EXP2 report

唐静雯 2015010846

1. Experimental design

a. Svm + bagging
   The training size each time = 3000, and repeat 4 times in bagging.

b. DT + bagging
   The training size each time = 3000, and repeat 4 times in bagging.

c. Svm + adaboost M1
   The training size = 3000, and there are 4 loops.

d. DT + adaboost M1
   The training size = 3000, and there are 4 loops.

2. Experimental results

	RMSE	Leaderboard RMSE
Svm + bagging	0.6123546	0.55791
DT + bagging	0.655788	0.81839
Svm+adaboostM1	0.9165151	0.70539
DT+adaboostM1	0.8944272	1.25646

3. Kaggle's evaluation set

Algorithm + ensemble method, features minimum frequency, training set volume, loops

My test	RMSE
Svm+bagging,30,3000,3	0.5599776
Svm+bagging,40,3000,3	0.5645937
Svm+bagging,30,3000,4	0.6115207

Kaggle's test	Leaderboard RMSE
Svm + adaboost M1, 20,6000, 8	**0.81785**
Svm + adaboost M1, 20,8000, 3	0.65926
Svm + adaboost M1, 30,18000, 3	0.64440
Svm + bagging, 20,5000, 10	0.53403
Svm + bagging,30,8000,30	0.51657
Svm + bagging,30,15000,5	0.48453
DT + bagging, 20,5000, 30	0.8929

Now rank on leaderboard: 57

4. Analysis and discussion

(a. Why do the algorithms mentioned above perform differently or similarly on the dataset?

From table1 we can see that on average svm works better than Decision Tree under the same amount of features, training examples and number of loops.

For the bad performance of Decision Tree, the reason may be that the same value of a feature (like one word "真的" which is neutral is included in two reviews) can mean different meanings because it will depend on the word after it. So take the words like them as nodes in a decision tree will not make sense and make some mistakes or misleading the classification.

For svm, in general the existence of one particular word will not lead the word to a different branch. That is the value in one dimension will not expert so much influence on a word's general position in the high dimensional space. So the language attributes will not do much harm to the svm method to do the classification.

(b. What is the difference between Bagging and AdaBoost?

Training set

Bagging: Randomly selected samples, tends to be independent.

Boosting: Decided by the previous one, dependent (actually derived from the same set of samples but with different weighting.)

Prediction function

Bagging: no weights; easier to parallelize. Use majority voting to decide the final choice.

Boosting: weights grow exponentially; sequential implementation. Finally use weighting for each classifier based on their performance.

Performance

In practice, bagging almost always helps.

On average, boosting helps more than bagging, but it is also common for boosting to hurt performance like in this case. Maybe because there are uncertainty about reviews because of the language complicity which can be regarded as noise.

Bagging doesn't work so well with stable models. Boosting might still help.

Boosting might hurt performance on noisy datasets. Bagging doesn't have this problem.

(c. Which combination is the best one and why?

Svm + bagging is the best.

As stated in the previous questions, the review classification may be not suitable for decision tree method for some certain frequent words should not be regarded as nodes which has an impact on the final prediction. Boosting performs badly maybe because there will be some noises in the data set because of the complexity of language expression. However bagging will be less susceptible to these noises.