Ensemble Methods Project

M2 Computer Science MALIA

Guillaume Metzler
Institut de Communication (ICOM)
Université de Lyon, Université Lumière Lyon 2
Laboratoire ERIC UR 3083, Lyon, France

guillaume.metzler@univ-lyon2.fr

Groups of two persons are allowed to do this project.

The work shall be submitted at the following mail adress

guillaume.metzler@univ-lyon2.fr

More precisely, the submitted work shall contain a python or a notebook which contains the code that has used to produce the results and report (taking the form of a pdf file) where you will present the data and used methods to get them.
This report shall focus on the comparison of the results and the analysis of the later.

1 Datasets

You can find several data sets at the following address

Github for the datasets.

For this work, you will have to use the 5 different datasets for which the name started as :

kaggle_source_cate_i, for i from 0 to 4.

For $i \in [\![0, 3]\!]$, you will find :

a train file which contains the features values of the training set.

a train_label file which contains the associated label.

a test file which contains the features values of the test set.

a test_label file which contains the associated label.

2 Work to perform

Methodology Your datasets can be divided into two groups according to a specific characteristic and maybe the procedure to used and/or the performance metrics you are going to use will be different.

For each of these groups of datasets, you will then need to compare the following algorithms :

- penalized logistic regression
- an ensemble method based on bagging
- a random forest method
- an ensemble method based on boosting
- an ensemble method based on stacking (learning the combination of voters)
- a gradient boosting method

What can be interesting also, is to test whether the combination is interesting. Thus, as a baseline, for instance, if you have decided to combine several linear SVM, to see the performance of a single linear SVM.

For the second group of data, it can also be interesting to test several sampling methods and/or cost-sensitive learning.

Code **The code that will be submitted shall provide the same results as the one presented in the report !**

What follows is a suggestion on how to write your report.

Introduction

You will rapidly present the context of your project and the main target and difficulties that are linked to your datasets.

Methodology

It mainly consists in presenting in few words and equations the way the methods are working. Saying differently, the way you will tackle the problem and used concepts.

You first introduce the notations that will be sued for the purpose.

You will then present the tools you will be using in the experimental part. We will start by talking about what we want to maximize, *i.e.* the performance measure by defining it before tackling the presentation of the algorithms (in a very short way).

For example, if you are testing an algorithm based on boosting, you briefly have to explain how it works. Don't hesitate to present the process in an abstract way, i.e. with mathematical notations and not just in words. Pseudo-code is also useful for summarizing the proposed approach.

If you are proposing several approaches for comparison purposes, you should take care to present the different approaches and justify why you are interested in them.

Note : it is not necessary to present all the algorithms used, but only those used to develop an "exotic" version.

Experiments

You will then draw up your experimental protocol, setting out the method(s) you have selected for the task in hand. This is generally composed of three parts.

Experimental Protocol

For each part, you will proceed as follows :

You will start by presenting the data you are using, usually in the form of a table that includes : name of the data set, sample size, size of the data set and any other information you consider relevant.

Briefly present the experiments you will be running, the different algorithms you will be testing, the range of hyper-parameters you will be using, and how these hyper-parameters are optimized (cross-validation in k-folds, simple validation or do you choose to fix them). What are your training/validation/test sets ?

The information you provide in this section should enable the reader to reproduce the results you present in the next subsection.

Results

Here, you will present and analyze the results obtained, using graphs and/or tables.
In addition to performance, you may also be interested in the speed of an algorithm.

The analysis should also highlight the advantages and disadvantages of the proposed methods.
This may involve using other performance measures/criteria to evaluate/comparison your algorithms.

As for the results, they can be presented in the form of a table similar to the one presented in Table 1.

Conclusion

The aim is to conclude your study. Review the proposed work and the main conclusions. It is also important to offer some perspectives on your work, based on the results obtained and the approach proposed. What method(s) not explored here could have been used to improve the results, and why do you think this is relevant given the experiments carried out ?

TABLE 1 – Mean test F1 over 10 iterations

Dataset	knn	svm_linear	svm_poly	svm_gauss
00.62% abalone20	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	**0.0** ± 0.0
01.39% abalone17	**13.7** ± 6.3	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0
02.36% yeast6	**44.0** ± 14.6	0.0 ± 0.0	38.9 ± 18.0	0.0 ± 0.0
03.31% wine4	**5.0** ± 5.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0
06.67% libras	81.1 ± 12.7	71.0 ± 10.0	**88.3** ± 7.9	75.6 ± 11.0
10.23% pageblocks	**84.9** ± 2.2	75.7 ± 2.9	82.3 ± 2.5	78.6 ± 2.5
10.98% yeast3	66.8 ± 5.9	65.6 ± 10.8	**76.4** ± 5.2	71.3 ± 8.2
13.60% abalone8	**22.7** ± 1.3	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0
14.29% segmentation	**85.1** ± 3.3	45.8 ± 3.7	64.1 ± 4.9	50.0 ± 4.6
22.73% hayes	14.8 ± 10.2	0.0 ± 0.0	**39.4** ± 19.9	0.0 ± 0.0
23.52% vehicle	88.1 ± 3.4	60.3 ± 5.3	**92.6** ± 1.7	56.7 ± 6.5
30.00% german	41.3 ± 6.4	5.3 ± 7.6	**56.9** ± 6.8	16.8 ± 6.6
32.71% glass	**70.2** ± 4.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0
33.15% wine	86.5 ± 4.3	89.6 ± 5.9	**90.1** ± 5.8	86.2 ± 6.1
34.90% pima	41.3 ± 6.6	5.0 ± 5.4	**44.7** ± 6.1	17.5 ± 4.8
35.90% iono	82.5 ± 4.1	88.9 ± 4.0	92.0 ± 3.4	**93.3** ± 3.4
37.50% autompg	82.4 ± 5.5	0.0 ± 0.0	**83.0** ± 3.4	0.0 ± 0.0
46.08% balance	95.3 ± 1.9	94.5 ± 1.2	**99.9** ± 0.3	97.3 ± 0.8
Mean	**55.9** ± 5.4	33.4 ± 3.2	52.7 ± 4.8	35.7 ± 3.0
Average Rank	2.00	3.72	1.72	2.56

3 Bonus

Try to apply one of the ensemble methods using the datasets as follows, in order to achieve the best results.

Training For the training procedure, you can use datasets containted in the following file :

- kaggle_source_cate_0_train.npy
- kaggle_source_cate_0_train_label.npy
- kaggle_target_cate_0_train.npy

Testing phase You will then evaluate the performance on both a *source* dataset and a *target* one :

- kaggle_source_cate_0_test.npy
- kaggle_source_cate_0_test_label.npy
- kaggle_target_cate_0_test.npy
- kaggle_target_cate_0_test_label.npy

The objective of this final section is to perform unsupervised domain adaptation. First, you will train a model in the classical way using the available information and assess the performance (separately) on both the source and target test data.

Your goal will then be to improve performance on the target test data using the labeled source data and the unlabeled target training data.