Machine Learning. Homework 3

Skoltech 2015

Deadline: **24 November 2015, 16:00**.

Description

This assignment is about classification. You need to predict whether a person receives a credit from the bank based on some information about this person.

You are provided with three files: `train.data`, `test.data` and `test.lab`. The first one is the training set and contains both the features and labels (last column). The last two files are the testing set: `test.data` contains the features, `test.lab` contains the corresponding labels. You should not use the testing labels until the very end!

You have to complete this assignment using IPython Notebook and provide your `.ipynb` file a result. You have to explain every code cell in your `.ipynb` file with markdown cells. If your experiments lack an analysis or corresponding conclusions, your grade will be reduced!

Task

1. Load the data. How many samples/features does it have? Determine the type of each feature (continuous or categorical). What are possible values of each feature? Do the data contain any missing values? How much data are missing? How many classes are in this problem? Can you say that the classes are more or less balanced?

2. Since the data contain missing values, you have to do something about it. For the training data, you could simply remove the samples with missing values. However, you cannot use this approach for the testing data because you have to predict the label for *each* testing sample, even if it contains missing values. Therefore you need to perform *imputation* for the testing test. (You may do imputation for the training set as well.)

3. Prepare the data for a classification algorithm. You need to construct the design matrix X and the vector of labels y that contain only numerical data. Do not forget to use *one-hot encoding* for categorical features!

4. Take the decision tree (DT) classifier from the `sklearn` module with default parameters. Estimate the accuracy of this classifier using *cross validation*. Use the fraction of correctly predicted labels as an accuracy metric. Try to vary some parameters of the classifier. How does the accuracy change?

5. Write the code that tunes the parameters of DT automatically using *grid search*. What are the best parameters? What is the resulting accuracy? How much time does this grid search take?

6. Run the SVM classifier from the `sklearn` module with default parameters. What is its accuracy? Try to normalize the features and see what changes.

7. Find the best parameters for SVM using grid search. What accuracy does it achieve?

8. Consider four classification algorithms: DT, SVM, Logistic Regression (LR) and KNN. Compare these algorithms in terms of training time (grid search) and resulting accuracy.

9. Choose any of these four algorithms, fit it on the training set and predict the labels for the testing set. If you used any feature transformation (e.g. normalization), do not forget to apply the same transformation to the testing set features beforehand!

10. Compare the actual testing labels with the ones predicted by your algorithm. What is the accuracy? Compare it with the cross validation accuracy you obtained previously.

11. **Bonus** (extra points): Try to increase the accuracy of the chosen algorithm using data preprocessing. You may try to use a different strategy for dealing with missing values, remove outliers and/or make up some (nonlinear) feature transformation. It may help to visualize your data (e.g. using a scatter plot matrix).