Ensemble Pruning for Text Categorization Based on Data Partitioning

Cagri Toraman and Fazli Can

Bilkent Information Retrieval Group,
Computer Engineering Department,
Bilkent University, 06800, Ankara, Turkey
{ctoraman,canf}@cs.bilkent.edu.tr

Abstract. Ensemble methods can improve the effectiveness in text categorization. Due to computation cost of ensemble approaches there is a need for pruning ensembles. In this work we study ensemble pruning based on data partitioning. We use a ranked-based pruning approach. For this purpose base classifiers are ranked and pruned according to their accuracies in a separate validation set. We employ four data partitioning methods with four machine learning categorization algorithms. We mainly aim to examine ensemble pruning in text categorization. We conduct experiments on two text collections: Reuters-21578 and BilCat-TRT. We show that we can prune 90% of ensemble members with almost no decrease in accuracy. We demonstrate that it is possible to increase accuracy of traditional ensembling with ensemble pruning.

Keywords: Data partitioning, ensemble pruning, text categorization.

1 Introduction

Ensemble of classifiers are known to perform better than individual classifiers when they are accurate and diverse [5], [19]. In text categorization, they are proven to perform better in some cases [6]. However, they are not efficient due to computational workload. For instance, in news portals, it is a burden to train a new ensemble model or test new documents. Various ensemble selection methods are proposed to overcome this problem [3]. The main idea is to increase the efficiency by reducing the size of ensemble without hurting the effectiveness. Besides, it can increase the effectiveness if selected classifiers are more accurate and diverse than base classifiers. In this work, we study these two aspects of ensemble selection by giving the accuracy (i.e. effectiveness) results [16].

Ensemble selection mainly consists of three stages: constructing base classifiers (ensemble members), selecting target classifiers among base classifiers, and combining their predictions. Base classifiers are constructed homogeneously or heterogeneously. Homogeneous classifiers are trained by the same algorithm and constructed by data partitioning methods in which training documents are manipulated [5], [6]. Heteregeneous classifiers are usually created by training different algorithms on the training set [3]. There are also mixed constructions in

M.V.M. Salem et al. (Eds.): AIRS 2011, LNCS 7097, pp. 352–361, 2011.

which data is partitioned and different algorithms are applied separately. There are various ensemble selection approaches [17]. In general, they search for an optimal subset of ensemble members. Searching evaluation is done with a validation (hillclimbing or hold-out) set, which can be used either in training or as a separate part of training set. Lastly, ensemble predictions are combined by simple/weighted voting, mixture of experts or stacking [17]. Voting is the most popular approach. It combines predictions of ensemble based on sum, production or other rules. It is called weighted when each prediction is multiplied by a coefficient.

Construction of base classifiers, training them and getting predictions from each of them require a considerable amount of time in text categorization when there are huge numbers of text documents. This becomes crucial when text documents become longer as experienced in news portals. There is a need for pruning as many base classifiers as possible. Therefore, in this study, we examine ensemble pruning in text categorization by applying different data partitioning methods for construction of base classifiers and popular classification algorithms to train them. We select a simple ranked-based ensemble pruning method in which base classifiers are ranked (ordered) according to their accuracy in a separate validation set and then pruned pre-defined amounts. We choose to use weighted voting to combine predictions of pruned ensemble.

Our answers to the following questions are the contributions of this study:

1. How much data can we prune without hurting the effectiveness using data partitioning?
2. Which partitioning and categorization methods are more suitable for ensemble pruning in the text categorization domain?
3. How do English and Turkish differ in ensemble pruning?
4. Can we increase effectiveness with ensemble pruning in the text categorization domain and which combination of partitioning method and categorization algorithm gives the highest accuracy?

The rest of the paper is organized as follows. Section 2 gives the related work on ensemble selection. Section 3 explains the experimental design and the datasets used in our study. Section 4 gives the experimental results. Finally, Section 5 concludes the paper.

2 Related Work

There are several ensemble selection studies. Tsoumakas et al. [17] give a taxonomy and short review on ensemble selection. Their taxonomy divides ensemble selection methods into search-based, clustering-based, ranked-based, and other methods. Search-based methods apply greedy search algorithms (forward or backward) to get the optimal ensemble. Clustering-based methods employ a clustering algorithm and then prune clusters. Ranked-based methods rank ensemble members once, and then prune a pre-defined amount of members. Our

approach also uses a ranked-based selection approach that examines different pruning levels.

Margineantu and Dietterich [11] study search-based ensemble pruning considering memory requirements. Classifiers constructed by the AdaBoost algorithm are pruned according to five different measures for greedy search based on accuracy or diversity. Their results show that it is possible to prune 60-80% ensemble members in some domains with good effectiveness performance.

Prodromidis et al. [14] define pre-pruning and post-pruning for ensemble selection in fraud detection domain using meta-learning. In our study, their pre-pruning corresponds to forward greedy search and post-pruning means backward greedy search. They produce their base classifiers in a mixed way such that they divide the train data into data partitions by time divisions and then apply different classification algorithms including decision trees to these partitions. They get up to 90% pruning with 60-80% of the original performance.

Caruana et al. [3] employ forward greedy search with heteregeneous ensembles on binary machine learning problems. They show that their selection approach outperforms traditional ensembling methods such as bagging and boosting. Caruana et al. [2] then examine some unexplored aspects of ensemble selection. They indicate that increasing validation set size improves performance. They also show that pruning up to 80-90% ensemble members rarely hurts the performance.

Martínez-Muñoz and Suárez [12] examine search-based ensemble pruning with bagging. They use CART trees and three different measures for forward greedy search. They show that 80% members can be removed with Margin Distance Minimization (MDM). Hernández-lobato et al. [7] study search-based ensemble pruning with bagging on regression problems. They decide to use 20% of ensemble members by looking regression errors. Martínez-Muñoz and Suárez [13] use training error defined in boosting in order to use in greedy search of ensemble pruning. Results are similar with the work by Hernández-lobato et al. [7].

In a recent work, Lu et al. [10] introduce ensemble selection by ordering according to a heuristic measure based on accuracy and diversity. Similar to our study, they prune the ordered (ranked) ensemble members using a pre-defined number of ensemble sizes. They compare their results with bagging and the approach used by Martínez-Muñoz and Suárez [12]. Their method usually outperforms the others when 15% and 30% of ensemble members are selected.

Our study is different from the above studies in terms of the production method of ensemble members, the way of ensemble selection, and the domain to which ensemble selection applied. We introduce a novel approach that examines data partitioning ensembles in ensemble selection. We also examine different classification algorithms that are popular in text categorization for ensemble selection. Our ensemble selection method is also simple such that we do not use greedy search or a genetic algorithm.

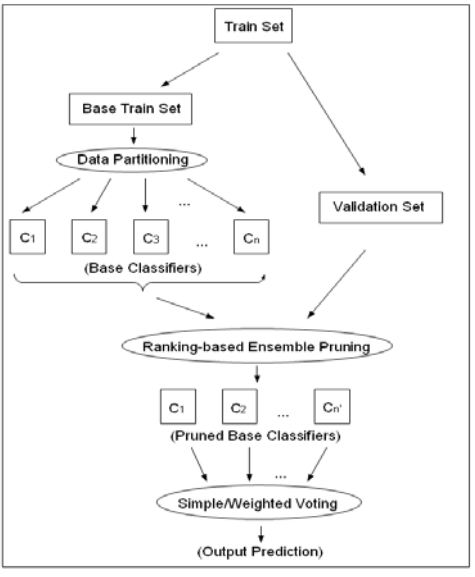

Fig. 1. Ensemble selection process used in this study (adapted from [16])

3 Experimental Environment

3.1 Experimental Design

Figure 1 represents the ensemble selection process used in this study. Firstly, the training set is divided into two separate parts. The base training set is used for training the base classifiers (i.e the ensemble). We construct the ensemble by dividing the base training set with homogeneous (in which base classifiers are trained by the same algorithm) data partitioning methods.

We apply four different partitioning methods: bagging, random-size sampling, disjunct, and fold partitioning [6].

- *Bagging* [1] creates ensemble members each of size N documents by randomly selecting documents with replacement where N is the size of the training set.
- *Disjunct partitioning* divides the training set into k equal-size partitions randomly and each k partition is trained separately.
- *Fold partitioning* divides the training set into k equal-size partitions and *k-1* partitions are trained for each partitions.
- *Random-size sampling* is similar to bagging, but the size of each ensemble member is chosen randomly.

The base classifiers are then trained with four popular machine learning algorithms: C4.5 decision tree [15], KNN (k-nearest Neighbor) [4] , NB (Naive Bayes) [8], and SVM (Support Vector Machines) [18]. KNN's k value is set as 1 and the default parameters are used for other classifiers.

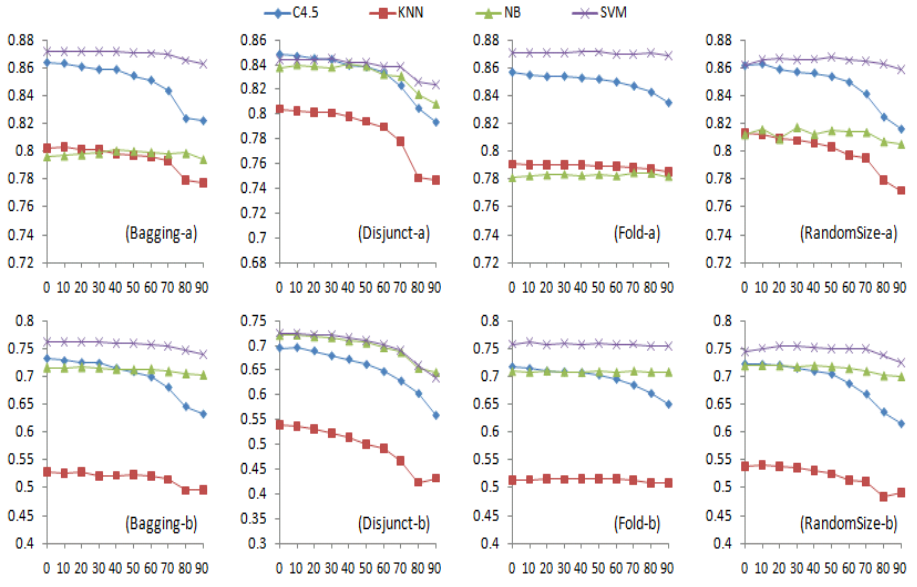

Fig. 2. Accuracy vs. pruning level: experimental results of different data partitioning and categorization methods on two datasets: (a) Reuters-21578 (b) BilCat-TRT. (Figures are not drawn to the same scale.)

After constructing the ensemble we decide to use simple solutions for ensemble selection since constructing data partitioning ensembles is a time-consuming process for large text collections. We choose ranking-based ensemble pruning that does not use complex search algorithms of other ensemble selection methods. Each ensemble member is ranked according to their accuracy on the validation set. We use a distinct part of the training set for the validation. The size of the validation set is set as 5% of the training set since we observe reasonable effectiveness and efficiency and accordingly, 5% of each category's documents are chosen randomly without replacement. After ranking, we prune the ranked-list 10% to 90% by 10% increments.

For the combination of the pruned base classifiers, we choose weighted voting that avoids the computational overload of stacking, mixture of experts etc. Class weight of each ensemble member is taken as its accuracy performance on the validation set. If the validation set of a class is empty (when number of documents in a class is not enough), then simple voting is applied.

Considering each four data partitioning methods with four classification algorithms, we use a thorough experimental approach and repeat the above ensemble pruning procedure for 16 different scenarios. All experiments are repeated 30 times and results are averaged. Documents are represented with term frequency vectors. Ensemble size (i.e data partitioning parameter) is set as 10 and the most frequent 100 unique words per category are used to increase efficiency. We use the classification accuracy for effectiveness measurement.

Table 1. The highest ensemble pruning degrees(%) obtained by unpaired t-test for each partitioning and categorization method on both datasets*

	Reuters-21578					BilCat-TRT				
	C4.5	KNN	NB	SVM	Avg.	C4.5	KNN	NB	SVM	Avg.
Bagging	10	10	90	60	42.5	10	20	60	50	35
Disjunct	10	30	60	40	35	10	0	20	30	15
Fold	0	0	90	50	35	10	60	90	90	62.5
Random-size	10	10	90	90	50	0	20	50	70	35
Avg.	7.5	12.5	82.5	60	40,6	7.5	25	55	60	36,8

* All accuracy differences between traditional ensemble and ensemble pruning approaches are statistically insignificant ($p > 0.05$) up to the pruning degrees given above. This means that, for example, with Reuters-21578, NB, and Bagging we can prune 90% of ensemble members with no statistically significant decrease in accuracy with respect to traditional ensemble approach.

3.2 Datasets

We use the following two datasets in the experiments: Reuters-21578 and BilCat-TRT (http://cs.bilkent.edu.tr/~ctoraman/datasets/ensemblePruning). The former one is a well-known benchmark dataset [9]. After splitting it with ModApte, eliminating multi-class documents and choosing the 10 most frequent topics, we get 5,753 training and 2,254 test news articles. The latter consists of 3,184 training and 1,660 test Turkish news articles. We choose these two datasets to observe the performance in both English and Turkish.

4 Experimental Results

4.1 Pruning Results

The four questions given in introduction are answered in this section. Firstly, Figure 2 gives the results of how much ensemble member we can prune with different data partitioning and categorization methods. These figures can be interpreted either heuristically or statistically. In heuristic way, one can look at Figure 2 and choose an appropriate pruning degree regarding some accuracy reduction. In general, fold partitioning seems to be more robust to accuracy reduction while disjunct partitioning is the weakest one. Similarly, NB and SVM are more suitable for ensemble pruning while C4.5 prunes the least number of base classifiers.

One can also apply some statistical methods to obtain a pruning degree regarding no accuracy reduction. We apply unpaired two-tail t-test between each pruning degree and traditional ensembling to check whether accuracy reduction is statistically significant. We apply unpaired t-test until difference becomes statistically significant. Pruning degrees regarding no accuracy reduction with unpaired t-test are listed in Table 1. We can prune up to %90 ensemble members using fold partitioning and NB on both datasets. Disjunct partitioning seems to

Table 2. Traditional ensembling accuracy and pruning's highest accuracy for each data partitioning and categorization method on Reuters-21578

	Traditional / Pruning's Highest (Pruning Degree)			
	C4.5	KNN	NB	SVM
Bagging	0.8646/-	0.8044/-	0.7928/0.8007(40%)**	0.8714/**0.8722(10%)**
Disjunct	0.8490/-	0.8024/-	0.8351/0.8404(40%)*	0.8414/0.8452(30%)**
Fold	0.8576/-	0.7921/-	0.7780/0.7846(60%)**	0.8718/-
Random-size	0.8624/0.8629(10%)	0.8139/-	0.8092/0.8174(30%)**	0.8565/0.8682(40%)**

Table 3. Traditional ensembling accuracy and pruning's highest accuracy for each data partitioning and categorization method on BilCat-TRT

	Traditional / Pruning's Highest (Pruning Degree)			
	C4.5	KNN	NB	SVM
Bagging	0.7325/-	0.5277/0.5282	0.7128/0.7163(20%)*	0.7605/**0.7620(20%)**
Disjunct	0.6987/-	0.5529/-	0.7209/0.7220(10%)	0.7206/-
Fold	0.7159/0.7171	0.5180/-	0.7076/0.7101(20%)*	0.7554/0.7612(10%)**
Random-size	0.7290/-	0.5423/-	0.7186/0.7205(10%)	0.7479/0.7549(30%)*

* Difference between traditional and pruning's highest is highly statistically significant when $p < 0.05$

** Difference between traditional and pruning's highest is extremely statistically significant when $p < 0.01$

be the worst method for ensemble pruning with no accuracy reduction. Similar to heuristic observations, we get better pruning degrees when either NB or SVM is used. Small amount of ensemble members are pruned using C4.5 and KNN with no accuracy reduction.

Table 1 also suggests that all partitioning and categorization methods prune similar number of ensemble members in both English and Turkish when no accuracy reduction is considered. However, NB prunes more or equal number of ensemble members with all partitioning methods in English than those of Turkish.

In some pruning degrees, we observe that ensemble pruning even increases accuracy of traditional ensembling. Table 2 and 3 list accuracies of traditional ensembling and highest increased accuracy that we can obtain by ensemble pruning using Reuters-21578 and BilCat-TRT respectively. If any degree of ensemble pruning makes no increase in accuracy, then we only give its traditional ensembling accuracy. We also give the pruning degree in which we get the highest accuracy within parentheses. Note that these pruning degrees are not the same as those in Table 1. Unpaired t-test is applied for all comparisons between traditional ensemble and pruning's highest increased accuracy. Results show that, in general, it is possible to increase accuracy with NB and SVM when ensemble pruning is applied. The combination when highest accuracies are seen is bagging with SVM on both datasets. Fold with SVM and random-size with SVM are almost as good as bagging with SVM.

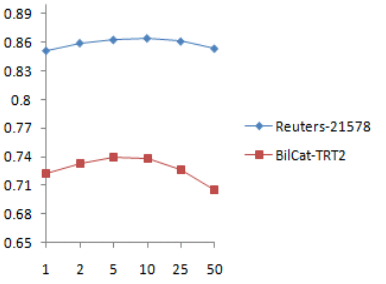

Fig. 3. Accuracy vs. validation set size: effect of different validation set size between 1% and 50% of original training set

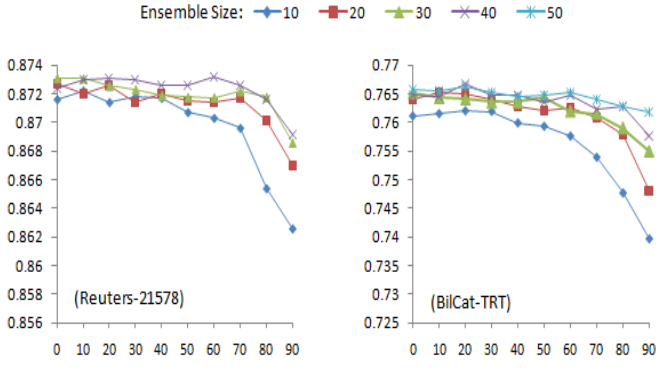

Fig. 4. Accuracy vs. pruning level: effect of different ensemble set size between 10 and 50 base classifiers. (Figures are not drawn to the same scale.)

4.2 Pruning-Related Parameters

Ranked-based ensemble pruning explained in Section 3.1 is a simple strategy that depends on choosing an appropriate validation set and ensemble size. In the previous experiments, validation set is chosen as 5% of the original training set and ensemble size is set as 10. These decisions are chosen for simplicity. However, other decisions may affect the accuracy result of ranked-based ensemble pruning. The following experiments are conducted on only bagging with SVM.

Different validation set sizes on both datasets are examined in Figure 3. Validation size experiments are conducted by 90% pruning of 10 base classifiers. We randomly select news documents for each category between 1% and 50% of the original training set and set this separate part as validation set. Figure 3 shows that if validation set size is either too small or too big, accuracy becomes reduced. Optimal validation set size is somewhere between 5% and 10% of the original training set.

Ensemble size is another parameter for ensemble pruning. Figure 4 displays pruning accuracies of different number of base classifiers between 10 and 50.

In ensemble set size experiments, validation set size is selected as 5% of the original training set. Accuracy is slightly increased with increasing number of base classifiers as expected. Moreover, accuracy reduction due to pruning becomes lower as ensemble size increases. However, efficiency is reduced due to the additional workload of training base classifiers. Thus, one should consider the trade-off between reduction in efficiency and increase in accuracy.

5 Conclusion

In this work we study ensemble pruning in text categorization. Ensembles are created with different data partitioning methods and trained by four different popular text categorization algorithms. The controlled experiments are conducted on English and Turkish datasets. We plan to perform further experiments with additional datasets. However, our statistical tests results provide strong evidence about the generalizability of our results. The main goals are to find how many ensemble members we can prune in text categorization without hurting accuracy, which data partitioning methods and categorization algorithms are more suitable for ensemble pruning, how English and Turkish differ in ensemble pruning, and lastly whether we can increase accuracy with ensemble pruning.

This study employs data partitioning methods with several classification algorithms in ensemble pruning. The main results of this study are:

1. Up to 90% of ensemble members can be pruned with almost no decrease in accuracy (See Table 1).
2. NB and SVM prune more ensemble members than C4.5 and KNN. Using disjunct partitioning prunes less members than other methods.
3. Pruning results are similar for both English and Turkish.
4. It is possible to increase accuracy with ensemble pruning (See Table 2 and 3). But pruning degrees are decreased in comparison to degree values without accuracy decrease (See Table 1). The best accuracy results are obtained by bagging with SVM on both datasets.

We also examine the effect of different ensemble and validation set sizes. It is seen that using 5-10% of the training set for validation is an appropriate decision for both datasets. We also find that accuracy reduction becomes smaller as ensemble size increases.

In future work different ensemble selection methods and validation measures can be studied. Additional test collections in other languages can be used in further experiments.

References

1. Breiman, L.: Bagging predictors. Mach. Learn. 24, 123–140 (1996)
2. Caruana, R., Munson, A., Niculescu-Mizil, A.: Getting the most out of ensemble selection. In: ICDM 2006, pp. 828–833. IEEE Computer Society, Washington, DC (2006)

3. Caruana, R., Niculescu-Mizil, A., Crew, G., Ksikes, A.: Ensemble selection from libraries of models. In: Proceedings of The Twenty-First Int. Conf. on ML, ICML 2004, p. 18 (2004)
4. Cover, T., Hart, P.: Nearest neighbor pattern classification. IEEE Transactions on Information Theory 13, 21–27 (1967)
5. Dietterich, T.G.: Ensemble Methods in Machine Learning. In: Kittler, J., Roli, F. (eds.) MCS 2000. LNCS, vol. 1857, pp. 1–15. Springer, Heidelberg (2000)
6. Dong, Y.S., Han, K.S.: Text classification based on data partitioning and parameter varying ensembles. In: Proceedings of the 2005 ACM Symposium on Applied Computing, SAC 2005, pp. 1044–1048 (2005)
7. Hernández-lobato, D., Martínez-Muñoz, G., Suárez, A.: Pruning in ordered regression bagging ensembles. In: Proceedings of IJCNN 2006, IEEE WCCI 2006, Vancouver, BC, pp. 1266–1273 (2006)
8. John, G.H., Langley, P.: Estimating continuous distributions in bayesian classifiers. In: UAI 1995, pp. 338–345 (1995)
9. Lewis, D.D., Ringuette, M.: A comparison of two learning algorithms for text categorization. In: Symposium on Document Analysis and Information Retrieval, pp. 81–93. ISRI, Univ. of Nevada, Las Vegas (1994)
10. Lu, Z., Wu, X., Zhu, X., Bongard, J.: Ensemble pruning via individual contribution ordering. In: Proceedings of the 16th ACM SIGKDD, KDD 2010, pp. 871–880 (2010)
11. Margineantu, D.D., Dietterich, T.G.: Pruning adaptive boosting. In: Proceedings of the Fourteenth International Conference on ML, ICML 1997, pp. 211–218 (1997)
12. Martínez-Muñoz, G., Suárez, A.: Aggregation ordering in bagging. In: Proc. of the IASTED, pp. 258–263. Acta Press (2004)
13. Martínez-Muñoz, G., Suárez, A.: Using boosting to prune bagging ensembles. Pattern Recognition Letters 28, 156–165 (2007)
14. Prodromidis, A.L., Stolfo, S.J., Chan, P.K.: Effective and efficient pruning of meta-classifiers in a distributed data mining system. Tech. rep. (1999)
15. Quinlan, J.R.: C4.5: Programs for machine learning. Morgan Kaufmann Publishers Inc., San Francisco (1993)
16. Toraman, C.: Text Categorization and Ensemble Pruning in Turkish News Portals. M.Sc. Thesis. Bilkent University, Ankara, Turkey (2011)
17. Tsoumakas, G., Partalas, I., Vlahavas, I.: A taxonomy and short review of ensemble selection. In: ECAI 2008, Workshop on Supervised and Unsupervised Ensemble Methods and Their Applications (2008)
18. Vapnik, V.: Estimation of Dependences Based on Empirical Data. Springer Inc., Secaucus (1982)
19. Witten, I.H., Frank, E.: Data Mining: Practical Machine Learning Tools and Techniques, 2nd edn. Morgan Kaufmann Series in Data Management Systems. Morgan Kaufmann Publishers Inc., San Francisco (2005)