Università degli Studi di Milano

Master Degree in Computer Science

Information Management course

Teacher: Alberto Ceselli

Lecture 17: 11/12/2012

Data Mining:

Concepts and Techniques

(3rd ed.)

— Chapter 8, 9 —

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

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

- 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 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 **X** be a data sample ("*evidence*"): class label is unknown
- Let H be a *hypothesis* that X belongs to class C
- Classification is to determine P(H|**X**), (*posteriori probability),* the probability that the hypothesis holds given the observed data sample **X**
- P(H) (*prior probability*), the initial probability
  - E.g., **X** buys computer, regardless of age, income
- P(**X**): probability that sample data is observed
- P(**X**|H) (likelyhood), the probability of observing the sample **X**, given that the hypothesis holds
  - E.g., Given that **X** buys computer, the prob. that X is 31..40, medium income

Bayesian Theorem

- Given training data **X**, *posteriori probability of a hypothesis* H, P(H|**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 prior/evidence

- Predicts **X** belongs to C_2 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
  $\mathbf{X} = (x_1, x_2, \ldots, x_n)$
- Suppose there are *m* classes $C_1, C_2, \ldots, C_m$.
- 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: $k*n^m$ 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):

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

- This greatly reduces the computation cost: Only counts the class distribution (k*n*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 μ and standard deviation σ

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

and $P(x_k|C_i)$ is

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

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.

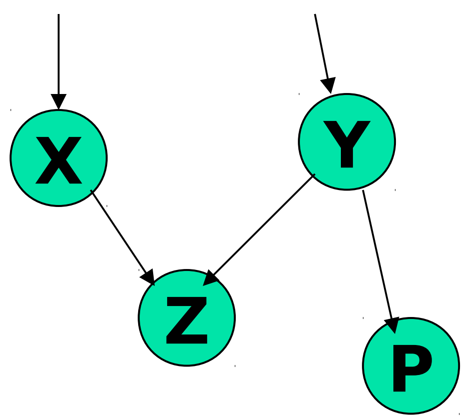
      Symptoms: fever, cough etc., Disease: lung cancer, diabetes, etc.
    - Dependencies among these cannot be modeled by Naïve Bayesian Classifier
- How to deal with these dependencies?
  → Bayesian Belief Networks

Bayesian Belief Networks

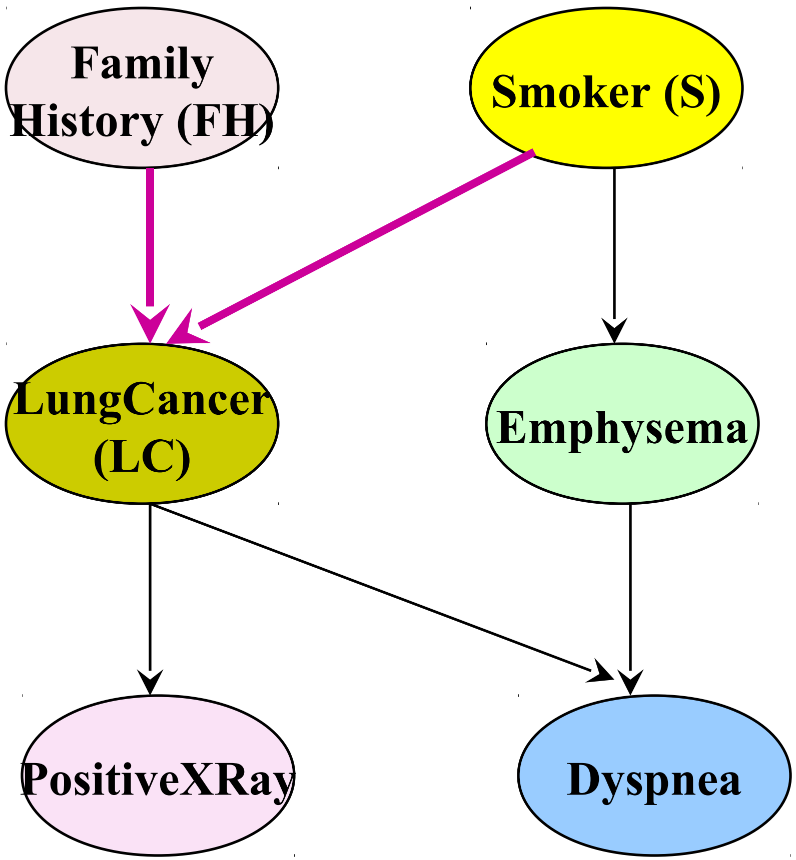
- **Bayesian belief networks** (also known as **Bayesian networks**, **probabilistic networks**): allow *class conditional independencies* between *subsets* of variables
- A (*directed acyclic*) graphical model of causal relationships
  - Represents <u>dependency</u> among the variables
  - Gives a specification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



Bayesian Belief Network

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:

$$P(x_1, \ldots, x_n) = \prod_{i=1}^{n} P(x_i \mid Parents(x_i))$$

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