Chapter 4: Classification & Prediction

- **4.1 Basic Concepts of Classification and Prediction**
- **4.2 Decision Tree Induction**
  - 4.2.1 The Algorithm
  - 4.2.2 Attribute Selection Measures
  - 4.2.3 Tree Pruning
  - 4.2.4 Scalability and Decision Tree Induction
- **4.3 Bayes Classification Methods**
  - 2.3.1 Naïve Bayesian Classification
  - 2.3.2 Note on Bayesian Belief Networks
- **4.4 Rule Based Classification**
- **4.5 Lazy Learners**
- **4.6 Prediction**
- **4.7 How to Evaluate and Improve Classification**

4.3 Bayes Classification Methods

- **What are Bayesian Classifiers?**
  - → Statistical classifiers
  - → Predict class membership probabilities: probability of a given tuple belonging to a particular class
  - → Based on Bayes' Theorem
- **Characteristics?**
  - → Comparable performance with decision tree and selected neural network classifiers
- **Bayesian Classifiers**
  - → Naïve Bayesian Classifiers
    - Assume independency between the effect of a given attribute on a given class and the other values of other attributes
  - → Bayesian Belief Networks
    - Graphical models
    - Allow the representation of dependencies among subsets of attributes

Bayes' Theorem In the Classification Context

- **X** is a data tuple. In Bayesian term it is considered **"evidence"**
- **H** is some **hypothesis** that X belongs to a specified class C

$$P(H \mid X) = \frac{P(X \mid H)P(H)}{P(X)}$$

- **P(H|X)** is the posterior probability of **H** conditioned on **X**

  Example: **predict** whether a costumer **will buy a computer** or **not**
  - Costumers are described by two attributes: **age** and **income**
  - **X** is a 35 years-old costumer with an income of 40k
  - **H** is the hypothesis that the costumer will buy a computer
  - **P(H|X)** reflects the probability that costumer **X will buy a computer** given that **we know** the costumers' **age** and **income**

Bayes' Theorem In the Classification Context

- **X** is a data tuple. In Bayesian term it is considered **"evidence"**
- **H** is some **hypothesis** that X belongs to a specified class C

$$P(H \mid X) = \frac{P(X \mid H)P(H)}{P(X)}$$

- **P(X|H)** is the posterior probability of **X** conditioned on **H**

  Example: **predict** whether a costumer **will buy a computer** or **not**
  - Costumers are described by two attributes: **age** and **income**
  - **X** is a 35 years-old costumer with an income of 40k
  - **H** is the hypothesis that the costumer will buy a computer
  - **P(X|H)** reflects the probability that costumer **X**, is 35 years-old and earns 40k, given that we know that the costumer will buy a computer

Bayes' Theorem In the Classification Context

- **X** is a data tuple. In Bayesian term it is considered **"evidence"**
- **H** is some **hypothesis** that X belongs to a specified class C

$$P(H \mid X) = \frac{P(X \mid H)P(H)}{P(X)}$$

- **P(H)** is the **prior** probability of **H**

  Example: **predict** whether a costumer **will buy a computer** or **not**

  - **H** is the hypothesis that the costumer will buy a computer
  - The prior probability of **H** is the probability that a costumer will buy a computer, regardless of age, income, or any other information for that matter
  - The posterior probability **P(H|X)** is **based on more information** than the prior probability **P(H)** which is **independent** from X

Bayes' Theorem In the Classification Context

- **X** is a data tuple. In Bayesian term it is considered **"evidence"**
- **H** is some **hypothesis** that X belongs to a specified class C

$$P(H \mid X) = \frac{P(X \mid H)P(H)}{P(X)}$$

- **P(X)** is the **prior** probability of **X**

  Example: **predict** whether a costumer **will buy a computer** or **not**
  - → Costumers are described by two attributes: **age** and **income**
  - → **X** is a 35 years-old costumer with an income of 40k
  - → **P(X)** is the probability that a person from our set of costumers is 35 years-old and earns 40k

Naïve Bayesian Classification

D: A training set of tuples and their associated class labels

Each tuple is represented by n-dimensional vector **$X(x_1,\ldots,x_n)$**, n measurements of n attributes $A_1,\ldots,A_n$

Classes: suppose there are **m** classes **$C_1,\ldots,C_m$**

Principle

- Given a tuple **X**, the classifier will predict that X belongs to the class having the **highest posterior probability** conditioned on X
- Predict that tuple X belongs to the class C_i if and only if

$$P(C_i \mid X) > P(C_j \mid X) \quad \text{for} \quad 1 \le j \le m, j \ne i$$

- Maximize **$P(C_i|X)$**: find the **maximum posteriori hypothesis**

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

- P(X) is **constant** for all classes, thus, **maximize $P(X|C_i)P(C_i)$**

Naïve Bayesian Classification

- To maximize **P(X|Ci)P(Ci)**, we need to know class prior probabilities
  - If the probabilities are not known, assume that **$P(C_1)=P(C_2)=\ldots=P(C_m)$ $\Rightarrow$ maximize $P(X|C_i)$**
  - Class prior probabilities can be estimated by **$P(C_i)=|C_{i,D}|/|D|$**
- Assume **Class Conditional Independence** to reduce computational cost of **$P(X|C_i)$**
  - given **$X(x_1,\ldots,x_n)$**, $P(X|C_i)$ is:

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

$$= P(x_1|C_i) \times P(x_2|C_i) \times \ldots \times P(x_n|C_i)$$

  - The probabilities $P(x_1|C_i)$, …$P(x_n|C_i)$ can be estimated from the training tuples

Estimating $P(x_i | C_i)$

- **Categorical Attributes**
  - Recall that $\mathbf{x_k}$ refers to the value of attribute A_k for tuple X
  - **X** is of the form $\mathbf{X(x_1,\ldots,x_n)}$
  - $\mathbf{P(x_k | C_i)}$ is the number of tuples of class C_i in D having the value x_k for A_k, divided by $|C_{i,D}|$, the number of tuples of class C_i in D
  - **Example**
    - **8** costumers in class $\mathbf{C_{yes}}$ (costumer will buy a computer)
    - **3** costumers among the 8 costumers **have high income**
    - $\mathbf{P(income{=}high | C_{yes})}$ the probability of a costumer having a high income knowing that he belongs to class C_{yes} is **3/8**
- **Continuous-Valued Attributes**
  - A continuous-valued attribute is assumed to have a **Gaussian (Normal)** distribution with mean μ and standard deviation σ

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

Estimating $P(x_i | C_i)$

- **Continuous-Valued Attributes**
  - The probability $P(x_k | C_i)$ is given by:

$$P(x_k \mid C_i) = g(x_k, \mu_{Ci}, \sigma_{Ci})$$

  - Estimate μ_{Ci} and σ_{Ci} the mean and standard variation of the values of attribute A_k for training tuples of class C_i
  - **Example**
    - **X** a 35 years-old costumer with an income of 40k (age, income)
    - Assume the age attribute is continuous-valued
    - Consider class C_{yes} (the costumer will buy a computer)
    - We find that in D, the costumers who will buy a computer are 38±12 years of age $\Rightarrow$ $\mu_{Cyes}=38$ and $\sigma_{Cyes}=12$

$$P(age = 35 \mid C_{yes}) = g(35, 38, 12)$$

Example

RID	age	income	student	credit-rating	class:buy_computer
1	youth	high	no	fair	no
2	youth	high	no	excellent	no
3	middle-aged	high	no	fair	yes
4	senior	medium	no	fair	yes
5	senior	low	yes	fair	yes
6	senior	low	yes	excellent	no
7	middle-aged	low	yes	excellent	yes
8	youth	medium	no	fair	no
9	youth	low	yes	fair	yes
10	senior	medium	yes	fair	yes
11	youth	medium	yes	excellent	yes
12	middle-aged	medium	no	excellent	yes
13	middle-aged	high	yes	fair	yes
14	senior	medium	no	excellent	no

Tuple to classify is

X (age=youth, income=medium, student=yes, credit=fair)

Maximize $P(X|C_i)P(C_i)$, for i=1,2

Example

Given **X (age=youth, income=medium, student=yes, credit=fair)**

Maximize **$P(X|C_i)P(C_i)$**, for i=1,2

First step: Compute $P(C_i)$. The prior probability of each class can be computed based on the training tuples:

P(buys_computer=yes)=9/14=0.643

P(buys_computer=no)=5/14=0.357

Second step: compute $P(X|C_i)$ using the following conditional prob.

P(age=youth|buys_computer=yes)=0.222

P(age=youth|buys_computer=no)=3/5=0.666

P(income=medium|buys_computer=yes)=0.444

P(income=medium|buys_computer=no)=2/5=0.400

P(student=yes|buys_computer=yes)=6/9=0.667

P(tudent=yes|buys_computer=no)=1/5=0.200

P(credit_rating=fair|buys_computer=yes)=6/9=0.667

P(credit_rating=fair|buys_computer=no)=2/5=0.400

Example

P(X | buys_computer=yes)= P(age=youth | buys_computer=yes)×
P(income=medium | buys_computer=yes) ×
P(student=yes | buys_computer=yes) ×
P(credit_rating=fair | buys_computer=yes)
= 0.044

P(X | buys_computer=no)= P(age=youth | buys_computer=no)×
P(income=medium | buys_computer=no) ×
P(student=yes | buys_computer=no) ×
P(credit_rating=fair | buys_computer=no)
= 0.019

Third step: compute **$P(X | C_i)P(C_i)$** for each class

P(X | buys_computer=yes)P(buys_computer=yes)=0.044 ×0.643=**0.028**

P(X | buys_computer=no)P(buys_computer=no)=0.019 ×0.357=**0.007**

The naïve Bayesian Classifier predicts buys_computer=yes for tuple X

Avoiding the 0-Probability Problem

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

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

- 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

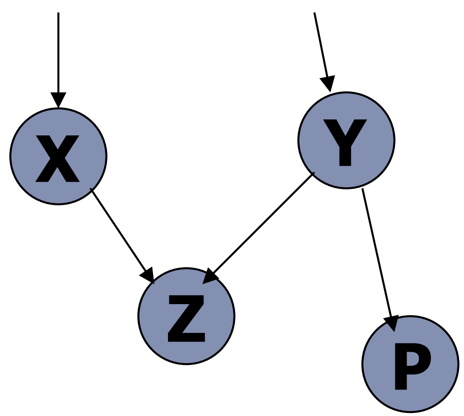
    Prob(income = high) = 11/1003
  - → The "corrected" prob. estimates are close to their "uncorrected" counterparts

Summary of Section 4.3

- Advantages
  - Easy to implement
  - 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

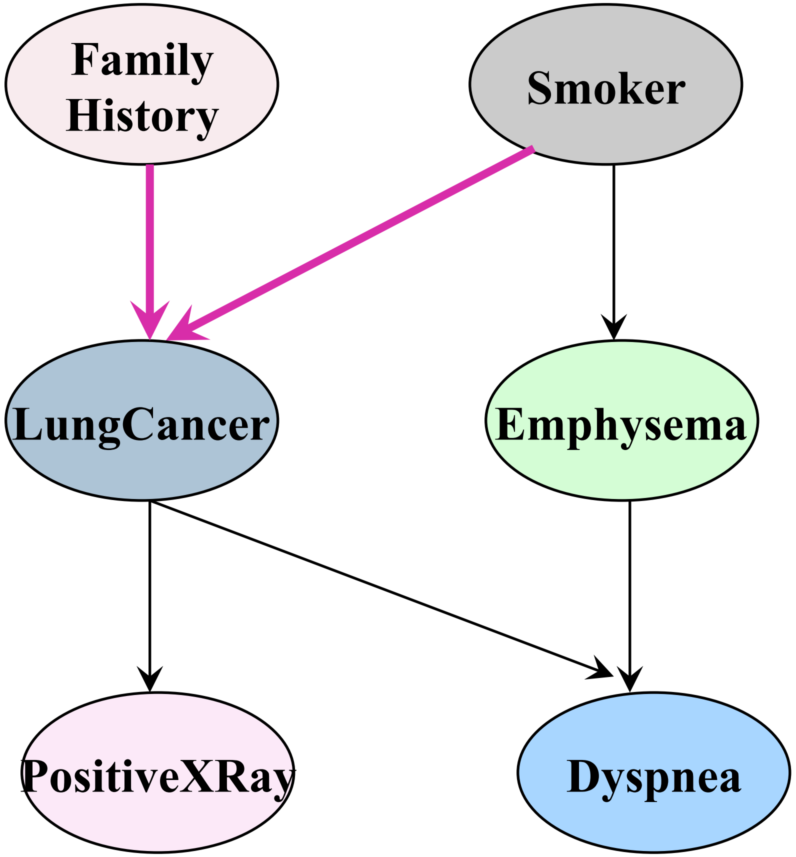
4.3.2 Bayesian Belief Networks

- Bayesian belief network allows a *subset* of the variables conditionally independent
- A graphical model of causal relationships
  - Represents dependency among the variables
  - Gives a specification of joint probability distribution



- 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 or cycles

Example



Bayesian Belief Networks

The **conditional probability table** (**CPT**) 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

CPT 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:

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

Training Bayesian Networks

- Several scenarios:
  - Given both the network structure and all variables observable: *learn only the CPTs*
  - Network structure known, some hidden variables: *gradient descent* (greedy hill-climbing) method, analogous to neural network learning
  - Network structure unknown, all variables observable: search through the model space to *reconstruct network topology*
  - Unknown structure, all hidden variables: No good algorithms known for this purpose

Summary of Section 4.3

- Bayesian Classifiers are **statistical classifiers**
- They provide **good accuracy**
- Naïve Bayesian classifier assumes **independency** between attributes
- **Causal relations** are captured by Bayesian Belief Networks