

LTAT.02.004 MACHINE LEARNING II

Graphical models

Sven Laur
University of Tartu

Discrete random variables

- ▷ A *random variable* X with possible *outcomes* $x \in \text{supp}(X)$
- ▷ Compact notation for probabilities

$$\Pr[x_1] := \Pr[\xi \leftarrow X_1 : \xi = x_1]$$

$$\Pr[x_1 \wedge x_2] := \Pr[\xi_1 \leftarrow X_1, \xi_2 \leftarrow X_2 : \xi_1 = x_1 \wedge \xi_2 = x_2]$$

- ▷ Bayes formula

$$\Pr[a|b] = \frac{\Pr[a \wedge b]}{\Pr[b]} = \frac{\Pr[b|a] \Pr[a]}{\Pr[b]}$$

- ▷ Independence of random variables $X_1 \dots X_m \perp Y_1, \dots Y_n$:

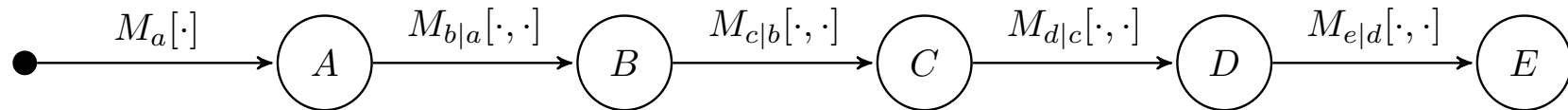
$$\Pr[x_1 \wedge \dots \wedge x_m \wedge y_1 \wedge \dots \wedge y_n] = \Pr[x_1 \wedge \dots \wedge x_m] \cdot \Pr[y_1 \wedge \dots \wedge y_n]$$

- ▷ Marginalisation over variables Y_1, \dots, Y_n :

$$\Pr[x_1 \wedge \dots \wedge x_m] = \sum_{y_1, \dots, y_n} \Pr[x_1 \wedge \dots \wedge x_m \wedge y_1 \wedge \dots \wedge y_n]$$

Common models

Markov chain



Definition. Let X_1, X_2, \dots be correlated random variables such that the probability of the observation x_{i+1} depends only on the observation x_i . Then the entire process is known as Markov chain.

Parametrisation. Markov chain is determined by specifying

- ▷ state spaces $\mathcal{S}_1 \dots, \mathcal{S}_n$
- ▷ initial probabilities $\Pr[x_1]$
- ▷ state transition probabilities $\Pr[x_{i+1}|x_i]$

What questions can we ask?

Sampling: What are typical outcomes of the chain?

- ▷ Synthesis of time-series, textures, sounds, games movements.

Stationary distribution: What happens if we run the chain infinitely long?

- ▷ Getting samples from an unnormalised posterior, optimisation tasks.

Likelihood estimation: What is a probability of an observation x_1, \dots, x_n ?

- ▷ Reasoning about probabilities and clustering sequences.

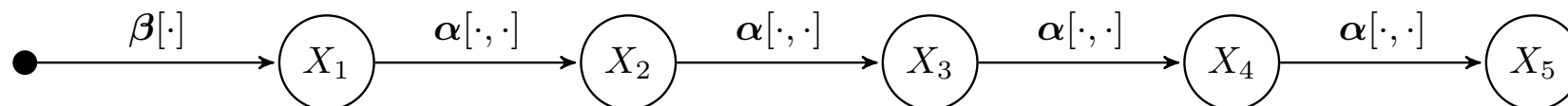
Decoding: What is the most probable outcome x_1, \dots, x_n ?

- ▷ Imputing missing values. Rudimentary logical reasoning.

Parameter estimation: What are the model parameters?

- ▷ Machine learning – finding parameters based on observations.

Parameter inference for homogenous case



For a sequence of observations $\mathbf{x} = (x_1, \dots, x_n)$ the log-likelihood is

$$\begin{aligned}\ell[\mathbf{x}] &= \log \underbrace{\Pr[x_1]}_{\beta[x_1]} + \sum_{i=1}^{n-1} \log \underbrace{\Pr[x_{i+1}|x_i]}_{\alpha[x_i, x_{i+1}]} \\ &= \log \beta[x_1] + \sum_{u_1, u_2} k(u_1, u_2) \log \alpha[u_1, u_2]\end{aligned}$$

where $k(u_1, u_2)$ is the count of bigrams u_1, u_2 in the sequence \mathbf{x} .

Posterior decomposition

As a result the log-likelihood of unnormalised posterior decomposes into the sum of independent terms

$$\begin{aligned}\log p[\boldsymbol{\alpha}, \boldsymbol{\beta} | \boldsymbol{x}] &= \sum_{u_1} k(u_1) \log \beta[u_1] + \log p(\boldsymbol{\beta}) \\ &+ \sum_{u_1, u_2} k(u_1, u_2) \log \alpha[u_1, u_2] + \sum_{u_1} \log p(\boldsymbol{\alpha}[u_1, \cdot])\end{aligned}$$

where

- ▷ $k(u_1)$ is the count u_1 at the beginning of the observed sequences
- ▷ $k(u_1, u_2)$ is the count of bigrams u_1, u_2 in the observed sequences.
- ▷ $p(\boldsymbol{\beta})$ is the prior for an entire vector of initial probabilities
- ▷ $p(\boldsymbol{\alpha}[u_1, \cdot])$ is the prior for the transition probabilities from u_1

Reduction to the dice throwing experiment

Posterior decomposition leads to many independent optimisation tasks

$$\sum_{u_1} k(u_1) \log \beta[u_1] + \log p(\beta) \rightarrow \max$$

$$\sum_{u_2} k(u_1, u_2) \log \alpha[u_1, u_2] + \log p(\alpha[u_1, \cdot]) \rightarrow \max$$

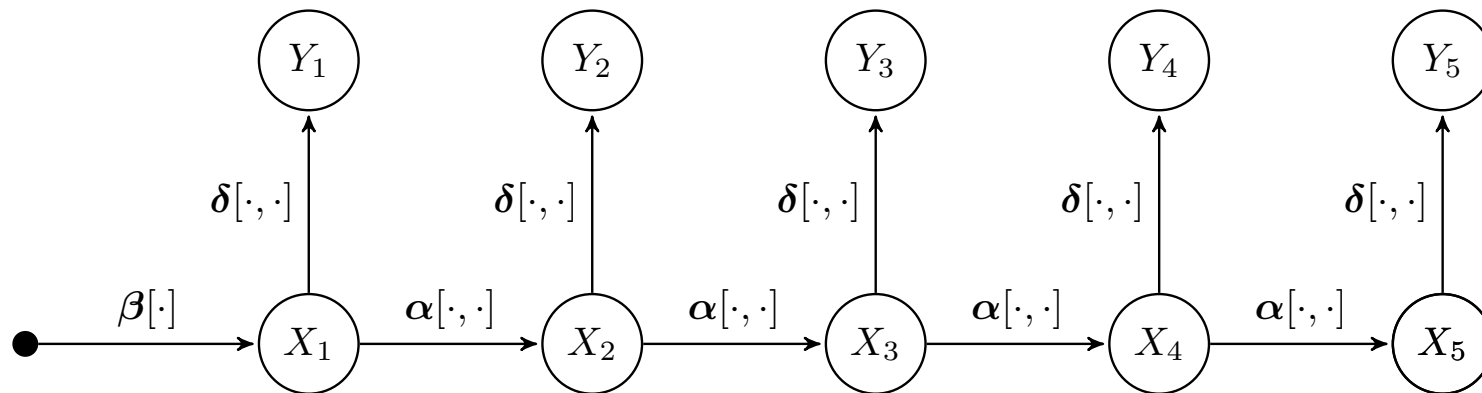
where each of these is equivalent to optimisation of dice throwing posterior. Thus Maximum A posteriori estimates for parameters are

$$\beta[u_1] = \frac{k(u_1) + c}{k(*) + mc} \quad \alpha[u_1, u_2] = \frac{k(u_1, u_2) + c}{k(u_1, *) + mc}$$

where

- ▷ $*$ is a wildcard symbol in the count queries
- ▷ m is the number of states and c is a constant for Laplacian smoothing.

Hidden Markov Model



Definition. Let X_1, X_2, \dots be hidden states that form a Markov chain and let Y_1, Y_2, \dots be observations that the probability of y_i depends only on the state x_i . Then the entire process is known as Hidden Markov Model.

Common tasks

- ▷ parameter estimation
- ▷ filtering, smoothing, prediction

Applications

Modelling and prediction

- ▷ stock prices
- ▷ linear control algorithms

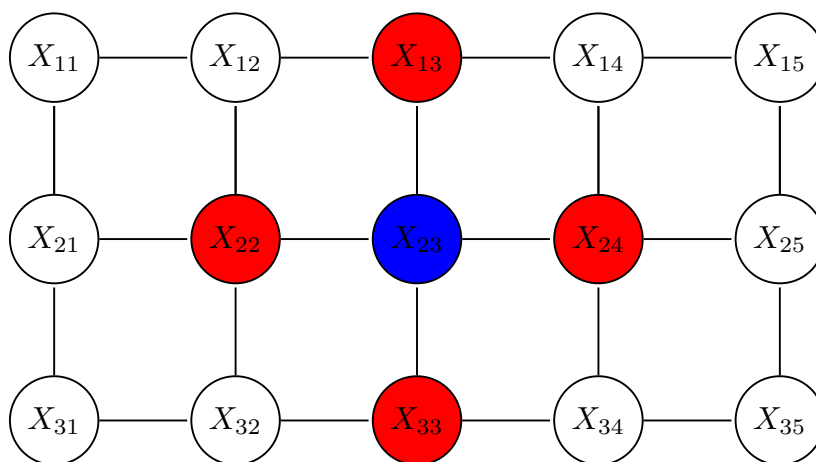
Sequence annotation

- ▷ fraud detection
- ▷ change detection
- ▷ functional motifs of DNA sequences

Decoding

- ▷ speech recognition
- ▷ communication over a noisy channels
- ▷ object tracking and data fusion

Random Markov Fields



Definition. Markov random field is specified by undirected graph connecting random variables X_1, X_2, \dots such that for any node X_i

$$\Pr [x_i | (x_j)_{j \neq i}] = \Pr [x_i | (x_j)_{j \in \mathcal{N}(X_i)}]$$

where the set of neighbours $\mathcal{N}(X_i)$ is also known as *Markov blanket* for X_i .

Hammersley-Clifford theorem

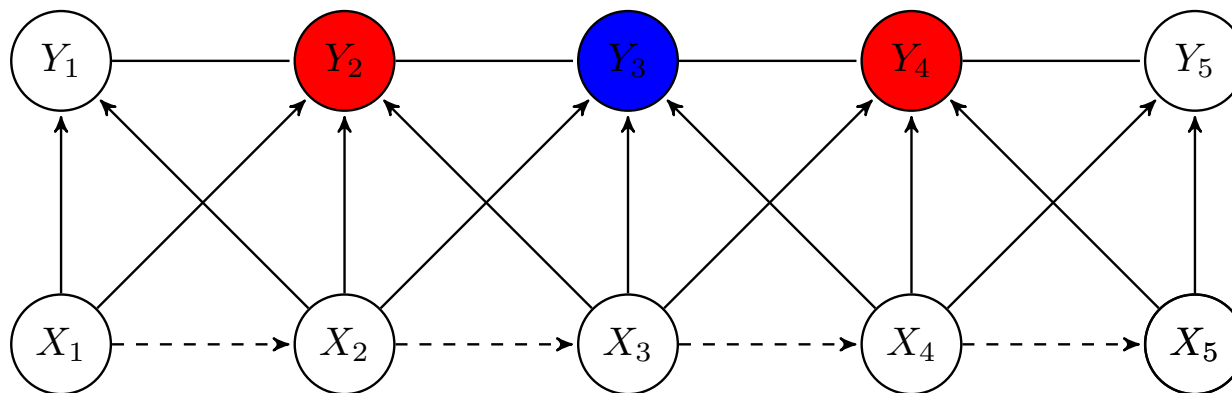
The probability of an observation $\mathbf{x} = (x_1, x_2, \dots)$ generated by a Markov random field can be expressed in the form

$$\Pr[\mathbf{x}] = \frac{1}{Z(\omega)} \cdot \exp \left(- \sum_{c \in \text{MaxClique}} \Psi_c(\mathbf{x}_c, \omega) \right)$$

where

- ▷ $Z(\omega)$ is a normalising constant
- ▷ MaxClique is the set of maximal cliques in the Markov random field
- ▷ Ψ_c is defined on the variables in the clique c

Conditional Random Fields



Definition. Let X_1, X_2, \dots and Y_1, Y_2, \dots be random variables. The entire process is conditional random field if random variables Y_1, Y_2, \dots conditioned for any sequence of observations x_1, x_2, \dots form a Markov random field

$$\Pr [y_i | (x_k)_{k=1}^{\infty}, (y_j)_{j \neq i}] = \Pr [y_i | (x_k)_{k=1}^{\infty}, (y_j)_{j \in \mathcal{N}(Y_i)}]$$

where the set of neighbours $\mathcal{N}(Y_i)$ is a *conditional Markov blanket* for Y_i .

Applications

Standard setting

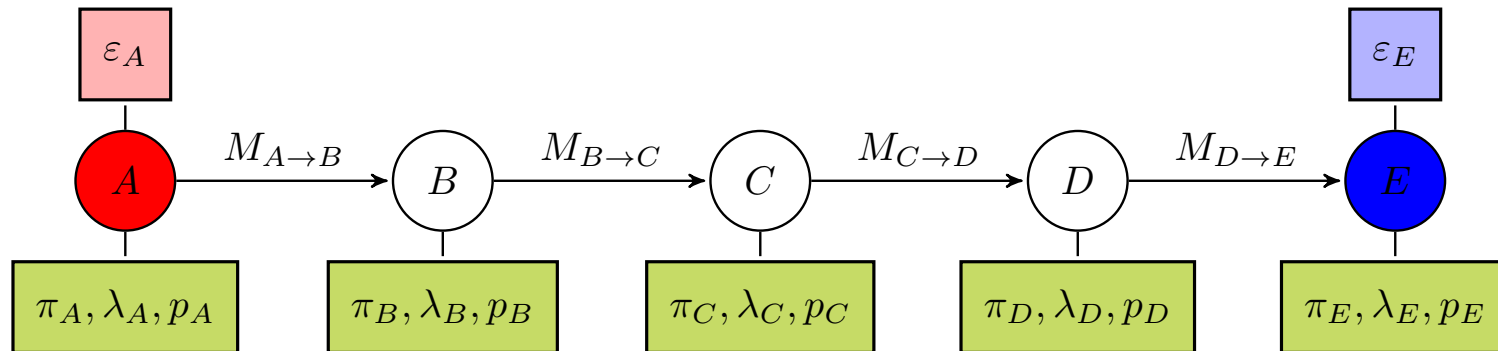
- ▷ The input x is used to predict labels y_1, y_2, \dots
- ▷ A correct label sequence must satisfy possibly unknown restrictions.
- ▷ These restrictions are captured by conditional random random field.

Instantiation

- ▷ Hammersley-Clifford theorem prescribes the format of $\Pr[y|x]$
- ▷ Clique features Ψ_c can depend on $(y_i)_{i \in c}, (x_i)_{i=1}^{\infty}$
- ▷ Features can be defined as linear combination of vertex and edge features.
- ▷ A vertex feature looks only variable y_i associated with the vertex.
- ▷ An edge feature looks only variables y_i, y_j associated with the edge.

Belief propagation

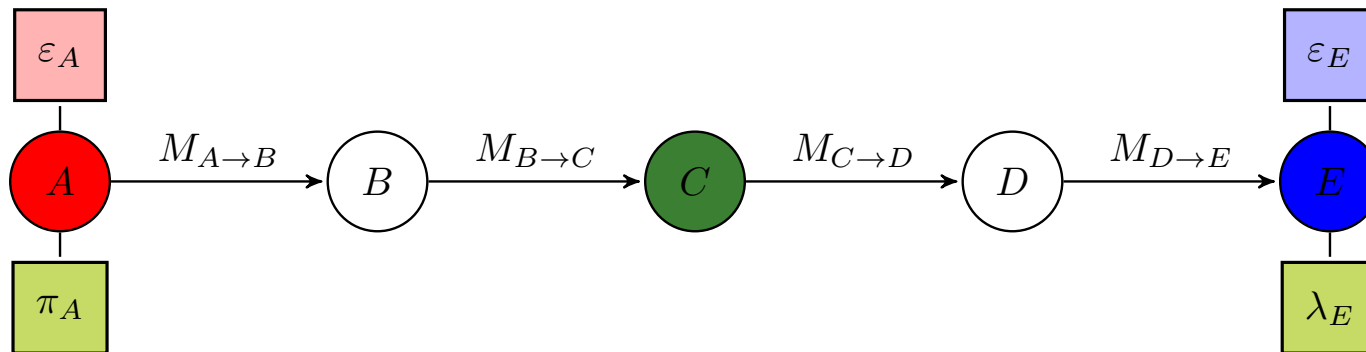
Belief propagation in a simple chain



Inference goal. Given evidence at the ends of the chain find marginal posterior probabilities for each node in the chain.

- ▷ Evidence ε_V is an observational data associated with the node V .
- ▷ Upstream **evidence⁻** is the evidence at the end of chain.
- ▷ Downstream **evidence⁺** is the evidence at the beginning of chain.
- ▷ Attributes π_V, λ_V, p_V are needed to compute marginal distributions.

Initialisation

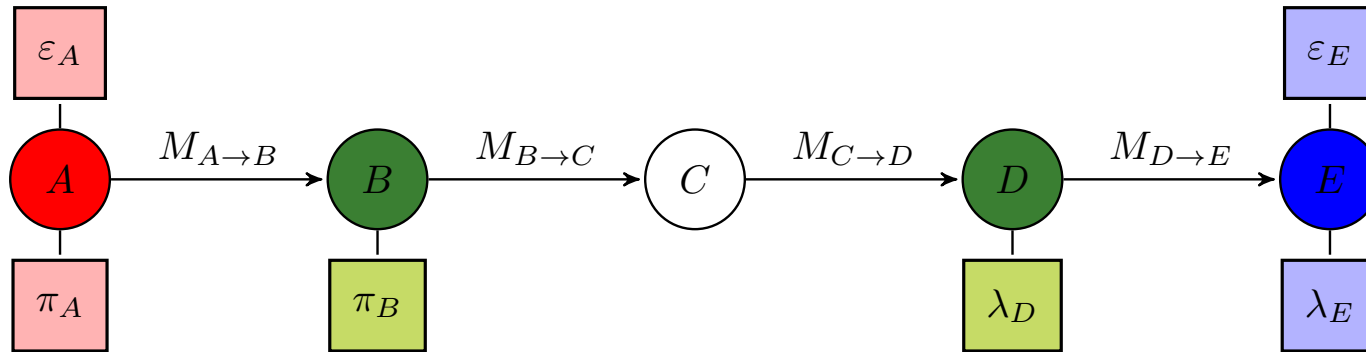


- ▷ Direct evidence ε_V determines the value of V .
- ▷ Indirect evidence ε_V determines the value distribution for V .
- ▷ We can assign the prior for the first and likelihood for the last node

$$\pi_A(a) = \Pr [A = a | \text{evidence}^+] = \Pr [A = a | \varepsilon_A]$$

$$\lambda_E(e) = \Pr [\text{evidence}^- | E = e] = \Pr [\varepsilon_E | E = e]$$

Belief propagation



Inference goal

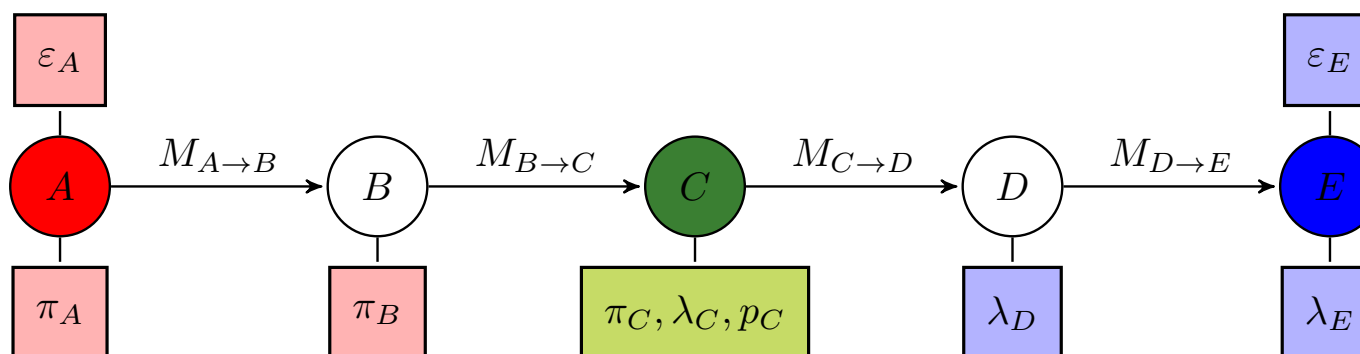
$$\pi_B(b) = \Pr [b | \text{evidence}^+]$$

$$\lambda_D(d) = \Pr [\text{evidence}^- | d]$$

Iterative propagation rules

- ▷ Marginalisation gives an update rule $\lambda_D = M_{D \rightarrow E} \lambda_E$.
- ▷ Marginalisation gives an update rule $\pi_B \propto \pi_A M_{A \rightarrow B}$.

Belief propagation



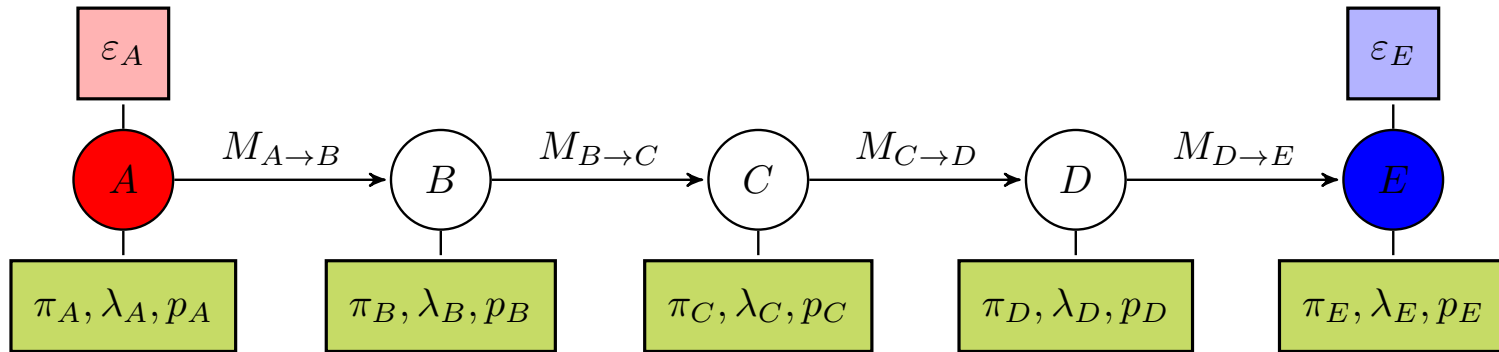
Inference goal

$$p_C(c) = \Pr [c | \text{evidence}^+, \text{evidence}^-]$$

Iterative update rule

- ▷ Bayes formula gives $p_C \propto \pi_C \otimes \lambda_C$.

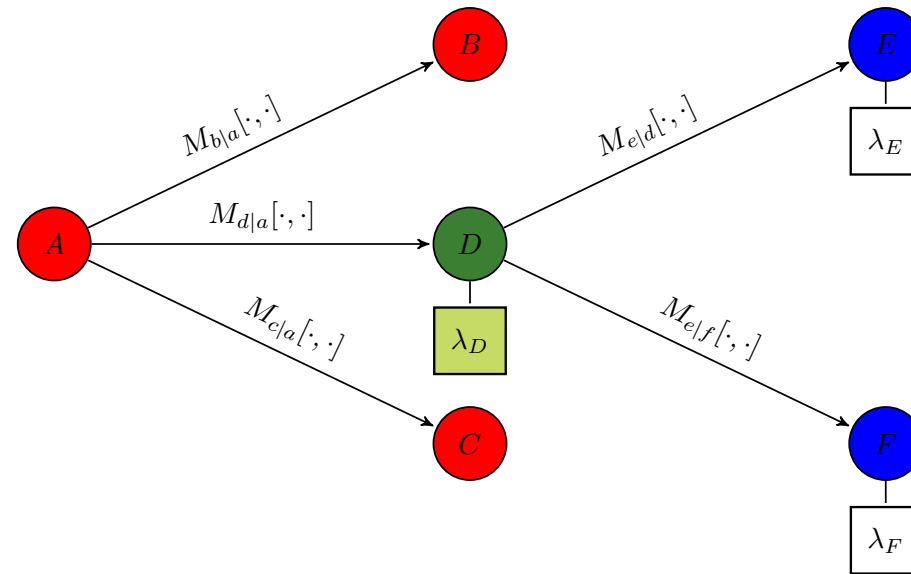
Belief propagation in a simple tree



Inference goal. Given evidence at the ends of the leafs and the root of tree find marginal posterior probabilities for each node in the tree.

- ▷ Evidence ε_V is an observational data associated with the node V .
- ▷ Upstream **evidence⁻** is the evidence at the end of chain.
- ▷ Downstream **evidence⁺** is the evidence at the beginning of chain.
- ▷ Attributes π_V, λ_V, p_V are needed to compute marginal distributions.

Likelihood propagation in a tree

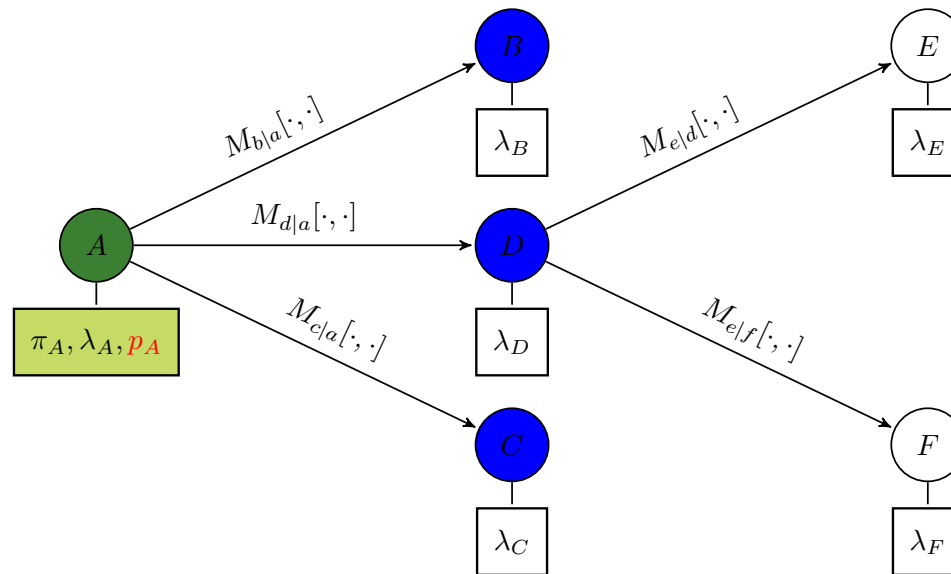


$$\lambda_D(d) = \Pr [\text{evidence}^-(D) | d]$$

Iterative propagation rules

- ▷ Independence gives a pooling rule $\lambda_D = \lambda_1 \otimes \lambda_2$
- ▷ Marginalisation gives rules $\lambda_1 = M_{e|d}\lambda_E$ and $\lambda_2 = M_{f|d}\lambda_F$.

Prior propagation in a tree



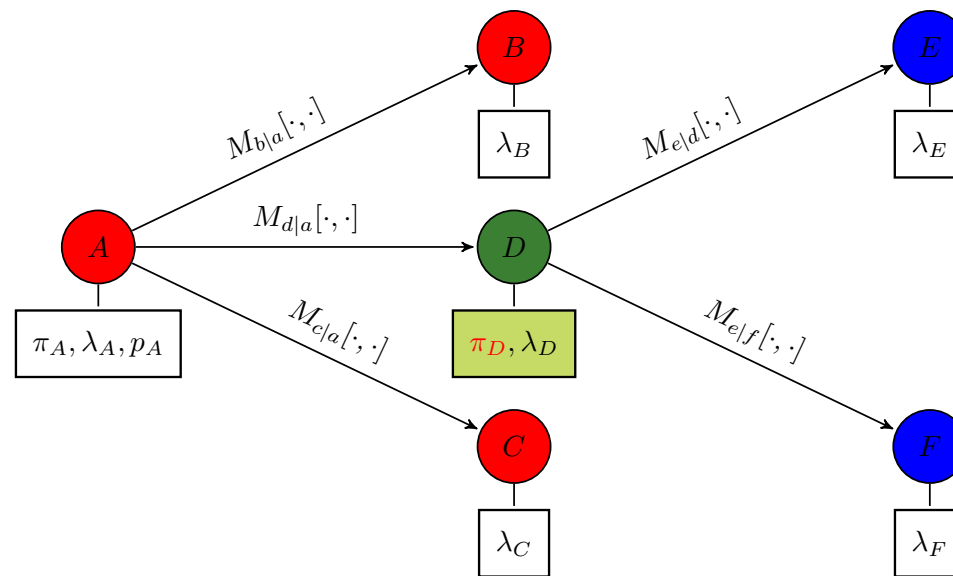
$$\pi_A(a) = \Pr [a | \text{evidence}^+(A)]$$

$$p_A(a) = \Pr [a | \text{evidence}^+(A), \text{evidence}^-(A)]$$

Evidence pooling

▷ Marginal conditional probability $p_A \propto \pi_A \otimes \lambda_A$

Prior propagation in a tree



$$\pi_D(d) = \Pr [d | \text{evidence}^+(D)]$$

$$\Pr [a | \text{evidence}^+(D)] = \Pr [a | \text{evidence}^+(A), \text{evidence}^-(A) \setminus \{D\}]$$

Iterative propagation rules

▷ Prior component can be updated $\pi_D \propto \pi_A M_{d|a} \otimes M_{b|a} \lambda_B \otimes M_{c|a} \lambda_C$.