## CS 2750 Machine Learning
## Lecture 19

# Learning Bayesian belief networks

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



## Learning probability distribution

**Basic settings:**

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

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

**Learning Bayesian belief networks:**

- parameterizations as defined by the structure of network

# Learning of BBN

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

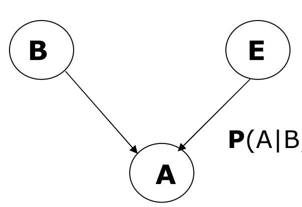
**Variables:**
- **Observable** – values present in every data sample
- **Hidden** – they values are never observed in data
- **Missing values** – values sometimes present, sometimes not

**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




**P**(A|B,E)

**4 estimation problems**
- **P**(A|B=T,E=T)
- **P**(A|B=T,E=F)
- **P**(A|B=F,E=T)
- **P**(A|B=F,E=F)

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

# Estimates of parameters of BBN

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

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

  - **Parameter independence**

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

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

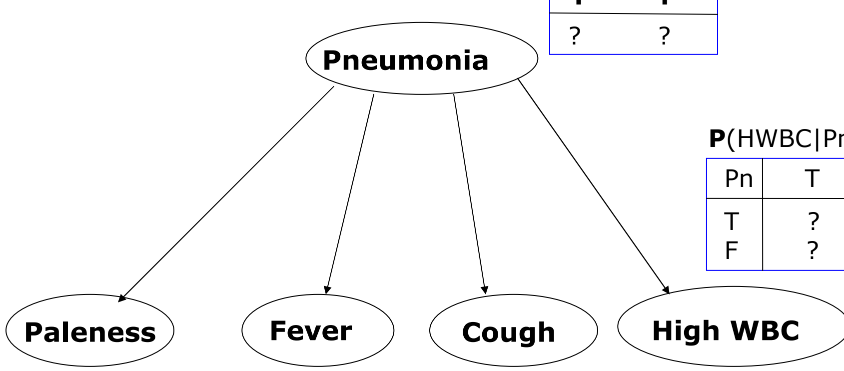


# Learning of BBN parameters. Example.

**Example:**

**P**(Pneumonia)

| **T** | **F** |
|---|---|
| ? | ? |




**P**(HWBC|Pneum)

| Pn | T | F |
|---|---|---|
| T | ? | ? |
| F | ? | ? |

**P**(Palen|Pneum) ?

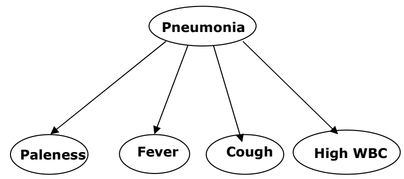
**P**(Fever|Pneum) ?

**P**(Cough|Pneum) ?

# 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 |






# 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:**

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

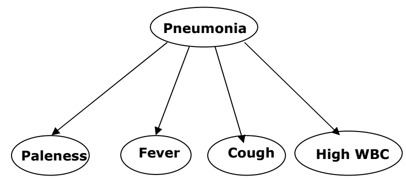
- **Problem:** How to pick the data to learn?

# Learning of BBN parameters. Example.

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

**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 |



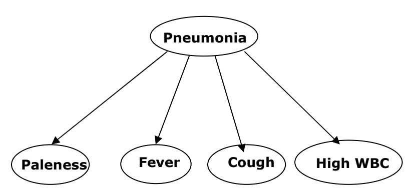



# Learning of BBN parameters. Example.

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

**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 |

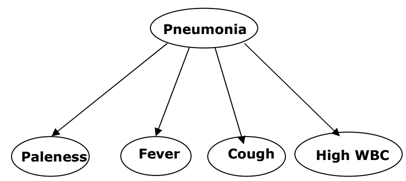


# Learning of BBN parameters. Example.

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

**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 |



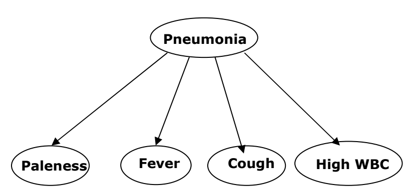



# Learning of BBN parameters. Example.

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

**Step 2:** Ignore the rest

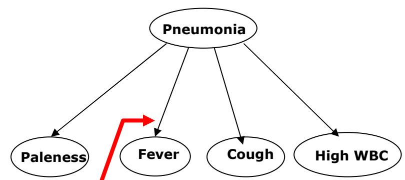
| Fev |
|---|
| F |
| F |
| T |
| T |
| T |



# Learning of BBN parameters. Example.

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

**Step 3a: Learning the ML estimate**




**Fev**
**F**
**F**
**T**
**T**
**T**

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

| T | F |
| --- | --- |
| 0.6 | 0.4 |



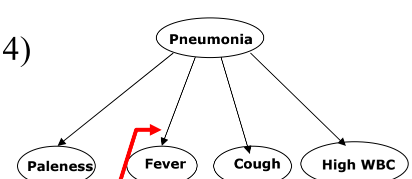
# Learning of BBN parameters. Bayesian learning.

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

**Step 3b: Learning the Bayesian estimate**

**Assume the prior**

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




**Fev**
**F**
**F**
**T**
**T**
**T**

**Posterior:**

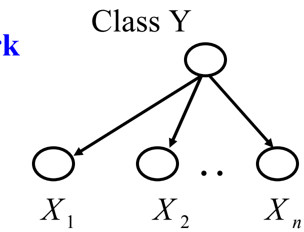
$$\theta_{Fever|Pneumonia=T} \sim Beta(6,6)$$

## Naïve Bayes model

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

- **used as a generative classifier model**
  - Class variable Y
  - Attributes are independent given Y

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




**Learning:** ML, Bayesian estimates of parameters

**Classification:** given $x$ we need to determine the class

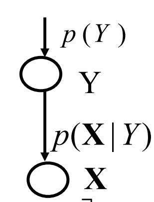
- Choose the class with the maximum posterior

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



## Naïve Bayes with Gaussians distributions

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




**1. Priors on classes**

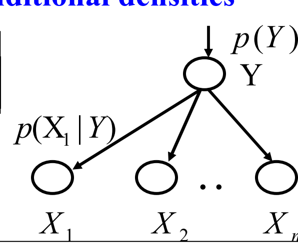
$p(Y = 1),\ p(Y = 2),\ p(Y = 3), \ldots$

**Before: Joint class conditional densities (for x)**

$$p(\mathbf{x} \mid \boldsymbol{\mu}_j, \mathbf{\Sigma}_j) = \frac{1}{(2\pi)^{d/2} \left|\mathbf{\Sigma}_j\right|^{1/2}} \exp\left[ -\frac{1}{2} (\mathbf{x} - \boldsymbol{\mu}_j)^T \mathbf{\Sigma}_j^{-1} (\mathbf{x} - \boldsymbol{\mu}_j) \right]$$

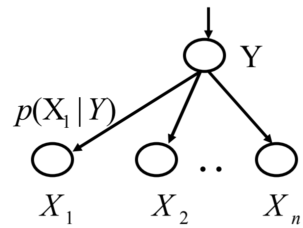
**Now: Naïve Bayes - independent class conditional densities**

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



# Naïve Bayes with Gaussians distributions

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




**1. Priors on classes**

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

?

**2. Class conditional densities**

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

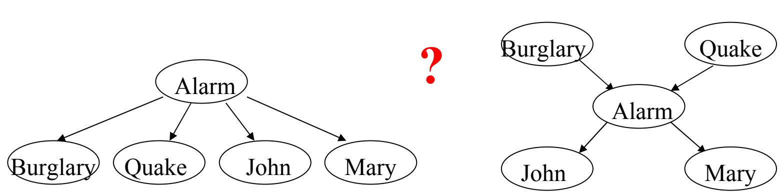


# Model selection

- **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?



# Learning the structure

Criteria we can choose to score the structure S

- **Marginal likelihood**

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

  $\xi$ - represents the prior knowledge

- **Maximum posterior probability**

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

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

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



# Learning of BBNs

- **Notation**:
  - $i$ ranges over all possible variables $i=1,..,n$
  - $j=1,..,q$ ranges over all possible parent combinations
  - $k=1,..,r$ ranges over all possible variable values
  - $\mathbf{\Theta}$ - parameters of the BBN

  $\Theta_{ij}$ is a vector of $\Theta_{ijk}$ 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 $k$

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

  $\alpha_{ijk}$ - prior counts (parameters of Beta and Dirichlet priors)

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

## Marginal likelihood

- Integrate over all possible parameter settings

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

- Using the assumption of parameter and sample independence

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

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

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

**Score is decomposable along variables !!!**



## Trick to compute the marginal likelihood

- Integrate over all possible parameter settings

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

- Posterior of parameters, given data and the structure

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

**Trick**

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

- Gives the solution

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

# Learning the structure

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

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

measures the goodness of fit of the BBN to data

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

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

- **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

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

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

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

**Bayesian information criterion (BIC)**

**Maximize:**

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



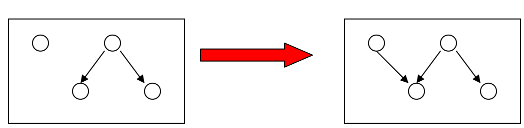
## Optimizing the structure

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

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

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

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

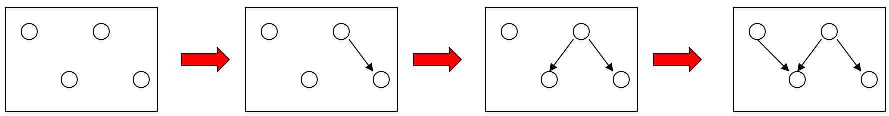


**Advantage:**

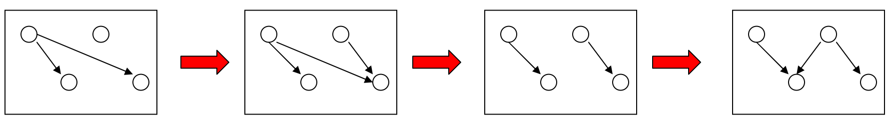
- we do not have to compute the whole score from scratch
- Recompute the partial score for the affected variable

# 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





# 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 ??

# Appendix: marginal likelihood

- **From the iid assumption:**

$$P(\boldsymbol{D} \mid S, \boldsymbol{\Theta}) = \prod_{h=1}^{N} \prod_{i=1}^{n} P(x_i^h \mid parents_i^h, \boldsymbol{\Theta})$$

- 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 D where $x_i$ has value k and parents with values j.

$$= \prod_{i}^{n} \prod_{j}^{q_i} \prod_{k}^{r_i} P(x_i = k \mid parents_i = j, \boldsymbol{\Theta})^{N_{ijk}}$$

$$= \prod_{i}^{n} \prod_{j}^{q_i} \prod_{k}^{r_i} \theta_{ijk}^{N_{ijk}}$$



# Appendix: the marginal likelihood

- **From parameter independence**

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

- **Priors for** $p(\Theta_{ij} \mid S, \xi)$
  - $\Theta_{ij} = (\Theta_{ij1}, \ldots, \Theta_{ijr_i})$ is a vector of parameters;
  - we use a Dirichlet distribution with parameters $\boldsymbol{\alpha}$ to represent it

$$P(\Theta_{ij} \mid S, \xi) = P(\Theta_{ij1}, \ldots, \Theta_{ijr_i} \mid S, \xi) = Dirichlet(\Theta_{ij1}, \ldots, \Theta_{ijr_i} \mid \alpha)$$

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

# Appendix: marginal likelihood

- **Combine things together:**

$$P(D \mid S_i) = \int_{\Theta} P(D \mid S_i, \Theta) P(\Theta \mid S_i) d\Theta$$

$$= \int \prod_{i}^{n} \prod_{j}^{q_i} \prod_{k}^{r_i} \Theta_{ijk}^{N_{ijk}} \cdot \frac{\Gamma(\sum_{k=1}^{r_i} \alpha_{ijk})}{\prod_{k=1}^{r_i} \Gamma(\alpha_{ijk})} \prod_{k=1}^{r_i} \Theta_{ijk}^{\alpha_{ijk}-1} d\Theta$$

$$= \prod_{i}^{n} \prod_{j}^{q_i} \frac{\Gamma(\sum_{k=1}^{r_i} \alpha_{ijk})}{\prod_{k=1}^{r_i} \Gamma(\alpha_{ijk})} \int \prod_{k=1}^{r_i} \Theta_{ijk}^{N_{ijk}+\alpha_{ijk}-1} d\Theta$$

$$= \prod_{i}^{n} \prod_{j}^{q_i} \frac{\Gamma(\alpha_{ij})}{\prod_{k=1}^{r_i} \Gamma(\alpha_{ijk})} \cdot \frac{\prod_{k=1}^{r_i} \Gamma(a_{ijk} + N_{ijk})}{\Gamma(\alpha_{ij} + N_{ij})}$$