Bayesian Analysis of High Dimensional Classification

Subhadeep Mukhopadhyay and Faming Liang

Citation: AIP Conference Proceedings **1193**, 243 (2009); doi: 10.1063/1.3275621

View online: http://dx.doi.org/10.1063/1.3275621

View Table of Contents: http://scitation.aip.org/content/aip/proceeding/aipcp/1193?ver=pdfcov

Published by the AIP Publishing

Bayesian Analysis of High Dimensional Classification

Subhadeep Mukhopadhyay and Faming Liang

Department of Statistics, Texas A&M University

Abstract.
Modern data mining and bioinformatics have presented an important playground for statistical learning techniques, where the number of input variables is possibly much larger than the sample size of the training data. In supervised learning, logistic regression or probit regression can be used to model a binary output and form perceptron classification rules based on Bayesian inference. In these cases , there is a lot of interest in searching for sparse model in High Dimensional regression(/classification) setup. we first discuss two common challenges for analyzing high dimensional data. The first one is the curse of dimensionality. The complexity of many existing algorithms scale exponentially with the dimensionality of the space and by virtue of that algorithms soon become computationally intractable and therefore inapplicable in many real applications. secondly, multicollinearities among the predictors which severely slowdown the algorithm. In order to make Bayesian analysis operational in high dimension we propose a novel *'Hierarchical stochastic approximation monte carlo algorithm'* (HSAMC), which overcomes the curse of dimensionality, multicollinearity of predictors in high dimension and also it possesses the self-adjusting mechanism to avoid the local minima separated by high energy barriers. Models and methods are illustrated by simulation inspired from from the feild of genomics . Numerical results indicate that HSAMC can work as a general model selection sampler in high dimensional complex model space.

Keywords: Hierarchical modeling; Markov chain Monte carlo; Model selection; Stochastic approximation ;
PACS: 02.70.Tt,05.50.+q

INTRODUCTION

The problem of interest here is to predict y a $\{0,1\}$ response based on x, a vector of predictors of dimension p, which is possibly much larger than the sample size n. We have $(\mathbf{X}_i, \mathrm{Y}_i)$ (i=1,2,...,n) independent replication of a random vector $(\mathbf{X}, \mathrm{Y})$. One is often interested in modeling the relation between y and **x**, selecting components of **x** that are most relevant to y, and predicting y using selected information from **x**. This raises modeling and computational challenges as the number of candidate predictor variables increases.

MCMC algorithms designed to explore the posterior distribution over regression model spaces (e.g., George and McCulloch 1993, 1997; Green 1995; Raftery, Madigan, and Hoeting 1997) rely on Gibbs sampling or on the Metropolis-Hastings algorithm but are increasingly ineffective in higher dimensions due to slow convergence. One reason is multimodality: on the energy landscape of the posterior, there are many local minima that are separated by high barriers. In simulation, the Markov chain may get stuck in a local energy minimum indefinitely, rendering the simulation ineffective.To alleviate this difficulty, many techniques have been proposed which can be categorized

CP1193, *Bayesian Inference and Maximum Entropy Methods in Science and Engineering: 29th International Workshop*
edited by P. Goggans and C.-Y. Chan

into two class broadly. The first idea is the use of auxiliary variables. Swendsen-Wang algorithm , simulated tempering, parallel tempering, evolutionary Monte Carlo, dynamic weighting, multicanonical weighting ,belong to this class. The second idea is the use of past samples. The multicanonical algorithm [1] is apparently the first work in this direction.Related works include the Wang-Landau (WL) algorithm [13], and the generalized Wang-Landau (GWL) algorithm (Liang 2004, 2005).Among the algorithms described here, the WL algorithm has received much recent attention in physics.

However, for many problems the slow convergence is not due to the multimodality, but the curse of dimensionality, that is, the number of samples increase exponentially with dimension to maintain a given level of accuracy. For example, the witch's hat distribution (Matthews 1993) has only one single mode, but the convergence time of the Gibbs sampler on it increases exponentially with dimension. For this kind of problems, although the difficulty of slow convergence can be resolved by the tempering or the importance weights based algorithms to some extent, the curse of dimensionality cannot be eliminated significantly, as these samplers always work in the same sample space.

In this paper, we provide a different treatment for the problem based on stochastic approximation Monte Carlo (SAMC) algorithm [10] and Hierarchical clustering with an aim to eliminate the curse of dimensionality suffered by the conventional MCMC methods in High dimensional variable selection problem.

MAIN FOCUS OF THE ARTICLE

This article presents how the stochastic approximation Monte Carlo (SAMC) [10] algorithm can be used for high dimensional classification problem, including variable selection, classifier building and class prediction.

A Brief Review for the SAMC Algorithm

Before describing the HSAMC algorithm, we first give a brief description of SAMC. The basic idea of SAMC stems from the Wang-Landau algorithm and can be briefly explained as follows. Let

$$f(x) = c\psi(x), \quad x \in \chi \tag{1}$$

denote the target probability density/mass function, where χ is the sample space and c is an unknown constant. Let $E_1, \ldots E_m$ denote a partition of χ, and let $\omega_i = \int_{E_i} \psi(x)dx$, for $i = 1, \ldots, m$. SAMC seeks to draw samples from the trial distribution

$$f_\omega(x) \propto \sum_{i=1}^{n} \frac{\pi_i \psi(x)}{\omega_i} I(x \in E_i), \tag{2}$$

where $I(\cdot)$ is an indicator function, π_i's are pre-specified constants such that $\pi_i > 0$ for all i and $\sum_{i=1}^{m} \pi_i = 1$. In Liang et al. (2007), $\pi = (\pi_1, \ldots, \pi_m)$ is called the desired sampling distribution of the subregions. If $\omega = (\omega_1, \ldots, \omega_m)$ can be well estimated, sampling from $f_\omega(x)$ will result in a "random walk" in the space of subregions (by regarding each subregion as a “point”) with each subregion being sampled with a frequency proportional to π_i. Hence, the local trap problem can be overcome essentially, provided that the sample space is partitioned appropriately. SAMC provides a systematic way to estimate $\omega_1, \dots, \omega_m$ under the framework of the stochastic approximation method [12]. The SAMC algorithm iterates between the following two steps.

SAMC Algorithm

(S1) Simulate a sample $\mathbf{X}_t$ by a single MH update with the invariant distribution

$$f_{\theta_t}(x) \propto \sum_{i=1}^{m} \frac{\psi(x)}{e^{\theta_{ti}}} \mathbf{I}(x \in \mathbf{E}_i) \tag{3}$$

(S2) set

$$\theta^* = \theta^{(t)} + \gamma_{t+1}(e_{t+1} - \pi) \tag{4}$$

where $e_t = e_{(t,1)}, \dots e_{(t,m)}$ and $e_{(t,i)} = 1$, if $x_t \in \mathbf{E}_i$ If $\theta \in \Theta$, set $\theta^{(t+1)} = \theta^*$, otherwise, set $\theta^{(t+1)} = \theta^{(t)} + c$, where $c = (c, \dots, c)$ is chosen such that $\theta^{(t)} + c \in \Theta$. Where θ_{ti} denote the working estimate of $\log \omega_i/\pi_i$, obtained at iteration t, $\theta_t = (\theta_{t1}, \dots \theta_{tm})$

A remarkable feature of SAMC is its self-adjusting mechanism. If a subregion is visited, θ_t will be updated accordingly such that this subregion has a smaller probability to be revisited in the next iteration. For more detail see Liang et.al 2007. However, this mechanism has not yet reached its maximum efficiency because of the presence of the Dimensionality curse in High dimension. In the next section we will explain how to get around to this problem in an elegant way.

Hierarchical SAMC algorithm for Variable Selection

In theory, SAMC is able to find the global energy minima but we still have the problem of “curse of dimensionality”. The ’trick’ we use to overcome this limitation, is to hierarchically organize subsets of huge number of variables into groups and move through the cluster to explore the whole space. This is the way we buildup the ladder (Linag 2003) to approximate the original high dimensional system by a system with a reduced dimension, the reduced system is again approximated by a system with a further reduced dimension, until one reaches a system of a manageable dimension, that is, the corresponding system is able to be sampled from easily by a local updating algorithm of SAMC. The idea is to use the information provided by the low-dimensional hierarchical regression models as a clue for the simulation from high-dimensional posterior, and, thus, to eliminate the curse of dimensionality significantly.

Hierarchical methods rely on a distance function to measure the “similarity” between clusters. Their computational complexity is usually $O(n^2)$, $n =$ number of sample. As we will consider mainly “small n large p problem”, we will mostly concentrate on this general method as clustering. Much more sophisticated method on hierarchical methods line BIRCH [14] and CURE [6] attempt to address the scalability problem.

Let $X_1,\ldots,X_p$ be the total number of possible predictor variables, where $X_1 = (X_{11},\ldots,X_{n1})'$ and n is the number of observations. Define the following notations:

- $\mathscr{L}_k$: the set of non-overlapping k clusters S_{k1}, …, S_{kk}.
- $M_k^{(t)}$: a model with k predictors and obtained at iteration t.
- $V_k^{(t)}$: the set of variables included in the model $M_k^{(t)}$.
- $\beta_k^{(t)}$: the set of regression coefficients of the variables included in the model $M_k^{(t)}$.
- $\xi_{k0}^{(t)}$: the number of vacant clusters, i.e., $\#\{j : |V_k^{(t)} \cap S_{kj}| = 0, j = 1,\ldots,k\}$.
- $\xi_{k1}^{(t)}$: the number of clusters that include a single variable of the model, i.e., $\#\{j : |V_k^{(t)} \cap S_{kj}| = 1, j = 1,\ldots,k\}$.
- $\tilde{\xi}_{k0}^{(t)}$: $\#\{j : |V_k^{(t)} \cap S_{k+1,j}| = 0, j = 1,\ldots,k+1\}$.
- $\tilde{\xi}_{k1}^{(t)}$: $\#\{j : |V_k^{(t)} \cap S_{k+1,j}| = 1, j = 1,\ldots,k+1\}$.
- $q_{k,A}$, $q_{k,D}$, $q_{k,E1}$, $q_{k,E2}$, $q_{k,C}$: denote, respectively, the probabilities to add a variable, delete a variable, exchange a variable within the cluster, exchange a variable between the cluster, and update the regression coefficient.

HSAMC Algorithm

(a) (Hierarchical clustering) Cluster the variables in a hierarchical clustering method, and identify $\mathscr{L}_1$, …, $\mathscr{L}_n$.

(b) At each level k, the model is updated in the following way depending on its configuration.

(b.0) (proposal setting) If $\xi_{k1}^{(t)} = 0$, i.e., no a cluster containing a single variable, then we set $q_{k,A} = 1/4$, $q_{k,D} = 0$, $q_{k,E1} = 3/16$, $q_{k,E2} = 3/16$, and $q_{k,C} = 3/8$. Otherwise, we set $q_{k,A} = 1/4$, $q_{k,D} = 1/4$, $q_{k,E1} = 1/8$, $q_{k,E2} = 1/8$, and $q_{k,C} = 1/4$.

(b.1) (extrapolation) This is to add a variable to the current model.

- Extrapolate $\mathscr{L}_k$ to $\mathscr{L}_{k+1}$.
- Identify the clusters satisfying the condition $|V_k^{(t)} \cap S_{k+1,j}| = 0$ and the clusters satisfying the condition $|V_k^{(t)} \cap S_{k+1,j}| = 1$. Denote the clusters by $S_{k+1,1}^{(0)},\ldots,S_{k+1,\tilde{\xi}_{k0}^{(t)}}^{(0)}$ and $S_{k+1,1}^{(1)},\ldots,S_{k+1,\tilde{\xi}_{k1}^{(t)}}^{(1)}$, respectively.
- Draw a random number i uniformly from the set $\{1,\ldots,\tilde{\xi}_{k0}^{(t)}\}$, select a variable randomly from the set $S_{k+1,i}^{(0)}$, and draw a normal random variable β^* from $N(0,\tau^2)$.
- Accept the new model, denoted by M_{k+1}^* with the probability $\min(1,r_a)$, where

$$r_a = \frac{e^{\theta_k^{(t)}}}{e^{\theta_{k+1}^{(t)}}} \frac{f(M_{k+1}^*|D)}{f(M_K^{(t)}|D)} \frac{q_{k+1,D}}{q_{k,A}} \frac{\tilde{\xi}_{k0}^{(t)}|S_{k+1,i}^{(0)}|}{\xi_{k1}^{(t)}+1} \frac{1}{\frac{1}{\tau}\phi(\beta^*/\tau)}$$

(b.2) (projection) This is to remove a variable from the current model.

- Project $\mathscr{L}_k$ to $\mathscr{L}_{k-1}$.

- Identify the clusters satisfying the condition $|V_k^{(t)} \cap S_{k,j}| = 0$ and the clusters satisfying the condition $|V_k^{(t)} \cap S_{k,j}| = 1$. Denote the clusters by $S_{k,1}^{(0)}, \ldots, S_{k,\xi_{k0}^{(t)}}^{(0)}$ and $S_{k,1}^{(1)}, \ldots, S_{k,\xi_{k1}^{(t)}}^{(1)}$, respectively.
- Draw a random number i uniformly from the set $\{1, \ldots, \xi_{k1}^{(t)}\}$.
- Accept the new model, denoted by M_{k-1}^* with the probability $\min(1, r_d)$, where

$$r_d = \frac{e^{\theta_k^{(t)}}}{e^{\theta_{k-1}^{(t)}}} \frac{f(M_{k-1}^*|D)}{f(M_K^{(t)}|D)} \frac{q_{k-1,A}}{q_{k,D}} \frac{\xi_{k1}^{(t)}}{(\xi_{k0}^{(t)}+1).|S_{k,i}^{(1)}|} \frac{1}{\tau} \phi(\beta_d^*/\tau)$$

where β_d^* denotes the regression coefficient of the variable to be removed.

(b.3) (Exchange within cluster) This is to exchange a variable within the same cluster.

- Randomly select a variable, say, the ith variable of the model.
- Identify the cluster that variable i belongs to, say, cluster S_{kj}.
- Accept the new model, denoted by M_k^* with the probability $\min(1, r_{e_1})$, where

$$r_{e_1} = \frac{f(M_k^*|D)}{f(M_K^{(t)}|D)}.$$

(b.4) (Exchange between cluster) This is to exchange a variable with another from a different cluster.

- Randomly select a variable, say, the ith variable of the model, and identify the cluster it belongs to, denoting the cluster by S_{ki}.
- Randomly select a cluster, say S_{kj}, other than S_{ki}.
- Randomly select a spare variable from S_{kj}. Let $|S_{kj}|^*$ denote the number of spare variables in S_{kj}, and let $|S_{ki}|^*$ denote the number of spare variables in S_{ki}.
- Draw a normal random variable β^* from $N(0, \tau^2)$.
- Accept the new model, denoted by M_k^* with the probability $\min(1, r_{e_2})$, where

$$r_{e_2} = \frac{f(M_k^*|D)}{f(M_K^{(t)}|D)} \frac{|S_{kj}|^*}{(|S_{ki}|^*+1)} \frac{\phi(\beta_d^*/\tau)}{\phi(\beta^*/\tau)},$$

where β_d^* denotes the regression coefficient of the variable to be exchanged.

(b.5) (Coefficient updating) Update β_k by a Metropolis-Hasting move or a Metropolis-within-Gibbs move.

(c) (Weight adjustment) Set

$$\theta^* = \theta^{(t)} + \gamma_{t+1}(e_{t+1} - \pi)$$

If $\theta \in \Theta$, set $\theta^{(t+1)} = \theta^*$, otherwise, set $\theta^{(t+1)} = \theta^{(t)} + c$, where $c = (c, \ldots, c)$ is chosen such that $\theta^{(t)} + c \in \Theta$.

PERFORMANCE OF HSAMC USING SIMULATION STUDY

In this section we report a simulation study based on Binary classification data,which demonstrate the effectiveness of HSAMC.All simulations are conducted using R Software. In this example $(n, p) = (50, 100)$, i.e, we will work in large p small n case and we put gaussian prior over the regression coeffitients to prevent from the overfitting. To show the applicability of our results to identify the underlying true model (Sparse) we have generated the class labels $(1/0)$ from the following model :

$$\log \frac{\Pr(\ Y = 1|\mathbf{X})}{\Pr(\ Y = 0|\mathbf{X})} \ = \ 1.6\mathbf{X}_1 - 4\mathbf{X}_2 + 5.1\mathbf{X}_3 + 1\mathbf{X}_4 - 1.5\mathbf{X}_5 + 2\mathbf{X}_6 \tag{5}$$

So, here the size of the true model is 6. To increase the size of the predictor set, we have added 94 irrelevant rows (variables) and created the data matrix X of order 100×50. Now to demonstrate the effectiveness of HSAMC, we purposefully generated the rows such that we have 8 distinct clusters and with in each cluster we have similar type of values which will create the multicollinearity problem. We select uniform distribution as the desired sampling distribution , that is, $\pi_i = 1/n, i = 1, 2, \ldots n$. The sequence γ_t is constructed with $t_0 = 10$, where $\gamma_t \ = \ \frac{t_0}{(\max t_0, t)}$, and t is the number of iteration. To give a overall description of the data we are using here, we have plotted it in the following Figure(1) :

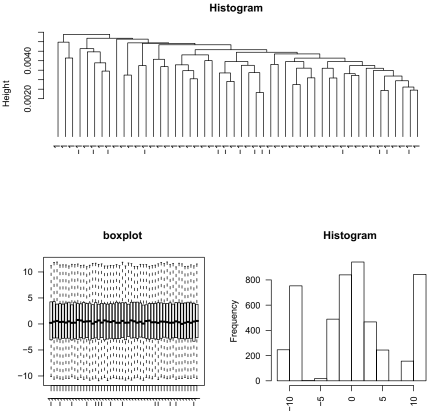

FIGURE 1: The figure shows boxplot, histogram and dendrogram of a hierarchical analysis. Hierarchical clustering is produced using average linkage clustering with a Pearson correlation measure of similarity

We emphasize that, the simulation time spent on low-energy and high-energy regions in SAMC by choosing the desired sampling distribution π can be controled almost exactly, up to the constant . SAMC can go to high-energy regions, but it spends only limited time there to help the system to escape from local energy minima, and also spends time exploring low-energy regions. This smart simulation time distribution scheme makes SAMC potentially more efficient for High Dimensional data, which is again confirmed by the Figure(2)

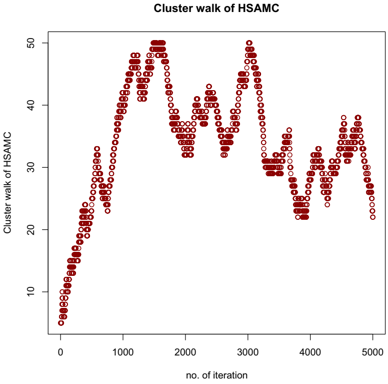

FIGURE 2: This figure shows as a function of iteration how the algorithm moves along with the clusters

In simulations,we can see that SAMC can overcome any difficulties in dimension-jumping moves and provide a full exploration for all models due to it's self-adjusting ability. Where as, the built-in hierarchical clustering helps SAMC to buildup the ladder and thus efficiently explore the model space in the presence of multicollinearity and dimensionality curse.

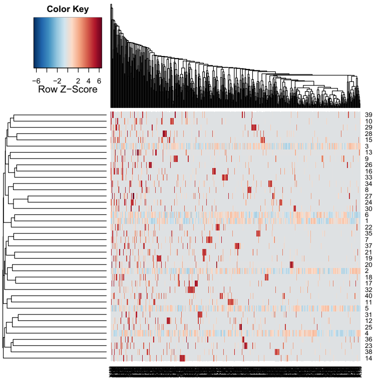

FIGURE 3: Computational Results of HSAMC

HSAMC was run for 5000 iteration and we have collected the last 600 MCMC sample for the first 40 variables only and displayed in the figure(3). The result is clear.There is clearly visible 6 bands where the algorithm moved frequently .The estimated coefficients are $1.27, -3.89, 4.772, 1.112, -1.667, 1.657$.

DISCUSSION

In this article we have introduced a highly efficient way to explore the high dimension and discussed it's application to simultaneous Model selection and estimation problem from a Bayesian point of view, which is beyond the ability of usual MCMC sampling.This paper shows how to improve SAMC for sampling from high-dimensional systems by constructing sequential structures through Hierarchical clustering for eliminating the curse of dimensionality .The HSAMC approach is quite general and, in addition to the binary regression models that we have considered here, can be applied to any generalized linear regression model as long as the marginal likelihood can be evaluated or approximated.

REFERENCES

1. Berg, B. A., and Neuhaus, T. (1991), *"Multicanonical Algorithms for 1st-Order Phase-Transitions,"* Physics Letters B, 267, 249-253.
2. Chris Hans, Adrian Dobra, Mike West (2007) *"Shotgun Stochastic Search for "Large p" Regression"* J. Amer. Statist. Assoc.,102(478): 507-516
3. Gelfand, A. E., and Smith, A. F. M. (1990), *"Sampling-Based Approaches for Calculating Marginal Densities"* Journal of the American Statistical Association,85, 398-409.
4. George, E. I., and McCulloch, R. E. (1993), *"Variable Selection via Gibbs Sampling"*, Journal of the American Statistical Association, 88, 881-889
5. Green, P. J. (1995), *"Reversible-Jump Markov Chain Monte Carlo-Computation and Bayesian Model Determination"* Biometrika, 82, 711-732.
6. Sudipto Guha, Rajeev Rastogi, Kyuscok Shim (2001) *"CURE: An Efficient Clustering Algorithm for Large Databases"* Information Systems ,Vol. 26, Issue 1
7. Lai, T. L(2003) *"Stochastic Approximation"* The Annals of Statistics, 31, 391-406.
8. Liang, F(2003). *"Use of sequential structure in simulation from high dimensional systems"* Physical Review E, 67, 56101-56107
9. Liang, F. (2007). *"Annealing Stochastic Approximation Monte Carlo for Neural Network Training"* Machine Learning 68 201-233.
10. Liang, F., Liu, C. and Carroll, R.J.(2007). *"Stochastic Approximation in Monte Carlo Computation"* J. Amer. Statist. Assoc., 102, 305-320
11. Raftery, A. E., Madigan, D., and Hoeting, J. (1997), *"Bayesian Model Averaging for Linear Regression Models"* Journal of the American Statistical Association, 92, 1197-1208
12. Robbins, H., and Monro, S. (1951), *"A Stochastic Approximation Method",* The Annals of Mathematical Statistics, 22, 400-407.
13. Wang, F., and Landau, D. P. (2001), *"Efficient, Multiple-Range Random-Walk Algorithm to Calculate the Density of States"* Physical Review Letters, 86,2050-2053.
14. Tian Zhang and Raghu Ramakrishnan and Miron Livny (1996) *"BIRCH: An Efficient Data Clustering Method for Very Large Databases"* Data Mining and Knowledge Discovery archive , Vol. 1 ,Issue 2