Data Mining

Lab 3: Data classification

Outline:

- The classification problem
- Instance based classifiers (k- NearestNeighbor)
- Decision trees (J48)
- Rule-based classifiers (OneR, JRip)

1. Classification

Quick reminder of main concepts:

- Aim of a classifier: establish to which class the input data belongs

- Constructing a classifier: extract a classification model using the data from a training set; the construction process is called supervised training/learning

- Performance evaluation: confusion matrix, accuracy, sensitivity, specificity, recall, precision, F-measure

Exercise 1: Visualization of decision surfaces

a) Launch BoundaryVisualizer (Weka GUI Chooser -> Visualization -> Boundary Visualizer)

b) Open the file "iris2D.arff" (it contains only two attributes: petalLength and petalWidth from iris.arff). Apply the following classifiers: IBk (k=1, k=2, k=5), J48, OneR, JRip, NaiveBayes and compare the results.

2. Instance Based Learning Classifiers

The simplest instance based classifier is the kNN (k nearest neighbours) .

Construction:

- The classification model consists of the training set (there is no specific training step – this is why these classifiers are considered lazy classifiers)

Usage:

- For an input data the class is predicted based on the following steps:
  - Step 1: Find the most similar k instances from the training set
  - Step 2: Identify the dominant class among the classes corresponding to the k instances selected at the previous step

Weka implementation: IB1 (only 1 neighbour is used), IBk (k neighbours are used)

Exercise 2: Choose the "right" number of neighbors for kNN

a) Open in Weka the file "breast-w.arff". For IBk identify the value(s) of k (out of {1,2,3,4,5,6,7,8,9,10}) which lead(s) to the best classifier performance (Hint: modify the value of k from IBk).

b) Open in Weka the file "glass.arff" and apply the same processing steps.

3. Decision trees

A decision tree is a classification tree which contains:

- Internal nodes: each internal node has an attribute associated and some splitting conditions (each splitting condition corresponds to a branch in the tree)
- Leaf nodes: each leaf node has a class associated

Top-down construction:

- For the root node one identifies the splitting attribute and the splitting conditions which leads to the highest informational gain based on the available dataset
- For each branch recursively apply the same procedure until a stopping condition is identified, e.g. the dataset corresponding to the current node is homogeneous (all data belong to the same class) or it is too small.

Remark: one of the most important aspects when constructing decision trees is the complexity level of the inferred tree. Simpler trees (i.e. fewer nodes) are always preferred. The simplification of the tree (pruning) can be done at different stages:

- During the tree construction (e.g. the branching process is stopped when the number of instances in the current dataset is smaller than a given threshold)
- After the tree construction some subtrees can be replaced with leaf nodes (as long as the classifier performance is not significantly decreased)

Decision tree usage: for a given instance one identify the tree branch which matches the corresponding attribute values; the label of the leaf node in that branch will be the predicted class.

Weka implementation: Id3, J48

Exercise 3: Comparing algorithms for decision trees construction

a) Open in Weka the file "weather_nominal.arff". Using Weka Explorer compare the performance of the following classifiers: Id3, J48 (in this cased both the "pruned" and "unpruned" variants are analyzed – this option is set in the list of parameters of J48)

b) Open in Weka the file "weather_numeric.arff" and try to apply the same processing steps as at (a).

c) Open in Weka the file "glass.arff" and analyze the impact on the J48 performance of the minimal number of objects which correspond to a leaf node (the value of the parameter minNumObj)

Exercise 4: Construction of a binary decision tree by interactive selection of data

Open in Weka the file "iris2D.arff" and apply the following processing steps:

a) Select "User Classifier" (group Tree from Classify)

b) In the window containing the root node go to Data visualizer and select a data subset with a high degree of purity (if it is possible all data should belong to the same class = the points have the same color):

- from Select Instance choose Rectangle,
- extend the rectangular frame over the data to be selected
- click on Submit

The effect of these steps is that the root node is split in two children nodes: one corresponding to the selected dataset and another one to the rest of data (the tree can be visualized in the panel Tree visualizer).

c) Select another subset (out of the remaining data) and continue the process until the entire dataset is covered.

4. Rule based classifiers

The classification rules have the following structure: IF <antecedent> THEN <class label>, where the antecedent part contains conditions concerning the attribute values (the individual conditions are combined using conjunction and/or disjunction operators)

Construction of the classification rules set. There are two main variants:

- Starting from a decision tree: from each branch one construct a classification rule (the conditions in the antecendent are obtained by using the conjunction operator)
- Directly from the data: by using the so-called covering algorithms

Weka implementation:

- Starting from decision trees: M5rules, PART
- Covering algorithms: OneR, JRip, PRISM

Exercise 5: Analysis of algorithm for extractin classification rules

a) Open the file “weather.nominal.arff” and apply successively (using Weka Explorer) the following algorithms: OneR, JRip, PRISM, PART

b) Identify the best rule based classifier for the dataset “weather.nominal.arff”