Universität Potsdam
Institut für Informatik
Lehrstuhl Maschinelles Lernen

Graphical Models

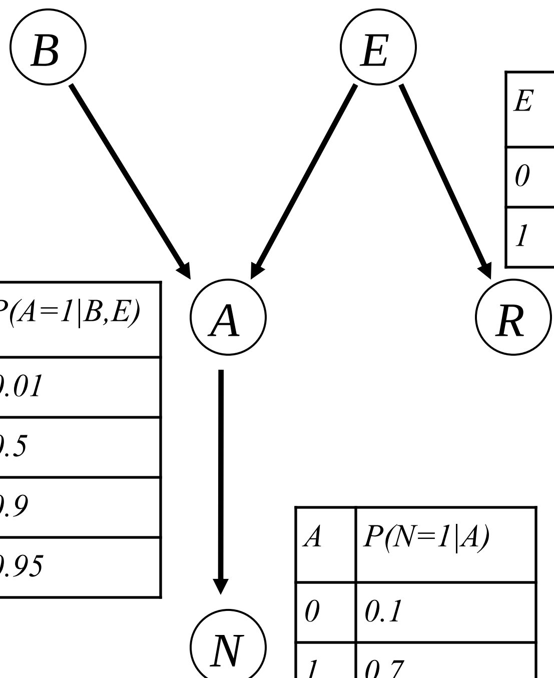
Niels Landwehr

Agenda

- Graphical models: syntax and semantics.
- Inference in graphical models (exact, approximate)
- Graphical models in machine learning.

Recap: Graphical Models

- Graphical model for „Alarm“ scenario



P(B=1)
0.1

P(E=1)
0.2

E	P(R=1|E)
0	0.01
1	0.5

B	E	P(A=1|B,E)
0	0	0.01
0	1	0.5
1	0	0.9
1	1	0.95

A	P(N=1|A)
0	0.1
1	0.7

Distribution modeled:

$$p(B,E,A,N,R) = p(B)\,p(E)\,p(A \mid E,B)\,p(N \mid A)\,p(R \mid E)$$

Graphical model:

- There is one node for each random variable
- For each factor of the form $p(X \mid X_1,\dots,X_k)$ there is a directed edge from the X_i to X in the graph
- Model is parameterized with conditional distributions $p(X \mid X_1,\dots,X_k)$

Recap: Problem Setting Inference

- Given: graphical model over random variables $\{X_1, \ldots, X_N\}$.

- Problem setting inference:
  - Variables with evidence $X_{i_1}, \ldots, X_{i_m}$ $\{i_1, \ldots, i_m\} \subseteq \{1, \ldots, N\}$
  - Query variable X_a $a \in \{1, \ldots, N\} \setminus \{i_1, \ldots, i_m\}$

  - Task: compute distribution over query variable given evidence.

Conditional distribution over random variable X_a

Evidence: observed values for variables $X_{i_1}, \ldots, X_{i_m}$

Compute $p(x_a \mid x_{i_1}, \ldots, x_{i_m})$

More generally also $p(x_{a_1}, \ldots, x_{a_k} \mid x_{i_1}, \ldots, x_{i_m})$

Recap: Message Passing Algorithm

- Algorithm: Message Passing on a linear chain
  - Input:

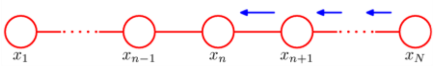
$$p(x_1,\ldots,x_N)=\psi_{1,2}(x_1,x_2),\ldots.,\psi_{N-1,N}(x_{N-1},x_N)$$

Query: $p(x_a)=?$

  - Recursively compute messages:

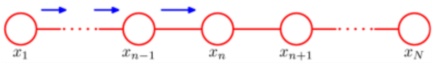
$$\mu_\beta(x_N)=\mathbf{1}$$

For $k=N$ -1,…, a: $\mu_\beta(x_k)=\sum_{x_{k+1}}\psi_{k,k+1}(x_k,x_{k+1})\mu_\beta(x_{k+1})$



$$\mu_\alpha(x_1)=\mathbf{1}$$

For $k=2,\ldots,a$: $\mu_\alpha(x_k)=\sum_{x_{k-1}}\psi_{k-1,k}(x_{k-1},x_k)\mu_\alpha(x_{k-1})$

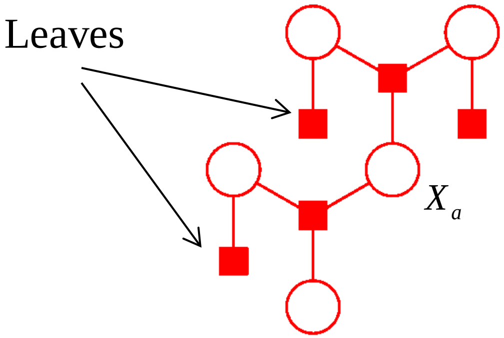


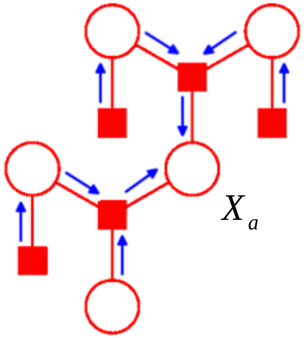
  - Output:

$p(x_a)=\mu_\alpha(x_a)\mu_\beta(x_a)$ (function of x_a, that is, distribution over x_a)

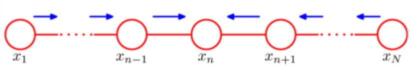
Recap: Inference on Factor Graphs

- If the orginial graph was a polytree, the resulting factor graph is an undirected tree (that is, it has no cycles).





Special case linear chain:



- Inference is then carried out on factor graph:
  - Take the query node X_a as the root of the undirected tree.
  - Send messages from the leaves to the root (there is always a unique path, because factor graph is undirected tree).
  - There are now two types of messages: factor messages and variable messages.

Agenda

- Graphical models: syntax and semantics.
- Inference in graphical models
  - Exact inference
  - Approximate inference
- Graphical models in machine learning.

Approximate Inference

- Exact inference in general graphical models is NP-hard.
- In practice, *approximate* inference algorithms therefore play an important role.

- We look at sampling-based approximate inference
  - Relatively easy to understand/implement.
  - Anytime algorithms (the longer the algorithm runs, the more accurate the result).

Sampling-based Inference

- General idea sampling:
  - We are interested in a distribution $p(\mathbf{z})$, where $\mathbf{z}$ is a set of random variables (e.g. conditional distribution over query variables in graphical model).
  - It is difficult to compute $p(\mathbf{z})$ directly.
  - Instead, we will generate „samples“

$$\mathbf{z}^{(k)} \sim p(\mathbf{z}) \qquad \text{i.i.d.,} \quad k = 1, \dots, K,$$

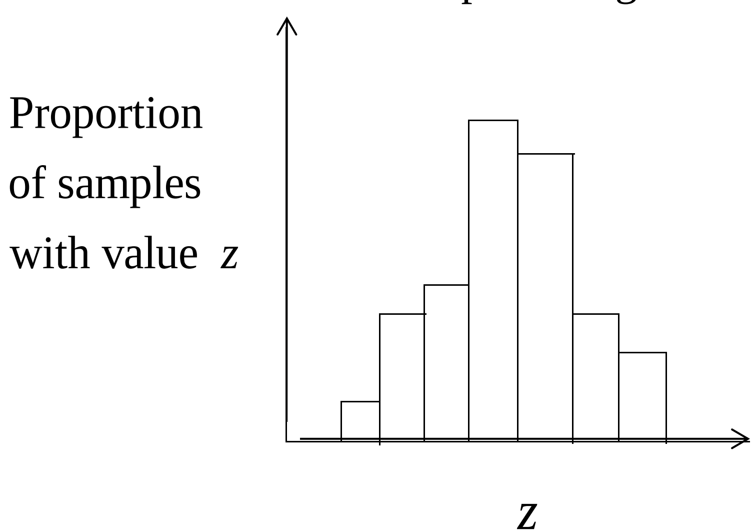
    every sample $\mathbf{z}^{(k)}$ completely assigns values to the random variables in $\mathbf{z}$.

- The samples $\mathbf{z}^{(1)}, \mathbf{z}^{(2)}, \dots, \mathbf{z}^{(K)}$ approximate the distribution $p(\mathbf{z})$.
- It is often easier to design a procedure for generating the $\mathbf{z}^{(k)}$ than it is to compute $p(\mathbf{z})$.

Sampling-based Inference

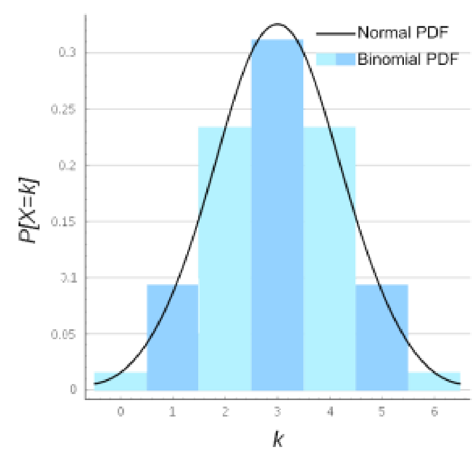
- Example:
  - One-dimensional distribution, $\mathbf{z} = \{z\}$.
  - Discrete variable with states {0,…,6}: number of „Heads“ from 6 coin tosses.
  - Tossing a coin 6 times gives us one sample.
  - K=100 experiments, with 6 coin tosses each.

Sample histogram



$$\xrightarrow{K \to \infty}$$

True distribution (Binomial)



Sampling Inference for Graphical Models

- Given a graphical model that represents a distribution by

$$p(x_1,\dots,x_N) = \prod_{i=1}^{N} p(x_i \mid pa(x_i)).$$

- Slightly more general problem setting: set of query variables

$$p(\mathbf{x}_A \mid \mathbf{x}_D) \;\approx\; ?$$

$\mathbf{x}_A \subseteq \mathbf{x} = \{x_1,\dots,x_N\}$ set of query variables

$\mathbf{x}_D \subseteq \mathbf{x} = \{x_1,\dots,x_N\}$ set of evidence variables

- Distribution $p(\mathbf{x}_A \mid \mathbf{x}_D)$ will be approximated by a set of samples.

- We first look at inference without evidence:

$$p(\mathbf{x}_A) \approx ? \qquad \mathbf{x}_A = \{x_{a_1},\dots,x_{a_m}\} \subseteq \{x_1,\dots,x_N\}$$

Sampling Inference for Graphical Models

- Goal: Drawing samples from marginal distribution $p(\mathbf{x}_A) = p(x_{a_1},\ldots,x_{a_m})$.

$$\mathbf{x}_A^{(k)} \sim p(\mathbf{x}_A) \qquad k = 1,\ldots,K$$

- It suffices to draw samples from the joint distribution $p(\mathbf{x}) = p(x_1,\ldots,x_N)$:

$$\mathbf{x}^{(k)} = (x_1^{(k)},\ldots,x_N^{(k)}) \sim p(x_1,\ldots,x_N) \qquad k = 1,\ldots,K$$

- We obtain samples from the marginal distribution $p(x_{a_1},\ldots,x_{a_m})$ simply by projecting to the $\{x_{a_1},\ldots,x_{a_m}\}$.

$$\mathbf{x}^{(k)} = (x_1^{(k)},\ldots,x_N^{(k)}) \sim p(x_1,\ldots,x_N) \qquad k = 1,\ldots,K$$

$$\Big\Downarrow \text{projection}$$

$$\mathbf{x}_A^{(k)} = (x_{a_1}^{(k)},\ldots,x_{a_m}^{(k)}) \sim p(x_{a_1},\ldots,x_{a_m}) \qquad k = 1,\ldots,K$$

Inference: Ancestral Sampling

- How do we generate samples $\mathbf{x}^{(k)} \sim p(\mathbf{x})$?

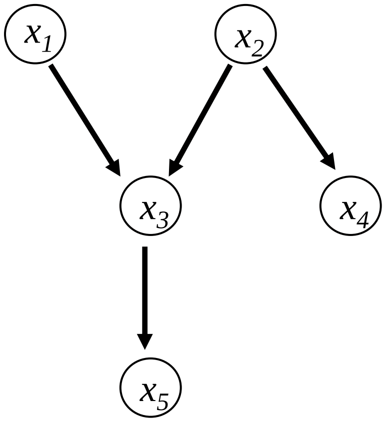
- Easy for directed graphical models: „Ancestral Sampling“
  - Exploit factorization of joint distribution

$$\mathbf{x}^{(k)} \sim p(\mathbf{x}) = p(x_1, \ldots, x_N)$$

$$= \prod_{i=1}^{N} p(x_i \mid pa(x_i))$$

Draw each new variable given states of previous variables

„Draw following the edges“



  - „Draw following the edges“

Inference: Ancestral Sampling

- We draw a sample $\mathbf{x}^{(k)} = (x_1^{(k)}, \ldots, x_N^{(k)})$ by successively drawing the individual $x_i^{(k)}$

$$x_1^{(k)} \sim p(x_1)$$
$$x_2^{(k)} \sim p(x_2 \mid pa(x_2))$$
$$\ldots$$
$$x_N^{(k)} \sim p(x_N \mid pa(x_N))$$

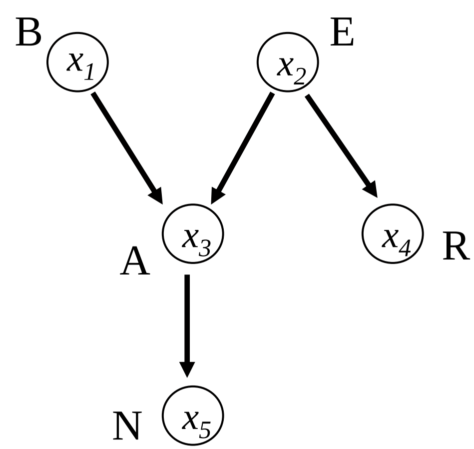
Already drawn values

$$\mathbf{x}^{(k)} \sim p(\mathbf{x}) = \prod_{i=1}^{N} p(x_i \mid pa(x_i))$$

Topological ordering: $pa(x_i) \subseteq \{x_1, \ldots, x_{i-1}\}$

- Example

$$x_1^{(k)} \sim p(x_1) \qquad \rightarrow x_1 = 1$$
$$x_2^{(k)} \sim p(x_2) \qquad \rightarrow x_2 = 0$$
$$x_3^{(k)} \sim p(x_3 \mid x_1 = 1, x_2 = 0) \qquad \rightarrow x_3 = 1$$
$$x_4^{(k)} \sim p(x_4 \mid x_2 = 0) \qquad \rightarrow x_4 = 0$$
$$x_5^{(k)} \sim p(x_5 \mid x_3 = 1) \qquad \rightarrow x_5 = 1$$



Inference: Ancestral Sampling

- We draw a sample $\mathbf{x}^{(k)} = (x_1^{(k)}, \ldots, x_N^{(k)})$ by successively drawing the individual $x_i^{(k)}$

$$x_1^{(k)} \sim p(x_1)$$
$$x_2^{(k)} \sim p(x_2 \mid pa(x_2))$$
$$\ldots$$
$$x_N^{(k)} \sim p(x_N \mid p$$

Already drawn values

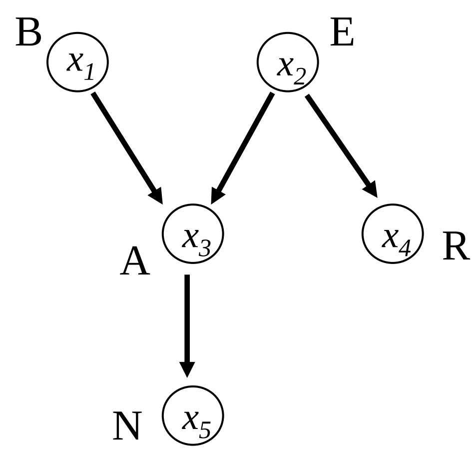
$$\mathbf{x}^{(k)} \sim p(\mathbf{x}) = \prod_{i=1}^{N} p(x_i \mid pa(x_i))$$

Topological ordering: $pa(x_i) \subseteq \{x_1, \ldots, x_{i-1}\}$

P(B=1)
0.1

- Example

$$x_1^{(k)} \sim p(x_1) \qquad \rightarrow x_1 = 1$$
$$x_2^{(k)} \sim p(x_2) \qquad \rightarrow x_2 = 0$$
$$x_3^{(k)} \sim p(x_3 \mid x_1 = 1, x_2 = 0) \qquad \rightarrow x_3 = 1$$
$$x_4^{(k)} \sim p(x_4 \mid x_2 = 0) \qquad \rightarrow x_4 = 0$$
$$x_5^{(k)} \sim p(x_5 \mid x_3 = 1) \qquad \rightarrow x_5 = 1$$



Inference: Ancestral Sampling

- We draw a sample $\mathbf{x}^{(k)} = (x_1^{(k)}, \ldots, x_N^{(k)})$ by successively drawing the individual $x_i^{(k)}$

$$x_1^{(k)} \sim p(x_1)$$
$$x_2^{(k)} \sim p(x_2 \mid pa(x_2))$$
$$\ldots$$
$$x_N^{(k)} \sim p(x_N \mid pa(x_N))$$

Already drawn values

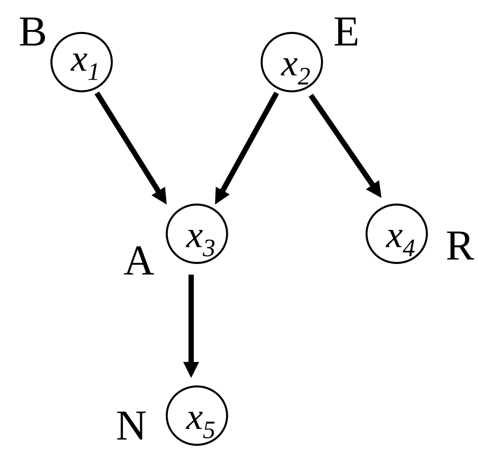
$$\mathbf{x}^{(k)} \sim p(\mathbf{x}) = \prod_{i=1}^{N} p(x_i \mid pa(x_i))$$

Topological ordering: $pa(x_i) \subseteq \{x_1, \ldots, x_{i-1}\}$

- Example

P(E=1)
0.2

$$x_1^{(k)} \sim p(x_1) \qquad \rightarrow x_1 = 1$$
$$x_2^{(k)} \sim p(x_2) \qquad \rightarrow x_2 = 0$$
$$x_3^{(k)} \sim p(x_3 \mid x_1 = 1, x_2 = 0) \qquad \rightarrow x_3 = 1$$
$$x_4^{(k)} \sim p(x_4 \mid x_2 = 0) \qquad \rightarrow x_4 = 0$$
$$x_5^{(k)} \sim p(x_5 \mid x_3 = 1) \qquad \rightarrow x_5 = 1$$



Inference: Ancestral Sampling

- We draw a sample $\mathbf{x}^{(k)} = (x_1^{(k)}, \ldots, x_N^{(k)})$ by successively drawing the individual $x_i^{(k)}$

$$x_1^{(k)} \sim p(x_1)$$
$$x_2^{(k)} \sim p(x_2 \mid pa(x_2))$$
$$\ldots$$
$$x_N^{(k)} \sim p(x_N \mid pa$$

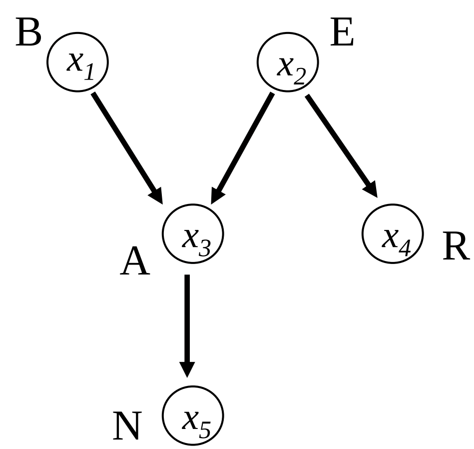
$$\mathbf{x}^{(k)} \sim p(\mathbf{x}) = \prod_{i=1}^{N} p(x_i \mid pa(x_i))$$

Topological ordering: $pa(x_i) \subseteq \{x_1, \ldots, x_{i-1}\}$

- Example

B	*E*	*P(A=1|B,E)*
0	*0*	*0.01*
0	*1*	*0.5*
1	*0*	*0.9*
1	*1*	*0.95*

$$x_1^{(k)} \sim p(x_1)$$
$$x_2^{(k)} \sim p(x_2) \qquad \to x_2 = 0$$
$$x_3^{(k)} \sim p(x_3 \mid x_1 = 1, x_2 = 0) \qquad \to x_3 = 1$$
$$x_4^{(k)} \sim p(x_4 \mid x_2 = 0) \qquad \to x_4 = 0$$
$$x_5^{(k)} \sim p(x_5 \mid x_3 = 1) \qquad \to x_5 = 1$$



Inference: Ancestral Sampling

- We draw a sample $\mathbf{x}^{(k)} = (x_1^{(k)}, \ldots, x_N^{(k)})$ by successively drawing the individual $x_i^{(k)}$

$$x_1^{(k)} \sim p(x_1)$$
$$x_2^{(k)} \sim p(x_2 \mid pa(x_2))$$
$$\ldots$$
$$x_N^{(k)} \sim p(x_N \mid pa$$

Already drawn

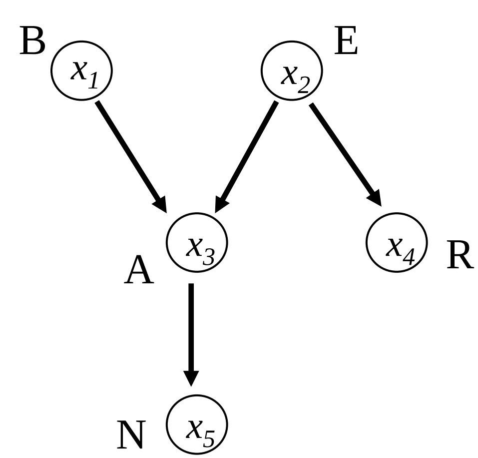
$$\mathbf{x}^{(k)} \sim p(\mathbf{x}) = \prod_{i=1}^{N} p(x_i \mid pa(x_i))$$

Topological ordering: $pa(x_i) \subseteq \{x_1, \ldots, x_{i-1}\}$

- Example

E	*P(R=1|E)*
0	*0.01*
1	*0.5*

$$x_1^{(k)} \sim p(x_1)$$
$$x_2^{(k)} \sim p(x_2)$$
$$x_3^{(k)} \sim p(x_3 \mid x_1 = 1, x_2 = 0) \quad \rightarrow x_3 = 1$$
$$x_4^{(k)} \sim p(x_4 \mid x_2 = 0) \quad \rightarrow x_4 = 0$$
$$x_5^{(k)} \sim p(x_5 \mid x_3 = 1) \quad \rightarrow x_5 = 1$$



Inference: Ancestral Sampling

- We draw a sample $\mathbf{x}^{(k)} = (x_1^{(k)}, \ldots, x_N^{(k)})$ by successively drawing the individual $x_i^{(k)}$

$$x_1^{(k)} \sim p(x_1)$$
$$x_2^{(k)} \sim p(x_2 \mid pa(x_2))$$
$$\ldots$$
$$x_N^{(k)} \sim p(x_N \mid pa(x_N))$$

Already drawn values

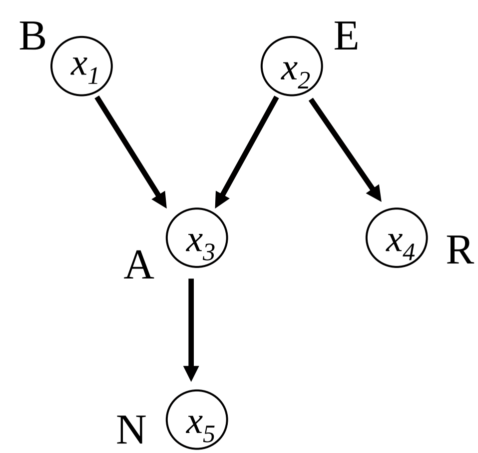
$$\mathbf{x}^{(k)} \sim p(\mathbf{x}) = \prod_{i=1}^{N} p(x_i \mid pa(x_i))$$

Topological ordering: $pa(x_i) \subseteq \{x_1, \ldots, x_{i-1}\}$

- Example

$$x_1^{(k)} \sim p(x_1)$$
$$x_2^{(k)} \sim p(x_2)$$
$$x_3^{(k)} \sim p(x_3 \mid x_1 =$$
$$x_4^{(k)} \sim p(x_4 \mid x_2 = \ldots) \qquad \rightarrow x_4 = 0$$
$$x_5^{(k)} \sim p(x_5 \mid x_3 = 1) \qquad \rightarrow x_5 = 1$$

A	P(N=1|A)
0	0.1
1	0.7



Inference: Ancestral Sampling

- We draw a sample $\mathbf{x}^{(k)} = (x_1^{(k)},\ldots,x_N^{(k)})$ by successively drawing the individual $x_i^{(k)}$

$$x_1^{(k)} \sim p(x_1)$$
$$x_2^{(k)} \sim p(x_2 \mid pa(x_2))$$
$$\ldots$$
$$x_N^{(k)} \sim p(x_N \mid pa(x_N))$$

Already drawn values

- Example

$$x_1^{(k)} \sim p(x_1) \qquad \to x_1 = 1$$
$$x_2^{(k)} \sim p(x_2) \qquad \to x_2 = 0$$
$$x_3^{(k)} \sim p(x_3 \mid x_1 = 1, x_2 = 0) \qquad \to x_3 = 1$$
$$x_4^{(k)} \sim p(x_4 \mid x_2 = 0) \qquad \to x_4 = 0$$
$$x_5^{(k)} \sim p(x_5 \mid x_3 = 1) \qquad \to x_5 = 1$$

$$\Rightarrow \quad \mathbf{x}^{(k)} = (1,0,1,0,1)$$

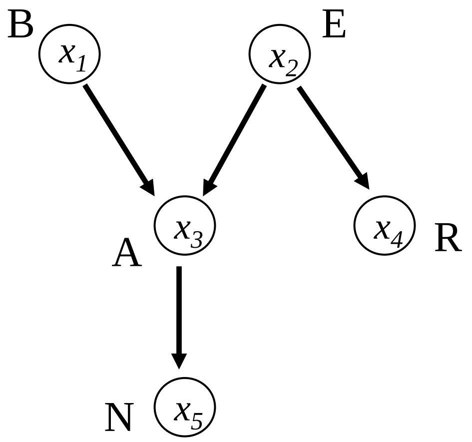
Example: Ancestral Sampling

- Example for estimation of marginal distribution from samples:

$$\mathbf{x}^{(1)} = (1,0,1,0,1)$$
$$\mathbf{x}^{(2)} = (0,0,0,0,0)$$
$$\mathbf{x}^{(3)} = (0,1,0,1,0)$$
$$\mathbf{x}^{(4)} = (0,1,1,0,1)$$
$$\mathbf{x}^{(5)} = (0,0,0,0,0)$$

$$\Rightarrow$$

$$p(x_3 = 1) \approx 0.4$$
$$p(x_4 = 1) \approx 0.2$$
$$p(x_5 = 1) \approx 0.4$$



- Analysis of Ancestral Sampling
  - + Directly draws from the right distribution.
  - + Efficient.
  - + Works for any graph structure.
  - - Only works without evidence.

Inference: Logic Sampling

- How do we obtain samples conditioned on evidence?

Observed variables

$$\mathbf{x}_A^{(k)} \sim p(\mathbf{x}_A \mid \mathbf{x}_D) = p(x_{a_1}, \ldots, x_{a_m} \mid x_{i_1}, \ldots, x_{i_l})$$

- Logic Sampling: Ancestral Sampling + reject samples that are not consistent with observations.
  - We generating complete samples

$$\mathbf{x}^{(k)} = (x_1^{(k)}, \ldots, x_N^{(k)}) \sim p(\mathbf{x})$$

  as before (ignoring the evidence).
  - We throw away samples in which the values drawn for the evidence variables do not correspond to the observations.
  - Problem: often almost all samples are rejected (specifically if there are many evidence variables).
  - Takes a long time to generate enough samples, often not practical.

Inference: MCMC

- Alternative strategy to generate samples: Markov Chain Monte Carlo („MCMC“)
- Idea:
  - Difficult to generate samples directly from $p(\mathbf{z})$.
  - Alternative strategy: construct sequence of samples

$$\mathbf{z}^{(0)} \rightarrow \mathbf{z}^{(1)} \rightarrow \mathbf{z}^{(2)} \rightarrow \mathbf{z}^{(3)} \rightarrow \mathbf{z}^{(4)} \rightarrow \mathbf{z}^{(5)} \rightarrow \ldots$$

$\mathbf{z}^{(0)}$ randomly initialized $\qquad \mathbf{z}^{(t+1)} \sim p(\mathbf{z}^{(t+1)} \mid \mathbf{z}^{(t)})$

    by iterative probabilistic update steps $\mathbf{z}^{(t+1)} \sim p(\mathbf{z}^{(t+1)} \mid \mathbf{z}^{(t)})$.
  - If updates are chosen appropriately, asymptotically it holds that

$$\mathbf{z}^{(T)} \sim p(\mathbf{z}) \qquad \text{approximately, for very large } T$$

Random variable: T-th sample

Markov Chains

- We study the sequence of samples

$$\mathbf{z}^{(0)} \to \mathbf{z}^{(1)} \to \mathbf{z}^{(2)} \to \mathbf{z}^{(3)} \to \mathbf{z}^{(4)} \to \mathbf{z}^{(5)} \to \ldots$$

  as random variables, $\mathbf{z}^{(t)}$ is called state of chain at time *t.*

- These random variables form a linear chain:

$$\mathbf{z}^{(0)} \sim p(\mathbf{z}^{(0)}) \qquad \mathbf{z}^{(1)} \sim p(\mathbf{z}^{(1)} \mid \mathbf{z}^{(0)}) \qquad \mathbf{z}^{(2)} \sim p(\mathbf{z}^{(2)} \mid \mathbf{z}^{(1)}) \qquad \mathbf{z}^{(3)} \sim p(\mathbf{z}^{(3)} \mid \mathbf{z}^{(2)})$$

$$\mathbf{z}^{(0)} \longrightarrow \mathbf{z}^{(1)} \longrightarrow \mathbf{z}^{(2)} \longrightarrow \mathbf{z}^{(3)} \longrightarrow \ldots$$

- Such linear chains are also called Markov chains.

Markov Chains

- The distribution over $\mathbf{z}^{(t+1)}$ can be computed based on the distribution over $\mathbf{z}^{(t)}$:

new state   current state

$$p(\mathbf{z}^{(t+1)}) = \sum_{\mathbf{z}^{(t)}} p(\mathbf{z}^{(t+1)} \mid \mathbf{z}^{(t)})\, p(\mathbf{z}^{(t)})$$

- A distribution $p(\mathbf{z}^{(t)})$ is called stationary, if $p(\mathbf{z}^{(t+1)}) = p(\mathbf{z}^{(t)})$.
- If chain has reached a stationary distribution at time t , the stationary distribution will be preserved:

$$p(\mathbf{z}^{(t+k)}) = p(\mathbf{z}^{(t)}) \qquad \text{for all k} \geq 0$$

- Under certain assumptions („ergodic chains“), Markov chains converge to a unique stationary distribution („equilibrium distribution“).

MCMC in Graphical Models

- Given a graphical model over random variables $\mathbf{x} = \{x_1, \ldots, x_N\}$, the model defines a distribution $p(\mathbf{x})$.
- For the time being we assume that there is no evidence.
- „Markov Chain Monte Carlo“ methods
  - From the graphical model, construct a sequence of samples by iterative probabilistic updates

$$\mathbf{x}^{(0)} \rightarrow \mathbf{x}^{(1)} \rightarrow \mathbf{x}^{(2)} \rightarrow \mathbf{x}^{(3)} \rightarrow \mathbf{x}^{(4)} \rightarrow \mathbf{x}^{(5)} \rightarrow \ldots$$

each $\mathbf{x}^{(t)}$ assignment of values to all nodes

$\mathbf{x}^{(0)}$ randomly initialized

$$\mathbf{x}^{(t+1)} \sim p(\mathbf{x}^{(t+1)} \mid \mathbf{x}^{(t)})$$

  - Goal: choose updates in such a way that we get an ergodic Markov chain with equilibrium distribution $p(\mathbf{x})$.
  - Most simple method: successively locally redraw a single variable conditioned on states of all other variables („Gibbs-Sampling“).

Inference: Gibbs Sampling

- Gibbs Sampling: one variant of MCMC.
- Probabilistic update step given by successively locally drawing a single random variable conditioned on state of all other variables.
  - Given old state $\mathbf{x} = (x_1, \ldots, x_N)$
  - Draw new state $\mathbf{x}' = (x_1', \ldots, x_N')$:

$$
\begin{aligned}
x_1' &\sim p(x_1 \mid \overbrace{x_2, \ldots, x_N}^{\text{states sampled in last update step}}) \\
x_2' &\sim p(x_2 \mid x_1', x_3, \ldots, x_N) \\
x_3' &\sim p(x_3 \mid x_1', x_2', x_4, \ldots, x_N) \\
&\ldots \\
x_N' &\sim p(x_N \mid x_1', x_2', \ldots, x_{N-1}')
\end{aligned}
$$

Random initialization in the beginning.

Inference: Gibbs Sampling

- Theorem: If $p(x_i \mid pa(x_i)) \neq 0$ for all i and all possible x_i, $pa(x_i)$, then the resulting Markov chain is ergodic with equilibrium distribution $p(\mathbf{x})$.

- Single Gibbs-step is easy: all variables except current query variable are observed, naive inference in time $\mathrm{O}(\mathrm{M\ N})$.

Gibbs Sampling With Evidence

- So far we have looked at inference without evidence.

- How do we obtain samples from the conditional distribution?

  Goal: $\mathbf{x}^{(T)} \sim p(\mathbf{x} \mid \mathbf{x}_D)$ approximately, for very large T

- Slight modification of Gibbs sampling algorithm:
  - Gibbs sampling always redraws a variable x_i, conditioned on the states of the other variables.
  - With evidence: only redraw the unobserved variables, the observed variables are fixed to their observed values.

Inference: Gibbs Sampling

- Summary Gibbs sampling algorithm:
  - $\mathbf{x}^{(0)}$ = random initialization of all random variables, consistent with evidence $\mathbf{x}_D$
  - For $t = 1,\ldots,T$: $\mathbf{x}^{(t+1)} = \text{Gibbs-update}(\mathbf{x}^{(t)})$ [Slide 27]
  - The samples $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \mathbf{x}^{(3)}, \ldots$ are asymptotically distributed according to $p(\mathbf{x} \mid \mathbf{x}_D)$
- Gibbs sampling gives reasonably good results in many practical applications
  - Individual update steps are efficient
  - Convergence is guaranteed (for $t \to \infty$)
  - Can draw samples from $p(\mathbf{x} \mid \mathbf{x}_D)$ without becoming very inefficient if evidence set is large (in contrast to logic sampling).

Inference: Gibbs Sampling

- Gibbs sampling: convergence
  - Convergence of Markov chain $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \mathbf{x}^{(3)}, \ldots$ is only guaranteed for $t \longrightarrow \infty$.
  - Practical solution: „burn-in“ iterations before samples are used (discard samples $\mathbf{x}^{(t)}$ for $t \leq T_{Burn-in}$).
  - There are also convergence tests to determine the number of burn-in iterations to use.

Inference: Summary

- Exact inference
  - Message passing algorithm.
  - Exact inference on polytrees (with Junction-Tree extension to general graphs).
  - Running time depends on graph structure, exponential in worst-case.

- Approximate inference
  - Sampling methods: approximation through a set of samples, exact results for $t \longrightarrow \infty$.
    - Ancestral sampling: simple, fast, no evidence.
    - Logic sampling: with evidence, but rarely feasible.
    - MCMC/Gibbs sampling: efficient approximate drawing of samples conditioned on evidence.

Agenda

- Graphical models: syntax and semantics.
- Inference in graphical models (exact, approximate)
- Graphical models in machine learning.

Recap: Parameter Estimate for Coin Tosses

- Recap: coin toss
  - Individual coin toss Bernoulli distributed with parameter μ

$$X \in \{0,1\}$$

$$X \sim \text{Bern}(X \mid \mu) = \mu^X (1-\mu)^{1-X}$$

$$\mu = p(X = 1 \mid \mu) \ \text{ unknown parameter}$$

- Parameter estimation problem:
  - We have observed N independent coin tosses, in the form of observations $L=\{x_1, \ldots, x_N\}$ of the random variables $X_1, \ldots, X_N$.
  - The true parameter μ is unknown, our goal is an estimate $\hat{\mu}$ or a posterior distribution $p(\mu \mid L)$.
  - Bayesian approach: posterior $\propto$ prior x likelihood

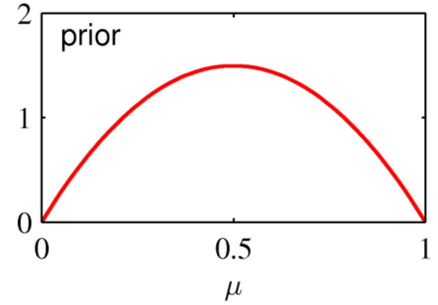
$$\underbrace{p(\mu \mid L)}_{posterior} \propto \underbrace{p(L \mid \mu)}_{likelihood} \underbrace{p(\mu)}_{prior}.$$

Recap: Parameter Estimate for Coin Tosses

- Prior: Beta distribution over coin toss parameter μ

$$p(\mu) = Beta(\mu \mid \alpha_k, \alpha_z) = \frac{\Gamma(\alpha_k + \alpha_z)}{\Gamma(\alpha_k)\Gamma(\alpha_z)} \mu^{\alpha_k - 1}(1-\mu)^{\alpha_z - 1}$$

$Beta(\mu \mid 2, 2)$



- Likelihood of N independent coin tosses:

$$\begin{aligned} p(X_1, \ldots, X_N \mid \mu) &= \prod_{i=1}^{N} p(X_i \mid \mu) \qquad i.i.d. \\ &= \prod_{i=1}^{N} \mathrm{Bern}(X_i \mid \mu) \\ &= \prod_{i=1}^{N} \mu^{X_i}(1-\mu)^{1-X_i} \end{aligned}$$

Coin Tosses as a Graphical Model

- Coin toss scenario as a graphical model?
- Random variables in coin toss scenario are $X_1, \ldots, X_N, \mu$.
- Joint distribution of data and parameter: prior x likelihood

$$p(X_1, \ldots, X_N, \mu) = p(\mu)\, p(X_1, \ldots, X_N \mid \mu) = p(\mu) \prod_{i=1}^{N} \underbrace{p(X_i \mid \mu)}_{\text{Bernoulli}}$$

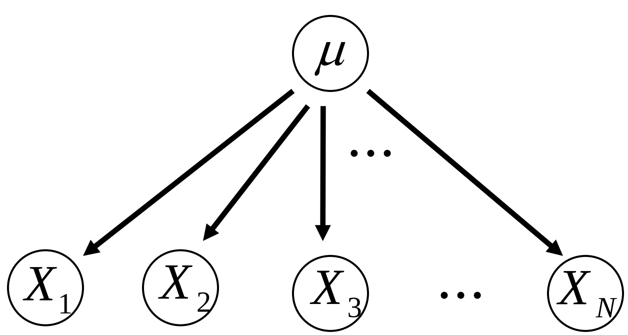
- Representation as a graphical model:

Coin Tosses as a Graphical Model

- Coin toss scenario as a graphical model?
- Random variables in coin toss scenario are $X_1, \dots, X_N, \mu.$
- Joint distribution of data and parameter: prior x likelihood

$$p(X_1, \dots, X_N, \mu) = p(\mu) p(X_1, \dots, X_N \mid \mu) = p(\mu) \prod_{i=1}^{N} \underbrace{p(X_i \mid \mu)}_{\text{Bernoulli}}$$

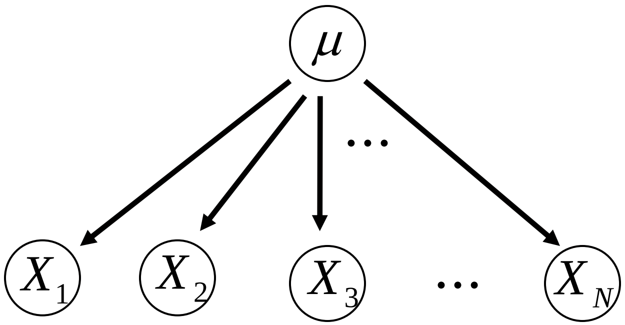
- Representation as a graphical model:



$$pa(\mu) = \varnothing$$
$$pa(X_i) = \{\mu\}$$

Coin Tosses as a Graphical Model

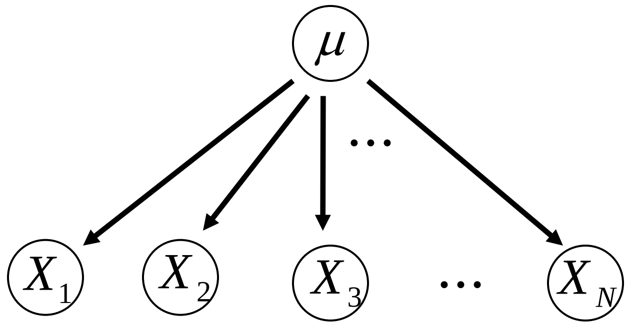
- Independent coin tosses: representation as a graphical model.



- D-separation
  - Does $X_N \perp X_1, \ldots, X_{N-1} \mid \varnothing$ hold?

Coin Tosses as a Graphical Model

- Independent coin tosses: representation as a graphical model.



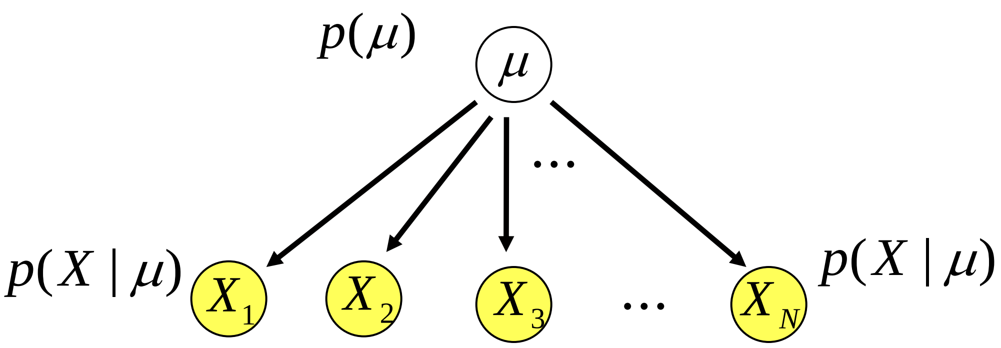
- D-separation
  - Does $X_N \perp X_1, \ldots, X_{N-1} \mid \varnothing$ hold?
  - No, path through μ is not blocked.
  - Intuitively: $X_1 = X_2 = \ldots = X_{N-1} = 1 \Rightarrow$ probably $\mu > 0.5 \Rightarrow$ probably $X_N = 1$
  - The unknown parameter μ couples the random variables $X_1, \ldots, X_N$.
  - But it holds that $X_N \perp X_1, \ldots, X_{N-1} \mid \mu$.

Parameter Estimation as Inference Problem

- MAP parameter estimation coin tosses:

$$\hat{\mu} = \arg\max_{\mu} p(\mu \mid x_1, \ldots, x_N).$$

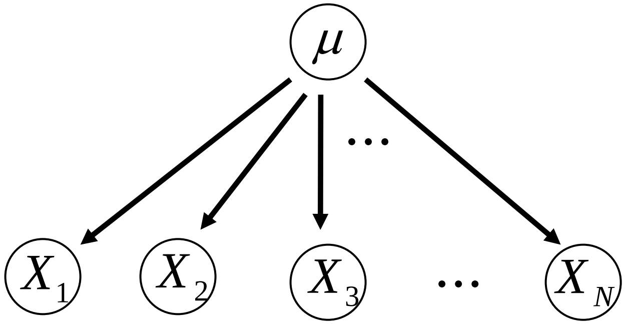
- Inference problem:



  - Evidence on the nodes $X_1, \ldots, X_N$.
  - Want: distribution $p(\mu \mid X_1, \ldots, X_N)$.

Plate Models

- Extension of graphical models: Plate notation.
- Independent coin tosses: representation as graphical model.

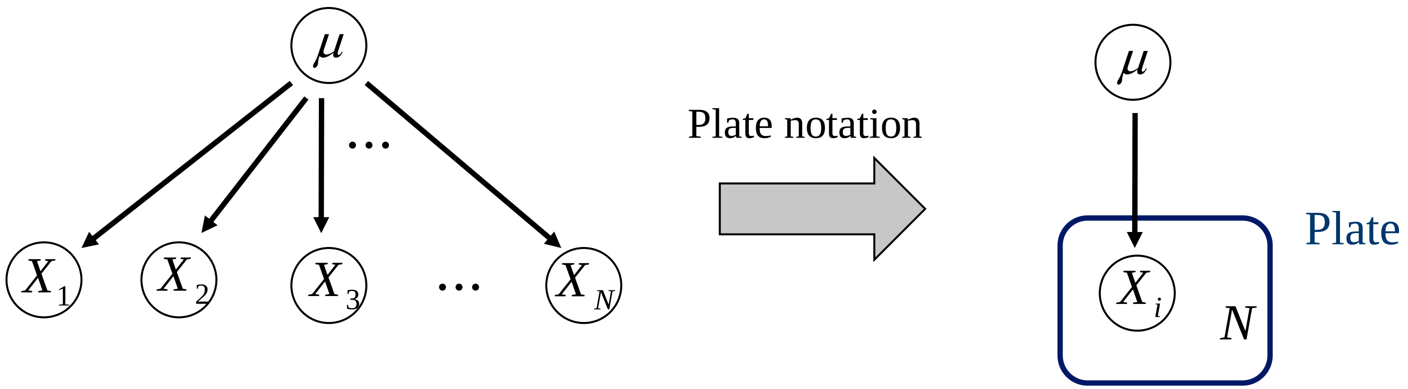


- Nodes $X_1, \dots, X_N$ are of the same form
  - Same domain (binary)
  - Same conditional distribution $p(X_i \mid \mu) = p(X_j \mid \mu)$.

- Shorthand notation in form of a „template“: Plate notation.

Plate Models

- Plate notation for coin tosses:



- A „Plate“ is a shorthand notation for N variables of the same form
  - Labeled with the number of variables, N
  - Variables have index (e.g. X_i).
- Plate models are often used in graphical models for machine learning.

Plate Models: Hyperparameters

- Role of „hyperparameters“ α_k, α_z ?
  - Not random variables, we only model the joint distribution of $X_1,\ldots,X_N,\mu$ given hyperparameters.

$$p(X_1,\ldots,X_N,\mu \mid \alpha_k,\alpha_z) = p(\mu \mid \alpha_k,\alpha_z)\prod_{i=1}^{N} p(X_i \mid \mu)$$

  - Hyperparameters are not nodes in the graphical model, but are often additionally depicted (with point instead of circle).