LTAT.02.004 Machine Learning II

Graphical models

Sven Laur
University of Tartu

Discrete random variables

- A *random variable* X with possible *outcomes* $x \in \mathrm{supp}(X)$
- 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 $X_1 \ldots X_m \perp Y_1, \ldots Y_n$:

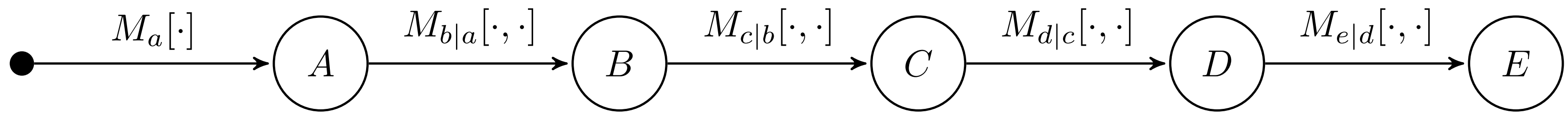
$$\Pr[x_1 \wedge \ldots \wedge x_m \wedge y_1 \wedge \ldots \wedge y_n] = \Pr[x_1 \wedge \ldots \wedge x_m] \cdot \Pr[y_1 \wedge \ldots \wedge y_n]$$

- Marginalisation over variables $Y_1, \ldots, Y_n$:

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

Common models

Markov chain



Definition. Let $X_1, X_2, \ldots$ 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 $\mathcal{S}_1 \ldots, \mathcal{S}_n$
- ▷ 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?
$\triangleright$ Synthesis of time-series, textures, sounds, games movements.

Stationary distribution: What happens if we run the chain infinitely long?
$\triangleright$ Getting samples from an unnormalised posterior, optimisation tasks.

Likelihood estimation: What is a probability of an observation $x_1, \ldots, x_n$?
$\triangleright$ Reasoning about probabilities and clustering sequences.

Decoding: What is the most probable outcome $x_1, \ldots, x_n$?
$\triangleright$ Imputing missing values. Rudimentary logical reasoning.

Parameter estimation: What is are the model parameters?
$\triangleright$ Machine learning – finding parameters based on observations.

Parameter inference for homogenous case

$$\bullet \xrightarrow{\boldsymbol{\beta}[\cdot]} X_1 \xrightarrow{\boldsymbol{\alpha}[\cdot,\cdot]} X_2 \xrightarrow{\boldsymbol{\alpha}[\cdot,\cdot]} X_3 \xrightarrow{\boldsymbol{\alpha}[\cdot,\cdot]} X_4 \xrightarrow{\boldsymbol{\alpha}[\cdot,\cdot]} X_5$$

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

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

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

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

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

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

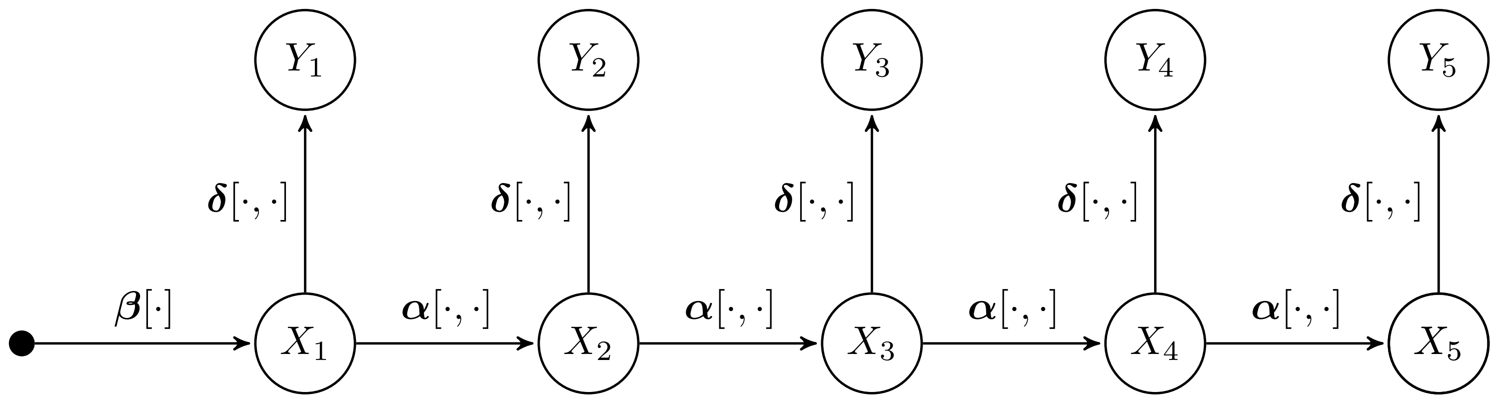
where each of these is equivalent to optimisation of dice throwing posterior. Thus Maximum Aposteriori estimates for parameters are

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

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 $X_1, X_2, \ldots$ be hidden states that form a Markov chain and let $Y_1, Y_2, \ldots$ 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

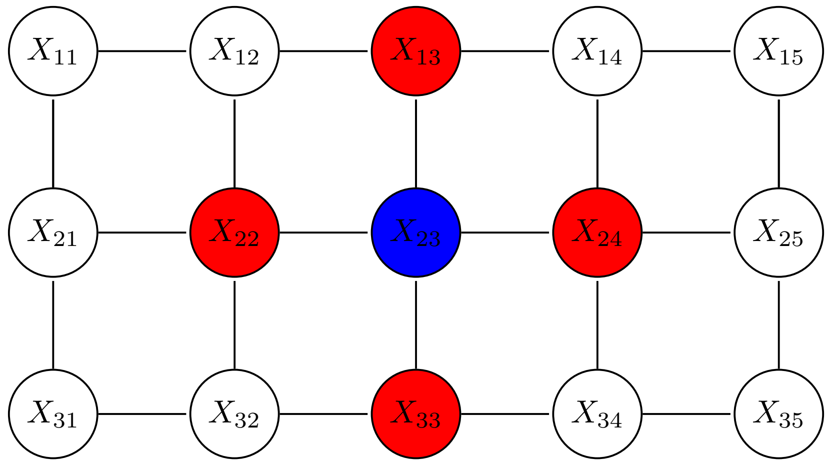
Sequence annotation

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

Decoding

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

Random Markov Fields



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

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

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

Hammersley-Clifford theorem

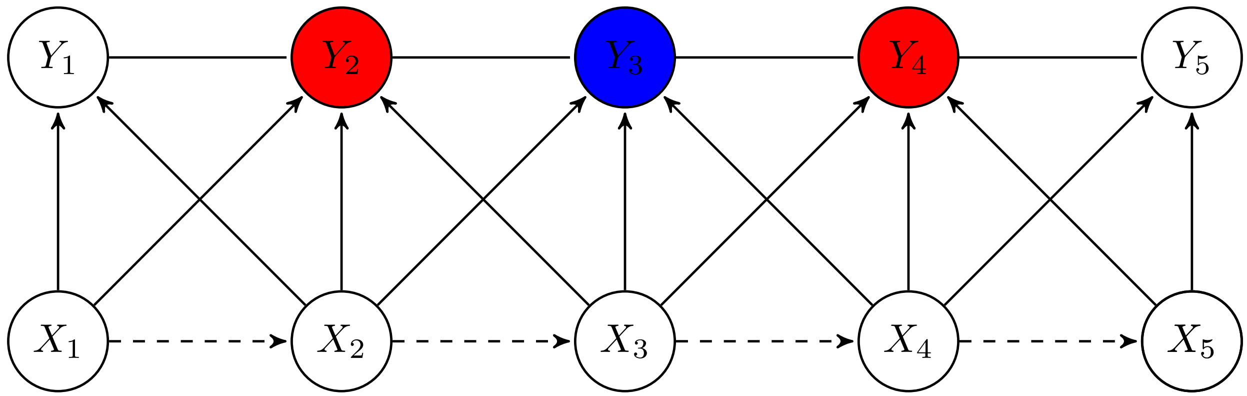
The probability of an observation $\boldsymbol{x} = (x_1, x_2, \ldots)$ generated by a Markov random field can be expressed in the form

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

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

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

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

Applications

Standard setting

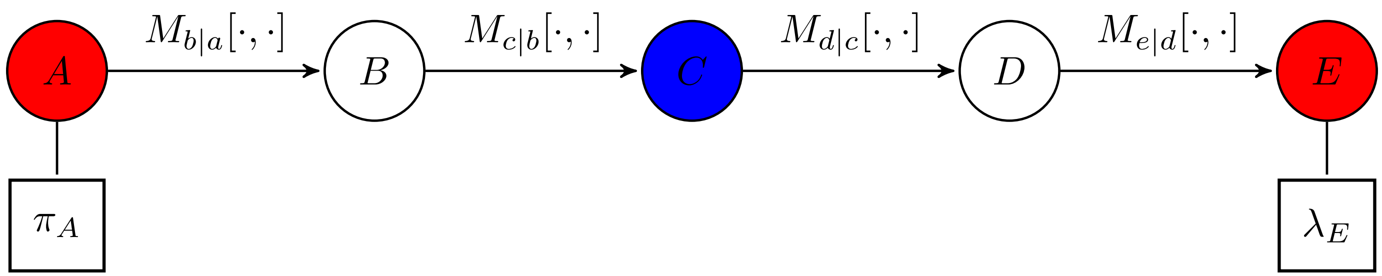
- The input $\boldsymbol{x}$ is used to predict labels $y_1, y_2, \ldots$.
- A correct label sequence must satisfy possibly unknown restrictions.
- These restrictions are captured by conditional random random field.

Instantiation

- Hammersley-Clifford theorem prescribes the format of $\Pr\left[\boldsymbol{y}|\boldsymbol{x}\right]$
- 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 chain



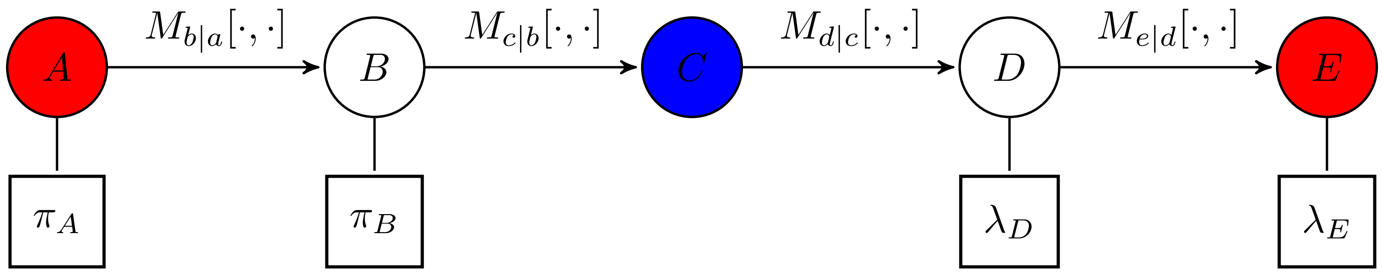
Evidence

- ▷ We know the values a and e for nodes A and E.
- ▷ We know a value distribution for nodes A and E.

Representation

- ▷ A prior vector π_A will represent value distribution in A.
- ▷ A likelihood vector λ_E will represent value distribution in E.

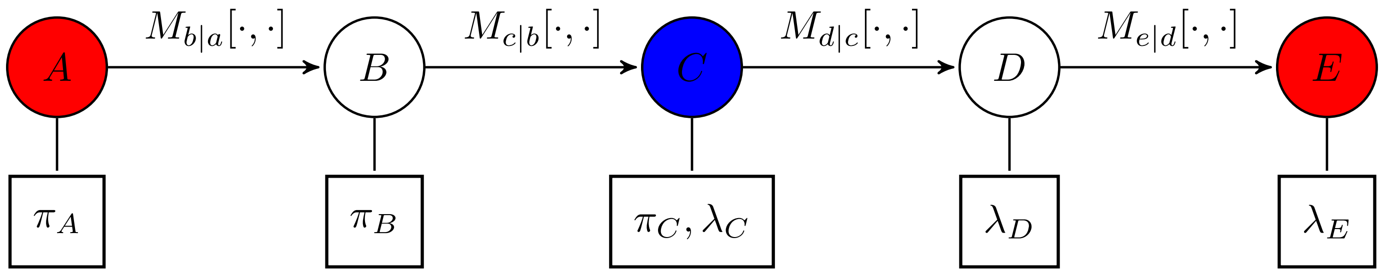
Belief propagation in a chain



Iterative propagation

- Marginalisation gives an update rule $\lambda_D = M_{e|d}\lambda_E$.
- Marginalisation gives an update rule $\pi_B \propto \pi_A M_{a|b}$.

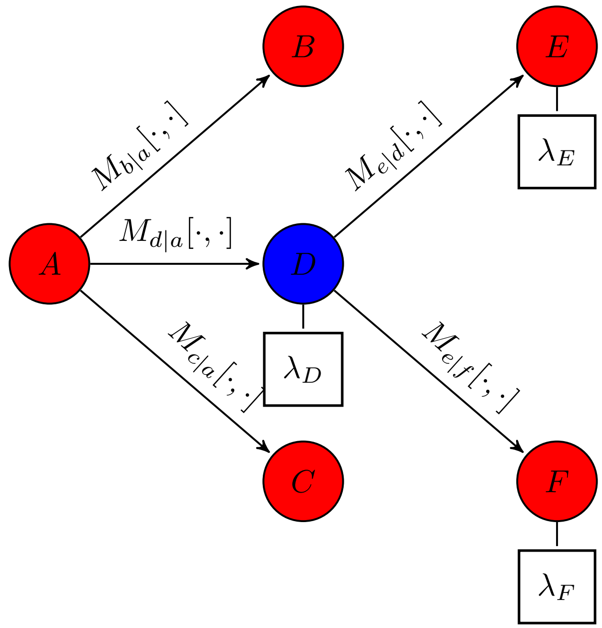
Belief propagation in a chain



Evidence pooling

▷ Marginal conditional probability $p_c = \Pr[c|\text{evidence}] \propto \pi_C \lambda_C$

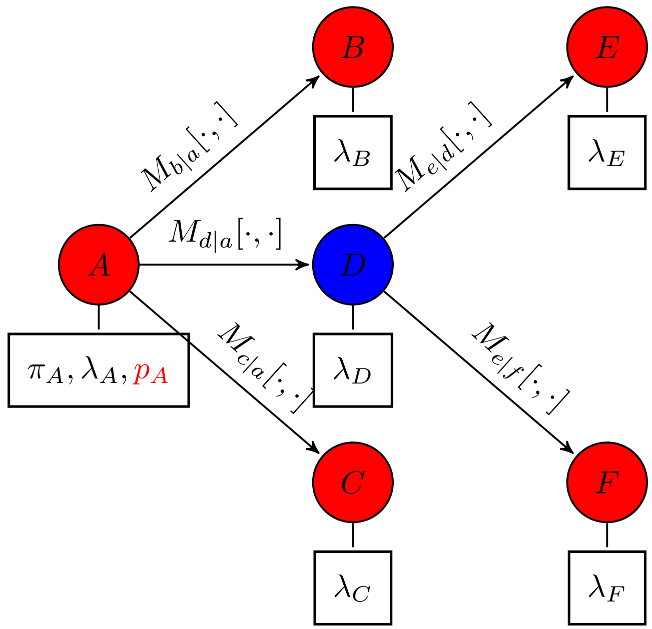
Likelihood propagation in a tree



Iterative propagation

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

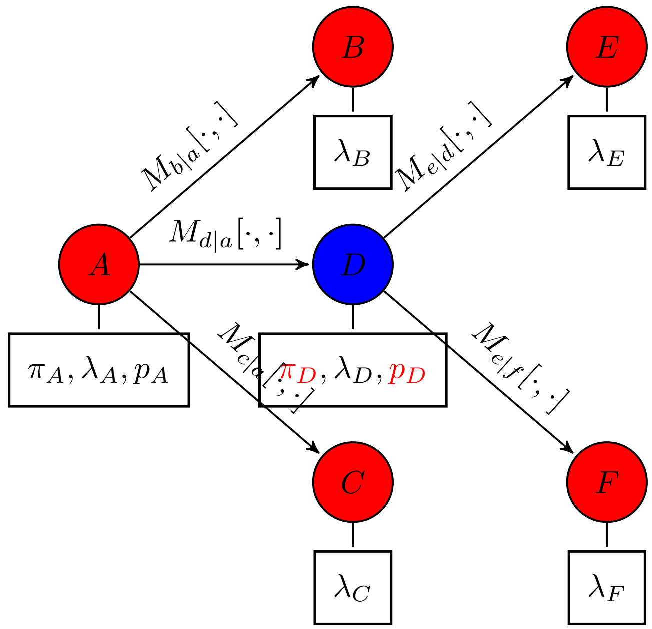
Prior propagation in a tree



Evidence pooling

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

Prior propagation in a tree



Iterative propagation

▷ Prior component can be updates $\pi_D \propto \frac{p_A}{M_{d|a}\lambda_D}$