

CS 2750 Machine Learning Lecture 12

Learning Bayesian belief networks

Milos Hauskrecht
milos@cs.pitt.edu
5329 Sennott Square

CS 2750 Machine Learning

Learning probability distribution

Basic settings:

- A set of random variables $\mathbf{X} = \{X_1, X_2, \dots, X_n\}$
- **A model of the distribution** over variables in \mathbf{X} with parameters Θ
- **Data** $D = \{D_1, D_2, \dots, D_N\}$

Objective: find parameters $\hat{\Theta}$ that describe the data the best

Learning Bayesian belief networks:

- parameterizations as defined by the structure of network

CS 2750 Machine Learning

Learning of BBN

Learning.

- **Learning of parameters of conditional probabilities**
- **Learning of the network structure**

Variables:

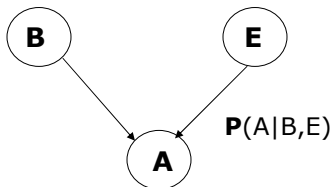
- **Observable** – values present in every data sample
- **Hidden** – they values are never observed in data
- **Missing values** – values sometimes present, sometimes not

Next: All variables are observable

1. Learning of parameters of BBN
2. Learning of the model (BBN structure)

Learning of parameters of BBN

- **Idea:** decompose the estimation problem for the full joint over a large number of variables to a set of smaller estimation problems corresponding to parent-variable conditionals.
- **Example:** Assume A,E,B are binary with *True*, *False* values



4 estimation problems

$$\left\{ \begin{array}{l} P(A|B=T,E=T) \\ P(A|B=T,E=F) \\ P(A|B=F,E=T) \\ P(A|B=F,E=F) \end{array} \right.$$

- **Assumption that enables the decomposition:** parameters of conditional distributions are independent

Estimates of parameters of BBN

- Two assumptions that permit the decomposition:
 - **Sample independence**

$$P(D | \Theta, \xi) = \prod_{u=1}^N P(D_u | \Theta, \xi)$$

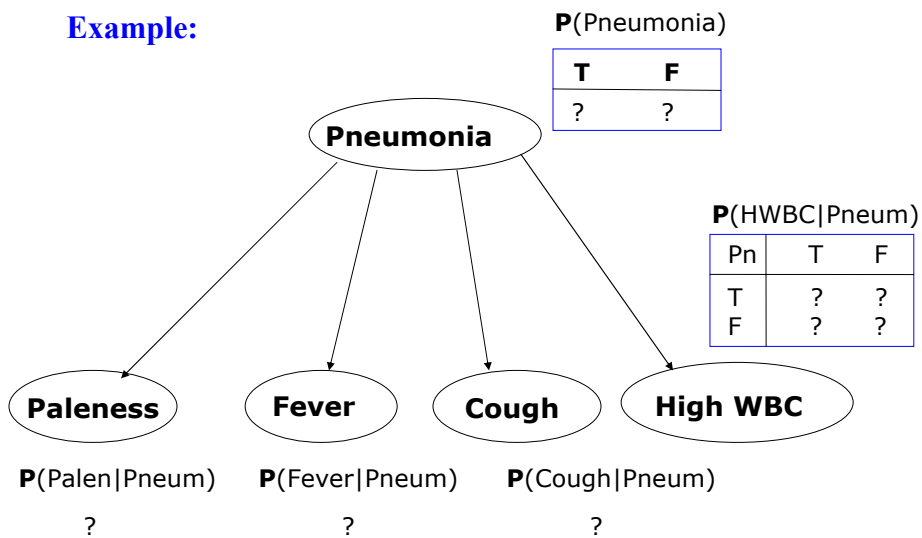
- **Parameter independence**

$$p(\Theta | D, \xi) = \prod_{i=1}^n \prod_{j=1}^{q_i} p(\theta_{ij} | D, \xi)$$

Parameters of each conditional (one for every assignment of values to parent variables) can be learned independently

Learning of BBN parameters. Example.

Example:

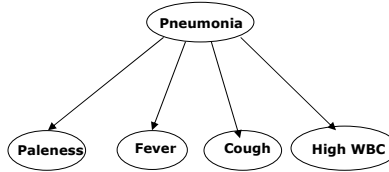


Learning of BBN parameters. Example.

Data D (different patient cases):

Pal Fev Cou HWB Pneu

T	T	T	T	F
T	F	F	F	F
F	F	T	T	T
F	F	T	F	T
F	T	T	T	T
T	F	T	F	F
F	F	F	F	F
T	T	F	F	F
T	T	T	T	T
F	T	F	T	T
T	F	F	T	F
F	T	F	F	F



CS 2750 Machine Learning

Estimates of parameters of BBN

- Much like multiple **coin toss or roll of a dice** problems.
- A “smaller” learning problem corresponds to the learning of exactly one conditional distribution

- **Example:**

$$P(\text{Fever} \mid \text{Pneumonia} = T)$$

- **Problem:** How to pick the data to learn?

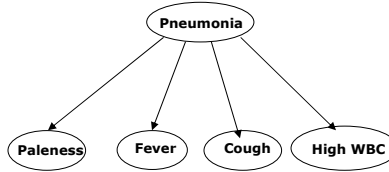
CS 2750 Machine Learning

Learning of BBN parameters. Example.

Learn: $P(\text{Fever} \mid \text{Pneumonia} = T)$

Step 1: Select data points with Pneumonia=T

Pal	Fev	Cou	HWB	Pneu
T	T	T	T	F
T	F	F	F	F
F	F	T	T	T
F	F	T	F	T
F	T	T	T	T
T	F	T	F	F
F	F	F	F	F
T	T	F	F	F
T	T	T	T	T
F	T	F	T	T
T	F	F	T	F
F	T	F	F	F

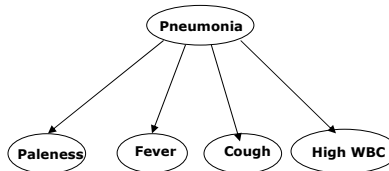


Learning of BBN parameters. Example.

Learn: $P(\text{Fever} \mid \text{Pneumonia} = T)$

Step 1: Ignore the rest

Pal	Fev	Cou	HWB	Pneu
F	F	T	T	T
F	F	T	F	T
F	T	T	T	T
T	T	T	T	T
F	T	F	T	T



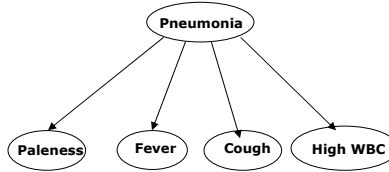
Learning of BBN parameters. Example.

Learn: $P(\text{Fever} \mid \text{Pneumonia} = T)$

Step 2: Select values of the random variable defining the distribution of Fever

Pal Fev Cou HWB Pneu

F	F	T	T	T
F	F	T	F	T
F	T	T	T	T
T	T	T	T	T
F	T	F	T	T



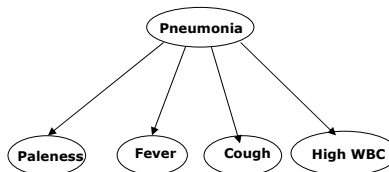
Learning of BBN parameters. Example.

Learn: $P(\text{Fever} \mid \text{Pneumonia} = T)$

Step 2: Ignore the rest

Fev

F
F
T
T
T



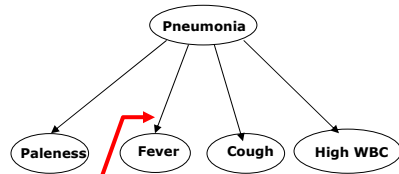
Learning of BBN parameters. Example.

Learn: $P(\text{Fever} \mid \text{Pneumonia} = T)$

Step 3a: Learning the ML estimate

Fev

F
F
T
T
T



$P(\text{Fever} \mid \text{Pneumonia} = T)$

T	F
0.6	0.4

CS 2750 Machine Learning

Learning of BBN parameters. Bayesian learning.

Learn: $P(\text{Fever} \mid \text{Pneumonia} = T)$

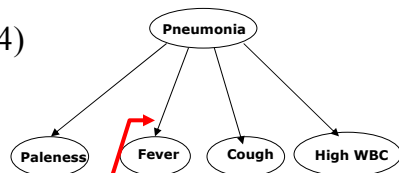
Step 3b: Learning the Bayesian estimate

Assume the prior

$$\theta_{\text{Fever}|\text{Pneumonia}=T} \sim \text{Beta}(3,4)$$

Fev

F
F
T
T
T



Posterior:

$$\theta_{\text{Fever}|\text{Pneumonia}=T} \sim \text{Beta}(6,6)$$

CS 2750 Machine Learning

Naïve Bayes model

A **special (simple) Bayesian belief network**

- **used as a generative classifier model**

- Class variable Y
- Attributes are independent given Y

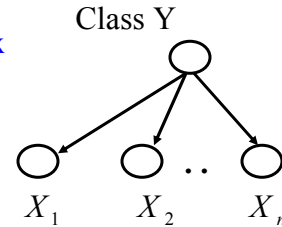
$$p(\mathbf{x} | Y = i, \Theta) = \prod_{j=1}^n p(x_j | Y = i, \Theta_{ij})$$

Learning: ML, Bayesian estimates of parameters

Classification: given x we need to determine the class

- Choose the class with the maximum posterior

$$p(Y = i | \mathbf{x}, \Theta) = \frac{p(Y = i | \Theta) p(\mathbf{x} | Y = i, \Theta)}{\sum_{j=1}^k p(Y = j | \Theta) p(\mathbf{x} | Y = j, \Theta)}$$



CS 2750 Machine Learning

Naïve Bayes with Gaussians distributions

Generative classification model $p(\mathbf{X}, Y)$

1. Priors on classes

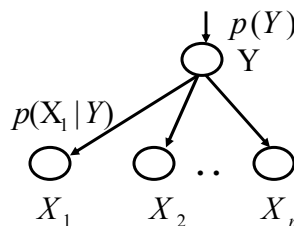
$$p(Y = 1), p(Y = 2), p(Y = 3), \dots$$

Before: Joint class conditional densities (for \mathbf{x})

$$p(\mathbf{x} | \mu_j, \Sigma_j) = \frac{1}{(2\pi)^{d/2} |\Sigma_j|^{1/2}} \exp \left[-\frac{1}{2} (\mathbf{x} - \mu_j)^T \Sigma_j^{-1} (\mathbf{x} - \mu_j) \right]$$

Now: Naïve Bayes - independent class conditional densities

$$p(x_i | \mu_{ji}, \sigma_{ji}) = \frac{1}{\sqrt{(2\pi)\sigma_{ji}}} \exp \left[-\frac{1}{2\sigma_{ji}^2} (x_i - \mu_{ji})^2 \right]$$



CS 2750 Machine Learning

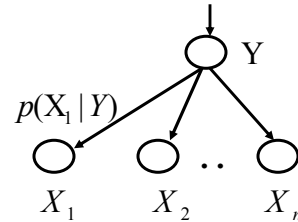
Naïve Bayes with Gaussians distributions

How to learn the generative model $p(\mathbf{X}, Y)$

1. Priors on classes

$$p(Y = 1), p(Y = 2), p(Y = 3), \dots$$

?



2. Class conditional densities

$$p(x_i | \mu_{ji}, \sigma_{ji}) = \frac{1}{\sqrt{(2\pi)\sigma_{ji}^2}} \exp\left[-\frac{1}{2\sigma_{ji}^2} (x_i - \mu_{ji})^2\right]$$

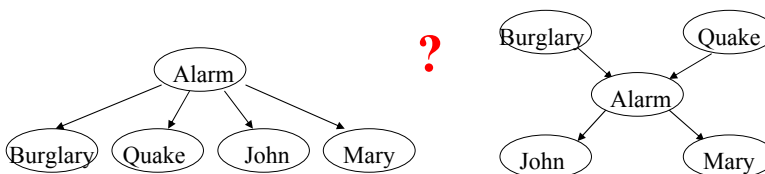
CS 2750 Machine Learning

Model selection

- **BBN has two components:**
 - **Structure of the network** (models conditional independences)
 - **A set of parameters** (conditional child-parent distributions)

We already know how to learn the parameters for the fixed structure

But how to learn the structure of the BBN?



CS 2750 Machine Learning

Learning the structure

Criteria we can choose to score the structure S

- **Marginal likelihood**

maximize $P(D | S, \xi)$

ξ - represents the prior knowledge

- **Maximum posterior probability**

maximize $P(S | D, \xi)$

$$P(S | D, \xi) = \frac{P(D | S, \xi)P(S | \xi)}{P(D | \xi)}$$

How to compute marginal likelihood $P(D | S, \xi)$?

Learning of BBNs

- **Notation:**

- i ranges over all possible variables $i=1, \dots, n$

- $j=1, \dots, q$ ranges over all possible parent combinations

- $k=1, \dots, r$ ranges over all possible variable values

- Θ - parameters of the BBN

- Θ_{ij} is a vector of Θ_{ijk} representing parameters of the conditional probability distribution; such that $\sum_{k=1}^r \Theta_{ijk} = 1$

- N_{ijk} - a number of instances in the dataset where parents of variable X_i take on values j and X_i has value k

$$N_{ij} = \sum_{k=1}^r N_{ijk}$$

- α_{ijk} - prior counts (parameters of Beta and Dirichlet priors)

$$\alpha_{ij} = \sum_{k=1}^r \alpha_{ijk}$$

Marginal likelihood

- Integrate over all possible parameter settings

$$P(D | S, \xi) = \int_{\Theta} P(D | S, \Theta, \xi) p(\Theta | S, \xi) d\Theta$$

- Using the assumption of parameter and sample independence

$$P(D | S, \xi) = \prod_{i=1}^n \prod_{j=1}^{q_i} \frac{\Gamma(\alpha_{ij})}{\Gamma(\alpha_{ij} + N_{ij})} \prod_{k=1}^{r_i} \frac{\Gamma(\alpha_{ijk} + N_{ijk})}{\Gamma(\alpha_{ijk})}$$

- We can use **log-likelihood score** instead

$$\log P(D | S, \xi) = \sum_{i=1}^n \left\{ \sum_{j=1}^{q_i} \log \frac{\Gamma(\alpha_{ij})}{\Gamma(\alpha_{ij} + N_{ij})} + \sum_{k=1}^{r_i} \log \frac{\Gamma(\alpha_{ijk} + N_{ijk})}{\Gamma(\alpha_{ijk})} \right\}$$

Score is decomposable along variables !!!

CS 2750 Machine Learning

Trick to compute the marginal likelihood

- Integrate over all possible parameter settings

$$P(D | S, \xi) = \int_{\Theta} P(D | S, \Theta, \xi) p(\Theta | S, \xi) d\Theta$$

- Posterior of parameters, given data and the structure

$$p(\Theta | D, S, \xi) = \frac{P(D | \Theta, S, \xi) p(\Theta | S, \xi)}{P(D | S, \xi)}$$

Trick

$$P(D | S, \xi) = \frac{P(D | \Theta, S, \xi) p(\Theta | S, \xi)}{p(\Theta | D, S, \xi)}$$

- Gives the solution

$$P(D | S, \xi) = \prod_{i=1}^n \prod_{j=1}^{q_i} \frac{\Gamma(\alpha_{ij})}{\Gamma(\alpha_{ij} + N_{ij})} \prod_{k=1}^{r_i} \frac{\Gamma(\alpha_{ijk} + N_{ijk})}{\Gamma(\alpha_{ijk})}$$

CS 2750 Machine Learning

Learning the structure

- **Likelihood of data for the BBN** (structure and parameters)

$$P(D | S, \Theta, \xi)$$

measures the goodness of fit of the BBN to data

- **Marginal likelihood** (for the structure only)

$$P(D | S, \xi)$$

- **Does not measure only a goodness of fit. It is:**
 - different for structures of different complexity
 - Incorporates preferences towards simpler structures, **implements Occam's razor !!!!**

Occam's Razor

- Why there is a preference towards simpler structures ?

Rewrite marginal likelihood as

$$P(D | S, \xi) = \frac{\int_{\Theta} P(D | S, \Theta, \xi) p(\Theta | S, \xi) d\Theta}{\int_{\Theta} p(\Theta | S, \xi) d\Theta}$$

We know that $\int_{\Theta} p(\Theta | S, \xi) d\Theta = 1$

Interpretation: in more complex structures there are more ways parameters can be set badly

- **The numerator:** count of good assignments
- **The denominator:** count of all assignments

Approximations of probabilistic scores

Approximations of the marginal likelihood and posterior scores

- **Information based measures**

- Akaike criterion
- Bayesian information criterion (BIC)
- Minimum description length (MDL)

- Reflect the tradeoff between the fit to data and preference towards simpler structures

Example: **Akaike criterion.**

Maximize: $score(S) = \log P(D | S, \Theta_{ML}, \xi) - \text{compl}(S)$

Bayesian information criterion (BIC)

Maximize: $score(S) = \log P(D | S, \Theta_{ML}, \xi) - \frac{1}{2} \text{compl}(S) \log N$

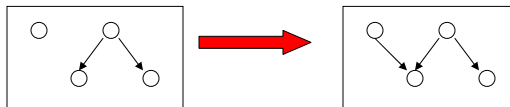
Optimizing the structure

Finding the best structure is a **combinatorial optimization** problem

- A good feature: the score **is decomposable along variables:**

$$\log P(D | S, \xi) = \sum_{i=1}^n \left\{ \sum_{j=1}^{q_i} \log \frac{\Gamma(\alpha_{ij})}{\Gamma(\alpha_{ij} + N_{ij})} + \sum_{k=1}^{r_i} \log \frac{\Gamma(\alpha_{ijk} + N_{ijk})}{\Gamma(\alpha_{ijk})} \right\}$$

Algorithm idea: Search the space of structures using local changes (additions and deletions of a link)



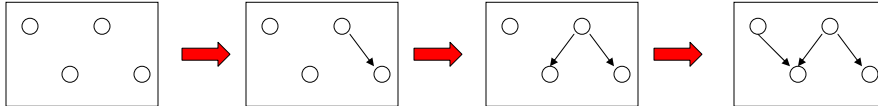
Advantage:

- we do not have to compute the whole score from scratch
- Recompute the partial score for the affected variable

Optimizing the structure. Algorithms

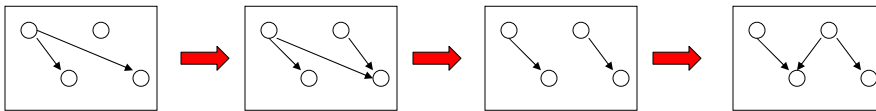
- **Greedy search**

- Start from structure with no links
- Add a link that yields the best score improvement



- **Metropolis algorithm (with simulated annealing)**

- Local additions and deletions
- Avoids being trapped in “local” optimal



CS 2750 Machine Learning

Next lecture

Learning with hidden and missing data

The variables in the BBN may not be perfectly observable in the dataset

Two cases:

- The value of a random variable (node) is never observable
 - Can be used to describe a variety of hidden modes, e.g., related to failure modes of a device
- The value of a variable that may not be available in every data entry

We would still like to learn the parameters of the BBN ??

CS 2750 Machine Learning

Appendix: marginal likelihood

- From the iid assumption:

$$P(\mathbf{D} | S, \Theta) = \prod_{h=1}^N \prod_{i=1}^n P(x_i^h | \text{parents}_i^h, \Theta)$$

- Let r_i = number of values that attribute x_i can take
 q_i = number of possible parent combinations
 N_{ijk} = number of cases in \mathbf{D} where x_i has value k and parents with values j .

$$\begin{aligned} &= \prod_i^n \prod_j^{q_i} \prod_k^{r_i} P(x_i = k | \text{parents}_i = j, \Theta)^{N_{ijk}} \\ &= \prod_i^n \prod_j^{q_i} \prod_k^{r_i} \theta_{ijk}^{N_{ijk}} \end{aligned}$$

Appendix: the marginal likelihood

- From parameter independence

$$p(\Theta | S, \xi) = \prod_{i=1}^n \prod_{j=1}^{q_i} p(\Theta_{ij} | S, \xi)$$

- Priors for $p(\Theta_{ij} | S, \xi)$

- $\Theta_{ij} = (\Theta_{ij1}, \dots, \Theta_{ijr_i})$ is a vector of parameters;
- we use a Dirichlet distribution with parameters α to represent it

$$P(\Theta_{ij} | S, \xi) = P(\Theta_{ij1}, \dots, \Theta_{ijr_i} | S, \xi) = \text{Dirichlet}(\Theta_{ij1}, \dots, \Theta_{ijr_i} | \alpha)$$

$$\begin{aligned} &= \frac{\Gamma(\sum_{k=1}^{r_i} \alpha_{ijk})}{\prod_{k=1}^{r_i} \Gamma(\alpha_{ijk})} \prod_{k=1}^{r_i} \Theta_{ijk}^{\alpha_{ijk}-1} \end{aligned}$$

Appendix: marginal likelihood

- **Combine things together:**

$$\begin{aligned}
 P(D | S_i) &= \int_{\Theta} P(D | S_i, \Theta) P(\Theta | S_i) d\Theta \\
 &= \int \prod_i^n \prod_j^{q_i} \prod_k^{r_i} \Theta^{N_{ijk}} \cdot \frac{\Gamma(\sum_{k=1}^{r_i} \alpha_{ijk})}{\prod_{k=1}^{r_i} \Gamma(\alpha_{ijk})} \prod_{k=1}^{r_i} \Theta^{\alpha_{ijk} - 1} d\Theta \\
 &= \prod_i^n \prod_j^{q_i} \frac{\Gamma(\sum_{k=1}^{r_i} \alpha_{ijk})}{\prod_{k=1}^{r_i} \Gamma(\alpha_{ijk})} \int \prod_{k=1}^{r_i} \Theta^{N_{ijk} + \alpha_{ijk} - 1} d\Theta \\
 &= \prod_i^n \prod_j^{q_i} \frac{\Gamma(\alpha_{ij})}{\prod_{k=1}^{r_i} \Gamma(\alpha_{ijk})} \cdot \frac{\prod_{k=1}^{r_i} \Gamma(\alpha_{ijk} + N_{ijk})}{\Gamma(\alpha_{ij} + N_{ij})}
 \end{aligned}$$

CS 2750 Machine Learning

Trick to compute the marginal likelihood

- Integrate over all possible parameter settings

$$P(D | S, \xi) = \int_{\Theta} P(D | S, \Theta, \xi) p(\Theta | S, \xi) d\Theta$$

- Posterior of parameters, given data and the structure

$$p(\Theta | D, S, \xi) = \frac{P(D | \Theta, S, \xi) p(\Theta | S, \xi)}{P(D | S, \xi)}$$

Trick

$$P(D | S, \xi) = \frac{P(D | \Theta, S, \xi) p(\Theta | S, \xi)}{p(\Theta | D, S, \xi)}$$

- Gives the solution

$$P(D | S, \xi) = \prod_{i=1}^n \prod_{j=1}^{q_i} \frac{\Gamma(\alpha_{ij})}{\Gamma(\alpha_{ij} + N_{ij})} \prod_{k=1}^{r_i} \frac{\Gamma(\alpha_{ijk} + N_{ijk})}{\Gamma(\alpha_{ijk})}$$

CS 2750 Machine Learning