An Ensemble Technique for Stable Learners with Performance Bounds[1]

Ian Davidson

Department of Computer Science, SUNY - Albany
1400 Washington Avenue, Albany, NY, 12210, davidson@cs.albany.edu

Abstract

Ensemble techniques such as bagging and DECORATE exploit the "instability" of learners, such as decision trees, to create a diverse set of models. However their application to stable learners such as naïve Bayes, does not yield as much improvement and can sometimes degrade performance a claim we empirically verify in this paper. But stable learners are desirable for their superior performance on some problems and algorithmic simplicity. We show that many learners commonly referred to as stable have Gaussian posterior distributions. Given such a well defined posterior distribution we can use both parametric and non-parametric bootstrapping to create a process that approximates taking draws from their posterior. By summing the *joint distribution* of the instance and the class we are approximating posterior model averaging a.k.a. the optimal Bayes classifier (OBC) which is known to minimize the Bayes error. We refer to our approach as *bootstrap model averaging.* Since model averaging removes the model uncertainty it works best when there is much model uncertainty and does no harm when there is little, a claim we empirically verify. Since the Gaussian distribution is mathematically well understood we can bound the increase over the OBC error using a Chebychev bound as a function of the number of models built. We empirically illustrate our approach's usefulness and verify our bound's correctness.

1 Introduction and Motivation

Ensemble approaches are popular because they can be readily applied to many problems and decrease predictive error. Bagging (Brieman 1996), boosting (Schapire 1992), arcing (Brieman 1998) and DECORATE (Melville, Mooney, 2003) have been empirically shown to have a better generalization error than using a single model for a large variety of data sets.

Although ensemble techniques are popular, they suffer from several problems. They are best suited to unstable learners such as decision trees that readily allow the creation of a diverse set of classifiers (Melville, Mooney, 2003) and do not decrease predictive error as much for stable learners. Why an ensemble technique works is not often known or cannot be easily tested meaning when they are applicable is difficult to predict. Therefore, each ensemble technique should be tried for completeness, just in case it works. Finally, how many models to build is not known a priori leaving the number of bags, boosting rounds or data sets to generate another unknown to be investigated empirically. Therefore, though they are useful, ensemble techniques require a large investment of time to verify how many models to build and which technique to apply.

In this paper we propose an ensemble technique for stable learners we refer to as bootstrap model averaging. We show that learners typically thought of as stable have Gaussian posteriors. Given this, we can use non-parametric and parametric bootstrapping to approximate taking draws from the posterior distribution. For each draw we sum the joint probability of a class and instance given the drawn model. We show that as the number of models built approaches infinity our ensemble approach is equivalent to the optimal Bayes classifier (OBC) when using the same prior distribution over the model space. The OBC is known to work well when there is much model uncertainity and is known to minimize the Bayesian error/risk. For a given model space and prior probability distribution over the space: no classifier can perform better, hence its name (Mitchell, 1997). In this respect OBC is highly desirable but it requires integration over the model space making it extremely time consuming. Our approach offers an approximation to OBC with performance bounds dependent on the number of models built or bootstrap samples taken. This allows the quantification of the closeness of the approximation and specifying a trade-off between closeness to the OBC error and time (number of models to build).

2 Paper Outline and Standards

We begin this paper by empirically verifying the often made claim that ensemble techniques do not work as well for stable learners. Next we illustrate the two conditions for learners to have a Gaussian posterior holds for many stable learners. Given the posterior of stable learners is known to be Gaussian we show that various types of bootstrapping can create a process that in the limit is equivalent to taking draws from the posterior. Our ensemble approach is to sum the joint

[1] This paper is an extended version of a paper with the same title that appeared in the AAAI 2004 proceedings.

probability of the class and instance given the model for each model derived by applying the learner to the bootstrapped data set. We empirically illustrate that bootstrap model averaging out performs bagging, boosting and DECORATE for stable learners. Since our approach is an approximation to posterior model averaging that is known to minimize the Bayesian risk, we can bound the increase over the minimal Bayes risk as a function of the number of models built. For artificial data sets, we can measure the error increase and we verify the correctness of our bound. Finally we conclude and describe future work.

Throughout this paper we use ten standard UCI data sets of varying properties including those that contain missing values. We use the WEKA software for all experiments[2]. All results were for 100 trials with a random division of available data into training and test set.

We denote the number of models with *T* and the independent variables as *x* and the dependent variable as *y*.

3 Ensembles and Learner Stability

The purpose of this section is to illustrate empirically that popular ensemble techniques do not work *as well* for stable learners. Later, we shall show our approach does increase predictive accuracy for these very same data sets. Creating a diverse set of models appears to be an important property for successful ensemble techniques (Krough and Vedelsby, 1995). However, creating a diverse set for stable learners appears to be difficult. Techniques such as bagging are known to work best with unstable learners (compare Table 1 and Table 4) as they reduce variance. We illustrate that techniques such as DECORATE and boosting also do not work as well with stable learners (compare Table 2 with Table 5 and Table 3 with Table 6). Our presented results show the following:

1) The mean error reduction for bagging, DECORATE and boosting for J48 is respectively 2.5%, 2.6% and 3.4%.while for naïve Bayes the reduction is 0.48%, -0.23% and 0.46% respectively.

2) Bagging, DECORATE and Boosting naïve Bayes produced only four statistically significant[3] decreases in error (out of thirty experiments) but twenty four significant decreases for J48. Furthermore, Boosting can significantly decrease error.

3) The ensemble techniques (particularly DECORATE and boosting) can significantly increase predictive error for naïve Bayes.

Table 1. J48 bagging % error. Training set size: 66%.

Dataset	Single Model	T=100	Improve (Stat. Sig.)
Iris	5.7	**5.2**	0.7 (Yes)
Breast	5.5	**3.7**	1.8 (Yes)
Soybean	10.9	**8.0**	2.9 (Yes)
Crx	15.2	**13.8**	1.4 (Yes)
Adult[4]	17.0	**15.6**	1.4 (Yes)
Labor	20.9	**17.2**	3.7 (Yes)
Glass	33.3	**27.5**	5.8 (Yes)
Vote	4.5	4.1	0.4 (No)
Audio.	22.6	**18.8**	3.8 (Yes)
Auto	26.5	**20.9**	5.6 (Yes)

Table 2. J48 DECORATE % error. Training set size: 66%.

Dataset	Training Data	T=100	Improve (Stat. Sig.)
Iris	5.7	**5.1**	0.6 (Yes)
Breast	5.5	**3.9**	1.6 (Yes)
Soybean	10.9	**6.7**	4.2 (Yes)
Crx	15.2	**14.5**	0.7 (Yes)
Adult	**17.0**	17.8	-0.8 (Yes)
Labor	20.9	**11.7**	9.2 (Yes)
Glass	33.3	**27. 8**	5.5 (Yes)
Vote	**4.5**	5.3	-0.8 (Yes)
Audio.	22.6	**20.4**	2.2 (Yes)
Auto	26.5	**22.8**	3.7 (Yes)

Table 3. J48 Boosting % error. Training set size: 66%

Dataset	Training Data	T=100	Improve (Stat. Sig.)
Iris	5.7	5.8	-0.1 (No)
Breast	5.5	**3.3**	2.2 (Yes)
Soybean	10.9	**7.6**	3.3 (Yes)
Crx	15.2	**14.5**	0.7 (Yes)
Adult	17.0	17.8	-0.8 (No)
Labor	20.9	**14.1**	6.8 (Yes)
Glass	33.3	**24.5**	8.8 (Yes)
Vote	4.5	5.1	-0.6 (No)
Audio.	22.6	**16.8**	5.8 (Yes)
Auto	26.5	**18.6**	7.9 (Yes)

Table 4. Naïve Bayes bagging. Training set size: 66%

Dataset	Single Model	T=100	Improve (Stat. Sig.)
Iris	4.0	4.1	-0.1 (No)
Breast	3.7	3.7	0.0 (No)
Soybean	7.9	8.1	0.2 (No)
Crx	15.2	14.8	0.4 (No)
Adult[5]	17.0	**15.6**	1.4 (Yes)
Labor	7.8	7.6	0.2 (No)
Glass	51.3	**49.0**	2.3 (Yes)
Vote	10.1	10.2	-0.1 (No)
Audio.	31.1	31.6	-0.5 (No)
Auto	44.0	43.0	1.0 (No)

[2] WEKA's implementation of bagging sums conditional probabilities not votes, we changed this implementation to sum votes

[3] Pairwise t-Test for means at the 95% confidence level

[4] Predicting SEX field

[5] Predicting SEX field

Table 5. Naïve Bayes Decorate % error,100 trials.

Dataset	Training Data	T=100	Improve (Stat. Sig.)
Iris	**4.0**	4.9	-0.9 (Yes)
Breast	3.7	3.9	-0.2 (No)
Soybean	7.9	8.4	-0.5 (No)
Crx	15.2	14.5	-0.7 (No)
Adult	17.0	17.8	-0.8 (No)
Labor	**7.8**	9.0	-1.2 (Yes)
Glass	51.3	50.4	0.9 (No)
Vote	10.1	10.5	-0.4 (No)
Audio.	31.1	31.1	0.0 (No)
Auto	44.0	42.5	1.5 (No)

Table 6. Naïve Bayes Boosting % error 100 trials

Dataset	Training Data	T=100	Improve (Stat. Sig.)
Iris	4.0	4.4	-0.4 (No)
Breast	3.7	4.4	-0.7 (No)
Soybean	**7.9**	10.0	-2.1 (Yes)
Crx	15.2	14.5	0.7 (No)
Adult	17.0	17.8	-0.8 (No)
Labor	**7.8**	10.1	-2.3 (Yes)
Glass	51.3	50.9	0.4 (No)
Vote	10.1	**6.8**	2.3 (Yes)
Audio.	31.1	**25.7**	5.4 (Yes)
Auto	44.0	42.9	1.1 (No)

4 Stable Learners Have Gaussian Distributions

The Bayesian central limit theorem tells us precisely what properties a learner must have to produce a Gaussian posterior. They are:

1) The training data must be independently and identically distributed given the model.
2) The posterior probability density function must be twice differentiable everywhere for all of the training data.

An additional property inherently implied in the work is:

3) The learner is a deterministic function of the training data.

Property 1) means that the training set observations are modeled as being independent of each other and are drawn from the same distribution (pool of data). Property 2) may seem prohibitive but holds for many learners in the machine learning literature. Consider a learner as a function *L(.)* that places a probability distribution over the model space and returns the most probable model. The property states that if we write the density function associated with the learner, the second order derivative is defined. Most probability density functions used in learning are from the expotential family and meet the above properties. A few distributions such as the Cauchy distribution do not meet the requirements, but they appear not to be in common use.

Note, the theorem **does not state** that for models of multivariate Gaussians that the posterior will be Gaussian, it states for **any** posterior density function that is twice differentiable and models the data as IID that the posterior will be Gaussian. The smooth Gaussian distribution denotes a well defined functional relationship between the data and the model parameters. The Gaussian posterior for stable learners we shall denote as:

$$\theta \sim N(\boldsymbol{\mu}_{\mathbf{Post}} = \mathrm{E}_{\mathrm{P}(\theta|D)}(\theta), \boldsymbol{\sigma}_{\mathbf{Post}}{}^2 = \mathrm{Var}_{\mathrm{P}(\theta|D)}(\theta)) \quad (1)$$

The mean of this Gaussian is the expected model parameters over the posterior distribution (ie. $\Sigma P(\theta|D).\theta$) and the variance is calculated over model parameters multiplied by their posterior probability. As an illustrative example, consider perhaps the simplest stable learner: the majority guesser. The majority guesser predicts the most populous class in the training set and for a two class problem effectively has one parameter, ρ, the probability of class +. We encode a positive label (dependent variable) for the i^{th} instance as $y_i = 1$ and negative label as $y_i = 0$. The posterior density function over the one independent column (x) is as follows:

$$\begin{aligned} P(\rho \mid D) &\propto P(\rho)\prod_{i=1}^{n} P(x_i \mid \rho), \text{assume a uniform prior} \\ &\propto \prod_{i=1}^{n} \rho^{y_i}(1-\rho)^{(1-y_i)} \end{aligned} \quad (2)$$

The mean of the posterior is $\Sigma_i \rho_i \mathrm{P}(\rho_i|\mathrm{D}) = 0.5$ for all data sets where i indexes all possible parameter values. The posterior standard deviation for a model space of size k is measured over the observations $\{\rho_1\mathrm{P}(\rho_1|\mathrm{D}) \ldots \rho_k\mathrm{P}(\rho_k|\mathrm{D})\}$.

We now show that as expected, that naïve Bayes is stable and decision trees unstable according to our definition. Consider a two class naïve Bayes classifier with a single Boolean attribute (true=1, false=0) without loss of generality as the attributes are independent of each other in naïve Bayes. This classifier effectively builds a model for each class and we focus on the model to predict one class (+), the other class (-) model will be identical in form. The parameters for the model are $\{\mathrm{P}(+), p, q=(1-p)\}$. We use the term n_+ to indicate the number of instances with a positive label and $n_{+,T}, n_{+,F}$ the number of positive labeled instances with a TRUE and FALSE attribute respectively. Writing the posterior distribution and ignoring the constant $P(D)$ for class + yields equation (3). Taking the second order derivative with respect to p yields equation (4), a standard result for binomial distributions.

$$\begin{aligned} P(\theta_+ \mid D) &\propto P(+)\prod_{i=1}^{n_+} P(x_i \mid +) \\ &\propto P(+)\prod_{i=1}^{n_+} p^{x_i}(1-p)^{(1-x_i)} \\ &\propto \left[n_+ / n\right] p^{n_{+,x_i=T}}(1-p)^{n_{+,x_i=F}} \end{aligned} \quad (3)$$

For a uniform and hence constant prior

$$\begin{aligned} P'(\theta_+ \mid D) &\propto n_{+,x_i=T}(pe+(1-p))^{n_{+,x_i=T}-1} pe \\ P''(\theta_+ \mid D) &\propto n_{+,x_i=T}(n_{+,x_i=T}-1)(pe+(1-p))^{n_{+,x_i=T}-2} + \\ &\quad n_{+,x_i=T}(pe+(1-p))^{n_{+,x_i=T}-1} pe \end{aligned} \quad (4)$$

Most learners that involve counting (with no independent attributes having missing values) will have a posterior density function that contain a binomial distribution and hence have posteriors that are twice differentiable. Examples include k-nearest neighbor (k>2), belief networks and even association rules. These are all considered by the machine learning community to be more stable compared to learners such as decision trees. Figure 1 illustrates for the vote data set the posterior distribution for the naïve Bayes classifier. As each column is modeled independent of each other the model space is effectively sixteen real values between 0 and 1.

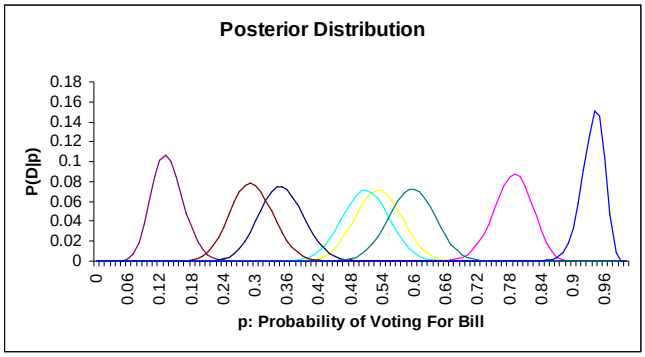


Figure 1. Posterior distribution for naïve Bayes parameters for attributes 1 – 8 only.

Now consider the posterior distribution for a decision tree learner with m attributes. If any of the attributes were continuous, the model parameters contain inequalities and hence the posterior density function cannot be differentiated. For Boolean or multistate attributes, the model consists of a disjunction of *nLeafs* conjunctions. Let the parameters for the i^{th} conjunction be $\{X_{i,1} \wedge \ldots X_{i,m}\}$ and the number of training instances following this path be n_{Path_i}. Then the posterior density function is:

$$P(\theta \mid D) \propto P(\theta) \prod_{i=1}^{nLeafs} P(Path_i)^{n_{Path_i}} \propto P(\theta) \prod_{i=1}^{nLeafs} P(X_{i,1} \wedge \ldots X_{i,m})^{n_{Path_i}} \quad (5)$$

Colloquially, the first order derivative measures the change in posterior density when the parameters are changed slightly. However, one cannot change the parameters of a conjunction slightly and the derivative of a conjunctive expression is undefined. Therefore, decision trees of either (or both) discrete or continuous attributes are not stable according to our definition.

We now discuss the optimal Bayes classifier and compare it to our ensemble approach bootstrap model averaging.

5 Optimal Bayesian Classifier and Minimal Risk

We begin this section by describing the OBC and why it is optimal. We find that deriving the conditions for optimality overcomes a common misunderstanding, namely that one should sum the joint probability of instance and class **not** the conditional probability of class given instance. The later summation inherently assumes that each instance is equally likely to be encountered which is rarely the case.

Without loss of generality consider a two class problem, y_1 and y_2, and a single test instance, x, training data D and model space Θ. For a particular instance the minimal risk is to predict the most probable class according to equation (6).

$$1 - P(y^* \mid x, \theta) \quad (6)$$

Where y^* is the most probable class

Generalizing this to minimize the risk for all instances involves an integration and factoring in the chance of encountering the instance as shown below.

$$\begin{aligned} &\int (1 - P(y^* \mid x, \theta))P(x)dx \\ &\int P(x)dx - \int P(y^* \mid x, \theta)P(x)dx \\ &1 - \int \frac{P(y^*, x, \theta)P(x)}{P(x)P(\theta)}dx \end{aligned} \quad (7)$$

as the test instance and the model are independent

$$1 - \int P(y^*, x \mid \theta)dx$$

Generalizing for all models to the remove model uncertainty involves integration and multiplying by posterior probability of the model.

$$\begin{aligned} &\int_{\theta \in \Theta} \int (1 - P(y^*, x \mid \theta))P(\theta \mid D)dx\,d\theta \\ &\int_{\theta \in \Theta} \int P(\theta \mid D) - \int_{\theta \in \Theta} \int P(y^*, x \mid \theta)P(\theta \mid D)dx\,d\theta \\ &1 - \int_{\theta \in \Theta} \int P(y^*, x \mid \theta)P(\theta \mid D)dx\,d\theta \end{aligned} \quad (8)$$

Consequently, choosing the class that minimizes the risk is equivalent to choosing the class that maximizes the summation of the joint probability of the class and test instance given the model over all models. Formally, the calculation that model averaging is performing is shown in equation (9).

$$\arg\max_i : q_i = \int_{\theta \in \Theta} P(y_i, x \mid \theta)P(\theta \mid D)\,d\theta \quad (9)$$

For a given data set, model space and prior distribution over the model space, no other approach can yield a smaller risk (Mitchell, 1997). However, OBC is a time consuming process as it involves performing an integration over the entire model space which could be high dimensional. Summing the *conditional probabilities* is equivalent to considering that each instance is equally likely.

5.1 Bootstrap Model Averaging Approximates Bayesian Model Averaging

In this section we show that creating multiple bootstrap samples ($B_1 \ldots B_T$) of the data, building a model from each ($\theta_1 \ldots \theta_T$) using a stable learner and summing the joint probability ($\Sigma_{i=1\ldots T} P(y_j, x|\theta_i)$) for a particular class estimates the degree of belief that class j is the true class for x. We need to show that the posterior distribution over models is equivalent to the distribution over the models that bootstrapping creates.

Consider the typical view of learning where a single training set of size n is available from the underlying distribution that generated the data F as shown in Figure 2.

$$D^n = \{ x_1, x_2 \ldots x_n \} \sim \mathbf{F}$$

$$\downarrow$$

$$\theta = L(D^n)$$

Figure 2. A typical view of learning

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 generated (is available). Consider T such data sets as indicated in Figure 3. If we were to build a model for each possible data set we would have a probability distribution over the model space.

$$D^n_1, D^n_2 \ldots D^n_T \sim \mathbf{F}$$

$$\downarrow$$

$$\theta_1 = L(D^n_1)$$
$$\theta_2 = L(D^n_2)$$
$$.$$
$$.$$
$$\theta_T = L(D^n_T)$$

Figure 3. A view of learning that considers training set uncertainty.

However, typically we do not have many different data sets so we can not compute the uncertainty over the model space from them. To consider the error or uncertainty for estimators/models using a single data set Efron (Efron, 1979) created the non-parametric and parametric bootstrapping approaches. In the training data set suppose that Boolean attribute i was TRUE p percent of the time and continuous attribute j was on average q. Then across the bootstrap samples the average values of attributes i and j will also be p and q respectively. Therefore, when we average over models learnt from $B_1 \ldots B_T$ we find that Average($L(B_1) \ldots L(B_T)$)$\approx L(D) \approx E_{P(\theta|D)}(\theta)$ as by definition the learner chooses the most probable model and the posterior distribution is unimodal for stable learners. Therefore $\mu_{Post} = \mu_{Boot}$ where the subscript *boot* indicates the bootstrapping distribution.

Furthermore, Variance($L(B_1) \ldots L(B_T)$) $\approx$ Variance$_{P(\theta|D)}(\theta)$. That is, $\sigma_{Post} = \sigma_{Boot}$ where as before the subscript *boot* indicates the bootstrapping distribution. An assumption of the above is that it holds in the limit as the number of samples approaches infinity, that is, $\lim_{T \to \infty}$. Later we shall derive bounds for how close this approximation is for a finite number of models. In the mean time we can visually verify that by creating just 100 samples, that the probability distribution created by bootstrapping approximates fairly well the posterior distribution as shown in Figure 4. A complete analysis of the differences is presented in the appendix.

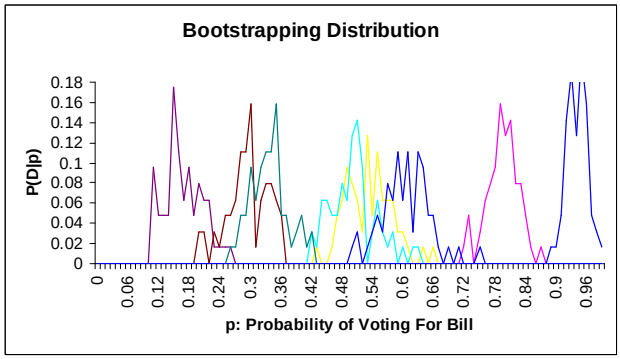


Figure 4. Distribution for naïve Bayes parameters from 100 bootstrap samples for all attributes in VOTE data set. attributes 1 – 8. Compare with Figure 1.

We can now outline our bootstrap model averaging algorithms and empirically verify their performance for improving the performance of the naïve Bayes classifier.

6 Two Algorithms for Bootstrap Model Averaging

6.1 Non-Parametric Bootstrap Model Averaging

Consider the underlying distribution, F, that created the training data D. If nothing is known of the structure of F we can empirically approximate it by sampling without replacement from D. This is the formulation of bootstrapping commonly known to the machine learning community. Our algorithm is therefore:

```
// Build the models
For i = 1 to t
   Bag_i = SampleWithReplacement(D)
   M_i = StableLearner(Bag_i)
End For
// The models M_1 … M_t  can now be treated
// as being drawn from the posterior.
// Sum the joint probability
```

$$y^* = \arg\max_i : \hat{q}_i = \sum_{i=1}^{T} P(y_i, x \mid M_i)$$

Algorithm #1): Non-parametric version of bootstrap model averaging.

This algorithm approximates the calculation in (9).

6.2 Parametric Bootstrap Model Averaging

If some knowledge of the structure of F is known then we can make use of this to perform a parametric bootstrap. Where as the non-parametric bootstrapping algorithm

previously described was independent of the underlying data set, parametric bootstrapping depends on the data set. Non-parametric bootstrapping used the actual instances in the training data to empirically approximate *F* hence each bootstrap consisted of instances that were part of the training data. With non-parametric bootstrapping a parametric distribution is fitted to the training data to approximate *F* and *virtual* instances that are not necessarily part of the training data are generated. For simplicity consider a two attribute problem one Boolean and one continuous. We can model the first attribute as a binomial distribution and the second as a Gaussian (for instance) as we have in this paper. The parametric bootstrap model averaging algorithm is as follows.

```
// Build the approximation to F
FModel = Model(D)
// Create the bags from the model
For i = 1 to t
   Bag_i = ParametricBootstrap(FModel)
   M_i = StableLearner(Bag_i)
End For
// The models M_1 … M_t  can now be treated
// as being drawn from the posterior.
// Sum the joint probability
```

$$y^* = \arg\max_i : \hat{q}_i = \sum_{i=1}^{T} P(y_i, x \mid M_i)$$

Algorithm #1): Parametric version of bootstrap model averaging

We now empirically verify the performance of these two algorithms for the stable naïve Bayes learner.

7 Bootstrap Model Averaging Naïve Bayes Classifiers

For a variety of data sets we empirically compare bootstrap model averaging against building a single model. Where as boosting and DECORATE quite often increased the predictive error we find that this does not occur for our approach. We find for six of the ten data sets that a significant decrease in error occurs. The average decrease is 1.01 which is twice as much as the previous ensemble techniques. We have included the summing of conditional probabilities to illustrate empirically that it is sub-optimal. Our results are shown in Table 7.

Table 7. Naïve Bayes bootstrap model averaging using 66% training set size. The last column indicates the results for summing conditional probabilities.

Dataset	Single Model	BMA	Improve (Stat. Sig.)	Boot Cond. Probs
Iris	4.0	4.0	0.0 (No)	4.1
Breast	3.7	3.7	0.0 (No)	3.7
Soybean	7.9	**7.5**	0.4 (Yes)	8.1
Crx	15.2	**14.3**	0.9 (Yes)	15.6
Adult[6]	17.0	**15.9**	1.1 (Yes)	17.0
Labor	7.8	**6.6**	1.2 (Yes)	7.5
Glass	51.3	51.0	0.3 (No)	52.0
Vote	10.1	10.1	0.0 (No)	10.2
Audio.	31.1	**28.0**	3.1 (Yes)	31.6
Auto	44.0	**40.7**	3.3 (Yes)	43.0

We note that all of these data sets contain missing values (with the exception of IRIS) so the naïve Bayes classifier may not be returning the most probable model, a requirement of our approach. This indicates our approach is useful even in less than ideal situations.

7.1 When Bootstrap Model Averaging Works Best

One of the benefits of our approach is that it can be explained as an approximation to posterior model averaging. It is well known that posterior model averaging removes model uncertainty (Neal 1992) (Davidson, 2000). Therefore, when there is no or little model uncertainty, such as when there is one sharply peaked posterior mode, then model averaging offers little benefit over just using the model at the posterior mode. This can be used to explain why in some data sets (Iris, Breast Cancer, Glass and Vote see Table 7) that our approach did not produce significant results beyond using a single model. For data sets which give rise to multiple explanations/models (posterior modes) or where the posterior mode is not well defined (ie. it is wide and flat) then model averaging will help.

Since model certainty increases with the data set size, we can create model uncertainty by using less of the available data. To this end we repeat our previous experiments but only using 10% of the training data. Therefore, the predictive errors of the single models decrease, and our approach significantly decreases predictive error in all but one of the data sets. Furthermore, our approach out performs other ensemble approaches. Achieving an average error decrease of 1.6% where as other ensemble approaches averaged 0.1% or less.

[6] Predicting SEX field

Table 8. Naïve Bayes classifier using a single model and a variety of ensemble approaches using **10%** training set size.

Dataset	Sing. Mod.	BMA	Decorate	Boosting	Bagging	Boot Cond.
Iris	5.8	**5.5**	6.5	6.3	5.9	7.7
Breast	4.0	**3.7**	4.0	4.7	3.9	4.4
Soy.	14.8	**13.0**	14.4	21.0	15.0	22.8
Crx	21.6	**20.7**	21.4	23.8	23.8	22.2
Adult	27.0	**25.9**	26.8	27.8	28.9	28.3
Labor	18.2	**15.6**	16.4	18.4	17.8	25.6
Glass	50.2	47.3	50.6	**46.7**	47.6	48.6
Vote	9.9	9.7	10.7	**6.1**	9.9	10.2
Audio.	54.0	**52.8**	54.7	54.8	53.4	55.0
Auto	52.8	**48.4**	53.0	49.5	50.8	54.8

We find that using our parametric bootstrap model averaging approach yields even better results, an average error improvement of 2.1% as shown in Table 9.

Table 9. Parametric bootstrap model averaging Naïve Bayes classifier using a single model and a variety of ensemble approaches using **10%** training set size.

Dataset	Single Model	BMA	Decorate	Boosting	Bagging
Iris	5.8	**5.0**	6.5	6.3	5.9
Breast	4.0	**3.3**	4.0	4.7	3.9
Soybean	14.8	**12.7**	14.4	21.0	15.0
Crx	21.6	**19.7**	21.4	23.8	23.8
Adult	27.0	**25.4**	26.8	27.8	28.9
Labor	18.2	**15.0**	16.4	18.4	17.8
Glass	50.2	**47.0**	50.6	**46.7**	47.6
Vote	9.9	9.5	10.7	**6.1**	9.9
Audio.	54.0	**52.0**	54.7	54.8	53.4
Auto	52.8	**47.8**	53.0	49.5	50.8

8 How Close is the Bootstrapping Distribution to the Posterior?

As shown in Figure 1 and Figure 4 the bootstrapping distribution only approximates the Gaussian posterior. However, since much is know of the well behaved Gaussian we can create useful bounds on how close this approximation is which we can then substitute into our risk calculations to determine the increase in risk beyond the OBC error.

8.1 Bounds for a Classifier

We now develop a bound that measures the closeness between both sets of parameters (($\boldsymbol{\mu}_{\mathbf{Post}}$, $\boldsymbol{\sigma}_{\mathbf{Post}}$) and ($\hat{\boldsymbol{\mu}}_{\mathbf{Boot}}, \hat{\boldsymbol{\sigma}}_{\mathbf{Boot}}$)) using a Chebychev inequality/bound. The Chebychev inequality (*for repeated experiments*) allows the definition of the number of samples, T, (in our case the number of bootstrap samples) required to obtain an estimate ($\hat{p}$) (calculated from those samples) that is within an error (ε, $0<\varepsilon<1$) of the true value (p). It is assumed that the samples are drawn from a distribution with mean of p and standard deviation of σ. The bound in its general form is:

$$P\left[\left|\hat{p}-p\right|>\varepsilon\right]<\frac{\sigma^2}{T(\varepsilon)^2} \qquad (10)$$

This expression can be interpreted as an upper bound on the chance that the error is larger than ε. In-turn we can upper bound the right-hand side by δ ($0<\delta<1$) which can be considered the maximum chance/risk we are willing to take that our estimate and true value differ by more than ε. We can use the posterior standard deviation estimate calculated from the bootstrap samples for σ. Then rearranging these terms to solve for the error yields:

$$\varepsilon>\sigma/\sqrt{T\delta} \qquad (11)$$
$$\left|\boldsymbol{\mu}_{\mathbf{Post}}-\hat{\boldsymbol{\mu}}_{\mathbf{Boot}}\right|>\varepsilon>\frac{\boldsymbol{\sigma}_{\mathbf{Post}}}{\sqrt{T\delta}}$$

The question of how close the means are, is now answered with respect to the chance (δ) that their difference threshold (ε) will be exceeded, for T models built. If we treat the numerator as a constant for a given problem we see that as T (number of models built) and δ (chance of failure) increases the parameter difference decreases as expected.

We now derive a bound for the standard deviation. We use a Chebychev bound but need to know the standard deviation of the posterior standard deviation. As the posterior is Gaussian the standard deviation is drawn from a chi-squared distribution, that is: $\mathbf{Stdev}(\boldsymbol{\sigma}_{\mathbf{Post}})=\sqrt{\boldsymbol{\sigma}_{\mathbf{Post}}{}^2/(2(n-1))}$. The probability the parameters differ by more than ε is then:

$$P\left[\left|\boldsymbol{\sigma}_{\mathbf{Post}}-\hat{\boldsymbol{\sigma}}_{\mathbf{Boot}}\right|>\varepsilon\right]<\boldsymbol{\sigma}_{\mathbf{Post}}{}^2/\left[2(n-1)(T\varepsilon^2)\right] \qquad (12)$$

Again we can bound the right hand side by the chance (δ) that this error will be exceeded and solve for ε. Note that these constants can differ from those in equation (11).

$$\begin{aligned}&\delta>\boldsymbol{\sigma}_{\mathbf{Post}}{}^2/\left[2(n-1)(T\varepsilon^2)\right]\\&\varepsilon^2>\boldsymbol{\sigma}_{\mathbf{Post}}{}^2/\left[2(n-1)(T\delta)\right]\\&\varepsilon>\boldsymbol{\sigma}_{\mathbf{Post}}/\sqrt{2(n-1)(T\delta)}\\&\left|\boldsymbol{\sigma}_{\mathbf{Post}}-\hat{\boldsymbol{\sigma}}_{\mathbf{Boot}}\right|>\varepsilon>\boldsymbol{\sigma}_{\mathbf{Post}}/\sqrt{2(n-1)(T\delta)}\end{aligned} \qquad (13)$$

Therefore, if we use T samples with our ensemble approach, then we know the difference in the calculated mean will be no more (with chance no more than δ)

than $\frac{\boldsymbol{\sigma}_{\mathbf{Post}}}{\sqrt{T\delta}}$ and the error in the standard deviation no more than $\boldsymbol{\sigma}_{\mathbf{Post}} / \sqrt{2(n-1)(T\delta)}$.

We can now rewrite the distribution obtained via bootstrap model averaging from *T* samples with respect to the posterior distribution in the worst case to be: $\mathbf{N}\left[\boldsymbol{\mu}_{\mathbf{Post}} \pm \frac{\boldsymbol{\sigma}_{\mathbf{Post}}}{\sqrt{T\delta}}, \boldsymbol{\sigma}_{\mathbf{Post}} \pm \boldsymbol{\sigma}_{\mathbf{Post}} / \sqrt{2(n-1)(T\delta)}\right]$. This is so as our just presented error calculations are only for differences and we do not know if the approach will exceed or be less than the true value. We can now substitute these errors into our risk calculations (equation (9)) to determine the approximation to the risk which we denote with the estimation symbol (hat).

$$\hat{R}_{i,\Theta,D} = 1 - \int_{\theta} P(y_i, x \mid \theta) \left[\begin{array}{l} P(\theta \mid \mu = \boldsymbol{\mu}_{\mathbf{Post}} \pm \frac{\boldsymbol{\sigma}_{\mathbf{Post}}}{\sqrt{T\delta}}, \\ \sigma = \boldsymbol{\sigma}_{\mathbf{Post}} \pm \boldsymbol{\sigma}_{\mathbf{Post}} / \sqrt{2(n-1)(T\delta)} \end{array} \right] \quad (14)$$

Performing this integration is equivalent to drawing an infinite number of models according to the bootstrap model averaging distribution over the model space. Note that $\hat{\theta}_i$ is the i^{th} model drawn from the bootstrapping distribution. Formally:

$$\hat{R}_{i,\Theta,D} = 1 - \sum_{i=1}^{\infty} P(y_i, x \mid \hat{\theta}_i)$$

where $\hat{\theta}_i \sim N(\boldsymbol{\mu}_{\mathbf{Post}} \pm \frac{\boldsymbol{\sigma}_{\mathbf{Post}}}{\sqrt{T\delta}}, \boldsymbol{\sigma}_{\mathbf{Post}} \pm \boldsymbol{\sigma}_{\mathbf{Post}} / \sqrt{2(n-1)(T\delta)})$

$\hat{\theta}_i = \{\theta_i \pm \beta_i\}$,

$\theta_i \sim N(\boldsymbol{\mu}_{\mathbf{Post}}, \boldsymbol{\sigma}_{\mathbf{Post}})$,

$\beta_i \sim N(\pm \frac{\boldsymbol{\sigma}_{\mathbf{Post}}}{\sqrt{T\delta}}, \pm \boldsymbol{\sigma}_{\mathbf{Post}} / \sqrt{2(n-1)(T\delta)})$

due to the additive nature of the Normal distribution.

$$\hat{R}_{i,\Theta,D} = 1 - \sum_{i=1}^{\infty} [P(y_i, x \mid \theta_i) \pm P(y_i, x \mid \beta_i)]$$

$$= 1 - \int_{\theta} [P(y_i, x \mid \theta) P(\theta \mid \mu = \boldsymbol{\mu}_{\mathbf{Post}}, \sigma = \boldsymbol{\sigma}_{\mathbf{Post}}) \pm$$

$$P(y_i, x \mid \theta) P(\theta \mid \mu = \frac{\boldsymbol{\sigma}_{\mathbf{Post}}}{\sqrt{T\delta}}, \sigma = \boldsymbol{\sigma}_{\mathbf{Post}} / \sqrt{2(n-1)(T\delta)})] d\theta$$

$$= 1 - \int_{\theta} P(y_i, x \mid \theta) P(\theta \mid \mu = \boldsymbol{\mu}_{\mathbf{Post}}, \sigma = \boldsymbol{\sigma}_{\mathbf{Post}}) \pm \left[\begin{array}{l} P(y_i, x \mid \theta) \times \\ P(\theta \mid \mu = \frac{\boldsymbol{\sigma}_{\mathbf{Post}}}{\sqrt{T\delta}}, \sigma = \frac{\boldsymbol{\sigma}_{\mathbf{Post}}}{\sqrt{2(n-1)(T\delta)}} \end{array} \right] d\theta$$

$$= R_{i,\Theta,D} \pm \left[\begin{array}{l} P(y_i, x \mid \theta) \times \\ \int_{\theta} P(\theta \mid \mu = \frac{\boldsymbol{\sigma}_{\mathbf{Post}}}{\sqrt{T\delta}}, \sigma = \boldsymbol{\sigma}_{\mathbf{Post}} / \sqrt{2(n-1)(T\delta)}) d\theta \end{array} \right]$$

therefore, $\left| R_{i,\Theta,D} - \hat{R}_{i,\Theta,D} \right| <$

$$\int_{\theta} P(y_i, x \mid \theta) \times P(\theta \mid \mu = \frac{\boldsymbol{\sigma}_{\mathbf{Post}}}{\sqrt{T\delta}}, \sigma = \boldsymbol{\sigma}_{\mathbf{Post}} / \sqrt{2(n-1)(T\delta)}) d\theta \quad (15)$$

As the number of models in the ensemble increases, the distribution in equation (15) becomes more peaked and its contribution is reduced as expected. This equation is an inequality as we have used *upper bounds* to perform a worse case analysis.

8.2 Experiments with a Majority Guesser with Uniform Prior

We provide this illustrate example to indicate how our bounds are used. Again, consider a majority guesser for a two-class problem, where each class is equiprobable in the training set of size 50 (ie. ρ=0.5). Then μ_{Post}=0.5, σ_{Post}=.0.057. For T=250 and δ=0.05 we find from equations (11) that the differences in means should be no more than 0.057/sqrt(250x0.05) = 0.00456 five percent of the time. Empirically we find that repeating the bootstrap model averaging approach 1000 times yields 33 (3.3%) occurrences where the means differ by more than the calculated error (0.00456). This is to be expected, as the equations provide an **upper bound** on the chance of failure. Equation (13) specifies that the variances should differ by no more than $0.057 / \sqrt{(98)12.5} \approx 0.0016$, no more than 5% of the time. Over 1000 experiments, we find that 38 times (3.8%) this error was exceeded.

Given these bounds hold, how can we use them in practice? We know the OBC gives us the optimal results and we have an approximation to this approach which we know the error of. This allows us to quantify the difference between q_i and its estimate using equation (15). We can then produce an error range for q_i to produce a joint probability *region* for each class. So long as these do not overlap, then our results will be the same as for OBC with a chance of failure no greater than δ. We can use as many bootstrap samples as required to prevent the regions from overlapping. In this way we are only drawing as many models as required given our tolerance to risk.

8.3 Experiments with Belief Networks

We now focus on differences in predictive error for the standard Boolean belief network shown in Figure 5.

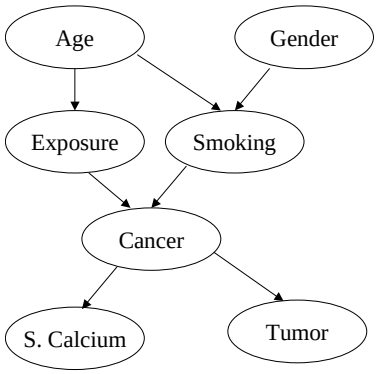


Figure 5. The Cancer Belief Network

As we know the parameters of the generating mechanism we can determine the difference between the true predictive distributions (equation (9) and our approaches estimate of

it. We used our ensemble approach to build a collection of models (T=500) so that the decision regions do not overlap using the chance of failure 5% (δ=0.05). We found that for 1000 random test queries (ie. Of the form P(Tumor = ? | Gender=M, Smoking=T)) that OBC and our approach made differing predictions 2% of the time over one hundred experiment repetitions.

9 Related Work

Previous work has explored the idea of aggregating probabilities rather than votes when creating models via bootstrapping. The work by Bauer and Kohavi (Bauer and Kohavi, 1998) mentions aggregating conditional probabilities (not joint probabilities) and find that this does not significantly improve the classification error for the naïve Bayes learner but does provide improvement for unstable learners for just T=15.

The work of Domingos (Domingos 2000) argues that bagging (bootstrapping with uniform votes) is an approximation to Bayesian model averaging by importance sampling. He then provides extensive empirical evidence showing that attempting to improve importance sampling does not yield better results than bagging. We note that the learners used in that work are not stable according to our definition and our work makes no claims on its applicability to unstable learners.

10 Conclusion and Future Work

Ensemble approaches are popular but typically require creating a diverse set of models. This is difficult for stable learners. In addition ensemble approaches are not always underpinned by a rigorous theory and determining the number of models is difficult. We showed that typical stable learners have Gaussian posteriors. The major requirement for a learner to have a Gaussian posterior is that the second order derivative for the posterior density function is defined. Learners with this property that model the data as IID have a Gaussian posterior according to the Bayesian central limit theorem. We created an ensemble technique for stable learners known as bootstrap model averaging that creates bootstrap samples of the data and builds a model from each. The models created from this process approximate being drawn from the posterior distribution. Rather than aggregating votes amongst these models (like bagging), the joint probability of the instance and class are summed and the most probable class is predicted. This is equivalent to OBC as the number of models reaches infinity and is much faster than MCMC techniques at performing posterior model averaging (Neal 1992, Davidson 2000). In the finite situation we developed bounds to determine how far above the OBC error our approach was.

We empirically demonstrate our approach for belief networks with artificial data sets. For the naïve Bayes classifier we were able reliably obtain statistically significant improvements where bagging, boosting and DECORATE could not. Importantly, our approach did not resulted in an increase in predictive error while all other ensemble techniques for at least one data set did. We illustrated that our approach works best when there is model uncertainty. We empirically simulated this by using fewer data points in the training data and found our approach provided a statistically significant benefit in all but one data set.

Our future work will involve generalization to latent variable models and creating a more complete list of stable and unstable learners. We are particularly interested in developing a more constructive form of the bounds described in this paper. For example the PAC learning literature shows that for a PAC learning drawing a single model from the posterior will yield a predictive error no worse than twice the Bayes error (Mitchell 1997).

Appendix: Posterior Distribution and Approximation to Posterior Using Bootstrapping

In this section we provide a real world example showing the approximation that bootstrapping provides for the vote data set using a naïve Bayes classifier. The vote data set consists of 16 Boolean attributes with a Boolean dependent variable, we shall focus on the model for Democrats. The parameters of the model to predict "democrat" are then 16 continuous values that signify the probability that a democrat will vote YES/SUCCESS on a particular bill. Figure 6 shows the Gaussian posterior distribution as expected.

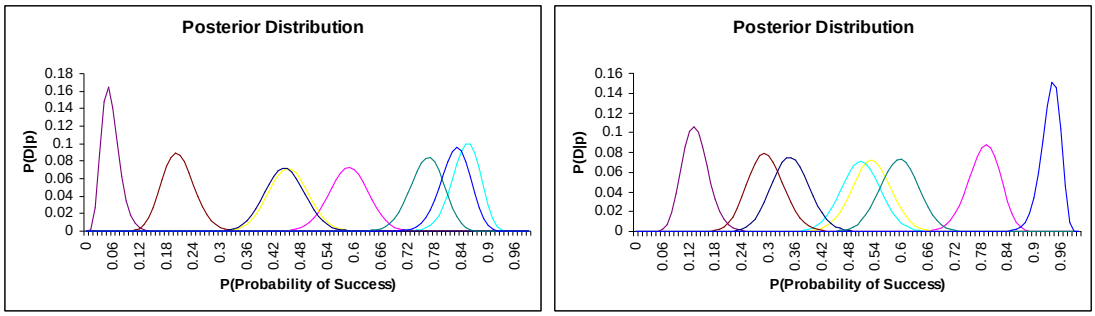
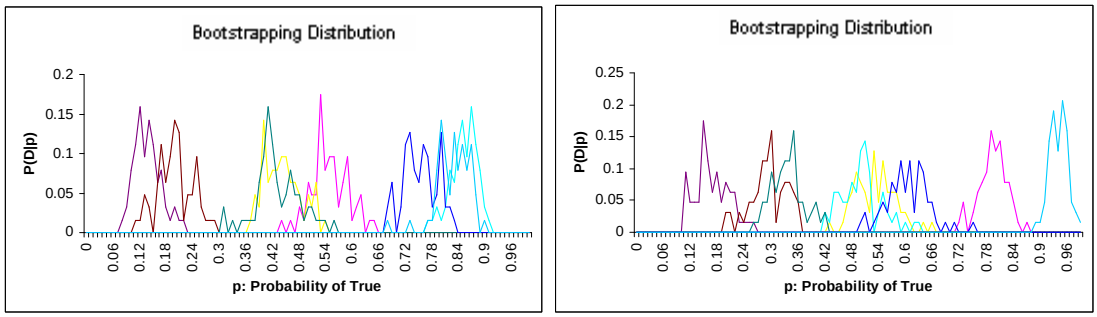


Figure 6. Posterior distribution for naïve Bayes parameters for all attributes in VOTE data set. Left figure (attributes 1 – 8), right figure attributes (9-16).

We created 100 bootstrap samples and built a naïve Bayes model from each. We then create a relative frequency distribution table for each of the sixteen parameter values over their possible values. We see that the distribution created by bootstrapping is approximately Gaussian.



KLDistance

Figure 7. Distribution for naïve Bayes parameters from 100 bootstrap samples for all attributes in VOTE data set. Left figure (attributes 1 – 8), right figure attributes (9-16).

We can better quantify the difference between the two distributions by considering the KL distance and difference between their means and standard deviations as shown in. Note these KL distances are over the actual probability distribution of the models built from the bootstrap samples not after fitting a Gaussian distribution to the data.

Table 10. The KL distance and difference between means and standard deviations of the posterior distribution and distribution obtained via bootstrapping.

Attribute	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**
Mean Diff	0.04	0.01	0	0.08	0.01	0.01	0.01	0	0.01	0	0.01	0.03	0	0.01	0.01	0
Stdev Diff	0.001	0.001	0.002	0.002	0	0	0.002	0.002	0.003	0.001	0.001	0.001	0.001	0.001	0.001	0.002
KL Dist	0.24	0.08	0.03	2.05	0.1	0.12	0.1	0.1	0.06	0.07	0.1	0.3	0.1	0.07	0.1	0.05

.

References

Bauer and Kohavi, An Empirical Evaluation of Ensemble Techniques, Journal of Machine Learning, 1998.

Brieman, L., Bagging Predictors, *Machine Learning 1996.*

Breiman, L., Arcing classifiers. Ann. Statistics,26(3), 1998.

Clarke B. and Barron A. Entropy, risk and the Bayesian central limit theorem. manuscript.

Davidson, I., Gibbs Sampling and the MML Principle, Uncertainty in A.I. 2000.

Domingos, P., Bayesian Averaging of Classifiers and the Overfitting Problem. ICML, 2000.

Duda R., Hart P., & Stork D., Pattern Classification. Wiley, 2001.

Efron, B. 1979. Bootstrap methods. Annals Statistics 7:1-26.

Krogh A. and Vedelsby J., Neural network ensembles, cross validation and active learning. NIPS 1995.

Melville, P. and Mooney, R. Constructing Diverse Classifier Ensembles Using Artificial Training Examples, IJCAI 2003.

Mitchell, T. (1997), Machine Learning, McGraw Hill.

Neal, R., Markov Chain Monte Carlo Sampling, Technical Report, Dept of Computer Science, University of Tornoto, 1992.

Schapire R.E.. The strength of weak learnability. Machine Learning, 5(2):197-227, 1990.