Machine Learning

A. Supervised Learning

A.8. A First Look at Bayesian and Markov Networks

Lars Schmidt-Thieme

Information Systems and Machine Learning Lab (ISMLL)
Institute for Computer Science
University of Hildesheim, Germany

Outline

Syllabus

Tue. 21.10.	(1)	0. Introduction
		A. Supervised Learning
Wed. 22.10.	(2)	A.1 Linear Regression
Tue. 28.10.	(3)	A.2 Linear Classification
Wed. 29.10.	(4)	A.3 Regularization
Tue. 4.11.	(5)	A.4 High-dimensional Data
Wed. 5.11.	(6)	A.5 Nearest-Neighbor Models
Tue. 11.11.	(7)	A.6 Decision Trees
Wed. 12.12.	(8)	A.7 Support Vector Machines
Tue. 18.11.	(9)	A.8 A First Look at Bayesian and Markov Networks
		B. Unsupervised Learning
Wed. 19.11.	(10)	B.1 Clustering
Tue. 25.11.	(11)	B.2 Dimensionality Reduction
Wed. 26.11.	(12)	B.3 Frequent Pattern Mining
		C. Reinforcement Learning
Tue. 2.12.	(13)	C.1 State Space Models
Wed. 3.12.	(14)	C.2 Markov Decision Processes

Outline

Joint Distribution

x_1 : the sun shines

$$\left.\begin{array}{l} p(x_1 = \text{false}) = 0.25 \\ p(x_1 = \text{true}) = 0.75 \end{array}\right\} \equiv p(x_1) = \begin{array}{|cc} \text{false} & \text{true} \\ \hline 0.25 & 0.75 \end{array} = (0.25, 0.75)$$

Joint Distribution

x_1 : the sun shines

$$\left.\begin{array}{l} p(x_1 = \text{false}) = 0.25 \\ p(x_1 = \text{true}) = 0.75 \end{array}\right\} \equiv p(x_1) = \begin{array}{|cc} \text{false} & \text{true} \\ \hline 0.25 & 0.75 \end{array} = (0.25, 0.75)$$

x_2 : it rains

$$\left.\begin{array}{l} p(x_2 = \text{false}) = 0.67 \\ p(x_2 = \text{true}) = 0.33 \end{array}\right\} \equiv p(x_2) = \begin{array}{|cc} \text{false} & \text{true} \\ \hline 0.67 & 0.33 \end{array} = (0.67, 0.33)$$

Joint Distribution

x_1 : the sun shines

$$\left.\begin{array}{l} p(x_1 = \text{false}) = 0.25 \\ p(x_1 = \text{true}) = 0.75 \end{array}\right\} \equiv p(x_1) = \begin{array}{|cc} \text{false} & \text{true} \\ \hline 0.25 & 0.75 \end{array} = (0.25, 0.75)$$

x_2 : it rains

$$\left.\begin{array}{l} p(x_2 = \text{false}) = 0.67 \\ p(x_2 = \text{true}) = 0.33 \end{array}\right\} \equiv p(x_2) = \begin{array}{|cc} \text{false} & \text{true} \\ \hline 0.67 & 0.33 \end{array} = (0.67, 0.33)$$

joint distribution:

$$\left.\begin{array}{ll} p(x_1 = \text{false}, x_2 = \text{false}) & = 0.07 \\ p(x_1 = \text{false}, x_2 = \text{true}) & = 0.18 \\ p(x_1 = \text{true}, x_2 = \text{false}) & = 0.6 \\ p(x_1 = \text{true}, x_2 = \text{true}) & = 0.15 \end{array}\right\} \equiv$$

$p(x_1, x_2)$		x_2 false	x_2 true
x_1	false	0.07	0.18
	true	0.6	0.15

Joint Distribution

x_1 : the sun shines

$$\left.\begin{array}{l} p(x_1 = \text{false}) = 0.25 \\ p(x_1 = \text{true}) = 0.75 \end{array}\right\} \equiv p(x_1) = \begin{array}{|cc} \text{false} & \text{true} \\ \hline 0.25 & 0.75 \end{array} = (0.25, 0.75)$$

x_2 : it rains

$$\left.\begin{array}{l} p(x_2 = \text{false}) = 0.67 \\ p(x_2 = \text{true}) = 0.33 \end{array}\right\} \equiv p(x_2) = \begin{array}{|cc} \text{false} & \text{true} \\ \hline 0.67 & 0.33 \end{array} = (0.67, 0.33)$$

joint distribution:

$$p(x_1, x_2) = \begin{array}{cc|cc} & & \multicolumn{2}{c}{x_2} \\ & & \text{false} & \text{true} \\ \hline x_1 & \text{false} & 0.07 & 0.18 \\ & \text{true} & 0.6 & 0.15 \end{array} = \begin{pmatrix} 0.07 & 0.18 \\ 0.6 & 0.15 \end{pmatrix}$$

Independence

for two variables:

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

for two variable subsets:

$$p(x_1, x_2, \ldots, x_M) = p(x_I) \cdot p(x_J), \quad I, J \subseteq \{1, \ldots, M\}, I \cap J = \emptyset$$

Note: $x_I := \{x_{m_1}, x_{m_2}, \ldots, x_{m_K}\}$ for $I := \{m_1, m_2, \ldots, m_K\}$.

Independence

for two variables:

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

for two variable subsets:

$$p(x_1, x_2, \ldots, x_M) = p(x_I) \cdot p(x_J), \quad I, J \subseteq \{1, \ldots, M\}, I \cap J = \emptyset$$

Examples:

$$\begin{pmatrix} 0.07 & 0.18 \\ 0.6 & 0.15 \end{pmatrix}$$

not independent

$$\begin{pmatrix} 0.17 & 0.08 \\ 0.5 & 0.25 \end{pmatrix}$$

independent

Note: $x_I := \{x_{m_1}, x_{m_2}, \ldots, x_{m_K}\}$ for $I := \{m_1, m_2, \ldots, m_K\}$.

Chain Rule

$$
\begin{aligned}
p(x_1, x_2, \ldots, x_M) = & p(x_1) \\
& \cdot p(x_2 \mid x_1) \\
& \cdot p(x_3 \mid x_1, x_2) \\
& \vdots \\
& \cdot p(x_M \mid x_1, x_2, \ldots, x_{m-1})
\end{aligned}
$$

Chain Rule

$$
\begin{aligned}
p(x_1, x_2, \ldots, x_M) = & p(x_1) \\
& \cdot p(x_2 \mid x_1) \\
& \cdot p(x_3 \mid x_1, x_2) \\
& \vdots \\
& \cdot p(x_M \mid x_1, x_2, \ldots, x_{m-1})
\end{aligned}
$$

Examples:

$$
\begin{pmatrix} 0.07 & 0.18 \\ 0.6 & 0.15 \end{pmatrix} = (0.25, 0.75) \cdot \begin{pmatrix} 0.28 & 0.72 \\ 0.8 & 0.2 \end{pmatrix}
$$

Chain Rule

$$p(x_1, x_2, \ldots, x_M) = p(x_1) \cdot p(x_2 \mid x_1) \cdot p(x_3 \mid x_1, x_2) \cdots \cdot p(x_M \mid x_1, x_2, \ldots, x_{m-1})$$

Examples:

$$\begin{pmatrix} 0.17 & 0.08 \\ 0.5 & 0.25 \end{pmatrix} = (0.25, 0.75) \cdot \begin{pmatrix} 0.67 & 0.33 \\ 0.67 & 0.33 \end{pmatrix}$$

Conditional Independence

two variables x, y are **independent conditionally on variable** z:

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

two variable sets are **independent conditionally on variables** $z_1, \ldots, z_K$:

$$\{x_1, \ldots, x_I\} \perp \{y_1, \ldots, y_J\} \mid \{z_1, \ldots, z_K\} :\Leftrightarrow$$
$$p(x_1, \ldots, x_I, y_1, \ldots, y_J \mid z_1, \ldots, z_K) = p(x_1, \ldots, x_I \mid z_1, \ldots, z_K) \cdot p(y_1, \ldots, y_J \mid z_1, \ldots, z_K)$$

Conditional Independence / Example

Example:

$$x_n \perp \{x_1, \ldots, x_{n-1}\} \mid x_{n-1} \quad \forall n \text{ (Markov property)}$$
$$\rightsquigarrow p(x_1, \ldots, x_N) = p(x_1)p(x_2 \mid x_1)p(x_3 \mid x_2) \cdots p(x_M \mid x_{M-1})$$

Graphical Models

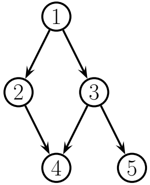
- represent joint distributions of variables by graphs
  - by directed graphs: **Bayesian networks**
  - by undirected graphs: **Markov networks**
  - by mixed directed/undirected graphs.
- nodes represent random variables
- absent edges represent conditional independence

Directed Graph Terminology

- **directed graph**: $G := (V, E),\ E \subseteq V \times V$
  - V set called **nodes** / **vertices**
  - E called **edges**, $(v, w) \in E$ edge from v to w.
- **adjacency matrix** $A \in \{0, 1\}^{N \times N}$

$$A_{v,w} := \delta((v, w) \in E), \quad v, w \in \{1, \ldots, N\}, N := |V|$$

- **parents**: $\text{pa}(v) := \{w \in V \mid (w, v) \in E\}$
- **children**: $\text{ch}(v) := \{w \in V \mid (v, w) \in E\}$
- **neighbors**: $\text{nbr}(v) := \text{pa}(v) \cup \text{ch}(v)$
- **family**: $\text{fam}(v) := \text{pa}(v) \cup \{v\}$
- **root**: v without parents.
- **leaf**: v without children.

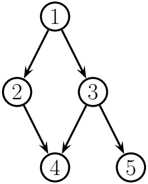


[Mur12, fig. 10.1a]

Note: $\delta(P) := 1$ if proposition P is true, $:= 0$ otherwise.

Directed Graph Terminology

- **path**: $p \in V^*$: $(p_i, p_{i+1}) \in E$ for all i.
  - $p = (p_1, \ldots, p_M)$, $p_m \in V$
  - **length** $|p| := M$
  - **starts at** p_1
  - **ends at** p_M
  - **paths** $G^* := \{p \in V^* \mid (p_i, p_{i+1}) \in E \quad \forall i = 1, \ldots, |p| - 1\}$.
  - $v \rightsquigarrow w$: **exists path from v to w**, i.e., $p \in G^* : p_1 = v, p_{|p|} = w$.
- **ancestors**: $\text{anc}(v) := \{w \in V \mid w \rightsquigarrow v\}$
- **descendants**: $\text{desc}(v) := \{w \in V \mid v \rightsquigarrow w\}$
- **in-degree** $|\text{pa}(v)|$
- **out-degree** $|\text{ch}(v)|$
- **degree** $|\text{nbr}(v)|$

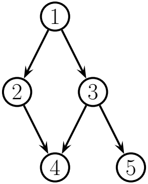


[Mur12, fig. 10.1a]

Note: $V^* := \bigcup_{M \in \mathbb{N}} V^M$ **finite V-sequences**.

Directed Graph Terminology

- **cycle**/**loop** at v: $v \rightsquigarrow v$
  - **self loop**: $(v, v) \in E$
- **directed acyclic graph / DAG**: directed graph without cycles.
- **topological ordering**: directed graph without cycles.
  - numbering of the nodes s.t. all nodes have lower number than their children.
  - exists for DAGs.



[Mur12, fig. 10.1a]

Bayesian Networks / Directed Graphical Models

A **Bayesian network** (aka **directed graphical model**) is a set of **conditional probability distributions/densities (CPDs)**

$$p(x_m \mid x_{\text{ctxt}(m)}), \quad m \in \{1, \ldots, M\}$$

s.t. the graph defined by

$$\begin{aligned} V :=& \{1, \ldots, M\} \\ E :=& \{(n, m) \mid m \in V, n \in \text{ctxt}(m)\}, \quad \text{i.e., } \text{pa}(m) := \text{ctxt}(m) \end{aligned}$$

is a DAG.

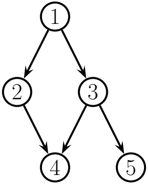
A Bayesian network defines a **factorization of the joint distribution**

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

Bayesian Networks / Example

For the DAG below,

$$p(x_1, x_2, x_3, x_4, x_5) = p(x_1)\, p(x_2 \mid x_1)\, p(x_3 \mid x_1)\, p(x_4 \mid x_2, x_3)\, p(x_5 \mid x_3)$$



[Mur12, fig. 10.1a]

Bayesian Networks / Example

For the DAG below,

$$p(x_1, x_2, x_3, x_4, x_5) = p(x_1)\, p(x_2 \mid x_1)\, p(x_3 \mid x_1)\, p(x_4 \mid x_2, x_3)\, p(x_5 \mid x_3)$$

If

- all variables are binary and
- all CPDs given as **conditional probability tables (CPTs)**,

then the BN is defined by the following 5 CPTs:

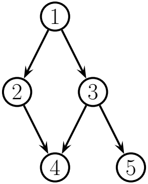
x_1	
0	...
1	...

	x_1	
x_2	0	1
0	...	...
1	...	...

	x_1	
x_3	0	1
0	...	...
1	...	...

	x_2	0		1	
	x_3	0	1	0	1
x_4	0	...	...	...	...
	1	...	...	...	...

	x_3	
x_5	0	1
0	...	...
1	...	...

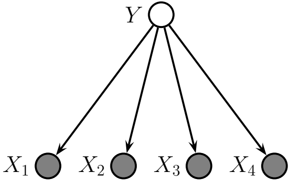


[Mur12, fig. 10.1a]

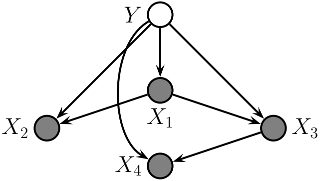
Outline

Naive Bayes Classifier

$$p(y, x_1, \ldots, x_M) = p(y)p(x_1 \mid y)p(x_2 \mid y) \cdots p(x_M \mid y)$$
$$= p(y) \prod_{m=1}^{M} p(x_m \mid y)$$



Naive Bayes Classifier



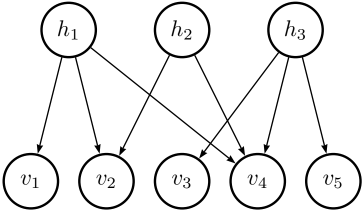
Tree Augmented Naive Bayes

[Mur12, fig. 10.2]

Medical Diagnosis

- bipartite graph
- observed variables $x_1, \ldots, x_M$ (symptoms)
- hidden variables $z_1, \ldots, z_K$ (diseases / causes)

$$p(x_1, \ldots, x_M, z_1, \ldots, z_M) = \prod_{k=1}^{K} p(z_k) \prod_{m=1}^{M} p(x_m \mid z_{\mathrm{pa}(m)})$$



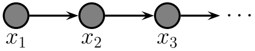
Note: In the diagram z is called h and x is called v.

[Mur12, fig. 10.5b]

Markov Models

first order:

$$p(x_1, \ldots, x_M) = p(x_1)p(x_2 \mid x_1)p(x_3 \mid x_2) \cdots p(x_M \mid x_{M-1})$$
$$= p(x_1) \prod_{m=1}^{M-1} p(x_{m+1} \mid x_m)$$

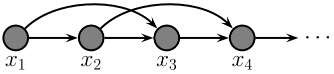


[Mur12, fig. 10.3a]

Markov Models / Second Order

second order:

$$p(x_1, \ldots, x_M) = p(x_1, x_2)p(x_3 \mid x_1, x_2)p(x_4 \mid x_2, x_3) \cdots p(x_M \mid x_{M-2}, x_{M-1})$$
$$= p(x_1, x_2) \prod_{m=2}^{M-1} p(x_{m+1} \mid x_{m-1}, x_m)$$



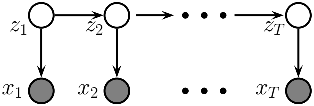
[Mur12, fig. 10.3b]

Hidden Markov Models

- observed variables $x_1, \ldots, x_M$
- hidden variables $z_1, \ldots, z_M$

$$p(x_1, \ldots, x_M, z_1, \ldots, z_M) = p(z_1) \prod_{m=1}^{M-1} p(z_{m+1} \mid z_m) \prod_{m=1}^{M} p(x_m \mid z_m)$$

- **transition model** $p(z_{m+1} \mid z_m)$
- **observation model** $p(x_m \mid z_m)$



[Mur12, fig. 10.4]

Outline

The Probabilistic Inference Problem

Given

- a Bayesian model $\theta := G = (V, E)$,
- a **query** consisting of
  - a set $X := \{x_1, \ldots, x_M\} \subseteq V$ of **predictor variables** (aka **observed**, **visible variables**)
  - with a **value** v_m for each x_m $(m = 1, \ldots, M)$ and
  - a set $Y := \{y_1, \ldots, y_J\} \subseteq V$ of **target variables** (aka **query variables**), with $X \cap Y = \emptyset$,

compute

$$p(Y \mid X = v; \theta) := p(y_1, \ldots, y_J \mid x_1 = v_1, x_2 = v_2, \ldots, x_M = v_M; \theta)$$
$$= (p(y_1 = w_1, \ldots, y_J = w_J \mid x_1 = v_1, x_2 = v_2, \ldots, x_M = v_M; \theta))_{w_1, \ldots, w_J}$$

Variables that are neither predictor variables nor target variables are called **nuisance variables**.

Inference Without Nuisance Variables

Without nuisance variables: $V = X \dot\cup Y$.

$$p(Y \mid X = v; \theta) \stackrel{\text{def}}{=} \frac{p(X = v, Y; \theta)}{p(X = v; \theta)} = \frac{p(X = v, Y; \theta)}{\sum_w p(X = v, Y = w; \theta)}$$

- first, clamp predictors X to their observed values v,
- then, normalize $p(X = v, Y; \theta)$ to sum to 1 (over Y).
- $p(X = v; \theta)$ **likelihood of the data** / **probability of evidence** is a constant.

Note: Summation over w is over all possible values of variables Y.

Inference With Nuisance Variables

Nuisance variables: $Z := \{z_1, \ldots, z_K\} := V \setminus (X \dot\cup Y)$.

1. add to target variables
2. answer resulting query without nuisance variables: $p(Y, Z \mid X)$.
3. **marginalize out** nuisance variables:

$$p(Y \mid X = v; \theta) \overset{\text{marginalization}}{=} \sum_u p(Y, Z = u \mid X = v; \theta)$$

Note: Summation over u is over all possible values of variables Z.

Inference With Nuisance Variables

Nuisance variables: $Z := \{z_1, \ldots, z_K\} := V \setminus (X \dot{\cup} Y)$.

1. add to target variables
2. answer resulting query without nuisance variables: $p(Y, Z \mid X)$.
3. **marginalize out** nuisance variables:

$$p(Y \mid X = v; \theta) \overset{\text{marginalization}}{=} \sum_u p(Y, Z = u \mid X = v; \theta)$$

Caveat: This is a naive algorithm never used in practice. See BN lecture for practically useful BN inference algorithms.

Note: Summation over u is over all possible values of variables Z.

Complexity of Inference

- for simplicity assume
  - all M predictor variables are nominal with L levels,
  - all K nuisance variables are nominal with L levels,
  - a single target variable: $Y = \{y\}, J = 1$
    also nominal with L levels.

- without (Conditional) Independencies:
  - full table p requires $L^{M+K+1} - 1$ cells storage.
  - inference requires $O(L^{K+1})$ operations.
    - for each $Y = w$ sum over all L^K many $Z = u$.

- with (Conditional) Independencies / Bayesian network:
  - CPDs p require $O((M + K + 1)L^{\text{max indegree}+1})$ cells storage.
  - inference requires $O((K + 1)L^{\text{treewidth}+1})$ operations.
    - treewidth=1 for a chain!

Note: See the Bayesian networks lecture for BN inference algorithms.

Outline

Learning Bayesian Networks

- **parameter learning**: given
  - the structure of the network (graph G) and
  - a regularization penalty $\text{Reg}(\theta)$,
  - data $x_1, \ldots, x_N$,

  learn the **CPDs** p.

$$\hat{\theta} := \arg\max_{\theta} \sum_{n=1}^{N} \log p(x_n; \theta) + \text{Reg}(\theta)$$

- **structure learning**: given
  - data,

  learn the **structure** G and the **CPDs** p.

Bayesian Approach

- in the Bayesian approach, parameters are also considered to be random variables, thus,
- learning is just a special type of inference (with the parameters as targets)
- information about the distribution of the parameters before seeing the data is required (**prior distribution** $p(\theta)$)

- **parameter learning**: given
  - the structure of the network (graph G) and
  - a prior distribution $p(\theta)$ of the parameters,
  - data $x_1, \ldots, x_N$,

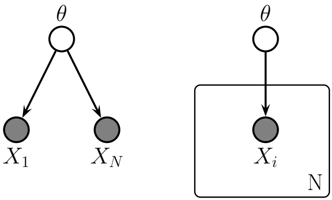
  learn the **CPDs** p.

$$\hat{\theta} := \arg\max_{\theta} \sum_{n=1}^{N} \log p(x_n; \theta) + \log p(\theta)$$

Plate Notation

- variables on plates are **duplicated**
  - the number of copies is given in the lower right corner.
- an **index** is used to differentiate copies of the same variable.

Example 1: data $x_1, \ldots, x_N$ is independently identically distributed (iid)

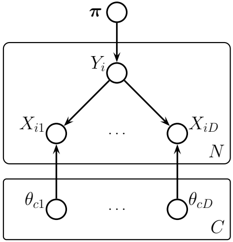


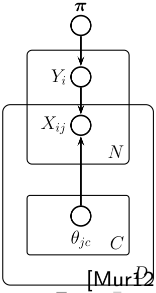
[Mur12, fig. 10.7]

Plate Notation

- variables on plates are **duplicated**
  - the number of copies is given in the lower right corner.
- an **index** is used to differentiate copies of the same variable.
- variables being in **several plates** will be duplicated for every combination, i.e., have several indices.
  - for clarity, the index should be added to the plate (but often is omitted).

Example 2: Naive Bayes classifier.





[Mur12, fig. 10.8]

Learning from Complete Data

Likelihood decomposes w.r.t. graph structure:

$$
\begin{aligned}
p(\mathcal{D} \mid \theta) :&= \prod_{n=1}^{N} p(x_n \mid \theta) \\
&= \prod_{n=1}^{N} \prod_{m=1}^{M} p(x_{n,m} \mid x_{n,\mathrm{pa}(m)}, \theta_m) \\
&= \prod_{m=1}^{M} \prod_{n=1}^{N} p(x_{n,m} \mid x_{n,\mathrm{pa}(m)}, \theta_m) \\
&= \prod_{m=1}^{M} p(\mathcal{D}_m \mid \theta_m)
\end{aligned}
$$

where θ_m are the parameters of $p(x_m \mid \mathrm{pa}(m))$

Note: In Bayesian contexts, often $p(\ldots \mid \theta)$ is used instead of $p(\ldots ; \theta)$.

Learning from Complete Data

If the prior also factorizes,

$$p(\theta) = \prod_{m=1}^{M} p(\theta_m)$$

then the posterior factorizes as well

$$p(\theta \mid \mathcal{D}) \propto p(\mathcal{D} \mid \theta)p(\theta) = \prod_{m=1}^{M} p(\mathcal{D}_m \mid \theta_m)p(\theta_m)$$

and the parameters θ_m of each CPD can be estimated independently.

Note: In Bayesian contexts, often $p(\ldots \mid \theta)$ is used instead of $p(\ldots ; \theta)$.

Learning from Complete Data / Dirichlet Prior

If

- all variables are nominal,
- variable m has L_m levels ($m = 1, \ldots, M$), and
- all CPDs are described by conditional probability tables (CPTs)

$$p(x_m \mid x_{\text{pa}(m)}) = \theta_{m,c,l}, \quad c := x_{\text{pa}(m)}, l := x_m$$
$$\text{with } \sum_{l=1}^{L} \theta_{m,c,l} = 1, \quad \forall m, c$$

a **Dirichlet distribution** for each row in the CPT

$$\theta_{m,c,\cdot} \sim \text{Dir}(\alpha_{m,c}), \quad \alpha_{m,c} \in (\mathbb{R}_0^+)^{L_m}$$

is a useful prior.

Learning from Complete Data / Dirichlet Prior

Then the posterior $p(\theta_{m,c,\cdot} \mid \mathcal{D})$ is also Dirichlet:

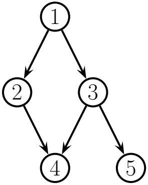
$$\theta_{m,c,\cdot} \mid \mathcal{D} \sim \text{Dir}(\alpha_{m,c} + N_{m,c})$$

$$N_{m,c,l} := \sum_{n=1}^{N} \delta(x_{n,m} = l, x_{n,\text{pa}(m)=c})$$

$$\text{with mean } \bar{\theta}_{m,c,l} = \frac{N_{m,c,l} + \alpha_{m,c,l}}{\sum_{l'=1}^{L} N_{m,c,l'} + \alpha_{m,c,l'}}$$

Learning from Complete Data / Example

graph structure:



data:

x_1	x_2	x_3	x_4	x_5
0	0	1	0	0
0	1	1	1	1
1	1	0	1	0
0	1	1	0	0
0	1	1	1	0

prior:

$$p(\theta_{m,c}) := \text{Dir}(1,1) \quad \forall m, c$$

learned parameters for CPT of x_4 ($m = 4$):

$c = x_{\text{pa}(m)}$		$N_{m,c,l}$		$\bar{\theta}_{m,c,l}$	
x_2	x_3	$N_{4,c,1}$	$N_{4,c,0}$	$\bar{\theta}_{4,c,1}$	$\bar{\theta}_{4,c,0}$
0	0	0	0	1/2	1/2
1	0	1	0	2/3	1/3
0	1	0	1	1/3	2/3
1	1	2	1	3/5	2/5

[Mur12, fig. 10.1a]

Learning with Missing and/or Hidden Variables

Learning with

- missing values or
- hidden variables

is more complicated as

- the likelihood no longer factorizes and
- neither is convex.

$\rightsquigarrow$ use iterative approximation algorithms to find a local MAP or ML minimum.

Summary

- **Bayesian Networks** define a joint probability distribution by a **factorization of conditional probability distributions (CPDs)** $p(x_n \mid \text{pa}(x_n))$
  - Conditions $\text{pa}(m)$ form a DAG.
  - For nominal variables, all CPDs can be represented as tables (CPTs).
  - Storage complexity is $O(L^{\text{max indegree}+1})$ (instead of $O(L^M)$).
- Many model classes essentially are Bayesian networks:
  - Naive Bayes classifier, Markov Models, Hidden Markov Models (HMMs)
- **Inference** in BN means to compute the (marginal joint) distribution of target variables given observed **evidence** of some predictor variables.
  - A Bayesian network can answer queries for arbitrary targets (not just a predefined one as most predictive models).
  - **Nuisance variables** (for a query) are variables neither observed nor used as targets.
  - Inference with nuisance variables can be done efficiently for DAGs with small tree width.

Summary (2/2)

- **Learning BN** has to distinguish between
  - **parameter learning**: learn just the CPDs for a given graph, vs.
  - **structure learning**: learn both, graph and CPDs.

- Parameter learning the **maximum aposteriori (MAP)** for BN with CPTs and **Dirichlet prior** can be done simply by counting the frequencies of families in the data.

- Some/most conditional independence assumptions are coded in the graph and can be read off by **d-separation**.

Further Readings

- [Mur12, chapter 10].

References

Kevin P. Murphy.
Machine learning: a probabilistic perspective.
The MIT Press, 2012.