Evolutionary Monte Carlo Methods for Clustering

Gopi GOSWAMI, Jun S. LIU, and Wing H. WONG

The problem of clustering a group of observations according to some objective function (e.g., K-means clustering, variable selection) or a density (e.g., posterior from a Dirichlet process mixture model prior) can be cast in the framework of Monte Carlo sampling for cluster indicators. We propose a new method called the evolutionary Monte Carlo clustering (EMCC) algorithm, in which three new “crossover moves,” based on swapping and reshuffling subcluster intersections, are proposed. We apply the EMCC algorithm to several clustering problems including Bernoulli clustering, biological sequence motif clustering, BIC based variable selection, and mixture of normals clustering. We compare EMCC's performance both as a sampler and as a stochastic optimizer with Gibbs sampling, “split-merge” Metropolis–Hastings algorithms, K-means clustering, and the MCLUST algorithm.

Key Words: Dirichlet process mixture model; Gibbs sampling; Integrated auto-correlation time; K-means; Metropolis–Hastings algorithm; Model-based clustering; Parallel tempering; Temperature ladder; Variable selection.

1. INTRODUCTION

The problem of clustering a given set of multidimensional objects arises in many different applications such as marketing, speech recognition, text mining, gene expression microarray studies, and biological sequence analysis, to name only a few. At the conceptual level, the main goal of clustering is to partition a set of objects into nonoverlapping “homogeneous” subgroups according to a certain “similarity” or “distance” measure. Some often-used measures include the Euclidean distance, Hamming distance, Pearson correlation, and entropy distance. Some methods, such as hierarchical clustering, do not directly

Gopi Goswami is a Post Doctoral Candidate, and Jun S. Liu is Professor, Department of Statistics, Harvard University, Seventh Floor, Science Center, 01 Oxford Street, Cambridge, MA 02138 (E-mail for correspondence: *goswami@stat.harvard.edu*). Wing H. Wong is Professor, Department of Statistics and Health Research and Policy, Stanford University, Sequoia Hall, 390 Serra Mall, Stanford, CA 94305 (E-mail: *whwong@stanford.edu*). We thank two anonymous referees and the associate editor for providing numerous helpful comments and suggestions. GG and JSL's work was supported by the National Science Foundation grant number DMS0204674, National Institute of Health grant number R01-GM078990, and Chinese National Natural Science Foundation grant number 10228102. WHW's work was supported by the National Science Foundation grant number DMS0505732.

Journal of Computational and Graphical Statistics, Volume 16, Number 4, Pages 1–22
DOI: 10.1198/106186007X255072

partition the set and decide on homogeneous groupings, but provide a tree structure ordering of the objects according to their similarities. Other methods, such as K-means and the self-organizing map, attempt to give the user the best partitioning of the set when provided with a pairwise similarity or distance measure and the total number of desired clusters; Hastie, Tibshirani, and Friedman (2001) provided a good introduction to these methods.

In general, a clustering problem can be posed as a sampling problem from a probability density over the space of all possible clusters. In some cases, such as the Bayesian Dirichlet process mixture models, this density arises naturally (Jensen and Liu 2007). In some other cases, such as the K-means algorithm, which minimizes the "goodness" of a proposed clustering determined by a given distance or similarity measure, one can recast it into a sampling problem using the Boltzmann distribution format (Liang and Wong 2000, 2001), and the simulated annealing framework.

In difficult multimodal high-dimensional problems, the standard Markov chain Monte Carlo (MCMC) techniques such as Gibbs sampling (Gibbs; Gelfand and Smith 1990) and the Metropolis–Hastings (MH; Hastings 1970) algorithm do not work very well. A class of MCMC sampling methods, known to be effective in such situations, capitalize on the "annealing" idea and use multiple parallel runs of MCMC chains, each corresponding to a "heated" version of the target distribution. We call this class of methods *population-based methods*, which include, for example, parallel tempering (PT; Geyer 1991), adaptive directional sampling (ADS; Gilks, Roberts, and George 1994), conjugate gradient Monte Carlo (CGMC; Liu, Liang, and Wong 2000), and evolutionary Monte Carlo (EMC; Liang and Wong 2000, 2001; Goswami and Liu 2007).

The new sampling recipe proposed in this article, the evolutionary Monte Carlo clustering (EMCC) algorithm, is structurally similar to EMC. We introduce three new crossover moves, namely, SCSC:TWO-NEW, SCSC:ONE-NEW, and SCRC in the PT framework. These new EMCC moves enhance the performance of the sampler with respect to dependency among its draws and the ability to escape local modes. As a result, the sampler performs much better than Gibbs sampling, "split-merge" MH algorithm, K-means, and the MCLUST algorithm (Fraley and Raftery 2002).

The article is organized as follows. In Section 2, we discuss the pros and cons of several existing approaches to clustering and motivate the need for the EMCC algorithm. In Section 3, we introduce the EMCC algorithm and its different moves. In Section 4 we briefly review the Dirichlet process mixture model (DPMM) prior and discuss Gibbs sampling from the resulting posterior; the DPMM prior has been used in the examples in this article. In Section 5, we look at applications of the EMCC algorithm to motif clustering, Beta mixture of Bernoulli clustering, Bayesian information criterion (BIC) based variable selection and mixture of multivariate normals clustering, and compare its performance both as a sampler and as a stochastic optimizer with Gibbs sampling, "split-merge" MH algorithm, K-means clustering, and the MCLUST algorithm. Finally, in Section 6 we provide some discussion.

2. A BRIEF OVERVIEW OF CLUSTERING METHODS

Approaches to clustering a set of observations can be roughly divided into two classes, namely, methods that rely on sampling over the space of possible clusters and methods that use optimization techniques instead of sampling.

Some of the methods that fall in the second category are K-means clustering, hierarchical clustering, and the MCLUST algorithm (Fraley and Raftery 2002). K-means performs a greedy search for the best clustering solution by iteratively minimizing an objective function (Section 5.3) and thus often gets stuck in some local mode. This problem is solved partially by restarting the algorithm with many random starting values and then choosing the solution with the minimum objective function value. The statistical software R takes this approach. Another route for avoiding the local minima trap is to start K-means from a hierarchical clustering solution to the problem. The statistical software S-PLUS takes this approach. But there is no guarantee that the global minimum of the objective function is achieved by the ultimate solution. Moreover, K-means takes the number of clusters as an input, and additional effort is needed to determine the “right” cluster size. The MCLUST algorithm, on the other hand, maximizes the likelihood functions of mixture Gaussian models with varying number of clusters individually, using the expectation–maximization (EM; Dempster, Laird, and Rubin 1977) algorithm, and then performs a BIC-based model selection to determine the “right” cluster size.

The sampling-based approaches to clustering avoid some of the problems of K-means and MCLUST. Many methods in this category use variations of the Gibbs and the MH algorithm to sample from the posterior induced by a Bayesian clustering model. One can either use a Bayes nonparametric hierarchical model (Liu 1996) to induce the clustering effect, or prescribe a prior distribution on the space of partitions of the data. Jain and Neal (2004) used the DPMM prior and Jensen and Liu (2007) used the uniform prior, both of which give positive probability to all possible clustering solutions with all possible numbers of clusters. Given the choice of prior, one can determine the “right” number of clusters in the light of the posterior distribution. Thus, if a MCMC sampler is efficient enough to sample from the posterior distribution, we can avoid the problem of specifying the number of clusters beforehand.

Unfortunately, both vanilla Gibbs and MH algorithms are known to perform poorly in high-dimensional multimodal sampling problems. In the MCMC literature, a powerful approach for improving the mixing property of a MCMC sampler is the tempering framework (e.g., PT and EMC). The EMC algorithm, which adds “crossover” moves to the PT structure has been shown to perform better than PT in many cases (Liang and Wong 2000). Our main innovation over vanilla EMC is the design of three new crossover moves that take advantage of the special structure of the space of clustering solutions. We discuss the main motivation behind these new moves in Section 3.2, and comment on their unique features in Section 6. In the following sections, we will use $\{m : n\}$ as a short hand for the set of positive integers $\{m, m+1, \ldots, n\}$.

3. THE EMCC ALGORITHM

Suppose we have d objects to be clustered. Let z_u be the cluster label for object u, $u \in \{1 : d\}$, and call $\underset{\sim}{z} \triangleq (z_1, z_2, \ldots, z_d)$ the *cluster indicator vector*. By convention, we take $z_u \in \{1 : d\}$, and thus $\underset{\sim}{z} \in Z \triangleq \{1 : d\}^d$. For example, $\underset{\sim}{z} = (k, k, \ldots, k)$ for any $k \in \{1 : d\}$ means that all our objects belong to only one cluster. Also, $\underset{\sim}{z}$ equals $(1, 2, \ldots, d)$ or any of its permutations means that each object forms its own cluster. Given $\underset{\sim}{z}$, we call $\mathcal{A} \triangleq \{A_k\}_{k=1}^{K}$ the *partition representation* of $\underset{\sim}{z}$, where the A_k's form a partition of $\{1 : d\}$, and each A_k corresponds to the set of indices (or objects) which have the same cluster label. Clearly, there is a many-to-one correspondence between $\underset{\sim}{z}$ and its partition representation, and we will use them interchangeably. In this article, our goal is to sample from the target density expressed in the Boltzmann distribution form

$$g(\underset{\sim}{z}) \; \propto \; \exp\{-H(\underset{\sim}{z})/\tau_{\min}\}, \quad \underset{\sim}{z} \in \underset{\sim}{Z}. \tag{3.1}$$

We call $H(\cdot)$ the *energy* function. For explicit form of $H(\cdot)$ as determined by the clustering formulation see, for example, Section 5.1. Clearly, lower energy values correspond to higher density regions of the sample space. We refer to samples with lower (higher) energy values as *good* (*bad*) samples throughout the article. Usually the temperature $\tau_{\min}$ is set at 1. But if we are interested in locating the mode(s) of $g(\underset{\sim}{z})$, we consider $\tau_{\min} \in (0, 1)$.

In PT and EMC, one needs to design a suitable temperature ladder, which is a decreasing sequence of positive numbers, $t_1 > t_2 > \cdots > t_N > 0$, such that $t_N = \tau_{\min}$. We denote t_1 by $\tau_{\max}$ for later reference. For $i \in \{1 : N\}$, we define the sequence of densities $f_i(\underset{\sim}{x}_i) \propto \exp\{-H(\underset{\sim}{x}_i)/t_i\}$, $\underset{\sim}{x}_i \in Z$; we use both $\underset{\sim}{z}$ and $\underset{\sim}{x}_i$ to denote cluster indicator vectors in the rest of the article. Now, we expand the sample space from Z to Z^N and define the new target density as

$$f(\mathbf{x}) \; \propto \; \prod_{i=1}^{N} f_i(\underset{\sim}{x}_i), \quad \mathbf{x} \triangleq (\underset{\sim}{x}_1, \underset{\sim}{x}_2, \ldots, \underset{\sim}{x}_N) \in Z^N. \tag{3.2}$$

Borrowing terminology from Liang and Wong (2001), we call $(\mathbf{x}, \mathbf{t}) \triangleq (\underset{\sim}{x}_1, t_1; \underset{\sim}{x}_2, t_2; \ldots; \underset{\sim}{x}_N, t_N)$ the *population*, and $\underset{\sim}{x}_i$ the ith *chromosome*. The EMCC algorithm samples from the new target density $f(\cdot)$ using moves described in the following subsections.

3.1 MUTATION

This move consists of updating a chosen chromosome using a "split-merge" MH step. More precisely, we choose $i \in \{1 : N\}$ according to a distribution $p(I = i \mid \mathbf{x})$ (e.g., uniform or deterministic). We set the "split" and the "merge" probability as $q_s \in (0, 1)$ and $q_m = 1 - q_s$, respectively. Let $\mathcal{A} \triangleq \{A_k\}_{k=1}^{K}$ be the partition representation of $\underset{\sim}{x}_i$. We randomly pick one of the d coordinates of $\underset{\sim}{x}_i$, say, the uth. Let $u \in A_{k_0}$, for $k_0 \in \{1 : K\}$. If $\left|A_{k_0}\right| = 1$, we randomly choose $k_1 \neq k_0$, and propose to merge u with A_{k_1}, which gives rise to a new partition $\mathcal{C} \triangleq \left[\mathcal{A} \setminus \left\{A_{k_0}, A_{k_1}\right\}\right] \cup \left\{A_{k_1} \cup \{u\}\right\}$. If on the other hand $\left|A_{k_0}\right| > 1$, we consider two different scenarios. If $K = 1$; that is, $\mathcal{A} = \{A_{k_0}\}$, we split u from the rest to form a new partition, $\mathcal{C} \triangleq \left\{A_{k_0} \setminus \{u\}, \{u\}\right\}$. If $K > 1$, then we split or merge u with probabilities q_s and q_m, respectively. Splitting gives a new partition $\mathcal{C} \triangleq \left[\mathcal{A} \setminus \left\{A_{k_0}\right\}\right] \cup \left\{A_{k_0} \setminus \{u\}, \{u\}\right\}$. For merging, we randomly choose $k_1 \neq k_0$ and merge u to A_{k_1}, giving a new partition $\mathcal{C} \triangleq \left[\mathcal{A} \setminus \left\{A_{k_0}, A_{k_1}\right\}\right] \cup \left\{A_{k_0} \setminus \{u\}, A_{k_1} \cup \{u\}\right\}$. Let $\underset{\sim}{y}_i$ be a cluster indicator vector representing the partition $\mathcal{C}$. We accept the new population $(\mathbf{y}, \mathbf{t}) = (\underset{\sim}{x}_1, t_1; \ldots; \underset{\sim}{y}_i, t_i; \ldots; \underset{\sim}{x}_N, t_N)$ with probability $\min(1, r_m)$, where

$$r_m = \frac{f_i(\underset{\sim}{y}_i)}{f_i(\underset{\sim}{x}_i)} \times \frac{p(I = i \mid \mathbf{y})}{p(I = i \mid \mathbf{x})} \times \frac{T(\underset{\sim}{y}_i, \underset{\sim}{x}_i)}{T(\underset{\sim}{x}_i, \underset{\sim}{y}_i)}. \tag{3.3}$$

Here $T(\underset{\sim}{x}_i, \underset{\sim}{y}_i)$ is the probability of generating $\underset{\sim}{y}_i$ from $\underset{\sim}{x}_i$ by the "split-merge" move, and is given by the following. Let $\mathcal{C}$ from the previous paragraph have L clusters, $\mathcal{C} \triangleq \{C_l\}_{l=1}^{L}$, and $u \in C_{l_0}$ for some $l_0 \in \{1 : L\}$. Then, $T(\cdot, \cdot)$ can be expressed as

$$T(\underset{\sim}{x}_i, \underset{\sim}{y}_i) = \begin{cases} 1 & \text{if} \quad \left|C_{l_0}\right| = 1, |\mathcal{A}| = 1 \\ q_s & \text{if} \quad \left|C_{l_0}\right| = 1, |\mathcal{A}| > 1 \\ 1/(|\mathcal{A}| - 1) & \text{if} \quad \left|C_{l_0}\right| > 1, \left|A_{k_0}\right| = 1 \\ q_m/(|\mathcal{A}| - 1) & \text{if} \quad \left|C_{l_0}\right| > 1, \left|A_{k_0}\right| > 1 \end{cases}.$$

This move is different from the "split-merge" Metropolis–Hastings move found in Jain and Neal (2004) and Dahl (2003); see Section 6 for further discussion.

3.2 THE NEW CROSSOVER MOVES

In general, a crossover move takes two chromosomes in the current population, which are called the *parents*, and recombines them to produce two new chromosomes, called the *children*, each inheriting some aspects of the parental configurations. We have developed two types of crossover moves, which swap or reallocate, respectively, the intersections of the clusters of the two chosen parent chromosomes to produce child chromosome(s), and hence we coin the terminology subcluster swap crossover (SCSC) and subcluster reallocation crossover (SCRC), respectively. The SCSC move is of two types, namely, SCSC:TWO-NEW and SCSC:ONE-NEW.

The main motivation behind the design of these new crossover moves is as follows. Since the crossover moves are disciplined by the Metropolis–Hastings acceptance–rejection rule, two new children have to be proposed to replace the two parents to maintain reversibility. But the children produced by crossing over two good parents are usually not as good as their parents, and replacing two good parents by their children, even good ones, impoverishes the population. It has been a long standing problem to design crossover moves, that preserve both good parents and good children to a certain degree. Our moves are the first to appear in the literature that satisfy this requirement.

We use the following notation in the subsections below. Let $\underset{\sim}{x}_i$ and $\underset{\sim}{x}_j$ be two parent chromosomes with partition representations $\mathcal{A} \triangleq \{A_k\}_{k=1}^{K}$ and $\mathcal{B} \triangleq \{B_l\}_{l=1}^{L}$, respectively. Given two sets of indices (k_1, l_1) and (k_2, l_2) with $k_1, k_2 \in \{1 : K\}$ and $l_1, l_2 \in \{1 : L\}$, and two sets G_1, G_2 (e.g., $G_i = A_{k_i} \cap B_{l_i}$, $i = 1, 2$), we define the disjoint collection of sets $\{S_{k,l} \mid k \in \{1:K\},\ l \in \{1:L\}\}$ where,

$$S_{k,l} = \begin{cases} G_1 & \text{for} \quad k = k_2,\ l = l_2 \\ G_2 & \text{for} \quad k = k_1,\ l = l_1 \\ A_k \cap B_l & \text{otherwise} \end{cases}. \tag{3.4}$$

Then we form $C_k \triangleq \cup_{l=1}^{L} S_{k,l},\ k \in \{1:K\}$ and $D_l \triangleq \cup_{k=1}^{K} S_{k,l},\ l \in \{1:L\}$. Let $\underset{\sim}{y}_i$ and $\underset{\sim}{y}_j$ be cluster indicators formed by the partition representations $\mathcal{C} \triangleq \{C_k\}_{k=1}^{K}$ and $\mathcal{D} \triangleq \{D_l\}_{l=1}^{L}$, respectively. We introduce the notation $(\underset{\sim}{y}_i, \underset{\sim}{y}_j) = \text{SCShuffle}((\underset{\sim}{x}_i, \underset{\sim}{x}_j); (k_1, l_1), G_1; (k_2, l_2), G_2)$.

3.3 SUBCLUSTER SWAP CROSSOVER (SCSC:TWO-NEW)

In this move, we sample $i, j \in \{1:N\},\ i \neq j$, according to distributions $p(I = i \mid \mathbf{x})$ and $p(J = j \mid \mathbf{x}, I = i)$. Let $\mathcal{A} \triangleq \{A_k\}_{k=1}^{K}$ and $\mathcal{B} \triangleq \{B_l\}_{l=1}^{L}$ be the partition representations of $\underset{\sim}{x}_i$ and $\underset{\sim}{x}_j$, respectively. Then, we randomly choose $k_1, k_2 \in \{1:K\}$ and $l_1, l_2 \in \{1:L\}$ such that $k_1 \neq k_2$ and $l_1 \neq l_2$, and $A_{k_u} \cap B_{l_u} \neq \emptyset,\ u = 1, 2$. Now, we obtain $(\underset{\sim}{y}_i, \underset{\sim}{y}_j) = \text{SCShuffle}((\underset{\sim}{x}_i, \underset{\sim}{x}_j); (k_1, l_1), A_{k_1} \cap B_{l_1}; (k_1, l_2), A_{k_2} \cap B_{l_2})$.

The choice of the k_i's and the l_i's forces the children $\underset{\sim}{y}_i, \underset{\sim}{y}_j$ to be distinct from each other and their parents $\underset{\sim}{x}_i, \underset{\sim}{x}_j$ (unless the parents have the same configuration to begin with), and hence the name of the present move. We replace the parents by the children in the population with probability $\min(1, r_{\text{scsc:tn}})$ where,

$$r_{\text{scsc:tn}} = \frac{f_i(\underset{\sim}{y}_i)\, f_j(\underset{\sim}{y}_j)}{f_i(\underset{\sim}{x}_i)\, f_j(\underset{\sim}{x}_j)} \times \frac{T_{i,j}(\mathbf{y}, \mathbf{x})}{T_{i,j}(\mathbf{x}, \mathbf{y})}, \tag{3.5}$$

with $T_{i,j}(\mathbf{x}, \mathbf{y}) = p(I = i \mid \mathbf{x}) p(J = j \mid \mathbf{x}, I = i) + p(I = j \mid \mathbf{x}) p(J = i \mid \mathbf{x}, I = j)$. If we randomly select I and J without replacement, we refer to this move as the `TWO-NEW-r-r` move, for future reference. On the other hand, if we take $p(I = i \mid \mathbf{x}) \propto \exp\{-H(\underset{\sim}{x}_i)/s\}$ and $p(J = j \mid \mathbf{x}, I = i) \propto \exp\{-H(\underset{\sim}{x}_j)/s\},\ j \neq i$, with selection temperature s positive and close to $\tau_{\min}$, we call this move the `TWO-NEW-b-b` move, for later use. A diagrammatic representation of this move is shown in Figure 1.

3.4 SUBCLUSTER SWAP CROSSOVER (SCSC:ONE-NEW)

In this crossover, we sample $i \in \{1:N\}$ according to distribution $p(I = i \mid \mathbf{x})$. Then, we randomly choose J as one of the two neighbors of I with equal probability (or the only possible neighbor when I is either 1 or N). Now, we choose $M \in \{I, J\}$ with probability $p(M = m \mid \mathbf{x}, I, J)$, and call the chromosome $\underset{\sim}{x}_M$ the *survivor-parent* and the other one the *nonsurvivor-parent*. Suppose we happen to choose $\underset{\sim}{x}_i$ as our survivor-parent and $\underset{\sim}{x}_j$ as the nonsurvivor-parent. Also, let $\mathcal{A} \triangleq \{A_k\}_{k=1}^{K}$ and $\mathcal{B} \triangleq \{B_l\}_{l=1}^{L}$ be the partition representations of $\underset{\sim}{x}_i$ and $\underset{\sim}{x}_j$, respectively. We randomly choose $k_1 \in \{1:K\}$ and $l_1, l_2 \in \{1:L\}$ such that $l_1 \neq l_2$ and $A_{k_1} \cap B_{l_u} \neq \emptyset,\ u = 1, 2$. Now, we produce the children as $(\underset{\sim}{y}_i, \underset{\sim}{y}_j) = \text{SCShuffle}((\underset{\sim}{x}_i, \underset{\sim}{x}_j); (k_1, l_1), A_{k_1} \cap B_{l_1}; (k_1, l_2), A_{k_1} \cap B_{l_2})$. By construction $\underset{\sim}{y}_i = \underset{\sim}{x}_i$, and hence the name SCSC:ONE-NEW. We call $\underset{\sim}{y}_j$ the *modified-child* produced

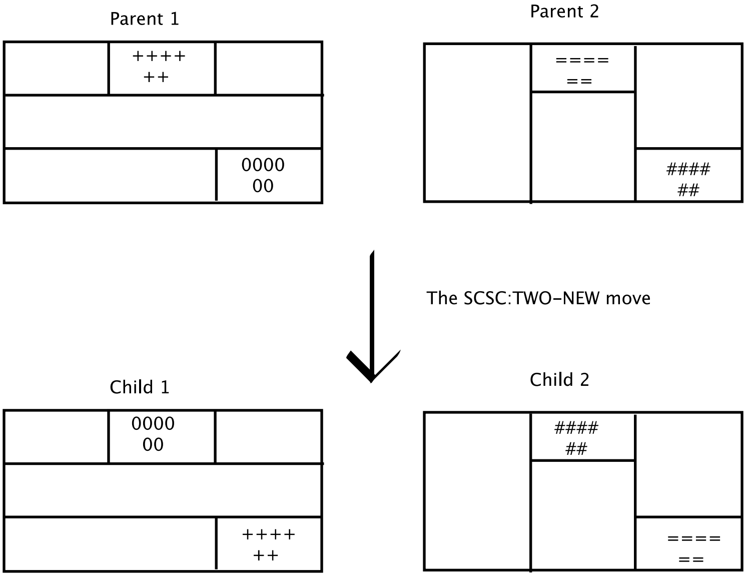

Figure 1. Diagrammatic representation of SCSC:TWO-NEW clustering (Section 3.3). The horizontal and the vertical lines form the partition representations of parent1 and parent2, respectively. The objects in the two sub-cluster intersections represented by symbols "+" and "0" in parent1 are swapped. Similarly, objects represented by symbols "=" and "#" in parent2 are also swapped. This swapping process produces child1 and child2 both of which are different from each other and from parent1 and parent2.

by modifying the nonsurvivor-parent $\underset{\sim}{x}_j$ with guidance from the survivor-parent $\underset{\sim}{x}_i$. The child $\underset{\sim}{y}_j$ is accepted to replace $\underset{\sim}{x}_j$ in the population with probability $\min(1, r_{\text{scsc:on}})$, where

$$r_{\text{scsc:on}} = \frac{f_j(\underset{\sim}{y}_j)}{f_j(\underset{\sim}{x}_j)} \times \frac{T_j(\mathbf{y},\mathbf{x})}{T_j(\mathbf{x},\mathbf{y})}. \tag{3.6}$$

For $T_j(\cdot,\cdot)$ we note that, unless j is at the boundary of the temperature ladder, both of the neighbors are its candidate survivor-parents. Let $h(\underset{\sim}{x}_i, \underset{\sim}{x}_j, \underset{\sim}{y}_j)$ be an indicator function that takes the value 1 if $\underset{\sim}{y}_j$ is the modified child from survivor-parent $\underset{\sim}{x}_i$ and nonsurvivor-parent $\underset{\sim}{x}_j$ and is 0 otherwise. Let $g(i, j, \mathbf{x}) \triangleq \{p(I = i \mid \mathbf{x}) \times p(J = j \mid \mathbf{x}, I = i) + p(I = j \mid \mathbf{x}) \times p(J = i \mid \mathbf{x}, I = j)\} \times p(M = i \mid \mathbf{x}, I = i, J = j)$, be the probability of choosing $\underset{\sim}{x}_i$ as the survivor-parent. Thus, we have

$$T_j(\mathbf{x},\mathbf{y}) = \begin{cases} g(1,2,\mathbf{x}) & \text{for } j = 1 \\ g(N, N-1, \mathbf{x}) & \text{for } j = N \\ \{g(j-1, j, \mathbf{x}) \times h(\underset{\sim}{x}_{j-1}, \underset{\sim}{x}_j, \underset{\sim}{y}_j) & \\ + g(j+1, j, \mathbf{x}) \times h(\underset{\sim}{x}_{j+1}, \underset{\sim}{x}_j, \underset{\sim}{y}_j)\} & \text{for } 1 < j < N \end{cases}. \tag{3.7}$$

If we randomly choose both I and M, then we call this move the `ONE-NEW-r-r` move for later use. On the other hand, if we take $p(I = i \mid \mathbf{x}) \propto \exp\{-H(\underset{\sim}{x}_i)/s\}$ and $p(M = m \mid \mathbf{x}, I, J) \propto \exp\{-H(\underset{\sim}{x}_m)/s\}$ with s positive and close to $\tau_{\min}$, we call this move the `ONE-NEW-b-b` for future reference. A diagrammatic representation of the present move is shown in Figure 2.

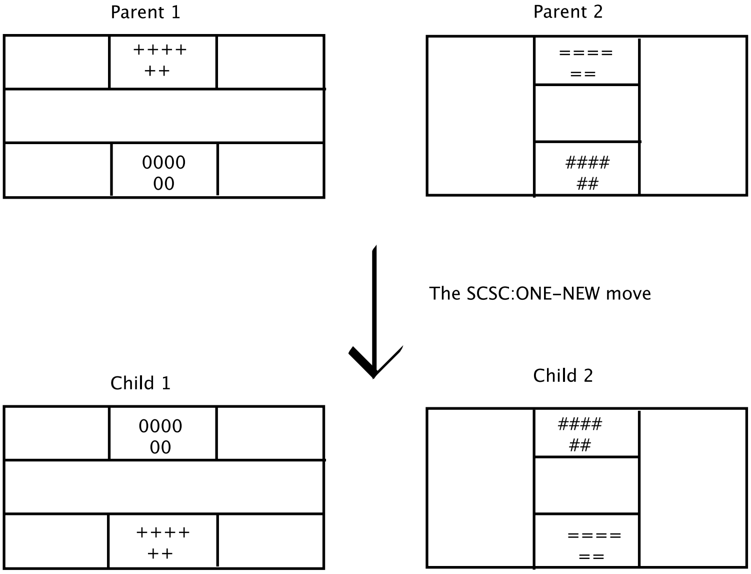

Figure 2. Diagrammatic representation of SCSC:ONE-NEW clustering (Section 3.4). The horizontal and the vertical lines form the partition representations of parent1 and parent2, respectively. The objects in the two subcluster intersections represented by symbols "+" and "0" in parent1 are swapped. Similarly, objects represented by symbols "=" and "#" in parent2 are also swapped. This swapping process produces child1 and child2, where child1 is different from parent1 but child2 is same as parent2.

3.5 SUBCLUSTER REALLOCATION CROSSOVER (SCRC)

The initial steps of this move are exactly the same as in the SCSC:ONE-NEW move described in Section 3.4. Following the steps of Section 3.4, we choose I, J, M, k_1, and $l_1 \neq l_2$. Then, we take $H \triangleq \left(A_{k_1} \cap B_{l_1}\right) \cup \left(A_{k_1} \cap B_{l_2}\right)$, and divide it into two nonempty subsets, H_1 and H_2. Let $m_u = \left|A_{k_1} \cap B_{l_u}\right|$ and $h_u = |H_u|$, $u = 1, 2$, and thus $h_1 + h_2 = m_1 + m_2$. We take a random sample of size h_1 from the $h_1 + h_2$ members of H to form H_1, and hence H_2. Then, we produce the children as $(\underset{\sim}{y}_i, \underset{\sim}{y}_j) =$ SCShuffle$((\underset{\sim}{x}_i, \underset{\sim}{x}_j); (k_1, l_1), H_1; (k_1, l_2), H_2)$. By construction, $\underset{\sim}{y}_i = \underset{\sim}{x}_i$. As in Section 3.4, we call $\underset{\sim}{y}_j$ the modified child produced by modifying the nonsurvivor-parent $\underset{\sim}{x}_j$ and with guidance from the survivor-parent $\underset{\sim}{x}_i$. The modified child has been produced by reallocating the elements of the subclusters of the nonsurvivor parent, and hence the name of the present move. The child $\underset{\sim}{y}_j$ is accepted to replace $\underset{\sim}{x}_j$ in the population with probability $\min(1, r_{\text{scrc}})$, where

$$r_{\text{scrc}} = \frac{f_j(\underset{\sim}{y}_j)}{f_j(\underset{\sim}{x}_j)} \times \frac{T_j(\mathbf{y}, \mathbf{x})}{T_j(\mathbf{x}, \mathbf{y})} \times \frac{S_j(\mathbf{y}, \mathbf{x})}{S_j(\mathbf{x}, \mathbf{y})}. \tag{3.8}$$

Here $T_j(\cdot, \cdot)$ has an expression similar to Equation (3.7) of Section 3.4 and we omit the details to avoid repetition. $S_j(\cdot, \cdot)$ is the reallocation probability, and its expression depends on the reallocation scheme used. If we require that $\{h_1, h_2\} = \{m_1, m_2\}$, then we call the move the `SAME-SIZE` move, for later reference. If, on the other hand, we pick a random size $h_1 \in \{1 : (|H| - 1)\}$, we call the move the `RANDOM-SIZE` move. For `SAME-SIZE`

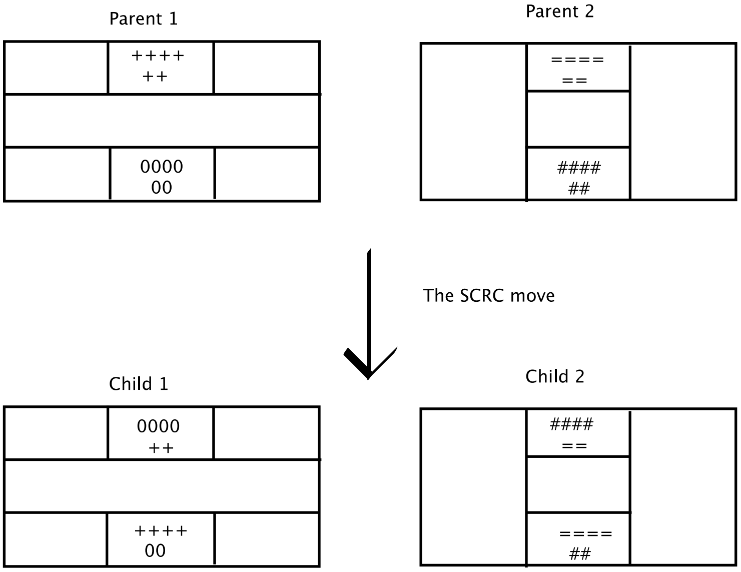

Figure 3. Diagrammatic representation of SCRC clustering (Section 3.5). The horizontal and the vertical lines form the partition representations of parent1 and parent2, respectively. The objects in the two subcluster intersections represented by symbols "+" and "0" in parent1 are (randomly) reallocated. Similarly, objects represented by symbols "=" and "#" in parent2 are also (randomly) reallocated. This swapping process produces child1 and child2, where child1 is different from parent1 but child2 is same as parent2.

since $\{h_1, h_2\} = \{m_1, m_2\}$, we have $S_j(\mathbf{y}, \mathbf{x})/S_j(\mathbf{x}, \mathbf{y}) = 1$. For RANDOM-SIZE

$$S_j(\mathbf{y}, \mathbf{x})/S_j(\mathbf{x}, \mathbf{y}) = \left(1\Big/\binom{h_1 + h_2}{m_1}\right) \Big/ \left(1\Big/\binom{m_1 + m_2}{h_1}\right) = \tfrac{m_1!m_2!}{h_1!h_2!},$$

where $\binom{n}{r}$ is the standard combination coefficient. Since $k(n) \triangleq \binom{n}{r}$ as a function of r, is maximized for $r = [n/2]$, $h_1 = h_2$ gives $\binom{h_1+h_2}{h_1} \geq \binom{h_1+h_2}{m_1}$, and hence $S_j(\mathbf{y}, \mathbf{x})/S_j(\mathbf{x}, \mathbf{y}) \geq 1$. Thus, RANDOM-SIZE prefers equal reallocation of members. A diagrammatic representation of this move is shown in Figure 3.

3.6 RANDOM EXCHANGE (RE)

This move is the same as the exchange move of PT. Briefly, we randomly select $i \in \{1 : N\}$, and set $j = i \pm 1$ with equal probabilities (with a small modification at the two ends of the temperature ladder so as to make the proposal symmetric). The new configuration $(\mathbf{y}, \mathbf{t}) = (\underset{\sim}{x}_1, t_1; \ldots; \underset{\sim}{x}_j, t_i; \ldots; \underset{\sim}{x}_i, t_j; \ldots; \underset{\sim}{x}_N, t_N)$ is accepted with probability $\min(1, r_{re})$, where

$$r_{re} = \frac{f_i(\underset{\sim}{x}_j)\, f_j(\underset{\sim}{x}_i)}{f_i(\underset{\sim}{x}_i)\, f_j(\underset{\sim}{x}_j)} = \exp\left[(H(\underset{\sim}{x}_j) - H(\underset{\sim}{x}_i)) \cdot (1/t_j - 1/t_i)\right]. \tag{3.9}$$

Since $r_{re} \geq 1$ if $j < i$ and $H(\underset{\sim}{x}_j) \leq H(\underset{\sim}{x}_i)$, a virtue of RE is to bring good samples down the ladder.

3.7 IMPLEMENTATION OF THE EMCC ALGORITHM

Let $\mathbf{x}^{(0)} = (\underset{\sim}{x}_1^{(0)}, \underset{\sim}{x}_2^{(0)}, \ldots, \underset{\sim}{x}_N^{(0)})$ be an initial configuration of the population. Fix $p_m \in (30\%, 50\%)$, and call it the *mutation rate*. Let $\{t_1 > t_2 > \cdots > t_N = \tau_{\min} > 0\}$ be a well-chosen temperature ladder. Then, one iteration of the EMCC algorithm consists of the following sequence of steps:

Algorithm 1 (EMCC).

1. Choose mutation or crossover with probability p_m and $(1 - p_m)$, respectively. For mutation, all the chromosomes are systematically updated M times each. For crossover, one selects, and crosses pairs of chromosomes using any one of the ways described in Sections 3.3, 3.4, and 3.5, $[N/2]$ times.

2. Apply RE to the population N times; that is, propose to exchange N pairs of neighboring chromosomes, where one pair is chosen in the way described in Section 3.6.

For the crossover step, we use one of `TWO-NEW-b-b`, `TWO-NEW-r-r`, `ONE-NEW-b-b`, `ONE-NEW-r-r`, `RANDOM-SIZE`, and `SAME-SIZE`; we refer to these six moves as the EMCC schemes or EMCC family of moves. At the end of the iterations the required sample from the target distribution consists of $\{x_N^{(t)} \mid t = 1, 2, \ldots\}$. For a general discussion on the construction of the underlying temperature ladder for the EMC algorithm, see Liang and Wong (2001, sect. 3); Goswami and Liu (2007) provide a specific recipe, which consists of two preliminary runs of the EMC algorithm. First preliminary run samples are used to determine $\tau_{\max}$. The samples from the second preliminary run are used to place the intermediate temperatures $t_2, t_3, \ldots, t_{N-1}$ between $\tau_{\max}(= t_1)$ and $(t_N =)\tau_{\min}$. This recipe is easily modified to work in the EMCC setting.

4. GIBBS SAMPLING

Dirichlet process priors are often used in nonparametric Bayes analysis (Liu 1996; Jain and Neal 2004). The problem of clustering can be formulated as a nonparametric Bayes problem. Let the data $\underset{\sim}{y} \triangleq (y_1, y_2, \ldots, y_d)$ be a set of observations such that $y_u \mid \theta_u \overset{\text{indep}}{\sim} F(\theta_u)$ and $\theta_u \mid G \overset{\text{iid}}{\sim} G$. Also, we assume that $G(\cdot)$ has a Dirichlet process prior with the baseline measure G_0 and the total mass parameter $\alpha (> 0)$ (Ferguson 1974). The hyper-parameters G_0 and α are set by the user. Lower values α correspond to smaller number of clusters being preferred by the prior. For a cluster indicator vector $\underset{\sim}{z}$ with partition representation $\mathcal{H} \triangleq \{H_s\}_{s=1}^S$, Dirichlet process induced prior (Blackwell and MacQueen 1973), the likelihood, and the resulting posterior, respectively, take the following forms:

$$p(\underset{\sim}{z}) = \alpha^S \frac{\prod_{s=1}^S (|H_s| - 1)!}{\prod_{u=1}^D (u - 1 + \alpha)} \tag{4.1}$$

$$p(\underset{\sim}{y} \mid \underset{\sim}{z}) = \int p(\underset{\sim}{y}, \underset{\sim}{\theta} \mid \underset{\sim}{z})\, d\underset{\sim}{\theta} = \prod_{s=1}^S \int \left[\left\{ \prod_{u \in H_s} F(y_u \mid \theta_{H_s}) \right\} G_0(\theta_{H_s})\, d\theta_{H_s} \right] \tag{4.2}$$

$$p(\underset{\sim}{z} \mid \underset{\sim}{y}) \propto p(\underset{\sim}{z}) \times p(\underset{\sim}{y} \mid \underset{\sim}{z}). \tag{4.3}$$

The following decomposition of the prior in Equation (4.1) into product of conditionals is used in the Gibbs sampling from the posterior in Equation (4.3). Given $z_1 \in \{1 : d\}$ we have

$$\begin{aligned} P(z_u = z \mid \underset{\sim}{z}_{1:(u-1)}) =& \frac{n_{u,z}}{u-1+\alpha} \quad \text{for} \quad z \in \underset{\sim}{z}_{1:(u-1)} \\ P(z_u = z \mid \underset{\sim}{z}_{1:(u-1)}) =& \frac{\alpha}{u-1+\alpha} \quad \text{otherwise,} \end{aligned}$$

where $u > 1$, $\underset{\sim}{z}_{1:(u-1)} = \{z_v\}_{v=1}^{(u-1)}$ and $n_{u,z} = |\{z_v \mid z_v = z, v \in \{1 : (u-1)\}\}|$. For Gibbs sampling from the posterior $p(\underset{\sim}{z} \mid \underset{\sim}{y})$, we need $p(z_u \mid \underset{\sim}{z}_{-u}, \underset{\sim}{y})$, where $\underset{\sim}{z}_{-u}$ denotes both the set $\{z_v \mid v \neq u, v \in \{1 : d\}\}$ and the $(d-1)$-tuple formed by it. Let the partition representation of $\underset{\sim}{z}_{-u}$ be $\widetilde{\mathcal{H}} \triangleq \{\widetilde{H}_s\}_{s=1}^{\widetilde{S}}$. Also, let $p(\underset{\sim}{y} \mid \widetilde{H}_s)$ be the likelihood for the sub-cluster $\widetilde{H}_s$. Note, for $s_0 \notin \underset{\sim}{z}_{-u}$, $\widetilde{H}_{s_0} = \{u\}$, and thus we have,

$$\begin{aligned} p(z_u = s_0 \mid \underset{\sim}{z}_{-u}, \underset{\sim}{y}) \propto\ & p(\underset{\sim}{y} \mid z_u = s_0, \underset{\sim}{z}_{-u}) p(z_u = s_0, \underset{\sim}{z}_{-u}) \\ \propto\ & p(\underset{\sim}{y} \mid z_u = s_0, \underset{\sim}{z}_{-u}) p(z_u = s_0 \mid \underset{\sim}{z}_{-u}) \propto p(y_u \mid \widetilde{H}_{s_0})\alpha. \end{aligned}$$

Now we take $s \in \underset{\sim}{z}_{-u}$ (clearly, $s \neq s_0$) and compute the ratio:

$$\begin{aligned} q(s, s_0) \triangleq\ & \frac{p(z_u = s \mid \underset{\sim}{z}_{-u}, \underset{\sim}{y})}{p(z_u = s_0 \mid \underset{\sim}{z}_{-u}, \underset{\sim}{y})} = \frac{p(\underset{\sim}{y} \mid z_u = s, \underset{\sim}{z}_{-u}) p(z_u = s \mid \underset{\sim}{z}_{-u})}{p(\underset{\sim}{y} \mid z_u = s_0, \underset{\sim}{z}_{-u}) p(z_u = s_0 \mid \underset{\sim}{z}_{-u})} \\ =\ & \frac{p(\underset{\sim}{y} \mid \widetilde{H}_s \cup \{u\}) \left|\widetilde{H}_s\right|}{p(\underset{\sim}{y} \mid \widetilde{H}_s) p(y_u \mid \widetilde{H}_{s_0})\alpha}. \end{aligned}$$

In some of the examples to follow the ratios above turn out be very easy to compute. We finally set

$$p(z_u = s \mid \underset{\sim}{z}_{-u}, \underset{\sim}{y}) = q(s, s_0) \cdot p(z_u = s_0 \mid \underset{\sim}{z}_{-u}, \underset{\sim}{y}) \propto \frac{p(\underset{\sim}{y} | \widetilde{H}_s \cup \{u\}) |\widetilde{H}_s|}{p(\underset{\sim}{y} | \widetilde{H}_s)}.$$

5. EXAMPLES

In the following sections, we call the sampler arising from application of the mutation move (Section 3.1) and the Gibbs move (Section 4) only to $f_N(\underset{\sim}{z}) = g(\underset{\sim}{z})$ (Section 3) the `MH` scheme and the `Gibbs` scheme, respectively. In the following examples, we used the same set of starting values across different schemes for a fair comparison. We took the split probability $q_s = 1/2$ for the mutation step (Section 3.1). The burn-in period for a chain producing T draws was taken to be $[T/4]$. It was argued by Geyer (1992) that fewer than 5% burn-in works well; we used a slightly larger default burn-in to be on the safe side.

To compare the performance of the various samplers, we used the *average integrated autocorrelation time* (AIAT) of several statistics computed from the MCMC draws. Let $\widehat{\tau}_{T_i}$ be the *integrated autocorrelation time* (IAT) computed from the T_i values of the (one-dimensional) statistic $S(\cdot)$, computed from the T_i post-burn-in draws from the ith run of a sampler. Then, for $R(\geq 1)$ different runs, the AIAT for a given statistic is defined as

$\text{AIAT}_R \triangleq \frac{1}{R}\sum_{i=1}^{R} \widehat{\tau}_{T_i}$. We computed $\widehat{\tau}_T$ following Geyer (1992). For a one-dimensional series $\{u_t\}_{t=1}^{T}$ of draws from a reversible Markov chain with sample auto-covariances $\{\widehat{\gamma}_j\}_{j=0}^{T-1}$, the monotone estimator of the variance of the sample mean is defined as

$$\widehat{\sigma}^2_{\text{mono},T} = -\widehat{\gamma}_0 + 2 \times \sum_{j=0}^{m} \widehat{\Gamma}^{\star}_j,$$

where $\widehat{\Gamma}^{\star}_j = \min\left\{\widehat{\Gamma}_0, \widehat{\Gamma}_1, \ldots, \widehat{\Gamma}_j\right\}$, with $\widehat{\Gamma}_j = \widehat{\gamma}_{2j} + \widehat{\gamma}_{2j+1}$, $j = 0, 1, \ldots$ and m is such that $\Gamma_{m+1} \leq 0$ for the first time. We took $\widehat{\tau}_T \triangleq \widehat{\sigma}^2_{\text{mono},T}/\widehat{\gamma}_0$; for a detailed discussion on computing IAT, please refer to Geyer (1992) or Goswami and Liu (2007). We looked at AIAT_R's of the following one-dimensional statistics of $\underset{\sim}{z}$, with partition representation $\mathcal{H} \triangleq \{H_s\}_{s=1}^{S}$:

- Number of distinct clusters: $n(\underset{\sim}{z}) \triangleq S(= |\mathcal{H}|)$.
- Entropy of the clusters (Dahl 2003): $e(\underset{\sim}{z}) \triangleq -\sum_{s=1}^{S} \frac{|H_s|}{d} \log\left(\frac{|H_s|}{d}\right)$.
- The log density: $l(\underset{\sim}{z}) \triangleq -H(\underset{\sim}{z})$, where $H(\cdot)$ is the energy function as in equation (3.1) .
- Proportion of observations in the largest cluster: $p_{\max}(\underset{\sim}{z}) \triangleq \max_{1\leq s\leq S} |H_s|/d$.

We also defined the *average maximum log density* achieved over R runs as $\text{AMLD}_R \triangleq \frac{1}{R}\sum_{i=1}^{R} \max_{1\leq j\leq T_i} l(z_{ij})$, where z_{ij} is the jth of the T_i samples from the ith run, and used it to detect which algorithm(s) were capable of escaping from local modes.

5.1 A SIMULATION STUDY WITH BERNOULLI–BETA CLUSTERING

This example is taken from Jain and Neal (2004), and it concerns clustering of vectors of Binomial observations with a conjugate Beta prior. Let $\boldsymbol{y} = (\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_d)$, where $\boldsymbol{y}_u = (y_{u1}, \ldots, y_{um})$, with $y_{uh} \mid \theta_{uh} \overset{\text{indep}}{\sim} \text{Bernoulli}(\theta_{uh})$. We take $\theta_{uh} \overset{\text{iid}}{\sim} \text{Beta}(\beta_{1h}, \beta_{0h})$. So, we have,

$$p(y_u \mid \theta_u) = \prod_{h=1}^{m} \theta_{uh}^{y_{uh}}(1-\theta_{uh})^{1-y_{uh}} \tag{5.1}$$

$$p(\theta_u) = \prod_{h=1}^{m} \frac{\Gamma(\beta_{1h}+\beta_{0h})}{\Gamma(\beta_{1h})\Gamma(\beta_{0h})}\theta_{uh}^{\beta_{1h}}(1-\theta_{uh})^{\beta_{0h}}. \tag{5.2}$$

Here $\beta_{1h}, \beta_{0h} > 0$, and are set by the user. In the notation of Section 4, $F(\cdot)$ and $G_0(\cdot)$ are given by Equations (5.1) and (5.2), respectively. The likelihood for $\underset{\sim}{z}$ (with partition representation $\mathcal{H} \triangleq \{H_s\}_{s=1}^{S}$) from Equation (4.2) becomes

$$p(\boldsymbol{y} \mid \underset{\sim}{z}) = \prod_{s=1}^{S}\prod_{h=1}^{m} \frac{\Gamma(x_{sh}+\beta_{1h})\Gamma(n_s - x_{sh}+\beta_{0h})\Gamma(\beta_{1h}+\beta_{0h})}{\Gamma(\beta_{1h})\Gamma(\beta_{0h})\Gamma(n_s+\beta_{1h}+\beta_{0h})}, \tag{5.3}$$

Table 1. True mixture distribution components for the Bernoulli-Beta clustering example (Section 5.1). The column entries corresponding to $h = 5$ through $h = 15$ are the same in each row of this table.

		$\theta_{ih},\ h = 1, 2, \ldots, 15$						
mixture no.	proportion	$h = 1$	$h = 2$	$h = 3$	$h = 4$	$h = 5$	$\cdots$	$h = 15$
1	0.2	0.95	0.95	0.95	0.95	0.95	$\cdots$	0.95
2	0.2	0.05	0.05	0.05	0.05	0.95	$\cdots$	0.95
3	0.2	0.95	0.05	0.05	0.95	0.95	$\cdots$	0.95
4	0.2	0.05	0.05	0.05	0.05	0.05	$\cdots$	0.05
5	0.2	0.95	0.95	0.95	0.95	0.05	$\cdots$	0.05

where $x_{sh} = \sum_{i \in H_s} y_{ih}$, and $n_s = |H_s|$. Now, combining $p(\boldsymbol{y} \mid \underset{\sim}{z})$ with the prior $p(\underset{\sim}{z})$ from Equation (4.1), we get the posterior $p(\underset{\sim}{z} \mid \boldsymbol{y})$, which becomes the density $g(\cdot)$ of equation (3.1) for EMCC with $H(\underset{\sim}{z}) = -\log(p(\underset{\sim}{z} \mid \boldsymbol{y}))$ and $\tau_{\min} = 1$. For `Gibbs`, we have for $u \in \{1 : d\}$ (Section 4):

$$p(z_u = s \mid \underset{\sim}{z}_{-u}, \boldsymbol{y}) \propto \begin{cases} \frac{n_s - 1}{d-1+\alpha} \prod_{h=1}^{m} \frac{\sum_{k \in H_s, k \neq u} \delta(y_{kh}, y_{uh}) + \beta_{y_{uh},h}}{n_s - 1 + \beta_{1h} + \beta_{0h}} & \text{if} \quad s \in \underset{\sim}{z}_{-u} \\ \frac{\alpha}{d-1+\alpha} \prod_{h=1}^{m} \frac{\beta_{y_{uh},h}}{\beta_{1h} + \beta_{0h}} & \text{otherwise.} \end{cases} \tag{5.4}$$

We took $m = 15$ and five distinct θ_s's, as shown in Table 1, and simulated 20 $\boldsymbol{y}_u$'s from each of the five $F(\cdot)$'s resulting in $d = 100$ data points. We constructed a temperature ladder of length 20 with $\tau_{\max} = t_1 = 20$. We ran `Gibbs` and `MH` and the six EMCC schemes 20 times each for fixed amount of CPU time. The acceptance rates were in (10%, 20%) for the SCSC:TWO-NEW family of moves, whereas for the SCSC:ONE-NEW and the SCRC family of moves, they were in (0.1%, 10%). From Table 2, we observe that the performance of `Gibbs` and `MH` are more or less comparable, but they did considerably worse than all the EMCC schemes, even though `Gibbs` and `MH` produced around 10 times more samples than the EMCC schemes at the same computational expense. Between the EMCC schemes, the two `TWO-NEW` schemes out-performed the other four, which can be explained by their higher acceptance rates (10%–20% as compared to 0.1%–10%); this is a problem-specific issue and we get a different comparative picture in Section 5.2.

5.2 MOTIF CLUSTERING

This example is taken from Jensen and Liu (2007). We have motif matrices $(\boldsymbol{Y}_1, \boldsymbol{Y}_2, \ldots, \boldsymbol{Y}_d)$ of fixed width ω; that is, $\boldsymbol{Y}_u$'s are matrices of dimension $\omega \times 4$ each. Here Y_{uvk} is the count for the nucleotide $k (\in \{A, G, T, C\})$ for the column $v (\in \{1 : \omega\})$ in the motif matrix $u (\in \{1 : d\})$. Let $d_u \triangleq \sum_k Y_{uvk},\ \ u \in \{1 : d\}$; note, d_u should not depend on v since all the column sums within a motif matrix $\boldsymbol{Y}_u$ are the same. We assume a product multinomial model for the columns of $\boldsymbol{Y}_u$'s; that is, we assume $p(\boldsymbol{Y}_u \mid \boldsymbol{\Theta}_u) = \prod_{v=1}^{\omega} p(\boldsymbol{Y}_{uv} \mid \boldsymbol{\theta}_{uv})$, with $\boldsymbol{Y}_{uv} \mid \boldsymbol{\theta}_{uv} \overset{\text{ind}}{\sim} \text{Multinomial}(d_u, \boldsymbol{\theta}_{uv})$. We take a four-dimensional Dirichlet distribution for the parameters, namely, we take

Table 2. Comparative performance of various algorithms for the Bernoulli Beta clustering example (Section 5.1). The statistics $n(\cdot), e(\cdot), l(\cdot)$, and $p_{\max}(\cdot)$ represent the number of clusters, entropy of clusters, log density of the cluster indicator vector, and the proportion of observations in the largest cluster, respectively. Also, AIAT_{20} and AMLD_{20} refer to the average integrated autocorrelation time and average maximum log density over 20 runs, respectively; for detailed definitions of the quantities mentioned, refer to Section 5.

	AIAT_{20} for statistic				
Method	$n(\underset{\sim}{z})$	$e(\underset{\sim}{z})$	$l(\underset{\sim}{z})$	$p_{\max}(\underset{\sim}{z})$	AMLD_{20}
`Gibbs`	102.489	98.73	35.174	81.975	−606.481
`MH`	96.327	161.879	163.48	82.721	−589.482
`TWO-NEW-b-b`	3.388	6.33	7.889	3.971	−589.782
`TWO-NEW-r-r`	2.653	4.506	6.622	4.703	−589.73
`ONE-NEW-b-b`	13.159	29.827	29.91	7.045	−589.44
`ONE-NEW-r-r`	21.324	21.683	17.336	12.166	−589.44
`RANDOM-SIZE`	20.072	50.111	49.102	7.359	−589.098
`SAME-SIZE`	23.284	55.469	54.159	6.869	-589.226

$\theta_{uv} \overset{\text{iid}}{\sim}$ Dirichlet(c, c, c, c), with $c > 0$. Thus, we have:

$$p(\boldsymbol{Y}_{uv} \mid \boldsymbol{\theta}_{uv}) = \frac{d_u!}{\prod_k Y_{uvk}!}\, \theta_{uvk}^{Y_{uvk}} \tag{5.5}$$

$$p(\boldsymbol{\theta}_{uv}) = \frac{\Gamma(4c)}{\Gamma^4(c)} \prod_k \theta_{uvk}^{c-1}. \tag{5.6}$$

In the notation of Section 4, (5.5) and (5.6) correspond to $F(\cdot)$ and $G_0(\cdot)$, respectively. The likelihood for $\underset{\sim}{z}$ (with partition representation $\mathcal{H} \triangleq \{H_s\}_{s=1}^{S}$) from Equation (4.2) becomes:

$$p(\mathbf{Y} \mid \underset{\sim}{z}) = \prod_{s=1}^{S} \prod_{v=1}^{\omega} \left[\left\{ \prod_{u \in H_s} \frac{d_u!}{\prod_k Y_{uvk}!} \right\} \times \left\{ \prod_k \frac{\Gamma(X_{svk} + c)}{\Gamma(c)} \right\} \times \frac{\Gamma(4c)}{\Gamma(\sum_k X_{svk} + 4c)} \right],$$

where $X_{svk} = \sum_{u \in H_s} Y_{uvk}$. Now, combining $p(\boldsymbol{y} \mid \underset{\sim}{z})$, with the prior $p(\underset{\sim}{z})$ from Equation (4.1), we get the posterior $p(\underset{\sim}{z} \mid \boldsymbol{y})$, which becomes the density $g(\cdot)$ of Equation (3.1) for EMCC. For Gibbs sampling, we have for $u \in \{1 : d\}$ (see Section 4):

$$p(z_u = s \mid \underset{\sim}{z}_{-u}, \boldsymbol{Y}) \propto \begin{cases} \frac{n_s - 1}{d-1+\alpha} \prod_{v=1}^{\omega} \frac{\prod_k \Gamma((Y_{uvk} + \widetilde{X}_{svk}) + c)\Gamma(\sum_k \widetilde{X}_{svk} + 4c)}{\prod_k \Gamma(\widetilde{X}_{svk} + c)\Gamma(\sum_k (Y_{uvk} + \widetilde{X}_{svk}) + 4c)} & \text{if} \quad s \in \underset{\sim}{z}_{-u} \\ \frac{\alpha}{d-1+\alpha} \prod_{v=1}^{\omega} \frac{\prod_k \Gamma(Y_{uvk} + c)}{\Gamma(\sum_k Y_{uvk} + 4c)} \cdot \frac{\Gamma(4c)}{[\Gamma(c)]^4} & \text{otherwise,} \end{cases} \tag{5.7}$$

where $\widetilde{X}_{svk} = \sum_{u \in \widetilde{H}_s} Y_{uvk}$, and $\widetilde{\mathcal{H}} \triangleq \{\widetilde{H}_s\}_{s=1}^{\widetilde{S}}$, the partition representation of $\underset{\sim}{z}_{-u}$. We considered a dataset from Jensen and Liu (2007) with $d = 90$ aligned motif matrices $\boldsymbol{Y}_u$'s each of width $\omega = 8$. For the EMCC schemes, we used a temperature ladder of length 33 with $\tau_{\max} = t_1 = 60$. We ran `Gibbs` and `MH` and the six EMCC schemes 50 times each for fixed amount of CPU time. The acceptance rates were in (10%, 25%) for the SCRC family of moves, whereas for the SCSC:ONE-NEW and the SCSC:TWO-NEW family of moves,

Table 3. Comparative performance of various algorithms for the motif clustering example (Section 5.2). The statistics $n(\cdot), e(\cdot), l(\cdot)$, and represent the number of clusters, entropy of clusters, and log density of the cluster indicator vector, respectively. Also, AIAT_{50} and AMLD_{50} refer to the average integrated autocorrelation time and average maximum log density over 50 runs, respectively; for detailed definitions of the quantities mentioned, refer to Section 5.

	AIAT_{50} for statistic			
Method	$n(\underset{\sim}{z})$	$e(\underset{\sim}{z})$	$l(\underset{\sim}{z})$	$AMLD_{50}$
`Gibbs`	222.136	310.25	247.105	−6467.902
`MH`	21.259	21.359	34.037	−6440.006
`TWO-NEW-b-b`	11.425	12.136	3.212	−6431.652
`TWO-NEW-r-r`	10.902	13.459	3.95	−6431.527
`ONE-NEW-b-b`	10.295	9.619	1.233	−6431.527
`ONE-NEW-r-r`	24.644	26.716	18.298	−6431.527
`RANDOM-SIZE`	5.749	5.392	1.065	−6431.527
`SAME-SIZE`	8.878	8.123	1.018	−6431.527

they were in (0.1%, 10%). In this example, the SCRC moves had the best acceptance rates among the EMCC schemes (compare with Section 5.1), and thus the least AIAT_{50} values in Table 3. We also see from this table that all the EMCC schemes outperformed both `Gibbs` and `MH` in every aspect; particularly, performance of `Gibbs` was much worse compared to all the other methods. In all the 50 runs, the EMCC family of moves achieved the log posterior value of $\log(p(\underset{\sim}{z} \mid \boldsymbol{y})) = -6431.527$, except the `TWO-NEW-b-b` move, which achieved this height 48 times. `MH` failed to achieve the same maximum almost half the times in the same amount of CPU time, and `Gibbs` failed to do so in all the runs.

5.3 OBJECTIVE FUNCTION BASED CLUSTERING

Here we generate five distinct $\boldsymbol{\theta}_s$'s from $\text{Normal}_m(\boldsymbol{\mu}, \tau^2 \boldsymbol{I}_m)$ and simulate 40 $\boldsymbol{Y}_u$'s from each of these five $\text{Normal}_m(\boldsymbol{\theta}_s, \sigma^2 \boldsymbol{I}_m)$, where $\boldsymbol{I}_m$ is the $m \times m$ identity matrix. Thus we have $d = 200$ data points, with $\boldsymbol{Y} = (\boldsymbol{Y}_1, \boldsymbol{Y}_2, \ldots, \boldsymbol{Y}_d)$. We take $m = 2, \boldsymbol{\mu} = \mathbf{0}, \sigma^2 = 1$ and $\tau^2 = 30$; one of the simulated datasets is shown in Figure 4, labeled "Original data."

The method of `K-means` is a popular clustering tool for continuous data. It minimizes the "within-cluster sum of squares" for a given number of clusters, namely, for a cluster indicator $\underset{\sim}{z}$ with partition representation $\mathcal{H} \triangleq \{H_s\}_{s=1}^{S}$, `K-means` minimizes $K(\underset{\sim}{z}) \triangleq \sum_{s=1}^{S} \sum_{u \in H_s} \left\| \boldsymbol{Y}_u - \bar{\boldsymbol{Y}}_s \right\|^2$, where $\bar{\boldsymbol{Y}}_s = \frac{1}{|H_s|} \sum_{u \in H_s} \boldsymbol{Y}_u$. We formulate the minimization of $K(\underset{\sim}{z})$ (for the data in Figure 4) over $\underset{\sim}{z}$ as a stochastic optimization problem by considering the following density

$$p(\underset{\sim}{z} \mid \boldsymbol{Y}) \propto \exp\left[-K(\underset{\sim}{z})/\tau_{\min}\right] \cdot 1_{\{S=5\}}(\underset{\sim}{z}), \tag{5.8}$$

where $\tau_{\min} = 0.5$, and $1_{\{S=5\}}(\underset{\sim}{z})$ is the indicator function for $\underset{\sim}{z}$'s with five clusters. We applied the EMCC algorithm to sample from $g(\underset{\sim}{z}) \triangleq p(\underset{\sim}{z} \mid \boldsymbol{Y})$ from Equation (5.8) using a temperature ladder of length 30, with $\tau_{\max} = t_1 = 60$. We ran all the EMCC schemes 50 times for 10^3 iterations each. We also ran the `K-means` algorithm 50 times with maximum number of iterations 10^4 and random starting values, using the function `kmeans(x,`

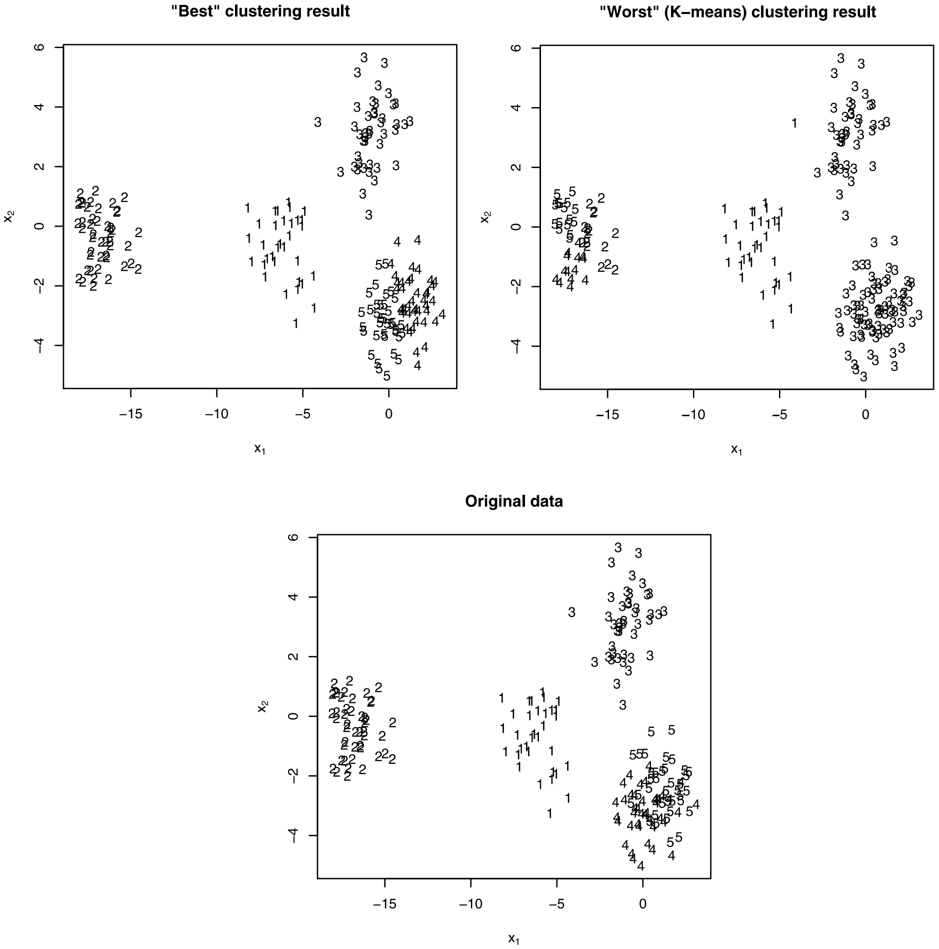

Figure 4. The best-clustering result, the worst-clustering result, and the original data for the objective function based clustering example (Section 5.3).

`centers = 5, iter.max = 10e4`) in R (R 2004). At the end of each of these 50 runs, we collected the minimum value $K(\underset{\sim}{z})$ obtained. We also saved the minimized values of $K(\underset{\sim}{z})$ from 50 runs of each of the six EMCC moves. Table 4 shows the six-point summary of the 50 minimized values for various methods. The number 339.7, which corresponds to the minimum of 50 minimized values of $K(\underset{\sim}{z})$ by all the methods, has been subtracted from all the entries of this table. We have the best-clustering result corresponding to $K(\underset{\sim}{z}) = 339.7$ in Figure 4. We can see from Table 4 that the `K-means` results are heavily right skewed; the plot of the worst-clustering from `K-means` appears in Figure 4. Thus we have shown that even in this extremely simple low-dimensional example, favoring the `K-means` "equal-variance" set-up, `K-means` clustering can fall into the local mode trap pretty often (at least more than 50% of the time), whereas stochastic optimization through the EMCC schemes give much better results.

Table 4. Six-point summary of the 50 minimized values of the $K(\cdot)$ function obtained from 50 runs of K-means and the various EMCC schemes for the objective function based clustering example (Section 5.3). The value 339.7 has been subtracted from the numbers in the table and then they have been rounded off.

	Summary statistics					
Method	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
`K-means`	0	0	0	56.4	40.2	1065.3
`TWO-NEW-b-b`	0	0	0.2	0.1	0.3	0.5
`TWO-NEW-r-r`	0	0	0	0.2	0.3	5.8
`ONE-NEW-b-b`	0	0	0	0.1	0.2	0.6
`ONE-NEW-r-r`	0	0	0.2	0.2	0.3	0.8
`RANDOM-SIZE`	0	0	0	0.1	0.2	0.4
`SAME-SIZE`	0	0	0	0.1	0.2	0.4

5.4 CLUSTERING BASED ON THE MIXTURE NORMAL DISTRIBUTION

Here we generated data $\boldsymbol{Y} = (\boldsymbol{Y}_1, \boldsymbol{Y}_2, \ldots, \boldsymbol{Y}_d)$ from a four-component equally weighted mixture of m-variate normal distributions. The means of the four mixture components ($\boldsymbol{\theta}_s$'s) appear in Table 5. We took $\boldsymbol{\Sigma}_s = \mathrm{AR}_{1m}[0.95]$ for $s = 1, 2$ and $\boldsymbol{\Sigma}_s = 0.2\boldsymbol{I}_m$ for $s = 3, 4$, where $\mathrm{AR}_{1m}[\rho]$ is a $m \times m$ matrix, with (a, b)th entry $\rho^{|a-b|}$. We set $m = 5$ and $d = 200$, and generated 50 datasets from this model; a sample dataset appears in Figure 5. We compared the performance of the model-based clustering tool `MCLUST` (Fraley and Raftery 2002), `K-means`, `MH`, and some of the EMCC moves in this setting. Note that the data generation procedure here adhered to the "ellipsoidal, varying volume, shape, and orientation" set-up of `MCLUST`.

For EMCC we took the standard Bayesian mixture Gaussian approach. More precisely, for given $\underset{\sim}{z}$ with partition representation $\mathcal{H} \triangleq \{H_s\}_{s=1}^{S}$, we let $\boldsymbol{\Sigma}_s \sim \text{Inv-Wishart}_{\nu_0}(\boldsymbol{\Lambda}_0^{-1})$ and $\boldsymbol{\theta}_s \mid \boldsymbol{\Sigma}_s \sim \text{Normal}_m(\boldsymbol{\mu}, \boldsymbol{\Sigma}_s/\kappa_0)$. We also took $\boldsymbol{Y}_u \mid \boldsymbol{\theta}_s \overset{\text{iid}}{\sim} \text{Normal}_m(\boldsymbol{\theta}_s, \boldsymbol{\Sigma}_s)$, $u \in H_s$, which gave the likelihood for $\underset{\sim}{z}$ as follows

$$p(\boldsymbol{Y} \mid \underset{\sim}{z}) = \prod_{s=1}^{S} \left\{ \frac{1}{\pi^{|H_s|m/2}} \left(\frac{\kappa_0}{\kappa_{1s}}\right)^{m/2} \prod_{l=1}^{m} \left\{ \Gamma\left(\frac{\nu_{1s}+1-l}{2}\right) \Big/ \Gamma\left(\frac{\nu_0+1-l}{2}\right) \right\} \times \frac{(\det(\boldsymbol{\Lambda}_0))^{\nu_0/2}}{(\det(\boldsymbol{\Lambda}_{1s}))^{\nu_{1s}/2}} \right\},$$

Table 5. The mixture component weights and means for the mixture normal clustering example (Section 5.4).

	Component	Coordinates				
Weights	means	1	2	3	4	5
0.25	θ_1	−3	0.75	0	0	0
0.25	θ_2	−3	−0.75	0	0	0
0.25	θ_3	3	−1.0	0	0	0
0.25	θ_4	3	1.0	0	0	0

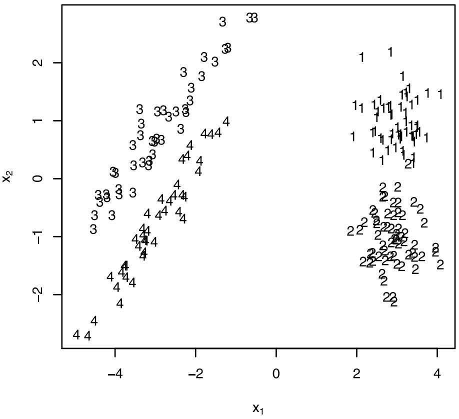

Figure 5. One of the 50 (five-dimensional) datasets used in the mixture normal clustering example (Section 5.4), projected to the first two coordinates.

where $\kappa_{1s} = \kappa_0 + |H_s|$, $\nu_{1s} = \nu_0 + |H_s|$, and $\boldsymbol{\Lambda}_{1s} = \boldsymbol{\Lambda}_0 + \sum_{u \in H_s} (\boldsymbol{Y}_u - \bar{\boldsymbol{Y}}_s)(\boldsymbol{Y}_u - \bar{\boldsymbol{Y}}_s)^T + \frac{|H_s|\kappa_0}{\kappa_{1s}} (\bar{\boldsymbol{Y}}_s - \boldsymbol{\mu})(\bar{\boldsymbol{Y}}_s - \boldsymbol{\mu})^T$, with $\bar{\boldsymbol{Y}}_s = \frac{1}{|H_s|} \sum_{u \in H_s} \boldsymbol{Y}_u$ (Gelman, Carlin, Stern, and Rubin 2004). We set $\nu_0 = 6$, $\kappa_0 = 0.05$, and $\boldsymbol{\Lambda}_0 = \boldsymbol{I}_m$. For EMCC, we considered a temperature ladder of length 30, with $\tau_{\max} = t_1 = 60$, and used $g(\underset{\sim}{z}) \triangleq p(\underset{\sim}{z} \mid \boldsymbol{Y}) \propto p(\boldsymbol{Y} \mid \underset{\sim}{z}) \cdot 1_{\{S=4\}}(\underset{\sim}{z})$. For `K-means` and `MCLUST`, we used the functions `kmeans(x, centers = 4, iter.max = 10e4, nstart = 10e4)` and `Mclust(x, minG = 4, maxG = 4)` in R (R 2004), respectively.

To study the clustering error rates for competing methods, we defined the disagreement between two cluster indicators, $d(\underset{\sim}{z}_1, \underset{\sim}{z}_2) \triangleq \sum_{\substack{u,v=1 \\ u>v}}^{d} \delta(z_{1u}, z_{2v})$, where $\delta(a, b) = 1$, if $a = b$, and 0 otherwise. To compare the performance of `K-means`, `MCLUST`, and EMCC, we computed $d(\underset{\sim}{z}_{\text{opt}}, \underset{\sim}{z}_{\text{truth}})$ for each of the generated 50 datasets, where $\underset{\sim}{z}_{\text{truth}}$ was the cluster indicator used to generate the dataset, and $\underset{\sim}{z}_{\text{opt}}$ was the optimal cluster for `K-means` and `MCLUST`, and the cluster corresponding to the posterior mode for `MH` and EMCC. In each row of Table 6, we recorded the number of $d(\underset{\sim}{z}_{\text{opt}}, \underset{\sim}{z}_{\text{truth}})$ values, as produced by different methods, falling in the range [0, 600) and [600, 6,000). It is easily noted from Table 6 that, not surprisingly, `K-means` performed poorly since the data generation was not in its home ground. The EMCC schemes and also `MH` performed much better than `MCLUST`, although the data-generation process here favored `MCLUST`; this could be partly explained by the fact that unlike `MCLUST`, in `MH` and EMCC, we integrated all the parameters in the model out and performed (stochastic) optimization on the "collapsed" space of the cluster indicators.

5.5 Variable Selection

We cast a variable selection problem into a clustering problem. We take $d = 60$ variables and $n = 100$ observations. We consider the Bayesian information criterion (BIC) as our variable selection criterion. Since exhaustive search in the space of $2^{60} - 1$ models is impossible, we compare the performance of algorithms below by their ability to locate the empirical maximum BIC, not necessarily the global maximum.

Table 6. Summary comparison of the $d(\cdot,\cdot)$ values for various methods for the mixture normal clustering example (Section 5.4). The two numbers in each row represent the number of $d(\cdot,\cdot)$ values falling in the intervals [0, 600) and [600, 6,000), respectively.

Method	$\#\{d(\cdot,\cdot) \in [0, 600)\}$	$\#\{d(\cdot,\cdot) \in [600, 6000)\}$
`K-means`	0	50
`MCLUST`	8	42
`MH`	31	19
`TWO-NEW-b-b`	42	8
`ONE-NEW-b-b`	41	9
`RANDOM-SIZE`	41	9

Let for $u \in \{1 : 20\}$, $\boldsymbol{Y}_{3u-1}, \boldsymbol{Y}_{3u-2} \overset{\text{iid}}{\sim}$ Normal$_n(\boldsymbol{0}, \boldsymbol{I}_n)$, the n-dimensional standard normal distribution. Also, let for $u \in \{1 : 20\}$, $\boldsymbol{Y}_{3u} \overset{\text{iid}}{\sim}$ Normal$_n(\boldsymbol{Y}_{3u-1}, 0.01^2\boldsymbol{I}_n)$. This generation procedure introduces high collinearity among the explanatory variables. We generate the dependent variable according to the model $\boldsymbol{Z} = \boldsymbol{Y}\boldsymbol{\beta} + \boldsymbol{\epsilon}$, where $\boldsymbol{Y} \triangleq [\boldsymbol{Y}_1 : \boldsymbol{Y}_2 : \ldots : \boldsymbol{Y}_d]$ is the matrix of explanatory variables; $\beta_{3u-1} = \beta_{3u-2} = 1$, $\beta_{3u} = 0$ for $u \in \{1 : 20\}$, and $\epsilon \sim$ Normal$_n(\boldsymbol{0}, \boldsymbol{I}_n)$.

We use the cluster indicator $\underset{\sim}{z}$ of length d to represent a regression model, where $z_u = 1$ and 0 indicate the inclusion and exclusion of explanatory variable $\boldsymbol{Y}_u$ in the model, respectively. We do not include an extra intercept term in the model, and hence we do not allow $\underset{\sim}{z} = (0, 0, \ldots, 0)$. Let $\{A_1 \triangleq \{u \mid z_u = 1\}, A_0 \triangleq \{u \mid z_u = 0\}\}$ form the partition representation of $\underset{\sim}{z}$. We define:

$$p(\underset{\sim}{z} \mid \boldsymbol{Y}, \boldsymbol{Z}) \propto \exp\left[-\mathrm{BIC}(\underset{\sim}{z})/\tau_{\min}\right] \cdot 1_{\{A_1 \neq \emptyset\}},$$

where $\tau_{\min} = 0.5$, and $\mathrm{BIC}(\underset{\sim}{z}) = n \cdot \log\left(\|\boldsymbol{Z} - \widehat{\boldsymbol{Z}}_{A_1}\|^2/n\right) + |A_1| \cdot \log(n)$, with $\widehat{\boldsymbol{Z}}_{A_1}$ being the (ordinary least square) prediction vector of $\boldsymbol{Z}$ based on $\{\boldsymbol{Y}_u,\ u \in A_1\}$; that is, all the explanatory variables in the model. Clustering based on the two clusters A_1 and A_0 requires tweaking the SCRC scheme in the following way; the two SCSC crossover schemes are not effective here. We select the two parents $\underset{\sim}{x}_i$ and $\underset{\sim}{x}_j$ exactly the same way as was done in Section 3.3, and take $\{k_1, k_2\} = \{l_1, l_2\} = \{0, 1\}$, so that $A_{k_i} \cap B_{l_i} \neq \emptyset, i = 1, 2$. If such a choice of $k_i, l_i, i = 1, 2$ is not possible, then this crossover cannot be performed. We take $H \triangleq \left(A_{k_1} \cap B_{l_1}\right) \cup \left(A_{k_2} \cap B_{l_2}\right)$ and proceed with the reallocation procedure of Section 3.5, to produce H_1 and H_2, with $\{h_1, h_2\} = \{m_1, m_2\}$. We obtain the children as $(\underset{\sim}{y}_i, \underset{\sim}{y}_j) = $ SCShuffle$((\underset{\sim}{x}_i, \underset{\sim}{x}_j); (k_1, l_1), H_1; (k_2, l_2), H_2)$, and replace the parents by their children with probability:

$$r = \frac{f_i(\underset{\sim}{y}_i)\, f_j(\underset{\sim}{y}_j)}{f(\underset{\sim}{x}_i)\, f_j(\underset{\sim}{x}_j)} \times \frac{T_{i,j}(\mathbf{y}, \mathbf{x})}{T_{i,j}(\mathbf{x}, \mathbf{y})}. \tag{5.9}$$

Here $T_{i,j}(\cdot,\cdot)$ has an expression similar to that of Section 3.3 and we omit the details to avoid repetition. We compared the performance of EMCC using this modified version of SCRC scheme and $\tau_{\max} = 16$ with the `MH` scheme. We ran the two algorithms on 25 randomly generated datasets from the data-generation model introduced earlier in this section,

Table 7. The first two rows show the six-point summary of the minimum BICs achieved by the MH scheme and the modified SCRC scheme on 25 randomly generated data sets, respectively, for the variable selection example (Section 5.5). The row labeled diff is the summary of the differences of the minimum BICs achieved by the MH scheme from those achieved by the modified SCRC scheme.

	Summary statistics					
Scheme	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
MH	154.40	119.30	113.00	113.80	104.30	88.11
mod-SCRC	154.30	119.00	112.70	113.70	104.30	87.94
diff	−0.29	−0.25	−0.18	−0.17	−0.09	0.00

for the same amount of computing time. EMCC found lower minimum BICs than MH in all the 25 runs; MH was able to match EMCC in only one of those runs. A comparison of the minimum BICs achieved by the two methods is given in Table 7. It is worth mentioning that the MH scheme, as described in this article, is a powerful method in itself; it was able to reach the minimum BIC level that could be achieved by PT or EMCC in all the variable selection examples (other than the present) we tried (e.g., Liang and Wong 2000, example 4.2).

6. DISCUSSION

This article demonstrates that the EMCC algorithm is an effective tool both for sampling clusters from the space of clustering solutions in high-dimensional settings and for finding the optimal clustering solution based on a given objective function. Moreover, the EMCC algorithm can be applied to any problem that can be cast into a cluster-sampling framework (e.g., the variable selection problem of Section 5.5).

The intuition behind the EMCC moves has an appealing justification. We choose two parent chromosomes either randomly or with probability proportional to (a function of) their fitness value. We take their “vote”; that is, we consider the subcluster intersections of the two parents, and randomly swap (in SCSC) or reshuffle (in SCRC) two of these intersections and thus form the child(ren). This way of producing child(ren) only perturbs the internal structure of the parents with respect to only two clusters and hence this process respects what the parents jointly have to say about the structure of the other (unperturbed) clusters, which ensures that good parents produce good children.

Note that if a SCSC or SCRC move is accepted, then we are able to change more than one coordinate of the parent chromosome(s) at once, which is not possible in Gibbs or “split-merge” MH (Section 3.1) sampling where only one coordinate of $\underset{\sim}{z}$ is proposed to be updated at a time. The flavor of “split-merge” MH move that can be found in Jain and Neal (2004) is different from ours. In the Jain–Neal “split-merge” method, one first chooses two coordinates of $\underset{\sim}{z}$. If these happen to lie in the same mother cluster, then the mother cluster is split into two child clusters containing those two coordinates. In case that the two chosen coordinates lie in different clusters, these clusters are merged. This procedure proposes drastic changes to the system by changing potentially a lot of coordinates of $\underset{\sim}{z}$ at a time;

and thus this is the other extreme of a Gibbs move. The SCSC and the SCRC moves take a somewhat middle ground between Gibbs on one end and the Jain–Neal "split-merge" sampler on the other.

The intuition behind introducing the SCRC moves is that it is not as drastic as the SCSC family of moves. Instead of swapping two subcluster intersections, as required in the SCSC:ONE-NEW move, in SCRC, we reshuffle their members randomly which understandably does not perturb the structure of the nonsurviving parent as much. In fact, the SCSC:ONE-NEW move can be considered as a special case of the SCRC move where the reshuffling is nothing but a deterministic swapping. SCSC:TWO-NEW is the most drastic in that it perturbs both the parents.

Note the EMCC schemes do not alter the number of clusters present in the parents in the process of producing child(ren). One might argue that discovery of new modes using the EMCC schemes may not be possible in difficult scenarios where the whole of $Z \triangleq \{1 : d\}^d$ needs to be searched quite rapidly. This drawback of the EMCC schemes is also a plus for problems where the number of clusters is always fixed (e.g., Sections 5.3, 5.5); in contrast, MH "split-merge," Gibbs, and the Jain–Neal "split-merge" method can produce invalid proposals.

The equi-energy sampler (EE; Kou, Zhou, and Wong 2006) is a powerful technique that does not fall in the category of population-based methods. The strength of the EE sampler lies in the *equi-energy* jump step where samples from different *energy rings* are proposed to be exchanged. Unfortunately, as of now, the set up of the EE sampler does not allow for any kind of crossover moves. The new crossover moves introduced here are complementary to and not substitutes for the EE sampler. We hope the ideas introduced in this article will enable discovery of similar crossover moves in the context of the EE sampler in the future.

We use AIAT as a measure of how well a sampler moves around the space; use of IAT for the same purpose can be found in Jain and Neal (2004) and Dahl (2003). However, unless we know from a different source that a sampler is visiting all the modes (which is almost always impossible) in a certain problem, the use of IATs might be misleading, since the sampler may move freely within a local mode producing low IATs whereas a "better" sampler which infrequently jumps around modes might produce an abnormally high IAT. On similar grounds, the use of AMLD as a measure of how well the sampler escapes from local modes is questionable. AMLD is a sound measure for stochastic optimization, but for sampling problems it portrays only a part of a grand picture.

7. IMPLEMENTATION

All the simulations in this article were done using C code interweaved with underlying random number generation C code of R (R 2004). The analysis of the simulation output was done using R. The first author is in the process of publishing the code as an R package called `EMCC`. Please check the following Web sites for release announcement and updates:

- *http://www.people.fas.harvard.edu/~junliu/*
- *http://www.r-project.org/*

REFERENCES

Blackwell, D,. and MacQueen, J. B. (1973), "Ferguson Distributions via Polya Urn Schemes," *The Annals of Statistics*, 1, 353–355.

Dahl, D. B. (2003), "An Improved Merge-Split Sampler for Conjugate Dirichlet Process Mixture Models," Technical report, Department of Statistics, University of Wisconsin.

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977), "Maximum Likelihood From Incomplete Data via the EM Algorithm," *Journal of the Royal Statistical Society*, Series B, 39, 1–22.

Ferguson, T. S. (1973), "A Bayesian Analysis of Some Nonparametric Problems," *The Annals of Statistics*, 1, 209–230.

——— (1974), "Prior Distributions on Spaces of Probability Measures," *The Annals of Statistics*, 2, 615–629.

Fraley, C., and Raftery, A. E. (2002), "Model-Based Clustering, Discriminant Analysis, and Density Estimation," *Journal of the American Statistical Association*, 97, 611–631.

Gelfand, A. E., and Smith, A. F. M. (1990), "Sampling-Based Approaches to Calculating Marginal Densities," *Journal of the American Statistical Association*, 85, 398–409.

Gelman, A., Carlin, J. B., Rubin, D. B., and Stern, H. S. (2004), *Bayesian Data Analysis*, New York: Chapman & Hall/CRC Press.

Geyer, C. J. (1991), "Markov Chain Monte Carlo Maximum Likelihood," in *Computing Science and Statistics. Proceedings of the 23rd Symposium on the Interface*, Fairfax, VA: Interface Foundation of North America, pp. 156–163.

——— (1994), "Practical Markov Chain Monte Carlo" (with discussion), *Statistical Science*, 7, 473–483.

Gilks, W. R., Roberts, G. O., and George, E. I. (1994), "Adaptive Direction Sampling," *The Statistician*, 43, 179–189.

Goswami, G., and Liu, J. S. (2007), "On Learning Strategies for Evolutionary Monte Carlo," *Statistics and Computing (electronic version is already published, to appear in print)*.

Hastie, T., Tibshirani, R., and Friedman, J. H. (2001), *The Elements of Statistical Learning: Data Mining, Inference, and Prediction: with 200 Full-color Illustrations*. New York: Springer-Verlag Inc.

Hastings, W. K. (1970), "Monte Carlo Sampling Methods Using Markov Chains and Their Applications," *Biometrika*, 57, 97–109.

Jain, S., and Neal, R. M. (2004), "A Split-Merge Markov Chain Monte Carlo Procedure for the Dirichlet Process Mixture Model," *Journal of Computational and Graphical Statistics*, 13, 158–182.

Jensen, S. T., and Liu, J. S. (2007), "Bayesian Clustering of Transcription Factor Binding Motifs," *Journal of the American Statistical Association*, (to appear).

Kou, S. S., Zhou, C. Q., and Wong, W. H. (2006), "Equi-Energy Sampler with Applications in Statistical Inference and Statistical Mechanics" (with discussion), *The Annals of Statistics*, 34, 1581–1652.

Liang, F., and Wong, W. H. (2000), "Evolutionary Monte Carlo: Applications to C_P Model Sampling and Change Point Problem," *Statistica Sinica*, 10, 317–342.

——— (2001), "Real-Parameter Evolutionary Monte Carlo with Applications to Bayesian Mixture Models," *Journal of the American Statistical Association*, 96, 653–666.

Liu, J. S. (1996), "Nonparametric Hierarchical Bayes via Sequential Imputations," *The Annals of Statistics*, 24, 911–930.

——— (2001), *Monte Carlo Strategies in Scientific Computing*, New York: Springer-Verlag Inc.

Liu, J. S., Liang, F., and Wong, W. H. (2000), "The Multiple-Try Method and Local Optimization in Metropolis Sampling," *Journal of the American Statistical Association*, 95, 121–134.

R Development Core Team (2004), *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria.

Roberts, G. O., and Gilks, W. R. (1994), "Convergence of Adaptive Direction Sampling," *Journal of Multivariate Analysis*, 49, 287–298.