

---

Università degli Studi di Milano  
Master Degree in Computer Science

# Information Management course

Teacher: Alberto Ceselli

Lecture 19: 10/12/2015

---

# **Data Mining: Concepts and Techniques**

**(3<sup>rd</sup> ed.)**


## **— Chapter 8, 9 —**

Jiawei Han, Micheline Kamber, and Jian Pei  
University of Illinois at Urbana-Champaign &  
Simon Fraser University

©2011 Han, Kamber & Pei. All rights reserved.

# Classification methods

---

- Classification: Basic Concepts
- Decision Tree Induction
- Bayes Classification Methods 
- Support Vector Machines
- Model Evaluation and Selection
- *Rule-Based Classification*
- *Techniques to Improve Classification Accuracy: Ensemble Methods*

# Bayesian Classification: Why?

- A statistical classifier: performs *probabilistic prediction*, i.e., predicts class membership probabilities
- Foundation: Based on Bayes' Theorem.
- Performance: A simple Bayesian classifier, *naïve Bayesian classifier*, has comparable performance with decision tree and selected neural network classifiers
- Incremental: Each training example can incrementally increase/decrease the probability that a hypothesis is correct — prior knowledge can be combined with observed data
- Standard: Even when Bayesian methods are computationally intractable, they can provide a standard of optimal decision making against which other methods can be measured

# Bayesian Classification Rationale: conditional probability

Class:

C1:buys\_computer = 'yes'

C2:buys\_computer = 'no'

$P(C1)?$

$P(C1|\text{student} = \text{yes})?$

income	student	credit	buys
high	no	fair	no
high	no	excellent	no
high	no	fair	yes
medium	no	fair	yes
low	yes	fair	yes
low	yes	excellent	no
low	yes	excellent	yes
medium	no	fair	no
low	yes	fair	yes
medium	yes	fair	yes
medium	yes	excellent	yes
medium	no	excellent	yes
high	yes	fair	yes
medium	no	excellent	no

# Bayesian Classification Rationale

- Let  $P(C_i|X)$  be the conditional probability of observing class  $C_i$  provided the set of attributes values of my element is  $X$
- Final aim: obtaining (an estimation of)  $P(C_i|X)$  for each  $i$  and for each  $X$  (classification model is the set of these values)
- $P(C_i|X) = P(C_i \cap X) / P(X)$
- How to compute  $P(X)$ ?
  - We would need a sufficient number of elements in the training set whose attribute values are  $X$
  - ... and therefore some elements for each possible combination of the attribute values (unrealistic)
- How to compute  $P(C_i \cap X)$ ? Same problems

# Bayesian Theorem: Basics

---

- Let  $\mathbf{X}$  be an **evidence** (data sample): unkn. class label
- Let  $H$  be a **hypothesis** on the class  $\mathbf{X}$  belongs (say “potential” class)
- Classification is to find  $P(H|\mathbf{X})$   
**a posteriori probability**: the probability that the hypothesis holds given the observed data sample  $\mathbf{X}$
- We can estimate:  
 $P(H)$  (**a priori probability**), an initial “blind” probability
  - E.g.,  $\mathbf{X}$  buys computer, regardless of age, income $P(\mathbf{X})$ : probability that a certain data sample is observed  
 $P(\mathbf{X}|H)$  (**likelihood**), the probability of observing the sample  $\mathbf{X}$ , given that the hypothesis  $H$  holds

# Bayesian classification: defs

age	income	student	credit_rating	PC
<=30	high	no	fair	no
<=30	high	no	excellent	no
31...40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31...40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31...40	medium	no	excellent	yes
31...40	high	yes	fair	yes
>40	medium	no	excellent	no

- Evidence **X** =  
(age = 31..40;  
income = medium;  
student = no;  
rating = excellent)
- Hypotesis H =  
(PC = yes)
- A priori Probability  
 $P(H) = 9/14$
- Likelihood  
 $P(\mathbf{X}|H) = 1/9$
- A posteriori Probability  
 $P(H|\mathbf{X}) = ???$



# Bayesian Theorem

---

- Given training data  $\mathbf{X}$ , *posteriori probability of a hypothesis*  $H$ ,  $P(H|\mathbf{X})$ , follows the Bayes theorem

$$P(H|X) = \frac{P(X|H)P(H)}{P(X)} = P(X|H) \times P(H) / P(X)$$

- Informally, this can be written as  
posteriori = likelihood x priori/evidence
- Predicts that  $\mathbf{X}$  belongs to  $C_i$  iff the probability  $P(C_i|\mathbf{X})$  is the highest among all the  $P(C_k|X)$  for all the  $k$  classes
- Practical difficulty: require initial knowledge of many probabilities, significant computational cost

# Bayesian Classification

---

- Let  $D$  be a training set of tuples and their associated class labels, and each tuple is represented by an  $n$ -D attribute vector  
 $\mathbf{X} = (x_1, x_2, \dots, x_n)$
- Suppose there are  $m$  classes  $C_1, C_2, \dots, C_m$ .
- Classification is to derive the maximum posteriori, i.e., the maximal  $P(C_i|\mathbf{X})$
- This can be derived from Bayes' theorem

$$P(C_i|X) = \frac{P(X|C_i)P(C_i)}{P(X)}$$

- Since  $P(X)$  is constant for all classes, only max

$$P(C_i|X) = P(X|C_i)P(C_i)$$

needs to be found (Maximum A Posteriori method)

# The “Optimal” Bayesian Classifier

---

- From a theoretical point of view, the Bayesian MAP classifier is optimal: no classifier can exist achieving a smaller error rate
- In order to compute

$$P(C_i|X) = P(X|C_i)P(C_i)$$

we need

$$P(C_i)$$

→ “easy”: just scan the DB once  
and

$$P(X|C_i)$$

→ if we have k classes and m attributes, each taking n possible values:  $k \cdot n^m$  probability values!

# Derivation of Naïve Bayes Classifier

- A simplified assumption: attributes are conditionally independent (i.e., no dependence relation between attributes) and identically distributed (iid):

$$P(X|C_i) = \prod_{k=1}^n P(x_k|C_i) = P(x_1|C_i) \times P(x_2|C_i) \times \dots \times P(x_n|C_i)$$

- This greatly reduces the computation cost: Only counts the class distribution ( $k \times n \times m$  probabilities)
- If  $A_k$  is categorical,  $P(x_k|C_i)$  is the # of tuples in  $C_i$  having value  $x_k$  for  $A_k$  divided by  $|C_{i,D}|$  (# of tuples of  $C_i$  in  $D$ )
- If  $A_k$  is continuous-valued,  $P(x_k|C_i)$  is usually computed based on Gaussian distribution with a mean  $\mu$  and standard deviation  $\sigma$

$$g(x, \mu, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

$$P(X|C_i) = g(x_k, \mu_{C_i}, \sigma_{C_i})$$

# Training a Naïve Bayesian Classifier (example)

## Training:

age	income	student	credit_rating	PC
<=30	high	no	fair	no
<=30	high	no	excellent	no
31...40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31...40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31...40	medium	no	excellent	yes
31...40	high	yes	fair	yes
>40	medium	no	excellent	no

- $P(PC = \text{yes}) = 9/14$
- $P(PC = \text{no}) = 5/14$
- $P(\text{age} = "<=30" \mid PC = \text{yes}) = 2/9$
- $P(\text{age} = "<=30" \mid PC = \text{no}) = 3/5$
- $P(\text{incm.} = \text{"med"} \mid PC = \text{yes}) = 4/9$
- $P(\text{incm.} = \text{"med"} \mid PC = \text{no}) = 2/5$
- $P(\text{student} = \text{"yes"} \mid PC = \text{yes}) = 6/9$
- $P(\text{student} = \text{"yes"} \mid PC = \text{no}) = 1/5$
- $P(\text{credit} = \text{"fair"} \mid PC = \text{"yes"}) = 6/9$
- $P(\text{credit} = \text{"fair"} \mid PC = \text{"no"}) = 2/5$
- $P(\text{all other combinations})$

...

## Using:

- $X = ("<=30"; "med"; "yes"; "fair")$
- $P(X \mid PC = \text{yes}) \rightarrow$   
 $P(\text{age} = "<=30" \mid PC = \text{yes}) * P(\text{incm.} = \text{"med"} \mid PC = \text{yes}) * P(\text{student} = \text{"yes"} \mid PC = \text{yes}) * P(\text{credit} = \text{"fair"} \mid PC = \text{"yes"}) \rightarrow 0.044$
- $P(X \mid PC = \text{no}) \rightarrow 0.019$
- $P(PC = \text{yes} \mid X) \rightarrow n * P(X \mid PC = \text{yes}) * P(PC = \text{yes}) \rightarrow n * 0.028$
- $P(PC = \text{no} \mid X) \rightarrow n * P(X \mid PC = \text{no}) * P(PC = \text{no}) \rightarrow n * 0.007$

**PREDICT "PC = yes"!!!**

# Avoiding the Zero-Probability Problem

- Naïve Bayesian prediction requires each conditional prob. be **non-zero**. Otherwise, the predicted prob. will be zero

$$P(X | C_i) = \prod_{k=1}^n P(x_k | C_i)$$

- Ex. Suppose a dataset with 1000 tuples, income=low (0), income= medium (990), and income = high (10)
- Use **Laplacian correction** (or Laplacian estimator)
  - *Adding 1 to each case*  
Prob(income = low) = 1/1003  
Prob(income = medium) = 991/1003  
Prob(income = high) = 11/1003

# Naïve Bayesian Classifier: Comments

---

- Advantages
  - Easy to implement and computationally efficient
  - Good results obtained in most of the cases
- Disadvantages
  - Assumption: class conditional independence, therefore loss of accuracy
  - Practically, dependencies exist among variables
    - E.g., hospitals: patients: Profile: age, family history, etc.  
Symptoms: fever, cough etc., Disease: lung cancer, diabetes, etc.
    - Dependencies among these cannot be modeled by Naïve Bayesian Classifier
- How to deal with these dependencies?  
→ Bayesian Belief Networks

# Bayesian Belief Networks

---

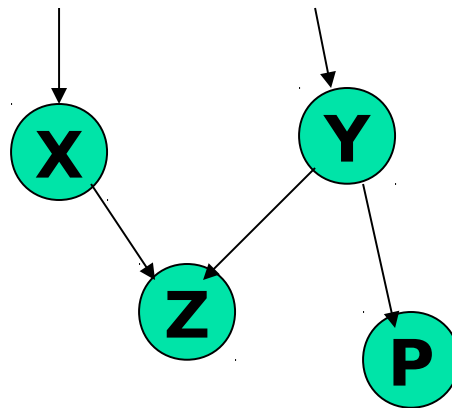
- **Bayesian belief networks** (also known as **Bayesian networks, probabilistic networks**): allow *class conditional independencies* between *subsets* of variables
- A (*directed acyclic*) graphical model of causal relationships
  - Represents dependency among the variables
  - Gives a specification of joint probability distribution



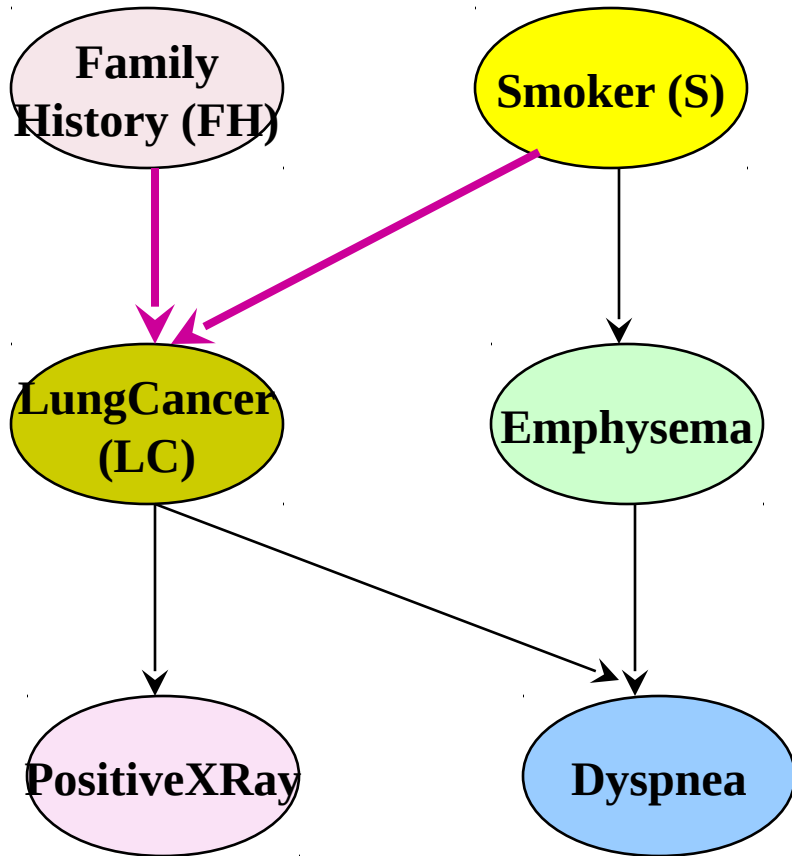
# Bayesian Belief Networks

---

- Nodes: random variables
- Links: dependency
- X and Y are the parents of Z, and Y is the parent of P
- No dependency between Z and P
- Has no loops/cycles



# Bayesian Belief Network: An Example



**CPT: Conditional Probability Table** for variable LungCancer:

	(FH, S)	(FH, ~S)	(~FH, S)	(~FH, ~S)
LC	0.8	0.5	0.7	0.1
~LC	0.2	0.5	0.3	0.9

shows the conditional probability for each possible combination of its parents

Derivation of the probability of a particular combination of values of **X**, from CPT:

**Bayesian Belief Network**

$$P(x_1, \dots, x_n) = \prod_{i=1}^n P(x_i | \text{Parents}(x_i))$$

# Training Bayesian Networks: Several Scenarios

---

- Scenario 1: Given both the network structure and all variables observable: *compute only the CPT entries*
- Scenario 2: Network structure known, some variables hidden: *gradient descent* (greedy hill-climbing) method, i.e., search for a solution along the steepest descent of a criterion function
  - Weights are initialized to random probability values
  - At each iteration, it moves towards what appears to be the best solution at the moment, w.o. backtracking
  - Weights are updated at each iteration & converge to local optimum

# Training Bayesian Networks: Several Scenarios

---

- Scenario 3: Network structure unknown, all variables observable: search through the model space to *reconstruct network topology*
- Scenario 4: Unknown structure, all hidden variables: No good algorithms known for this purpose
- D. Heckerman. [A Tutorial on Learning with Bayesian Networks](#). In *Learning in Graphical Models*, M. Jordan, ed.. MIT Press, 1999.

# Bayesian Belief Networks: Comments

---

- Advantages
  - Computationally heavier than naïve classifier, but still tractable
  - Handle (approximating) dependencies
  - Very good results (provided a meaningful network is designed & tuned)
- Disadvantages
  - Need expert problem knowledge or external mining algorithms for designing the network