Model-Based Variable Clustering with Application to Neurophysiology

– Ph.D Thesis Proposal

Liuxia Wang
Department of Statistics
Carnegie Mellon University

August 20, 2004

Abstract

Cluster analysis is the method to put objects into groups on the basis of the property of the objects obtained from the observed data. Most of current cluster analysis is used to classify observations. Motivated by applications to neurophysiology, we are interested in variable clustering, which classifies variables according to their association across repeated observations. The association for variable clustering is typically measured as correlation. Even for scalar data, it is already non–trivial to estimate a correlation matrix from a limited data. The problem becomes much harder for vector or function data. Motivated by applications to neurophysiology, we propose the variable clustering model based on variation across replications. We start with scalar variables, then extend the model to vector data. By using a basis representation (as in James and Sugar, 2003), we are able to construct one functional variable clustering model. The models perform variable clustering and estimation simultaneously. Some simulation results using MCMC, as well as comparisons with the existing models, are displayed. Also, we apply the model to analyze neuronal data. we found that some neurons are generally associated regardless of movement direction and time. The correlation pattern of some neuons switches between states of high correlation and near independence depending on the movement direction and time. There are also some pairs of neurons which are completely independent with each other in all movement directions and time periods.

1 Introduction

Cluster analysis is used to classify observations. Variable clustering instead classifies variables according to their association across repeated observations. We are interested in applications of variable clustering to neurophysiology. Three features make variable clustering a challenging statistical problem in this context. First, the association for variable clustering is typically measured as correlation. Even for scalar data, it is already non–trivial to estimate a correlations matrix from a limited data (Barnard, et al, 2000; Daniels and Kass, 2001; Liechty, et al, 2004). Secondly, correlation is only defined for scalar variables. For vector-valued variables, one may define a correlation matrix to measure the association between two sets of variables. However, it is difficult to define a suitable scalar summary of that matrix. Thirdly, when dealing with functional variables, the problem becomes even harder because of high dimensions.

In the following, I will introduce some background in neuroscience and further explain the motivation for our approach, then I will give the outline of my proposal.

1.1 Motivation

The goal of the field of cognitive neuroscience is to understand how the brain works. A major proportion of cells in the brain are neurons whose main function is to produce signals that initiate and control human reactions. In reality, human functioning is the result of the simultaneous interaction between millions of neurons. In general, neuronal data is the recorded action potential (or *spike*) activity (electrical discharges) obtained using microelectrodes (Evarts, 1968). The data are usually a collection of sequences of spikes along time in multiple replications (or *trials* in neuroscience) of an experiment. The experiments of our colleagues involve arm movement tasks carried out by monkeys.

In neurophysiology, there is great interest in studying whether neuronal firing patterns among neurons change when the task is changed. Recent research shows that the firing characteristics of neurons can change during the performance of certain tasks (Graziano, et al., 2002; Li, et al., 2001). When a monkey is performing a center-out reaching task (explained below), in different periods of the movement the neuronal responses are different. For example, the firing rates change over time. One may ask if the correlation activity among the neurons is changing during movement. In addition to its inherent scientific interest , taking account of correlation can be helpful in reconstructing movement from neuronal signals (Gao, et al. 2002). We would like to study *time-varying* or *activity-related* clustering. Instead of clustering the neurons once for all, we would like to see how the grouping pattern changes over time. If we partition the time interval into several smaller time bins, we may examine how the neurons are associated with each other at different periods of movement. This may also provide some information about the way the neurons' interactions are related to their functions.

One general way to describe the function of a neuron's behavior is firing rate function, which denotes the strength of the response of neuron to the stimulus along time. Cai, Kass and Ventura (2003) use replication-dependent conditional intensity functions to describe trial-to-trial variability, which is how the neurons' response to the stimulus varies from trial to trial. As mentioned before, neurons interact with each other when the subject is performing a behavioral task. When dealing with multiple neurons, it is necessary to take account of correlation. Much of the correlation among neurons is due to trial-to-trial variability. Cai, *et al* (2004) model firing rate functions for two neurons

recorded simultaneously. However, it is of interest to fit firing rate functions for more than two neurons simultaneously. One natural question then becomes which neurons are correlated, which are not.

Another motivation comes from Bayesian sequential methods for neural decoding and control of neural prostheses. Brockwell, *et al.* (2003) assume that the neurons are conditionally independent. To improve the performance of the method, it is desirable to take the correlation into consideration. Because of computational difficulties, the model works well if 12 or less than 12 neurons are correlated, but independent of the others. The question is how we can put the neurons into many groups such that the neurons within group are correlated, while the neurons from different groups are not correlated.

1.2 Proposal Outline

This proposal is organized as follows. In Section 2, we will review several recent models related to variable clustering and functional clustering. In the following three sections, we will present the clustering models for Normal data. We start with scalar variables in Section 3, then extend the model to vector data in Section 4. By using a basis representation (as in James and Sugar, 2003), we are able to construct a functional variable clustering model in Section 5. Some simulation results using MCMC, as well as comparisons with the existing methods, are displayed. Also, we use our model to analyze neuronal data in Section 6, where a brief review relevant neurophysiological background is given. At the end, topics for future work are listed.

2 Review of Existing Clustering Methods

Most clustering done in practice is based largely on heuristic but intuitively reasonable procedures, such as hierarchical agglomerative clustering and k-means clustering (Hartigan,1975; Kaufman and Rousseeuw, 1990). These methods are relatively easy to apply and often give good results. However, they can not make statistical inference, they do not take account of measurement error in the data or estimation, they do not provide assessment of the clustering uncertainty, and they do not make inference on the number of clusters (Oh and Raftery, 2003).

Model-based clustering is a framework for clustering analysis based on probabilistic models, in which objects are assumed to follow a finite mixture of probability distribution such that each component distribution represents a cluster. For reviews, see Cheeseman and Stutz (1995), Fraley and Raftery (2002) and Oh and Raftery (2003). Model-based clustering approaches not only overcome the disadvantages of the heuristic clustering methods mentioned above, but also cluster the objects and estimate the component parameters simultaneously.

The objects to be clustered can be categorized as "*observations*" and "*variables*". Observation object means that each observation corresponds one object, which is always in the form of scalars or vectors. The goal is to cluster all the scalar or vector observations. For variable objects, multiple replications are observed, and the goal is to cluster the variables based on some notion of similarity across replications. Up to now, most of the model-based clustering methods have focused on clustering of observation objects, (Banfield and Raftery, 1993; Fraley and Raftery, 1998, 2002; James and Sugar, 2003). Recently, several algorithms have been developed that can cluster variables (Liechty, et al. 2004, Oh and Raftery, 2003). In the scalar case, similarity may be defined using correlation, and then familiar observation–clustering methods may be applied by taking "distance" to be a function of

correlation. For example, if the distance between two variables is defined to be $1-r$ (r is correlation), hierarchical agglomerative clustering may be applied. This can sometimes give good results (Johnson and Wichern, 1998).

Recently, Liechty, Liechty and Muller (2004) proposed a model for estimating a correlation matrix, which I refer to as the LLM model. The can be applied to variable clustering. See Section 2.1. Another model, based on multidimensional scaling (MDS), is developed by Oh and Raftery (2003). This model carries out multidimensional scaling and clustering simultaneously. Section 2.2 will show more details. In Section 2.3, I will briefly review a model-based functional clustering introduced by James and Sugar (2003), where infinite–dimensional data such as curves are grouped.

2.1 LLM Model

LLM model (Liechty, et al. 2004) is a model–based variable clustering on the basis of correlation. Under the assumption of normality, the sample correlation follows inverse-Wishart distribution. Normal priors are assigned to each correlation elements under the assumption that the correlation matrix is partitioned to several block, with the same correlation within each blocks. This model works pretty well in clustering of variables as well as estimating of correlation matrix. Reversible jump Markov chain Monte Carlo (Green, 1995) is adopted to infer the number of clusters. However, because the model is built directly on the correlation, instead of the real observed data, it may lose some information contained in the data. Secondly, in this model, the correlations among the variables within each group are assumed to have a common mean. When the correlations within a group have both positive and negative values, the common mean will tend to be zero, therefore this model can not distinguish between the clusters. For example, see simulation result in Section 3.5.2.

2.2 MDS Model

Oh and Raftery (2003) develop a clustering model using multidimensional scaling. The intuition is that the objects have latent positions in a Euclidean space, which is solved via multidimensional scaling. Meanwhile, the latent positions are generated from a mixture of multivariate Normal distribution, each one corresponding to one cluster. The model carries out multidimensional scaling and clustering simultaneously. However, similarly to LLM model, all the information from the data are transmitted via correlation matrix, which may not be a good representation of all information contained in the data.

2.3 Functional Clustering

The objects can also be classified into "*Scalar/Vector data*" and "*Functional data*". The methods reviewed above for scalar/vector data are well developed in terms of both observations and variables, but there are limited ways to cluster the functional data. Of course, functional data can be treated as high dimensional vectors. However, it does not only cause intensive computation, but also the estimations are unstable because of its high dimension (Friedman, 1989, Hastie, el al. 1995). James and Sugar (2003) provide a flexible model-based approach to cluster functional data. First, the data is converted into a vector coefficients of using basis functions, then a mixture model is used fr these. This model deals with functional observations, and clusters the curves on the basis of spatial closeness of

the functions. We are instead interested in clustering the vectors of functions, each vector comprising many noisy replications of a function.

3 Clustering Models Based on Replications

In neurophysiology, it is of interest to study how the neurons interact with each other when one specific task is performed by human or primate animals. To this end, multiple neurons' firing activity is collected when a monkey performing 3D center–out reaching task repeatedly. Many neurons work with each other to complete this task. One way to know how the neurons interact with each other is to study the correlation of the firing rate. Trial-to-trial variation is a major source of the correlation among the neurons. Intuitively, when two variables are highly correlated, they share a source of variation in common as they vary from trial to trial. Therefore, we would like to put the neurons with similar across–trial variation into one group. Statistically, the clustering criteria in our model is based on the variation of observations across replications.

In Section 3.1, a probabilistic model for clustering neuronal variables with the consideration of trial-to-trial variability is developed. In addition to clustering, the model also can estimate the correlation matrix as well as variance-covariance matrix via Equation (2). Certain topics relevant to clustering are discussed thereafter. Section 3.5 contains some simulations to examine the accuracy of out model in terms of clustering error rate and estimation of correlation matrix when comparing with LLM model and sample correlation.

3.1 Model

Let y_{ij} denote the firing rate for i^{th} neuron, j^{th} trial. z_i is the indicator of which group the i^{th} neuron belongs to. Let μ_i denote the mean firing rate for i^{th} neuron, which can be treated as fixed effect, and γ_{kj} denote the trial–to–trial variation, which is treated as the random effect. We also add the weights w_i's to each variable to specify the amplitude of the variation for different variable. The model is

$$\begin{aligned} y_{ij} &= \mu_i + w_i\gamma_{z_ij} + \epsilon_{ij}, \quad i = 1, \cdots, N, j = 1, \cdots, N_i \\ \gamma_{kj} &\sim N(0,1), \quad \text{and} \quad \epsilon_{ij} \sim N(0, \sigma_k^2) \quad \text{where } k = z_i. \end{aligned} \tag{1}$$

Markov chain Monte Carlo (MCMC) is adopted to implement the model. Conjugate priors are assigned to all the parameters, hence the full conditional posterior density functions for all parameters are conjugate.

Remark 1 *First, note that, in Model (1), the variables in the same group form a typical factor analysis model, with γ as the latent variable. Secondly, the correlation is estimated via Equation (2) after the relevant parameters, w's, σ's and z's are estimated.*

$$corr(y_{ij}, y_{i'j'}) = \begin{cases} \frac{w_i w_{i'}}{\sqrt{w_i^2+\sigma_k^2}\sqrt{w_{i'}^2+\sigma_k^2}} & \text{if } z_i = z_{i'} \text{ and } j = j', \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

Also, the variance-covariance matrix can be assessed similarly.

In some cases (e.g. Liechty, et al. 2004), it is natural to assume that there is a common correlation among the variable in the same group. Consequently, the variables may share the similar weights ws within group, which may be specified using the common within-group variation. The model is then simplified as

$$y_{ij} = \gamma_i + \gamma_{z_i j} + \epsilon_{ij}, \quad i = 1, \cdots, N, j = 1, \cdots, N_i \tag{3}$$

where $\gamma_{kj} \sim N(0, \kappa_k^2)$ with $k = z_i$ and $\epsilon_{ij} \sim N(0, \sigma_k^2)$. Note that the model within group is a mixed–effects model, which is easier to deal with. The correlation matrix is estimated similarly to Equation (2), and it produces a special correlation structure with the same off–diagonal elements for the correlation within groups.

3.2 Inference on the number of clusters

A fundamental problem in clustering analysis is the determination of "true" number of groups. Numerous approaches to this problem have been suggested over the years, for example, Bayes factor (Banfield et al. 1993), Reversible Jump MCMC (Green, 1995; Liechty, et al. 2004), gap statistics (Tibshirani, et al. 2001) and information theoretic approach (Sugar and James, 2003). Gaussian model-based approach using approximate Bayes factors (Kass and Raftery, 1995) is used widely. Let Y denote the observed data, M_k be the model with k clusters and θ_k denote the parameters for Model M_k. Then

$$P(M_k|Y) = \frac{P(Y|M_k)P(M_k)}{\sum_k P(Y|M_k)P(M_k)}, \text{ where } P(Y|M_k) = \int p(Y|\theta_k, M_k)p(\theta_k)d\theta_k,$$

which is also called integrated likelihood for Model M_k. Kass and Wasserman (1995) showed that, for a variety of models,

$$BIC_k \approx 2\log(P(Y|M_k)) \approx 2l(Y|\theta, z) - \nu_k \log(n). \tag{4}$$

In our model, the observations across replications are not independent, and the number of parameters ν_k also involve the number of observations n, the argument in Raftery (1995), i.e., the second approximation in Equation (4), does not hold for this model.

Sugar and James (2003) develop an information theoretic approach to find the number of clusters. It makes use of rate distortion theory. The procedure is based on "distortion", which is a measure of within cluster dispersion, and is defined as

$$d_K = \frac{1}{p} \min_{c_1, \cdots, c_K} E[(X - c_x)'\Gamma_x(X - c_x)], \tag{5}$$

where X is a random variable having a mixture distribution of K components, each with covariance Γ_i and center c_i, and p is the dimension of X. The number of clusters is estimated by $K^* = \text{argmax}_K J_K = \text{argmax}_K[\hat{d}_K^{-j} - \hat{d}_{K-1}^{-j}]$, where the typical value for j is $p/2$.

3.3 Identifiability

In the literature of mixture model, label switching in MCMC samples can be a tough problem. In the mixture model, the density function is invariant under arbitrary permutation of component labels. Thus, the posterior density function would be invariant under arbitrary permutation of component

labels unless strong information prior is used (Stephens, 2000). This may cause label switching during the MCMC iterations, hence typical averages of MCMC samples of the parameters do not produce reasonable estimates for the mixture parameters. Typically, some constraints are enforced on the parameters (Roeder and Wasserman, 1998; Richardson and Green, 1998). However, this does not always work (Celeux et al., 2000). Therefore, Celeux et al. (2000) provide a clustering-like algorithm for relabeling. One computational difficulty of this procedure is that it compares all the possible permutation of cluster labels. Inspired by Stephen (2000), we adopt optimization technique from operation research to overcome this difficulty, see Appendix A for more details.

Another identifiability problem of our model comes from the factor analysis model. If the signs of w within the same group are changed, the model will stay the same. However, the correlation structure and relative relationship of the amplitude of the variation will not be changed even if the signs are different. Once MCMC becomes stable, the sign of the w's becomes stable too.

3.4 Definition of error rate

After the clustering algorithm is defined, we want to see how stable the algorithm is using simulation studies. Therefore, it is necessary to define some measurement to describe the similarity between the estimated clustering and the reference or true clustering. Several ways to calibrate the clusterings through the relationship between pairs of objects have been proposed, such as Rand (1977), Fowlkes and Mallows (1983). These measures are independent of the cluster labels. However, they only give partial information about similarity of two clusterings in the sense that only pairs of objects are concerned. Furthermore, these similarity definitions only work when an object is either correctly or incorrectly classified. When we use MCMC algorithm, we obtain a posterior probability of an object's grouping. To make use of these probabilities, we define the following procedure, which is similar to the success rate in Ben–Hur and Guyon (2003) except that we are defining error rate with posterior probability of an object's grouping. The posterior probability is defined as

$$p_{ik} = \mathrm{P}(z_i{=}\mathrm{k}) = \frac{\#\text{ of iterations that the variable is classified into Cluster } k}{\text{total }\#\text{ of iterations}} \tag{6}$$

Let C_e denote the estimated clustering and C_t denote the true clustering in the simulation study or a reference clustering otherwise. Define a cost matrix with element

$$M_{kh} = 1 - \frac{1}{|C_t = k|} \sum_{i \in \{C_t = k\}} p_{ih} = \frac{1}{|C_t = k|} \sum_{i \in \{C_t = k\}} (1 - p_{ih}),$$

where $|A|$ denotes the cardinality of the set A and M_{kh} means the error caused by misclassifying the objects in Cluster k into Cluster h. The error rate is defined as

$$\text{error rate} = \frac{1}{G} \sum_{k} M_{kk} \tag{7}$$

where G is the number of clusters. However, in clustering, no knowledge about the clusters is assumed, so the labels $1, 2, \cdots, G$ are arbitrary, and any permutation of the labels represents the same clustering. The label-switching algorithm in Celeux, et al. (2000) only makes sure the labels are consistent over MCMC sampling, but it does not concern about if it is consistent with another clustering. If a

different labels are used, the cost matrix will be different. Therefore, we will select one permutation which minimizes the sum of the diagonal elements. We use the optimization algorithm in Jonker and Volgenant (1987) to do this.

3.5 Simulation Studies

To examine the accuracy of or model, we will compare its performance with the model proposed by Liechty, *et al.* (2004) in terms of clustering error rate and risk in estimation of the correlation matrix. The hierarchical agglomerative clustering based on sample correlation is also considered, where the distance between two variables is defined to be $1-r$ (r is correlation), then hierarchical agglomerative clustering is applied

3.5.1 Methods

The error rate is assessed via Formula (7). The risk in the estimation of correlation matrix is assessed by Stein's loss function

$$l(\Sigma, \hat{\Sigma}) = tr(\hat{\Sigma}\Sigma^{-1}) - \log(|\hat{\Sigma}\Sigma^{-1}|) - p$$

(Lin and Perlman, 1985; Yang and Berger, 1994), where p is the dimension of the matrix.

Here we present two examples from the two models (1) and (3) above respectively. Then compare them to the model in Liechty et al (2004) and the classical estimation via sample correlation. For each example, 30 data sets, with 30 replications in each one, are drawn from multivariate normal distribution with mean $(2, 6, 2, 6, 2, 6, 2, 6)$, which is chosen randomly, and variance-covariance matrix as shown in R_1 and R_2 below. For each MCMC run, 4000 iterations are obtained with burn-in 500. The comparison includes the Stein's loss of the correlation matrix and the error rate via Formula (7). The mean error rate is the average of the error rates over the 30 data sets.

We consider two types of variance-covariance matrices. The first one, R_1, has homogeneous correlation within group, which this kind of correlation structure considered in Liechty *et al.* (2004). The second one, R_2, has non-homogeneous structure so that we can show the power of the model we propose here.

$$R_1 = \begin{pmatrix} 1 & .4 & .4 & .4 & 0 & 0 & 0 & 0 \\ .4 & 1 & .4 & .4 & 0 & 0 & 0 & 0 \\ .4 & .4 & 1 & .4 & 0 & 0 & 0 & 0 \\ .4 & .4 & .4 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & .5 & .5 & .5 \\ 0 & 0 & 0 & 0 & .5 & 1 & .5 & .5 \\ 0 & 0 & 0 & 0 & .5 & .5 & 1 & .5 \\ 0 & 0 & 0 & 0 & .5 & .5 & .5 & 1 \end{pmatrix}, \quad R_2 = \begin{pmatrix} 1 & .59 & -.72 & .94 & 0 & 0 & 0 & 0 \\ .59 & 1 & -.47 & .61 & 0 & 0 & 0 & 0 \\ -.72 & -.47 & 1 & -.74 & 0 & 0 & 0 & 0 \\ .94 & .61 & -.74 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -.52 & .55 & .63 \\ 0 & 0 & 0 & 0 & -.52 & 1 & -.71 & -.81 \\ 0 & 0 & 0 & 0 & .55 & -.71 & 1 & .86 \\ 0 & 0 & 0 & 0 & .63 & -.81 & .86 & 1 \end{pmatrix}.$$

3.5.2 Results

Table 1 contains the risk in the correlation matrix estimation and the mean error rate, as well as the corresponding standard errors. For the first example, all the three methods work great in terms of

		Risk (std error)	Error rate
Example 1	**LLM**	7.774(.289)	.0141(.0052)
	Sample Correlation	1.094(.048)	.0172(.0085)
	Model 3	.095(.019)	.0035(.0015)
Example 2	**LLM**	10.961(4.159)	.1076(.0194)
	Sample correlation	1.325(.376)	.0083(.0059)
	Model 1	.114(.071)	.0012(.00039)

Table 1: *Comparison of LLM, sample correlation and the models proposed in this paper in terms of correlation estimation and clustering error rate. Here, LLM refers to the model in Liechty, et al. (2004). Model 1 denotes the general model specified by Equation (1), and Model 3 is the simpler version of Model 1, and specified by Equation (3). The sample size is 30.*

clustering error rate. However, we still can see our model is much better than the other two. LLM works almost as well as sample correlation. From the aspect of estimation of correlation matrix, without any question, our model beats the other two dramatically. The sample correlation works even better than LLM. As for the second example, it is easy to see that Model (1) beats LLM both in the estimation of the correlation matrix and the error rate of clustering. The error rate is about 100 times smaller.

The simulations above show that Model 1 as well as the simpler version–Model 3 improves the accuracy in terms of clustering error rates and the risk in the estimation of correlation matrix. When the within-group correlations are non-homogeneous, the clustering behavior of LLM is dramatically bad. The reason is that LLM model clusters the variables in terms of the common correlation for each group. The correlation like R_2 leads to zero mean within–group correlation for both groups, therefore, LLM model can not distinguish between these two groups. I also want to point out that hierarchical clustering using sample correlation as measure of similarity does not work as well as our model based on replication, but it is easy to apply. On the other hand, it has some disadvantages discussed in Section 2, and it does not extend to vector or functional variable clustering.

4 Extension to Vector Data

In some real problems, it is unavoidable to deal with vector variables. What's more, vector–valued variable clustering is essential in functional variable clustering, not only because functional data is always represented in form of high-dimensional discrete vector, but also because the functions can be expressed via low-dimensional vectors by projecting them to some finite dimensional functional basis, which can be seen in Section 5. When the observations are vectors, we can write the model similarly to Equation (1). Let $\underline{y}_{ij}$ be the vector observation for i^{th} variable and j^{th} trial, and z_i be the cluster member indicator.

To simplify the notation, we use $*$ to denote the arithmetic multiplication between vectors or vector and matrix. Let a and b denote two vectors of length q,

$$a * b = (a_1 b_1, \cdots, a_q b_q)'. \tag{8}$$

With this notation, the model can be written as

$$\underline{y}_{ij} = \underline{\mu}_i + \underline{w}_i * \underline{\gamma}_{z_i j} + \underline{\epsilon}_{ij}, \quad i = 1, \cdots, N, j = 1, \cdots, N_i \tag{9}$$

$$\underline{\gamma}_{kj} \sim N(0, I), \quad \text{and} \quad \underline{\epsilon}_{ij} \sim N(0, \Sigma_i) \quad k = z_i, \tag{10}$$

where symbol "$\underline{\ }$" denotes "vector". $\underline{w}$ is the weight for the trial variation, "$*$" implies that the amplitude of trial variation for different dimension is different. We may use MCMC to implement the model with conjugate priors. Note that the conjugate prior to Σ_i is

$$\Sigma_i \sim IW(\nu, V_0),$$

where IW denotes "inverse Wishart distribution". We set V_0 to be diagonal so that the estimation of Σ_i shrinks to diagonality. Daniels and Kass (1999) introduce some non-conjugate priors to variance-covariance matrix, which may work better but are computationally expensive. The initial values for MCMC are similar to the scalar model. Some simulations were conducted using the data generated from the model, and it was shown that both the clustering and estimation of parameters are well done.

5 Functional Variable Clustering Model

In Neurophysiology, one popular way to describe the neuron's activity is firing rate function. Therefore, we may cluster the neurons via the information from the firing rate functions. The association among the neurons is measured by how similarly the firing rate functions vary from trial to trial. For each experiment trial, the firing rate function can be estimated via some nonparametric methods (Kass and Venture, 2001; Cai, et al. 2003).

In James and Sugar (2003), they project the function onto a finite spline basis, then model the coefficients using Gaussian distribution. We will implement this idea to cluster functional variables with repeated replications. Instead of clustering the functions based on the common coefficients, we will cluster them based on how closely they are associated across replications. The association among the firing rate functions are represented by the association among the coefficients of the functions.

Let Y_{ij} be the firing rate function for i^{th} neuron and j^{th} trial, which is observed at time points $(t_1, \cdots, t_n)$. For simplicity, we assume that all the functions Y_{ij} are evaluated at the common time points $(t_1, \cdots, t_n)$. Actually, this model does not necessarily have this constraint. Let B be the $d-$dimensional spline basis vectors and z_i be the cluster membership variable. Then

$$Y_{ij} = B\eta_{ij} + \epsilon_{ij}, \quad i = 1, \cdots, N, j = 1, \cdots, M_i \tag{11}$$

where N is the number of neurons to be clustered, M_i is the number of replications for each neuron, and ϵ_i, the measurement error term, has mean zero, variance $\sigma^2_{z_i}$ and is uncorrelated with each other and η_i. Furthermore, analogously to Equation (9), we model the coefficients via

$$\eta_{ij} = \beta_i + w_i * \gamma_{z_i j} + \xi_{ij}, \quad \gamma_{z_i j} \sim N(0, I), \quad \xi_{ij} \sim N(0, \Gamma_{z_i}) \tag{12}$$

where β_i denotes the overall mean of the coefficients for the i^{th} curve, and is treated as fixed effects.

Note that the parameters η, β, λ and w are all vectors of size $d+2$, where d is the number of knots for the spline basis. Actually, this is a two-stage model. In the first stage, which is specified by Equation (11), the coefficients of the functions are evaluated, namely, the functional data is projected to lower dimensional spline space. The second stage, specified by Equation (12), is exactly the vectorial version of clustering model in Equations (9) and (10). Since, essentially, the model is built on the coefficients, we can use the vector version of clustering algorithm in Section 4 to complete this procedure.

One important issue related to functional data is the selection of the spline basis. Most procedures use equally spaced knots, which reduces the problem to selecting the correct number of knots. One natural approach is to take the dimension of the basis, d, to corresponding to the largest cross-validation likelihood (James, *et al.*, 2000). This works well but is computationally expensive. An alternative approach is to calculate AIC or BIC, which apply a penalty term involving the number of parameters to the likelihood (Rice and Wu, 2001). This method appears to be fairly robust to any reasonable number of knots. One may also use BARS (Bayesian Adaptive Regression Splines) to estimate the number and the locations of knots (DiMatteo, et al. 2001).

6 Application to Neuronal Data

Neurons receive information, process it and pass it to downstream neurons. There are billions of neurons in the brain which control the whole body's action. The neurons pass the information in the form of action potentials (also called *spikes*, see Figure 1). For the same neuron, every spike has the same waveform lasting about a millisecond. Different neurons have different waveforms, which are used to identify neurons. Because the precision of most equipment in collecting data is 1 ms, the output of every neuron consists of a sequence of spikes, which is called a *spike train*, see the bottom plot in Figure 1, which represents the spike sequence for one trial, with each short bar representing one spike.

6.1 Explanation of the Experiment

W analyze data that were collected in the laboratory of Dr. Andrew Schwartz, Department of Neurobiology, University of Pittsburgh. The experiment involved a series of 3D center-out reaching tasks. The monkey faces a computer screen which displays eight targets located at eight corners of a cube and one center bar. At the beginning, the monkey's hand can be anywhere. When the center light flashes, the monkey is required to move his hand to the center. Then a light at one of the eight targets appears so that the monkey moves his hand from the center to the highlighted target. The targets are highlighted in a random order. A sensor is attached to the monkey's hand so that the computer can keep track of the hand position. Several electrodes are plugged into the monkey's primary motor cortex area (also called M1 area) so that the activity of the neurons is recorded.

The data file consists of information from 100 trials regarding about the hand movement and neural activity of 65 neurons recorded simultaneously. Correspondingly, the file can be classified as "Behavior data" and "Neuronal data". Behavioral data include the information regarding success index (see if the trial is a success or not); the sequence of the targets (sequence of the targets appearing on the screen); 3-dimensional hand position when the monkey reaches the highlighted target and the time landmarks

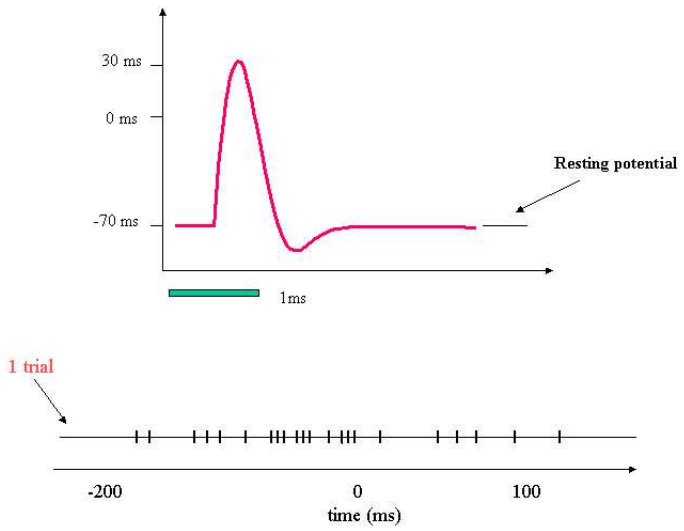

Figure 1: *The upper plot shows a typical spike, and the bottom one represents the spike sequence for one trial, with each short bar representing one spike.*

(Capture, Hold A, Delay, Reaction, Movement and hold B). Neuronal data contain the sequence of spikes obtained from 65 neurons in the primary motor cortex of a monkey's brain. Neurons can be identified by their locations in M1. From the neuronal data, we found that 3 channels actually obtain information from the same neuron. Therefore, essentially, we have 63 neurons.

Since our interest is in the relationship between the hand movement and the activity of neurons, we will only use the Reaction period and Movement period for our analysis later. The delay period is quite short (only about 10 milliseconds), so we will not include this part of action when we consider the movement.

Because the onset and offset of the movement are different across trials, it is necessary to align them so that the trials are comparable. *Landmark Registration* (Ramsay and Silverman, 1997) is adopted to deal with this issue. For this data set, the landmarks we use are onset of Reaction (i.e. the time when target cue appears.), onset of Movement, maximum of Movement and offset of Movement. For each movement direction, all trials are registered based upon these four landmarks. The alignment is based on the behavior, and because of the strong relationship between the behavior and neurons' activity (Moran and Schwartz, 1999), the spikes are aligned using the same transformation.

6.2 Clustering Result

For each movement direction, the whole aligned time is partitioned into 6 bins equally, with bin–width 162.8 ± 12.3 ms. The number of spikes were counted for every neuron, every trial, on each time bin in each movement direction, and the counts were square–root transformed so that they look more like Normal. For each time bin in each movement direction, variable clustering algorithm based on trial–to–trial variation was run to put the 63 neurons into 7 groups, which is chosen by the information

theoretical algorithm of Sugar and James (2003), see Section 3.2.

The output of the clustering algorithm gives the posterior probability that two neurons are in a same group. If this probability is greater than 0.8, we can confidently say that these two neurons are in the same group. If it is less than 0.2, these two neurons are unlikely to be in the same group. If it is between 0.2 and 0.8, we are not sure if they are in the same group and this case maybe categorized as "uncertain", but the posterior probability may provide auxiliary information. Similarly, we can define the probability for any number of neurons in the same group, but the cutoff points may be changed correspondingly. We use the cutoff points of .6 for 3–5 neurons and .4 for more than 6 neurons. Because of the computational difficulty, we can not calculate the probability of all possible combinations of all the 63 neurons. We begin with pairs, which may provide information about further grouping. For the interesting groups with more than two neurons, we will further check the probability that they are actually in the same group.

Following the above strategy, we can work on clustering of the neurons for each bin and each movement direction. Totally, we have 48 combinations of time bins and movement directions. Then we calculate the percentage of the number of combinations where the neurons are grouped over all combinations. If this percentage is big, we say that the neurons are grouped.

From the analysis outlined above we found 3 types of phenomena.

(1) Some neurons are grouped.

From the analysis, basically, we have three groups with strong association:

(a) Neuron 18, 24, and 25 are in the same group. In 32 out of 48 combinations, the probability that these three neurons are in the same group is greater than 0.6.

(b) Neuron 36, 38, 39, 61 and 62 are in the same group. In 12 out of 48 combinations, the probability that these five neurons are in the same group is greater than 0.6.

(c) Neuron 1, 12, 14, 22, 27, 28, 35 and 44 are in the same group. In 13 out of 48 combinations, the probability that these eight neurons are in the same group is greater than 0.4.

(2) Two neurons may be completely independent of each other across all movement directions and all time bins.

9 pairs have this strong independence, and another 122 pairs are almost independent of each other in the sense that in more than 80% of 48 combinations, they are independent. Especially, Neuron 43 is almost independent all the other neurons.

(3) There is one pair whose groupings are switched between "correlated" and "independent".

Neurons 35 and 36 are such a special pair. In 20 of 48 combinations, they are in the same group, and in 23 of combinations, they are independent of each other. The switching of the grouping shows a clear time-varying pattern. Figure 2 shows the probabilities of these two neuron in the same group for all the combinations of movement direction and time bin. The y-axis represents the time bins and the x–axis corresponds to 8 movement directions. The darker the block is, the bigger the probability is. The clearest pattern appears in Directions 3 and 5. In Direction 3, these two neurons are completely independent at the first 2 bins, then they work together afterward. While for Direction 5, these two neurons are not correlated until the 5^{th} bin, then they interact strongly after the maximum of velocity is reached. For the other movement directions,

the pattern is not that apparent, but one still may see it. However, the patterns are different across movement directions.

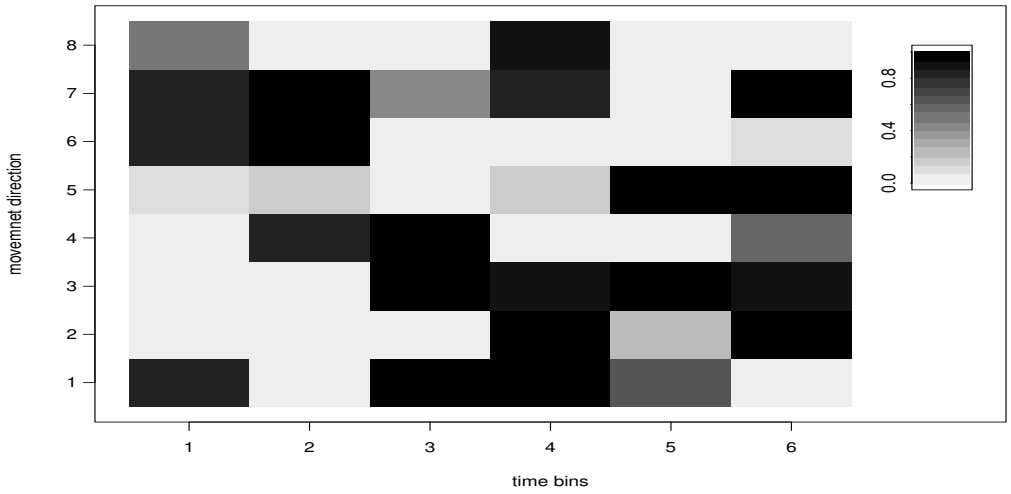

Figure 2: *The posterior probabilities that Neurons 35 and 36 are in the same group for all time bins in every movement direction. The y-axis represents the time bins and the x-axis represents the movement directions. This pair shows a clear temporal clustering pattern. For example, in Direction 3, these two neurons are completely independent at the first 2 bins, then they work together afterward.*

7 Future Work

7.1 Further Analysis on Neuronal Data

In Section 6.2, we study how the clustering patterns of the neurons evolve over time. Since the neurons in Primary Motor Cortex display directional tuning, meaning that their firing rate reaches the maximum when hand movement is in one certain direction (Georgopoulos, *et al*, 1982, Schwartz, *et al*, 1988), therefore it is also of interest to see how the clusterings change over direction. However, the neurons in this data set are not well tuned. Therefore, we would like to collect a new data set, and see whether the clustering is related to the movement directions of the neurons.

7.2 Wrong Model Simulation

In the simulations, usually we generate the data from the model, and then test how the model works. In this way, we always can get the expected result. However, what if the data are not generated from the model? For example, how does the Normal-based variable clustering methods work on the Poisson

data? To address this question, we may try the following two approaches – simulations where the true clustering is known and cross–validation where the true clustering is unknown.

7.3 Non-Gaussian Models

Normal assumption is always tractable in the sense that the results are easily derived and the theoretical foundation is well developed. While in some circumstances, the normality assumption is not satisfied. For example, in neuroscience, the stochastic nature of the spike occurrences generates a Poisson process when the number of trials is relatively large (Daley and Vere-Jones, 1988). Therefore, we would like to extend the trial–to–trial–based clustering methods to the non-Gaussian models. It was noticed that the normal models can work well on the Poisson data after square root transformation (DiMatteo, et. al, 2001). Therefore, we also would like to compare the performance of the normal model and Poisson model using Poisson data.

When the bin-width is very small, say, 1 ms, the observations of the number of spikes within each bin will be 0's or 1's, the binary data. To handle this kind of data, we may extend the model to logit model to cluster the neurons based on the trial–to-trial variation of logit of the probability that one spike occurs.

It is also well known that loglinear model is powerful to estimate the high order interaction and conditional independent structures, (Martignon, et al, 2000; Fienberg, 1980). We may try to cluster the neurons based on the high order interaction instead of correlation and compare these two approaches.

7.4 More factors

In the models specified by Equations (1), (9), and (12), only one factor or latent variable is involved in each model. However, it is quite possible that more factors have contribution to the model, therefore we would like extend the models to include more factors to make the model more sensible and accurate. The techniques related to factor analysis, such as determining the number of factors, are necessary.

A Label-switching Algorithm

To solve label-switching problem in mixture model, Celeux et al. (2000) proposed a relabeling procedure, which is a clustering-like algorithm. Let $\xi^1,\ \xi^2,\ \cdots$ be the sequence of $d-$dimensional MCMC vector samples. The procedure is initialized using the first m vectors $\xi^1,\ \xi^2, \cdots, \xi^m$, where m is typically 100 or so. Reference centers and component-wise variances for $(i = 1, \cdots, d)$ are defined as

$$\bar{\xi_i} = \frac{1}{m}\sum_{j=1}^{m} \xi_i^j,$$

and

$$s_i = \frac{1}{m}\sum_{i=1} m(\xi_i^j - \bar{\xi_i})^2.$$

Set $s_i^{[0]} = s_i,\ i = 1, \cdots, d$. And denote $\bar{\xi}_i^{[0]} = \bar{\xi}$, then the $(k! - 1)$ other clusters $\bar{\xi}_2^{[0]},\ \bar{\xi}_3^{[0]}, \cdots, \bar{\xi}_{k!}^{[0]}$ can be deduced from $\bar{\xi}_i^{[0]}$ by permuting the labeling of the mixture components. After this initializing stage, the rth iteration of the clustering procedure runs as follows:

1. Allocate ξ^{m+r} to the cluster j^* that minimize

$$\| \xi^{m+r} - \bar{\xi}_j^{[r-1]} \|^2 = \sum_{i=1}^{d} \frac{(\xi_i^{m+r} - \bar{\xi}_{ij}^{[r-1]})^2}{s_i^{[r-1]}} \tag{13}$$

   where $\bar{\xi}_{ij}^{[r-1]}$ is the ith coordinate of $\bar{\xi}_j^{[r-1]}$. If $j^* \neq 1$, permute the coordinates of ξ^{m+r} to get $j^* = 1$.

2. Update the $k!$ centers and the d normalizing coefficients:

$$\bar{\xi}_1^{[r]} = \frac{m + r - 1}{m + r} \bar{\xi}_1^{[r-1]} + \frac{1}{m + r} \xi^{m+r}$$

   Derive the $k! - 1$ other centers by permutation, and take

$$s_i^{[r]} = \frac{m + r - 1}{m + r} s_i^{[r-1]} + \frac{m + r - 1}{m + r} (\bar{\xi}_{i1}^{[r-1]} - \bar{\xi}_{i1}^{[r]})^2 + \frac{1}{m + r} (\bar{\xi}_{i1}^{m+r} - \bar{\xi}_{i1}^{[r]})^2$$

   The mode of reference corresponds to $j = 1$ at each iteration. Note that the normalization of the distances in (13) make the procedure independent of location-scale transformations of the parameters even though the resulting estimates depend on the parametrization of ξ.

Actually step 2 is a special case of the Transportation problem in operations research, known as *assignment problem*. The basic idea is to choose one permutation ν to minimize

$$\sum_{j=1}^{d} c(j, \nu(j)) \tag{14}$$

where

$$c(j, l) = \frac{(\xi_l^{m+r} - \bar{\xi}_j^{[r-1]})^2}{s_j^{[r-1]}},$$

which is called *cost* matrix. Equation (14) is equivalent to the integer programming problem – choosing $y_{jl}, j, l = 1, \cdots, d$ to minimize

$$\sum_j \sum_l y_{jl} c(j, l) \tag{15}$$

subject to $y_{jl} = 0$ or 1, and $\sum_j y_{jl} = \sum_j y_{jl} = 1$. If $\hat{y_{jl}}$ is an optimal solution to problem (15), then the corresponding optimal solution to problem (14) is $\hat{\nu(j)} = l$ if and only if $\hat{y_{jl}} = 1$.

We revised the C++ code written by Jonker and Volgenant, who followed the algorithm in Jonker et al. (1987).

References

[1] Anderson, T.M. (2003). *An Introduction to Multivariate Statistical Analysis*, (Third edition), John Wiley & Sons, Inc, New Jersey.

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

[3] Barnard, J, McCulloch, R, and Meng, X. (2000). Modeling covariance matrices in terms of standard deviations and correlations, with applications to shrinkage. *Statistica Sinica* 10, 1281–1311.

[4] Ben-Hur, A. and Guyon, I. (2003). Detecting stable clusters using principal component analysis. *Methods in Molecular Biology*, M.J. Brownstein and A. Khodursky (eds.) Humana press, 159-182.

[5] Brockwell, A.E., Rojas, A.L. and Kass, R.E. (2004). Recursive Bayesian decoding of motor cortical signals by particle filtering, *Journal of Neurophysiology*, 91, 1899–1907.

[6] Cai, C., Kass, R.E. and Ventura, V. (2003). Trial-to-trial variability and its effect on time-varying dependence between two neurons, unpublished.

[7] Celeux, G. ,Hurn, M. and Robert, C.P. (2000). Computational and inferential difficulties with mixture posterior distribution. *JASA*, 95, 957 –97.

[8] Cheeseman, P. and Stutz, J. (1995). Bayesian classification (autoClass): Theory and results, in *Advances in Knowledge Discovery and Data Mining*, Usama M. Fayyad, Gregory Piatetsky-Shapiro, Padhraic Smyth, Ramasamy Uthurusamy, Eds. The AAAI Press, Menlo Park, expected fall.

[9] Daley, D.J. and Vere-Jones, D. (1988). *Introduction to the Theory of Point Processes*, Springer-Verlag, New York.

[10] Daniels, M.J. and Kass, R.E.(1999). Nonconjugate Bayesian estimation of covariance matrices and its use in hierarchical models. *JASA*, 94, 1254–1263.

[11] Daniels, M.J. and Kass, R.E. (2001). Shrinkage estimators for covariance matrices. *Biometrics*, 57, 1173–1184.

[12] DiMatteo, I., Genovese, C.R. and Kass, R.E., (2001). Bayesian curve fitting with free-knot spline, *Biometrika*, 88 (4): 1055 - 1071.

[13] Evarts, E.V. (1968). Relation of pyramidal tract activity of force extended during voluntary movement, *Journal of Neurophysiology*. 31, 14–27.

[14] Fowlkes, E. and Mallows, C. (1983). A method for comparing two hierarchical clusterings. *JASA*, 78, 553–584.

[15] Fraley, C. and Raftery, A.E., (1998). How many clusters? Which clustering method? - Answers via model-based clustering analysis. *The Computer Journal*, 41, 578–588.

[16] Fraley, C. and Raftery, A.E., (2002a). Model-based clustering,discriminant analysis, and density estimation, *JASA*, Vol 97, 611 - 631.

[17] Fraley, C. and Raftery, A.E., (2002b). MCLUST, software for model-based clustering, density estimation and discriminant analysis, *Technical Report No. 415*, Department of Statistics, University of Washington.

[18] Friedman, J.H. (1989). Regularized discriminant analysis, *JASA*, 84, 165–175.

[19] Gao, Y., Black, M.J., Bienenstock, E., Shoham, S. and Gonoghue, J.P. (2002). Probabilistic inference of hand motion from neural activity in motor cortex, in *Advances in Neural Information Processing Systems*, 14, MIT Press.

[20] Graziano, M.S., Taylor, C.S. and Moore, T. (2002).Complex movement evoked by microstimulation of pre-central cortex, *Neuron*, 34, 841–851.

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

[22] Hartigan, J.A. (1975). *Clustering Algorithm*, Wiley, New York.

[23] Hastie, T.J., Buja, A. and Tibshirani, R.J. (1995). Penalized discriminant analysis, *Annals of Statistics*, 23, 73–102.

[24] James, G.M., Hastie, T.J. and Sugar, C.A. (2000). Principal component models for sparse functional data, *Biometrics*, 87, 587–602.

[25] James, G.M. and Sugar C.A. (2003). Clustering for sparsely sampled functional data, *JASA*, 98, 397 – 408.

[26] Johnson, R.A. and Wichern, D.W., (1998). *Applied Multivariate Statistical Analysis*, Fourth edition. Prentice Hall.

[27] Jonker, R. and Volgenant, A. (1987). A shortest augmenting path algorithm for dense and sparse linear assignment problems," *Computing* 38, 325-34.

[28] Kass, R.E. and Raftery, A.E. (1995). Bayes factors, *JASA*, Vol 90, 773–795.

[29] Kass, R.E. and Vanture, V. (2001). A spike-train probability model, *Neural Computation*, 13, 1713–172.

[30] Kass, R.E., Wasserman, L., (1995). A reference Bayesian test for nested hypotheses and its relationship to Schwarz criterion, *JASA*, 90, 431:928-934.

[31] Kaufman, L. and Rousseeuw, P. (1990). *Finding Groups in Data: An Introduction to Clustering Analysis,* Wiley, New York.

[32] Li, C.S., Padoa-Schioppa, M. and Bizzi, E. (2001). Neuronal correlates of motor performance and motor learning in primary motor cortex of monkey adapting to an external force field, *Neuron.* 30, 593–607.

[33] Liechty, J.C., Liechty, M.W. and Müller, P., (2004). Bayesian correlation estimation, *Biometirka*, 91, 1–14.

[34] Lin, S.P. and Perlman, M.D. (1985). An improved procedure for the estimation of a correlation matrix. In *Statistical Theory and Data Analysis*, Elsevier Science Publishers B.V., 369–379.

[35] Meilă, M. (2002). Comparing clusterings, *Technical Report, 418*, Department of Statistics, University of Washington.

[36] Moran, D.W. and Schwartz, A.B. (1999). Motor cortical representation of speed and direction during reaching. *Journal of Neurophysiology*, 82, 2676–2692.

[37] Oh, M-S and Raftery, A. (2003). Model-based clustering with dissimilarities: a Bayesian approach, *Technical Report*, No. 441, Department of Statistics, University of Washington.

[38] Ramsay, J.O. and Silverman, B.W. (1997). *Functional Data Analysis*. New York: Springer.

[39] Rice, J.A. and Wu, C.O. (2001). Nonparametric mixed effects models for unequally sampled noisy curves, *Biometrics*, 57, 253–259.

[40] Richardson, S. and Green, P.J. (1997). On Bayesian analysis of mixtures with an unknown number of components (with discussion). *Journal of the Royal Statistical Society, Series B*, 59, 731–792.

[41] Roeder, K. and Wasserman, L. (1997). Practical Bayesian density estimation using mixture of Normals, *JASA*, 92, 894–902.

[42] Rousseeuw, P.J. and van Zomeren, B.C. (1990). Unmasking multivariate outliers and leverage points (with discussion), *JASA*, 85, 633-651.

[43] Schwarz, G. (1978). Estimating the dimension of a model, *Annals of Statistics*, 6, 461–464.

[44] Sugar, C.A. and James, G.M. (2003). Finding the number of clusters in a data set: an information theoretic approach. *JASA*, 98, 750-763..

[45] Stephens, M. (2000). Dealing with label-switching in mixture models. *JRSS, Series B*, 62, 795–809.

[46] Tibshrani, R., Walther, G. and Hastie, T. (2001). Estimating the number of clusters in a data set via the gap statistics. *JRSS, Series B*,63, 411–423.

[47] Yang, R. and Berger, J.O. (1994). Estimation of a covariance matrix using the reference prior. *The Annals of Statistics*, 22, 1195–1211.

[48] 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.