

Praktikum zu Moderne Methoden der Datenanalyse

Exercise 10: Data Mining Cup: NeuroBayes[®]

This is the last part of the Data Mining Cup exercise. In exercise 9 we used a Neural Network for the classification of customers. Everybody who tried to train a Neural Network for this task could realize that it is not an easy task. There is the possibility of overtraining and the problem of not finding the global minimum (keep in mind, that training a Neural Network means, minimizing a loss function in multidimensional space). In this exercise we will continue with Neural Networks and the presentation of techniques, which can help to train a Neural Network.

For this purpose we will use the NeuroBayes package which was also used successfully by many diploma students in the Data Mining Cups of last years. Type the following command to set up the package:

```
source /home/staff/tkuhr/Praktikum/Exercise10/setup_neurobayes.sh
```

You can find the documentation of the package at `$NEUROBAYES/doc/`.

- **Exercise 10.1:**

The NeuroBayes package can automatically generate code for the training of a Neural Network. The parameters for this task have to be provided in a steering file. The steering file we will use is `/home/staff/tkuhr/Praktikum/Exercise10/dmc.codegen`. Copy it to your working directory and open it in an editor. Select your favourite set of variables you want to use as network input by removing the `#` in front of them. Then perform the network training by typing

```
teacher.sh dmc.codegen
```

This will create a directory with prefix `teacher_` followed by date and time. Change to this directory, look at the plots in `analysis*.gz` and try to understand them. The root file `outputExpert_*.root` contains a tree of the input variables and the Neural Network output for each event.

- **Exercise 10.2:**

Select a value for a cut on the Neural Network output and calculate a score. You can use the `MakeClass` method of the tree in `output.root` for this task like in exercise 7.2.

- **Exercise 10.3:**

A common problem of Neural Networks is the high number of degrees of freedom due to many connections so that statistical noise can be stored. In the worst case, the network can memorize individual events which were used for the training. In order to avoid to train on statistical fluctuations, one can regularize the network (remove insignificant connections).

To switch on regularization, comment out the line containing `REG 'OFF'` in `dmc.codegen`. Run the teacher and the expert once more and compare the performance with the Neural Network without regularization.

- **Exercise 10.4:**

Another technique, which can help in the Neural Network training is preprocessing input variables in a way, which makes it easier to learn correlations between them. The `NeuroBayes` package offers many possibilities for this task. In this exercise we want to transform the input variables to a Gaussian with mean zero and width 1 and we want to decorrelate them. To achieve this, set the `PREPRO` option in the steering file to 12. Run the teacher and the expert and check the performance of the network.

- **Exercise 10.5:**

Up to now we didn't include the real cost matrix to our training, which means, that the Neural Network didn't assign different weights to the different kind of errors as it is done when calculating a score. The `NeuroBayes` package allows using different cost functions. Enable the `COST_FUNCTION` option in `dmc.codegen` to train a Neural Network with the appropriate cost matrix.

- **Exercise 10.6:**

In the training of a Neural Network with low statistics, overtraining is rather probable. One way how to check, that the network is not overtrained is to split the training sample to N subsamples, training a Neural Network N times with $N - 1$ subsamples and applying the result to the subsample which was not used in the training. With this procedure, called cross validation, one gets N classified samples, where the classified events were not used in the training. Run a cross validation by enabling the `CROSS_VALIDATION` option in the steering file. The result can be found in the file `nbOutputPlots_crossval.pdf` in the created subdirectory.

- **Exercise 10.7:**

Try to improve your network by changing your selection of input variables and/or by using individual preprocessing of them (see manual). This is your chance to play with the Network and make your best classification.

- **Exercise 10.8:**

Change to the subdirectory of your best network and use the file `outputExpertClass_*.root` to produce a text file with your prediction for the classification dataset like in exercise 7.3. Give the text file to a tutor to obtain a score.

- **Exercise 10.9:**

Prepare a short (maximum 10 minutes) presentation, in which you explain at least one of your solutions to your fellow students. You can use the blackboard or prepare slides in electronic format (e.g. with OpenOffice). A Beamer will be available for the presentations.

The presentations will take place in the last session and are a *Scheinkriterium*. If you want to use slides in electronic format, save it as a pdf file with name *YourFirstName_YourLastName.pdf*. Send this file via email not later than Wednesday 11.02.09 to `Thomas.Kuhr@ekp.uni-karlsruhe.de` or put it at a world readable location (e.g. your home directory on the computer pool with access rights set by `chmod go+rx ~`) and send only the location. Include your matriculation number and your full name in the email.