Machine Learning and Pervasive Computing

Stephan Sigg

Georg-August-University Goettingen, Computer Networks

21.01.2015

Overview and Structure

22.10.2014	Organisation
22.10.3014	Introduction (Def.: Machine learning, Supervised/Unsupervised, Examples)
29.10.2014	Machine Learning Basics (Toolchain, Features, Metrics, Rule-based)
05.11.2014	A simple Supervised learning algorithm
12.11.2014	Excursion: Avoiding local optima with random search
19.11.2014	–
26.11.2014	Bayesian learner
03.12.2014	–
10.12.2014	Decision tree learner
17.12.2014	k-nearest neighbour
07.01.2015	Support Vector Machines
14.01.2015	Artificial Neural Networks and Self Organizing Maps
21.01.2015	Hidden Markov models and Conditional random fields
28.01.2015	High dimensional data, Unsupervised learning
04.02.2015	Anomaly detection, Online learning, Recom. systems

Outline

Markov chains

Markov processes

- Intensively studied
- Major branch in the theory of stochastic processes

A. A. Markov (1856 – 1922)

Extended by A. Kolmogorov to chains of infinitely many states

- 'Anfangsgründe der Theorie der Markoffschen Ketten mit unendlich vielen möglichen Zuständen' (1936) [1]

[1] A. Kolmogorov, *Anfangsgründe der Theorie der Markoffschen Ketten mit unendlich vielen möglichen Zuständen*, 1936.

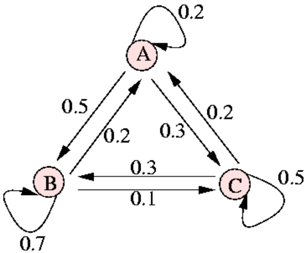
Markov chains

- Theory applied to a variety of algorithmic problems
- Standard tool in many probabilistic applications

Intuitive graphical representation

- Suitable for graphical illustration of stochastic processes

Popular for their simplicity and easy applicability to huge set of problems[2]



[2] William Feller, *An introduction to probability theory and its applications*, Wiley, 1968.

Markov chains

Independent trials of events

Dependent trials of events

Markov chains

Independent trials of events

- Set of possible outcomes of a measurement E_i associated with occurrence probability p_i
- Probability to observe sample sequence:
  - $P\{(E_1, E_2, \ldots, E_i)\} = p_1 p_2 \cdots p_i$

Dependent trials of events

Markov chains

Independent trials of events

- Set of possible outcomes of a measurement E_i associated with occurrence probability p_i
- Probability to observe sample sequence:
  - $P\{(E_1, E_2, \ldots, E_i)\} = p_1 p_2 \cdots p_i$

Dependent trials of events

- Probability to observe specific sequence $E_1, E_2, \ldots, E_i$ obtained by conditional probability:

$$P(E_i | E_1, E_2, \ldots, E_{i-1})$$

Markov chains

Independent random variables

Dependent random variables

Markov chains

Independent random variables

- Number of coin tosses until 'head' is observed
- Radioactive atoms always have same probability of decaying at next trial

Dependent random variables

Markov chains

Independent random variables

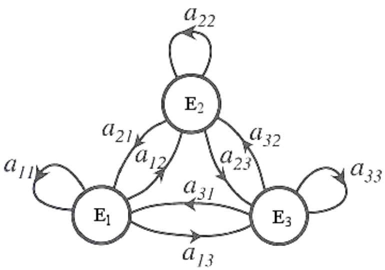
- Number of coin tosses until 'head' is observed
- Radioactive atoms always have same probability of decaying at next trial

Dependent random variables

- Knowledge that no car has passed for five minutes increases expectation that it will come soon.
- Coin tossing:
  - Probability that the cumulative numbers of heads and tails will equalize at the second trial is $\frac{1}{2}$
  - Given that they did not, the probability that they equalize after two additional trials is only $\frac{1}{4}$

Markov property

In the theory of stochastic processes the described lack of memory is connected with the Markov property.



Outcome depends exclusively on outcome of directly preceding trial

- Every sequence (E_i, E_j) has a conditional probability p_{ij}
- Additionally: Probability a_i of the event E_i

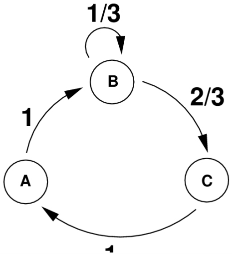
Markov chains

Markov chain

A sequence of observations $E_1, E_2, \ldots$ is called a Markov chain if the probabilities of sample sequences are defined by

$$P(E_1, E_2, \ldots, E_i) = a_1 \cdot p_{12} \cdot p_{23} \cdot \cdots \cdot p_{(i-1)i}.$$

and fixed conditional probabilities p_{ij} that the event E_i is observed directly in advance of E_j.



Markov chains

Described by probability a for initial distribution and matrix P of transition probabilities.

$$P = \begin{bmatrix} p_{11} & p_{12} & p_{13} & \cdots \\ p_{21} & p_{22} & p_{23} & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix}$$

P is called a stochastic matrix

(Square matrix with non-negative entries that sum to 1 in each row)

Markov chains

p_{ij}^k denotes probability that E_j is observed exactly k observations after E_i was observed.

Calculated as the sum of the probabilities for all possible paths $E_i E_{i_1} \cdots E_{i_{k-1}} E_j$ of length k

We already know

$$p_{ij}^1 = p_{ij}$$

Consequently:

$$\begin{aligned} p_{ij}^2 &= \sum_\nu p_{i\nu} \cdot p_{\nu j} \\ p_{ij}^3 &= \sum_\nu p_{i\nu} \cdot p_{\nu j}^2 \end{aligned}$$

Markov chains

By mathematical induction:

$$p_{ij}^{n+1} = \sum_{\nu} p_{i\nu} \cdot p_{\nu j}^{n}$$

and

$$p_{ij}^{n+m} = \sum_{\nu} p_{i\nu}^{m} \cdot p_{\nu j}^{n} = \sum_{\nu} p_{i\nu}^{n} \cdot p_{\nu j}^{m}$$

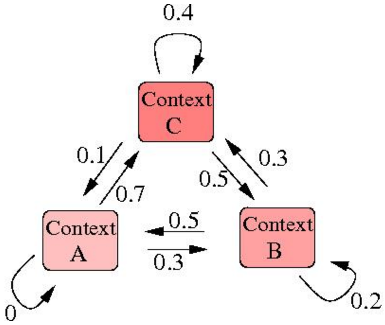
Similar to matrix P we can create a matrix P^n that contains all p_{ij}^n

p_{ij}^{n+1} obtained from P^{n+1}: Multiply row i of P with column j of P^n

Symbolically: $P^{n+m} = P^n P^m$.

$$P^n = \begin{bmatrix} p_{11}^n & p_{12}^n & p_{13}^n & \cdots \\ p_{21}^n & p_{22}^n & p_{23}^n & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix}$$

Markov chains



	Context A	Context B	Context C
Context A	0	0.3	0.7
Context B	0.5	0.2	0.3
Context C	0.1	0.5	0.4

	Context A	Context B	Context C
Context A	0.22	0.41	0.37
Context B	0.13	0.34	0.53
Context C	0.29	0.33	0.38

	Context A	Context B	Context C
Context A	0.242	0.333	0.425
Context B	0.223	0.372	0.405
Context C	0.203	0.343	0.454

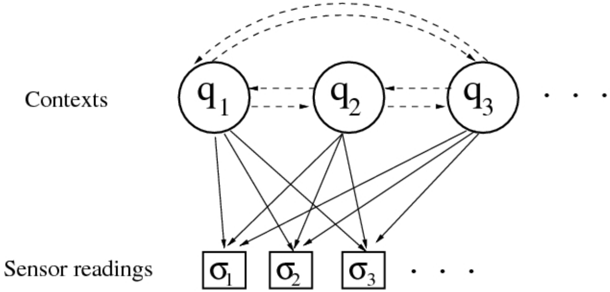
Outline

Hidden Markov Models

Make a sequence of decisions for a process that is not directly observable[3]

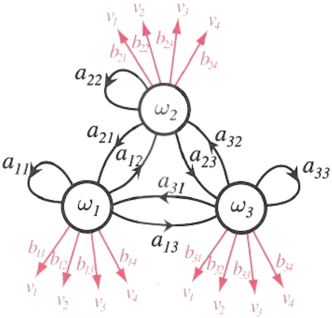
Current states of the process might be impacted by prior states

HMM often utilised in speech recognition or gesture recognition



[3] Richard O. Duda, Peter E. Hart and David G. Stork, *Pattern classification*, Wiley-interscience, 2001.

Hidden Markov Models

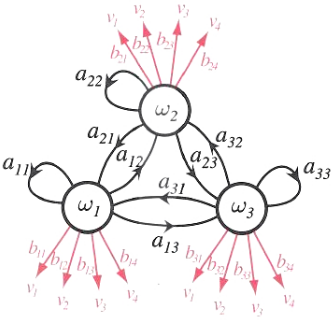


At every time step t the system is in an internal state $\omega(t)$

Additionally, we assume that it emits a (visible) symbol $v(t)$

Only access to visible symbols and not to internal states

Hidden Markov Models



Probability to be in state $\omega_j(t)$ and emit symbol $v_k(t)$:

$$P(v_k(t)|\omega_j(t)) = b_{jk}$$

Transition probabilities: $p_{ij} = P(\omega_j(t+1)|\omega_i(t))$
Emission probability: $b_{jk} = P(v_k(t)|\omega_j(t))$

Hidden Markov Models

Central issues in hidden Markov models:

Evaluation problem Determine the probability that a particular sequence of visible symbols V^n was generated by a given hidden Markov model

Decoding problem Determine the most likely sequence of hidden states ω^n that led to a specific sequence of observations V^n

Learning problem Given a set of training observations of visible symbols, determine the parameters p_{ij} and b_{jk} for a given HMM

Hidden Markov Models – Evaluation problem

Probability that model produces a sequence V^n:

$$P(V^n) = \sum_{\overline{\omega}^n} P(V^n|\overline{\omega}^n)P(\overline{\omega}^n)$$

Also:

$$\begin{aligned} P(\overline{\omega}^n) &= \prod_{t=1}^{n} P(\omega(t)|\omega(t-1)) \\ P(V^n|\overline{\omega}^n) &= \prod_{t=1}^{n} P(v(t)|\omega(t)) \end{aligned}$$

Together:

$$P(V^n) = \sum_{\overline{\omega}^n} \prod_{t=1}^{n} P(v(t)|\omega(t))P(\omega(t)|\omega(t-1))$$

Hidden Markov Models – Evaluation problem

Probability that model produces a sequence V^n:

$$P(V^n) = \sum_{\overline{\omega}^n} \prod_{t=1}^{n} P(v(t)|\omega(t))P(\omega(t)|\omega(t-1))$$

Formally complex but straightforward

Naive computational complexity

- $\mathcal{O}(c^n n)$

Hidden Markov Models – Evaluation problem

Probability that model produces a sequence V^n:

$$P(V^n) = \sum_{\overline{\omega}^n} \prod_{t=1}^{n} P(v(t)|\omega(t))P(\omega(t)|\omega(t-1))$$

Computationally less complex algorithm:

- Calculate $P(V^n)$ recursively
- $P(v(t)|\omega(t))P(\omega(t)|\omega(t-1))$ involves only $v(t), \omega(t)$ and $\omega(t-1)$

$$\alpha_j(t) = \begin{cases} 0 & t = 0 \text{ and } j \neq \text{ initial state} \\ 1 & t = 0 \text{ and } j = \text{ initial state} \\ \left[\sum_i \alpha_i(t-1)p_{ij}\right] b_{jk} & \text{otherwise } {\scriptstyle (b_{jk} \text{ leads to observed } v(t))} \end{cases}$$

Hidden Markov Models – Evaluation problem

Forward Algorithm

Computational complexity: $O(c^2 n)$

Forward algorithm

initialise $t \leftarrow 0, p_{ij}, b_{jk}, V^n, \alpha_j(0)$
    for $t \leftarrow t+1$
        $j \leftarrow 0$
        for $j \leftarrow j+1$
            $\alpha_j(t) \leftarrow b_{jk} \sum_{i=1}^{c} \alpha_i(t-1) p_{ij}$
        until $j = c$
    until $t = n$
return $P(V^n) \leftarrow \alpha_j(n)$ for the final state
end

Hidden Markov Models – Decoding problem

Given a sequence V^n, find most probable sequence of hidden states

Enumeration of every possible path will cost $O(c^n)$

- Not feasible

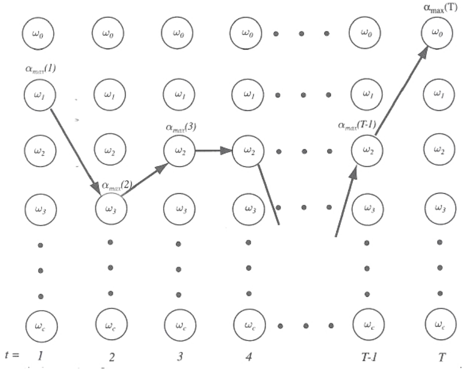
Hidden Markov Models – Decoding problem

Given a sequence V^n, find most probable sequence of hidden states

Decoding algorithm

initialise: path $\leftarrow \{\}, t \leftarrow 0$
    for $t \leftarrow t + 1$
        $j \leftarrow 0;$
        for $j \leftarrow j + 1$
            $\alpha_j(t) \leftarrow b_{jk} \sum_{i=1}^{c} \alpha_i(t-1) p_{ij}$
        until $j = c$
        $j' \leftarrow arg\max_j \alpha_j(t)$
        append $\omega_{j'}$ to path
    until $t = n$
return path
end

Hidden Markov Models – Decoding problem



Computational time of the decoding algorithm

- $O(c^2n)$

Hidden Markov Models – Learning problem

Determine the model parameters p_{ij} and b_{jk}

- Given: Training sample of observed values V^n

No method known to obtain the optimal or most likely set of parameters from the data

- However, we can nearly always determine a good solution by the forward-backward algorithm
- General expectation maximisation algorithm
- Iteratively update weights in order to better explain the observed training sequences

Hidden Markov Models – Learning problem

Probability that the model is in state $\omega_i(t)$ and will generate the remainder of the given target sequence:

$$\beta_i(t) = \begin{cases} 0 & t = n \text{ and } \omega_i(t) \text{ not final hidden state} \\ 1 & t = n \text{ and } \omega_i(t) \text{ final hidden state} \\ \sum_j \beta_j(t+1) p_{ij} b_{jk} & \text{otherwise } (b_{jk} \text{ leads to } v(t+1)) \end{cases}$$

Hidden Markov Models – Learning problem

$\alpha_i(t)$ and $\beta_i(t)$ only estimates of their true values since transition probabilities p_{ij}, b_{jk} unknown

Probability of transition between $\omega_i(t-1)$ and $\omega_j(t)$ can be estimated

- Provided that the model generated the entire training sequence V^n by **any** path

$$\gamma_{ij}(t) = \frac{\alpha(t-1)p_{ij}b_{jk}\beta_j(t)}{P(V^n|\Omega)}$$

Probability that model generated sequence V^n:

$$P(V^n|\Omega)$$

Hidden Markov Models – Learning problem

Calculate improved estimate for p_{ij} and b_{jk}

$$\overline{p_{ij}} = \frac{\sum_{t=1}^{n} \gamma_{ij}(t)}{\sum_{t=1}^{n} \sum_{k} \gamma_{ik}(t)}$$

$$\overline{b_{jk}} = \frac{\sum_{t=1,v(t)=v_k}^{n} \sum_{l} \gamma_{jl}(t)}{\sum_{t=1}^{n} \sum_{l} \gamma_{jl}(t)}$$

Start with rough estimates of p_{ij} and b_{jk}

Calculate improved estimates

Repeat until some convergence is reached

Hidden Markov Models – Learning problem

Forward-Backward algorithm

initialise p_{ij}, b_{jk}, V^n, convergence criterion $\Delta, t \leftarrow 0$
    do $t \leftarrow t+1$
        compute $\overline{p_{ij}(t)}$
        compute $\overline{b_{jk}(t)}$
        $p_{ij}(t) \leftarrow \overline{p_{ij}(t)}$
        $b_{jk}(t) \leftarrow \overline{b_{jk}(t)}$
    until $\max_{i,j,k}[p_{ij}(z) - p_{ij}(z-1), b_{jk}(t) - b_{jk}(t-1)] < \Delta$
                (convergence achieved)
return $p_{ij} \leftarrow p_{ij}(t)$, $b_{jk} \leftarrow b_{jk}(t)$
end

Outline

Markov chains

Hidden Markov Models

- Evaluation
- Deconding
- Learning

Probabilistic Graphical Models

Probabilistic graphical models

Introduction

In the previous models, probabilistic inference was a prominent aspect.

We will now discuss probabilistic graphical models

Some of the classification approaches discussed earlier can be described by such models

Probabilistic graphical models

Introduction

In the previous models, probabilistic inference was a prominent aspect.

We will now discuss probabilistic graphical models

Some of the classification approaches discussed earlier can be described by such models

Benefits of probabilistic graphical models

→ Simple way to visualise the structure of a probabilistic model

→ Insights into properties of the model, including conditional independence

→ Graphical representation of complex computations required to perform inference and learning

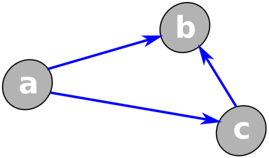
Probabilistic graphical models

Definition

A probabilistic graphical model comprises vertices connected by edges

Vertices represent random variables or groups of variables

Edges represent probabilistic relationships between variables

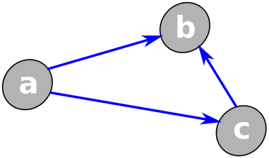


Probabilistic graphical models

Definition
A probabilistic graphical model comprises vertices connected by edges

Vertices represent random variables or groups of variables

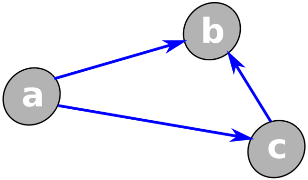
Edges represent probabilistic relationships between variables



Probabilistic graphical model

The graph captures the way in which the joint distribution over all of the random variables can be decomposed into a product of factors each depending only on a subset of variables

Probabilistic graphical models



Example

Consider an arbitrary joint distribution $\mathcal{P}[a, b, c]$.

We can then write

$$\begin{aligned}\mathcal{P}[a, b, c] &= \mathcal{P}[b|a, c]\mathcal{P}[a, c] \\ &= \mathcal{P}[b|a, c]\mathcal{P}[c|a]\mathcal{P}[a]\end{aligned}$$

Probabilistic graphical models

Example

Similarly we can define a joint distribution

$$\mathcal{P}[x_1, \ldots, x_n] = \mathcal{P}[x_n | x_1, \ldots, x_{n-1}] \ldots \mathcal{P}[x_2 | x_1] \mathcal{P}[x_1]$$

Probabilistic graphical models

Example

Similarly we can define a joint distribution

$$\mathcal{P}[x_1, \ldots, x_n] = \mathcal{P}[x_n | x_1, \ldots, x_{n-1}] \ldots \mathcal{P}[x_2 | x_1] \mathcal{P}[x_1]$$

These graphs are fully connected.
(One edge between every pair of nodes)

Probabilistic graphical models

Example

Similarly we can define a joint distribution

$$\mathcal{P}[x_1, \ldots, x_n] \quad = \quad \mathcal{P}[x_n | x_1, \ldots, x_{n-1}] \ldots \mathcal{P}[x_2 | x_1] \mathcal{P}[x_1]$$

These graphs are fully connected.
(One edge between every pair of nodes)

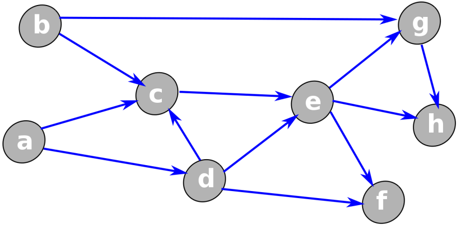
The actual absence of links in the graph covers intersting information about the properties of the class of distributions represented

Probabilistic graphical models

Definition

A general distribution for a graph with n nodes is

$$\mathcal{P}[x] = \prod_{i=1}^{n} \mathcal{P}[x_i | \text{parents of vertex } x_i]$$

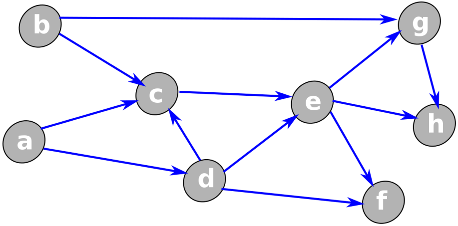


Probabilistic graphical models

Definition

A general distribution for a graph with n nodes is

$$\mathcal{P}[x] = \prod_{i=1}^{n} \mathcal{P}[x_i | \text{parents of vertex } x_i]$$



Remark: Bayesian networks are represented in this way

Probabilistic graphical models

Example: Bayesian Curve fitting

W Polynomial coefficients

$X = (x_1, \ldots, x_n)^T$ Input data

$Y = (y_1, \ldots, y_n)^T$ Observed data (Ground truth)

σ^2 Noise variance

α representation of the precision of the Gaussian prior over W

$$\mathcal{P}[Y, W] = \mathcal{P}[W] \prod_{i=1}^{n} \mathcal{P}[y_i | W]$$

(omitting deterministic parameters)

Probabilistic graphical models

Example: Bayesian Curve fitting

W Polynomial coefficients

$X = (x_1, \ldots, x_n)^T$ Input data

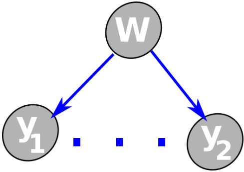
$Y = (y_1, \ldots, y_n)^T$ Observed data (Ground truth)

σ^2 Noise variance

α representation of the precision of the Gaussian prior over W

$$\mathcal{P}[Y, W] = \mathcal{P}[W] \prod_{i=1}^{n} \mathcal{P}[y_i | W]$$

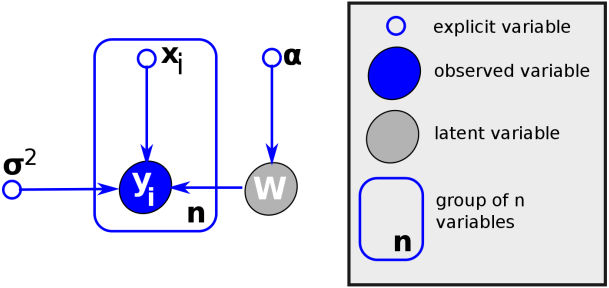
(omitting deterministic parameters)



Probabilistic graphical models

Example: Bayesian Curve fitting

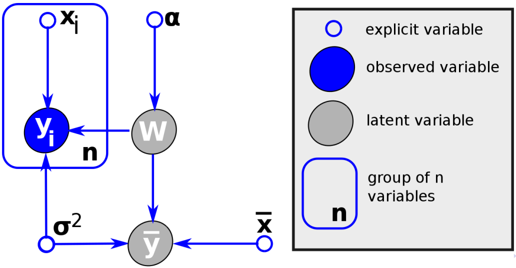
$$\mathcal{P}[Y, W | X, \alpha, \sigma^2] = \mathcal{P}[W|\alpha] \prod_{i=1}^{n} \mathcal{P}[y_i | W, x_i, \sigma^2]$$



Probabilistic graphical models

Prediction of $\overline{y}$ given the model and a new sample $\overline{x}$ as

$$\mathcal{P}[\overline{y}, Y, W|\overline{x}, X, \alpha, \sigma^2] \quad = \quad \left[\prod_{i=1}^{n} \mathcal{P}[y_i|W, x_i, \sigma^2]\right] \mathcal{P}[W|\alpha]\mathcal{P}[\overline{y}|\overline{x}, W, \sigma^2]$$



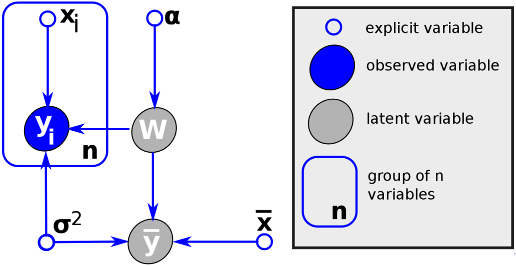
Probabilistic graphical models

Prediction of $\overline{y}$ given the model and a new sample $\overline{x}$ as

$$\mathcal{P}[\overline{y}, Y, W|\overline{x}, X, \alpha, \sigma^2] \quad = \quad \left[\prod_{i=1}^{n} \mathcal{P}[y_i|W, x_i, \sigma^2]\right] \mathcal{P}[W|\alpha]\mathcal{P}[\overline{y}|\overline{x}, W, \sigma^2]$$

Sum rule of probability leads to predictive distribution for $\overline{y}$:

$$\mathcal{P}[\overline{y}|\overline{x}, X, \alpha, Y, \sigma^2] \quad \propto \quad \int \mathcal{P}[\overline{y}, Y, W|\overline{x}, X, \alpha, \sigma^2]dW$$



Probabilistic graphical models

Conditional independence between nodes of the graph

Consider variables a, b and c and assume the conditional distribution

$$\mathcal{P}[a|b, c] = \mathcal{P}[a|c]$$

Then: a is conditionally independent of b given c

Probabilistic graphical models

Conditional independence between nodes of the graph

Consider variables a, b and c and assume the conditional distribution

$$\mathcal{P}[a|b, c] = \mathcal{P}[a|c]$$

Then: a is conditionally independent of b given c

Notation: $a \perp\!\!\!\perp b \mid c$

Probabilistic graphical models

Conditional independence between nodes of the graph

Consider variables a, b and c and assume the conditional distribution

$$\mathcal{P}[a|b,c] = \mathcal{P}[a|c]$$

Then: a is conditionally independent of b given c

Notation: $a \perp\!\!\!\perp b \mid c$

Importance of conditional independence in probabilistic models

Conditional independence in probabilistic models for pattern recognition

- simplifies the structure of a model and
- the computations needed to perform inference and learning

Probabilistic graphical models

Conditional independence between nodes of the graph

Conditional independence can be read directly from the graph !

Probabilistic graphical models

Conditional independence between nodes of the graph

Conditional independence can be read directly from the graph !

Example

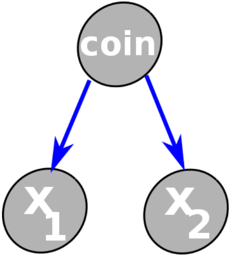
Assume a random experiment containing a biased and a fair coin.

$$\text{Biased: } \mathcal{P}[\text{head}] = 0.8,\ \mathcal{P}[\text{tail}] = 0.2$$

$$\text{Fair: } \mathcal{P}[\text{head}] = \mathcal{P}[\text{tail}] = 0.5$$

The experiment consists of two steps:

1. Choose which coin to toss
2. Toss the coin twice



Probabilistic graphical models

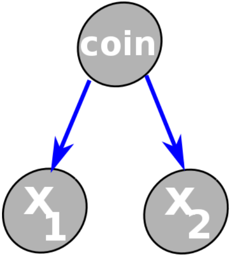
Conditional independence between nodes of the graph

Conditional independence can be read directly from the graph !

Example

If we are ignorant of which coin we chose, the result of the first toss impacts our expectation of what we see in the second toss:

→ e.g. if the first toss came out head, this will increase our expectation to see head also in the second toss



Probabilistic graphical models

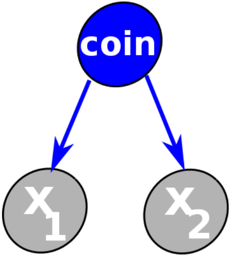
Conditional independence between nodes of the graph

Conditional independence can be read directly from the graph !

Example

However, if we were given information about which coin we chose, the x_1 and x_2 independent.

→ Since we know the distribution expected by both coins, knowledge of the outcome of x_1 does not change the expected outcome of x_2



Probabilistic graphical models

Conditional independence between nodes of the graph

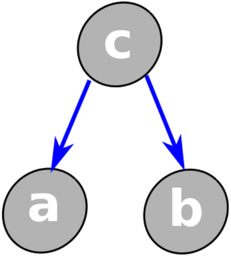
$$\mathcal{P}[a, b, c] = \mathcal{P}[a|c]\mathcal{P}[b|c]\mathcal{P}[c]$$

If none of the variables are observed, we can investigate whether a and b are independent by marginalizing both sides with respect to c:

$$\mathcal{P}[a, b] = \sum_c \mathcal{P}[a|c]\mathcal{P}[b|c]\mathcal{P}[c]$$

Since this does not factorize into $\mathcal{P}[a]\mathcal{P}[b]$ in general, we conclude

$$a \not\perp\!\!\!\perp b \mid \emptyset$$

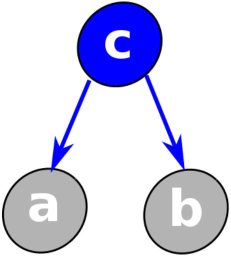


Probabilistic graphical models

Conditional independence between nodes of the graph

If, however, c is observed, we obtain

$$\begin{aligned}\mathcal{P}[a,b|c] &= \frac{\mathcal{P}[a,b,c]}{\mathcal{P}[c]} \\ &= \frac{\mathcal{P}[a|c]\mathcal{P}[b|c]\mathcal{P}[c]}{\mathcal{P}[c]} \\ &= \mathcal{P}[a|c]\mathcal{P}[b|c]\end{aligned}$$



And thus obtain the conditional independence property

$$a \perp\!\!\!\perp b \mid c$$

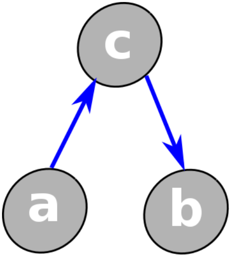
Probabilistic graphical models

Conditional independence between nodes of the graph

$$\mathcal{P}[a,b,c] = \mathcal{P}[a]\mathcal{P}[c|a]\mathcal{P}[b|c]$$

Marginalizing over c leads to

$$\begin{aligned}\mathcal{P}[a,b] &= \mathcal{P}[a]\sum_c \mathcal{P}[c|a]\mathcal{P}[b|c] \\ &= \mathcal{P}[a]\mathcal{P}[b|a]\end{aligned}$$



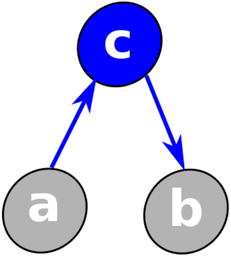
This does not factorize into $\mathcal{P}[a]\mathcal{P}[b]$ in general and therefore

$$a \not\perp\!\!\!\perp b \mid \emptyset$$

Probabilistic graphical models

Conditional independence between nodes of the graph

$$
\begin{aligned}
\mathcal{P}[a, b|c] &= \frac{\mathcal{P}[a, b, c]}{\mathcal{P}[c]} \\
&= \frac{\mathcal{P}[a]\mathcal{P}[c|a]\mathcal{P}[b|c]}{\mathcal{P}[c]} \\
&= \mathcal{P}[a|c]\mathcal{P}[b|c]
\end{aligned}
$$



And therefore

$$a \perp\!\!\!\perp b \mid c$$

Probabilistic graphical models

Conditional independence between nodes of the graph

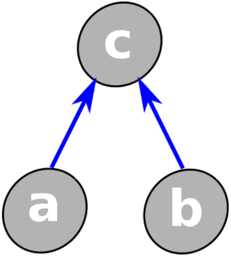
$$\mathcal{P}[a, b, c] = \mathcal{P}[a]\mathcal{P}[b]\mathcal{P}[c|a, b]$$

Marginalizing over c leads to

$$\mathcal{P}[a, b] = \mathcal{P}[a]\mathcal{P}[b]$$

So, in this case, we obtain

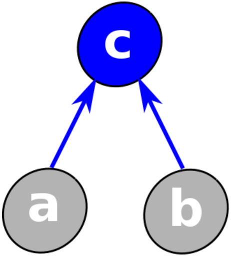
$$a \perp\!\!\!\perp b \mid \emptyset$$



Probabilistic graphical models

Conditional independence between nodes of the graph

$$
\begin{aligned}
\mathcal{P}[a, b|c] &= \frac{\mathcal{P}[a, b, c]}{\mathcal{P}[c]} \\
&= \frac{\mathcal{P}[a]\mathcal{P}[b]\mathcal{P}[c|a, b]}{\mathcal{P}[c]}
\end{aligned}
$$



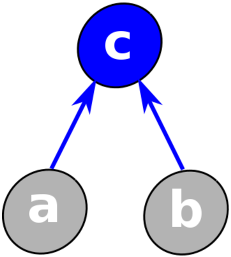
Which does not in general factorize into $\mathcal{P}[a|c]\mathcal{P}[b|c]$ and so

$$a \not\perp\!\!\!\perp b \mid c$$

Probabilistic graphical models

Conditional independence between nodes of the graph

$$\mathcal{P}[a, b|c] = \frac{\mathcal{P}[a, b, c]}{\mathcal{P}[c]} = \frac{\mathcal{P}[a]\mathcal{P}[b]\mathcal{P}[c|a, b]}{\mathcal{P}[c]}$$



Which does not in general factorize into $\mathcal{P}[a|c]\mathcal{P}[b|c]$ and so

$$a \not\perp\!\!\!\perp b \mid c$$

This rule applies also if, instead of c, any its descendants are observed !

Probabilistic graphical models

Conditional independence between nodes of the graph

D-separation

Consider a general directed graph in which A, B and C are arbitrary nonintersecting sets of nodes

A is d-separated from B by C when all possible paths from A to B contain a node such that either

a) the node is in the set C and the arrows meet head-to-tail or tail-to-tail

b) the node is not in the set C nor any of its descendants and the arrows meet head-to-head

Probabilistic graphical models

The concept of d-separation helps us to understand the probability distributions that are expressed by a particular graphical model:

Probabilistic graphical models

The concept of d-separation helps us to understand the probability distributions that are expressed by a particular graphical model:

We have seen above that the joint distribution of a graph is given as its factorization:

$$\mathcal{P}[x] = \prod_{i=1}^{n} \mathcal{P}[x_i | \text{parents of vertex } x_i]$$

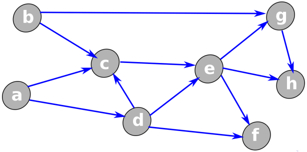
Probabilistic graphical models

The concept of d-separation helps us to understand the probability distributions that are expressed by a particular graphical model:

We have seen above that the joint distribution of a graph is given as its factorization:

$$\mathcal{P}[x] = \prod_{i=1}^{n} \mathcal{P}[x_i | \text{parents of vertex } x_i]$$

The graph literally filters those distributions which can express it in terms of the factorization implied by the graph.



Probabilistic graphical models

The concept of d-separation helps us to understand the probability distributions that are expressed by a particular graphical model:

We have seen above that the joint distribution of a graph is given as its factorization:

$$\mathcal{P}[x] = \prod_{i=1}^{n} \mathcal{P}[x_i | \text{parents of vertex } x_i]$$

The graph literally filters those distributions which can express it in terms of the factorization implied by the graph.

It can be shown that the set of distributions that pass the filter is precisely the set of distributions that fulfills the set of conditional independence properties defined by the d-separation property.

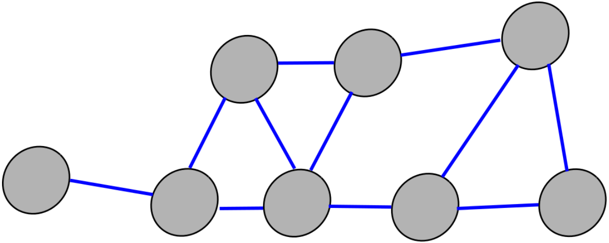
Probabilistic graphical models

Undirected graphical models

Undirected graphical models

Also graphical models that are described by undirected graphs specify

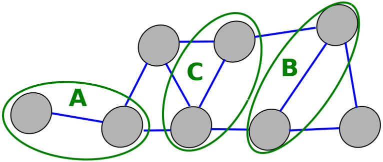
a) a factorization
b) a set of conditional independence relations



Probabilistic graphical models

Undirected graphical models

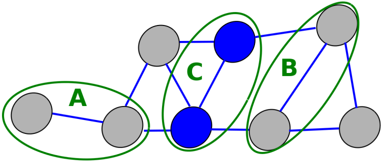
Assume three test of nodes A, B and C in such an undirected graph



Probabilistic graphical models

Undirected graphical models

Assume three test of nodes A, B and C in such an undirected graph



Conditional independence in undirected graphs

$A \perp\!\!\!\perp B \mid C$ if all paths between A and B contain an observed node from the set C

$A \not\!\perp\!\!\!\perp B \mid C$ if at least one path between A and B does not contain any observed node.

Probabilistic graphical models

Factorization rule for undirected graphs

Two nodes *a* and *b* in a graph are conditionally independent (given all other nodes) if they are not connected by an edge

$\rightarrow$ Since there is no direct path between the nodes

Probabilistic graphical models

Factorization rule for undirected graphs

Two nodes a and b in a graph are conditionally independent (given all other nodes) if they are not connected by an edge

$\rightarrow$ Since there is no direct path between the nodes

Therefore, the joint distribution described by the graph is given by functions of the variables of the maximal cliques in the graph

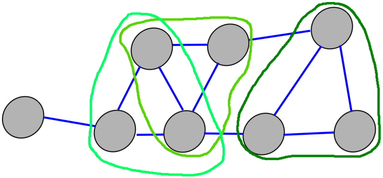
Probabilistic graphical models

Factorization rule for undirected graphs

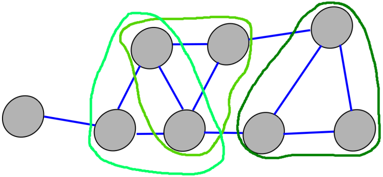
Two nodes a and b in a graph are conditionally independent (given all other nodes) if they are not connected by an edge

$\rightarrow$ Since there is no direct path between the nodes

Therefore, the joint distribution described by the graph is given by functions of the variables of the maximal cliques in the graph



Probabilistic graphical models



The joint distribution is written as a product of potential functions $\phi_C(X_C)$ over the maximal cliques X_C of the graph:

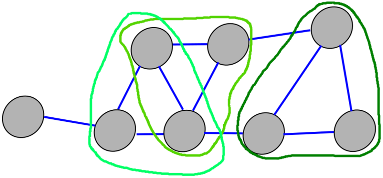
$$\mathcal{P}[X] = \frac{1}{Z} \prod_C \phi_C(X_C)$$

Here, Z is a normalisation constant given by

$$Z = \sum_X \prod_C \phi_C(X_C)$$

to ensure that the distribution $\mathcal{P}[X]$ is correctly normalised.

Probabilistic graphical models



The joint distribution is written as a product of potential functions $\phi_C(X_C)$ over the maximal cliques X_C of the graph:

$$\mathcal{P}[X] = \frac{1}{Z}\prod_C \phi_C(X_C)$$

Gibbs distribution

Here, Z is a normalisation constant given by

$$Z = \sum_X \prod_C \phi_C(X_C)$$

to ensure that the distribution $\mathcal{P}[X]$ is correctly normalised.

Probabilistic graphical models

Conditional random fields

Distinguishing between observed variables X and target variables Y, in the unnormalized measure

$$\mathcal{P}[X, Y] = \prod_C \phi_C(X_C)$$

we can define a conditional random field as

$$\begin{aligned} \mathcal{P}[Y|X] &= \frac{1}{Z(X)} \prod_C \phi_C(X_C) \\ Z(X) &= \sum_X \mathcal{P}[X, Y] \end{aligned}$$

Probabilistic graphical models

Conditional random fields

Distinguishing between observed variables X and target variables Y, in the unnormalized measure

$$\mathcal{P}[X, Y] = \prod_C \phi_C(X_C)$$

we can define a conditional random field as

$$\begin{aligned} \mathcal{P}[Y|X] &= \frac{1}{Z(X)} \prod_C \phi_C(X_C) \\ Z(X) &= \sum_X \mathcal{P}[X, Y] \end{aligned}$$

Compared to the Bayesian models represented in directed graphs, the CRF removes from the model any dependency between the input variables x_i

Outline

Markov chains

Hidden Markov Models
- Evaluation
- Deconding
- Learning

Probabilistic Graphical Models

Questions?

Stephan Sigg
stephan.sigg@cs.uni-goettingen.de

Literature

- C.M. Bishop: Pattern recognition and machine learning, Springer, 2007.
- R.O. Duda, P.E. Hart, D.G. Stork: Pattern Classification, Wiley, 2001.