Data Mining

Lab 4: Data Classification (II)

Summary:

- Probabilistic models
- Neural networks
- Support vector machines

1. Probabilistic models

They are based on the estimation of tha probability that a data belongs to a class ($P(C_k|(a_1,a_2,\ldots,a_n))$). During the training process are computed, based on the training set, some probabilities which are later used to estimate the probability for a given data to belong to each of the classes. The class having the maximal probability will be considered the class to which the data belongs.

Examples:

- Naïve Bayes: it is based on the simplifying assumption that all attributes are independent (consequently the probability to observe a data instance is the product of the probabilities to observe each of the attributes, i.e. $P(a_1,a_2,\ldots,a_n)=P(a_1)P(a_2)\ldots P(a_n)$.
- Bayesian Network (Belief network): it allows to describe the relationships between attributes by using a directed acyclic graph.

Exercise 1:

a) Open the file "weather.nominal.arff". Apply the classifiers Naïve Bayes and Bayes Net
b) Use the Bayes Network Editor (from Weka GUI Choser -> Tools) to construct a Bayesian network. Main steps:
   - Select the dataset: Tools->Set Data
   - Place the nodes (each attribute, including that corresponding to the class will have an associated node) by using Edit->Add node
   - Specify the relationships by using Edit->Add arc (for instance one can specify an arc from the class node to the nodes corresponding to the attributes)
   - Network training (Tools->Learn network (this corresponds to the learning of the structure), Tools->Learn CPT (this corresponds to the computation of the tables with probabilities))
   - Usage of the Bayes network classifier: select the values which correspond to the instance to be classified (by right clicking on each node associated to an attribute and by using Set evidence); the decision is based on the values of the probabilities associated to the class.

Remark: details on using the Weka implementation of Bayesian networks can be find at http://www.cs.waikato.ac.nz/~remco/weka_bn/

2. Neural networks

The main steps in designing a neural network for classification are:

- *The choice of the architecture*. In the case of multi-layer perceptrons one have to choose:
  - The number of input units = the number of attributes
  - The number of hidden layers and units – it depends on the complexity of the problem (there is no unique solution)
  - Number of output units:
    - Binary classification: one unit (the result is interpreted by using a threshold value, e.g. if the output value is larger than 0.5 then class 1 otherwise class 2) or two units (the unit which produces the larger value will indicate the class)
- *The choice of the activation functions*. For the neural networks to be trained using the backpropagation algorithm the activation functions should be differentiable, thus the typical choices are:
  - The hidden units have sigmoidal activation functions (logistic or tanh function).
  - The output functions have sigmoidal or linear functions.
- The choice of the training algorithm and of its parameters. In the case of the backpropagation algorithm the typical parameters (also used in Weka) are:
  - Number of training epochs
  - Learning rate
  - Coefficient of the momentum term

Weka implementation: MultilayerPerceptron = feedforward neural network trained using Backpropagation (the variant with momentum term); the network architecture has by default one hidden layer but the number of layers (and of units per each layer) can be changed.

Exercise 2: Analysis of the network architecture on the classification performance

a) Open the file "breast-w.arff"
b) Train a neural network using the default settings (one hidden layer with K=(nr attributes+nr classes)/2 units)
c) Compare the performance of the classification (accuracy values) for the following values of the Hidden Layers (H) parameter:
    a. 'a' (K=(nr attributes+nr classes)/2)
    b. 'i' (K=nr attributes)
    c. 'o' (K=nr classes)
    d. 't' (K = nr attributes+nr classes)
    e. 4,2 (two hidden layers with 4 and 2 units, respectively)
d) Analyze the influence of the learning rate and of the coefficient of the "momentum" term.
e) Compare the results obtained by using the MultilayerPerceptron with that obtained by using a RBF network – radial basis function network (functions->RBF Network)

Remark. By activating the option GUI from Multilayer Perceptron one can visualize the network architecture.

3. Classification using Support Vector Machines (SVM)

The Weka implementation of SVM uses a fast algorithm for solving the optimization problem involved in the estimation of the weights corresponding to support vectors (Sequential Minimal Optimization). The standard SVM is designed for binary classification. In the case of the classification in M classes the classification problem is transformed in several binary classification problems.

Exercise 3:

a) Open in Weka the file “breast-w.arff” and use SVM (functions->SMO); analyze the influence of the kernel function on the classification accuracy (variants: PolyKernel, NormalizedPolyKernel, RBFKernel)

b) Open in Weka the file “arrhythmia.arff” and apply the SVM classifier; try to solve the same problem using a neural network (multilayer perceptron or RBF network).