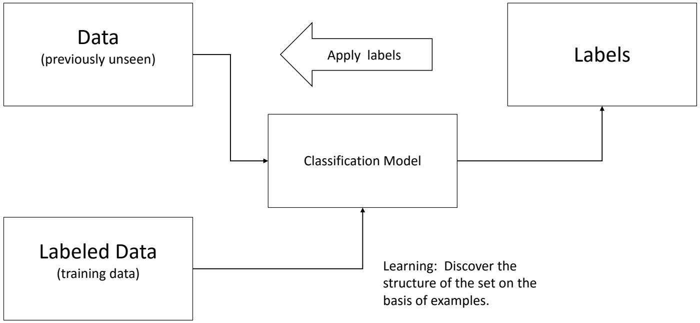
Data Mining, Lecture 8: Classification

S. Nõmm

[1]Department of Computer Science, Tallinn University of Technology

March 21, 2016

Introduction



Data
(previously unseen)

Apply labels

Labels

Classification Model

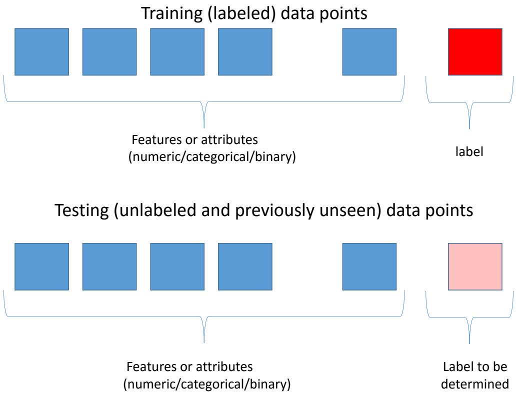
Labeled Data
(training data)

Learning: Discover the structure of the set on the basis of examples.

Introduction

- Classification problem may be seen as learning the structure of a data set of examples, already partitioned into categories or classes (labeled data set).
- Typically, learning leads the model.
- Model is used to *estimate* labels of the **previously unseen** data.
- Majority of the classification algorithms consist of two phases: *Training* and *Testing*
- Usually output of the classification is either *Label* or *Numeric score*.

Data


Training (labeled) data points

Features or attributes (numeric/categorical/binary)

label

Testing (unlabeled and previously unseen) data points

Features or attributes (numeric/categorical/binary)

Label to be determined

Feature selection

- Filter models: a subset of features is evaluated with the use of a class-sensitive discriminative criterion.
    - Gini index.
    - Entropy.
    - Fisher score.
    - Fisher linear discriminant.
- Wrapper models.
- Embedded models.

Gini index

- Measures the discriminative power of a particular feature.
- Typically, it is used for categorical variables, but it can be generalized to numeric attributes by the process of discretization.
- Let $v_1, \ldots, v_r$ are the possible values of the particular categorical attribute.
- Let p_j denotes the fraction of the data points containing attribute value v_i belonging to the class $j \in \{1, \ldots, k\}$ to the data points containing attribute value v_i then *Gini* index defined as follows:

$$G(v_i) = 1 - \sum_{j=1}^{k} p_j^2.$$

- The value $1 - 1/k$ indicates that the different classes are distributed evenly for a particular attribute value.
- Lower values of the Gini index imply greater discrimination.

Gini index

The value-specific Gini index may be converted into an attribute wise Gini index.

- $$G(v_i) = 1 - \sum_{j=1}^{k} p_j^2.$$
- Let n_j denote the number of data points that take the value v_i, $\sum_i = 1^r n_i = n$. Overall Gini index is defined as
  $$G = \sum_{i=1}^{r} \frac{n_i G(v_i)}{n}.$$
- Lower values of the Gini index imply greater discriminative power.
- More classical definition:
  $$G = \frac{\sum_{i=1}^{n} \sum_{j=1}^{n} |x_i - x_j|}{2n^2 \mu}$$

Entropy

- The class-based entropy measure is related to notions of information gain resulting from fixing a specific attribute value.
- The class- base entropy is defied as follows:

$$E(v_i) = -\sum_{j=1}^{k} p_i \log_2(p_j))$$

  takes its values in $[0, \log_2(k)]$, whereas greater values indicate greater mixing.
- By analogy with Gini index one may define overall Entropy as

$$E = \sum_{i=1}^{r} \frac{n_i E(v_i)}{n}.$$

Fisher score

- The Fisher score is naturally designed for numeric attributes to measure the ratio of the average interclass separation to the average intraclass separation.
- The larger the Fisher score, the greater the discriminatory power of the attribute.
- Let μ_j and σ_j denote the mean and the standard deviation of the of the data points belonging to the class j, for a particular feature. And let p_j be the fraction of the points belonging to the class j. Finally let μ define the mean of the entire data set. The Fisher index is defined as follows:

$$F = \frac{\displaystyle\sum_{j=1}^{k} p_j(\mu_j - \mu)^2}{\displaystyle\sum_{j=1}^{k} p_j \sigma_j^2}$$

- The attributes with the largest value of the Fisher score may be selected for use with the classification algorithm.

Wrapper Models

- Filter models are agnostic to the particular classification algorithm being used. In some cases, it may be useful to leverage the characteristics of the specific classification algorithm to select features.
- Wrapper models can optimize the feature selection process to the classification algorithm at hand.
- Let $\mathcal{A}$ denote a specific classification algorithm. The basic strategy in wrapper models is to iteratively refine a current set of features F by successively adding features to it.
- The algorithm starts with the F set to be empty then two following steps are repeated
    - Create an augmented set of features F by adding one or more features to the current feature set.
    - Use a classification algorithm $\mathcal{A}$ to evaluate the accuracy of the set of features F. Use the accuracy to either accept or reject the augmentation of F.

Decision Trees

- Decision trees are a classification methodology, wherein the classification process is modeled with the use of a set of hierarchical decisions on the feature variables, arranged in a tree-like structure.
- The decision at a particular node of the tree, which is referred to as the split criterion, is typically a condition on one or more feature variables in the training data.
- Split Criteria, Stopping criterion and pruning.

Rule-Based Classifiers

- Rule-based classifiers use a set of ifthen rules $\mathcal{R} = \{\mathcal{R}_1, \ldots, \mathcal{R}_m\}$ to match antecedents to consequents. A rule is typically expressed in the following form:

$$\text{IF} \quad Condition \quad \text{THEN} \quad Condition$$

- The condition on the left-hand side of the rule, also referred to as the antecedent, may contain a variety of logical operators.
- The right-hand side of the rule is referred to as the consequent, and it contains the class variable.
- The training phase of a rule-based algorithm creates a set of rules. The classification phase for a test instance discovers all rules that are triggered by the test instance.
- In some cases, methods are required to resolve the conflicts in class label prediction.

Properties of the rule sets

- Mutually exclusive rules: Each rule covers a disjoint partition of the data. Therefore, at most one rule can be triggered by a test instance.
- Exhaustive rules: The entire data space is covered by at least one rule. Therefore, every test instance triggers at least one rule.
- In cases where rule sets are not mutually exclusive, conflicts in the rules triggered by a test instance can be resolved in one of two ways:
    - Rule ordering: The rules are ordered by priority, which may be defined in a variety of ways. One possibility is to use a quality measure of the rule for ordering.
    - Unordered rules: No priority is imposed on the rule ordering. The dominant class label among all the triggered rules may be reported. Such an approach can be more robust because it is not sensitive to the choice of the single rule selected by a rule ordering scheme.

Associative classifiers

- Associative classifiers are a popular strategy because they rely on association pattern mining, for which many efficient algorithmic alternatives exist.
- The basic strategy for an associative classifier is as follows:
    - Mine all class-based association rules at a given level of minimum support and confidence.
    - For a given test instance, use the mined rules for classification.

Some other methods

- Probabilistic Classifiers
  - Naive Bayes Classifier
  - Logistic Regression
- Support Vector Machines
- Neural Networks
- Instance Based Learning

Bayes theorem

- Let us suppose that there k classes are given.
- The *posterior probability* of a class C_k for an input x is:

$$p(C_k \mid x) = \frac{p(\boldsymbol{x} \mid C_k)p(C_k)}{p(x)}$$

- $p(\boldsymbol{x} \mid C_k)$ is the likelihood, $p(C_k)$ is the *prior probability*, $p(x)$ is the *marginal data likelihood*.
- $p(C_k)$ is the probability of a class $p(C_k)$ *a priori*, before getting about any knowledge about the data.
- $p(C_k \mid \boldsymbol{x})$ is the class probability *a posteriori*, after getting knowledge about the data.
- Bayes theorem updates prior distribution into posterior on the basis of empiric information.

Conditional and unconditional independence

- If X and Y are *unconditionally independent* then their joint distribution is the product of the marginal distributions:

$$X \perp Y \Leftrightarrow p(X, Y) = p(X)p(Y)$$

- If the influence is mediated through a third variable Z, then X and Y are said to be *conditionally independent*

$$X \perp Y \mid Z \Leftrightarrow p(X, Y \mid Z) = p(X \mid Z)p(Y \mid Z)$$

- Conditional independence does not imply unconditional independence and vice versa:

$$X \perp Y \mid Z \not\Leftrightarrow X \perp Y$$

Example: Spam detection

- Inputs $\boldsymbol{x}$ are the e-mail messages (text documents)
- m labeled training pairs (x_i, y_i), where $y_i \in \{0, 1\}$. 0 - indicates "clear" message and 1 - spam
- Task is to classify a new e-mail spam/not a spam
- According to Bayes theorem

$$p(y \mid x) = \frac{p(\boldsymbol{x} \mid y)p(y)}{p(\boldsymbol{x})} \propto p(\boldsymbol{x} \mid y)$$

- The demoniator may be computed as

$$p(\boldsymbol{x}) = \sum_{y'} p(\boldsymbol{x} \mid y')p(y')$$

Feature representation

- Amount of the training data may pose a problem in computing likelihood $p(\boldsymbol{x} \mid y)$. (Low amout of training data may prevent reliable computation of the likelihood).
- Consider the document as the set of words
- for the given vocabulary V present each document as a binary vector.
- If word belong to the vocabulary corresponding element take the value 1 and 0 otherwise.
- This approach will lead to the following likelihood function

$$p(\boldsymbol{x} \mid y) = \prod_{j=1}^{|V|} p(x_j \mid y)$$

Naïve Bayes assumption

- Likelihood is computed as:

$$p(\boldsymbol{x} \mid y) = \prod_{j=1}^{n} p(x_j \mid y)$$

- *Naïve Bayes assumption:* the features are conditionally independent given the class label.
- the word *naïve* reveres to the fact that actually features are not expected to be independent or conditionally independent.
- Model has relatively few parameters and therefore immune to overfilling.

Naïve Bayes model

- Parameters of the model

$$\begin{aligned}\theta_{j|y=1} &= p(x_1 = 1 \mid y = 1)\\ \theta_{j|y=0} &= p(x_1 = 1 \mid y = 0)\\ \theta_y &= p(y = 1)\end{aligned}$$

- The MLE estiamtes of the parameters are:

$$\begin{aligned}\theta_{j|y=1} &= \frac{\sum_{i=1}^{m} \mathbb{I}(x_{i,j} = 1, y_i = 1)}{\sum_{i=1}^{m} \mathbb{I}(y_i = 1)}\\ \theta_{j|y=0} &= \frac{\sum_{i=1}^{m} \mathbb{I}(x_{i,j} = 1, y_i = 0)}{\sum_{i=1}^{m} \mathbb{I}(y_i = 0)}\\ \theta_y &= \frac{\sum_{i=1}^{m} \mathbb{I}(y_i = 1)}{m}\end{aligned}$$

Prediction with naïve Bayes model

- the goal is to find wether a new element is of class 1 or 0 (in the example of spam filtering wether given e-mail message is spam or not).
- According to Bayes theorem.

$$p(y=1 \mid \boldsymbol{x}, \boldsymbol{\theta}) \propto p(\boldsymbol{x} \mid y, \boldsymbol{\theta}) p(y \mid \boldsymbol{\theta}) = p(y=1 \mid \theta) \prod_{j=1}^{n} p(x_{i,j} \mid y=1, \boldsymbol{\theta})$$

$$p(y=0 \mid \boldsymbol{x}, \boldsymbol{\theta}) \propto p(\boldsymbol{x} \mid y, \boldsymbol{\theta}) p(y \mid \boldsymbol{\theta}) = p(y=0 \mid \theta) \prod_{j=1}^{n} p(x_{i,j} \mid y=0, \boldsymbol{\theta})$$

- Predict the class with highest posterior probability:

$$y^* = \arg \max_{y \in \{0,1\}} p(y \mid \boldsymbol{x}, \boldsymbol{\theta})$$