Machine Learning EXP2: Ensemble

Description

This assignment needs to implement some **ensemble learning algorithms** and test on a given dataset.

The dataset used in our experiment is reviews (in English) from Amazon, which contains more than 50,000 reviews with quality labels (high_quality: 1, low_quality: 0)

Task

Compare different ensemble learning algorithms with different base classifiers. Two ensemble learning algorithms are required (Bagging and AdaBoost.M1); and two base classifiers are required (SVM and Decision Tree). Thus, you should at least compare 4 combinations:

- Bagging + Decision Tree
- Bagging + SVM
- AdaBoost.M1 + Decision Tree
- AdaBoost.M1 + SVM

You should design, extract and select the features by yourself.

You are allowed to use existing classifier implementations in the experiment, but you need to implement the ensemble learning algorithms by yourself.

Optional Tasks

- Try other base classifier (such as K-NN, Naive Bayes...)
- Analyze the effect of different (kinds of) features
- Tune the parameters of ensemble learning algorithms, and analyse their effect on performance
- Any other methods you'd like to get higher score

Evaluation

We use AUC as evluation metric, which do not need assign label to each test case given a threshold. To calculate AUC, you only need to output $p \in [0, 1]$ representing the probablity that a test case is positive.

See the link above for more details about calculation.

File Description

- train.csv: training set
- test.csv: test set, without information about votes and label

You should generate results on each reviews in `test.csv`, where `reviewerID` and `asin` are assured to have been appeared in `train.csv`.

These files can be loaded with `pandas` using python.

```
import pandas as pd
train_df = pd.read_csv('train.csv', sep='\t')
```

Data Fields

- Id: identify test cases (only appear in test set)
- reviewerID: unique id of each reviwer
- asin: unique id of each item
- reviewText: content of review in English, without preprocessing
- overall: the rating user gives to item (from 1 to 5)
- votes_up: number of up votes to this review (only appear in training set)
- votes_all: number of total votes to this review (only appear in training set)
- label: 0 for low quality, and 1 for high quality (only appear in training set)

Reviews with votes_up / votes_all ≥ 0.9 are considered as high quality reviews. All the reviews are assured to have at least 5 votes_all.

Submission Format

Go here to participant the Kaggle competition. You can **submit up to 3 times per day**.

The submission file should be the results for each review in `test.csv` . Each row contains `Id` (corresponding to test set) and `Predicted` result (a double number), splited by comma. The file should be like this (header should be included):

```
Id,Predicted
0,0.9
1,0.45
2,0.78
...
```

Submission (learn.tsinghua)

Source code

With necessary comments. No restriction on programming languages, but make sure that TA can run your code easily.

README

A text file that briefly describes how to run your code and produce the reported results. Please also make sure your name, your student ID, your name on Kaggle and your contact information included.

Report

A pdf file that includes the following information:

- Your experimental design
- The experimental results: the results of 4 required combinations
- Performance of different methods on Kaggle's evaluation set and the rank on the leaderboard
- Your analysis and discussion. For example: why do the algorithms mentioned above perform differently or similarly on the dataset? What is the difference between Bagging and AdaBoost? Which combination is the best one and why?

Deadline & Other Information

DEADLINE: Thursday May 16 23:59, 2019 (UTC+8)

Upload the packed file (ZIP format is preferred) with your name and student number in filename to learn.tsinghua.edu.cn. Late submissions **WILL NOT BE ACCEPTED**.

Feel free to contact the TA for further information.

THUwangcy@gmail.com 17888802343

Some Toolkits

SVM

- Sklearn: https://scikit-learn.org/stable/modules/svm.html
- LibSVM: https://www.csie.ntu.edu.tw/~cjlin/libsvm/
- SVM-light: http://svmlight.joachims.org/

Decision Tree

- Sklearn: https://scikit-learn.org/stable/modules/tree.html
- C4.5: http://www.rulequest.com/Personal/
- C5.0: http://www.rulequest.com/see5-info.html

Other classifiers

- Sklearn provides many common classifiers in Python http://scikit-learn.org/stable/
- Weka: Data Mining Software in Java http://www.cs.waikato.ac.nz/ml/weka/
- Matlab also has lots of packages for machine learning

Please note that even if the package provides ensemble learning tools, you **SHOULD NOT** use them. The implementation of the ensemble learning algorithms (Bagging and AdaBoost.M1) must be done by yourself.