Approaches to the Label-Switching Problem of Classification, Based on Partition-Space Relabeling and Label-Invariant Visualization

David Farrar,
Statistical Consulting Center and Department of Statistics,
Virginia Polytechnic Institute and State University, Blacksburg

July 15, 2006

ABSTRACT

In the context of interest, a method of cluster analysis is used to classify a set of units into a fixed number of classes. Simulation procedures with various conceptual foundations may be used to evaluate uncertainty, stability, or sampling error of such a classification. However simulation approaches may be subject to a label-switching problem, when a likelihood function, posterior density, or some objective function is invariant under permutation of class labels. We suggest a relabeling algorithm that maximizes a simple measure of agreement among classifications. However, it is known that effective summaries and visualization tools can be based on sample concurrence fractions, which we define as sample fractions with given pairs of units falling in the same cluster, and which are invariant under permutation of class labels. We expand the study of concurrence fractions by presenting a matrix theory, which is employed in relabeling, as well as in elaboration of visualization tools. We explore an ordination approach treating concurrence fractions as similarities between pairs of units. A matrix result supports straightforward application of the method of principal coordinates, leading to ordination plots in which Euclidean distances between pairs of units have a simple relationship to concurrence fractions. The use of concurrence fractions complements relabeling, by providing an efficient initial labeling.

Keywords. Consensus matrix, label-switching, model-based clustering, Monte Carlo simulation, principal coordinates analysis, similarity and dissimilarity

CONTENTS

Introduction 1

1. Partitions and Classifications, with Matrix Representations 4

2. A Label-Invariant approach to post-processing.. 5

2.1 The ECCP matrix 5
2.2 Graphical Procedures for General Dissimilarity Information 6
2.3 Label-Invariant Identification of Clusters from Simulation Output, with Evaluation of Classification Uncertainty 7
2.4 Conditional Estimation of Class-Specific Parameters...................................... 7

3. Relabeling and Related Computations 8

3.1 The Alignment Matrix.. 8
3.2 Matrix Representation of the Criterion for Alignment of Two Classifications .. 9
3.3 A Criterion for Alignment of a Sample of Classifications............................... 10
3.4 A Relabeling Algorithm.. 11

4. Implementation with Multiple MCMC Chains 12

5. Example 12

6. Discussion... 15

Acknowledgement 15

References 16

Appendices 19

A.1 Representing Classifications and Samples of Classifications in R, with a Timing Comparison... 19

A.2 Technical Results Related to Concurrence Fractions.................................... 22

INTRODUCTION

An important type of non-hierarchical cluster analysis divides a set of *n* objects into *k* disjoint subsets (Mardia et al., 1977; Seber et al., 1984). Various Monte Carlo methods may be useful in these situations, which have different conceptual foundations, but which have in common the generation of a sample of classifications, of size n_{MC} say. In model-based clustering, Markov chain Monte Carlo (MCMC) procedures may be used to explore a likelihood surface, or to a sample a posterior, allowing assessment of uncertainty in a classification jointly with uncertainty for other model unknowns (Celeux, Hurn, and Robert, 2000; Denison et al., 2002; Diebolt and Robert, 1994; Farrar et al., 2006a; Lipkovich, 2002; Viele and Tong, 2002). Non-Bayesian simulation approaches include re-sampling, sub-sampling, and parametric bootstrap sampling, used to evaluate stability or sampling error (Dudoit and Fridlyand, 2002; MacLachlan and Peel, 2000; Monti et al., 2003; Qin and Self, 2005; Rocke and Dai, 2003).

Simulation procedures with various foundations may be subject to the *label-switching* problem, as the problem has been termed in the literature of finite mixture models (Celeux et al., 2000; Stephens 2000; Früwirth-Schnatter, 2001; see also Lipkovich, 2002**)**. The term reflects the possibility that the same classification may recur in a sample, with the classes indexed differently. In the context of statistical modeling, the problem may be viewed as a form of model non-identification. The wider problem is that a function used to represent the quality of a classification is invariant under a permutation of class labels. Label-switching is expected to be particularly a problem if, as in our applications, a posterior sample is formed by pooling multiple chains generated using Markov chain Monte Carlo.

One type of remedy involves adjustments of the class labels in the sample. The term *relabeling* is used in the literature of finite mixture models (Celeux et al., 2000; Qin and Self, 2006; Stephens, 2000). A term suggested by Lipkovich (2002) is *alignment*, based on analogy to the alignment problem of factor analysis (Clarkson, 1979; Ichikawa and

Konishi, 1995). Relabeling (or alignment) may be desirable in computation of class-specific summaries or estimates.

Various relabeling procedures may be plausible in a given situation, associated with different model unknowns. Much of model-based clustering is based on finite mixture models (e.g., Fraley and Raftery, 1998, 2002), for which one standard computational approach is a Gibbs sampler (e.g., Bensmail et al., 1997; Diebolt and Robert, 1994; Viele and Tong, 2002; Wasserman, 2002). In this context, Stephens (2000) suggests relabeling based on comparison of membership probabilities, as computed for a Gibbs sampler. Alternatively, relabeling may be based on class-specific parameter estimates (Celeux et al., 2000). Lipkovich (2002) suggests adjusting class labels to maximize some correlation among classifications in MCMC output. We follow Lipkovich in emphasizing manipulations of classifications, for reasons that include possible utility in connection with diverse Monte Carlo procedures.

However, useful summaries are available that are unaffected by any indexing of classifications in the sample. Particularly helpful are summaries that depend on recording, for particular pairs of units, whether both members of the pair fall in the same or different classes. Monti et al. (2003, hereafter MTMG) used the term *consensus matrix* to describe an $n \times n$ matrix that gives, for each pair of units, the sample fraction with both members of the pair falling in the same cluster. The primary interest of MTMG related to the use of subsampling to evaluate stability of classifications. They discuss applications for evaluating the degree of support for clusters, for ranking units according to their value for representing particular clusters, and for estimating the number of clusters. MTMG did not relate their approach to the label-switching problem, and did not actually mention the very important property of label-invariance. For Tibsharini et al. (2001) develop a concept of prediction strength based on the concurrence fractions – termed by those authors *co-membership* probabilities – to be used in cross-validation.

We expand in several ways on the work of MTMG. A simple matrix theory is introduced, and used in the contexts of relabeling and visualization. In applications to

Bayes posterior sampling, we have used the term *estimated co-clustering probability* (ECCP) in place of the consensus matrix of MTMG (Farrar et al., 2006a). However, a term that will be apparent from out matrix approach, and that may be appropriate for more general applications, is *sample concurrence fractions*.

We find that an effective visualization of information in the ECCP matrix can be based on multidimensional scaling (MDS). We provide a justification for applying a particularly straightforward MDS procedure, the method of principal coordinates (Gower, 1966). Indeed, we observe that when the ECCP matrix is transformed as customary for principal coordinates analysis, the result is a non-negative definite matrix, as assumed by the procedure. Euclidean distances among units, computed using the approach, have a simple relationship to the ECCP.

We present a relabeling approach, approximately maximizing a particularly transparent criterion of similarity among partitions. In case of exhaustive evaluation of $k\,!$ permutations of labels for a classification, only limited special computations are required for each label permutation. We introduce an effective initialization based on concurrence fractions. For the initial labeling we select *k* units that with high probability belong to distinct clusters. These are treated in effect as a training sample, requiring each to belong to a distinct cluster where possible. Some information is given on computing time.

The study is organized as follows. In Section 1 we present some terminology and matrix theory. Label-invariant summarization based on concurrence fractions is the topic of Section 2, while Section 3 presents our approach to relabeling. In Section 4 we attempt an integration of the procedures with other post-processing tasks in processing MCMC output with multiple independent chains. Section 5 illustrates the procedures based on work in progress involving model-based clustering. Some technical results are presented in appendices.

Any computational procedure for manipulating classifications will require some approach for representing a classification in computer code. An appendix gives some remarks on representing classifications in R, which we have used to implement the methods. We report large differences in computational performance associated with alternative representations, in an R implementation.

Our applied interest is in use of model-based clustering to evaluate regional variation in ecological stressor-response relationships, where the units for clustering are environmental monitoring stations or groups of nearby stations (Lamon and Stow 2004; Lipkovich, 2002). We have applied the methods considered here routinely, in that context.

1. PARTITIONS AND CLASSIFICATIONS, WITH MATRIX REPRESENTATIONS

We will find it convenient to distinguish between a *partition* and a *classification* or *labeled partition*. A representation of a partition is to state whether or not, for each pair of units, both members of the pair fall in the same group. The groups of units that define a partition will be termed *clusters*. A classification is obtained by associating some label or index with each cluster in a partition. Thus for any partition comprising *k* clusters, there are $k\,!$ classifications associated with permutations of k labels. The groups with associated labels will be termed *classes*. In our context, the "labels" will be $1, \ldots, k$, so that labeling is effectively an ordering or indexing. However, references to labels are conventional, particularly in finite mixture literature.

Our distinction is clarified somewhat in the following matrix representation. Borrowing terminology from experimental design, a classification in our terminology may be represented by an $n \times k$ *incidence* matrix $\mathbf{Z}$. The value in the *i*th row and *j*th column of the matrix equals 1 if the *i*th unit belongs to the *k*th cluster, according to a specific choice of class labels, and otherwise equals zero. A partition may be represented by a *concurrence* matrix $\mathbf{M}$, a symmetric $n \times n$ matrix where the value in the *i*th row and *j*th column is 1 if units *i* and *j* belong to the same cluster, and otherwise zero. $\mathbf{M}$ is termed a *connectivity matrix* by MTMG. By the definition of a partition, $\mathbf{M}$ can be arranged into a block-diagonal form by some permutation of the order of units. However, in our applications the order of units is fixed.

Associated with a concurrence matrix **M**, representing some partition, there are $k!$ incidence matrices associated with the possible classifications, identical up to a permutation of the order of columns. Indeed, for a concurrence matrix **M**, suppose $\mathbf{Z}^*$ is an incidence matrix derived by permuting columns of another incidence matrix **Z**, so that both represent the same partition. Then $\mathbf{Z}^* = \mathbf{ZB}$ where $\mathbf{B}$ is a permutation matrix. Properties of permutation matrices are discussed in sources such as Harville (1997). Using the orthonormal property of **B**, we have

$$\mathbf{Z}^*\mathbf{Z}^{*\mathrm{T}} = \mathbf{ZB}\left(\mathbf{ZB}\right)^{\mathrm{T}} = \mathbf{ZBB}^{\mathrm{T}}\mathbf{Z}^{\mathrm{T}} = \mathbf{ZZ}^{\mathrm{T}} = \mathbf{M}\,.$$

The final inequality appears as Lemma 1 in the appendix.

Our interest in concurrence matrices is based on their relationship to the ECCP (or consensus matrix of MTMG), as described in the next section.

2. A LABEL-INVARIANT APPROACH TO POST-PROCESSING

2.1 The ECCP matrix

We suppose that some Monte Carlo scheme such as resampling or Bayes posterior sampling generates a random sample of classifications, and let $\mathbf{M}_1^*, \ldots, \mathbf{M}_{n_{\mathrm{MC}}}^*$ denote the corresponding concurrence matrices. The matrix **ECCP** (or consensus matrix of MTMG) gives the sample fractions where pairs of units occur in the same cluster,

$$\mathbf{ECCP} = \frac{1}{n_{\mathrm{MC}}}\sum_{s=1}^{n_{\mathrm{MC}}}\mathbf{M}_s^*\,. \qquad \textbf{(1)}$$

We will let ECCP_{ij} denote the value in the *i*th row and *j*th column, equal to the sample fraction with the *i*th and *j*th units falling in the same cluster. The matrix is symmetric with diagonal values equal to unity. Like the concurrence matrices, the ECCP do not depend on the labeling of particular classifications in the sample. In the remainder of this section we outline several applications.

Our term ECCP reflects an emphasis on Bayes posterior sampling, where the ECCP are estimates of posterior probabilities. For a more general term, we suggest *concurrence fractions*.

2.2 Graphical Procedures for General Dissimilarity Information

Some effective graphical tools for evaluation of the simulation sample can be based on the ECCP, and thus do not require relabeling. MTMG rely for graphical display primarily on direct display of the matrix, with relative magnitudes conveyed using different colors or intensities, and with matrix rows and columns sorted following the approach of Bar-Joseph (2001). (In R, the contributed package `cba` of Buchta and Hahsler, 2005, includes an implementation of that sorting approach.)

We find that useful graphical displays can be based on statistical procedures, such as cluster analysis and multidimensional scaling, for evaluation of general similarity or dissimilarity information (Mardia et al., 1977; Seber et al., 1984). We currently rely primarily on two graphs, an average-linkage dendrogram, and a plot of principal coordinates. In any case the quantities

$$d_{ij} = 1 - \mathrm{ECCP}_{ij}\,, \qquad (2)$$

interpreted as sample fractions with particular units falling in different clusters, and treated as measures of dissimilarity, are found to play an important role.

Principal coordinates represents a straightforward solution of the problem of ordination or multidimensional scaling (MDS). This is the problem of finding a correspondence between our *n* units and coordinates in an *n*-dimensional Euclidean space, such that distances between the coordinates have a definite, ideally simple relationship to a matrix of similarities. Issues in MDS then are properties to require for the similarities, and the computation of coordinates for plotting. The method of principal coordinates, encapsulated essentially in Theorem 1 (appendix), is applicable when a particular transformation of the matrix of similarities is non-negative definite, and then provides a useful set of coordinates. We observe (Corollary 1) that we do obtain a non-negative definite matrix, when the transformation is applied to ECCP. Moreover, squared Euclidean distances generated by the method are proportional to dissimilarities of the form (2).

2.3 Label-Invariant Identification of Clusters from Simulation Output, with Evaluation of Classification Uncertainty

An obvious approach for assigning units to clusters depends on relabeling the simulation sample. After relabeling, we may compute the sample frequency with a given unit belonging to each class, and assign the unit to the class that includes it with highest frequency. To quantify uncertainty, it is natural to subtract the maximum membership probability from one (Fraley and Raftery, 2005; Lipkovich, 2002).

A label-invariant approach can be derived from the dendrogram described in the previous section. From the dendrogram we may extract k clusters by "cutting" the dendrogram at an appropriate plotting height. Using R library functions, a dendrogram object may be generated using the function `hclust`, and clusters extracted from the object using function `cutree`. When the units are classified according to this procedure, a plausible index of uncertainty for co-clustering of units i and j is

$$U_{ij} = \mathrm{ECCP}_{ij} \left(1 - \mathrm{ECCP}_{ij}\right) \qquad (3)$$

2.4 Conditional Estimation of Class-Specific Parameters

It addition to inference of a classification, we may be interested in inference of class-specific parameters. A naive approach is to estimate the classification, then estimate class-specific parameters with the classification fixed, treated as if known to be the true classification. Based on cluster analysis literature, we expect such a conditional approach to be biased, overestimating the differences among groups (Gordon, 1966). A more sophisticated approach may involve an evaluation of joint uncertainty for the partition and other model unknowns. Relabeling may be needed for estimation of class-specific quantities.

The naive, conditional procedure may be termed *greedy*, a term that can be used when a set of unknowns are estimated in a specified order, treating the unknowns estimated at a given point in the series as known, for purposes of estimating the next unknown. We suggest that the greedy approach may be relatively easy to implement, and may be convenient for a qualitative description of differences among classes.

3. RELABELING AND RELATED COMPUTATIONS

Relabeling of the sample may be considered desirable in the context of class-specific summaries, for example if we compute the sample fraction with a particular unit assigned to a particular class. In this section we first develop the alignment criterion, to be maximized by adjustment of class labels, before presenting our relabeling algorithm. In developing the alignment criterion, we first consider in Section 3.1 the comparison of two classifications, before considering the general case of a sample with two or more classifications.

3.1 The Alignment Matrix

As a preliminary, it is helpful to review the following tabular approach for comparing classifications. We illustrate the approach with two classifications of the same 5 units into 2 groups, represented by P = (1,1,1,2,2) and Q = (2,2,2,1,2). Here, each classification is represented by a vector with the ith coordinate giving the class index for the ith unit. It is useful to cross-classify the units according to their classes under P and Q:

Table 1 Example of an alignment matrix

Cluster Index in *P*	**Cluster Index in *Q***	
	1	**2**
1	0	3
2	1	1

The cell count in the ith row and jth column, say n_{ij} , is the number of units that fall in the ith class under P and in the jth class under Q. We suggest calling such a table an *alignment matrix*.

The alignment matrix is apparently useful for multiple computations. We suggest that the matrix trace, which gives the count of units with the same label under each classification, can be taken as a quantity to be maximized in a relabeling algorithm. Also, certain indices used to express the agreement between (unlabeled) partitions are computed from the matrix (Rand, 1971; Hubert and Arabie, 1985).

3.2 Matrix Representation of a Criterion for Aligning Two Classifications

For mutual alignment of two classifications, our suggested approach is to maximize the trace of the alignment matrix described in the previous section. The suggested criterion is related to the index of Cohen (1960), used for quantifying inter-observer reliability in recording a categorical variable.

Let two classifications of the same units be represented using incidence matrices $\mathbf{Z}_1$ and $\mathbf{Z}_2$. We first observe that the alignment matrix can be can be expressed as $\mathbf{Z}_1^{\mathrm{T}}\mathbf{Z}_2$. Therefore the proposed alignment value, is the trace

$$\mathrm{A} = \mathtt{tr}\left(\mathbf{Z}_1^{\mathrm{T}}\mathbf{Z}_2\right)$$

In maximizing this quantity, we will adopt a convention of holding the labels fixed for the first classification and varying the labels for the second. For a matrix representation, the optimal alignment can be represented as choosing a permutation matrix **B** maximizing

$$\mathrm{A}\left(\mathbf{B}\right) = \mathtt{tr}\left(\mathbf{Z}_1^{\mathrm{T}}\mathbf{Z}_2\mathbf{B}\right). \tag{4}$$

For our example, if we fix the labels for *P* and swap the class labels for *Q*, the recomputed alignment matrix is $\begin{pmatrix} 3 & 0 \\ 1 & 1 \end{pmatrix}$. The trace increases from 1 to 4, which we take to suggest that the swap of class labels is appropriate.

While we use permutation matrices to complete our matrix representation, use of such matrices in computer code would be inefficient.

3.3 A Criterion for Alignment of a Sample of Classifications

The suggested criterion for alignment of two classifications can be generalized for relabeling a sample with two or more classifications. In our approach, each classification is relabeled based on comparison to a summary of other classifications in the sample, treating the latter as mutually aligned.

Suppose we align n_{MC} classifications which, prior to relabeling, have incidence matrices $\mathbf{Z}_1, \ldots, \mathbf{Z}_{n_{\mathrm{MC}}}$. Without loss of generality, consider alignment of the first classification. It seems that a useful generalization of the alignment matrix is $\left(\mathbf{Z}_2 \; + \; + \cdots + \; \mathbf{Z}_{n_{\mathrm{MC}}}\right)^{\mathrm{T}} \mathbf{Z}_1$. We again maximize the trace over column permutations, i.e., choose a permutation matrix **B** maximizing

$$\mathrm{A}\left(\mathbf{B}\right) = \mathtt{tr}\left(\left(\sum_{j=2}^{n_{\mathrm{MC}}} \mathbf{Z}_j\right)^{\mathrm{T}} \mathbf{Z}_1\mathbf{B}\right). \tag{5}$$

The approach can be iterated, on each iteration maximizing (**5**) with incidence matrices carried forward from the previous iteration.

An objective function that may be optimized by such a procedure, neglecting local optima, is

$$\sum_{i \neq j} \mathrm{tr}\left(\mathbf{Z}_i^{\mathrm{T}} \mathbf{Z}_j\right),$$

the sum of values given by Expression (4), over pairs of classifications. An iteration of our approach will not lead to a decrease in value of this objective function. Indeed, after a single iteration the objective function will increase by a value equal to the increase in $\mathrm{A}\left(\mathbf{B}\right)$.

3.4 A Relabeling Algorithm

Our relabeling algorithm follows Stephens (2000), in relabeling individual items in a sample (in our case, individual classifications) based on comparison to summaries of other items in the sample. However, whereas in Stephens' algorithm the comparisons are based on membership probabilities, as generated by a Gibbs sampling implementation of a finite mixture model, for our algorithm the comparisons are between classifications. Also, our approach incorporates an initial relabeling based on concurrence fractions. The steps of the suggested algorithm are as follows:

(1) Use a label-invariant procedure to select *k* cluster representative units (CRU), one for each class.

(2) Use the CRU from (1) in a preliminary relabeling of each classification in the sample, where possible defining class *j* in a classification as the class that includes CRU_j.

(3) Adjust the labels for each classification so as to maximize (5).

For the selection of CRU in Step (1), note that for an ideal set, there would be zero probability that any two belong to the same cluster. Accordingly, we select k units that minimize the sum of cells in the corresponding $k \times k$ submatrix of **ECCP**.

For the preliminary relabeling of Step (2), it can happen that a cluster contains more than one CRU, so that other clusters will contain no CRU. In that case, considering our objective of generating a preliminary relabeling in Step (2), the problematic classifications in the sample are simply ignored.

Although Step (3) may be iterated, currently execute a single iteration, based on the apparent effectiveness of the initial labeling. We rely in Step (3) on exhaustive evaluation of the $k!$ label permutations, a manageable number given that we currently consider 2-4 classes. We note that the alignment table does not need to be recomputed for each label permutation. For the particular case of 2 classes, Step (4) amounts to noting whether or not the trace of the alignment matrix is larger than the sum of values in non-diagonal positions.

4. IMPLEMENTATION WITH MULTIPLE MCMC CHAINS

In our MCMC applications to model-based clustering, we rely on multiple chains, which are initiated randomly and independently. Independent chains may be useful for detecting local optima and assessing convergence (Gelman et al., 2003). Particularly, with multiple chains, the methods discussed may need to be integrated with other operations. For some calculations we may delete a leading burn-in sequence form each chain, so that the remainder of each chain is approximately at steady state. We suggest that is helpful to distinguish between computations that (1) do not require relabeling or deletion of a burn-in (e.g., plotting the likelihood); (2) require deletion of the burn-in but do not require relabeling (e.g., methods based on ECCP); or (3) require both burn-in deletion and relabeling (e.g., class-specific summaries).

For our current work we generate 10 independent chains of equal length and plot log-likelihoods end-to-end on one graph, to detect local optima. We then delete a burn-in from each chain. We evaluate convergence based on comparison of the chains using a method of Gelman and Rubin (1992), as implemented in `coda` (Plummer et al., 2005), with the burn-in deleted from each chain. We currently diagnose convergence only for the log-likelihood, which does not depend on cluster labels. Criteria of convergence of a classification have not been developed.

5. EXAMPLE

We illustrate the procedures using selected results from an analysis in preparation (Farrar, 2006b), relating a measure of ecological quality to two land-use predictor variables, for monitoring stations on streams in Maryland, USA (Mercurio et al., 1999). For cluster analysis, the stations were grouped into 17 combinations of 2 physiographic regions and 12 river basins, which served as the units clustered. We applied a form of model-based clustering, with class regression models relating our response variable to two predictors.

The regression model was additive in *B*-splines for the two predictors. We assumed flat priors for regression model parameters. Our prior for the partition assumes that each partition is equally probable *a priori*, subject to a minimum count of 20 stations per cluster. The posterior distribution was sampled using a Metropolis algorithm. We implemented the approach with the number of classes fixed at 2-5.

For each model, we generated 10 chains of length 20,000, thinned each chain by removing odd-numbered iterates to reduce memory demand, and deleted the first half of each chain to allow for equilibration. Analysis of convergence suggested that fewer than 1000 iterations are required, beyond our burn-in.

For the sake of illustration, our Figure 1 shows some results based on a model with 3 classes. From a plot of the log-likelihood, it appears that our sampling procedure is prone to become trapped in regions of the model space associated with relatively low likelihood. Subsequent analyses are based on chains 1, 2, 3, 5, and 9, chosen by inspection of the log-likelihood plot. In a histogram of the ECCP, it is seen that many values are close to either 1 or 0, suggesting that there is often high confidence that, respectively, a pair of units belongs to the same cluster or to separate clusters. The two additional graphs are a single linkage dendrogram based on the ECCP, and an ordination plot. These graphs are in good qualitative agreement with regard to the units with the greatest uncertainty in assignment to classes. In addition, we have computed class membership probabilities based on pooled chains, relabeled according to our suggested procedure. The results are in good agreement with those shown in Figure 1.

We have carried out a preliminary evaluation of the relabeling algorithm with 2-4 class models, in Windows XP on a 1.6 Gigahertz Pentium processor. Generally the computational expense was larger with increasing *k*. For 2-4 classes, the initial labeling required 0.6-1 second per thousand sample size, while the refinement in the final step required 1-2 seconds per thousand. Such computing times are negligible relative to the hours of computing required for generating the samples, but can still represent a nuisance in some situations. Relabeling may have been relatively rapid in our applications because of the small number of units and classes. Also, in our implementation results of relabeling were stored without rearranging output matrices or other objects used to represent the unlabeled sample. The latter approach would require additional time.

An indication of the effectiveness of the initial labeling is the fraction of classifications revised in the final step of the algorithm. We find that the result of the initial labeling is often close to our final result. There is some indication that the initial labeling is more effective for a small number of classes. The fraction of classifications revised was 0 for 2 classes, fewer than 1 per thousand for 3 classes, and 1% for 4 classes.

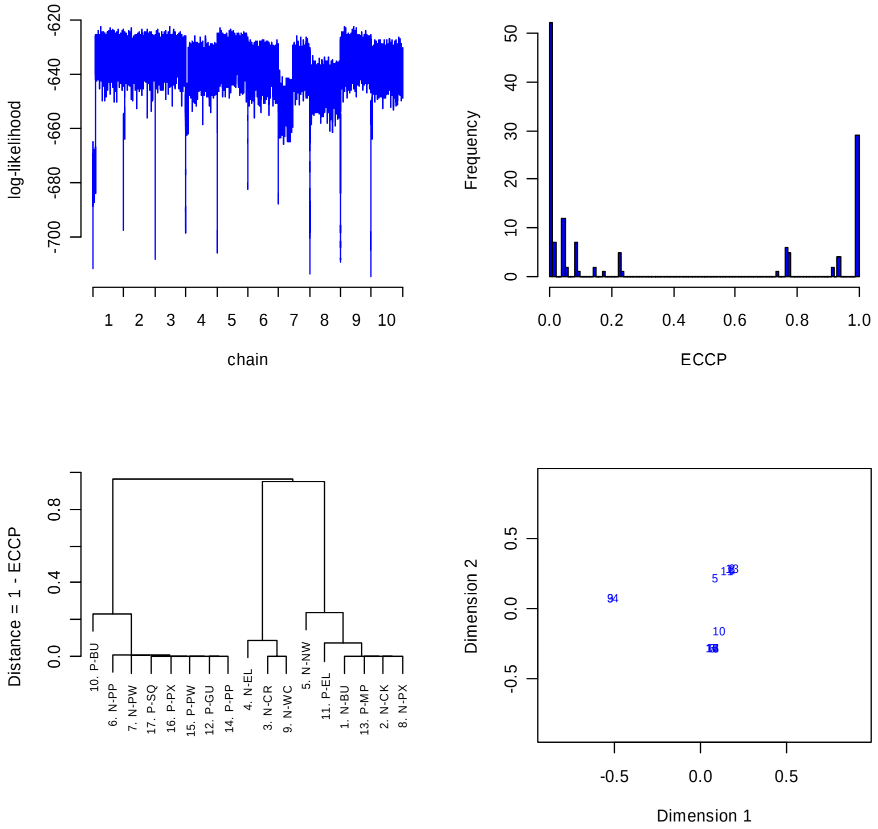

Figure 1 *Illustration of Application to MCMC results*. Clockwise from upper right: Plot of log-likelihood for 10 chains, histogram of ECCP, ordination plot, and average-linkage dendrogram.

6. DISCUSSION

Our relabeling procedure can be said to operate in “partition space,” one of three spaces where proposed relabeling algorithms operate. Alternatives are relabeling in the space of class distribution parameters (Celeux et al., 2000; Qin and Self, 2005) and in a space of estimated class probabilities (Stephens, 2000). Also, following Stephens, we perform relabeling by maximizing a measure of agreement between a single classification and a summary of classifications in the sample. An alternative is to assign labels based on comparisons to a single classification in the sample, perhaps representing a maximum likelihood estimate (Lipkovich, 2002; Qin and Self, 2005). From our viewpoint, the multiplicity of reasonable relabeling approaches adds somewhat to the appeal of label-invariant procedures.

In view of our principal emphasis on data analysis, quantitative comparisons of labeling procedures have not been undertaken. However, some comparisons among relabeling approaches may be of interest. Criteria for comparisons may relate to computational efficiency or effects on inference.

In particular, we do not know of any evaluation of sampling properties of cluster analysis procedures that incorporate relabeling. Two conjectures are (1) that it is best to carry out relabeling in a space where the clusters are relatively distinct; and (2) when performing inference in one space, we may prefer to carry out relabeling in a different space. The second conjecture is based on an expectation that relabeling will restrict overlap of clusters, particularly in the space where relabeling is carried out.

ACKNOWLEDGEMENT

Funding for this research was received from U.S. EPA STAR Grant RD 83136801-0. The research has not been subjected to any USEPA review and so no USEPA endorsement should be inferred.

REFERENCES

Bar-Joseph, Z., Demaine, E.D., Gifford, G.K., and Jaakkola, T. (2001) Fast Optimal Leaf Ordering for Hierarchical Clustering. Bioinformatics, Vol. 17 Suppl. 1, pp. 22-29.

Bensmail, H., Celeux, G., Raftery, A., and Robert, C. (1997). Inference in model-based clustering. Statistics and Computing 7:1-10.

Buchta, C., and Hahsler, M. (2005). `cba`: Clustering for Business Analytics. [Available with manual from the web site of the Comprehensive R archive network (CRAN) http://cran.r-project.org/]

Celeux, G., Hurn, M., and Robert, C. (2000). Computational and inferential difficulties with mixture posterior distributions. Journal of the American Statistical Association 95:957-970.

Clarkson, D.B. (1979). Estimating the standard errors of rotated factor loadings by jack-knife. Psychometrika, 44, 3, 297-314.

Cohen, J. (1960). A coefficient of agreement for nominal scales. Educational and Psychological Measurement 20:37–46.

Denison, D.G.T., Holmes, C.C., Mallick, B.K., and Smith, A.F.M. (2002). Bayesian Methods for Nonlinear Classification and Regression. Wiley, New York.

Diebolt, J., and Robert, C.P. (1994). Estimation of finite mixture distributions by Bayesian sampling. Journal of the Royal Statistical Society B 56:363-375.

Dudoit, S., and Fridlyand, J. (2002). A prediction-based method for estimating the number of clusters in a dataset. Genome Biology 3:0036.1 - 0036.21.

Farrar, D.B., Bates Prins, S.C., and Smith, E.P. (2006a). A Finite Mixture Approach for Identification of Geographic Regions with Distinctive Ecological Stressor-Response Relationships. Technical Report 06-3. Virginia Tech Department of Statistics.

Farrar, D.B., and Smith, E.P. (2006b). Model-Based Clustering with Additive *B*-Spline Regressions, with Application to Ecological Stressor-Response Modeling. (In preparation)

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

Fraley, C., and Raftery, A.E. (2002). Model-based clustering, discriminant analysis, density estimation. Journal of the American Statistical Association 97:611-631.

Früwirth-Schnatter, S. (2001). Markov chain Monte Carlo estimation of classical and dynamic switching and mixture models. J. American Statistical Assoc. 96:194-209.

Gelman, A., and Rubin, D.B. (1992). Inference from iterative simulation using multiple sequences. Statistical Science 7:457-511.

Gelman, A., Carlin, J.B., Stern, H.S., and Rubin, D.B. (2003). Bayesian Data Analysis. (2nd edition) Chapman and Hall, New York.

Gordon, A.D. (1999). Classification. (2nd edition) Chapman & Hall/CRC, New York.

Gower, J.C. (1996). Some distance properties of latent root and vector methods used in multivariate analysis. Biometrika 53:325-338.

Harville, D.A. (1997). Matrix Algebra from a Statistician's Perspective. Springer, New York.

Hubert, L., and Arabie, P. (1985). Comparing partitions. J. Classification 2:193-218.

Ichikawa, M. and Konishi, S. (1995). Application of the bootstrap methods in factor analysis. Psychometrika, 60:77-93.

Lamon, E.C., and Stow, C.A. (2004). Bayesian methods for regional-scale eutrophication models. Water Research 38:2764-2774.

Lipkovich, I.A. (2002). Bayesian Model Averaging and Variable Selection in Multivariate Ecological Models. Dissertation, Virginia Polytechnic Inst. and State. University.

Mardia, K.V., Kent, J.T., and Bibby, J.M. (1979). Multivariate Analysis. Academic Press, New York.

McLachlan, G., and Peel, D. (2000). Finite Mixture Models. Wiley, New York.

Mercurio, G., Chaillou, J.C., and Roth, N.E. (1999). Guide to using 1995-1997 Maryland Biological Stream Survey Data. Versar Inc.

Monti, S., Tamayo, P., Mesirov, J., and Golub, T. (2003). Consensus clustering: a resampling-based method for class discovery and visualization of gene expression microarray data. Machine Learning 52:91-118.

Plummer, M., Best, N., Cowles, K., and Vines, K. (2005). The coda Package. [Available on the web site of the Comprehensive R Archive Network http://cran.r-project.org/]

Qin, L., and Self, S.G. (2006). The clustering of regression models method with applications in gene expression data. Biometrics 62:526-533.

Rand, W.M. (1971). Objective criteria for the evaluation of clustering methods. J. Amer. Statist. Assoc. 66:846-850.

Rocke, D.M., and Dai, J. (2003). Sampling and subsampling for cluster analysis in data mining with applications to sky survey data. Data Mining and Knowledge Discovery 7:215-232.

Seber, G.A.F. (1984). Multivariate Observations. Wiley, New York.

Stephens, M. (2000). Dealing with label-switching in mixture models. Journal of the Royal Statistical Society B 62:795-809.

Tibshirani, R., Walther, G., Botstein, D., and Brown, P. (2001). Cluster validation by prediction strength. Technical Report, Standford University.

Viele, K., and Tong, B. (2002). Modeling with mixtures of linear regressions. Statistics and Computing 12:315-330.

Wasserman, L. (2000). Asymptotic inference for mixture models using data-dependent priors. Journal of the Royal Statistical Society B 62:159-180.

APPENDICES

A.1 Representing Classifications and Samples of Classifications in R, with a Timing Comparison

Any software library for generating and manipulating samples of classifications will require an approach for representing classifications, and samples of classifications, in computer code. We discuss two representations in R, of which perhaps the less obvious seems actually to provide better performance. R code demonstrates each approach, and provides a timing comparison. We discuss this topic in part because it provides background for use of any of our programs. Regarding notation, we consider a Monte Carlo sample of classifications of size n_{MC} , each of which divides *n* objects into *k* groups.

Perhaps the most obvious representation of a sample of classifications is a matrix that records a class index for each unit, in each classification. R functions that use a vector of cluster indices to represent a classification include the library function `cutree`, which extracts clusters from a dendrogram object generated using the `hclust` function. We use this approach for functions that return a single classification. Summarizing a matrix approach:

- A single classification is represented by a vector of length *n*, with *i*th value the class index (in 1, ... , *k*), for the *i*th unit.
- A sample of classifications is represented by an $n_{\mathrm{MC}} \times n$ matrix where each row represents a single classification.

However, for many operations involving a sample of classification we obtain much better performance using an approach based on the R list class:

- A single cluster is represented by a vector giving the unit indices (each in 1,..., *n*) of units that belong to the cluster.
- A single classification is represented by a list of clusters of length *k*.
- A sample of classifications is represented by a list of classifications.

This list representation is elegant for retrieving subsets of the data associated with particular clusters, and for computing various useful cluster analytic results, such as the co-clustering frequencies (consensus values of Monti et al., 2003).

The two approaches are demonstrated in the R program below. The R library function `system.time` is used to time the operations based on each approach. For each approach we simulate three operations: generation of classifications, retrieval of data for each cluster (in each classification), and retrieval of classifications from the sample. We generate classifications of the unit square using a Voronoi approach as implemented in the function `Tess.fn` (S. Prins), assuming *n* = 100 units and a minimum cluster count of 10 units. To simulate retrieval of data, we generate (just once) a matrix of dimension 100*5, simulating analysis of 5 variables observed for 100 units. A loop generating the sample of classifications is followed by a loop in which each classification is retrieved.

An experiment involving generation and manipulation of 100,000 classifications ran in about a minute with the list representation, but required over an hour with the matrix representation.

```
# Timing comparison for alternative representation of a sample of classifications
# * Classifications are generated using function 'Tess.fn' (S. Bates).  Assume loaded

nMC   <- 1e+5 # num. simulated classifications
n.obs <- 100  # num. simulated units
K     <- 5    # num. classes per classification
n.min <- 10   # min cluster size for Tess.fn
n.vars<- 5    # num. variables in simulated data
#-- Tesselation stuff:
sim.x <- runif(n.obs)   # simulate grid variables just once
sim.y <- runif(n.obs)
myTessi <-function() Tess.fn(
            k=K, gx=sim.x,gy=sim.y,
            gxlim=c(0,1),gylim=c(0,1),
            n.min=10,n.obs=n.obs)$z
#-- simulated data matrix (non-grid vars) with all values equal.
X <- matrix(rep(pi,n.obs*n.vars),n.obs,n.vars)
#-- function for translating classification from vector to list representation:
asClusterList <- function(class,N,K) { # reformat classification from vector to list
  outlist <- vector(K,mode="list")     # init output list to NA
  for (k in 1:K) outlist[[k]] <- (1:N)[class=k]  # indices for kth cluster
  return(outlist)
}
#-- list representation (time simulated generation and access)
cat("\nlist storage:")
parttn.sample<- NULL
t.list<-system.time(
{
  parttn.sample <- vector(nMC,mode="list")
  for(i in 1:nMC) {
    if(!(i %% 5000)) cat("\n",i)
    parttn.i <- asClusterList(myTessi(),n.obs,K)
    for(k in 1:K) x.ik <- X[parttn.i[[k]],]   # retrieve data
    parttn.sample[[i]]<- parttn.i             # store classification
  };cat("\n"); #for
  for(i in 1:nMC)   # simulate access of stored classifications
    parttn.i <- parttn.sample[[i]];  # sim. access
}#timed expression
         ) #system.time(..
#-- simulate matrix representation
parttn.sample<- NULL  # remove any large object from memory
cat("\nmatrix storage:")
t.mtrx <-system.time(
{
  #-- generation and retrieval of data
  parttn.sample <- matrix(NA,nMC,n.obs);  # init. tessln. sample.
  for(i in 1:nMC) {
    if(!(i %% 5000)) cat("\n",i);     # report progress to monitor
    parttn.i <- myTessi()            # generate classification
    for(k in 1:K) x.ik <- X[parttn.i==k,]; # retrieve data for each cluster
    parttn.sample[i,]<- parttn.i     # store classification
  }; cat("\n"); #for
  for(i in 1:nMC) parttn.i <- parttn.sample[i,]; # simulate accessing
} #timed expression
  ); # system time(..
cat("\nmatrix");print(t.mtrx)
cat("\nlist");print(t.list)
rm(parttn.sample)
```

A.2 Technical Results Related to Concurrence Fractions

Lemma 1. Expansion and Factorization of a Concurrence Matrix. For **Z** an incidence matrix representing some classification and **M** the concurrence matrix for the corresponding partition

$$\mathbf{M} = \sum_{l=1}^{k} \mathbf{z}_l \mathbf{z}_l^{\mathrm{T}} = \mathbf{Z}\mathbf{Z}^{\mathrm{T}},$$

where $\mathbf{z}_l$ denotes the *l*th column in $\mathbf{Z}$.

Proof. Regarding the first identity we observe that $\mathbf{z}_l \mathbf{z}_l^{\mathrm{T}}$ is a symmetric $n \times n$ matrix where the value in the *i*th row and *j*th column equals unity if units *i* and *j* both fall in cluster *l*, and otherwise equals zero. $\mathbf{M}$ is evidently the sum indicated by the first identity, given that two units can co-occur in at most one cluster. For the second identity we have

$$\begin{aligned}\sum_{l=1}^{k} \mathbf{z}_l \mathbf{z}_l^{\mathrm{T}} &= \sum_{l=1}^{k} \mathbf{Z}\mathbf{e}_l \left(\mathbf{Z}\mathbf{e}_l\right)^{\mathrm{T}} = \sum_{l=1}^{k} \mathbf{Z}\mathbf{e}_l\mathbf{e}_l^{\mathrm{T}}\mathbf{Z}^{\mathrm{T}} = \mathbf{Z}\left(\sum_{l=1}^{k} \mathbf{e}_l\mathbf{e}_l^{\mathrm{T}}\right)\mathbf{Z}^{\mathrm{T}} = \mathbf{Z}\mathbf{I}\mathbf{Z}^{\mathrm{T}} \\ &= \mathbf{Z}\mathbf{Z}^{\mathrm{T}}\end{aligned}$$

where $\mathbf{e}_l$ a vector of length *k* with all values 0, except for a 1 in the *l*th position.

The following statement of the method of principal coordinates follows Seber (1984).

Theorem 1. Principal Coordinates Analysis of a Matrix of Similarities. Let **C** denote a matrix of similarities among *n* units, with typical element $c_{ij} \in [0, 1]$, subject to $c_{ii} = 1 \; (i = 1, \ldots, n)$. Form the matrix $\mathbf{F} = \left(\mathbf{I} - \overline{\mathbf{J}}\right) \mathbf{C} \left(\mathbf{I} - \overline{\mathbf{J}}\right)$ where $\mathbf{I}$ is an identify matrix and $\overline{\mathbf{J}}$ is a square matrix with each value equal to $1 / n$. Form the matrix $\mathbf{G} = \left(\gamma_1^{1/2}\mathbf{v}_1 \vdots \ldots \vdots \gamma_p^{1/2}\mathbf{v}_p\right)$ where $\gamma_1, \ldots, \gamma_p$ are positive eigenvalues and $\mathbf{v}_1, \ldots, \mathbf{v}_p$ are corresponding eigenvectors in the spectral representation of **F**. If $\mathbf{F}$ is non-negative definite, the *i*th row of **G** gives coordinates for the *i*th unit $(i = 1, \ldots, n)$ in *n* dimensions, with the squared Euclidean distance between coordinates for the *i*th and *j*th units equaling $2\left(1-c_{ij}\right)$.

Corollary 1. Principal Coordinates Analysis of ECCP. When **ECCP**, the average of a sample average of concurrence matrices $\mathbf{M}_1^*, \ldots, \mathbf{M}_{n_{\mathrm{MC}}}^*$, is transformed for principal coordinates analysis as indicated in Theorem 1, the matrix that results is non-negative definite. Rows of the matrix $\left(\mathbf{I} - \overline{\mathbf{J}}\right) \mathbf{ECCP} \left(\mathbf{I} - \overline{\mathbf{J}}\right)$ give coordinates of corresponding units, separated by squared Euclidean distances $2\left(1-\mathrm{ECCP}_{ij}\right)$.

Proof. The theorem is an application of Theorem 1 if **ECCP** is shown to be non-negative definite. This holds because **ECCP** is the average of non-negative definite concurrence matrices $\mathbf{M}_1^*, \ldots, \mathbf{M}_{n_{\mathrm{MC}}}^*$ from a Monte Carlo sample of size n_{MC}. For any concurrence matrix **M** we have, for some incidence matrix $\mathbf{Z}$,

$$\left(\mathbf{I} - \overline{\mathbf{J}}\right) \mathbf{M} \left(\mathbf{I} - \overline{\mathbf{J}}\right) = \left(\mathbf{I} - \overline{\mathbf{J}}\right) \mathbf{Z} \left[\left(\mathbf{I} - \overline{\mathbf{J}}\right) \mathbf{Z}\right]^{\mathrm{T}}$$

A matrix of this form is non-negative definite (Harville, 1997, Corollary 14.2.14).