Bayesian Networks

Lecture series „Machine Learning“

Niels Landwehr

Research Group „Data Science“
Institute of Computer Science
University of Hildesheim

Agenda

- Bayesian networks: syntax and semantics
- Independence in Bayesian networks
- Bayesian networks in machine learning
- A first look at Inference and parameter estimation

Agenda

- Bayesian networks: syntax and semantics
- Independence in Bayesian networks
- Bayesian networks in machine learning
- A first look at Inference and parameter estimation

Review: Probabilities

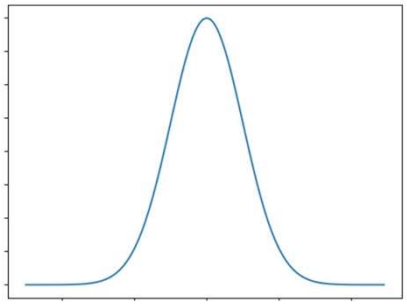
- In this lecture, we will talk about models that represent and manipulate joint distributions over a set of random variables

- **Review: random variables and probability distributions**
- Discrete random variables: distribution represented by discrete probabilities. Example: random variable $x \in \{a, b, c\}$ with discrete distribution $p(x)$

	$x = a$	$x = b$	$x = c$
$p(x)$	0.3	0.3	0.4

- Continuous random variables: distribution represented by density function. Example: normally distributed continuous random variable $x \in \mathbb{R}$

$$p(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$$



Review: Probabilities

- If there are several random variables $x, y, z, \ldots$, those have a **joint distribution** $p(x, y, z)$

- From the joint distribution, we can recover the distribution over individual variables or subsets of variables („**marginal distribution**“) by summing out the remaining variables

$$p(x) = \sum_y p(x, y) \qquad p(x) = \int_y p(x, y) dy \qquad \text{„sum rule“}$$

- The **conditional distribution** of one variable given another variable is

This is again a distribution over the random variable x

$$p(x \mid y) = \frac{p(x, y)}{p(y)}$$

- By the **product rule**, we can recover the joint distribution

$$p(x, y) = p(x \mid y) p(y)$$

- The product rule also holds more generally for sets of variables

$$p(x_1, \ldots, x_K, y_1, \ldots, y_M) = p(x_1, \ldots, x_K \mid y_1, \ldots, y_M) p(y_1, \ldots, y_M)$$

Review: Independence

- **Independence of random variables**: random variables x, y are called independent if

$$p(x,y) = p(x)p(y)$$

or equivalently $p(x \mid y) = p(x)$

or equivalently $p(y \mid x) = p(y)$

- **Conditional independence of random variables**: random variables x, y are called conditionally independent given random variable z if

$$p(x,y \mid z) = p(x \mid z)p(y \mid z)$$

or equivalently $p(x \mid y,z) = p(x \mid z)$

or equivalently $p(y \mid x,z) = p(y \mid z)$

- Conditional independence is simply the definition of independence applied to the conditional distribution $p(x,y \mid z)$

Bayesian Networks: Representing Joint Distributions

- **Bayesian networks** are a formalism for efficiently representing the joint distribution $p(x_1,...,x_M)$ over a set of random variables $\{x_1,...,x_M\}$

- From the joint distribution $p(x_1,...,x_M)$, we can compute
  - All marginal distributions, using the sum rule

$$p(x_{m_1},...,x_{m_K}) \qquad \{m_1,...,m_K\} \subseteq \{1,...,M\}$$

  - All conditional distributions, based on the marginal distributions

$$p(x_{m_1},...,x_{m_K} \mid x_{m_{K+1}},...,x_{m_{K+L}}) \qquad \{m_1,...,m_K,m_{K+1},...,m_{K+L}\} \subseteq \{1,...,M\}$$

- In this way, all probabilistic queries about the random variables under study can be answered, that is, we can probabilistically reason about all possible events/outcomes

Bayesian Networks: Representing Joint Distributions

- Bayesian networks combine graphs and probability theory in order to represent the joint distribution $p(x_1,...,x_M)$
  - Compact representation: avoid the generally exponential size of a naive representation of $p(x_1,...,x_M)$
  - Structured representation: easy to understand structure of the distribution and easy to bring in prior knowledge
  - Powerful algorithms for inference, that is, answering probabilistic queries (not discussed in this lecture)
  - Can be learned from data, that is, observations of the joint state of the random variables

- Bayesian networks, as a graphical representation of probability distributions, are also widely used as a tool for writing down and analyzing statistical models in machine learning

Bayesian Networks: Representing Joint Distributions

- Let‘s start with a toy example to introduce the main ideas behind Bayesian networks
- Consider the following toy scenario
  - We own a house in Los Angeles which has a burglar alarm
  - We are on holiday. Our neighbor has promised to call us if he hears the alarm go off. In case we think that someone broke into the house, we want to come back
  - Unfortunately, the neighbor is not always at home
  - Unfortunately, the alarm can also be triggered by small earthquakes

- We model this domain using *M=5* binary random variables

  b Burglary – Someone broke into the house

  e Earthquake – An earthquake occured

  a Alarm – The alarm goes off

  n NeighborCalls – Our neighbor calls

  r RadioReport – There is a report about an earthquake on the radio

Example: Inference Problem

- We want to model the joint distribution $p(b,e,a,n,r)$ over these random variables
- How can this distribution be modeled compactly and efficiently?

- Before talking about models, we give an example for a probabilistic query (also called **inference problem**):
  - Let‘s say the neighbor has called, $n=1$ (state of other variables unknown)
  - What is the probability that a burglary has occured, $b=1$?
  - Depends on many factors: how likely is earth quake, how likely is burglary, how likely are both to set of alarm, how likely that neighbor calls us, ...
  - We can compute this as follows (naively, exponential in number of variables)

$$p(b=1 \mid n=1) = \frac{p(b=1,n=1)}{p(n=1)}$$

$$= \frac{\sum_{e\in\{0,1\}} \sum_{a\in\{0,1\}} \sum_{r\in\{0,1\}} p(b=1,e,a,n=1,r)}{\sum_{b\in\{0,1\}} \sum_{e\in\{0,1\}} \sum_{a\in\{0,1\}} \sum_{r\in\{0,1\}} p(b,e,a,n=1,r)}$$

Sum rule: to get $p(b=1,n=1)$ need to sum over all possible values for other variables

Representing Joint Distribution: Full Table

- Back to modeling the joint distribution $p(b,e,a,n,r)$ over all random variables

- **First attempt**: do not make any assumptions about dependencies, model the distribution by a full table that gives the probability for each joint state:

2^M

b	e	a	n	r	$p(b,e,a,n,r)$
0	0	0	0	0	0.6
1	0	0	0	0	0.005
0	1	0	0	0	0.01
...	...	...	...	...	...


Any distribution $p(b,e,a,n,r)$ can be represented


Number of parameters exponential in *M*


Not easy to bring in prior knowledge: not clear what the probability for a specific joint state is

- **Second attempt**: to reduce the number of parameters, assume that all variables are independent (this obviously does not make much sense...)

$$p(b,e,a,n,r) = p(b)\,p(e)\,p(a)\,p(n)\,p(r)$$

b	$p(b)$
0	0.99
1	0.01

e	$p(e)$
0	0.95
1	0.05

...


Number of parameters linear in *M*


Independence is unrealistic and does not allow any useful inference

Representing Joint Distribution: Graphical Models

- Graphical models: to model the joint distribution compactly, make selective independence assumption that are motivated by our understanding of the domain
- First, choose an ordering of the variables, let's say $b < e < a < n < r$
- Then, apply the product rule to factorize the joint distribution:

$$p(b,e,a,n,r) = p(b,e,a,n)\,p(r \mid b,e,a,n)$$

product rule, applied to {b,e,a,n} and {r}

$$= p(b,e,a)\,p(n \mid b,e,a)\,p(r \mid b,e,a,n)$$

product rule, applied to {b,e,a} and {n}

$$= p(b,e)\,p(a \mid b,e)\,p(n \mid b,e,a)\,p(r \mid b,e,a,n)$$

product rule, applied to {b,e} and {a}

$$= p(b)\,p(e \mid b)\,p(a \mid b,e)\,p(n \mid b,e,a)\,p(r \mid b,e,a,n)$$

product rule, applied to {b} and {e}

- The result is a product of factors that each describe the distribution over one variable given variables that came earlier in the ordering
- **Which of these dependencies really exist? Can we simplify the factors?**

Independence Assumptions

- Factorization according to product rule of last slide has given

$$p(b,e,a,n,r) = p(b)\,p(e\,|\,b)\,p(a\,|\,b,e)\,p(n\,|\,b,e,a)\,p(r\,|\,b,e,a,n)$$

- We now make conditional independence assumptions, which result in random variable being removed from the condition part:

$p(e\,|\,b) = p(e)$ Earthquake does not depend on burglary

$p(a\,|\,b,e) = p(a\,|\,b,e)$ Alarm does depend on earthquake and burglary

$p(n\,|\,b,e,a) = p(n\,|\,a)$ Whether neighbor calls us only depends on alarm (we assume neighbor does not directly observe burglary)

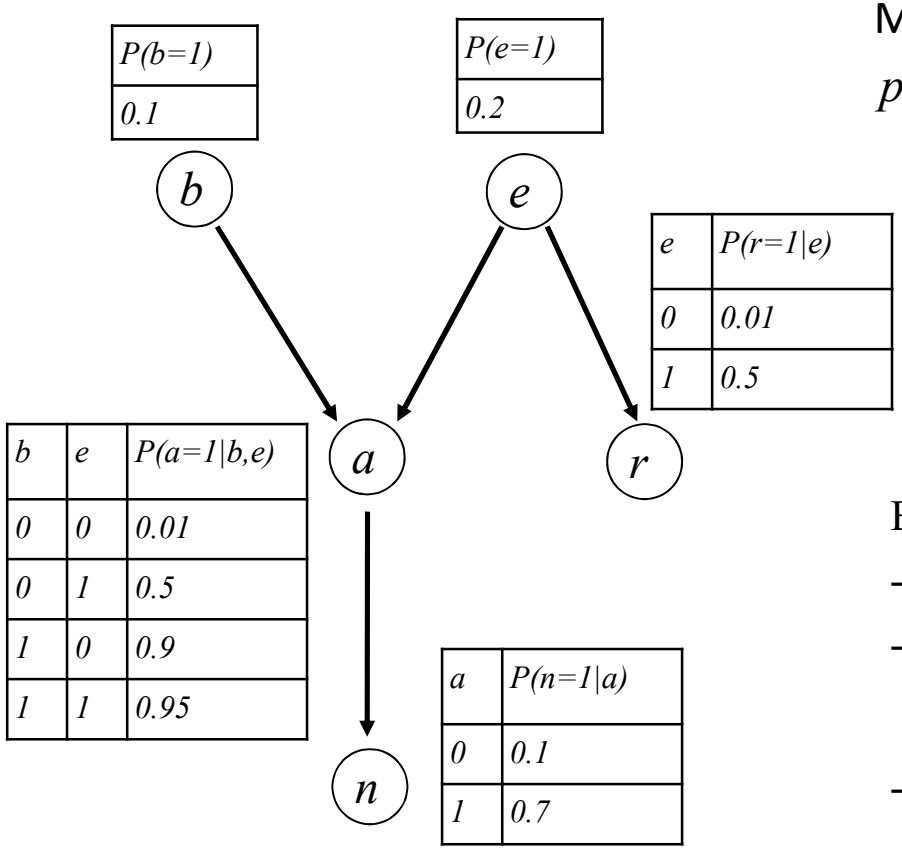
$p(r\,|\,b,e,a,n) = p(r\,|\,e)$ Whether there is a report about an earthquake on the radio only depends on the earthquake occuring

- Assumptions lead to a representation of the joint distribution with simpler factors:

$$p(b,e,a,n,r) = p(b)\,p(e)\,p(a\,|\,b,e)\,p(n\,|\,a)\,p(r\,|\,e)$$

Bayesian Network

- A Bayesian network for the domain is a graphical representation of this simplified distribution:



Modeled distribution:

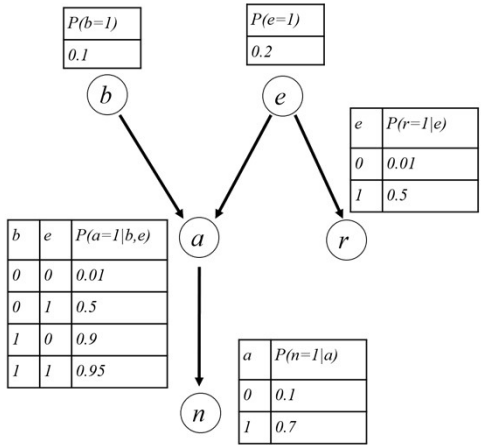
$$p(b,e,a,n,r) = p(b)\,p(e)\,p(a\,|\,b,e)\,p(n\,|\,a)\,p(r\,|\,e)$$

Bayesian network:

- Every random variable is a node
- For every factor of the form $p(x_m \,|\, x_{m_1},...,x_{m_K})$ we add directed edged from the x_{m_k} to x_m
- Model is parameterized with the conditional distributions $p(x_m \,|\, x_{m_1},...,x_{m_K})$

Bayesian Network: Example

- Bayesian network for the „Alarm“ domain:



- What is the number of parameters in the model?
  - Limited by $M \cdot 2^K$, where K is the maximum number of parents of a node
  - In this case, $1+1+2+2+4=10$ parameters compared to $2^5-1=31$ for the full table
  - The distributions $p(x_m \mid x_{m_1},...,x_{m_K})$ also make more sense to human experts than a full joint state: e.g. what is the a priori probability for earthquake or burglary, what is the probability that alarm goes off in both cases...

Bayesian Network: Formal Definition

- **Bayesian network: formal definition**
- Let $\{x_1,...,x_M\}$ be a set of random variables. A Bayesian network over the random variables $\{x_1,...,x_M\}$ is a directed graph with
  - Node set $\{x_1,...,x_M\}$
  - There are no directed cycles in the graph, that is, no pathes of the form $x_{m_1} \to x_{m_2} \to ... \to x_{m_K} \to x_{m_1}$
  - Nodes are associated with parameterized conditional distributions $p(x_m \mid pa(x_m))$ where $pa(x_m) \subset \{x_1,...,x_M\}$ is the set of parents of the node x_m in the graph

- The Bayesian network defines a joint distribution over the given random variables by

$$p(x_1,...,x_M) = \prod_{m=1}^{M} p(x_m \mid pa(x_m))$$

Directed Acyclic Graph

- Why does the graph have to be acyclic?
  - Proposition from graph theory:

$$G \text{ is acyclic } \Leftrightarrow \text{ there is an ordering } \leq_G \text{ of the nodes such that the directed edges respect the ordering } (N \rightarrow N' \Rightarrow N \leq_G N')$$

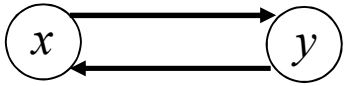
  - With this property, a factorization of the form

$$p(x_1,...,x_M) = \prod_{m=1}^{M} p(x_m \mid pa(x_m))$$

For the application of the product rule, it is required that $pa(x_m)$ come before x_m in the variable ordering

    can be derived from the product rule and conditional independence assumptions (see Slide 12), by ordering the variables according to $\leq_G$

- If there are cycles in the graph, the factorization does not work:



$$p(x,y) \neq p(x \mid y)p(y \mid x)$$

Agenda

- Bayesian networks: syntax and semantics
- Independence in Bayesian networks
- Bayesian networks in machine learning
- A first look at Inference and parameter estimation

Independence in Bayesian Networks

- The graph structure of a Bayesian network implies (conditional) independencies between the random variables in the network

- **Notation**: for random variables x, y, z we write

$$x \perp y \mid z \Leftrightarrow p(x \mid y, z) = p(x \mid z)$$

- We extend this defition to sets of random variables A, B, C as follows:

$$A \perp \mathrm{B} \mid C \Leftrightarrow p(A \mid B, C) = p(A \mid C)$$

For example,

$$\underbrace{\{x_1, x_2, x_4\}}_{A} \perp \underbrace{\{x_3, x_5\}}_{B} \mid \underbrace{\{x_6, x_7\}}_{C} \Leftrightarrow p(x_1, x_2, x_4 \mid x_3, x_5, x_6, x_7) = p(x_1, x_2, x_4 \mid x_6, x_7)$$

Independence in Bayesian Networks

- **Which independencies of the form $A \perp \mathrm{B} \mid C$ hold in the distribution represented by a Bayesian network?**

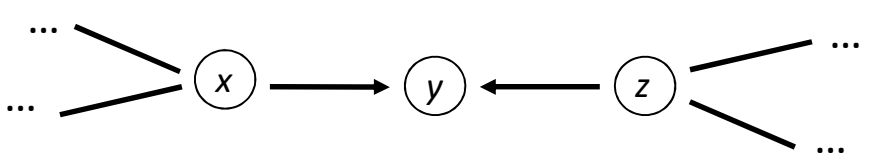
- In principle, can compute this directly from the joint distribution

$$p(x_1, ..., x_M) = \prod_{m=1}^{M} p(x_m \mid pa(x_m))$$

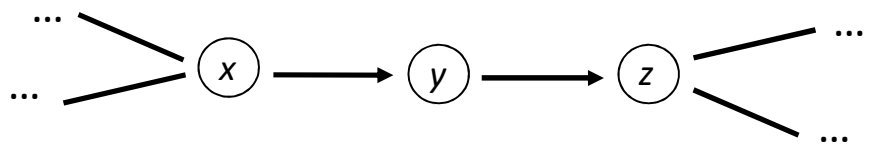
- However, for Bayesian networks there is a more convenient way that directly works with the graph structure
  - **D-separation** criterion: set of simple rules that enable us to „read of“ all independencies from the graph structure
  - The „D“ in D-separation stands for „directed“, as for Bayesian networks we work with directed graphs
  - There are also graph-based probabilistic models that work with undirected graphs, called „Markov Networks“ (not covered in this lecture)
  - Bayesian networks and Markov networks are both called „Graphical Models“

D-Separation

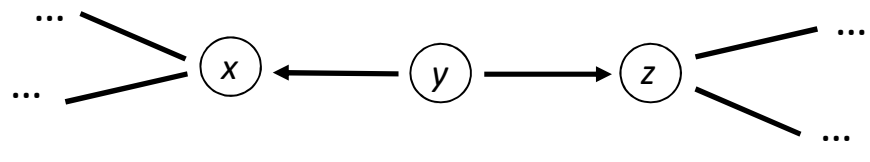
- D-separation is determined by looking at pathes between nodes in the network
- Notation:



Path between x and z has a „**converging**“ connection at y (also called „head to head“)



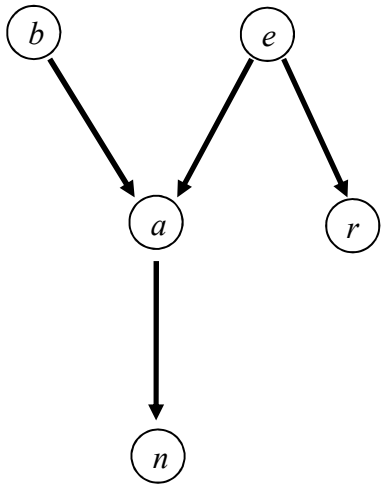
Path between x and z has a „**serial**“ connection at y (also called „head to tail“)



Path between x and z has a „**diverging**“ connection at y („tail-to-tail“)

Diverging Connection

- D-separation is determined by looking at pathes between nodes in the network



Joint distribution:

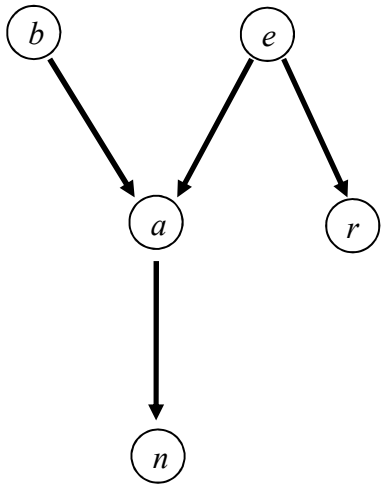
$$p(b,e,a,n,r) =$$
$$p(b)\,p(e)\,p(a\,|\,b,e)\,p(n\,|\,a)\,p(r\,|\,e)$$

b=„Burglary“ *n*=„Neighbor calls“
e=„Earthquake“ *r*=„Radio report“
a=„Alarm“

- Consider the path $a \leftarrow e \rightarrow r$. Does it hold that $a \perp r \,|\, \varnothing$?

Diverging Connection

- D-separation is determined by looking at pathes between nodes in the network



Joint distribution:

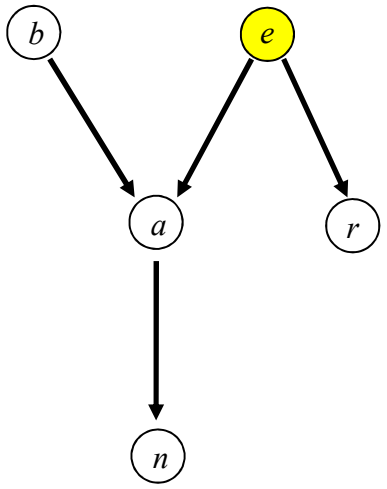
$$p(b,e,a,n,r) =$$
$$p(b)\,p(e)\,p(a\,|\,b,e)\,p(n\,|\,a)\,p(r\,|\,e)$$

b=„Burglary“ *n*=„Neighbor calls“
e=„Earthquake“ *r*=„Radio report“
a=„Alarm“

- Consider the path $a \leftarrow e \rightarrow r$. Does it hold that $a \perp r \,|\, \varnothing$?
  - No, $p(a\,|\,r) \neq p(a)$ (can be calculated from the joint distribution)
  - Intuitively: radio report $\Rightarrow$ probably earthquake $\Rightarrow$ triggers alarm
    $p(a\,|\,r=1) > p(a)$
  - We say the random variable r influences the random variable a through the diverging connection $r \leftarrow e \rightarrow a$

Diverging Connection: Observation

- D-separation is determined by looking at pathes between nodes in the network



Joint distribution:

$$p(b,e,a,n,r) =$$
$$p(b)\,p(e)\,p(a\,|\,b,e)\,p(n\,|\,a)\,p(r\,|\,e)$$

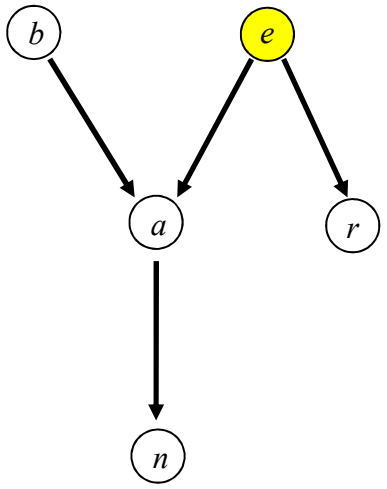
b=„Burglary“ *n*=„Neighbor calls“
e=„Earthquake“ *r*=„Radio report“
a=„Alarm“

observed node

- Consider the path $a \leftarrow e \rightarrow r$. Does it hold that $a \perp r \,|\, e$?

Diverging Connection: Observation

- D-separation is determined by looking at pathes between nodes in the network



Joint distribution:

$$p(b,e,a,n,r) = p(b)\,p(e)\,p(a\,|\,b,e)\,p(n\,|\,a)\,p(r\,|\,e)$$

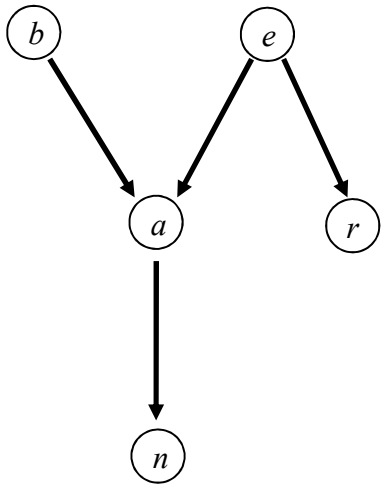
b=„Burglary“ *n*=„Neighbor calls“
e=„Earthquake“ *r*=„Radio report“
a=„Alarm“

observed node

- Consider the path $a \leftarrow e \rightarrow r$. Does it hold that $a \perp r \,|\, e$?
  - Yes, $p(a\,|\,r,e) = p(a\,|\,e)$ (can be calculated from the joint distribution)
  - Intuitively: if we already know whether an earthquake occured, hearing about it on the radio does not make alarm more or less likely
  - We say that the diverging connection $r \leftarrow e \rightarrow a$ is **blocked** by the observation of the random variable *e*

Serial Connection

- D-separation is determined by looking at pathes between nodes in the network



Joint distribution:

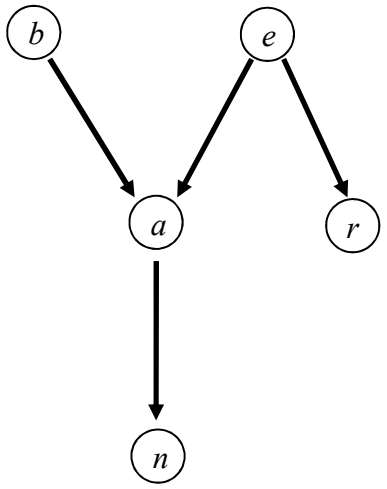
$$p(b,e,a,n,r) =$$
$$p(b)\,p(e)\,p(a\,|\,b,e)\,p(n\,|\,a)\,p(r\,|\,e)$$

b=„Burglary“ *n*=„Neighbor calls“
e=„Earthquake“ *r*=„Radio report“
a=„Alarm“

- Consider the path $n \leftarrow a \leftarrow b$. Does it hold that $n \perp b \,|\, \varnothing$?

Serial Connection

- D-separation is determined by looking at pathes between nodes in the network



Joint distribution:

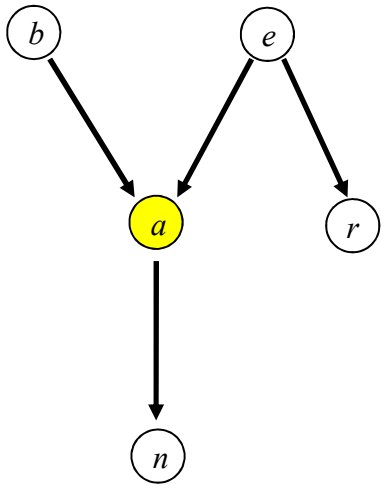
$$p(b,e,a,n,r) =$$
$$p(b)p(e)p(a \mid b,e)p(n \mid a)p(r \mid e)$$

b=„Burglary“ *n*=„Neighbor calls“
e=„Earthquake“ *r*=„Radio report“
a=„Alarm“

- Consider the path $n \leftarrow a \leftarrow b$. Does it hold that $n \perp b \mid \varnothing$?
  - No, $p(n \mid b) \neq p(n)$ (can be calculated from the joint distribution)
  - Intuitively: burglary $\Rightarrow$ probably alarm $\Rightarrow$ probably neighbor will call us (that is the idea!). Therefore $p(n \mid b = 1) > p(n)$.
  - We say the random variable b influences the random variable n through the serial connection $n \leftarrow a \leftarrow b$

Serial Connection: Observation

- D-separation is determined by looking at pathes between nodes in the network



Joint distribution:

$$p(b,e,a,n,r) =$$
$$p(b)\,p(e)\,p(a\,|\,b,e)\,p(n\,|\,a)\,p(r\,|\,e)$$

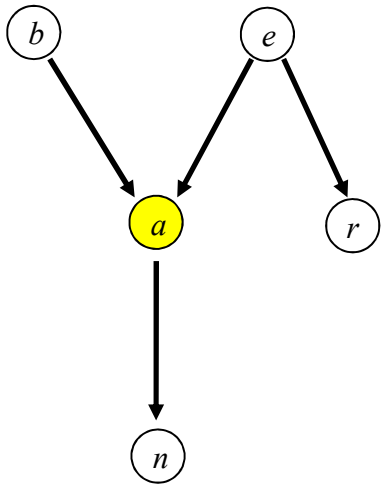
b=„Burglary“ *n*=„Neighbor calls“
e=„Earthquake“ *r*=„Radio report“
a=„Alarm“

observed node

- Consider the path $n \leftarrow a \leftarrow b$. Does it hold that $n \perp b \,|\, a$?

Serial Connection: Observation

- D-separation is determined by looking at pathes between nodes in the network



Joint distribution:

$$p(b,e,a,n,r) = p(b)\,p(e)\,p(a \mid b,e)\,p(n \mid a)\,p(r \mid e)$$

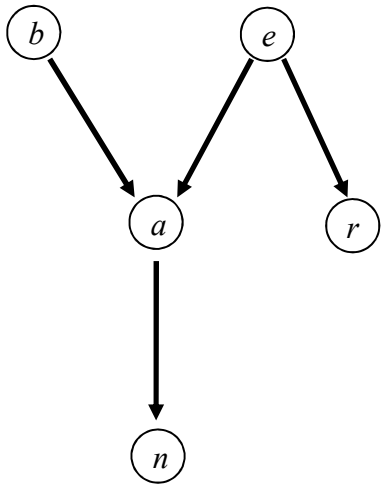
b=„Burglary“ *n*=„Neighbor calls“
e=„Earthquake“ *r*=„Radio report“
a=„Alarm“

observed node

- Consider the path $n \leftarrow a \leftarrow b$. Does it hold that $n \perp b \mid a$?
  - Yes, $p(n \mid b,a) = p(n \mid a)$ (can be calculated from the joint distribution)
  - Intuitively: if we already know that the alarm went off, neighbor calling does not give us any additional information (neighbor only observes the alarm)
  - We say the serial connection $n \leftarrow a \leftarrow b$ is blocked by the observation of the random variable *a*

Converging Connection

- D-separation is determined by looking at pathes between nodes in the network



Joint distribution:

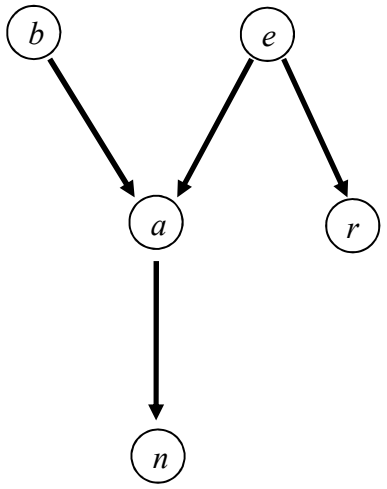
$$p(b,e,a,n,r) =$$
$$p(b)\,p(e)\,p(a\,|\,b,e)\,p(n\,|\,a)\,p(r\,|\,e)$$

b=„Burglary“ *n*=„Neighbor calls“
e=„Earthquake“ *r*=„Radio report“
a=„Alarm“

- Consider the path $b \rightarrow a \leftarrow e$. Does it hold that $b \perp e \,|\, \varnothing$?

Converging Connection

- D-separation is determined by looking at pathes between nodes in the network



Joint distribution:

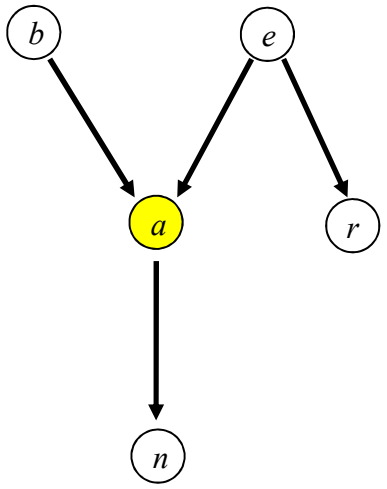
$$p(b,e,a,n,r) =$$
$$p(b)\,p(e)\,p(a\,|\,b,e)\,p(n\,|\,a)\,p(r\,|\,e)$$

b=„Burglary“ *n*=„Neighbor calls“
e=„Earthquake“ *r*=„Radio report“
a=„Alarm“

- Consider the path $b \rightarrow a \leftarrow e$. Does it hold that $b \perp e \,|\, \varnothing$?
  - Yes, $p(b\,|\,e) = p(b)$ (can be calculated from joint distribution)
  - Intuitively: burglaries do not happen more or less often on days with earthquakes
  - The converging connection at a is blocked when a is **not observed**

Converging Connection: Observation

- D-separation is determined by looking at pathes between nodes in the network



Joint distribution:

$$p(b,e,a,n,r) =$$
$$p(b)\,p(e)\,p(a\,|\,b,e)\,p(n\,|\,a)\,p(r\,|\,e)$$

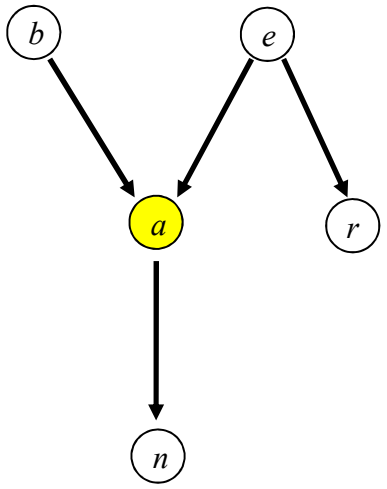
b=„Burglary“ *n*=„Neighbor calls“
e=„Earthquake“ *r*=„Radio report“
a=„Alarm“

observed node

- Consider the path $b \rightarrow a \leftarrow e$. Does it hold that $b \perp e \,|\, a$?

Converging Connection: Observation

- D-separation is determined by looking at pathes between nodes in the network



Joint distribution:

$$p(b,e,a,n,r) =$$
$$p(b)\,p(e)\,p(a\,|\,b,e)\,p(n\,|\,a)\,p(r\,|\,e)$$

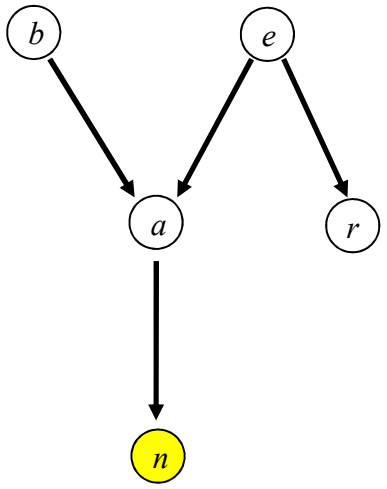
b=„Burglary“ *n*=„Neighbor calls“
e=„Earthquake“ *r*=„Radio report“
a=„Alarm“

observed node

- Consider the path $b \rightarrow a \leftarrow e$. Does it hold that $b \perp e \,|\, a$?
  - No, $p(b\,|\,e,a) \neq p(b\,|\,a)$ (can be calculated from the joint distribution)
  - Intuitively: an alarm was observed. If we know that an earthquake has occured, that explains the alarm and makes burglary less likely („explaining away“)
  - We say the converging connection $b \rightarrow a \leftarrow e$ is **unblocked** by the observation of the random variable *a*

Converging Connection: Observation Descendent

- D-separation is determined by looking at pathes between nodes in the network



Joint distribution:

$$p(b,e,a,n,r) =$$
$$p(b)\,p(e)\,p(a\,|\,b,e)\,p(n\,|\,a)\,p(r\,|\,e)$$

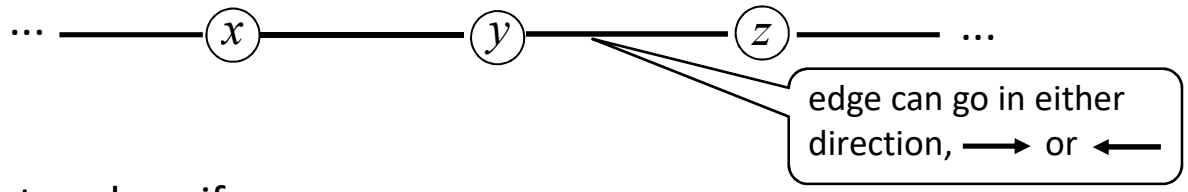
b=„Burglary“ *n*=„Neighbor calls“
e=„Earthquake“ *r*=„Radio report“
a=„Alarm“

observed node

- Consider the path $b \rightarrow a \leftarrow e$. Does it hold that $b \perp e \,|\, n$?
  - No, $p(b\,|\,e,a) \neq p(b\,|\,a)$ (can be calculated from the joint distribution)
  - Intuitively: neighbor calls can be seen as an indirect observation of alarm. Observation of earthquake still explains the alarm and makes burglary less likely
  - We say the converging connection $b \rightarrow a \leftarrow e$ is **unblocked** by the observation of the descendent node with random variable *n*

Summary: Blocked Pathes

- Assume a path of the form
- Motivated by the discussion in the example, we call a path of the form



blocked at node y if

- there is a diverging connection at y and y is observed or
- there is a serial connection at y and y is observed or
- there is a converging connection at y and neither y nor any of its descendents $y' \in \text{descendents}(y)$ is observed, where

$$\text{descendents}(y) = \{y' \in G \mid \text{there is a directed path from } y \text{ to } y' \text{ in } G\}$$

- Otherwise, the path is **open** at the node y

Summary: Blocked Pathes

- We also call the overall path blocked if it is blocked at any of the nodes on the path
- More formally:
  - Let x, x' be random variables, and C a set of observed random variables in the Bayesian network
  - A path $x - x_{m_1} - ... - x_{m_k} - x'$ between x, x' is blocked given C if there is a node x_{m_i} on the path such that the path is blocked at x_{m_i} given C

- We can now define the D-separation criterion:
  - Let A, B, C denote disjunctive sets of random variables in the network
  - We say that A and B are D-separated given C if and only if any path from a node $x \in A$ to a node $x' \in B$ is blocked given C

Correctness and Completeness of D-Separation

- D-separation as a criterion for determining independence between sets of random variables in a Bayesian network is correct and complete in the following sense

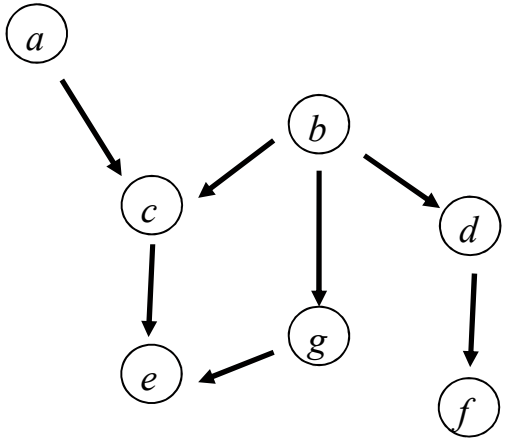
- Given a Bayesian network over random variables $\{x_1,...,x_M\}$ with graph structure G
- As explained above, the Bayesian network defines a joint probability distribution as

$$p(x_1,...,x_M) = \prod_{m=1}^{M} p(x_m \mid pa(x_m)),$$

  which is a function of the conditional probability distributions $p(x_m \mid pa(x_m))$
- **Theorem (correctness, completenes of D-separation)**
  - If A and B are D-separated given C, then it holds that $p(A \mid B,C) = p(A \mid C)$
  - There are no other independencies that hold irrespective of the choice of the conditional probability distributions $p(x_m \mid pa(x_m))$

Example: D-Separation

- Example D-separation:



Does it hold that $a \perp f \mid d$?

Does it hold that $b \perp e \mid c$?

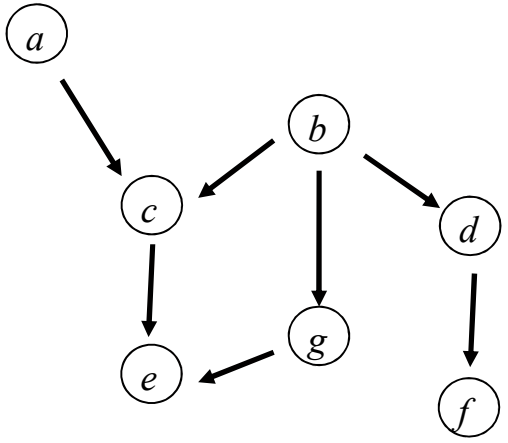
Does it hold that $a \perp e \mid c$?

- See above: a path is blocked at node y if
  - there is a diverging connection at y and y is observed or
  - there is a serial connection at y and y is observed or
  - there is a converging connection at y and neither y nor any of its descendents $y' \in \text{descendents}(y)$ is observed, where

$$\text{descendents}(y) = \{y' \in G \mid \text{there is a directed path from } y \text{ to } y' \text{ in } G\}$$

Example: D-Separation

- Example D-separation:



Does it hold that $a \perp f \mid d$? Yes

Does it hold that $b \perp e \mid c$? No $(b - g - e)$

Does it hold that $a \perp e \mid c$? No $(a - c - b - g - e)$

- See above: a path is blocked at node y if
  - there is a diverging connection at y and y is observed or
  - there is a serial connection at y and y is observed or
  - there is a converging connection at y and neither y nor any of its descendents $y' \in \text{descendents}(y)$ is observed, where

$$\text{descendents}(y) = \{y' \in G \mid \text{there is a directed path from } y \text{ to } y' \text{ in } G\}$$

Agenda

- Bayesian networks: syntax and semantics
- Independence in Bayesian networks
- Bayesian networks in machine learning
- A first look at Inference and parameter estimation

Bayesian Networks in Machine Learning

- Probabilistic models in machine learning define distributions over random variables
  - Targets and features or targets given features (e.g. discriminant analysis, logistic regression, naive Bayes)
  - In Bayesian learning, we also treat model parameters as random variables

- Bayesian networks (or more generally graphical models) can be a useful formalism for expressing and understanding such probabilistic models
  - They give insight into the structure of the model
  - They can be used to quickly sketch new models or adapt existing models
  - Problems such as prediction or parameter estimation can be seen as inference problems in the corresponding Bayesian network

- We will now look at some examples of Bayesian network representation of machine learning models

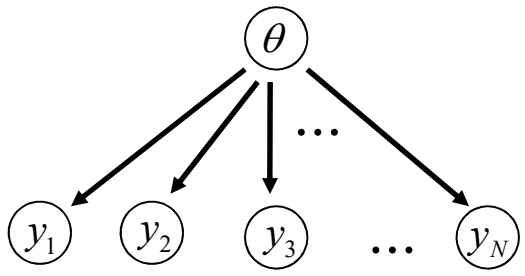
Coin Tosses

- As the simplest example, let's consider the coin toss example we discussed in the lecture on Bayesian learning

- Review: Bayesian coin toss model defines a joint distribution over the model parameter θ and the observed data points $y_1,...,y_N$:

$$p(y_1,...,y_N,\theta) = \underbrace{p(\theta)}_{\text{Beta}} \prod_{n=1}^{N} \underbrace{p(y_n \mid \theta)}_{\text{Bernoulli}}$$

- This is the Bayesian network formulation of a coin toss experiment



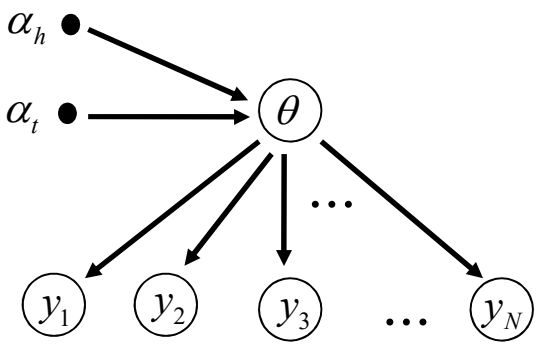
$$pa(\theta) = \varnothing$$
$$pa(y_n) = \{\theta\}$$

Hyperparameter

- In the coin toss model, the parameters of the Beta distribution α_h, α_t are hyperparameters of the model

$$p(y_1,\ldots,y_N,\theta) = \underbrace{p(\theta \mid \alpha_h,\alpha_t)}_{\text{Beta}} \prod_{n=1}^{N} \underbrace{p(y_n \mid \theta)}_{\text{Bernoulli}}$$

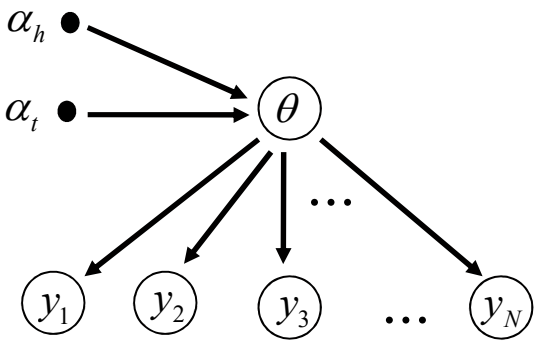
- The hyperparameters are not random variables, because we do not define a distribution over them. However, they influence the distribution over the other variables
- In the Bayesian network notation, they are often denoted using filled dots rather than circles:

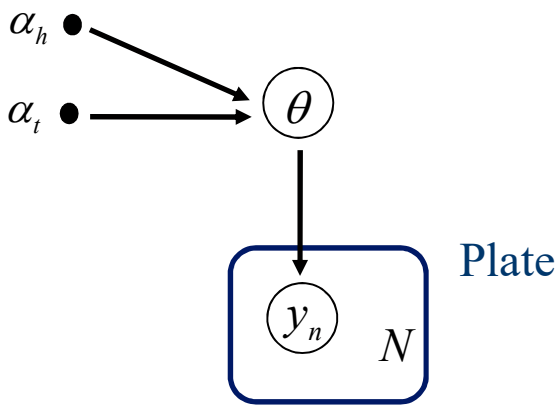


$$pa(\theta) = \varnothing$$
$$pa(y_n) = \{\theta\}$$

Plate Notation

- The plate notation is an extension of Bayesian networks (and more generally graphical models) to more easily represent repeated structures in models
- Plates are a „template“ (shorthand) notation for several random variables of the same form
  - variables all have the same domain
  - variables all have the same parents and same conditional distribution given parents
- For example, in the coin toss model:

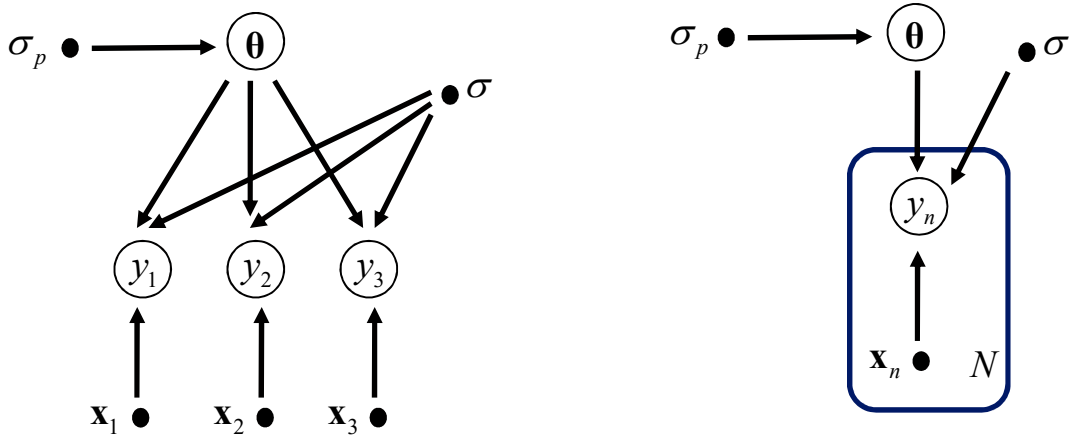




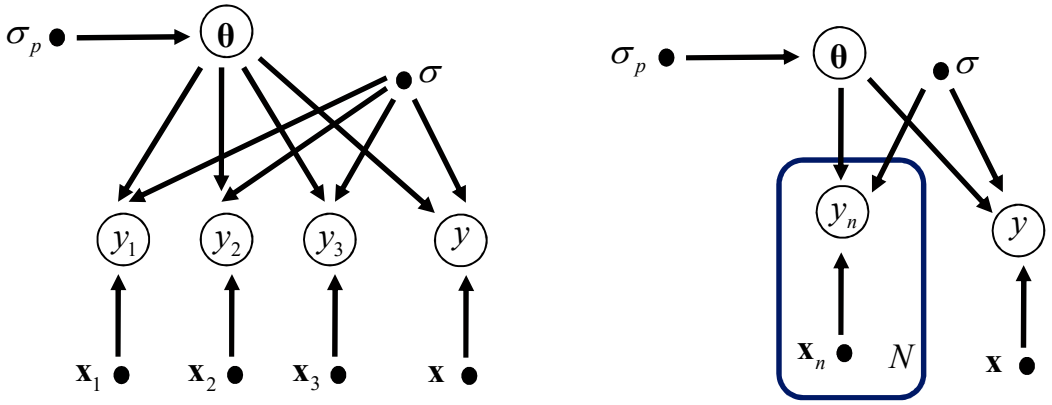
- Plate is labeled with the number of variables (N) and variables is indexed (y_n)

Bayesian Linear Regression

- Further example: Bayesian network representation of Bayesian linear regression model (N=3 in left figure)



- Bayesian linear regression with test instance:



Markov Model

- An important probabilistic model for sequential data is the so-called **Markov model**
- It models a joint distribution over a sequence of random variables $q_1, \ldots, q_T$:

$$p(q_1, \ldots, q_T) = p(q_1) \prod_{t=2}^{T} p(q_t \mid q_{t-1})$$

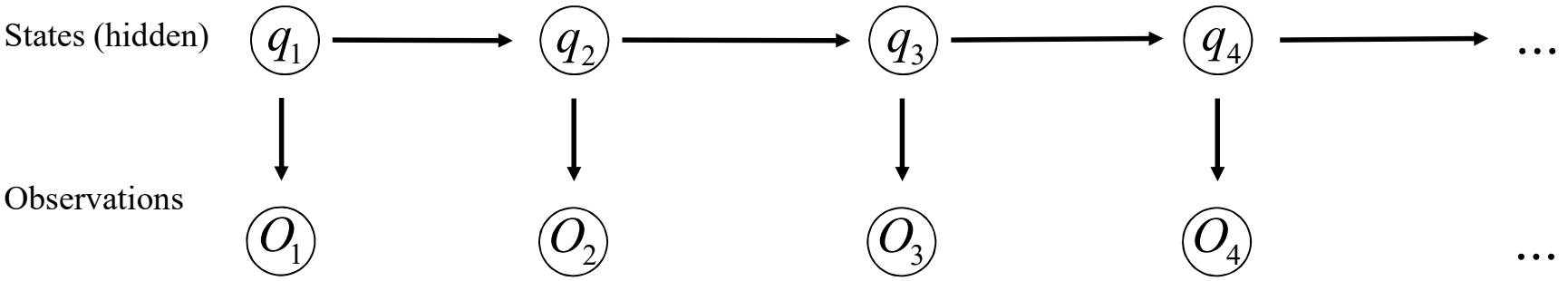
- In the joint distribution, every variable q_t only depends on its predecessor q_{t-1}: this is known as the **Markov assumption**

- Markov models are widely used to describe sequential data or sequential processes

Hidden Markov Model

- An extension of the Markov model is the so-called **hidden Markov model**
- It models a joint distribution over outputs $o_1, \ldots, o_T$ and hidden states $q_1, \ldots, q_T$:

$$p(q_1, \ldots, q_T, O_1, \ldots, O_T) = p(q_1)\, p(O_1 \mid q_1) \prod_{t=2}^{T} p(q_t \mid q_{t-1})\, p(O_t \mid q_t)$$



- Hidden Markov models are used for modeling sequential systems that are only indirectly observable
  - there is an „inner“ hidden state of the system that cannot be observed
  - the outputs allow us to indirectly observe or estimate the hidden states

Agenda

- Bayesian networks: syntax and semantics
- Independence in Bayesian networks
- Bayesian networks in machine learning
- A first look at Inference and parameter estimation

Inference Problem

- **Inference** in a Bayesian network is the computation of the joint distribution over a set of query variables, given observations on some of the other variables

- More formally, assume a Bayesian network that represents a joint distribution by

$$p(x_1,...,x_M) = \prod_{m=1}^{M} p(x_m \mid pa(x_m))$$

- Let
  - $x_{m_1},...,x_{m_K}$ be a set of **query variables**,
  - $x_{m_{K+1}},...,x_{m_L}$ be a set of **observed variables**, for which values are given,
  - and $x_{m_{L+1}},...,x_{m_M}$ the set of remaining variables (not query and not observed, also sometimes called **nuisance variables**)

- The **inference** task is to compute $p(x_{m_1},...,x_{m_K} \mid x_{m_{K+1}},...,x_{m_L})$

Inference Problem

- The rules of probability tell us how to compute the required conditional distribution:

$$p(x_{m_1},...,x_{m_K} \mid x_{m_{K+1}},...,x_{m_L}) = \frac{p(x_{m_1},...,x_{m_K},x_{m_{K+1}},...,x_{m_L})}{p(x_{m_{K+1}},...,x_{m_L})}$$

- It suffices to compute $p(x_{m_1},...,x_{m_K},x_{m_{K+1}},...,x_{m_L})$ for any possible joint state of the variables $x_{m_1},...,x_{m_K}$ (remember the values of $x_{m_{K+1}},...,x_{m_L}$ are fixed) and then normalize such that the probabilities sum to one

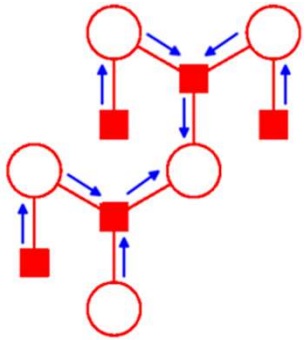
- The probabilities $p(x_{m_1},...,x_{m_K},x_{m_{K+1}},...,x_{m_L})$ can be computed by summing out the remaining variables:

$$p(x_{m_1},...,x_{m_K},x_{m_{K+1}},...,x_{m_L}) = \sum_{x_{m_{L+1}},...,x_{m_M}} p(x_{m_1},...,x_{m_K},x_{m_{K+1}},...,x_{m_L},x_{m_{L+1}},...,x_{m_M})$$

- What is the computational complexity?
  - Exponential in the number of query variables K (also for storing the results)
  - Exponential in the number of nuisance variables $M-(K+L)$

Inference Problem

- Better inference algorithms exist, especially with better complexity in the number of nuisance variables
  - Exact: by exploiting the graph structure, e.g. „message passing“



  - Approximate: e.g. by sampling values repeatedly (Markov-Chain Monte Carlo, MCMC)
  - ...

- More in the lecture on Bayesian networks!

Learning Bayesian Networks From Data

- Assume we have a Bayesian network with known graph structure encoding the structure of the joint distribution:

$$p(x_1,...,x_M) = \prod_{m=1}^{M} p(x_m \mid pa(x_m))$$

- Assume all variables are discrete
- To fully define the joint distribution, we need to specify for each variable $x_m \in \{x_1,...,x_M\}$ the conditional probabilities

$$p(x_m \mid pa(x_m)),$$

typically in the form of a table (see Alarm example above)

- Assume the conditional probabilities in the network are jointly parameterized by a parameter vector $\boldsymbol{\theta} \in \mathbb{R}^D$ (that is, $\boldsymbol{\theta}$ contains all the numbers in all of the probability tables), such that conditional distributions take the form

$$p(x_m \mid pa(x_m), \boldsymbol{\theta})$$

Learning Bayesian Networks From Data

- We can learn these conditional distributions (that is, the parameter vector $\boldsymbol{\theta}$) from a set of observations (training data) of the variables $\{x_1,...,x_M\}$

- Training data for the Bayesian network consist of a set of instances which each are a full observation of the joint state of all random variables in the network:
  - The overall data set is $\mathcal{D} = \{\mathbf{x}_1,...,\mathbf{x}_N\}$
  - Each instance is of the form $\mathbf{x}_n = (x_{n,1},...,x_{n,M})$, where $x_{n,m}$ is an observed value for the variable x_m

- We can train the network to maximize the likelihood

$$p(\mathcal{D}\,|\,\boldsymbol{\theta}) = \prod_{n=1}^{N} p(\mathbf{x}_n\,|\,\boldsymbol{\theta})$$

$$= \prod_{n=1}^{N} \prod_{m=1}^{M} p(x_{n,m}\,|\,pa(x_{n,m}),\boldsymbol{\theta})$$

Learning Bayesian Networks From Data

- Let for a particular variable x_m the set of parents be $pa(x_m) = \{x_{m_1}, ..., x_{m_K}\}$

- For any joint state of the parents $x_{m_1} = v_1, ..., x_{m_K} = v_K$ and state of the variable $x_m = v$ let the probability of the variable state given the parent states be given by

$$\theta_{v,v_1,...,v_K} = p(x_m = v \mid x_{m_1} = v_1, ..., x_{m_K} = v_K, \boldsymbol{\theta})$$

- Then the maximum likelihood solution is

$$\hat{\theta}_{v,v_1,...,v_K} = \frac{\sum_{n=1}^{N} I(x_{n,m} = v \wedge x_{n,m_1} = v_1 \wedge ... \wedge x_{n,m_K} = v_K)}{\sum_{n=1}^{N} I(x_{n,m_1} = v_1 \wedge ... \wedge x_{n,m_K} = v_K)}$$

how often was joint state observed?

how often was parent state observed?

- Can also introduce a regularizer for the parameter vector $\boldsymbol{\theta}$ or employ a Dirichlet prior for maximum a posteriori estimation in a Bayesian setting (no details)
- Can also learn from partial observations, but this is more involved (no details)

Summary: Bayesian Networks

- Bayesian networks are a formalism for representing the joint distribution over a set of random variables $\{x_1,...,x_M\}$

- The representation is based on a combination of a graph structure that expresses the structure of the distribution by

$$p(x_1,...,x_M) = \prod_{m=1}^{M} p(x_m \mid pa(x_m))$$

  and parameterized conditional probabilities $p(x_m \mid pa(x_m))$

- Independencies between variables can be derived by D-separation: a set of simple rules that check if all pathes of influence between the variables are blocked

- As a Bayesian network represents the joint distribution, any probabilistic inference query can in principle be answered (though naive algorithms are exponential)

- Bayesian networks can be learned from data by maximizing the likelihood

Further Reading

- Further reading: Murphy 2012, Chapter 10

- Kevin P. Murphy. Machine Learning: A Probabilistic Perspective. The MIT Press, 2012.