Bayesian Analysis (2013) **8**, Number 4, pp. 883–908

Clustered Bayesian Model Averaging

Qingzhao Yu *, Steven N. MacEachern † and Mario Peruggia ‡

Abstract. It is sometimes preferable to conduct statistical analyses based on the combination of several models rather than on the selection of a single model, thus taking into account the uncertainty about the true model. Models are usually combined using constant weights that do not distinguish between different regions of the covariate space. However, a procedure that performs well in a given situation may not do so in another situation. In this paper, we propose the concept of local Bayes factors, where we calculate the Bayes factors by restricting the models to regions of the covariate space. The covariate space is split in such a way that the relative model efficiencies of the various Bayesian models are about the same in the same region while differing in different regions. An algorithm for clustered Bayes averaging is then proposed for model combination, where local Bayes factors are used to guide the weighting of the Bayesian models. Simulations and real data studies show that clustered Bayesian averaging results in better predictive performance compared to a single Bayesian model or Bayesian model averaging where models are combined using the same weights over the entire covariate space.

Keywords: Bayesian Model Averaging, Clustered Bayes Factor, Local Averaging

1 Introduction

The purpose of statistical analysis is usually two-fold: description and prediction (Breiman, 2001). For a given quantity of interest, we use statistical methods to look for "important" factors and to explore the relationship between these factors and the quantity of interest. We then use the collected information for future prediction. A traditional method in statistical data analysis is to choose or build one "best" model or procedure to use for estimation and/or prediction. This process is called model selection.

There is a large body of literature on model selection. In the field of linear regression, model selection research has focused mainly on variable selection methods, which range from the traditional approaches based on R-squared, c_p, and other adjusted criteria, to the information criteria such as the Akaike information criterion (AIC), the Bayesian information criterion (BIC), etc., to regularization methods, such as the least absolute shrinkage and selection operator (LASSO, Tibshirani, 1996), which penalize the loss function for model complexity. Despite the theoretical and methodological developments in the field, concerns about model selection are evident and growing. The major concern is that selecting a single model does not take into account model uncertainty, thus underestimating the variability associated with the estimation or prediction. Another

*Louisiana State University Health Science Center qyu@lsuhsc.edu
†Department of Statistics, The Ohio State University snm@stat.osu.edu
‡Department of Statistics, The Ohio State University peruggia@stat.osu.edu

 DOI:10.1214/13-BA859

concern is that selection methods are typically unstable—a small change in the data may result in a big change in the selected model, in terms of both included variables and estimated parameter values.

Methods have been developed to address the problems stemming from model selection, mainly through the adoption of multiple models. Representative classical methods include the bootstrap method (Rao and Tibshirani, 1996), "bagging" (Breiman, 1996), frequentist model averaging using the focused information criterion (FIC, Hjort and Claeskens, 2003), and adaptive regression by mixing (ARM, Yang, 2001).

The bootstrap method and bagging focus mainly on building different models based on different sets of training data. The model building process for each set of training data is usually the same and the models are combined with equal weights. The ARM algorithm allows the candidate models to be very different and assigns weights to candidate models by measuring relative model performance. In this algorithm, a training data set is randomly split into two parts, one for model building and the other for model performance assessment.

Bayesian model averaging (BMA, Raftery et al., 1997) is a Bayesian version of those methods that averages over selected linear models. The linear models are combined with weights proportional to their empirical marginal likelihood over the training data. Let D denote the observed data set, and let $p(\theta_i|M_i)$ be the prior distribution for the parameters θ_i of model M_i. The key component informing Bayesian model choice is the marginal likelihood of the models, given by

$$p(M_i|D) = \frac{p(D|M_i)p(M_i)}{\sum_j p(D|M_j)p(M_j)}, \tag{1}$$

where $p(D|M_i) = \int p(D|M_i, \theta_i)p(\theta_i|M_i)d\theta_i$ is the marginal likelihood of model M_i, calculated by integrating the joint density of the data D and the parameters θ_i over the parameter space of θ_i. In BMA, the various linear models under consideration are weighted by their respective marginal likelihoods and are updated with newly collected data. Asymptotically, the weights are governed by the Kullback-Leibler divergence from the fitted model to the true model. BMA incorporates model uncertainty into the posterior inferences. Under BMA, the posterior distribution of a specific quantity of interest, Δ, is

$$p(\Delta|D) = \sum_{i\in\Gamma} p(\Delta|M_i, D)p(M_i|D), \tag{2}$$

where Γ is the set of models under consideration. BMA is exactly the application of Bayes theorem to a "hyper-model" consisting of a distribution across models and a distribution on the parameters within each model. Compared with the selection of a single model, BMA has shown better predictive performance in practice and gives more stable results in general.

With the backing of Bayes theorem, BMA appears to be the ultimate means of making inference. One specifies the prior distribution both across and within models and then performs the standard Bayesian update. However, there are reasons to turn to alternative methods. Traditional departures have been driven by several issues.

First, it can be difficult to specify a full Bayesian model, including the prior distribution and the likelihood. As a consequence, many methods have been developed to minimize the impact of the prior specification, notably objective Bayesian methods. Some of these methods rely on improper prior distributions and use a training sample to pass from the improper prior distribution to a proper partial posterior distribution before commencing with model averaging (Berger and Pericchi, 1996) or they construct prior distributions that are sufficiently proper for the computation of a Bayes factor (Casella and Moreno, 2006). Other methods attempt to remove the 'bad parts' of the prior distribution by stepping beyond Bayes theorem, for example by restricting Γ in Equation (2) to include only those models receiving relatively high marginal likelihood as in "Occam's window" (Madigan and Raftery, 1994). The robust Bayes literature (e.g., Berger 1994) formally examines the sensitivity of posterior inference to changes in the prior distribution. We do not directly address prior specification in this paper.

A second reason for departing from BMA is computational. When the number of models to be investigated is large or fitting the models is difficult, the computational cost of BMA can be prohibitive. Thus, many resort to Markov chain Monte Carlo methods and rely on relatively simple prior distributions, often of conjugate or conditionally conjugate form (e.g., George and McCulloch 1993, Madigan and York 1995, and Clyde et al. 2011). When run forever, these strategies typically produce BMA, but for large problems and relatively modest computational effort, they may be best viewed as model search strategies (Hans et al. 2007). The impact of computation has been a longstanding concern, and it is one motivation for techniques such as the pseudo-Bayes factor (Geisser and Eddy 1979), an early approach to Bayesian model selection. We propose a set of computationally convenient variations of our method in the following sections, with particular attention given to techniques relying on the pseudo-Bayes factor.

The main focus of our work is a non-traditional departure from Bayesian methods that focuses on a different shortcoming in their implementation: namely that the "true model" will most often lie outside the support of the hyper-model that is to be used for formal inference. When this happens, our Bayesian model cannot capture the "true model", even asymptotically. Standard calculations show that, under mild regularity conditions, the posterior will assign probability tending to one to a single "closest" model. However, it will typically be the case that different models will provide a better approximation to the true model in different regions of the covariate space. BMA and other global weighting methods do not take this different regional performance into account. Instead, each model is given the same weight over the entire covariate space, and this weight is driven as much by the distribution of covariates in the study as by the responses. There is potential for improving the statistical analysis if we take the regional differences into account.

The importance of accounting for differing regional model performance has been addressed in the literature on mixture-of-experts models (Jordan and Jacobs 1994, Xu et al. 1995). These models place a mixing distribution over a collection of models. The mixing distribution allows for local weights, and each observation is presumed to be drawn from a "random" expert (or model). From a fully Bayesian point of view, mixture of expert models have difficulty handling improper or vague expert-specific prior

distributions.

In this article, we react to the non-regionality of standard Bayesian methods in a fashion which still allows us to make strong use of Bayes theorem and to perform computations in a feasible time-frame. We propose a method, clustered Bayesian averaging (CBA), that combines different Bayesian analysis procedures using adaptive weights. The Bayesian procedures could be parametric models and/or nonparametric procedures. For CBA, the covariate space is split in such a way that the relative performances of the various procedures are about the same in a given region but differ from one region to another. Therefore, the procedures are assigned different weights in different covariate regions. CBA is intended to be a theoretically-adjustable and practical algorithm for combining Bayesian procedures that aims to improve prediction.

The article is organized as follows. In Section 2, we introduce the concept of the local Bayes factor and in Section 3 we use this concept to develop the clustered Bayesian averaging methodology, describing a basic algorithm and several variants on it. Illustrations with simulation and real data are provided in Section 4. Conclusions and a discussion are given in Section 5.

2 Local Bayes Factor

The Bayes factor (BF) was mentioned by Wald (1947) in his work on sequential analysis and developed by Jeffreys (1961) as a Bayesian approach to hypothesis testing. Many since, including Kass and Raftery (1995), have used the BF to compare the performance of Bayesian models. Assume that two models M_1 and M_2 are built to fit the data D. Given prior probabilities $p(M_1)$ and $p(M_2) = 1 - p(M_1)$ for the models, the data can be used according to Equation (1) to calculate the posterior probabilities $p(M_1|D)$ and $p(M_2|D) = 1 - p(M_1|D)$. Passing from probabilities to odds, the posterior odds for the two models are given by

$$\frac{p(M_1|D)}{p(M_2|D)} = \frac{p(D|M_1)}{p(D|M_2)}\frac{p(M_1)}{p(M_2)} = B_{12}\frac{p(M_1)}{p(M_2)},$$

where the Bayes factor is defined as

$$B_{12} = \frac{p(D|M_1)}{p(D|M_2)}.$$

The posterior odds are used to guide Bayesian model averaging. If the prior belief is that the two models are equally likely, the posterior odds equal the Bayes factor. To combine the two models, it is natural to weight M_1 by $B_{12}/(1 + B_{12})$ and M_2 by $1/(1 + B_{12})$.

When calculating the Bayes factor, one should be concerned that improper prior distributions can sometimes result in an undefined Bayes factor. To solve this problem without multiple use of the data, the fractional Bayes factor (O'Hagan 1995) and the intrinsic Bayes factor (Berger and Pericchi 1996) were developed. Both methods are most easily viewed as relying on a partial update, moving from the prior distribution

to a partial posterior distribution, followed by the remaining update which is used to compute the Bayes factor.

For the fractional Bayes factor, a fractional power of the likelihood is taken to update the improper priors, resulting in proper partial posteriors. The remaining fraction of the likelihood is used for computation of the Bayes factor. Specifically, for an update based on a fraction f, the partial posterior distribution for M_i is given by

$$p(\theta_i|M_i, D^f) = \frac{p(D|M_i, \theta_i)^f p(\theta_i|M_i)}{\int p(D|M_i, \theta_i)^f p(\theta_i|M_i) d\theta_i},$$

where $f \in (0, 1)$ is assumed to be large enough that the denominator is finite and where the apparent conditioning on D^f is merely notation to represent the partial update. The fractional Bayes factor for M_1 over M_2 then becomes

$$FBF_{12} = \frac{\int p(D|M_1, \theta_1)^{(1-f)} p(\theta_1|M_1, D^f) d\theta_1}{\int p(D|M_2, \theta_2)^{(1-f)} p(\theta_2|M_2, D^f) d\theta_2}.$$

For the intrinsic Bayes factor, the data set is split into a training data set which is used to produce a proper partial posterior and the remaining data which are used to calculate the Bayes factor. The full methodology for the intrinsic Bayes factor makes use of many repeated divisions of the data into training data and remaining data. The common implementation uses minimal training data sets, in the sense that they contain just enough data to produce proper partial posterior distributions for the models under consideration. The intrinsic Bayes factor and various modifications to it are now routinely used for Bayesian model selection and model averaging, although not without controversy (Kadane and Lazar, 2004). In the sequel, we draw on ideas stemming from both the fractional Bayes factor and the intrinsic Bayes factor. Details of these partial updates, tailored to the setting of the local Bayes factor, appear shortly.

In keeping with our goal of allowing the final inference to react to differential local performance, we create a means of obtaining regional, rather than global, weights for the two models. For this purpose, we define a local Bayes factor.

Definition 1. *Using the notation introduced previously, suppose that the covariate space* Λ *can be partitioned into a finite number* K *of disjoint subregions such that* $\Lambda = \Lambda_1 \cup \Lambda_2 \cup \ldots \cup \Lambda_K$. *The* local Bayes factor *of model* M_1 *over model* M_2 *in subregion* Λ_k *is defined to be the Bayes factor calculated by restricting the covariate space to* Λ_k*:* $B_{12k} = p(D_k|M_1)/p(D_k|M_2)$*, where* D_k *denotes the subset of the data* D *falling in* Λ_k*. The local log Bayes factor is defined as* $\log(B_{12k})$.

Note that $p(D_k|M_i)$ would typically be $\int p(D_k|M_i, \grave{} _i) p(\grave{} _i|M_i) d\grave{} _i$, where the prior distribution of $\grave{} _i$, $p(\grave{} _i|M_i)$, is specified without regard to the region of the covariate space under consideration. The updating would be purely local, leading to two problems. First, for the linear model, restriction to a small region of covariate space limits the leverage of cases, producing instability in the likelihood surface. For vague prior distributions, this translates to unstable posterior means for the regression coefficients and so to unstable (highly variable) predictions. A similar phenomenon occurs with non-linear

models. Second, with the covariate space partitioned, there is less information within each region for computation of the Bayes factor, leading to less ability to discriminate between or appropriately weight rival models. A small effective sample size coupled with vague prior distributions can generate unstable Bayes factors (Xu et al. 2011).

One way to address both of these sources of instability is through a substantial partial global update. By making use of the entire covariate space, leverage is unconstrained and the partial posterior is far more stable. Stability of the partial posterior also leads to much more stable joint marginal likelihoods of the data, lending stability to the local Bayes factors. As an added bonus, this stability considerably improves a computationally convenient independence approximation. The global partial update also allows us to use improper prior distributions for one or both of the models.

Formally, for the partial update, a portion of the data, D^*, is used to update the models M_1 and M_2. These data are not included in the local Bayes factor calculation. The updated distributions are then used as the starting point for the local Bayes factor calculation. In this case, with D^* representing the data used for the partial update and D_k representing the data used to compute the local Bayes factor in region Λ_k, the integral defining $p(D_k|M_i)$ is to be interpreted as $\int p(D_k|M_i, D^*, \grave{} _i)p(\grave{} _i|M_i, D^*)d\grave{} _i$. That is, $p(M_i)$ is replaced by $p(M_i|D^*)$ and $p(D_k|M_i)$ is replaced by $p(D_k|M_i, D^*)$, resulting in the local Bayes factor

$$B_{12k} = \frac{p(D_k|M_1, D^*)}{p(D_k|M_2, D^*)}.$$

We have the following property for local Bayes factors under the strong assumption that observations from different regions are independent under both model M_1 and model M_2.

Lemma 2. *If the data from different regions are independent (with respect to their distributions under models M_1 and M_2), the log Bayes factor of M_1 over M_2 is the summation of the local log Bayes factors. In other words, using the notation of Definition 1,* $\log(B_{12}) = \sum_{k=1}^{K} \log B_{12k}$.

The proof of Lemma 2 is straightforward as

$$\log B_{12} = \log \frac{p(D|M_1)}{p(D|M_2)} = \log \frac{\prod_{k=1}^{K} p(D_k|M_1)}{\prod_{k=1}^{K} p(D_k|M_2)} = \sum_{k=1}^{K} \log \frac{p(D_k|M_1)}{p(D_k|M_2)},$$

with the latter two equalities justified by the independence assumption.

Of course, the data from different regions will not be independent under typical Bayesian models, although it is common for all of the observations to be conditionally independent (given parameters in the model). Under many measures, the dependence among observations is strongest when the prior distribution is vague. Under a more concentrated prior, the dependence is weaker and, in the limiting case of a prior that concentrates at a given parameter vector, conditional independence *is* independence. The partial update serves to concentrate the partial posterior distribution near a parameter vector in each

model, rendering the calculations done under independence a good approximation to the full joint calculation. These partial updates have a long history in Bayesian statistics. The size of the data set D^* ranges from the bare minimum needed to obtain a proper partial posterior (Berger and Pericchi 1996) to half of the data (Lempers 1971), to all of the data except a single case (Geisser and Eddy 1979). As Geisser and Eddy note, the independence approximation simplifies computation. In our search to split the covariate space into regions of differential performance, we investigate two versions of the partial update, one based on a half-data update and the other based on an all-but-one case update. When making use of this independence approximation for splitting the covariate space and for computation of the local Bayes factor, we use the terminology "local pseudo Bayes factor" paralleling the definition of pseudo-likelihood.

In the development of our technique, we will partition the covariate space. We approximate calculations by assuming that the observations in the data set D are independent and write $D = \{\mathbf{d}_1, \ldots, \mathbf{d}_n\}$, where $\mathbf{d}_j$ is the (scalar or) vector of data collected for the j-th observation. Therefore, the marginal pseudo-likelihood is $p(D|M_i) = \prod_{j=1}^{n} p(\mathbf{d}_j|M_i)$ and the log marginal pseudo-likelihood is $l(D|M_i) = \sum_{j=1}^{n} \log[p(\mathbf{d}_j|M_i)]$. We denote by l_{ij} the log marginal pseudo-likelihood for model i at the j-th observation, $\log[p(\mathbf{d}_j|M_i)]$. When used with a partial update, these quantities become $p(D|M_i, D^*) = \prod_{j=1}^{n} p(\mathbf{d}_j|M_i, D^*)$ and $l(D|M_i, D^*) = \sum_{j=1}^{n} \log[p(\mathbf{d}_j|M_i, D^*)]$.

The relative performance of models M_1 and M_2 over the covariate space is determined by the difference in expected log marginal likelihood of the response. The surface changes as data accrue and the prior distributions are updated. Under mild regularity conditions, the surface stabilizes, tending to a well-defined limit over compact sets as the sample size grows. In subsequent sections, we focus on this surface after a substantial partial update of the models. The data not used for the partial update give us insight into the behavior of the surface, providing a noisy version of the surface, observed at (typically) irregularly spaced locations. Examination of the surface often provides evidence of differential model performance in different regions of the covariate space.

To compare and/or combine models, we must estimate the surface. This can be done in many ways, ranging from fitting low-dimensional parametric models to fitting flexible nonparametric models. In this work we pursue a path based on clustering techniques. We would like to divide the covariate space into regions such that $l_{1j} - l_{2j}$ is similar within regions and differs substantially between regions. In this, we recognize the clustering problem, and we find clusters such that the within cluster variances of $l_{1j} - l_{2j}$ are small compared to the between cluster variances. This goal can be easily fulfilled using any standard clustering method. Here, we make use of the regression tree method to perform the clustering. We set $l_{1j} - l_{2j}$ as the response variable and use all the covariates as explanatory variables. When a tree is fit, each leaf of the tree yields a cluster of observations. In a typical regression application, a tree method averages the observed responses which fall in each leaf to obtain an estimate of the mean response for the leaf. For the purpose of model comparison, we need the summation of the contributions from the observed responses falling in the leaf, which is the local log pseudo-Bayes factor.

3 Clustered Bayesian Averaging

In this section, we propose to use the concept of a local Bayes factor to improve statistical predictions. Consider the regression setting

$$y_i = g(\mathbf{x}_i) + \epsilon_i, \;\; i = 1, \ldots, n,$$

where $\{\mathbf{x}_i\}_{i=1}^n$ are i.i.d. draws from the joint distribution of $\mathbf{x}$, g is a deterministic function, and $\{\epsilon_i\}_{i=1}^n$ are i.i.d. noise terms, with mean 0, variance σ^2, independent of $\{\mathbf{x}_i\}_{i=1}^n$. The explanatory variables $\mathbf{x}$ can be multidimensional with unknown distribution. After observing a training data set $\mathbf{Z} = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$, the goal is to predict $\{y_j\}_{i=1}^T$ on a test data set $\mathbf{Z}_T = \{\mathbf{x}_j\}_{j=1}^T$.

Suppose that a finite collection of J Bayesian regression procedures has been proposed to estimate g. Procedure j may involve preliminary work, such as traditional model building and refinement, but ends up producing a model for the response, $f_j(y|\mathbf{x}, \theta_j)$, based on a covariate vector $\mathbf{x}$ and a parameter vector θ_j. The parameter vector θ_j follows a prior distribution $p_j(\theta_j)$ and has posterior distribution $p_j(\theta_j|\mathbf{Z})$.

No assumptions are made about the Bayesian regression procedures—both linear and nonlinear models can be used. Examples of procedures that will be used in this article include Bayesian additive regression trees (BART; Chipman et al. 2008); Bayesian model averaging (BMA, Raftery et al., 1997); treed Gaussian processes (Gramacy and Lee, 2008); Bayesian linear models where variables in the model are selected using AIC or BIC and flat priors are adopted for the parameters (denoted by BAIC and BBIC in this paper); and Bayesian combination of linear models where a few informative Bayesian linear models (called human models in this paper) are averaged. In the human models, variables are selected subjectively and possibly transformed, and informative priors are specified. Furthermore, the prior weights of the various models are chosen subjectively and updated using Bayes theorem if data not used for model building become available.

3.1 The Basic Algorithm

For combining the Bayesian regression procedures, we propose Algorithm 3, called clustered Bayesian averaging (CBA), which assigns local weights to each procedure. For simplicity, we only consider the combination of two Bayesian regression procedures in this paper. The method can be easily extended to more than two procedures. We also assume that n is even.

Algorithm 3. *Clustered Bayesian Averaging for Two Models.*

(i) Specify the Bayesian models $f_j(y|\mathbf{x}, \grave{} _j)$ *and the prior distributions* $p_j(\grave{} _j)$, *for* $j = 1, 2$.

(ii) Repeat the following steps for $q = 1, 2, \ldots, Q$, *where* Q *is a large pre-specified number:*

(a) Randomly permute the cases in the data set $\mathbf{Z}$, *and then split it into two parts* $\mathbf{Z}^{(1)} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{n/2}$ *and* $\mathbf{Z}^{(2)} = \{(\mathbf{x}_i, y_i)\}_{i=n/2+1}^{n}$.

(b) *Obtain the posterior distribution,* $p_j(\grave{} _j|\mathbf{Z}^{(1)})$*, of the parameters* $\grave{} _j$ *of model* $f_j(y|\mathbf{x}, \grave{} _j)$ *conditional on* $\mathbf{Z}^{(1)}$*, for* $j = 1, 2$*.*

(c) *For each* f_j*, use* $p_j(\grave{} _j|\mathbf{Z}^{(1)})$ *as the prior distribution and evaluate the marginal likelihood of each observation in* $\mathbf{Z}^{(2)}$*. Denote the marginal likelihood of the observation with index* k *by* M_{jk}*.*

(d) *Calculate the log-Bayes factor evaluated at each observation in* $\mathbf{Z}^{(2)}$*, obtaining*

$$z_k = \log BF_k = \log \frac{M_{1k}}{M_{2k}}, \quad \text{for } k = n/2 + 1, \ldots, n.$$

(e) *Run classification and regression trees (CART, Breiman et al., 1984) to construct a regression tree using* $\{z_k\}_{k=n/2+1}^n$ *as the response and the associated covariate values* $\{\mathbf{x}_k\}_{k=n/2+1}^n$ *as the predictors. This clusters the values* $\{z_k\}_{k=n/2+1}^n$ *and creates a partition of the covariate space.*

(f) *Approximate the local Bayes factor for each element of the partition by making use of the independence approximation. For a given element of the partition, use* $\sum z_k$*, with the sum running over observations in* $\mathbf{Z}^{(2)}$ *that fall in the element.*

(g) *For a covariate vector* $\mathbf{x}_t$*, let* $\hat{z}_{tq}$ *denote the log local Bayes factor from Step 2f.*

(iii) *For* $j = 1, 2$*, obtain the posterior distribution,* $p_j(\grave{} _j|(\mathbf{x}_i, y_i)_1^n)$*, for the parameters* $\grave{} _j$ *in* $f_j(y|\mathbf{x}, \grave{} _j)$ *based on all observations in* $\mathbf{Z}$*. The updated models are used to make predictions.*

(iv) *For* $j = 1, 2$*, denote by* $\hat{y}_{jt}$ *the prediction from model* j *conditional on covariate* $\mathbf{x}_t$*, and let* π_j *represent the prior probability of model* j*. Define* $\hat{z}'_{tq} = \hat{z}_{tq} + \log(\pi_1) - \log(\pi_2)$ *and let* $w_{1t} = \frac{1}{Q}\sum_{q=1}^{Q} [exp(\hat{z}'_{tq})]/[1 + exp(\hat{z}'_{tq})]$*. The final prediction for observation* t *is calculated as*

$$\hat{y}_t = w_{1t} \cdot \hat{y}_{1t} + (1 - w_{1t}) \cdot \hat{y}_{2t}.$$

Comments on Algorithm 3

(i) In Step 2c, if the analytic form of the marginal likelihood cannot be obtained but values of $\grave{} _{ji}$ can be simulated from $p_j(\grave{} _j|\mathbf{Z}^{(1)})$, for $i = 1, \ldots, N$, then M_{jk} can be estimated as

$$\widehat{M}_{jk} = \frac{1}{N}\sum_{i=1}^{N} f_j(y_k|\mathbf{x}_k, \grave{} _{ji}),$$

where $k = n/2 + 1, \ldots, n$.

(ii) Because the training data are split repeatedly, the final weights (the means of the weights from the different splits) for each procedure become smoother across regions. As a consequence, the averaged model is more stable. Also, this method is less likely than BMA to select a single model in a given region.

(iii) The method can be easily extended to combine more than two procedures using the geometric mean of the estimated Bayes factors (see Yu et al. 2011). The extension requires replacement of Step 2e with a partitioning of covariate space based on a multivariate summary of models in Step 2d. Possible summaries include designation of a reference model M_1 and use of Bayes factors for other models with respect to this model, use of the average marginal likelihood or maximum marginal likelihood in place of M_{1k} in the equation in Step 2d, or simply setting $z_k = \log(M_{ik})$ for model i. A multivariate version of CART can be used to partition the covariate space in Step 2e. The performance of these procedures has been robust to differences in implementation in the examples we have examined.

3.2 Modified algorithms

The basic algorithm can be modified in various ways to address theoretical and computational concerns.

CBA.boot: Bootstrapped algorithm
A first modification is motivated by resampling considerations. The algorithm remains exactly the same as the basic algorithm, except for the fact that the random split in Step 2a does not cut the training data set into two halves. Instead, a bootstrap sample of n observations $\mathbf{Z}_{\mathrm{B}}^{(1)}$ is generated and used to fit the models (this set plays the same role as $\mathbf{Z}^{(1)}$ in the original algorithm). The observations not included in $\mathbf{Z}_{\mathrm{B}}^{(1)}$ are included in the set $\mathbf{Z}_{\mathrm{B}}^{(2)}$ and are used to derive the weights for the models (this set plays the same role as $\mathbf{Z}^{(2)}$ in the original algorithm). The updating in Step 3 is based on the original $\mathbf{Z}$.

CBA.pll: Pseudo-loglikelihood algorithm
In this modification of the algorithm we pursue a different strategy to effect covariate space clustering. Rather than dividing the training data set into two parts and using the log-Bayes factors of Step 2d to obtain a partition using CART, we feed directly into CART the pseudo-loglikelihood of each observation in the entire training data set. Note that, in this implementation, no random splits of the training data set are involved and therefore we set $Q = 1$ at the beginning of Step 2.

CBA.pll.loc: Pseudo-loglikelihood algorithm with local updating
Here we modify the pseudo-loglikelihood algorithm to incorporate local updating. First, as in the original pseudo-loglikelihood algorithm, we perform covariate space clustering based on the whole training data set. Next, we perform repeatedly the following two steps: a) calculate the pseudo-loglikelihood conditional only on the training data in each cluster, and b) use the newly calculated pseudo-loglikelihood to perform a new clustering. Steps (a) and (b) are repeated until the calculated pseudo-loglikelihood values converge to constants. The algorithm is terminated after a prespecified large number of iterations if convergence is not attained. Because the local updating is conditional only on the observations that fall in a given cluster, if we work with improper priors, we need to enforce the condition that the size of a given cluster is large enough to guarantee propriety of the posterior.

CBA.pll.boot: Bootstrapped pseudo-loglikelihood algorithm
This and the following modification of the basic pseudo-loglikelihood algorithm are intended to produce smooth weights. In the bootstrap method, the pseudo-loglikelihoods of the training data are resampled *with* replacement and the bootstrap sample is used for covariate space clustering. The process is repeated several times, and the prediction weights are the average weights computed in the various iterations.

CBA.pll.sub(p): Subset pseudo-loglikelihood algorithm
An alternative to resampling with replacement is to perform the covariate space clustering by using subsets of varying sizes (a fraction p of the size of the training data set) sampled *without* replacement from the training data set. Similarly to the CBA.pll.boot modification, for a given subset size, the process is repeated several times, and the prediction weights are the average weights computed in the various iterations. (In the examples, we considered subsets of size 0.5 to 0.95 times the size of the training data set, in increments of 0.05.)

CBA.l.g(f): Mixed local and global updating algorithm
This modification is motivated by the desire to perform some amount of local updating that is in agreement with the derived partitioning of the covariate space. First, we perform a covariate space clustering based on the basic pseudo-loglikelihood algorithm. The model updating in Step 3, motivated by the fractional Bayes factor, combines elements of global and local updating. Specifically, a fractional power f of the likelihood of all of the data is used for global updating and an additional fractional power $(1 - f)$ of the likelihood of those data falling within a region is used for local updating.

CBA.med: CBA with median weighting
This and the following modifications are based on alternative ways of computing the model weights. The weights given to the various procedures as calculated in Step 4 of the basic algorithm are the means of the weights from the split training data. As for the intrinsic Bayes factor (Berger and Pericchi 1998), when the training data are very noisy, using the median of the weights may be a more robust choice.

CBA.thresh: CBA with threshold weighting
Another weighting strategy is based on thresholding the weights. For example, when the final weight of a model is smaller than 0.25, we downweight that model to 0. Similarly, if the weight is larger than 0.75, the model receives a weight of 1. In other words, we choose a single model (rather than an average of the two models) for prediction in a given region when that model is considerably better than the other model over that region. The suggested thresholding value is loosely based on Kass and Raftery's (1995) rule of thumb for Bayes factors, in which odds of 3 to 1 are taken as the transition from negligible to positive evidence in favor of a model.

4 Experiments

In this section, we illustrate the use of CBA with simulations and real data. Throughout, we examine the performance of the method when the models are inadequate–that is,

the models cannot match the "true" data-generating mechanism, even with the best possible parameter values plugged in. We refer to such models as "biased" models. All of the simulations involve biased models, and we strongly believe that the real data analyses do as well. In Simulation 1, we show the effect of CBA when very simple models are combined. The performance of CBA when combining similar models (Bayesian linear regressions with predictive variables chosen by AIC or BIC) is explored in Simulation 2. The performance of CBA when combining very different models is explored in Simulation 3 (Bayesian linear models and Bayesian additive regression trees (BART)) and Simulation 4 (Bayesian linear model and nonlinear model). Then, we perform CBA on two real data sets. First, we present the results for a data set on daily ozone concentrations where the models being combined were built by three different analysts. Next, we present the results for a breast cancer data set where the two survival models being combined are differently motivated.

The comparisons also involve a suite of competing techniques, including BART, BMA based on all potential linear models resulting from the inclusion/exclusion of the available predictors, and those based on selection of a model by AIC or BIC. We also compare CBA to two versions of treed Gaussian processes (Gramacy and Lee, 2008), which average over treed partitions of the covariate space. One version (which we denote by 'tgp') fits separate Gaussian processes and one (which we denote by 'tgpllm') fits simpler linear models within each component of a partition.

These methods have been implemented with freely available software at the default settings. For each of the simulations, models are fit to a data set of 200 observations and evaluated on a separate test set of 100 observations. Each simulation was replicated 100 times and the tables show averages of summaries across the 100 replicates.

4.1 Simulation 1: CBA for simple models

This simulation shows that, when the true model and the models to be combined are all simple (and the truth can be represented as a linear combination of these models), CBA can successfully reduce the bias of the combined set of models. Here, we consider only linear regression models. The true regression function is given by:

$$y_i = 0.9 + 1.6x_{1i} + 1.6x_{2i} + 1.7x_{3i} + \epsilon_i,$$

where the predictors x_1, x_2, and x_3 are iid uniform random variables on $[0, 1]$ and the errors ϵ_i are iid normal random variables with mean 0. Various values of the error variance are considered to obtain noise-to-signal (NTS) ratios ranging from 0.5 to 4.

Two biased Bayesian linear models are fit:

- Model 1: $g(\mathbf{x}) = \alpha_0 + \alpha_1 x_1 + \alpha_2 x_2$;
- Model 2: $g(\mathbf{x}) = \beta_0 + \beta_1 x_1 + \beta_2 x_3$;

with flat noninformative priors for all the parameters. The two models are then combined using CBA. In this example, the average of the two models' mean functions is a near match for the true mean function.

Table 1 contains the means of the sum of squared errors (SSE) over the 100 replicates in the simulation. As expected, model averaging via CBA and its variants allows us to closely match the true mean function, leading to a substantial decrease in SSE compared to either single model. We see that the effect of bias decreases as the error variance (and hence NTS ratio) increases, following the decomposition $E(SSE) = Bias^2 + Error\ Variance$. CBA outperforms treed Gaussian processes, except for small NTS ratios. We attribute this to treed Gaussian processes chasing the data more than our implementations of CBA.

The improvement in SSE does not carry over to improvement in log-likelihood, as shown in Table 2. To understand this phenomenon, we note that the target $g(\mathbf{x})$ is roughly the mean of the two model-specific $g(\mathbf{x})$. At this midpoint, the likelihood under the two models will be approximately the same, and so we see no appreciable change in out-of-sample log-likelihood.

NTS	Model 1	Model 2	CBA	CBA.pll
0.5	40.81	40.24	29.66	33.25
1.0	93.45	95.09	82.52	88.47
1.5	182.25	186.24	172.06	180.59
2.0	307.20	313.68	297.11	303.75
2.5	468.30	477.42	458.32	467.96
3.0	665.56	677.46	656.07	662.25
3.5	898.98	913.79	890.07	895.02
4.0	1168.55	1186.42	1160.22	1164.32
NTS	CBA.pll.sub(.5/.75/.95)	CBA.pll.boot	tgp	tgpllm
0.5	27.22/27.04/27.29	27.60	25.15	24.95
1.0	82.27/81.94/82.45	82.42	81.47	82.27
1.5	172.12/172.43/173.05	172.92	176.78	176.24
2.0	297.52/298.15/299.32	298.32	308.14	306.77
2.5	458.94/460.64/461.06	460.77	481.20	481.08
3.0	656.68/658.41/658.78	658.48	682.49	676.97
3.5	890.18/891.85/892.17	892.08	942.91	927.51
4.0	1160.26/1161.57/1161.57	1161.37	1206.88	1215.37

Table 1: CBA performance on simple models and comparisons with treed Gaussian processes. Cells show the average SSE of each method over the 100 replicates of the simulation.

4.2 Simulation 2: CBA on models built via AIC/BIC

We assess the effect of CBA when the models to be combined are similar. For this illustration, we consider linear regression models where variables are selected according to AIC or BIC. The true underlying regression function is one of the following functions of the variables x_j. The x_j are independently drawn from the Uniform $[0, 1]$ distribution:

- Case 1: $g(\mathbf{x}) = 0.9 + 1.5x_1 + 1.6x_2 + 1.7x_3 + 1.5x_4 + 0.4x_5 + 0.3x_6 + 0.2x_7 + 0.1x_8$.

NTS	Model 1	Model 2	CBA
0.5	-97.55	-97.52	-97.54
1.0	-139.15	-139.10	-139.13
1.5	-172.53	-172.49	-172.51
2.0	-198.60	-198.58	-198.59
2.5	-219.68	-219.64	-219.66
3.0	-237.23	-237.21	-237.22
3.5	-252.24	-252.24	-252.24
4.0	-265.37	-265.35	-265.36

Table 2: CBA performance on simple models. Cells show the average log marginal likelihood on the test data over the 100 replicates of the simulation.

- Case 2: $g(\mathbf{x}) = 1 + x_1 + x_2 + x_3 + x_4$.
- Case 3: $g(\mathbf{x}) = \begin{cases} 1 + x_1 + x_2 + x_3 + x_4, & \text{if } x_2 < u, \\ 0.9 + 1.5x_1 + 1.6x_2 + 1.7x_3 + 1.5x_4 + 0.4x_5 + \\ \quad 0.3x_6 + 0.2x_7 + 0.1x_8, & \text{otherwise.} \end{cases}$
- Case 4: $g(\mathbf{x}) = \begin{cases} 1 + x_1 + x_2 + x_3 + x_4, & \text{if } x_9 < u, \\ 0.9 + 1.5x_1 + 1.6x_2 + 1.7x_3 + 1.5x_4 + 0.4x_5 + \\ \quad 0.3x_6 + 0.2x_7 + 0.1x_8, & \text{otherwise.} \end{cases}$
- Case 5: $g(\mathbf{x}) = \begin{cases} 1 + x_1 + x_2 + x_3 + x_4, & \text{if } x_6 < u, \\ 0.9 + 1.5x_1 + 1.6x_2 + 1.7x_3 + 1.5x_4 + 0.4x_5 + \\ \quad 0.3x_6 + 0.2x_7 + 0.1x_8, & \text{otherwise.} \end{cases}$

The NTS ratio is 2. In each replicate or the simulation, the variable u in cases 3 to 5 is a fixed number drawn from a uniform distribution in $[0, 1]$ and is neither used for model building nor for weight calculation. Case 1 is a more complex linear model than Case 2 because it contains more variables with various coefficients. For Cases 3, 4, and 5, one variable determines a binary split of the covariate space, so that different regression models are specified in different regions. In Case 3 the splitting variable x_2 is used for building both models; in Case 4 the splitting variable x_9 is not used in building either model; in Case 5, the splitting variable x_6 is used for building model 2 but not model 1.

The performance of CBA is compared with that of BMA, treed Gaussian processes, and of the models chosen by BAIC or BBIC in Table 3, where the cells show the average SSE from the 100 replicates of the simulation. Because the linear models chosen by AIC or BIC are not very different, the improvement from use of CBA is also not very large. However, CBA consistently performs better than BAIC, BBIC, BMA, and treed Gaussian processes. This is especially true for Cases 3, 4, and 5, where the true models differ in different regions of the covariate space. The values for CBA.l.g(0.6) and CBA.l.g(0.2) indicate that performing a mix of local and global updating may be

useful, provided the fraction of local updating is not too small.

Case	BAIC	BBIC	CBA	CBA.med	CBA.l.g(.2)
1	345.91	348.77	344.39	344.35	348.6698
2	135.75	134.99	134.58	134.59	135.519
3	633.55	632.43	627.38	627.44	628.8996
4	516.49	516.90	510.70	510.19	519.6447
5	511.78	513.51	507.46	507.55	514.4719
Case	CBA.l.g(0.6)	CBA.pll	BMA	tgp	tgpllm
1	343.85	346.21	345.30	358.78	359.41
2	134.7059	134.25	134.30	140.54	138.15
3	625.5657	622.48	628.76	652.99	643.97
4	512.6894	513.65	516.92	519.32	521.00
5	508.4137	506.27	513.83	528.22	521.11

Table 3: CBA performance on Bayesian linear models and comparisons with other methods. Cells show the average SSE over the 100 replicates of the simulation.

Table 4 shows performance of CBA in terms of log marginal likelihoods. The performance of CBA is essentially equivalent to that of BAIC and BBIC, although the numbers do tilt slightly in favor of CBA.

Case	BAIC	BBIC	CBA	CBA.med
1	-205.53	-205.95	-205.36	-205.36
2	-158.52	-158.22	-158.02	-158.02
3	-233.47	-233.24	-232.86	-232.86
4	-223.83	-223.72	-223.13	-223.09
5	-223.04	-223.10	-222.50	-222.51

Table 4: CBA performance on Bayesian linear models. Cells show the average log marginal likelihood over the 100 replicates of the simulation.

4.3 Simulation 3: CBA on models built via AIC/BIC and BART

In this simulation, we assess the performance of CBA when very different models are combined. Specifically, we combine Bayesian linear models where covariates are chosen by AIC/BIC with Bayesian Additive Regression Trees (BART) models. The underlying true models are as follows:

- Case 1: $g(\mathbf{x}) = (x_1 - 0.2)^2$.
- Case 2: $g(\mathbf{x}) = e^{(3x_1 - 0.5)}$.
- Case 3: $g(\mathbf{x}) = 10\sin(\pi x_1 x_2 + 20x_3 - 0.5)^2 + 10x_4 + 5x_5$.
- Case 4: $g(\mathbf{x}) = (x_1^2 + (x_2 x_3 - 1/(x_2 x_4))^2)^{0.5}$.

Case	BART	AIC	BIC	CBA:AIC	CBA:BIC
1	2.60	2.93	2.96	2.54	2.55
2	720.88	780.49	786.59	698.29	701.80
3	2017.42	2315.60	2280.60	2022.40	2010.24
4	12392168	12763852	12660616.7	12048327	11994398
5	9.25	11.22	11.24	9.10	9.08
Case	CBA.pll:AIC	CBA.pll:BIC	tgp	tgpllm	BMA
1	2.60	2.62	2.76	2.64	2.95
2	705.90	713.08	767.62	739.05	785.26
3	2103.13	2031.11	2418.47	2217.74	2269.81
4	12415111	12324536	11606904	11665620	12675323
5	9.35	9.35	9.86	10.60	11.37

Table 5: CBA performance when combining Bayesian linear models and BART models, and comparisons with other methods. CBA:AIC and CBA:BIC refer to the methods that use CBA to combine a BART model with a Bayesian linear model whose covariates are chosen by AIC and BIC, respectively. The table entries show average SSE over the 100 replicates of the simulation.

- Case 5: $g(\mathbf{x}) = \tan^{-1}((x_2x_3 - 1/(x_2x_3))/x_1)$.

The NTS ratio is 0.8. In Cases 1 and 2, x_1 has a Uniform $[0, 1]$ distribution. The functions in Cases 3 to 5 were considered in the papers on multivariate adaptive regression splines (MARS, Friedman, 1991) and bagging (Breiman, 1996). For Case 3, there are 10 independent predictors $x_1, \ldots, x_{10}$, all of them uniformly distributed over $[0, 1]$. For Cases 4 and 5, there are also 10 independent uniform predictors, but the supports of their distributions vary. Specifically, $x_5, \ldots, x_{10}$ are supported over $[0, 1]$ and x_1 to x_4 are supported over these ranges:

$$\begin{aligned} 0 &\leq x_1 \leq 100, \\ 20 &\leq x_2/2\pi \leq 280, \\ 0 &\leq x_3 \leq 1, \\ 1 &\leq x_4 \leq 11. \end{aligned}$$

Table 5 shows the average SSE for each case. We observe that BART outperforms the linear models. BMA is competitive with choice of the linear model by AIC or by BIC. CBA is consistently the top performer among all these methods and outperforms treed Gaussian processes in all but one case.

In terms of log marginal likelihood, as shown in Table 6, CBA shows an overall modest improvement. For both SSE and log marginal likelihood, the CBA half-sample update provided better results than the all-but-one update of the pseudo-likelihood variant of CBA.

Case	BART	AIC	BIC	
1	40.05	34.50	34.03	
2	-241.00	-244.88	-245.26	
3	-292.09	-299.63	-298.72	
4	-729.01	-730.46	-729.94	
5	-24.95	-33.75	-33.79	
Case	CBA:AIC	CBA:BIC	CBA.pll:AIC	CBA.pll:BIC
1	40.94	40.67	40.02	39.52
2	-239.76	-240.00	-239.95	-240.34
3	-292.57	-292.24	-294.24	-292.61
4	-727.42	-727.18	-729.29	-728.88
5	-23.42	-23.37	-25.08	-25.10

Table 6: CBA performance when combining Bayesian linear models and BART models. CBA:AIC and CBA:BIC refer to the methods that use CBA to combine a BART model with a Bayesian linear model whose covariates are chosen by AIC and BIC, respectively. The table entries show the average log marginal likelihood over the 100 replicates of the simulation.

4.4 Simulation 4: CBA on linear and non-linear models

In this simulation we evaluate the performance of several versions of CBA when a linear model and a non-linear model are combined. The true model is designed so that there are special local effects: $g(\mathbf{x}) = 2x^{0.8}\sin[z\pi(x^{-1} + 0.01x)]$. The error term is normally distributed with a standard deviation of 0.4. We fit two models:

- Model 1: $g(\mathbf{x}) = a + bx$.
- Model 2: $g(\mathbf{x}) = c + d\sin(2\pi x) + e\sin(4\pi x)$.

Non-informative priors are used for all parameters. For prediction purposes, the posterior mean is used to estimate the parameters. CBA is compared with the combined model where each single model is assigned equal weight (convex synthesis or CS, Yu et al. 2011).

The boxplots of Figure 1 show the performance of the methods on 100 replicates of the simulation. We observe that, compared with the single models, the various types of CBA, with the assignment of local weights, improve predictive performance as measured by SSE. The performance of CS is comparable to that of the linear model. As shown in Figure 2, similar conclusions can be drawn in terms of log marginal likelihoods, except for the fact that the performance of CBA.boot is only slightly better than that of CS.

The top panel of Figure 3 shows the original data and the fitted values from the various models. The bottom panel shows the weights assigned to Model 1 by the various versions of CBA in one of the simulations. The figure shows that CBA works better. The improvement comes from CBA's ability to adaptively assign weights to the two models according to individual model performance in a given covariate region.

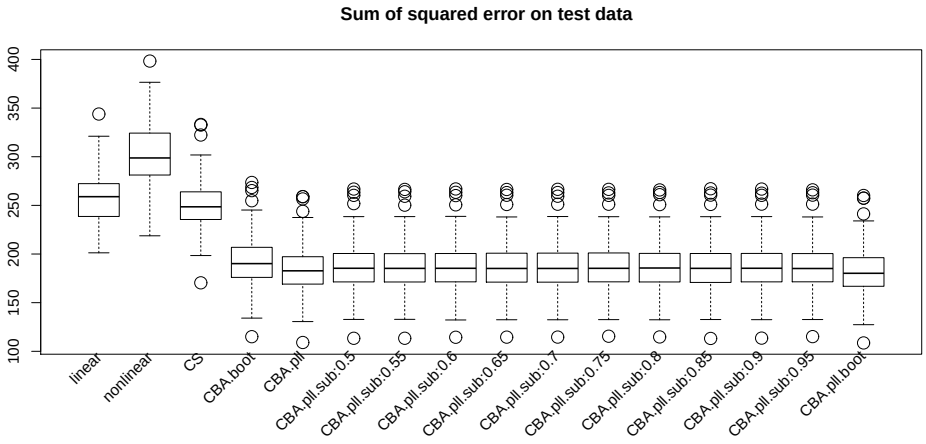

Figure 1: Comparison of various types of CBA with single models and convex synthesis (CS) in terms of sum of squared errors of prediction.

4.5 CBA on ozone data

The ozone data set (contained and documented in the software package R) consists of daily measurements of ozone concentration and eight meteorological quantities in the Los Angeles basin for 330 days in the year 1976. A detailed description of the data set can be found in Breiman (2001). Each of us analyzed one third of the data obtained by random partitioning and constructed a Bayesian model for predicting ozone concentration. Based on a set of ground rules that we had specified, each model produced a distribution for ozone concentration supported on the non-negative integers. In this way, we were able to consider three pairs of analysts, with one third of the data reserved for evaluation of the pair's combined analysis.

For details of the Bayesian models built, the readers are referred to Yu et. al (2011). Tables 7 and 8 shows the results of the analyses. CBA is compared with convex synthesis and mean human prediction (Yu et. al 2011). In convex synthesis analyses were updated over the whole training data set and were equally weighted to predict the test data. In mean human prediction, one of the models was selected with probability 0.5 to predict the test data. In the SSE simulation, CBA is also compared with two versions of treed Gaussian processes. We observe that, on average, CBA outperforms the other methods (including CBA.pll) and can significantly improve the performance of any single model. The improvement holds for both SSE and log-likelihood.

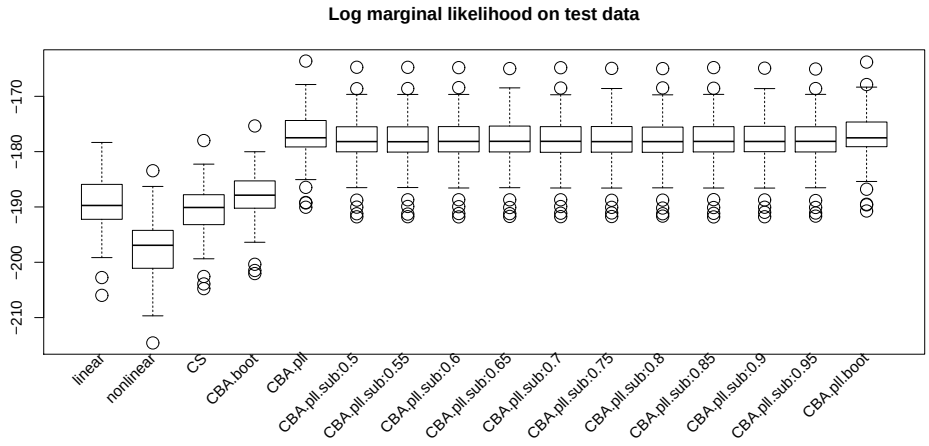

Figure 2: Comparison of various types of CBA with single models and convex synthesis (CS) in terms of log marginal likelihood.

4.6 CBA on SEER data

We evaluate the performance of CBA in classification problems by predicting the five-year survival rate for female breast cancer patients. The primary data used in this analysis are derived from individual patient information collected by the Surveillance, Epidemiology, and End Results (SEER) Program. We used the SEER 1973-2009 research data file from August 2012 through the SEER*stat software. More information about the SEER program and SEER*stat can be found at the SEER website: seer.cancer.gov

Patients included in the analysis are women aged 35 to 59 at diagnosis with malignant breast cancer, who have at least six lymph nodes sampled, and have been actively followed up. In addition, to be included in the analysis, the patients must have known tumor size and number of nodes positive for tumor. The inclusion rules and the variables chosen for building the models are similar to those used by the Adjuvant! Online program (for details, see Ravdin et al., 2001). We used cases diagnosed between 2000 and 2002 as training data and cases diagnosed in 2003 as test data. There are a total of 44,355 cases in the training data set and 11,877 cases in the test data set. There are 405 cases in the test data set that were lost to follow-up within 5 years.

To build the first model, we used the following variables:

(i) age group (5 levels: $35-39, 40-44, 45-49, 50-54, 55-59$)

(ii) tumor grade (4 levels: well-, moderately-, poorly- or un-differentiated, and undefined)

(iii) tumor size (5 levels: $0-10, 11-20, 21-30, 31-50, >50$cm)

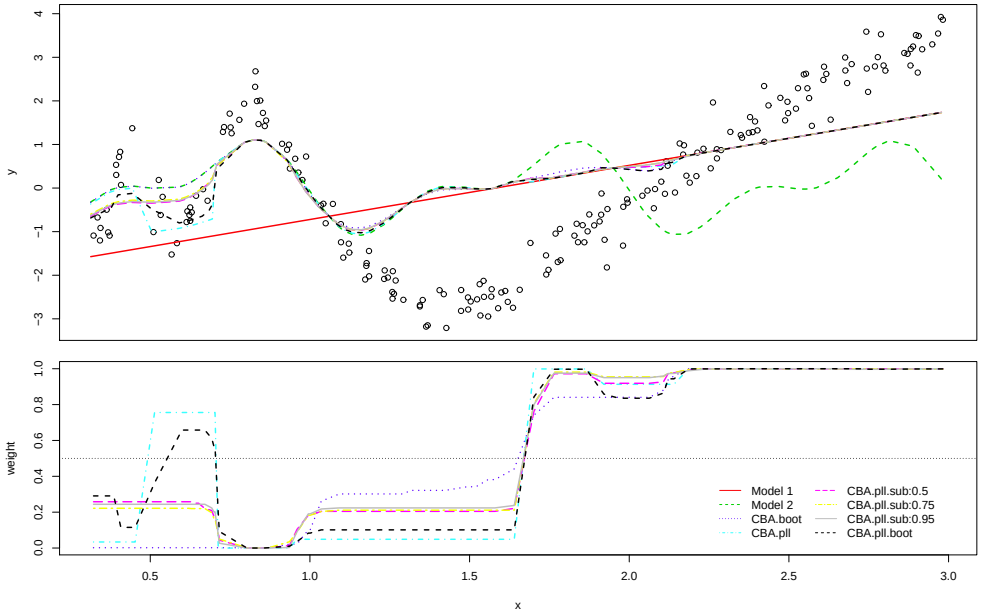

Figure 3: Comparison of various types of CBA with single models. The original data and the fitted values from the various models are plotted in the top panel. The bottom panel shows the weights assigned to Model 1 by the various versions of CBA in one of the simulations.

(iv) number of positive nodes (4 levels: $0, 1-3, 4-9, >9$)

(v) er (estrogen receptor - 3 levels: positive, negative, borderline or unknown)

The prior distributions for the survival rates were set as Beta(0.986, 0.014) for the first year and Beta(0.97, 0.03) for the second to the fifth year, independently. The data-driven priors were chosen so that their means are close to the mean survival rate for the whole population in the same data sets. The sum of the parameters in the beta distributions is 1, giving the prior an influence of about 1 case. For Model 1, the covariate space is divided by the covariates into 1,200 ($5\times4\times5\times4\times3$) regions. Posterior survival rates are updated in each region independently.

To build Model 2 we used the same covariates as in Model 1, but in the original format, where age group, tumor size, and number of positive nodes are treated as continuous. Logistic regression models were fit for each time period to predict the follow-up probabilities, and the probabilities of death for the followed-up cases. The estimated probabilities were then used for predicting 5-year survival rates.

Test Data	1	2	3
Analyst 1	-	12.31	14.65
Analyst 2	17.96	-	15.66
Analyst 3	15.96	14.21	-
Mean Human	16.96	13.26	15.15
Convex Synthesis	15.98	11.93	13.39
CBA	15.67	11.89	13.52
CBA.pll	15.57	13.65	14.49
tgp	26.19	13.06	14.06
tgpllm	25.68	12.74	14.23

Table 7: Comparison of CBA with mean human prediction, convex synthesis, and treed Gaussian processes by sum of squared errors for log ozone. The column labels Test Data 1, 2, and 3 indicate which third of the data was used as the test data (with the other two thirds having been used for model building).

Test Data	1	2	3
Analyst 1	-	-274.68	-300.05
Analyst 2	-291.10	-	-300.07
Analyst 3	-286.06	-284.96	-
Mean Human	-288.58	-279.82	-300.06
Convex Synthesis	-284.55	-275.69	-293.36
CBA	-283.54	-275.58	-293.94
CBA.pll	-283.96	-280.11	-301.10

Table 8: Comparison of CBA with mean human prediction and convex synthesis by log marginal likelihood for log ozone. The column labels Test Data 1, 2, and 3 indicate which third of the data was used as the test data (with the other two thirds having been used for model building).

We used the likelihood ratio evaluated at the MLEs for the two models to approximate the likelihood ratios needed to calculate the CBA weights. In the weight calculation, we included all variables used to build Models 1 and 2, and added the two variables laterality and progesterone receptor (pr) status. These two variables were considered in the model-building phase, but were found to be less important in predicting survival than the other variables. Table 9 compares CBA's performance with that of the two individual models and that of convex synthesis with equal weights in terms of SSE, misclassification, and log-likelihood. We find that CBA provides superior prediction to any of the competitors that we considered in terms of misclassification and SSE. CBA.med performs particularly well for log-likelihood.

	# of misclassifications	SSE	Log-likelihood
Model 1	1475	1149.58	-3889.39
Model 2	1502	1173.09	-3952.40
Convex Synthesis	1478	1138.48	-3826.73
CBA.med	1450	1125.48	-3802.92
CBA	1465	1135.74	-3869.61

Table 9: Predictive results on the SEER test data.

5 Discussion and Future Research

In this paper we developed the concept of a local Bayes factor and we propose to use local Bayes factors to combine models for prediction in a way that accounts for regionally-different performance across the covariate space. We proposed the clustered Bayesian averaging algorithm where training data are used for both model updating and evaluation. Given various models to be combined for prediction, we split the covariate space into regions where the relative model efficiency is about the same. Then local Bayes factors are approximated in a computationally efficient fashion in each region to derive weights for model combination. We show through simulations and real data examples that CBA can effectively improve prediction performance.

We considered several variants of the basic CBA algorithm to evaluate the impact of certain implementation choices on the overall performance of the procedure, both in terms of out-of-sample prediction accuracy and likelihood evaluation. The following motivating questions were behind the modified CBA algorithms (either one at a time or in various combinations). Can repeated random splits of the training data be avoided, thus reducing the overall computational effort? Are there alternative, effective ways to perform the splits of the training data? How best can we produce posterior weights that are both smooth and robust to sampling variability? What mix of local and global updating is most beneficial? On the one hand, with the procedure emphasizing local features of the inferential problem, it is conceptually appealing to update the distributions of the model parameters conditional only on local data. On the other hand, empirical evidence has shown us that relying exclusively on local updating produces unstable results. Not surprisingly, the relative merits of the different variants changed from simulation to simulation, depending on the nature of the true underlying model and of the models contributing to the locally weighted analysis. However, a few unifying themes emerged. First, a substantial amount of global updating is needed for stable and reliable performance. Second, smooth weights tend to produce better out-of-sample performance, but, with our implementation, smoothness comes with additional computational cost. (In many circumstances there is little or no deterioration in performance associated with the use of discontinuous weights, and in some circumstances discontinuity can improve performance.)

The next question becomes how to best formalize the notions that underlie CBA. There is a natural path to further development of the local Bayes factor. The first step is to create a set of diagnostics, both graphical and numerical, to aid in determination of

whether there is a need to turn to local weights. The difficulty in this task lies in the fact that relative regional performance varies systematically when models are compared, and we seek evidence of variation that does not conform to these systematic patterns. The patterns are easily identified for low-dimensional parametric models, but they are more challenging to identify for more complex models.

The second step is to develop an array of techniques to capture the changes in the local Bayes factor. Choice of a specific technique would ideally be motivated by diagnostics on the log-surface. Tree-based techniques favor weights that are discontinuous and have jumps; a Gaussian process for the log local Bayes factor induces continuity but must account for the systematic trends inherent in the model comparison; particular forms for the local Bayes factor relate to mixture of experts models. Ideally, the diagnostics of the first step and problem specific knowledge would combine to determine the form of these changes.

The third step is to create effective computational strategies to compute the local Bayes factors. Ideally, further work would enable one to replace the independence approximations we have made with milder assumptions of conditional independence.

Fourth, an investigation of the theoretical properties of the algorithms would enhance our understanding of their benefits. The methods, in any form, focus on a trade-off between fidelity to the expected log Bayes factor surface (with better fidelity following from smaller regions) and quicker movement toward choice of a model (hastened by larger regions and effectively greater amounts of data). Asymptotic results will establish consistency for various implementations of the method. Realistic asymptotics are rendered more difficult because the basic premise of the work is that the true data-generating mechanism lies outside the class of models that are being fit.

References

Berger, J.O. (1994). "An overview of robust Bayesian analysis" (with discussion) , *Test*, 3, 5-124.

Berger, J.O. and Pericchi, L.R. (1996). "The Intrinsic Bayes Factor for Model Selection and Prediction", *Journal of the American Statistical Association*, 91(433), 109-122.

Berger, J. O., and Pericchi, L. R. (1998). "Accurate and Stable Bayesian Model Selection: The Median Intrinsic Bayes Factor," *Sankhya, Series B*, 60, 118.

Breiman L., Friedman J.H., Olshen R.A., and Stone, C.J. (1984), *Classification and Regression Trees*, Wadsworth, Belmont, Ca.

Breiman L. (1996), "Bagging Predictors," *Machine Learning*, 26, 123-140.

Breiman L. (2001), "Statistical Modeling: The Two Cultures," *Statistical Science*, 16, 199-215, (Disc: p216–231).

Casella, G. and Moreno, E. (2006). "Objective Bayesian variable selection," *Journal of the American Statistical Association*, 101, 157-167.

Chipman H. A., George E. I., and McCulloch R. E. (2010), "BART: Bayesian Additive Regression Tree", *Annals of Applied Statistics*, 4(1), 266-298.

Clyde, M. A., Ghosh, J. and Littman, M. (2011). "Bayesian Adaptive Sampling for Variable Selection and Model Averaging," *Journal of Computational and Graphical Statistics*, 20, 80-101.

Friedman, J.H. (1991), "Multivariate Adaptive Regression Splines" (with discussion), *Annals of Statistics*, 19, 1-67.

Geisser, S. and Eddy, W.F. (1979). "A Predictive Approach to Model Selection," *Journal of the American Statistical Association*, 74, 153-160.

George, E.I. and McCulloch, R.E. (1993), "Variable Selection Via Gibbs Sampling," *Journal of the American Statistical Association*, 88, 881-889.

Gramacy, R. B. and Lee, H. K. H. (2008), "Bayesian Treed Gaussian Process Models with an Application to Computer Modeling," *Journal of the American Statistical Association*, 103, 1119-1130.

Hans, C., Dobra, A., and West, M. (2007), "Shotgun Stochastic Search for "Large p" Regression," *Journal of the American Statistical Association*, 102, 507-516.

Hjort, N.L. and Claeskens, G. (2003), "Frequentist Model Average Estimators", *Journal of the American Statistical Association*, 98, 879-899.

Jeffreys, H. (1961), *Theory of Probability* (3rd ed.), Oxford, U.K.: Oxford University Press.

Jordan, M.I., and Jacobs, R.A. (1994), "Hierarchical Mixtures of Experts and the EM Algorithm," *Neural Computation*, 6, 181-214.

Kadane, J. B. and Lazar, N. A. (2004). "Methods and criteria for model selection" *Journal of the American Statistical Association*, 99, 279-290.

Kass, R.E. and Raftery, A.E. (1995), "Bayes factors," *Journal of the American Statistical Association*, 90, 773-795.

Lempers, F. B. (1971). *Posterior Probabilities of Alternative Linear Models*, Rotterdam: University Press.

Madigan D. and Raftery A. E., (1994). "Model Selection and Accounting for Model Uncertainty in Graphical Models Using Occam's Window," *Journal of the American Statistical Association*, 89, 1535-1546.

Madigan, D. and York, J. (1995). "Bayesian graphical models for discrete data," *Inter-*

national Statistical Review, 63, 215-232.

O'Hagan A. (1995), "Fractional Bayes Factors for Model Comparison," *Journal of the Royal Statistical Society, Series B*, 57(1), 99-138.

Raftery A.E., Madigan D. and Hoeting J.A. (1997), "Bayesian Model Averaging for Linear Regression Models," *Journal of the American Statistical Association*, 92,179-191.

Rao, J.S., Tibshirani R. (1996), "The out-of-bootstrap method for model averaging and selection," *Technical Report*, Department of Statistics, University of Toronto.

Ravdin P.M., Siminoff, L.A., Davis, G.J., Mercer, M.B., Hewlett, J., Gerson, N., and Parker, H.L. (2001), "Computer program to assist in making decisions about adjuvant therapy for women with early breast cancer," *Journal of Clinical Oncology*, 19(4), 980-991.

Tibshirani R. (1996), "Regression Shrinkage and Selection via the Lasso," *Journal of the Royal Statistical Society, Series B*, 58(1), 267-288.

Xu, L., Jordan, M.I., and Hinton, G.E. (1995), "An alternative model for mixtures of experts," in *Advances in Neural Information Processing Systems (NIPS) 7*, Tesauro, G., Touretzky, D.S., and Leen, T.K. (Eds.), Cambridge, MA: MIT Press, 633-640.

Xu, X., Lu, P., MacEachern, S.N. and Xu, R. (2011). "Calibrated Bayes Factors for Model Comparison and Prediction," *Technical Report No. 855*, Department of Statistics, The Ohio State University.

Yang, Y. (2001), "Adaptive Regression by Mixing," *Journal of the American Statistical Association*, 96, 574-588.

Yu, Q., MacEachern, S.N., and Peruggia, M. (2011), "Bayesian Synthesis: Combining subjective analyses, with an application to ozone data," *Annals of Applied Statistics*,5(2B), 1678-1698.

Wald, A. (1947). *Sequential Analysis*, New York: John Wiley and Sons.

Acknowledgments

This work is supported by the National Science Foundation under award numbers SES-1024709, DMS-1007682 and DMS-1209194. The authors wish to thank the reviewers for their very constructive comments and suggestions.