Bayesian Averaging of Classifiers and the Overfitting Problem

Pedro Domingos PEDROD@CS.WASHINGTON.EDU
Department of Computer Science and Engineering, University of Washington, Seattle, WA 98195, U.S.A.

Abstract

Although Bayesian model averaging is theoretically the optimal method for combining learned models, it has seen very little use in machine learning. In this paper we study its application to combining rule sets, and compare it with bagging and partitioning, two popular but more *ad hoc* alternatives. Our experiments show that, surprisingly, Bayesian model averaging's error rates are consistently higher than the other methods'. Further investigation shows this to be due to a marked tendency to overfit on the part of Bayesian model averaging, contradicting previous beliefs that it solves (or avoids) the overfitting problem.

1. Introduction

A learner's error rate can often be much reduced by learning several models instead of one, and then combining them in some way to make predictions (Drucker et al., 1994; Freund & Schapire, 1996; Quinlan, 1996; Maclin & Opitz, 1997; Bauer & Kohavi, 1999). In recent years a large number of more or less *ad hoc* methods for this purpose have been proposed and successfully applied, including bagging (Breiman, 1996a), boosting (Freund & Schapire, 1996), stacking (Wolpert, 1992), error-correcting output codes (Kong & Dietterich, 1995), and others. Bayesian learning theory (Bernardo & Smith, 1994; Buntine, 1990) provides a potential explanation for their success, and an optimal method for combining models. In the Bayesian view, using a single model to make predictions ignores the uncertainty left by finite data as to which is the "correct" model; thus all possible models in the model space under consideration should be used when making predictions, with each model weighted by its probability of being the "correct" model. This *posterior* probability is the product of the model's *prior* probability, which reflects our domain knowledge (or assumptions) before collecting data, and the *likelihood*, which is the probability of the data given the model. This method of making predictions is called *Bayesian model averaging*.

Given the "correct" model space and prior distribution, Bayesian model averaging is the optimal method for making predictions; in other words, no other approach can consistently achieve lower error rates than it does. Bayesian model averaging has been claimed to obviate the overfitting problem, by "canceling out" the effects of different overfitted models until only the true "main" effect remains (Buntine, 1990). In spite of this, it has not been widely used in machine learning (some exceptions are Buntine (1990); Oliver and Hand (1995); Ali and Pazzani (1996)). In this paper we apply it to the combination of rule models and find that, surprisingly, it produces consistently higher error rates than two *ad hoc* methods (bagging and partitioning). Several possible causes for this are empirically rejected, and we show that Bayesian model averaging performs poorly because, contrary to previous belief, it is highly sensitive to overfitting.

The next section briefly introduces the basic notions of Bayesian theory, and their application to classification problems. We then describe the experiments conducted and discuss the results.

2. Bayesian Model Averaging in Classification

In classification learning problems, the goal is to correctly predict the classes of unseen examples given a training set of classified examples. Given a space of possible models, classical statistical inference selects the single model with highest likelihood given the training data and uses it to make predictions. Modern Bayesian learning differs from this approach in two main respects (Buntine, 1990; Bernardo & Smith, 1994; Chickering & Heckerman, 1997): the computation of posterior probabilities from prior probabilities and likelihoods, and their use in model averaging. In Bayesian theory, each candidate model in the model space is explicitly assigned a prior probability, reflecting our subjective degree of belief that it is the "correct" model, prior to seeing the data. Let n be the training set size, $\vec{x}$ the examples in the training set, $\vec{c}$ the corresponding class labels, and h a model (or hypothesis) in the model space H. Then, by Bayes' theorem, and assuming the examples are drawn independently, the *posterior probability* of h given $(\vec{x}, \vec{c})$ is given by:

$$Pr(h|\vec{x}, \vec{c}) = \frac{Pr(h)}{Pr(\vec{x}, \vec{c})} \prod_{i=1}^{n} Pr(x_i, c_i|h) \quad (1)$$

where $Pr(h)$ is the *prior probability* of h, and the product of $Pr(x_i, c_i|h)$ terms is the *likelihood*. The *data prior* $Pr(\vec{x}, \vec{c})$ is the same for all models, and can thus be ignored. In order to compute the likelihood it is necessary to compute the probability of a class label c_i given an unlabeled example x_i and a hypothesis h, since $Pr(x_i, c_i|h) = Pr(x_i|h)Pr(c_i|x_i, h)$. This probability, $Pr(c_i|x_i, h)$, can be called the *noise model*, and is distinct from the classification model h, which simply produces a class prediction with no probabilities attached.[1]

In the literature on computational learning theory, a *uniform class noise model* is often assumed (see, e.g., Kearns and Vazirani (1994)). In this model, each example's class is corrupted with probability ϵ, and thus $Pr(c_i|x_i, h) = 1 - \epsilon$ if h predicts the correct class c_i for x_i, and $Pr(c_i|x_i, h) = \epsilon$ if h predicts an incorrect class. Equation 1 then becomes:

$$Pr(h|\vec{x}, \vec{c}) \propto Pr(h)\ (1 - \epsilon)^s \epsilon^{n-s} \quad (2)$$

where s is the number of examples correctly classified by h, and the noise level ϵ can be estimated by the models' average error rate. An alternative approach (Buntine, 1990) is to rely on the fact that, implicitly or explicitly, every classification model divides the instance space into regions, and labels each region with a class. For example, if the model is a decision tree (Quinlan, 1993), each leaf corresponds to a region. A class probability can then be estimated separately for each region, by making:

$$Pr(c_i|x_i, h) = \frac{n_{r,c_i}}{n_r} \quad (3)$$

where r is the region x_i is in, n_r is the total number of training examples in r, and n_{r,c_i} is the number of examples of class c_i in r.

Finally, an unseen example x is assigned to the class that maximizes:

$$Pr(c|x, \vec{x}, \vec{c}, H) = \sum_{h \in H} Pr(c|x, h)\ Pr(h|\vec{x}, \vec{c}) \quad (4)$$

[1]Since a classification model h does not predict the example distribution $Pr(x_i|h)$, but only the class given x_i, $Pr(x_i|h) = Pr(x_i)$, and is therefore the same for all models and can be ignored.

If a "pure" classification model is used, $Pr(c|x, h)$ is 1 for the class predicted by h for x, and 0 for all others. Alternatively, a model supplying class probabilities such as those in Equation 3 can be used. Since there is typically no closed form for Equation 4, and the model space used typically contains far too many models to allow the full summation to be carried out, some procedure for approximating Equation 4 is necessary. Since $Pr(h|\vec{x}, \vec{c})$ is often very peaked, using only the model with highest posterior can be an acceptable approximation. Alternatively, a sampling scheme can be used. Two widely-used methods are importance sampling (Bernardo & Smith, 1994) and Markov chain Monte Carlo (Neal, 1993; Gilks et al., 1996).

3. Bayesian Model Averaging of Bagged C4.5 Rule Sets

Bagging (Breiman, 1996a) is a simple and effective way to reduce the error rate of many classification learning algorithms. For example, in the empirical study described below, it reduces the error of a rule learner in 19 of 26 databases, by 4% on average. In the bagging procedure, given a training set of size s, a "bootstrap" replicate of it is constructed by taking s samples *with replacement* from the training set. Thus a new training set of the same size is produced, where each of the original examples may appear once, more than once, or not at all. The learning algorithm is then applied to this sample. This procedure is repeated m times, and the resulting m models are aggregated by uniform voting (i.e., all models have equal weight).

Bagging can be viewed as a form of importance sampling (Bernardo & Smith, 1994). Suppose we want to approximate the sum (or integral) of a function $f(x)p(x)$ by sampling, where $p(x)$ is a probability distribution. This is the form of Equation 4, with $p(x)$ being the model posterior probabilities. Since, given a probability distribution $q(x)$ (known as the importance sampling distribution),

$$\sum f(x)p(x) = \sum f(x) \left[\frac{p(x)}{q(x)}\right] q(x) \quad (5)$$

we can approximate the sum by sampling according to $q(x)$, and computing the average of $f(x_i)p(x_i)/q(x_i)$ for the points x_i sampled. Thus each sampled value $f(x_i)$ will have a weight equal to the ratio $p(x_i)/q(x_i)$. In particular, if $p(x) = q(x)$ all samples should be weighed equally. This is what bagging does. Thus it will be a good importance-sampling approximation of Bayesian model averaging to the extent that the learning algorithm used to generate samples does so according to the model posterior probability. Since most classification learners find a model by minimizing a function of the empirical error, and posterior probability decreases with it, this is a reasonable hypothesis.[2]

Given that bagging can be viewed as an approximation of Bayesian model averaging by importance sampling, and the latter is the optimal prediction procedure, then modifying bagging to more closely approximate it should lead to further error reductions. One way to do this is by noticing that in practice the probability of the same model being sampled twice when sampling from a very large model space is negligible. (This was verified in the empirical study below, and has been noted by many authors (e.g., Breiman (1996a); Charniak (1993).) In this case, weighting models by their posteriors leads to a better approximation of Bayesian model averaging than weighting them uniformly. This can be shown as follows. With either method, the models that were not sampled make the same contribution to the error in approximating Equation 4, so they can be ignored for purposes of comparing the two methods. When weighting models by their posteriors the error in approximating the terms that were sampled is zero, because the exact values of these terms are obtained (modulo errors in the $Pr(c|x,h)$ factors, which again are the same for both methods). With uniform weights the error cannot be zero for all sampled terms, unless they all have exactly the same posterior. Thus the approximation error when using posteriors as weights is less than or equal to the error for uniform weights (with equality occurring only in the very unlikely case of all equal posteriors). Note that, unlike importance sampling, using the posteriors as weights would not yield good approximations of Bayesian model averaging in the large sample limit, since it would effectively weight models by the square of their posteriors. However, for the reasons above it is clearly preferable for realistic sample sizes (e.g., the 10 to 100 models typically used in bagging and other multiple model methods).

[2]Of course, this ignores a number of subtleties. One is that, given such a learner, only models at local maxima of the posterior will have a nonzero probability of being selected. But since most of the probability mass is concentrated around these maxima (in the large-sample limit, all of it), this is a reasonable approximation. Another subtlety is that in this regime the probability of each locally MAP model being selected depends on the size of its basin of attraction in model space, not the amplitude of its posterior. Thus we are implicitly assuming that higher peaks of the posterior will typically have larger basins of attraction than lower ones, which is also reasonable. In any case, none of what follows depends on these aspects of the approximation, since they will be equally present in all the alternatives compared. Also, in Sections 5 and 6 we report on experiments where the model space was exhaustively sampled, making these considerations irrelevant.

Whether a better approximation of Bayesian model averaging leads to lower errors can be tested empirically by comparing bagging (i.e., uniform weights) with the better approximation (i.e., models weighted by their posteriors). In order to do this, a base learner is needed to produce the component models. The C4.5 release 8 learner with the C4.5RULES postprocessor (Quinlan, 1993) was used for this purpose. C4.5RULES transforms decision trees learned by C4.5 (Quinlan, 1993) into rule sets, and tends to be slightly more accurate. A non-informative, uniform prior was used. This is appropriate, since all rule sets are induced in a similar manner from randomly selected examples, and there is thus no *a priori* reason to suppose one will be more accurate than another.

Twenty-six databases from the UCI repository were used (Blake & Merz, 2000). Bagging's error rate was compared with that obtained by weighting the models according to Equation 1, using both a uniform class noise model (Equation 2) and Equation 3. Equation 4 was used in both the "pure classification" and "class probability" forms described. Error was measured by ten-fold cross-validation. Every version of the closer approximation of Bayesian model averaging performed worse than "pure" bagging on a large majority of the data sets (e.g., 19 out of 26), and worse on average. The best-performing combination was that of uniform class noise and "pure classification." Results for this version (labeled "BMA"), bagging and the single model learned from the entire training set in each fold are shown in Table 1. For this version, the experiments were repeated with $m = 10$, 50, and 100, with similar results. Inspection of the posteriors showed them to be extremely skewed, with a single rule model typically dominating to the extent of dictating the outcome by itself. This occurred even though all models tended to have similar error rates. In other words, Bayesian model averaging effectively performed very little averaging, acting more like a model selection mechanism (i.e., selecting the most probable model from the m induced).

4. Bayesian Model Averaging of Partitioned RISE Rule Sets

The surprising results of the previous section might be specific to the bagging procedure, and/or to the use of C4.5 as the base learner. In order to test this, this section reports similar experiments using a different multiple model method (partitioning) and base learner (RISE).

RISE (Domingos, 1996a) is a rule induction system that assigns each test example to the class of the near-

Table 1. Bayesian model averaging of bagged C4.5 rule sets: average error rates and their standard deviations.

Database	Single	Bagging	BMA
Annealing	6.5±0.7	5.1±0.3	5.6±0.7
Audiology	26.5±2.8	23.0±2.1	24.0±2.3
Breast cancer	31.2±4.5	29.7±2.8	37.1±3.4
Credit	14.3±0.9	12.8±1.1	17.8±1.2
Diabetes	25.1±1.7	24.2±1.9	27.5±1.8
Echocardio	33.5±4.2	29.7±4.6	34.3±4.0
Glass	34.1±3.0	22.9±3.2	29.4±2.8
Heart	22.1±1.8	17.2±1.5	23.1±1.9
Hepatitis	19.9±4.2	16.0±4.2	22.5±4.3
Horse colic	16.3±1.3	14.0±1.7	16.7±1.7
Iris	5.3±1.7	5.3±1.7	6.7±2.0
LED	41.0±3.5	39.0±5.2	40.0±4.9
Labor	19.7±4.2	9.0±3.9	12.3±3.7
Lenses	20.0±6.9	23.3±6.7	26.7±7.9
Liver	33.4±2.1	25.8±2.1	33.0±2.2
Lung cancer	45.0±12.2	55.0±10.0	44.2±10.2
Lymphogr.	19.7±2.8	23.7±4.4	19.0±3.7
Post-oper.	31.1±6.2	37.8±6.0	34.4±5.9
Pr. tumor	59.0±2.3	56.3±2.1	56.3±1.7
Promoters	18.3±3.5	13.4±3.3	17.1±1.9
Solar flare	28.8±2.8	30.6±3.1	29.7±2.1
Sonar	24.6±2.8	19.7±2.8	27.3±3.2
Soybean	0.0±0.0	2.0±2.0	2.0±2.0
Voting	4.4±1.1	3.2±0.7	4.6±0.6
Wine	11.2±3.2	6.7±2.0	11.3±2.4
Zoo	9.9±3.0	9.0±3.1	7.0±2.6

est rule according to a similarity measure, and thus implicitly partitions the instance space into the regions won by each of the rules. Its learning time on large databases can be much reduced by randomly partitioning the database into several smaller ones and learning a model on each one separately (Domingos, 1996b). Partitioning can be viewed as an importance-sampling approximation of Bayesian model averaging in the same way that bagging can. Given an unseen example, partitioned RISE classifies it by letting the multiple models induced vote, with each model's vote given by Equation 3 (with the Laplace correction (Niblett, 1987; Good, 1965)). This approach was compared with Bayesian model averaging (i.e., weighing predictions by the posterior probabilities of the corresponding models, using Equations 1 and 3) on eight of the larger databases in the UCI repository. As with bagging, and for similar reasons, uniform priors were used. In the shuttle domain, the pre-defined training and test sets (corresponding to different shuttle flights) were used. For all other databases, ten runs were carried out, in each run randomly dividing the data into two-thirds for training and one-third for testing. The average error rates and their standard deviations are shown in Table 2. The "Single" column shows the error rate obtained by learning on the full database at once, without partitioning. The next two columns show the results obtained by learning on partitions of 100 examples each and combining the resulting models using RISE's method and Bayesian model averaging.

Table 2. Bayesian model averaging of partitioned RISE rule sets: average error rates and their standard deviations.

Database	Single	Partitioning	BMA
Credit	17.4±0.5	13.6±0.6	14.6±0.4
Diabetes	28.4±0.8	25.6±0.7	30.6±0.8
Annealing	2.5±0.3	6.4±0.5	8.8±0.5
Chess	1.6±0.2	5.5±0.2	6.1±0.3
Hypothyroid	2.1±0.1	3.0±0.1	3.3±0.3
Splice	7.5±0.3	5.0±0.2	6.8±0.4
Mushroom	0.0±0.0	1.1±0.0	6.8±0.4
Shuttle	0.0	0.5	0.7

Model averaging produced higher error rates than RISE's method in every domain. As in the previous section, inspection of the posteriors typically showed a single rule model dominating to the extent of dictating the outcome by itself. Thus the observations that were made for bagging C4.5 rule sets are also valid for RISE with partitioning.

5. Bayesian Model Averaging of Foreign Exchange Trading Rules

In the previous sections, Bayesian model averaging could not be applied in its ideal form, due to the very large number of possible models, and this might be the reason for its disappointing performance. Conceivably, if all the terms in Equation 4 were included, the single most probable model would no longer dominate to the point of single-handedly determining the predictions made. This issue can be addressed by applying model averaging in model spaces that are sufficiently restricted for the exact computation of Equation 4 to be feasible. One significant application where these arise is foreign exchange prediction, where the goal is to maximize the return from investing in a foreign currency by predicting whether it will rise or fall against the US dollar. An approach that is used by some traders, and that has been validated by large-scale empirical studies (LeBaron, 1991), involves the use of so-called *technical rules* of the form "If the s-day moving average of the currency's exchange rate rises above the t-day one, buy; else sell." The fact

Table 3. Percent five-year return on investment for four currencies: German mark (DM), British pound (BP), Swiss franc (SF), and Canadian dollar (CD).

Currency	B&H	Best	Unif.	BMA	BMA_P
DM	29.5	47.2	52.4	47.2	47.2
BP	−7.0	7.9	19.5	7.9	8.3
SF	34.4	58.3	44.4	58.3	47.7
CD	−8.3	−2.7	0.0	−5.0	−7.0

that there is clearly no single "right" rule of this type suggests that the use of Bayesian model averaging is appropriate. The choice of s and t, with $t > s$, can be made empirically. If a maximum value t_{max} is set for t (and, in practice, moving averages of more than a month or so are never considered), the total number of possible rules is $t_{max}(t_{max} - 1)/2$. It is thus possible to compare the return yielded by the single most accurate rule with that yielded by averaging *all* possible rules according to their posterior probabilities. These are computed assuming a uniform prior on rules/hypotheses and ignoring terms that are the same for all rules (see Equation 1):

$$Pr(h|\vec{x}, \vec{c}) \propto \prod_{i=1}^{n} Pr(c_i|x_i, h) \tag{6}$$

Let the two classes be + (rise/buy) and − (fall/sell). For each rule h, $Pr(c_i|x_i, h)$ can take only four values: $Pr(+|+)$, $Pr(-|+)$, $Pr(+|-)$ and $Pr(-|-)$. Let n_{-+} be the number of examples in the sample which are of class − but for which rule h predicts +, and similarly for the other combinations. Let $n_{\cdot+}$ be the total number of examples for which h predicts +, and similarly for $n_{\cdot-}$. Then, estimating probabilities from the sample as in Equation 3:

$$\begin{aligned}&\hat{Pr}(h|\vec{x}, \vec{c}) \propto \\ &\left(\frac{n_{++}}{n_{\cdot+}}\right)^{n_{++}} \left(\frac{n_{-+}}{n_{\cdot+}}\right)^{n_{-+}} \left(\frac{n_{+-}}{n_{\cdot-}}\right)^{n_{+-}} \left(\frac{n_{--}}{n_{\cdot-}}\right)^{n_{--}}\end{aligned} \tag{7}$$

Tests were conducted using daily data on five currencies for the years 1973–87, from the Chicago Mercantile Exchange (Weigend et al., 1992). The first ten years were used for training (2341 examples) and the remaining five for testing. A maximum t of two weeks was used. The results for four currencies, in terms of the five-year return on investment obtained, are shown in Table 3. In the fifth currency, the Japanese yen, all averaging methods led to zero return, due to the fact that downward movements were in the majority for all rules both when the rule held and when it did not, leading the program to hold U.S. dollars throughout. This reflects a limitation of making only binary predictions, and not of multiple model methods. The first column of Table 3 shows the result of buying the foreign currency on the first day and holding it throughout the five-year period. The second column corresponds to applying the single best rule, and is a clear improvement over the former. The remaining columns show the results of applying various model averaging methods.

Uniform averaging (giving the same weight to all rules) produced further improvements over the single best rule in all but one currency. However, Bayesian model averaging (fourth column) produced results that were very similar to that of the single best rule. Inspection of the posteriors showed this to be due in each case to the presence of a dominant peak in the (s, t) plane. This occurs even though the rule error rates typically differ by very little, and are very close to chance (error rates below 45% are rare in this extremely noisy domain). Thus it is not the case that averaging over all models in the space will make this phenomenon disappear.

A further aspect in which the applications described so far differ from the exact Bayesian procedure is that averaging was only performed over the classification models, and not over the parameters $Pr(x_i, c_i|h)$ of the noise model. The maximum-likelihood values of these parameters were used in place of integration over the possible parameter values weighted by their posterior probabilities. For example, $Pr(-|+)$ was estimated by $\frac{n_{-+}}{n_{\cdot+}}$, but in theory integration over the [0, 1] interval should be performed, with each probability weighted by its posterior probability given the observed frequencies. Although this is again a common approximation, it might have a negative impact on the performance of Bayesian model averaging. Bayesian averaging was thus reapplied with integration over the probability values, using uniform priors and binomial likelihoods. This led to no improvement ("BMA_P" column). We attribute this to the sample being large enough to concentrate most of the posterior's volume around the maximum-likelihood peak. This was confirmed by examining the curves of the posterior distributions.

6. Bayesian Model Averaging of Conjunctions

Because the results of the previous section might be specific to the foreign exchange domain, the following experiment was carried out using artificially generated Boolean domains. Classes were assigned at random to examples described by a features. All conjunctions of 3 of those features were then generated (a total of $a(a-1)(a-2)/6$), and their posterior probabilities were estimated from a random sample composed of half the possible examples. The experiment was repeated ten times for each of $a = 7, 8, 9, \ldots, 13$. Because the class was random, the error rate of both Bayesian model averaging and the best conjunction[3] was always approximately 50%. However, even in this situation of pure noise and no possible "right" conjunction, the posterior distributions were still highly asymmetric (e.g., the average posterior excluding the maximum was on average 14% of the maximum for $a = 7$, and decreased to 6% for $a = 13$). As a result, Bayesian model averaging still made the same prediction as the "best" conjunction on average 83.9% of the time for $a = 7$, decreasing to 64.4% for $a = 13$.[4]

7. The Overfitting Problem in Bayesian Model Averaging

The observations of the previous sections all point to the conclusion that Bayesian model averaging's disappointing results are not the effect of deviations from the Bayesian ideal (e.g., sampling terms from Equation 4, or using maximum likelihood estimates of the parameters), but rather stem from some deeper problem. In this section this problem is identified, and seen to be a form of overfitting that occurs when Bayesian averaging is applied.

The reason Bayesian model averaging produces very skewed posteriors even when model errors are similar, and as a result effectively performs very little averaging, lies in the form of the likelihood's dependence on the sample. This is most easily seen in the case of a uniform noise model. In Equation 2, the likelihood of a model $(1-\epsilon)^s \epsilon^{n-s}$ increases exponentially with the proportion of correctly classified examples s/n. As a result of this exponential dependence, even small random variations in the sample will cause some hypotheses to appear far more likely than others. Similar behavior occurs when Equation 3 is used as the noise model, with the difference that each example's contribution to the likelihood's exponential variation now depends on the region the example is in, instead of being the same for all examples. This behavior will occur even if the "true" values of the noise parameters are known exactly (ϵ in the uniform model, and the $Pr(x_i, c_i|h)$ values in Equation 3). The posterior's exponential sensitivity to the sample is a direct consequence of Bayes' theorem and the assumption that the examples are drawn independently, which is usually a valid one in classification problems. Dependence between the examples will slow the exponential growth, but will only make it disappear in the limit of all examples being completely determined by the first.

To see the impact of this exponential behavior, consider any two of the conjunctions in the previous section, h_1 and h_2. Using the notation of Section 5, for each conjunction $Pr(+|+) = Pr(-|+) = Pr(+|-) = Pr(-|-) = \frac{1}{2}$, by design. By Equation 6, $Pr(h_1|\vec{x},\vec{c})/Pr(h_2|\vec{x},\vec{c}) = 1$. In other words, given a sufficiently large sample, the two conjunctions should appear approximately equally likely. Now suppose that: $n = 4000$; for conjunction h_1, $n_{++} = n_{--} = 1050$ and $n_{-+} = n_{+-} = 950$; and for conjunction h_2, $n_{++} = n_{--} = 1010$ and $n_{-+} = n_{+-} = 990$. The resulting estimates of $\hat{Pr}(+|+), \ldots, \hat{Pr}(-|-)$ for both conjunctions are quite good; all are within 1% to 5% of the true values. However, the estimated ratio of conjunction posteriors is, by Equation 7:

$$\frac{\hat{Pr}(h_1|\vec{x},\vec{c})}{\hat{Pr}(h_2|\vec{x},\vec{c})} = \frac{\left(\frac{1050}{2000}\right)^{1050}\left(\frac{950}{2000}\right)^{950}\left(\frac{950}{2000}\right)^{950}\left(\frac{1050}{2000}\right)^{1050}}{\left(\frac{1010}{2000}\right)^{1010}\left(\frac{990}{2000}\right)^{990}\left(\frac{990}{2000}\right)^{990}\left(\frac{1010}{2000}\right)^{1010}} \simeq 120$$

In other words, even though the two conjunctions should appear similarly likely and have similar weights in the averaging process, h_1 actually has a far greater weight than h_2; enough so, in fact, that Bayes-averaging between h_1 and 100 conjunctions with observed frequencies similar to h_2's is equivalent to always taking only h_1 into account. If only a single parameter $Pr(\pm|\pm)$ were being estimated, a deviation of 5% or more from a true value of $\frac{1}{2}$ given a sample of size 2000 would have a probability of $p_{5\%} = 0.013$ (binomial distribution, with $p = \frac{1}{2}$ and $n = 2000$). This is a reasonably small value, even if not necessarily negligible. However, two independent parameters are being estimated for each model in the space, and the probability of a deviation of 5% or more in any one parameter increases with the number of parameters being estimated (exponentially if they are independent). With $a = 10$, there are 120 conjunctions of 3 Boolean features (Section 6), and thus 240 parameters to estimate. Assuming independence to simplify, this raises the probability of a deviation of 5% or more to $1 - (1 - p_{5\%})^{240} \simeq 0.957$. In other words, it is highly likely to occur. In practice this value will be smaller due to dependences between the conjunctions, but the pattern is clear. In most applications, the model space contains far more than 120 distinct models; for most modern machine learning methods (e.g., rule induction, decision-tree induction, neural

[3] Predicting the class with highest probability given that the conjunction is satisfied when it is (estimated from the sample), and similarly when it is not.

[4] This decrease was not due to a flattening of the posteriors as the sample size increased (the opposite occurred), but to the class probabilities given the value of each conjunction converging to the 50% limit.

networks, instance-based learning), it can be as high as doubly exponential in the number of attributes. Thus, even if only a very small fraction of the terms in Equation 4 is considered, the probability of one term being very large purely by chance is very high, and this outlier then dictates the behavior of Bayesian model averaging. Further, the more terms that are sampled in order to better approximate Equation 4, the more likely such an outlier is to appear, and the more likely (in this respect) Bayesian model averaging is to perform poorly.

This is an example of *overfitting*: preferring a hypothesis that does not truly have the lowest error of any hypothesis considered, but that by chance has the lowest error on the training data (Mitchell, 1997). The observation that Bayesian model averaging is highly prone to overfitting, the more so the better Equation 4 is approximated, contradicts the common belief among Bayesians that it solves the overfitting problem, and that in the limit overfitting cannot occur if Equation 4 is computed exactly (see, e.g., Buntine (1990)). While uniform averaging as done in bagging can indeed reduce overfitting by canceling out spurious variations in the models (Rao & Potts, 1997), Bayesian model averaging in effect acts more like model selection than model averaging, and is equivalent to amplifying the search process that produces overfitting in the underlying learner in the first place. Although overfitting is often identified with inducing "overly complex" hypotheses, this is a superficial view: overfitting is the result of attempting too many hypotheses, and consequently finding a poor hypothesis that appears good (Jensen & Cohen, in press). In this light, Bayesian model averaging's potential to aggravate the overfitting problem relative to learning a single model becomes clear.

The net effect of Bayesian model averaging will depend on which effect prevails: the increased overfitting (small if few models are considered), or the reduction in error potentially obtained by giving some weight to alternative models (typically a small effect, given Bayesian model averaging's highly skewed weights). While Buntine (1990) and Ali and Pazzani (1996) obtained error reductions with Bayesian model averaging in some domains, these were typically small compared with those obtainable using uniform weights (whether in bagging or using Buntine's option trees, as done by Kohavi and Kunz (1997)).[5] Thus, the improvements produced by multiple models in these references would presumably have been greater if the model posteriors had been ignored. The fact that, in the studies with bagging reported in this article, Bayesian model averaging generally did not produce even those improvements may be attributable to the effect of a third factor: the fact that, when bootstrapping is used, each model is effectively learned from a smaller sample, while the multiple models in Buntine (1990) and Ali and Pazzani (1996) were learned on the entire sample, using variations in the algorithm.

Multiple model methods can be placed on a spectrum according to the asymmetry of the weights they produce. Bagging is at one extreme, with uniform weights, and never increases overfitting (Breiman, 1996b). Methods like boosting and stacking produce weights that are more variable, and can sometimes lead to overfitting (see, e.g., Margineantu and Dietterich (1997)). Bayesian model averaging is at the opposite end of the spectrum, producing highly asymmetric weights, and being correspondingly more prone to overfitting.

8. Conclusions

This paper found that, contrary to previous belief, Bayesian model averaging does not obviate the overfitting problem in classification, and may in fact aggravate it. Bayesian averaging's tendency to overfit derives from the likelihood's exponential sensitivity to random fluctuations in the sample, and increases with the number of models considered. The problem of successfully applying it in machine learning remains an open one.

Acknowledgments

This research was partly supported by PRAXIS XXI and NATO. The author is grateful to all those who provided the data sets used in the experiments.

References

Ali, K., & Pazzani, M. (1996). Classification using Bayes averaging of multiple, relational rule-based models. In D. Fisher and H.-J. Lenz (Eds.), *Learning from data: Artificial intelligence and statistics V*, 207–217. New York, NY: Springer.

Bauer, E., & Kohavi, R. (1999). An empirical comparison of voting classification algorithms: Bagging, boosting and variants. *Machine Learning*, *36*, 105–142.

Bernardo, J. M., & Smith, A. F. M. (1994). *Bayesian theory*. New York, NY: Wiley.

Blake, C., & Merz, C. J. (2000). *UCI repository of*

[5]Note also that only one of the many variations of Bayesian model averaging in Buntine (1990) consistently reduced error.

machine learning databases. Department of Information and Computer Science, University of California at Irvine, Irvine, CA. http://www.ics.uci.edu/~mlearn/MLRepository.html.

Breiman, L. (1996a). Bagging predictors. *Machine Learning*, *24*, 123–140.

Breiman, L. (1996b). *Bias, variance and arcing classifiers* (Technical Report 460). Statistics Department, University of California at Berkeley, Berkeley, CA.

Buntine, W. (1990). *A theory of learning classification rules*. Doctoral dissertation, School of Computing Science, University of Technology, Sydney, Australia.

Charniak, E. (1993). *Statistical language learning*. Cambridge, MA: MIT Press.

Chickering, D. M., & Heckerman, D. (1997). Efficient approximations for the marginal likelihood of Bayesian networks with hidden variables. *Machine Learning*, *29*, 181–212.

Domingos, P. (1996a). Unifying instance-based and rule-based induction. *Machine Learning*, *24*, 141–168.

Domingos, P. (1996b). Using partitioning to speed up specific-to-general rule induction. *Proceedings of the AAAI-96 Workshop on Integrating Multiple Learned Models for Improving and Scaling Machine Learning Algorithms* (pp. 29–34). Portland, OR: AAAI Press.

Drucker, H., Cortes, C., Jackel, L. D., LeCun, Y., & Vapnik, V. (1994). Boosting and other machine learning algorithms. *Proceedings of the Eleventh International Conference on Machine Learning* (pp. 53–61). New Brunswick, NJ: Morgan Kaufmann.

Freund, Y., & Schapire, R. E. (1996). Experiments with a new boosting algorithm. *Proceedings of the Thirteenth International Conference on Machine Learning* (pp. 148–156). Bari, Italy: Morgan Kaufmann.

Gilks, W. R., Richardson, S., & Spiegelhalter, D. J. (Eds.). (1996). *Markov chain Monte Carlo in practice*. London, UK: Chapman and Hall.

Good, I. J. (1965). *The estimation of probabilities: An essay on modern Bayesian methods*. Cambridge, MA: MIT Press.

Jensen, D., & Cohen, P. R. (in press). Multiple comparisons in induction algorithms. *Machine Learning*.

Kearns, M. J., & Vazirani, U. V. (1994). *An introduction to computational learning theory*. Cambridge, MA: MIT Press.

Kohavi, R., & Kunz, C. (1997). Option decision trees with majority votes. *Proceedings of the Fourteenth International Conference on Machine Learning* (pp. 161–169). Nashville, TN: Morgan Kaufmann.

Kong, E. B., & Dietterich, T. G. (1995). Error-correcting output coding corrects bias and variance. *Proceedings of the Twelfth International Conference on Machine Learning* (pp. 313–321). Tahoe City, CA: Morgan Kaufmann.

LeBaron, B. (1991). *Technical trading rules and regime shifts in foreign exchange* (Technical Report). Department of Economics, University of Wisconsin at Madison, Madison, WI.

Maclin, R., & Opitz, D. (1997). An empirical evaluation of bagging and boosting. *Proceedings of the Fourteenth National Conference on Artificial Intelligence*. Providence, RI: AAAI Press.

Margineantu, D., & Dietterich, T. (1997). Pruning adaptive boosting. *Proceedings of the Fourteenth International Conference on Machine Learning* (pp. 211–218). Nashville, TN: Morgan Kaufmann.

Mitchell, T. M. (1997). *Machine learning*. New York, NY: McGraw-Hill.

Neal, R. M. (1993). *Probabilistic inference using Markov chain Monte Carlo methods* (Technical Report CRG-TR-93-1). Department of Computer Science, University of Toronto, Toronto, Canada.

Niblett, T. (1987). Constructing decision trees in noisy domains. *Proceedings of the Second European Working Session on Learning* (pp. 67–78). Bled, Yugoslavia: Sigma.

Oliver, J. J., & Hand, D. J. (1995). On pruning and averaging decision trees. *Proceedings of the Twelfth International Conference on Machine Learning* (pp. 430–437). Tahoe City, CA: Morgan Kaufmann.

Quinlan, J. R. (1993). *C4.5: Programs for machine learning*. San Mateo, CA: Morgan Kaufmann.

Quinlan, J. R. (1996). Bagging, boosting, and C4.5. *Proceedings of the Thirteenth National Conference on Artificial Intelligence* (pp. 725–730). Portland, OR: AAAI Press.

Rao, J. S., & Potts, W. J. E. (1997). Visualizing bagged decision trees. *Proceedings of the Third International Conference on Knowledge Discovery and Data Mining* (pp. 243–246). Newport Beach, CA: AAAI Press.

Weigend, A. S., Huberman, B. A., & Rumelhart, D. E. (1992). Predicting sunspots and exchange rates with connectionist networks. In M. Casdagli and S. Eubank (Eds.), *Nonlinear modeling and forecasting*, 395–432. Redwood City, CA: Addison-Wesley.

Wolpert, D. (1992). Stacked generalization. *Neural Networks*, *5*, 241–259.