J. R. Statist. Soc. B (2008)
70, *Part* 1, *pp.* 119–139

Clustering using objective functions and stochastic search

James G. Booth,

Cornell University, Ithaca, USA

and George Casella and James P. Hobert

University of Florida, Gainesville, USA

[Received December 2005. Final revision August 2007]

Summary. A new approach to clustering multivariate data, based on a multilevel linear mixed model, is proposed. A key feature of the model is that observations from the same cluster are correlated, because they share cluster-specific random effects. The inclusion of cluster-specific random effects allows parsimonious departure from an assumed base model for cluster mean profiles. This departure is captured statistically via the posterior expectation, or best linear unbiased predictor. One of the parameters in the model is the true underlying partition of the data, and the posterior distribution of this parameter, which is known up to a normalizing constant, is used to cluster the data. The problem of finding partitions with high posterior probability is not amenable to deterministic methods such as the EM algorithm. Thus, we propose a stochastic search algorithm that is driven by a Markov chain that is a mixture of two Metropolis–Hastings algorithms—one that makes small scale changes to individual objects and another that performs large scale moves involving entire clusters. The methodology proposed is fundamentally different from the well-known finite mixture model approach to clustering, which does not explicitly include the partition as a parameter, and involves an independent and identically distributed structure.

Keywords: Bayesian model; Best linear unbiased predictor; Cluster analysis; Linear mixed model; Markov chain Monte Carlo methods; Metropolis–Hastings algorithm; Microarray; Quadratic penalized splines; Set partition; Yeast cell cycle

1. Introduction

Clustering and classification are some of the most fundamental data analysis tools in use today. Many standard clustering algorithms are based on the assumption that the measurements to be clustered are realizations of random vectors from some parametric statistical model. These models usually place no restriction on the mean structure via covariates or otherwise. However, in many applications there is potential for parsimonious representation of the mean. For example, microarray experiments often yield time-series-type data where each p-dimensional vector consists of measurements at p different time points. In such cases, it seems natural to model the mean via regression, especially when tempered with the ability to detect clusters that are well defined but deviate from a specified parametric form. We provide general clustering methods that achieve this balance, i.e. they allow us to exploit covariate information without overemphasizing conformance with a model. Related ideas have been considered in some recent works including Serban and Wasserman (2005), Hitchcock *et al.* (2006), McLachlan *et al.* (2004),

Address for correspondence: James G. Booth, Department of Biological Statistics and Computational Biology, Cornell University, Ithaca, NY 14850, USA.
E-mail: jb383@cornell.edu

 1369–7412/08/70119

Celeux *et al.* (2005) and Heard *et al.* (2006), whose model is a special case of the model that is considered here.

The basic clustering problem is simple to state. Given a set of n distinguishable objects, we wish to distribute the objects into groups or clusters in such a way that the objects within each group are similar whereas the groups themselves are different. Let the integers $\mathbb{N}_n := \{1, 2, \ldots, n\}$ serve as labels for our n distinguishable objects. In mathematical terms, the output of a clustering algorithm is a partition of $\mathbb{N}_n$, i.e. an unordered collection of non-empty subsets of $\mathbb{N}_n$. Unfortunately, it is not always clear exactly how to quantify the similarity of objects within clusters nor the difference between clusters. However, suppose that, in addition to the n objects, there is an objective function $\pi : \mathbb{P}_n \to \mathbb{R}^+$, where $\mathbb{P}_n$ denotes the set of all possible partitions of $\mathbb{N}_n$, which assigns a score to each partition reflecting the extent to which it achieves the overall clustering goal that was described above. In this case, the cluster analysis is tantamount to the straightforward optimization problem of finding the partition with the highest score. In this paper we propose a general method for constructing objective functions, through the use of linear mixed models, that take into account covariate information.

Suppose that the objects to be clustered are n p-dimensional vectors denoted by $Y_i = (Y_{i1}, \ldots, Y_{ip})^{\mathrm{T}}$, $i = 1, 2, \ldots, n$. The standard, model-based approach to clustering (see, for example, McLachlan and Basford (1988) and McLachlan and Peel (2000)) begins with the assumption that these n vectors are realizations of n independent and identically distributed random vectors from the K-component mixture density

$$\sum_{k=1}^{K} \tau_k \, f(\cdot; \theta_k), \tag{1}$$

where K is a fixed positive integer in $\mathbb{N}_n$, $\tau_k \in (0, 1)$, $\Sigma_{k=1}^{K} \tau_k = 1$, $\{ f(\cdot; \theta) : \theta \in \Theta \}$ is a parametric family of densities on $\mathbb{R}^p$ and $\theta_k \in \Theta$ is an unknown vector of parameters that is associated with the kth component. A partition of the data is typically obtained as a by-product of an EM algorithm that is designed to find the maximum likelihood estimates of the parameters, $\{(\tau_k)_{k=1}^{K}, (\theta_k)_{k=1}^{K}\}$. The missing data are n multinomial K-vectors, $W_1, \ldots, W_n$, indicating the origin of each Y_i. A partition of the data is obtained through the so-called *maximum likelihood classification rule*, which assigns observation i to the mixture component that is associated with the largest co-ordinate of the conditional expectation of W_i calculated during the final E-step.

The motivation for using expression (1) as the basis for cluster analysis must surely be the fact that this model would be correct if the data were actually a random sample from a heterogeneous population with K groups whose sizes are proportional to the τ_i. However, in many applications of cluster analysis this sampling scheme is quite unrealistic. We contend that a more realistic assumption is that there is some fixed unknown partition of $\mathbb{N}_n$, ω, that has $c = c(\omega)$ clusters denoted by $\mathcal{C}_1, \ldots, \mathcal{C}_c$ and that the data are a realization from a density of the form

$$f(y | \theta_1, \ldots, \theta_c, \omega) = \prod_{k=1}^{c} \prod_{j \in \mathcal{C}_k} f(y_j | \theta_k). \tag{2}$$

Of course, $\cup_{k=1}^{c} \mathcal{C}_k = \mathbb{N}_n$ and $\mathcal{C}_i \cap \mathcal{C}_j = \emptyset$ whenever $i \neq j$. Note that, unlike the mixture model, model (2) contains a parameter ω that is directly relevant to the basic clustering problem. Another standard approach to clustering is to maximize equation (2) in (ω, θ), to call the result $(\hat{\omega}, \hat{\theta})$, and then to use $\hat{\omega}$ as the partition (Scott and Symons, 1971; Symons, 1981; Banfield and Raftery, 1993). In this context, equation (2) is called the *classification likelihood*.

We propose an objective function that is based on a generalization of model (2) that takes into account covariate information and allows for dependence among data vectors in the same

cluster. To be specific, our objective function is the posterior distribution $\pi(\omega|y_1, \ldots, y_n)$, which is constructed by placing priors on all the parameters in the model and marginalizing over all parameters except ω. This general approach was suggested several decades ago in Binder (1978). However, stochastic search methods for finding partitions with high posterior probability were not feasible at that time. Also, the linear mixed model formulation that is proposed here is a generalization of that proposed in Heard *et al.* (2006). In particular, our model allows for the data vectors within a cluster to be correlated, which allows for parsimonious representation of the cluster means through the use of penalized splines.

Finding the partitions that yield the highest values of the objective function is a challenging optimization problem. The reason is that the total number of partitions of $\mathbb{N}_n$, $B_n = \#(\mathbb{P}_n)$, which is called the *Bell number* (Stanley (1997), page 33), grows extremely rapidly with n; for example, $B_{40} = 1.6 \times 10^{35}$ and $B_{100} = 4.8 \times 10^{115}$. Thus, even for moderately large n, it is computationally infeasible to enumerate $\mathbb{P}_n$. Not surprisingly, standard clustering algorithms typically fail to optimize any objective function globally.

A second contribution of this paper is the development of a stochastic search algorithm for finding the maximizer of the objective function. The basic idea is to construct a Metropolis–Hastings (MH) Markov chain whose stationary distribution is proportional to the objective function. Of course, the key to success with the MH algorithm is the choice of the candidate transition kernel. We propose a mixture of two MH algorithms—one that makes small scale changes to individual objects and another that performs large scale moves involving entire clusters. Thus, our approach is contrary to the claim that was made in Heard *et al.* (2006), section 7, that 'it is rather difficult to construct efficient dimension-changing moves in the vast space of possible clusterings'. They used a deterministic, greedy, agglomerative hierarchical algorithm.

In general, partitioning the data by finding the maximizer of an objective function alleviates several well-known difficulties that are associated with the standard clustering procedure. For example, one practical problem with the standard, EM-based procedure for the mixture model that was described above is that K must be fixed *a priori*. Fraley and Raftery (2002) suggested optimizing the Bayes information criterion to solve this problem, but this means that the EM algorithm must be run once for every possible value of K that the user wishes to consider. Although this may not be overly burdensome from a computational standpoint, it is not very satisfying. In contrast, our objective function can be evaluated at any given partition, regardless of the number of clusters, and hence the fixed K problem is not an issue.

One might argue that the methods that are proposed in this paper are computationally burdensome relative to more conventional clustering algorithms because of the stochastic search ingredient. However, it is well known that methods such as the K-means and the mixture model-based approach are sensitive to starting values. For example, the K-means algorithm can converge to substantially different solutions when rerun with a different random-number generator seed. Different solutions resulting from different starting values must be compared, presumably by using an objective function (such as a least squares criterion for K-means). Since it is impossible to rerun the algorithm from every possible starting point, the only way to gain confidence in the solutions that are provided by these algorithms is to perform some type of stochastic search. This fact has been recognized by other researchers. For example, Selim and Alsutan (1991) attempted to minimize the K-means least squares criterion by using a simulated annealing algorithm, and Celeux and Govaert (1992) proposed two stochastic clustering methods that were based on the EM algorithm. The approach that is described in this paper can be viewed as a formalization of this process that leads to a probability-based criterion for selecting good partitions that is based on a flexible class of statistical models.

The remainder of this paper is organized as follows. In Section 2, we describe the mixed model framework that leads to a probability-based objective function for cluster analysis. Our stochastic search procedure for maximizing the objective function is discussed in Section 3. Section 4 contains the results of a simulation study in which the model-based algorithm proposed is compared with the K-means method. In Section 5 we apply the method to a well-known data set that was obtained from a microarray experiment concerning the yeast cell cycle, and a second microarray time course data set concerning corneal wound healing in rats. We conclude in Section 6 with some discussion.

2. Model-based objective functions

Suppose that the data vector Y_i that is measured on the ith object actually consists of r replicate profiles, i.e.

$$Y_i = (Y_{i11}, \ldots, Y_{i1p}, \ldots, Y_{ir1}, \ldots, Y_{irp})^{\mathrm{T}} = (Y_{i1}^{\mathrm{T}}, \ldots, Y_{ir}^{\mathrm{T}})^{\mathrm{T}},$$

for $i = 1, \ldots, n$. A particular setting where this data structure arises is microarray experiments in which replicate measurements are made on each gene (Celeux *et al.*, 2005). Of course, if there is no replication, the second subscript can be omitted. We now describe a model for the data vectors, $Y_1, \ldots, Y_n$.

Fix $\omega \in \mathbb{P}_n$ and let $\theta = (\theta_k)_{k=1}^{c(\omega)}$ denote a set of cluster-specific parameter vectors where $\theta_k \in \Theta$. We assume that, given (ω, θ), the data vectors are partitioned into c clusters according to ω and that the clusters of data are mutually independent. However, the random vectors within each cluster may be correlated and the joint distribution depends on the value of the corresponding θ_k. In the most general case, we suppose that dependence among the Y_i within a cluster, and among replicate profiles from the same object, is induced by cluster- and object-specific random effects. To be specific, for $l \in \{1, 2\}$, let $\{g_l(\cdot|\theta) : \theta \in \Theta\}$ denote a parametric family of densities, each having support $S_l \subset \mathbb{R}^{s_l}$, and let $\{h(\cdot|u, v, \theta) : u \in S_1, v \in S_2, \theta \in \Theta\}$ denote another family with common support that is a subset of $\mathbb{R}^p$. Then, for a given fixed value of (ω, θ), the joint density of $Y = (Y_1^{\mathrm{T}}, \ldots, Y_n^{\mathrm{T}})^{\mathrm{T}}$ is given by

$$f(y|\theta, \omega) = \prod_{k=1}^{c(\omega)} \int_{S_2} \left[\prod_{i \in \mathcal{C}_k} \int_{S_1} \left\{ \prod_{j=1}^{r} h(y_{ij}|u_i, v_k, \theta_k) \right\} g_1(u_i|\theta_k)\, \mathrm{d}u_i \right] g_2(v_k|\theta_k)\, \mathrm{d}v_k. \tag{3}$$

The density h may depend on known covariates, but this is suppressed notationally. This model is similar in structure to the 'parametric partition models' that were used by Hartigan (1990) and Crowley (1997) and also to a model that was used by Consonni and Veronese (1995). However, in those models, there is within-cluster independence given (ω, θ). Furthermore, these researchers were not specifically concerned with cluster analysis.

The objective function that we propose is the marginal posterior of ω, which is calculated by putting priors on ω and θ and then integrating the nuisance parameter θ out of the full posterior distribution. Since the dimension of θ depends on ω, it is natural to use a hierarchical prior of the form $\pi(\theta|\omega)\,\pi(\omega)$ (see Green (1995)). As a prior for $\omega \in \mathbb{P}_n$, we use

$$\pi_n(\omega) = \frac{\Gamma(m) m^{c(\omega)}}{\Gamma(n+m)} \prod_{k=1}^{c(\omega)} \Gamma(n_k), \tag{4}$$

where $n_k = \#(\mathcal{C}_k)$ and $m > 0$ is a parameter. This distribution was used as a prior in Crowley (1997). Clearly, as m decreases, more weight is put on the set of partitions having a small number of clusters. In fact, we show in Appendix A that, if $\omega \sim \pi_n$, then the expected number of clusters

is given by

$$E\{c(\omega)\} = m \sum_{i=0}^{n-1} \frac{1}{m+i}.$$

This function is clearly increasing in m and has limiting values of 1 and n as m approaches 0 and ∞ respectively. Not surprisingly, the hyperparameter m is of critical importance, and use of a default value, such as $m = 1$, is not a good general practice. As we shall demonstrate later, a reasonable strategy seems to be to choose m sufficiently small that the *a priori* expected number of clusters is very close to its lower bound of 1.

Distribution (4) has two desirable properties. Firstly, $\pi_n(\omega)$ depends only on $c(\omega)$ and n_1, $n_2, \ldots, n_{c(\omega)}$, so any two partitions that share the same values of $c(\omega)$ and $n_1, n_2, \ldots, n_{c(\omega)}$ (i.e. differing only by a permutation of the labels $\{1, 2, \ldots, n\}$) will have the same probability under π_n. For example, when $n = 3$ the Bell number is 5 and the partitions are

$$\omega_1 : \{1,2,3\}, \qquad \omega_2 : \{1,2\}\{3\}, \qquad \omega_3 : \{1,3\}\{2\}, \qquad \omega_4 : \{2,3\}\{1\}, \qquad \omega_5 : \{1\}\{2\}\{3\}. \tag{5}$$

In this case $\pi_3(\omega_2) = \pi_3(\omega_3) = \pi_3(\omega_4)$. This property is called *exchangeability* (Pitman, 2005) and is a minimal requirement in our context given that the assignment of the labels $\{1, 2, \ldots, n\}$ to the n data vectors is arbitrary.

Secondly, these distributions enjoy a form of *consistency* that is now described. Consider the action of *deleting* object $n+1$ from $\omega \in \mathbb{P}_{n+1}$, which results in an element of $\mathbb{P}_n$. Suppose that $\omega^* \in \mathbb{P}_n$ and let $S \subset \mathbb{P}_{n+1}$ denote the set of elements that become ω^* when $n+1$ is deleted. The consistency property is that $\Sigma_{\omega \in S} \pi_{n+1}(\omega) = \pi_n(\omega^*)$ (see McCullagh and Yang (2006), and the references therein). If this property were to fail, then we could, for example, have

$$\pi_3[\{1,2,3\}] \neq \pi_4[\{1,2,3,4\}] + \pi_4[\{1,2,3\}\{4\}],$$

which seems unreasonable. Why should the prior probability that the first three data points are in the same cluster depend on whether $n = 3$ or $n = 4$? We note that the priors that were used by Consonni and Veronese (1995) and Heard *et al.* (2006), equation (9), satisfy the exchangeability property, but not the consistency property.

As for $\pi(\theta|\omega)$, we assume that, conditional on ω, the random vectors $\theta_1, \ldots, \theta_c$ are exchangeable, but the precise form will depend on the specific structure of the model. The marginal posterior of ω is given by

$$\pi(\omega|y) \propto \int f(y|\theta,\omega)\, \pi(\theta|\omega)\, \pi_n(\omega)\, \mathrm{d}\theta. \tag{6}$$

We propose to use this marginal posterior as an objective function for cluster analysis.

In this paper, we focus on a particular version of equation (3) in which the joint distribution of the response vectors in $\mathcal{C}_k$, given (θ, ω), is described by a linear mixed model. To avoid excessive subscripting, assume for the time being that $\mathcal{C}_k = \{1, \ldots, n_k\}$. We assume that the data vectors corresponding to objects in the kth cluster follow the model

$$Y_{ij} = X\beta_k + Z_1 U_i + Z_2 V_k + \varepsilon_{ij}, \tag{7}$$

where $i = 1, \ldots, n_k$, $j = 1, \ldots, r$, the ε_{ij} are independent and identically distributed $N_p(0, \sigma_k^2 I_p)$, the U_i are independent and identically distributed $N_{s_1}(0, \lambda_1 \sigma_k^2 I_{s_1})$ and $V_k \sim N_{s_2}(0, \lambda_2 \sigma_k^2 I_{s_2})$. We assume that the ε_{ij}, U_i and V_k are mutually independent. In terms of the general model, we have taken $g_l(\cdot; \theta_k)$ to be an s_l-variate normal density with zero mean and covariance matrix $\lambda_l \sigma_k^2 I_{s_l}$, for $l \in \{1, 2\}$. (Of course, if there are no replicates, the $Z_1 U_i$-term would be absent from the

model.) The matrix X is $p \times q$ ($q < p$) with full column rank, β_k is a q-dimensional regression parameter and the matrix Z_l is $p \times s_l$ with rank $s_l^* \leqslant s_l$. In this case, $\theta_k = (\beta_k, \sigma_k^2)$, and λ_1 and λ_2 are tuning parameters. (A default, data-driven, method for choosing λ_1 and λ_2 is proposed in Section 4.) Specification of the model is completed by taking the prior

$$\pi(\beta, \sigma^2|\omega) \propto \prod_{k=1}^{c(\omega)} (1/\sigma_k^2)^{\alpha+1}.$$

We now work out the exact form of expression (6) under these specific assumptions.

Let Y_k^* denote the $n_k rp \times 1$ vector consisting of all the responses in $\mathcal{C}_k$ stacked on top of one another. Then, it is readily verified that $Y_k^* \sim N_{n_k rp}\{(1_{n_k r} \otimes X)\beta_k, \sigma_k^2 M_k\}$, where $M_k = I_{n_k} \otimes A + J_{n_k} \otimes B$, 1_m is a 1-vector of length m, $J_m = 1_m 1_m^{\mathrm{T}}$, and the matrices A and B are given by $A = I_r \otimes I_p + J_r \otimes \lambda_1 Z_1 Z_1^{\mathrm{T}}$ and $B = J_r \otimes \lambda_2 Z_2 Z_2^{\mathrm{T}}$. Let $\bar{Y}_k$ represent the mean profile in the kth cluster, i.e. the average of the $n_k r$ p-dimensional vectors that comprise Y_k^*. Also define

$$W_k = (I_p + r\lambda_1 Z_1 Z_1^{\mathrm{T}} + n_k r \lambda_2 Z_2 Z_2^{\mathrm{T}})^{-1}.$$

In Appendix B, it is shown that, for a fixed ω, the statistics $\hat{\beta}_k = (X^{\mathrm{T}} W_k X)^{-1} X^{\mathrm{T}} W_k \bar{Y}_k$ and

$$\hat{\sigma}_k^2 = \frac{1}{n_k rp} \sum_{i \in \mathcal{C}_k} (Y_i - 1_r \otimes \bar{Y}_k)^{\mathrm{T}} A^{-1} (Y_i - 1_r \otimes \bar{Y}_k) + \frac{1}{p} (\bar{Y}_k - X\hat{\beta}_k)^{\mathrm{T}} W_k (\bar{Y}_k - X\hat{\beta}_k), \tag{8}$$

for $k = 1, \ldots, c(\omega)$, jointly maximize equation (3). Furthermore, the joint density of the measurements on the objects in cluster k can be written as

$$f(y_k^*|\theta, \omega) = |2\pi\sigma_k^2 M_k|^{-1/2} \exp\left[-\frac{n_k r}{2\sigma_k^2} \{(\beta_k - \hat{\beta}_k)^{\mathrm{T}} X^{\mathrm{T}} W_k X (\beta_k - \hat{\beta}_k) + p\hat{\sigma}_k^2\} \right]. \tag{9}$$

Let $\delta_1, \ldots, \delta_p$ be the eigenvalues of the matrix $D = (I_p + r\lambda_1 Z_1 Z_1^{\mathrm{T}})^{-1} \lambda_2 Z_2 Z_2^{\mathrm{T}}$ and note that

$$|M_k| = |A|^{n_k} \prod_{i=1}^{p} (1 + n_k r \delta_i).$$

Now, integrating the product of the density (9) and $(1/\sigma_k^2)^{\alpha+1}$ with respect to β_k and σ_k^2, we obtain

$$\frac{2^{\alpha} \Gamma\{(n_k rp - q)/2 + \alpha\}}{\pi^{(n_k rp - q)/2} (n_k rp \hat{\sigma}_k^2)^{(n_k rp - q)/2 + \alpha} |n_k r X' W_k X|^{1/2} |A|^{n_k/2} \prod_{i=1}^{p} (1 + n_k r \delta_i)^{1/2}}.$$

Finally, taking the product over the clusters, and multiplying by the prior $\pi_n(\omega)$, yields our proposed objective function for cluster analysis:

$$\pi(\omega|y) \propto \pi_n(\omega) \prod_{k=1}^{c(\omega)} \frac{2^{\alpha} \pi^{q/2} \Gamma\{(n_k rp - q)/2 + \alpha\} \prod_{i=1}^{p} (1 + n_k r \delta_i)^{-1/2}}{(n_k r)^{q/2} (n_k rp \hat{\sigma}_k^2)^{(n_k rp - q)/2 + \alpha} |X^{\mathrm{T}} W_k X|^{1/2}}. \tag{10}$$

If each measurement is rescaled by a factor a, say, then the value of $\hat{\sigma}_k^2$ changes to $a^2 \hat{\sigma}_k^2$, resulting in a multiplicative change in expression (10) of $\Pi_{k=1}^{c} a^{-(n_k rp - q + 2\alpha)} \propto a^{c(q - 2\alpha)}$. This motivates the choice $q/2$ for the hyperparameter α on the grounds of scale invariance. This is of practical importance in microarray studies, for example, where the responses are log-expression ratios, and the choice of base is arbitrary.

The posterior $\pi(\omega|y)$ is an intuitively reasonable objective function for clustering in that it rewards large homogeneous clusters, i.e. those in which $\hat{\sigma}_k^2$ is small and n_k is large, but also includes penalties for extreme partitions in which $c(\omega)$ is very small or very large. Moreover, examining the form of $\hat{\sigma}_k^2$ shows that there are two distinct parts to this variance. The first piece measures a within-cluster variance and will help to identify clusters of similar objects even if they do not follow the specified model. The second piece measures lack of fit and will identify clusters of objects whose average profile closely follows the assumed model that is determined by the matrix X.

Another nice feature of the mixed model approach is that the predicted mean for a given cluster is a compromise between a projection onto the columns of X and the columns of $X|Z_2$. This allows cluster means to deviate from the base model while retaining parsimony. Owing to the flat prior specification for β, provided that $n_k rp - q > 1$, the posterior expectation of $X\beta_k + Z_2 V_k$ is equal to the best linear unbiased predictor given by

$$\begin{aligned} X\hat{\beta}_k + Z_2\hat{V}_k &= X\hat{\beta}_k + n_k r\lambda_2 Z_2 Z_2^{\mathrm{T}} W_k(\bar{Y}_k - X\hat{\beta}_k) \\ &= n_k r\lambda_2 Z_2 Z_2^{\mathrm{T}} W_k \bar{Y}_k + (I - n_k r\lambda_2 Z_2 Z_2^{\mathrm{T}} W_k) X\hat{\beta}_k. \end{aligned} \tag{11}$$

Thus, predicted values in large clusters shrink towards the more complex model. In particular, if $Z_1 = 0$ and $Z_2 = I$, the predicted values in the kth cluster are a convex combination of $X\hat{\beta}_k$ and $\bar{Y}_k$, the estimate that is based on a completely unstructured mean.

The model that was considered by Heard *et al.* (2006), section 3, is a special case of model (7) in which $X = Z_1 = 0$. One key difference between their Bayesian model and ours is their use of independent, proper, inverse gamma priors for $\sigma_1^2, \ldots, \sigma_{c(\omega)}^2$. Their shape and scale parameters are set at 10^{-2} to reflect little prior information. However, this results in a posterior distribution that is not scale invariant in the sense that was described above.

3. Stochastic search

Once we have constructed an objective function $\pi : \mathbb{P}_n \to \mathbb{R}^+$ that measures the goodness of partitions, we are left with a potentially difficult optimization problem. As explained in Section 1, B_n grows extremely rapidly with n. Therefore, unless n is very small, it is impossible to maximize π by brute force enumeration. An alternative approach might be to generate a completely random sample of partitions from $\mathbb{P}_n$ and to evaluate their π-values. Surprisingly, simulation from the uniform distribution on $\mathbb{P}_n$ is a non-trivial problem—see, for example, Pitman (1997). Moreover, this method is quite inefficient. For example, even if n is only 20, and 1 billion random partitions are generated, the probability that none of the top 1000 partitions are observed is about 0.98. Thus, it is clear that a more intelligent search algorithm is required.

Our task is a special case of the general problem of finding the maximum of an objective function over a large combinatorial set. Problems of this type are often amenable to Markov chain Monte Carlo optimization, which entails running a Markov chain on the combinatorial set of interest and evaluating the objective function at the successive states (see, for example, section 12.6 of Jerrum and Sinclair (1996)). Before introducing the Markov chains that we have developed for this problem, we digress briefly to discuss the issue of multimodality.

The problem of local modes is well known to plague the likelihood function of the mixture model (1). However, the mixture model likelihood is a function of the model parameters. In contrast, our objective function is over the discrete space of partitions. Thus, the multimodality issue is very different in the context of our model. Multimodality on the partition space would mean that there are two or more partitions with high posterior probabilities that are separated by

regions of low probability. This possibility seems implausible for models that are even remotely reasonable descriptions of the data. Moreover, our empirical findings are consistent with this intuition, i.e. we have found no evidence of multimodality in the specific versions of $\pi(\omega|y)$ with which we have worked. (As we explain in Section 5, we have 'hunted' for local modes by rerunning our search algorithm from multiple starting points in $\mathbb{P}_n$. So this is not a case of 'ignorance is bliss'.) In the next two subsections, we describe a Markov chain with state space $\mathbb{P}_n$ and stationary mass function proportional to π that is very simple to simulate and seems to be effective at honing in on the maximum of the objective function.

3.1. A Metropolis–Hastings algorithm based on a biased random walk

In general, the MH algorithm allows us to simulate a Markov chain with a prespecified stationary distribution by 'correcting' an easy-to-simulate candidate Markov chain. When an MH algorithm is used to solve combinatorial optimization problems, the candidate Markov chain is often taken to be a random walk on a graph that defines a neighbourhood structure for the combinatorial set in question. Let G_n be a connected, undirected graph with vertex set $\mathbb{P}_n$ such that there is an edge between two vertices ω_i and ω_j, if and only if it is possible to go from partition ω_i to partition ω_j by moving exactly one of the n objects in ω_i to a different cluster. The graph G_n determines a neighbourhood structure on $\mathbb{P}_n$, i.e. ω_i and ω_j are neighbours if and only if they share an edge. For example, when $n=3$, the five possible partitions are given in expression (5). The only two partitions that do not share an edge in G_3 are ω_1 and ω_5.

Let $d(\omega)$ denote the number of neighbours (or degree) of the vertex ω in G_n. An obvious candidate Markov chain for the MH algorithm is the nearest neighbour random walk on G_n which moves from ω' to ω with probability $1/d(\omega')$ if ω and ω' are neighbours and 0 otherwise. Our simple example with $n=3$ illustrates that different partitions may have different numbers of neighbours. Consequently, the transition matrix of this random walk is not symmetric. For example, $\mathrm{pr}(\omega_1 \to \omega_2) = \frac{1}{3}$, but $\mathrm{pr}(\omega_2 \to \omega_1) = \frac{1}{4}$.

Consider programming the MH algorithm with the nearest neighbour random walk as the candidate. Let the current state be ω'. To simulate the next state, we require a method of sampling uniformly at random from the $d(\omega')$ neighbours of ω'—call the selected neighbour ω—and an algorithm for calculating $d(\omega)$, so that the acceptance probability may be computed. This can become quite computationally intensive, as we must enumerate all the possible neighbours. However, it turns out that a slightly different candidate leads to an MH algorithm that is as effective and much simpler to programme.

The alternative candidate Markov chain evolves as follows. Let c denote the number of clusters in the current state (partition). There are two cases: $c=1$ and $c \geqslant 2$. If $c=1$, choose one of the n objects uniformly at random and move the chosen object to its own cluster. If $c \geqslant 2$, choose one of the n objects uniformly at random. If the chosen object is a singleton (i.e. forms its own cluster), then move it to one of the other $c-1$ clusters, each with probability $1/(c-1)$. If the chosen object is not a singleton, then move it to one of the other $c-1$ clusters, each with probability $1/c$, or make the chosen object its own cluster with probability $1/c$. As with the nearest neighbour random walk, the move $\omega' \to \omega$ has positive probability if and only if ω' and ω share an edge in G_n. We call this Markov chain the biased random walk on G_n.

At first glance, one might think that what we have just described is simply an algorithm for simulating the nearest neighbour random walk. This is not so, however. For example, under the new dynamics, in the $n=3$ example, $\mathrm{pr}(\omega_1 \to \omega_2) = \mathrm{pr}(\omega_2 \to \omega_1) = \frac{1}{3}$. In fact, straightforward arguments show that the transition matrix of the biased random walk is symmetric. Therefore, when this chain is used as the candidate in the MH algorithm, the acceptance

probability is simply $\min\{1, \pi(\omega)/\pi(\omega')\}$, i.e. running this algorithm does not require finding or counting the neighbours of ω and ω'. Hence, this alternative candidate results in an MH algorithm that is much easier to programme and faster in the sense of more iterations per unit time.

Crowley (1997) developed and employed a Markov chain on $\mathbb{P}_n$ that can be viewed as a deterministic scan Gibbs sampler. Our MH algorithm based on the biased random walk has the same basic structure as a random-scan version of Crowley's algorithm. In both cases, at each iteration, one of the n objects is chosen at random and the chosen object is either moved to a new cluster or left where it is. However, one iteration of the random-scan Gibbs sampler typically requires many more evaluations of the objective function than the biased random-walk algorithm.

3.2. Adding split–merge moves

The biased random walk yields proposals that are small or local in the sense that only one object at a time is moved from one cluster to another. Consequently, a large scale change that would result in a substantial increase in the objective function (e.g. merging two similar clusters) is very unlikely to occur because, taken together, the sequence of moves leading to this change is very unlikely.

We now describe an alternative candidate Markov chain that proposes more drastic changes to the current state and which addresses the limitation of the biased random walk that was described above. This candidate chain was used by Green (1995) in his analysis of Consonni and Veronese's (1995) model. At each iteration, we randomly decide between a merge move with probability $p_{\mathrm{m}} \in (0, 1)$ and a split move with probability $1 - p_{\mathrm{m}}$. A merge proposal is constructed by merging two randomly chosen clusters in the current partition. A split proposal is created by randomly choosing a cluster and then randomly splitting it into two clusters conditionally on neither being empty. A split move is automatically proposed whenever the current state consists of a single cluster, and likewise a merge move is automatically proposed when the current state consists of n clusters. Suppose that $\omega, \omega' \in \mathbb{P}_n$ and that it is possible to arrive at ω by merging two clusters in ω'. Let n^* denote the size of the cluster in ω that must be split to arrive at ω'. (n^* will necessarily be greater than or equal to 2.) Under these dynamics

$$\mathrm{pr}(\omega' \to \omega) = \frac{2p_{\mathrm{m}}}{c(\omega')\{c(\omega') - 1\}}$$

and

$$\mathrm{pr}(\omega \to \omega') = \frac{1 - p_{\mathrm{m}}}{(2^{n^*-1} - 1) \sum_{k=1}^{c(\omega)} I[\#\{\mathcal{C}_k(\omega)\} \geqslant 2]}.$$

To use this Markov chain as the candidate in an MH algorithm, we would accept a proposed move from $\omega' \to \omega$ with probability $\min(1, R)$ where

$$R = \frac{\pi(\omega)\,\mathrm{pr}(\omega \to \omega')}{\pi(\omega')\,\mathrm{pr}(\omega' \to \omega)},$$

and we would accept a proposed move from $\omega \to \omega'$ with probability $\min(1, 1/R)$.

It is a simple matter to add split–merge moves to the biased random-walk algorithm described above. For example, at each iteration we could make a transition according to the biased random-

walk MH algorithm or the split–merge algorithm with probabilities p_b and $1 - p_b$ respectively. Since π is invariant for each MH kernel, it is invariant for the mixture (see, for example, Besag *et al.* (1995)).

4. Simulation study

In this section we report results from a simulation study in which our method is compared with the K-means method (Hartigan and Wong, 1979), which is one of the most widely used clustering algorithms. The K-means algorithm is a deterministic greedy algorithm and, like most clustering algorithms, a limitation is that it requires a fixed, user-specified, number of clusters. At each iteration, the current partition is updated by moving a single object to a different cluster. The particular move that is made is the one that results in the largest decrease in the within-cluster variation. Its implementation in the R package (R Foundation for Statistical Computing, 2006) involves a random starting partition. Thus, the method does not return the same answer when it is applied multiple times to the same input data. We view this as a good feature, as it serves to emphasize a key point of this paper: that some form of stochastic search is unavoidable if the user is to have confidence in the output from a clustering algorithm.

We compared the performance of the K-means algorithm with our model-based stochastic search approach by using 10 data sets that were generated according to the following regime. Each data set consisted of $n = 200$ pairs of replicate profiles of dimension $p = 11$, and each contained 13 clusters of sizes 100, 20, 20, 10, 10, 10, 5, 5, 5, 5, 5, 3 and 2. Each cluster had a distinct mean profile, with the mean profile for the cluster of size 100 being identically 0. (This was motivated in part by applications in genetics in which most genes do not respond to a given treatment.) The remaining 12 mean profiles were functions of the following three types: $f_1(t) = a(t/T)^{\alpha-1}(1 - t/T)^{\beta-1}$, $f_2(t) = a - b\exp(-\alpha t)$ and $f_3(t) = a\sin(2\pi t/\beta)$ for $0 \leqslant t \leqslant T = 10$. Four different choices of parameters were selected for each type of function.

We added random noise to each profile at the observation, replicate and cluster level. Specifically, each replicate profile has the form (7) with the (normal) distributional assumptions for U_i and ε_{ij} that were described in Section 2. However, the components of V_k were generated as identically distributed but positively correlated normal variates to produce a smooth perturbation of the cluster mean (an exception was that V_k was identically 0 in the largest cluster). It is important to keep in mind that, even though the theoretical model generating the data had exactly 13 clusters, the added noise makes it unlikely that 13 is the 'right number of clusters' for a given data set. In fact, a cursory examination of the simulated data suggested an average of about nine distinguishable clusters.

We assessed performance on the basis of the number of pairs of objects in the 12 non-zero mean clusters that were correctly clustered together or apart (using the theoretical model as the gold standard). Specifically, consider the 2×2 table with counts $\{n_{ij}\}$ that is formed by cross-classifying all $\binom{200}{2}$ pairs of objects according to whether they belong together and whether they were clustered together. We used four statistics to measure performance:

(a) the χ^2-statistic,

$$\chi^2 = \frac{(n_{11} - n_{1+})^2}{n_{1+}} + \frac{(n_{22} - n_{2+})^2}{n_{2+}}, \tag{12}$$

which compares the observed diagonal counts with the ideal, $n_{11} = n_{1+}$ and $n_{22} = n_{2+}$;

(b) Yule's Q association measure (Agresti (1990), page 23),

$$\hat{Q} = \frac{n_{11}n_{22} - n_{12}n_{21}}{n_{11}n_{22} + n_{12}n_{21}}; \tag{13}$$

(c) the sensitivity n_{11}/n_{1+};
(d) the specificity n_{22}/n_{2+}.

Note that $\hat{Q} \in [-1, 1]$ and values near 1 correspond to better performance. Also, the maximum value of χ^2 equals $n_{1+} + n_{2+} = n_{++}$, when the main diagonal counts are both 0. We define $\psi = 1 - \chi^2/n_{++}$, so that values that are close to 1 correspond to better performance for all four measures.

The K-means algorithm was implemented on each of the 10 data sets with K taking values 5, 10 and 20. Each of the K-means partitions were then used as starting values for model-based stochastic searches. To emulate a situation in which there is no obvious parametric base model for the profiles, we used a quadratic penalized spline model that can be formulated as the best linear unbiased predictor from a linear mixed model fit (Ruppert *et al.*, 2003). In particular, we

Table 1. Simulation results comparing the K-means method with model-based stochastic search ('Splines')†

Method	*K*	*Yule's Q*	ψ	*Sensitivity*	*Specificity*
K-means	5	0.952	0.945	0.867	0.769
		(0.025)	(0.013)	(0.045)	(0.031)
	10	0.465	0.899	0.409	0.956
		(0.057)	(0.009)	(0.030)	(0.003)
	20	−0.173	0.815	0.183	0.989
		(0.026)	(0.003)	(0.007)	(0.001)
Splines,	12.60	0.994	0.991	0.930	0.913
$\log(m) = 0$	(0.371)	(0.002)	(0.002)	(0.014)	(0.015)
	13.4	0.965	0.989	0.858	0.945
	(0.542)	(0.016)	(0.004)	(0.037)	(0.006)
	14.50	0.946	0.985	0.793	0.962
	(0.582)	(0.013)	(0.003)	(0.031)	(0.004)
Splines,	7.30	0.999	0.976	0.971	0.838
$\log(m) = -10$	(0.260)	(0.001)	(0.007)	(0.010)	(0.028)
	8.60	0.992	0.994	0.928	0.930
	(0.163)	(0.003)	(0.001)	(0.018)	(0.006)
	9.30	0.967	0.989	0.846	0.953
	(0.260)	(0.011)	(0.003)	(0.031)	(0.004)
Splines,	6.10	0.999	0.960	0.979	0.784
$\log(m) = -20$	(0.180)	(0.000)	(0.009)	(0.007)	(0.029)
	8.00	0.992	0.991	0.931	0.916
	(0.211)	(0.005)	(0.003)	(0.020)	(0.016)
	8.30	0.969	0.990	0.863	0.945
	(0.260)	(0.013)	(0.003)	(0.034)	(0.006)
Splines,	5.80	0.998	0.956	0.973	0.765
$\log(m) = -30$	(0.133)	(0.001)	(0.008)	(0.011)	(0.023)
	7.30	0.997	0.991	0.964	0.902
	(0.153)	(0.002)	(0.002)	(0.012)	(0.012)
	7.60	0.995	0.991	0.936	0.909
	(0.267)	(0.002)	(0.004)	(0.012)	(0.017)

†Values given are averages over 10 data sets with standard errors given below in parentheses. The stochastic searches were started at the K-means solutions for $K = 5, 10, 20$. The average number of clusters in the final partitions are given in the column that is headed by K. We note that the values of $E\{c(\omega)\}$ corresponding to $\log(m) = 0, -10, -20, -30$ are 5.878 030 94, 1.000 266 63, 1.000 000 01 and 1.000 000 00 respectively.

used three knots, $\tau_1 = 2.5$, $\tau_2 = 5$ and $\tau_3 = 7.5$, so that the X-matrix consists of three columns ($q = 3$): a column of 1s, 11 equally spaced time points, $t_i = i$, for $i = 0, 1, \ldots, 10$, and a column of the squared time points. The matrix Z_1 is an identity matrix ($s_1 = 11$) and Z_2 is an 11×3 matrix with (i, j)th entry given by $(t_i - \tau_j)_+^2$ (so that $s_2 = 3$). The parameters λ_1 and λ_2 were set equal to analysis-of-variance estimates based on the first K-means partition and a 'default' model in which the cluster mean profiles are assumed to be constant (i.e. $X = 1_p$), and the variance parameters are assumed homogeneous across clusters. (This is another difference between our implementation and that of Heard *et al.* (2006), section 5.2.) They took an empirical Bayes approach in which the maximization is accomplished by running their algorithm for a range of values of λ_2.) Each stochastic search was run for 10^5 iterations with the parameters p_b and p_m from Section 3.2 set equal to 0.9 and 0.5 respectively.

Average values of the four performance measures, and their standard errors, are reported in Table 1 for the K-means partitions, and those obtained by our model-driven stochastic search. Also given are the average number of clusters in the final partitions, at each combination of initial number of clusters and the prior parameter m. This simulation shares some general features with others that we have performed. First, the final number of clusters, $c(\omega)$, tends to increase with the initial number. This is an indication that the algorithm has not fully converged. However, the numbers are not drastically different. For example, with $\log(m) = 0$, $c(\omega)$ ranged from 12.6 when the initial number was 5, to 14.5 when the initial number was 20. Second, the final number of clusters tends to decrease as m decreases owing to the penalty term, $c(\omega)\log(m)$, in the log-posterior. Third, setting $\log(m) = 0$ leads to overestimation of the number of distinguishable clusters (about 9 in this simulation), but there is a large range of values of m (which is consistent with $E\{c(\omega)\} \approx 1$) that yield similar reasonable results. Fourth, specificity generally decreases with $c(\omega)$, whereas sensitivity increases. This is to be expected, since, the larger $c(\omega)$, the less likely a pair of objects from different theoretical clusters is to be placed in the same empirical cluster.

It is interesting that the K-means algorithm performs best when $K = 5$. When $K = 10$ (which is close to the correct value), the performance measures deteriorate dramatically, with the exception of sensitivity. Not surprisingly, the best overall performances of the model-based stochastic search occur at settings which result in $c(\omega)$ close to 9. However, the model-based approach is almost uniformly better than the K-means method, for the range of initial $c(\omega)$-values and prior parameter values that we considered.

5. Examples

5.1. Application to yeast cell cycle data

To illustrate the clustering method that is proposed in this paper we consider the expression profiles of yeast genes from the α-factor synchronization experiment discussed by Spellman *et al.* (1998). The data consist of log-expression ratios of 104 genes, which are known to be cell cycle regulated. The measurements are taken from 18 complementary DNA microarrays equally spaced at intervals of 7 min. About 80% of the profiles are complete, and all except one had four or fewer missing values. The one gene with more than four missing values has been omitted in the subsequent analysis. The data are available on line from Stanford University's 'Yeast cell cycle analysis project' Web site at `http://genome-www.stanford.edu/cellcycle/data/rawdata/`.

The primary goal of cluster analysis in this context is to find groups of genes that are all part of a team performing some function, i.e. groups of co-regulated genes. An example of such a group is provided in Fig. 1, which shows the profiles of a subset of eight histones that are

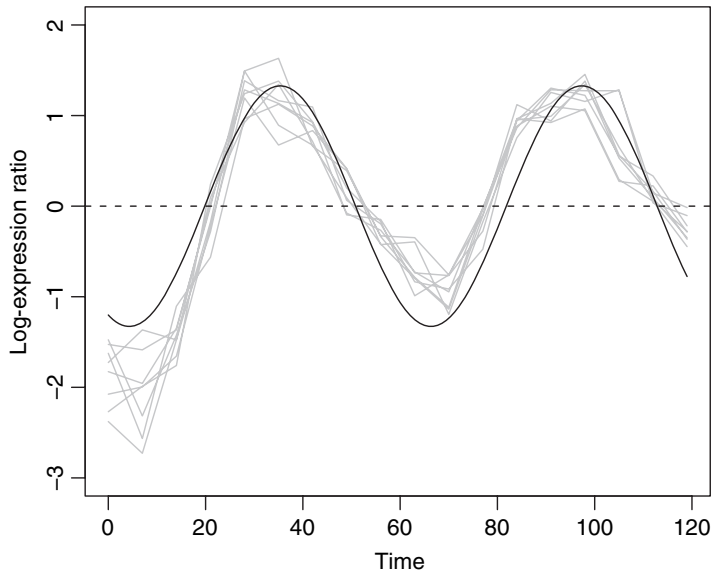

Fig. 1. Gene expression profiles for the eight histones: ———, first-order Fourier series model (14) fit to the pointwise average profile

known to show peak expression during the short synthesis phase. A naive way to cluster these data is to ignore the time aspect and simply to apply a standard clustering algorithm to the 103 18-dimensional vectors. However, implicit in the analysis of Spellman *et al.* (1998) is the first-order Fourier series model

$$E\{y(t)\} = \tfrac{1}{2}a_0 + a_1\cos(2\pi t/T) + b_1\sin(2\pi t/T), \tag{14}$$

where $y(t)$ denotes log-expression ratio at time t, and T is the period of the cell cycle. Although it is tempting to make use of this model to represent the mean structure parsimoniously, one must recognize that strict adherence to such a model could cause problems. For example, the least squares fit of model (14) to the eight histone profiles is overlayed in Fig. 1. The model is reasonably effective in identifying the phase of peak expression of each gene, but there is clearly a substantial lack of fit to these gene profiles. One of the key features of the mixed model methodology that we propose is the allowance for parsimonious deviation from an overly simplistic base model.

We used the first-order Fourier series model (14) to fill in the missing values and to register the profiles. More specifically, if T is known, then the model is linear in the intercept and slope parameters, a_0 and (a_1, b_1). In our analysis we fixed T at 62, which is the least squares estimate of T that was obtained by Booth *et al.* (2004) in a previous analysis of these same data. We then estimated the regression parameters for each gene separately via least squares and used the resulting fitted models to fill in the missing data. The possibility of incorporating the imputation of missing values into the Bayesian analysis is discussed in the next section. However, having balanced data greatly simplifies the computations in the cluster analysis. Indeed, when the data are balanced, the estimated regression coefficients for a given cluster are simply averages of the least squares estimates for the genes in the cluster. As missing values comprised less than 2% of the data, this substitution has little effect on our conclusions. Finally, to register the profiles at the same overall level, we further modified the data by subtracting the estimated intercept from

each profile. This is similar to the mean subtraction that was used in Spellman *et al.* (1998).

In our cluster analysis, we used the linear mixed model (7) with $Z_1 = 0$ (no replicates), $Z_2 = I$ and an X-matrix based on model (14), i.e. X is 18×2 with the row corresponding to time t equal to $(\cos(2\pi t/62), \sin(2\pi t/62))$. The intercept term is absent because of the registration. To obtain a value for λ_2, we first applied the K-means clustering algorithm to the data with the number of clusters fixed at 5, corresponding to the number of phases of the cell cycle, and then obtained the analysis-of-variance estimate $\hat{\lambda}_2 = 1.45$ on the basis of the K-means partition, and the default model that was described in Section 4. (A method for incorporating the tuning parameters in the Bayesian analysis is discussed in the next section.) A second analysis-of-variance estimate, $\hat{\lambda}_2 = 2.39$, was obtained by using a K-means partition with 10 clusters.

Using the two K-means solutions as starting partitions, along with their associated $\hat{\lambda}_2$-values, we searched for the maximizer of the objective function by running 10^5 iterations of the MH algorithm with $p_b = 0.9$ and $p_m = 0.5$. We used $\log(m) = -20$, which corresponds to an *a priori* expected number of clusters that is 1 to seven decimal places. To demonstrate the insensitivity of our stochastic search algorithm to the starting value, we performed four additional runs of the algorithm with the same two λ_2-values starting with all genes in a single cluster as well as all genes in separate clusters. The best partitions found were remarkably similar regardless of the starting partition and the value of λ_2. In particular, the $\hat{Q}$ association measure (13) was greater than 0.99 in all pairwise comparisons between the six runs. Moreover, the final number of clusters in the six runs ranged only from 9 to 11. One of the model-based solutions with nine clusters is shown in Fig. 2, with the eight histones captured in cluster 1. This cluster was perfectly identified by all six model-based solutions.

In comparison, the association between the two K-means solutions was $\hat{Q} = 0.92$, and the pairwise association between these and the model-based solutions ranged from 0.86 to 0.97. In addition, the histone cluster was not perfectly identified in either K-means partition. A key point is that, for the K-means procedure to be effective, some form of stochastic search must be incorporated, with good partitions being identified presumably by its least squares criterion. Similar comments can be made about other clustering algorithms that converge to different solutions depending on the starting values.

5.2. Corneal wound healing

Our next example concerns 646 gene expression profiles that were obtained from Affymetrix gene chip microarrays at days 0, 1, 2, 3, 4, 5, 6, 7, 14, 21, 42 and 98 of a study of corneal wound healing in rats at the University of Florida. There were two technical replicate measurements at each time point. The day 0 sample was taken before photorefractive keratectomy (corrective eye surgery) and hence represents a baseline value to which the profiles are expected to return over the treatment period. Unlike the yeast cell cycle example, here there is no obvious parametric base model for the profiles. As in the simulation, we used a quadratic penalized spline model. Specifically, consider the equally spaced and centred timescale, $t_j = j - 5.5$, $j = 0, 1, \ldots, 11$. Then, the X-matrix in model (7) is 12×3 with jth row given by $(1, t_j, t_j^2)$, and Z_1 and Z_2 are both 12×5, with the entries in column $i = 1, \ldots, 5$ equal to $(t_j - \tau_i)_+^2$, $j = 0, \ldots, 11$, where $\tau_i = 2(i-3)$.

As with the cell cycle data, an initial partition of the data was obtained by applying the K-means procedure. In this case we set the number of clusters equal to 20. Analysis-of-variance fitting of the default model based on the K-means partition resulted in the estimates $(\hat{\lambda}_1, \hat{\lambda}_2)$ equal to (0.0,1.31). The best partition found after running the stochastic search algorithm for 10^5 iterations, with $\log(m) = -30$, and using the K-means solutions to initiate the Markov chain, consisted of 16 clusters. These are show in Fig. 3. Also shown are the (unnormalized)

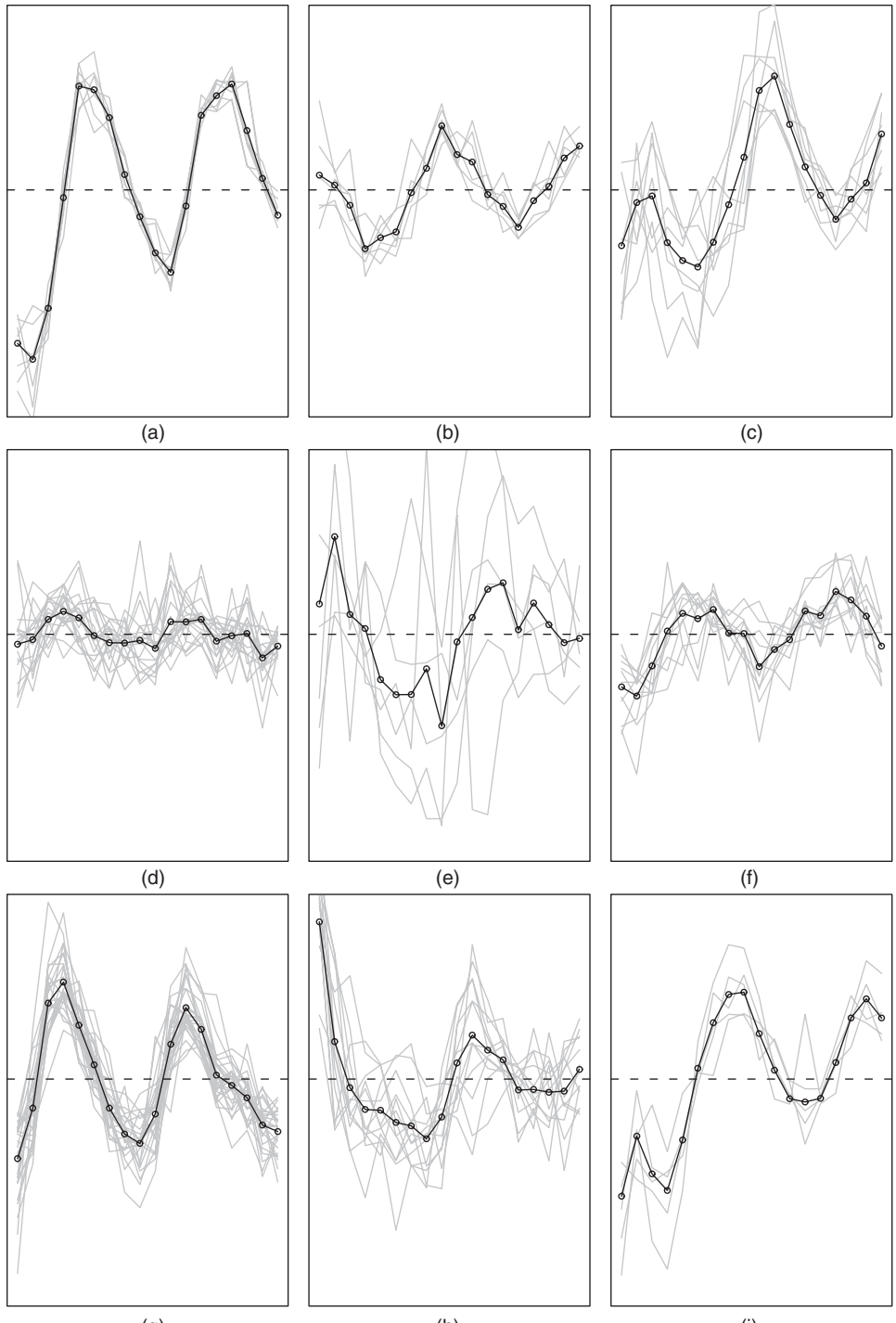

Fig. 2. Clusters of yeast cell cycle gene profiles (———, best linear unbiased predictors calculated by using equation (11)): (a) cluster 1 (eight histones); (b) cluster 2; (c) cluster 3; (d) cluster 4; (e) cluster 5; (f) cluster 6; (g) cluster 7; (h) cluster 8; (i) cluster 9

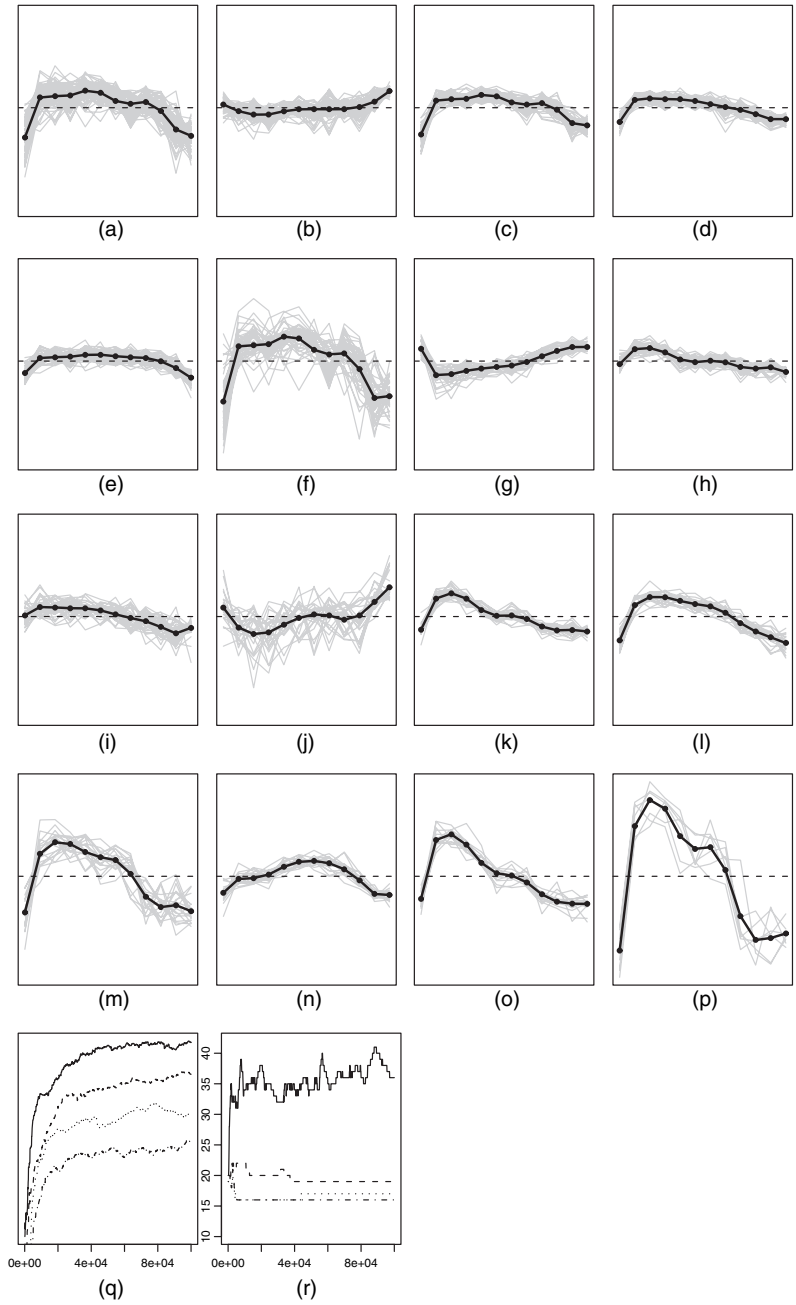

Fig. 3. 16 clusters from the corneal wound healing experiment arranged according to cluster size (———, averages over the two replicates for each gene; ———, best linear unbiased predictors at the 12 time points 0, 1, 2, 3, 4, 5, 6, 7, 14, 21, 42 and 98 days; the timescale is transformed so that the points are equally spaced): (a) cluster 1, 87 genes; (b) cluster 2, 79 genes; (c) cluster 3, 66 genes; (d) cluster 4, 59 genes; (e) cluster 5, 57 genes; (f) cluster 6, 48 genes; (g) cluster 7, 40 genes; (h) cluster 8, 40 genes; (i) cluster 9, 39 genes; (j) cluster 10, 26 genes; (k) cluster 11, 24 genes; (l) cluster 12, 23 genes; (m) cluster 13, 20 genes; (n) cluster 14, 18 genes; (o) cluster 15, 13 genes; (p) cluster 16, 7 genes; (q), (r) (unnormalized) log-posteriors and numbers of clusters as a function of iteration number for four additional runs of the algorithm with $\log(m) = 0$ (———), $\log(m) = -10$ (-------), $\log(m) = -20$ (·····) and $\log(m) = -30$ (·-·-·-)

log-posteriors and the number of clusters as a function of iteration number for four additional runs of the algorithm with $\log(m)$ equal to 0, -10, -20 and -30. (These values of $\log(m)$ correspond to $E\{c(\omega)\}$ values of 7.048 788 96, 1.000 319 94, 1.000 000 02 and 1.000 000 00 respectively.) The plot of the log-posteriors serves as a crude graphical convergence diagnostic. (The reader should keep in mind that the posteriors are not normalized so the curves are not directly comparable.) The plot showing the progression in the numbers of clusters illustrates the effect of the prior parameter m. The algorithm tends to overestimate the number of clusters when $\log(m) = 0$ but is relatively stable for values in the range from $\log(m) = -10$ to $\log(m) = -30$. These findings are consistent with those in the simulation study and in other examples that we have considered.

Finally, we provide another demonstration that the algorithm is fairly robust to the starting value. We performed four additional runs of the algorithm with the same $(\hat{\lambda}_1, \hat{\lambda}_2)$. The first was started with all genes in the same cluster, the second with every gene in its own cluster and the third and fouth runs were started at two other K-means solutions (based on 20 clusters). The final partitions were all quite similar to the partition that was reported above. Indeed, the $\hat{Q}$ association measure was greater than 0.84 in all pairwise comparisons among these five runs. Moreover, this number jumps up to 0.89 if we remove the first run from consideration. This is quite consistent with our empirical experience which suggests that it takes the algorithm longer to converge when the starting partition has too few clusters.

6. Discussion

In our implementation the 'tuning' parameter, $\lambda = (\lambda_1, \lambda_2)$, was fixed throughout the stochastic search. We chose its value by fitting a default linear mixed model to an initial partition that was obtained by using the K-means procedure. An alternative approach is to incorporate λ into the Bayesian analysis by specifying a prior distribution. However, no choice of prior leads to a tractable form for $\pi(\omega|y)$. One possibility for exploring the joint posterior of λ and ω, $\pi(\lambda, \omega|y)$, is to use an MH algorithm that updates ω (as in Section 3) with probability p and λ with probability $1 - p$. We have successfully implemented this approach for λ_2 by using a Student t random-walk candidate. To be specific, given the current value λ_2', the candidate is $\lambda_2 = \kappa T + \lambda_2'$ where T is a standard Student t-variate and κ is chosen to match the fit of a Gaussian distribution to the posterior as a function of λ_2 at the initial partition. (The posterior $\pi(\lambda, \omega|y)$ is integrable with respect to λ_2 if $Z_2 = I$.) A major disadvantage of this approach is that, instead of yielding evaluations of the marginal posterior, $\pi(\omega|y)$, it yields evaluations of $\pi(\omega|\lambda, y)$ and approximate samples from $\pi(\omega|y)$. It is unclear how these can be used to approximate effectively the maximizer of $\pi(\omega|y)$.

Unfortunately, a similar issue arises when we attempt to deal with missing values in the multivariate profiles in a formal way. In particular, integrating the missing data out of the likelihood leads to an intractable form for the marginal posterior distribution on the space of partitions. In other words, $\pi(\omega|y_{\text{obs}})$ is intractable, where y_{obs} denotes the observed data. An exception is the special case $X = Z_1 = 0$, which was considered in Heard *et al.* (2006). However, missing data complicate the computations even in this setting. In the general case, the data augmentation algorithm (which is also known as the two-variable Gibbs sampler) could presumably be used to simulate a Markov chain whose stationary distribution is $\pi(\omega, y_{\text{miss}}|y_{\text{obs}})$, where y_{miss} denotes the missing data, but the problem that was described above is apparent here as well, i.e. this method would not provide us with evaluations of $\pi(\omega|y_{\text{obs}})$.

In conclusion, we have proposed a multilevel mixed model for clustering multivariate data. Our model leads to a tractable, probability-based, objective function for identifying good partitions. One key difference between the approach proposed, and most conventional clustering

algorithms, is that it is not necessary to specify the number of clusters in advance. A second difference is that measurements on different objects, within the same cluster, are correlated because they share cluster-specific random effects. The inclusion of such random effects allows for parsimonious deviation of the mean profile for a given cluster from a given base model, that may be captured statistically via the best linear unbiased predictor. We also allow for a second level of dependence when replicate observations are obtained on each object, which is a situation that is quite common in microarray experiments. This second type of dependence can also be incorporated in the mixture model framework (1) by letting $f(\cdot;\theta_k)$ be the density of the entire observation vector for an object from cluster k. For example, both McLachlan *et al.* (2004) and Celeux *et al.* (2005) proposed mixed model formulations of the density f in which dependence between replicate observations is induced, as it is in our model, through the inclusion of object-specific random effects. However, in contrast with the mixed models formulation that is proposed in this paper, these models still assume that the observation vectors from different objects in the same cluster are independent and identically distributed. An approach to modelling dependence between the observation vectors, within the mixture model framework, is to suppose that there is a hidden Markov chain on the component indicator vectors. McLachlan and Peel (2000), chapter 13, pointed out that a hidden Markov model might be realistic, 'when the observations appear sequentially in time and tend to cluster or to alternate between different possible components'. However, it is difficult to justify this approach when the ordering of the observations is arbitrary.

Acknowledgements

The authors are grateful to Dr Henry Baker, College of Medicine, University of Florida, for providing the wound healing data set, and to Peter McCullagh and three reviewers for comments and suggestions that led to a much improved version of the paper. This research was partially supported by National Science Foundation grants DMS-04-05543 (Booth and Casella) and DMS-05-03648 (Hobert).

Appendix A: Expected number of clusters under Crowley's prior

Assuming that ω is a random partition with mass function (4), we have

$$\mathrm{pr}\{c(\omega)=k\}=\sum_{\omega:c(\omega)=k}\pi_n(\omega)=\frac{\Gamma(m)m^k}{\Gamma(n+m)}\sum_{\omega:c(\omega)=k}\prod_{j=1}^{k}\Gamma(n_j).$$

Hence,

$$\begin{aligned}
E\{c(\omega)\}=\sum_{k=1}^{n}k\,\mathrm{pr}\{c(\omega)=k\}&=\frac{\Gamma(m)}{\Gamma(n+m)}\sum_{k=1}^{n}km^k\sum_{\omega:c(\omega)=k}\prod_{j=1}^{k}\Gamma(n_j)\\
&=\frac{m\,\Gamma(m)}{\Gamma(n+m)}\sum_{k=1}^{n}km^{k-1}\sum_{\omega:c(\omega)=k}\prod_{j=1}^{k}\Gamma(n_j)\\
&=\frac{m\,\Gamma(m)}{\Gamma(n+m)}\frac{\partial}{\partial m}\left\{\sum_{k=1}^{n}\sum_{\omega:c(\omega)=k}m^k\prod_{j=1}^{k}\Gamma(n_j)\right\}\\
&=\frac{m\,\Gamma(m)}{\Gamma(n+m)}\frac{\partial}{\partial m}\left\{\frac{\Gamma(n+m)}{\Gamma(m)}\right\}\\
&=m\{\psi(n+m)-\psi(m)\}=m\sum_{i=0}^{n-1}\frac{1}{m+i}
\end{aligned}$$

where $\psi(\cdot)$ is the derivative of the log-gamma function.

Appendix B: Maximizing the likelihood

For a fixed partition, the maximizer of equation (3) with respect to β_k is given by

$$\hat{\beta}_k = \left\{(1_{n_k r} \otimes X)^{\mathrm{T}} M_k^{-1} (1_{n_k r} \otimes X)\right\}^{-1} (1_{n_k r} \otimes X)^{\mathrm{T}} M_k^{-1} Y_k^*.$$

To show that $\hat{\beta}_k = (X^{\mathrm{T}} W_k X)^{-1} X^{\mathrm{T}} W_k \bar{Y}_k$, we shall establish the following two facts:

(a) $(1_{n_k} \otimes 1_r \otimes X)^{\mathrm{T}} M_k^{-1} (1_{n_k} \otimes 1_r \otimes X) = n_k r X^{\mathrm{T}} W_k X$;
(b) $(1_{n_k} \otimes 1_r \otimes X)^{\mathrm{T}} M_k^{-1} Y_k^* = n_k r W_k \bar{Y}_k$.

A key matrix inversion result that will be used is

$$(I_{ml} + J_m \otimes C)^{-1} = I_{ml} - J_m \otimes (I_l + mC)^{-1} C. \tag{15}$$

Using result (15), we obtain

$$\begin{aligned} M_k^{-1} &= \{I_{n_k rp} - J_{n_k} \otimes (I_{rp} + n_k A^{-1} B)^{-1} A^{-1} B\}(I_{n_k} \otimes A^{-1}) \\ &= I_{n_k} \otimes A^{-1} - J_{n_k} \otimes (I_{rp} + n_k A^{-1} B)^{-1} A^{-1} B A^{-1}. \end{aligned} \tag{16}$$

Now,

$$\begin{aligned} n_k A^{-1} B &= J_r \otimes \{I_p - (I_p + r\lambda_1 Z_1 Z_1^{\mathrm{T}})^{-1} r\lambda_1 Z_1 Z_1^{\mathrm{T}}\} n_k \lambda_2 Z_2 Z_2^{\mathrm{T}} \\ &= J_r \otimes (I_p + r\lambda_1 Z_1 Z_1^{\mathrm{T}})^{-1} n_k \lambda_2 Z_2 Z_2^{\mathrm{T}} = J_r \otimes n_k D, \end{aligned}$$

where $D = (I_p + r\lambda_1 Z_1 Z_1^{\mathrm{T}})^{-1} \lambda_2 Z_2 Z_2^{\mathrm{T}}$. It follows that

$$\begin{aligned} (I_{rp} + n_k A^{-1} B)^{-1} A^{-1} &= (I_{rp} + J_r \otimes n_k D)^{-1} \{I_{rp} - J_r \otimes (I_p + r\lambda_1 Z_1 Z_1^{\mathrm{T}})^{-1} \lambda_1 Z_1 Z_1^{\mathrm{T}}\} \\ &= I_{rp} - J_r \otimes (I_p + rn_k D)^{-1} n_k D - J_r \otimes (I_p + r\lambda_1 Z_1 Z_1^{\mathrm{T}})^{-1} \lambda_1 Z_1 Z_1^{\mathrm{T}} \\ &\quad - J_r \otimes (I_p + rn_k D)^{-1} n_k D (I_p + r\lambda_1 Z_1 Z_1^{\mathrm{T}})^{-1} r\lambda_1 Z_1 Z_1^{\mathrm{T}} \\ &= I_{rp} - J_r \otimes (I_p + r\lambda_1 Z_1 Z_1^{\mathrm{T}})^{-1} \lambda_1 Z_1 Z_1^{\mathrm{T}} - J_r \otimes (I_p + rn_k D)^{-1} n_k D (I_p + r\lambda_1 Z_1 Z_1^{\mathrm{T}})^{-1}. \end{aligned}$$

This combined with equation (16) yields

$$\begin{aligned} M_k^{-1} (1_{n_k} \otimes 1_r \otimes X) &= 1_{n_k} \otimes \{A^{-1} - (I_{rp} + n_k A^{-1} B)^{-1} n_k A^{-1} B A^{-1}\} (1_r \otimes X) \\ &= 1_{n_k} \otimes (I_{rp} + n_k A^{-1} B)^{-1} A^{-1} (1_r \otimes X) \\ &= 1_{n_k r} \otimes W_k X, \end{aligned} \tag{17}$$

where we have used the fact that

$$\begin{aligned} W_k &= (I_p + r\lambda_1 Z_1 Z_1^{\mathrm{T}} + n_k r\lambda_2 Z_2 Z_2^{\mathrm{T}})^{-1} \\ &= \{I_p + (I_p + r\lambda_1 Z_1 Z_1^{\mathrm{T}})^{-1} n_k r\lambda_2 Z_2 Z_2^{\mathrm{T}}\}^{-1} (I_p + r\lambda_1 Z_1 Z_1^{\mathrm{T}})^{-1} \\ &= (I_p + rn_k D)^{-1} (I_p + r\lambda_1 Z_1 Z_1^{\mathrm{T}})^{-1} \\ &= I_p - (I_p + r\lambda_1 Z_1 Z_1^{\mathrm{T}})^{-1} r\lambda_1 Z_1 Z_1^{\mathrm{T}} - (I_p + rn_k D)^{-1} rn_k D (I_p + r\lambda_1 Z_1 Z_1^{\mathrm{T}})^{-1}. \end{aligned}$$

The two facts now follow directly from equation (17). Thus, we may now write

$$(Y_k^* - (1_{n_k r} \otimes X)\beta_k)^{\mathrm{T}} M_k^{-1} (Y_k^* - (1_{n_k r} \otimes X)\beta_k) = n_k r \{(\beta_k - \hat{\beta}_k)^{\mathrm{T}} X^{\mathrm{T}} W_k X (\beta_k - \hat{\beta}_k) + p\hat{\sigma}_k^2\}, \tag{18}$$

where

$$\hat{\sigma}_k^2 = \frac{1}{n_k rp} (Y_k^* - (1_{n_k r} \otimes X)\hat{\beta}_k)^{\mathrm{T}} M_k^{-1} (Y_k^* - (1_{n_k r} \otimes X)\hat{\beta}_k).$$

We conclude by establishing statistic (8). By adding and subtracting the term $1_{n_k r} \otimes \bar{Y}_k$ and multiplying, we obtain

$$\begin{aligned} n_k rp\hat{\sigma}_k^2 &= (Y_k^* - 1_{n_k r} \otimes \bar{Y}_k)^{\mathrm{T}} M_k^{-1} (Y_k^* - 1_{n_k r} \otimes \bar{Y}_k) + (1_{n_k r} \otimes (\bar{Y}_k - X\hat{\beta}_k))^{\mathrm{T}} M_k^{-1} (1_{n_k r} \otimes (\bar{Y}_k - X\hat{\beta}_k)) \\ &\quad + 2(Y_k^* - 1_{n_k r} \otimes \bar{Y}_k)^{\mathrm{T}} M_k^{-1} (1_{n_k r} \otimes (\bar{Y}_k - X\hat{\beta}_k)). \end{aligned}$$

Arguments that are similar to those above show that the cross-term (third term) is 0 and that the second term can be written as $n_k r(\bar{Y}_k - X\hat{\beta}_k)^{\mathrm{T}} W_k (\bar{Y}_k - X\hat{\beta}_k)$. Finally, the first term is

$$(Y_k^* - 1_{n_k r} \otimes \bar{Y}_k)^{\mathrm{T}} (I_{n_k} \otimes A^{-1} + J_{n_k} \otimes (I_{rp} + n_k A^{-1} B)^{-1} A^{-1} B A^{-1})(Y_k^* - 1_{n_k r} \otimes \bar{Y}_k).$$

It is easy to show that $BA^{-1} = J_r \otimes D^{\mathrm{T}}$ and it follows from what was done above that

$$(I_{rp} + n_k A^{-1} B)^{-1} A^{-1} = I_{rp} - J_r \otimes H_k,$$

where

$$H_k = (I_p + r\lambda_1 Z_1 Z_1^{\mathrm{T}})^{-1} \lambda_1 Z_1 Z_1^{\mathrm{T}} + (I_p + r n_k D)^{-1} n_k D (I_p + r\lambda_1 Z_1 Z_1^{\mathrm{T}})^{-1}.$$

Hence,

$$(I_{rp} + n_k A^{-1} B)^{-1} A^{-1} B A^{-1} = J_r \otimes (I_p - r H_k) D^{\mathrm{T}} = J_r \otimes G_k,$$

say. It follows that

$$\begin{aligned}(Y_k^* - 1_{n_k r} \otimes \bar{Y}_k)^{\mathrm{T}} (J_{n_k} \otimes (I_{rp} + n_k A^{-1} B)^{-1} A^{-1} B A^{-1})(Y_k^* - 1_{n_k r} \otimes \bar{Y}_k) \\ = (Y_k^* - 1_{n_k r} \otimes \bar{Y}_k)^{\mathrm{T}} (J_{n_k r} \otimes G_k)(Y_k^* - 1_{n_k r} \otimes \bar{Y}_k) = 0,\end{aligned}$$

and so

$$\begin{aligned}(Y_k^* - 1_{n_k r} \otimes \bar{Y}_k)^{\mathrm{T}} M_k^{-1} (Y_k^* - 1_{n_k r} \otimes \bar{Y}_k) &= (Y_k^* - 1_{n_k r} \otimes \bar{Y}_k)^{\mathrm{T}} (I_{n_k} \otimes A^{-1})(Y_k^* - 1_{n_k r} \otimes \bar{Y}_k) \\ &= \sum_{i \in C_k} (Y_i - 1_r \otimes \bar{Y}_k)^{\mathrm{T}} A^{-1} (Y_i - 1_r \otimes \bar{Y}_k).\end{aligned}$$

References

Agresti, A. (1990) *Categorical Data Analysis*. New York: Wiley.

Banfield, J. D. and Raftery, A. E. (1993) Model-based Gaussian and non-Gaussian clustering. *Biometrics*, **49**, 803–821.

Besag, J., Green, P., Higdon, D. and Mengersen, K. (1995) Bayesian computation and stochastic systems (with discussion). *Statist. Sci.*, **10**, 3–66.

Binder, D. A. (1978) Bayesian cluster analysis. *Biometrika*, **65**, 31–38.

Booth, J. G., Casella, G., Cooke, J. E. K. and Davis, J. M. (2004) Clustering periodically-expressed genes using microarray data: a statistical analysis of the yeast cell cycle data. *Technical Report*. Department of Biological Statistics and Computational Biology, Cornell University, Ithaca.

Celeux, G. and Govaert, G. (1992) A classification EM algorithm for clustering and two stochastic versions. *Computnl Statist. Data Anal.*, **14**, 315–332.

Celeux, G., Lavergne, C. and Martin, O. (2005) Mixture of linear mixed models: application to repeated data clustering. *Statist. Modllng*, **5**, 243–267.

Consonni, G. and Veronese, P. (1995) A Bayesian method for combining results from several binomial experiments. *J. Am. Statist. Ass.*, **90**, 935–944.

Crowley, E. M. (1997) Product partition models for normal means. *J. Am. Statist. Ass.*, **92**, 192–198.

Fraley, C. and Raftery, A. E. (2002) Model-based clustering, discriminant analysis, and density estimation. *J. Am. Statist. Ass.*, **97**, 611–631.

Green, P. J. (1995) Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. *Biometrika*, **82**, 711–732.

Hartigan, J. A. (1990) Partition models. *Communs Statist. Theory Meth.*, **19**, 2745–2756.

Hartigan, J. A. and Wong, M. A. (1979) Algorithm AS 136: A *K*-means clustering algorithm. *Appl. Statist.*, **28**, 100–108.

Heard, N. A., Holmes, C. C. and Stephens, D. A. (2006) A quantitative study of gene regulation involved in the immune response of Anopheline mosquitoes: an application of Bayesian hierarchical clustering of curves. *J. Am. Statist. Ass.*, **101**, 18–29.

Hitchcock, D. B., Casella, G. and Booth, J. (2006) Improved estimation of dissimilarities by smoothing functional data. *J. Am. Statist. Ass.*, **101**, 211–222.

Jerrum, M. and Sinclair, A. (1996) The Markov chain Monte Carlo method: an approach to approximate counting and integration. In *Approximation Algorithms for NP-hard Problems*. Boston: PWS.

McCullagh, P. and Yang, J. (2006) Stochastic classification models. *Technical Report*. Department of Statistics, University of Chicago, Chicago.

McLachlan, G. J. and Basford, K. E. (1998) *Mixture Models: Inference and Applications to Clustering*. New York: Dekker.
McLachlan, G. J., Do, K.-A. and Ambroise, C. (2004) *Analyzing Microarray Gene Expression Data*. New York: Wiley.
McLachlan, G. J. and Peel, D. (2000) *Finite Mixture Models*. New York: Wiley.
Pitman, J. (1997) Some probabilistic aspects of set partitions. *Am. Math. Mnthly*, **104**, 201–209.
Pitman, J. (2005) Combinatorial stochastic processes. *Lect. Notes Math.*, **1875**.
R Foundation for Statistical Computing (2006) *R: a Language and Environment for Statistical Computing*. Vienna: R Foundation for Statistical Computing. (Available from `http://www.R-project.org`.)
Ruppert, D., Wand, M. P. and Carroll, R. J. (2003) *Semiparametric Regression*. New York: Cambridge University Press.
Scott, A. J. and Symons, M. J. (1971) Clustering methods based on likelihood ratio criteria. *Biometrics*, **27**, 387–397.
Selim, S. Z. and Alsutan, K. (1991) A simulated annealing algorithm for the clustering problem. *J. Pattn Recogn. Soc.*, **24**, 1003–1008.
Serban, N. and Wasserman, L. (2005) CATS: clustering after transformation and smoothing. *J. Am. Statist. Ass.*, **100**, 990–999.
Spellman, P., Sherlock, G., Zhang, M. Q., Iyer, R. I., Anders, K., Eisen, M. B., Brown, P. O., Botstein, D. and Futcher, B. (1998) Comprehensive identification of cell cycle-regulated genes of the yeast *saccharomyces cerevisiae* by microarray hybridization. *Molec. Biol. Cell*, **9**, 3273–3297.
Stanley, R. P. (1997) *Enumerate Combinatorics*, vol. I. New York: Cambridge University Press.
Symons, M. J. (1981) Clustering criteria and multivariate normal mixtures. *Biometrics*, **37**, 35–43.