Comput Stat (2008) 23:643–659
DOI 10.1007/s00180-007-0103-7

ORIGINAL PAPER

A bootstrap-based aggregate classifier for model-based clustering

José G. Dias · Jeroen K. Vermunt

Accepted: 19 November 2007 / Published online: 19 February 2008

Abstract In model-based clustering, a situation in which true class labels are unknown and that is therefore also referred to as unsupervised learning, observations are typically classified by the Bayes modal rule. In this study, we assess whether alternative classifiers from the classification or supervised-learning literature—developed for situations in which class labels are known—can improve the Bayes rule. More specifically, we investigate the performance of bootstrap-based aggregate (bagging) rules after adapting these to the model-based clustering context. It is argued that specific issues, such as the label-switching problem, have to be carefully addressed when using bootstrap methods in model-based clustering. Our two Monte Carlo studies show that classification based on the Bayes rule is rather stable and difficult to improve by bootstrap-based aggregate rules, even for sparse data. An empirical example illustrates the various approaches described in this paper.

Keywords Classification rules · Classifiers · Bootstrap estimation · Model-based clustering · Latent class model

J. G. Dias (✉)
Department of Quantitative Methods, ISCTE,
Higher Institute of Social Sciences and Business Studies and UNIDE,
Edifício ISCTE, Av. das Forças Armadas,
1649-026 Lisboa, Portugal
e-mail: jose.dias@iscte.pt

J. K. Vermunt
Department of Methodology and Statistics, Tilburg University,
P.O. Box 90153, 5000 LE Tilburg, The Netherlands
e-mail: j.k.vermunt@uvt.nl

1 Introduction

Model-based clustering by finite mixture (FM) models is formulated as follows (McLachlan and Peel 2000). Let $\mathbf{y}$ denote a J-dimensional observation and $D = \{\mathbf{y}_1, \ldots, \mathbf{y}_n\}$ a sample of size n. Each data point is assumed to be a realization of the random variable $\mathbf{Y}$ with S-component mixture probability function

$$f(\mathbf{y}_i; \boldsymbol{\varphi}) = \sum_{s=1}^{S} \pi_s f_s(\mathbf{y}_i; \boldsymbol{\theta}_s), \tag{1}$$

where π_s are positive mixing proportions—also referred to as prior class membership probabilities—that sum to one, $\boldsymbol{\theta}_s$ are the parameters of the conditional distribution of component s defined by $f_s(\mathbf{y}_i; \boldsymbol{\theta}_s)$, and $\boldsymbol{\varphi} = \{\pi_1, \ldots, \pi_{S-1}, \ \boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_S\}$ is the vector of unknown parameters.

Assuming i.i.d. observations, the log-likelihood function corresponding to an FM model is obtained as $\ell(\boldsymbol{\varphi}; \mathbf{y}) = \sum_{i=1}^{n} \log f(\mathbf{y}_i; \boldsymbol{\varphi})$. For the most common forms of $f_s(\mathbf{y}_i; \boldsymbol{\theta}_s)$, it is straightforward to maximize the log-likelihood function by means of the EM algorithm (Dempster et al. 1977).

From the prior class membership probabilities and the component-specific densities defining the FM model of interest, one can easily derive the posterior probability that a particular observation was generated by a given component or cluster. Using Bayes' theorem we obtain the a posteriori probability that observation i belongs to class s as follows:

$$\alpha_{is} = \frac{\pi_s f_s(\mathbf{y}_i; \boldsymbol{\theta}_s)}{\sum_{v=1}^{S} \pi_v f_v(\mathbf{y}_i; \boldsymbol{\theta}_v)}. \tag{2}$$

The maximum likelihood (ML) estimates of the posterior probabilities—denoted as $\hat{\alpha}_{is}$—play an important role in the classification of cases into clusters. Magidson and Vermunt (2001) showed that these quantities can not only be used for classification purposes, but also for profiling the clusters by comparing (and plotting) the average value of $\hat{\alpha}_{is}$ across subgroups defined by covariate categories.

Whereas the $\hat{\alpha}_{is}$ define a soft partitioning/clustering of the data set at hand, an additional step is needed to transform this soft partition into a hard partition. Let c_i represent the true class label of observation i, which in clustering applications is unknown and that can, therefore, be consider to be missing data. An alternative representation of the class membership of case i is by a set of S indicator variables $z_{is} = I(c_i = s)$, where $I(.)$ stands for the indicator function—$I(A) = 1$, if A is true and zero otherwise—i.e., $z_{is} = 1$, if $c_i = s$ and 0 otherwise. The Bayes modal rule assigns observation i to the class with maximum a posteriori probability. At the ML estimate, this yields the following classification rule:

$$\hat{c}_i = \arg\max_{s} \hat{\alpha}_{is}, \quad i = 1, \ldots, n, \tag{3}$$

which is equivalent to

$$\hat{z}_{is} = I\left(\max_{v} \hat{\alpha}_{iv} = \hat{\alpha}_{is}\right), \quad s, v = 1, \ldots, S, \; i = 1, \ldots, n. \tag{4}$$

Whereas the performance of the Bayes modal rule has been extensively studied for supervised learning applications—i.e., classification tasks based on known class labels c_i—little is known on its performance in model-based clustering applications. It is not clear at all whether the rule is optimal in all circumstances nor whether there is room for improvements in model-based clustering classification by means of so-called aggregation methods. In this study we, therefore, not only assess the quality of the simple Bayes modal rule, but also adapt the bootstrap-based aggregate rule called bagging to the unsupervised learning situation.

The remaining of this paper is organized as follows. In the two sections, we discuss the Bayes rule and introduce an aggregate classification rule for model-based clustering. Then, the implementation of this aggregate rule using the bootstrap method is discussed, where special attention is given to the label-switching problem. Subsequently, two Monte Carlo studies are presented. Whereas the first study compares the performance of the proposed classification rules under a set of conditions, the second study focuses on the effect of sparseness. Then, an empirical example is presented. The paper ends with some final remarks.

2 The Bayes' classifier

2.1 Known class labels

As mentioned earlier, in classification or supervised-learning problems, the class labels c_i are known. The purpose of the analysis it to construct a classifier that can be used for classifying new cases; that is, for assigning observations that were not used in the analysis to one of the classes. The best-known classification method is discriminant analysis, which yields as output a set of discriminant functions. By assigning observations to the class for which the discriminant function is largest, one minimizes the probability of incorrect classification (McLachlan 1992; Duda et al. 2001).

Let $\mathcal{C} = \{1, \ldots, S\}$ denote the set of class labels used in a classification task; $c_i \in \mathcal{C}$ indicates the class membership of observation i; that is, $c_i = s$ means that observation i is generated by or belongs to class s. Let $D_c = \{(c_i, \mathbf{y}_i), i = 1, \ldots, n\}$ be a sample of n independent observations consisting of the class labels c_i and the "explanatory variables" $\mathbf{y}_i$. A classification method uses the data set D_c to construct a function $\hat{\mu}(\mathbf{y}; D_c)$ that yields a prediction $E[C|\mathbf{Y} = \mathbf{y}; D_c]$ for new observations, i.e., based on D_c one defines a rule to predict the value of c_{n+1} given $\mathbf{y}_{n+1} \in \mathcal{R}^J$.

The classifier output is commonly defined as an S-dimensional vector $(\hat{\mu}_1(\mathbf{y}; D_c), \ldots, \hat{\mu}_S(\mathbf{y}; D_c))$, where $\hat{\mu}_s(\mathbf{y}; D_c)$ can be interpreted as the degree of support given by classifier $\hat{\mu}$ to the hypothesis that $\mathbf{y}$ comes from class s, $s = 1, \ldots, S$. Without loss of generality, one can transform $\hat{\mu}_s(\mathbf{y}; D_c)$ to lie within the interval $[0, 1]$ and to sum to one, yielding what is sometimes referred to as "soft labels". In some applications, "crisp" class labels are required, i.e., $\hat{\mu}_s(\mathbf{y}; D_c) \in \{0, 1\}$, with

$\sum_{s=1}^{S} \hat{\mu}_s(\mathbf{y}; D_c) = 1$. These are typically obtained by "hardening" the soft labels by assigning 1 to the largest value (the winning class label), and 0 to the remaining ones, yielding a hardening method referred to as the Bayes modal rule. Its underlying idea is supported by decision theory as follows. Let s' and s indicate the predicted classification (the decision) and the true state of nature of observation i, respectively. Then, the decision is correct if $s' = s$ and in error otherwise. The loss function of interest is the so-called zero-one loss function, which is defined as follows:

$$L(c_i = s' | c_i = s) = \begin{cases} 0, & s' = s \\ 1, & s' \neq s \end{cases} \tag{5}$$

for $s', s = 1, \ldots, S$. The conditional risk associated with this loss function is (Duda et al. 2001, p. 27)

$$R(c_i = s' | \mathbf{y}_i, D_c) = 1 - p(c_i = s | \mathbf{y}_i, D_c). \tag{6}$$

Therefore, under zero-one loss, the misclassification risk is minimized if and only if observation i is assigned to the component s for which $p(c_i = s | \mathbf{y}_i; D_c)$ is largest (McLachlan 1992); that is,

$$\hat{c}_i = \arg\max_{s} p(c_i = s | \mathbf{y}_i; D_c), \tag{7}$$

which defines the Bayes classification rule or Bayes classifier.

Because in classification problems such as discriminant analysis the class labels are known, the correct allocation rates can be directly computed by comparing predicted with observed class labels. Using the definition of $\hat{z}_{is}$ from Eq. (4), the correct allocation rate for class s (A_s) and the overall correct allocation rate (A) can be obtained as follows:

$$A_s = \frac{1}{n_s} \sum_{i=1}^{n} z_{is} I(z_{is} = \hat{z}_{is}), \tag{8}$$

$$A = \frac{1}{n} \sum_{s=1}^{S} n_s A_s, \tag{9}$$

where $n_s = \sum_{i=1}^{n} z_{is}$, and $n = \sum_{s=1}^{S} n_s$. Note that A is simply a weighted sum of the S class-specific A_s values.

2.2 Unknown class labels

Contrary to the classification framework described above, in the clustering (or unsupervised learning) framework we would like to focus on, the class labels c_i are unobserved (latent or missing). For simplicity of exposition, we restrict ourselves to the situation in which the number of labels/classes is known, which means that S is treated as fixed.

In clustering problems the observed data is $D = \{\mathbf{y}_i, i = 1, \ldots, n\}$. The FM model with S components in Eq. (1) defines the marginal distribution of the observed data; that is, the distribution after integrating out (summing over) the missing data. The hypothetical complete data may be referred to as $\{(\mathbf{z}_i, \mathbf{y}_i), i = 1, \ldots, n\}$, where the indicators variables $\mathbf{z}_i = (z_{i1}, \ldots, z_{iS})$ are assumed to come from a multinomial distribution defined by the component proportions $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_S)$, $\mathbf{z}_i \sim \mathcal{M}(1; \boldsymbol{\pi})$. The expected value of missing data $\mathbf{z}_i$ conditional on the observed data and the unknown model parameters $\boldsymbol{\varphi}$ is given by the Bayes' theorem in Eq. (2). In practice one will plug in the ML estimates $\hat{\boldsymbol{\varphi}}$ to obtain estimates for α_{is}. Equivalently to the classification setting, in the clustering setting the Bayes modal rule assigns observation i to the class with maximum a posteriori probability (see Eqs. 3 and 4).

For clustering problems, the class-specific and overall accuracy of the classification cannot be directly computed using Eqs. (8) and (9) because the z_{is} appearing in these equations are unobserved; that is, the correct allocation rates depend on the unknowns $\mathbf{z}_i$ that have to be estimated. A natural alternative for A_s and A is

$$\hat{A}_s = \frac{1}{n\hat{\pi}_s} \sum_{i=1}^{n} \hat{z}_{is}\hat{\alpha}_{is}, \tag{10}$$

$$\hat{A} = \frac{1}{n} \sum_{i=1}^{n} \max_s \hat{\alpha}_{is}, \tag{11}$$

which, as shown by Basford and McLachlan (1985), are consistent but biased estimators of A_s and A, respectively. To remove the bias in these estimators, these authors proposed using bootstrapping techniques.

3 Aggregate classifiers

3.1 Known class labels

An aggregate classifier or aggregate classification rule μ_A is given by the general expression

$$\mu_A(\mathbf{y}) = E_F[\hat{\mu}(\mathbf{y}, D_c)], \tag{12}$$

where the expectation is over samples D_c distributed according to F. The underlying idea of using an aggregate classifier is that one wishes to reduce the effect of the specific data set that is used to build the classifier. Theoretically, this is achieved by integrating over—or aggregating over—all possible data sets that can be generated from F, the true population distribution of D_c.

Although in practice one typically has no more than a single data set from F at hand to construct a classifier, it turns out to be possible to mimic the process underlying the aggregate classifier by means of the bootstrap method, a rather general computer intensive resampling technique introduced by Efron (1979). The bootstrap can be used, among other things, to determine standard errors, biases, and confidence intervals of model parameters in situations in which theoretical statistics are difficult to obtain.

Breiman (1996a) was the first who proposed using the bootstrap method as a tool for building more stable classifiers.

The bootstrap technique is easily stated. Suppose we have a random sample D from an unknown probability distribution F and we wish to estimate the unknown parameter $\boldsymbol{\varphi} = t(F)$. Let $S(D, F)$ be a statistic. Whereas standard statistical inference assumes that the underlying sampling distribution of $S(D, F)$ is known, in the bootstrap method, F is estimated by $\hat{F}$ based on D. More specifically, the bootstrap provides an approximation of the sampling distribution based on $S(D^*, \hat{F})$, where the bootstrap sample $D^* = \left\{\mathbf{y}_1^*, \mathbf{y}_2^*, \ldots, \mathbf{y}_n^*\right\}$ is a random sample of size n drawn from $\hat{F}$, and $\hat{\boldsymbol{\varphi}}^* = S(D^*, \hat{F})$ is a bootstrap replication of $\hat{\boldsymbol{\varphi}}$. In fact, the bootstrap performs a Monte Carlo evaluation of the properties of $\hat{\boldsymbol{\varphi}}$ using repeated sampling, say B times, from $\hat{F}$ to approximate the sampling distribution of $\hat{\boldsymbol{\varphi}}$. The B samples are obtained using the following iterative scheme:

1. Draw a bootstrap sample $D^{(*b)} = \{\mathbf{y}_i^{(*b)}\}$, $i = 1, \ldots, n$, with $\mathbf{y}_i^{(*b)} \sim \hat{F}$;
2. Estimate $\hat{\boldsymbol{\varphi}}^{(*b)} = S(D^{(*b)}, \hat{F})$.

Breiman (1996a) introduced what he called the bagging (bootstrap aggregating) procedure, in which the aggregate rule $\mu_A(\mathbf{y})$ is obtained by generating B of D_c using the bootstrap method. The aggregate classifier is defined as follows:

$$\hat{\mu}_A(\mathbf{y}) = E_{\hat{F}}[\hat{\mu}(\mathbf{y}, D_c^*)], \tag{13}$$

where D_c^* is a bootstrap sample from the empirical distribution function; that is, a sample that is obtained by sampling with replacement from the observed data set. The bootstrap expectation is obtained by Monte Carlo integration: for every bootstrap resample from $\hat{F}$, one computes $\hat{\mu}^{(*b)}(\mathbf{y}; D_c^{(*b)})$, $b = 1, \ldots, B$ and subsequently approximates Eq. (13) by the ergodic mean $\hat{\mu}_A^*(\mathbf{y}) \approx B^{-1} \sum_{b=1}^{B} \hat{\mu}^{(*b)}(\mathbf{y}; D_c^{(*b)})$.

The aggregate classifier $\hat{\mu}_A$ will usually perform better than the original classifier $\hat{\mu}(\mathbf{y}, D_c)$ based on a single observed data because its variance is either equal or smaller than the variance of $\hat{\mu}(\mathbf{y}, D_c)$, where the expected variance reduction is larger when the original predictor is more "unstable". Because both procedures have a similar bias, the mean square error of the bagged estimator will be lower, particularly for unstable predictors (Breiman 1996b; Bauer and Kohavi 1999). Heuristically, a predictor is said to be unstable if small changes in the data D_c can cause large changes in the classifier (Breiman 1996b). It has been shown that certain regression and classification methods (e.g., regression trees) prediction/classification can be very unstable. For a more rigorous treatment of the instability issue and for theoretical results explaining the improved performance of aggregate classifiers, we refer to Bühlmann and Yu (2002).

Various refinements of the bagging procedure have been proposed in the classification literature, such as boosting and arcing—two procedures that adaptively change the weights of the training patterns based on their performance at previous iterations (Freud and Schapire 1996; Breiman 1998)—and double-bagging—a procedure that uses information of observations not included in a given bootstrap replication

(Hothorn and Lausen 2003). Though a comparison by Bauer and Kohavi (1999) showed that these refinements may improve classification performance, here will use only the more standard implementation of bagging because the other procedures cannot easily be adapted to the situation in which class labels are unknown.

3.2 Unknown class labels

In the clustering setting, an aggregate classifier generalizes the Bayes rule described in Eq. (3) in the sense that it classifies observations taking into account the aggregate results, i.e., it classifies $\mathbf{y}_i$ into components by majority vote over the class labels produced from the B bootstrap resamples. This concept of voting is borrowed from the bagging procedure for classification problems which was described above.

Let $\hat{z}_{is}^{(*b)}$ be the classification of observation i based on bootstrap replicate b, $b = 1, \ldots, B$, which are obtained by plugging in $\hat{\alpha}_{is}^{(*b)}$, the bth sample estimates of the posterior probabilities, in Eq. (4). By majority vote, observation i ($i = 1, \ldots, n$) is assigned to the class with the maximum number of assignments across the bootstrap resamples

$$\hat{c}_i^{(*)} = \arg\max_s \sum_{b=1}^{B} \hat{z}_{is}^{(*b)}. \qquad (14)$$

A question of interest is whether classification based by the Bayes rule—yielding the $\hat{c}_i$ of Eq. (3)—is always the most adequate or whether it can be improved by the model-based aggregate classifier described in Eq. (14). In other words, is the Bayes rule a stable classifier in the clustering setting or is there room for improvement? This question is addressed below by means of a simulation study.

4 Implementation of the bootstrap in FM modeling

This section discusses various issues that are of interest when implementing bootstrap methods. Two general issues are choices regarding the number of replications and the type of bootstrap, and two issues which are specific for FM models are the problems of local maxima and label switching.

4.1 Number of bootstrap samples

Efron and Tibshirani (1993, p. 13) suggested using a B value between 50 and 200 when the bootstrap is used for the computation of standard errors. For confidence intervals, on the other hand, a much larger B value of at least 1000 is required (Efron 1987). In the boosting literature, the advice is generally to use a $B = 50$ (Breiman 1996a). In our study, we compared results for B equal to 21, 51, and 101.

4.2 Parametric versus nonparametric bootstrap

There are two types of bootstrap procedures that differ in the way F is approximated. The parametric bootstrap assumes a parametric form for F and estimates the unknown parameters by their sample quantities ($\hat{F}_{\text{par}}$). That is, one draws B samples of size n from the parametric estimate of the function F—the function defined by the ML estimates of the unknown model parameters. In the nonparametric bootstrap procedure, the approximation of F ($\hat{F}_{\text{nonpar}}$) is obtained by its nonparametric maximum likelihood estimate; that is, by the empirical distribution function which puts equal mass $1/n$ at each observation. In that procedure, sampling from $\hat{F}$ means sampling with replacement from the data D. While the nonparametric approach is the one that is typically used in bagging rules, here both bootstrap procedures are investigated since we would like to know whether the parametric bootstrap is a better choice in certain situations.

4.3 Starting values

Estimating of the parameters of the FM model for each bootstrap replication b ($\hat{\boldsymbol{\varphi}}^{(*b)}$) requires the use of an iterative algorithm. The EM algorithm is an elegant alternative, but its success in converging to the global maximum depends among others on the quality of the starting values. Because the original sample D and the replicated sample $D^{(*b)}$ will usually not differ very much, McLachlan and Peel (2000) suggested using the ML estimate of $\boldsymbol{\varphi}$ from D as the starting value for the bootstrap runs. For a latent class (LC) model, Dias (2005) showed that this strategy performs well in comparison with starting the EM algorithm 10 times with random values for the parameters $\boldsymbol{\varphi}$. Therefore, in our analysis, within the bootstrap procedure, the EM algorithm was started from the ML estimates for sample D.

4.4 Label-switching problem

The likelihood function of a FM model is invariant under permutations of the S components, i.e., any rearrangement of the component indices yields the same log-likelihood value. Typical for the bootstrap is that permutations of the components may occur across replications, resulting in a distortion of the distribution of interest. This problem, which is well-known in the Bayesian analysis of mixture models by Markov chain Monte Carlo (MCMC) techniques, is usually referred to as the label-switching problem.

One way to deal with this problem is to impose inequality constraints on a particular set of model parameters, for example, that $\pi_1^{(*b)} < \pi_2^{(*b)} < \cdots < \pi_S^{(*b)}$, $b = 1, \ldots, B$ (Richardson and Green 1997). However, for Bayesian estimation of mixture models it has been shown that such a simple strategy can seriously distort the results, especially when the true class sizes are similar to one another (Stephens 1997; Celeux et al. 2000). Because similar problems are very likely to occur in the bootstrap, it is a better option to use a method proposed by Stephens (2000) which determines the "right" order of the clusters by inspecting the posterior probabilities $\alpha_{is}^{(*b)}$ defined in Eq. (2). This method has been shown to outperform other methods in the Bayesian MCMC context (Dias and Wedel 2004).

Let $\upsilon_{(*b)}(\boldsymbol{\varphi}^{(*b)})$ define a permutation of the parameters for the bth bootstrap sample and $\mathbf{Q}^{(b-1)} = (q_{is}^{(b-1)})$ be the bootstrap estimate of $\alpha = (\alpha_{is})$, based on the previous $b-1$ bootstrap samples. Stephen's algorithm for reordering the clusters is initialized with a small number of runs, say B^*: $\mathbf{Q}^{(0)} = \left(\frac{1}{B^*}\sum_{v=1}^{B^*}\hat{\alpha}_{is}^{(v)}\right)$. Then, for the bth bootstrap sample, choose $\upsilon_{(*b)}$ that minimizes the Kullback-Leibler divergence between the posterior probabilities $\hat{\alpha}_{is}\left\{\upsilon_{(*b)}(\hat{\boldsymbol{\varphi}}^{(*b)})\right\}$ and the estimate of the posterior probabilities $\mathbf{Q}^{(b-1)}$, and subsequently compute $\mathbf{Q}^{(b)}$.

A difference between an MCMC run and a bootstrap is that in the latter the label is likely to occur at every resample. An initial estimate using a small number of B^* bootstrap estimates (without taking into account label switching) is, therefore, not appropriate and a better solution seems to be to take $\mathbf{Q}^{(0)}$ as the ML solution as the starting configuration. An even simpler alternative is a non-adaptive procedure with $\mathbf{Q}^{(b-1)}= \mathbf{Q}^{(0)}$, i.e., which implies relabeling each bootstrap sample according to its distance to the ML solution. We, however, found examples where this procedure fails in removing multimodality and do, therefore, recommend not to use it.

5 Model Carlo studies

Below, we first introduce the specific FM model that was used in our two Monte-Carlo studies, namely, the latent class model. Then we describe the design of and the results obtained with the two simulation studies. It should be noted that because Monte-Carlo (MC) studies of bootstrap methods are extremely computer intensive, only factors believed to have a strong impact on the results were investigated. That is also the reason why we restricted ourselves to a single, relatively simple, type of FM model. The programming for the simulations was done in MATLAB (MathWorks 2002).

5.1 The latent class model

We deal with FMs of conditionally independent multinomial distributions for nominal response variables, also known as latent class (LC) models (Goodman 1974). Let Y_j have L_j nominal categories, i.e., $y_{ij} \in \{1, \ldots, L_j\}$. Each category l is associated with a binary variable defined by the indicator function $I(y_{ij} = l)$, which takes on the value 1 if the condition $y_{ij} = l$ holds and 0 otherwise.

The general FM model defined in Eq. (1) obtains the form of a LC model with S latent classes by using a class-specific conditional density of the form $f_s(\mathbf{y}_i; \theta_s) = \prod_{j=1}^{J}\prod_{l=1}^{L_j}\theta_{sjl}^{I(y_{ij}=l)}$. Here, θ_{sjl} is the probability that an observation belonging to component s gives response l to variable j. As usual, $\sum_{l=1}^{L_j}\theta_{sjl} = 1$.

Based on the sufficient conditions for the identifiability of LC models provided by McHugh (1956) and Goodman (1974), it can easily be shown that all models used

Table 1 Parameter values and corresponding separation levels

Categories	Level of separation					
	Well separated		Moderately separated		Ill separated	
	$s=1$	$s=2$	$s=1$	$s=2$	$s=1$	$s=2$
2 categories						
$l=1$	0.1	0.9	0.25	0.75	0.4	0.6
$l=2$	0.9	0.1	0.75	0.25	0.6	0.4
E	0.9572		0.6153		0.1364	
3 categories						
$l=1$	0.10	0.85	0.15	0.45	0.40	0.35
$l=2$	0.05	0.10	0.25	0.40	0.35	0.25
$l=3$	0.85	0.05	0.60	0.15	0.25	0.40
E	0.9681		0.5855		0.0943	

E computed with $a = 1$, $J = 5$, and $n = 10^6$ observations

in this simulation study are identified. For parameter estimation, we used the EM algorithm started for 50 different sets of starting values.

5.2 Monte Carlo study I

The factors that were varied in our first Monte Carlo (MC) study are: (1) the number of categories of the observed variables (L_j), (2) the sample size (n), (3) the component sizes, and (4) the level of separation of components. The number of categories L_j was either 2 or 3. For the sample size n, we used the values 300, 600, and 1200. Component sizes were generated using the expression $\pi_s = a^{s-1}\left(\sum_{r=1}^{S} a^{r-1}\right)^{-1}$, with $s = 1, \ldots, S$ and $a \geq 1$. Setting $a = 1$ yields equal proportions, whereas larger values of a yield more unequal component sizes. We used two different values for a: 1 and 2. The level of separation of components depends on the differences between the θ_{sjl} across classes, and can be controlled by the relative entropy E, a measure lying in the [0,1] interval defined as $1 - \left(-\sum_{i=1}^{n}\sum_{s=1}^{S}\alpha_{is}\log\alpha_{is}\right)/(n\log S)$. We used three separation levels: well-separated components (E close to 1), moderately-separated components, and ill-separated components (E close to 0). Table 1 shows the relative entropy and the parameter values for different levels of separation of the components. For simplicity, we use the same θ_{sjl} values for all J observed variables.

Because the ML estimate also suffers from the label-switching problem, for each data set the ML solution was ordered according to the minimum KL distance from the population values. The need to deal with this non-identifiability introduces a small bias in favor of the Bayes rule, i.e., it tends to be slightly more similar to the true value than expected. Because it was difficult to recover the natural order of ill-separated components when dealing with more than 2 latent classes, the number of classes in

Table 2 Results for study I (mean correct classification rates)

Design	ML allocation rate	Difference between aggregate and Bayes rules					
		NP			PAR		
		$B = 21$	51	101	21	51	101
Sample size							
300	0.8203	−0.0018	−0.0014	−0.0012	−0.0012	−0.0012	−0.0009
600	0.8249	−0.0029	−0.0026	−0.0027	−0.0020	−0.0014	−0.0011
1200	0.8282	−0.0038	−0.0041	−0.0042	−0.0021	−0.0016	−0.0017
No. of categories							
2	0.8322	−0.0039	−0.0041	−0.0040	−0.0022	−0.0018	−0.0017
3	0.8167	−0.0018	−0.0013	−0.0014	−0.0013	−0.0010	−0.0008
Comp. size							
Equal	0.8140	−0.0042	−0.0043	−0.0047	−0.0013	−0.0013	−0.0011
Unequal	0.8349	−0.0014	−0.0011	−0.0008	−0.0022	−0.0015	−0.0014
Comp. separation							
Well	0.9897	−0.0001	−0.0001	0.0000	−0.0001	−0.0001	0.0000
Moderate	0.8816	−0.0008	−0.0005	−0.0003	−0.0007	−0.0004	−0.0003
Ill	0.6020	−0.0077	−0.0076	−0.0078	−0.0046	−0.0038	−0.0035
Total	0.8245	−0.0028	−0.0027	−0.0027	−0.0018	−0.0014	−0.0013

the LC model was restricted to two ($S = 2$). The number of observed variables was set to five ($J = 5$), yielding an LC model that is identified.

This MC study uses a $2^2 \times 3^2$ factorial design with 36 cells. Within each cell, 200 data sets (replications) were generated. For each data set within each cell, 101 nonparametric (NP) and parametric (PAR) bootstrap samples were generated. This means that 7200 samples and 1454400 resamples were considered in this study. For the resamples we used the ML solution of the sample as starting set. The EM algorithm was stopped when the difference between two subsequent values of the log-likelihood values was smaller than 10^{-6} (tolerance level).

Table 2 presents the main results of the first MC study. Note that in a simulation study like ours we know the true class labels (c_i). Therefore, we can compute the correct allocation rates (using Eq. 9) for the Bayes and aggregate classifiers rules. The correct allocation rate reported in the second column is the average for the Bayes rule across the 7200 samples. The remaining columns report the difference between the average correct allocation rate for the corresponding aggregate classifier and the Bayes rule.

The main finding that can be derived from Table 2 is the remarkable performance of the Bayes rule for clustering problems. Contrary to what we expected, the aggregate classifiers perform slightly worse than the Bayes rule, though the differences are smaller for the larger B values. The parametric bootstrap version of the classifier performs slightly better than the nonparametric version.

Increasing the sample size seems to improve the performance of all classifiers, but the aggregate classifiers improve at a slightly smaller rate. Increasing the number of

Table 3 Multiple regression of differences between bootstrap and Bayes classifiers (study I)

	$B = 21$	$B = 51$	$B = 101$
Intercept	−0.0011*	−0.0016*	−0.0016**
Sample size			
300	–	–	–
600	−0.0009*	−0.0007	−0.0008
1200	−0.0014**	−0.0016***	−0.0019***
No. of categories			
2	–	–	–
3	0.0016***	0.0018***	0.0017***
Comp. size			
Equal	–	–	–
Unequal	0.0009*	0.0015***	0.0018***
Comp. separation			
Well	–	–	–
Moderate	−0.0007	−0.0004	−0.0003
Ill	−0.0061***	−0.0056***	−0.0056***
Bootstrap method			
Nonparametric	–	–	–
Parametric	0.0011**	0.0013***	0.0015***
Sparseness	–	–	–
F statistic	36.4083***	39.7149***	43.1717***

*** $p < 0.001$
** $p < 0.01$
* $p < 0.05$

categories of the observed variables reduces the performance of the Bayes rule as well as the difference between the two rules. When latent class are of unequal size, the performance of the Bayes rule is better than with equal class sizes. For equal class proportions, the parametric bootstrap classifier presents more similar results to the Bayes rule than the nonparametric bootstrap classifier; for unequal proportions, allocation rates for nonparametric bootstrap classifiers are more similar to the Bayes rule than the parametric bootstrap classifier. Finally, one can observe that the level of separation of components has a huge impact on the correct allocation rate for the Bayes rule. For well-separated components almost all observations are correctly classified (0.99); however, for ill-separated components the correct allocation rate is only 0.60. For well-separated and moderately separated components, both classifiers virtually yield the same results.

To shed further light on the effect of the design factors on the relative performance of these rules, a regression analysis was performed with "difference in correct allocation rate compared to the standard classifier" as dependent variable (Model I in Table 3). One observes that the relation between the design factors and the relative performance of these rules is significant. Taking into account that the first level of each design factor was used as the reference category, it can be seen that increasing the sample size increases the difference in favor of the Bayes rule and that reducing the level of separation of components has a similar effect. Increasing the number of categories has the largest impact on improving the bootstrap classifier in comparison to the Bayes rule (0.0017 for $B = 101$). Unequal component sizes also improve the aggregate rule in comparison to the Bayes rule. Given the massive amount of data, significance levels should be interpreted with some care in the sense that even very small effects may be significant.

5.3 Monte Carlo study II

The second study was aimed at having a closer look at the effects of the level of sparseness, defined as $n^{-1}\prod_{j=1}^{J} L_j$, and of the number of bootstrap replications on the differential performance of the two classification rules. The number of components (S) was set to 2, classes were of equal size, and the sample size was fixed to 600. For the level of separation of components, we used the same three sets of parameter values as in Table 1. The level of sparseness was manipulated by varying the number of categories the observed variables (2 or 3) as well as the number of observed variables (5, 8, 11, or 14). Compared to the Study I, apart from using LC models with a larger number of variables, we also increased the number of resamples up to B=1001 for both the nonparametric and parametric procedure, making it possible to assess the potential improvements for large B. Within each cell we simulated 30 data sets, which means that 360 samples and 720720 resamples had to be analyzed.

Table 4 presents the main results of the second MC study. We conclude that the results from both classifiers are virtually identical with a slightly better performance for the Bayes classifier (Total). We also observe that increasing the number of bootstrap resamples (B), despite of reducing the difference to the Bayes classifier, has a rather small impact on the performance of the bootstrap classifier. As expected, the relative performance of the aggregate classifier improves when the number of variables (and thus also sparseness) increases. When components are well separated, both classifiers can retrieve the right classification. However, for the smallest level of separation the aggregate classifier tends to perform slightly worse.

As before a regression analysis was performed to establish the relation between the difference in performance of the two classifiers and the design factors (Table 5). As can be seen, using the parametric bootstrap (compared to the nonparametric) improves the relative performance of the aggregate classifier. Moreover, increasing the number of variables reduces the difference in performance (only significant from 5 to 8 variables). On the other hand, reducing the level of separation of components reduces the performance of the aggregate classifier in relation to the Bayes classifier (significantly for Ill-separated level).

6 An empirical example

This section illustrates the topics related to uncertainty in LC modeling using the original version of the well-known Stouffer-Toby data set (Stouffer and Toby 1951, p. 406), a data set has been used by various other authors (e.g., Goodman 1974). It contains the information for 216 respondents with respect to whether they tend towards particularistic or universalistic values when confronted with four different role conflict situations. We set $S = 2$.

Table 4 Results for study II (mean correct classification rates)

Number of variables	Level of separation of components	ML allocation rate	Difference between aggregate and Bayes rules				
			$B = 11$	51	101	501	1001
Nonparametric							
	Well	0.9912	0	0	0	0	0
5	Moderate	0.8982	−0.0014	−0.0008	−0.0001	0	−0.0001
	Ill	0.5787	−0.0073	−0.0031	−0.0040	−0.0044	−0.0043
	Well	0.9977	0.0001	0	0.0001	0.0001	0.0001
8	Moderate	0.9286	0.0007	−0.0002	0.0001	0.0005	0.0002
	Ill	0.6474	−0.0365	−0.0365	−0.0387	−0.0374	−0.0363
	Well	0.9998	0	0	0	0	0
11	Moderate	0.9662	−0.0002	−0.0002	−0.0001	0.0001	0.0001
	Ill	0.7053	−0.0195	−0.0168	−0.0148	−0.0134	−0.0151
	Well	0.9999	−0.0001	0	0	0	0
14	Moderate	0.9765	0	−0.0003	0.0001	0.0001	0.0001
	Ill	0.7338	−0.0188	−0.0097	−0.0080	−0.0065	−0.0076
Parametric							
	Well	0.9912	0	0	0	0	0
5	Moderate	0.8982	−0.0006	−0.0002	−0.0004	−0.0001	−0.0001
	Ill	0.5787	−0.0035	−0.0053	−0.0076	−0.0043	−0.0027
	Well	0.9977	0.0001	0.0001	0	0	0
8	Moderate	0.9286	−0.0001	−0.0008	−0.0003	0	−0.0001
	Ill	0.6474	−0.0017	−0.0071	−0.0037	−0.0030	−0.0042
	Well	0.9998	0	0	0	0	0
11	Moderate	0.9662	0	0	0	0	0
	Ill	0.7053	−0.0001	−0.0001	−0.0001	−0.0001	−0.0001
	Well	0.9999	0	0	0	0	0
14	Moderate	0.9765	0	0	0	0	0
	Ill	0.7338	0.0001	0.0001	0.0001	0.0001	0.0001
Total		0.8686	−0.0037	−0.0034	−0.0032	−0.0028	−0.0029

Table 6 presents Bayes- and aggregate-classifier results for the Stouffer-Toby data set. We conclude that for the most problematic patterns, (2, 1, 2, 2), (2, 2, 2, 1), and (2, 2, 1, 2), the aggregate classifier confirms the decision by the Bayes classifier. Both rules give the same hard partition of the data set, including for the most uncertain patterns. This empirical application also suggests that a voting system of 100 resamples may not be enough to obtain a stable classification.

7 Final remarks

We introduced a model-based clustering aggregate classifier as an alternative to the Bayes rule. More specifically, we showed how to transform the bagging majority vote

Table 5 Multiple regression of differences between bootstrap and Bayes classifiers (study II)

	$B = 11$	$B = 51$	$B = 101$	$B = 501$	$B = 1001$
Intercept	−0.0017	−0.0004	−0.0010	−0.0008	−0.0006
No. of variables					
5	–	–	–	–	–
8	−0.0041	−0.0059***	−0.0051***	−0.0052***	−0.0056***
11	−0.0011	−0.0013	−0.0005	−0.0008	−0.0013
14	−0.0010	−0.0001	0.0007	0.0004	−0.0001
Comp. separation					
Well	–	–	–	–	–
Moderate	−0.0002	−0.0003	−0.0001	0.0001	0.00001
Ill	−0.0109***	−0.0098***	−0.0096***	−0.0086***	−0.0088***
Bootstrap method					
Nonparametric	–	–	–	–	–
Parametric	0.0064***	0.0045***	0.0045***	0.0045***	0.0047***
F statistic	10.7355***	14.6982***	13.8089***	12.8620***	13.6017***

$^{***}p < 0.001$

Table 6 Estimated class-1 allocation rates (empirical example)

Patterns	Posterior probabilities	Voting share (proportion)					
		NP			PAR		
		$B = 100$	1000	5000	100	1000	5000
(1,1,1,1)	0.000	0.000	0.000	0.000	0.000	0.000	0.000
(1,1,1,2)	0.001	0.020	0.009	0.006	0.000	0.002	0.002
(1,1,2,1)	0.001	0.000	0.002	0.001	0.000	0.000	0.000
(1,1,2,2)	0.017	0.040	0.026	0.021	0.020	0.012	0.012
(1,2,1,1)	0.001	0.000	0.000	0.000	0.000	0.000	0.000
(1,2,1,2)	0.013	0.040	0.026	0.020	0.010	0.012	0.010
(1,2,2,1)	0.018	0.030	0.006	0.005	0.020	0.005	0.003
(1,2,2,2)	0.287	0.430	0.383	0.364	0.370	0.336	0.347
(2,1,1,1)	0.001	0.000	0.000	0.000	0.000	0.000	0.000
(2,1,1,2)	0.031	0.050	0.020	0.016	0.010	0.008	0.009
(2,1,2,1)	0.045	0.030	0.011	0.007	0.000	0.000	0.001
(2,1,2,2)	0.505	0.500	0.528	0.545	0.580	0.548	0.538
(2,2,1,1)	0.033	0.020	0.003	0.003	0.000	0.001	0.001
(2,2,1,2)	0.425	0.400	0.463	0.448	0.420	0.438	0.436
(2,2,2,1)	0.518	0.620	0.588	0.590	0.600	0.588	0.574
(2,2,2,2)	0.959	1.000	1.000	1.000	1.000	1.000	1.000

Particularistic (1), $B = 5000$ resamples, ML and KL strategies procedure from the classification literature into a classification rule that can be used in the context of model-based clustering; that is, when true class labels are unknown. We studied variants based on the parametric and on the nonparametric bootstrap procedures for different numbers of bootstrap resamples. An important complication in the implementation of these bootstrap-based aggregate classifiers for FM models was the label-switching problem resulting from the non-identifiability of the class labels, a problem that had not received any attention so far in the literature on bootstrapping with mixture models. We introduced an adaptation of the Stephens' method to the bootstrap methodology as an alternative to using inequalities constraints which may yield distortions of the geometry of the bootstrap distribution.

From our Monte Carlo studies we concluded that the Bayes rule is remarkably stable as a classifier for model-based procedures. Even for sparse data, the aggregate classifier, which averages over a larger number of resamples, can hardly beat the Bayes rule. It should be emphasized that results for the aggregate classifier are conservative due to the setting of the non-identifiability of the FM model by using the ML solution to define the *true* classification of each observation, which is more favorable to the Bayes rule. Moreover, our simulation studies showed that the aggregate classifier works better with the parametric than with the nonparametric bootstrap.

Our results focused on FMs of independent multinomial distributions (LC models). Future research could be aimed at extending our findings and proposals to other model-based clustering procedures based on FM models, such as multivariate normal mixture models and FMs of regression models.

References

Basford KE, McLachlan GJ (1985) Estimation of allocation rates in a cluster analysis context. J Am Stat Assoc 80:286–293

Bauer E, Kohavi R (1999) An empirical comparison of voting classification algorithms: bagging, boosting, and variants. Mach Learn 36:105–139

Breiman L (1996a) Bagging predictors. Mach Learn 24:123–140

Breiman L (1996b) Heuristics of instability and stabilization in model selection. Ann Stat 24:2350–2383

Breiman L (1998) Arcing classifier (with discussion). Ann Stat 26:801–849

Bühlmann P, Yu B (2002) Analyzing bagging. Ann Stat 30:927–961

Celeux G, Hurn M, Robert CP (2000) Computational and inferential difficulties with mixture posterior distributions. J Am Stat Assoc 95:957–970

Dempster AP, Laird NM, Rubin DB (1977) Maximum likelihood from incomplete data via the EM algorithm (with discussion). J R Stat Soc B 39:1–38

Dias JG (2005) Bootstrapping latent class models. In: Weihs C, Gaul W (eds) Classification—the ubiquitous challenge. Springer, Berlin, pp 121–128

Dias JG, Wedel M (2004) An empirical comparison of EM, SEM and MCMC performance for problematic Gaussian mixture likelihoods. Stat Comput 14:323–332

Duda RO, Hart PO, Stork DG (2001) Pattern classification, 2nd edn. Wiley, New York

Efron B (1979) Bootstrap methods: another look at the jackknife. Ann Stat 7:1–26

Efron B (1987) Better bootstrap confidence intervals (with discussion). J Am Stat Assoc 82:171–200

Efron B, Tibshirani R (1993) An introduction to the bootstrap. Chapman & Hall, London

Freud Y, Schapire RE (1996) Experiments with a new boosting algorithm. In: Saitta L (ed) Proceedings 13th International Conference on Machine Learning. Morgan Kaufmann, San Francisco, pp 148–156

Goodman LA (1974) Exploratory latent structure analysis using both identifiable and unidentifiable models. Biometrika 61:215–231

Hothorn T, Lausen B (2003) Double-bagging: combining classifiers by bootstrap aggregation. Pattern Recognit 36:1303–1309

Magidson J, Vermunt JK (2001) Latent class factor and cluster models, bi-plots and related graphical displays. Sociol Methodol 31:223–264

MathWorks (2002) MATLAB 6.5. The MathWorks Inc., Natick, MA

McHugh RB (1956) Efficient estimation and local identification in latent class analysis. Psychometrika 21:331–347

McLachlan GJ (1992) Discriminant analysis and statistical pattern recognition. Wiley, New York

McLachlan GJ, Peel D (2000) Finite mixture models. Wiley, New York

Richardson S, Green PJ (1997) On Bayesian analysis of mixtures with an unknown number of components (with discussion). J R Stat Soc B 59:731–792

Stephens M (1997) Discussion on 'On Bayesian analysis of mixtures with an unknown number of components (with discussion)'. J R Stat Soc B 59:768–769

Stephens M (2000) Dealing with label switching in mixture models. J R Stat Soc B 62:795–809

Stouffer SA, Toby J (1951) Role conflict and personality. Am J Sociol 56:395–406