

Pruning and Dynamic Scheduling of Cost-sensitive Ensembles

Wei Fan

IBM T.J. Watson Research
Hawthorne, NY 10532
weifan@us.ibm.com

Fang Chu

Computer Science Department
University of California
Los Angeles, CA 90095
fchu@cs.ucla.edu

Haixun Wang and Philip S. Yu

IBM T.J. Watson Research
Hawthorne, NY 10532
{haixun,psyu}@us.ibm.com

Abstract

Previous research has shown that averaging ensemble can scale up learning over very large cost-sensitive datasets with linear speedup independent of the learning algorithms. At the same time, it achieves the same or even better accuracy than a single model computed from the entire dataset. However, one major drawback is its inefficiency in prediction since every base model in the ensemble has to be consulted in order to produce a final prediction. In this paper, we propose several approaches to reduce the number of base classifiers. Among various methods explored, our empirical studies have shown that the benefit-based greedy approach can safely remove more than 90% of the base models while maintaining or even exceeding the prediction accuracy of the original ensemble. Assuming that each base classifier consumes one unit of prediction time, the removal of 90% of base classifiers translates to a prediction speedup of 10 times. On top of pruning, we propose a novel dynamic scheduling approach to further reduce the “expected” number of classifiers employed in prediction. It measures the confidence of a prediction by a subset of classifiers in the pruned ensemble. This confidence is used to decide if more classifiers are needed in order to produce a prediction that is the same as the original unpruned ensemble. This approach reduces the “expected” number of classifiers by another 25% to 75% without loss of accuracy.

Introduction

Recently, cost-sensitive learning has been extensively studied due to its wide application in areas such as fraud detection and intrusion detection. In such applications where examples carry different benefits and incur different penalties when misclassified, it is simply not enough to maximize the accuracy based on the assumption that all examples are equally important. Credit card fraud detection, for example, seeks to maximize the total benefit - the transaction amount of correctly detected frauds minus the cost to investigate all (correctly and incorrectly) predicted frauds. The problem of cost-sensitive learning is made more difficult by the fact that most real-world cost-sensitive datasets are very large and potentially distributed. A typical

credit card company has millions of new transactions on a daily basis and numerous local authorization centers nationwide. However, most traditional learning algorithms are designed for centralized and memory-resident data. (Fan *et al.* 2002) introduced a scalable method that fits well into the large-sized and distributed scenario. Their method computes a number of base classifiers from partitions of the entire dataset. The predictions by the base classifiers are combined by averaging to compute the final prediction. Their approach achieves linear speedup and linear scaled speedup relative to the number of partitions while producing a similar or even higher benefit when compared to a single classifier computed from the entire dataset as a whole. Another difference from traditional heuristic-based ensemble approaches is that the accuracy of averaging ensemble can be estimated by statistical sampling techniques from a subset of base models, so the performance is known before learning every base model. However, the satisfactory performance and properties of averaging ensembles is discounted by elongated process of prediction, since all base classifiers have to be consulted before a final prediction can be made. This paper addresses the issue of averaging ensemble pruning and scheduling in a cost-sensitive context.

Our Contributions

In this paper, we focus on using pruning methods to reduce the size of averaging ensembles in order to increase system throughput. We also introduce a new strategy, *dynamic scheduling*, for further prediction speedup.

Previously proposed pruning methods choose either the most accurate or the most diverse classifiers. Those proposals work well for *cost-insensitive* applications where all examples are equally important. We have tested previously proposed accuracy-based method using mean square error and diversity-based method using KL-distance in *cost-sensitive* applications. However, none of these methods yield pruned ensembles with good total benefits (for example, the total recovered fraudulent transaction amount). In our work, we use total benefits directly as the criterion. We have applied greedy algorithm with and without backfitting to classifier selection. Empirical evaluations on several datasets have revealed that the benefit-based

greedy approach with and without backfitting are both capable of pruning more than 90% of the ensemble while maintaining or exceeding the total benefits of the original unpruned ensemble on the test set. The more sophisticated backfitting does not lead to significantly higher benefits than greedy algorithm itself. We have also studied whether it helps to consider both diversity and total benefits at the same time. Experimental results show that total benefits is a good criterion by itself.

Another contribution is the proposal of dynamic scheduling. We find that not all the classifiers in the pruned ensemble are needed all the times for every example. Some examples are “easier” to predict than others. We propose a dynamic scheduling method that measures the confidence of a prediction by a subset of classifiers in the pruned ensemble and decides if more classifiers in the pruned ensemble need to be employed to generate a prediction that is likely to be the same as the original unpruned ensemble. In our experiment, the average or “expected” number of classifiers in the pruned ensemble is reduced by another 25% to 75% without loss of accuracy. Assuming each base classifier costs one unit of prediction time, the overall prediction throughput increases after pruning and dynamic scheduling is up to 40 ($=\frac{1}{10\%} \times \frac{1}{25\%}$) times at the best case. The complexity of benefit-based greedy algorithm is $O(k \cdot k' \cdot n)$, where n is the number of examples, k is the number of classifiers in the original ensemble and k' is the number of remaining classifiers in the pruned ensemble. More complicated methods (such as backfitting) and measures (such as KL-distance) will incur an additional term that is usually non-linear. The complexity of dynamic scheduling is $O(k' \cdot n)$.

Difference from Previous Work

Previous work mainly studies ensembles combined with either voting or meta-learning for cost-insensitive applications. (Margineantu & Dietterich 1997) first studied various pruning methods on AdaBoost ensembles. Their evaluation is mainly on *cost-insensitive* problems. We have adopted and tested several of their approaches for averaging ensemble in *cost-sensitive* applications; however, the results are not very satisfying. We have proposed new approaches using total benefits as a criterion and obtained significant improvement on both accuracy and efficiency. (Prodromidis 1999) proposed CART-like cost-based pruning methods to remove unnecessary classifiers in meta-learning trees.

The rest of the paper is organized as follows. Next we summarize the averaging method. We then discuss pruning criterion, classifier selection procedure, and scheduling. It is followed by experimental design and results.

Averaging Cost-Sensitive Ensembles

We briefly review how an averaging ensemble is constructed. Using credit card fraud detection for illustration. Each transaction x has a transaction amount of $y(x)$.

For undetected fraudulent transaction, the credit card issuer loses the whole transaction amount $y(x)$. Suppose that there is a fixed cost \$90 to investigate and dispute a detected fraud. In this case, the *optimal decision policy* is to predict x being fraudulent or positive iff $(R(x) = P(\text{fraud}|x) \cdot y(x)) > \90 , where $P(\text{fraud}|x)$ is the estimated probability that x is a fraud. $R(x)$ is called *risk*. Only those x 's with $y(x) > \$90$ are worthy of being detected. The benefit for true positives is $y(x) - \$90$ and the “benefit” for false positives is $-\$90$. For some applications such as charity donation, the benefit for each example such as donation amount is not known. A separate dataset with different feature set is given to train a model to estimate the amount $Y(x)$ when someone does donate. The averaging ensemble method solves the problem of how to estimate $R(x)$ efficiently for large and possibly distributed datasets.

Assume that the dataset is partitioned into k disjoint subsets S_i and a base classifier C_i is trained from each S_i . Given an example x , C_i outputs a member probability $P_i(x)$. $R_i(x) = P_i(x) \cdot y(x)$ is the individual risk. If the benefit of x is not given, a separate dataset is available and partitioned similarly into k subsets and a model (eg. regression) $Y_i(x)$ is trained from each subset. Using *Simple Averaging*, the combined risk is $\frac{\sum_i R_i(x)}{k} = \frac{\sum_i P_i(x) \cdot Y_i(x)}{k}$. The training complexity of averaging ensembles is $k \cdot O(f(\frac{n}{k}))$ and the speedup and scaled speedup are both linear $O(k)$.

Pruning Methods

Given an averaging ensemble of k classifiers and an upper bound $k' < k$, pruning is used to identify a subset of “good” k' classifiers which give the same level of total benefits as the original ensemble. Primary issues include pruning criteria and procedures of classifier selection. We also introduce a strategy that schedules the selected classifiers based on estimated prediction confidence to reduce the expected number of classifiers used in prediction by a pruned ensemble.

Pruning Criteria

We start with some notations. x is an instance; it may be either *positive* or *negative*. $P(x)$ (and its variations with subscripts) is the estimated probability that x is positive. $B(x)$ is the benefit to predict x to be positive; it only depends on the example x , not on any methods. The total benefit of a method is the cumulative benefits for a data set, i.e., $\sum_x B(x)$. C_i is a classifier and outputs probability $P_i(x)$ for example x . $S = \{C_1, \dots, C_k\}$ is the original ensemble, and S' is a pruned ensemble, $S' \subset S$, with k' base classifiers. The average probability estimate by an ensemble S for x is $P_S = \frac{\sum_{C_i \in S} P_i(x)}{|S|}$.

The first strategy is to choose a subset of classifiers that minimizes classification error. Probability estimate $P_{S'}(x)$ is compared with the true probability $P_T(x)$ (0 or 1). Use mean-square-error, the criterion is

$$MSE = \sqrt{\sum_x (P_{S'}(x) - P_T(x))^2}.$$

The second criterion directly uses total benefits. It chooses a subset of classifiers that maximizes the total benefits.

The third criterion favors diverse classifiers. KL-distance is a widely adopted measure. It may not be suitable for *cost-sensitive* learning as the difference between two classifiers is not reflected in the number of examples on which they disagree, but directly related to the benefits of those examples. We modify KL-distance for cost-sensitive scenarios. KL-distance is computed on *normalized* benefits, $\bar{B}(x) = \frac{B(x)}{\sum_x B(x)}$. Suppose the normalized benefits for x by two classifiers \mathcal{C}_i and \mathcal{C}_j are $\bar{B}_i(x)$ and $\bar{B}_j(x)$. The modified KL-distance is defined as $KL(\mathcal{C}_i, \mathcal{C}_j) = \sum_x \bar{B}_i(x) \log \frac{\bar{B}_i(x)}{\bar{B}_j(x)}$.

Classifier Selection

Combinatorial search will be too costly. Instead, we have applied greedy algorithm both with and without backfitting to decide the sequence of classifiers to consider. Greedy by itself always starts with the classifier that maximizes or minimizes the chosen criterion (KL-distance starts with the one that has the highest total benefit.) At the following steps, it chooses one of the remaining classifiers in $S - S'$ to join with previously chosen classifiers to generate a candidate pruned ensemble that maximizes or minimizes the chosen measure. In both benefit and accuracy based greedy approaches, when a new classifier is considered, combined probabilities $P_{S'}(x)$ are incrementally updated for n examples. The complexity is $O(n \cdot \sum_1^{k'} (k - i)) \approx O(k \cdot k' \cdot n)$ for small k' . However, in diversity-based greedy algorithm, the KL-distance between the candidate and every previously chosen classifier have to be computed and summed up. The complexity is $O(n \cdot \sum_1^{k'} (k - i) \cdot i) \approx O(k \cdot k'^2 \cdot n)$ for small k' .

Greedy is a local-optimum algorithm. We tested to use backfitting to avoid local optimum. It greedily chooses a classifier at each step, then perturbs the chosen ensemble by replacing some randomly chosen classifiers. When perturbed ensemble generates a higher total benefit, it replaces the older ensemble.

Dynamic Scheduling

A pruned ensemble aims at reaching or exceeding the level of total benefits of the original ensemble. We may not need to use *every classifier* in a pruned ensemble to predict on *every example* to reach the same total benefits. For example, if the average probability estimate by 2 classifiers is 0.6 (such as $\frac{0.7+0.5}{2}$), and the average probability estimate by the original ensemble with 256 classifiers is 0.6, they will make exactly the same prediction and we don't need to consult any additional classifiers. Actually, probability estimates by the conjectured 2 classifiers and 256 classifiers

even need not be the same to make the same predictions due to the *error tolerance* property of optimal decision making policy. Let us re-write this policy for the credit card fraud detection dataset, $P(x) > (T(x) = \frac{\$90}{y(x)})$. $T(x)$ is called *decision-threshold*. As long as, $P(x) > T(x)$ (or $P(x) \leq T(x)$), the prediction will be the same. Assume that $y(x) = \$900$, consequently $T(x) = 0.1$, both $P(x) = 0.2$ and $P(x) = 0.4$ will produce the same prediction. This property actually helps reduce the “expected” number of classifiers consulted for prediction.

For a given pruned ensemble S' with size k' , we first order the base classifiers into a “pipeline” according to their total benefits. To classify x , the classifier with the highest benefits will always be consulted first, followed by classifiers with decreasing total benefits. This pipeline procedure stops as soon as “a confident prediction” is made or there are no more classifiers in the pipeline. Assuming that $\mathcal{C}_1, \dots, \mathcal{C}_{k'}$ is the ordered classifiers. The set of classifiers at pipeline stage i is $S'_i = \{\mathcal{C}_1, \dots, \mathcal{C}_i\}$. Since the target is to reach the total benefits of the original ensemble, the confidence is calculated based on errors to the estimated probability $P_S(x)$ by the original ensemble. At pipeline stage i , the averaged probability for x is $P_{S'_i}(x)$. The error is simply $\epsilon_i(x) = P_{S'_i}(x) - P_S(x)$. In order to compute confidence, we divide the range of probability ($P_{S'_i}(x) \in [0, 1]$) into several bins and confidence is computed from examples in the same bin. We use $b(P_{S'_i}(x))$, or $\hat{b}_i(x)$ as a short form, to map $P_{S'_i}(x)$ to the bin it belongs to. We then calculate the average $\mu_i(\hat{b}_i(x))$ and variance $\sigma_i^2(\hat{b}_i(x))$ of error $\epsilon_i(x)$ for examples in the same bin $\hat{b}_i(x)$. These statistics measure the difference between $P_{S'_i}(x)$ and $P_S(x)$. To classify an unknown instance x , when $P_{S'_i}(x)$ is computed by the first i classifiers in the pipeline, we first determine the group it belongs to, $\hat{b}_i(x)$, then apply the following decision rules.

$$\begin{cases} (P_{S'_i}(x) - \mu_i(\hat{b}_i(x)) - \mathbf{t} \cdot \sigma_i(\hat{b}_i(x))) > T(x), & \text{positive} \\ (P_{S'_i}(x) - \mu_i(\hat{b}_i(x)) + \mathbf{t} \cdot \sigma_i(\hat{b}_i(x))) \leq T(x), & \text{negative} \\ \text{otherwise,} & \text{uncertain} \end{cases}$$

\mathbf{t} is a confidence interval parameter. Assuming normal distribution, $\mathbf{t} = 3$ has 99.7% confidence. When the prediction is uncertain, the next classifier in the pipeline (\mathcal{C}_{i+1}) has to be employed. If there are no more classifiers, the current prediction is returned. Thus, not all examples need to use all classifiers in the pipeline to compute a confident prediction, the “expected” number of classifiers can be reduced. Dynamic scheduling updates probabilities and group examples at each pipeline stage. The cost to compute confidence mainly comes from updating estimated probabilities for all n examples; the complexity to train a dynamic schedule is therefore $O(k' \cdot n)$. During classification, we map probabilities to confidence using a hash table. Decision trees output limited number of unique estimated probabilities since there are limited number of nodes. Besides binning, a more fine-grained approach is to group examples with the same value of $P_{S'_i}(x)$. However, some $P_{S'_i}(x)$ values computed at classification may not be seen at training time, the rea-

Dataset		rw	smth	ctl	ctl+smth
Donation	S	12489.6	14291.2	13292.7	14424.9
	E	14138.9	12483.0	14907.3	12994.4
Credit	S	552400	765009	733980	782545
	E	817869	705281	791024	730883
Adult	S	16225	15392	16443	15925
	E	16432	12764	16255	13695

Table 1: Total benefits by single model (S) and ensembles (E). If the result of the ensemble is higher than single model, it will be highlighted in **bold**.

son is that some particular combination of leaf nodes from different trees are not encountered on training data. If this happens, the pipeline runs to the end.

Experiments and Results

We evaluate the ensemble pruning approaches on three cost-sensitive problems: credit card fraud detection, donation and adult. Details about these datasets can be found in (Fan *et al.* 2002). Credit card dataset has been discussed when introducing averaging ensembles. In donation dataset, suppose that the cost of requesting a charitable donation from an individual x is \$0.68 and the best estimate of the amount that x might donate is $Y(x)$. The optimal decision to solicit x iff $P(\text{donate}|x)Y(x) > \0.68 . The total benefit is the total charity received minus the cost of mailing. In adult dataset, positive examples carry a benefit of 2 and negatives carry a benefit of 1. There is no penalty for misclassification. The optimal decision is to predict x to be positive iff $P(\text{positive}|x) \cdot 2 > (1 - P(\text{positive}|x)) \cdot 1$.

The base classifiers in the ensembles are C4.5 decision trees that output calibrated probabilities (Zadrozny & Elkan 2001): raw (rw), smoothing (smth), curtailment (ctl) or curtailment plus smoothing (ctl+smth). The size of the original ensemble or the original number of base classifiers in the ensemble was chosen to be, $k \in \{64, 128, 256\}$. For each dataset, we apply different pruning techniques on the 12 ensembles (3 ensemble sizes (k) \times 4 types of probability outputs.)

First, we report the baseline benefits that should be maintained after pruning. In Table 1, we cite the total benefits of both the averaging ensembles (E) and the single models (S). The ensemble’s result (E) is the average total benefits over 3 ensembles with different sizes ($k \in \{64, 128, 256\}$). When the total benefit of the ensembles is higher than that of the single model, it will be highlighted in **bold** font. The averaging ensemble generally produces a higher total benefit with raw and curtailed probabilities, and their results are the highest among all three datasets.

Next, we discuss the experiments and results of the pruning methods. The number of classifiers selected k' by pruning is chosen to be $\frac{i}{32}$ -th ($i \in \{1, 2, 3, 4\}$) (roughly 3%, 6%, 9% and 12%) of the original size of ensembles k , where $k \in \{64, 128, 256\}$. The pruning algorithm chooses

Methods	rw	smth	ctl	ctl+smth
Accuracy	11812.8	13147.6	14120.4	14046.9
Diversity	8148.4	11295.3	12013.9	12048.5
Benefit	12965.2 \sqrt	13281.6	14642.8 \sqrt	14171.6

(a) Donation

Methods	rw	smth	ctl	ctl+smth
Accuracy	757382	715740	792291	735731
Diversity	752203	722291	799461	746575
Benefit	776838 \sqrt	730516	809212\sqrt	752473

(b) Credit Card

Methods	rw	smth	ctl	ctl+smth
Accuracy	16362	12884	16242	15567
Diversity	16092	12611	15902	14973
Benefit	16362 \sqrt	13299	16381	15817

(c) Adult

Table 2: Total benefits of pruning on different datasets. For each column of competitive methods, the highest result is highlighted. If the result is higher than the original unpruned ensemble (as shown in Table 1), it will be highlighted in **bold** font; otherwise if it is higher than at least the single classifier, it is marked with a \sqrt .

classifier based on their performance on training data, and the pruned ensembles are compared on their total benefits on test data. The results are shown in Table 2. Each number is the average benefits of 12 pruned ensembles (4 pruned ensemble sizes (k') \times 3 original (k) ensembles.) In each column where competitive pruning methods were applied on the same data with the same probability outputs, the highest pruning result is highlighted as follows. If the result is higher than the respective unpruned ensemble baseline (as shown in Table 1), it will be shown in **bold** font; otherwise, if it is higher than the single classifier (Table 1), it will be followed by a \sqrt .

As we study the results, “benefit-based greedy” method consistently beats the other pruning methods; all the highlighted results in Table 2 are by benefit-based greedy algorithm. Every pruned ensemble produced by benefit-based greedy method has higher benefits than either the original ensemble or the single model or both. Comparing among different probability estimation methods (raw, smoothing and etc), pruning works the best with curtailment. For both donation and credit card dataset, the pruned curtailment ensemble has the second highest benefit (very close to the highest) among all ensembles and single models. One very interesting observation is that pruning significantly increases the benefits of ensemble with smoothing. In general, the ensemble method is able to increase benefits for ensembles with raw and curtailed probabilities, but this ability fails with smoothed probabilities. However, benefit-based greedy algorithm has the capability to choose the good set of classifiers to obtain high

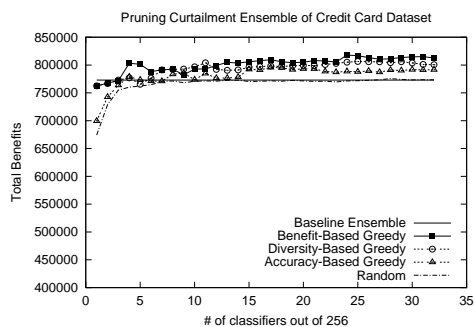


Figure 1: Pruning Effect on Credit Card Curtailment Ensembles

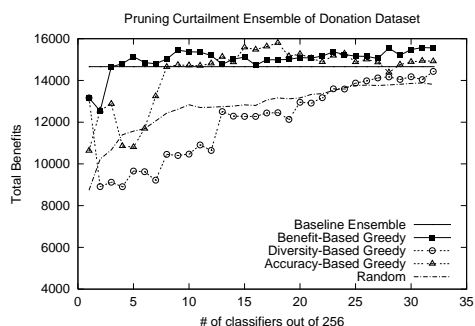


Figure 2: Pruning Effect on Donation Curtailment Ensembles

benefits. As a summary, benefit-based greedy approach can remove more than 90% of the base classifiers while maintaining or exceeding the accuracy of the unpruned ensemble.

One important study is to examine how well pruning works with ensembles containing a large number of base classifiers whose prediction is slower than ensembles with relatively smaller number of classifiers. We show result for credit card and donation datasets in Figures 1 and 2. (The result of adult is similar.) The original ensemble has $k = 256$ base classifiers. Pruning chooses continuously up to $k' = 32$ classifiers. As a comparison, we also plot the result of a “random selection” method that chooses classifiers arbitrarily. The random result is the average of 10 random choices. As we compare all the curves in Figures 1 and 2, an obvious observation is that random selection doesn’t work well. The performance increase by benefit-based greedy method over baseline of the original ensemble and other pruning methods (accuracy and diversity) are shown clearly throughout the different choices of $k' \leq 32$. In the donation dataset, benefit-based greedy method already reaches the accuracy of the original ensemble when $k' = 3$ out of $k = 256$. The benefit still increases when more classifiers are chosen. When $k' = 31$, the total benefit is \$15573, which is significantly higher than the KDD98 Cup winning entry of \$14712.

Besides these basic methods and experiments, we experimented other variations. In one trial, we applied backfitting

to benefit-based greedy approach to seek further improvement. The maximal number of classifiers allowed to be replaced by backfitting is set to be 12. The additional effort on backfitting does not bring better results. In addition, we also try to consider both total benefits and diversity in the same time. To choose the next classifier in this new method, first the top 10 candidate classifiers (chosen from $k - k'$ remaining classifiers) are selected. At the second step, we select the one with the greatest KL-distance. However, this more sophisticated method brings no significant difference. These experiments strengthen our observation that benefit-based greedy algorithm is a promising pruning method for cost-sensitive ensembles.

When a subset of classifiers is chosen, we apply dynamic scheduling to reduce the “expected” number of classifier even further, as shown in Figure 3, where x-axis is the number of classifiers after pruning by benefit-based greedy method, or the expected or average number of classifiers consulted per example for dynamic scheduling. The curve for dynamic scheduling is obtained as follows. When dynamic scheduling is applied on a pruned ensemble with k' base classifiers, many examples from the test data uses $k'' < k'$ classifiers instead of k' classifiers. We calculate both the expected number of classifiers used when a prediction is confidently made and the resultant total benefits. These two numbers are plotted in Figure 3. The visual effect of dynamic scheduling is to “compress and shift” the curve of benefit-based greedy method towards the left or smaller number of classifiers. To see some detailed result, after dynamic scheduling is applied, on average, using 8 classifiers gives similar total benefits as using 32 classifiers all the time. Similar results are obtained on the other two datasets. As a summary, when there are totally 32 classifiers to be scheduled, dynamic scheduling cut the average number of classifiers to 75% for credit card dataset, 33% for adult and 25% for donation.

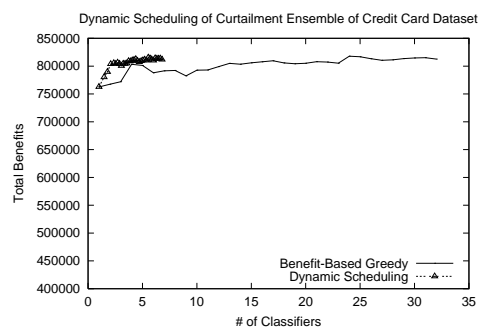


Figure 3: Total benefits of pruned credit card curtailment ensemble when dynamic scheduling is employed. X-axis is the number of classifiers after pruning for benefit-based greedy method, or the average number of classifiers consulted per example for dynamic scheduling.

Discussion and Future Work

As a sanity check, we ran the benefit-based greedy method up to $k' = k = 256$ for the credit card dataset as shown in

Figure 4. The curves on donation and adult are very similar. There are a few important observations. First, when $k' = 256$, the result is the same as the original ensemble. Second, even the most accurate base classifier (with x -axis value of 1) by itself is less accurate than the original ensemble, implying that ensemble is necessary. The curve resembles an “overfitting” curve that researchers in machine learning always use. It takes only three classifiers to reach the accuracy of the 256 classifier ensemble and the total benefit continues to increase as more classifiers are inserted, and begins to drop after about 50 classifiers until eventually it converges to the baseline. A conjectured reason for the shape of the curve is that there may be random noises from multiple sources. Averaging can cancel uncorrelated errors from the same source. Some classifiers may have errors from unique sources. In the beginning, errors from the same sources start to cancel out, and consequently the accuracy starts to increase. When more classifiers are introduced, more errors from unique sources that cannot be canceled by averaging are also added. These errors accumulate and decrease the overall accuracy eventually. More formal study is needed to explain this phenomenon and verify the conjecture. Nonetheless, the “overfitting” curve validates the necessity of pruning for both accuracy and efficiency reasons.

Both pruning and dynamic scheduling are performed after learning the complete ensemble. A possibly more efficient approach is to generate only the subset of classifiers that would not be pruned. One possibility is to apply boosting and bagging-like approaches that choose the subset of examples based on the performance of previously generated classifiers.

Related Work

Besides related work discussed in introduction, one of the first work to scale up learning by combining class labels is meta-learning (Chan 1996). Other incomplete list of significant contributions to scalable learning (mostly are on learning class labels or cost-insensitive learning) are scalable decision tree learner SPRINT (Shafer, Agrawl, & Mehta 1996), rule-based algorithm for multi-processor learning DRL (Provost & Hennesy 1996) among many others. There

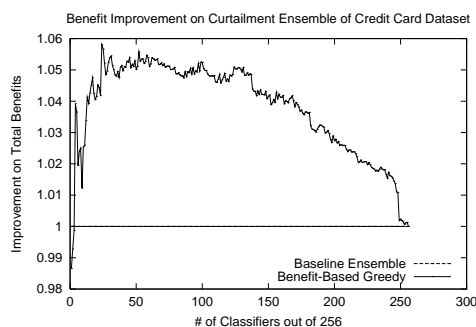


Figure 4: “Full spectrum” sanity check on credit card dataset with curtailed probabilities. Result is normalized over the total benefits of the original ensemble with 256 classifiers

are many recent publications on cost-sensitive learning, Domingos (Domingos 1999) converts a cost-sensitive problem into an equivalent cost-insensitive problem by mislabelling. A recent workshop on cost-sensitive learning can be found in (Dietterich *et al.* 2000).

Conclusion

Our studies have found that benefit-based greedy pruning can remove more than 90% of the base classifiers but with the same or higher accuracy than the original ensemble. We also found out that when dynamic scheduling is applied to a pruned ensemble, the expected number of classifiers used can be reduced by another 25% to 75% without loss of accuracy. Based on our studies, we conclude that benefit-based greedy approach plus dynamic scheduling is an effective and efficient approach to prune cost-sensitive averaging ensembles. In comparison with a single model trained from the entire dataset, pruned averaging ensemble achieves significantly higher learning efficiency, the same or better accuracy and comparable prediction efficiency.

References

- Chan, P. 1996. *An Extensible Meta-learning Approach for Scalable and Accurate Inductive Learning*. Ph.D. Dissertation, Columbia University.
- Dietterich, T.; Margineatu, D.; Provost, F.; and Turney, P., eds. 2000. *Cost-Sensitive Learning Workshop (ICML-00)*.
- Domingos, P. 1999. MetaCost: a general method for making classifiers cost-sensitive. In *Proceedings of Fifth International Conference on Knowledge Discovery and Data Mining (KDD-99)*.
- Fan, W.; Wang, H.; Yu, P. S.; and Stolfo, S. 2002. A framework for scalable cost-sensitive learning based on combining probabilities and benefits. In *Second SIAM International Conference on Data Mining (SDM2002)*.
- Margineantu, D., and Dietterich, T. 1997. Pruning adaptive boosting. In *Proceedings of Fourteenth International Conference on Machine Learning (ICML-97)*.
- Prodromidis, A. 1999. *Management of Intelligent Learning Agents in Distributed Data Mining Systems*. Ph.D. Dissertation, Columbia University.
- Provost, F. J., and Hennesy, D. 1996. Scaling up: Distributed machine learning with cooperation. In *Proceedings of Thirteenth National Conference on Artificial Intelligence (AAAI-96)*.
- Shafer, J.; Agrawl, R.; and Mehta, M. 1996. SPRINT: A scalable parallel classifier for data mining. In *Proceedings of Twenty-second International Conference on Very Large Databases (VLDB-96)*, 544–555. San Francisco, California: Morgan Kaufmann.
- Zadrozny, B., and Elkan, C. 2001. Obtaining calibrated probability estimates from decision trees and naive bayesian classifiers. In *Proceedings of Eighteenth International Conference on Machine Learning (ICML'2001)*.