Search and Data Mining: Techniques

Classification

Anya Yarygina

Boris Novikov

Introduction

- Classification and prediction
- Bayesian classification
- Classification by decision tree induction
- Support vector machines
- Lazy learners
- Prediction
- Accuracy and error measures
- Evaluating the accuracy of a classifier or predictor
- Ensemble methods

Reference

Jiawei Han, Micheline Kamber. Data Mining: Concepts and Techniques, 2nd Edition. The Morgan Kaufmann Series in Data Management Systems, 2006

What is classification? What is prediction?

- Classification
  - model or classifier is constructed to predict categorical labels
- (Numeric) prediction
  - model or predictor constructed predicts a continuous-valued function, or ordered value

How does classification work?

- Learning (training): training data are analyzed by classification algorithm to construct a classifier or learned model
- Classification: test data are used to estimate the accuracy of the classifier; classifier is applied to the classification of new data

- Training (testing) set: set of tuples and their associated class labels
$$\{(X^i, y^i)\}$$
- Tuple is represented by an n-dimensional attribute vector
$$X = (x_1, \dots, x_n)$$
- Learning of mapping function
$$y = f(X)$$

Issues regarding classification and prediction

- Preparing data for classification and prediction
  - Data cleaning
    - Noise
    - Missing values
  - Relevance analysis
    - Correlation analysis
    - Attribute subset selection
  - Data transformation and reduction
    - Normalization
    - Discretizing
    - PCA
- Comparing classification and prediction methods
  - Accuracy
  - Speed
  - Robustness
  - Scalability
  - Interpretability

Naïve Bayesian classification

- Training set of tuples and associated labels

$$\{(X^i, C^i)\}$$
$$X = (x_1, \dots, x_n)$$
$$C_1, \dots, C_m$$

- Given a tuple X, the classifier will predict the class based on posterior probability conditioned on X

$$P(C_i|X) > P(C_j|X)$$

- Maximum posteriori hypothesis

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

- Class conditional independence

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

- Estimate $P(x_k|C_i)$
  - Categorical attribute
  - Continuous-valued attribute

$$g(x, \mu, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$
$$P(x_k|C_i) = g(x_k, \mu_{C_i}, \sigma_{C_i})$$

  - Laplacian correction

- Bayes theorem

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

Other classification techniques

- Classification by decision tree induction
- Support vector machines
- Lazy learners

Regression

Linear regression

- Straight-line regression analysis

$$y = w_0 + w_1 x$$

  - Response variable
  - Single predictor variable
  - Regression coefficients
  - Method of least squares
    - Best-fitting straight line as the one that minimizes the error between the actual data and the estimate of the line

$$w_1 = \frac{\sum_i (x_i - \bar{x})(y_i - \bar{y})}{\sum_i (x_i - \bar{x})^2}$$

$$w_0 = \bar{y} - w_1 \bar{x}$$

- Multiple linear regression

$$y = WX$$

$$y = w_0 + w_1 x_1 + w_2 x_2$$

Non-linear regression

- Polynomial regression

$$y = w_0 + w_1 x + w_2 x^2 + w_3 x^3$$

Accuracy and error measures

Classifier accuracy measures

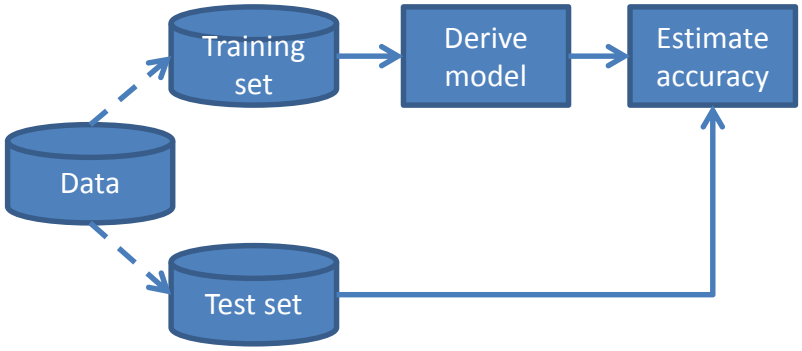
- Accuracy
- Error rate
- Confusion matrix
- $sensitivity = \frac{t\_pos}{pos}$
- $specificity = \frac{t\_nes}{neg}$
- $precision = \frac{t\_pos}{(t\_pos+f\_pos)}$

Predictor error measures

- Absolute error:
  $|y_i - y_i'|$
- Squared error:
  $(y_i - y_i')^2$
- Mean absolute error:
  $\frac{\sum_{i=1}^{d} |y_i - y_i'|}{d}$
- Mean squared error:
  $\frac{\sum_{i=1}^{d} (y_i - y_i')^2}{d}$
- Relative absolute error:
  $\frac{\sum_{i=1}^{d} |y_i - y_i'|}{\sum_{i=1}^{d} |y_i - \bar{y}|}$
- Relative squared error:
  $\frac{\sum_{i=1}^{d} (y_i - y_i')^2}{\sum_{i=1}^{d} (y_i - \bar{y})^2}$

Evaluating the accuracy of classifier or predictor

- Holdout method
  - Training set
  - Test set
- Random sampling
  - Holdout method repeated k times
- Cross validation
  - k mutually exclusive subsets (folds) of approximately equal size
  - Training and testing is performed k times: one fold for testing, others for training
  - Leave-one-out
- Bootstrap
  - Samples the given training tuples with replacement
  - .632 bootstrap
    - Bootstrap sample of size equal to all set for training
    - Not selected tuples for testing



Ensemble methods

Bagging

Ensemble of models for learning scheme where each model gives an equally-weighted predictions

Boosting

- Ensemble of classifiers where each one gives a weighted vote

Ensemble methods

Bagging

- Method

for $i = 1$ to k **do** // create k models:
    create bootstrap sample, D_i, by sampling D with replacement;
    use D_i to derive a model, M_i;
endfor

- Application

if classification **then**
    let each of the k models classify $\boldsymbol{X}$ and return the majority vote;
if prediction **then**
    let each of the k models predict a value for $\boldsymbol{X}$ and return the average predicted value;

Boosting

- Method

initialize the weight of each tuple in D to $1/d$;
for $i = 1$ to k **do** // for each round:
    sample D with replacement according to the tuple weights to obtain D_i;
    use training set D_i to derive a model, M_i;
    compute $error(M_i)$, the error rate of M_i (Equation 6.66)
    if $error(M_i) > 0.5$ **then**
        reinitialize the weights to $1/d$
        go back to step 3 and try again;
    endif
    for each tuple in D_i that was correctly classified **do**
        multiply the weight of the tuple by $error(M_i)/(1 - error(M_i))$; // update weights
    normalize the weight of each tuple;
endfor

$$error(M_i) = \sum_j w_j * err(X_j)$$

- Application

initialize weight of each class to 0;
for $i = 1$ to k **do** // for each classifier:
    $w_i = log\frac{1-error(M_i)}{error(M_i)}$; // weight of the classifier's vote
    $c = M_i(\boldsymbol{X})$; // get class prediction for $\boldsymbol{X}$ from M_i
    add w_i to weight for class c
endfor
return the class with the largest weight;

Outline

- Classification and prediction
- Bayesian classification
- Classification by decision tree induction
- Support vector machines
- Lazy learners
- Prediction
- Accuracy and error measures
- Evaluating the accuracy of a classifier or predictor
- Ensemble methods