An Ensemble Technique for Stable Learners with Performance Bounds

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, creating a diverse set of models for stable learners such as naïve Bayes is difficult as they are relatively insensitive to training data changes. Furthermore, many popular ensemble techniques do not have a rigorous underlying theory and often provide no insight into how many models to build. We formally define stable learner as having a second order derivative of the posterior density function and propose an ensemble technique specifically for stable learners. Our ensemble technique, *bootstrap model averaging,* creates a number of bootstrap samples from the training data, builds a model from each and then sums the joint instance and class probability over all models built. We show that for stable learners our ensemble technique for infinite bootstrap samples approximates posterior model averaging (aka the optimal Bayes classifier (OBC)). For finite bootstrap samples we estimate the increase over the OBC error using Chebychev bounds. We empirically illustrate our approach's usefulness for several stable learners and verify our bound's correctness.

1 Introduction and Motivation

Ensemble approaches are popular because they 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.

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, a claim we empirically verify in this paper. Why an ensemble technique works is not often known 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 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. For now, we define only the behavior of a stable learner as building similar models from slight variations of a data set, precise properties we leave until later sections. Examples of stable learners include naïve Bayes classifiers and belief networks when applied to data sets with no missing values. 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 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. 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

We begin this paper by algorithmically outlining our ensemble approach. Some ensembles approaches such as bagging are known to not perform well for stable learners as it is difficult to create a diverse set of models, we verify this and show similar results for DECORATE. We then formally define stable learners and describe their properties. The subsequent section shows our approach approximates the OBC in the limit as the number of models built approaches infinity. For the non-ideal case, we bound the increase over the OBC error 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. We next empirically demonstrate our approach on real world data sets for naïve Bayes and belief networks showing that the results are better than a single model and bagging. Finally we conclude and describe future work. Throughout this paper we use five standard UCI data sets, similar results were obtained for other data sets but are not presented due to space limitations.

3 Bootstrap Model Averaging

From the n instance training data set, D, a number of (T) bootstrap samples (sampling with replacement) of size n are

drawn and a model built for each. Let θ_t represent the model learnt from bootstrap sample (X_t). Each model votes for every class according to the joint probability of the class and the test set instance given the model. Let the sum of the joint probability for class i be r_i. Formally bootstrap model averaging chooses the highest probability class:

$$\arg\max_i : r_i = \sum_t^T P(y_i, x \mid \theta_t) \quad (1)$$

Generative learners such as naïve Bayes classifiers and mixture models estimate the joint probability.

4 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 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 (see Table 1 and Table 3) as they reduce variance. We illustrate that techniques such as DECORATE also does not work as well with stable learners (see Table 2 and Table 4). Our presented results show the following:

1) The mean error reduction for bagging (DECORATE) C4.5 is 1.5% (1.3%), for naïve Bayes 0.13% (0.5%).

2) Bagging and DECORATE naïve Bayes produced only one statistically significant[1] decrease in error but nine significant decreases for C4.5.

3) Bagging and DECORATE generalization errors are non-monotonic functions of the number of bags/rounds.

Table 1. C4.5 bagging 10 X-fold % error, 100 trials.

Dataset	Training Data	T=10	T=50	T=100	Improve (Stat. Sig.)
Iris	5.4	4.7	5.4	4.7	0.7 (Yes)
Breast	5.2	4.2	3.4	3.2	2.0 (Yes)
Soybean	7.6	6.6	6.8	6.1	1.5 (Yes)
Crx	15.2	13.5	14.2	13.8	1.8 (Yes)
Adult[2]	17.0	16.4	16.2	15.6	1.4 (Yes)

Table 2. Naïve Bayes bagging 10 X-fold % error,100 trials.

Dataset	Training Data	T=10	T=50	T=100	Improve (Stat. Sig.)
Iris	5.2	5.3	5.3	5.3	0 (No)
Breast	2.7	2.7	2.8	2.8	0 (No)
Soybean	9.25	9.35	9.2	9.3	0 (No)
Crx	22.5	22.1	22.4	22.0	0.48 (No)
Adult	19.3	19.3	19.2	19.2	0.16 (No)

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

[2] Predicting SEX field

Table 3. C4.5 DECORATE 10 X-fold % error, 100 trials.

Dataset	Training Data	T=10	T=50	T=100	Improve (Stat. Sig.)
Iris	5.4	5.4	6.0	5.4	0.0 (No)
Breast	5.2	3.6	3.6	3.4	1.8 (Yes)
Soybean	7.6	5.8	6.1	6.4	1.5 (Yes)
Crx	15.2	14.9	14.5	14.5	0.7 (Yes)
Adult	17.0	18.4	14.5	17.8	2.5 (Yes)

Table 4. Naïve Bayes DECORATE 10 X-fold error, 100 trials

Dataset	Training Data	T=10	T=50	T=100	Improve (Stat. Sig.)
Iris	5.2	5.2	5.2	5.1	0.1 (No)
Breast	2.7	3.0	3.0	3.0	-0.3 (No)
Soybean	9.25	9.3	9.3	9.3	-0.0 (No)
Crx	22.5	20.2	20.2	20.2	2.3 (Yes)
Adult	19.3	19.4	19.4	19.4	0.16 (No)

5 Stable Learners

So far we have only colloquially mentioned that a stable learner converges to a set of similar models from slight variations of the training data set, we now formalize the definition. A stable learner has a posterior density function whose second order derivative with respect to the model parameters **is defined**. This second order derivative is the rate of change, of the rate of change of the posterior probability to changes in the model parameters for a given data set. A learner/estimator that has the above mentioned property and one other is all that is required for the Bayesian central limit theorem (Clarke and Barron) to hold.

The Bayesian central limit theorem tells us precisely what properties a learner must have to produce a Gaussian posterior. Those 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 not required for the theorem to hold but we add into our definition of stable learners is:

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

Property 1) means that the training set observations can be considered independent of each other and are drawn from the same distribution (pool of data). Property 2) we have discussed earlier in this section may seem prohibitive but holds for many learners in the machine learning literature. 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. Conversely small changes to the data will result in only small changes to the posterior probabilities. We could have defined the second order derivative with respect to changes in the data, but the number of parameters in a model is far less than the number of data points and hence easier to calculate. The Gaussian posterior for stable learners is:

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

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 simply 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 for the i^{th} instance as $y_i = 1$ and negative label as $y_i = 0$. The posterior density function over the one parameter is:

$$\begin{aligned} P(\rho \mid D) &\propto P(\rho)\prod_{i=1}^{n} P(x_i \mid \rho), \text{assume a uniform prior} \\ &\propto \prod_{i=1}^{n} \rho^{y_i}(1-\rho)^{(1-y_i)} \end{aligned} \qquad (3)$$

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 over $\{\rho_1\mathrm{P}(\rho_1|\mathrm{D}) \ \ldots \ \rho_k\mathrm{P}(\rho_k|\mathrm{D})\}$. 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. 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 for class + yields equation (4). Taking the second order derivative with respect to p yields equation (5), a standard result for binomial distributions.

$$\begin{aligned} P(\theta_+ \mid D) &\propto P(+)\prod_{i=1}^{n_+} P(x_i \mid +) \\ &\propto P(+)\prod_{i=1}^{n_+} p^{x_i}(1-p)^{(1-x_i)} \\ &\propto [n_+/n] p^{n_{+,x_i=T}}(1-p)^{n_{+,x_i=F}} \end{aligned} \qquad (4)$$

For a uniform and hence constant prior

$$\begin{aligned} P'(\theta_+ \mid D) &\propto n_{+,x_i=T}(pe+(1-p))^{n_{+,x_i=T}-1} pe \\ P''(\theta_+ \mid D) &\propto n_{+,x_i=T}(n_{+,x_i=T}-1)(pe+(1-p))^{n_{+,x_i=T}-2} + \\ &\quad n_{+,x_i=T}(pe+(1-p))^{n_{+,x_i=T}-1} pe \end{aligned} \qquad (5)$$

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 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.

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:

$$\begin{aligned} P(\theta \mid D) &\propto P(\theta)\prod_{i=1}^{nLeafs} P(Path_i)^{n_{Path_i}} \\ &\propto P(\theta)\prod_{i=1}^{nLeafs} P(X_{i,1} \wedge \ldots X_{i,m})^{n_{Path_i}} \end{aligned} \qquad (6)$$

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.

6 Optimal Bayesian Classifier and Bootstrap Model Averaging

We begin this section by describing the OBC and why it is optimal. Then we describe our approach and show it is equivalent to OBC for stable learners in the limit as the number of models approaches infinity. For the finite case we then derive bounds that measure the difference between the two ensemble approaches.

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. Generalizing our results for the entire instance space involves just adding another integral over the instance space. The OBC sums the belief that the instance x is in class y_i for each model weighted by the posterior probability of the model. The approach chooses the class that maximizes equation (7).

$$\arg\max_i : q_i = \int_{\theta\in\Theta} P(y_i, x \mid \theta) P(\theta \mid D)\, d\theta \qquad (7)$$

To minimize a zero-one loss function it is well known that the optimal decision rule is to select the class with the highest probability (Duda, Hart, Stork, 2001). The risk associated with selecting class i is then 1-P($Class_i$) . If this

approach is applied for all models, the overall risk or expected loss (Duda, Hart, Stork, 2001) associated with choosing class/action i using a model space Θ and training data D is:

$$R_{i,\Theta,D} = \int_\theta \left(1 - P(y_i, x \mid \theta)\right) p(\theta \mid D)\, d\theta \quad (8)$$
$$= \int_\theta p(\theta \mid D)\, d\theta - \int_\theta P(y_i, x \mid \theta)\, p(\theta \mid D)\, d\theta$$
$$= 1 - \int_\theta P(y_i, x \mid \theta)\, p(\theta \mid D)\, d\theta$$

Consequently, choosing the class that maximizes (7) minimizes the risk. 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.

6.1 Bootstrap Model Averaging Approximates Bayesian Model Averaging

In this section we show that bootstrap model averaging approximates the OBC for stable learners. The posterior probability distribution for a stable learner according to the Bayesian central limit theorem is a Gaussian as shown in (2). We could obtain the exact values of these parameters of this distribution, but this would effectively involve the computation to perform the integration in equation (7). However, for a Gaussian the sample mean and sample standard deviation are *sufficient statistics* for the population mean and standard deviation and we can use them to perform our inference on the test set instance. Generally, that is $P(\boldsymbol{\mu_\theta}, \boldsymbol{\sigma_\theta} \mid D) = P(\boldsymbol{\mu_\theta}, \boldsymbol{\sigma_\theta} \mid \hat{\boldsymbol{\mu}}_\theta, \hat{\boldsymbol{\sigma}}_\theta)$. We note that the population parameters in our situation are the posterior mean and variance calculated from *all* of the models in the model space (equation (2)) and sample parameters are calculated from a small subset of models sampled. For stable learners this relationship is shown in equation (9).

$$P(\boldsymbol{\mu_\theta} = \mathrm{E}_{P(\theta \mid D)}(\theta), \boldsymbol{\sigma_\theta}^2 = \mathrm{Var}_{P(\theta \mid D)}(\theta) \mid D) \quad (9)$$
$$= P(\boldsymbol{\mu_\theta} = \mathrm{E}_{P(\theta \mid D)}(\theta), \boldsymbol{\sigma_\theta}^2 = \mathrm{Var}_{P(\theta \mid D)}(\theta) \mid \hat{\boldsymbol{\mu}}_\theta, \hat{\boldsymbol{\sigma}}_\theta^2)$$

If the bootstrap model averaging approach yields a *sample* (posterior) mean and standard deviation then it produces sufficient statistics for the posterior distribution. Stable learners are by definition (property 3) deterministic functions written as L(.). The learner when applied to the training data places a fixed probability distribution over the model space with a prescribed mean and variance (equation (2)) and returns the most probable model. When the learner is applied within our ensemble approach of bootstrap model averaging it is effectively a deterministic function of the bootstrap samples. The variability over the models within our approach is due to variability across the bootstrap samples.

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 $X_1 \ldots X_T$ we find that Average($L(X_1) \ldots L(X_T)$)$\approx L(D) \approx \mathrm{E}_{P(\theta \mid D)}(\theta)$ by definition as the learner chooses the most probable model and the posterior distribution is unimodal. Furthermore, Efron (Efron, 1979) created the bootstrapping approach to estimate the population variance. The variance of the attributes across the bootstrap samples approximately equals the variance across randomly drawn samples which is a function of the training data variance. Therefore, $\mathrm{Variance}(L(X_1) \ldots L(X_T)) \approx \mathrm{Variance}_{P(\theta \mid D)}(\theta)$. An assumption of the above is that it holds in the limit as the number of samples approaches infinity, that is, $\lim_{T \to \infty}$. We now develop a bound on the closeness of the approximation of bootstrap model averaging to the OBC in the non-ideal case when $T << \infty$.

In earlier work 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.

6.2 Bounds for a Classifier

We now develop a bound that measures the closeness between both sets of parameters (($\boldsymbol{\mu_\theta}, \boldsymbol{\sigma_\theta}$) and ($\hat{\boldsymbol{\mu}}_\theta, \hat{\boldsymbol{\sigma}}_\theta$)) 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:

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

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:

$$\varepsilon > \sigma \ / \sqrt{T\delta} \quad (11)$$
$$\left|\boldsymbol{\mu_\theta} - \hat{\boldsymbol{\mu}}_\theta\right| > \varepsilon > \frac{\boldsymbol{\sigma_\theta}}{\sqrt{T\delta}}$$

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 error 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: $\mathbf{Stdev}(\hat{\boldsymbol{\sigma}}_{\boldsymbol{\theta}}) = \sqrt{\boldsymbol{\sigma}_{\boldsymbol{\theta}}^{\,2}/(2(n-1))}$ .The probability the parameters differ by more than ε is then:

$$P\left[\left|\boldsymbol{\sigma}_{\boldsymbol{\theta}} - \hat{\boldsymbol{\sigma}}_{\boldsymbol{\theta}}\right| > \varepsilon\right] < \boldsymbol{\sigma}_{\boldsymbol{\theta}}^{\,2} / \left[2(n-1)(T\varepsilon^2)\right] \quad (12)$$

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).

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

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

We can see that the posterior distribution $\mathbf{N}(\boldsymbol{\mu}_{\boldsymbol{\theta}}, \boldsymbol{\sigma}_{\boldsymbol{\theta}})$ and the distribution obtained via bootstrap model averaging from T samples is in the worst case: $\mathbf{N}\left[\boldsymbol{\mu}_{\boldsymbol{\theta}} \pm \frac{\boldsymbol{\sigma}_{\boldsymbol{\theta}}}{\sqrt{T\delta}}, \boldsymbol{\sigma}_{\boldsymbol{\theta}} \pm \boldsymbol{\sigma}_{\boldsymbol{\theta}} / \sqrt{2(n-1)(T\delta)}\right]$. This is so as our 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 to determine the approximation to the risk which we denote with the estimation symbol (hat).

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

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 θ_i is the i^{th} model drawn from a distribution. Formally:

$$\begin{aligned} &\hat{R}_{i,\Theta,D} = 1 - \sum_{i=1}^{\infty} P(y_i, x \mid \hat{\theta}_i) \\ &\text{where } \hat{\theta}_i \sim N(\boldsymbol{\mu}_{\boldsymbol{\theta}} \pm \frac{\boldsymbol{\sigma}_{\boldsymbol{\theta}}}{\sqrt{T\delta}},\ \boldsymbol{\sigma}_{\boldsymbol{\theta}} \pm \boldsymbol{\sigma}_{\boldsymbol{\theta}} / \sqrt{2(n-1)(T\delta)}) \\ &\quad \hat{\theta}_i = \{\theta_i \pm \beta_i\}, \\ &\quad \theta_i \sim N(\boldsymbol{\mu}_{\boldsymbol{\theta}}, \boldsymbol{\sigma}_{\boldsymbol{\theta}}), \beta_i \sim N(\pm \frac{\boldsymbol{\sigma}_{\boldsymbol{\theta}}}{\sqrt{T\delta}}, \pm \boldsymbol{\sigma}_{\boldsymbol{\theta}} / \sqrt{2(n-1)(T\delta)}) \\ &\text{due to the additive nature of the Normal distribution.} \\ &\hat{R}_{i,\Theta,D} = 1 - \sum_{i=1}^{\infty} \left[P(y_i, x \mid \theta_i) \pm P(y_i, x \mid \beta_i)\right] \\ &\qquad = 1 - \left[\begin{array}{l} \int_{\theta} [P(y_i, x \mid \theta) P(\theta \mid \boldsymbol{\mu}_{\boldsymbol{\theta}}, \boldsymbol{\sigma}_{\boldsymbol{\theta}}) \pm \\ \quad P(y_i, x \mid \theta) P(\theta \mid \frac{\boldsymbol{\sigma}_{\boldsymbol{\theta}}}{\sqrt{T\delta}}, \boldsymbol{\sigma}_{\boldsymbol{\theta}} / \sqrt{2(n-1)(T\delta)})] \end{array} \right] \\ &\text{therefore, } \left|R_{i,\Theta,D} - \hat{R}_{i,\Theta,D}\right| \\ &\qquad < \int_{\theta} P(y_i, x \mid \theta) P(\theta \mid \frac{\boldsymbol{\sigma}_{\boldsymbol{\theta}}}{\sqrt{T\delta}}, \boldsymbol{\sigma}_{\boldsymbol{\theta}} / \sqrt{2(n-1)(T\delta)}) \end{aligned} \quad (15)$$

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.

7 Experimental Results

7.1 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_θ=0.5, σ_θ=.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 $0.057 / \sqrt{(98)5} \approx 0.0025$, 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 r_i and q_i 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.

7.2 Belief Networks

We now focus on differences in predictive error for the standard Boolean belief network shown in Figure 1.

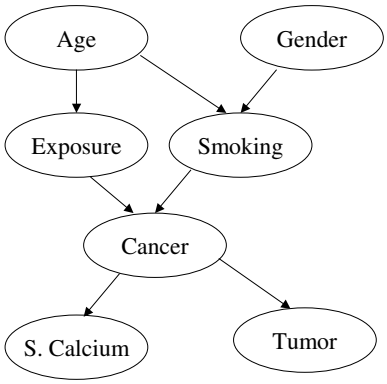

Figure 1. The Cancer Belief Network

As we know the parameters of the generating mechanism we can determine the difference between the true predictive distributions: r_i (equation (1)) and our approaches estimate of it: q_i (equation (7)). We used our ensemble approach to build a collection of enough models (n=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.

7.3 Naïve Bayes for Real World Datasets

For a variety of data sets we empirically compare bootstrap model averaging against building a single model. As we do not know the true population parameters, we can not test the accuracy of our bounds as for artificial data sets. Therefore, in this section we test our approaches' usefulness using empirical experiments. Where as bagging and DECORATE did not increase the predictive accuracy we find that this is not the case for our approach. Furthermore, we find that the improvement increases as a function of the number of bootstrap samples. We find for all but one data set that the decrease in error is significant and that the error decreases as a function of the number of models.

Table 5. Naïve Bayes bootstrap model averaging 10 X-fold error (%) over 100 trials.

Dataset	Training Data	T= 10	T= 50	T= 100	Improvement (Stat. Sig.)[3]
Iris	5.2	4.9	4.6	4.5	0.7 (Yes)
Breast	2.7	2.6	2.6	2.5	0.2 (No)
Soybean	9.25	8.5	7.7	7.5	1.75 (Yes)
Crx	22.5	22.3	22.1	22.0	0.5 (Yes)
Adult	19.3	18.9	18.7	18.2	0.9 (Yes)

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.

8 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 increasing the number of models to build does not always decrease predictive error. We formally specified the notion of stable learners as having a defined second order derivative for the posterior density function. 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. 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. 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 to obtain statistically significant improvements where bagging and DECORATE could not. Our future work will involve generalization to latent variable models and creating a list of stable and unstable learners.

References

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.

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.

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

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