Sensorgestützte intelligente Umgebungen: Introduction to Machine Learning

Due on 11 June 2010

Oscar Martinez Mozos – Email: ommozos@unizar.es

Exercise 1

1. Download and install the framework for machine learning WEKA from:

   `http://www.cs.waikato.ac.nz/ml/weka/`

2. Have a look at the tutorial:

   `http://weka.wikispaces.com/Primer`

   Concentrate only on the information referring "supervised classifiers" which are the classifiers we saw in the lecture.

Exercise 2

1. Have a look at the section "Dataset" of the tutorial. Here you will find an explanation about how to prepare the data files containing the training data.
2. Download and extract the following file:

   `http://www.informatik.uni-freiburg.de/~omartine/local/examples.arff`

   If you remember from the lecture we are actually trying to classify segments composed of a set points in 2D. The file `examples.arff` contains the features corresponding to each segment together with its classification. The actual label of each segment is define in the attribute `person` which is 1 for segments corresponding to people and -1 for segments corresponding to other objects in the room.

   The features correspond to the ones in the paper:

   > K. O. Arras, O. M. Mozos, and W. Burgard,
   > Using boosted features for the detection of people in 2D range data,
   > in Proceedings of the IEEE International Conference on Robotics and Automation (ICRA),
   > pp. 3402-3407, 2007.

   That you can download here:

   `http://www.informatik.uni-freiburg.de/~omartine/publications/arras2007icra.pdf`

3. In the "Explorer" load the `examples.arff` file in the "Prepocess" tab. Move to the "Classify" tab. Divide the data into training (70%) and test (30%) using "Percentage split". Then choose the "bayes/NaiveBayes" classifier and apply it to the data. Using the output of WEKA write a confusion matrix for the results using the terms true positives (TP), true negatives (TN), false positives (FP), false negatives (FN). Write the results in percentages.
4. Change the classifier and choose the "meta/Adaboost.M1". Write the confusion matrix of the results.
5. Now change the classifier to "rules/ZeroR". This is the simplest classifier. Have a look at the percentage of correct classified instances and incorrect classified instances. Compare these results with the ones from "bayes/NaiveBayes". Which classifier is better according to this values?

   Despite the results, the "rules/ZeroR" classifier is bad, and you cannot trust it. Why? What is this classifier actually doing? (Hint: have a look a its confusion matrix).