ELSEVIER

Computational Statistics & Data Analysis 31 (1999) 1–11

COMPUTATIONAL STATISTICS & DATA ANALYSIS

www.elsevier.com/locate/csda

Some computational issues in cluster analysis with no a priori metric ☆

Dan Coleman[a], Xioapeng Dong[b], Johanna Hardin[c], David M. Rocke[d,*], David L. Woodruff[e]

[a] *Hyseq Inc., Sunnyvale, USA*
[b] *Cadence Design Systems Inc., San Jose, USA*
[c] *Graduate Group in Statistics, University of California, Davis, USA*
[d] *Center for Image Processing and Integrated Computing, University of California, One Shields Avenue, Davis, CA 95616-8553, USA*
[e] *Graduate School of Management, University of California, Davis, USA*

Received 1 April 1998; received in revised form 1 March 1999

Abstract

Recent interest in data mining and knowledge discovery underscores the need for methods by which patterns can be discovered in data without any prior knowledge of their existence. In this paper, we explore computational methods of finding clusters of multivariate data points when there is no metric given a priori. We are given a sample, X, of n points in $\Re^p$ that come from g distinct multivariate normal populations with unknown parameters each of which contributes in excess of p points. Based on the assumption that we are given the number of groups, g, and a computational budget of T seconds of computer time, the paper reviews choices for cluster finding that have been described in the literature and introduces a new method that is a structured combination of two of them. We investigate these algorithms on some real data sets and describe simulation experiments. A principal conclusion is strong support for the contention that a two-stage algorithm based on a combinatorial search followed by the EM algorithm is the best way to find clusters.

Keywords: Cluster analysis; Data mining; EM algorithm; Mixture models

☆ Supported by the National Science Foundation (DMS 93-01344, DMS 95-10511, DMS 96-26843, ACI 96-19020, DMS 98-70172) and the National Institute of Environmental Health Science, National Institutes of Health (PHS P42 ES04699).

* Corresponding author.

0167-9473/99/$ - see front matter
PII: S 0167-9473(99)00009-2

1. Introduction

The basic problem of cluster analysis is to begin with a sample of n, p-dimensional points and then to classify the points into clusters purely from their location in p-dimensional space. Many current clustering methods assume that the appropriate distance function is known (for example, they may use Euclidean distance); but it is frequently more appropriate to use a distance measure that depends on the shape of the clusters. For example, if a cluster is multivariate normal with mean μ and covariance matrix Σ, the appropriate distance between a point x and the center μ of the cluster is the Mahalanobis distance $(x-\mu)'\Sigma^{-1}(x-\mu)$. The difficulty here is that the shape of the clusters is not known until the clusters have been identified, and the clusters cannot be effectively identified unless the shapes are known.

The majority of the extant literature on cluster analysis concerns methods which assume that a similarity measure or metric is known a priori – often the Euclidean metric is used. To sample this literature, see Lance and Williams (1967), Johnson (1967), Gower (1967), Mulvey and Crowder (1989), de Ammorim et al. (1992), Gersho and Gray (1992), Dorndorf and Pesch (1994). In some cases there may be sufficient prior knowledge about the problem at hand to allow the pre-specification of the metric, but the case in which this prior knowledge is unavailable is common, and may represent the overwhelming proportion of instances. Our unwillingness to rely on any a priori metric leads us to require that our estimators and algorithms be *affine invariance*, which means that affine transformations of the data $X \to AX + b$ will produce the same clusterings (A is nonsingular) as would have been produced using the same algorithm on the untransformed data. Essentially, affine invariance of clustering is equivalent to assuming that the metric is quadratic but otherwise unspecified.

When we refer to a *metric*, we mean a quadratic-form metric d of the form

$$d_S(x, y) = (x - y)'S^{-1}(x - y)$$

with S a positive-definite symmetric matrix. Such metrics comprise the class of metrics (distance functions) that are equivariant with respect to translations, rotations, and stretchings (affine transformations) under the assumption that the matrix S transforms with the data. Although there are other distance functions (the city-block metric, for example), the affine-equivariant metrics are particularly appropriate for use in clustering, since the results do not depend on irrelevant factors such as the units of measurements used or the orientation of the clusters in space.

These metrics can arise naturally in a number of ways, such as probability mixture models with multivariate normal or other elliptically symmetric distributions, or as intuitionally justified measures of quality of fit. For very high-dimensional data sets, metrics more complicated than the quadratic ones may not be feasible due to the curse of dimensionality. We will refer interchangeably to the matrix S or the distance function $d_S(x, y)$ as the metric.

Let us suppose that we are given a sample, $X = \{x_1, x_2, \ldots, x_n\}$, of n points in $\Re^p$ that come from g distinct populations with unknown parameters each of which contributes in excess of p points. We will assume that g is known. This is not

because we believe that the estimation of the number of clusters is necessarily easy (see e.g., Windham and Cutler, 1992; Banfield and Raftery, 1993), but because we are concentrating in this paper on the computational problems associated with finding the "best" g-cluster solution, for fixed g. It is an obviously necessary component of any successful overall clustering solution that the g-cluster sub-problems should be efficiently solvable.

2. Model-based clustering

Much of the work in affine equivariant clustering methods falls into what is often called model-based clustering (Banfield and Raftery, 1993). In model-based clustering, we judge a clustering by the likelihood under a particular model for example mixtures of multivariate normals. These multivariate normal clusters can be assumed all to have the same covariance matrix, or one may let the covariance matrices vary as well as the cluster means. A thorough examination of such criteria is given by Banfield and Raftery (1993). Their paper proposes a number of criteria that maximize the classification likelihood under various assumptions about the relative sizes and shapes of the clusters.

We use normal likelihood models not because we believe that clusters are necessarily normal, but because the criteria and methods produced by this model often have a natural and reasonable interpretation in terms of hypervolumes, quadratic distances, and others. Furthermore, although not all clusters are elliptical, the curse of dimensionality requires that we make some structural assumptions in order to be able to progress in even the most rudimentary way. Further work on clustering in which some of the data do not lie in elliptical clusters is an ongoing effort of our group.

2.1. Mixture likelihood

Within model-based clustering there are two main approaches: classification likelihood and mixture likelihood. Let τ_{ij} be a number indicating the cluster membership of point j in cluster i. In the mixture likelihood approach, we choose $\tau_{ij} \in [0,1]$ and the cluster parameters $\boldsymbol{\mu}_i$ and $\boldsymbol{\Sigma}_i$ to maximize the mixture likelihood

$$\mathscr{L}(\theta; \boldsymbol{X}) = \prod_{j=1}^{n} \left[\sum_{i=1}^{g} \pi_i \phi(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i; \boldsymbol{x}_j) \right],$$

where $\theta = (\pi_1, \ldots, \pi_g, \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1, \ldots, \boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g)$. The mixture frequencies are $\pi_i = n^{-1} \sum_{j=1}^{n} \tau_{ij}$, with $\sum_{i=1}^{g} \pi_i = 1$ and $\sum_{i=1}^{g} \tau_{ij} = 1$. The mean and covariance of the ith cluster are $\boldsymbol{\mu}_i$ and $\boldsymbol{\Sigma}_i$, and ϕ is the multivariate normal density.

In this formulation, we can interpret τ_{ij} as $\mathbb{P}(\boldsymbol{x}_j \in \text{Cluster}_i)$. The EM algorithm (Dempster et al., 1977; McLachlan and Basford, 1988), is the usual method of obtaining a solution to the mixture likelihood problem. It consists of an iterative re-weighting ascent from a given set of starting points,

$$\hat{\tau}_{ij}^{(0)} = \mathbb{P}(\boldsymbol{x}_j \in \text{Cluster}_i).$$

The EM algorithm terminates at a local maximum of the likelihood. Most commonly, the initial starting point is obtained from a random assignment of points to clusters, in which all the $\hat{\tau}_{ij}^{(0)}$ are therefore either 0 or 1.

If we have a computational budget of T seconds of computer time, and we have no prior information about the clusters, then a reasonable and commonly used choice is to repeatedly restart the algorithm from randomly generated allocations, terminating when T seconds have elapsed. Afterwards the clustering and parameter estimates corresponding to the largest local maximum are selected.

Typically there are many local maxima and the EM algorithm is very sensitive to the starting values given (see Everitt, 1993, Section 6.3.3; McLachlan and Basford, 1988, Section 3.2). The fraction of possible starting points that lead to the global maximum, or even to a local maximum corresponding to a "good" clustering, is often very small.

2.2. *Classification likelihood*

The classification likelihood is a restricted form of the mixture likelihood in which $\tau_{ij} \in \{0,1\}$, so that each point is uniquely assigned to exactly one cluster (i.e., $\sum_{i=1}^{g} \tau_{ij} = 1$). The initial random allocations used as starting points by many implementations of the EM algorithm are an example of such an assignment. Other assignments can be derived from combinatorial search techniques more sophisticated than random allocation.

Our proposal is to obtain starting values for the EM algorithm from combinatorial search algorithms which use the classification likelihood approach, assigning each point to exactly one cluster. Specifically, starting values for the EM algorithm are obtained from a clustering by setting $\hat{\tau}_{ij}^{(0)} = 1$ if point j has been assigned to cluster i and $\hat{\tau}_{ij}^{(0)} = 0$ otherwise.

Given a clustering, the data can be reordered as $\boldsymbol{x}_{ij}$ where i gives cluster membership and j gives observation number within the cluster. The likelihood conditional on a clustering is then

$$\mathscr{L}(\hat{\theta}; \boldsymbol{X}|\text{clustering}) = \prod_{i=1}^{g} \prod_{j=1}^{n_i} \phi(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i; \boldsymbol{x}_{ij}).$$

The determinant criterion is derived by maximizing the log of this likelihood with the assumption of homogeneous but unrestricted cluster variances

$$\mathscr{L}(\hat{\theta}; \boldsymbol{X}|\text{clustering}) = \text{constant} - \frac{n}{2}\log|\hat{\boldsymbol{\Sigma}}|, \tag{2.1}$$

where $\hat{\boldsymbol{\Sigma}}$ is the pooled covariance matrix across the g groups. Maximization of (2.1) is equivalent to minimization of

$$|\boldsymbol{W}| = \left|\sum_{i=1}^{g} \boldsymbol{W}_i\right|,$$

where

$$W_i = \sum_{j=1}^{n_i} (x_{ij} - \bar{x}_i)(x_{ij} - \bar{x}_i)' \quad \text{and} \quad \bar{x}_i = \frac{1}{n_i} \sum_{j=1}^{n_i} x_{ij}.$$

There are no known algorithms that can solve this problem to provable optimality in a reasonable amount of computational time (i.e. bounded by a polynomial function of the size of the data). An algorithm first proposed by Späth (1985), that we will refer to as FI (first improving), is an effective method of finding good solutions. In Section 5 we will explain the details of this algorithm and compare it with two others.

In Section 3 two data sets are studied in order to gain insight into the issues associated with cluster finding. In Section 4 we describe extensive simulation experiments that help to quantify the response of the algorithms to T, n, p, and g. The computer experiments were conducted on DEC Alpha 3000/700 workstations using implementations of EM written by McLachlan and Basford (1988), FI by Späth (1985), and *mclust* written by Fraley (1996); all other programs were written by the authors.

3. Examples: Iris data and Diabetes data

In this section two algorithms are compared: the EM algorithm started from a random clustering, EMRS and the EM algorithm preceded by FI which we call EMLS (for local search). Two real data sets are used: the Fisher Iris data due to Fisher (1936) and a data set concerning Chemical and Overt Diabetes due to Reaven and Miller (1979). These data sets are of similar size. However, using the true clustering, the Iris data form equally sized clusters with more-or-less homogeneous covariance matrices, while the Diabetes data form clusterings of different sizes with what appears to be a heterogeneous covariance structure. Our purpose in comparing the two algorithms using these two data sets is to highlight a few issues in cluster finding with no a priori metric and to motivate the comparative study of the algorithms using simulation studies.

We start with the Iris data set that consists of 150 observations and four variables: sepal length and width, and petal length and width. The observations consist of three clusters each of size 50 corresponding to three species of Iris: Setosa, Versicolor and Virginica. As is well known, the Setosa cluster is clearly separated from the other two, while the Virginica and Versicolor clusters overlap to some extent.

Runs of 100 ascents indicate that EMRS and EMLS work equally well when a homogeneous covariance structure is assumed. Each finds the clustering with likelihood 2199.925 in which all the observations of Cluster 1 are correctly classified, 2 observations of Cluster 2 are incorrectly classified as Cluster 3 and 1 observation in Cluster 3 is incorrectly classified as Cluster 2, this is summarized as $(50,0,0)$, $(0,48,2)$, $(0,1,49)$; the error rate is $3/150 = 2\%$.

The assumption of heterogeneous covariances introduces an additional 20 parameters, and the mixture likelihood is infinite if a cluster lies in a hyper-plane. This will always occur if a cluster contains less than p observations, but can also occur in other ways. In the Iris data, a subset of 29 observations from the Setosa cluster forms a three-dimensional subspace, all with the fourth variable, petal width, equal to 0.2. In order to yield finite likelihoods, we must ignore ascents where the likelihood diverges. From many starting points, the EM algorithm terminates with finite likelihood. The best likelihood using EMLS is 8223.998 and its classification is $(50,0,0)$, $(0,43,7)$, $(0,0,50)$ (error rate 4.7%). The largest finite likelihood found by the EMRS algorithm is 9809.822 corresponding to the classification, $(50,0,0)$, $(0,50,0)$, $(0,35,15)$ (error rate 10%). Note that the EMRS (random start) method finds a higher likelihood that corresponds to a poorer classification. This corresponds to a false peak in the likelihood that is avoided by EMLS, since it first searches for a plausible starting point (none of which is near the false peak).

The Diabetes data set has 145 observations and three variables: insulin area, glucose area and steady-state plasma glucose response (SSPG). The subjects were clinically classified as normal, chemical diabetes and overt diabetes, forming three clusters of sizes 76, 36 and 33, respectively. To study this data set, runs of 500 ascents were made. The clinical classifications were used to determine error rates.

Assuming homogeneous variances, the largest likelihood found by EMLS is 116.04, which corresponds to a classification $(74,2,0)$, $(19,17,0)$, $(10,2,21)$ with an error rate of 23%. The largest likelihood found by EMRS is 104.28, which is lower, but the classification $(64,12,0)$, $(6,30,0)$, $(5,7,21)$ has a comparable error rate of 20%.

Heterogeneity introduces 12 additional parameters. The algorithm EMLS finds the best likelihood at 361.26 corresponding to the classification $(58,7,11)$, $(0,24,12)$, $(0,3,30)$, with an error rate 22%. The best likelihood found by EMRS is 359.31, which is not as high but corresponds to the classification $(65,11,0)$, $(1,34,1)$, $(0,3,30)$ with an error rate of just 11%. This classification compares favorably to those considered in Table 6 of Banfield and Raftery. Classifications with lower likelihoods and lower error rates than the maximum likelihood solution can also be found by EMLS. This perplexing situation of a larger likelihood corresponding to a poor classification and a smaller likelihood corresponding to a good classification may be due to many factors other than the above-mentioned false peak including: the model may be over-parameterized or wrong, outliers may be present in the data, or the "true" classification may be wrong. Of course, real data sets are finite and are not generated by a process that is truly multivariate normal. There is no guarantee that the classification errors should be monotone in the likelihood.

Both EMRS and EMLS are very fast. They require less than 30 s for a run of 100 ascents on either data set. Augmenting random starts of the EM algorithm with those from a local search would appear to offer advantages for finding the maximum likelihood classifications. We defer quantification of this assessment to a large set of simulation experiments that are described in the next section.

4. Simulations

In order to explore the capabilities of the algorithms and responses to their parameters and the parameters of the data, we conducted extensive simulation experiments using pseudo-random data sets. In these data sets, each cluster is multivariate standard normal and has a mean that lies along a randomly selected diagonal away from the origin. The estimators compared here are all versions of the EM algorithm with two differences. The starting value for each trial is either a random partition of the data (EMRS) or is the result of a local combinatorial search (EMLS). The criterion used is either for homogeneous covariance matrices or for heterogeneous ones.

We measure the distance of the cluster means from the origin in terms of the unit of measurement $Q_p = \sqrt{\chi^2_{p;\,0.001}}$, which is more or less the radius of the sphere around the mean that contains almost all the points in the cluster. If the clusters are centered at a distance of at least $2Q_p$ from each other then these spheres should not overlap. We implement clusters at a distance of dQ_p from the origin by adding $\pm dQ_p^*$ to each component, where $Q_p^* = \sqrt{\chi^2_{p;\,0.001}/p}$. This generation mechanism is sufficient for use with affine-equivariant methods, such as the methods under consideration here. For non-affine-equivariant methods, the data should then be standardized as described in Rocke and Woodruff (1996, p. 1053).

The data sets were constructed by varying the factors g, p, n, and d. The number of clusters, g was 2, 3 or 5; the dimension p ranged over 2, 3, 5, and 10 (except for 5 clusters, where the dimension 2 and 3 cases were omitted). The three values for n were roughly 5, 10 and 20 times p times g, which we refer to as small, medium, and large; the increase in the number of points with the dimension was supposed to reduce the comparative disparity between dimensions. For this set of experiments, d was set at 2.0, which provides adequate separation between the clusters. For each combination of parameters, 20 data sets were randomly generated. Each of the four algorithms was executed for three different values of T (corresponding to 2, 5, and 10 random restarts of the underlying ascent or descent algorithm). This large design enables a reasonable characterization of performance of the algorithms.

We score the results on a simple scale: the classification is either (completely) correct or not. With widely separated clusters, an iterative algorithm is unlikely to be nearly correct, because it would then after some iterations become entirely correct. Complete correctness is not a reasonable criterion for overlapping clusters, but overlapping clusters is a problem beyond the scope of the present paper. Thus, in any cell of Tables 1–4, an entry is the percent of cases for which the algorithm returned an incorrect ("Bad") classification.

Table 1 shows the dependence of the overall results on the estimator. We give results separately by the number of clusters, since design choices may make the different cases not completely comparable. The main conclusion to be drawn here is that the local search methods are far more effective than the random start methods. As an aside, the local search methods also took considerably less time (due to reduced EM iteration cost).

Table 1
Percent "Bad" classifications by estimator and by number of clusters

	EMRS		EMLS	
Number of clusters (g)	Heterogeneous	Homogeneous	Heterogeneous	Homogeneous
2	40.3	56.4	12.6	12.9
3	27.9	23.9	0.7	0.1
5	30.2	17.4	2.5	3.4

Table 2
Percent "Bad" classifications by dimension and by number of clusters

Number of clusters	$p=2$	$p=3$	$p=5$	$p=10$
2	12.2	19.2	31.9	58.9
3	3.3	7.9	15.1	26.3
5	–	–	6.7	20.5

Table 3
Percent "Bad" classifications by sample size and by number of clusters

Number of clusters	Small n	Medium n	Large n
2	29.0	29.9	32.8
3	16.6	9.6	14.0
5	16.0	9.4	14.9

Table 4
Percent "Bad" classifications by number of iterations and by number of clusters

Number of clusters	2 iterations	5 iterations	10 iterations
2	45.2	28.3	18.1
3	20.3	11.4	7.8
5	19.7	12.2	8.3

Table 2 shows that increasing the dimension makes the problem more difficult, uniformly across estimators. The relatively poor classification results for high dimension are perhaps unsurprising, but this decrease in performance as the dimension rises occurs even with a proportionate increase in the sample size, since n is proportional to gp.

Table 3 shows that the overall results are insensitive to the sample size. Most likely, this is due to two factors that work in opposite directions. More data mean better estimates if a good estimate can be found, but more data also increase the computational complexity of the problem.

Table 4 shows that more trials produce better results. This conclusion (like the results in Tables 2 and 3) remains true for each estimator individually as well as

in the aggregate. With 10 iterations, the most effective estimator EMLS obtained the correct solution about 98% of the time.

5. Comparing local search and hierarchical agglomeration

Given that it seems important to precede the EM algorithm with a combinatorial search for a good starting point, the logical next question is which classes of combinatorial search algorithms are most effective. In this section three combinatorial algorithms for maximizing the determinant criterion are compared. The first two methods are local searches in a single exchange neighborhood, first improvement (FI) and steepest ascent (SA). The third method is hierarchical agglomeration (HA).

Local searches start with a random partition (clustering) of the data into g clusters with the restriction, maintained throughout the algorithm, that each cluster have at least h observations, $h > p$. At each stage a new partition is selected from the *neighborhood* of the current partition. The single exchange neighborhood of a partition is all partitions that can be obtained by moving one observation from one cluster to another. Hence, the number of neighbors under the single exchange neighborhood consists of $n(g-1)$ partitions (or fewer if one or more clusters contains h members).

When employing local search in a deterministic way, one must decide how to explore a neighborhood. Two extremes are 'steepest ascent', which we have called SA and 'first improving', which we have called FI. In the SA algorithm, the objective function is evaluated for all partitions in the neighborhood and the partition that yields the greatest increase in the objective function becomes the new incumbent partition. The algorithm stops when there are no members of the neighborhood of the current solution that have higher likelihood. In the case of FI, the objective function is evaluated one observation at a time. At the beginning of the algorithm a random ordering of the data is made. The *current* observation at stage k is the $\mathrm{mod}(k-1,n)+1$ observation in this ordering. All moves of the current observation are evaluated, and such a move is executed if it improves the objective function, without checking to see if some other point move would be better. The current observation then changes to the next point. The algorithm stops when no moves are made in one entire pass of the data, which is the same stopping criterion as for SA.

FI and SA have the same neighborhood structure so they have the same local maxima. If they are started on a local maximum they both stop. However, since they employ different algorithms to select moves, in general they may terminate at different local maxima given the same starting point.

Which of these two algorithms is 'better' depends on the speed at which the objective function can be updated and the preponderance of local maxima in the objective function, which is data set dependent. Comparisons between first improving and steepest ascent have been conducted on a number of different optimization problems from problem domains other than statistics (see, e.g. Anderson, 1996), and it seems that the preferred method depends strongly on the problem and weakly on

Table 5
Comparison of search algorithms applied to the Iris data

	FI	SA	HA
Avg time for ascent	0.0262	0.1853	63.6617
Std error of time for ascent	0.0120	0.0303	0.3364
Success probability per ascent	0.1460	0.1060	0.0

the data. As evidence below indicates, the algorithm FI seems to be preferred for the problem of classification to maximize the determinant criterion.

An alternative to the local search algorithms that we have described is hierarchical agglomeration (HA), based on an algorithm attributed to Ward (1963) and currently implemented in Splus as *mclust* (see Fraley, 1996). In the language of local search, HA is based on a *constructive neighborhood*, meaning that full solutions are built up from partial solutions. In HA, the observations are partitioned into a large number of small clusters. Pairs of the clusters are then merged so as to minimize the increase in the objective function (determinant) for the resulting combination. If all the data are considered as separate clusters, then at the first stage $n(n-1)/2$ objective function evaluations are required, and the entire algorithm requires $(n^3-n)/6$ objective function updates. In implementing the determinant criterion, the objective function is zero until a cluster of $p+1$ observations is obtained, so a surrogate criterion for the first stages must be used. In *Splus* the trace and the trace+determinant are used. As an aside, we note that use of the trace renders the algorithm not strictly affine equivariant.

For each algorithm five hundred ascents are made. An ascent of the algorithm is a random starting point followed to a local maximum. A starting point for FI and SA is a partition, for HA as implemented in *mclust* it is a reordering of the data.

The largest conditional log likelihood found in the 1500 ascents of the Iris data set is 757.57. An ascent is deemed a success if this value is obtained. This likelihood corresponds to the partition found by Friedman and Rubin (1967) and has three observations incorrectly classified. Summary statistics for the Iris data are given in Table 5. None of the ascents of HA yielded the maximum, the highest likelihood found was 754.2. The results for the Diabetes data set are similar. The algorithm FI is about 6.75 times faster per ascent than SA in both data sets. If FI is run 6 times (which is still faster than running SA once), the success rate is $1-(1-0.1460)^6=0.61$ in the Iris data set making it far superior to SA, which has a success rate of 0.1060. Otherwise stated, it requires an estimated 30 iterations of FI to insure a 99% chance of success. SA requires 42 tries to insure the same success rate, which will take over 9 times as long to run. Results for the Glucose data and for simulated data sets were qualitatively the same. We conclude that FI seems to be preferred for this problem.

We obtained some additional insight into the behavior of the *mclust* implementation of hierarchical agglomeration by examining the classifications produced by HA

when run on random permutations of the Iris data. Ideally, the possible clusterings obtained should not depend on the order of the data in the data file, but in this case there was a strong dependency. For each random ordering, several classifications occurred, but a different random ordering would have a different possible set of classifications, possibly with some overlap. Many of the classifications had a large number of errors. Use of the iterative relocations routine *mreloc* after hierarchical agglomeration did not improve the situation.

6. Conclusions and directions for further research

We have described a computational investigation of methods for a very important problem in multivariate statistics: finding clusters in the absence of an a priori metric. Our most important finding is that use of combinatorial local search to produce starting points for the EM algorithm is much superior in every respect to the use of random partitions of the data as starting points. Both methods seem to produce better classifications with a fixed number of clusters than hierarchical agglomeration does.

References

de Ammorim, S.G., Barthelemy, J-P., Ribeiro, C.C., 1992. Clustering and clique partitioning: simulated annealing and tabu search approaches. J. Classification 9, 17–41.

Anderson, E.J., 1996. Mechanisms for local search. European J. Oper. Res. 88, 139–151.

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

Dorndorf, U., Pesch, E., 1994. Fast clustering algorithms. ORSA J. Comput. 6, 141–153.

Everitt, B.S., 1993. Cluster Analysis. Hodder & Stoughton, London.

Fisher, R.A., 1936. The use of multiple measurements in taxonomic problems. Ann. of Eugenics 7, Part II, 179–188.

Gersho, A., Gray, R.M., 1992. Vector Quantization and Signal Compression. Kluwer Academic Publishers, Dordrecht.

McLachlan, G.J., Basford, K.E., 1988. Mixture Models: Inference and Applications to Clustering. Marcel Dekker, New York.

Mulvey, J.M., Crowder, H.P., 1989. Cluster analysis: an application of Lagrangian relaxation. Management Sci. 25, 329–340.

Rocke, D.M., Woodruff, D.L., 1996. Identification of outliers in multivariate data. J. Amer. Statist. Assoc. 91, 1047–1061.

Ward, J.H., 1963. Hierarchical groupings to optimize and objective function. J. Amer. Statist. Soc. 58, 236–244.

Windham, M.P., Cutler, A., 1992. Information ratios for validating mixture analyses. J. Amer. Statist. Assoc. 87, 1188–1192.