Collective Agreement-based Pruning of Ensembles

Lior Rokach
Department of Information System Engineering,
Ben-Gurion University of the Negev
liorrk@bgu.ac.il

Abstract

The main idea of ensemble methodology is to weigh several individual pattern classifiers, and combine them to reach a better classification performance. Nevertheless, some ensembles superfluously contain too many members, which results in large storage requirements and in some cases it may even reduce classification performance. The goal of ensemble pruning is to identify a subset of ensemble members that performs at least as good as the original ensemble and discard any other members as redundant members.

In this paper we present the Collective Agreement-based Pruning (CAP) method. Rather than ranking individual member, CAP ranks the worth of ensemble subsets by considering the individual predictive ability of each member along with the degree of redundancy among them. Subsets whose members highly agree with the class while having low inter-agreement are preferred.

1. Introduction

The main idea of an ensemble methodology is to combine a set of models, each of which solves the same original task, in order to obtain a better composite global model, with more accurate and reliable estimates or decisions than can be obtained from using a single model. In fact, ensemble methodology imitates our second nature to seek several opinions before making any crucial decision. We weigh the individual opinions, and combine them to reach a final decision (Polikar, 2006).

The ensemble idea has been seeded long time by Tukey (1977). However the main progress in the field has been made when Nineties. Hansen and Salamon (1990) suggested an ensemble of similarly configured neural networks to improve the predictive performance of a single ANN. About the same time Schapire (1990) laid the foundations

for the award winning AdaBoost (Freund and Schapire, 1996) algorithm by showing that a strong classifier in probably approximately correct (PAC) sense can be generated by combining "weak" classifiers (that is, simple classifiers whose classification power is only slightly better than random classification).

In the past few years, experimental studies conducted by the machine-learning community show that combining the outputs of multiple classifiers reduces the generalization error (Bauer and Kohavi, 1999). Ensemble methods are very effective, mainly due to the phenomenon that various types of classifiers have different ``inductive biases". Indeed, ensemble methods can effectively make use of such diversity to reduce the variance-error (Ali and Pazzani, 1996) without increasing the bias-error. In certain situations, an ensemble can also reduce bias-error, as shown by the theory of large margin classifiers (Bartlett and Shawe-Taylor, 1998).

The ensemble methodology is applicable in many fields such as: finance (Leigh et al., 2002), bioinformatics (Tan et al., 2003), medicine (Mangiameli et al., 2004), cheminformatics (Merkwirth et al., 2004), manufacturing (Rokach, 2008), geography (Bruzzone et al. 2004), and Image Retrieval (Lin et al., 2006).

Creating an ensemble in which each classifier is as different as possible while still being consistent with the training set is theoretically known to be an important feature for obtaining improved ensemble performance (Kuncheva, 2005). Diversified classifiers lead to uncorrelated errors, which in turn improve classification accuracy.

For regression problems, the bias-variance-covariance decomposition has been suggested to explain why and how diversity between individual models contributes toward overall ensemble accuracy. In such cases it can be easily shown that the ensemble error can be reduced by increasing ensemble diversity while maintaining the average error of a single model. Nevertheless, in the classification context, there is no complete and agreed upon theory (Brown et al., 2005).

As in decision tree induction, it is sometimes useful to let the ensemble grow freely and then prune the ensemble in order to get more effective and compact ensembles. In an empirical study that was conducted in order to understand the affect of ensemble sizes on ensemble accuracy and diversity, it has been shown that it is feasible to keep a small ensemble while maintaining accuracy and diversity similar to those of a full

ensemble (Liu et al., 2004). Ensemble pruning is important for two reasons: efficiency and predictive performance (Tsoumakas et al., 2008). Having a large ensemble results in computational overhead. Empirical examinations indicate that pruned ensembles may improve the accuracy performance in comparing to the original ensemble (Margineantu and Dietterich, 1997).

Roughly speaking the two most popular approaches for selecting an ensemble subset are Ranking-based and Search Based Methods (see Tsoumakas et al., 2008 for additional approaches).

Ranking-based

The idea of this approach is to *once* rank the individual members according to a certain criterion and choosing the top ranked classifiers according to a threshold. For example Prodromidis et al. (1999) suggest ranking classifiers according to their classification performance on a separate validation set and their ability to correctly classify specific classes. Similarly Caruana et al. (2004) presented a forward stepwise selection procedure in order to select the most relevant classifiers (that maximize the ensemble's performance) among thousands of classifiers. The algorithm FS-PP-EROS generates a selective ensemble of rough subspaces (Hu et al., 2007). The algorithm performs an accuracy-guided forward search to select the most relevant members. The experimental results show that FS-PP-EROS outperforms bagging and random subspace methods in terms of accuracy and size of ensemble systems.

In attribute bagging (Bryll et al., 2003), classification accuracy of randomly selected *m*-attribute subsets is evaluated by using the wrapper approach and only the classifiers constructed on the highest ranking subsets participate in the ensemble voting. Margineantu and Dietterich (1997) present an agreement based ensemble pruning which measures the Kappa statistics between any pair of classifiers. Then pairs of classifiers are selected in ascending order of their agreement level till the desired ensemble size is reached.

Search Based Methods

Instead of separately ranking the members, one can perform a heuristic search in the space of the possible different ensemble subsets while evaluating the collective merit of a candidate subset. The GASEN algorithm was developed for selecting the most appropriate classifiers in a given ensemble (Zhou et al., 2002). In the initialization phase, GASEN assigns a random weight to each of the classifiers. Consequently, it uses genetic algorithms to evolve those weights so that they can characterize to some extent the fitness of the classifiers in joining the ensemble. Finally, it removes from the ensemble those classifiers whose weight is less than a predefined threshold value. A revised version of the GASEN algorithm called GASEN-b has been suggested (Zhou and Tang, 2003). In this algorithm, instead of assigning a weight to each classifier, a bit is assigned to each classifier indicating whether it will be used in the final ensemble. In an experimental study the researchers showed that ensembles generated by a selective ensemble algorithm, which selects some of the trained C4.5 decision trees to make up an ensemble, may be not only smaller in size but also stronger in the generalization than ensembles generated by non-selective algorithms. A similar approach can also be found in (Kim et al., 2002).

Rokach et al. (2006) suggest first to rank the classifiers according to their ROC performance. Then, they suggest evaluating the performance of the ensemble subset by using the top ranked members. The subset size is increased gradually until there are several sequential points with no performance improvement.

Prodromidis and Stolfo (2001) introduce a backwards correlation based pruning. The main idea is to remove the members that are least correlated to a meta-classifier which is trained based on the classifiers' outputs. In each iteration they remove one member and recompute the new reduced meta-classifier (with the remaining members). The meta-classifier in this case is used to evaluate the collective merit of the ensemble.

Windeatt and Ardeshir (2001) compared several subset evaluation methods that were applied to Boosting and Bagging. Specifically the following pruning methods have been compared: Minimum Error Pruning (MEP), Error-based Pruning (EBP), Reduced-Error Pruning(REP), Critical Value Pruning (CVP) and Cost-Complexity Pruning (CCP).

The results indicate that if a single pruning method needs to be selected then overall the popular EBP makes a good choice.

Zhang el al. (2006) formulate the ensemble pruning problem as a quadratic integer programming problem to look for a subset of classifiers that has the optimal accuracy-diversity trade-off. Using a semi-definite programming (SDP) technique, they efficiently approximate the optimal solution, despite the fact that the quadratic problem is NP-hard.

Which approach to use?

Search Based Methods provide a better classification performance than the ranking based methods (Prodromidis et al., 1999). However Search Based methods are usually computational expensive due to their need for searching a large space. Thus one should select a feasible search strategy. Moreover independently to the chosen search strategy, the computational complexity for a evaluating a single candidate subset usually is at least linear in the number of instances in the training set (see Tsoumakas et al., 2008 for complexity analysis of existing evolution measures.)

The aim of this work is developing a low computational complexity evaluation measure to direct the space search. Specifically the computational complexity of the evaluation measure depends only on the ensemble size and does not depend on the training set size. Like in the ranking method of Margineantu and Dietterich (1997) we first calculate the agreement level among all pairs of members. In addition we calculate the agreement level between each member's output and the real label. Then while exploring the space, we use the measure to evaluate the merit of a candidate subset. The measure prefers ensemble subset whose members' classification agrees with the real class, yet the members disagree with each other.

2. Problem Formulation

In a typical classification problem, a training set of labelled examples is given. The training set can be described in a variety of languages, most frequently, as a collection of

patterns denoted as $S = (< \boldsymbol{x_1}, y_1 >, ..., < \boldsymbol{x_m}, y_m >)$ where $\boldsymbol{x_q} \in X$ is a vector of feature values charactering the pattern and $y \in \{c_1, ..., c_k\}$ indicates the pattern's class. Usually, it is assumed that the training set records are generated randomly and independently according to some fixed and unknown joint probability distribution *D*.

Let $\Omega = \{M_1, ..., M_n\}$ represent an ensemble of *n* classifiers. M_i is a classifier that can predict the class $M_i(\boldsymbol{x_q})$ of an observation $\boldsymbol{x_q}$. The problem of ensemble pruning is to find the best subset such that the combination of the selected classifiers will have the highest possible degree of accuracy. Consequently the problem can be formally phrased as follows:

Given an ensemble $\Omega = \{M_1, ..., M_n\}$*, a combination method C, and a training set S from a distribution D over the labeled instance space, the goal is to find an optimal subset* $Z_{opt} \subseteq \Omega$ *. which minimizes the generalization error over the distribution D of the classification of classifiers in* Z_{opt} *combined using method C.*

Note that we assume that the ensemble is given, thus we do not attempt to improve the creation of the original ensemble.

It has been shown that the pruning effect is more noticeable on ensemble whose the diversity among its members is high (Margineantu and Dietterich, 1997). Boosting algorithms create diverse classifiers by using widely different parts of the training set at each iteration (Zhang et al., 2006). Specifically we employ the most popular methods for creating the ensemble: Bagging and AdaBoost. Bagging (Breiman, 1996) employs bootstrap sampling to generate several training sets and then trains a classifier from each generated training set. Note that, since sampling with replacement is used, some of the original instances may appear more than once in the same generated training set and some may not be included at all. The classifier predictions are often combined via majority voting. AdaBoost (Freund and Schapire, 1996) sequentially constructs a series of classifiers, where the training instances that are wrongly classified by a certain classifier will get a higher weight in the training of its subsequent classifier. The classifiers' predictions are combined via weighted voting where the weights are

determined by the algorithm itself based on the training error of each classifier. Specifically the weight of classifier *i* is determined by Equation 1:

$$\alpha_i = \frac{1}{2}\ln\left(\frac{1-\varepsilon_i}{\varepsilon_i}\right) \quad (1)$$

where ε_i is the training error of classifier *i*.

The ensemble pruning problem resemble to the well known feature selection problem. However, instead of selecting features one should select the ensemble's members (Liu et al., 2004). This lead to the idea of adapting the Correlation-based Feature Selection method (Hall, 2000) to the current problem. The CFS algorithm is suitable to this case, because in many ensembles there are many correlated base-classifiers.

3. Collective Agreement -based Ensemble Pruning Method

The Collective Agreement-based Ensemble Pruning (CAP) calculates the member-class and member-member agreements based on the training data. Member-class agreement indicates how much the member's classifications agree with the real label while member-member agreement is the agreement between the classifications of two members. Based on performance criterion adopted from test theory, the merit of an ensemble subset Z with n_z members can be estimated from:

$$Merit_z = \frac{n_z \overline{\kappa}_{cm}}{\sqrt{n_z + n_z(n_z-1)\overline{\kappa}_{mm}}} \quad (2)$$

where $\overline{\kappa}_{cf}$ is the mean agreement between the Z's members and the class and $\overline{\kappa}_{mm}$ is the average member-member agreements in Z. Specifically, the Kappa statistics is used to measure the agreement:

$$\kappa_{i,j} = \frac{\vartheta_{i,j} - \theta_{i,j}}{1-\theta_{i,j}} \quad (3)$$

where $\vartheta_{i,j}$ is the proportion of instances on which the classifiers *i* and *j* agree with each other on the training set, and $\theta_{i,j}$ is the probability that the two classifiers agree by chance.

Alternatively one can use the symmetrical uncertainty (a modified information gain measure) to measure the agreement between two members (Hall, 2000):

$$SU_{i,j} = \frac{H(\hat{\mathbf{y}}_i) + H(\hat{\mathbf{y}}_j) - H(\hat{\mathbf{y}}_i, \hat{\mathbf{y}}_j)}{H(\hat{\mathbf{y}}_i) + H(\hat{\mathbf{y}}_j)} \qquad (4)$$

where $\hat{\mathbf{y}}_i$ is the classification vector of classifier *i* and *H* is the entropy function.

As the search space is huge (2^n), we are using best first search strategy as the preferred strategy. It explores the search space by making local changes to the current ensemble subset. Best first search strategy begins with an empty ensemble subset. If the path being explored does not achieve an improved merit, the best first strategy backtracks to a more promising previous subset and continues the search from there. The search stops if five consecutive iterations obtain non-improving subsets.

The pseudocode of the proposed algorithm is presented in Figure 1. The algorithm gets as input the training set, the ensemble of classifiers, the method for calculating the agreement measure (for example Kappa statistics) and the search strategy. It first calculates the classifiers' output (prediction) on each instance in the training set (Lines 1-5). Then it calculates the mutual agreement matrix among the classifiers' outputs and the agreement between each classifier's output and the actual class (Lines 6-11). Finally it searches the space according to the given search strategy. The search procedure uses the merit calculation for evaluating a certain solution (Lines 14-24).

CAP (S,Q,CT,k)

Input: S Training set

Ω Ensemble of classifiers $\{M_1,...,M_n\}$

Agr A method for calcuating the agreement measure

Src A search startegy

Output: Z Pruned ensemble set

1: FOR each $< \boldsymbol{x_q}, y_q >\in S$ /* Getting members' classifications */

2: FOR each $M_i \in \Omega$

3: $\hat{y}_{i,q} \leftarrow M_i(\boldsymbol{x_q})$

4: END FOR

5: END FOR

6: FOR each $M_i \in \Omega$ /* Preparing the agreement matrix */

7: $CM_i = Agr(\mathbf{y}, \hat{\mathbf{y}}_i)$

8: FOR each $M_j \in \Omega; j > i$

9: $MM_{i,j} = Agr(\hat{\mathbf{y}}_i, \hat{\mathbf{y}}_j)$

10: END FOR

11: END FOR

12: $Z \leftarrow Src(\Omega, MM, CM)$ /* Searching the space using the merit function */

13: Return Z

EvaluateMerit (Z,CM,MM)

Input: Z The Ensemble Subset

CM Class-Member agreement vecor

MM Member-Member agreement matrix

Output: $Merit_z$ - The merit of Z.

14: $n_z \leftarrow |Z|$

15: $\overline{\kappa}_{cm} \leftarrow 0$

16: $\overline{\kappa}_{mm} \leftarrow 0$

17: FOR each $M_i \in Z$

18: $\overline{\kappa}_{cm} \leftarrow \overline{\kappa}_{cm} + CM_i$

19: FOR each $M_j \in Z; j > i$

20: $\overline{\kappa}_{mm} \leftarrow \overline{\kappa}_{mm} + MM_{i,j}$

21: END FOR

22: END FOR

23: $Merit_z \leftarrow \frac{n_z \overline{\kappa}_{cm}}{\sqrt{n_z + n_z(n_z - 1)\overline{\kappa}_{mm}}}$

24: Return $Merit_z$

Figure 1: A Pseudocode of Collective Agreement-based Pruning of Ensembles

The computational complexity of the agreement matrix calculation (lines 6-11) is $o(n^2 m)$ assuming that the complexity of the agreement measure is $o(m)$. This assumption is true for the two measures presented in equations 3 and 4. The computational complexity of the merit evaluation (lines 14-24) is: $o(n^2)$. If the search strategy imposes a partial ordering on the search space, then the merit can be calculated incrementally. For example if backward search is used then it is requires one addition to the numerator and up to *n* additions/subtractions in the denominator.

Note that the actual computational complexity depends on the computational complexity of the classifier making a classification (line 3) and the computational complexity of the search strategy which is being used (line 12). Nevertheless neither the computational complexity of evaluating a solution's merit nor the search space size depends on the training set size. Thus, the proposed method makes it possible to thoroughly search the space for problems with large training sets. For example the complexity for a forward selection or backward elimination is $o(n^2)$. Best first search is exhaustive, but the use of a stopping criterion makes the probability of exploring the entire search space small.

4. Experimental Study

In order to illustrate to evaluate the performance of the proposed Agreement-based Ensemble Pruning algorithm, a comparative experiment was conducted on benchmark datasets. The following subsections describe the experimental set-up and the results obtained.

4.1 Datasets

The selected algorithms were examined on 30 datasets, which were selected manually from the UCI Machine Learning Repository and are widely used by the pattern recognition community for evaluating learning algorithms. The datasets vary across such dimensions as the number of target classes, of instances, of input features and their type (nominal, numeric).

4.2 Algorithms Used

We use Adaboost to create the original ensemble for pruning. Following Zhang et al. (2006) the ensemble size was set to one hundred. If the training error converges to zero before the ensemble sizes reaches one hundred, then all the subsequent members will be replications of the last member because no instance weight is changed. Therefore in such cases, the AdaBoost.M1 procedure is repeated using a different random seed. Note that the original AdaBoost weights are used to weight the predictions of the selected members.

As for the induction algorithm that was used for training the base classifier, we have examined the C4.5 decision tree algorithm (Quinlan, 1993) and decision stump. The C4.5 algorithm is frequently used for comparing ensembles. Decision stump is a weak learner consisting of a one-level tree which known to be benefit from boosting strategy (Kotsiantis et al., 2006).

The new algorithm is compared to the following ensemble pruning methods:

1. GASEN-b - genetic-algorithm (number of generation=200 and population size=50). The GASEN employs a wrapper evaluator, in which a candidate subset is evaluated by repeatedly sampling the training set and measuring the accuracy of the subset ensemble over a holdout validation dataset.
2. Kappa members ranking (as used by Margineantu and Dietterich, 1997). Kappa members ranking can prune the ensemble to any pre-set size. Thus, in order to make a fair comparison, we set it to the same size obtained by our algorithm which makes.

We also evaluated the following configurations of the proposed approach:

3. CAP-F-K – Using Forward-Selection search strategy with kappa statistics as the agreement measure.

4. CAP-BF-K – Using Best First search strategy with kappa statistics as the agreement measure.
5. CAP-GA-K – Using genetic algorithm search strategy (number of generation=200 and population size=50) with kappa statistics as the agreement measure.
6. CAP-BF-SU – Using Best First search strategy with symmetrical uncertainty as the agreement measure.
7. CAP-GA-SU – Using genetic algorithm search strategy (number of generations=200 and population size=50) with symmetrical uncertainty as the agreement measure.

4.3 Metrics Measured

In this experiment the following metrics were measured:

1. Generalized Accuracy: This represents the probability that an instance was classified correctly. In order to estimate the generalized accuracy, a 10-fold cross-validation procedure was repeated five times. For each 10-fold cross- validation, the training set was randomly partitioned into 10 disjoint instance subsets. Each subset was utilized once in a test set and nine times in a training set. The same cross-validation folds were implemented for all algorithms. Since the average accuracy is a random variable, the confidence interval was estimated by using the normal approximation of the binomial distribution. In order to conclude which algorithm performs best over multiple datasets, we followed the procedure proposed in Demsar (2006). We first used the adjusted Friedman test in order to reject the null hypothesis and then the Bonferroni-Dunn test to examine whether the new algorithm performs significantly better than existing algorithms.
2. Computational Cost: Since this paper focuses on reducing the complexity cost, the running time required for pruning the ensemble was measured.
3. Pruned Ensemble Size.

4.4 Accuracy Performance

Table 1 presents the mean accuracy and the standard deviation over five runs of 10 fold cross-validation using C4.5 algorithm as the base classifier. The shaded boxes represent cases where the difference between CAP-Best-First-Kappa and the corresponding method is statistically significant with 95% confidence using t-test. A win-loss-tie summarization based on mean value and t test is attached at the bottom of the table. Generally Kappa measure slightly outperforms symmetrical uncertainty and GA search outperforms Best First search.

Using adjusted Friedman test the null hypothesis that all pruning methods perform the same over multiple data sets and the observed differences are merely random has been reject with $F_F(8,232) = 10.05, p < 0.001$. We proceed with a post-hoc Bonferroni-Dunn test using CAP-Best-First-Kappa as the controlled method. We concluded that all variations of CAP method perform almost the same. Still CAP-Best-First-Kappa significantly outperforms CAP-Best-First-Symmetrical-Uncertainty with z= 2.26, p<0.05.

The accuracy of the proposed pruned ensemble is similar to the accuracy of the original ensemble (no pruning). CAP-Best-First-Kappa significantly outperforms Kappa Ranking with z=4.14, p<0.001. Moreover CAP-GA-Kappa significantly outperforms GASEN-b with z=2.32, p<0.01. This indicates that the using collective merit measure is more accurate than using the wrapper approach when GA search strategy is used. This conclusion is not expected, because wrapper approach is generally considered to be slow but accurate mean to direct the search process.

Table 2 presents the mean accuracy and the standard deviation over five runs of 10 fold cross-validation using Decision Stump algorithm as the base classifier. The shaded boxes represent cases where the difference between CAP-Best-First-Kappa and the corresponding method is statistically significant with 95% confidence using t-test. A win-loss-tie summarization based on mean value and t test is attached at the bottom of the table. All pruning methods slightly reduce the accuracy performance when compared to the No-Pruning results. Nevertheless CAP-Best-First-Kappa significantly outperforms No-Pruning in the Wine dataset. Generally Kappa measure slightly outperforms symmetrical uncertainty.

Using adjusted Friedman test the null hypothesis that all pruning methods perform the same over multiple data sets and the observed differences are merely random has been reject with $F_F(8, 232) = 7.168, p < 0.001$. We proceed with a post-hoc Bonferroni-Dunn test using CAP-Best-First-Kappa as the controlled method. CAP-Best-First-Kappa significantly outperforms Kappa Ranking with z=4.336, p<0.001. This is consistent with the superiority of CAP-Best-First-Kappa over Kappa Ranking in the previous table (using C4.5 as nase classifier). Thus, we conclude that the usage of Kappa statistics is not sufficient to obtain favorable results, but the collective merit measure (Eq. 2) is required. Moreover, CAP-Best-First-Kappa significantly outperforms CAP-Best-First-Symmetrical-Uncertainty with z= 1.6, p<0.05. Again this result is consistent with the superiority obtained in the previous table. This leads to the conclusion that Kappa is a better measure than symmetrical uncertainty for pruning ensembles.

Table 1: Mean accuracy using C4.5 as base classifier

Dataset	# Instances	# Features	A single C4.5		No Pruning		Kappa Ranking		GASEN-b		CAP-F-K		CAP-GA-K		CAP-BF-K		CAP-GA-SU		CAP-BF-SU	
			Mean	S.D.	Mean	S.D.	Mean	S.D.	Mean	S.D.	Mean	S.D.	Mean	S.D.	Mean	S.D.	Mean	S.D.	Mean	S.D.
Anneal	898	39	98.51	0.24	99.67	0.16	98.9	2.1	98.17	3.2	95.69	4.88	99.55	0.17	95.69	4.88	99.44	0.19	95.64	4.88
Audiology	200	70	77.05	1.2	84.07	1.3	80.15	1.6	83.18	1.2	85	1.9	85	1.3	85.17	0.9	83.5	1.16	83	0.9
Aust Credit	690	15	86.49	0.69	85.48	0.69	83.17	0.98	82.9	0.74	85.1	0.74	85.28	0.74	85.28	0.75	85.33	0.72	85.39	0.74
Autos	205	26	82.94	2.17	87.39	1.75	85.52	1.17	84.94	1.4	87.1	1.85	85.79	1.67	86.9	1.81	85.55	1.73	87.1	1.83
Diabetes	768	9	74.85	1.35	74.11	1.14	70.13	1	75.9	1.7	73.85	1.09	74.09	1.11	73.83	1.08	74.74	1.12	74.48	1.23
Glass	214	10	69.05	2.02	78.03	1.84	76.02	1.67	81.02	1.17	77.58	1.77	78.47	1.5	77.58	1.78	77.77	1.74	77.58	1.89
Hepatitis	155	20	79.82	1.44	77.33	1.44	75.33	2.01	77.19	2.11	78.89	1.01	79.53	1.17	79.16	1.11	76.45	1.62	77.08	1.65
Ionosphere	351	51	89.63	1.21	93.85	1.08	92.93	0.7	92.91	1.1	92.94	2.28	91.6	3.38	92.94	2.28	91.46	3.34	92.88	2.27
Iris	150	5	94.93	0.91	94.67	0.93	88.72	1.8	96.67	0.79	94.4	0.91	94.4	0.91	94.4	0.91	94.53	0.92	94.27	0.95
Kr-vs-kp	3197	37	99.44	0.06	99.56	0.06	99.41	0.12	99.56	0.06	99.53	0.06	99.55	0.06	99.54	0.06	99.52	0.06	99.54	0.06
Labor	57	17	77.13	2.88	88.6	2.48	82.17	2.19	77	3.01	87.27	2.51	87.53	2.49	87.53	2.57	87.47	2.45	86.8	2.41
LED17	220	25	61.73	1.57	62.73	1.68	61.95	1.29	63.02	2.78	62.36	1.67	62.45	1.71	62.36	1.67	62.18	1.66	62	1.65
Letter	15000	17	75.26	1.1	87.72	2.17	82.12	1.34	87.26	3.6	86.94	2.3	88.06	2.9	87.71	1.95	85.82	2.6	85.61	1.78
Lung	31	56	46.83	4.16	54.17	4.98	53.5	3.9	55	5.86	55.67	4.83	57.5	4.92	57.5	4.98	56.17	4.46	52.5	4.7
Lymphogra	148	19	75.7	2.94	84.07	2.47	83.66	2.2	84.66	1.84	84.35	2.51	85.12	2.59	84.34	2.45	85.48	2.68	83.54	2.39
Monks1	124	6	79.87	1.65	98.72	0.5	96.09	0.45	96.86	0.91	95.46	0.96	96.6	0.82	96.29	0.87	96.77	0.86	96.95	0.8
Monks2	169	6	57.74	1.71	60.47	1.98	61	1.29	60.96	2.02	60.58	1.99	61.43	1.84	61.43	1.83	58.95	1.88	59.53	1.91
Monks3	122	6	90.1	1.08	89.15	1.37	88.95	1.45	86.73	1.5	89	1.09	89.81	1.15	89	1.13	89.79	1.23	88.96	1.33
MUSH	8124	22	100	0	100	0	100	0	100	0	100	0	100	0.01	100	0.01	100	0.01	100	0.01
Nurse	12960	8	97.54	0.06	98.22	0.06	97.34	0.1	98.15	0.56	98.13	0.06	98.19	0.06	98.13	0.06	98.18	0.06	98.12	0.06
Optic	5628	64	63.02	0.31	91.14	0.17	91.07	0.18	89.16	0.27	91.14	0.17	91.14	0.17	91.14	0.17	91.14	0.17	91.14	0.17
Sonar	208	60	70.46	1.72	79.94	1.49	72.14	1.33	72	1.74	79.98	1.41	80.27	1.38	79.98	1.4	79.31	1.43	79.59	1.52
Soybean	683	35	91.5	0.8	92.82	1.4	91.9	1.9	92.24	2.1	92.86	1.17	93.26	1.96	92.97	1.62	91.66	2.42	91.48	1.17
Splice	1000	60	91.1	0.52	94.64	0.39	92.88	0.57	93.1	0.5	94.66	0.39	94.68	0.39	94.68	0.39	94.58	0.4	94.56	0.4
TTT	958	9	84.78	0.56	99.12	0.19	99	0.25	96.45	0.27	99.12	0.19	99.14	0.18	99.14	0.18	99.1	0.19	99.14	0.18
Vehicle	846	19	71.87	1.21	77.85	0.98	73.12	1.2	75.63	0.95	77.92	0.98	78.42	1.13	77.92	0.98	78.42	1.15	77.9	0.98
Vote	290	16	95.93	0.62	95.45	0.61	95.52	0.44	95.52	0.47	95.38	0.61	95.45	0.61	95.45	0.61	95.17	0.64	95.52	0.59
Waveform	5000	41	75.04	0.48	84.95	0.47	80.62	0.3	83.74	0.2	84.95	0.47	84.66	0.52	84.95	0.47	84.66	0.52	84.95	0.47
Wine	178	13	84.15	1.42	94.4	1.06	93.14	1.845	95	0.92	94.06	0.95	94.95	1.04	94.06	1.07	94.95	1.04	94.5	0.99
Zoo	101	8	92.69	1.18	100	1.2	100	1.7	100	2.1	98.17	2.76	100	2.79	100	2.76	97.15	2.76	96.33	2.76
Mean			**81.17**	**1.24**	**86.94**	**1.20**	**84.88**	**1.24**	**85.83**	**1.50**	**86.60**	**1.45**	**87.06**	**1.36**	**86.84**	**1.42**	**86.51**	**1.37**	**86.20**	**1.42**
Significant W-L-T			**17-1-12**		**1-1-28**		**9-1-20**		**9-2-19**		**0-0-30**		**0-1-29**				**3-1-26**		**4-0-26**	

Table 2: Mean accuracy using Decision Stump as base classifier

Dataset	# Instances	# Features	A single C4.5		No Pruning		Kappa Ranking		GASEN-b		CAP-F-K		CAP-GA-K		CAP-BF-K		CAP-GA-SU		CAP-BF-SU	
			Mean	S.D.	Mean	S.D.	Mean	S.D.	Mean	S.D.	Mean	S.D.	Mean	S.D.	Mean	S.D.	Mean	S.D.	Mean	S.D.
Anneal	898	39	77.19	0.97	83.63	0.14	82.46	1.00	83.17	0.14	83.05	0.13	82.46	0.70	83.63	0.14	83.63	0.14	83.63	0.14
Audiology	200	70	47.00	0.08	47.00	0.19	47.00	0.08	47.00	0.02	47.00	0.00	47.00	0.07	47.00	0.10	47.00	0.10	47.00	0.04
Aust Credit	690	15	85.51	0.91	85.88	0.92	85.70	0.91	85.65	0.81	85.36	0.89	85.30	0.91	85.30	0.90	85.70	0.91	85.48	0.89
Autos	205	26	44.88	1.53	44.88	1.53	44.88	1.56	44.88	1.51	44.88	1.53	44.88	1.52	44.88	1.54	44.88	1.51	44.88	1.51
Diabetes	768	9	72.06	1.33	75.57	1.02	72.66	1.36	74.16	1.01	74.56	1.01	73.51	0.95	75.13	1.32	72.66	1.34	73.38	1.15
Glass	214	10	44.90	0.76	45.32	0.76	44.95	0.78	45.29	0.76	45.61	0.76	45.12	0.19	45.79	0.19	44.96	0.19	44.96	0.19
Hepatitis	155	20	81.69	1.75	81.07	1.74	80.15	1.76	65.08	5.60	80.16	1.91	80.94	2.01	80.16	1.91	81.05	1.71	80.79	1.88
Ionosphere	351	51	82.51	1.31	92.43	1.15	76.08	1.33	82.51	1.14	84.59	1.14	81.35	1.38	84.78	1.39	76.08	1.44	76.08	1.26
Iris	150	5	66.67	0.00	94.13	1.49	60.40	0.00	77.33	3.35	93.20	1.49	60.40	1.56	93.07	1.50	75.35	1.47	74.67	3.75
Kr-vs-kp	3197	37	66.05	0.43	95.19	0.31	74.21	0.45	82.72	0.31	74.21	1.15	75.07	0.97	74.21	1.15	89.47	0.31	88.74	0.61
Labor	57	17	78.53	4.02	91.07	3.12	90.19	4.13	79.33	4.48	91.33	3.02	92.07	2.88	91.33	3.02	90.50	3.10	90.20	3.07
LED17	220	25	20.27	0.77	23.18	0.77	22.95	0.78	23.09	0.55	23.09	0.53	22.98	0.53	23.04	0.53	23.08	0.76	22.96	0.53
Letter	15000	17	7.00	0.43	6.99	0.43	7.00	0.45	6.97	0.43	6.94	0.43	6.98	0.43	6.96	0.43	6.95	0.42	6.95	0.43
Lung	31	56	42.50	3.98	49.67	4.67	45.33	4.17	40.83	5.42	45.33	4.34	45.67	4.36	45.33	4.34	47.54	4.65	47.33	4.48
Lymphogra	148	19	74.84	2.85	75.10	2.55	74.83	2.89	75.06	2.52	74.83	2.52	75.57	2.76	74.90	2.69	75.57	2.76	74.90	2.69
Monks1	124	6	73.44	2.00	69.17	2.89	61.23	2.05	66.35	2.74	69.01	2.65	61.23	3.21	69.33	2.64	69.25	2.88	68.59	2.69
Monks2	169	6	59.40	1.52	53.84	2.40	48.74	1.49	55.11	2.72	54.12	2.52	53.01	2.55	54.69	2.29	49.23	2.37	48.75	2.58
Monks3	122	6	71.45	3.14	89.50	1.94	72.19	3.26	78.01	4.25	89.51	2.55	82.85	3.48	89.51	2.55	72.76	1.91	72.19	2.70
MUSH	8124	22	88.68	0.28	99.91	0.03	96.15	0.29	96.70	0.80	97.06	0.29	97.35	0.15	97.16	0.27	96.20	0.03	96.15	0.64
Nurse	12960	8	66.25	0.01	66.25	0.01	64.54	0.01	65.67	0.01	64.54	0.01	66.25	0.01	64.54	0.01	66.22	0.01	66.25	0.01
Optic	5628	64	19.32	0.10	27.92	0.10	27.64	0.10	27.76	0.10	27.64	0.02	27.64	0.02	27.71	0.02	27.79	0.10	27.92	0.02
Sonar	208	60	66.35	2.02	72.56	2.50	69.64	2.11	70.80	2.47	71.03	2.31	70.57	2.12	70.64	2.34	69.64	2.49	70.09	2.23
Soybean	683	35	27.96	0.69	27.82	0.69	27.97	0.69	27.94	0.70	27.96	0.69	27.96	0.69	27.96	0.69	27.96	0.69	27.96	0.69
Splice	1000	60	63.90	0.88	85.16	1.07	81.86	0.87	83.33	1.07	83.12	1.23	83.66	1.12	83.12	1.23	81.86	1.07	82.66	1.17
TTT	958	9	69.94	1.07	89.21	0.79	68.60	1.06	75.40	0.79	68.61	1.21	83.44	1.16	68.61	1.21	71.25	0.79	71.29	1.05
Vehicle	846	19	39.76	0.40	40.07	0.40	40.07	0.42	40.07	0.40	40.10	0.40	40.07	0.15	40.10	0.15	40.07	0.15	40.07	0.15
Vote	290	16	95.86	0.97	95.17	1.06	94.92	0.97	95.41	1.00	95.93	1.08	95.52	1.03	96.00	1.08	94.92	1.07	95.72	0.99
Waveform	5000	41	56.77	0.32	67.83	0.94	65.78	0.32	66.75	0.94	65.79	0.93	66.90	0.50	66.22	0.85	66.88	0.50	66.73	0.53
Wine	178	13	57.33	1.21	57.77	1.22	63.46	1.23	65.32	1.24	71.29	2.61	63.46	3.53	71.29	2.61	64.02	1.24	63.75	3.10
Zoo	101	8	60.40	0.61	60.40	0.61	60.40	0.64	60.45	0.94	60.40	0.61	60.40	0.61	60.40	0.61	60.40	0.61	60.40	0.61
Mean			**60.28**	**1.21**	**66.46**	**1.25**	**61.73**	**1.24**	**62.94**	**1.61**	**64.68**	**1.33**	**63.32**	**1.38**	**64.76**	**1.32**	**63.43**	**1.22**	**63.35**	**1.39**
Significant W-L-T			**15-1-14**		**1-3-26**		**7-0-23**		**7-2-21**		**0-0-30**		**4-1-25**				**6-1-23**		**5-1-24**	

4.5 Pruned Ensemble Size

Table 3 presents the mean pruned ensemble size obtained by each method on each dataset when C4.5 is used as a base classifier. The last row in the table specifies the mean ensemble size over all datasets. All CAP methods usually converge to similar ensembles sizes. GASEN-b usually converges to a smaller ensemble size. Combining the results of Table 1 and Table 3 indicates that it is possible to keep almost the same accuracy of the original ensemble but using only circa 45% of its members.

4.6 Pruning time

Table 4 presents the mean pruning time (in milliseconds) required by each method for various datasets when C4.5 is used as a base classifier. The last row in the table specifies the mean pruning time over all datasets. We conducted all experiments on the following hardware configuration: a desktop computer implementing a Windows XP operating system with Intel Pentium 4-2.8GHz, and 2GB of physical memory. Kappa Ranking is the faster method. All CAP methods have similar complexity costs. Still the Kappa metric tends to be faster than the Symmetrical-Uncertainty metric and Best First Search run faster than GA search. The table reveals that for large datasets (such as Letter and Nurse) the computational cost of GASEN-b is significantly higher than the CAP methods.

Table 3: Pruned ensemble size using C4.5 as base classifier

	No Pruning	GASEN-b	Kappa Ranking	CAP-F-K	CAP-GA-K	CAP-BF-K	CAP-GA-SU	CAP-BF-SU
Anneal	100	17.44	16.45	15.50	29.84	16.45	25.96	15.92
Audiology	100	23.22	26.40	26.53	25.98	26.40	35.86	33.41
Aust credit	100	34.88	32.26	31.96	32.22	32.26	31.07	31.88
Autos	100	44.89	53.26	52.67	56.20	53.26	52.20	50.30
Diabetes	100	19.91	60.35	60.24	73.40	60.35	71.76	60.37
Glass	100	18.27	60.22	60.02	61.72	60.22	64.48	63.92
Hepatitis	100	15.80	33.44	33.43	33.85	33.44	32.53	34.66
Ionosphere	100	60.45	62.98	60.14	67.72	62.98	60.24	56.59
Iris	100	12.11	9.42	9.42	9.84	9.42	9.08	9.84
Kr-vs-kp	100	21.48	17.18	16.86	17.56	17.18	18.02	17.30
Labor	100	12.53	22.85	23.57	23.90	22.85	21.99	22.48
LED17	100	12.82	75.39	74.76	78.10	75.39	74.22	73.57
LETTER	100	17.33	12.00	11.88	11.41	12.00	11.98	12.38
Lung Cancer	100	31.44	20.98	21.20	21.95	20.98	20.51	21.55
Lymphography	100	53.60	47.25	45.20	53.52	47.25	45.40	41.39
Monks1	100	10.09	17.66	17.57	17.68	17.66	18.50	16.96
Monks2	100	9.66	40.11	41.02	39.98	40.11	39.86	41.14
Monks3	100	11.96	23.73	24.42	24.63	23.73	24.23	24.18
MUSH	100	43.22	67.20	76.93	78.50	67.20	78.99	80.00
Nurse	100	11.86	8.08	8.09	8.40	8.08	8.24	7.96
OPTIC	100	45.55	73.81	77.59	80.00	73.81	77.81	78.24
Sonar	100	62.75	73.22	72.28	70.74	73.22	69.91	71.89
Soybean	100	23.32	17.60	17.85	17.52	17.60	17.32	17.26
Splice	100	69.97	79.46	78.74	73.66	79.46	79.30	77.90
TTT	100	65.33	79.71	74.90	79.11	79.71	79.86	74.85
Vehicle	100	18.00	63.62	63.62	63.48	63.62	63.76	63.84
Vote	100	31.18	22.21	21.26	21.66	22.21	22.46	21.14
Waveform	100	27.52	62.40	62.40	62.40	62.40	62.40	80.00
Wine	100	45.01	42.02	42.57	41.62	42.02	42.24	41.54
Zoo	100	17.18	12.42	12.62	12.34	12.42	12.70	12.01
Mean	**100**	**29.65**	**41.12**	**41.17**	**42.96**	**41.12**	**42.43**	**41.82**

Table 4: Pruning time in milliseconds using C4.5 as base classifier

	GASEN-b	Kappa Ranking	CAP-F-K	CAP-GA-K	CAP-BF-K	CAP-GA-SU	CAP-BF-SU
Anneal	4,751	556	2,473	6,714	2,401	7,308	2,401
Audiology	940	280	360	410	340	510	410
Aust Credit	162,301	318	357	3,967	405	631	430
Autos	1,450	42	45	1,589	75	2,033	67
Diabetes	4,490	514	715	1,210	731	5,562	780
Glass	1,300	40	102	359	98	1,964	99
Hepatitis	47,633	10	16	3,414	70	240	78
Ionosphere	1,450	84	159	381	122	4,469	17
Iris	11,689	15	137	3,020	148	373	153
Kr-vs-kp	1,467,428	101	145	2,785	861	836	259
Labor	131	8	10	3,496	42	191	8
LED17	116,716	29	108	3,673	64	96	26
LETTER	1,991,674	9,870	12,129	22,269	13,560	32,721	14,789
Lung Cancer	29,344	1	9	3,185	38	173	12
Lymphography	410	43	79	408	128	1,040	80
Monks1	38,483	9	10	3,410	55	213	52
Monks2	9,286	21	88	3,764	167	284	160
Monks3	27,549	17	21	3,678	78	230	46
MUSH	74,340	2,094	4,922	6,587	5,539	5,503	6,524
Nurse	159,086	14	184	1,639	759	931	2,146
OPTIC	1,393,426	13	206	3,944	935	18,266	2,222
Sonar	780,696	11	142	3,852	193	393	435
Soybean	1,470	510	390	560	420	690	620
Splice	27,723	89	106	3,430	138	343	555
TTT	217,138	30	251	4,004	300	613	667
Vehicle	9,450	100	103	38	61	8,645	224
Vote	45,546	11	72	3,226	96	359	144
Waveform	195,780	140	602	1,972	6,915	268,853	11,399
Wine	290	21	94	3,403	170	318	164
Zoo	129	44	135	2,573	139	405	185
Average	**227,403**	**501**	**806**	**3,732**	**1,168**	**12,423**	**1,505**

5. Conclusions

In this paper we presented the Collective Agreement-based Pruning method for pruning ensembles. The basic idea is that the merit of a certain subset is estimated using the pairwise agreement among members. The computational complexity for obtaining this evaluation does not depend on the training set size, which make it feasible for large datasets. We have examined two metrics for measuring the agreement among members: Symmetrical uncertainty and Kappa statistics where the latter demonstrates a better performance.

The experimental study reveals that CAP typically eliminated well over half the members. In most cases, classification accuracy using the pruned ensemble equaled to the accuracy using the original ensemble. The experimental study also indicates that CAP obtained comparable results to the wrapper approach using the same GA search strategy. CAP executes faster than wrapper especially in larger datasets.

Additional issues to be further studied include: evaluating CAP with other base classifier such as neural networks and other techniques for generating the ensemble (such as bagging).

References

1. Ali K. M., Pazzani M. J., Error Reduction through Learning Multiple Descriptions, Machine Learning, 24: 3, 173-202, 1996.
2. Bartlett P. and Shawe-Taylor J., Generalization Performance of Support Vector Machines and Other Pattern Classifiers, In ``Advances in Kernel Methods, Support Vector Learning", Bernhard Scholkopf, Christopher J. C. Burges, and Alexander J. Smola (eds.), MIT Press, Cambridge, USA, 1998.
3. Bauer, E. and Kohavi, R., ``An Empirical Comparison of Voting Classification Algorithms: Bagging, Boosting, and Variants". Machine Learning, 35: 1-38, 1999.
4. Breiman L., Bagging predictors, Machine Learning, 24(2):123-140, 1996.
5. Brown G., Wyatt J., Harris R., Yao X., Diversity creation methods: a survey and categorisation, Information Fusion, 6(1):5--20.
6. Bruzzone L., Cossu R., Vernazza G., Detection of land-cover transitions by combining multidate classifiers, Pattern Recognition Letters, 25(13): 1491--1500, 2004.
7. Bryll R., Gutierrez-Osuna R., Quek F., Attribute bagging: improving accuracy of classifier ensembles by using random feature subsets, Pattern Recognition Volume 36 (2003): 1291-1302
8. Caruana R., Niculescu-Mizil A. , Crew G. , Ksikes A., Ensemble selection from libraries of models, Twenty-first international conference on Machine learning, July 04-08, 2004, Banff, Alberta, Canada.
9. Demsar J., Statistical Comparisons of Classifiers over Multiple Data Sets, Journal of Machine Learning Research, 7 (2006):1-30.
10. Freund Y. and Schapire R. E., Experiments with a new boosting algorithm. In Machine Learning: Proceedings of the Thirteenth International Conference, pages 325-332, 1996.
11. Friedman, J. H., ``Multivariate Adaptive Regression Splines", The Annual Of Statistics, 19, 1-141, 1991.

12. Hansen L.K. and Salamon P., "Neural network ensembles," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 12, no. 10, pp. 993-1001, 1990.
13. Hu Q., Yu D., Xie Z., Li X., EROS: Ensemble rough subspaces, Pattern Recognition 40 (2007) 3728 – 3739.
14. Kotsiantis S. B., Kanellopoulos D. and Pintelas P. E. (2006), Local Boosting of Decision Stumps for Regression and Classification Problems, JOURNAL OF COMPUTERS, 1(4):30-37
15. Kuncheva L.I. Diversity in multiple classifier systems (Editorial), Information Fusion, 6 (1), 2005, 3-4.
16. Leigh W., Purvis R., Ragusa J. M., Forecasting the NYSE composite index with technical analysis, pattern recognizer, neural networks, and genetic algorithm: a case study in romantic decision support, Decision Support Systems 32(4): 361--377, 2002.
17. Lin H., Kao Y., Yang F., Wang P., Content-Based Image Retrieval Trained By Adaboost For Mobile Application, International Journal of Pattern Recognition and Artificial Intelligence, 20(4):525-541, 2006.
18. Liu H., Mandvikar A., Mody J., An Empirical Study of Building Compact Ensembles. WAIM 2004: pp. 622-627.
19. Mangiameli P., West D., Rampal R., Model selection for medical diagnosis decision support systems, Decision Support Systems, 36(3): 247--259, 2004.
20. Margineantu D. and Dietterich T., Pruning adaptive boosting. In Proc. Fourteenth Intl. Conf. Machine Learning, pages 211--218, 1997.
21. Merkwirth C., Mauser H., Schulz-Gasch T., Roche O., Stahl M., Lengauer T., Ensemble methods for classification in cheminformatics, Journal of Chemical Information and Modeling, 44(6):1971--1978, 2004.
22. Merz C. J, and Murphy P.M., UCI Repository of machine learning databases. Irvine, CA: University of California, Department of Information and Computer Science, h1998.
23. Polikar R., "Ensemble Based Systems in Decision Making," IEEE Circuits and Systems Magazine, vol.6, no. 3, pp. 21-45, 2006
24. Prodromidis, A. L., Stolfo, S. J. and Chan, P. K., Effective and efficient pruning of metaclassifiers in a distributed Data Mining system. Technical report CUCS-017-99, Columbia Univ., 1999.
25. Quinlan, J. R., Bagging, Boosting, and C4.5. In Proceedings of the Thirteenth National Conference on Artificial Intelligence, pages 725-730, 1996.
26. Rokach L., Mining manufacturing data using genetic algorithm-based feature set decomposition, Int. J. Intelligent Systems Technologies and Applications, 4(1):57-78, 2008.
27. Rokach L., R. Arbel, O. Maimon, "Selective Voting - Getting More For Less in Sensor Fusion", International Journal of Pattern Recognition and Artificial Intelligence, 20(3):329-350, 2006.
28. Schapire R.E., "The strength of weak learnability," Machine Learning, vol. 5, no. 2, pp. 197-227, 1990.

29. Tan A. C., Gilbert D., Deville Y., Multi-class Protein Fold Classification using a New Ensemble Machine Learning Approach. Genome Informatics, 14:206--217, 2003.
30. Tsoumakas G., Partalas I., Vlahavas I., A Taxonomy and Short Review of Ensemble Selection, ECAI, Workshop on Supervised and Unsupervised Ensemble Methods and Their Applications (SUEMA-2008), 2008 (accepted for presentation).
31. Tukey J.W., Exploratory data analysis, Addison-Wesley, Reading, Mass, 1977
32. Windeatt T. and Ardeshir G., An Empirical Comparison of Pruning Methods for Ensemble Classifiers, IDA2001, LNCS 2189, pp. 208–217, 2001.
33. Zhou, Z. H., and Tang, W., Selective Ensemble of Decision Trees, in Guoyin Wang, Qing Liu, Yiyu Yao, Andrzej Skowron (Eds.): Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing, 9th International Conference, RSFDGrC, Chongqing, China, Proceedings. Lecture Notes in Computer Science 2639, pp.476-483, 2003.
34. Zhou, Z. H., Wu J., Tang W., Ensembling neural networks: many could be better than all. Artificial Intelligence 137: 239-263, 2002.