Advanced Topics in Machine Learning Programming Task

Jannis Becke, Christian Lausberger and Maik Riestock
Otto von Guericke University Magdeburg - Data and Knowledge Engineering Group

1. APPROACH

Our general approach to solve the task was as follows:

- analysed the data set and its various attributes
- cleaned the data (remove attributes, missing values)
- used different libraries which provides different machine learning algorithms
- evaluated the quality of selected algorithms with accuracy metric and cross-validation

In order to implement the classification algorithms we used different libraries:

Weka[1] - collection of machine learning algorithms for data mining tasks which provides a lot of basic functionality like data storage, filtering, visualisation, and additionally many classifier and clustering algorithms

Collective Classification[2] - provides semi-supervised algorithms and extends the Weka library

LIBSVM[3] - is an integrated software for support vector classification and provides a implementation of SVM classifier for Weka

For our solution we selected these four classifiers:

- NaiveBayes
- Decision Table
- SVM
- YATSI

2. DATABASE

source what it is about what the attributes are about

3. PREPROCESSING

3.1 attributes

remove of attributes

[1] http://www.cs.waikato.ac.nz/ml/weka/
[2] http://www.cms.waikato.ac.nz/~fracpete/projects/collective-classification/
[3] http://www.csie.ntu.edu.tw/~cjlin/libsvm/

This documentation was created in the context of the course Advanced Topics in Machine Learning summer term 2014/15. This course was held by: Prof. Dr. Andreas Nurnberger, M.Sc. Tatiana Gossen; Research group Data and Knowledge Engineering Group, Otto-von-Guericke-University of Magdeburg, Germany.

3.2 instances

handling of missing values
remove of instances?

3.3 conclusion

4. CLASSIFICATION

The attribute to be classified from the data set is *Readmitted* and exist in three characteristics, *No*, *<30* and *>30*. These values denote if the patient was readmitted in more or less than 30 days, and No for no record of readmission. [**?**]

The distribution of the values in these three instances differs greatly from 54864 instances in *No* and 35545 instances in *<30* to 11357 instances in *>30*. These unequal distribution led to a problem that is addressed in section One Class Problem.

When selecting classifiers The following criteria were important: performance, type specification from task description and feasibility. In addition, many classifiers omitted because they could not solve a 3 classes problem.

Used classification algorithms:

- NaiveBayes, a supervised algorithm
- DecisionTable, a supervised algorithm
- LibSVM, a algorithm based on SVM from
- YATSI, a semi-supervised algorithm

For the evaluation of the trained classifiers, we used the two common evaluation methods:
Percentage Split - used with a 50%/50% split
Cross-Validation - used with 10 folds

By means of the presented classification and evaluation methods, we achieved the following results:

Table I. Classifier result for each evaluation method

Algorithm	Evaluation	Correctly Classified
NaiveBayes	Percentage Split	57.32%
NaiveBayes	Cross-Validation	57.14%
DecisionTable	Cross-Validation	57.75%
SVM	Percentage Split	64.87%
SVM	Cross-Validation	-2%
YATSI	Percentage Split	53.22%
YATSI	Cross-Validation	53.30%

explain the results
kaum unterschied zwischen den evaluierungs methoden

During the review of the results we noticed, that some classification algorithms put all test instances in one class. The problem

arises from it that the number of instances between the classes varies greatly. To counteract this problem, we tried two approaches presented in section One Class Problem.

5. ONE CLASS PROBLEM

The results some classification algorithms put all test instances in one class. The problem arises from it that the number of instances between the classes varies greatly. To counteract this problem, we tried two approaches presented in the following.

5.1 Even Classes

In this approach, we have adjusted the number of instances of each class. We have first determines the class with least number of instances. Then we have the other classes cleared by random selection so much instances until the classes were equalized.

With this newly created data set we have carried out our classifications again. Surprisingly, now showed the same problem. Except that now the characteristics, >30, with the former fewest instances, get all test instances. Because of these poor results, we did not pursue this approach. This facilitates the task of classification.

5.2 2 Class Problem

In this approach, we converted the 3-class problem to a 2-class problem. We have the attribute *Readmitted* changed so that the smaller characteristics, <30 and >30, were summarized. That reduced the expressiveness of the attribute. Now you can only make a statement about if the patient was readmitted or not.

With this newly created data set we have carried out our classifications again. We achieved better results than with the original data set. As this is not compatible with the tasks description, we did not pursue this approach.

6. PROGRAM EXECUTION

In order to execute our program you need to open a console and enter the following command:

```
java -jar <PATH TO JAR>\ATMLProgramming.jar
```

The database file should be saved in the same directory as the jar file. After executing the jar file the menu will guide you through the whole program and ask for some parameters before the classification starts.

7. CONCLUSION

List of Tables

APPENDIX

A. TASK DESCRIPTION

Use the Diabetes 130-US hospitals for years 1999-2008 Data Set. This dataset contains records with 55 features for more than 100000 patients. 55 features include information about the diabetic encounters, including demographics, diagnoses, diabetic medications, number of visits in the year preceding the encounter, and payer information. Note that even though some attributes are codified using numeric values, they are nominal (not numeric) attributes. Browse additional information about the attributes in the paper cited on the dataset website. The goal of this assignment is to find suitable methods in the area of machine learning to determine the readmission attribute of a patient and to estimate the quality of selected approaches.

In order to achieve this goal, the following tasks have to be completed:

- analyse the data set and its various attributes
- clean the data (e.g. missing values)
- select an appropriate subset of the attributes and explain your choice
- use different suitable machine learning algorithms (either implement them, or use existing libraries, e.g. Weka)
- determine the quality of your model (e.g. through cross-validation, log loss3, confusion matrix)
- compare your results between different algorithms

Specifically, different suitable machine learning algorithms should include:

- at least two classification algorithms, one of which, SVM, you learn during the course
- at least one algorithm for semi-supervised classification. Use a part of the dataset as unlabeled data for learning (omit the label)