Introduction to Bayesian networks

Wim Wiegerinck
SNN, Radboud University Nijmegen

Contents

- Reasoning with uncertainty. Rules vs probability theory
- Probabilistic models
- Conditional independence
- Bayesian networks
- Some common graphical models
- Inference
- Learning
- Summary

Reasoning with uncertainty

Why reasoning with uncertainty?

- finite amount of data.
- insufficient detailed knowledge of conditions and circumstances.
- intrinsic noise.

Encoding knowledge in realistic domain $\rightarrow$ many exceptions

- enumerating them $\rightarrow$ too complex.
- ignoring them $\rightarrow$ too simple.
- summarizing them $\rightarrow$ uncertainty.

How to reason?

- knowledge is expressed by rules. Reasoning by combining rules,
- knowledge is expressed by a probabilistic model. Reasoning by probability theory.

Wet Grass

- Mr. Holmes has a garden with a nice grass. One morning, when Mr. Holmes leaves the house, he notices that his grass is wet. Is it due to rain, or has he forgotten to turn off the sprinkler? His belief in both events increases.
- Next he notices that the grass of his neighbor, Dr. Watson, is also wet. Holmes is almost sure that that it has been raining.
- Holmes thinks: If it rained then that explains why my grass is wet, so there is no reason to believe that the sprinkler has been on.

Rules with uncertainty

Rules

if *condition* with certainty x **then** *fact* with certainty $f(x)$

E.g.

if Rained **then** Wet Grass with certainty f

if Sprinkler **then** Wet Grass with certainty g

How to combine rules?

if Wet Grass **then** Rained, Sprinkler?

if Rained **then** Wet Grass **then** Sprinkler has been on ??

Problem: Rules are *context free*

Context in probability theory

A *conditional probability* $P(A|C) = x$ statement means:

Given that I know C (context), then the probability of A is x

Conditional probability is related to marginal probabilities,

$$P(A|C) = \frac{P(A,C)}{P(C)} = \frac{P(A,C)}{\sum_{A'} P(A',C)}$$

Probability theory is *context sensitive*.

Wet grass with probabilities

Rained	Sprinkler	Wet	P
T	T	T	0.01
T	T	F	0.00
T	F	T	0.50
T	F	F	0.05
F	T	T	0.05
F	T	F	0.01
F	F	T	0.00
F	F	F	0.38

$$
\begin{aligned}
P(R) &= 0.56 \\
P(R|W) &= 0.92 \\
P(S) &= 0.07 \\
P(S|W) &= 0.11 \\
P(S|R) &= 0.02 \\
P(S|R,W) &= 0.02
\end{aligned}
$$

Probabilistic models

Probabilistic model

- Random variables $x_1, \ldots, x_n$
- Joint Probability Distribution (JPD) $P(x_1, \ldots, x_n)$.

Inference: computation of the probabilities of interest

- Marginal $P(x_i) = \sum_{\{x_1,\ldots,x_{i-1},x_{i+1},x_n\}} P(x_1, \ldots, x_n)$
- Conditional probability $P(x_i|x_j, x_k) = \frac{P(x_i,x_j,x_j)}{P(x_j,x_k)}$.

Problems with naive probabilistic approach

Naive approach: Probabilistic model requires the literal encoding of all entries in $P(x_1, \ldots, x_n)$. Unless n is extremely small, this approach leads to several severe problems:

- The definition of P would require a table with 2^n entries.
- Inference would require summations over exponentially many terms. For instance, computing the marginal $P(x_i)$ would require summing $P(x_1, \ldots, x_n)$ over all 2^{n-1} combinations of the remaining $n-1$ variables.
- The numbers defining the table of P are not 'meaningfull'.

Solution: simplify model + computation by assuming

$$\boxed{\text{conditional independencies}} \Rightarrow \boxed{\textit{Graphical models}}$$

Conditional independence

- Variables x and y are (marginally) independent if
  - $P(x,y) = P(x)P(y)$
- Equivalently (but maybe more intuitive), variables x and y are (marginally) independent if
  knowing y tells me nothing about x,i.e.,
  - $P(x|y) = P(x)$

Similarly,

- Variables x and y are <u>conditionally independent given z</u> if
  - $P(x,y|z) = P(x|z)P(y|z)$
- Equivalently (but maybe more intuitive), variables x and y are conditionally independent given z if
  given z, knowing y tells me nothing about x, i.e.,
  - $P(x|y,z) = P(x|z)$

Conditional independence (example)

NB:

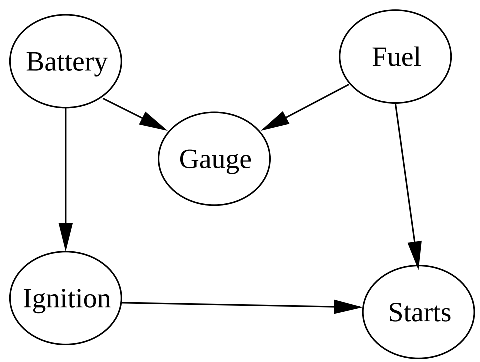
Two variables x, y that are marginally independent can be conditional dependent given z and vice versa

Example: I have a book tabulating population weight distributions by age and sex. I want to know the weight y of John.

- Given: Peter weighs x kg. What does x tell about y?
- Next, it is given that John and Peter are of the same age. What does x tell about y?
- Next, it is given that John and Peter are both 10 years old and that for 10 year old boys, the mean weight = 35 kg, sd = 6 kg What does x tell about y?

Bayesian networks

- A Bayesian network is a JPD defined by a factorization into conditional probability distributions.
- The factorization is uniquely represented by a Directed Acyclic Graph (DAG), hence the term 'Graphical model'.
- The model parameters are the conditional probability tables (rather than entries in the full JPD).

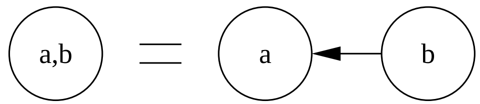


$$P(B, F, G, I, S) = P(B)P(F)P(G|B, F)P(I|B)P(S|F, I)$$

Chain rule

In general,

$$P(a,b) = P(a|b)P(b)$$



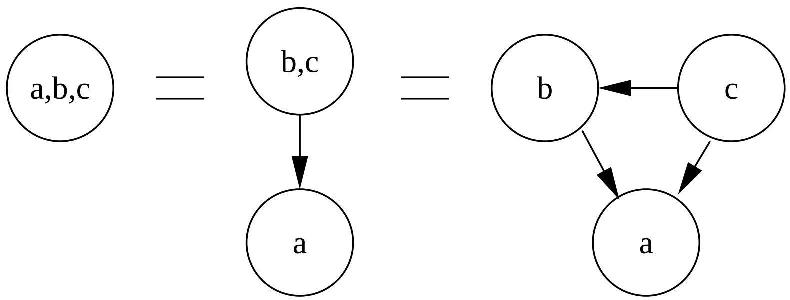
So,

$$P(x_1,\ldots,x_n) = P(x_1|x_2,\ldots,x_n)P(x_2,\ldots,x_n)$$
$$= P(x_1|x_2,\ldots,x_n)P(x_2|x_3,\ldots,x_n)P(x_3,\ldots,x_n),$$

etc

and we end up with the product,

$$P(x_1,\ldots,x_n) = \prod_{i=1}^{n} P(x_i|x_{i+1}\ldots x_n)$$



Bayesian networks and chain rule

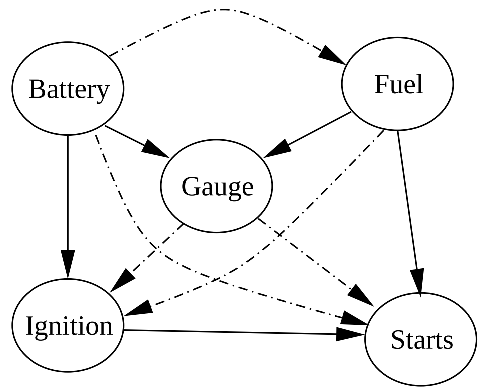
Bayesian network:

- Ordering $x_1, \ldots, x_n$
- Parent sets $\pi_i \subset \{x_1, \ldots, x_{i-1}\}$
- Conditional probability distributions satisfy

$$P(x_i | x_{i-1} \ldots x_1) = P(x_i | \pi_i)$$

- Joint distribution of the Bayesian network

$$P(x_1, \ldots, x_n) = \prod_{i=1}^{n} P(x_i | \pi_i)$$

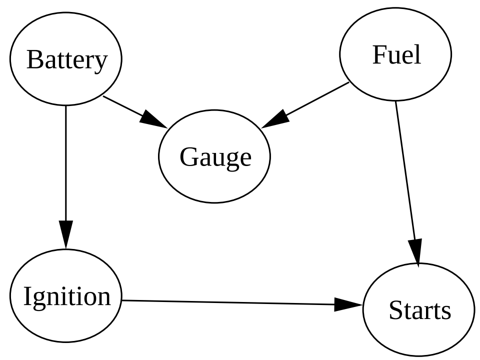


$$\begin{aligned} p(F|B) &= p(F) \\ p(I|B, F, G) &= p(I|B) \\ p(S|B, F, G, I) &= p(S|F, I) \end{aligned}$$

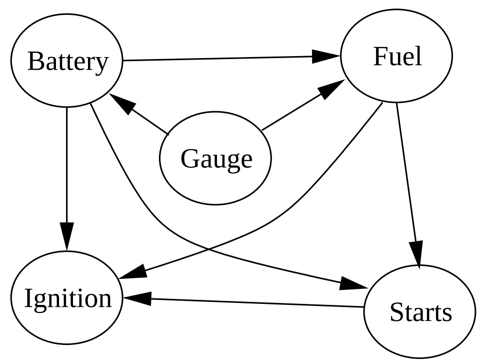
Compactness and node ordering

All ordenings are valid, but may yield awkward results.
Often, causal ordering gives simple and understandable models

- Ordering = F, B, G, I, S

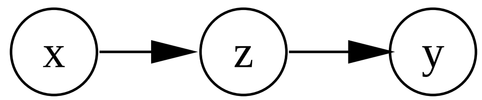


- Ordering = G, B, F, S, I
  (less conditional independency assumptions)

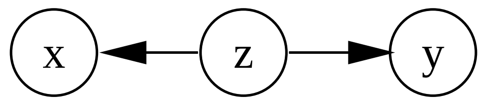


Conditional independence relations in BN's

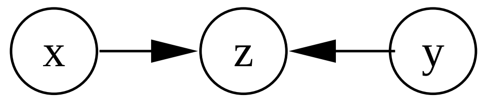
- Serial connections:
  z clamped: x and y independent



- diverging connections
  z clamped: x and y independent



- converging connections
  z free: x and y independent
  (otherwise: 'explaining away')



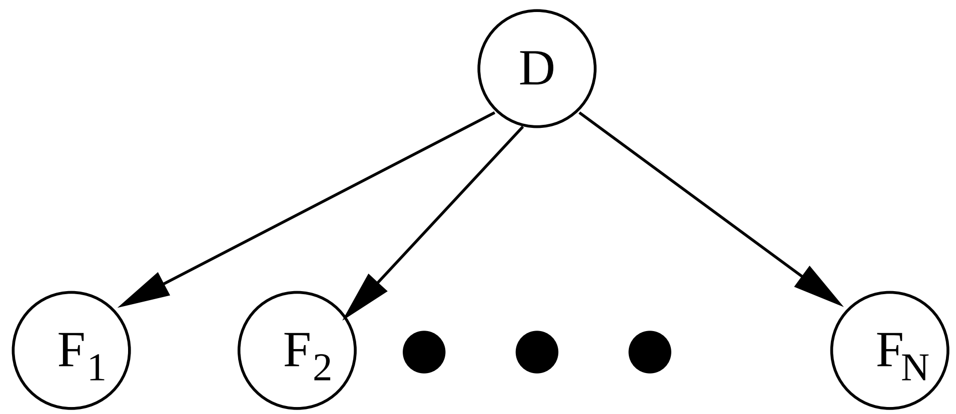
Formalized in d-separation

Naive Bayes

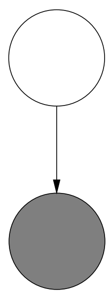
Model often used for (naive) medical diagnosis.

D : disease state, $D = \{d_1, d_2, \ldots, d_n\}$, mutually exclusive
F_i : Finding (symptom)
$P(D, F_1, F_2, \ldots, F_N) = P(D)P(F_1|D)P(F_2|D)\ldots P(F_N|D)$

- Only n (N+1) parameters.
- Given D, the F_i's are independent (diverging connections).



Hidden variable models for datamining



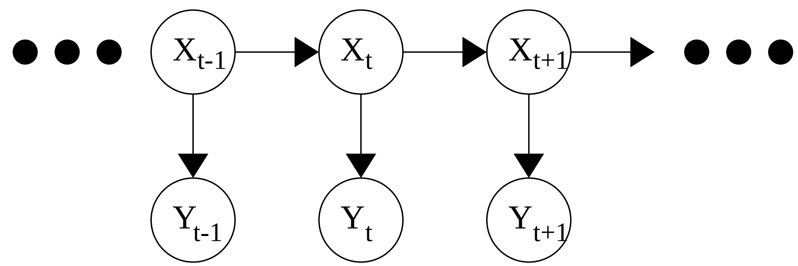
X: hiddens (= latent),

Y: observations

- Mixture of Gaussians: X discrete, Y Gaussian.
  Clustering
- Factor Analysis: X, Y Gaussian, $Y = WX + \text{noise}$.
  Dimensionality reduction.
- Independent Component Analysis (ICA):
  X non-Gaussian, $Y = WX + \text{noise}$.
  Blind source separation (cocktail party problem).
- Extensions: mixtures of ...

Temporal models: Dynamic Bayesian networks

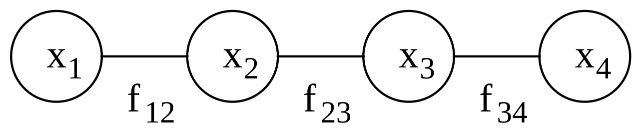
- Examples: Hidden Markov Model (HMM), Kalman filter. Typical applications: speech recognition (HMM), Tracking (KF).
- X_t hidden states, Y_t observations.
- Time invariant transition probabilities $P(X_t|X_{t-1})$ and observation model $P(Y_t|X_t)$.



Marginals by local computation

Compute the "marginal" $f(x_3) = \sum_{x_1,x_2,x_4} f(x_1,x_2,x_3,x_4)$ where

$$f(x_1,x_2,x_3,x_4) = f_{12}(x_1,x_2)f_{23}(x_2,x_3)f_{34}(x_3,x_4)$$



First, sum over x_1:

$$\begin{aligned}
\textstyle\sum_{x_1} f(x_1,x_2,x_3,x_4) \ &= \\
&f_{12}(0,x_2)f_{23}(x_2,x_3)f_{34}(x_3,x_4) + f_{12}(1,x_2)f_{23}(x_2,x_3)f_{34}(x_3,x_4) \\
&= \left[\, f_{12}(0,x_2) + f_{12}(1,x_2) \,\right] \ f_{23}(x_2,x_3)f_{34}(x_3,x_4) = \\
&= \left[\, \textstyle\sum_{x_1} f_{12}(x_1,x_2) \,\right] f_{23}(x_2,x_3)f_{34}(x_3,x_4) \\
&\equiv \ \lambda_{12}(x_2)f_{23}(x_2,x_3)\ f_{34}(x_3,x_4) \\
&\equiv f^*_{23}(x_2,x_3)\ f_{34}(x_3,x_4)
\end{aligned}$$

$\lambda_{12}(x_2)$: message from node 1 to node 2.

Next, sum over x_2,

$$\sum_{x_2} f_{23}^*(x_2, x_3) f_{34}(x_3, x_4) \equiv \lambda_{23}(x_3) f_{34}(x_3, x_4) = f_{34}^*(x_3, x_4)$$

Finally, sum over x_4,

$$\sum_{x_4} f_{34}^*(x_3, x_4) = f(x_3)$$

Sum over x_4 can also be interpreted as messages:
$\lambda_{43}(x_3) = \sum_{x_4} f_{34}(x_3, x_4)$

So, $f(x_3)$ is product over incoming messages,

$$\begin{aligned} f(x_3) &= \sum_{x_4} f_{34}^*(x_3, x_4) \\ &= \sum_{x_4} \lambda_{23}(x_3) f_{34}(x_3, x_4) \\ &= \lambda_{23}(x_3) \lambda_{43}(x_3) \end{aligned}$$

Message propagation (computing marginals by local computations)

First compute messages, from “outside inwards”

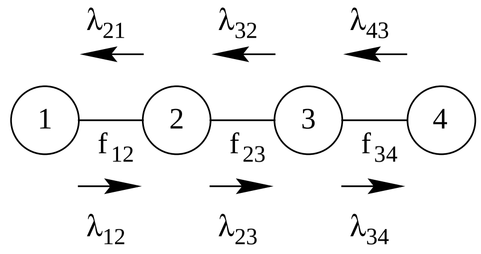
$\lambda_{12}(x_2) = \sum_{x_1} f_{12}(x_1, x_2)$

$\lambda_{23}(x_3) = \sum_{x_2} \lambda_{12}(x_2) f_{23}(x_2, x_3)$

$\lambda_{34}(x_4) = \sum_{x_3} \lambda_{23}(x_2) f_{34}(x_2, x_3)$

(NB $\lambda_{34}(x_4) = \sum_{x_3 x_2 x_1} f(x_1, x_2, x_3, x_4)$)

$\lambda_{43}(x_3) = \sum_{x_4} f_{34}(x_3, x_4)$ etc



All marginals can now be computed

$f(x_1) = \lambda_{21}(x_1)$

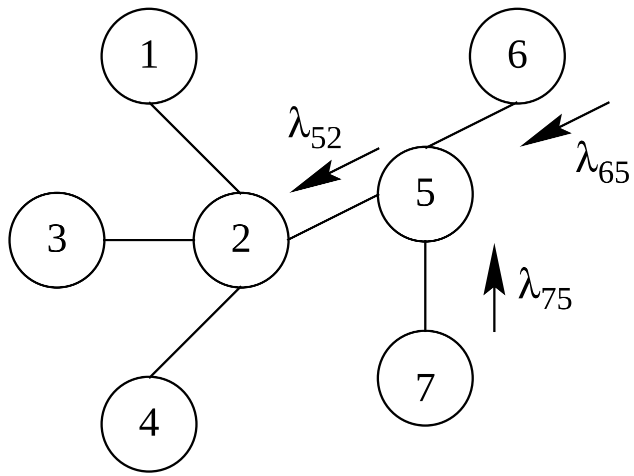
$f(x_2) = \lambda_{12}(x_3)\lambda_{32}(x_3)$ etc

Message propagation in chains

- A node i can send its message to its neighbour as soon as i received the messages of its other neighbour.
- If a node received the messages from its both neighbours, the node marginal can be computed by multiplying messages.
- Computational complexity is linear in the number of nodes.

Messages in trees

$$f(x_1, x_2, x_3, x_4, x_5, x_6, x_7) =$$
$$f_{12}(x_1, x_2) f_{23}(x_2, x_3) \times$$
$$f_{24}(x_2, x_4) f_{25}(x_2, x_5) \times$$
$$f_{56}(x_5, x_6) f_{57}(x_5, x_7)$$



Messages:

$\lambda_{65}(x_5) = \sum_{x_6} f_{56}(x_5, x_6)$

$\lambda_{75}(x_5) = \sum_{x_7} f_{57}(x_5, x_7)$

$\lambda_{52}(x_2) = \sum_{x_5} \lambda_{65}(x_5) \lambda_{75}(x_5) f_{25}(x_2, x_5)$

Marginals:

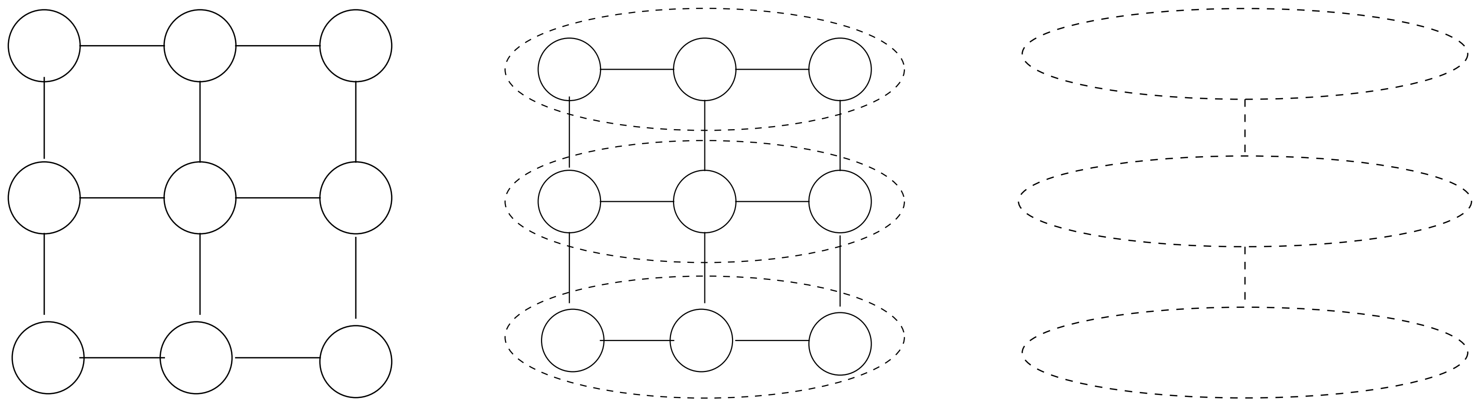
$f_5(x_5) = \lambda_{25}(x_5) \lambda_{65}(x_5) \lambda_{75}(x_5) = \sum_{\{x_{i \neq 5}\}} f(\vec{x})$

Message propagation in trees

- A node i can send its message to its neighbour as soon as i received the messages of its other neighbours.
- If a node received the messages from all its neighbours, the node marginal can be computed from the messages.
- Computational complexity is linear in the number of nodes,

Inference in general (loopy) graphs

- Graph is to be transformed into a tree of clusters of nodes (cliques or super-nodes).
- Computation is linear in number of clusters
- Computation is exponential in the number of nodes in the clusters.
- To find the optimal clustering is NP hard, but reasonable heuristics exists.
- Efficient implementation is the junction tree algorithm.



Learning from data

$D = \left\{x^1, x^2, \ldots, x^p\right\}$: data set

θ: model parameters (e.g. conditional probability tables)

Full Bayesian learning: *probability distribution of models.*

E.g. prediction of x^{new}

$$P(x^{\text{new}}|D) = \int_\theta P(x^{\text{new}}|D,\theta)P(\theta|D)d\theta = \int_\theta P(x^{\text{new}}|\theta)P(\theta|D)d\theta$$

Bayes Rule: $P(\theta|D) = \dfrac{P(D|\theta)P(\theta)}{P(D)}$

$P(\theta)$: *Prior*

$P(D\,|\,\theta)$: *Likelihood*

$P(\theta\,|\,D)$: *Posterior*

MAP (Maximum a posteriori): maximize $P(D|\theta)P(\theta)$ w.r.t θ

Maximum likelihood: maximize $P(D|\theta)$ w.r.t θ

ML in a Bayesian network

Fully observable: Maximum likelihood solution (conditional counting):

$$P(A=a|B=b,C=c) = \frac{N(A=a,B=b,C=c)}{\sum_a N(A=a,B=b,C=c)}$$

Missing data (Hidden variables): EM-algorithm:

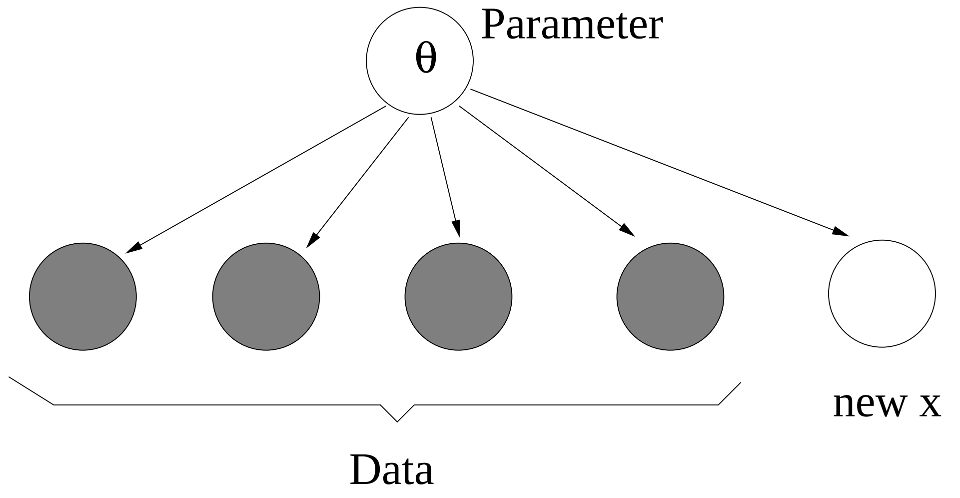
E-step Estimate missing values by the *current model* P_t, (e.g. if B is hidden:)

$$\tilde{N}(A=a,B=b,C=c) = N(A=a,C=C)\,P_t(B=b|A=a,C=c).$$

M-step Maximize likelihood

$$P_{t+1}(A=a|B=b,C=c) = \frac{\tilde{N}(A=a,B=b,C=c)}{\sum_a \tilde{N}(A=a,B=b,C=c)}$$

Learning from data as a graphical model



$$P(x^{\text{new}}|x^1,\ldots,x^p) \propto \int_\theta P(x^{\text{new}}|\theta)\prod_{i=1}^{p} P(x^i|\theta)P(\theta)d\theta$$

Summary

Probability theory provides a consistent and correct framework for reasoning with uncertainty.

Bayesian networks are a class of probabilistic models.

- Convenient graphical language to describe a large class of models.
- Large scale modeling feasible by exploiting conditional independencies.
- Efficient inference (linear in the number of clusters, exponential in the cluster-size).
- Modeling with expert knowledge easily combined with learning from data.
- Learning and reasoning in same framework