When Efficient Model Averaging Out-Performs Boosting and Bagging

Ian Davidson[1] and Wei Fan[2]

[1] Department of Computer Science, University at Albany - State University of New York, Albany, NY 12222. Email: davidson@cs.albany.edu.
[2] IBM T.J. Watson, 19 Skyline Drive, Hawthorne, NY 10532.
Email: weifan@us.ibm.com

Abstract. Bayesian model averaging also known as the Bayes optimal classifier (BOC) is an ensemble technique used extensively in the statistics literature. However, compared to other ensemble techniques such as bagging and boosting, BOC is less known and rarely used in data mining. This is partly due to model averaging being perceived as being inefficient and because bagging and boosting consistently outperforms a single model, which raises the question: "Do we even need BOC in datamining?". We show that the answer to this question is "yes" by illustrating that several recent *efficient* model averaging approaches *can* significantly outperform bagging and boosting in realistic difficult situations such as extensive class label noise, sample selection bias and many-class problems. To our knowledge the insights that model averaging can outperform bagging and boosting in these situations has not been published in the machine learning, mining or statistical communities.

1 Introduction and Motivation

The typical aim of classification data mining is to build the most accurate model. Research activity in the machine learning and data mining fields has produced a large number of ensemble techniques such as boosting and bagging that can combine multiple base models to increase accuracy beyond the single best model. However, the use of the classic Bayesian statistical approach to multiple models, Bayesian model averaging, is rarely reported in data mining and even when it is, yields indifferent results [2, 6]. The Bayesian approach of using multiple models is to weight each model's belief in a prediction ($P(y_i|\mathbf{x}, \theta)$) by the model's posterior probability ($P(\theta|D)$) and is known as Bayesian model averaging in the statistical literature [16] and the Bayes optimal classifier (BOC) in the machine learning and data mining literature [19]. Given a test set instance, $\mathbf{x}$, to be classified into one of k classes ($y_1...y_k$), a training dataset D and model space Θ the BOC chooses the class that maximizes equation (1).

$$\text{argmax}_{y_i} : P(y_i) = \int_{\theta \in \Theta} P(y_i|\mathbf{x}, \theta) P(\theta|D) d\theta \tag{1}$$

The BOC is claimed to be optimal for a number of reasons: Kohavi claims it reduces bias and variance [17] while Buntine claims it prevents overfitting [2] and

Domingos [6] states “no other method can consistently achieve lower error rates“. In addition there is significant experimental evidence supporting its effectiveness in the statistical literature [16]. For example, a reply comment in [16] is: *“In problems where model uncertainty is present, I have found model averaging to consistently yield predictive performance improvements over single selected models. I have also found this phenomenon to persist under a wide variety of prior choices. Many colleagues have also reported similar experiences.”.*

Yet there seems to be little interest in using model averaging in data mining. Instead the two most common ensemble model techniques are bagging [1] and boosting [14]. These techniques are rightfully extensively used as they are relatively easy to implement, have been shown to work for a variety of learners and a variety of data sets. In contrast model averaging is typically more difficult to implement, particularly since BOC in its rudimentary form requires performing an integration over the entire model space which is computationally prohibitive for the *complex* model spaces such as trees and large datasets used in the data mining.

Compared to model averaging, ensemble techniques such as bagging and boosting **combine** not average and the base unit is a **vote** not a probability. Minka [20] succinctly notes the difference that BOC is a method of “soft model selection” using multiple models when there is *uncertainty* to which model is the best model. For example, if all posterior probabilities are the same then model uncertainty is maximum. In contrast techniques such as bagging and boosting are methods to *combine* multiple models to create a new (and potentially more appropriate) model space than the base model space [20].

Which type of technique (averaging or combining) will work best depends on the properties of the target concept and the data set available to find it. Consider the circles example in the Minka paper reproduced in Figure 1. The base model class consists of just these three circles (inside the circle predicts ’o’, outside ’x’). The top most circle has the greatest single model accuracy misclassifying only a small number of class ’x’ as ’o’ compared to the other two models. However, given only a small number of data points a model averaging approach would weight each model approximately equally (which is the strategy that maximizes accuracy) because equation 1 integrates over **all** models. Since model combination strategies have no such requirement they may choose only a subset of the three available models and hence not uniformly vote amongst all models leading to a sub-optimal result. We shall show that model averaging can outperform the model combination approaches of bagging and boosting when there is significant model uncertainty and averaging out model uncertainty is desirable. In later sections we shall discuss and experimentally explore realistic difficult situations where this occurs such as many-class problems and extensive class label noise. These results may seem to contradict the results of earlier work such as [6], but we note that we purposefully choose data sets and conditions when model uncertainty is great and this is not the case for other studies.

The aim of this paper is two fold. Firstly, we intend to explore if when model uncertainty occurs model averaging outperform bagging and boosting. Secondly, we will compare the computational efficiency of several recent efficient model averaging techniques against other ensemble techniques. We begin this paper by

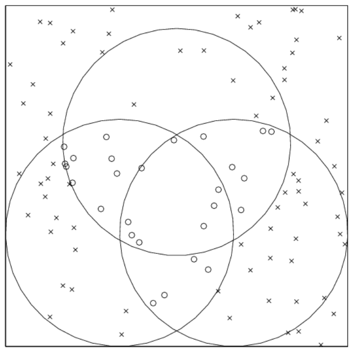

Fig. 1. A simple classification problem where the target concept is: 'o' if the instance falls inside at least two circles, otherwise 'x'. A uniform vote amongst all three models is an optimal prediction strategy.

discussing when model uncertainty is most likely to exist for various properties of data sets in section 2. We then describe two recent efficient approaches to model averaging the authors have created *Random Decision Trees* [10] and *Parametric Bootstrap Model Averaging* [3]. We then conduct extensive experiments to test if model averaging can outperform boosting and bagging in the situations outlined in section 2.

2 Why and When Averaging Will Outperform Combining

In this section we compare and contrast model averaging and model combination and conclude the section by describing data set conditions where model averaging should perform better than model combination. In later sections we empirically test these claims.

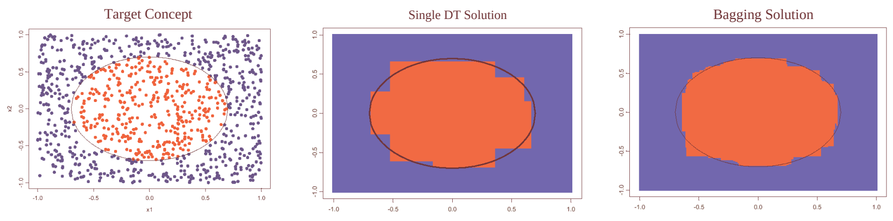

Fig. 2. A problem where bagging multiple trees better approximates the target concept than a single tree. Target concept is inside and on the ellipse is one class and outside another. Reproduced from Andrew Moore AUTONLAB tutorial site.

It may appear the model averaging and techniques such as boosting and bagging are similar but how the multiple models are used are quite different. Many ensemble techniques explicitly or implicitly combine multiple models. For example, the serial nature of boosting to focus on misclassified instances *effectively* means that the original tree built from the unweighted training set has subsequent additional trees grafted onto the leaf nodes containing many misclassified instances. In that way the model space that boosting is searching is a combination of the base model class (trees). Similarly, it has been noted that bagging models can create a new model space which is more expressive than the base model space. Consider the example given in Figure 2. The base model class of trees, can only find a decision surface such that one side is parallel to one of the dimensions. This does not approximate well the true decision surface of an ellipse. However, bagging many trees results in a significantly better approximation.

In contrast model averaging never explicitly combines models. Rather, each model's predictions are weighted by the belief (posterior) that it is the true model. In this way, though the final predictions may be different than any single model in the entire model class [19], the model class/space is no more complex than the original. Furthermore, in the presence of excessive number of training instances equation 1 will simply have most of the posterior mass on a single model and perform no better than a single tree. As the statistical literature [16] and Minka [20] note model averaging should perform well when there is substantial model uncertainty where best-model uncertainty can be quantified as being one minus the posterior probability of the most probable model. We note that there are other general measures of model uncertainty such as the entropy in the posterior [3]. Formally:

Definition 1. ***Best-Model Uncertainty*** *is the degree of belief that the most probable model (θ^*) is* **not** *the best model in the model class (Θ). That is:*
ModelUncertainty$(\Theta, D) = argmin_{\theta \in \Theta}\ (1 - P(\theta|D))$

When there exists considerable model uncertainty in the original model space it is not advisable to combine multiple base models to get an even more complex model. When model uncertainty exists, it is because there is insufficient data to conclusively state that one model is the best and hence building combinations of models can be perceived as building an overly complex model given the amount of data.

We see from definition 1 that model uncertainty is likely to exists if there is no highly probable model. By a simple expansion of $P(\theta|D) = \frac{P(\theta)P(D|\theta)}{P(D)}$ using Bayes theorem we see that this can occur if the numerator, particularly the likelihood is small. In decision tree learning this can occur for a number of reasons. Since each tree path forms a distinct part of a model we can say that $P(D|\theta) = \Pi_{1\ldots n} P(D_i|Leaf(D_i))$ where $Leaf(D_i)$ is a function returning the leaf the i^{th} instance is assigned to. The term $P(D_i|Leaf(D_i)) = \frac{Count(Class(D_i), Leaf(D_i))}{Number(Leaf(D_i))}$ where $Count(Class(D_i), Leaf(D_i))$ returns the number of training set instances having the same label as D_i and $Number(Leaf(D_i))$ the total number of instances at the leaf node. The leaf nodes of the tree may be poorly or incorrectly populated for any number of reasons. The (non-exhaustive) set of data set conditions we believe where this will occur and shall explore in this paper are:

- Training sets with excessive class label errors (i.e. security/fraud applications where labeling is difficult)

– High dimensional data but a relatively few number of instances (i.e. bioinformatics problems)
– Multi-class problems involving a large number of classes (i.e. fault diagnosis)

In addition, even if model uncertainty is not particularly great, it may be worth removing model uncertainty by averaging. In this paper we shall explore a situation discussed in our recent work, sample selection bias [8]. In this situation, since the training and test data sets are drawn from different distributions, even a highly probable model for the training set may produce poor predictions on the test set.

3 Efficient Model Averaging Techniques

In this section we shall describe our efficient model averaging techniques for completeness since they are not as well known as boosting and bagging. If the reader understands that our techniques *Random Decision Trees* (RDT) and *Parametric Bootstrap Model Averaging* (PBMA) are efficient approaches to model averaging then this section can be skipped without loss of flow in the first reading of this paper.

3.1 Random Decision Trees (RDT)

RDT were originally proposed in [10, 9]. The construction of RDT is significantly different from the construction of regular decision trees as RDT constructs multiple decision trees randomly. Rather than using purity functions (i.e. information gain) like traditional decision tree algorithms, RDT chooses a feature/column to split on *randomly*. A discrete feature is chosen only once in the decision path from the root of the tree to the current node. A continuous feature can be chosen multiple times in the same decision path with a different decision threshold each time. The tree stops growing if either the current node becomes empty or the depth of the tree exceeds some predefined limit. Since both feature and decision thresholds for continuous features are chosen randomly, each RDT is likely to be different. In our experiments, the depth of the tree is limited to be up to the number of features in order to give each feature equal opportunity to be chosen in any decision path.

During classification, each random tree computes a posterior probability at the leaf node. That is, if there are n examples at a leaf node and q_i belong to class label y_i, the posterior probability $P(y_i|\mathbf{x}, \theta) = \frac{q_i}{n}$. For a given test instance, the posterior probability outputs from multiple decision trees are **averaged** to produce a final probability estimate with the most probable class for an instance being the prediction. As found in [10], typically 30 random trees give satisfactory results and as little as 10 random trees produce results better than using a single traditionally grown decision tree [9]. Although quite different from well-accepted methods that employ purity split functions to construct decision trees, random trees have been shown to have accuracy comparable to or higher than bagging and random forest but at a fraction of the training cost for a variety of data sets including those with many irrelevant attributes [10, 12]. RDT are grown to a fix depth (the number of attributes in the training data set) and hence are not pruned. Though with RDT each *tree structure* is created randomly the leaf probabilities are estimated from the training data set. Therefore, in the limit as the number of trees approaches infinity, RDT effectively computes the following:

$$P(y_i|\mathbf{x}, \Theta) = \sum_{\theta \in \Theta} P(\theta|D).\frac{q_{i,\theta}}{n_\theta} \tag{2}$$

RDT though superficially similar to random forest of trees (RFT) are different in a number of ways (see [12] for details). Furthermore, RDT are different to the random trees referred to by Dietterich [4] as in that work the splits are randomly chosen from the twenty most informative splits and it is quite feasible that many of these twenty choices are for the same continuous attribute.

3.2 Parametric Bootstrap Model Averaging

The bootstrap model averaging approach [3] is a frequentist approach to model averaging. The philosophy of model averaging is to remove model uncertainty that arises due to **a single finite data set**. Consider the typical view of mining where a single training set of size n is available from the underlying distribution that generated the data F. However, this view masks the underlying uncertainty in the data, namely that the training data we have is one of many that could have been chosen/generated. If we were to build a model for each possible data set we would have a probability distribution over the model space. This probability distribution is the frequentist analog to the posterior distribution and gives the model uncertainty due to data set fluctuations. It differs from the Bayesian posterior since no prior distribution over the model space is used in its calculations.

However, typically we do not have the luxury of many different data sets drawn independently of the same size so we can not compute the uncertainty over the model space from them. To approximate this distribution using a *single data set*, Efron [7] created the non-parametric and parametric bootstrapping approaches. Non-parametric bootstrapping which has been used extensively by the machine learning community is an example of attempting to simulate draws from F when no knowledge of its form is known. Parametric bootstrapping which has received little attention is used when some underlying knowledge on the form of F is known. In this paper we make use of a simple generative parametric model of F which assumes that the independent variables are conditionally independent given the dependent variable value. The parameters for this model are estimated from the data and virtual training examples are drawn/bootstrapped from it to build models. More complex models of F could be used and remains an active research area. Formally the bootstrap model averaging approach is shown in equation 3. In this paper, the learner, L, used with PBMA is J48 (Weka's equivalent to C4.5) to produce trees built with the default parameters.

$$\mathrm{P}(y_i|\mathbf{x}, \Theta) = \sum_{D', \theta_i = L(D')} P(y_i|\mathbf{x}, \theta_i) P(D'|D) \tag{3}$$

4 Experiments

We now illustrate the performance of model averaging techniques and other ensemble techniques when considering model uncertainty is beneficial (see section 2). We find that bagging and boosting does not perform well in these circumstances.

4.1 Experiment Methodologies

Class Noise and Biased Sample Experiments We being by illustrating the efficiency of PBMA and RDT by using only 30 models but use 100 models for bagging and boosting. We use the Weka software to perform bagging and AdaBoost using the J48 decision tree base learner (equivalent to C4.5). This requires changing Weka slightly as the default implementation of bagging in Weka aggregates conditional probabilities and *not* votes [13]. In all experiments, regardless of the ensemble technique (PBMA, Boosting, Bagging), the default parameters were used for J48. We randomly divide the data into 80% for training and the remaining 20% for testing unless otherwise stated. We investigate the following situations 1) where removing model uncertainty due to noise in the class labels is beneficial and 2) when the training data set is biased and hence the most probable model happens to be wrong. Experiments are repeated 100 times.

Large Number of Classes Experiments We tested bagging, boosting, RDT and PBMA on four data sets that contain large number of classes. We tried building, 100, 250 and 500 trees and calculated the ten-fold cross-validated accuracy ten times. A pair-wise student t-test using the same folds, shows that the poorer performing model average technique, PBMA, outperforms both bagging and boosting for all data stets at the 99% confidence interval.

4.2 Model Uncertainty Due to Class Label Noise

If there exists a deterministic relationship between the independent variables and class label, and the model space is suitable, it is likely any reasonable learner will find a highly probable model that can correctly map feature vectors of independent variables to their correct class labels. Hence the model uncertainty would be relatively small. However, the addition of class label noise, i.e., the correct class label being flipped to another class, would almost certainly reduce the probability of the "correct model" otherwise trained from dataset without label noise, hence increasing model uncertainty. We took several UCI data sets and introduced 20% class label noise in the training sets only by randomly flipping 20% of class labels. We then try the model averaging approaches and compare them against the other ensemble techniques. Table 1 shows that the model averaging techniques outperform the other ensemble techniques and single models in each situation. A pair-wise student t-test using the same training and test divisions, shows that the poorer performing model average technique, PBMA, outperforms both bagging and boosting at the 99% confidence interval for all data sets.

As expected [14] boosting performs quite poorly as it apparently continues to build models to fit the added noise while bagging can only occasionally significantly outperform a single model.

	Breast	**Vote**	**Pima**	**Credit**	**Wine**
UP. Tree	10.6	5.2	31.8	36.5	37.6
P. Tree	4.5	12.6	31.8	37.4	37.0
RDT	**0.5**	3.7	**30.0**	**30.6**	**26.4**
PBMA	2.5	**2.2**	30.9	30.7	26.7
Bagging	4.0	4.2	33.9	34.3	34.3
Boosting	15.7	19.3	34.0	42.4	38.2

Table 1. Average error of various techniques on standard UCI data sets with class label noise in training set only over one hundred repetitions. UP=Unpruned, P=Pruned

4.3 Biased Training Sets

Recent work by Zadrozny and ourselves [26, 8] has classified many learners including decision trees and naive Bayes as global learners that can be effected by sample bias. In particular the sample bias investigated is that the probability of the instance being selected in the training data (denoted by the event, $s = 1$) is conditionally independent of the instance label (y) given the instance's description (x), formally: $P(s = 1|x, y) = P(s = 1|x)$. This type of sample bias is effectively when instances are not chosen randomly but instead depend on their description but not their label. It occurs prevalently in applications where the occurrence of particular instances' change but not the concept (i.e. relationship between x and y). For the rest of this paper when we refer to **sample bias**, we refer to this type of bias. Decision trees are known to be unstable, and is categorized as "global" classifier in [26]. The structure of decision tree is sensitive to sample selection bias, which makes it unlikely to find that most probable decision tree otherwise trained from dataset without sample selection bias.

Artificial Sample Selection Bias We can artificially introduce sample bias into the previously mentioned UCI data sets by first dividing the data into training and test sets. We then sort only the training set on the attributes and remove the first 10% of all instances. Note this introduces a training sample bias but does not change the relationship between the independent and class-labels as our previous experiments did. Table 2 shows that model averaging performs better than other ensemble techniques even though only 30 models are used and 100 models are used for bagging and boosting. Unlike the previous situation, boosting performs better than bagging on average **but both perform worse than a single tree**.

Sample Selection Bias in Newsgroup Classification We now focus on the newsgroup data where the training and test data sets are drawn from similar but not identical distributions. We perform experiments on the 20 Newsgroups [22] datasets using the standard bydate version division into training (60%) and test (40%) based on posting date. The division creates a temporal bias. For example, in the GUNS newsgroup the word "Waco" occurs extensively in news articles in the training set but not in the test set as interest in the topic fades.

	Breast	**Vote**	**Pima**	**Credit**	**Wine**
UP. Tree	1.5	3.7	26.5	3.1	35.3
P. Tree	2.1	3.7	27.6	12.2	35.3
RDT	**0.5**	**2.7**	26.1	**1.0**	**27.8**
PBMA	1.0	3.1	**25.7**	1.3	28.0
Bagging	4.04	3.7	28.0	5.61	38.2
Boosting	2.5	4.4	28.4	5.02	38.1

Table 2. Error of various techniques on UCI data sets with biased training sets over ten repetitions. UP=Unpruned, P=Pruned

However, since the proportion of each class label is the **same** in the training and test data sets there is no class label bias. We used the tool Rainbow [18] to extract features from these news articles. The feature vector for a document consists of the frequencies of top ten words by selecting words with highest average mutual information with the class variable. To better understand the performance of model averaging we treat the problem as binary classification between a variety of newsgroups. Table 3 illustrates the improvement that model averaging provides. Again, the better performance is obtained using only 30 models for the model averaging approaches as opposed to 100 models for bagging and boosting. In this situation boosting performs better than bagging but the model averaging approaches outperform both and the single best model.

	BaseBall Hockey	**Christian Sale**	**M.Cycle Guns**	**MidEast Guns**	**MidEast Electrical**	**Mac. Religion**
Unpruned Tree	15.7	7.9	10.5	20.3	14.4	18.4
Pruned Tree	15.7	7.4	10.2	20.3	14.4	18.7
RDT	**11.9**	6.6	**8.5**	10.5	**7.2**	12.7
PBMA	12.0	**6.0**	8.6	**9.8**	7.4	**11.9**
Bagging	14.8	7.9	10.5	20.3	14.4	18.4
Boosting	12.6	7.7	9.2	10.7	11.7	13.2

Table 3. Error of various techniques on newsgroup data. Training set is first 60% of the year's postings and the test set the last 40%.

4.4 Model Uncertainty Due to Large Numbers of Classes

We examine four data sets where the number of classes is large: Digit (10), Glass (6), Protein (6), Soybean (19). Furthermore, the number of columns is relatively high and instances relatively small. For each data set the number of columns and instances is: Digit (17, 469), Glass (10, 214), Protein (21, 116) and Soybean (36, 683) respectively. Assuming a uniform distribution of classes amongst the instances there are as little as 19 instances per class (not leaf) and hence model uncertainty is likely to be high. When comparing the model combination approaches bagging, boosting to the model averaging approaches RDT and PBMA we find that the averaging approaches work significantly better except for the Glass data set when only PBMA outperform both combination techniques (see Figure 3).

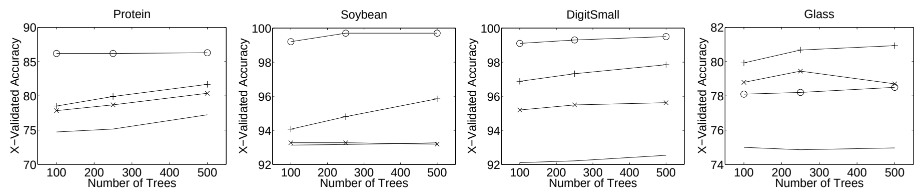

Fig. 3. Plots of Accuracy against number of trees for Bagging (-), Boosting (-x-), RDT (-o-) and PBMA (-+-).

5 Discussion

It is often stated that bagging and boosting are approaches to model averaging, but it is better to say that they are approaches to combine multiple models. Only BOC of the three ensemble approaches is performing averaging in that equation 1 explicitly calculates the expectation (average) of the class label conditional probability over the posterior distribution. Furthermore, considerable experimental evidence suggests that bagging is *not* an approximation to the BOC [5]. We have already discussed why model averaging should work in our situations earlier (see section 2) but not why model combining techniques should perform relatively poorly.

In our experiments boosting performed poorly when there existed class label noise while bagging performed poorly when there was sample bias. Boosting can be interpreted as performing various functions. In this paper we shall use the popular view that boosting is a technique to maximize the margins [14]. In particular boosting, tends to increase the margins of training examples by focusing on those with the smallest margin. However, with a noisy class labels some of the instances with small margins may by incorrectly labeled and hence focusing on them produces poor results.

It is interesting to consider that bagging which uses non-parametric bootstrapping faired poorly on problems with sample bias but PBMA which uses parametric bootstrapping did not. A viable explanation comes from the bootstrapping literature [7]. Non-parametric bootstrapping by continually resampling attempts to create an empirical estimate of F, the distribution the training data is drawn from, using the (biased) training data. Parametric bootstrapping instead draws virtual instances, that could be quite different from the training data set, after creating a parametric model of F. If this parametric model can at least partially correct the sample bias then it should perform better than its non-parametric counterpart.

6 Conclusion

Model averaging is a method of choice for statisticians to make use of multiple models. However, model averaging is rarely used in data mining and instead other ensemble techniques such as boosting and bagging are used extensively. We believe this is for two primary reasons: a) Model averaging is perceived as being computationally inefficient and b) boosting and bagging have been shown to work for a variety of problems. A valid question is then: "Does the data mining practioner require model averaging as an ensemble technique?".

The answer to this question is yes, there are situations where model averaging can significantly outperform both bagging and boosting using two recent efficient approaches to model averaging. This is because a) boosting and bagging are model *combination* not averaging approaches and b) model averaging works best when there is model uncertainty. The situations where averaging to remove model uncertainty is beneficial we explored in this paper include when the number of classes is large, excessive class label noise and training sample bias occurs.. We believe that these are not academic situations but will occur in fields such as bioinformatics and security and demonstrated that one of the situations (training sample bias) occurs in the newsgroup data sets.

We illustrated that the Random Decision Tree [10, 9], and Parametric Bootstrap Model Averaging approaches [3] are two efficient model averaging approaches for complex model spaces since they average over only a fraction of available models and in most of our experiments used less than one third as many models as used for bagging and boosting. We explored the situations of noisy class labels in the training set, biased training sets and many class problems where both model averaging techniques outperformed a single model, bagging and boosting. We believe the demonstration of model averaging to address the presented situations is novel and is not reported in the machine learning, data mining or even the statistical literature [24].

We believe that one of the most important contributions of this paper is to show that model averaging is the preferable ensemble technique when there is model uncertainty and/or factoring model uncertainty is beneficial. Though we focused on RDT and PBMA as they are two efficient model averaging approaches other model averaging techniques such as averaging over the extended fanned set of a best tree [21] should yield similar if not better results.

References

1. Breiman, L. (1996), Bagging Predictors, Machine Learning, Vol. 24, No. 2.
2. Buntine W. (1990). *A Theory of Learning Classification Rules*, Ph.D. Thesis, UTS, Sydney.
3. Davidson I. (2004). An Ensemble Technique for Stable Learners with Performance Bounds, AAAI 2004: 330-335.
4. Dietterich, T. G. (2000). An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization. *Machine Learning*, 40 (2) 139-158.

5. Domingos, P., (1997). Why Does Bagging Work? A Bayesian Account and its Implications. KDD 1997: 155-158
6. Domingos P. (2000). Bayesian Averaging of Classifiers and the Overfitting Problem, *AAAI 2000*
7. Efron, B. (1982). The jackknife, the bootstrap, and other resampling plans. SIAM Monograph 38.
8. Fan W., Davidson I., Zadrozny B. and Yu P., (2005) An Improved Categorization of Classifier's Sensitivity on Sample Selection Bias, 5th IEEE International Conference on Data Mining, ICDM 2005.
9. Fan, W. (2004). On the optimality of probability estimation by random decision trees, AAAI 2004: 336-341.
10. Fan W., Wang H., Yu P.S, Ma S., (2003). Is random model better? On its accuracy and Efficiency. ICDM 2003: 51-58.
11. Fei Tony Liu, Kai Ming Ting, Wei Fan, (2005). Maximizing Tree Diversity by Building Complete-Random Decision Trees. PAKDD 2005: 605-610
12. Fei Tony Liu, Kai Ming Ting, Wei Fan, (2005). Maximizing Tree Diversity by Building Complete-Random Decision Trees. PAKDD 2005: 605-610
13. Frank, Eibe, *Personal Communication*, 2004.
14. Freund Y., Schapire R., (1999). A Short Introduction to Boosting. Journal of Japanese Society for Artificial Intelligence, 14(5):771-780,
15. Grunwald P.D., Langford J., (2004). Suboptimal behavior of Bayes and MDL in classification under misspecification. COLT 2004: 331-347.
16. Hoeting J., Madigan D., Raftery A., and Volinsky C., (1999). "Bayesian Model Averaging: A Tutorial", Statistical Science 14, 382-401.
17. Kohavi R., Wolpert D., (1996). Bias Plus Variance Decomposition for 0-1 Loss Functions, ICML 1996: 275-283.
18. McCallum, A. (1998). Bow: A toolkit for statistical language modeling, text retrieval, classification and clustering. CMU TR.
19. Mitchell T., (1997). *Machine Learning*, McGraw-Hill, 1997.
20. Minka, T.P., Bayesian model averaging is not model combination, MIT Media Lab note (7/6/00), available from: http://research.microsoft.com/ minka/papers/bma.html
21. Oliver, J. and Hand, D., (1995), On Pruning and Averaging Decision Trees, ICML 1995.
22. Rennie, J. *20 Newsgroups*, (2003). Technical Report, Dept C.S., MIT.
23. Special Issue on the use of MCMC in Machine Learning, (2003), Journal of Machine Learning, Volume 50.
24. Chris Volinsky, *Personal communication*, ATT Research Labs, maintainer of the BMA Web Site: http://www.research.att.com/ volinsky/bma.html
25. Webb, G. I., (1996). Further Experimental Evidence against the Utility of Occam's Razor, JAIR 4 397-417.
26. Zadrozny B., (2004). Learning and evaluating classifiers under sample selection bias, ICML 2004.