Published in final edited form as:
J Stat Comput Simul. 2015 ; 85(6): 1090–1101. doi:10.1080/00949655.2013.861839.

Model-Averaged ℓ_1 Regularization using Markov Chain Monte Carlo Model Composition

Chris Fraley[1] and **Daniel Percival**[2]

[1]Insilicos and University of Washington, Seattle, WA / cfraley@insilicos.com

[2]Department of Statistics, Carnegie Mellon University / dancsi@gmail.com

Abstract

Bayesian Model Averaging (BMA) is an effective technique for addressing model uncertainty in variable selection problems. However, current BMA approaches have computational difficulty dealing with data in which there are many more measurements (variables) than samples. This paper presents a method for combining ℓ_1 regularization and Markov chain Monte Carlo model composition techniques for BMA. By treating the ℓ_1 regularization path as a model space, we propose a method to resolve the model uncertainty issues arising in model averaging from solution path point selection. We show that this method is computationally and empirically effective for regression and classification in high-dimensional datasets. We apply our technique in simulations, as well as to some applications that arise in genomics.

Keywords

model averaging; Markov chains; model composition; MCMCMC; ℓ_1 regularization; lasso; high-dimensional; variable selection

1 Introduction

Variable selection now plays a central role in statistical analysis. In particular, modern data applications increasingly involve "wide" data sets ($p > n$ in statistical terminology). Variable selection and dimension reduction are critical in the analysis of such applications. In both regression and classification problems, building models using only a few variables often yields better interpretive and predictive results. For example, in microarray gene expression data there are typically thousands of candidate predictor genes and only a handful of samples. In such a setting, dimension reduction is necessary for any analysis to proceed. Moreover, there is an interest in identifying small numbers of predictor variables that may serve as biomarkers for diagnostic tests and development of therapies.

Modeling techniques for variable selection and sparse modeling typically ignore the issue of model uncertainty. Often, an analyst performs variable selection by simply applying an appropriate technique to choose in advance a subset of the many candidate variables. The analyst next fits a model using these variables, as if this collection of variables comprised the true model. However, this process ignores a critical issue. Once a set of variables is chosen, how certain are we that this is the correct set? Could another set of variables appear

to model the data as well or better? These questions are at the heart of model uncertainty in variable selection.

A general approach to take model uncertainty into account is to, instead of picking a single "final" model, combine many models together, resulting in an ensemble model. Bayesian model averaging (BMA) takes this general approach and seeks to address model uncertainty by taking a weighted average over a class of models under consideration (see Hoeting et al. 1999). The general BMA procedure begins with a set of potential models, called a model space. Using the available data, BMA estimates quantities of interest via a weighted average taken over the elements of the model space. For practical applications to problems where variable selection is necessary, BMA presents two main challenges. First, a complete model space consisting of all subsets of predictors is usually computationally impractical, even for datasets of modest dimension. Second, exact calculation of the weighted average is often intractable. Both of these problems necessitate approximation methods.

Hoeting et al. (1999) identify two approaches to address these challenges. The first approach (Volinsky et al. 1997) uses the 'leaps-and-bounds' algorithm (Furnival and Wilson 1974) to obtain a set of candidate models. The second approach uses Markov chain Monte Carlo model composition (MCMCMC) to directly approximate the posterior distribution (Madigan and York 1995). For BMA, the 'leaps-and-bounds' approach was extended iteratively by Yeung et al. (2005, 2012) to apply to 'wide' data sets, in which there are many more measurements or features than samples, as is common in bioinformatics applications. However, this approach is computationally slow for data sets where p is very large. Fraley and Seligman (2010) replace the models obtained by 'leaps-and-bounds' with those defined by the ℓ_1 regularization path, yielding a method suitable for wide as well as narrow data sets. In this paper we treat the entire ℓ_1 regularization path as a model space for MCMCMC, and develop a combination technique of regularization and model averaging in the following sections with the aim of resolving the model uncertainty issues arising from path point selection.

Bayesian approaches to variable and model selection have been developed and applied with some success to high dimensional data (Brown et al. 2002, Savitsky et al. 2011). Bayesian approaches to lasso have also been developed (Park and Casella 2008, Hans 2009, Hans 2010), but have not yet been sucessfully extended to high dimensional data.

Aggregation methods are another class of techniques that take the ensemble approach to addressing modeling uncertainty. Aggregation procedures offer flexible ways to combine many linear models into a single estimator (see, e.g. Rigolett and Tsybakov 2011, Rigolett 2012). These methods have significant theoretical support, including minimax optimal rates over many important classes of target functions (Yang 2004, Rigolett and Tsybakov 2011). The latter focused on sparse modeling with aggregation, including an MCMCMC-like algorithm to constructing the aggregation estimator. This algorithm takes the form of a random walk over the binary hypercube of the 2^p all-subsets models. Since this approach does not approximate the model space, it is computationally unsuitable for wide data sets.

In this paper, we explore a simple approximation for the model space for use in BMA for regression and classification problems involving high dimensional, wide data sets. Our technique is based on the lasso (Tibshirani 1996), a now ubiquitous technique for variable selection and sparse modeling. We use the solution path of the lasso, parameterized by the tuning parameter λ, as a model space for BMA. This combination of ideas leads to a simple and effective method for addressing model uncertainty in variable selection in settings where it is most critical.

2 Model Averaging and Regularization Paths

Our general approach involves combining two analytical methods. In the sequel, we review the basic concepts behind the two methods: the lasso and Bayesian model averaging. As we will show, we can combine these two methods advantageously, with each lending its strength to combat the other's weakness. The lasso is a useful tool for high dimensional regression and variable selection, but its typical use suffers from model uncertainty issues. Bayesian model averaging is designed to address model uncertainty, but is hard to apply in high dimensional settings. The lasso in turn offers a natural approximation to the complete "all-subsets" model space in high dimensions, allowing a BMA analysis to proceed.

2.1 The Lasso

We first review the lasso, a common technique for sparse regression and variable selection in high dimensional problems (Tibshirani 1996). We begin with notation. Suppose we have n pairs data points measured on p variables and corresponding univariate response values $\{y_i, x_i\}_{i=1}^n$. Let x_{ij} denote the measurement of the ith data point on the jth variable. Then, we define the ℓ_1 regularized estimator as:

$$\hat{\beta}(\lambda) = arg\,min \sum_{i=1}^{n} L(y_i, x_i, \beta) + \lambda \sum_{j=1}^{p} |\beta_j|.$$

In the above, β is a p-dimensional parameter vector, and $L(y_i, x_i, \beta)$ is a loss function, typically restricted to be convex to allow for numerical optimization. This loss function measures the goodness of fit of the model, parameterized by β, to the data. For example, under square error loss, we have the familiar lasso estimator:

$$\hat{\beta}(\lambda) = \sum_{i=1}^{n} \left(y_i - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 + \lambda \sum_{j=1}^{p} |\beta_j|.$$

Finally, and most importantly for our combination of ℓ_1 regularization and the lasso, $\lambda \geq 0$ is a tuning parameter controlling the balance between the goodness of fit represented in the loss function and the second regularization term. As λ increases from 0, the ℓ_1 regularization term becomes more important. The key property of ℓ_1 regularization is that it encourages many entries of β to be exactly zero, giving the lasso its appealing variable selection property. Notice that for all values of λ greater than some λ_{max}, $\hat{\beta}(\lambda) = 0$ — the all zero or

null model solution. For all values of λ between zero and λ_{max}, $\hat{\beta}(\lambda)$ will take on non-trivial values. In turn, for each $\lambda \in [0, \lambda_{max}]$, the parameter $\hat{\beta}(\lambda)$ defines a model. We refer to this set of models, the solutions viewed as a function of λ, as the solution path of the ℓ_1 regularized method.

2.2 Bayesian Model Averaging

We now give a brief description of Bayesian Model Averaging (BMA), which is the basic framework of our technique. In the following section, we describe how we combine the ℓ_1 regularization and BMA by using the solution path of an ℓ_1 regularized method as the model space for BMA. Bayesian model averaging is a technique for combining the information contained in many models of data. The basic elements of this problem are first the data, denoted *D*, and a set of models of these data $\mathscr{M}$. Following a Bayesian approach, we define a prior distribution over the model space **P**(*M*), and a likelihood function of the data given a model $M \in \mathscr{M}$:$\mathbf{P}(D|M)$, which is the integrated likelihood of the model *M*:

$$\mathbf{P}(D|M) = \int \mathbf{P}(D|\beta, M)\, \mathbf{P}(\beta|M)\, d\beta.$$

In the above, β represents the vector of parameters associated with the model *M*. From these, we calculate the posterior distribution over the model space, given the data: **P**(*M*|*D*). We may use this posterior distribution to calculate the posterior probabilities of quantities of interest Δ via the weighted average:

$$\mathbf{P}(\Delta|D) = \sum_{M \in \mathscr{M}} \mathbf{P}(\Delta|M, D)\, \mathbf{P}(M|D).$$

For example, in a linear regression setting, we might take $\mathscr{M}$ to be the collection of linear models using all subsets of the available candidate predictors. For a given *M*, which represents a linear model restricted to a particular subset of the candidate variables. Thus, the parameter associated with *M* would be the vector of linear coefficients β, with nonzero entries appropriately restricted. Δ could be, for example, a single entry of β, or the probability that this entry equals zero.

In practice, we can calculate the posterior probability **P**(*M*|*D*) using a technique called Markov Chain Monte Carlo Model Composition (MCMCMC, or MC^3). We do this by constructing a Markov chain over the model space $\mathscr{M}$ with stationary distribution equal to the posterior distribution over the model space. Indexing the chain as {*M*(*t*)}, *t* = 1, 2...*T*, we can estimate Δ by taking the following average:

$$\hat{\Delta} = \frac{1}{T} \sum_{t=1}^{T} g(M(t)),$$

where *g*(*M*(*t*)) calculates the quantity Δ for the model *M*(*t*). This quantity will converge almost surely by the law of large numbers to the true value of Δ.

Such a chain with the desired equilibrium property can be constructed via the Metropolis-Hastings (M-H) algorithm. Consider a model $M \in \mathscr{M}$. At each step in the M-H algorithm, we must "propose" a new model $M' \in \mathscr{M}$ via a proposal distribution **P**(*M*′|*M*). One way to construct such a distribution is to first define a neighborhood around *M*, and then propose a new model from this neighborhood. For example, in the case of linear regression, the set of models using all possible subsets of a set of candidate covariates can be considered the model space. Given a particular model, all models with one additional or fewer covariates included could be considered to be the neighborhood of that model in this model space (cf. to the algorithm in Chapter 7 of Rigolett and Tsybakov 2011).

Once we have constructed our proposal distribution, and drawn from it, this new proposal is then "accepted" with probability:

$$\mathbf{P}\left(\text{Accept} \quad \text{M'}\right) = min\left(1, \frac{\mathbf{P}\left(M'|M\right)\mathbf{P}\left(M'|D\right)\mathbf{P}\left(M'\right)}{\mathbf{P}\left(M|M'\right)\mathbf{P}\left(M|D\right)\mathbf{P}\left(M\right)}\right).$$

Note that if the proposal distribution is symmetric, e.g. **P**(*M*′|*M*) = **P**(*M*|*M*′), then this is simply the ratio of posterior model probabilities. If we accept *M*′, then *M*′ is now considered as *M* for the next iteration, and we repeat the procedure. The equilibrium distribution of this chain is **P**(*M*|*D*).

This approach presents several computational challenges. First, for many problems, the model space $\mathscr{M}$ may be hard to define, or the most intuitive approach may give a model space that is computationally infeasible, e.g. the all subsets model space described above in a high dimensional linear regression setting. Second, the integrated likelihood **P**(*D*|*M*) is difficult to compute for arbitrary models. Numerical approximations are necessary in this case.

3 Model Averaging using Regularization Paths

3.1 Combination Algorithm

We propose to combine MCMCMC and regularization by using the ℓ_1 regularization path as the model space $\mathscr{M}$ for MCMCMC. We show that this model space gives quite natural solutions to the common issues with MCMCMC. Additionally, we suggest that this approach has significant computational advantages over the "leaps-and-bounds" approach for wide data sets.

The ℓ_1 regularization path can be represented by the interval [0, λ_{max}], which is simply the range of possible λ penalization values. Under this representation, λ_{max} represents the intercept only model and 0 represents the unpenalized regression. For example, in linear regression, a point $a \in (0, \lambda_{max})$ yields the following regression coefficient:

$$\hat{\beta}_{\alpha}=argmin\left(\sum_{i=1}^{n}\left(y_i-\sum_{j=1}^{p}\beta_j x_{ij}\right)^2+\alpha\sum_{j=1}^{p}|\beta_j|\right).$$

We can thus write M_a to represent a model from this space at the point $\lambda = a$ along the ℓ_1 regularization path. Taking the regularization path as the model space, the model space can also be represented by the interval [0, λ_{max}]. A similar setup can be used for the "elastic net" regularization technique (Zou and Hastie 2005). Since neighborhoods on the real line are intuitive concepts, this makes this model space framework quite natural.

Based on the above reasoning, we propose the following combination algorithm for applying MCMCMC to the models space $\mathscr{M}$ generated by an ℓ_1 regularization path:

1. Begin by choosing a starting point $a \in [0, \lambda_{max}]$. This may be done randomly, or deterministically.
2. Propose a new a'(discussed below).
3. Calculate $\mathbf{P}\left(\text{Accept} \quad M_{\alpha'}\right)=min\left(1,\frac{\mathbf{P}\left(M'|M\right)\mathbf{P}\left(M'|D\right)\mathbf{P}\left(M'\right)}{\mathbf{P}\left(M|M'\right)\mathbf{P}(M|D)\mathbf{P}(M)}\right)$
4. Draw a single $U(0, 1)$ to determine acceptance. Set $a = a'$ if accepted, keep $a = a$ otherwise.
5. Repeat steps 2 - 4 T times.

The MCMCMC method described was implemented for elastic net paths for both normal linear regression and for binomial family generalized linear models. The elastic net models were fit using the R package glmnet (Friedman et al. 2008). Friedman et al. 2010 give a technical discussion of the underlying computational algorithm. Note that ℓ_1 regularization is a special case of the elastic net.

The approximation a orded by using the model space defined by the ℓ_1 regularization path has great computational benefits over related methods. For example, in their study of sparse aggregation, Rigolett and Tsybakov (2011) propose a related algorithm for model space search. In each proposal step, the algorithm proposes to add or drop a single variable from the model. This makes the algorithm a random walk over the model space generated from all-subsets regression. This approach thus scales by an additional factor of the number of variables, making it inefficient for high dimensional settings.

3.2 Algorithmic Issues

3.2.1 Choosing a proposal distribution P(*M* ′|*M*)—In the second step, the proposal distribution must be carefully chosen to ensure that the resulting Markov chain has good properties. Three properties of great concern are the following. (1) The first concern is the acceptance rate: how often a proposed model (a) is accepted. An acceptance rate that is near zero or one indicates that the chain is not mixing properly. (2) Second, the auto correlation of the chain is a concern: high auto correlation is bad since it is indicative of the chain being

stuck in a certain spot or small loop. (3) We are finally concerned with the chain becoming stuck at either end of the interval [0, λ_{max}]

Using a proposal distribution which is either one of a $U(\alpha - \gamma, \alpha + \gamma)$ or a $N(\alpha, \gamma)$ can produce good results with respect to concerns (1) and (2), for some suitable $\gamma > 0$. However, concern (3) is not fully addressed, since this scheme has the potential to become trapped at the ends of the interval representing the regularization path. When the chain is near the end, a significant part of the density of either proposal distribution lies outside of the interval.

There are three ways to deal with this problem. First, a straightforward solution to this problem is that if a point outside of the interval is chosen, we simply map it to the closest endpoint. However, this creates an attracting effect where the chain tends to get stuck at the ends of the interval. Second, the distributions could be truncated to exclude any support outside of the model space interval. This solution produces the opposite effect: the endpoints now are repelling. A third solution is to re-parametrize the model space to be constraint free. This can be accomplished by a transformation such as a logit transform. However, there remain issues with the endpoints of the interval, which are distorted to occupy large areas of the real line by the transform, again creating an attracting effect.

Consequently, none of these solutions are completely satisfying. However, we have found that in practice, a uniform proposal distribution is the most appealing when combined with a careful choice of the tuning parameter γ. This choice must take into account both the attracting concern and the number of modes in the posterior distribution of the model space. If the posterior is suspected to be unimodal, then a small, relative to the value of λ_{max}, tuning parameter is acceptable, and can be chosen to be small enough to avoid ever reaching one of the endpoints. If the posterior has many modes, then a large tuning parameter can allow the chain to jump between modes easily.

3.2.2 Model Space Prior P(*M*)—The next two important concerns in this algorithm relate to the quantity $\frac{\mathbf{P}\left(M_{\alpha'}|D\right)}{\mathbf{P}(M_\alpha|D)}$, for use in step 3. Note that $\mathbf{P}(M|D) \propto \mathbf{P}(M, D)\mathrm{P}(M)$.The term $\mathbf{P}(M)$ represents a prior on the model space. A simple solution is a flat prior, so that all models are considered a priori equally likely, e.g.: $\mathbf{P}(M) = \mathbf{P}(M')$ for all $M, M' \in \mathscr{M}$. This effectively eliminates these terms from all calculations.

Alternate ideas include expert elicitation and sparsity inducing priors. Hoeting et al. (1999) suggest the following prior, for a model with p candidate predictor variables:

$$\mathbf{P}(M_i) \propto \prod_{j=1}^{p} \pi_j^{\delta_{ij}} (1 - \pi_j)^{1-\delta_{ij}}.$$

Here δ_{ij} equals 1 if variable j is included in model i, and equals zero otherwise. π_j is a prior probability of variable j being included in the model. If we consider each variable equally likely to be included or not: $\pi_j = .5$ for all j, then we return to the flat prior.

Priors over model-space like objects also play a role in aggregation methods. In the linear regression setting, aggregation methods estimate the vector of linear coefficients as a linear combination of the linear coefficients of all models in the model space:

$$\hat{\beta}_{aggregate} = \sum_{M \in \mathscr{M}} w(M)\, \hat{\beta}(M).$$

In the above, $\hat{\beta}(M)$ is the vector of linear coefficients associated with model *M* and the weight *w*(*M*) can be a product of a goodness of fit component and a prior weight. Rigolett and Tsybakov (2011) suggested the following prior for sparse linear regression:

$$\mathbf{P}(M) \propto \left(\frac{|\hat{\beta}(M)|_0}{2 \quad e \quad p} \right)^{|\hat{\beta}(M)|_0},$$

where *e* is the base of the natural logarithm, and $|\cdot|_0$ denotes the ℓ_0 norm, the count of the number of nonzero entries. They showed that this prior has appealing theoretical and practical properties. In particular, it strongly encourages sparse aggregation estimates, a property that extends to the model averaging approach as well.

In our applications, we have no reason to encode strong prior beliefs about particular variables. Since employing an ℓ_1 regularization path as the model space is already a sparsity inducing technique, we avoid sparsity inducing priors. For these two reasons, we use a flat prior in our examples.

3.2.3 Approximating Integrated Likelihood—The integrated likelihood term **P**(*M*, *D*) is a notoriously difficult quantity to compute. In our implementation, we use the BIC approximation. This approximation is often used in BMA applications; Yeung et al. (2005) employ BIC in their iterative BMA scheme.

One possible way to improve upon the BIC approximation is to apply some sort of Monte Carlo integration technique. As noted in Tibshirani (1996), the ℓ_1 penalized regression model corresponds to a Bayesian regression model with independent double exponential priors on each coordinate of β, using the posterior mode as the coefficient estimates. This gives us a basic setup for a Monte Carlo integration. However, in many cases this is a very high dimensional integral.

4 Simulation Study

In this section, we present the results of a simulation study. We adopt the setting from Wu and Lange (2008). Our goal is to compare the out of sample performance and the variable selection properties between a modeling approach which incorporates averaging versus one which does not. We compare our model averaging using the ℓ_1 solution path (L1MC3) to ordinary ℓ_1 regularization (lasso), where we choose a single solution (a single value of λ) from the path (Friedman et al. 2008, 2010), as well as iterative BMA (iBMA) (Yeung et al. 2005, 2012), which uses a model space approximation based on the “leaps and bounds”

approach. We consider two prediction settings: linear regression and classification. In all settings, we generate an $n \times p$ data matrix with independent standard normal entries. We then fix a vector of true coefficients β with only the first five entries set to be nonzero. We generate a response y in two settings: linear regression and classification. In the linear regression setting, we fix $\beta = \{1, 1, 1, 1, 1, 0, \ldots\}$, and in the classification setting we set a somewhat stronger signal: $\beta = \{5, 5, 5, 5, 5, 0, \ldots\}$. In linear regression, we generate y according to:

$$y_i = x_i^T \beta + \epsilon_i,$$

where the ε_i are also independent standard normal random variables. In the classification setting, we draw y according to the logistic model:

$$y_i \sim \text{Bernoulli}\left(\frac{exp\left(x_i^T \beta\right)}{1 + exp\left(x_i^T \beta\right)}\right).$$

In each simulation, we generate an additional test set consisting of 2000 data points and responses. We employ 10-fold cross validation to choose an optimal λ in for the lasso. In our model averaging methods, we ran the MCMCMC algorithm for 1,000 iterations, with the first 250 discarded to allow for convergence. We reduced the autocorrelation of the chain by thinning, taking only every kth iteration of the chain for use in inference. We used lagged autoregression to pick the thinning parameter k as the minimal lag giving an $R^2 < .1$. We use the median model from the estimated posterior. We compare both estimators to oracle out of sample predictions using the true parameter values. This comparison allows us to measure against an optimal (up to noise) prediction method.

In each case, we employ a different numerical measure of out of sample prediction performance. In the linear regression case, we use the mean square test error. In the classification case, we use area under the ROC curve (AUC) as a measure of out of sample classification performance. In addition, we report the number of variables included in the final model, and how many of these variables were “true” selections (nonzero entries in the true parameter β).

Table 1 displays the results of the simulation study. In terms of out of sample performance, we see that the solution path model averaging approach (L1MC3) outperforms the single model approach (lasso), and compares favorably to the out of sample performance using the true parameters. Iterative BMA falls behind both methods in terms of prediction performance. These results holds in both the classification and regression setting. L1MC3 tends to select a few more variables than necessary in both settings, primarily resulting in false positive errors — selections of variables that correspond to zero entries of the true parameter β. False negative errors, non-selections of nonzero entries of the true parameter β, are uncommon. The lasso, in comparison, performs nearly perfect selection in the linear regression settings. The lasso has much worse selection performance in the classification setting, though it primarily displays too many false positive selections. iBMA has more false

positives than L1MC3 in the linear regression setting, and has very poor selection performance in the classification settings, including many more false negative errors. These comparisons are based on the median model reported by BMA. This excludes more sophisticated techniques and approaches that we might consider if we draw upon more information contained in the posterior distribution over the solution path.

4.1 Example: Gene Network Inference

Yeung et al. (2012) applied iterative Bayesian Model Averaging in an application to gene network inference from time-series perturbation experiments in yeast data. They began by formulating the network inference problem as a variable selection problem. The expression level of gene *g* at time *t*, *X*(*g*, *t*), is modeled as linear combination of the expression levels of potential regulator genes at time *t* – 1:

$$X(g,t) = \beta_0 + \sum_{h \in \{\text{potential regulators}\}} \beta_h X(h, t-1) + \epsilon.$$

The data consist of several experiments for which measurements are taken over time, which are combined in the modeling. Typically, the set of regulators is unknown, and instead a much larger set of potential regulators is available. Sparse modeling approaches to high-dimensional data are therefore needed because there could be thousands of potential regulators. Further, such approaches allow the analyst to make conclusions about which genes are linked in the regulatory network. The networkBMA package (Fraley et al. 2012) implements iterative BMA for this application.

We applied L1MC3 to this example, and compared it to lasso and iterative BMA on the 100-gene simulated time series data from the DREAM4 competition (Stolovitsky et al. 2009; Marbach et al. 2009, 2010; Prill et al 2010), for which the underlying network is known. The data consist of 5 different gene networks giving expression values for 100 genes. The networks are sparse, containing 176, 249, 195, 211, 193 edges out of a possible 9900 (there are no self edges). Ten replicates were generated for each of the 5 networks, with expressions measured over 21 time points from 0 to 1000, in increments of 50. A perturbation is applied to approximately 1/3 of the genes at the initial time point, and removed at the halfway point (time 500). The identity of the genes to which the perturbation is applied is not made available, and is not obvious from data visualization.

The results of our analysis are displayed in Tables 2 and Table 3. These two tables give results related to the most important feature of the problem: the efficacy of the network recovery. We used L1MC3 to select a network by thresholding the posterior probability of selection for each network edge at 50%. We see that L1MC3, in most cases, makes fewer false positive errors in most of the datasets when compared to the competitor BMA method. The lasso by comparison identifies many more of the true edges in the network, at the cost of a great deal more false positives. The overall accuracy of the methods is encapsulated with Precision-Recall values in Table 2. The Precision-Recall approach to evaluation was developed in the setting of information retrieval, in which the number of relevant instances is much smaller than the number of possible instances (e.g. Powers 2011). In this case, the

precision is the fraction of pairs identified by the method that are edges in the network, and the recall (or sensitivity) is the fraction of edges in the network that are identified by the method.

4.2 Example: Classifying Gene Expression Data

In this example, we apply L1MC3 to classification of gene expression data. We use data from the DREAM5 competition Systems Genetics sub-challenge (Stolovitsky et al. 2009). The data consist of 28,295 gene expression measurements from soybeans over 200 training and 30 test cases. For each case, we have 941 binary genotype classifications from an experiment on pathenogenesis of soybean data. We treat each of the genotypes as the reponse in a classification problem, giving a total of 941 separate classification problems.

We again compare to iterative BMA (Yeung et al. 2005) and lasso via glmnet (Friedman et al. 2008, 2010). In addition to using all potential variables (gene expression measurements), we also did the analysis with using only top 100 and 1000 variables as our pool of potential predictors, ranked according to the between sum-of-squares / within sum-of-squares criterion. Since iBMA is impractical on the full set of variables, this gave a total of 8 methods for comparison.

These data are very difficult to classify, and none of the methods is successful in all the genotype classification tasks. In 73 of the 941 cases, no method achieved an ROC AUC above .5. Table 4 shows the L1MC3 has the best overall classification performance. We see that L1MC3 has a better ROC AUC using the “top” genes for a majority of the test experiments. Table 5 also shows that both L1MC3 and iterative BMA identify very few variables as having as high probability of being in the model. In contrast, the lasso selects many more variables when used with the “top” sets of genes.

5 Conclusion

We proposed a simple algorithm, L1MC3, which applies Bayesian model averaging via Markov Chain Monte Carlo model composition to ℓ_1 regularization paths, and demonstrated that this technique can give comparable or better results, in terms of prediction performance and sparsity, than competing approaches on data that have many more variables than samples. Our algorithm is based on iterative simulation, and thus requires considerable computational resources, as does iterative BMA for high-dimensional datasets. While standard lasso has fewer parameters set than L1MC3 (which is built on lasso), iterative BMA has more.

There are several areas of potential improvement in this combination algorithm. First, there are issues with the proposal distribution in the Metropolis-Hastings step. The default tuning parameters gave satisfactory performance in our examples, but we cannot rule out the possibility of being trapped near the edges of the regularization path in other cases. These issues might be resolved through adaptive methods for choosing the tuning parameters, applying a transformation to the ℓ_1 path, and/or alternative proposal distributions.

Second, while the BIC approximation for integrated likelihood is fast, it is poor in many cases. BIC approximations are worst when the number of observations is very small compared to the number of covariates, a situation where regularization techniques are most needed. A better alternative for approximating the integrated likelihood would be highly desirable, but this would also be among the most challenging improvements to make. When information on prior probabilities of the parameters is available, it would be possible to replace the flat prior on the variables as well as to incorporate the prior odds into the integrated likelihood approximation (Lo et al. 2012).

Acknowledgments

This work was supported by National Institutes of Health SBIR awards 5R44GM074313-03 and 7R44GM074313-04, and by NIH grant 5R01GM084163. We thank Dr. Ka Yee Yeung for many fruitful discussions on genomics, for making iterative BMA available, and for making us aware of the DREAM competition as a source of benchmark data. We also thank an anonymous referee for comments leading to a number improvements in this article.

References

1. Brown PJ, Vannucci M, Fearn T. Bayes model averaging with selection of regressors. Journal of the Royal Statistical Society, Series B. 2002; 64:519–536.
2. Efron B, Hastie T, Tibshirani R. Least angle regression. The Annals of Statistics. May.2004 32:407–451.
3. Fraley, C.; Seligman, M. Model-averaged ℓ_1 penalized logistic regression. the 41st Symposium on the Interface: Computing Science and Statistics; Seattle, Washington. June 2010;
4. Fraley C, Yeung KY, Raftery AE. networkBMA: Regression-based network inference using BMA. R package distributed through Bioconductor. 2012
5. Friedman J, Hastie T, Tibshirani R. glmnet: Lasso and elastic net regularized generalized linear models. 2008 R language; available through CRAN; revised in 2010.
6. Friedman J, Hastie T, Tibshirani R. Regularization paths for generalized linear models via coordinate descent. Journal of Statistical Software. 2010; 33
7. Furnival GM, Wilson RW. Regression by leaps and bounds. Technometrics. 1974; 16:499–511.
8. Hans C. Bayesian lasso regression. Biometrika. 2009; 103:835–845.
9. Hans C. Model uncertainty and variable selection in Bayesian lasso regression. Statistics and Computing. 2010; 20:221–229.
10. Hoeting JA, Madigan D, Raftery AE, Volinsky CT. Bayesian model averaging: a tutorial. Statistical Science. 1999; 14(4):382–417.
11. Lo K, Raftery AE, Dombek KM, Zhu J, Schadt EE, Bumgarner RE, Yeung KY. Integrating external biological knowledge in the construction of regulatory networks from time-series expression data. BMC Systems Biology. 2012; 6:101. [PubMed: 22898396]
12. Madigan D, York J. Bayesian graphical models for discrete data. International Statistical Review. 1995; 63:215–232.
13. Marbach D, Prill RJ, Schaffter T, Mattiussi C, Floreano D, Stolovitzky G. Revealing strengths and weaknesses of methods for gene network inference. Proceedings of the National Academy of Sciences. 2010; 107:6286–6291.
14. Marbach D, Schaffter T, Mattiussi C, Floreano D. Generating realistic in silico gene networks for performance assessment of reverse engineering methods. Journal of Computational Biology. 2009; 16:229–239. [PubMed: 19183003]
15. Park M-Y, Hastie T. An L_1 regularization path algorithm for generalized linear models. Journal of the Royal Statistical Society, Series B. 2007; 69:659–677.
16. Park T, Casella G. The Bayesian lasso. Journal of the American Statistical Association. 2008; 103:681–686.

17. Powers DMW. Evaluation: From precision, recall and F-measure to ROC, informedness, markedness and correlation. Journal of Machine Learning Technologies. 2011; 2:37–63.

18. Prill RJ, Marbach D, Saez-Rodriguez J, Sorger PK, Alex-opoulos LG, Clarke ND, Altan-Bonnet G. Towards a rigorous assessment of systems biology models: the DREAM3 challenges. PLoS ONE. 2010; 5:e9202. [PubMed: 20186320]

19. Rigollet P. Kullback-Leibler aggregation and misspecified generalized linear models. The Annals of Statistics. 2012; 40:639–665.

20. Rigollet P, Tsybakov A. Exponential screening and optimal rates of sparse estimation. The Annals of Statistics. 2011; 39(2):731–771.

21. Savitsky T, Vannucci M, Sha N. Variable selection for nonpara-metric process priors. Statistical Science. 2011; 26:130–149. [PubMed: 24089585]

22. Stolovitzky, G.; Califano, A.; Prill, R.; Rodriguez, JS. DREAM: Dialogue for Reverse Engineering Assessments and Methods. 2009. http://wiki.c2b2.columbia.edu/dream

23. Tibshirani R. Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society, Series B. 1996; 58:385–395.

24. Volinsky CT, Madigan D, Raftery AE, Kronmal RA. Bayesian model averaging in proportional hazard models: Assessing stroke risk. Journal of the Royal Statistical Society, Series C — Applied Statistics. 1997; 46:433–448.

25. Wu TT, Lange K. Coordinate descent algorithms for lasso penalized regression. Annals of Applied Statistics. 2008; 2(1):224–244.

26. Yang Y. Aggregating regression procedures to improve performance. Bernoulli. 2004; 10(1):25–47.

27. Yeung KY. iterativeBMA: The iterative Bayesian model averaging (BMA) algorithm. 2005 R language; available through Bioconductor; revised in 2010.

28. Yeung KY, Bumgarner RE, Raftery AE. Bayesian model averaging: Development of an improved, multi-class, gene selection tool for microarray data. Bioinformatics. 2005; 21:2394–2402. [PubMed: 15713736]

29. Yeung KY, Dombek KM, Lo K, Mittler JE, Zhu J, Schadt EE, Bumgarner RE, Raftery AE. Construction of regulatory networks using expression time-series data of a genotyped population. Proceedings of the National Academy of Sciences. Nov.2011 108:19436–19441.

30. Zou H, Hastie T. Regularization and variable selection via the elastic net. Journal of the Royal Statistical Society, Series B. 2005; 67:301–320.

Table 1

Simulation Study: comparison of L1MC3 model averaging with iterative BMA (iBMA) and single model Lasso. In terms of out of sample errors, L1MC3 outperforms iBMA and the single model Lasso approach, and compares favorably to the out of sample performance using the true parameters. Moreover, the L1MC3 approach tends to generate fewer false positive (noise) selections than iBMA, and many fewer than lasso in the case of classification.

Linear Regression: n = 200; p=5000; β = {1, 1, 1, 1, 1, 0, 0, ...}

Model	Test Error MSE	Selected Variables	
		True	Noise
true	1.00 ± .03	5	0
L1MC3	1.47 ± 0.14	5 ± 0	2.58 ± 1.83
lasso	2.06 ± 0.48	5 ± 0	.29 ± 1.04
iBMA	5.03 ± .33	5 ± .04	4.58 ± 2.32

Linear Regression: n=500; p=5000; β = {1, 1, 1, 1, 1, 0, 0, ...}

Model	Test Error MSE	Selected Variables	
		True	Noise
true	1.00 ± .03	5	0
L1MC3	1.20 ± 0.06	5 ± .00	.43 ± .70
lasso	1.27 ± 0.09	5 ± .00	.18 ± .67
iBMA	5.09 ± .04	5 ± .00	2.21 ± 1.68

Classification: n=200; p=5000; β = {5, 5, 5, 5, 5, 0, 0, ...}

Model	Test Error ROC AUC	Selected Variables	
		True	Noise
true	0.99 ± .00	5	0
L1MC3	0.95 ± 0.03	4.96 ± .19	.38 ± .63
lasso	0.94 ± 0.07	4.88 ± .70	57.2 ± 29.8
iBMA	0.72 ± .08	1.51 ± .73	1.17 ± 1.45

Classification: n=500; p=5000; β = {5, 5, 5, 5, 5, 0, 0, ...}

Model	Test Error ROC AUC	Selected Variables	
		True	Noise
true	0.99 ± .00	5	0
L1MC3	0.98 ± .00	5 ± .00	.00 ± 0.04
lasso	0.87 ± .19	3.96 ± 1.95	112.7 ± 74.8
iBMA	0.68 ± .06	1.24 ± .54	5.72 ± 1.63

Table 2

Precision-Recall values for the L1MC3, lasso, and iterative BMA (iBMA) approaches to network inference via variable selection for the five DREAM4 100-gene time series network datasets. L1MC3 shows the best performance in all but one case.

	# 1	# 2	# 3	# 4	# 5
L1MC3	.129	.061	.152	.102	.111
lasso	.108	.057	.130	.102	.115
iBMA	.045	.039	.040	.046	.040

Table 3

TP / FP for the L1MC3, lasso, and iterative BMA (iBMA) approaches to network inference via variable selection the five DREAM4 100-gene time series network datasets, using all 10 replicates in this case. TP is the number of edges correctly identified and FP is the number of gene pairs mistakenly identified as edge. The values are given for a probability threshhold of 50%. The L1MC3 and iBMA approaches produce much sparser networks than lasso. While L1MC3 identifies somewhat fewer true network edges than iBMA, it also makes considerably fewer mistaken identifications.

	# 1	# 2	# 3	# 4	# 5
true	176	249	195	211	193
L1MC3	54 / 319	40 / 378	57 / 310	60 / 349	53 / 369
lasso	135 / 5504	172 / 5787	140 / 1312	155 / 5526	150 / 5682
iBMA	59 / 581	51 / 603	58 / 137	61 / 633	54 / 619

Table 4

Percentages of cases in which the AUC for the method listed vertically to the left of the table is greater than that of the methods listed horizontally at the top of the table. Results are for the DREAM 5 data. Cases in which both AUCs are less than or equal to .5 are excluded. L1MC3 with 100 and 1000 variables give the best overall classification performance on these data.

	L1MC3			lasso			iterative BMA	
	100	1000	all	100	1000	all	100	1000
L1MC3/100		.38	.67	.69	.61	.85	.70	.59
L1MC3/1000	.62		.72	.62	.72	.83	.62	.64
L1MC3/all	.32	.28		.42	.48	.84	.44	.40
lasso/100	.31	.38	.57		.55	.81	.48	.49
lasso/1000	.39	.28	.51	.45		.77	.44	.42
lasso/all	.15	.17	.16	.19	.23		.19	.20
iBMA/100	.30	.38	.55	.51	.56	.80		.49
iBMA/1000	.41	.36	.60	.51	.58	.80	.51	

Table 5

Mean and standard deviation of the number of variables selected for each of the methods across the training experiments for the DREAM 5 data. In the case of L1MC3 and iterative BMA, the number is given for variables with the probability .5 of having a nonzero coefficient. Iterative BMA was not included for all variables because the length of time required to complete the computations was prohibitive.

	100	1000	all
L1MC3	2.4 ± 3.4	2.3 ± .9	2.5 ± .9
lasso	38.2 ± 10.1	70.2 ± 38.8	5.8 ± 12.1
iterative BMA	5.1 ± 1.4	5.4 ± 1.3	—