Boosting SVM Classifiers By Ensemble

Yan-Shi Dong
Shanghai Jiao Tong University
ysdong@126.com

Ke-Song Han
Motorola Labs, China Research Center
a18186@motorola.com

ABSTRACT

By far, the support vector machines (SVM) achieve the state-of-the-art performance for the text classification (TC) tasks. Due to the complexity of the TC problems, it becomes a challenge to systematically develop classifiers with better performance. We try to attack this problem by ensemble methods, which are often used for boosting weak classifiers, such as decision tree, neural networks, etc., and whether they are effective for strong classifiers is not clear.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval; I.5.2 [**Pattern Recognition**]: Design Methodology

General Terms

Algorithms, Experimentation

Keywords

Classifier design and evaluation, Information filtering, Machine learning, Neural nets, Text processing

1. INTRODUCTION

Text classification (TC) has become one of the key techniques for handling and organizing online data. The SVM comes out as one of the most effective text classifiers [3]. While SVM already performs so well, the next question comes forth naturally: can we develop a systematic ways to significantly boost a well-performed classifier, like SVM, for TC tasks further more?

Ensemble of multiple classifiers, i.e. combining the outputs of several base classifiers to form an integrated output, has become an effective classification method for many domains, but whether it is beneficial for TC remains open. Besides, many potential ensemble methods have not been explored so far. Moreover, most of the widely studied ensembles use unstable classifiers or weak classifiers, such as decision trees, neural networks, etc. as base classifiers, but few are reported about the effectiveness of the ensembles of strong classifiers, like SVMs. Hence, thorough evaluations of many potentially effective ensemble methods for TC are worth being carried out.

2. ENSEMBLES OF CLASSIFIERS

Most of the ensembles of classifiers can be decomposed into two kinds of components. The first component is to create base classifiers with necessary accuracy and diversity. We adopt the data partitioning method to achieve this goal. This method applies a homogenous training algorithm with the same parameter settings on the different subsets of the training dataset. In this paper, five types of procedures for partitioning data are evaluated: bagging, boosting, disjunct partitioning, fold partitioning and cluster partitioning. The disjunct partitioning means to randomly split the training dataset into *k* equal-sized non-overlapping sets, each taken as a subset for training a base classifier. The fold partitioning splits the training dataset as the disjunct partitioning does, however combines all but *i*-th sets into a subset. The clustering partitioning uses a clustering algorithm to agglomerate the training dataset into several non-overlapping clusters, each taken as a subset [1]. Besides, we incorporate biased sampling into these partitioning methods, i.e. only divide the negative examples into several subsets at first, leaving aside the positive examples, and then merge all of the positive examples with each subset of negative examples to create the final subsets for training.

The second component of ensemble of classifier is to integrate all of the outputs of base classifiers into a numeric value as the final output. There are two type of methods for this task. The simple type is to weighted average the outputs of base classifiers, where the weights could be uniform, e.g. bagging, or predetermined. This type of combining methods avoids any learning procedures. The more complex type is to introduce a stacking classifier to learn the optimal mappings from the individual outputs of base classifiers to a final output. The stacking classifier can be in any forms. In this paper we used neural networks as the stacking classifiers because of their theoretical ability to learning any complex functions. Stacking classifier must be trained using stacking datasets, which is formed from the outputs of its base classifiers on the examples by *k*-folding cross-validation.

Besides, optimal subsets of the base classifiers in an ensemble can be selected for stacking, and the ensemble on an optimal subset might perform better than that on all of the base classifiers [2]. Many methods of optimal subset selection exist, but for simplicity, here we only selected several best-performed base classifiers as the optimal subset. The performance of the base classifiers are also estimated by cross-validation.

3. EXPERIEMENTAL SETTINGS

We compared all of the classifiers in an identical framework. We chose words as the representational units, without any further processing. Linear kernels, cosine normalized TFIDF term weighting and the tradeoff parameter C of 100 is used for the SVM classifiers. We adopt the simplest settings in the neural networks, i.e. the number of input units is equal to the number of base classifiers, with an output unit and zero hidden units. The number of partitions in the ensembles of classifiers is five wherever necessary. The k-means clustering algorithm is used in the cluster partitioning. Besides, we apply Scut thresholding [4] to estimate the optimal thresholds of the classifiers. The macro F_1 measure is chosen as the criteria for evaluating the performance of classifiers as well as for thresholding. Each classifier was trained

WWW 2005, May 10-14, 2005, Chiba, Japan.
ACM 1-59593-051-5/05/0005.

and tested for eight times, and their evaluation results were averaged.

A well-accepted benchmark collection, Reuters-21578 corpus, was adopted as one of the evaluation dataset. The ModApte split is used to divide the corpus into two datasets for training and testing. The ten most frequent topics were chosen as the target categories. We select all of the words with document frequency in the training dataset larger than or equal to five as the features. Another collection is the Usenet articles collected by Lang from 20 different newsgroups, called 20-Newsgroup collection here. We chose the chronologically lattermost 500 documents in each category as testing examples, and the remaining documents as training examples. We select all of the words occurred in the training dataset as the features.

4. EVALUATION RESULTS

Table 1. Overall comparisons of various ensemble classifiers.

Classifier	Reuters-21578		20-Newsgroup
	Macro F1	**Micro F1**	**Macro F1**
svm	82.88	91.06	68.40
aver-part-unbias	85.82	92.28	71.02
aver-part-bias	85.40	92.25	65.05
nn-part-unbias	**86.02**	92.34	**75.43**
nn-part-bias	85.87	**92.37**	69.60
aver-fold-unbias	84.71	91.98	66.48
aver-fold-bias	84.03	91.63	65.38
nn-fold-unbias	84.81	91.91	72.40
nn-fold-bias	84.85	91.97	68.91
aver-bag-unbias	84.16	91.72	59.91
aver-bag-bias	83.98	91.64	NA
aver-boost-unbias	69.04	79.39	37.08
aver-boost-bias	79.72	89.72	NA
nn-cluster-bias	78.84	88.49	68.34

Notion: *Nn* and *aver* represents using neural networks and uniform averaging as stacking classifiers, respectively. *Part*, *fold*, *clust*, *bag* and *boost* represents disjunct, fold and clustering data partitioning, as well as bagging and boosting, respectively. *Bias* and *unbias* represents biased and unbiased sampling, respectively.

Table 1 shows the overall evaluation results of various classifiers on the Reuters-21578 and 20-Newsgroup. From Table 1, the following conclusions could be drawn:

- Except boosting, biased data partitioning ensemble can not achieve better performance than unbiased methods. In fact, on 20-Newsgroup, most of the unbiased ensemble outperform the corresponding biased ones.
- Stacking by neural networks is more effective than by uniform averaging.
- Boosting can not improve the performance of SVMs, and in particular, unbiased boosting performs the worst. We guess the reason of this phenominon is that SVMs are kind of strong classifiers.
- The cluster partitioning ensemble also can not improve the performance of SVMs.
- The unbiased disjunct partitioning ensemble significantly achieves best performance among all of the classifiers, including the single SVM. The advantage of fold data partitioning ensemble over the SVM is less obvious.

Now we further investigate the effectiveness of ensemble and best base classifier selection. Figure 1 shows four curves, respectively representing the macro F1 measures of four types of ensemble classifiers on Reuters-21578 collection when the selected number of the best base classifiers, *n*, are changed. From Figure 1 we can observe that:

- The performance of the disjunct partitioning ensemble classifiers saliently and monotonously degrades when *n* decreases, and turned out near or below the baseline, the F_1 measure of the SVM, when the *n* decreases to one. This further demonstrates the effects of ensemble—the ensemble of classifiers significant out-performs any of its base classifiers. Every base classifier contributes more or less for the final performance of a classifier ensemble, and none of single base classifier dominates.
- The above-mentioned trends cannot be observed for the fold partitioning ensembles. The curves become relatively flat, but always higher than the baseline.

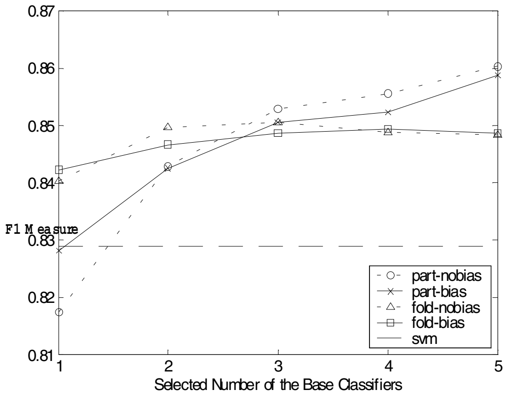

Figure 1. The F1 measures of ensemble classifiers containing different number of best base classifiers on Reuters-21578

5. Conclusions

We have experimentally compared five types of data partitioning ensemble of SVMs on two well-accepted benchmark collections, i.e. Reuters-21578 and 20-Newsgroup, and found that disjunct partitioning ensembles of SVMs with stacking performed best and consistently outperformed the single SVM. We also found bagging and cluster partitioning ensembles are not effective to combine strong classifiers like SVM, and boosting always achieves worse results on all of the collections.

6. References

[1] Chang, Y. I. Boosting SVM classifiers with logistic regression. See www.stat.sinica.edu.tw/library/c_tec_rep/2003-03.pdf, 2003.

[2] Dzeroski, S., and Zenko, B. Is combining classifiers better than selecting the best one? In Proc. 9th ICML, Morgan Kaufmann, San Francisco, CA, USA, 123-130, 2002.

[3] Joachims, T. Learning to Classify Text Using Support Vector Machines: Methods, Theory and Algorithms, Kluwer Academic Publishers, Norwell, MA, USA, 2002.

[4] Yang, Y. A study on thresholding strategies for text categorization. In Proc. 24th ACM SIGIR, ACM Press, New York, NY, USA, 137-145, 2001.