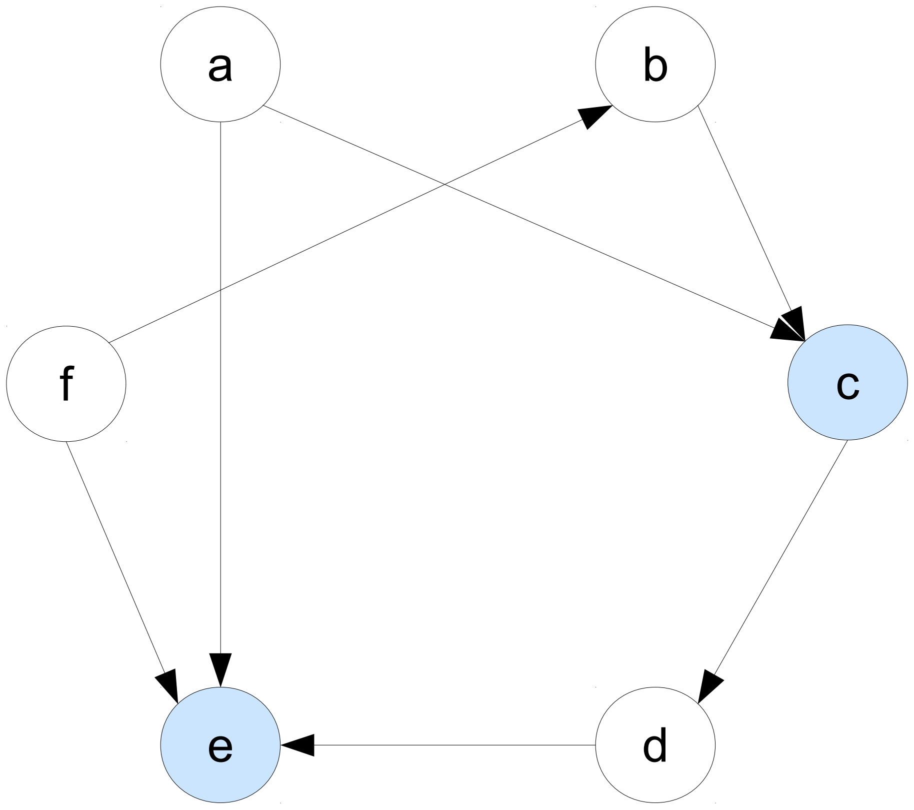
In this case, you need

- |d|.|e|.|f| additions
- (|d| +3)* |f| + 3 multiplications

$$p(a,b,c)=p(a)\,p(c|a,b)\sum_f p(b|f)\,p(f)\left(\sum_d p(d|c)\left(\sum_e p(e|a,d,f)\right)\right)$$

Posterior distribution

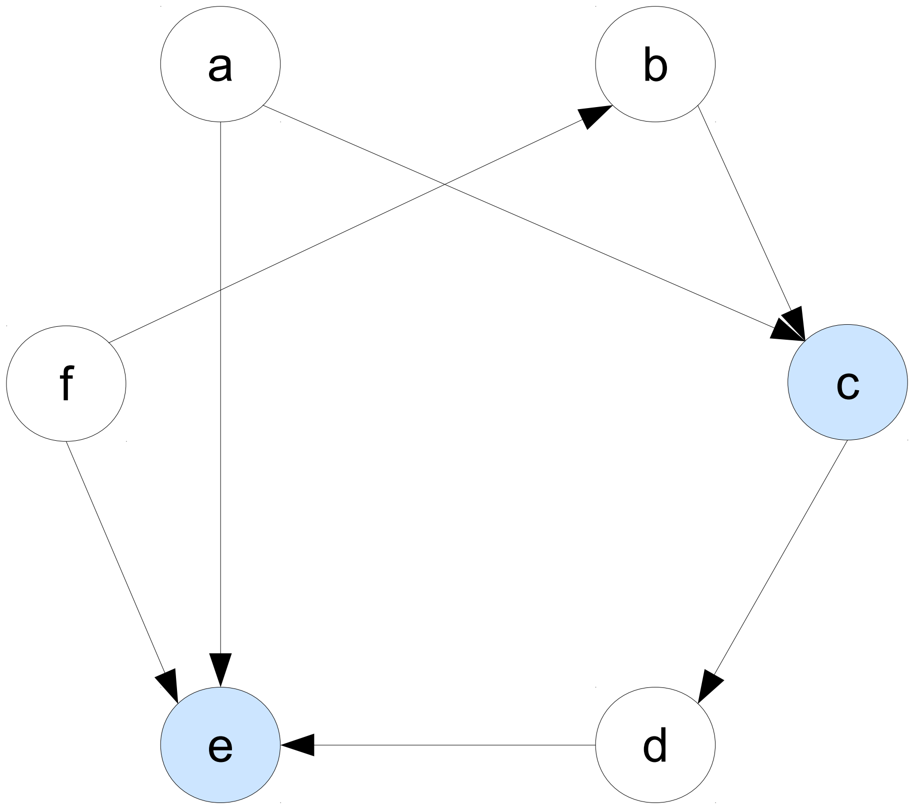
- We are not much interested in the joint distribution
- More often we want to know the posteriors
  - for some of the random variables given some observed data



Posterior given the some variables

- Joint a, b, d, f given c, e

$$p(a, b, d, f | c, e) = ?$$



Posterior given the some variables

- Joint prob. of a, b, d, f given c, e

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

$$= \frac{p(a,b,c,d,e,f)}{\sum_{a,b,d,f} p(a,b,c,d,e,f)}$$

Posterior given the known variables

- Joint prob. of a, b, d, f given c = C, e = E

$$p(a, b, d, f \mid c=C, e=E) = \frac{p(a, b, c=C, d, e=E, f)}{p(c=C, e=E)}$$

$$= \frac{p(a, b, c=C, d, e=E, f)}{\sum_{a,b,d,f} p(a, b, c=C, d, e=E, f)}$$

Inference in Bayesian networks

- Simple marginalisation is inefficient
- Use dynamic programming

- Belief propagation
  - using dynamic programming
  - exact on trees
  - equivalent to Forward-Bacward algorithm for HMMs

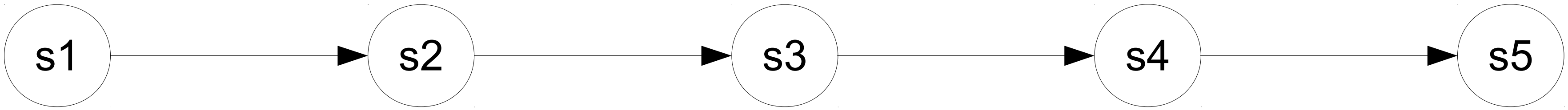
Belief propagation on a chain

- Compute p(s5)

$$p(s_5)=\sum_{s_1, s_2, s_3, s_4} p(s_1, s_2, s_3, s_4, s_5)$$

$$=\sum_{s_4} p(s_5|s_4)\sum_{s_3} p(s_4|s_3)\sum_{s_2} p(s_3|s_2)\sum_{s_1} p(s_2|s_1)\, p(s_1)$$

- Use dynamic programming
  - aka message passing algorithm



Forward message passing

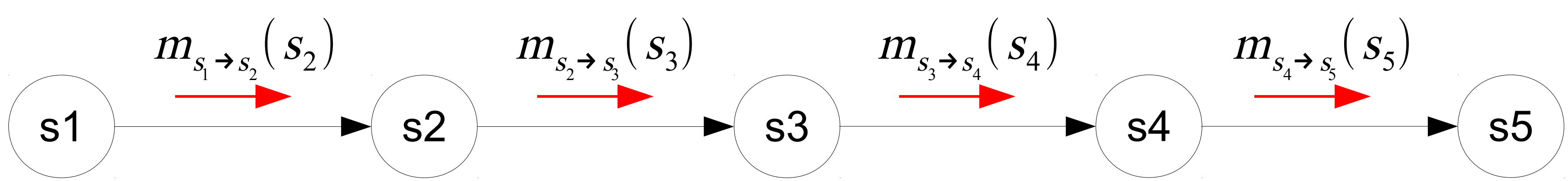
$$p(s_5) = \sum_{s_4} p(s_5|s_4) \sum_{s_3} p(s_4|s_3) \sum_{s_2} p(s_3|s_2) \sum_{s_1} p(s_2|s_1)\, p(s_1)$$

$$p(s_5) = \sum_{s_4} p(s_5|s_4) \sum_{s_3} p(s_4|s_3) \sum_{s_2} p(s_3|s_2)\, m_{s_1 \to s_2}(s_2)$$

$$p(s_5) = \sum_{s_4} p(s_5|s_4) \sum_{s_3} p(s_4|s_3)\, m_{s_2 \to s_3}(s_3)$$

$$p(s_5) = \sum_{s_4} p(s_5|s_4)\, m_{s_3 \to s_4}(s_4)$$

$$p(s_5) = m_{s_4 \to s_5}(s_5)$$



Backward message passing

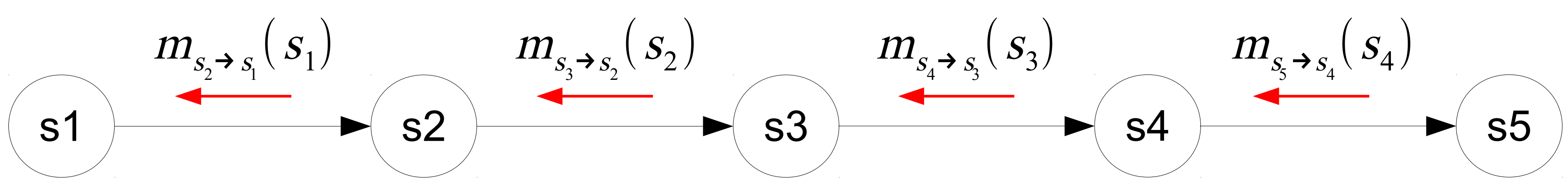
$$p(s_1)=\sum_{s_2} p(s_2|s_1)\,p(s_1)\sum_{s_3} p(s_3|s_2)\sum_{s_4} p(s_4|s_3)\sum_{s_5} p(s_5|s_4)$$

$$p(s_1)=\sum_{s_2} p(s_2|s_1)\,p(s_1)\sum_{s_3} p(s_3|s_2)\sum_{s_4} p(s_4|s_3)\,m_{s_5\to s_4}(s_4)$$

$$p(s_1)=\sum_{s_2} p(s_2|s_1)\,p(s_1)\sum_{s_3} p(s_3|s_2)\,m_{s_4\to s_3}(s_3)$$

$$p(s_1)=\sum_{s_2} p(s_2|s_1)\,p(s_1)\,m_{s_3\to s_2}(s_2)$$

$$p(s_1)=m_{s_2\to s_1}(s_1)$$



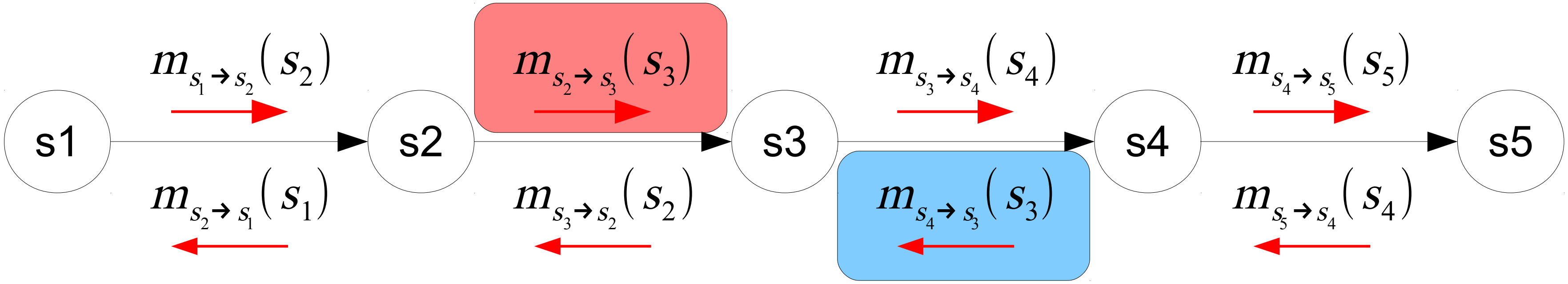
Message passing

$$p(s_3=S_3)=\sum_{s_5} p(s_5|s_4)\sum_{s_4} p(s_4|s_3=S_3)\sum_{s_2} p(s_3=S_3|s_2)\sum_{s_1} p(s_2|s_1)\,p(s_1)$$

$$p(s_3=S_3)=\sum_{s_5} p(s_5|s_4)\sum_{s_4} p(s_4|s_3=S_3)\,m_{s_2\rightarrow s_3}(s_3=S_3)$$

$$p(s_3=S_3)=m_{s_2\rightarrow s_3}(s_3=S_3)\sum_{s_4} p(s_4|s_3=S_3)\sum_{s_5} p(s_5|s_4)$$

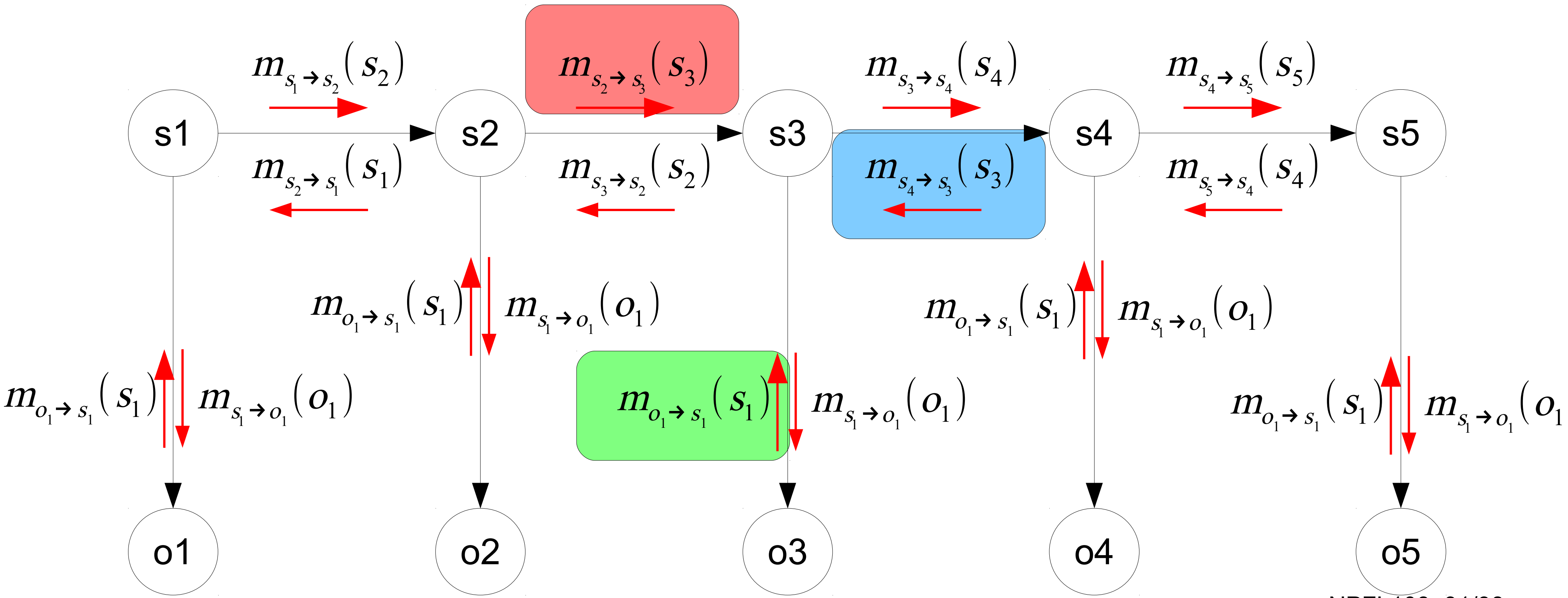
$$p(s_3=S_3)=m_{s_2\rightarrow s_3}(s_3=S_3)\,m_{s_4\rightarrow s_3}(s_3=S_3)$$



Belief propagation on a tree

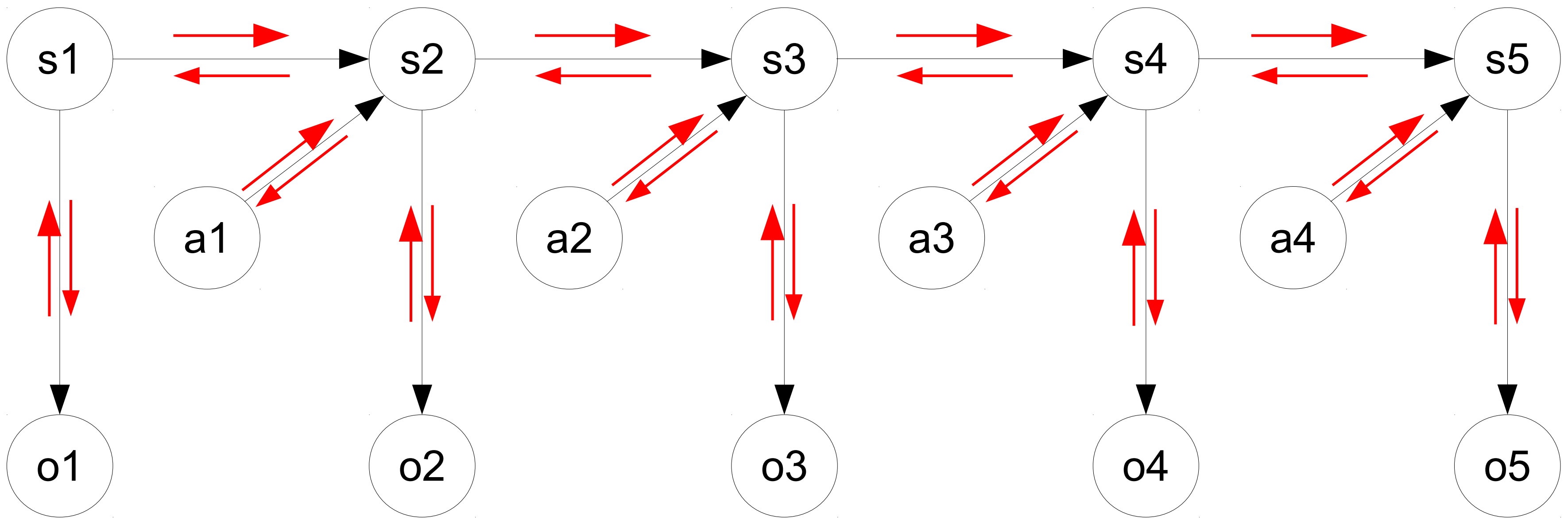
- We are interested in p(s3 = S3)

$$p(s_3 = S_3) = m_{s_2 \to s_3}(s_3 = S_3)\, m_{o_3 \to s_3}(s_3 = S_3)\, m_{s_4 \to s_3}(s_3 = S_3)$$



Belief propagation on a tree

- The same algorithm scales to an arbitrary tree



Inference in general Bayesian Networks

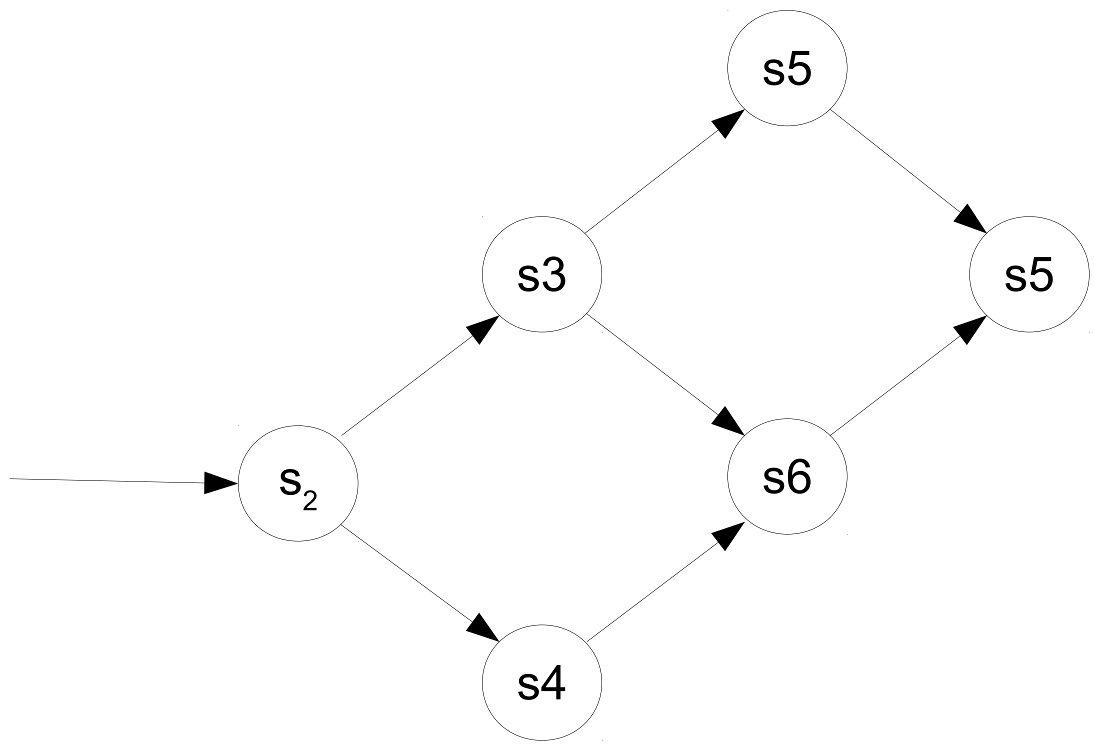
- Exact inference intractable
  - Approximation techniques necessary

- Loopy belief propagation
  - Infers the marginal distribution for the nodes
  - Approximate the joint distribution by a product of marginals

$$p(s_1, s_2, s_3, s_4, s_5) \approx p(s_1)\, p(s_2)\, p(s_3)\, p(s_4)(s_5)$$

BP in a general Bayesian Network

- It is not a tree any more

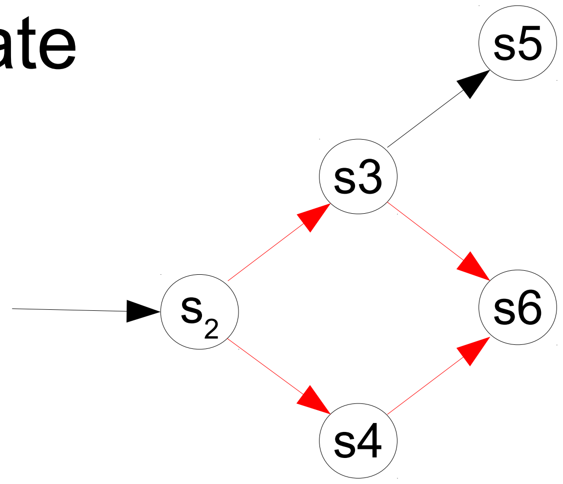


BP in a general Bayesian Network

- If used on networks with cycles then it is inexact
  - can be used iteratively → Loopy Belief Propagation
  - it converges to some local optimum
  - most of the time it works

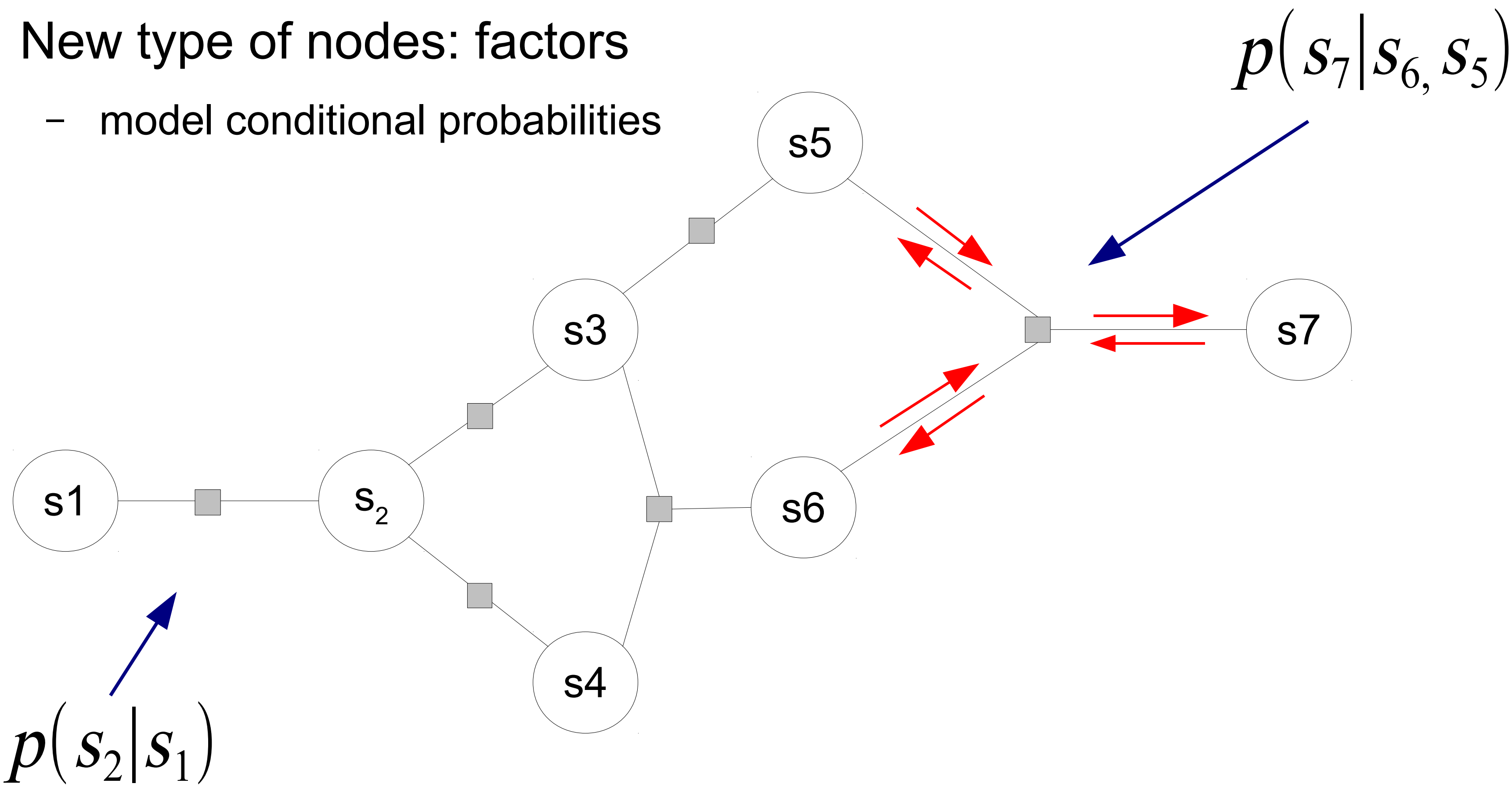
- Iterate until convergence
  - there are multiple ways how the iterate

→ Loopy belief propagation



Factor graphs

- Very often used to explain LPB, EP, etc.
- LPB cannot be easily described on BN
  - New type of nodes: factors
    - model conditional probabilities



- Then messages are defined between variable and factors nodes

Loopy belief propagation

- **Iterate over nodes n \in N**

$$F(X_n) \quad X_n = ne(f_j) - s_n$$

$$p(s_n) = \prod_{j \in ne(s_n)} m_{f_j \to s_n}(s_n)$$

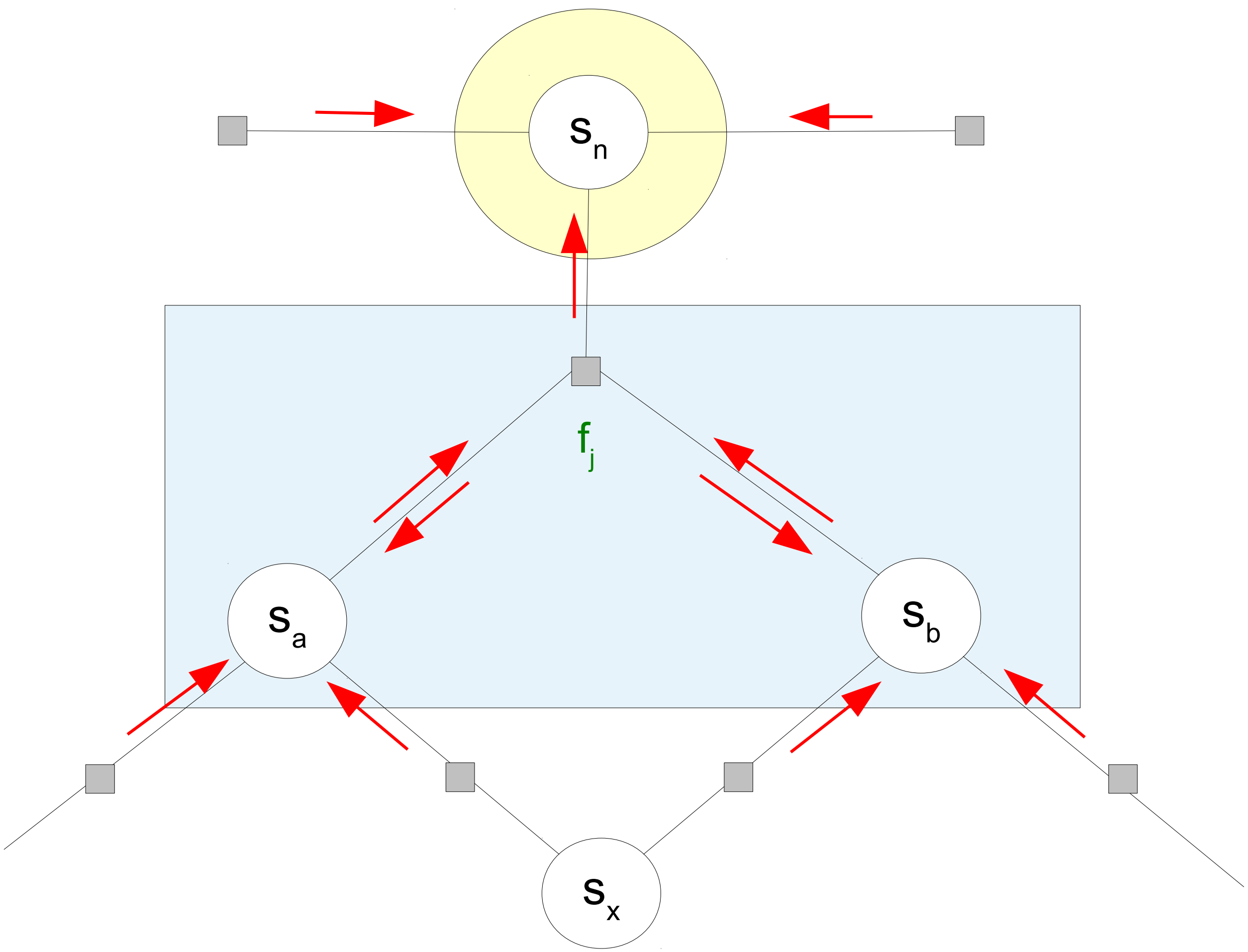
$$m_{f_j \to s_n}(s_n) = \sum_{s_a} \sum_{s_b} \sum_{\ldots} P(s_n | s_a, s_b, \ldots) \prod_{k \in a, b, \ldots} m_{s_k \to f_j}(s_k)$$

$$k \in ne(f_j) - s_n$$

$$m_{s_k \to f_j}(s_k) = \frac{p(s_k)}{m_{f_j \to s_k}(s_k)} = \frac{\prod_{l \in ne(s_k)} m_{f_l \to s_k}(s_k)}{m_{f_j \to s_k}(s_k)} = \prod_{l \in ne(s_k) - f_j} m_{f_l \to s_k}(s_k)$$

- **Not all messages are available at the beginning**
  - Try some reasonable initial values for
    - probabilities and messages 1s

Messages to a factor in LPB



k - variables

l - factors