Turning Bayesian Model Averaging Into Bayesian Model Combination

Kristine Monteith, James L. Carroll, Kevin Seppi, and Tony Martinez
kristinemonteith@gmail.com, jlcarroll@lanl.gov, kseppi@byu.edu, and martinez@cs.byu.edu

***Abstract*— Bayesian methods are theoretically optimal in many situations. Bayesian model averaging is generally considered the standard model for creating ensembles of learners using Bayesian methods, but this technique is often outperformed by more *ad hoc* methods in empirical studies. The reason for this failure has important theoretical implications for our understanding of why ensembles work. It has been proposed that Bayesian model averaging struggles in practice because it accounts for uncertainty about which model is correct but still operates under the assumption that only one of them is. In order to more effectively access the benefits inherent in ensembles, Bayesian strategies should therefore be directed more towards model combination rather than the model selection implicit in Bayesian model averaging. This work provides empirical verification for this hypothesis using several different Bayesian model combination approaches tested on a wide variety of classification problems. We show that even the most simplistic of Bayesian model combination strategies outperforms the traditional *ad hoc* techniques of bagging and boosting, as well as outperforming BMA over a wide variety of cases. This suggests that the power of ensembles does not come from their ability to account for model uncertainty, but instead comes from the changes in representational and preferential bias inherent in the process of combining several different models.**

I. Introduction

Learner error can often be reduced by combining information from a set of models. This poses the challenge of finding effective ways to create combinations of learners. A number of *ad hoc* strategies have been proposed to address this task. For example, bagging [1] employs one of the simplest methods of combining the information presented in an ensemble: allowing each learner to have one vote toward the final classification of an instance. Boosting [2], attempts to focus on harder instances during the course of training, and votes are weighted by the accuracy that a given learner achieves on the data set.

One possible explanation for the success of ensemble learners is based on Bayesian learning theory [3]. Supposedly, using a single model for learning ignores the uncertainty about model correctness that results from a finite amount of data. Under this assumption, ensembles work because they can more effectively deal with this uncertainty about model correctness. Strategies such as bagging compensate for this uncertainty simply by incorporating a set of models into the learning process while Bayesian model averaging (BMA) should provide the "optimal" ensemble procedure. Bayesian model averaging accounts for uncertainty of model correctness by integrating over the model space and weighting each model by the probability of its being the "correct" model. BMA is the generally accepted method for applying Bayesian learning theory to the task of model combination. Although the result of BMA is a combination of models, this combination is actually just integrating out the system's uncertainty as to *which* model is correct in the sense of being the Data Generating Model (DGM) assuming that one and only one of the models is indeed the DGM. Thus, BMA is actually a model *selection* procedure that deals with uncertainty about its selection using a combination.

One might expect Bayesian model averaging to perform well since Bayesian techniques have been applied to many other tasks with high success. For example, even simple single model classifiers such as Naïve Bayes [4] and Flexible Bayes [5] can achieve remarkably high accuracy on certain problems. More complex distributions can be represented by Bayesian mixture models. Sampling techniques such as Markov Chain Monte Carlo can be used to infer parameters in relatively complex models [6]. Specific models are also commonly used for specific tasks. The latent Dirichlet allocation model is commonly used to identify topics present in a set of documents [7]. However, when it comes to the task of ensemble creation, the standard technique of Bayesian model averaging encounters some problems.

In an empirical study, Domingos [8] showed that Bayesian model averaging is prone to higher error rates than more *ad hoc* methods. Specifically, Bayesian model averaging resulted in higher average error rates than bagging and partitioning in a variety of experiments. A similar result was obtained by Clarke [9], who compared BMA to stacking. At first, these results appear to be surprising given the supposed optimality of Bayesian techniques and their success in so many other areas.

Domingos argued that the problem with BMA is that it places too much weight on the maximum likelihood classifier. Even slight differences in error rate between classifiers result in much higher weighting of the more accurate classifier in the ensemble. Yet Bayesian model averaging is theoretically the optimal method for dealing with uncertainty about which hypothesis in the hypothesis space is correct. Given the superior performance of *ad hoc* methods in empirical studies, it would appear that ensemble performance is due to more than just their ability to deal with model uncertainty.

While comparing BMA to stacking, Clarke empirically

This work was partially supported by the Advanced Radiography Science Campaign at Los Alamos National Laboratory. LA-UR 11-02743.

noticed that when the Data Generating Model (DGM) is not one of the component models in the ensemble, BMA tends to converge to the *model* closest to the DGM rather than to the *combination* closest to the DGM [9]. He also empirically noted that, in the cases he studied, when the DGM is not one of the component models of an ensemble, there usually existed a combination of models that could more closely replicate the behavior of the DMG than could any individual model on their own.

Three years earlier, Minka theorized that Bayesian model averaging is outperformed by other strategies because it fails to take advantage of the enriched hypothesis space that an ensemble can provide [10]. If Minka is correct, an ensemble does more than just deal with uncertainty about which model is the correct model; it can augment the hypothesis space with hypotheses that its individual members may not be able to even represent on their own. Further, ensembles may change the preferential bias of a learning algorithm, predisposing the algorithm towards combinations of models that tend to overfit less than single learners. As Minka states in his paper, "...the only flaw with BMA is the belief that it is an algorithm for model combination." Yet, despite this fact, people continue to employ BMA in the very case where BMA is unlikely to perform well, namely the case where the DGM is not one of the component ensemble members [9]. In this situation, Bagging and other *ad hoc* strategies should have an advantage over Bayesian model averaging because they incorporate more information from the enriched hypothesis space provided by an ensemble. This suggests that if Bayesian methods are to be effectively used in ensemble creation strategies, efforts should be directed towards creation of Bayesian mixture models that directly infer the optimal *combination* of the component models. Such strategies would take advantage of both the optimality of Bayesian learning strategies and the error reduction advantages that can result from combinations of models.

There are several ways in which an ensemble combination can be generated using Bayesian principles. Bayesian inference could be used to generate the optimal combination (ensemble member weights) given a set of fixed (and already trained) learners. Alternatively, Bayesian inference could be used to infer the optimal set of component model parameters given a fixed ensemble combination scheme. Finally, these two approaches could be used simultaneously. In this work we will provide empirical evidence for Minka's hypothesis by examining the first two of these three possibilities. In Section II we review Minka's argument that Bayesian model averaging assumes that a single ensemble member is the DGM. Section III then proposes several possibilities for generating ensemble weighs given a set of fixed component models using the same Bayesian principles as BMA, but directing them towards the task of model *combination* instead of model *selection*. More complicated strategies are clearly possible, but even the simple models presented here outperform bagging, boosting, and Bayesian model averaging in terms of error reduction on 50 data sets. As a complement to these techniques, we present a strategy in Section IV that uses Bayesian methods to learn optimal component model parameters given a fixed combination of weights. Again, while there is clear potential for more sophisticated strategies, even this simple one outperforms more *ad hoc* methods of model learning in terms of error reduction.

II. Bayesian Averaging of Linear Combinations of Models

With traditional Bayesian model averaging, the class value assigned to a given example by the overall model is determined by taking the probability of each class value as predicted by a single model, multiplying by the probability that the model is the Data Generating Model (DGM) given a sample of data, and summing these values for all models in the hypothesis space. Let n be the size of a data set D. Each individual example d_i is comprised of a vector of attribute values x_i and an associated class value y_i. The model space is approximated by a finite set of learners, H, with h being an individual hypothesis in that space. Equation 1 illustrates how the probability of a class value is determined for a given example. The class value assigned to the instance will be the one with the maximum probability.

$$p(y_i|x_i, D, H) = \sum_{h \in H} p(y_i|x_i, h)p(h|D) \tag{1}$$

By Bayes' Theorem, the *posterior probability* of h given D (the posterior probability that h is the DGM) can be calculated as shown in Equation 2. Here, $p(h)$ represents the *prior probability* of h and the product of the $p(d_i|h)$ determines the *likelihood*.

$$p(h|D) = \frac{p(h)}{p(D)} \prod_{i=1}^{n} p(d_i|h) \tag{2}$$

Bayesian model averaging strategies commonly assume a *uniform class noise model* when determining likelihood [8]. With this model, the class of each example is assumed to be corrupted with probability ϵ. This means that $p(d_i|h)$ is $1 - \epsilon$ if h correctly predicts class y_i for example x_i and ϵ otherwise. Equation 2 can be rewritten as shown in Equation 3. (Since the prior probability of the data $p(D)$ is the same for each model, the equation becomes a statement of proportionality and $p(D)$ can be ignored.)

$$p(h|D) \propto p(h)(1 - \epsilon)^r(\epsilon)^{n-r} \tag{3}$$

r is the number of examples correctly classified by h. ϵ can be estimated by the average error rate of the model on the data. This method of calculating likelihood tends to weight even slightly more accurate classifiers much more heavily. For example, on a data set with 100 examples, a learner that achieved 95% accuracy would be weighted as 17 times more likely than a learner that achieved an accuracy of 94%.

$$(1 - \tfrac{5}{100})^{95}(\tfrac{5}{100})^5 = 2.39 * 10^{-9}$$
$$(1 - \tfrac{6}{100})^{94}(\tfrac{6}{100})^6 = 1.39 * 10^{-10}$$

Using these posterior probabilities to weight learner classifications is clearly an effective way of exploiting the model with the highest accuracy while still allowing influence from other models to account for the uncertainty about which model is correct. It is somewhat ineffective, however, at taking advantage of information provided by the entire set of models [9]. If the goal is to use optimal Bayesian techniques and still capitalize on the possible advantages inherent in learner combinations, these techniques could be modified in order to produce optimal methods of model combination rather than model selection.

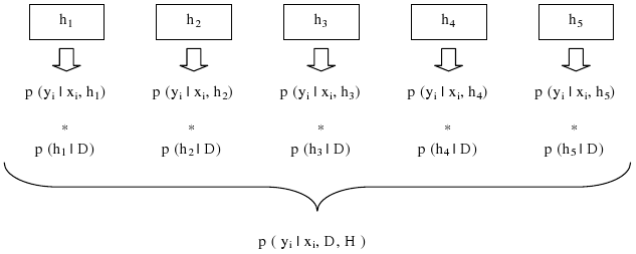

Fig. 1. Bayesian model averaging. Since the probability of the most likely hypothesis is often much higher than the probability of the other hypothesis, $p(y_i|x_i, D, H)$ will be predominantly determined by $p(h_{mostLikely}|D)$.

III. Bayesian Model Combination

Bayesian model averaging can easily be modified to produce an optimal technique for model combination rather than model selection. This strategy is referred to here as Bayesian model combination (BMC). Equation 1 is modified as follows:

$$p(y_i|x_i, D, H, E) = \sum_{e \in E} p(y_i|x_i, H, e)p(e|D) \qquad (4)$$

where e is an element in the space E of possible model combinations. In this case, the outputs from individual hypotheses are combined in a variety of ways to create a set of diverse ensembles. The output from each ensemble is then weighted by the probability that the ensemble is correct given the training data. Now, instead of integrating out uncertainty about which ensemble *member* is correct, we are instead integrating out uncertainty about which *model combination* is correct.

Although the space of potential model combinations is very large, as we shall see, it can easily be sampled from in order to produce a reasonable finite set of potential model combinations to test.

A. BMC with a Linear Combinations of Models

For the first set of Bayesian model combination experiments, ensembles were created using linear combinations of outputs from the base classifiers. Ensembles consisted of m decision trees whose votes were combined using various weights. In order to systematically generate a diverse collection of ensembles, nested for loops were used to assign incrementally increasing values to the base components.

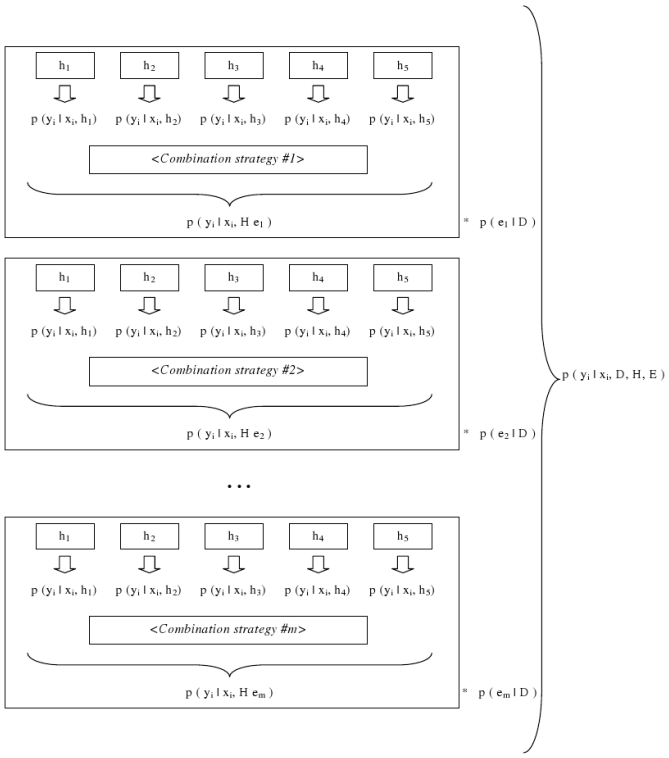

Fig. 2. Bayesian model combination. In this case, $p(y_i|x_i, D, H, E)$ will be predominantly determined by $p(e_{mostLikely}|D)$. The model is now heavily weighting the most probable combination of hypotheses instead of the most probable single hypothesis.

These values were then normalized to produce a vector of weights. Table I illustrates how weights were assigned. For the reported experiments $m = 10$ and ensemble weightings were assigned using an increment value of three. This allowed for the creation of $59,049$ different ensembles from the same ten base classifiers.

TABLE I

Weight assignments for individual components in a simple Bayesian model combination learner. Each component is weighted with a uniform prior in these experiments.

Raw weights	Normalized weights	$p(e)$
1 1 1 1 1 1 1 1 1 1	0.10 0.10 0.10 0.10 0.10 0.10 0.10 0.10 0.10 0.10	$\frac{1}{59049}$
1 1 1 1 1 1 1 1 1 2	0.09 0.09 0.09 0.09 0.09 0.09 0.09 0.09 0.09 0.18	$\frac{1}{59049}$
1 1 1 1 1 1 1 1 1 3	0.08 0.08 0.08 0.08 0.08 0.08 0.08 0.08 0.08 0.25	$\frac{1}{59049}$
1 1 1 1 1 1 1 1 2 1	0.09 0.09 0.09 0.09 0.09 0.09 0.09 0.09 0.18 0.09	$\frac{1}{59049}$
...	...	...
3 3 3 3 3 3 3 3 3 3	0.10 0.10 0.10 0.10 0.10 0.10 0.10 0.10 0.10 0.10	$\frac{1}{59049}$

This version of Bayesian model combination is compared to the strategies of bagging, boosting, and traditional Bayesian model averaging. Experiments were implemented using modified Weka code [11]. Ten J48 decision trees (Weka's implementation of the C4.5 algorithm) with reduced-error pruning were used as the base classifiers in each of the algorithms. Bagging and boosting were implemented using Weka defaults. For bagging, training data for the component classifiers was obtained by drawing with replacement from the initial training set until a new training set the same size

as the original set was created [1]. Training sets for the boosting algorithm were generated in a similar manner, but instances misclassified by initial component classifiers were more likely to appear in the training data for subsequent classifiers [2]. Bayesian model averaging and Bayesian model combination were implemented using the same ten decision trees that were used in the bagging experiments as component classifiers.

Probabilities for class predictions by individual learners and ensembles were estimated using Weka defaults. For the individual J48 decision trees, $p(y_i|x_i, h)$ was estimated based on the purity of classification at the leaf node. For the ensemble, $p(y_i|x_i, e)$ was calculated by averaging probability estimates from the individual trees.

Posterior probabilities for ensembles in the Bayesian model combination approach were estimated the same way they were estimated for individual learners in Bayesian model averaging. Equation 3 can be easily applied to calculate $p(e|D)$ instead of $p(h|D)$. The class of each example is assumed to be corrupted with probability ϵ, so $p(d_i|e)$ is $1 - \epsilon$ if e correctly predicts class y_i for example x_i and ϵ otherwise.

Empirical results, shown in Table II, demonstrate the efficacy of this Bayesian model combination strategy. Experiments were conducted on the twenty-six data sets cited by Domingos, but since this selection of data sets proved insufficient to draw conclusions about the statistical significance of mean differences in accuracy, an additional twenty-four datasets were included. All data sets were obtained from the UCI repository [12]. Error was calculated using ten-fold cross-validation.

Just as in Domingo's experiments, these results show that Bayesian model averaging achieves a lower average accuracy on the data sets than either bagging or boosting. However, a strategy that iterates over combinations of models allows a Bayesian method to compete with the *ad hoc* methods. An application of the Friedman test reveals significant differences in average accuracy among the various strategies. ($27.77 \sim \chi^2, DF = 4, p <= 0.01$). The Bonferroni-Dunn *post hoc* test indicates that the improvement in accuracy of this Bayesian model combination strategy exceeds the critical difference for significance at a confidence level of 95% for two of the other four strategies (Critical difference = 0.87, Mean rank differences: $1.26, 0.81, 0.18, 1.25$).

TABLE II
AVERAGE ACCURACY OF VARIOUS ENSEMBLE COMBINATION STRATEGIES

	J48	Bagging	Boosting	BMA	BMC
anneal	98.44	98.22	99.55	98.22	98.89
audiology	77.88	76.55	84.96	76.11	82.30
autos	81.46	69.76	83.90	70.24	84.39
balance-scale	76.64	82.88	78.88	82.88	81.44
bupa	68.70	71.01	71.59	70.43	69.86
cancer-wisc.	93.85	95.14	95.71	95.28	95.42
cancer-yugo.	75.52	67.83	69.58	68.18	73.08
car	92.36	92.19	96.12	92.01	93.87
cmc	52.14	53.63	50.78	41.96	53.22
credit-a	86.09	85.07	84.20	84.93	85.65
credit-g	70.50	74.40	69.60	74.30	72.90
dermatology	93.99	92.08	95.63	92.08	95.36
diabetes	73.83	74.61	72.40	74.61	72.92
echo	97.30	97.30	95.95	97.30	97.30
ecoli-c	84.23	83.04	81.25	82.74	84.82
glass	66.82	69.63	74.30	68.69	70.56
haberman	71.90	73.20	72.55	73.20	74.51
heart-cleveland	77.56	82.18	82.18	82.18	80.86
heart-h	80.95	78.57	78.57	78.57	79.59
heart-statlog	76.67	79.26	80.37	78.52	80.00
hepatitis	83.87	84.52	85.81	83.87	84.52
horse-colic	85.33	85.33	83.42	85.05	85.87
hypothyroid	99.58	99.55	99.58	99.55	99.60
ionosphere	91.45	90.88	93.16	90.60	93.16
iris	96.00	94.00	93.33	94.00	95.33
kr-vs-kp	99.44	99.12	99.50	99.12	99.44
labor	73.68	85.96	89.47	87.72	84.21
led	100.00	100.00	100.00	100.00	100.00
lenses	83.33	66.67	70.83	58.33	79.17
letter	100.00	100.00	100.00	100.00	100.00
liver-disorders	68.70	71.01	71.59	70.43	69.86
lungcancer	50.00	50.00	53.12	46.88	53.12
lymph	77.03	78.38	81.08	79.05	79.73
monks	96.53	99.54	100.00	96.99	100.00
page-blocks	96.88	97.24	97.02	97.26	97.26
postop	70.00	71.11	56.67	71.11	68.89
primary-tumor	39.82	45.13	40.12	45.13	41.59
promoters	81.13	83.96	85.85	85.85	81.13
segment	96.93	96.97	98.48	96.88	97.66
sick	98.81	98.49	99.18	98.46	98.94
solar-flare	97.83	97.83	96.59	97.83	97.83
sonar	71.15	77.40	77.88	77.40	75.48
soybean	91.51	86.82	92.83	86.38	93.56
spect	78.28	81.65	80.15	82.02	79.03
tic-tac-toe	85.07	92.07	96.35	91.65	93.53
vehicle	72.46	72.70	76.24	72.81	76.36
vote	94.79	94.58	95.66	94.58	95.66
wine	93.82	94.94	96.63	93.26	95.51
yeast	56.00	60.04	56.40	31.20	60.24
zoo	92.08	87.13	96.04	86.14	93.07
average:	82.37	82.79	83.62	81.64	83.93

B. BMC with Sampling from a Dirichlet Distribution

Our previous implementation of BMC used a systematic method for sampling the space of potential model combinations. But as we shall see, further improvements in accuracy can be achieved using a slightly more sophisticated stochastic strategy for creating a set of potential model combinations. Instead of assigning weights incrementally, the weights for each combination of the base classifiers can be obtained by sampling from a Dirichlet distribution.

In this next set of experiments, weights for the first q combinations were drawn from a Dirichlet distribution with uniform alpha values. $p(e|D)$ was then calculated for each combination, and the weights from the most probable combination were used to update the alpha values for the distribution from which the next q weight assignments were drawn. Table III illustrates how weights were assigned in these experiments.

The same ten base classifiers from the previous section were used in these experiments. Alpha values were updated with a q value of three, and $59,049$ Dirchlet-generated weight assignments were considered. Results are shown in Table IV.

An application of the Friedman test reveals significant differences in average accuracy among the various strategies.

TABLE III

SAMPLE WEIGHT ASSIGNMENTS FOR INDIVIDUAL COMPONENTS IN A BAYESIAN MODEL COMBINATION LEARNER EMPLOYING A DIRICHLET DISTRIBUTION. AFTER A SET OF COMBINATIONS ARE GENERATED, THE WEIGHTS OF THE MOST PROBABLE COMBINATION ARE USED TO UPDATE THE ALPHA VALUES OF THE DIRICHLET FROM WHICH THE NEXT SET OF COMBINATIONS WILL BE DRAWN. AS WITH THE FIRST EXPERIMENTS, EACH COMPONENT IS WEIGHTED WITH A UNIFORM PRIOR.

Weights	$p(e\|D)$	$p(e)$
Initial alpha values: 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00		
0.06 0.26 0.08 0.11 0.09 0.20 0.17 0.00 0.02 0.01	0.00	$\frac{1}{59049}$
0.10 0.15 0.14 0.28 0.04 0.00 0.17 0.03 0.07 0.02	0.03	$\frac{1}{59049}$
0.00 0.10 0.04 0.04 0.03 0.03 0.09 0.02 0.29 0.36	0.02	$\frac{1}{59049}$
New alpha values: 1.10 1.15 1.14 1.28 1.04 1.00 1.17 1.03 1.07 1.02		
0.07 0.00 0.04 0.12 0.26 0.15 0.07 0.13 0.01 0.13	0.03	$\frac{1}{59049}$
0.16 0.13 0.15 0.05 0.00 0.04 0.07 0.14 0.13 0.12	0.02	$\frac{1}{59049}$
0.01 0.05 0.07 0.15 0.04 0.08 0.26 0.01 0.26 0.08	0.02	$\frac{1}{59049}$
New alpha values: 1.17 1.15 1.19 1.40 1.31 1.16 1.24 1.17 1.07 1.15		
0.02 0.02 0.03 0.28 0.20 0.04 0.04 0.00 0.18 0.19	0.02	$\frac{1}{59049}$
0.35 0.12 0.13 0.06 0.08 0.07 0.09 0.02 0.06 0.01	0.00	$\frac{1}{59049}$
0.07 0.14 0.02 0.01 0.17 0.01 0.17 0.15 0.14 0.12	0.03	$\frac{1}{59049}$

($28.76 \sim \chi^2, DF = 4, p <= 0.01$). The Bonferroni-Dunn *post hoc* test indicates that the improvement in accuracy of Bayesian model combination with Dirichlet sampling exceeds the critical difference for significance at a confidence level of 95% for three of the other four strategies (Critical difference = 0.87, Mean rank differences: $1.33, 0.87, 0.29, 1.31$).

IV. BAYESIAN MODEL PARAMETER LEARNING GIVEN A FIXED COMBINATION OF MODELS

The previous experiments effectively use Bayesian techniques to determine the optimal combination of a fixed set of learners. Alternately, Bayesian techniques can be used to update learners given a fixed combination of weights. There are likely many models for which this sort of strategy could be applied, but one simple illustrative case involves the CMAC neural network topology [13].

The CMAC is modeled on the human cerebellum. It functions by mapping weights $w[i]$ to tiles which are interpreted spatially, as illustrated in Figure 3. Inputs are mapped to the correct bins by means of an association function $b[i](x)$, where $b[i](x) = 0$ when x does not fall within the spacial region assigned to bin i and where $b[i](x) = 1$ when it does. The output of the system can be computed as follows:

$$f_{CMAC}(x) = \sum_i w[i]b[i](x) \tag{5}$$

Note that the CMAC outputs continuous values, so the experiments in this section will involve data sets with real rather than discrete target values. The error at location x is calculated as shown:

$$e(x) = f_{CMAC}(x) - f_{observed}(x) \tag{6}$$

Traditionally, weights are updated as follows:

$$\Delta w[i] = \alpha \frac{e(x)}{\sum_i b[i](x)} \tag{7}$$

TABLE IV

AVERAGE ACCURACY OF VARIOUS ENSEMBLE COMBINATION STRATEGIES

	J48	Bagging	Boosting	BMA	BMC-D
anneal	98.44	98.22	99.55	98.22	98.89
audiology	77.88	76.55	84.96	76.11	82.30
autos	81.46	69.76	83.90	70.24	84.88
balance-scale	76.64	82.88	78.88	82.88	81.92
bupa	68.70	71.01	71.59	70.43	71.88
cancer-wisc.	93.85	95.14	95.71	95.28	95.14
cancer-yugo.	75.52	67.83	69.58	68.18	73.08
car	92.36	92.19	96.12	92.01	93.75
cmc	52.14	53.63	50.78	41.96	52.95
credit-a	86.09	85.07	84.20	84.93	85.07
credit-g	70.50	74.40	69.60	74.30	73.10
dermatology	93.99	92.08	95.63	92.08	95.36
diabetes	73.83	74.61	72.40	74.61	74.35
echo	97.30	97.30	95.95	97.30	97.30
ecoli-c	84.23	83.04	81.25	82.74	84.52
glass	66.82	69.63	74.30	68.69	70.09
haberman	71.90	73.20	72.55	73.20	74.51
heart-cleveland	77.56	82.18	82.18	82.18	79.87
heart-h	80.95	78.57	78.57	78.57	79.59
heart-statlog	76.67	79.26	80.37	78.52	80.00
hepatitis	83.87	84.52	85.81	83.87	83.87
horse-colic	85.33	85.33	83.42	85.05	86.14
hypothyroid	99.58	99.55	99.58	99.55	99.60
ionosphere	91.45	90.88	93.16	90.60	93.45
iris	96.00	94.00	93.33	94.00	95.33
kr-vs-kp	99.44	99.12	99.50	99.12	99.44
labor	73.68	85.96	89.47	87.72	84.21
led	100.00	100.00	100.00	100.00	100.00
lenses	83.33	66.67	70.83	58.33	79.17
letter	100.00	100.00	100.00	100.00	100.00
liver-disorders	68.70	71.01	71.59	70.43	71.88
lungcancer	50.00	50.00	53.12	46.88	56.25
lymph	77.03	78.38	81.08	79.05	80.41
monks	96.53	99.54	100.00	96.99	100.00
page-blocks	96.88	97.24	97.02	97.26	97.24
postop	70.00	71.11	56.67	71.11	67.78
primary-tumor	39.82	45.13	40.12	45.13	41.30
promoters	81.13	83.96	85.85	85.85	81.13
segment	96.93	96.97	98.48	96.88	97.45
sick	98.81	98.49	99.18	98.46	98.97
solar-flare	97.83	97.83	96.59	97.83	97.83
sonar	71.15	77.40	77.88	77.40	74.52
soybean	91.51	86.82	92.83	86.38	93.12
spect	78.28	81.65	80.15	82.02	79.03
tic-tac-toe	85.07	92.07	96.35	91.65	93.53
vehicle	72.46	72.70	76.24	72.81	76.48
vote	94.79	94.58	95.66	94.58	95.44
wine	93.82	94.94	96.63	93.26	95.51
yeast	56.00	60.04	56.40	31.20	60.51
zoo	92.08	87.13	96.04	86.14	93.07
average:	82.37	82.79	83.62	81.64	84.02

where α is the learning rate. The output y of the network at any position x is the sum of the weights for the tiles that overlap that position.

Though not a traditional view, the CMAC can be thought of as an ensemble where each layer learns information about a given function and outputs are calculated by combining information from each layer using a fixed weighting scheme (each layer is equally weighted with all the others). The ensemble-like structure suggests that the CMAC could also be reasonably trained using ensemble techniques such as

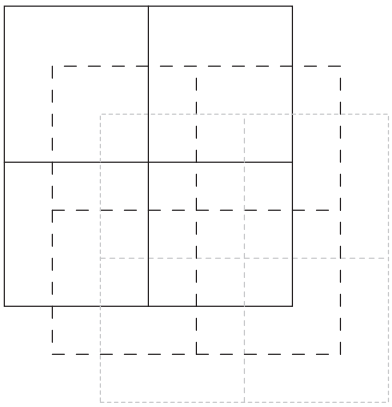

Fig. 3. Tile structure for a CMAC with three layers and four tiles per layer

bagging or Bayesian model averaging, treating the layers as individual learners and altering the weightings of layer outputs according to the given technique. With one task specifically designed to match the assumptions made by BMA, that ensemble creation technique is effective in reducing error. However, once again, a Bayesian strategy that allows for a model combination approach does better on a wider variety of tasks.

Carroll, Monson, and Seppi [14] showed how Bayesian techniques can be applied to CMAC learning. Further details on BCMAC training can be found elsewhere in the literature [15], but a brief overview is provided here. A function, f, is assumed to be stationary, and all observations y are assumed to have linear Gaussian noise with covariance Σ_y. The relationship between the data D and the CMAC's representation for f can be modeled as follows:

$$p(\mathbf{y}|\mathbf{x}, f) = N(\mathbf{y}; f(\mathbf{x}), \mathbf{\Sigma}_y). \tag{8}$$

This can be rewritten as:

$$p(\mathbf{y}|\mathbf{x}, f) = N(\mathbf{y}|\mathbf{H}\mathbf{w}, \mathbf{\Sigma}_y), \tag{9}$$

where $\mathbf{H}$ can be thought of as an association matrix. $\mathbf{H}_{i,j} = 1$ if tile j influences the training example i. Weight values are represented by the vector $\mathbf{w}$. Weights of the model are related to observations according to a multivariate normal model [16] with prior parameters $\boldsymbol{\mu_0}$ and $\mathbf{\Sigma}_0$. The parameters of the posterior distributions for the mean and covariance can then be found by:

$$\boldsymbol{\mu}_1 = \boldsymbol{\mu}_0 + \mathbf{K}_1(\mathbf{y} - \mathbf{H}\boldsymbol{\mu}_0), \tag{10}$$

and

$$\mathbf{\Sigma}_1 = (\mathbf{I} - \mathbf{K}_1\mathbf{H})(\mathbf{\Sigma}_0), \tag{11}$$

where

$$\mathbf{K}_1 = (\mathbf{\Sigma}_0)\mathbf{H}^T(\mathbf{H}(\mathbf{\Sigma}_0)\mathbf{H}^T + \mathbf{\Sigma}_y)^{-1}. \tag{12}$$

These equations are identical to the Kalman filter for a single time step. This observation means that, given a prior over CMAC weights and some training data, a well-known and widely studied filtering technique can be applied to solve in closed form for both the posterior distribution over the CMAC weights and the posterior predictive distribution over CMAC outputs.

The benefits of this strategy are demonstrated in the following experiments. The layers of the CMAC were learned using the traditional CMAC learning rule, bagging, Bayesian model averaging, and the BCMAC learning rule. All of the CMACs were constructed with five layers and between three and seven tiles tiles per dimension on each layer. With the bagging CMAC, layers were trained individual on size n subsets selected with replacement from the initial training set of size n. Outputs of each layer were then weighted equally when calculating the final output for a given example. The Bayesian model averaging CMAC was constructed in a similar manner, but layer outputs were weighted by a likelihood term calculated using a normal noise model. Priors for the BCMAC were calculated empirically based on the data sets.

Experiments are conducted on three numeric data sets provided by Weka for machine learning tasks [11]. Because the CMAC was designed for continuous values, these sets were selected for their limited number of numerical features and numeric class values. Algorithm performance was also tested on twoDimEgg, a variant of the two-dimensional egg carton function $y = sin(x_1 * 2.5) + sin(x_2 * 2.5)$, and step2d, a stepwise function which returns 1 if $x_1^2 + x_2^2 < 10$ and -1 otherwise. This rather simple function was specifically chosen to have a steep, curved boundary, a situation which have been shown to be difficult for CMAC based learning algorithms.

In order to further test the theory that BMA performs poorly because it performs optimal model selection instead of optimal model combination, the final data set, optimalBMA, was constructed to provide a situation where model selection would perform well [9]. The function assigns -1 to all values left of a vertical boundary and 1 to all values to the right. This boundary was aligned with the edge of one of the tiles in the CMAC. Thus, one of the layers would exactly replicate the DGM in the sense of providing correct outputs for each example while every other layer would provide at least some incorrect outputs. The goal of an ensemble strategy would be to select this layer.

TABLE V
AVERAGE ERROR RATES OF FOUR LEARNING STRATEGIES

	CMAC	Bagging	BMA	BCMAC
elusage	0.047	0.045	0.045	0.035
gascon	0.140	0.135	0.134	0.041
longley	0.097	0.119	0.119	0.062
step2d	0.019	0.018	0.022	0.018
twoDimEgg	0.025	0.109	0.270	0.018
optimalBMA	0.005	0.071	0.006	0.002

The BCMAC achieves a substantially lower error rate than the Bayesian model averaging strategy on all data sets studied, except for the case of optimalBMA where the results are nearly indistinguishable. In fact, with the exception of one tie with bagging on the step2d function, BCMAC

outperforms all of the other three algorithms in terms of error reduction over the other five data sets. As with the previous experiments, bagging was often able to achieve a lower error rate than Bayesian model averaging. However, Bayesian model averaging substantially outperforms bagging on the optimalBMA data set, where placing all of the weight on one component is the best strategy. BMA was outperformed by the *ad hoc* techniques, except in the one case where model selection was required. This again provides further empirical justification for Minka's proposition on the theory of ensemble learning.

V. Conclusion

Despite the theoretical optimality of Bayesian methods and their successful application to a wide variety of tasks, the standard technique of Bayesian model averaging struggles in empirical studies. Minka theorized that since the algorithm places so much emphasis on the most likely ensemble member, it fails to take advantage of the benefits inherent in model combinations. However, as we have shown, if BMA is modified to integrate over combinations of models rather than over individual learners, it can achieve much better results.

Domingos described a number of situations in which Bayesian model averaging is outperformed by standard *ad hoc* ensemble creation methods. We have shown that even the most simplistic of Bayesian model combination strategies outperforms the traditional *ad hoc* techniques of bagging and boosting, as well as outperforming BMA in a significant number of cases. We have demonstrated with the BCMAC experiments that, in the rare instances where model selection is indeed the correct approach, Bayesian model averaging performs well. On most problems, however, a Bayesian technique geared toward selecting a combination of models results in lower error rates.

This work has some theoretical implications for why ensembles work. The results suggest the effectiveness of ensembles is due, at least in part, to the enriched hypothesis space and more general bias that can be provided by a combination of models. We have demonstrated that there are a wide variety of potential methods for applying Bayesian techniques to model combination. We have shown that it is possible to fix the component learners and then learn the optimal model combination in a Bayesian fashion (both versions of BMC). We have also shown that in some situations it is possible to fix the model combination strategy, and learn optimal models given the known combination (BCMAC).

Future work will involve the investigation of more sophisticated methods of Bayesian model combination. For example, the simple Bayesian model combination strategies presented in Section III could be modified to allow for non-linear combinations of models. Other possible strategies might take spatial considerations into account, developing learners to specialize in different areas of the feature space or training learners with the sampling techniques used in boosting.

In this paper, we have shown how Bayesian inference can be used to generate the optimal combination (ensemble member weights) given a set of fixed (and already trained) learners. We have also shown how Bayesian inference can be used to infer the optimal set of component model parameters given a fixed ensemble combination scheme. Future work will involve using these two approaches could be used simultaneously. One way to accomplish this could involve an expectation maximization strategy. An optimal combination could be determined given a set of learners, and then the learners could be updated given the new combination strategy. Alternatively, strategies could be developed that would allow learners and combinations to be inferred simultaneously. The BCMAC can be solved in closed form because both weights and outputs are distributed normally. Other learners with similar Normal distribution properties might also be solved in a similar fashion. Gaussian processes should be explored as a potential rich framework for building such learners.

References

[1] L. Breiman, "Bagging predictors," *Machine Learning*, vol. 24, no. 2, pp. 123–140, 1996.

[2] Y. Freund and R. E. Schapire, "Experiments with a new boosting algorithm," *Proceedings of the Thirteenth International Conference on Machine Learning*, 1996.

[3] J. M. Bernardo and A. F. M. Smith, *Bayesian theory*. New York, NY: Wiley, 1994.

[4] K. Lang, "Newsweeder: Learning to filter netnews," *Proceedings of the Twelfth International Conference on Machine Learning*, pp. 331–339, 1995.

[5] G. H. John and P. Langley, "Estimating continuous distributions in bayesian classifiers," *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence*, p. 338345, 1995.

[6] W. R. Gilks, "Markov chain monte carlo," *Encyclopedia of Biostatistics*, 2005.

[7] D. Blei, A. Ng, and M. Jordan, "Latent dirichlet allocation," *Advances in Neural Information Processing Systems*, vol. 14, 2002.

[8] P. Domingos, "Bayesian averaging of classifiers and the overfitting problem," *Proceedings of the Seventeenth International Conference on Machine Learning*, 2000.

[9] B. Clarke, "Comparing bayesian model averaging and stacking when model approximation error cannot be ignored," *Journal of Machine Learning Research*, vol. 4, 2003.

[10] T. Minka, "Bayesian model averaging is not model combination," *MIT Media Lab Note December 2000*, 2000.

[11] I. H. Witten and E. Frank, *Data Mining: Practical Machine Learning Tools and Techniques*, 2nd ed. San Francisco: Morgan Kaufmann, 2005. [Online]. Available: http://sourceforge.net/projects/weka/files/datasets/

[12] S. Hettich, C. L. Blake, and C. J. Merz, "Uci repository of machine learning databases," 1998. [Online]. Available: http://www.ics.uci.edu/ mlearn/MLRepository.html

[13] J. S. Albus, "A new approach to manipulator control: The cerebellar model articulation controller (CMAC)," *Journal of Dynamic Systems, Measurement, and Control*, vol. 97, no. 3, pp. 220–227, 1975.

[14] J. L. Carroll, C. K. Monson, and K. D. Seppi, "A bayesian CMAC for high assurance supervised learning," *Applications of Neural Networks in High-Assurance Systems, IJCNN Workshop*, 2007.

[15] J. L. Carroll, "A bayesian decision theoretical approach to supervised learning, selective sampling, and empirical function optimization," Ph.D. dissertation, Brigham Young University, March 2010. [Online]. Available: http://james.jlcarroll.net/publications/

[16] M. H. DeGroot, *Optimal Statistical Decisions*. New York, NY: McGraw-Hill Book Company, 1970.