Australian & New Zealand Journal of Statistics

Check for updates

Aust. N. Z. J. Stat. (2024) doi: 10.1111/anzs.12413

Semi-supervised Gaussian mixture modelling with a missing-data mechanism in R

Ziyang Lyu[1,*], Daniel Ahfock[2], Ryan Thompson[1,3] and Geoffrey J. McLachlan[2]

University of New South Wales, University of Queensland and Commonwealth Scientific and Industrial Research Organisation

Summary

Semi-supervised learning is being extensively applied to estimate classifiers from training data in which not all the labels of the feature vectors are available. We present gmmsslm, an R package for estimating the Bayes' classifier from such partially classified data in the case where the feature vector has a multivariate Gaussian (normal) distribution in each of the pre-defined classes. Our package implements a recently proposed Gaussian mixture modelling framework that incorporates a missingness mechanism for the missing labels in which the probability of a missing label is represented via a logistic model with covariates that depend on the entropy of the feature vector. Under this framework, it has been shown that the accuracy of the Bayes' classifier formed from the Gaussian mixture model fitted to the partially classified training data can even have lower error rate than if it were estimated from the sample completely classified. This result was established in the particular case of two Gaussian classes with a common covariance matrix. Here we focus on the effective implementation of an algorithm for multiple Gaussian classes with arbitrary covariance matrices. A strategy for initialising the algorithm is discussed and illustrated. The new package is demonstrated on some real data.

Key words: Bayes' rule; entropy; mixture model; partially classified sample; semi-supervised learning.

1. Introduction

Classifiers such as neural networks often achieve their strong performance through a completely supervised learning approach, which requires a fully classified (labelled) dataset. However, missing labels often occur in practice because of the difficulty determining the true label for an observation (the feature vector). For example, in medicine and defence, images can often be correctly classified only by a limited number of experts in the field. Hence, a training sample might not be completely classified, with images difficult to classify left without their class labels. Moreover, in medicine, there might be scans that can be diagnosed confidently only after an invasive procedure, possibly regarded as unethical to perform at the time.

*Author to whom correspondence should be addressed.

[1]UNSW Data Science Hub and School of Mathematics and Statistics, University of New South Wales, Sydney 2052, NSW Australia. e-mail: ziyang.lyu@unsw.edu.au

[2]School of Mathematics and Physics, University of Queensland, Brisbane, QLD 4072, Australia.

[3]Data61, Commonwealth Scientific and Industrial Research Organisation, Eveleigh 2015, NSW Australia.

Acknowledgements. Open access publishing facilitated by University of New South Wales, as part of the Wiley - University of New South Wales agreement via the Council of Australian University Librarians.

Semi-supervised learning (Chapelle, Schölkopf, & Zien 2006) addresses the issue of missing labels. Classic approaches that belong to the semi-supervised learning paradigm include generative models (Pan *et al*. 2006; Kim & Kang 2007; Fujino, Ueda, & Saito 2008), graph-based models (Blum & Mitchell 1998; Szummer & Jaakkola 2001; Zhou *et al*. 2003) and semi-supervised support vector machines (Vapnik 1998; Joachims 1999; Lanckriet *et al*. 2004). Gaussian mixture models are a fundamental class of statistical models particularly relevant to semi-supervised learning. Given a partially classified sample, the traditional optimisation objective is a joint likelihood over the labelled and unlabelled data. This problem of maximum likelihood (ML) with missing data is amenable to the expectation-maximisation algorithm of Dempster, Laird, & Rubin (1977); see, for example, McLachlan & Peel (2000), McLachlan & Krishnan (2008) and the recent review of McLachlan, Lee, & Rathnayake (2019) on finite mixture models. Although Gaussian mixture models in the semi-supervised setting are now well studied (Pan *et al*. 2006; Kim & Kang 2007; Côme *et al*. 2009; Huang & Hasegawa-Johnson 2010; Szczurek *et al*. 2010), there is often a critical assumption that the missing-label process can be ignored for likelihood-based inference (McLachlan 1975, 1977; Ganesalingam & McLachlan 1978; Chawla & Karakoulas 2005).

Recently, Ahfock & McLachlan (2020) introduced a novel approach that treats labels of unclassified observations as missing data, leveraging a framework for handling missingness, as in the groundbreaking work by Rubin (1976) on incomplete data analysis. Ahfock & McLachlan (2020) conducted an asymptotic analysis to demonstrate that a partially classified sample can provide more valuable information than a fully labelled sample, specifically in the two-class Gaussian homoscedastic model; see also the review by Ahfock & McLachlan (2023). By building upon their framework, we propose to model the probability of a missing label as a logistic regression model where the mean depends on a single covariate equal to an entropy-based measure. This approach enables the implementation of an algorithm for estimating the Bayes' classifier through the full likelihood, catering to multiple Gaussian classes with arbitrary covariance matrices. Given the complexities that emerged in the derived expressions, transitioning from the specific two-class homoscedastic Gaussian model to this expanded framework poses a significant technical challenge. Our development and presentation of an R package designed for semi-supervised learning from a statistical viewpoint marks an important advancement. Its potential spans various statistical analyses and application scenarios.

In the context of our more general framework, we introduce the R package gmmsslm (Gaussian mixture model-based semi-supervised learning with a missing-data mechanism), which is available open-source on the Comprehensive R Archive Network at https://cran.r-project.org/package=gmmsslm. This package implements three distinct Gaussian mixture modelling approaches: (i) partially classified samples, taking into account the missing-data mechanism; (ii) partially classified samples, disregarding the missing-data mechanism; and (iii) fully classified samples. Although various packages exist for estimating mixture models, such as bgmm (Biecek *et al*. 2012), EMMIX (McLachlan *et al*. 1999), flexmix (Grün & Leisch 2007), mclust (Fraley & Raftery 2007), mixtools (Benaglia *et al*. 2009) and Rmixmod (Lebret *et al*. 2015), none accommodates a missing-data mechanism. In gmmsslm, the missingness mechanism is specified through a multinomial logistic regression concerning the entropy of feature vectors. The package applies to an arbitrary number of classes possessing multivariate Gaussian distributions with potentially dissimilar covariance matrices.

This paper is structured as follows. Section 2 provides a concise review of the statistical model used in gmmsslm. Section 3 describes how users can implement gmmsslm in applications. Section 4 illustrates a practical application of the package. Finally, Section 5 concludes the paper.

2. Gaussian mixture model with a missing-data mechanism

2.1. Gaussian mixture model

We let $\boldsymbol{Y}$ be a p-dimensional vector of features on an entity to be assigned to one of g pre-defined classes $C_1, \dots, C_g$ occurring in proportions $\pi_1, \dots, \pi_g$, where $\sum_{i=1}^{g} \pi_i = 1$. The random variable $\boldsymbol{Y}$ corresponding to the realisation $\boldsymbol{y}$ is assumed to have density $f_i(\boldsymbol{y}; \boldsymbol{\omega}_i)$ with a vector $\boldsymbol{\omega}_i$ of unknown parameters in Class C_i $(i = 1, \dots, g)$. The vector of all unknown parameters is given by $\boldsymbol{\theta} = (\pi_1, \dots, \pi_{g-1}, \boldsymbol{\omega}_1^\top, \dots, \boldsymbol{\omega}_g^\top)^\top$.

In the sequel, we assume that the class-conditional densities of $\boldsymbol{y}$ are multivariate Gaussian with

$$f_i(\boldsymbol{y}; \boldsymbol{\omega}_i) = \phi(\boldsymbol{y}; \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) \quad (i = 1, \dots, g),$$

where $\phi(\boldsymbol{y}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$ denotes the p-variate Gaussian density function with mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$. The vector $\boldsymbol{\theta}$ of all unknown parameters now consists of the elements of the means $\boldsymbol{\mu}_i$ and the $p(p+1)/2$ distinct elements of the covariance matrices $\boldsymbol{\Sigma}_i$, along with the mixing proportions. In order to estimate $\boldsymbol{\theta}$, it is customary in practice to have available a training sample. We let $\boldsymbol{x}_{\mathrm{CC}} = (\boldsymbol{x}_1^\top, \dots, \boldsymbol{x}_n^\top)^\top$ contain n independent realisations of $\boldsymbol{X} = (\boldsymbol{Y}^\top, Z)^\top$ as the completely classified training data, where Z denotes the class membership of $\boldsymbol{Y}$, being equal to i if $\boldsymbol{Y}$ belongs to class C_i $(i = 1, \dots, g)$ and zero otherwise, and where $\boldsymbol{x}_j = (\boldsymbol{y}_j^\top, z_j)^\top$ $(j = 1, \dots, n)$. For a partially classified training sample $\boldsymbol{x}_{\mathrm{PC}}$ in semi-supervised learning, we introduce the missing-label indicator m_j, which equals 1 if z_j is missing and 0 if it is available $(j = 1, \dots, n)$. Thus, $\boldsymbol{x}_{\mathrm{PC}}$ consists of those observations $\boldsymbol{x}_j$ in $\boldsymbol{x}_{\mathrm{CC}}$ with $m_j = 0$, but only the feature vector $\boldsymbol{y}_j$ in $\boldsymbol{x}_{\mathrm{CC}}$ if $m_j = 1$ (i.e. the label z_j is missing). The presence of unclassified feature observations in the training data (i.e. features with missing labels) necessitates the consideration and fitting of the unconditional density of $\boldsymbol{Y}$, which is given by the g-component Gaussian mixture density

$$f(\boldsymbol{y}_j; \boldsymbol{\theta}) = \sum_{i=1}^{g} \pi_i \phi\big(\boldsymbol{y}_j; \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i\big).$$

The optimal (Bayes') rule of allocation $R(\boldsymbol{y}; \boldsymbol{\theta})$ assigns an entity with feature vector $\boldsymbol{y}$ to Class C_k, that is, $R(\boldsymbol{y}; \boldsymbol{\theta}) = k$ if $k = \operatorname{argmax}_i \tau_i(\boldsymbol{y}; \boldsymbol{\theta})$, where $\tau_i(\boldsymbol{y}; \boldsymbol{\theta}) = \pi_i f_i(\boldsymbol{y}; \boldsymbol{\omega}_i) / \sum_{h=1}^{g} \pi_h f_h(\boldsymbol{y}; \boldsymbol{\omega}_h)$ is the posterior probability that the entity belongs to Class C_i given $\boldsymbol{Y} = \boldsymbol{y}$ $(i = 1, \dots, g)$.

We define the log likelihoods

$$\log L_{\mathrm{C}}(\boldsymbol{\theta}) = \sum_{j=1}^{n} (1 - m_j) \sum_{i=1}^{g} z_{ij} \log\big\{\pi_i f_i(\boldsymbol{y}_j; \boldsymbol{\omega}_i)\big\}, \tag{1}$$

$$\log L_{\mathrm{UC}}(\boldsymbol{\theta}) = \sum_{j=1}^{n} m_j \log\left\{\sum_{i=1}^{g} \pi_i f_i(\boldsymbol{y}_j; \boldsymbol{\omega}_i)\right\},$$

$$\log L_{\mathrm{PC}}^{(\mathrm{ig})}(\boldsymbol{\theta}) = \log L_{\mathrm{C}}(\boldsymbol{\theta}) + \log L_{\mathrm{UC}}(\boldsymbol{\theta}). \tag{2}$$

In (1), $z_{ij} = 1$ if $z_j = i$ and $z_{ij} = 0$ otherwise. If one ignores the 'missingness' of the class labels, $L_{\mathrm{C}}(\boldsymbol{\theta})$ and $L_{\mathrm{UC}}(\boldsymbol{\theta})$ are the likelihood functions formed from the classified and unclassified data, respectively. The likelihood function $L_{\mathrm{PC}}^{(\mathrm{ig})}(\boldsymbol{\theta})$ is formed from the partially classified sample $\boldsymbol{x}_{\mathrm{PC}}$, ignoring the missing-data mechanism for the labels. The likelihood $L_{\mathrm{CC}}(\boldsymbol{\theta})$ for the completely classified sample $\boldsymbol{x}_{\mathrm{CC}}$ is recovered from (1) by taking all $m_j = 0$.

2.2. Missing-data mechanism

In the present context, it is appropriate to dispense with the missing-data mechanism when performing likelihood inference in situations where the missing labels can be viewed as missing completely at random (MCAR) in the framework proposed by Rubin (1976) for handling missingness in incomplete data analysis. The reader is referred to Mealli & Rubin (2015) for precise definitions of MCAR and its less restrictive version of missing at random (MAR). The MCAR case here holds if the missingness of labels is independent of both the features and labels, while with the MAR case this missingness is allowed to depend on the features but not the labels. As highlighted by McLachlan & Gordon (1989), one can legitimately ignore missingness in certain MAR situations, such as when dealing with truncated features. However, for the MAR scenario being discussed here, one cannot ignore the missingness in carrying out the likelihood analysis. Indeed, the use of the missingness provides a way of improving the performance of the classifier to be formed.

Ahfock & McLachlan (2020) noted that it is common in practice for unlabelled images (i.e. the features with missing labels) to fall in regions of the feature space where there is class overlap. This finding led them to argue that the unlabelled observations can carry additional information that can be used to improve the efficiency of the parameter estimation of $\boldsymbol{\theta}$. Additional theoretical motivation is available in Appendix A. They noted that in these situations the difficulty of classifying an observation can be quantified using the Shannon entropy of an entity with feature vector $\boldsymbol{y}$, which is defined by

$$e(\boldsymbol{y}_j;\boldsymbol{\theta}) = -\sum_{i=1}^{g} \tau_i(\boldsymbol{y}_j;\boldsymbol{\theta}) \log \tau_i(\boldsymbol{y}_j;\boldsymbol{\theta}).$$

Let M_j denote the random variable corresponding to the realised value m_j of the missing-label indicator for the observation $\boldsymbol{y}_j$. The missingness mechanism of Rubin (1976) is specified in the present context as

$$\Pr\{M_j = 1|\boldsymbol{y}_j, z_j\} = \Pr\{M_j = 1|\boldsymbol{y}_j\} = q(\boldsymbol{y}_j;\boldsymbol{\theta},\boldsymbol{\xi}),$$

where $\boldsymbol{\xi} = (\xi_0, \xi_1)^\top$ is distinct from $\boldsymbol{\theta}$. The conditional probability $q(\boldsymbol{y}_j;\boldsymbol{\theta},\boldsymbol{\xi})$ is taken to be a logistic function of the Shannon entropy $e(\boldsymbol{y}_j;\boldsymbol{\theta})$, yielding

$$q(\boldsymbol{y}_j;\boldsymbol{\theta},\boldsymbol{\xi}) = \frac{\exp\{\xi_0 + \xi_1 \log e(\boldsymbol{y}_j;\boldsymbol{\theta})\}}{1 + \exp\{\xi_0 + \xi_1 \log e(\boldsymbol{y}_j;\boldsymbol{\theta})\}}. \tag{3}$$

In the special case of $g = 2$ with equal covariance matrices, the negative log entropy in the conditional probability (3) can be replaced by the square of the discriminant function. Details can be found in Appendix B.

2.3. Full likelihood function based on a missingness mechanism

We let $\boldsymbol{\Psi} = \left(\boldsymbol{\theta}^\top, \boldsymbol{\xi}^\top\right)^\top$ be the vector of all unknown parameters. Henceforth, we let f be a generic symbol for a density or probability function where appropriate. To construct the full likelihood function $L_{\mathrm{PC}}^{(\mathrm{full})}(\boldsymbol{\Psi})$ from the partially classified sample $\boldsymbol{x}_{\mathrm{PC}}$, we need expressions for

$$f(\boldsymbol{y}_j, z_j, m_j = 0) \quad \text{and} \quad f(\boldsymbol{y}_j, m_j = 1)$$

corresponding to the classified and unclassified observations $\boldsymbol{y}_j$, respectively. For a classified observation $\boldsymbol{y}_j$, it follows that

$$\begin{aligned} f(\boldsymbol{y}_j, z_j, m_j = 0) &= f(z_j) f(\boldsymbol{y}_j | z_j) \Pr\{M_j = 0 | \boldsymbol{y}_j, z_j\} \\ &= \prod_{i=1}^{g} \{\pi_i f_i(\boldsymbol{y}_j; \boldsymbol{\omega}_i)\}^{z_{ij}} \{1 - q(\boldsymbol{y}_j; \boldsymbol{\theta}, \boldsymbol{\xi})\}, \end{aligned}$$

while for an unclassified observation $\boldsymbol{y}_j$, we have

$$f(\boldsymbol{y}_j, m_j = 1) = f(\boldsymbol{y}_j) \Pr\{M_j = 1 | \boldsymbol{y}_j\} = \sum_{i=1}^{g} \pi_i f_i(\boldsymbol{y}_j; \boldsymbol{\omega}_i) q(\boldsymbol{y}_j; \boldsymbol{\theta}, \boldsymbol{\xi}).$$

The full likelihood can then be expressed as

$$L_{\mathrm{PC}}^{(\mathrm{full})}(\boldsymbol{\Psi}) = \prod_{j=1}^{n} \{f(\boldsymbol{y}_j, z_j, m_j = 0)\}^{1-m_j} \{f(\boldsymbol{y}_j, m_j = 1)\}^{m_j}.$$

Thus, the full log likelihood follows as

$$\log L_{\mathrm{PC}}^{(\mathrm{full})}(\boldsymbol{\Psi}) = \log L_{\mathrm{PC}}^{(\mathrm{ig})}(\boldsymbol{\theta}) + \log L_{\mathrm{PC}}^{(\mathrm{miss})}(\boldsymbol{\Psi}), \tag{4}$$

where $\log L_{\mathrm{PC}}^{(\mathrm{ig})}(\boldsymbol{\theta})$ is given in (2) and

$$\log L_{\mathrm{PC}}^{(\mathrm{miss})}(\boldsymbol{\Psi}) = \sum_{j=1}^{n} \left[(1 - m_j) \log\left\{1 - q(\boldsymbol{y}_j; \boldsymbol{\theta}, \boldsymbol{\xi})\right\} + m_j \log q(\boldsymbol{y}_j; \boldsymbol{\theta}, \boldsymbol{\xi})\right]$$

is the log likelihood formed on the basis of the missing-label indicators m_j.

Using the expectation conditional maximisation (ECM) algorithm, we obtain the ML estimations $\hat{\boldsymbol{\theta}}_{\mathrm{CC}}$, $\hat{\boldsymbol{\theta}}_{\mathrm{PC}}^{(\mathrm{ig})}$ and $\hat{\boldsymbol{\theta}}_{\mathrm{PC}}^{(\mathrm{full})}$ for $\boldsymbol{\theta}$ based on the log likelihoods $\log L_{\mathrm{CC}}(\boldsymbol{\theta})$, $\log L_{\mathrm{PC}}^{(\mathrm{ig})}(\boldsymbol{\theta})$ and $\log L_{\mathrm{PC}}^{(\mathrm{full})}(\boldsymbol{\Psi})$, respectively (see Appendix C for details). Notice that $\hat{\boldsymbol{\theta}}_{\mathrm{PC}}^{(\mathrm{full})}$ is a subvector of $\hat{\boldsymbol{\Psi}}_{\mathrm{PC}}^{(\mathrm{full})}$. Similarly, we let $R(\hat{\boldsymbol{\theta}}_{\mathrm{CC}})$, $R\left(\hat{\boldsymbol{\theta}}_{\mathrm{PC}}^{(\mathrm{ig})}\right)$ and $R\left(\hat{\boldsymbol{\theta}}_{\mathrm{PC}}^{(\mathrm{full})}\right)$ denote the estimated Bayes' rule obtained by plugging in the estimates $\hat{\boldsymbol{\theta}}_{\mathrm{CC}}$, $\hat{\boldsymbol{\theta}}_{\mathrm{PC}}^{(\mathrm{ig})}$ and $\hat{\boldsymbol{\theta}}_{\mathrm{PC}}^{(\mathrm{full})}$, respectively. The overall conditional error rate of the rule $R\left(\boldsymbol{y}; \hat{\boldsymbol{\theta}}_{\mathrm{CC}}\right)$ is then

$$\mathrm{err}(\hat{\boldsymbol{\theta}}_{\mathrm{CC}}; \boldsymbol{\theta}) = 1 - \sum_{i=1}^{g} \pi_i \Pr\left\{R(\boldsymbol{y}; \hat{\boldsymbol{\theta}}_{\mathrm{CC}}) = i \,|\, \hat{\boldsymbol{\theta}}_{\mathrm{CC}}, Z = i\right\}. \tag{5}$$

The corresponding conditional error rates $\mathrm{err}\left(\hat{\boldsymbol{\theta}}_{\mathrm{PC}}^{(\mathrm{ig})}; \boldsymbol{\theta}\right)$ of the rule $R\left(\boldsymbol{y}; \hat{\boldsymbol{\theta}}_{\mathrm{PC}}^{(\mathrm{ig})}\right)$ and $\mathrm{err}\left(\hat{\boldsymbol{\theta}}_{\mathrm{PC}}^{(\mathrm{full})}; \boldsymbol{\theta}\right)$ of the rule $R\left(\boldsymbol{y}; \hat{\boldsymbol{\theta}}_{\mathrm{PC}}^{(\mathrm{full})}\right)$ are defined likewise. The optimal error rate $\mathrm{err}(\boldsymbol{\theta})$ follows by replacing $\hat{\boldsymbol{\theta}}_{\mathrm{CC}}$ with $\boldsymbol{\theta}$ in (5).

3. Package description

In this section, we describe the gmmsslm package. The package has been crafted to implement three distinct Gaussian mixture models, which can be fit using the function `gmmsslm()`. These models utilise different log likelihood functions, namely (1), (2) and (4). Furthermore, the package provides several auxiliary functions including the Bayes rule classifier `bayesclassifier()`, conditional error rate `erate()` and Shannon entropy `get_entropy()`. The general syntax for using these functions is illustrated below:

```
gmmsslm(dat, zm, paralist, xi = NULL, type)
bayesclassifier(dat, p, g, paralist)
erate(dat, p, g, paralist, clust)
get_entropy(dat, n, p, g, paralist)
```

Details for some of the main command arguments are as follows:

- `n`: Number of observations.
- `p`: Dimension of observation vector.
- `g`: Number of multivariate normal classes.
- `dat`: An $n \times p$ matrix where each row represents an individual observation.
- `zm`: An n-dimensional vector containing the class labels including the missing label denoted as `NA`.
- `pi`: A g-dimensional vector for the initial values of the mixing proportions.
- `mu`: A $p \times g$ matrix for the initial values of the location parameters.
- `sigma`: A $p \times p$ matrix, or a $p \times p \times g$ array, for the initial values of the covariance matrices. The model is fit with a common covariance matrix if `sigma` is a $p \times p$ covariance matrix, otherwise the model is fit with unequal covariance matrices.
- `paralist`: A list containing `pi`, `mu` and `sigma`. When `paralist` is provided, individual inputs for `pi`, `mu` and `sigma` are not required.
- `xi`: A two-dimensional vector containing the initial values of the coefficients in the logistic function of the Shannon entropy. The default value of `xi` is `NULL`, except when `type = 'full'` is selected.
- `type`: One of three types of Gaussian mixture models as follows: `'full'` fits the model given by (4) to a partially classified sample on the basis of the full likelihood by taking into account the missing-data mechanism; `'ign'` fits the model given by (2) to a partially classified sample based on the likelihood that the missing-data mechanism is ignored; and `'com'` fits the model given by (1) to a completed classified sample.
- `clust`: An n-dimensional vector of the class partition.

The main function `gmmsslm()` provides a comprehensive summary output, detailed as follows:

- `Likelihood`: Value of the objective likelihood.
- `VarianceStructure`: Describes the structure of the covariance matrix associated with the Gaussian mixture model fitting. Possible structures include ‘A common covariance matrix’ and ‘Unequal covariance matrices’.
- `Convergence`: State of convergence.
- `Iteration`: Number of iterations executed.
- `TotalObservation`: Total number of observations.

- `Dimension`: Dimension of the data.
- `ModelType`: Type of model fit. The ‘full’ model type considers a partially classified sample based on the full likelihood, accounting for the missing-data mechanism. The ‘ign’ type fits the model to a partially classified sample, dismissing the missing-data mechanism. The ‘com’ type is dedicated to a completely classified sample.
- `Parameters:` Estimated parameters, which include $\boldsymbol{\pi}$, $\boldsymbol{\mu}$, $\boldsymbol{\Sigma}$ and $\boldsymbol{\xi}$ (relevant only if `ModelType = ‘full’`).

4. Example: Gastrointestinal lesions data

In this section, we demonstrate the practical application of the gmmsslm package using the gastrointestinal lesions data from Mesejo *et al*. (2016). The dataset comprises 76 colonoscopy videos, the histology (classification ground truth) and the opinions of the endoscopists (four experts and three beginners). White light- and narrow-band imaging methods were used to classify whether the lesions were benign or malignant. Each of the $n = 76$ observations consists of four selected features extracted from the colonoscopy videos. A panel of seven endoscopists viewed the videos to give their opinion as to whether each patient needed resection (malignant) or no resection (benign). We formed our partially classified sample as follows. Feature vectors for which all seven endoscopists agreed were taken to be classified with labels specified either as 1 (resection) or 2 (no resection) using the ground truth labels. Observations for which there was no total agreement among the endoscopists were taken as having missing labels, denoted by `NA`.

We begin by loading the gmmsslm package. The `gastro_data` dataset from gmmsslm is then loaded. The feature matrix is denoted as `X`, where `n` represents the total number of observations and `p` their dimensionality. We also define `g = 2` as the number of classes.

```
library(gmmsslm)
data('gastro_data')
X <- as.matrix(gastro_data[,1:4])
n <- nrow(X)
p <- ncol(X)
g <- 2
```

An initial pair plot is constructed to visualise the locations of the labelled and unlabelled data.

```
classagree <- ifelse(is.na(gastro_data$class_agreement), 3,
                     gastro_data$class_agreement)
shapevec <- plyr::mapvalues(unclass(classagree), 1:3,
           to = c(16, 17, 15))
colvec <- plyr::mapvalues(unclass(classagree), 1:3,
           to = c('BLUE', 'RED', 'BLACK'))
pairs(X, pch = shapevec, col = colvec)
```

Figure 1 shows a plot of the data with the class labels of the feature vectors. The black squares denote the unlabelled observations, red triangles denote the benign observations

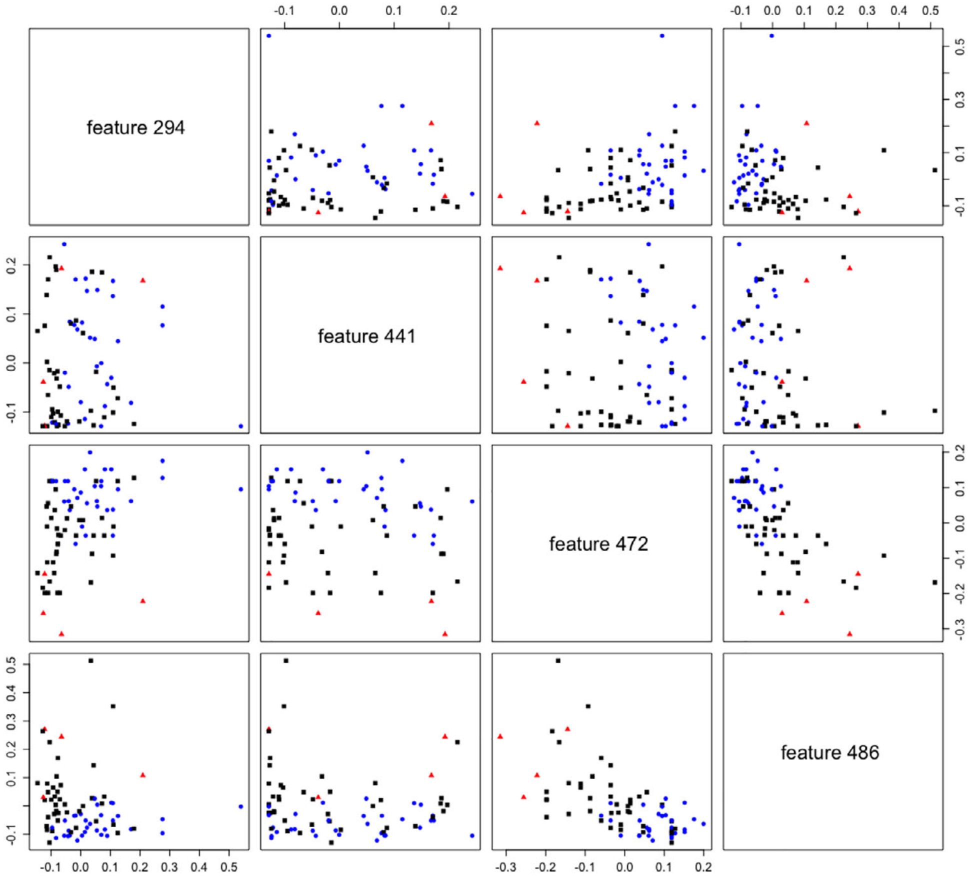

Figure 1. Gastrointestinal dataset. The black squares correspond to the unlabelled observations, the red triangles denote the benign observations and the blue circles denote the malignant observations. Observations are treated as unlabelled if fewer than seven endoscopists assigned the same class label to the feature vector.

and blue circles denote the malignant observations. The unlabelled observations tend to be located in regions of class overlap.

We can now verify if the incomplete data is appropriate for modelling with our proposed approach. Specifically, we need to ascertain whether there exists a missing-data mechanism in the missing labels. The function `plot_missingness()` is employed for this purpose, producing a box plot of the initial estimated entropies for labelled and unlabelled groups. Furthermore, it offers a plot showcasing a Nadaraya–Watson kernel estimate for the probability of missing labels against the negative log entropy. The `plot_missingness()` function requires initial estimates of $\boldsymbol{\pi}$, $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$. To acquire these estimates, we utilise `initialvalue(dat, zm, g, ncov)`. Note that the `ncov` parameter specifies the structure of `sigma`. By default, `ncov = 2`. When set to `ncov = 1`, it denotes a common covariance matrix, whereas `ncov = 2` signifies unequal covariance matrices.

```
zm <- gastro_data$class_agreement
inits <- initialvalue(dat = X, zm = zm, g = g, ncov = 2)
plot_missingness(dat = X, g = 2, parlist = inits, zm = zm,
        bandwidth = 3, range.x = c(0, 4), ylim = c(0.2, 0.8),
        kernel = 'normal')
```

Figure 2a compares the box plots of the estimated entropies in the labelled and unlabelled groups. Figure 2b presents the Nadaraya–Watson kernel estimate of the probability of missing labels. From Figure 2a, we find that the unlabelled observations typically have higher entropy than the labelled observations. Figure 2b shows that the estimated probability of a missing class label increases as the log entropy increases. This relation is in accordance with (3). The higher the entropy of a feature vector, the higher the probability of its class label being unavailable.

To process the data with the Gaussian mixture model with a missing-data mechanism, we utilise the `gmmsslm()` function and set `type = 'full'`. Prior to this, initial values for ξ_0 and ξ_1 are essential. We leverage the `glm()` function from the base package stats as follows:

```
en <- get_entropy(dat = X, n = n, p = p, g = g,
     paralist = inits)
m <- ifelse(gastro_data$missinglabel_indicator == 1, 0, 1)
xi_inits <- coef(glm(m ~ en, family = 'binomial'))
fullfit <- gmmsslm(dat = X, zm = zm, paralist = inits,
   xi = xi_inits, type = 'full')
```

The `gmmsslm()` function yields a `gmmsslm` object as its output. To extract a comprehensive summary from the fitted model, we can employ the `summary()` function. This summary reports various aspects, including the likelihood value, fitted variance

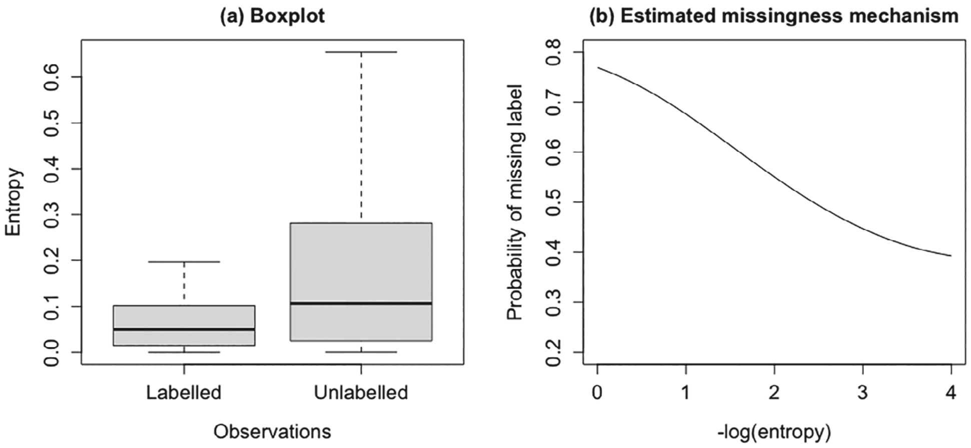

Figure 2. Analysing the gastrointestinal dataset concerning the relationship between entropy and observations, both labelled and unlabelled.

structure, convergence status, number of iterations, total number of observations, dimension, model type and estimated parameters.

```
summary(fullfit)
```

```
   Table:
                        [,1]
   Likelihood           "-244.9371"
   VarianceStructure    "Unequal covariance matrices"
   Convergence          "1"
   Iteration            "205"
   TotalObservation     "76"
   Dimension            "4"
   ModelType            "full"

   Parameters:

   $pi
   [1] 0.7758885 0.2241115

   $mu
                   [,1]        [,2]
   [1,]  0.018823669 -0.06329461
   [2,]  0.002130841 -0.00751177
   [3,]  0.050076855 -0.16765507
   [4,] -0.042663822  0.14306282

   $sigma
   , , 1

                    [,1]          [,2]         [,3]          [,4]
   [1,]  1.302248e-02  0.0002572603  0.002515218 -8.461135e-05
   [2,]  2.572603e-04  0.0128121495 -0.001279286 -6.947834e-04
   [3,]  2.515218e-03 -0.0012792861  0.005025436 -1.812703e-03
   [4,] -8.461135e-05 -0.0006947834 -0.001812703  3.683971e-03

   , , 2

                  [,1]          [,2]           [,3]         [,4]
   [1,]  0.0086012175  0.0009785018 -0.0004327248  0.004907910
   [2,]  0.0009785018  0.0147164175 -0.0051017162 -0.003288298
   [3,] -0.0004327248 -0.0051017162  0.0039913872  0.001410450
   [4,]  0.0049079101 -0.0032882976  0.0014104502  0.018480950

   $xi
   (Intercept)          en
    1.8276442   0.1434896
```

The initial values have minimal impact on the ECM algorithm's outcomes in this instance. For further exploration of this aspect, interested readers can refer to Data S1 containing relevant code for their own sensitivity analysis. Additionally, we can utilise the `predict()` function to obtain predicted labels derived from the fit by `gmmsslm()`. This function returns the predicted labels for the unclassified data initially input into `gmmsslm()`.

```
> predict(fullfit)
  [1] 1 1 1 1 1 1 1 1 1 2 1 2 1 1 2 1 1 2 1 1 1 1 2 2 2 2
  [27] 2 2 1 2 1 2 1 2 2 1 1 1 1 1 1
```

Subsequently, we use the function `paraextract()` to extract the parameter list from the fit by `gmmsslm()`; then we compute the conditional error rate for the unlabelled data.

```
X_ul <- X[m == 1,]
labeltruth = vector(length = n)
labeltruth[gastro_data$‘ground truth’ == ’resection’] <- 1
labeltruth[gastro_data$‘ground truth’ == ’no-resection’] <- 2
clust_ul <- labeltruth[m == 1]
err_ful <- erate(dat = X[m == 1,], p, g,
    paralist = paraextract(fullfit), clust = clust_ul)
err_ful
[1] 0.2215959
```

We now use `gmmsslm()` to compare the performance of the Bayes classifier with the unknown parameter vector $\boldsymbol{\theta}$ estimated by $\hat{\boldsymbol{\theta}}_{\mathrm{CC}}$, $\hat{\boldsymbol{\theta}}_{\mathrm{PC}}^{(\mathrm{ig})}$ and $\hat{\boldsymbol{\theta}}_{\mathrm{PC}}^{(\mathrm{full})}$. Error rates are estimated using leave-one-out cross-validation.

```
clust_com <- clust_ign <- clust_ful <- numeric(n)
for(i in 1:n){
  comfit <- gmmsslm(dat = X[-i,], zm = labeltruth[-i],
            paralist = inits, type = ’com’)
  ignfit <- gmmsslm(dat = X[-i,], zm = zm[-i],
            paralist = inits, type = ’ign’)
  fullfit <- gmmsslm(dat = X[-i,], zm = zm[-i],
            paralist = inits, xi = xi_inits, type = ’full’)
  clust_com[i] <- bayesclassifier(dat = X[i,], p, g,
                paralist = paraextract(comfit))
  clust_ign[i] <- bayesclassifier(dat = X[i,], p, g,
                paralist = paraextract(ignfit))
  clust_ful[i] <- bayesclassifier(dat = X[i,], p, g,
                paralist = paraextract(fullfit))
  }
  1 - mean(clust_com == labeltruth)
  1 - mean(clust_ign == labeltruth)
  1 - mean(clust_ful == labeltruth)
```

The estimated conditional error rates are reported in Table 1. The classifier based on the estimates of the parameters using the full likelihood for the partially classified training

Table 1. Results for the Gastrointestinal dataset. The dataset has $n = 76$ observations, $p = 4$ features and $g = 2$ classes.

	n_{c} (classified)	n_{uc} (unclassified)	Error rate
$R(\hat{\boldsymbol{\theta}}_{\mathrm{PC}}^{(\mathrm{full})})$	35	41	0.158
$R(\hat{\boldsymbol{\theta}}_{\mathrm{PC}}^{(\mathrm{ig})})$	35	41	0.211
$R(\hat{\boldsymbol{\theta}}_{\mathrm{CC}})$	76	0	0.171

sample has lower estimated error rate than that of the rule that would be formed if the sample were completely classified.

5. Summary

The R package gmmsslm implements the semi-supervised learning approach proposed by Ahfock & McLachlan (2020) for estimating the Bayes' classifier from a partially classified training sample in which some of the feature vectors have missing labels. It uses a generative model approach whereby the joint distribution of the feature vector and its ground-truth label is adopted. Each of g pre-specified classes to which a feature vector can belong has the multivariate Gaussian distribution. The conditional probability that a feature vector has a missing label is formulated in a framework in which the missingness mechanism models this probability to depend on the entropy of the feature vector using a logistic model. The parameters in the Bayes classifier are estimated by ML via an ECM algorithm. The package applies to classes with equal or unequal covariance matrices in their multivariate Gaussian distributions. In application to a real-world medical dataset, the estimated error rate of the Bayes' classifier based on the partially classified training sample is lower than that of the Bayes classifier formed from a completely classified sample.

Appendix A

Under the model (3) for MAR labels in the case of the two-class homoscedastic Gaussian model, Ahfock & McLachlan (2020) derived the following theorem, which motivated the development of a package to implement this semi-supervised learning approach for possibly multiple classes with multivariate Gaussian distributions.

Theorem 1. *The Fisher information about $\boldsymbol{\beta}$ in the partially classified sample $\boldsymbol{x}_{PC}$ via the full likelihood function $L_{PC}^{(full)}(\boldsymbol{\Psi})$ can be decomposed as*

$$\boldsymbol{I}_{PC}^{(full)}(\boldsymbol{\beta}) = \boldsymbol{I}_{CC}(\boldsymbol{\beta}) - \gamma(\boldsymbol{\Psi})\boldsymbol{I}_{CC}^{(clr)}(\boldsymbol{\beta}) + \boldsymbol{I}_{PC}^{(miss)}(\boldsymbol{\beta}), \tag{A1}$$

where $\gamma(\boldsymbol{\Psi})$ is the proportion of missing labels, $\boldsymbol{I}_{CC}(\boldsymbol{\beta})$ is the information about $\boldsymbol{\beta}$ in the completely classified sample $\boldsymbol{x}_{CC}, \boldsymbol{I}_{CC}^{(clr)}(\boldsymbol{\beta})$ is the conditional information about $\boldsymbol{\beta}$ under the logistic regression model fitted to the class labels in $\boldsymbol{x}_{CC}$ and $\boldsymbol{I}_{PC}^{(miss)}(\boldsymbol{\beta})$ is the information about $\boldsymbol{\beta}$ in the missing-label indicators under the assumed logistic model for their distribution given their associated features in the partially classified sample $\boldsymbol{x}_{PC}$.

The expression (A1) for the Fisher information about the vector of discriminant function coefficients contains the additional term $\boldsymbol{I}_{\mathrm{PC}}^{(\mathrm{miss})}(\boldsymbol{\beta})$, arising from the additional information about $\boldsymbol{\beta}$ in the missing-label indicators m_j. This term has the potential to compensate for the loss of information in not knowing the true labels of those unclassified features in the partially unclassified sample. The compensation depends on the extent to which the probability of a missing label for a feature depends on its entropy. It follows that if

$$\boldsymbol{I}_{\mathrm{PC}}^{(\mathrm{miss})}(\boldsymbol{\beta}) > \gamma(\boldsymbol{\Psi})\boldsymbol{I}_{\mathrm{CC}}^{(\mathrm{clr})}(\boldsymbol{\beta}), \tag{A2}$$

there is an increase in the information about $\boldsymbol{\beta}$ from the partially classified sample over the information $\boldsymbol{I}_{\mathrm{CC}}(\boldsymbol{\beta})$ from the completely classified sample. The inequality in (A2) is used to mean that the difference of the left- and right-hand sides of the inequality is a positive definite matrix.

The reader is referred to equation (36) in Ahfock & McLachlan (2020) for a precise definition of the conditional information term $\boldsymbol{I}_{\mathrm{CC}}^{(\mathrm{clr})}(\boldsymbol{\beta})$. By deriving the asymptotic relative efficiency of the Bayes rule using the full ML estimate of $\boldsymbol{\beta}$, Ahfock & McLachlan (2020) showed that the asymptotic expected excess error rate using the partially classified sample $\boldsymbol{x}_{\mathrm{PC}}$ can be much lower than the corresponding excess rate using the completely classified sample $\boldsymbol{x}_{\mathrm{CC}}$. The contribution to the Fisher information from the missingness mechanism can be relatively high if ξ_1 is large, as the location of the unclassified features in the feature space provides information about regions of high uncertainty and, hence, where the entropy is high.

However, when extending to multiple classes with multivariate Gaussians, providing a similar asymptotic analysis becomes considerably more complex. We are inclined to believe that the foundational ideas of Theorem 1 should hold for the general case. However, a comprehensive analytical verification would be extensive and is beyond the scope of this paper. We recognise the importance of such a verification and are planning to address this in our future research, possibly by leveraging numerical analysis techniques.

Appendix B

Since the expression for the optimal error rate is complicated, it is difficult to give a theoretical analysis of likelihood inference based on a missing-data mechanism in the general case. In the particular case of $g = 2$ classes and under the assumption of equal covariance $\boldsymbol{\Sigma}_1 = \boldsymbol{\Sigma}_2 = \boldsymbol{\Sigma}$, Ahfock & McLachlan (2020) provided a Taylor series approximation for the logarithm of entropy as

$$\log\left\{e\left(\boldsymbol{y}_j;\boldsymbol{\theta}\right)\right\} = -\left[\log\{\log(2)\} + d^2\left(\boldsymbol{y}_j;\boldsymbol{\beta}\right)/\{8\log(2)\}\right] + O\left(d^4\left(\boldsymbol{y}_j;\boldsymbol{\beta}\right)\right),$$

where

$$d\left(\boldsymbol{y}_j;\boldsymbol{\beta}\right) = \beta_0 + \boldsymbol{\beta}_1^{\top}\boldsymbol{y}_j,$$

with

$$\beta_0 = \log(\pi_1/\pi_2) - \frac{1}{2}(\boldsymbol{\mu}_1 + \boldsymbol{\mu}_2)^{\top}\boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2) \quad \text{and} \quad \boldsymbol{\beta}_1 = \boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2).$$

This expression provides a linear relationship between the negative log entropy $\log\{e(\boldsymbol{y}_j;\boldsymbol{\theta})\}$ and the square of the discriminant function $d^2(\boldsymbol{y}_j;\boldsymbol{\beta})$. Therefore, the negative log entropy in the conditional probability (3) can be replaced by the square of the discriminant function

$$q(\boldsymbol{y}_j;\boldsymbol{\beta},\boldsymbol{\xi})=\frac{\exp\{\xi_0-\xi_1 d^2(\boldsymbol{y}_j;\boldsymbol{\beta})\}}{1+\exp\{\xi_0-\xi_1 d^2(\boldsymbol{y}_j;\boldsymbol{\beta})\}}.$$

The missing-label indicator based on the entropy ($g>2$) or the square of the discriminant function ($g=2$) can be generated using the function `rlabel(dat, pi, mu, sigma, xi)`. The element of the outputs represent a missing label when equal to 1 and an available label when equal to 0.

Appendix C

We apply the ECM algorithm of Meng & Rubin (1993) to compute the ML estimate of $\boldsymbol{\Psi}$ on the basis of the full likelihood $L_{\mathrm{PC}}^{(\mathrm{full})}(\boldsymbol{\Psi})$. The adopted ECM framework makes the obvious choice of declaring the 'missing' data to be the missing labels $\boldsymbol{z}_j$ for those features $\boldsymbol{y}_j$ with $m_j=1$.

E step: It handles the presence of the introduced missing data by forming on the $(k+1)$th iteration the so-called Q-function $Q\left(\boldsymbol{\Psi};\boldsymbol{\Psi}^{(k)}\right)$ equal to the expectation of the complete data log likelihood conditional on the observed data $\boldsymbol{y}$, using the current estimate $\boldsymbol{\Psi}^{(k)}$ for $\boldsymbol{\Psi}$. As this complete data log likelihood is linear in the missing class labels, this expectation conditional on $\boldsymbol{y}$ is effected by replacing the unobservable z_{ij} by its conditional expectation given $\boldsymbol{y}, z_{ij}^{(k)}$, where

$$\begin{aligned}z_{ij}^{(k)}&=\mathrm{E}_{\boldsymbol{\Psi}^{(k)}}\{Z_{ij}|\boldsymbol{y}\}\\&=\mathrm{Pr}_{\boldsymbol{\Psi}^{(k)}}\{Z_{ij}=1|\boldsymbol{y}\}\\&=\tau_i\left(\boldsymbol{y}_j;\boldsymbol{\Psi}^{(k)}\right)\\&=\frac{\pi_i^{(k)}\phi\left(\boldsymbol{y}_j;\boldsymbol{\mu}_i^{(k)},\boldsymbol{\Sigma}_i^{(k)}\right)}{\sum_{h=1}^{g}\pi_h^{(k)}\phi\left(\boldsymbol{y}_j;\boldsymbol{\mu}_h^{(k)},\boldsymbol{\Sigma}_h^{(k)}\right)}.\end{aligned}$$

Accordingly, we have that

$$\begin{aligned}Q\left(\boldsymbol{\Psi};\boldsymbol{\Psi}^{(k)}\right)=&\sum_{j=1}^{n}(1-m_j)\sum_{i=1}^{g}z_{ij}\{\log\pi_i+\phi(\boldsymbol{y}_j;\boldsymbol{\mu}_i,\boldsymbol{\Sigma}_i)\}\\&+\sum_{j=1}^{n}m_j\sum_{i=1}^{g}z_{ij}^{(k)}\{\log\pi_i+\phi(\boldsymbol{y}_j;\boldsymbol{\mu}_i,\boldsymbol{\Sigma}_i)\}\\&+\sum_{j=1}^{n}\left[(1-m_j)\log\{1-q(\boldsymbol{y}_j;\boldsymbol{\theta},\boldsymbol{\xi})\}+m_j\log q(\boldsymbol{y}_j;\boldsymbol{\theta},\boldsymbol{\xi})\right].\end{aligned}$$

We calculate the updated value $\boldsymbol{\Psi}^{(k+1)}$ of $\boldsymbol{\Psi}$ using two conditional maximisation (CM) steps.

CM-step 1: We fix $\boldsymbol{\xi}$ at its current value $\boldsymbol{\xi}^{(k)}$ and update $\boldsymbol{\theta}$ to $\boldsymbol{\theta}^{(k+1)}$ given by

$$\boldsymbol{\theta}^{(k+1)}=\arg\max_{\boldsymbol{\theta}}\ Q\left(\boldsymbol{\theta},\boldsymbol{\xi}^{(k)};\boldsymbol{\theta}^{(k)},\boldsymbol{\xi}^{(k)}\right),$$

where

$$\begin{aligned}Q(\boldsymbol{\theta},\boldsymbol{\xi}^{(k)};\boldsymbol{\theta}^{(k)},\boldsymbol{\xi}^{(k)}) &= \sum_{j=1}^{n}(1-m_j)\sum_{i=1}^{g} z_{ij}^{(k)}\{\log\pi_i + \log f_i(\boldsymbol{y}_j;\boldsymbol{\omega}_i)\}\\ &+\sum_{j=1}^{n} m_j \sum_{i=1}^{g} z_{ij}^{(k)}\{\log\pi_i + \log f_i(\boldsymbol{y}_j;\boldsymbol{\omega}_i)\}\\ &+\sum_{j=1}^{n}\left[(1-m_j)\log\{1-q(\boldsymbol{y}_j;\boldsymbol{\theta},\boldsymbol{\xi}^{(k)})\} + m_j\log q\left(\boldsymbol{y}_j;\boldsymbol{\theta},\boldsymbol{\xi}^{(k)}\right)\right].\end{aligned}$$

CM-step 2: We now fix $\boldsymbol{\theta}$ at its updated value $\boldsymbol{\theta}^{(k+1)}$ and update $\boldsymbol{\xi}$ to $\boldsymbol{\xi}^{(k+1)}$ as

$$\boldsymbol{\xi}^{(k+1)} = \underset{\boldsymbol{\xi}}{\arg\max}\ Q\left(\boldsymbol{\theta}^{(k+1)},\boldsymbol{\xi};\boldsymbol{\theta}^{(k)},\boldsymbol{\xi}^{(k)}\right),$$

which reduces to

$$\boldsymbol{\xi}^{(k+1)} = \underset{\boldsymbol{\xi}}{\arg\max}\ \log L_{\mathrm{PC}}^{(\mathrm{miss})}(\boldsymbol{\theta}^{(k+1)},\boldsymbol{\xi}),$$

on retaining only terms that depend on $\boldsymbol{\xi}$.

As $L_{\mathrm{PC}}^{(\mathrm{miss})}(\boldsymbol{\theta}^{(k+1)},\boldsymbol{\xi})$ belongs to the regular exponential family, we use the function `glm()`. The estimate $\boldsymbol{\Psi}_{\mathrm{PC}}^{(\mathrm{full})}$ is given by the limiting value of $\boldsymbol{\Psi}^{(k)}$ as k tends to infinity. We take the ECM algorithm as having converged when

$$\log L_{\mathrm{PC}}^{(\mathrm{full})}\left(\boldsymbol{\Psi}^{(k+1)}\right) - \log L_{\mathrm{PC}}^{(\mathrm{full})}(\boldsymbol{\Psi}^{(k)})$$

is less than some arbitrarily specified value.

Supporting information

Additional supporting information may be found in the online version of this article at http://wileyonlinelibrary.com/journal/anzs.

Data S1. ANZJS code.

References

Ahfock, D. & McLachlan, G.J. (2020). An apparent paradox: A classifier based on a partially classified sample may have smaller expected error rate than that if the sample were completely classified. *Statistics and Computing*, **30**, 1779–1790.

Ahfock, D. & McLachlan, G.J. (2023). Semi-supervised learning of classifiers from a statistical perspective: A brief review. *Econometrics and Statistics*, **26**, 124–138.

Benaglia, T., Chauveau, D., Hunter, D.R. & Young, D.S. (2009). mixtools: An R package for analyzing mixture models. *Journal of Statistical Software*, **32**, 1–29.

Biecek, P., Szczurek, E., Vingron, M. & Tiuryn, J. (2012). The R package bgmm: Mixture modeling with uncertain knowledge. *Journal of Statistical Software*, **47**, 1–31.

Blum, A. & Mitchell, T. (1998). Combining labeled and unlabeled data with co-training. In *Proceedings of the 11th Annual Conference on Computational Learning Theory*, pp. 92–100. New York, NY: Association for Computing Machinery. https://doi.org/10.1145/279943.279962

Chapelle, O., Schölkopf, B. & Zien, A. (2006). Semi-Supervised Learning. In *Adaptive Computation and Machine Learning*. Cambridge, MA: MIT Press. https://doi.org/10.1109/TNN.2009.2015974

Chawla, N.V. & Karakoulas, G. (2005). Learning from labeled and unlabeled data: An empirical study across techniques and domains. *Journal of Artificial Intelligence Research*, **23**, 331–366.

Côme, E., Oukhellou, L., Denoeux, T. & Aknin, P. (2009). Learning from partially supervised data using mixture models and belief functions. *Pattern Recognition*, **42**, 334–348.

Dempster, A.P., Laird, N.M. & Rubin, D.B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, **39**, 1–22.

Fraley, C. & Raftery, A. (2007). Model-based methods of classification: Using the mclust software in chemometrics. *Journal of Statistical Software*, **18**, 1–13.

Fujino, A., Ueda, N. & Saito, K. (2008). Semisupervised learning for a hybrid generative/discriminative classifier based on the maximum entropy principle. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **30**, 424–437.

Ganesalingam, S. & McLachlan, G.J. (1978). The efficiency of a linear discriminant function based on unclassified initial samples. *Biometrika*, **65**, 658–662.

Grün, B. & Leisch, F. (2007). Flexmix: An R package for finite mixture modelling. *R News*, **7**, 8–13.

Huang, J.T. & Hasegawa-Johnson, M. (2010). Semi-supervised training of Gaussian mixture models by conditional entropy minimization. In *Proceedings of the 11th Annual Conference of the International Speech Communication Association*, pp. 1353–1356. International Speech Communication Association. https://doi.org/10.21437/Interspeech.2010-19

Joachims, T. (1999). Transductive inference for text classification using support vector machines. In *Proceedings of the 16th International Conference on Machine Learning*, Vol. 99, pp. 200–209. San Francisco, CA: Morgan Kaufmann Publishers.

Kim, S.C. & Kang, T.J. (2007). Texture classification and segmentation using wavelet packet frame and Gaussian mixture model. *Pattern Recognition*, **40**, 1207–1221.

Lanckriet, G.R.G., Cristianini, N., Bartlett, P., Ghaoui, L.E.G. & Jordan, M.I. (2004). Learning the kernel matrix with semidefinite programming. *Journal of Machine Learning Research*, **5**, 27–72.

Lebret, R., Iovleff, S., Langrognet, F., Biernacki, C., Celeux, G. & Govaert, G. (2015). Rmixmod: The R package of the model-based unsupervised, supervised, and semi-supervised classification Mixmod library. *Journal of Statistical Software*, **67**, 1–29.

McLachlan, G. & Peel, D. (2000). Finite Mixture Models. In *Wiley Series in Probability and Statistics*. New York, NY: John Wiley and Sons. https://doi.org/10.1002/0471721182

McLachlan, G.J. (1975). Iterative reclassification procedure for constructing an asymptotically optimal rule of allocation in discriminant analysis. *Journal of the American Statistical Association*, **70**, 365–369.

McLachlan, G.J. (1977). Estimating the linear discriminant function from initial samples containing a small number of unclassified observations. *Journal of the American Statistical Association*, **72**, 403–406.

McLachlan, G.J. & Gordon, R.D. (1989). Mixture models for partially unclassified data: A case study of renal venous renin in hypertension. *Statistics in Medicine*, **8**, 1291–1300.

McLachlan, G.J. & Krishnan, T. (2008). The EM Algorithm and Extensions. In *Wiley Series in Probability and Statistics*, 2nd edn. Hoboken, NJ: John Wiley and Sons. https://doi.org/10.1002/9780470191613

McLachlan, G.J., Lee, S.X. & Rathnayake, S.I. (2019). Finite mixture models. *Annual Review of Statistics and Its Application*, **6**, 355–378.

McLachlan, G.J., Peel, D., Basford, K.E. & Adams, P. (1999). The EMMIX software for the fitting of mixtures of normal and t-components. *Journal of Statistical Software*, **4**, 1–14.

Mealli, F. & Rubin, D.B. (2015). Clarifying missing at random and related definitions, and implications when coupled with exchangeability. *Biometrika*, **102**, 995–1000.

Meng, X.L. & Rubin, D.B. (1993). Maximum likelihood estimation via the ECM algorithm: A general framework. *Biometrika*, **80**, 267–278.

Mesejo, P., Pizarro, D., Abergel, A., *et al*. (2016). Computer-aided classification of gastrointestinal lesions in regular colonoscopy. *IEEE Transactions on Medical Imaging*, **35**, 2051–2063.

Pan, W., Shen, X., Jiang, A. & Hebbel, R.P. (2006). Semi-supervised learning via penalized mixture model with application to microarray sample classification. *Bioinformatics*, **22**, 2388–2395.

Rubin, D.B. (1976). Inference and missing data. *Biometrika*, **63**, 581–592.

Szczurek, E., Biecek, P., Tiuryn, J. & Vingron, M. (2010). Introducing knowledge into differential expression analysis. *Journal of Computational Biology*, **17**, 953–967.

Szummer, M. & Jaakkola, T. (2001). Partially labeled classification with Markov random walks. *Advances in Neural Information Processing Systems*, **14**, 945–952.

Vapnik, V. (1998). The support vector method of function estimation. In *Nonlinear Modeling: Advanced Black-Box Techniques*, eds., Suykens, J.A.K. & Vandewalle, J., pp. 55–85. Boston, MA: Spinger. https://doi.org/10.1007/978-1-4615-5703-6_3

Zhou, D., Bousquet, O., Lal, T.N., Weston, J. & Schölkopf, B. (2003). Learning with local and global consistency. In *Advances in Neural Information Processing Systems*, Vol. 16, pp. 321–328. Cambridge, MA: MIT Press.