

pgmm Version 1.0 for R: Model-based Clustering and Classification via Latent Gaussian Mixture Models

Technical Report 2011-320, Department of Mathematics & Statistics, University of Guelph.

Paul D. McNicholas^{‡§} T. Brendan Murphy[¶] Aaron F. McDaid[¶]
K. Raju Jampani[‡] Larry Banks[‡]

December 2011

Abstract

An R package is presented that implements methodology for classification and clustering using a family of latent Gaussian mixture models known as parsimonious Gaussian mixture models (PGMM). Both mixture model-based clustering and classification are available within the `pgmm` package, where the latter is a semi-supervised version of the former. Parameter estimation for each of the twelve members of the PGMM family is carried out using an alternating expectation-conditional maximization algorithm, and the Bayesian information criterion or the integrated completed likelihood can be used for model selection. The `pgmm` package allows the user to provide starting values for labels, giving tremendous flexibility. Examples are used to illustrate the application of software for clustering and classification.

[‡]Department of Mathematics & Statistics, University of Guelph, Guelph, Ontario, Canada.

[§]Corresponding author. E-mail: paul.mcnicholas@uoguelph.ca

[¶]School of Mathematics Sciences, University College Dublin, Ireland.

1 Introduction

The idiom ‘model-based clustering’ is commonly used to describe mixture model-based clustering, whereby parametric finite mixture models are used to cluster data. The density of a G -component parametric finite mixture model can be written $f(\mathbf{x} \mid \boldsymbol{\vartheta}) = \sum_{g=1}^G \pi_g \rho_g(\mathbf{x} \mid \boldsymbol{\theta}_g)$, where $\pi_g > 0$, such that $\sum_{g=1}^G \pi_g = 1$ is the mixing proportion for component g , $\rho_g(\mathbf{x} \mid \boldsymbol{\theta}_g)$ is the density of a random vector \mathbf{X} with parameters $\boldsymbol{\theta}_g$, and $\boldsymbol{\vartheta} = (\boldsymbol{\pi}, \boldsymbol{\theta}_1, \dots, \boldsymbol{\theta}_G)$ is the vector of parameters with $\boldsymbol{\pi} = (\pi_1, \dots, \pi_G)$.

The finite Gaussian mixture model has been used throughout the model-based clustering literature (cf. McNicholas, 2011). The density of a finite Gaussian mixture model is given by $\xi(\mathbf{x} \mid \boldsymbol{\vartheta}) = \sum_{g=1}^G \pi_g \phi(\mathbf{x} \mid \boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g)$, where $\phi(\mathbf{x} \mid \boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g)$ is the density of a multivariate Gaussian random variable \mathbf{X} with mean $\boldsymbol{\mu}_g$ and covariance matrix $\boldsymbol{\Sigma}_g$, and $\boldsymbol{\vartheta} = (\boldsymbol{\pi}, \boldsymbol{\mu}_1, \dots, \boldsymbol{\mu}_G, \boldsymbol{\Sigma}_1, \dots, \boldsymbol{\Sigma}_G)$ is again the vector of parameters. Parameters can be estimated using some variant of the expectation-maximization (EM) algorithm (Dempster et al., 1977) but other approaches are also available. The Gaussian mixture density has a total of $(G - 1) + Gp + Gp(p + 1)/2$ parameters, of which $Gp(p + 1)/2$ are in the component covariance matrices $\boldsymbol{\Sigma}_1, \dots, \boldsymbol{\Sigma}_G$. Therefore, these covariance matrices $\boldsymbol{\Sigma}_1, \dots, \boldsymbol{\Sigma}_G$ are often decomposed to facilitate the construction of more parsimonious models for clustering (see Celeux and Govaert, 1995; McNicholas and Murphy, 2008, for examples).

2 Parsimonious Gaussian mixture models

2.1 The models

First, consider the factor analysis model (Spearman, 1904) where we assume that a p -dimensional random vector \mathbf{X} can be modelled using a q -dimensional vector of latent factors

\mathbf{U} , where $q \ll p$. The factor analysis model can be written $\mathbf{X} = \boldsymbol{\mu} + \boldsymbol{\Lambda}\mathbf{U} + \boldsymbol{\epsilon}$, where $\boldsymbol{\Lambda}$ is a $p \times q$ matrix of factor loadings, the latent variables $\mathbf{U} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_q)$, and $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Psi})$, where $\boldsymbol{\Psi}$ is a $p \times p$ diagonal matrix. The marginal distribution of \mathbf{X} arising from this model is $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Lambda}\boldsymbol{\Lambda}' + \boldsymbol{\Psi})$. Ghahramani and Hinton (1997) introduced a mixture of factor analyzers model, which has density of the form in Equation ?? but with $\boldsymbol{\Sigma}_g = \boldsymbol{\Lambda}_g\boldsymbol{\Lambda}_g' + \boldsymbol{\Psi}$. Tipping and Bishop (1999) proposed the mixture of probabilistic principal component analyzers model in which $\boldsymbol{\Sigma}_g = \boldsymbol{\Lambda}_g\boldsymbol{\Lambda}_g' + \psi_g\mathbf{I}_p$ and McLachlan and Peel (2000b) used the most general mixture of factor analyzers, i.e., with $\boldsymbol{\Sigma}_g = \boldsymbol{\Lambda}_g\boldsymbol{\Lambda}_g' + \boldsymbol{\Psi}_g$.

McNicholas and Murphy (2008) developed a family of eight parsimonious Gaussian mixture models (PGMMs) for clustering by imposing, or not, each of the constraints $\boldsymbol{\Lambda}_g = \boldsymbol{\Lambda}$, $\boldsymbol{\Psi}_g = \boldsymbol{\Psi}$, and $\boldsymbol{\Psi}_g = \psi_g\mathbf{I}_p$ upon the mixture of factor analyzers component covariance structure $\boldsymbol{\Sigma}_g = \boldsymbol{\Lambda}_g\boldsymbol{\Lambda}_g' + \boldsymbol{\Psi}_g$. This family includes both the mixture of factor analyzers model and the mixture of probabilistic principal component analyzers model as members. This family of mixture models has several computational advantages, some of which are described in Section 2.4. McNicholas (2010) introduced a model-based classification framework based on the PGMM family. This classification framework, which is a semi-supervised version of the model-based clustering framework introduced by McNicholas and Murphy (2008), can give excellent classification performance when applied to real data (cf. McNicholas, 2010) and is illustrated in sections 4.5 and 4.6.

McNicholas and Murphy (2010) introduced a modified factor analysis model, thereby extending the PGMM family of models. Specifically, they further parameterized the factor analysis covariance structure by writing $\boldsymbol{\Psi}_g = \omega_g\boldsymbol{\Delta}_g$, where $\omega_g \in \mathbb{R}^+$ and $\boldsymbol{\Delta}_g = \text{diag}\{\delta_1, \delta_2, \dots, \delta_p\}$, such that $|\boldsymbol{\Delta}_g| = 1$ for $g = 1, 2, \dots, G$. The resulting covariance structure $\boldsymbol{\Sigma}_g = \boldsymbol{\Lambda}_g\boldsymbol{\Lambda}_g' + \omega_g\boldsymbol{\Delta}_g$ is known as the modified factor analysis covariance structure. In addition to the constraint $\boldsymbol{\Lambda}_g = \boldsymbol{\Lambda}$, we can impose valid combinations of the constraints

$\omega_g = \omega$, $\Delta_g = \Delta$, and $\Delta_g = \mathbf{I}_p$ to give a family of twelve Gaussian mixture models (Table 1). Although McNicholas and Murphy (2010) used these models for clustering, they can also be used for model-based classification in the fashion described by McNicholas (2010).

Table 1: The covariance structure, nomenclature, and number of free covariance parameters for each member of the PGMM and EPGMM families.

PGMM	EPGMM	Covariance Structure	No. of Free Cov. Parameters
CCC	CCCC	$\Sigma_g = \Lambda\Lambda' + \omega\mathbf{I}_p$	$[pq - q(q - 1)/2] + 1$
CUC	CCUC	$\Sigma_g = \Lambda\Lambda' + \omega_g\mathbf{I}_p$	$[pq - q(q - 1)/2] + G$
UCC	UCCC	$\Sigma_g = \Lambda_g\Lambda_g' + \omega\mathbf{I}_p$	$G[pq - q(q - 1)/2] + 1$
UUC	UCUC	$\Sigma_g = \Lambda_g\Lambda_g' + \omega_g\mathbf{I}_p$	$G[pq - q(q - 1)/2] + G$
CCU	CCCU	$\Sigma_g = \Lambda\Lambda' + \omega\Delta$	$[pq - q(q - 1)/2] + p$
–	CCUU	$\Sigma_g = \Lambda\Lambda' + \omega_g\Delta$	$[pq - q(q - 1)/2] + [G + (p - 1)]$
UCU	UCCU	$\Sigma_g = \Lambda_g\Lambda_g' + \omega\Delta$	$G[pq - q(q - 1)/2] + p$
–	UCUU	$\Sigma_g = \Lambda_g\Lambda_g' + \omega\Delta_g$	$G[pq - q(q - 1)/2] + [G + (p - 1)]$
–	CUCU	$\Sigma_g = \Lambda\Lambda' + \omega\Delta_g$	$[pq - q(q - 1)/2] + [1 + G(p - 1)]$
CUU	CUUU	$\Sigma_g = \Lambda\Lambda' + \omega_g\Delta_g$	$[pq - q(q - 1)/2] + Gp$
–	UUCU	$\Sigma_g = \Lambda_g\Lambda_g' + \omega\Delta_g$	$G[pq - q(q - 1)/2] + [1 + G(p - 1)]$
UUU	UUUU	$\Sigma_g = \Lambda_g\Lambda_g' + \omega_g\Delta_g$	$G[pq - q(q - 1)/2] + Gp$

The `pgmm` package (McNicholas et al., 2011) implements all twelve models for both model-based clustering and classification and is therefore an implementation of the work of McNicholas and Murphy (2008, 2010) and McNicholas (2010). McNicholas and Murphy (2010) referred to their family of models (Table 1) as the ‘expanded PGMM’ (EPGMM) family. Hereafter, we shall use ‘PGMM family’ to mean the family comprising all twelve models.

Consider the CUU mixture model for model-based clustering. In this case, the likelihood is given by

$$\mathcal{L}(\boldsymbol{\vartheta}) = \prod_{i=1}^n \sum_{g=1}^G \pi_g \phi(\mathbf{x}_i \mid \boldsymbol{\mu}_g, \Lambda\Lambda' + \omega_g\Delta_g),$$

where $\boldsymbol{\vartheta} = (\boldsymbol{\pi}, \boldsymbol{\mu}_1, \dots, \boldsymbol{\mu}_G, \Lambda, \omega_1, \dots, \omega_G, \Delta_1, \dots, \Delta_G)$ and $\phi(\mathbf{x}_i \mid \boldsymbol{\mu}_g, \Lambda\Lambda' + \omega_g\Delta_g)$ is the density of a p -dimensional multivariate Gaussian random vector \mathbf{X}_i with mean $\boldsymbol{\mu}_g$ and covariance matrix $\Lambda\Lambda' + \omega_g\Delta_g$. Before writing down the likelihood in the model-based classification

case, we need to introduce Z_{ig} to denote group memberships. We define Z_{ig} so that $z_{ig} = 1$ if observation i is in component g and $z_{ig} = 0$ otherwise. Suppose we observe n p -dimensional data vectors and that k of these are known to belong to one of G_1 groups. Without loss of generality, we can order these data so that the first k have known group memberships: $\mathbf{x}_1, \dots, \mathbf{x}_k, \mathbf{x}_{k+1}, \dots, \mathbf{x}_n$. Then the likelihood for these data in the model-based classification framework, using the CUU model, can be written

$$\mathcal{L}(\boldsymbol{\vartheta}) = \underbrace{\prod_{i=1}^k \prod_{g=1}^{G_1} [\pi_g \phi(\mathbf{x}_i | \boldsymbol{\mu}_g, \boldsymbol{\Lambda} \boldsymbol{\Lambda}' + \omega_g \boldsymbol{\Delta}_g)]^{z_{ig}}}_{\text{Known Group Memberships}} \times \underbrace{\prod_{j=k+1}^n \sum_{h=1}^{G_2} \pi_h \phi(\mathbf{x}_j | \boldsymbol{\mu}_h, \boldsymbol{\Lambda} \boldsymbol{\Lambda}' + \omega_h \boldsymbol{\Delta}_h)}_{\text{Unknown Group Memberships}},$$

for $G_2 \geq G_1$, where $\boldsymbol{\vartheta}$ and $\phi(\mathbf{x}_i | \boldsymbol{\mu}_g, \boldsymbol{\Lambda} \boldsymbol{\Lambda}' + \omega_g \boldsymbol{\Delta}_g)$ are defined as before. Note that model-based clustering can be considered a special case of model-based classification, with $k = 0$ and $G_2 = G_1$. Note that within both clustering and classification frameworks, the unknown group memberships are replaced by their conditional expected values, which we denote

$$\hat{z}_{jg} = \frac{\hat{\pi}_g \phi(\mathbf{x}_j | \hat{\boldsymbol{\mu}}_g, \hat{\boldsymbol{\Lambda}}_g \hat{\boldsymbol{\Lambda}}_g' + \hat{\boldsymbol{\Psi}}_g)}{\sum_{h=1}^G \hat{\pi}_h \phi(\mathbf{x}_j | \hat{\boldsymbol{\mu}}_h, \hat{\boldsymbol{\Lambda}}_h \hat{\boldsymbol{\Lambda}}_h' + \hat{\boldsymbol{\Psi}}_h)},$$

for $j = k + 1, \dots, n$.

2.2 Parameter estimation: AECM algorithms

The EM algorithm is an iterative algorithm for maximum likelihood estimation when data are incomplete or are formulated as being incomplete. The EM algorithm considers the likelihood on the basis of ‘complete-data’, that is the observed data together with missing or latent data. In the E-step, the expected value of the complete-data log-likelihood is computed and, in the M-step, this expected value is maximized with respect to the model

parameters. The E- and M-steps are then iterated until convergence. The expectation-conditional maximization (ECM) algorithm (Meng and Rubin, 1993) is a variant of the EM algorithm in which the M-step is replaced by a number of, typically, more computationally efficient CM-steps. The alternating ECM (AECM) algorithm (Meng and van Dyk, 1997) extends the ECM algorithm by allowing different complete-data at each stage of an iteration.

In the case of the PGMM family of models, there are two sources of missing data — the group memberships and the latent factors — and so the AECM algorithm is used for parameter estimation. McLachlan and Peel (2000a, Chapter 8) give details of the AECM algorithm for the mixtures of factor analyzers model (UUU in our notation). Details on the AECM algorithms used for the PGMMs for model-based clustering are given by McNicholas and Murphy (2008, 2010) and McNicholas et al. (2010), and details on parameter estimation for the PGMMs for model-based classification are given by McNicholas (2010).

As described by McNicholas et al. (2010) and others, the Aitken acceleration (Aitken, 1926) is used to determine convergence of these AECM algorithms. The Aitken acceleration at iteration t is given by

$$a^{(t)} = \frac{l^{(t+1)} - l^{(t)}}{l^{(t)} - l^{(t-1)}},$$

where $l^{(t+1)}$, $l^{(t)}$, and $l^{(t-1)}$ are the log-likelihood values from iterations $t + 1$, t , and $t - 1$, respectively. Using the Aitken acceleration, we can compute an asymptotic estimate of the log-likelihood at iteration $t + 1$: $l_{\infty}^{(t+1)} = l^{(t)} + (l^{(t+1)} - l^{(t)})/(1 - a^{(t)})$. For further details, see Böhning et al. (1994). In the `pgmm` package, the stopping criterion proposed by Lindsay (1995) is used so that an AECM algorithm is stopped when

$$l_{\infty}^{(t+1)} - l^{(t+1)} < \varepsilon, \tag{1}$$

for ε small. The default setting is $\varepsilon = 0.1$, but this can be changed by the user (cf. Section 3).

For both clustering and classification applications, the final predicted classifications for the members of the PGMM family are taken to be the maximum *a posteriori* (MAP) classifications given \hat{z}_{ig} , where $\text{MAP}\{\hat{z}_{ig}\} = 1$ if $\max_g\{z_{ig}\}$ occurs in component g and $\text{MAP}\{\hat{z}_{ig}\} = 0$ otherwise, for $j = k + 1, \dots, n$.

2.3 Model selection

The Bayesian information criterion (BIC; Schwarz, 1978) is the most popular method used to select the ‘best’ member of a family of mixture models: $\text{BIC} = 2l(\mathbf{x}, \hat{\boldsymbol{\theta}}) - m \log n$, where $l(\mathbf{x}, \hat{\boldsymbol{\theta}})$ is the maximized log-likelihood, $\hat{\boldsymbol{\theta}}$ is the maximum likelihood estimate of $\boldsymbol{\theta}$, m is the number of free parameters in the model, and n is the number of observations. Lopes and West (2004) demonstrated that the BIC is effective at choosing the number of factors in a factor analysis model and Keribin (2000) give a theoretical justification for its use under certain regularity conditions.

The BIC focuses on mixture component selection rather than cluster selection *per se*. To try to shift this focus, Biernacki et al. (2000) proposed the integrated completed likelihood (ICL) as an alternative to the BIC. The ICL essentially penalizes the BIC for estimated mean entropy, thereby penalizing mixture components that are more spread out. In applications, an approximate ICL is used: $\text{ICL} \approx \text{BIC} + \sum_{i=k+1}^n \sum_{g=1}^G \text{MAP}\{\hat{z}_{ig}\} \log \hat{z}_{ig}$. Both the BIC and the ICL are available for model selection within `pgmm`.

2.4 A computational nicety: the Woodbury identity

Following several others (McLachlan and Peel, 2000a; McNicholas and Murphy, 2008, 2010; McNicholas, 2010; McNicholas et al., 2010; Andrews and McNicholas, 2011a,b), we make use of the Woodbury identity (Woodbury, 1950) in our computations. Each AECM algorithm

requires the inversion of the $p \times p$ covariance matrices $\Sigma_1, \dots, \Sigma_G$ at each iteration; this becomes increasingly computationally expensive as the number of variables p gets larger. One of the main computational advantages of the PGMM approach is that the Woodbury identity can be used to avoid the inversion of any non-diagonal $p \times p$ matrices. In general, for an $n \times n$ matrix \mathbf{A} , an $n \times k$ matrix \mathbf{U} , a $k \times k$ matrix \mathbf{C} , and a $k \times n$ matrix \mathbf{V} , the Woodbury identity is $(\mathbf{A} + \mathbf{UCV})^{-1} = \mathbf{A}^{-1} - \mathbf{A}^{-1}\mathbf{U}(\mathbf{C}^{-1} + \mathbf{VA}^{-1}\mathbf{U})^{-1}\mathbf{VA}^{-1}$. Now, setting $\mathbf{U} = \Lambda_g$, $\mathbf{V} = \Lambda'_g$, $\mathbf{A} = \omega_g \Delta_g$, and $\mathbf{C} = \mathbf{I}_q$ gives

$$(\omega_g \Delta_g + \Lambda_g \Lambda'_g)^{-1} = (\omega_g \Delta_g)^{-1} - (\omega_g \Delta_g)^{-1} \Lambda_g [\mathbf{I}_q + \Lambda'_g (\omega_g \Delta_g)^{-1} \Lambda_g]^{-1} \Lambda'_g (\omega_g \Delta_g)^{-1}. \quad (2)$$

While the matrix on the left-hand-side of Equation 2 is a $p \times p$ matrix, the matrices that require inversion on the right hand side are either diagonal or much smaller ($q \times q$). This is a significant computational advantage, especially when p is large. Furthermore, the determinant of the covariance matrix can be computed using a related formula: $|\Lambda_g \Lambda'_g + \omega_g \Delta_g| = |\omega_g \Delta_g| / |\mathbf{I}_q - \Lambda'_g (\Lambda_g \Lambda'_g + \omega_g \Delta_g)^{-1} \Lambda_g|$. The `pgmm` package uses this identity and (2) to increase computational speed.

2.5 Parameter initialization

The initialization of the covariance parameters in `pgmm` is carried out in a similar fashion to that described by McNicholas and Murphy (2008). For model-based clustering, three different options are given for starting values of the group memberships \hat{z}_{ig} : random, k -means, and user-specified. Initialization of the \hat{z}_{ig} for model-based classification follows Andrews and McNicholas (2011b), by initializing the \hat{z}_{ig} uniformly so that $\hat{z}_{jg} = 1/G$ for $g = 1, \dots, G$ and $j = k + 1, \dots, n$.

3 Notes on the code

The `pgmm` package offers one function and three data sets. The function, `pgmmEM()`, is essentially an R (R Development Core Team, 2011) wrapper for code written in the C language; however, initialization of the covariance parameters and the group memberships is carried out in R. Communication between R and C is achieved using the `.C()` function. The `pgmmEM()` function has the form

```
pgmmEM<-function(x,class=NULL,icl=FALSE,zstart=2,cccStart=TRUE,loop=3,
zlist=NULL,qmax=2,qmin=1,Gmax=2,Gmin=1,modelSubset=NULL,seed=123456,
tol=0.1,relax=FALSE)
```

and takes the following arguments:

- `x` is a matrix or data frame such that rows correspond to observations and columns correspond to variables.
- `class` is either `NULL` or a vector of length n . If `NULL` then model-based clustering is performed. If a vector, then model-based classification is performed. In this latter case, the i th entry of `class` is either zero, indicating that the component membership of observation i is unknown, or corresponds to the component membership of observation i . See examples in sections 4.5 and 4.6.
- `icl` is logical. If `TRUE`, then the ICL is used for model selection. Otherwise, the BIC is used for model selection.
- `zstart` is a number that controls what starting values are used: (1) random; (2) k-means; or (3) user-specified via `zlist`.
- `cccStart` is logical. If `TRUE`, then random starting values are put through the CCC model and the resulting group memberships are used as starting values for the models specified in `modelSubset`. Only relevant for `zstart=1`.

- `loop` is a number specifying how many different random starts should be used. Only relevant for `zstart=1`.
- `zlist` is a list comprising `Gmin:Gmax` vectors of initial classifications such that `zlist[[k]]` gives the starting values for the k -component model. Only relevant for `zstart=3`.
- `qmax` is the maximum number of factors to be used.
- `qmin` is the minimum number of factors to be used.
- `Gmax` is the maximum number of components to be used.
- `Gmin` is the minimum number of components to be used.
- `modelSubset` is a vector of strings giving the models to be used.
- `seed` is the pseudo-random number seed to be used.
- `tol` specifies the ε value for the convergence criteria (Equation 1). Values of `tol` greater than the default are not accepted.
- `relax` is logical. By default, $q \leq p/2$ but setting `relax=TRUE` relaxes this constraint and allows $q \leq 3p/4$.

There are several checks on the accuracy of the parameters entered by the user. For example, `qmin` must be less than `qmax` and `zlist` must be either `NULL` or a list of integers. Ideally, `qmax` should be much less than the number of variables; however, to allow for low-dimensional data, `pgmmEM()` only returns an error if the number of variables is not at least twice the value of `qmax`. Furthermore, the user has the option to `relax` this constraint. In terms of structure, initializations are carried out in `R` and the initial values are passed to `C`. The precise use of the `class` and `models` settings is best illustrated using examples (cf. Section 4). We also use these examples to introduce the three data sets included in `pgmm`.

In terms of output, the `pgmmEM()` function returns an object of class `"pgmm"`, which is a list containing the following items:

- `map` is a vector of integers, taking values in `Gmin:Gmax`, indicating the maximum *a posteriori* classifications for the best model.
- `model` is a string giving the name of the best model.
- `g` is the number of groups for the best model.
- `q` is the number of factors for the best model.
- `zhat` is a matrix giving the raw values upon which `map` is based.
- `plot_info` stores information to enable `plot()`.
- `summ_info` stores information to enable `summary()`.

In addition, the object will contain one of the following:

- `bic` is a list containing the BIC for each model.
- `icl` is a list containing the ICL for each model.

Printed output is also produced, as illustrated in the following section. This output explicitly states the method that was used to obtain starting values for the \hat{z}_{jg} so as to emphasize that the performance of `pgmmEM()` depends on the starting values. Dedicated `print()`, `summary()`, and `plot()` functions are available for objects of class "pgmm".

4 Data Analysis Examples

4.1 Model-based clustering of coffee: *k*-means starts

Streuli (1973) reported the chemical composition of coffee samples collected from around the world. Forty-three samples were collected from 29 countries and beans were either from the Arabica or Robusta species. Twelve of the thirteen chemical constituents reported by Streuli (1973) are given in `coffee`; the omitted variable is total chlorogenic acid. We use

`pgmmEM()` to cluster these data using k -means starting values; we use all 12 members of the PGMM family and the ICL is used for model selection.

```
data("coffee")
x<-coffee[,-c(1,2)]
x<-scale(x)
coffee_clust<-pgmmEM(x,zstart=2,qmax=3,Gmax=3,icl=TRUE)
```

Based on k -means starting values, the best model (ICL) for the range of factors and components used is a CCUU model with $q = 1$ and $G = 2$.

The ICL for this model is -1292.821.

```
table(coffee[,1],coffee_clust$map)
```

1	2	
1	36	0
2	0	7

```
plot(coffee_clust,onlyAll=TRUE)
```

In this example, one of the four new models introduced by McNicholas and Murphy (2010) was selected and gave perfect classifications. Figure 1 gives the range of values of q plotted by the ICL value and the number of components G . For all twelve members of the PGMM family, the ICL prefers a one-factor model ($q = 1$); a $G = 2$ component model is selected for ten of the twelve members. Note that `onlyAll=FALSE`, which is the default in `plot()`, gives the image in Figure 1 but then prompts the user to cycle through each plot individually.

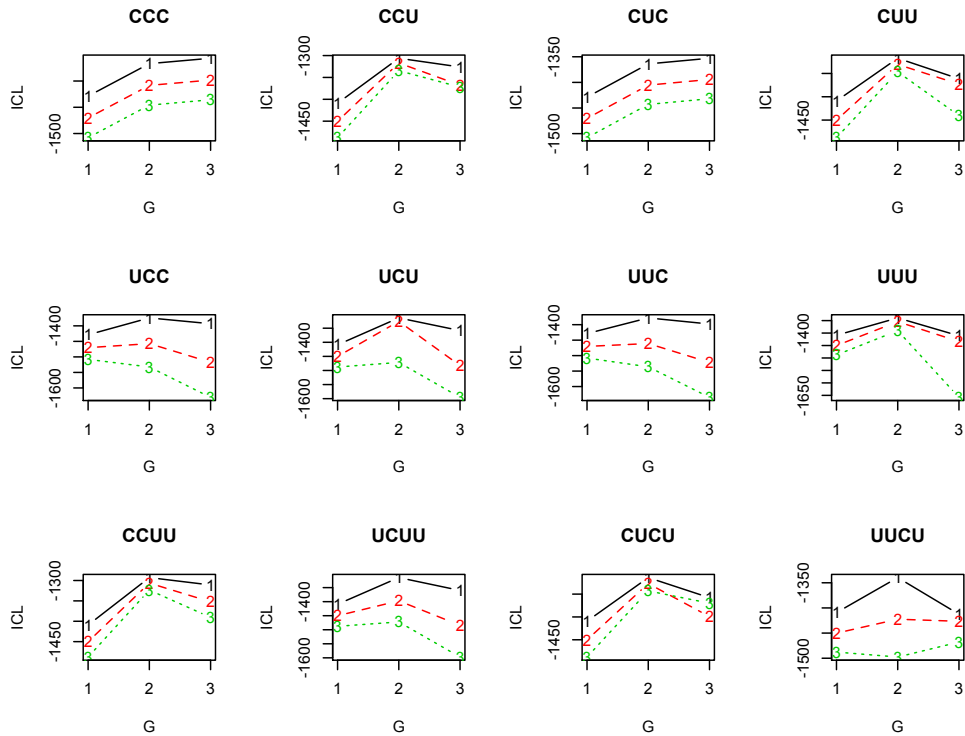


Figure 1: The range of values of q plotted by the ICL value and the number of components G for the PGMM models run on the coffee data.

4.2 Model-based clustering coffee: custom starts

To illustrate the mechanism for using user-specified starting values, we also cluster the coffee data using starting values obtained from hierarchical clustering. To do this, we will need to pass a list of starting values to `pgmmEM()`, as follows. Again, we use all 12 members of the PGMM family.

```
data("coffee")
x<-coffee[,-c(1,2)]
x<-scale(x)
hcl<-hclust(dist(x))
```

```

z<-list()
for(g in 1:4){z[[g]]<-cutree(hcl,k=g)}
coffee_clust2<-pgmmEM(x,zstart=3,qmax=4,Gmax=4,zlist=z)

```

Based on custom starting values, the best model (BIC) for the range of factors and components used is a UUC model with $q = 2$ and $G = 3$.

The BIC for this model is -1174.209.

```
table(coffee[,1],coffee_clust2$map)
```

```

      1  2  3
1 36  0  0
2  0  5  2

```

Here a three-component model is selected, wherein the seven Robusta samples have been broken across two groups. Looking at `coffee[,2]`, we can see that this classification separates the samples from Togo and Madagascar into a group of their own. Note that the chosen model (UUC) is a mixture of probabilistic principal component analyzers.

4.3 Model-based clustering of Italian wines: three random starts

Forina et al. (1986) reported data on three types of wine from the Piedmont region of Italy. The `wine` data consist of 178 samples of 27 measurements. The code and the resulting output for clustering of these data, using the CUU model with three different random starts for the group memberships, are as follows. In addition to giving a classification table, we give the BIC values.

```

data("wine")
x<-wine[,-1]

```

```
x<-scale(x)
wine_clust<-pgmmEM(x,zstart=1,loop=3,qmax=6,Gmax=4)
```

Based on 3 random starts, the best model (BIC) for the range of factors and components used is a CUU model with $q = 4$ and $G = 3$.

The BIC for this model is -11427.65.

```
table(wine[,1],wine_clust$map)
```

	1	2	3
1	59	0	0
2	1	69	1
3	0	0	48

This clustering analysis, which is based on three random starts, gave predicted classifications that are slightly inferior to the findings of McNicholas and Murphy (2008), who misclassified only one sample. However, the BIC for this model is actually greater than the BIC from their model (-11,454.32); this phenomenon, whereby the model with the greater BIC gives worse classification performance, has been previously observed. An alternative to the BIC that consistently performs better has yet to emerge.

4.4 Model-based clustering of Italian wines: k -means starts

The `wine` data (cf. Section 4.3) were clustered using all twelve models with k -means starting values for the group memberships.

```
data("wine")
x<-wine[,-1]
x<-scale(x)
```

```
wine_clust2<-pgmmEM(x,zstart=2,qmax=6,Gmax=4)
```

Based on k -means starting values, the best model (BIC) for the range of factors and components used is a CUU model with $q = 6$ and $G = 3$.

The BIC for this model is -11479.09.

```
table(wine[,1],wine_clust2$map)
```

```
      1  2  3
1     0  0 59
2    67  1  3
3     0 48  0
```

Here, with k -means starting values, we misclassify four samples, which is two samples more than in Section 4.3. Using k -means starting values is, however, faster than using random starts and thus k -means starts are used as the default in `pgmmEM()`.

4.5 Model-based classification of olive oil data into areas

Forina and Tiscornia (1982) and Forina et al. (1983) reported the percent composition of eight fatty acids found by lipid fractionation of 572 Italian olive oils. The oils come from nine areas (cf. Table 2) across three regions of Italy (Southern Italy, Sardinia, and Northern Italy). These data are described and then analyzed within the model-based classification framework by McNicholas (2010). Here, we consider model-based classification of these data by treating some of the olive oil samples as having unknown group memberships. We use three members of the PGMM family: CUC, CUU, and CUCU.

```
data("olive")
x<-olive[,-c(1,2)]
```

```

x<-scale(x)
cls<-olive[,2]
for(i in 1:dim(olive)[1]){if(i%%3==0){cls[i]<-0}}
olive_class<-pgmmEM(x,cls,qmax=4,Gmax=9,Gmin=9,modelSubset=c("CUC","CUU",
"CUCU"))

```

Based on the labelled and unlabelled data provided, the best model (BIC) for the range of factors and components used is a CUU model with $q = 4$ and $G = 9$. The BIC for this model is -5984.987.

```

cls_ind<-(cls==0)
table(olive[cls_ind,2],olive_class$map[cls_ind])

```

	1	2	3	4	5	6	7	8	9
1	7	0	0	1	0	0	0	0	0
2	0	19	0	0	0	0	0	0	0
3	0	0	67	1	0	0	0	0	0
4	1	0	0	11	0	0	0	0	0
5	0	0	0	0	21	1	0	0	0
6	0	0	0	0	1	10	0	0	0
7	0	0	0	0	0	0	16	1	0
8	0	0	0	0	0	0	0	16	0
9	0	0	0	0	0	0	4	0	13

From this output, we can see that this model-based classification approach gave very good performance, with just 10 of 190 samples misclassified. Furthermore, all of the misclassifications were from areas in the same region (Table 2).

Table 2: Model-based classifications for the olive oil data cross-tabulated against area, where one-third of the olive oil samples were treated as having unknown group membership. The table is sub-divided into the three regions to illustrate that no out-of-region misclassifications occurred.

	1	2	3	4	5	6	7	8	9
North Apulia	7	0	0	1	0	0	0	0	0
Calabria	0	19	0	0	0	0	0	0	0
South Apulia	0	0	67	1	0	0	0	0	0
Sicily	1	0	0	11	0	0	0	0	0
Inland Sardinia	0	0	0	0	21	1	0	0	0
Coastal Sardinia	0	0	0	0	1	10	0	0	0
East Liguria	0	0	0	0	0	0	16	1	0
West Liguria	0	0	0	0	0	0	0	16	0
Umbria	0	0	0	0	0	0	4	0	13

Notably, the chosen number of factors $q = 4$ is at the end of the range we used ($q = 1, \dots, 4$). Therefore, it is good practice to run `pgmmEM()` again with an extended range; extending the range beyond $q = p/2 = 4$ requires setting `relax=TRUE`.

```
olive_class2<-pgmmEM(x,cls,qmax=6,qmin=4,Gmax=9,Gmin=9,modelSubset=c("CUC",
"CUCU","CUU"),relax=TRUE)
```

Based on the labelled and unlabelled data provided, the best model (BIC) for the range of factors and components used is a CUU model with $q = 5$ and $G = 9$. The BIC for this model is -5833.272.

```
cls_ind<-(cls==0)
table(olive[cls_ind,2],olive_class2$map[cls_ind])
```

```

      1  2  3  4  5  6  7  8  9
1  8  0  0  0  0  0  0  0  0
2  0 19  0  0  0  0  0  0  0
```

```

3  0  0 67  1  0  0  0  0  0
4  1  0  0 11  0  0  0  0  0
5  0  0  0  0 20  2  0  0  0
6  0  0  0  0  1 10  0  0  0
7  0  0  0  0  0  0 15  2  0
8  0  0  0  0  0  0  0 16  0
9  0  0  0  0  0  0  2  0 15

```

This time, a $q = 5$ factor model was selected and we misclassified just 9/190 samples. Again, no out-of-region misclassifications occurred.

4.6 Model-based classification of olive oil data into regions

To illustrate that `pgmmEM()` can run model-based classification for values of G exceeding the number of known groups, consider the following analysis of the olive oil data. Here, we take two-thirds of the oils from Southern Italy and Sardinia as having known group memberships with regard to region ($g \in \{1, 2\}$). All oils from Northern Italy are treated as having unknown group memberships. We then use the CUU model for model-based classification for $G = 2, 3, 4$.

```

data("olive")
x<-olive[,-c(1,2)]
x<-scale(x)
cls2<-olive[,1]
for(i in 1:dim(olive)[1]){if(i%%3==0||i>420){cls2[i]<-0}}
olive_class3<-pgmmEM(x,cls2,qmax=6,Gmax=4,Gmin=2,modelSubset=c("CUU"),
relax=TRUE)

```

Based on the labelled and unlabelled data provided, the best model (BIC) for the range of factors and components used is a CUU model with $q = 5$ and $G = 3$. The BIC for this model is -6487.319.

```
cls_ind2<-(cls2==0)
table(olive[cls_ind2,1],olive_class3$map[cls_ind2])
```

	1	2	3
1	107	0	0
2	0	33	1
3	0	4	147

Despite providing known labels for just two of the three regions, a three component model was selected. Again, the classification performance is excellent, with only 5/292 observations misclassified. This example, and those in Section 4.5, illustrate the efficacy of the semi-supervised approach described by McNicholas (2010).

5 Summary

The R package `pgmm` offers an implementation of the PGMM model-based clustering and classification techniques introduced by McNicholas and Murphy (2008, 2010) and McNicholas (2010). Some features of the code, which is essentially an R wrapper for C, have been explained and examples given. These examples elucidate how the code may be used for model-based clustering and model-based classification. The ability to choose between random starts, k -means starts, or to pass user-specified starting values to `pgmmEM()` gives the user great flexibility. In addition to illustrating various settings within the code, the examples in Section 4 introduce the three data sets contained within `pgmm`: `coffee`, `wine`,

and `olive`. One of the examples is used to illustrate that while the BIC is generally considered the most effective model selection technique for mixture model selection within the literature, the model with the highest BIC is not necessarily the best classifier. The `pgmm` package gives the user the option to use the BIC or the ICL for model selection. A clustering example with higher dimensional data is also presented as well as a model-based classification application where the true number of groups exceeds the number of known groups.

Acknowledgements

This work was supported by a Discovery Grant from the Natural Sciences and Engineering Research Council of Canada (McNicholas); the University Research Chair in Computational Statistics from the University of Guelph (McNicholas); an Early Researcher Award from the Ontario Ministry of Research and Innovation (McNicholas); and a Basic Research Grant (04/BR/M0057) and a Research Frontiers Grant (2007/RFP/MATF281) from Science Foundation Ireland (Murphy). Most of the computing equipment used for the development of this software was purchased through grants from the Canada Foundation for Innovation Leaders Opportunity Fund and the Ontario Ministry for Research and Innovation Small Infrastructure Fund (McNicholas).

References

- Aitken, A. C. (1926). On bernoulli's numerical solution of algebraic equations. *Proceedings of the Royal Society of Edinburgh* 46, 289–305.
- Andrews, J. L. and P. D. McNicholas (2011a). Extending mixtures of multivariate t-factor analyzers. *Statistics and Computing* 21(3), 361–373.

- Andrews, J. L. and P. D. McNicholas (2011b). Mixtures of modified t-factor analyzers for model-based clustering, classification, and discriminant analysis. *Journal of Statistical Planning and Inference* 141(4), 1479–1486.
- Biernacki, C., G. Celeux, and G. Govaert (2000). Assessing a mixture model for clustering with the integrated completed likelihood. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22(7), 719–725.
- Böhning, D., E. Dietz, R. Schaub, P. Schlattmann, and B. Lindsay (1994). The distribution of the likelihood ratio for mixtures of densities from the one-parameter exponential family. *Annals of the Institute of Statistical Mathematics* 46, 373–388.
- Celeux, G. and G. Govaert (1995). Gaussian parsimonious clustering models. *Pattern Recognition* 28(5), 781–793.
- Dempster, A. P., N. M. Laird, and D. B. Rubin (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B* 39(1), 1–38.
- Forina, M., C. Armanino, M. Castino, and M. Ubigli (1986). Multivariate data analysis as a discriminating method of the origin of wines. *Vitis* 25, 189–201.
- Forina, M., C. Armanino, S. Lanteri, and E. Tiscornia (1983). Classification of olive oils from their fatty acid composition. In H. Martens and H. Russwurm Jr (Eds.), *Food Research and Data Analysis*, pp. 189–214. London: Applied Science Publishers.
- Forina, M. and E. Tiscornia (1982). Pattern recognition methods in the prediction of Italian olive oil origin by their fatty acid content. *Annali di Chimica* 72, 143–155.
- Ghahramani, Z. and G. E. Hinton (1997). The EM algorithm for factor analyzers. Technical Report CRG-TR-96-1, University Of Toronto, Toronto.

- Keribin, C. (2000). Consistent estimation of the order of mixture models. *Sankhyā. The Indian Journal of Statistics. Series A* 62(1), 49–66.
- Lindsay, B. G. (1995). Mixture models: Theory, geometry and applications. In *NSF-CBMS Regional Conference Series in Probability and Statistics*, Volume 5. California: Institute of Mathematical Statistics: Hayward.
- Lopes, H. F. and M. West (2004). Bayesian model assessment in factor analysis. *Statistica Sinica* 14, 41–67.
- McLachlan, G. J. and D. Peel (2000a). *Finite Mixture Models*. New York: John Wiley & Sons.
- McLachlan, G. J. and D. Peel (2000b). Mixtures of factor analyzers. In *Proceedings of the Seventh International Conference on Machine Learning*, San Francisco, pp. 599–606. Morgan Kaufmann.
- McNicholas, P. D. (2010). Model-based classification using latent Gaussian mixture models. *Journal of Statistical Planning and Inference* 140(5), 1175–1181.
- McNicholas, P. D. (2011). On model-based clustering, classification, and discriminant analysis. *Journal of the Iranian Statistical Society* 10(2), 181–199.
- McNicholas, P. D., K. R. Jampani, A. F. McDaid, T. B. Murphy, and L. Banks (2011). *pgmm: Parsimonious Gaussian Mixture Models*. R package version 1.0.
- McNicholas, P. D. and T. B. Murphy (2008). Parsimonious Gaussian mixture models. *Statistics and Computing* 18(3), 285–296.
- McNicholas, P. D. and T. B. Murphy (2010). Model-based clustering of microarray expression data via latent Gaussian mixture models. *Bioinformatics* 26(21), 2705–2712.

- McNicholas, P. D., T. B. Murphy, A. F. McDaid, and D. Frost (2010). Serial and parallel implementations of model-based clustering via parsimonious Gaussian mixture models. *Computational Statistics and Data Analysis* 54(3), 711–723.
- Meng, X.-L. and D. B. Rubin (1993). Maximum likelihood estimation via the ECM algorithm: A general framework. *Biometrika* 80, 267–278.
- Meng, X.-L. and D. van Dyk (1997). The EM algorithm — an old folk song sung to a fast new tune (with discussion). *Journal of the Royal Statistical Society. Series B* 59(3), 511–567.
- R Development Core Team (2011). *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing.
- Schwarz, G. (1978). Estimating the dimension of a model. *The Annals of Statistics* 6(2), 461–464.
- Spearman, C. (1904). The proof and measurement of association between two things. *American Journal of Psychology* 15, 72–101.
- Streuli, H. (1973). Der heutige stand der kaffeechemie. In *Association Scientifique Internationale du Cafe, 6th International Colloquium on Coffee Chemistry*, Bogatá, Columbia, pp. 61–72.
- Tipping, T. E. and C. M. Bishop (1999). Mixtures of probabilistic principal component analysers. *Neural Computation* 11(2), 443–482.
- Woodbury, M. A. (1950). *Inverting modified matrices*. Statistical Research Group, Memorandum Report 42. Princeton University, Princeton, New Jersey.