Chapter 6. Classification and Prediction

- What is classification? What is prediction?
- Issues regarding classification and prediction
- Classification by decision tree induction
- Bayesian classification
- Rule-based classification
- Classification by back propagation
- Support Vector Machines (SVM)
- Associative classification
- Lazy learners (or learning from your neighbors)
- Other classification methods
- Prediction
- Accuracy and error measures
- Ensemble methods
- Model selection
- Summary

Bayesian Classification: Why?

- A statistical classifier:
  - performs *probabilistic prediction, i.e.,* predicts the membership probabilities for different classes
  - Foundation: Based on Bayes' theorem
- Performance:
  - A simple *naïve Bayesian classifier* has comparable performance with decision trees and neural network classifiers

Bayesian Classification: Why?

- Incremental:
  - Each training example can incrementally increase/decrease the probability that a hypothesis is correct
  - Prior knowledge can be combined with observed data, rather than training from the scratch
- Standard:
  - They can provide a standard of optimal decision making against which other methods can be measured

Bayesian Theorem: Basics

- Let **X** be a data sample (*evidence*) whose class label is unknown
- Let H be a *hypothesis* that X belongs to a class C
- Classification
  - to determine P(H|**X**), the probability that the hypothesis holds when the observed data sample **X** is given

Bayesian Theorem: Basics

- P(H) (*prior probability*)
  - The initial probability (independent of a specific **X**)
  - E.g., **X** will buy computer, regardless of age, income, …
- P(**X**)
  - The probability that sample data is observed
- P(**X**|H) (*posteriori probability*)
  - The probability of observing the sample **X**, given that the hypothesis holds
  - E.g., Given that **X** will buy computer, the prob. that X is 31..40, medium income

Bayesian Theorem

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

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

Bayesian Theorem

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

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

- Informally, this can be written as

  likelihood = posteriori * prior / evidence

- Predicts **X** belongs to C_i iff the probability P(C_i|**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

Towards Naïve Bayesian Classifier

- 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 maximum $P(C_i|\mathbf{X})$
- This can be derived from Bayes' theorem

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

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

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

needs to be maximized

Derivation of Naïve Bayes Classifier

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

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

- This greatly reduces the computation cost: Only counts the class distribution

Derivation of Naïve Bayes Classifier

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

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

Naïve Bayesian Classifier: Training Dataset

Class:
C1:buys_computer = 'yes'
C2:buys_computer = 'no'

Data sample
X = (age <=30,
Income = medium,
Student = yes
Credit_rating = Fair)

age	income	student	credit_rating	comp
<=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

Naïve Bayesian Classifier: An Example

- $P(C_i)$: P(buys_computer = "yes") = 9/14 = 0.643
  P(buys_computer = "no") = 5/14= 0.357

- Compute $P(X|C_i)$ for each class
  P(age = "<=30" | buys_computer = "yes") = 2/9 = 0.222
  P(age = "<= 30" | buys_computer = "no") = 3/5 = 0.6
  P(income = "medium" | buys_computer = "yes") = 4/9 = 0.444
  P(income = "medium" | buys_computer = "no") = 2/5 = 0.4
  P(student = "yes" | buys_computer = "yes) = 6/9 = 0.667
  P(student = "yes" | buys_computer = "no") = 1/5 = 0.2
  P(credit_rating = "fair" | buys_computer = "yes") = 6/9 = 0.667
  P(credit_rating = "fair" | buys_computer = "no") = 2/5 = 0.4

- **X = (age <= 30 , income = medium, student = yes, credit_rating = fair)**

$P(X|C_i)$: P(X|buys_computer = "yes") = 0.222 x 0.444 x 0.667 x 0.667 = 0.044
P(X|buys_computer = "no") = 0.6 x 0.4 x 0.2 x 0.4 = 0.019

$P(X|C_i)$*$P(C_i)$: P(X|buys_computer = "yes") * P(buys_computer = "yes") = 0.028
P(X|buys_computer = "no") * P(buys_computer = "no") = 0.007

Therefore, X belongs to class ("buys_computer = yes")

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 the idea of 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, not allowing zero probability

Naïve Bayesian Classifier: Comments

- 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., Patients' Profiles: age, family history; Symptoms: fever, cough; Disease: cold, lung cancer, diabetes
    - Dependencies among these cannot be modeled by Naïve Bayesian Classifier
- How to deal with these dependencies?
  - Bayesian Belief Networks (not dealt with here)

Chapter 6. Classification and Prediction

- What is classification? What is prediction?
- Issues regarding classification and prediction
- Classification by decision tree induction
- Bayesian classification
- Rule-based classification
- Classification by back propagation
- Support Vector Machines (SVM)
- Associative classification
- Lazy learners (or learning from your neighbors)
- Other classification methods
- Prediction
- Accuracy and error measures
- Ensemble methods
- Model selection
- Summary

Using IF-THEN Rules for Classification

- Represent the knowledge in the form of IF-THEN rules
  - R: IF *age* = youth AND *student* = yes
    THEN *buys_computer* = yes
  - Rule antecedent/precondition vs. rule consequent
- Assessment of a rule R: *coverage* and *accuracy* (see Ex. 6.6)
  - n_{covers} = # of tuples *covered* by R
  - $n_{correct}$ = # of tuples *correctly classified* by R

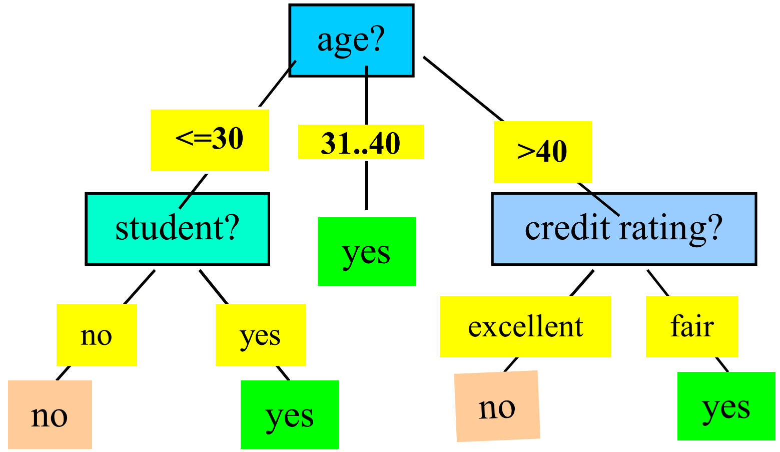
coverage(R) = n_{covers} /|D| /* D: training data set */

accuracy(R) = $n_{correct}$ / n_{covers}

Using IF-THEN Rules for Classification

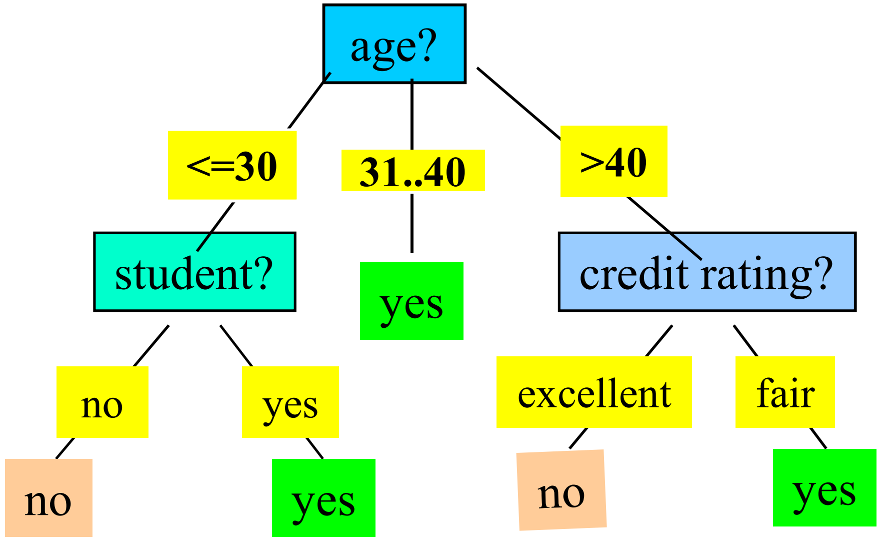
- If more than one rule is triggered, need **conflict resolution**
  - Size ordering: assign the highest priority to the triggering rules that have the "toughest" requirement (i.e., with the *most attribute test*)
  - Class-based ordering: decreasing order of *prevalence (frequency) or misclassification cost per class*
  - Rule-based ordering (**decision list**): rules are organized into one long priority list
    - According to some measure of rule quality or by experts

Rule Extraction from a Decision Tree



- Rules are easier to understand than a large tree
- One rule is created for each path from the root to a leaf
- Each attribute-value pair along a path forms a conjunction: the leaf holds the class prediction
- Rules are mutually exclusive and exhaustive

Rule Extraction from a Decision Tree



- Example: Rule extraction from our *buys_computer* decision-tree

  IF *age* = young AND *student* = *no*, THEN *buys_computer* = *no*

  IF *age* = young AND *student* = *yes*, THEN *buys_computer* = *yes*

  IF *age* = mid-age, THEN *buys_computer* = *yes*

  IF *age* = old AND *credit_rating* = *excellent*, THEN *buys_computer* = *yes*

  IF *age* = young AND *credit_rating* = *fair*, THEN *buys_computer* = *no*

Chapter 6. Classification and Prediction

- What is classification? What is prediction?
- Issues regarding classification and prediction
- Classification by decision tree induction
- Bayesian classification
- Rule-based classification
- Classification by back propagation
- Support Vector Machines (SVM)
- Associative classification
- Lazy learners (or learning from your neighbors)
- Other classification methods
- Prediction
- Accuracy and error measures
- Ensemble methods
- Model selection
- Summary

Associative Classification

- Associative classification
  - Association rules are generated and analyzed for use in classification
  - Search for strong associations between frequent patterns (conjunctions of attribute-value pairs) and class labels
  - Classification: Based on evaluating a set of rules in the form of

$$P_1 \wedge p_2 \ldots \wedge p_l \rightarrow \text{“}A_{class} = C\text{”} \text{ (conf, sup)}$$

- Why effective?
  - It explores highly confident associations among multiple attributes
    - May overcome some constraints introduced by decision-tree induction, which considers only one attribute at a time
  - In many studies, associative classification has been found to be more accurate than some traditional classification methods, such as C4.5

Chapter 6. Classification and Prediction

- What is classification? What is prediction?
- Issues regarding classification and prediction
- Classification by decision tree induction
- Bayesian classification
- Rule-based classification
- Classification by back propagation
- Support Vector Machines (SVM)
- Associative classification
- Lazy learners (or learning from your neighbors)
- Other classification methods
- Prediction
- Accuracy and error measures
- Ensemble methods
- Model selection
- Summary

Lazy vs. Eager Learning

- Lazy vs. eager learning
  - Eager learning (the previously discussed methods)
    - Given a set of training set, constructs a classification model *before receiving a* new test tuple to classify
  - Lazy learning
    - Simply stores training data (or only minor processing) and *just waits* until a test tuple is given
- Lazy: much less time in training but more time in predicting

Lazy vs. Eager Learning

- Accuracy
  - A eager method must commit to a *single hypothesis* that covers the entire instance space
  - A lazy method effectively uses a richer hypothesis space since it uses many local linear functions

Lazy Learner: Instance-Based Methods

- Instance-based learning:
  - Store training examples and delay the processing (i.e., lazy evaluation)
    - Until a new instance is received to be classified
- Typical example: *k*-nearest neighbor approach

The *k*-Nearest Neighbor Algorithm

- All instances correspond to points in n-D space
  - A distance, dist($\mathbf{X_1}$, $\mathbf{X_2}$), is defined over the space
- k-nearest neighbors are retrieved in terms of the distance
- The test sample is classified by the class label of a majority of the k-nearest neighbors