On Potts Model Clustering, Kernel *K*-Means, and Density Estimation

Alejandro MURUA, Larissa STANBERRY, and Werner STUETZLE

Many clustering methods, such as K-means, kernel K-means, and MNcut clustering, follow the same recipe: (i) choose a measure of similarity between observations; (ii) define a figure of merit assigning a large value to partitions of the data that put similar observations in the same cluster; and (iii) optimize this figure of merit over partitions. Potts model clustering represents an interesting variation on this recipe. Blatt, Wiseman, and Domany defined a new figure of merit for partitions that is formally similar to the Hamiltonian of the Potts model for ferromagnetism, extensively studied in statistical physics. For each temperature T, the Hamiltonian defines a distribution assigning a probability to each possible configuration of the physical system or, in the language of clustering, to each partition. Instead of searching for a single partition optimizing the Hamiltonian, they sampled a large number of partitions from this distribution for a range of temperatures. They proposed a heuristic for choosing an appropriate temperature and from the sample of partitions associated with this chosen temperature, they then derived what we call a *consensus clustering*: two observations are put in the same consensus cluster if they belong to the same cluster in the majority of the random partitions. In a sense, the consensus clustering is an "average" of plausible configurations, and we would expect it to be more stable (over different samples) than the configuration optimizing the Hamiltonian.

The goal of this article is to contribute to the understanding of Potts model clustering and to propose extensions and improvements: (1) We show that the Hamiltonian used in Potts model clustering is closely related to the kernel K-means and MNCut criteria. (2) We propose a modification of the Hamiltonian penalizing unequal cluster sizes and show that it can be interpreted as a weighted version of the kernel K-means criterion. (3) We introduce a new version of the Wolff algorithm to simulate configurations from the distribution defined by the penalized Hamiltonian, leading to penalized Potts model clustering. (4) We note a link between kernel based clustering methods and nonparametric density estimation and exploit it to automatically determine locally adaptive kernel bandwidths. (5) We propose a new simple rule for selecting a good temperature T.

As an illustration we apply Potts model clustering to gene expression data and compare our results to those obtained by model based clustering and a nonparametric dendrogram sharpening method.

Key Words: Consensus clusters; Gene expression; Monte Carlo; Multiway normalized cut; Penalized Wolff algorithm; Superparamagnetic method.

Alejandro Murua is Associate Professor, Département de Mathématiques et de Statistique, Université de Montréal, Canada (E-mail: *murua@dms.umontreal.ca*). Larissa Stanberry is a Graduate Student, and Werner Stuetzle is Professor, Department of Statistics, University of Washington, Seattle, WA.

Journal of Computational and Graphical Statistics, Volume 17, Number 3, Pages 629–658
DOI: 10.1198/106186008X318855

1. INTRODUCTION

The goal of clustering is to identify distinct groups in a dataset and assign a group label to each observation. Clustering is a common problem in emerging fields such as bioinformatics and text mining. In a typical bioinformatics application we may have microarray data measuring the expression levels of thousands of genes for the same organism under different experimental conditions. Genes with similar expression patterns across experiments may have related functions. Clustering of genes can also be a first step toward modeling and understanding gene regulatory networks (Eisen et al. 1998). In text mining, the goal of clustering may be to partition a collection of documents, such as Web pages returned by a search engine, into subsets representing different topics (Tantrum, Murua, and Stuetzle 2003, 2004).

One of the most popular clustering algorithms is K-means. Let $x_i \in \mathbf{R}^d$, $i = 1, \ldots, n$ be our data. Suppose we want to partition the data into q clusters. Let $z_{ki} = 1$ if x_i belongs to the kth cluster, and zero, otherwise. K-means finds cluster centers $\{m_k\}_{k=1}^q$ and cluster memberships z_{ki} by minimizing $\sum_{k=1}^q \sum_{i=1}^n z_{ki}(x_i - m_k)^t(x_i - m_k)/n$. This is equivalent to maximizing $\sum_{i=1}^n \sum_{j=1}^n (<x_i, x_j> /n) \sum_{k=1}^q z_{ki}z_{kj}/n_k$, where n_k is the number of data points forming the kth cluster, $k = 1, \ldots, q$, and $<\cdot, \cdot>$ denotes the inner product in $\mathbf{R}^d$. Define weights $w(i, j, \{z_{ki}\}) = \sum_{k=1}^q z_{ki}z_{kj}/n_k$. The weight $w(i, j, \{z_{ki}\})$ is $1/n_{k'}$ if x_i and x_j share the same label k', and it is zero, otherwise. Using this new notation, the K-means figure of merit is

$$\frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{n} w(i, j, \{z_{ki}\}) \ <x_i, x_j> . \tag{1.1}$$

One can see that (a) K-means penalizes the assignment of the same label to dissimilar data points ($<x_i, x_j> \ < \ 0$); (b) K-means favors the assignment of the same label to very similar points (large $<x_i, x_j>$); and that (c) the effect of the weights is in part to try to assign data points that are not very similar, but still similar ($<x_i, x_j> \ > \ 0$), to small clusters (small n_k). The K-means criterion (1.1) can be generalized by modifying the weights $w(i, j, \{z_{ki}\})$, replacing $<x_i, x_j>$ with a more general similarity measure $s(x_i, x_j)$, or both. The criterion (1.1) then becomes

$$\frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{n} w(i, j, \{z_{ki}\})\, s(x_i, x_j)\,. \tag{1.2}$$

We show in Section 2.1 that choosing a similarity measure derived from a Mercer kernel, that is, $s(x_i, x_j) = k(x_i, x_j)$ for some square-integrable symmetric positive function $k : \mathbf{R}^2 \rightarrow [0, +\infty)$, leads to the kernel K-means criterion (Girolami 2002). An additional modification of the weights results in the Multiway Normalized Cut (MNCut) criterion (see the Appendix). The figure of merit proposed by Blatt, Wiseman, and Domany (1996a,b, 1997) in their articles introducing what we call *Potts model clustering* fits into this framework by choosing weights $\delta_{ij} = \sum z_{ki}\, z_{kj}$, leading to the criterion

$$\frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{n} s(x_i,\, x_j)\, \delta_{ij}\,. \tag{1.3}$$

The similarity $s(x_i, x_j)$ between observations i and j receives the weight one if they are assigned to the same cluster, and the weight zero, otherwise, independent of the cluster sizes. Hence, unlike K-means (which by (a) above, penalizes the assignment of the same label to dissimilar data points), the criterion given by Equation (1.3) favors the assignment of the same label to similar points. A procedure based on this criterion is able to deal with nonspherical shapes and may be able to assign the same label to objects that are far apart (e.g., the extremes of a snake-shaped cluster) which is highly unlikely to occur in K-means. Maximizing (1.3) is equivalent to minimizing

$$H(\{z_{ki}\}) = \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} s(x_i,\, x_j) - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} s(x_i,\, x_j)\, \delta_{ij} = \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} (1-\delta_{ij})\, s(x_i,\, x_j)\,. \tag{1.4}$$

The function $H(\{z_{ki}\})$ is just another criterion measuring the quality of clustering, and one could simply find the cluster memberships $\{z_{ki}\}$ minimizing H. However, Blatt et al. (1996a,b, 1997) pursued a different approach. They pointed out that $H(\{z_{ki}\})$ has a physical interpretation when $s(\cdot, \cdot)$ is positive and symmetric: it corresponds to the Hamiltonian (Sokal 1996, Sect. 6) of a Potts model for describing ferromagnetism phenomena. The Potts model is a probabilistic model of the system formed by the particles (data points), and their *interactions* given by the similarity measure. The distribution of the system depends on the temperature T. For each T there is a probability $p_T(\{z_{ki}\})$ associated with each configuration of the system's labels

$$p_T(\{z_{ki}\}) \propto \exp\left\{-\frac{1}{T} H(\{z_{ki}\})\right\} = \exp\left\{-\frac{1}{2\,T}\sum_{i=1}^{n}\sum_{j=1}^{n} (1-\delta_{ij})\, s(x_i,\, x_j)\right\}. \tag{1.5}$$

Note that the Potts model gives low probability to configurations assigning different labels to similar observations. The maximum probability is achieved when all the observations are assigned the same label (e.g., single cluster). In contrast, K-means achieves its minimum when each observation is assigned its own label (e.g., n singleton clusters). This is another difference between the two criteria (1.3) and (1.1). The key to understand these differences lies in the similarity measure $s(x_i, x_j)$. If $s(x_i, x_j)$ is always positive, then the criterion does not penalize assigning the same label to dissimilar observations. This is the case of the Potts model but not of K-means (since $s(x_i, x_j) = <x_i,\, x_j>$ may be negative).

Blatt et al. (1996a,b, 1997) first simulated a large number M of configurations $\{z_{ki}\}$ according to the distribution (1.5) for a range of temperatures. This can be done efficiently using the Swendsen–Wang Markov chain Monte Carlo (MCMC) algorithm (Swendsen and Wang 1987; Wang and Swendsen 1990). They proposed a heuristic for choosing an appropriate temperature. In a second step they then extracted what we call a *consensus clustering* from the M configurations associated with the chosen temperature. The consensus clustering assigns two observations to the same cluster if they belong to the same cluster in the majority of the randomly generated configurations. The consensus clusters are the connected components of the graph over the observations with an edge between any pair belonging to the same cluster in the majority of the configurations.

In a sense, the consensus clustering is an "average" of plausible configurations, and we would expect it to be more stable (over different samples) than the configuration minimizing H. There is abundant statistical and machine learning literature that exploits the idea of combining several partitions (Dimitriadou, Weingessel, and Hornik 2001; Dudoit and Fridlyand 2003; Fern and Brodley 2003; Fred and Jain 2002; Topchy, Jain, and Punch 2005). A great advantage of consensus clustering is that there is no need to specify the number of clusters in the data before starting a search. The number of clusters in a random configuration is itself random and governed by the distribution (1.5), and forming the consensus does not require any parameters—the clusters and their number are estimated simultaneously. Another advantage is that this approach avoids the combinatorial search for the configuration optimizing H. We also noticed in experiments where the true group structure of the data was known that the consensus clustering tends to be closer to the truth than the clustering found by optimizing the figure of merit.

Potts model clustering, also known as the superparamagnetic clustering method, has been a subject of intensive research since its introduction by Blatt et al. (1996). The physical aspects of the method and its dependence on the definition of the neighbors, type of interactions, number of possible states, and size of the dataset have been studied by Wiseman, Blatt, and Domany (1998), and by Agrawal and Domany (2003). Ott et al. (2004) introduced a sequential extension to deal with inhomogeneities in shape, density, and size of clusters. Reichardt and Bornholdt (2004) introduced a spin glass Hamiltonian with a global diversity constraint to identify probable community assignments in complex networks. Stanberry, Murua, and Cordes (2007) applied the method to study functional connectivity patterns in fMRI data and examined the dependence of the method on neighborhood structure, signal-to-noise ratio, and spatial dependence in the data. Potts model clustering has been applied to different fields such as computer vision (Domany et al. 1999), gene expression data (Getz et al. 2000; Domany 2003; Einav et al. 2005), high-dimensional chemical data (Ott et al. 2004, 2005) and neuronal spike detection (Quiroga, Nadasdy, and Ben-Shaul 2004).

The objective of this article is to improve and extend Potts model clustering based on statistical methodology and machine learning techniques. More specifically, (1) we show that the Hamiltonian used in Potts model clustering is related to the kernel K-means and MNCut criteria. All three criteria are weighted averages of interpoint similarities. The weights and the similarities differentiate the methods (see Section 2 and the Appendix). (2) We propose a modification of the Hamiltonian, penalizing unequal cluster sizes, and show that it can be interpreted as a weighted version of the kernel K-means criterion (see Section 3). (3) We introduce a new version of the Wolff algorithm (Wolff 1989) to simulate configurations from the distribution defined by the penalized Hamiltonian, leading to a penalized Potts model clustering (see Section 3.3). (4) We note a link between kernel-based methods and nonparametric density estimation and exploit it to automatically determine kernel bandwidths. While most kernel-based clustering methods, including Blatt, Wiseman, and Domany's version of Potts model clustering, use kernels with fixed, predetermined bandwidth over the entire feature space, our approach produces adaptive bandwidths (Abramson 1982; Silverman 1986) (see Section 4). (5) We propose a simple rule to select a good temperature T. Our rule is based on monitoring a series of cluster splitting measures that follow the trajectories over temperature of the cluster sizes. We measure similarity among the clustering partitions generated within and across temperatures by the adjusted Rand index (Hubert and Arabie 1985) and its variance. Small variances are indicators of stable partitions and hence, possible good partitions. Relevant cluster splitting is also measured through the variation in the upper tail of the distribution of the cluster sizes. The rule proposed by Blatt et al. (1996a,b), namely, the variance of the size of the largest cluster, is a special case of our rule. Our experiments in Section 6 show that our rule performs well.

We apply our proposed Potts model clustering methodology to gene expression data and compare our results to those obtained by model-based clustering (Banfield and Raftery 1993; Celeux and Govaert 1995), and the hierarchical clustering with dendrogram sharpening method introduced by McKinney (1995). The former has been shown to perform moderately well for gene expression data (Yeung et al. 2001) when the clustering is done over the genes. However, in many situations the clustering of interest is on the subjects; for example, being able to differentiate among several subtypes of cancer in order to deliver the optimal treatment. In this case, the data are high-dimensional, with dimensions on the order of 10^4 genes. Potts model clustering is suitable for this kind of data since the clustering does not depend on the data dimension, but only on the similarities between the data points, and their spatial arrangement. In general, Gaussian model-based clustering cannot directly be applied to this type of data, since one would need many more patients than genes in order to estimate the cluster parameters. Throughout our experiments we have observed that Potts model clustering suggested an appropriate number of clusters for the data.

The remainder of the article is organized as follows. In Section 2 we describe kernel K-means and its connection to Potts model clustering. In Section 3 we study the distribution of labels for different variants of the Potts model and introduce the penalized Wolff algorithm. Section 4 deals with the connection between kernel-based methods and kernel density estimation and introduces methods for adaptive bandwidth selection. In Section 5 we address the problem of temperature selection for Potts model clustering. In Section 6 we present the results of a simulation performed with the goal of shedding some light on the performance of Potts model clustering and our suggested procedure to select an appropriate temperature. In this section we also apply Potts model clustering to microarray data and illustrate our method for adaptive kernel bandwidth selection. Section 7 contains a discussion and some ideas for future work.

2. CONNECTIONS BETWEEN KERNEL K-MEANS AND POTTS MODEL CLUSTERING

2.1 KERNEL K-MEANS

Instead of working directly with the original feature data vectors x_i's, one could work with a suitable transformation of them, say $\Phi : \mathbf{R}^d \to \mathbf{H}$ where, in general, $\mathbf{H}$ is a higher dimensional (and possible infinite-dimensional) Hilbert space. K-means in this new feature space $\mathbf{H}$ corresponds to finding z_{ki}'s and μ_k's that minimize

$$\frac{1}{n}\sum_{k=1}^{q}\sum_{i=1}^{n} z_{ki} D(\Phi(x_i), \mu_k), \tag{2.1}$$

where $D(\cdot,\cdot)$ denotes the distance in $\mathbf{H}$. The mean estimates are given by $\hat{\mu}_k = n_k^{-1}\sum_{i=1}^{n}\hat{z}_{ki}\Phi(x_i)$, $k = 1,\ldots,q$. Let $<\cdot,\cdot>$ denote the inner product in $\mathbf{H}$. Note that

$$\begin{aligned} D(\Phi(x_i), \hat{\mu}_k) &= \ < \Phi(x_i) - \hat{\mu}_k,\ \Phi(x_i) - \hat{\mu}_k > \\ &= \ < \Phi(x_i), \Phi(x_i) > - < \Phi(x_i), \hat{\mu}_k > \\ &\quad - < \hat{\mu}_k, \Phi(x_i) > + < \hat{\mu}_k, \hat{\mu}_k > . \end{aligned}$$

Assume that there exists a kernel function in $\mathbf{R}^d \times \mathbf{R}^d$ for which the inner product in $\mathbf{H}$ can be expressed as $< \Phi(x_i), \Phi(x_j) >= k(x_i, x_j)$. In this case K-means does not need to know explicitly the transformation $\Phi(\cdot)$. It only needs to know the kernel $k(\cdot,\cdot)$. This is the well-known kernel K-means method (Girolami 2002; Zhang and Rudnicky 2002; Dhillon, Guan, and Kulis 2004).

Girolami (2002) showed that Equation (2.1) can be written as

$$\frac{1}{n}\sum_{k=1}^{q}\sum_{i=1}^{n} z_{ki} k_{ii} - \sum_{k=1}^{q}\gamma_k R_k, \tag{2.2}$$

where $k_{ij} = k(x_i, x_j)$, $\gamma_k = n_k/n$ is the proportion of data points falling in cluster k, and $R_k = n_k^{-2}\sum_{i=1}^{n}\sum_{j=1}^{n} z_{ki}z_{kj}k_{ij}$, $i, j = 1,\ldots,n$, $k = 1,\ldots,q$. Since the first term in (2.2) does not depend on the label assignments (note that $\sum_{k=1}^{q}\sum_{i=1}^{n} z_{ki}k_{ii} = \sum_{i=1}^{n}(\sum_{k=1}^{q} z_{ki})k_{ii} = \sum_{i=1}^{n} k_{ii}$), minimizing (2.1) is equivalent to maximizing

$$\sum_{k=1}^{q}\gamma_k R_k = \frac{1}{n}\sum_{i=1}^{n}\sum_{k=1}^{q} z_{ki}\frac{1}{n_k}\sum_{j=1}^{n} z_{kj}k_{ij} = \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{n} k_{ij}\sum_{k=1}^{q} z_{ki}z_{kj}\frac{1}{n_k}, \tag{2.3}$$

which is exactly the criterion given by (1.2) with the weights of the K-means method. If we assume that $k(x_i, x_j) = k(x_i - x_j)$ for all i, j (e.g., Gaussian kernel), then $\hat{p}(x_i|k) = \frac{1}{n_k}\sum_{j=1}^{n} z_{kj}k_{ij}$ can be seen as a nonparametric estimate of the conditional density score associated with observing x_i given cluster k, $p(x_i|k)$ (Silverman 1986). From now on we will assume that the kernel $k(\cdot,\cdot)$ is of this form. Therefore (2.3) can be interpreted as an average of these conditional density scores, and the goal of kernel K-means in this case is to maximize this average. Girolami (2002) gave a different interpretation to (2.3). In his view, each R_k provides a measure of compactness of the corresponding kth cluster, $k = 1, \dots, q$. This is derived from the convolution (reproductive-kernel) property of the Gaussian kernel:

$$\begin{aligned}\int_{\text{cluster } k} p(x|k)^2\,dx &\approx \int p(x|k)^2\,dx \\ &\approx \int \Big(\frac{1}{n_k}\sum_{i=1}^{n} z_{ki}k(x-x_i)\Big)^2\,dx = \frac{1}{n_k^2}\sum_{i=1}^{n}\sum_{j=1}^{n} z_{ki}z_{kj}k_{ij} = R_k.\end{aligned}$$

2.2 WEIGHTED KERNEL *K*-MEANS

On the other hand, $\gamma_k\,\hat{p}(x_i|k)$ can be seen as an estimate of the density score associated with observing x_i in cluster k. Hence,

$$\sum_{k=1}^{q} \gamma_k^2 R_k = \frac{1}{n}\sum_{i=1}^{n}\sum_{k=1}^{q} z_{ki}\gamma_k \frac{1}{n_k}\sum_{j=1}^{n} z_{kj}k_{ij} = \frac{1}{n}\sum_{i=1}^{n}\sum_{k=1}^{q} z_{ki}\gamma_k\hat{p}(x_i|k) \tag{2.4}$$

can be interpreted as an average of the density scores associated with observing the data points in the corresponding clusters. This slight modification of K-means leads to a weighted K-means approach that penalizes the distribution of the cluster sizes. Consider weights given by the γ_k's, and the minimization of

$$\frac{1}{n}\sum_{k=1}^{q}\gamma_k\sum_{i=1}^{n} z_{ki}\,D(\Phi(x_i), \mu_k). \tag{2.5}$$

A straightforward computation leads to the maximization of

$$\sum_{k=1}^{q}\gamma_k^2 R_k - \sum_{k=1}^{q}\gamma_k^2. \tag{2.6}$$

The role of the last term is to penalize the nonuniform distribution of the cluster sizes, that is, to avoid clusters that are too large or too small. In the next section we show that the criterion given by Equation (2.4) is connected to Potts model clustering. Moreover, we also show that (2.6) is connected to a modified (penalized) version of Potts model clustering.

2.3 POTTS MODEL CLUSTERING

Without loss of generality, assume that the observations are the vertices of a graph. So far we have worked with a complete graph (i.e., all graph nodes are connected). In many practical situations (e.g., images) it may be convenient to work with a reduced graph. For example one can build a K-nearest-neighbor graph such that for each point x_i there is an edge between x_i and its K nearest neighbors. If the K-nearest-neighbor graph contains more than one connected set, then the graph can be augmented by adding edges of the minimum-spanning graph, so that there is a path from any point to any other point in the resulting graph.

Let $\alpha_{ij} = 1$ if $i \neq j$ and the points x_i and x_j are neighbors in the graph (i.e., there is an edge connecting these points), and zero, otherwise. A sensible clustering criterion is to penalize different labels between neighboring points. This leads to the minimization of (compare with (1.4))

$$H(\{z_{ki}\}) = \frac{1}{2}\sum_{i=1}^{n}\sum_{k=1}^{q}\sum_{j=1}^{n} z_{ki}(1-z_{kj})k_{ij}\alpha_{ij} = \sum_{\alpha_{ij}=1} k_{ij}(1-\delta_{ij}). \tag{2.7}$$

Equation (2.7) is the Potts model criterion on a graph. It represents the Hamiltonian (Sokal 1996, Sect. 6) of the system, which has log-density equal to minus this quantity. If the graph is a K-nearest-neighbor graph, the Hamiltonian only involves $O(n)$ terms, whereas for the complete graph in (1.4) it involves $O(n^2)$ terms. Thus, it is computationally advantageous to work with Potts models on graphs. Although, in general, the graph depends on the interpoint distances themselves, in many interesting situations, such as in images, the graph neighborhood relationship is an intrinsic property of the data. Moreover, as seen in (2.8) below, working on a graph corresponds to multiplying the weights $w(i, j, \{z_{ki}\})$ by α_{ij}. This holds for every method based on (1.2) not just for the Potts model clustering.

A trivial calculation shows that $H(\{z_{ki}\}) = \text{constant} - \frac{1}{2}\sum_{j=1}^{n}\sum_{i=1}^{n}\alpha_{ij}k_{ij}\delta_{ij}$. The constant in the right-hand side is independent of the labels. Therefore, maximizing the likelihood of the Potts model (i.e., minimizing (2.7)), excluding the trivial all-in-one cluster solution, is equivalent to maximizing

$$\frac{1}{2}\sum_{j=1}^{n}\sum_{i=1}^{n}\alpha_{ij}k_{ij}\delta_{ij} = \frac{1}{2}\sum_{j=1}^{n}\sum_{k=1}^{q}\sum_{i=1}^{n} z_{ki}z_{kj}k_{ij}\alpha_{ij}. \tag{2.8}$$

Note that $\hat{p}_{\text{potts}}(x_i|k) = n_k^{-1}\sum_{j=1}^{n} z_{kj}k_{ij}\alpha_{ij}$ is an estimate of $p(x_i|k)$. We can rewrite (2.8) as

$$\frac{n}{2}\sum_{i=1}^{n}\sum_{k=1}^{q} z_{ki} \;\; \gamma_k \; \hat{p}_{\text{potts}}\,(x_i|k). \tag{2.9}$$

Hence, Equation (2.8) is equivalent to Equation (2.4) and it can be interpreted in the same manner.

2.4 THE CONNECTION WITH WEIGHTED K-MEANS

Adding the term $(n^2/2)\sum_{k=1}^{q}\gamma_k^2$ to the expression in (2.8) leads to an expression similar to (2.6), derived from the weighted K-means criterion. We refer to this latter model as the penalized Potts model. As in weighted K-means, the distribution of the cluster sizes are shrunk towards the uniform distribution. It is easy to see (see Section 3.3) that both criteria are exactly the same for the complete graph (i.e., $\alpha_{ij} = 1$ for all $i, j = 1, \ldots, n$). When the graph is a reduced graph (e.g., K-nearest-neighbor graph) the criteria differ. From a computational point of view, it is advantageous to use small neighborhoods with penalized Potts model clustering. In Sections 3.3 and 3.4, we develop an extended Potts model and a "penalized" Wolff algorithm with the aim of optimizing this criterion.

3. SIMULATING THE LABELING DISTRIBUTION

A great advantage of the Potts model clustering method over other kernel-based clustering methods is that it can estimate the clusters and their number simultaneously. Cluster membership is based on the proportion of times that any two observations are assigned to the same cluster. These proportions are estimated using MCMC techniques such as the Swendsen–Wang algorithm (Swendsen and Wang 1987), or the Wolff algorithm (Wolff 1989). For completeness, we briefly outline the procedures here.

3.1 THE SWENDSEN–WANG AND WOLFF ALGORITHMS

Perhaps the simplest way to generate samples from the Potts model is through a Gibbs sampler (Grenander 1983; Geman and Geman 1984), also known in physics as the heat bath algorithm (Creutz 1979), on the labels $\{z_{ki}\}$. This reduces to finding the full conditionals of each vector $(z_{1i}, z_{2i}, \ldots, z_{qi})$ given the current values of the remaining labels for $j \neq i$, for $i = 1, \ldots, n$. Although the conditionals are easy to obtain and work with, the sampling is rather inefficient. For example, to assign two points, say x_i and x_j, to the same label may take a full sweep of the data, let alone assigning several points to the same updated label. Fortunately, there exists a very efficient way to generate samples from the Potts model by model augmentation.

Let $p_{ij} = 1 - \exp\{-k_{ij}\}$. The Potts model density is given by

$$p(\{z_{ik}\}) = Z^{-1} \exp\left\{ - \sum_{\alpha_{ij}=1} k_{ij}(1 - \delta_{ij}) \right\} = Z^{-1} \prod_{\alpha_{ij}=1} (1 - p_{ij}) + p_{ij}\delta_{ij},$$

where $Z = \sum_{\{z_{ki}\}} \exp\{-H(\{z_{ki}\})\}$ is the normalizing constant. Following Sokal's derivation (Sokal 1996), since the sum of any two real numbers x, y, can be written as $x + y = \sum_{b=0}^{1} x(1 - b) + yb$, it follows that $Z = \sum_{\{z_{ki}\}} \sum_{\{b_{ij}\}} \prod_{\alpha_{ij}=1} \{(1 - p_{ij})(1 - b_{ij}) + p_{ij} b_{ij} \delta_{ij}\}$, where the $\{b_{ij}\}$ are binary $0 - 1$ variables. They are said to be the *bonds* between the vertices of the graph. The joint density of labels and bonds is

$$p(\{z_{ki}\}, \{b_{ij}\}) = Z^{-1} \prod_{\alpha_{ij}=1} \{(1 - p_{ij})(1 - b_{ij}) + p_{ij} b_{ij} \delta_{ij}\}, \tag{3.1}$$

which is known as the Fortuin–Kasteleyn–Swendsen–Wang model (Sokal 1996, p. 46). The marginal density over the labels is exactly the Potts model. The marginal over the bonds is known as the *random-cluster* model. The interpretation of the bond variables in model (3.1) is the following. The bond b_{ij} is said to be *frozen* if $b_{ij} = 1$, and the points x_i and x_j are neighbors ($\alpha_{ij} = 1$) and have the same label ($\delta_{ij} = 1$). Otherwise, the bond is not frozen: $b_{ij} = 0$. The bond b_{ij} becomes frozen with probability $p_{ij} = 1 - \exp\{-k_{ij}\}$. A set for which any two points can be connected by a path of frozen bonds is said to be a connected set. Only subsets containing points with the same label can form a connected set. The Swendsen–Wang algorithm uses (3.1) to generate samples from the Potts model. This is a Gibbs sampler with two steps:

Step 1. Given the labels $\{z_{ki}\}$, each bond becomes frozen independently of the others with probability p_{ij} if $\alpha_{ij} = 1$ and $\delta_{ij} = 1$; otherwise, the bond is set to 0.

Step 2. Given the bonds $\{b_{ij}\}$, each connected subset is assigned the same label. The assignment is done independently and chosen uniformly at random from the set of labels.

The connected sets formed by frozen bonds allow for cluster splitting (Step 1). Merging is produced by the label assignment (Step 2). In the Swendsen–Wang algorithm both merging and splitting are done *in parallel*, since a multitude of sites in the graph are updated simultaneously in each iteration of the algorithm.

The Wolff algorithm (Wolff 1989) is a variant of the second step above. Instead of updating all connected sets, a point in the graph is chosen uniformly at random; the associated connected set is then updated as in the Swendsen–Wang algorithm. The advantage of this variant is that large clusters are updated often.

3.2 THE CONSENSUS CLUSTERS

Several (simulated) samples drawn from the Potts model are collected. The clustering is estimated by counting how many times any two given points are given the same label. The consensus clusters are based on MCMC estimates $\hat{Q}_{ij}$ of the probabilities (under the Potts model) $Q_{ij} = p(z_{ki} = z_{kj} \text{ for some } k \in \{1, \ldots, q\}) = p(\delta_{ij} = 1)$. If $\hat{Q}_{ij}$ is larger than a certain threshold (usually 0.5), then points x_i, x_j are assigned to the same cluster.

3.3 PENALIZED POTTS MODEL CLUSTERING

Penalized Potts model clustering aims at maximizing (see the right-hand side of (2.8))

$$\frac{1}{2}\sum_{j=1}^{n}\sum_{i=1}^{n}\alpha_{ij}k_{ij}\delta_{ij} - \frac{n^2}{2}\sum_{k=1}^{q}\gamma_k^2, \tag{3.2}$$

which is the same as maximizing $(1/2)\sum_{j=1}^{n}\sum_{i=1}^{n}\alpha_{ij}k_{ij}\delta_{ij} - (n^2/2)\sum_{k=1}^{q}(\gamma_k - (1/q))^2$. Hence, the penalty term tends to balance the cluster sizes. Noting that $\sum_{i=1}^{n} z_{ki} = n_k$, we obtain

$$n^2\sum_{k=1}^{q}\gamma_k^2 = \sum_{k=1}^{q}n_k^2 = \sum_{k=1}^{q}\sum_{j=1}^{n}\sum_{i=1}^{n}z_{ki}z_{kj} = \sum_{j=1}^{n}\sum_{i=1}^{n}\alpha_{ij}\,\delta_{ij} + \sum_{j=1}^{n}\sum_{i=1}^{n}(1-\alpha_{ij})\,\delta_{ij}. \tag{3.3}$$

Rewriting k_{ij} as $k_{ij} + 1$, and using (3.3), the penalized criterion (3.2) can be written as

$$\frac{1}{2}\sum_{j=1}^{n}\sum_{i=1}^{n}\alpha_{ij}k_{ij}\delta_{ij} - \frac{1}{2}\sum_{j=1}^{n}\sum_{i=1}^{n}(1-\alpha_{ij})\,\delta_{ij}. \tag{3.4}$$

Therefore, penalized Potts model clustering imposes a penalty whenever nonneighboring points are assigned the same label.

3.4 SIMULATING THE LABELING DISTRIBUTION: THE PENALIZED WOLFF ALGORITHM

We have developed a variant of the Wolff algorithm to estimate the cluster structure under the criterion (3.2). The last term in (3.4) can be seen as a penalty on the graph formed by connecting all nonneighboring points. Hence, in a similar fashion as the bonds were introduced into the original graph in the Swendsen–Wang algorithm, we augment the model by introducing bonds between nonneighboring points.

Let $\{d_{ij}\}$ be the set of nonneighbor bonds. d_{ij} is set to 1 (i.e., becomes frozen) with probability $q_{ij} = 1 - e^{-1}$ only if x_i and x_j are not neighbors, and $\delta_{ij} = 0$. We say that there is a connected path between points x_i and x_j if there exists a path of consecutive frozen-bond edges that starts at x_i and finishes at x_j. The connected path of point x_i is the set of points connected to x_i by a connected path.

The penalized Wolff's algorithm works as follows:

1. Given the labels $\{z_{ki}\}$ and the nonneighbor bonds $\{d_{ij}\}$, set each bond b_{ij} to 1 independently of other bonds with probability $p_{ij} = 1 - e^{-k_{ij}}$ if the following four conditions hold: (i) $\alpha_{ij} = 1$, (ii) $\delta_{ij} = 1$, (iii) there is no nonneighbor frozen-bond between the point x_j and another point in the connected path of x_i, and (iv) there is no nonneighbor frozen-bond between the point x_i and another point in the connected path of x_j. Otherwise, set the bond to 0.

2. Given the labels $\{z_{ki}\}$ and the bonds $\{b_{ij}\}$, set each nonneighbor bond d_{ij} to 1 independently of other nonneighbor bonds with probability $q_{ij} = 1-e^{-1}$ if the following three conditions hold: (i) $\alpha_{ij} = 0$, (ii) $\delta_{ij} = 0$, and (iii) there is no connected path between points x_i and x_j. Otherwise, set the nonneighbor bond to 0.

3. Given the bonds $\{b_{ij}\}$, $\{d_{ij}\}$, choose a point x uniformly at random. Find the associated connected subset $A = A(x)$ and the associated set $B(A)$ of nonneighbor points that have a nonneighbor frozen bond with at least one of the points in the connected subset A. Form the set $C(B)$ of all labels associated with points in $B(A)$ and its complement $\overline{C}(B)$. This latter set is the set of *admissible* labels. Choose a label uniformly at random from the set of admissible labels. Assign this label to all points in A.

The final cluster structure is estimated as explained in Section 3.2.

4. THE CONNECTION WITH DENSITY ESTIMATION

Equation (2.9) connects Potts model clustering with density estimation. The interaction term k_{ij} can be thought of as the contribution of the point x_j when evaluating the kernel density at the point x_i. This interpretation of Potts model clustering leads to some improvements of the model as shown later. By analogy with kernel density estimation, one could use an adaptive bandwidth in the interaction terms. Using the quick estimate of the density at x_i, $\hat{p}_{knn}(x_i)$, obtained at the time the K-nearest-neighbor graph of the data was constructed (see the beginning of Section 2.3), we derive a localized bandwidth (Breiman, Meisel, and Purcell 1977; Abramson 1982; Silverman 1986, Sect. 5.3) $\lambda_{knn}(x_i) \propto \exp\{-0.5\,(\log[\hat{p}_{knn}(x_i)] - (1/n)\ \sum_{j=1}^{n} \log[\hat{p}_{knn}(x_j)])\}$. Since the Potts model uses a symmetric kernel, we symmetrize the adaptive bandwidth kernel by replacing k_{ij} with $k_{ij}^{(s)} = 0.5\,\{k(\lambda_{knn}^{-1}(x_i)(x_i - x_j)) + k(\lambda_{knn}^{-1}(x_j)(x_j - x_i))\}$. In our experiments, this choice of bandwidth often improved the estimation of the clustering structure in the data. We refer to the algorithm run with these bandwidths as the *adaptive* Potts model clustering algorithm.

4.1 BANDWIDTH ESTIMATION

The adaptive bandwidth given above can be used as a starting value to simultaneously estimate the local bandwidths and the clustering. The Swendsen–Wang and Wolf penalized algorithms are extended by a Metropolis–Hastings step (Metropolis 1953; Hastings 1970):

1. For given bandwidths $\{\lambda_i\}$, update the labels as in the Swendsen–Wang, Wolff, or penalized-Wolff algorithm.

2. For given labels, update the bandwidths independently of each other through a Metropolis–Hastings procedure.

In what follows we describe a Metropolis–Hastings sampler for the bandwidth with an inverse chi-squared prior. Recall that for a given target density $\pi(\lambda)$ and proposal density $q(\lambda^*|\lambda)$, the Metropolis–Hastings algorithm proceeds as follows: given the current state λ, an update to state λ^* is proposed with density $q(\lambda^*|\lambda)$; the update is accepted with probability

$$A(\lambda_i^*, \lambda_i) = \min\{1, [q(\lambda_i|\lambda_i^*)\pi(\lambda_i^*)]/[q(\lambda_i^*|\lambda_i)\pi(\lambda_i)]\}.$$

In our particular case the joint density of labels and bandwidths is proportional to

$$\pi(\lambda_i) = \prod_i \lambda_i^{-(\nu+2)/2} \exp\left\{-\frac{1}{2}\frac{\nu s_i^2}{\lambda_i}\right\} \times \exp\left\{-\frac{1}{2}\sum_{\alpha_{ij}=1} k_{ij}^{(s)}(\lambda_i, \lambda_j)(1-\delta_{ij})\right\},$$

where s_i^2 are the prior scales, and ν is the prior degrees of freedom. We have used the notation $k_{ij}^{(s)}(\lambda_i, \lambda_j)$ to make explicit the dependency of the symmetrized kernel on both bandwidths λ_i and λ_j. At each location x_i, consider an inverse chi-squared proposal density $q(\lambda_i^*|\lambda_i)$ with scale $s_i^2 + (1/\nu)\sum_{j,\,\alpha_{ij}=1} \lambda_i k_{ij}^{(s)}(\lambda_i, \lambda_j)(1-\delta_{ij})$, and ν degrees of freedom. Then the acceptance ratio for the proposal is $R(\lambda_i^*, \lambda_i) = \exp\{-(1/2)\sum_{j,\,\alpha_{ij}=1}(1 - \delta_{ij})[k_{ij}^{(s)}(\lambda_i^*, \lambda_j) - k_{ij}^{(s)}(\lambda_i, \lambda_j)]\}$. The update λ_i^* is accepted with probability $A(\lambda_i^*, \lambda_i) = \min\{1,\ R(\lambda_i^*, \lambda_i)\}$.

4.2 SMOOTHING PRIORS

If bandwidths associated with nearby points are expected to be similar, then a penalty prior on the smoothness of the bandwidths can be used. We experimented with two priors on the log-bandwidths: a Gaussian and a Laplace-type prior. As expected, the Gaussian prior yields smoother bandwidths, whereas the Laplace prior yields piecewise-constant looking bandwidths (see Section 6.3).

Let $\tau_i = \log \lambda_i, i = 1, \ldots, n$. The Gaussian prior for the bandwidths has the form

$$p_n(\{\tau_i\}|\{z_{ki}\}) \propto \exp\left\{\frac{-1}{2\sigma^2}\sum_{i=1}^{n}\sum_{\alpha_{ij}=1}(\tau_i-\tau_j)^2\delta_{ij}\right\} \times \exp\left\{\frac{-1}{2\,\sigma_0^2}\sum_{i=1}^{n}(\tau_i - \log\lambda_{knn}(x_i))^2\right\}.$$

Hence, only bandwidths of neighboring points with the same label are expected to be similar. The variance σ^2 acts as a penalty cost. As before, bandwidth updates are generated using a Metropolis–Hastings sampler. Our proposal τ_i^* is generated from the Gaussian density with mean μ_i and variance σ_i^2 given by $\mu_i = \sigma_i^2\ (2\sigma^{-2}\sum_{\alpha_{ij}=1}\tau_j\delta_{ij} + \sigma_0^{-2}\log\lambda_{knn}(x_i))$, $\sigma_i^2 = (2m_i\sigma^{-2}+\sigma_0^{-2})^{-1}$, where m_i is the number of neighbors of x_i with the same label as x_i. The acceptance ratio $R_2(\tau_i^*, \tau_i)$ is given by $R_2(\tau_i^*, \tau_i) = \exp\{-(1/2)\sum_{j,\,\alpha_{ij}=1}[k_{ij}^{(s)}(\lambda_i^*,\lambda_j) - k_{ij}^{(s)}(\lambda_i,\lambda_j)](1-\delta_{ij})\}$.

Similarly, our Laplace-type prior has the form

$$p_\ell(\{\tau_i\}|\{z_{ki}\}) \propto \exp\left\{\frac{-\Delta}{2}\sum_{i=1}^{n}\sum_{\alpha_{ij}=1}|\tau_i-\tau_j|\delta_{ij}\right\},$$

where Δ is the penalty cost parameter. In this case, our proposal τ_i^* is generated from the Laplace density with location equal to the median of the bandwidths τ_i and τ_j's associated with neighboring points with the same label as x_i. Let $\tilde{\mu}_i$ denote this median, and $\tilde{\mu}_i^*$ denote the median of the bandwidths τ_i^* and τ_j's associated with neighboring points with the same label as x_i. The acceptance ratio $R_1(\tau_i^*, \tau_i)$ is given by

$$\begin{aligned} R_1(\tau_i^*, \tau_i) &= \exp\{-\Delta[|\tau_i - \tilde{\mu}_i^*| - |\tau_i^* - \tilde{\mu}_i|]\} \\ &\quad\times \exp\left\{-\Delta[\sum_{j,\,\alpha_{ij}=1}(|\tau_i^*-\tau_j| - |\tau_i-\tau_j|)\delta_{ij}\right\} \\ &\quad\times \exp\left\{-\frac{1}{2}\sum_{j,\,\alpha_{ij}=1}[k_{ij}^{(s)}(\lambda_i^*,\lambda_j) - k_{ij}^{(s)}(\lambda_i,\lambda_j)](1-\delta_{ij})\right\}. \end{aligned}$$

4.3 DENSITY ESTIMATION

Let λ_i be the median bandwidth at x_i estimated from one of the procedures outlined above. Let $p(x|k) = n_k^{-1}\sum_{i=1}^{n} z_{ki}\ k([x-x_i]/\lambda_i)$. The density estimator at x is $\hat{f}(x) = \sum_{k=1}^{q}\gamma_k\, p(x|k) = \frac{1}{n}\sum_{i=1}^{n}k([x-x_i]/\lambda_i)$. Section 6.3 gives an idea of how this estimate works.

5. TEMPERATURE SELECTION

An important parameter in the Potts model is the *temperature, T*. It modifies the weights ($w(i, j, \{z_{ki}\}) = 1/T$ (see (1.2)), and the Hamiltonian ($H(\{z_{ki}\})/T$). At any given temperature, the clustering is estimated by counting how many times any two given points are assigned the same label. As seen in Section 3.2, the label assignments are based on the probabilities $\{Q_{ij} = Q_{ij}(T)\}$. It turns out that these probabilities are directly related to the number of times any two given points occur in the same connected subset, and consequently, to probabilities under the *random clusters model* resulting from integrating out the labels in the joint density of labels and bonds (3.1) (Edwards and Sokal 1988). The log-density of the random clusters model for given T is proportional to $\sum_{\alpha_{ij}=1,b_{ij}=1} \log(1 - e^{-k_{ij}/T}) - \sum_{\alpha_{ij}=1,b_{ij}=0} k_{ij}/T + C(\{b_{ij}\}) \times \log(q)$, where $C(\{b_{ij}\})$ denotes the number of connected components given the current values of the bonds. This function favors more clusters when T is large, and fewer clusters when T is small. Hence, T acts as a *clustering smoothing parameter.* By varying the temperature from low to high values, Potts model clustering can be seen as a hierarchical splitting procedure. Thus, the key problem is to find the "right" temperature associated with the "true" clustering. This is a hard problem and more research is needed to solve it. The current strategy is to try several values of T while monitoring some T-dependent statistics of the "goodness-of-clustering." Based on the physical model underlying the Potts model, Blatt et al. (1996a,b) suggested monitoring the variance of the *magnetization* of the system. In statistical terms, this corresponds to the variance of the size of the largest cluster. Our experiments suggest that this is not always a good measure of splitting, since smaller clusters might split before the larger ones. Hence, an extension of the variance of the magnetization is to monitor the variance of the size of (possible many of) the largest clusters. The number of clusters to monitor depends on the number of clusters one expect to observe. Peaks on these variances indicate important splits. It is assumed that the true clustering of the data corresponds to a temperature nearby one of these peaks. Hence, locating the peaks is crucial. In order to get rough estimates of their location, one could quickly travel over a vast range of temperatures. Once the peaks are located, longer simulations could be run around them to study the cluster structures that they yield.

5.1 CHOOSING A GOOD TEMPERATURE

Monitoring cluster splitting is a way of measuring variation in clustering or partitions of the data. Hence, we have conceived a rule based on two approaches to measure this variation: (a) the distribution of the size of the largest clusters (an extension of the variance of the magnetization measure); and (b) the adjusted Rand index (Rand 1971; Milligan and Cooper 1986; Hubert and Arabie 1985).

5.1.1 (a) The Distribution of the Size of the Largest Clusters

For any given T, let $S_\ell(T)$ be the cluster size associated to the ℓth largest cluster found in a partition drawn from the Potts model, $\ell = 1, \ldots, G$, $S_1(T) \geq S_2(T) \geq \cdots \geq S_G(T)$.

G is a parameter of the procedure that depends on our prior belief on the true number of clusters. In our experiments we set $G = 6$. We monitor the variance curves as a function of temperature: $\mathrm{Mvar}(L, G, T) = \sum_{\ell=L}^{G} \mathrm{var}(S_\ell(T))/(G - L + 1)$, $L = 1, \ldots, G - 1$, and choose a clustering associated with a temperature immediately following one of the local maxima of these curves. Usually all the curves peak around the same temperatures, so the choice of L is not very relevant. However, we do recommend using $L > 1$, since the largest cluster is usually very large in comparison with the others, and its variance masks the changes in the other clusters.

5.1.2 The Adjusted Rand Index

There are many known measures for comparison of partitions, such as the Folkes-Wallace and the adjusted Rand indexes (Milligan and Cooper 1986; Hubert and Arabie 1985), that are popular in the clustering literature (Yeung et al. 2001; Tantrum et al. 2004). The Rand index (Rand 1971) is the fraction of pairs of points that are either in the same clusters in both partitions or in different clusters in both partitions. The adjusted Rand index (ARI) adjusts the Rand index so that its expected value is zero when the partitions are random. The larger the ARI, the more similar the two partitions are. Suppose that at each temperature T, M partitions $\{\mathcal{P}_m\}_{m=1}^{M}$ are generated from the Potts model. A representative partition is given by the consensus clustering $\mathcal{P}(T)$ among the M partitions. Let r_m be the ARI between $\mathcal{P}_m$ and $\mathcal{P}(T)$. The average of these indexes, $\bar{r}(T)$, measures the similarity among the M partitions, whereas $\mathrm{var}(r(T)) = \sum_{m=1}^{M}(r_m - \bar{r}(T))^2/(M-1)$ measures their instability. In principle, a good temperature T_0 is a temperature for which $\bar{r}(T_0)$ is high and $\mathrm{var}(r(T_0))$ is low. We also expect that the consensus partitions associated with temperatures in a neighborhood of T_0 be similar to the consensus partition found at T_0. In other words, the system induced by the Potts model should be more or less stable at temperatures close to T_0. An important implication is that the choice of T_0 should not be too critical as long as it is chosen in the neighborhood of the optimal temperature. The similarity between two consensus partitions $\mathcal{P}(T - \Delta T)$ and $\mathcal{P}(T)$ generated at consecutive temperatures $T - \Delta T$, and T is measured by their ARI, $R(T)$.

5.1.3 The Rule to Select T_0

We select the first candidate temperature T^* located at the end of the last significant maximum of $\mathrm{Mvar}(L, G, T)$ that precedes a sharp drop in the cluster-size variation. Then we check for temperatures near T^* that have high $\bar{r}(T)$, low $\mathrm{var}(r(T))$, and are found in a more or less stable region of $R(T)$ (i.e., a plateau of $R(T)$). The final choice T_0 is a compromise between all these conditions. Figure 1 illustrates the procedure. Our experiments in the next section show that this procedure performs well.

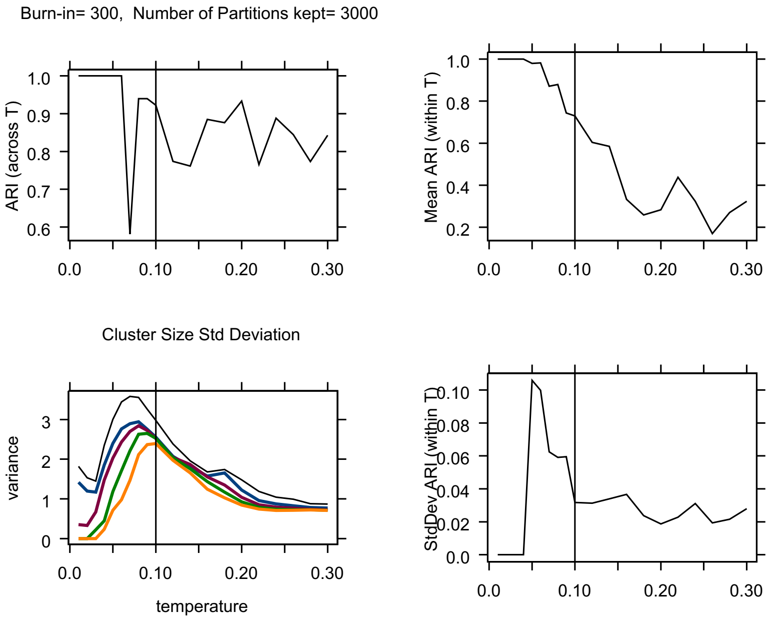

Figure 1. Plots of the adjusted Rand index (ARI) across temperatures, $R(T)$ (upper left corner), mean ARI within temperatures, $\bar{r}(T)$ (upper right corner), standard deviation of the ARI within temperatures, $\sqrt{\text{var (r(T))}}$ (lower right corner), and mean standard deviations, $\sqrt{\text{Mvar (L, G = 6, T)}}$, $L = 2, \ldots, 6$ (lower left corner). The vertical line corresponds to the chosen temperature $T_0 = 0.10$. The statistics were computed for the artificially generated dataset $\mathcal{D}_2$ based on 3,000 partitions kept after a burn-in period of 300.

6. EXPERIMENTAL RESULTS

6.1 Performance On Simulated Data

In this section we report the results of a simulation carried out to study the performance of Potts model clustering on three different artificially generated datasets and for different values of the Swendsen–Wang simulation parameters. The datasets were: $\mathcal{D}_1$ consisting of 200 points in two clusters, $\mathcal{D}_2$ consisting of 200 points in four clusters, and $\mathcal{D}_3$ consisting of 340 points in eight clusters. The data are plotted in Figure 2. The clusters are either Gaussian clumps or uniformly scattered around arcs.

6.1.1 The Adjusted Rand Index

The goal here is to find the right combination of the burn-in and partitions kept (after the burn-in period) parameters of the Swendsen–Wang algorithm within Potts model clustering. The burn-in parameter corresponds to the number of initial partitions generated by the algorithm that are discarded from further analysis. The partitions generated after the burn-in are the partitions kept to compute the consensus clustering and all relevant statistics. In our simulation, the burn-in and partitions-kept parameters were set to values in $\{100, 300, 600, 1000, 3000\}$. We ran Potts model clustering on each dataset $\mathcal{D}_i$ $(i = 1, 2, 3)$ with all 25 combinations of the two parameters. For each combination we

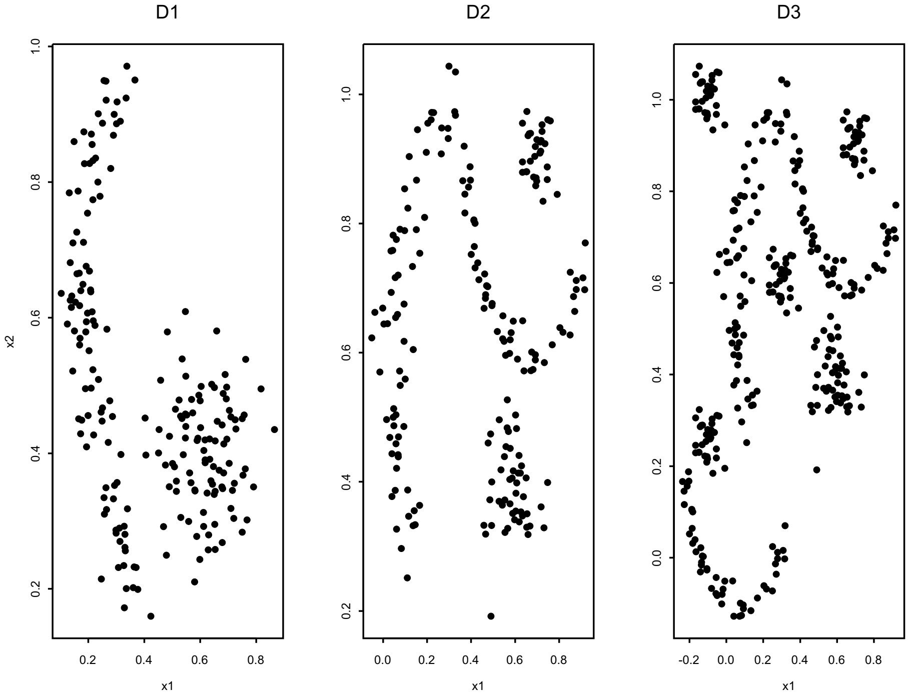

Figure 2. The three artificially generated datasets: (left) $\mathcal{D}_1$ consisting of one clump and one arc; (middle) $\mathcal{D}_2$ consisting of two clumps and two arcs; and (right) $\mathcal{D}_3$ consisting of five clumps and three arcs.

measured the number of clusters associated with the partition chosen according to the procedure explained in Section 5.1, and the ARI between the true partition and the chosen partition.

We always run Potts model clustering starting at a very cold temperature T_1 (for which no splitting is observed). After a fixed number of iterations, say M, the temperature is switched to a warmer one $T_2 > T_1$. At T_2 the initial partition is the last partition drawn at T_1. Then again after M iterations the temperature is increased to T_3 with initial partition given by the last partition drawn at T_2. This procedure is repeated until reaching the last hottest temperature T_n. Hence, for an intermediate temperature T_k, $(k-1)M$ draws already have been generated before the first partition at T_k is drawn. Since the grid of temperatures at which the algorithm is run is not too coarse, the initial partition at each temperature T_k is very likely to be a good starting partition for the Swendsen–Wang algorithm. This observation may explain the results of the simulation summarized in Figure 3. It appears that the performance of Potts model clustering is not sensitive to the burn-in parameter. The performance with only 100 partitions-kept is very different and much poorer than the performance with at least 300 partitions-kept. The analysis of variance on the resulting ARIs confirmed these observations. An HSD (honestly significant difference) Tukey method for all pairwise comparisons (Tukey 1949) at a significance $\alpha = 0.01$ revealed no significant difference between the performances with 300, 600, 1000 or 3000 partitions-kept. How-

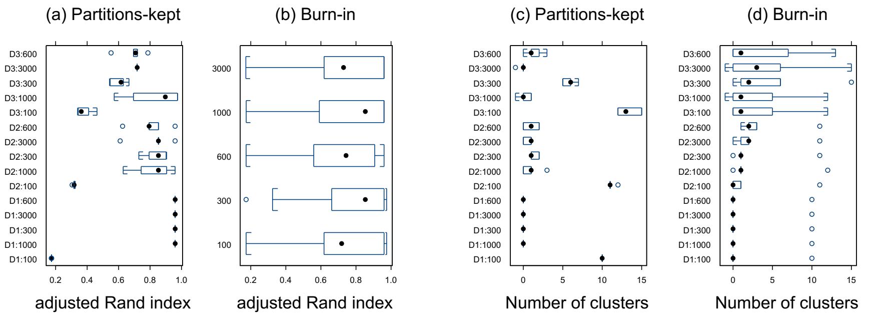

Figure 3. The performance of Potts model clustering for the three artificial datasets $\{\mathcal{D}_i\}_{i=1}^{3}$. The figure shows boxplots of (a) the adjusted Rand index (ARI) associated with each combination of dataset and partitions-kept parameter; (b) the ARI associated with each value of the burn-in parameter; (c) the difference between the estimated and the true number of clusters associated with each combination of dataset and partitions-kept parameter; (d) the difference between the estimated and the true number of clusters associated with each combination of data set and burn-in parameter.

ever the analysis of variance on the resulting difference between the square-roots of the estimated number of clusters and the true number of clusters showed a significant interaction between the data sets and the number of partitions kept. As observed in Figure 3, the performance with at most 300 partitions-kept is worse than the performance with at least 600 partitions-kept for $\mathcal{D}_3$, but not for the other datasets. This indicates that the appropriate number of partitions-kept is dependent on and increasing with the complexity of the data.

6.1.2 The Number Of Clusters

Potts model clustering yielded the correct number of clusters 55% of the time. Eighty-eight percent of the time the estimated number of clusters was off by at most one cluster; 92% of the time it was within three clusters. Increasing the number of partitions kept increased the proportion of correct estimates of the number of clusters. The proportion of times the estimates were correct when 300 partitions were kept was 0.33; this proportion increased to 0.53, 0.60, and 0.73 when the number of partition kept was increased to 600, 1000, and 3000, respectively.

In conclusion, a small burn-in and a moderate data-dependent number of partitions kept (e.g., about 1000) were enough to obtain a good performance. Although there was no evidence that keeping several thousands partitions would improve the ARI in a significant way, keeping a large number of partitions would probably improve the estimation of the number of clusters.

6.2 Applications to Gene expression Data

We applied Potts model clustering to two different gene expression data sets: the subtypes of acute lymphoblastic leukemia data (Yeoh et al. 2002), and the yeast cell cycle data (Cho et al. 1998). The points were normalized to norm one. We constructed a 10-nearest-neighbor graph for each dataset and used a Gaussian kernel, $k(x_i, x_j) \propto$

Table 1. Partition matrix for the subtypes of acute lymphoblastic ALL data.

True	Estimated clusters					
Clusters	1	2	3	4	5	6
BCR-ABL	0	0	0	9	0	0
E2A-PBX1	18	0	0	0	0	0
Hyperploid> 50	0	14	0	28	0	0
MLL rearrangement	3	0	9	2	0	0
OTHERS	5	0	2	39	0	6
T-ALL	0	0	1	0	27	0
TEL-AML1	0	0	0	0	0	52

$\exp\{-||x_i - x_j||^2/(2\sigma^2)\}$ whose bandwidth σ was estimated adaptively as explained in Section 4. The final clusters were forced to have at least five points (smaller clusters were merged with their closest clusters). For comparison purposes we ran both the nonadaptive (fixed bandwidth) and penalized versions of Potts model clustering, and the (nonparametric) dendrogram sharpening method (McKinney 1995; Stanberry, Nandy, and Cordes 2003) on both datasets. We also applied model-based clustering (Banfield and Raftery 1993) to the yeast cycle data. The same values of the burn-in and partitions-kept parameters were used for all three versions of Potts model clustering. As Potts model clustering, the dendrogram sharpening and model-based Gaussian clustering methods do not require prior assumption about the number or location of clusters. The reader is referred to the articles cited above for detailed descriptions of these methods.

The ALL data consist of oligonucleotide microarray gene expression levels of 12,558 genes for each of 360 ALL patients. Yeoh et al. (2002) divided the patients into seven diagnostic groups corresponding to six known leukemia subtypes (T-ALL, E2A-PBX1, BCR-ABL, TEL-AML1, MLL rearrangement, and Hyperploid$>$ 50 chromosomes), and one unknown type, labeled OTHER. The data were taken from the Kent Ridge Bio-Medical Data Set Repository, where they have been split into training and test sets. For our experiments we selected the training set comprising 215 patients.

In view of the simulation results described in Section 6.1, 1000 partitions were kept after a burn-in period of 300. Figure 4 shows the results of the adaptive Potts model clustering algorithm. The vertical line corresponds to the temperature $T_0 = 0.67$ chosen according to the procedure described in Section 5.1. A comparison of the corresponding clustering structure with that obtained in (Yeoh et al. 2002) is shown in Table 1. The rows of this matrix represent the seven subtypes of ALL leukemia assigned by Yeoh et al. (2002). The columns represent the six estimated clusters from the adaptive-bandwidth Potts model clustering algorithm. Each cell (i, j) counts the number of points shared in the corresponding ith true and jth estimated clusters. The associated ARI between these two partitions is 0.56. The adaptive Potts model clustering produced clusters very similar to the seven subtypes of ALL leukemia reported by Yeoh et al. (2002); except that Hyperploid$>$50, OTHERS, and BCR-ABL appear to be difficult to separate. The partition clearly separates the E2A-PBX1

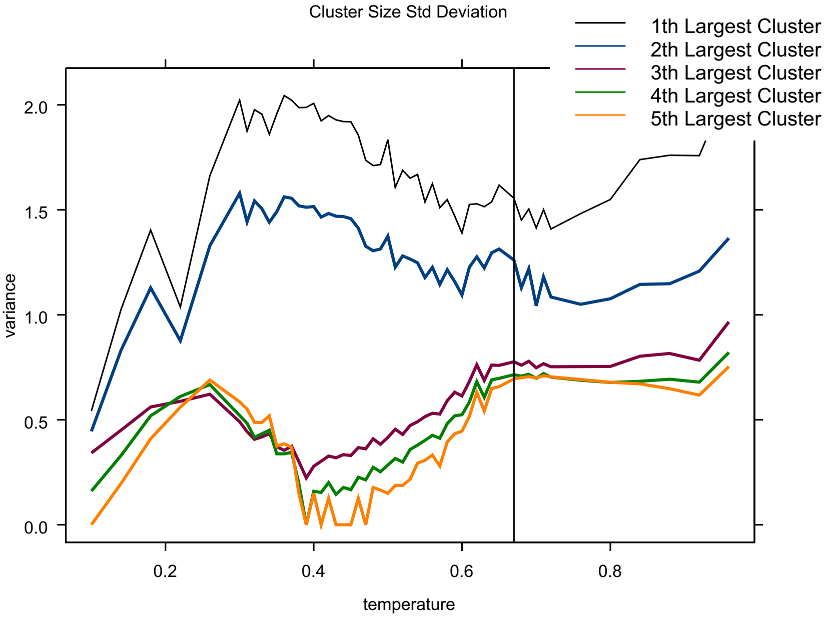

Figure 4. The mean standard deviations $\sqrt{\mathrm{Mvar}\,(\mathrm{L}, \mathrm{G} = 6, \mathrm{T})}$, $L = 2, \ldots, 6$, as a function of temperature, resulting from the application of the adaptive-bandwidth Potts model clustering on the ALL data. The vertical line corresponds to $T_0 = 0.67$.

and MLL rearrangement subtypes from the others, which is important, since these types of leukemia do not respond well to conventional treatments.

We applied both the nonadaptive and penalized Potts model clustering to these data with constant bandwidth equal to the mean of the distances. Both methods yielded similar ARI of 0.43 and 0.44 with seven and six clusters, respectively. The adaptive penalized Potts model clustering yielded six clusters with an associated ARI of 0.53. The main difference from the partition produced by the adaptive nonpenalized version is that OTHERS was split among five clusters. The dendrogram sharpening method yielded a three-cluster partition with an ARI of 0.23.

The yeast cell cycle data record the fluctuations of the expression levels of about 6,000 genes over two cell cycles comprising 17 time points. We use the five-phase subset of the data (Cho et al. 1998). It consists of 420 genes of which 386 have been assigned to one of five phases of the cells cycle. The clustering results should reveal five groups of genes associated with the five phases. We ran the adaptive Potts model clustering algorithm with 500 iterations of the Swendsen–Wang algorithm: 250 partitions were kept after a burn-in period of 250. Despite the small number of partitions kept in the analysis, Potts model clustering was able to find a good nine-cluster partition of the data. Table 2 shows the corresponding partition matrix. The associated ARI was slightly over 0.46. A run of the algorithm at the same temperature, forcing the cluster size to be least 20, yielded a six-cluster partition with an associated ARI of 0.45. Hence, it is important to have a good prior estimate of the size of the smallest cluster. The nonadaptive Potts model clustering algorithm yielded 16 clus-

Table 2. Partition matrix for the yeast cell cycle data.

True	Estimated clusters								
Clusters	1	2	3	4	5	6	7	8	9
1	51	5	6	0	0	2	0	0	3
2	16	0	117	1	0	1	0	0	0
3	5	0	34	13	10	3	5	4	2
4	2	2	0	3	5	2	17	10	12
5	4	5	0	0	0	1	1	0	44

ters with an associated ARI of 0.43. Penalized Potts model clustering yielded nine clusters with an associated ARI of 0.45. Both the adaptive and penalized algorithms yielded similar clustering structures for the data. Yeung et al. (2001) analyzed the labeled subset of these data using model-based clustering based on Gaussian mixtures (Banfield and Raftery 1993). They reported four clusters with an ARI of about 0.43. The dendrogram sharpening method yielded a four-cluster partition with an ARI of 0.45.

6.3 EXPLORING KERNEL DENSITY ESTIMATION

In this section we explore the connection between Potts model clustering and kernel density estimation as outlined in Section 4. We compare the three different strategies of bandwidth estimation described in Sections 4.1 and 4.2 on two one-dimensional and one two-dimensional datasets: an artificial dataset, the galaxy data, and the Old Faithful geyser data.

We ran Potts model clustering with bandwidth estimation using the Euclidean distance as a measure of similarity between the points, and the Gaussian kernel. We ran the modified Swendsen–Wang Metropolis–Hastings algorithms (see Sections 4.1 and 4.2) with 600 complete iterations. The first 300 iterations were discarded from the analysis. The remaining 300 iterations were used to cluster the data and to compute the cluster size variances in the largest six clusters. The final clusters were forced to have at least five points. Unless otherwise noted, all runs were initialized with constant bandwidth equal to the mean distance between any two points. We observed that all three bandwidth estimation algorithms (Gamma bandwidth selection, and Gaussian and Laplace smoothing priors) selected clusterings at similar temperatures for these datasets. Bandwidth smoothing did not appear to influence the clustering. The main difference was in the bandwidth estimates.

The artificial dataset was created with the purpose of comparing the kernel density estimates with the true density of the data. The data were generated from a Gaussian mixture with five components centered at −0.3, 0.0, 0.2, 0.8, and 1.0; and with variances equal to 1.0, 2.0, 0.7, 0.4, 0.2, respectively. The component proportions were proportional to 0.2, 0.1, 0.1, 0.2, 0.3. One-hundred fifty points were drawn from this distribution. Figure 5 shows the bandwidths means for the Gamma updates, the Gaussian updates with $\sigma_0^2 = 0.1$, $\sigma^2 = 1.0$, and the Laplace updates with prior parameter $\Delta = 100$ and

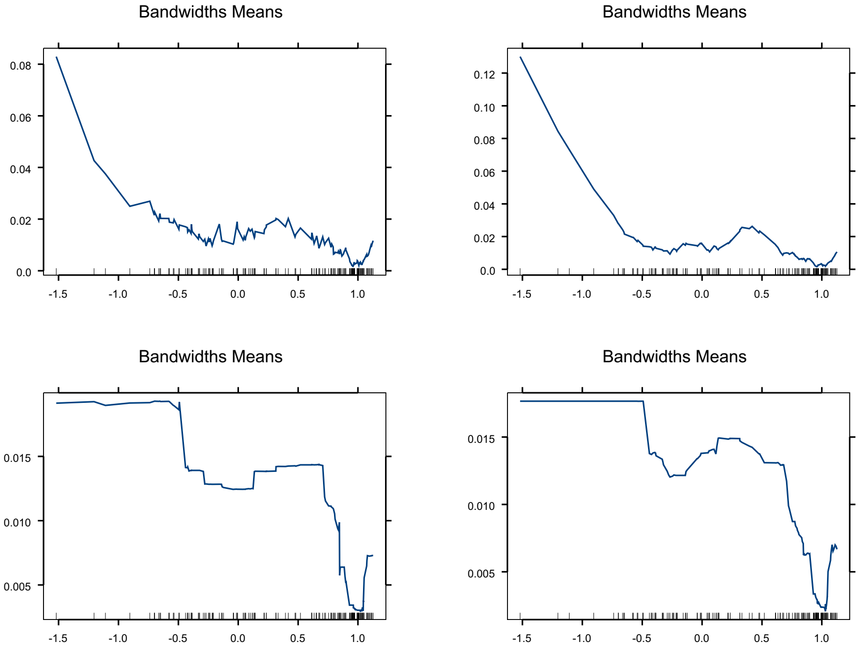

Figure 5. Artificial dataset: Bandwidths means for different Metropolis-Hastings bandwidth updating strategies: Gamma (top left), Gaussian with $\sigma_0^2 = 0.1$, $\sigma^2 = 1.0$ (top right), and Laplace with $\Delta = 100$ (bottom left) and $\Delta = 10,000$ (bottom right).

$\Delta = 10000$. The associated kernel density estimators as well as the true density and the adaptive bandwidth kernel density estimator are depicted in Figure 6. One can clearly appreciate the smoothness in the bandwidths introduced by the Gaussian smoothing prior, and the nearly piece-wise constant shape yielded by the Laplace smoothing prior. All associated kernel density estimators look very similar to each other. They all introduce smoothness to the estimator as compared with the one based only on the adaptive bandwidth.

The galaxy dataset provided with S-Plus version 6.2 consists of 323 measurements of the radial velocity (in km/second) of a spiral galaxy (NGC7531) measured at points in the area of the sky covered by it (Buta 1987; Chambers and Hastie 1992). Figure 7 shows the bandwidth medians and the associated kernel density estimators yielded by Potts model clustering with Gamma and Gaussian penalty ($\sigma^2 = 1.0$) updates. The bandwidth smoothness introduced by the Gaussian smoothing prior is obvious. The figures clearly show eight to nine modes in the density estimates which correspond to the clusters found by the Potts model algorithms.

The Old Faithful dataset provided with the S-Plus version 6.2 consists of 299 measurements of the waiting time between eruptions and the duration of the eruption for the Old Faithful geyser in Yellowstone National Park (Azzalini and Bowman 1990). The bandwidth medians yielded by the different bandwidth update algorithms are shown in Figure 8. The corresponding two-dimensional contours and kernel density estimates are shown in Figure 9. Note the spikes in the bandwidths near previous durations 2.0 and 4.0. These mark

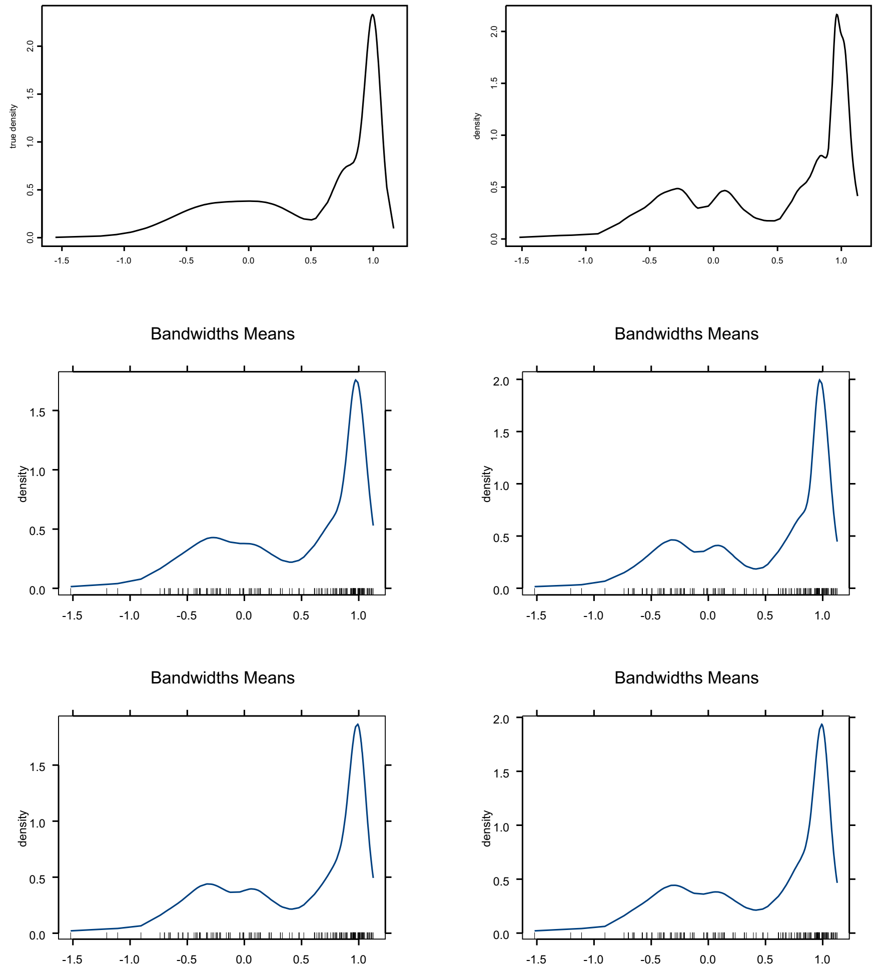

Figure 6. Artificial dataset: Kernel density estimator means for different Metropolis–Hastings bandwidth update strategies: True density (top left), kernel density with adaptive bandwidth (top right), Gamma (center left), Gaussian with $\sigma_0^2 = 0.1,\ \sigma^2 = 1.0$ (center right), and Laplace with $\Delta = 100$ (bottom left) and $\Delta = 10{,}000$ (bottom right).

the boundaries between the two main clusters of points. Also note that the bandwidths tend to increase in the “empty” region. The spikes are probably due to the sharp increase in the density at the clusters. The Laplace smoothing prior updates yield almost piece-wise constant bandwidths within the two main clusters and the empty region. The updates are highly variable in the boundary regions. The contours in Figure 9 show the smoothness introduced by the Gaussian and Laplace smoothing priors. Overall, the Laplace updates appeared to have produced the smoothest looking kernel density estimator, although the Gaussian updates seemed to have yielded the smoothest bandwidths.

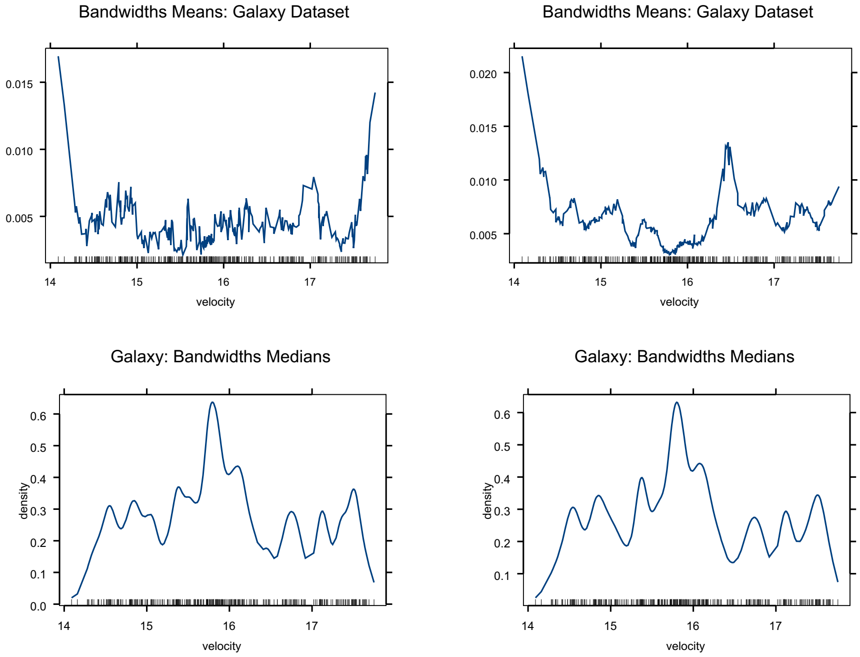

Figure 7. Galaxy dataset: Bandwidths medians and associated kernel density estimators for different Metropolis-Hastings bandwidth update strategies: Gamma (left column), and Gaussian with $\sigma_0^2 = \sigma^2 = 1.0$ (right column).

7. DISCUSSION

The main contribution of this article is to uncover and exploit the close connection between Potts model clustering and kernel K-means and kernel density estimation. Interpreting the Hamiltonian of the Potts model in terms of the conditional densities given the cluster labels motivates a variant of Potts model clustering that incorporates a penalty for unequal cluster sizes. A modification of the Wolff algorithm allows us to simulate configurations from the distribution defined by this penalized Hamiltonian, leading to penalized Potts model clustering. The link to kernel density estimation suggests replacing constant bandwidth with adaptive bandwidth kernels, a generalization long recognized as advantageous in the context of density estimation that also turns out to be beneficial for clustering.

There are several directions for future work: (i) To use more general penalty terms for penalized Potts model clustering. The algorithm outlined in this article uses a constant penalty ($k_p(x_i, x_j) = 1$) for nonneighbor points with the same label assignment. But one could use a different kernel $k_p(x_i, x_j)$ for interactions between nonneighbor points. For example, we could make this penalty an increasing function of the distance. (ii) To develop more computationally efficient ways of choosing the temperature. Our current method requires simulating configurations at several different temperatures. It would be more efficient if a good temperature could be discovered in a single run of the algorithm. We think a strategy close in spirit to simulated tempering (Marinari and Parisi 1992; Geyer and Thompson 1995) and parallel tempering (Geyer 1991) may be worth investigating.

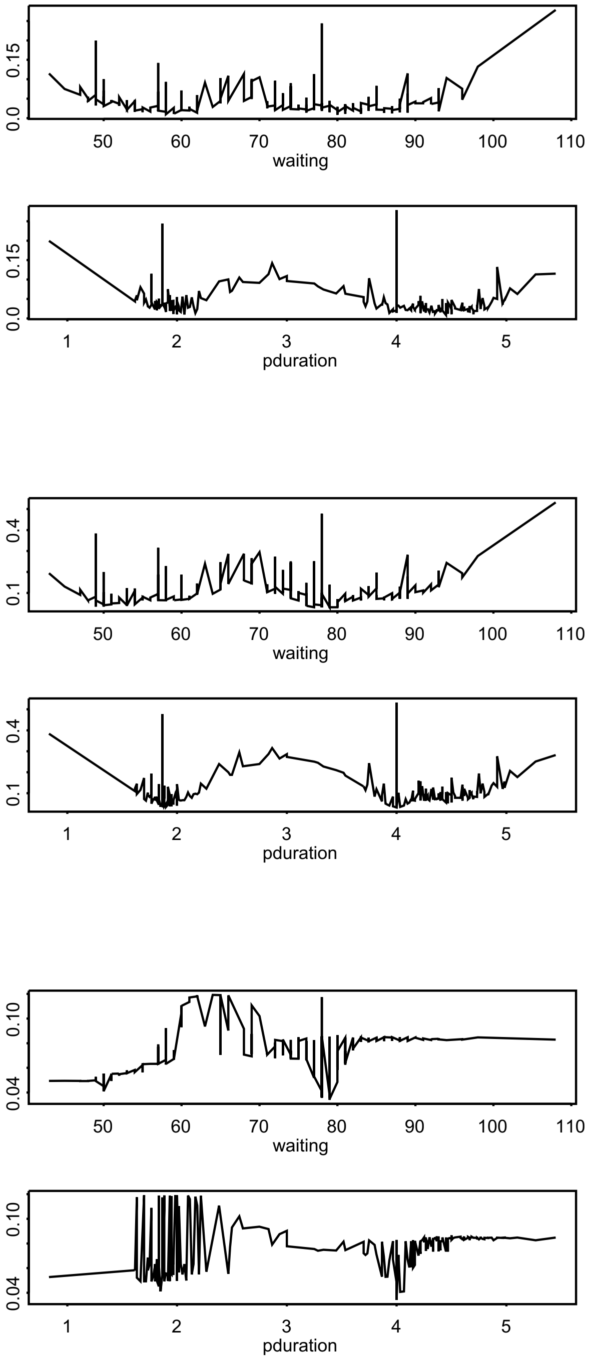

Figure 8. Old Faithful: Bandwidth medians yielded by the Gamma update (top), Gaussian with $\sigma_0^2 = 0.01$, $\sigma^2 = 1.0$ (center), and Laplace with $\Delta = 100$ (bottom). The horizontal axes correspond to the waiting time to eruption (upper plots) and to the previous eruption duration (lower plots).

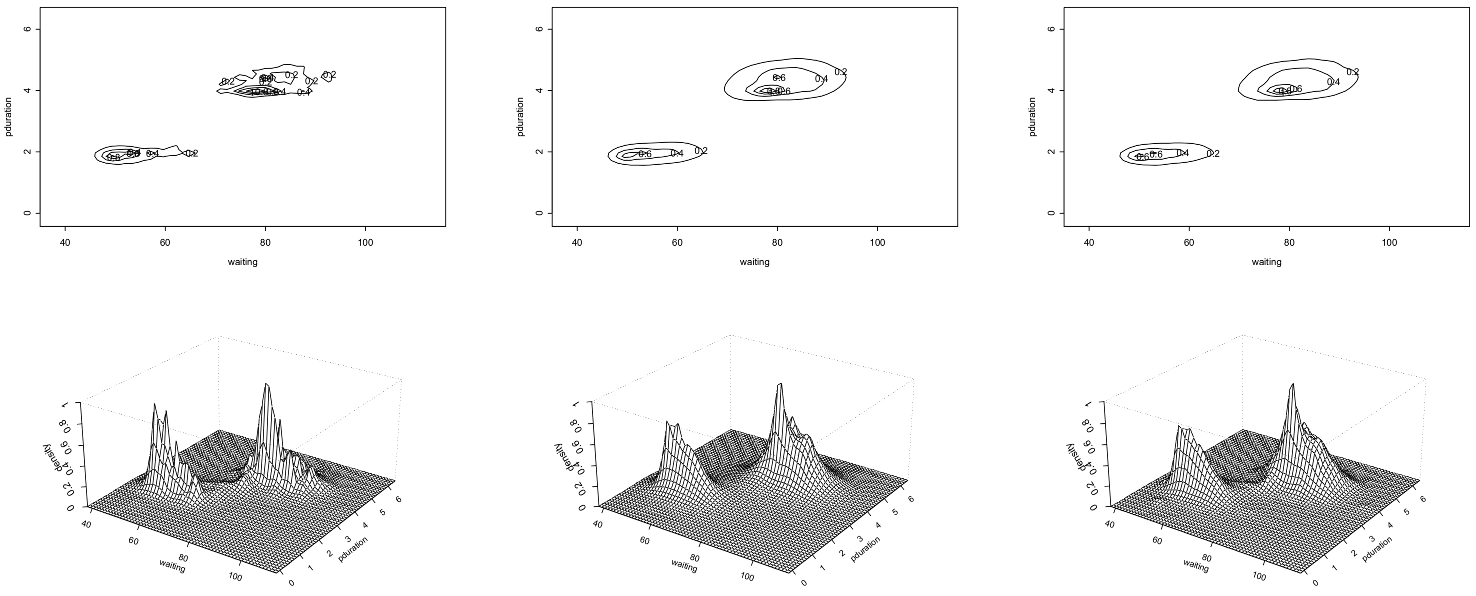

Figure 9. Old Faithful: Kernel density estimators associated with the bandwidth medians yielded by the Gamma update (left), Gaussian with $\sigma_0^2 = 0.01, \ \sigma^2 = 1.0$ (center), and Laplace with $\Delta = 100$ (right).

And (iii) to consider an extension to semisupervised learning. In semi-supervised learning one is given the true labels for a (typically small) subset of the observations. This information could be incorporated by assigning a large similarity to pairs of observations known to have the same label, and a small similarity to pairs known to have different labels.

A. APPENDIX: MULTIWAY NORMALIZED CUT

The normalized cut between any two clusters k, and k' is defined as (Shi and Malik 2000; Meila and Xu 2003; Yu and Shi 2003)

$$\mathrm{NCut}(k,k') = \left(\frac{1}{\mathrm{vol}\,(k)} + \frac{1}{\mathrm{vol}\,(k')}\right)\sum_{i=1}^{n}\sum_{j=1}^{n} z_{ki} z_{k'j} k_{ij}, \tag{A.1}$$

where $\mathrm{vol}\,(\ell) = \sum_{i=1}^{n}\sum_{j=1}^{n} z_{\ell i} k_{ij}$, $\ell = 1,\dots,q$. The MNCut of any given partition is then defined as

$$\mathrm{MNCut} = \sum_{k=1}^{q}\sum_{k'=k+1}^{q} \mathrm{NCut}\,(k,k') = \frac{1}{2}\sum_{k=1}^{q}\sum_{k'=1}^{q}\mathrm{NCut}\,(k,k') - \frac{1}{2}\sum_{k=1}^{q}\mathrm{NCut}\,(k,k). \tag{A.2}$$

The goal of MNCut is to find the set of labels $\{z_{ki}\}$ that minimize (A.2). Using (A.1), one easily obtains

$$\mathrm{MNCut} = q - \sum_{k=1}^{q} \frac{\sum_{j=1}^{n}\sum_{i=1}^{n} z_{kj} z_{ki} k_{ij}}{\sum_{i=1}^{n}\sum_{j=1}^{n} z_{ki} k_{ij}} = q - \sum_{k=1}^{q} w_k \gamma_k R_k,$$

where $w_k = n/\sum_{j=1}^{n} \hat{p}(x_j|k)$. Writing $\sum_{k=1}^{q} w_k \gamma_k R_k$ as

$$\sum_{k=1}^{q} \frac{\sum_{i=1}^{n} z_{ki}\hat{p}(x_i|k)}{\sum_{j=1}^{n}\hat{p}(x_j|k)} = \sum_{k=1}^{q} \frac{\text{mass assigned to cluster } k \text{ given cluster } k}{\text{total mass in cluster } k \text{ given cluster } k}$$

one sees that MNCut tries to maximize the total mass assigned to the clusters so that the data points assigned to the corresponding clusters account for almost all the mass associated with the conditional densities. A straightforward calculation shows that, as in the kernel K-means and Potts model clustering cases, $\sum_{k=1}^{q} w_k \gamma_k R_k$ is another way to write (1.2) with weights given by

$$w(i,j,\{z_{ki}\},k) = \left\{ \begin{array}{ll} 0 & \text{if} \quad \delta_{ij} = 0 \\ \left(\sum_{i=1}^{n} z_{ki} \sum_{j=1}^{n} k_{ij}\right)^{-1} & \text{if} \quad z_{ki} = z_{kj} = 1 \end{array} \right\}.$$

Hence, minimizing (A.2) is again linked to some sort of weighted K-means procedure with cluster dependent weights w_k. Indeed, it is straightforward to verify that a weighted K-means with weights w_k maximizes $\sum_{k=1}^{q} w_k\gamma_k(R_k - 1)$. Note that $\sum_{k=1}^{q} w_k\gamma_k = \sum_{k=1}^{q} 1/m(k)$, where $m(k) = \sum_{j=1}^{n}\hat{p}(x_j|k)/n_k$ is the “average” mass in cluster k. The weighted K-means with weights given by w_k’s penalizes clusterings with large variations in average masses across the clusters. Thus, unlike the weighted K-means with weights γ_k’s that penalizes unequal cluster sizes, the MNCut induced weighted K-means penalizes unequal cluster masses.

ACKNOWLEDGMENTS

The authors thank the referees for their valuable comments. This research was partially supported by NSERC grant 327689-06, and by NSF grant 0505824.

[Received January 2006. Revised September 2007.]

REFERENCES

Abramson, I. S. (1982), "On Bandwidth Variation in Kernel Estimates—A Square Root Law," *The Annals of Statistics,* 10, 1217–1223.

Agrawal, H., and Domany, E. (2003), "Potts Ferromagnets on Coexpressed Gene Networks: Identifying Maximally Stable Partitions," *Physical Review Letters,* 90, 158102.

Azzalini, A., and Bowman, A. W. (1990), "A Look at Some Data on the Old Faithful Geyser," *Applied Statistics,* 39, 357–365.

Banfield, J. D., and Raftery, A. E. (1993), "Model-Based Gaussian and Non-Gaussian Clustering," *Biometrics,* 49, 803–821.

Blatt, M., Wiseman, S., and Domany, E. (1996a), "Super-Paramagnetic Clustering of Data," *Physical Review Letters,* 76, 3251–3255.

——— (1996b), "Clustering Data Through an Analogy to the Potts Model," in *Advances in Neural Information Processing Systems,* eds. D. S. Touretzky, M. C. Mozer, and M. E. Hasselmo, Cambridge: MIT Press, pp. 416–422.

——— (1997), "Data Clustering Using a Model Granular Magnet," *Neural Computation,* 9, 1805–1842.

Breiman, L., Meisel, W., and Purcell, E. (1977), "Variable Kernel Estimates of Multivariate Densities," *Technometrics*, 19, 135–144.

Buta, R. (1987), "The Structure and Dynamics of Ringed Galaxies, III: Surface Photometry and Kinematics of the Ringed Nonbarred Spiral NGC7531," *The Astrophysical Journal Supplement Series,* 64, 1–37.

Celeux, G., and Govaert, G. (1995), "Gaussian Parsimonious Clustering Models," *Pattern Recognition,* 28, 781–793.

Chambers, J. M., and Hastie, T. J. (eds.) (1992), "Statistical Models in S," Pacific Grove, CA: Wadsworth and Brooks, p. 352.

Creutz, M. (1979), "Confinement and the Critical Dimensionality of Space–Time," *Physics Review Letters,* 43, 553–556.

Cho, R. J., Campbell, M. J., Winzeler, E. A., Steinmetz, L., Conway, A., Wodicka, L., Wolfsberg, T. G., Gabrielian, A. E., Landsman, D., Lockhart, D. J., and Davis, R. W. (1998), "A Genome-Wide Transcriptional Analysis of the Mitotic Cell Cycle," *Molecular Cell,* 2, 65–73.

Dhillon, I. S., Guan, Y., and Kulis, B. (2004), "Kernel *k*-means, Spectral Clustering and Normalized Cuts," in *Proceedings of the Tenth ACM SIGKDD International Conference,* pp. 551–556.

Dimitriadou, E., Weingessel, A., and Hornik, K. (2001), "Voting-Merging: An Ensemble Method for Clustering," in *Proceedings of the International Conference on Artificial Neural Networks,* pp. 217–224.

Domany, E. (2003), "Cluster Analysis of Gene Expression Data," *Journal of Statistical Physics,* 110 (3-6), 1117.

Domany, E., Blatt, M., Gdalyahu, Y., and Weinshall, D. (1999), "Superparamagnetic Clustering of Data: Application to Computer Vision," *Computer Physics Communications,* 121–122, 5–12.

Dudoit, S., and Fridlyand, J. (2003), "Bagging to Improve the Accuracy of a Clustering Procedure," *Bioinformatics*, 19, 1090–1099.

Edwards, R. G., and Sokal, A. D. (1988), "Generalization of the Fortuin-Kasteleyn-Swendsen-Wang Representation and Monte Carlo Algorithm," *Physical Review D,* 38, 2009–2012.

Einav, U., Tabach, Y., Getz, G., Yitzhaky, A., Ozbek, U., Amarglio, N., Izraeli, S., Rechavi, G., and Domany, E. (2005), "Gene Expression Analysis Reveals a Strong Signature of an Interferon-Unduced Pathway in Childhood Lymphoblastic Leukemia as Well as in Breast and Ovarian Cancer," *Oncogene,* 24, 6367–6375.

Eisen, M.B., Spellman, P.T., Brown, P.O., and Botstein D. (1998), "Cluster Analysis and Display of Genome-Wide Expression Patterns," *PNAS*, 95, 14863–14868.

Fern, X. Z., and Brodley, C. E. (2003), "Random Projection for High Dimensional Data Clustering: A Cluster Ensemble Approach," in *Proceedings of the 20th International Conference on Machine Learning*, 186–193.

Fred, A. L. N., and Jain, A. K. (2002), "Data Clustering Using Evidence Accumulation," in *Proceedings of the 16th International Conference Pattern Recognition,* pp. 276–280.

Geman, S., and Geman, D. (1984), "Stochastic Relaxation, Gibbs Distributions and the Bayesian Restoration of Images," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 6, 721–741.

Getz, G., Levine, E., Domany, E., and Zhang, M. (2000), "Super-Paramagnetic Clustering of Yeast Gene Expression Profiles," *Physica A,* 279 (1–4), 457–464.

Geyer, C. J. (1991), "Markov Chain Monte Carlo Maximum Likelihood," in *in Computing Science and Statistics: Proceedings of the 23rd Symposium on the Interface*, ed. E. M. Keramigas, pp. 156–163.

Geyer, C. J., and Thompson, E. .A. (1995), "Annealing Markov Chain Monte Carlo with Applications to Ancestral Inference," *Journal of the American Statistical Association*, 90, 909–920.

Girolami, M. (2002), "Mercer Kernel Based Clustering in Feature Space," *IEEE Transactions on Neural Networks*, 13, 669–688.

Grenander, U. (1983), "Tutorial on Pattern Theory," Technical Report, Division of Applied Mathematics, Brown University.

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

Hubert, L., and Arabie, P. (1985), "Comparing Partitions," *Journal of Classification*, 2, 193–218.

Kent Ridge Bio-Medical Dataset Repository, available online at *http://sdmc.lit.org.sg/GEDatasets/Datasets.html.*

Marinari, E., and Parisi, G. (1992), "Simulated Tempering: A New Monte Carlo Scheme," *Europhysics Letters,* 19, 451.

McKinney, S. (1995), "Autopaint: A Toolkit for Visualizing Data in 4 or more Dimensions," unpublished PhD Thesis, University of Washington, Seattle.

Meila, M., and Xu, L (2003), "Multiway Cuts and Spectral Clustering," Technical Report, University of Washington, Seattle.

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., and Teller, E. (1953), "Equations of State Calculations by Fast Computing Machines," *Journal of Chemical Physics*, 21, 1087–1092.

Milligan, G. W., and Cooper, M. C. (1986), "A Study of the Comparability of External Criteria for Hierarchical Cluster Analysis," *Multivariate behavioral Research*, 21, 441–458.

Ott, T., Kern, A., Schuffenhauer, A., Popov, M., Acklin, P., Jacoby, E., and Stoop, R. (2004), "Sequential Superparamagnetic Clustering for Unbiased Classification of High-Dimensional Chemical Data," *Journal of Chemical Information and Computer Sciences*, 44, 1358–1364.

Ott, T., Kern, A., Steeb, W., and Stoop, R. (2005), "Sequential Clustering: Tracking Down the Most Natural Clusters," *Journal of Statistical Mechanics: Theory and Experiment*, 11, P11014.

Quiroga, R. Q., Nadasdy, Z., and Ben-Shaul, Y. (2004), "Unsupervised Spike Detection and Sorting with Wavelets and Superparamagnetic Clustering," *Neural Computation*, 16, 1661–1687.

Rand, W. M. (1971), "Objective Criteria for the Evaluation of Clustering Methods," *Journal of the American Statistical Association*, 66, 846–850.

Reichardt, J., and Bornholdt, S. (2004), "Detecting Fuzzy Community Structures in Complex Networks with a Potts Model," *Physical Review Letters*, 93, 218701.

Shi, J., and Malik, J. (2000), "Normalized Cuts and Image Segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22, 888–905.

Silverman, B. W. (1986), *Density Estimation for Statistics and Data Analysis*, London: Chapman and Hall.

Sokal, A. D. (1996), "Monte Carlo Methods in Statistical Mechanics: Foundations and New Algorithms," *Lectures at the Cargese Summer School on Functional Integration: Basics and Applications*.

Stanberry, L., Murua, A., and Cordes, D. (2007), "Functional Connectivity Mapping using the Ferromagnetic Potts Spin Model," *Human Brain Mapping*, 29, 422–440.

Stanberry, L., Nandy, R., and Cordes, D. (2003), "Cluster Analysis of fMRI Data using Dendrogram Sharpening," *Human Brain Mapping,* 20, 201–219.

Swendsen, R. H., and Wang, J. S. (1987), *Physical Review Letters,* 58, 86.

Tantrum, J., Murua, A., and Stuetzle, W. (2003), “Assessment and Pruning of Hierarchical Model Based Clustering,” *KDD 2003,* Washington, DC, USA.

——— (2004), “Hierarchical Model-Based Clustering of Large Datasets through Fractionation and Refractionation,” *Information Systems,* 29, 315–326.

Topchy, A., Jain, A. K., and Punch, W. (2005), “Clustering Ensembles: Models of Consensus and Weak Partitions,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27, 1866–1881.

Tukey, J. W. (1949), “Comparing Individual Means in the Analysis of Variance,” *Biometrics*, 5, 99–114.

Wang, S., and Swendsen, R. H. (1990), “Cluster Monte Carlo Algorithms,” *Physica A,* 167, 565–579.

Wiseman, S., Blatt, M., and Domany, E. (1998), “Superparamagnetic Clustering of Data,” *Physical Review E*, 57, 3767–3783.

Wolff, U. (1989), “Collective Monte Carlo Updating for Spin Systems,” *Physical Review Letters*, 62, 361–364.

Yeoh, E. J., Ross, M. E., Shurtleff, S. A., Williams, W. K., Patel, D., Mahfouz, R., Behm, F. G., Raimondi, S. C., Relling, M. V., Patel, A., Cheng, C., Campana, D., Wilkins, D., Zhou, X., Li, J., Liu, H., Pui, C. H., Evans, W. E., Naeve, C., Wong, L., and Downing, J. R. (2002), “Classification, Subtype Discovery, and Prediction of Outcome in Pediatric Acute Lymphoblastic Leukemia by Gene Expression Profiling,” *Cancer Cell*, 1, 133–143.

Yeung, K. Y., Fraley, C., Murua, A., Raftery, A. E., and Ruzzo, W. L. (2001), “Model-Based Clustering and Data Transformations for Gene Expression Data,” *Bioinformatics*, 17, 977–987.

Yu, S. X., and Shi, J. (2003), “Multiclass Spectral Clustering,” *International Conference on Computer Vision, 2003.*

Zhang, R., and Rudnicky, A. I. (2002), “A Large Scale Clustering Scheme for Kernel K-means,” in *Proceedings of the 16th International Conference Pattern Recognition*, pp. 289–292.