

Machine Learning

B. Supervised Learning: Nonlinear Models

B.5. 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

Syllabus

- Fri. 26.10. (1) 0. Introduction
- A. Supervised Learning: Linear Models & Fundamentals**
- Fri. 2.11. (2) A.1 Linear Regression
- Fri. 9.11. (3) A.2 Linear Classification
- Fri. 16.11. (4) A.3 Regularization
- Fri. 23.11. (5) A.4 High-dimensional Data
- B. Supervised Learning: Nonlinear Models**
- Fri. 30.11. (6) B.1 Nearest-Neighbor Models
- Fri. 7.12. (7) B.2 Neural Networks
- Fri. 14.12. (8) B.3 Decision Trees
- Fri. 21.12. (9) B.4 Support Vector Machines
- *Christmas Break* —
- Fri. 11.1. (10) B.5 A First Look at Bayesian and Markov Networks
- C. Unsupervised Learning**
- Fri. 18.1. (11) C.1 Clustering
- Fri. 25.1. (12) C.2 Dimensionality Reduction
- Fri. 1.2. (13) C.3 Frequent Pattern Mining
- Fri. 8.2. (14) Q&A

Outline

1. Introduction
2. Examples
3. Inference
4. Learning

Outline

1. Introduction

2. Examples

3. Inference

4. Learning

Joint Distribution

x_1 : the sun shines

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

Joint Distribution

x_1 : the sun shines

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

x_2 : it rains

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

Joint Distribution

x_1 : the sun shines

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

x_2 : it rains

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

joint distribution:

$$\left. \begin{array}{l} p(x_1 = \text{false}, x_2 = \text{false}) = 0.07 \\ p(x_1 = \text{false}, x_2 = \text{true}) = 0.18 \\ p(x_1 = \text{true}, x_2 = \text{false}) = 0.6 \\ p(x_1 = \text{true}, x_2 = \text{true}) = 0.15 \end{array} \right\} \equiv \begin{array}{c|cc} & & x_2 \\ & & \text{false} \quad \text{true} \\ \hline x_1 & \text{false} & 0.07 \quad 0.18 \\ & \text{true} & 0.6 \quad 0.15 \end{array}$$

Joint Distribution

x_1 : the sun shines

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

x_2 : it rains

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

joint distribution:

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

Independence

for two variables:

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

for two variable subsets:

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

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

Independence

for two variables:

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

for two variable subsets:

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

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

Chain Rule

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

Chain Rule

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

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

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

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, \dots, z_K** :

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

Conditional Independence / Example

Example:

$$x_n \perp \{x_1, \dots, x_{n-2}\} \mid x_{n-1} \quad \forall n \text{ (Markov property)}$$
$$\rightsquigarrow p(x_1, \dots, x_N) = p(x_1)p(x_2 \mid x_1)p(x_3 \mid x_2) \cdots p(x_M \mid x_{M-1})$$

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

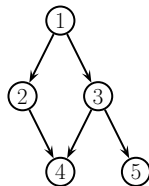
Directed Graph Terminology

- ▶ **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** $A \in \{0, 1\}^{N \times N}$

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

- ▶ **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.

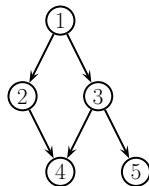


Note: $\delta(P) := 1$ if proposition P is true, $:= 0$ otherwise.

[Murphy, 2012, fig. 10.1a]

Directed Graph Terminology

- ▶ **path**: $p \in V^* := \bigcup_{M \in \mathbb{N}} V^M$: $(p_m, p_{m+1}) \in E$ for all m .
 - ▶ $p = (p_1, \dots, p_M)$, $p_m \in V$
 - ▶ **length** $|p| := M$
 - ▶ **starts at** p_1
 - ▶ **ends at** p_M
 - ▶ **paths** $G^* := \{p \in V^* \mid (p_m, p_{m+1}) \in E \quad \forall m = 1, \dots, |p| - 1\}$.
 - ▶ $v \rightsquigarrow w$: **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** $|\text{nbr}(v)|$

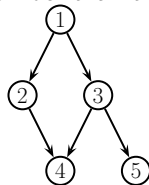


[Murphy, 2012, fig. 10.1a]

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**:
 - ▶ numbering of the nodes s.t. all nodes have lower number than their children.
 - ▶ exists for DAGs.



[Murphy, 2012, fig. 10.1a]

Bayesian Networks / Directed Graphical Models

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

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

s.t. the graph defined by

$$V := \{1, \dots, M\}$$

$$E := \{(n, m) \mid m \in V, n \in \text{ctxt}(m)\}, \quad \text{i.e., } \text{pa}(m) := \text{ctxt}(m)$$

is a DAG.

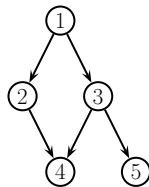
A Bayesian network defines a **factorization of the joint distribution**

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

Bayesian Networks / Example

For the DAG below,

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



[Murphy, 2012, fig. 10.1a]

Bayesian Networks / Example

For the DAG below,

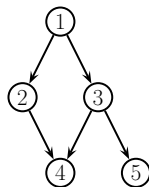
$$p(x_1, x_2, x_3, x_4, x_5) = p(x_1) p(x_2 \mid x_1) p(x_3 \mid x_1) p(x_4 \mid x_2, x_3) p(x_5 \mid x_3)$$

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		x_2		x_1		x_3		x_1	
0	...	0	0 1	0	...	0	0 1	0	...
1	...	1	...	1	...	1	...	1	...

x_2		x_3		x_5	
	0 1		0 1		0 1
x_4	0	0	...
	1	1	...

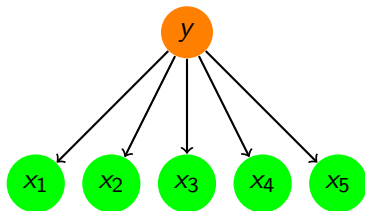


[Murphy, 2012, fig. 10.1a]

Outline

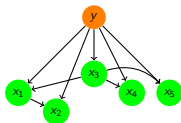
1. Introduction
2. Examples
3. Inference
4. Learning

Naive Bayes Classifier

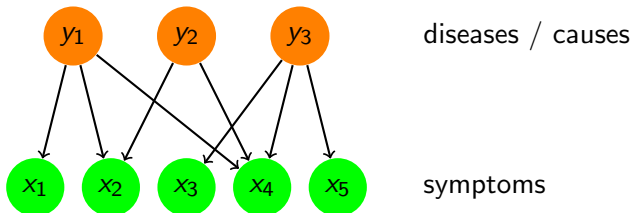


$$\begin{aligned}
 p(x_1, \dots, x_M, y) &= p(y)p(x_1 | y)p(x_2 | y) \cdots p(x_M | y) \\
 &= p(y) \prod_{m=1}^M p(x_m | y)
 \end{aligned}$$

more powerful generalization:
tree-augmented naive Bayes:



Medical Diagnosis



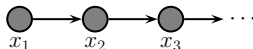
$$p(x_1, \dots, x_M, y_1, \dots, y_T) = \prod_{t=1}^T p(y_t) \prod_{m=1}^M p(x_m | y_{pa(m)})$$

- ▶ bipartite graph
- ▶ predictor variables x_1, \dots, x_M (symptoms)
- ▶ target variables y_1, \dots, y_T (diseases / causes)
 - ▶ multi-label (\leftrightarrow Naive Bayes: single-label)
 - ▶ y 's also could be hidden

Markov Models

first order:

$$\begin{aligned} p(x_1, \dots, x_M) &= p(x_1)p(x_2 | x_1)p(x_3 | x_2) \cdots p(x_M | x_{M-1}) \\ &= p(x_1) \prod_{m=1}^{M-1} p(x_{m+1} | x_m) \end{aligned}$$

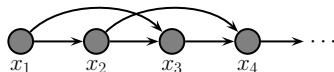


[Murphy, 2012, fig. 10.3a]

Markov Models / Second Order

second order:

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



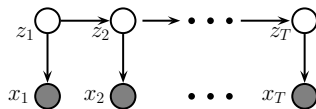
[Murphy, 2012, fig. 10.3b]

Hidden Markov Models

- ▶ observed variables x_1, \dots, x_M
- ▶ hidden variables z_1, \dots, z_M

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

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



[Murphy, 2012, fig. 10.4]

Outline

1. Introduction
2. Examples
- 3. Inference**
4. Learning

The Probabilistic Inference Problem

Given

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

compute

$$p(Y \mid X = v; \theta) := p(y_1, \dots, y_T \mid x_1 = v_1, x_2 = v_2, \dots, x_M = v_M; \theta)$$

$$= (p(y_1 = w_1, \dots, y_T = w_T \mid x_1 = v_1, x_2 = v_2, \dots, x_M = v_M; \theta))_{w_1, \dots, w_T}$$

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

Inference Without Nuisance Variables

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

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

- ▶ 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: $Z := \{z_1, \dots, z_K\} := V \setminus (X \cup Y)$.

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

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

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

Inference With Nuisance Variables

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

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

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

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: $Y = \{y\}$, $T = 1$
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 $O((M + K + 1)L^{\max \text{ indegree} + 1})$ 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

1. Introduction
2. Examples
3. Inference
- 4. Learning**

Learning Bayesian Networks

- ▶ **parameter learning**: given
 - ▶ the structure of the network (graph G),
 - ▶ a regularization penalty $\text{Reg}(\theta)$ — for the parameters θ of the CPTs, and
 - ▶ data x_1, \dots, x_N ,learn the **CPTs** 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 **CPTs** p .

Bayesian Approach

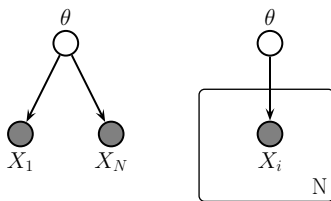
- ▶ in the Bayesian approach, parameters are also considered to be random variables, thus,
- ▶ learning is just a special type of inference (with the parameters as targets)
- ▶ information about the distribution of the parameters before seeing the data is required (**prior distribution** $p(\theta)$)
- ▶ **parameter learning**: given
 - ▶ the structure of the network (graph G) and
 - ▶ a prior distribution $p(\theta)$ of the parameters,
 - ▶ data x_1, \dots, x_N ,learn the **CPTs** 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, \dots, x_N is independently identically distributed (iid)

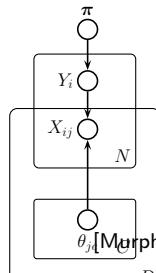
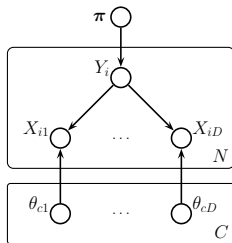


[Murphy, 2012, fig. 10.7]

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.



[Murphy, 2012, fig. 10.8]

Learning from Complete Data

Likelihood decomposes w.r.t. graph structure:

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

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

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

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 CPT can be estimated independently.

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

Learning from Complete Data / Dirichlet Prior

If

- ▶ all variables are nominal,
- ▶ variable m has L_m levels ($m = 1, \dots, M$), and

parameters θ of CPTs are

$$p(x_m \mid x_{\text{pa}(m)}) = \theta_{m,c,l}, \quad c := x_{\text{pa}(m)}, l := x_m$$

$$\text{with } \sum_{l=1}^{L_m} \theta_{m,c,l} = 1, \quad \forall m, c$$

and 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:

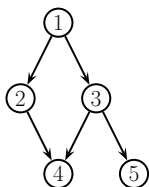
$$\theta_{m,c,\cdot} \mid \mathcal{D} \sim \text{Dir}(\alpha_{m,c} + N_{m,c})$$

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

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

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:

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

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

[Murphy, 2012, fig. 10.1a]

Learning BN from Complete Data / Algorithm

```

1 learn-bn-params( $\mathcal{D}^{\text{train}} := \{x_1, \dots, x_N\} \subset \mathcal{X}_1 \times \dots \times \mathcal{X}_M, G, \alpha$ ) :
2   for  $n := 1 : N$ :
3     for  $m := 1 : M$ :
4        $\alpha_{m, x_n, m, x_n, \text{pa}(m)} \text{ += } 1$ 
5   return  $\alpha$ 
  
```

where

- ▶ $\mathcal{X}_m := \{1, \dots, L_m\}$ discrete domains of variable X_m
(having L_m different levels)
- ▶ G is a DAG on $\{1, \dots, M\}$
- ▶ $(\alpha_{m,l,c})_{m=1:M, l=1:L_m, c \in \prod_{c \in \text{pa}(m)} L_c} \geq 0$ the Dirichlet prior of the parameters

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.

↪ use iterative approximation algorithms to find a local MAP or ML optimum.

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 $O(L^{\max \text{ indegree} + 1})$ (instead of $O(L^M)$).
- ▶ Many model classes essentially are Bayesian networks:
 - ▶ **Naive Bayes classifier, Markov Models, Hidden Markov Models**
- ▶ **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.

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 a posteriori (MAP)** for BN with CPTs and **Dirichlet prior** can be done simply by counting the frequencies of families in the data.

Further Readings

- ▶ [Murphy, 2012, chapter 10].

References

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