Why Does Bagging Work? A Bayesian Account and its Implications

Pedro Domingos[1,2]
Department of Information and Computer Science
University of California, Irvine
Irvine, California 92697, U.S.A.
pedrod@ics.uci.edu
http://www.ics.uci.edu/~pedrod

Abstract

The error rate of decision-tree and other classification learners can often be much reduced by *bagging*: learning multiple models from bootstrap samples of the database, and combining them by uniform voting. In this paper we empirically test two alternative explanations for this, both based on Bayesian learning theory: (1) bagging works because it is an approximation to the optimal procedure of Bayesian model averaging, with an appropriate implicit prior; (2) bagging works because it effectively shifts the prior to a more appropriate region of model space. All the experimental evidence contradicts the first hypothesis, and confirms the second.

Bagging

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 decision-tree 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. On average, 63% of the original examples will appear in the bootstrap sample. The learning algorithm is then applied to this training set. This procedure is repeated m times, and the resulting m models are aggregated by uniform voting. Bagging is one of several "multiple model" approaches that have recently received much attention (see, for example, (Chan, Stolfo, & Wolpert 1996)). Other procedures of this type include boosting (Freund & Schapire 1996) and stacking (Wolpert 1992).

Two related explanations have been proposed for bagging's success, both in a classical statistical framework.

Breiman (1996a) relates bagging to the notion of an *order-correct* learner. A learner is order-correct for an example x if, given many different training sets, it predicts the correct class for x more often than any other. Breiman shows that, given sufficient replicates, bagging turns an order-correct learner into a nearly-optimal one. Although this line of reasoning has intuitive value, its usefulness is limited, because it is seldom (or never) known *a priori* whether a learner is order-correct for a given example or not, or what regions of the instance space it will be order-correct in and not. Thus it is not possible to judge from an application domain's characteristics whether bagging will be successful in it or not. On the other hand, Breiman provides a qualitative description of the learners with which bagging can be expected to work: they have to be unstable, in the sense that small variations in the training set can lead them to produce very different models. Decision trees and neural networks are examples of such learners. In contrast, nearest-neighbor methods are stable, and bagging is of little value when applied to them.

In a related study, Friedman (1996) relates the success of bagging to the notions of *bias* and *variance* of a learning algorithm. Several alternative definitions of bias and variance for classification learners have been proposed (Kong & Dietterich 1995; Kohavi & Wolpert 1996; Breiman 1996b; Friedman 1996). Loosely, bias measures the systematic component of a learner's error (i.e., its average error over many different training sets), and variance measures the additional error that is due to the variation in the model produced from one training set to another. Friedman suggests that bagging works by reducing variance without changing the bias. Again, this explanation has intuitive value, but leaves unanswered the question of how the success of bagging relates to domain characteristics.

In this paper, an alternate line of reasoning is pursued, one that draws on Bayesian learning theory (Buntine 1990; Bernardo & Smith 1994). In this theory, knowledge of the domain is (or assumptions about it are) contained in the prior probability assigned to the

[1] Partly supported by a NATO scholarship.

different models in the model space under consideration, the training set is taken into account by computing the likelihood of each model given the data, and optimal classification is performed by voting among the models, with each model's weight being its posterior probability (the product of the prior probability and the likelihood). Simply choosing the single best model, as most learners do, is only an approximation to the optimal procedure. Since bagging effectively samples the model space, and then performs classification by voting among the models found, the question naturally arises: can bagging's success be due to its being a closer approximation to the optimal Bayesian procedure than simply choosing the best model? And if so, what prior assumptions does its uniform voting procedure imply? Alternatively, bagging can be regarded as forming a single new model composed of the m bagged models, and choosing it. In this case, bagging in effect changes the model space, or at least redistributes probability from the component models to the composite ones. These two interpretations (bagging approximates Bayesian model averaging vs. bagging changes the priors) have very different implications. In the first, the underlying learner is assumed to fit the domain well, and error is reduced by refining the inference procedure (averaging models, instead of selecting one). In the second, inference according to a single model is an acceptable approximation, but the underlying learner is making incorrect assumptions about the domain, and error is reduced by changing it.

This paper tests these two hypotheses empirically, and discusses the implications of the results. After a brief review of Bayesian theory, the first hypothesis is tested, followed by the second one.

Bayesian Learning Theory

Modern Bayesian approaches to learning differ from classical statistical ones in two main respects (Buntine 1990): 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's theorem, and assuming the examples are drawn independently, the posterior probability of h given $(\vec{x}, \vec{c})$ is given by:

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

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. If a *uniform class noise model* is assumed (i.e., each example's class is corrupted with probability ϵ), $Pr(x_i, c_i|h) = 1 - \epsilon$ if h predicts the correct class c_i for x_i, and $Pr(x_i, c_i|h) = \epsilon$ if h predicts an incorrect class. Equation 1 then becomes:

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

where s is the number of examples correctly classified by h. An alternative approach (Buntine 1990) relies on the fact that, implicitly or explicitly, a 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 noise level can then be estimated separately for each region, by making:

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

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, a test example x is assigned to the class that maximizes:

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

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 (e.g., Markov chain Monte Carlo (Madigan *et al.* 1996)) can be used.

Empirical Tests of the First Hypothesis

This section empirically tests the following hypothesis:

1. *Bagging reduces a classification learner's error rate because it more closely approximates Equation 4 than the single model output by the learner.*

Obviously, this hypothesis assumes the learner indeed outputs a single model, and is only relevant for those *(learner, domain)* pairs where bagging does in fact reduce error. If this hypothesis is correct, given m bootstrap replicates of the training set, bagging samples m high-posterior terms from Equation 4. This is plausible, since a sensible learner will produce a high-posterior model given its training set. More problematic is the fact that bagging assigns uniform weight to all models sampled, i.e., it assumes they all have the same posterior probability, irrespective of their likelihood. One possible interpretation (Hypothesis 1a) is that this is simply an imperfection in bagging's approximation, and better results would be obtained by correctly weighing the models by their posteriors, assuming (conservatively) an uninformed prior (e.g., uniform). Alternatively, bagging may be regarded as assuming a prior probability distribution that approximately cancels the likelihood for these models (Hypothesis 1b). Both these variants of Hypothesis 1 were tested.

A decision-tree learner, C4.5 release 8 (Quinlan 1993), was used in all experiments.[3] Twenty-six databases from the UCI repository were used[4] (Merz, Murphy, & Aha 1997). Hypothesis 1a was tested by comparing bagging's error with that obtained by weighing 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. Initially, $m = 25$ bootstrap replicates were used. Error was measured by ten-fold cross-validation. In all cases, non-uniform weighting performed worse than bagging on a large majority of the datasets (e.g., 19 out of 26), and worse on average. The best-performing combination was that of uniform class noise and "pure classification." For this version, the experiments were repeated with $m = 10$, 50, and 100, with similar results. Since Bayesian model averaging with a uniform prior consistently performs worse than bagging, it is unlikely that bagging works because it is an approximation to that procedure, and the empirical evidence thus contradicts Hypothesis 1a.

If Hypothesis 1b is correct, exactly what prior is being assumed by bagging? Since the likelihood decreases exponentially with training-set error (see Equation 2, and consider that $\epsilon \leq \frac{1}{2}$ by definition, and the training-set error is $(n-s)/n$), a prior that cancels the likelihood must increase exponentially with training-set error, at least in the region(s) of model space occupied by the models sampled by bootstrapping. This use of training-set information in the prior is not strictly allowed by Bayesian theory, but is nevertheless common (Cheeseman 1990). Although counter-intuitive, penalizing models that have lower error on the training data simply corresponds to an assumption that the models *overfit* the data, or more precisely, that the models that have lower error on the training data will in fact haver higher error on test data. Since learners that overfit are also necessarily unstable learners, or learners with high variance, and these are the learners for which Breiman (1996a) and Friedman (1996) found bagging will work, it is plausible that bagging incorporates a prior that is appropriate to those learners. Whether this assumption that bagging incorporates an error-favoring prior is correct for the databases and learner used can be tested by checking the sign and magnitude of the correlation between each model's in-bag error (i.e., on the data it was learned on) and out-of-bag error (i.e., on the remaining data). Doing this results in the observation that, although the models are almost always overfitted in the sense that their error is lower in-bag than out-of-bag, the correlation between in-bag and out-of-bag error is positive in all but four of the 26 databases, and is greater than 0.5 in half the databases where it is positive. Thus lower in-bag error is almost always accompanied by lower out-of-bag error, and using an error-favoring prior should increase error, not reduce it, as bagging does. This evidence contradicts Hypothesis 1b.

Empirical Tests of the Second Hypothesis

This section empirically tests the following hypothesis:

2. *Bagging reduces a classification learner's error rate because it changes the learner's model space and/or prior distribution to one that better fits the domain.*

As before, a learner outputting a single model is assumed, and this hypothesis is only relevant in cases where bagging indeed reduces error. Most learners do not employ an explicit prior distribution on models. However, since they invariably have learning biases that lead them to induce some models more often than others, and these biases are known before the learner is applied to any given training set, these biases can be considered to imply a corresponding prior probability distribution. Specifically, most decision tree and rule learners (including C4.5) incorporate a *simplicity bias*: they give preference to simpler models, on the assumption that these models will have lower error on a test set than more complex ones, even if they have higher error on the training set. Simplicity can be measured in different ways, but is typically related to the size of the decision tree or rule set produced (e.g., the number of nodes in the tree, or number of conditions in all the rules). Because, given m replicates, bagging produces m models, its output is (in naive terms) on the order of m times more complex that that of the base learner. A plausible hypothesis is then that, when it works, bagging is in effect counteracting an inappropriate simplicity bias, by shifting probability to more complex models.

If this instantiation of Hypothesis 2 is to be effectively tested, bagging's output complexity must be more carefully evaluated. Syntactically, sets of decision trees constitute a different model space from individual decision trees, even if the set of classifications they can represent is the same, and thus it is questionable to directly compare the complexity of models

[3] The C4.5RULES post-processor was used, since it tended to reduce error.

[4] Audiology, annealing, breast cancer, credit, diabetes, echocardiogram, glass, heart disease, hepatitis, horse colic, iris, labor, lenses, LED, lung cancer, liver disease, lymphography, post-operative, promoters, primary tumor, solar flare, sonar, soybean, voting, wine, and zoology.

in the two representations. A bagged set of m decision trees $\{DT_1, DT_2, \ldots, DT_m\}$ can be trivially transformed into a single decision tree by placing a replica of DT_2 at each of DT_1's leaves, then a replica of DT_3 at each leaf of each replica of DT_2, and so on up to DT_m, and then labeling each leaf node of the composite tree with the class assigned to it by the bagged ensemble. However, this produces a decision tree whose complexity is exponential in m, and is likely to be far more complex than necessary to replicate the bagged ensemble's classification decisions. Rather, we would like to find the simplest decision tree extensionally representing the same model as a bagged ensemble, and compare its complexity with that of the single tree induced from the whole training set. Although this is likely to be an NP-complete problem (Hyafil & Rivest 1976), an approximation to this approach can be obtained by simply applying the base learner to a training set composed of a large number of examples generated at random, and classified according to the bagged ensemble. The decision tree produced in this way models the bagged ensemble's division of the instance space into class regions. Exactly the same simplicity bias is applied to learning this tree as to learning a tree directly from the original training set, making the complexities of the two directly comparable.

This "meta-learning" procedure was carried out for the 26 databases previously mentioned, using C4.5 as before. Details and full results are given elsewhere (Domingos 1997). In all but four of the 22 databases where bagging improves on the single rule set, meta-learning also produces a rule set with lower error, with over 99% confidence according to sign and Wilcoxon tests. Its error reductions are on average 60% of bagging's, indicating that the rule set produced is only an approximation of the bagged ensemble's behavior. Measuring complexity as the total number of antecedents and consequents in all rules, the new rule set is more complex than the directly-learned one in every case, typically by a factor of 2 to 6. Moreover, varying the pruning level during meta-learning and the size of the meta-learning training set lead to the observation that error and complexity are inversely correlated (i.e., complexity tends to increase when error decreases). Thus the empirical evidence agrees with the hypothesis that bagging works by effectively changing a single-model learner to another single-model learner, with a different implicit prior distribution over models, one that is less biased in favor of simple models.

Conclusion

This paper tested two alternative explanations for bagging's success. Given the empirical evidence, it is unlikely that bagging works because it is an approximation to Bayesian model averaging, and it is plausible that it works at least in part because it corrects for an overly-strong simplicity bias in the underlying learner.

References

Bernardo, J. M., and Smith, A. F. L. 1994. *Bayesian Theory.* New York, NY: Wiley.

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. L. 1990. *A Theory of Learning Classification Rules.* Ph.D. Dissertation, School of Computing Science, University of Technology, Sydney, Australia.

Chan, P.; Stolfo, S.; and Wolpert, D., eds. 1996. *Proc. AAAI-96 Workshop on Integrating Multiple Learned Models.* Portland, OR: AAAI Press.

Cheeseman, P. 1990. On finding the most probable model. In Shrager, J., and Langley, P., eds., *Computational Models of Scientific Discovery and Theory Formation.* San Mateo, CA: Morgan Kaufmann.

Domingos, P. 1997. Knowledge acquisition from examples via multiple models. In *Proc. Fourteenth International Conference on Machine Learning.* Nashville, TN: Morgan Kaufmann.

Freund, Y., and Schapire, R. E. 1996. Experiments with a new boosting algorithm. In *Proc. Thirteenth International Conference on Machine Learning*, 148–156. Bari, Italy: Morgan Kaufmann.

Friedman, J. H. 1996. On bias, variance, 0/1 - loss, and the curse-of-dimensionality. Technical report, Department of Statistics and Stanford Linear Accelerator Center, Stanford University, Stanford, CA.

Hyafil, L., and Rivest, R. L. 1976. Constructing optimal binary decision trees is NP-complete. *Information Processing Letters* 5:15–17.

Kohavi, R., and Wolpert, D. H. 1996. Bias plus variance decomposition for zero-one loss functions. In *Proc. Thirteenth International Conference on Machine Learning*, 275–283. Bari, Italy: Morgan Kaufmann.

Kong, E. B., and Dietterich, T. G. 1995. Error-correcting output coding corrects bias and variance. In *Proc. Twelfth International Conference on Machine Learning*, 313–321. Tahoe City, CA: Morgan Kaufmann.

Madigan, D.; Raftery, A. E.; Volinsky, C. T.; and Hoeting, J. A. 1996. Bayesian model averaging. In *Proc. AAAI-96 Workshop on Integrating Multiple Learned Models*, 77–83. Portland, OR: AAAI Press.

Merz, C. J.; Murphy, P. M.; and Aha, D. W. 1997. UCI repository of machine learning databases. Department of Information and Computer Science, University of California at Irvine, Irvine, CA.

Quinlan, J. R. 1993. *C4.5: Programs for Machine Learning.* San Mateo, CA: Morgan Kaufmann.

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