Stockholms universitet

Applying the EM algorithm to pixel-based approximations of bimodal data

Magnus Pierrau

Kandidatuppsats i matematisk statistik
Bachelor Thesis in Mathematical Statistics

Kandidatuppsats 2019:9
Matematisk statistik
Juni 2019

www.math.su.se

Matematisk statistik
Matematiska institutionen
Stockholms universitet
106 91 Stockholm

Stockholm University

Mathematical Statistics
Stockholm University
Bachelor Thesis **2019:9**
http://www.math.su.se

Applying the EM algorithm to pixel-based approximations of bimodal data

Magnus Pierrau*

June 2019

Abstract

In this thesis we develop and study an approximative version of the Expectation-Maximization algorithm, modified to analyze quantized data. The quantized data consists of a grid of pixels and emulates low resolution images of simplified protein structures. We perform several experiments designed to capture the performance of the approximative and regular EM algorithms under different circumstances. We find that the approximative variant consistently performs worse than the regular algorithm but provides acceptable results. However, as the size of the pixels decrease, the approximative variant approaches the results of the regular algorithm. We find that the approximative variant has a highly unstable convergence rate and that the assumption of homoscedasticity is not suited well for this variant.

*Postal address: Mathematical Statistics, Stockholm University, SE-106 91, Sweden. E-mail: magnus.pierrau@gmail.com. Supervisor: Ola Hössjer, Christian Blau.

Acknowledgements

I would like to sincerely thank my supervisors Prof. Ola Hössjer and Christian Blau for their invaluable feedback and encouragement, and without whom this thesis would not have been possible. I would also like to thank Linnea Axelsson for introducing me to the research at the SciLifeLab and Assoc. Prof. Chun-Biu Li for encouraging me to pursue this topic. Finally, I would like to thank my good friends Dasha Medvedeva and Lauge Gregers Hedegaard for their support and suggestions during the writing process.

Contents

1 Introduction

The topic of this thesis is inspired by the rapid development in protein structure determination methods in the past five years. The dramatic increase in understanding protein structure and function builds partly on an increase in experimental methods and better microscopes, but not the least on improved numerical algorithms for protein structure determination. One of these experimental methods is based on using cryo-electron microscopy (cryo-EM), in which proteins (extracted from e.g. frog eggs) are very rapidly frozen to a temperature of -150 °C. This keeps the protein structures intact within a layer of vitrified ice (supercooled water). The proteins are then bombarded from above with electrons, creating a through-view of the protein, captured on a detector below. The method of using cryo-EM imaging to create high resolution models of proteins is a field that has seen a surge lately due to increased quality in image resolution. The developers of cryo-EM were awarded the Nobel Prize in Chemistry in 2017 [1].

The most common algorithm that is used for protein structure determination from cryo-EM data is a Bayesian *Expectation-Maximisation algorithm*. Its most common implementation is a software called **RELION** (**RE**guralized **LI**kelihood **O**ptimizatio**N**), which for example is used by researchers at the Science for Life Laboratory (SciLifeLab) at Karolinska Institutet [2].

Protein structures are highly complex, and the data analyzed by RELION are 2D images of proteins in various structural states and angular positions. These images cannot easily be emulated by data from some two-dimensional probability distribution, but to fit the scope of this thesis we simplify this problem grossly by simulating data from a bivariate Gaussian Mixture Model with two underlying clusters and consider this an emulation of an image of a protein. Furthermore, we will ignore issues of interfering colored noise and the strength of the signal from the imaging process that researchers at SciLifeLab face.

The purpose of this thesis is thus to consider the canonical example of learning the underlying distributions of a Gaussian Mixture Model (GMM) under several different conditions, by using the EM algorithm. We will also develop and study an approximative variant of the EM algorithm which applies to data distorted by quantization. The quantization corresponds to "pixlifying" data to resemble imaging data from cryo-EM or other microscopy applications analyzed by RELION.

Some theory is presented and derived both for the regular EM algorithm and the approximative variant. We then perform several simulation experiments to study the performance of the approximative variant on the quantized data and compare it to that of the regular algorithm on undistorted data. Finally, we discuss the challenges that can arise in the implementation of this algorithm and some areas of improvement and future research.

2 Model

The data inferred on is simulated from a bivariate Gaussian Mixture Model, as defined in section 2.1 below. The EM algorithm is a powerful tool when there exist latent data, which in the case of the GMM is which underlying Gaussian distribution each data point originates from. The Gaussian Mixture Model can be considered when data is not unimodal and thus cannot be modelled by a single Gaussian distribution in a satisfactory way.

2.1 Gaussian Mixture Model (GMM)

The observed data, $\boldsymbol{y}$, is a vector of outcomes of the random variable $\boldsymbol{Y}$, with the ith element being the cartesian coordinates of $\boldsymbol{y}_i \in \mathcal{Y} = \mathbb{R}^2$. The latent data, $\boldsymbol{z}$ carries the information about which of the p underlying Gaussian distributions that $\boldsymbol{y}_i$ originated from. The restriction of bimodal data in this thesis implies that $p = 2$. The complete data is thus $\boldsymbol{X} = (\boldsymbol{Y}, \boldsymbol{Z})$. Throughout this paper the variables $\boldsymbol{y}$ and $\boldsymbol{x}$ will be used to denote the observed and the complete data respectively, and we will therefore denote the cartesian coordinates by $\boldsymbol{y}_i = (y_{i1}, y_{i2})$.

Table 1: Type of data

Data	Notation	Definition	State space
Observed	$\boldsymbol{y} \sim \boldsymbol{Y}$	$(\boldsymbol{y}_1, \boldsymbol{y}_2, ..., \boldsymbol{y}_n)^T$	$\mathcal{Y} = \mathbb{R}^2$
Latent	$\boldsymbol{z} \sim \boldsymbol{Z}$	$(z_1, z_2, ..., z_n)^T$	$\mathcal{Z} = \{1, \dots, p\}$
Complete	$\boldsymbol{x} \sim \boldsymbol{X}$	$(\boldsymbol{y}, \boldsymbol{z})$	$\mathcal{X} = \mathcal{Y} \times \mathcal{Z} = \mathbb{R}^2 \times \{1, \dots, p\}$

We will throughout the rest of this paper refer to each underlying Gaussian distribution as either "the underlying distribution" or "the cluster". Based on the assumption that there exist two underlying Gaussian distributions, we wish to infer on the respective mean vectors, covariance matrices and weights of each distribution. With w_j being the weight of the jth underlying distribution, we have the restriction $\sum_{j=1}^{p} w_j = 1$. Since $p = 2$ this is equal to the restriction $w_1 = 1 - w_2$. The weight provides the expected proportion between the number of data points originating from each respective distribution. With each underlying distribution having parameters $\{w_j, \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j\}$ we yield the parameter vector

$$\boldsymbol{\theta} = \Big(\{w_j\}, \{\boldsymbol{\mu}_j\}, \{\boldsymbol{\Sigma}_j\}\Big) \quad j = 1, 2. \tag{1}$$

The probability of observing some $\boldsymbol{x}_i$ as an outcome of the GMM-distributed random variable $\boldsymbol{X}_i$ is the sum of the probabilities of each of the p underlying

distributions multiplied by their respective weight. For a model with p underlying bivariate Gaussian distributions with respective parameters $(w_1, \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)$, ... , $(w_p, \boldsymbol{\mu}_p, \boldsymbol{\Sigma}_p)$, where $\boldsymbol{\mu}_j$ is the vector of expected mean values, $\boldsymbol{\Sigma}_j$ the covariance matrix and w_j the weight of the jth underlying distribution, we have the following probability density function:

$$p(\boldsymbol{x}_i|\boldsymbol{\theta}) = \sum_{j=1}^{p} w_j \cdot \phi(\boldsymbol{x}_i \mid \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j), \tag{2}$$

where

$$\phi(\boldsymbol{x}_i \mid \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j) = \frac{1}{\sqrt{(2\pi)^2 \det(\boldsymbol{\Sigma}_j)}} \exp\left\{-\frac{1}{2}(\boldsymbol{x}_i - \boldsymbol{\mu}_j)^T \boldsymbol{\Sigma}_j^{-1} (\boldsymbol{x}_i - \boldsymbol{\mu}_j)\right\}.$$

2.2 A pixel-based approximative GMM model

Many challenges can appear when analysing data from the real world. Data can be distorted or compromised by for example white or colored noise or poor resolution.

We choose to consider the problem of distortion by poor resolution of images. To emulate images that are analyzed at the SciLifeLab (and in other research) we quantize the GMM data by creating a grid around data and counting the number of data points in each square, as visualized in Figure 1. To draw a parallel to image analysis we can see each square as a *pixel* and the number of observations as the brightness, or the *intensity* of that pixel. A higher number of observations in a square corresponds to a higher intensity of that pixel. Henceforth we will mainly refer to these grid squares as pixels, but sometimes as "squares" or "boxes" if clarity is needed.

To apply this model on real images an assumption of linearity between the intensity of a pixel and the number of observations, or the amount of mass, inside the area of each box is needed. While the reality of this assumption can be questioned, it is a reasonable first assumption, that we will hold on to.

With Δ denoting the side length of each pixel, the grid is constructed by finding the lowest and highest y_{i1}– and y_{i2}–coordinates of the observed data and rounding them to their respective closest lower multiple of Δ. Let $\boldsymbol{y_1} = (y_{11}, y_{21}, ..., y_{n1})^T$ and $\boldsymbol{y_2} = (y_{12}, y_{22}, ..., y_{n2})^T$ be the vectors containing all respective y_{i1}– and y_{i2}–coordinates (note that these are not the same as $\boldsymbol{y}_1$ and $\boldsymbol{y}_2$ in section 2.1) and let $\lfloor y_{ij} \rfloor$ be the integer part of y_{ij}. Then,

$$\begin{aligned}
y_{\min,1} &= \lfloor \min(\boldsymbol{y_1}/\Delta) \rfloor \cdot \Delta, \\
y_{\min,2} &= \lfloor \min(\boldsymbol{y_2}/\Delta) \rfloor \cdot \Delta, \\
y_{\max,1} &= \lfloor \max(\boldsymbol{y_1}/\Delta) \rfloor \cdot \Delta, \\
y_{\max,2} &= \lfloor \max(\boldsymbol{y_2}/\Delta) \rfloor \cdot \Delta.
\end{aligned}$$

For example, if $\Delta = 0.4$ and the lowest y_{i1}-coordinate of the data set is 3.4, then $y_{\min,1} = 3.2$, since it is the closest lower multiple of 0.4. With $K =$

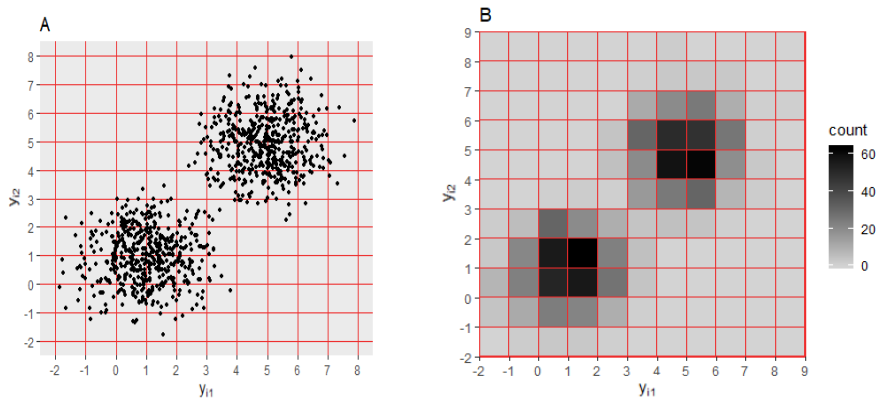

Figure 1: Example of regular GMM data (**A**) and its quantization (**B**) with pixel side length $\Delta = 1$.

$(y_{\max,1} - y_{\min,1})/\Delta$ and $L = (y_{\max,2} - y_{\min,2})/\Delta$ being the scaled range of the y_{i1}– and y_{i2}–coordinates of the data, we get a $K \times L$ grid of pixels covering the data set. Each pixel has sides of length Δ and subsequently an area of $A = \Delta^2$.

Our new algorithm will come to sum over each pixel and we thus need to index them. We let $q = K \cdot L$ be the total number of pixels and denote pixel k by b_k, with $k = 1, 2, ..., q$. The centerpoint of each box will come into play later on and it is denoted by $\boldsymbol{c}_k$. As will the intensity of each pixel, which we denote by N_k.

3 Parameter estimation and the EM algorithm

This section gives an overview of the theoretical framework of the EM algorithm, as well as some useful mathematical results for its expansion. Here we will also develop and discuss the approximative variant of the EM algorithm for the quantized data.

3.1 Maximum Likelihood-estimation

To infer on the underlying parameters of the GMM we want to calculate the parameter vector which maximizes the *likelihood function* of $\boldsymbol{\theta}$ given the complete data $\boldsymbol{y}$, over the parameter space Ω. This vector is called the *maximum likelihood estimate* (MLE) of $\boldsymbol{\theta}$ and is denoted $\hat{\boldsymbol{\theta}}$. The MLE is most often found by solving the *score equation* $\frac{d}{d\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\theta}|\boldsymbol{y}) = 0$ with regards to $\boldsymbol{\theta}$, where $\mathcal{L}(\boldsymbol{\theta}|\boldsymbol{y}) \coloneqq p(\boldsymbol{y}|\boldsymbol{\theta})$ is the likelihood function and $p(\boldsymbol{y}|\boldsymbol{\theta})$ is the density function of $\boldsymbol{Y}$ given $\boldsymbol{\theta}$. The solution provides the set of parameters that are most likely to have produced

the observed data.

For ease of calculation it is a common method to maximise the natural logarithm of $\mathcal{L}(\boldsymbol{\theta}|\boldsymbol{y})$, called the *log likelihood function* and denoted ℓ. This is possible since $\log(\cdot)$ is a monotonic and injective function, which results in ℓ and $\mathcal{L}$ having the same argmax over $\boldsymbol{\theta}$. Thus, from equation (1) we have that our MLE is

$$\hat{\boldsymbol{\theta}}_{\text{MLE}} = \left(\{\hat{w}_j\}, \{\hat{\boldsymbol{\mu}}_j\}, \{\hat{\boldsymbol{\Sigma}}_j\}\right), \quad j = 1, 2. \tag{3}$$

3.2 The general EM algorithm

The *Expectation-Maximization algorithm* is an iterative algorithm that, for each iteration provides an estimate, $\boldsymbol{\theta}^{(0)}, \boldsymbol{\theta}^{(1)}, ..., \boldsymbol{\theta}^{(m)}$, of $\boldsymbol{\theta}$, which, under some conditions, converges to $\hat{\boldsymbol{\theta}}_{\text{MLE}}$, as m tends to infinity. Since the resulting estimate from the algorithm is an approximation of the MLE (the algorithm stops at some point), we will henceforth denote the theoretical MLE by $\hat{\boldsymbol{\theta}}_{\text{MLE}}$ and the resulting estimates by $\hat{\boldsymbol{\theta}}$.

One reason for the popularity of EM is its general form, making it applicable to many different problems and areas, for example genetics, neural networks, text mining and clustering. Another reason is the promise of the likelihood function of $\boldsymbol{\theta}$ increasing in each iteration – a result of the *monotonicity theorem*. Thus, as m tends to infinity, the estimates $\boldsymbol{\theta}^{(m)}$ converges to $\hat{\boldsymbol{\theta}}_{\text{MLE}}$, given that the function has no local maxima. A third reason is its simple analytical and numerical form for distributions of the exponential family, as we will see later in section 3.2.4 [3][4].

The algorithm had been proposed already in the 1950's under special circumstances, and its convergence was proven by Rolf Sundberg in his doctoral thesis [7] at Stockholm University in 1976. Sundberg did extensive work on the algorithm and especially for the case of mixtures and convolutions of exponential family distributions. The algorithm was eventually generalized and explained thoroughly by Dempster, Laird and Rubin in their 1977 publication [5]. Maximum likelihood fitting has since then become a very common tool for fitting mixture distributions and the EM algorithm has been studied and developed for special cases in countless papers. [4][6].

In the sections below we will give a more thorough mathematical presentation of the algorithm, but to put it in words; in every iterative step of the algorithm we form the function Q, a conditional expectation of the log likelihood of our current parameter guess. We then solve for the parameters which maximize Q, and substitute the result as our new parameter guess in the next iteration. This process then repeats until convergence. This Q-function is the heart of the EM algorithm and is a function of the parameter vector $\boldsymbol{\theta}$. We will present it for

the complete data $\boldsymbol{X}$ here and use it to describe the iterative process, before looking more closely at how to expand and interpret it more explicitly:

$$Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(m)}) = E_{\boldsymbol{X}|\boldsymbol{y},\boldsymbol{\theta}^{(m)}}\left[\log p(\boldsymbol{X}|\boldsymbol{\theta})\right]. \tag{4}$$

The subscript $\boldsymbol{X}|\boldsymbol{y},\boldsymbol{\theta}^{(m)}$ in (4) is used to explicitly convey that the expectation is with regards to the complete data $\boldsymbol{X}$ conditioned on the observed data $\boldsymbol{y}$ and $\boldsymbol{\theta}^{(m)}$, the estimate of $\boldsymbol{\theta}$ in the mth iteration of the algorithm. The estimate in the next step of the algorithm, $\boldsymbol{\theta}^{(m+1)}$, is based on the current parameter estimates by relations detailed in section 3.2.4.

The function in (4) is called the Q-function since the original paper [5] used a Q to notate it, but, as put in [4]: "We like to say that the Q stands for *quixotic* because it is a bit crazy, hopeful and beautiful to think you can find the maximum likelihood estimate of $\boldsymbol{\theta}$ in this way that iterates round-and-round like a windmill. If Don Quixote had been a statistician, it is just the sort of thing he might have done."

3.2.1 The iterative steps

To clarify we split up the algorithm into five smaller steps [4]:

Step 1. Let $m = 0$ and make an initial guess $\boldsymbol{\theta}^{(m)}$ for $\boldsymbol{\theta}$.

Step 2. Assume now that the initial guess $\boldsymbol{\theta}^{(m)}$ is correct and formulate the conditional probability distribution $p(\boldsymbol{X}|\boldsymbol{\theta})$.

Step 3. Using the conditional probability distribution from the previous step, form the conditional expected log-likelihood, (4).

Step 4. Maximise (4) over each and every parameter in $\boldsymbol{\theta}$ and solve for that respective parameter.

Step 5. Let the parameters solved for in step 4 be the new $\boldsymbol{\theta}$-guess, $\boldsymbol{\theta}^{(m+1)}$, set $m \coloneqq m + 1$ and repeat from step 2 until reaching some predetermined level of convergence.

Steps 2 and 3 are commonly comprised to one step, called the *E-step*, for *expectation* and step 4 the *M-step*, for *maximisation* and hence gives the algorithm its name [5].

A standard stopping criterion is $\|\theta_i^{(m)} - \theta_i^{(m-1)}\| < \varepsilon$ for some $\varepsilon > 0$ and for all components θ_i in $\boldsymbol{\theta}$. Throughout this thesis we will use $\varepsilon = 10^{-5}$. When decreasing the precision to 10^{-3} we found that the number of iterations until convergence decreased on average, but the standard errors of the parameter estimates increased. Some discussion on alternative stopping criteria is found in

section 5.

To proceed we need to expand (4), for which two mathematical results are necessary. The proof of result 1 is straightforward and is laid out in section 6.1 of the appendix, while the proof of result 2 can be found in section 1.4.2 of [4].

3.2.2 Result 1 - Independence of observed data

The Q-function above was formulated in terms of complete data, $\boldsymbol{X}$. However, this result will allow us to solely condition on the latent data $\boldsymbol{Z}$ when evaluating the Q-function:

$$\begin{aligned} Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(m)}) &= E_{\boldsymbol{X}|\boldsymbol{y},\boldsymbol{\theta}^{(m)}}\left[\log p(\boldsymbol{X}|\boldsymbol{\theta})\right] \\ &= E_{\boldsymbol{Z}|\boldsymbol{y},\boldsymbol{\theta}^{(m)}}\left[\log p(\boldsymbol{y},\boldsymbol{Z}|\boldsymbol{\theta})\right]. \end{aligned} \tag{5}$$

■

3.2.3 Result 2 - Decomposition of Q-function

Another important result for us to be able to expand (4), as well as implement this algorithm into a program script, is the ability to decompose the Q-function into a sum of its parts:

Suppose all $\boldsymbol{X}_i$ are independent and identically distributed. It then holds for all $\boldsymbol{X}_i \in \mathcal{X}$ that fulfill the Markov relationship $\boldsymbol{\theta} \to \boldsymbol{X}_i \to \boldsymbol{Y}_i$ (i.e. that $\boldsymbol{y}_i$ is a function only of $\boldsymbol{x}_i$) for all $i = 1, ..., n$, and all $\boldsymbol{\theta} \in \Omega$, that

$$Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(m)}) = \sum_{i=1}^{n} Q_i(\boldsymbol{\theta}|\boldsymbol{\theta}^{(m)}), \tag{6}$$

where

$$Q_i(\boldsymbol{\theta}|\boldsymbol{\theta}^{(m)}) = E_{\boldsymbol{X}_i|\boldsymbol{y}_i,\boldsymbol{\theta}^{(m)}}[\log p(\boldsymbol{X}_i|\boldsymbol{\theta})], \quad i = 1, ..., n.$$

■

By using (5) we can rewrite $Q_i(\boldsymbol{\theta}|\boldsymbol{\theta}^{(m)})$ as a conditional expectation with regards to the latent data:

$$\begin{aligned} Q_i(\boldsymbol{\theta}|\boldsymbol{\theta}^{(m)}) &= E_{\boldsymbol{X}_i|\boldsymbol{y}_i,\boldsymbol{\theta}^{(m)}}\left[\log p(\boldsymbol{X}_i|\boldsymbol{\theta})\right] \\ &= E_{\boldsymbol{Z}_i|\boldsymbol{y}_i,\boldsymbol{\theta}^{(m)}}\left[\log p(\boldsymbol{y}_i,\boldsymbol{Z}_i|\boldsymbol{\theta})\right] \\ &= \sum_{j=1}^{p} P(\boldsymbol{Z}_i = j|\boldsymbol{y}_i,\boldsymbol{\theta}^{(m)})\ \log p(\boldsymbol{y}_i,\boldsymbol{Z}_i = j|\boldsymbol{\theta}). \end{aligned}$$

This, together with equation (6) from result 2, finally lets us express (4) as

$$\begin{aligned} Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(m)}) &= \sum_{i=1}^{n} Q_i(\boldsymbol{\theta}|\boldsymbol{\theta}^{(m)}) \\ &= \sum_{i=1}^{n}\sum_{j=1}^{p} P(\boldsymbol{Z}_i = j|\boldsymbol{y}_i, \boldsymbol{\theta}^{(m)}) \log p(\boldsymbol{y}_i, \boldsymbol{Z}_i = j|\boldsymbol{\theta}). \end{aligned} \tag{7}$$

The expansion in (7) is done by first decomposing the expression as a sum over all Q_i components, and then inserting the definition of the conditional expectation, which sums over all j.

The probability $P(\boldsymbol{Z}_i = j|\boldsymbol{y}_i, \boldsymbol{\theta}^{(m)})$ is denoted $\gamma_{ij}^{(m)}$ and is interpreted as the relative probability in the mth iteration that observation $\boldsymbol{y}_i$ originates from cluster j. This is called the *responsibility* of cluster j for $\boldsymbol{y}_{ij}$. It is calculated as

$$\gamma_{ij}^{(m)} \coloneqq \frac{w_j^{(m)} \phi(\boldsymbol{y}_i|\boldsymbol{\mu}_j^{(m)}, \boldsymbol{\Sigma}_j^{(m)})}{\sum_{l=1}^{p} w_l^{(m)} \phi(\boldsymbol{y}_i|\boldsymbol{\mu}_l^{(m)}, \boldsymbol{\Sigma}_l^{(m)})},$$

and consequently $\sum_{j=1}^{p} \gamma_{ij}^{(m)} = 1$.

Figure 2 visualizes the product $\gamma_{i1}^{(m)} \cdot \gamma_{i2}^{(m)}$ of the regular EM algorithm for a data set of size $n = 1000$ at the mth iteration. The product is maximised when $\gamma_{i1}^{(m)} = \gamma_{i2}^{(m)} = 0.5$, i.e. when both distributions are given equal responsibility for $\boldsymbol{y}_i$. The red area of overlapping data points, in this example, creates a clear region of uncertainty, where data is equally likely to belong to either distribution, whereas points that are further away from the opposite cluster center have a higher relative probability, indicated with the blue color.

We let $\boldsymbol{\gamma}^{(m)}$ denote the $n \times p$ matrix with $\gamma_{ij}^{(m)}$ as the ijth element. The column sums of $\boldsymbol{\gamma}^{(m)}$ will come into play later and are denoted $n_j^{(m)} = \sum_{i=1}^{n} \gamma_{ij}^{(m)}$, for $j = 1, ..., p$.

The second part of (7) is the log likelihood function of $\boldsymbol{\theta}$ given the complete data $\boldsymbol{x}_i$ for observation i and has the explicit form

$$\begin{aligned} \log p(\boldsymbol{y}_i, \boldsymbol{Z}_i = j|\boldsymbol{\theta}) &= \log\left(\frac{w_j}{\sqrt{(2\pi)^2 \det(\boldsymbol{\Sigma}_j)}} \exp\left\{ -\frac{1}{2}(\boldsymbol{y_i} - \boldsymbol{\mu}_j)^T \boldsymbol{\Sigma}_j^{-1} (\boldsymbol{y_i} - \boldsymbol{\mu}_j) \right\} \right) \\ &= \log w_j - \log 2\pi - \frac{1}{2} \log\det(\boldsymbol{\Sigma}_j) - \frac{1}{2}(\boldsymbol{y_i} - \boldsymbol{\mu}_j)^T \boldsymbol{\Sigma}_j^{-1} (\boldsymbol{y_i} - \boldsymbol{\mu}_j) \\ &\propto \log w_j - \frac{1}{2} \log\det(\boldsymbol{\Sigma}_j) - \frac{1}{2}(\boldsymbol{y_i} - \boldsymbol{\mu}_j)^T \boldsymbol{\Sigma}_j^{-1} (\boldsymbol{y_i} - \boldsymbol{\mu}_j), \end{aligned}$$

where the proportionality in the last step is with regards to $\boldsymbol{\theta}$.

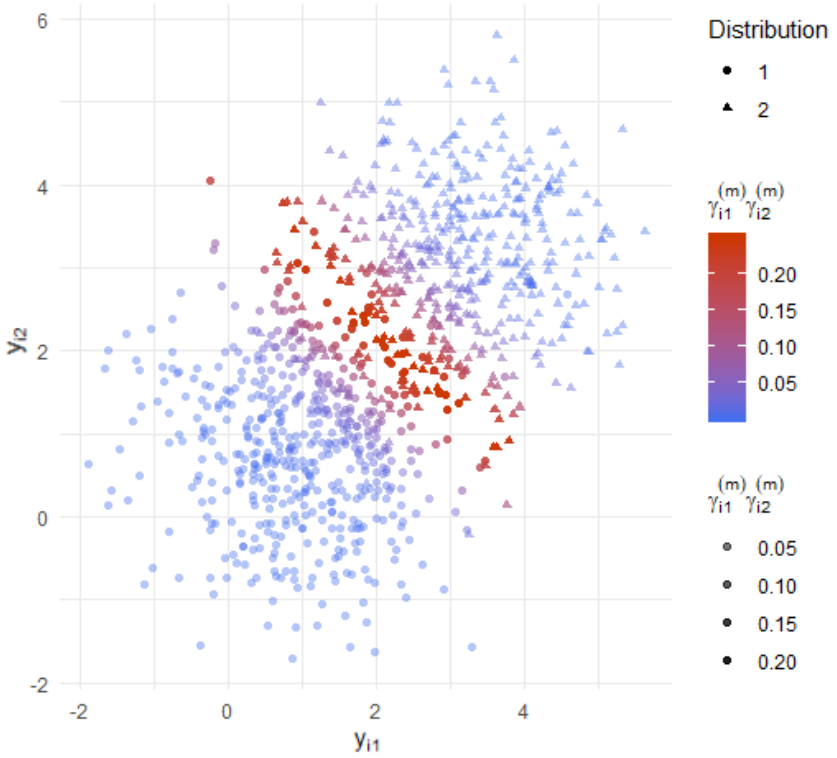

Figure 2: A visualization of $\gamma_{i1}^{(m)} \cdot \gamma_{i2}^{(m)}$ for a dataset of $n = 1000$, $\boldsymbol{\mu}_1 = (1,1)^T$, $\boldsymbol{\mu}_2 = (3,3)^T$, $\boldsymbol{\Sigma}_1 = \boldsymbol{\Sigma}_2 = \mathbb{I}_2$ and $w_1 = w_2 = 0.5$.

3.2.4 Iterative relations for parameter estimates

In some applications of the EM algorithm the Q-function can be highly complicated and not possible to optimize analytically, especially in cases with higher dimensions. However, in the case of the GMM, the maximisation in step 4 is analytically tractable and gives us explicit expressions for the next parameter guess $\boldsymbol{\theta}^{(m+1)}$.

This is due to each underlying Gaussian distribution in the GMM belonging to the exponential family. One can show that the score function of Q then involves data in terms of a sum of all minimal sufficient statistics $T(\boldsymbol{X}_i)$ for all $\boldsymbol{X}_i$ multiplied by a function of $\boldsymbol{\theta}$. It can then be shown that the maximum likelihood estimate is the inverse of the expectation of $T(\boldsymbol{X})$, and since in the case of the GMM (and in most important cases), these inverses exist, this yields us an explicit analytical form for the MLE [7].

After maximisation with respect to each respective parameter we get the fol-

lowing results (the derivations can be found in the appendix, section 6.2):

$$w_j^{(m+1)} = \frac{n_j^{(m)}}{\sum_{l=1}^{p} n_l^{(m)}}, \tag{8}$$

$$\boldsymbol{\mu}_j^{(m+1)} = \frac{1}{n_j^{(m)}} \sum_{i=1}^{n} \gamma_{ij}^{(m)} \boldsymbol{y}_i, \tag{9}$$

$$\boldsymbol{\Sigma}_j^{(m+1)} = \frac{1}{n_j^{(m)}} \sum_{i=1}^{n} \gamma_{ij}^{(m)} \left(\boldsymbol{y}_i - \boldsymbol{\mu}_j^{(m+1)}\right)^T \left(\boldsymbol{y}_i - \boldsymbol{\mu}_j^{(m+1)}\right), \tag{10}$$

all for $j = 1, ..., p$.

We observe especially that in the iterative step for $\boldsymbol{\Sigma}_j^{(m+1)}$ we use the new estimate of $\boldsymbol{\mu}_j^{(m+1)}$ in the calculations. It is therefore important to perform the calculations in a correct order, i.e. (9) before (10), when implementing the algorithm. These formulae can both be interpreted as a form of weighted averages for the coordinates and sample variates, using the responsibilities $\gamma_{ij}^{(m)}$ as weights.

Later we will discuss the possibility of applying the restriction of homoscedasticity, i.e. that $\boldsymbol{\Sigma}_1 = ... = \boldsymbol{\Sigma}_p = \boldsymbol{\Sigma}$, in order to make the algorithm more robust. When applying this restriction we get the following iterative step for $\boldsymbol{\Sigma}$:

$$\boldsymbol{\Sigma}^{(m+1)} = \frac{1}{\sum_{j=1}^{p} n_j^{(m)}} \sum_{j=1}^{p} \sum_{i=1}^{n} \gamma_{ij}^{(m)} \left(\boldsymbol{y}_i - \boldsymbol{\mu}_j^{(m+1)}\right)^T \left(\boldsymbol{y}_i - \boldsymbol{\mu}_j^{(m+1)}\right).$$

3.2.5 Initiation

The algorithm requires initial parameter guesses $\boldsymbol{\theta}^{(0)}$ as starting values. There are several ways to choose the initial guesses – for example by simulating $\boldsymbol{\mu}_1^{(0)}$ and $\boldsymbol{\mu}_2^{(0)}$ from $\boldsymbol{N}(\bar{\boldsymbol{y}}, \boldsymbol{V})$, where $\bar{\boldsymbol{y}}$ is the sample mean and $\boldsymbol{V}$ the sample covariance matrix of data. This $\boldsymbol{V}$ can then be chosen as our $\boldsymbol{\Sigma}_j^{(0)}$ and $w^{(0)}$ as $1/p$ [6]. However, we choose a different initiation for $\boldsymbol{\mu}^{(0)}$ by applying the commonly used method of *k-means clustering*, laid out in section 6.3 of the appendix. In our case $k = p$ but we will use the term k-means. This method provides us with rough estimates, which are expected to be in a vicinity of the mean vectors, and we use these as the initial guesses $\boldsymbol{\mu}_1^{(0)}$ and $\boldsymbol{\mu}_2^{(0)}$ [4]. The parameters w_j will be initiated by $w_1^{(0)} = w_2^{(0)} = 1/p = 0.5$ throughout all experiments. We will use different $\boldsymbol{\mu}^{(0)}$ and $\boldsymbol{\Sigma}^{(0)}$ as means of initiation in the different simulation experiments in section 4, and we will discuss these choices there.

3.3 EM for quantized data - the approximative grid-based method

After distorting data we no longer know the exact $\boldsymbol{y}_{i1}$- and $\boldsymbol{y}_{i2}$-coordinates of each observation – only which pixel they were observed in. To use the EM algorithm on this data we will have to rework the algorithm and its iterative steps. To begin with we will have to rewrite the formula for equation (7).

We consider some pixel b_k from the grid and assume that it has an intensity of one, i.e. that a single data point, $\boldsymbol{y}_i$ has been observed inside of it. To calculate $P(\boldsymbol{y}_i \in b_k)$, i.e. the probability of observing $\boldsymbol{y}_i$ in b_j, we have to integrate the probability density function $p(\cdot, \cdot)$ of the assumed underlying distribution over the area of b_k. We thus have to evaluate a double integral of the density function of a GMM over the area of the pixel:

$$
\begin{aligned}
P(\boldsymbol{y}_i \in b_k) &= \iint_{b_k} p(y_{i1}, y_{i2})\, dy_{i1}\, dy_{i2} \\
&= \iint_{b_k} \sum_{j=1}^{2} \frac{w_j}{\sqrt{(2\pi)^2 \det(\boldsymbol{\Sigma}_j)}} \exp\left\{-\frac{1}{2}(\boldsymbol{y}_i - \boldsymbol{\mu}_j)^T \boldsymbol{\Sigma}_j^{-1} (\boldsymbol{y}_i - \boldsymbol{\mu}_j)\right\} dy_{i1}\, dy_{i2}.
\end{aligned}
$$

However, it is known that there exists no analytical closed form for the integral of the density function of a bivariate Gaussian distribution, and we thus have to approximate the sought for probability. We could for example use Monte Carlo-approximation, but we choose to settle for a more practical alternative.

Figure 3: Visualization of the relative error between the Monte Carlo approximation using 1000 random samples and the point estimate approximation of $P(\boldsymbol{y}_i \in b_k)$ for $\Delta = 0.5$ (**A**) and $\Delta = 1$ (**B**). Calculations are based on a Gaussian Mixture Model with parameters $w_1 = w_2 = 0.5$, $\boldsymbol{\mu}_1 = (2, 2)^T$, $\boldsymbol{\mu}_2 = (5, 5)^T$ and $\boldsymbol{\Sigma}_1 = \boldsymbol{\Sigma}_2 = \mathbb{I}_2$. The points indicate the two cluster centers $\boldsymbol{\mu}_1$ and $\boldsymbol{\mu}_2$. All relative errors larger than 1 are truncated to the same color in order to allow a more interpretable visualization and a direct comparison between **A** and **B**. See Figure 14 in appendix 6.5 for a larger version of Figure 3**B**.

The approximation of our choice is a point estimate, where we take the value of the density function at the centerpoint of the pixel and multiply it by its area, A. It is practical since it allows us to retain the closed analytical expressions for the iterative formulae with only minor modifications, which would not be possible by Monte Carlo-approximation.

From Figure 3 we note that for both $\Delta = 0.5$ and 1 the relative error is high in the fringes of the figure, but lower around the mean vectors, where the majority of the data points for this distribution are located.

When decreasing the size of the pixels, i.e. as $\Delta \to 0$, we have that $\boldsymbol{c}_k(\boldsymbol{y}_i) \to \boldsymbol{y}_i$, and thus $p(\boldsymbol{c}_k(\boldsymbol{y}_i)) \to p(\boldsymbol{y}_i)$, i.e. the approximative probability will converge to its true value. As Δ decreases each pixel will eventually only contain one single observation, implying that $q \to n$ and $\Gamma_{kj}^{(m)} \to \gamma_{ij}^{(m)}$. Thus, as $\Delta \to 0$ the approximative variant converges to the regular EM algorithm.

All this indicates that the approximation is feasible on the data scale that we operate at throughout this paper.

3.3.1 Deriving the Q-function for the approximative method

With $\boldsymbol{c}_k$ defined as the centerpoint of b_k, and with B_i the pixel which $\boldsymbol{y}_i$ was observed in, we have that the estimated probability of $\boldsymbol{y}_i$ being observed in b_k, given that $\boldsymbol{y}_i$ originated from some Gaussian distribution with parameters $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ is

$$\hat{p}(B_i = b_k|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = A \cdot \phi(\boldsymbol{c}_k|\boldsymbol{\mu}, \boldsymbol{\Sigma}).$$

By conditioning on which cluster $\boldsymbol{y}_i$ was observed in we can now find the joint probability that $\boldsymbol{y}_i$ is observed in b_k and also originates from the underlying cluster center j:

$$\begin{aligned}\hat{p}(B_i = b_k, Z_i = j|\boldsymbol{\theta}) &= P(Z_i = j) \cdot \hat{p}(B_i = b_k|Z_i = j, \boldsymbol{\theta}) \\ &= w_j \cdot A \cdot \phi(\boldsymbol{c}_k|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j). \end{aligned} \tag{11}$$

In order to rewrite the Q-function we also need to find $\hat{P}(Z_i = j|B_i, \boldsymbol{\theta}^{(m)})$, which is interpreted as the approximative relative probability that y_i originated from cluster center j, conditioned on both which pixel it was observed in as well as on the current guess $\boldsymbol{\theta}^{(m)}$:

$$\begin{aligned}\hat{P}(Z_i = j|B_i = b_k, \boldsymbol{\theta}^{(m)}) &= \frac{w_j^{(m)} \cdot A \cdot \phi(\boldsymbol{c}_k|\boldsymbol{\mu}_j^{(m)}, \boldsymbol{\Sigma}_j^{(m)})}{\sum_{l=1}^{p} w_l^{(m)} \cdot A \cdot \phi(\boldsymbol{c}_k|\boldsymbol{\mu}_l^{(m)}, \boldsymbol{\Sigma}_l^{(m)})} \\ &= \frac{w_j^{(m)} \cdot \phi(\boldsymbol{c}_k|\boldsymbol{\mu}_j^{(m)}, \boldsymbol{\Sigma}_j^{(m)})}{\sum_{l=1}^{p} w_l^{(m)} \cdot \phi(\boldsymbol{c}_k|\boldsymbol{\mu}_l^{(m)}, \boldsymbol{\Sigma}_l^{(m)})} \coloneqq \gamma_{kj}^{(m)}. \end{aligned} \tag{12}$$

We note here that since every pixel has the same area, the A's cancel out, which makes (12) almost identical to $\gamma_{ij}^{(m)}$, with the sole difference of exchanging $\boldsymbol{c}_k$ with $\boldsymbol{y}_i$. Thus, the difference in relative probability between the underlying distributions is attributed to a shift of each $\boldsymbol{y}_i$ to the centerpoint of that pixel.

We note also that $\boldsymbol{c}_k$ is a vector valued function of $\boldsymbol{y}_i$ (and of Δ as well, but since Δ is a fixed variable we will use the notation $\boldsymbol{c}_k(\boldsymbol{y}_i)$ in the expression below, and $\boldsymbol{c}_k$ thereafter, except if clarity is needed):

$$\boldsymbol{c}_k \coloneqq \boldsymbol{c}_k(\boldsymbol{y}_i) = (\lfloor \frac{\boldsymbol{y}_i}{\Delta} \rfloor + \frac{1}{2}) \cdot \Delta,$$

where $\lfloor \frac{\boldsymbol{y}_i}{\Delta} \rfloor = (\lfloor \frac{y_{i1}}{\Delta} \rfloor, \lfloor \frac{y_{i2}}{\Delta} \rfloor)$, i.e. the floor function is taken componentwise of $\boldsymbol{y}_i$. Now, by using the approximations (11) and (12), we can express the Q-function for the quantized data:

$$\begin{aligned}
\hat{Q}(\boldsymbol{\theta}|\boldsymbol{\theta}^{(m)}) &= \sum_{i=1}^{n}\sum_{j=1}^{p} \hat{P}(Z_i = j|B_i, \boldsymbol{\theta}^{(m)}) \ \log \hat{p}(B_i, Z_i = j|\boldsymbol{\theta}) \\
&= \sum_{i=1}^{n}\sum_{j=1}^{p} \frac{w_j^{(m)} \cdot \phi(\boldsymbol{c}_k(\boldsymbol{y}_i)|\boldsymbol{\mu}_j^{(m)}, \boldsymbol{\Sigma}_j^{(m)})}{\sum_{l=1}^{p} w_l^{(m)} \cdot \phi(\boldsymbol{c}_k(\boldsymbol{y}_i)|\boldsymbol{\mu}_l^{(m)}, \boldsymbol{\Sigma}_l^{(m)})} \log\left(w_j \cdot A \cdot \phi(\boldsymbol{c}_k(\boldsymbol{y}_i)|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)\right).
\end{aligned} \tag{13}$$

We note that every $\boldsymbol{y}_i$ that was observed in the same pixel will yield the same expression for its respective $\hat{Q}_i(\boldsymbol{\theta}|\boldsymbol{\theta}^{(m)})$, since they all map to the same $\boldsymbol{c}_k$. Therefore, we can, instead of summing over every $\boldsymbol{y}_i$, sum over every pixel b_k and multiply each term in the sum with its respective pixel intensity, N_k. Thus, (13) becomes

$$\begin{aligned}
&\sum_{k=1}^{q} N_k \cdot \sum_{j=1}^{p} \frac{w_j^{(m)} \cdot \phi(\boldsymbol{c}_k|\boldsymbol{\mu}_j^{(m)}, \boldsymbol{\Sigma}_j^{(m)})}{\sum_{l=1}^{p} w_l^{(m)} \cdot \phi(\boldsymbol{c}_k|\boldsymbol{\mu}_l^{(m)}, \boldsymbol{\Sigma}_l^{(m)})} \log\left(w_j \cdot A \cdot \phi(\boldsymbol{c}_k|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)\right) \\
&\propto \sum_{k=1}^{q} N_k \cdot \sum_{j=1}^{p} \frac{w_j^{(m)} \cdot \phi(\boldsymbol{c}_k|\boldsymbol{\mu}_j^{(m)}, \boldsymbol{\Sigma}_j^{(m)})}{\sum_{l=1}^{p} w_l^{(m)} \cdot \phi(\boldsymbol{c}_k|\boldsymbol{\mu}_l^{(m)}, \boldsymbol{\Sigma}_l^{(m)})} \log\left(w_j \cdot \phi(\boldsymbol{c}_k|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)\right) \qquad (14) \\
&= \sum_{k=1}^{q} N_k \cdot \sum_{j=1}^{p} \frac{w_j^{(m)} \cdot \phi(\boldsymbol{c}_k|\boldsymbol{\mu}_j^{(m)}, \boldsymbol{\Sigma}_j^{(m)})}{\sum_{l=1}^{p} w_l^{(m)} \cdot \phi(\boldsymbol{c}_k|\boldsymbol{\mu}_l^{(m)}, \boldsymbol{\Sigma}_l^{(m)})} \log\left(p(\boldsymbol{c}_k, Z_k = j|\boldsymbol{\theta})\right) \\
&= \sum_{k=1}^{q}\sum_{j=1}^{p} N_k \cdot \gamma_{kj}^{(m)} \cdot \log p(\boldsymbol{c}_k, Z_i = j|\boldsymbol{\theta}) \\
&= \sum_{k=1}^{q}\sum_{j=1}^{p} \Gamma_{kj}^{(m)} \cdot \log p(\boldsymbol{c}_k, Z_i = j|\boldsymbol{\theta}),
\end{aligned}$$

with $\Gamma_{kj}^{(m)} \coloneqq N_k \cdot \gamma_{kj}^{(m)}$, Z_k denoting the cluster origin of all observations in b_k, and p being the density function of the GMM (equation (2)). Step (14) above is due to the sum of all terms involving $\log A$ being proportional to $\boldsymbol{\theta}$ and thus

will disappear in the maximization step.

The term $\Gamma_{kj}^{(m)}$ can be interpreted as the number of observations in b_k that is attributed to cluster j. In contrast to $\gamma_{ij}^{(m)}$, $\Gamma_{kj}^{(m)}$ does not sum to 1 over all j's, and thus gives more weight to pixels with a high intensity. Figure 4 illustrates $\gamma_{k1}^{(m)} \cdot \gamma_{k2}^{(m)}$ for the same data set as in Figure 2. We can discern a clear region of uncertainty between the two clusters, as for the unquantized example.

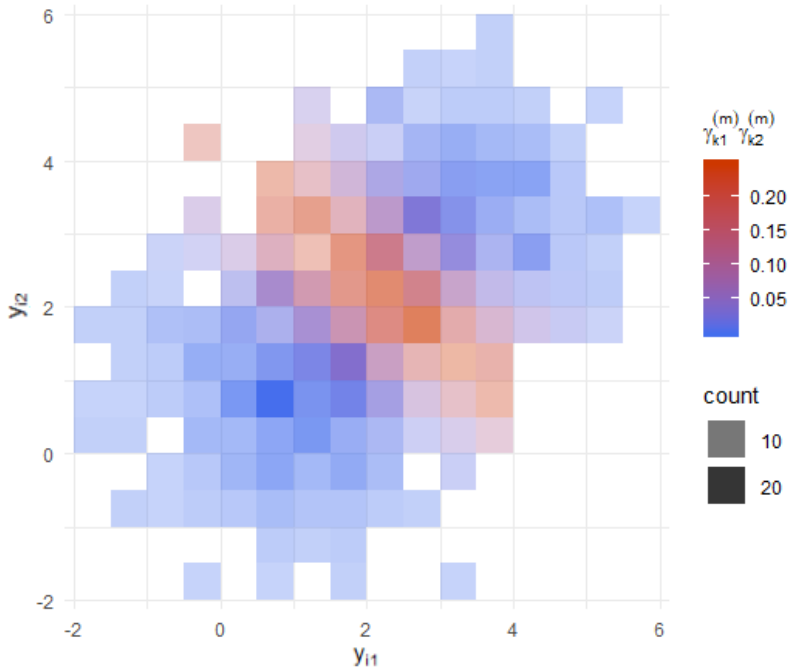

Figure 4: A visualization of $\gamma_{k1}^{(m)} \cdot \gamma_{k2}^{(m)}$ from a representative quantized dataset with $\Delta = 0.5$, $n = 1000$ and parameters $\boldsymbol{\mu}_1 = (1,1)^T$, $\boldsymbol{\mu}_2 = (3,3)^T$, $\boldsymbol{\Sigma}_1 = \boldsymbol{\Sigma}_2 = \mathbb{I}_2$ and $w_1 = w_2 = 0.5$. The darker pixels indicate a higher N_k (intensity/count).

3.3.2 Iterative relations for the approximative method

To find the new iteration formulae we modify the original equations (8), (9) and (10). Instead of summing over all $\boldsymbol{y}_i$ we sum over the closest pixel midpoints, $\boldsymbol{c}_k(\boldsymbol{y}_i)$. All data points observed in the same pixel will thus have identical expressions for $\gamma_{ij}^{(m)}$ and $\boldsymbol{c}_k(\boldsymbol{y}_i) - \mu_j^{(m+1)}$, and so instead of summing over every data point we can sum over all pixels and for each term multiply it by its respective intensity.

With $q_j^{(m)} = \sum_{k=1}^{q} \Gamma_{kj}^{(m)}$ for $j = 1, ..., p$, we get the following iterative relations:

$$w_j^{(m+1)} = \frac{q_j^{(m)}}{\sum_{l=1}^{p} q_l^{(m)}}, \tag{15}$$

$$\boldsymbol{\mu}_j^{(m+1)} = \frac{1}{q_j^{(m)}} \sum_{k=1}^{q} \Gamma_{kj}^{(m)} \boldsymbol{c}_k, \tag{16}$$

$$\boldsymbol{\Sigma}_j^{(m+1)} = \frac{1}{q_j^{(m)}} \sum_{k=1}^{q} \Gamma_{kj}^{(m)} \left(\boldsymbol{c}_k - \boldsymbol{\mu}_j^{(m+1)}\right)^T \left(\boldsymbol{c}_k - \boldsymbol{\mu}_j^{(m+1)}\right), \tag{17}$$

for $j = 1, ..., p$.

These relations are derived by analytically maximising (13) with respect to each respective parameter, but with $\boldsymbol{c}_k$ instead of $\boldsymbol{y}_i$. The multiplicative constant of A disappears in the maximisation over the parameters and the calculations are then analogous to those for equations (8) – (10) for the original EM algorithm.

As with the regular EM algorithm we can apply the restriction $\boldsymbol{\Sigma}_1 = ... = \boldsymbol{\Sigma}_p = \boldsymbol{\Sigma}$, which gives rise to the following iterative relation:

$$\boldsymbol{\Sigma}^{(m+1)} = \frac{1}{\sum_{l=1}^{p} q_l^{(m)}} \sum_{j=1}^{p} \sum_{k=1}^{q} \Gamma_{kj}^{(m)} \left(\boldsymbol{c}_k - \boldsymbol{\mu}_j^{(m+1)}\right)^T \left(\boldsymbol{c}_k - \boldsymbol{\mu}_j^{(m+1)}\right).$$

Equations (15) – (17) are next to identical to the formulae for the regular EM algorithm, with the difference of replacing $\boldsymbol{y}_i$ by $\boldsymbol{c}_k$, then $\gamma_{ij}^{(m)}$ by $\Gamma_{kj}^{(m)}$ and finally summing over every pixel $k = 1, ..., q$, instead of every data point $i = 1, ..., n$.

4 Method and results

In this section we perform multiple simulation experiments to study how both the regular and approximative EM algorithms perform under different circumstances. There are countless scenarios to study, and we choose to restrict our experiments to a few interesting ones. In the coming three sections we will vary some underlying assumptions of the distribution, the initial values and the overlap between the two clusters.

To quantify the performance of each algorithm we will simulate R different data sets from the same underlying distribution, quantize the simulated data and then apply each algorithm and the particular variants specified in that experiment, to each of the R data sets. With $\hat{\boldsymbol{\theta}}_{ri}$ denoting the estimate of the ith component of the parameter vector from the rth data set, and $\boldsymbol{\theta}_i$ the true value, we then calculate the standard error of each parameter estimate for each

variant of the algorithms by

$$\text{SE}(\hat{\boldsymbol{\theta}}_i) = \sqrt{\frac{1}{R-1}\sum_{r=1}^{R}(\hat{\boldsymbol{\theta}}_{ri} - \boldsymbol{\theta}_i)^2}.$$

We compare the errors of the approximative EM algorithm applied to the quantized data to the ones of the regular EM algorithm applied to the original data. We also consider some different statistics of the number of iterations until convergence for each algorithm.

In section 4.1, we study the results of a canonical setup to see how well the algorithms perform on non-overlapping data and with non-disruptive initial starting values, both in the homoscedastic and heteroscedastic case. The results here will serve as reference points for the later experiments. In section 4.2 we inspect how supplying different initial starting values of $\boldsymbol{\mu}^{(0)}$ and $\boldsymbol{\Sigma}^{(0)}$ to the algorithms affect the standard errors and number of iterations, and lastly, in section 4.3, we study how well the algorithms perform when the underlying distributions overlap to different extents.

Throughout section 4 the following notation is used: Σ_{jkl} indicates element k,l of covariance matrix $\boldsymbol{\Sigma}_j$ and μ_{jk} indicates element k of $\boldsymbol{\mu}_j$, for cluster j. Since $w_2 = 1 - w_1$ the two parameters have the same standard error and will therefore be presented in all tables as w. For most tables we will present a selection of representative parameters, chosen to display general trends in the results, to make the tables more orderly. The full tables for all experiments can be found in section 6.6 of the appendix.

4.1 Experiment 1 – Canonical setup

To compare the algorithms on a canonical setup we simulate $R = 1000$ data sets from a homoscedastic GMM of size $n = 1000$ data points with underlying parameters $w_1 = w_2 = 0.5$, $\boldsymbol{\mu}_1 = (1,1)^T$, $\boldsymbol{\mu}_2 = (5,5)^T$ and $\boldsymbol{\Sigma}_1 = \boldsymbol{\Sigma}_2 = \mathbb{I}_2$, where $\mathbb{I}_2$ is the identity matrix of size 2. In order to deduce how the approximative algorithm performs for varying pixel sizes we will consider the three values $\Delta = 0.1, 0.5$ and 1. We will also compare the results when assuming equal covariance matrices and not.

To initiate the algorithms we use the initial guesses of $\boldsymbol{\Sigma}_j^{(0)} = \mathbb{I}_2$ and $w_j^{(0)} = 0.5$ for $j = 1, 2$. For the two mean vectors $\boldsymbol{\mu}_j^{(0)}$ we use k-means clustering. Note that the initial guesses $w_j^{(0)}$ and $\boldsymbol{\Sigma}_j^{(0)}$ equal the true parameter values. In section 4.2 we will consider how the algorithms handles more challenging and compromising initial guesses. As seen in Figure 5A, there is very little overlap between the underlying distributions. In the case of $\Delta = 0.1$ the distortion is minimal, with the highest intensity of any pixel being 4, as seen in Figure 5B. As we increase Δ to 0.5 and 1 (fig. 5C and D) the quantized data becomes cruder and less

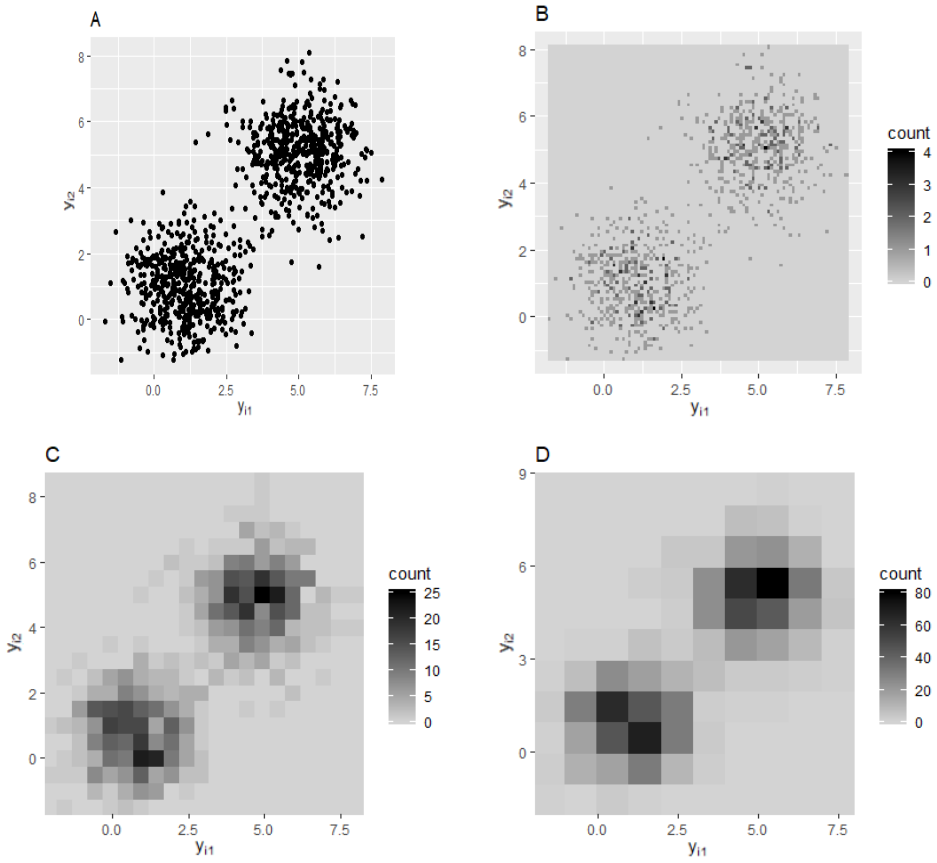

Figure 5: A representative of the $R = 1000$ data sets visualized at varying levels of Δ: A: Original data. B: Quantized data at $\Delta = 0.1$. C: Quantized data at $\Delta = 0.5$. D: Quantized data at $\Delta = 1$.

detailed, but still with a clear outline of two separate clusters.

Table 2 indicates that the standard errors increase with the size of the pixels. This feels intuitive, as when we increase the size of the pixels, we remove more and more information, and thus make the approximation less accurate. We note also from Table 2 that at $\Delta = 0.1$ the approximative variant performs equally well as the regular EM algorithm. Although it cannot be discerned from these tables, the difference in errors appear at the fifth decimal place, for all parameter estimates.

Table 3 shows that while the mean and median for $\Delta = 0.1$ and the regular algorithm are similar in size, there is a discrepancy between these statistics for

Table 2: **Experiment 1** - Standard errors of representative parameter estimates from the regular EM algorithm (Regular) and the approximative variant (Approx.), with $R = 1000$ and $n = 1000$ on varying levels of Δ. See Table 11 in appendix for more information.

		Standard error of						
EM variant	Δ	$\hat{\mu}_{11}$	$\hat{\mu}_{22}$	$\hat{\Sigma}_{111}$	$\hat{\Sigma}_{112}$	$\hat{\Sigma}_{222}$	$\hat{\Sigma}_{212}$	$\hat{w}$
Regular	-	0.045	0.047	0.064	0.047	0.066	0.048	0.016
Approx.	0.1	0.045	0.047	0.064	0.047	0.066	0.048	0.016
Approx.	0.5	0.077	0.076	0.147	0.140	0.164	0.125	0.017
Approx.	1	0.099	0.154	0.207	0.189	0.235	0.215	0.025

Table 3: **Experiment 1** - Mean and median number of iterations for the regular EM algorithm and the approximative variant on varying levels of Δ.

		No. of iterations	
EM variant	Δ	Mean	Median
Regular	-	6.6	7
Approx.	0.1	7.0	7
Approx.	0.5	13.9	8
Approx.	1	34.4	10

$\Delta = 0.5$ and especially for $\Delta = 1$. The larger mean indicates that there might exist a tail of high values skewing the mean.

By inspecting Figure 6B we find that there are indeed a few extreme observations that disrupt the convergence of the mean, creating several large spikes in the convergence curve. We desire a curve that might initially fluctuate, but eventually stabilizes towards some value, as seen in Figure 6A for the regular algorithm. By analyzing the particular data sets with a very large number of iterations we find that two of these have significantly erroneous parameter estimates for all parameters.

We observe this behaviour in this experiment for $\Delta = 0.5$ as well, but not for $\Delta = 0.1$. The convergence to unwanted local maxima is a known shortcoming of the EM algorithm, and one that is exemplified and discussed further in section 5.1.

After removing these two erroneous estimates for $\Delta = 1$ we obtain a reduced mean, but the convergence is still compromised by one extreme observation, as seen in Figure 6C. This observation has converged to the desired local maximum, but a very large number of iterations (9617) was required for converge. By running five identical simulations with different random seeds we find that the same does not occur in repeated simulations (see Figure 6C).

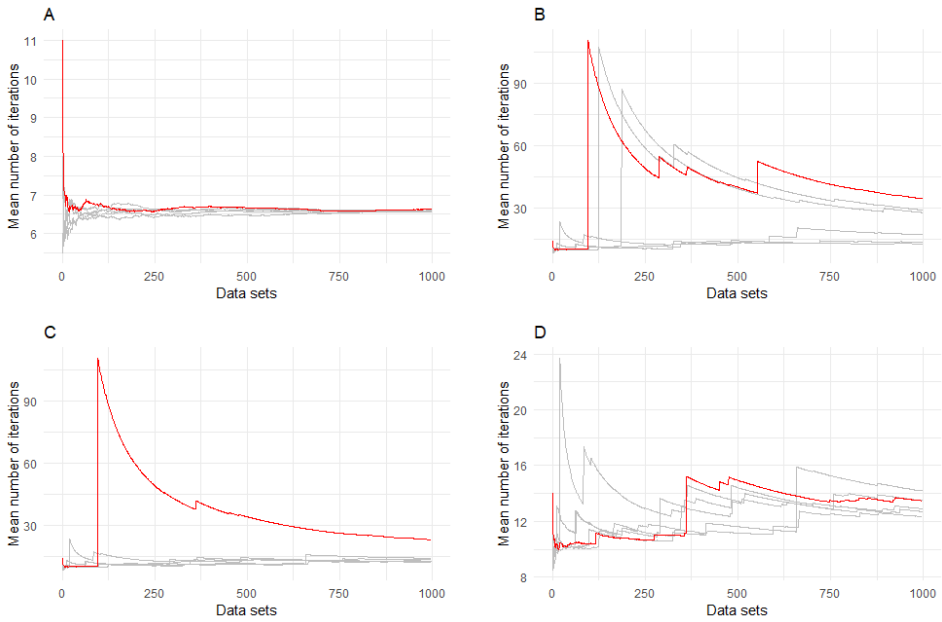

Figure 6: **A**-**D** visualizes the convergence of the mean number of iterations until convergence of the regular EM algorithm (**A**) and the approximative variant with $\Delta = 1$ (**B**-**D**). The curve highlighted in red corresponds to the results presented in Table 4. The grey curves visualize five different simulation runs for comparison. **A**: Regular EM algorithm. **B**: Approximative variant with no data sets removed. **C**: Approximative variant with two data sets removed. Both observations corresponded to undesired parameter estimates. **D**: Approximative variant with the two data sets from **C** removed, as well as an additional extreme data set that corresponds to a desired parameter estimate with a long convergence time.

If we remove the extreme observation the convergence is similar to that of the other simulation runs (Figure 6D). This indicates that the extreme observation is not systematically reoccurring and we choose to regard it as an outlier. Table 4 illustrates some summarizing statistics after removal of the two erroneous observations as well as the one with a very large number of iterations. By considering Figure 6D we note that the convergence is unstable and does not seem to level out, even after removing the unjustified outlier. This indicates that a longer simulation run is required to make a proper conclusion on the convergence of the mean for $\Delta = 1$.
The number of iterations aside, as we remove the undesired estimates, the standard errors for the parameter estimates greatly improve, as seen by comparing tables 5 and 2. Removing the desired but extreme observation does not impact the estimates and is therefore excluded from the table. The trend of having

Table 4: **Experiment 1** - Mean, median and max number of iterations for the regular EM algorithm and the approximative variant after removing RM extreme observations, with $R = 1000$, $n = 1000$ and for varying levels of Δ.

			No. of iterations		
EM variant	Δ	RM	Mean	Median	Max
Regular	-	-	6.6	7	11
Approx.	0.1	-	7.0	7	11
Approx.	0.5	1	11.9	8	2468
Approx.	1	2	23.1	10	9617
Approx.	1	3	13.5	10	1475

Table 5: **Experiment 1** - Standard errors of representative parameter estimates from the regular EM algorithm and the approximative variant after removing RM undesired observations, with $R = 1000$, $n = 1000$ and on varying levels of Δ. See Table 11 in appendix for more information.

			Standard error of						
EM variant	Δ	RM	$\hat{\mu}_{11}$	$\hat{\mu}_{22}$	$\hat{\Sigma}_{111}$	$\hat{\Sigma}_{112}$	$\hat{\Sigma}_{222}$	$\hat{\Sigma}_{212}$	$\hat{w}$
Regular	-	-	0.045	0.047	0.064	0.047	0.066	0.048	0.016
Approx.	0.1	-	0.045	0.047	0.064	0.047	0.066	0.048	0.016
Approx.	0.5	1	0.046	0.048	0.069	0.049	0.071	0.049	0.016
Approx.	1	2	0.047	0.049	0.107	0.052	0.108	0.052	0.016

larger errors with larger Δ is still apparent but weaker. The largest increase is seen for the estimates of Σ_{111} and Σ_{222}, which both increase from around 0.065 when $\Delta = 0.1$ to 0.107 for $\Delta = 1$ (relative increase of almost 65%). The remaining parameters experience an increase of about $0.002 - 0.005$ (4% relative increase).

The issue of inflated iteration means and parameter standard errors will prove to reoccur throughout the experiments and the following tables will therefore only present results after removal of reoccurring local maxima convergences. The number of removed observations are indicated by the RM column. The full tables with unedited results can be found in section 6.6 of the appendix and will be referred to in table captions.

4.1.1 Assuming homoscedasticity

A method of alleviating the issue of undesired local maxima is to impose the restriction of homoscedasticity, i.e. equal variance between the two clusters. This assumption is common in implementations of the EM algorithm even when there is no real natural or physiological reason for this assumption to be true [4][6].

After applying the restriction to the algorithms, we simulate $R = 1000$ new

Table 6: **Experiment 1** - Standard errors of representative parameter estimates from the regular EM algorithm and the approximative variant under the restriction $\mathbf{\Sigma}_1 = \mathbf{\Sigma}_2 = \mathbf{\Sigma}$ after removing RM undesired observations, with $R = 1000$, $n = 1000$ and on varying levels of Δ. See Table 12 in appendix for full results.

			Standard error of				
EM variant	Δ	RM	$\hat{\mu}_{12}$	$\hat{\mu}_{21}$	$\hat{\Sigma}_{22}$	$\hat{\Sigma}_{12}$	w
Regular	-	-	0.045	0.045	0.047	0.032	0.016
Appr.	0.1	20	0.044	0.045	0.047	0.032	0.016
Appr.	0.5	19	0.044	0.045	0.052	0.033	0.016
Appr.	1	23	0.045	0.046	0.097	0.034	0.016

data sets according to the aforementioned parameters and analyze the standard errors of the estimated parameters for both algorithms. We observe all data sets carefully and note that there is a large increase in the number of observations converging to the undesired local maximum, on all levels of Δ. After removing these erroneous data we obtain the result of Table 6.

By comparing tables 6 and 5 we note that, for all variants, on all levels of Δ, the errors for the $\boldsymbol{\mu}_j$ parameter estimates see a marginal decrease of $5-7\%$, while the errors for the Σ_{j11} and Σ_{j22} estimates decrease by around 0.01-0.02 for both $j = 1, 2$, which corresponds to a relative decrease of between 25-35% depending on parameter. The largest decrease in error is observed for $\hat{\Sigma}_{112}$ and $\hat{\Sigma}_{212}$ – the covariance estimates, which decrease by 35% from 0.052 to 0.034. The parameter estimates of Σ_{111} and Σ_{222} see the smallest decrease, from 0.107 to 0.97, and still show the largest errors of any parameter estimates. The observed decrease in standard errors for the $\hat{\mathbf{\Sigma}}_{jkl}$ agrees with our expectation, given that this is indeed the restricted parameter, and especially given that the underlying assumption of homoscedasticity *is actually true*.

We find that even after removing undesired results there still exist a large discrepancy between mean and median number of iterations for the approximative variant. Figure 7 shows that applying the restriction has created a heavy tail of large iterations and increased the variance for the approximative variant on all levels of Δ. The regular EM algorithm sees a decreased mean number of iterations and maintains a similar variance. See Table 13 of the appendix for full results.

The restriction improves the parameter estimates but makes the algorithm more unreliable, as the number of erroneous results has increased by an order of magnitude(!) – from around 0.2% to 2%, as well as increasing the variance of the number of iterations until convergence. If there exists feasible means to sort out the undesired results, this is an effective method to increase the accuracy of the estimates. However, if the underlying assumption of homoscedasticity

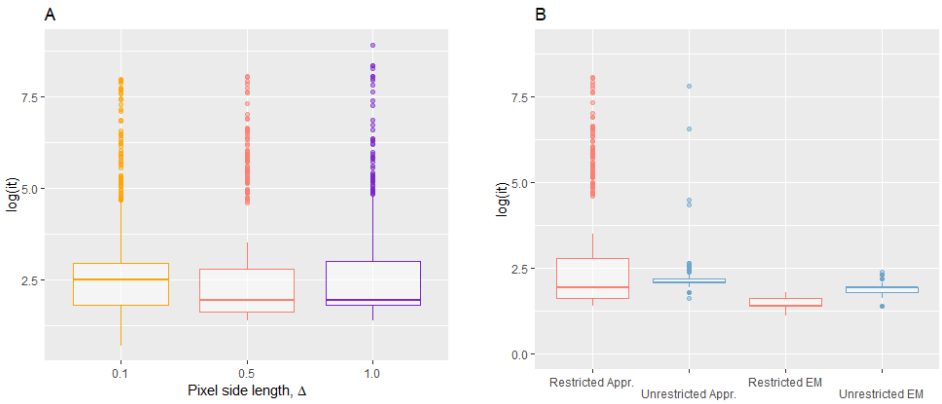

Figure 7: **A**: Boxplot of the natural logarithm of number of iterations until convergence for the approximative EM algorithm on $R = 1000$ dataset on three levels of Δ and assuming homoscedasticity. **B**: Boxplot of the natural logarithm of number of iterations for both algorithms with and without the assumption of homoscedasticity, with $R = 1000$, $n = 1000$ and $\Delta = 0.5$.

is unrealistic and it is hard to interpret the results visually, perhaps as a consequence of high-dimensional data, then this restriction is not optimal for the approximative variant.

4.2 Experiment 2 – Impact of initial values on parameter estimations and iterations

In section 4.1 the algorithms were initiated with the true parameter values as initial guesses for the weights and covariance matrices – only the initial guess of the means were unprecise. "Unprecise" should also be taken lightly, since we used k-means clustering to provide the initiating mean vectors. This provided us with an uncompromised point of reference but most likely leads to a decreased mean number of iterations and more precise parameter estimates. To investigate how sensitive both algorithms are to inaccurate guesses of the different parameters we will now fix all parameters except one at a time and use the same methodology as in section 4.1, without any restrictions.

Since we have seen similar trends at varying levels of Δ we choose to only regard the case when $\Delta = 0.5$ and consider these results as representative for the approximative algorithm when applied to these data sets. However, we do keep in mind that there might be nuances in the results for other Δ that are lost by doing this.

Unless specified, we will use $w_1^{(0)} = w_2^{(0)} = 0.5$, $\boldsymbol{\Sigma}_1^{(0)} = \boldsymbol{\Sigma}_2^{(0)} = \mathbb{I}_2$ and k-means for $\boldsymbol{\mu}_j^{(0)}$ to initiate the algorithms in the coming sections.

4.2.1 Disruptive initial values of mean vectors

Here we choose to disrupt the algorithms by using $\boldsymbol{\mu}_1^{(0)} = (0, 4)^T$ and $\boldsymbol{\mu}_2^{(0)} = (6, 2)^T$ as initiation. These guesses are chosen since they are not too close the undesired local maxima in the middle between the clusters and are equally far ($\sqrt{10}$ units in Euclidean distance) from their respective true values. This way the guess is disruptive in a sense that we are not close to the true value, but at the same time generous by hopefully steering the algorithm away from undesired local maximum results.

After removing one observation from the results of the approximative variant initiated by k-means we find no differences in the parameter estimates between the different initiation methods, and the same size of errors as in the canonical experiment. The only differences that occur are observed at the sixth and seventh decimal place (the largest relative difference being approximatively .0005%). The results suggest that neither of the two algorithms are sensitive to this particular disruptive initial guess of the two mean vectors. However, to state this for disruptive mean vectors in general we would have to explore many other scenarios as well. Full results are found in Table 14 of the appendix, section 6.6.

As in experiment 1 there is one undesired observation which greatly inflates the mean number of iterations for the approximative EM with k-means initiation. We find two more extreme observations which converged to the desired local maxima. If we only remove the one faulty observation the disruptive guess decreases the mean, whilst if all three extreme observations are removed the disruptive guess increases the mean by 5 iterations. Full results are found in Table 15 of the appendix.

From Figure 8 we find that both algorithms increase in iterations for the majority of the data sets, with the regular algorithm increasing slightly more. However, unlike the regular algorithm, the approximative variant also sees an increase in variance of the convergence in all data sets, reconfirming that the approximative variant is less reliable when it comes to convergence rate.

For the approximative variant there are 16 data sets that see a decrease in number of iterations required for convergence when using the disruptive guess. Three of these differences were considerably larger than the others and decreased with 699, 2001 and 2459 iterations respectively. This is attributed to the approximative algorithm being the only one of the four variants (regular and approximate variant with and without disruptive guess respectively) with a high number of iterations on those specific data sets when initiated by k-means but not when using the disruptive guess. These results suggest that the algorithms take a few iterations to first reach the vicinity of the cluster centers from the disruptive guess before converging at an equal rate as when undisrupted.

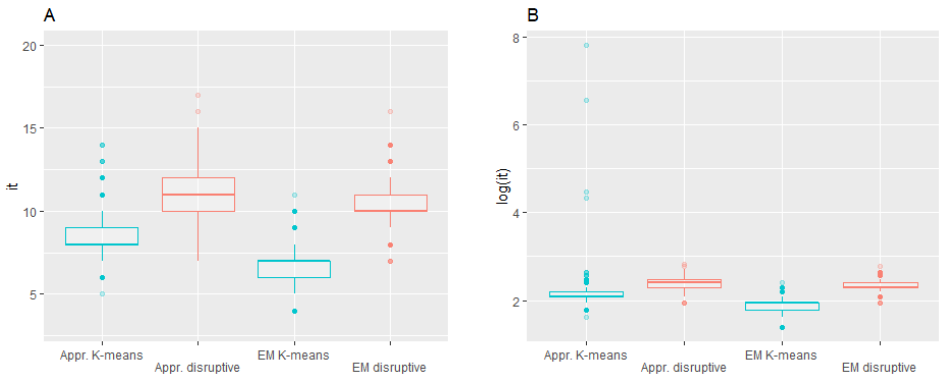

Figure 8: Boxplots of number of iterations for the regular EM algorithm (EM) and the approximative algorithm (Appr.) with disruptive $\boldsymbol{\mu}^{(0)}$ (disruptive) and by $\boldsymbol{\mu}^{(0)}$ generated by k-means clustering (K-means). **A**: Linear scale of number of iterations which excludes four observations valued 77, 88, 709 and 2468. These values still affect the statistics but are not visualized for a clearer visualization of the body of the results. **B**: Natural logarithmic scale of number of iterations with no observations removed.

4.2.2 Disruptive initial values of covariance matrices

In this section we try to disrupt the algorithms by using inaccurate initial guesses of $\boldsymbol{\Sigma}_j^{(0)}$ for both $j = 1, 2$. The experiment is carried out as in section 4.2.1, using $\Delta = 0.5$, $n = 1000$ and $R = 1000$. The initial guesses of the covariance matrices are chosen to be $\boldsymbol{\Sigma}_1^{(0)} = \begin{pmatrix} 6 & 3 \\ 3 & 6 \end{pmatrix}$ and $\boldsymbol{\Sigma}_2^{(0)} = \begin{pmatrix} 6 & -3 \\ -3 & 6 \end{pmatrix}$. This allows the variance to be great enough to encompass both clusters, which could potentially produce more undesired results, i.e. erroneous local maxima.

After carrying out the simulations we find that, as in the previous experiments, there are no noteworthy differences in any of the standard errors (results are found in Table 16 of the appendix). The differences appear at the sixth or seventh decimal place for all parameter estimates and are similar to those of the canonical case.

Just as for the case with a disruptive $\boldsymbol{\mu}^{(0)}$, we find from Table 7 that the mean and median number of iterations increase when we introduce the disruptive $\boldsymbol{\Sigma}^{(0)}$-guess. When comparing every particular data set between the two methods of initiation we find that the number of required iterations for convergence increases for every single data set for the regular EM algorithm, on average by 5 iterations.

For the approximative variant we find that the number of iterations increased

Table 7: **Experiment 2** - Summary table for number of iterations of the regular and approximative EM algorithms initiated by disruptive values of $\boldsymbol{\Sigma}_j^{(0)}$ (disr.) and by the correct guess $\mathbb{I}_2$ with $R = 1000$, $n = 1000$ and $\Delta = 0.5$. See Table 17 for full results.

			No. of iterations		
EM variant	$\boldsymbol{\Sigma}^{(0)}$	RM	Mean	Median	Max
Regular	disr.	-	11.7	12	17
Regular	$\mathbb{I}_2$	-	6.6	7	11
Appr.	disr.	1	12.1	12	18
Appr.	$\mathbb{I}_2$	1	8.7	8	72

for all but two of the data sets by on average 3.5 iterations. We conclude that when it comes to convergence, the regular EM algorithm gains more by an accurate initial guess than the approximative variant but none of the algorithms are sensitive to this disruptive $\boldsymbol{\Sigma}^{(0)}$ for producing good parameter estimates.

When initiating the algorithm with an extremely inaccurate $\boldsymbol{\Sigma}^{(0)}$ we found that the parameter estimates were unaffected but that the number of erroneous results increased greatly from 0.1-0.2% to 4-5%. Complete results and some comments are presented in section 6.4 of the appendix.

4.3 Experiment 3 – Impact of overlap between distributions

The data in sections 4.1 and 4.2 is relatively easy to handle for the algorithm since there is little overlapping data and no covariance present. One of the real challenges for a cluster-finding algorithm like EM, is when there is overlapping data and we will therefore now examine how the algorithms handle data with varying degree of overlap.

On the other hand, if we would use exactly the same parameters for both underlying Gaussian distributions and allow them to overlap completely, then it would be *too* hard for the algorithm to infer on the weights due to non-identifiability, since data would just look like (and theoretically be) a sample of $n = 1000$ data points from one single Gaussian distribution. Since the main idea of the experiments is to study how well the algorithms estimates two *different* underlying Gaussian distributions, we will avoid completely overlapping data. This is done by introducing covariance between y_{i1} and y_{i2} by simulating data from a GMM with parameters

$$\boldsymbol{\Sigma}_1 = \begin{pmatrix} 1 & -0.7 \\ -0.7 & 1 \end{pmatrix} \quad \text{and} \quad \boldsymbol{\Sigma}_2 = \begin{pmatrix} 1 & 0.7 \\ 0.7 & 1 \end{pmatrix} \tag{18}$$

This will tilt the two underlying Gaussian distributions in opposite directions of each other, preventing them from overlapping completely, even when $\boldsymbol{\mu}_1 = \boldsymbol{\mu}_2$.

For weights we use $w_1 = w_2 = 0.5$. We will denote the two underlying clusters with 1 and 2, with 1 being the one with a negative correlation (red) and 2 with positive correlation (yellow). To execute the experiment in a consistent manner we will simulate one data set from the underlying GMM and then shift the part of the data set that originated from distribution 1, towards the mean vector of distribution 2, so that the variability within each data set remains the same, and only their relative position is shifted, as illustrated in Figure 9.

Table 8: Table of mean vector coordinates and their Euclidean distance for each step in experiment 3.

Step	$\boldsymbol{\mu}_1$	$\boldsymbol{\mu}_2$	$\|\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1\|$
1	$(2,2)^T$	$(5,5)^T$	$3\sqrt{2}$
2	$(\frac{7}{2},\frac{7}{2})^T$	$(5,5)^T$	$\frac{3}{2}\sqrt{2}$
3	$(5,5)^T$	$(5,5)^T$	0

This will allow us to study the effect of the overlap without having to consider the randomness between different dataset in different steps. We will regard data at three levels of overlap, by decreasing the distance between $\boldsymbol{\mu}_1$ and $\boldsymbol{\mu}_2$

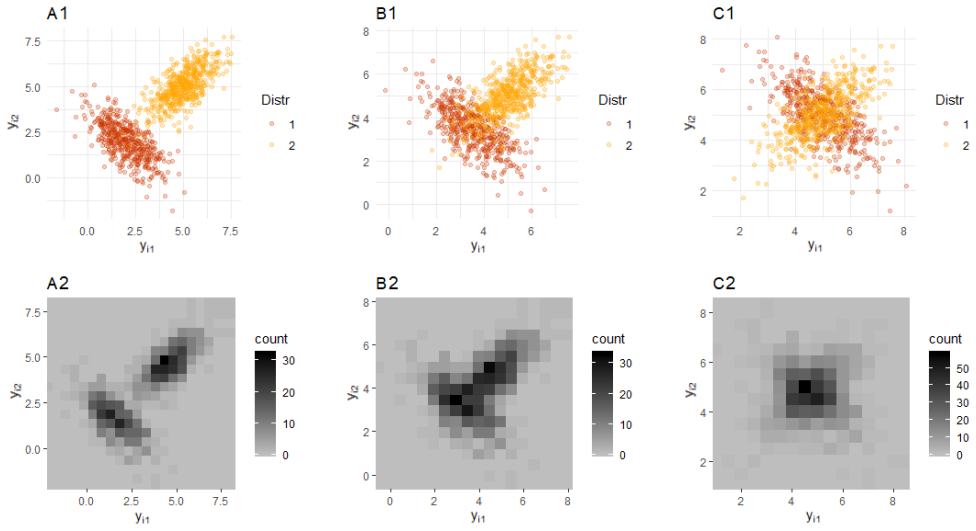

Figure 9: Visualization of one of the $R = 1000$ datasets. Simulated with $\boldsymbol{\Sigma}_1$ and $\boldsymbol{\Sigma}_2$ from equation (18), $w_1 = w_2 = 0.5$, $\boldsymbol{\mu}_j$ according to Table 8 and with $n = 1000$. **A1**: Non-overlapping data (step 1). **A2**: Non-overlapping quantized data (step 1). **B1**: Partially overlapping data (step 2). **B2**: Partially overlapping quantized data (step 2). **C1**: Overlapping data (step 3). **C2**: Overlapping quantized data (step 3). $\Delta = 0.5$.

by shifting $\boldsymbol{\mu}_1$ with 1.5 units in each coordinate direction per step (see Table 8). The steps are solely determined by ocular inspection of the simulated data sets and what to us appear as a reasonably challenging level of overlap.

The Euclidean distance between cluster centers is not always very indicative of the amount of overlap between distributions if unaware of the variance of these. An alternative metric is the *Mahalanobis distance*, which intuitively indicates how many standard deviates the mean vectors are from each other. The metric is defined as:

$$D_M(\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \boldsymbol{\Sigma}) = \sqrt{(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^T \boldsymbol{\Sigma}^{-1} (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)}$$

where $\boldsymbol{\Sigma}^{-1}$ is the inverse of the shared covariance matrix of the two distributions with mean vectors $\boldsymbol{\mu}_1$ and $\boldsymbol{\mu}_2$. However, in our case we have that $\boldsymbol{\Sigma}_1 \neq \boldsymbol{\Sigma}_2$ and thus consider a sort of "mean" covariance matrix by calculating the mean of the elements in $\boldsymbol{\Sigma}_1$ and $\boldsymbol{\Sigma}_2$. From (18) we get that the "mean" covariance matrix is: $\bar{\boldsymbol{\Sigma}} = \frac{1}{2}(\boldsymbol{\Sigma}_1 + \boldsymbol{\Sigma}_2) = \mathbb{I}_2$. The Mahalanobis distance using $\mathbb{I}_2$ becomes

$$D_M(\boldsymbol{\mu}_1, \boldsymbol{\mu}_1, \mathbb{I}_2) = \sqrt{(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^T \mathbb{I}_2 (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)} = ||\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2||, \tag{19}$$

and thus we find it feasible to use the Euclidean distance as a metric here.

We shift data according to Table 8 for each of the $R = 1000$ data sets, and as in previous experiments we will regard the standard errors and mean number of iterations of the two algorithms, with $\Delta = 0.5$ for the approximative variant. As seen in Table 9, the standard error of the estimated $\boldsymbol{\mu}_1$-vector behaves as

Table 9: **Experiment 3** - Standard errors of parameter estimates for the regular EM algorithm and the approximative variant with varying levels of overlap and with $R = 1000$, $n = 1000$ and $\Delta = 0.5$. Full results are found in Table 20 of the appendix.

		Standard error of						
EM variant	Step	$\hat{\mu}_{12}$	$\hat{\mu}_{22}$	$\hat{\Sigma}_{111}$	$\hat{\Sigma}_{112}$	$\hat{\Sigma}_{222}$	$\hat{\Sigma}_{212}$	$\hat{w}$
Regular	1	0.045	0.050	0.061	0.055	0.075	0.066	0.016
Regular	2	0.049	0.079	0.066	0.066	0.100	0.083	0.028
Regular	3	0.050	0.052	0.075	0.078	0.078	0.078	0.041
Appr.	1	0.046	0.050	0.065	0.056	0.080	0.069	0.016
Appr.	2	0.050	0.083	0.070	0.069	0.106	0.088	0.029
Appr.	3	0.052	0.053	0.081	0.082	0.084	0.082	0.044

expected – the more overlap, the higher the error, albeit with marginal impact. The same trend is observed for the error of the $\boldsymbol{\Sigma}_1$- and w- estimates. However, the errors tell us that the estimates of distribution 2 are worse when there is *some* overlap, in step 2, compared to *full* overlap, in step 3, which at first feels counterintuitive.

As seen in Figure 9 there is overlapping data in the lower left of distribution 2. Since this overlap is located mainly around the center of distribution 1 the responsibility for those data points are attributed to distribution 1, which, if true, would mean that distribution 2 has a smaller variance for both y_{i1} and y_{i2}. Meanwhile, in step 3, the overlap takes place around the centers of both underlying distributions and thus does not "amputate" any data from either distribution. That is, it doesn't increase the difficulty of estimating $\mathbf{\Sigma}_2$ compared to estimating $\mathbf{\Sigma}_1$, as in step 2. The estimates are therefore improved in step 3.

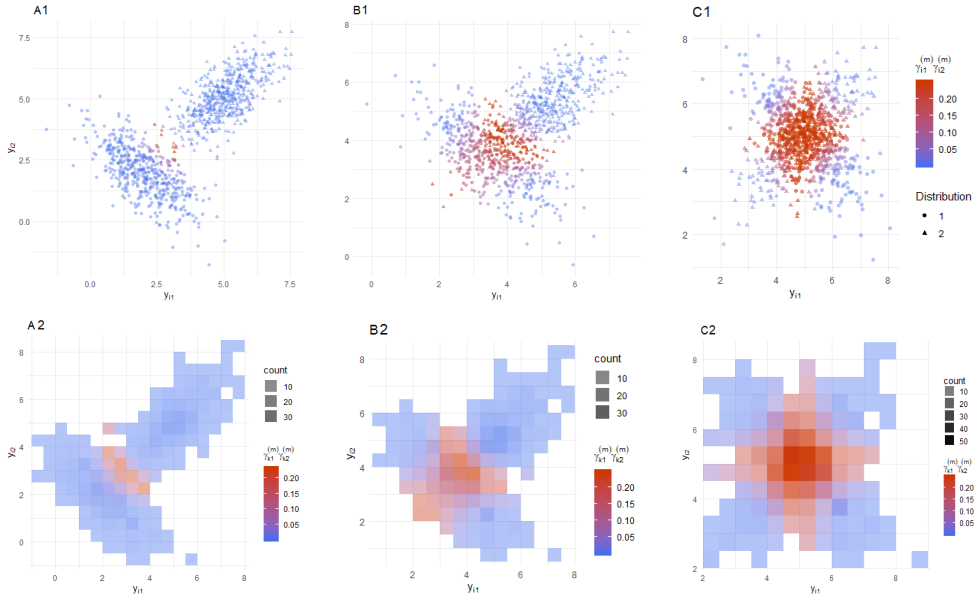

Figure 10: A visualization of $\gamma_{k1}^{(m)} \cdot \gamma_{k2}^{(m)}$ from a representative dataset from the simulations in experiment 3. The darker the pixels indicate a higher N_k (intensity/count). **A1-C1** illustrates steps 1-3 for the regular EM algorithm. **A2-C2** illustrates steps 1-3 for the approximative EM algorithm.

The errors of the weights, on the other hand, increase with each step, for both algorithms. The explanation for this is that the iterative formula for $w_j^{(m+1)}$, (equations (8) and (15) for the regular and approximative variants respectively) is a function only of the relative probabilities in $\boldsymbol{\gamma}^{(m)}$ for the two distributions. When data is overlapping, it is harder to determine which underlying distribution is responsible for each overlapping point, and especially so when they overlap around the centers of both distributions simultaneously.

This is visualized in figures 10C1 and C2 for the regular and approximative

algorithms respectively. The probability is then similar for both distributions, i.e. $\gamma_{ij}^{(m)}$ is close to 0.5 for both $j = 1, 2$, and thus those data points add equal weight to the estimates of w_1 and w_2. If they were not overlapping, the relative probability for some $\boldsymbol{y}_i$ would be closer to either 0 or 1 depending on which cluster it originated from, and would thus contribute more to one $\hat{w}_j$ than the other, as seen in figures 10A1 and A2. As seen in Figure 11, the mean number

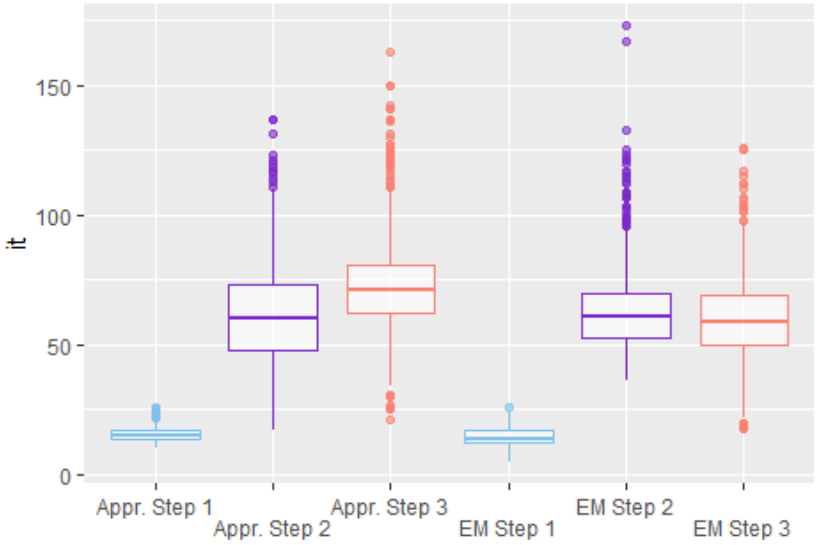

Figure 11: Boxplot of number of iterations until convergence for the regular and approximative EM algorithm in the three steps of experiment 3.

of iterations increases dramatically when introducing overlap, from around 15 iterations in step 1 up to 60 in step 2 for both algorithms, and with a large increase in variance for both variants. In step 3 we see a slight increase for the approximate variant but a decrease for the regular algorithm. Except for this we are unable to discern any difference in trends between the regular and approximative variant, with the latter showing marginally worse errors, as in previous experiments. We note that there are no extreme observations or convergences to undesired local maxima, as in previous examples, and that the median and mean behave equally for both variants of the algorithm for all steps in the experiment.

5 Discussion

To conclude, we have found that it is possible to apply the EM algorithm to the problem of analyzing quantized bimodal data. The experiments have shown that the approximative variant performs relatively well on the data sets handled

in these particular settings, especially for $\Delta = 0.1$. The parameter estimates are consistently worse for the approximative variant than for the regular algorithm and does generally increase with larger Δ. However, the errors are still very much acceptable and in some cases for low Δ similar to those of the regular EM algorithm. We found that initiations by disruptive $\boldsymbol{\mu}_j^{(0)}$ and $\boldsymbol{\Sigma}_j^{(0)}$, when imposed separately, do not seem to affect either algorithms' parameter estimates significantly. However, the approximative variant was found to perform considerably worse when it comes to convergence, both quantitatively and qualitatively. It provides more erroneous results as well as takes a higher number of iterations to converge.

The assumption of equal covariance matrices improved the parameter estimates slightly but on the other hand greatly increased the unreliability of the approximative variant. We did not explore the results when the underlying assumption was false. Both algorithms handled data with two overlapping distributions of equal weight relatively well, but results are most likely depending on the characteristics and degree of overlap, and we cannot draw any conclusions on overlapping data in general.

Below follow some comments on the approximation done in section 3.3, challenges when implementing the algorithm, improvements of the approximative algorithm and undesired convergence towards local maxima.

5.1 Convergence unto undesired local maxima

The problem of convergence unto undesired local maxima is sometimes encountered when applying any of the two algorithms to some data. The algorithms allocate the majority of the responsibility for the observed data points to one cluster – for instance to cluster $j = 1$. This leads to $\hat{w}_1$ increasing towards 1 and consequentially $\hat{w}_2 = 1 - \hat{w}_1$ towards zero. This in turn leads to all $\gamma_{i2}^{(m)}$ tending to zero, and since the iterative formulae for the new estimates include the $\gamma_{ij}^{(m)}$-term, all estimates with $j = 2$ will carry a very small weight and can thus be greatly inflated without affecting the overall likelihood notably. In the canonical case in section 4.1, this leads to $\hat{\boldsymbol{\mu}}_1$ converging to some point in between the two clusters, with the elements of $\hat{\boldsymbol{\Sigma}}_1$ large enough to cover all observations, while $\hat{\boldsymbol{\mu}}_2$ and $\hat{\boldsymbol{\Sigma}}_2$ may take on any arbitrary value within the parameter space Ω. This corresponds to the algorithm finding an undesired local maximum in the probability landscape and converging there, as is visualized in Figure 12.

In the experiments we have used the common stopping criteria of $\|\theta_i^{(m)} - \theta_i^{(m-1)}\| < 10^{-5}$, for all components θ_i of $\boldsymbol{\theta}$. When parameters have different magnitudes, this can lead to some estimate converging before others, and a relative change in estimates might therefore be preferred. The criteria used is more an indicator of halted progress, but not necessarily one of convergence. One could imagine a region in the parameter landscape Ω that is *almost* a sad-

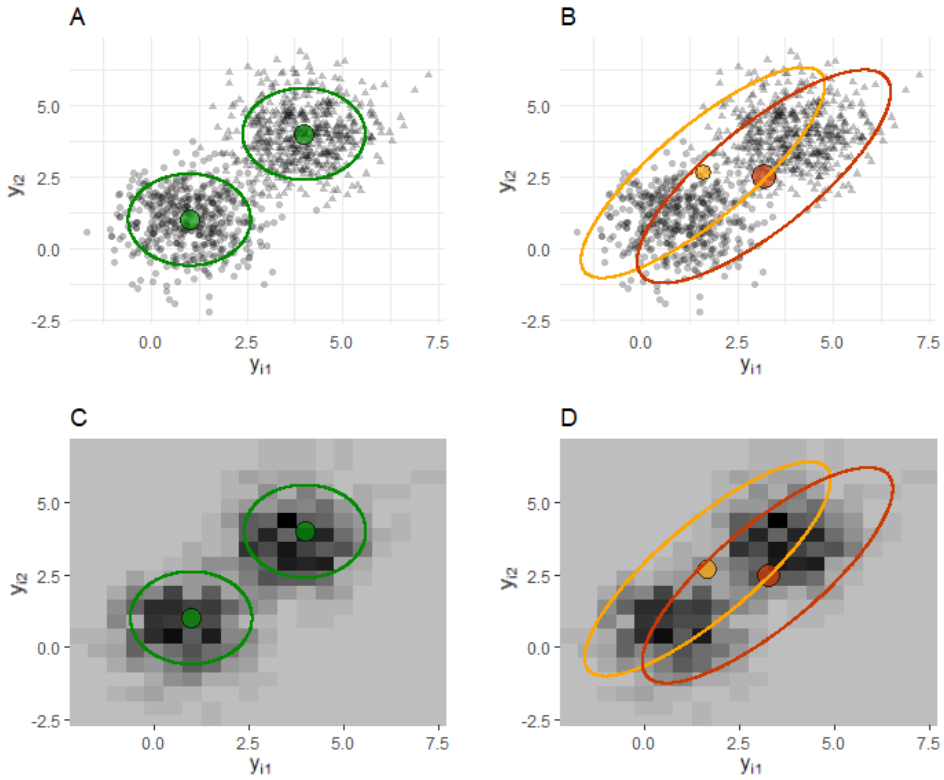

Figure 12: Data simulated from parameters specified in section 4.1 with $n = 1000$ and $\Delta = 0.5$. The filled dots indicate the mean vectors and the ellipses 99% confidence ellipses of the true parameters (green) and their estimates (yellow and red). The relative size of the dots corresponds to the estimates of w_j. **A**: Example of true parameter estimates. **B**: Example of undesired parameter estimates to local maximum corresponding to a local maximum of the likelihood in between clusters, found by regular EM algorithm. **C**: True parameter estimates over quantized data. **D**: Example of undesired parameter estimates to local maxmimum corresponding to a local maximum of the likelihood in between clusters found by approximative variant.

dlepoint, but actually has a very small incremental slope. If the chosen ε is larger than the slope, the algorithm stops there even though this is not a local maxima in the probability landscape [6].

A measure of convergence that would be preferred is the difference in the likelihood function of $\boldsymbol{\theta}^{(m)}$, i.e. to stop the algorithm when $\mathcal{L}(\boldsymbol{\theta}^{(m)}|\boldsymbol{y}) - \mathcal{L}(\boldsymbol{\theta}^{(m-1)}|\boldsymbol{y}) < \varepsilon$ for some small ε. From an implementational point of view this is a clearer measure of convergence, since it is a one-dimensional scalar, whilst the current

implementation applies the same convergence restriction ε to every parameter difference. Perhaps this would have decreased the number of iterations on the extreme observations, as probably there was just one parameter not converging as fast as the others. With the likelihood function as convergence criterion, all parameter estimates contribute to that criterion.

Another way of improving the algorithm is to run the algorithm multiple times on every dataset but with different initial values for $\boldsymbol{\mu}_j^{(0)}$ and then choosing the result with the highest likelihood as the final estimate.

By plotting the estimates, we could observe how they change in each iteration, as well as if there is some parameter that seems to converge prior to the others, and especially at what point erroneous estimates arise. By studying this we could find a reasonable cut-off limit for a maximum number of iterations to decrease the computational time.

With all that said, the current convergence criteria provide reasonable results when it comes to parameter estimates but is not optimal in terms of convergence rate.

5.2 The point estimate approximation

We have to keep in mind that the size of the pixels will stand in relative proportion to the spread of data. A smaller variance requires smaller pixels not to lose too much information when discretizing, and vice versa. If applying this theory to data from another distribution, we have to consider, not the absolute, but the relative size of the pixels in proportion to the spread of the data. A point of further investigation is how the relation between the size of the pixels and the variance of data impacts the estimates.

A possible way of modifying the algorithm would be to incorporate a more precise approximation of $P(\boldsymbol{y}_i \in b_k)$ in $\gamma_{ij}^{(m)}$, which is not required to be a closed expression, since the value is evaluated numerically, rather than solved for analytically. This would theoretically yield a higher precision but would greatly increase the computational cost, as for example Monte Carlo-approximation of each pixel would need at least 1000 operations instead of 1 in order to be precise.

5.3 Improving the approximative EM method

By taking pixels into account where no data points have been observed, the approximative algorithm could be improved. For example, this could prevent the algorithm from converging to the local maxima between the two clusters, as seen in experiment 1 and visualized in Figure 12D, by double-checking if the result is feasible. If the algorithm has produced an erroneous result in an empty or low-intensity pixel, for instance b_k, in between two non-overlapping distributions, we could use the probability of observing none or very few points

in that pixel to decide if the result is probable or not. The number of points b_k is $\text{Bin}(N_k, \pi_k)$ distributed, with $\pi_k = A \cdot p_{\boldsymbol{Y}}(\boldsymbol{c}_k(\boldsymbol{y}_i))$, where $p_{\boldsymbol{Y}}$ is the density function of the GMM, c_k the centerpoint of and A the area of b_k. We could then either choose to reinitiate the algorithm or choose another of the multiple initiation results as the final one, if using these techniques (which we did not implement).

A problem that has reoccurred throughout the implementation of both the regular and the approximate EM algorithm is computer precision. When using the r-function DMVNORM to calculate values of the density function of a multivariate Gaussian distribution, the probability of 0 was occasionally returned for some extreme observations. This led to matrices of non-full rank, NaN:s and crashing code. This was mainly an issue initially when exposing the algorithm to extreme data sets with high covariance and $n < 100$. Introducing the assumption of equal covariance matrices did ease this problem for some cases, but for us to be able to compare the performance of the algorithm with and without this assumption we had to solve it in another manner. One ad hoc solution is to reinitiate the algorithm when faced with this problem, however there was no room to apply this to our implementation. We solved this problem by regarding trivial data sets. In this way when the problem rarely occurred we solved it by changing the random seed until success. Another, more general solution is to imply prior information via a Bayesian approach (then called MAP-estimation), as discussed in section 5.3.1 below [4].

The RELION software used at the SciLifeLab uses MAP-estimation, which allows researchers to introduce prior knowledge about, for example the structures of proteins and features of the interfering noise. The results from RELION also suffer from similar problems that appear in our implementation of the EM algorithm. Some results are discarded by researchers based on prior knowledge of what protein structures are supposed to look like and that the results either do not resemble the desired structure, or that they resemble noise. This is a consequence of the EM algorithm converging to a local maximum and corresponds to our erroneous results visualized in Figure 12.

5.3.1 Maximum A Posteriori (MAP)-estimation in EM

A method that is often used when trying to infer on parameters is to impose some prior information on the parameters. This would probably be an effective way of avoiding erroneous results, if for example imposing prior knowledge on the range of w_j or the size of $\boldsymbol{\Sigma}_j$. The idea is that instead of maximising the log likelihood function we maximise the natural logarithm of the *posteriori* distribution. By Bayes' law we have that:

$$p(\boldsymbol{\theta}|\boldsymbol{x}) = \frac{p(\boldsymbol{x}|\boldsymbol{\theta})p(\boldsymbol{\theta})}{\int_\Omega p(\boldsymbol{x}|\boldsymbol{\theta})p(\boldsymbol{\theta})d\boldsymbol{\theta}} = \frac{p(\boldsymbol{x}|\boldsymbol{\theta})p(\boldsymbol{\theta})}{p(\boldsymbol{x})}, \tag{20}$$

where $\boldsymbol{x} = (\boldsymbol{x}_1, ..., \boldsymbol{x}_n)^T$ and $\boldsymbol{x}_i = (\boldsymbol{y}_i, z_i)$.

The expression with the integral in the denominator of (20), called the *marginal likelihood*, integrates over all $\boldsymbol{\theta} \in \Omega$ and is thus only depending on $\boldsymbol{x}$. Since the denominator $f(\boldsymbol{x})$ does not depend on $\boldsymbol{\theta}$ and since the denominator is positive for all $\boldsymbol{x}$, we yield that

$$\arg\max_{\boldsymbol{\theta}} p(\boldsymbol{\theta}|\boldsymbol{x}) = \arg\max_{\boldsymbol{\theta}} p(\boldsymbol{x}|\boldsymbol{\theta}) p(\boldsymbol{\theta}). \tag{21}$$

We can now conveniently introduce prior information to the algorithm via $p(\boldsymbol{\theta})$, as long as $p(\cdot)$ is a *proper prior*, i.e. that its probability mass integrates to 1, or else the integral in the denominator of equation (20) diverges [6].

In the M-step of the original EM algorithm we maximise $Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(m)})$ with regards to $\boldsymbol{\theta}$, i.e. we want to find the arg max of the joint log likelihood function of $\boldsymbol{Z}$ and $\boldsymbol{y}$, conditioned on $\boldsymbol{\theta}$. By instead regarding considering the posterior distribution and applying the result of equation (21) we can expand the algorithm to take prior information into account:

$$\begin{aligned} &\arg\max_{\boldsymbol{\theta}} E_{\boldsymbol{Z}|\boldsymbol{y},\boldsymbol{\theta}^{(m)}} \left[\log p(\boldsymbol{\theta}|\boldsymbol{Z}, \boldsymbol{y})\right] \\ =&\arg\max_{\boldsymbol{\theta}} E_{\boldsymbol{Z}|\boldsymbol{y},\boldsymbol{\theta}^{(m)}} \left[\log p(\boldsymbol{Z}, \boldsymbol{y}|\boldsymbol{\theta}) + \log p(\boldsymbol{\theta})\right]. \end{aligned}$$

It is practical to use some conjugate prior, for example the inverse gamma distribution, which after maximisation results in an additional additive term in the iterative equations (8) – (10) and (15) – (17).

6 Appendix

6.1 Proof of Result 1 - Independence of Y

Here follow the proof of result 1. It suffices to condition on the latent data $\boldsymbol{Z}$ in the expectation of equation (4):

$$\begin{aligned} Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(m)}) &= E_{\boldsymbol{X}|\boldsymbol{y},\boldsymbol{\theta}^{(m)}}\left[\log p(\boldsymbol{X}|\boldsymbol{\theta})\right] \\ &= \int_{\mathcal{X}} \log p(\boldsymbol{x}|\boldsymbol{\theta}) p(\boldsymbol{x}|\boldsymbol{y},\boldsymbol{\theta}^{(m)}) d\boldsymbol{x} \\ &= \int_{\mathcal{X}} \log p(\boldsymbol{y},\boldsymbol{z}|\boldsymbol{\theta}) p(\boldsymbol{y},\boldsymbol{z}|\boldsymbol{y},\boldsymbol{\theta}^{(m)}) d\boldsymbol{x} \\ &= \int_{\mathcal{X}} \log p(\boldsymbol{y},\boldsymbol{z}|\boldsymbol{\theta}) p(\boldsymbol{z}|\boldsymbol{y},\boldsymbol{\theta}^{(m)})) d\boldsymbol{x} \\ &= \int_{\mathcal{Z}} \log p(\boldsymbol{y},\boldsymbol{z}|\boldsymbol{\theta}) p(\boldsymbol{z}|\boldsymbol{y},\boldsymbol{\theta}^{(m)}) d\boldsymbol{z} \qquad (22) \\ &= E_{\boldsymbol{Z}|\boldsymbol{y},\boldsymbol{\theta}^{(m)}}\left[\log p(\boldsymbol{y},\boldsymbol{Z}|\boldsymbol{\theta})\right]. \end{aligned}$$

■

Here (22) follows as the only random part of $\boldsymbol{x} = (\boldsymbol{y},\boldsymbol{z})$ is $\boldsymbol{z}$ ($\boldsymbol{y}$ is already observed). We can thus equally well integrate over $\mathcal{Z}$.

6.2 Proof of update formulae

We will here provide the derivations of the update formulae for equations (8) - (10). The calculations here follow the proof laid out in [4].

We find $\boldsymbol{\mu}_j^{(m+1)}$ by maximizing (7) with respect to $\boldsymbol{\mu}_j$. This is done by solving the score equation:

$$\begin{aligned} 0 = \frac{dQ(\boldsymbol{\theta}|\boldsymbol{\theta}^{(m)})}{d\boldsymbol{\mu}_j} &= \frac{d}{d\boldsymbol{\mu}_j}\left(\sum_{i=1}^{n}\sum_{l=1}^{p}\gamma_{ij}^{(m)}\log p(\boldsymbol{y}_i, Z_i = l|\boldsymbol{\theta})\right) \\ &= \sum_{i=1}^{n} -\frac{\gamma_{ij}^{(m)}}{2}\frac{d}{d\boldsymbol{\mu}_j}\left(\left(\boldsymbol{y}_i - \boldsymbol{\mu}_j\right)^T \boldsymbol{\Sigma}_j^{-1}\left(\boldsymbol{y}_i - \boldsymbol{\mu}_j\right) + C_1\right), \qquad (23) \end{aligned}$$

where C_1 is independent of $\boldsymbol{\mu}_j$ and thus becomes 0 after differentiation. Similarly, all terms in the sum with $l \neq j$ are constants that disappear after differentiation.

It can be shown from straightforward but tedious algebraic manipulations that if $\boldsymbol{W}$ is symmetric, and $\boldsymbol{x}$ and $\boldsymbol{s}$ are vectors, then

$$\frac{d}{d\boldsymbol{s}}(\boldsymbol{x}-\boldsymbol{s})^T\boldsymbol{W}(\boldsymbol{x}-\boldsymbol{s}) = -2\boldsymbol{W}(\boldsymbol{x}-\boldsymbol{s}).$$

Thus, (23) becomes

$$\begin{aligned}
&\sum_{i=1}^{n} -\frac{\gamma_{ij}^{(m)}}{2}\left(-2\boldsymbol{\Sigma}_j^{-1}\left(\boldsymbol{y}_i-\boldsymbol{\mu}_j\right)\right)\\
&=\sum_{i=1}^{n}\gamma_{ij}^{(m)}\boldsymbol{\Sigma}_j^{-1}\left(\boldsymbol{y}_i-\boldsymbol{\mu}_j\right)\\
&=\boldsymbol{\Sigma}_j^{-1}\sum_{i=1}^{n}\gamma_{ij}^{(m)}\left(\boldsymbol{y}_i-\boldsymbol{\mu}_j\right)\\
&=\boldsymbol{\Sigma}_j^{-1}\left(\sum_{i=1}^{n}\gamma_{ij}^{(m)}\boldsymbol{y}_i-\boldsymbol{\mu}_j\sum_{i=1}^{n}\gamma_{ij}^{(m)}\right),
\end{aligned}$$

where moving $\boldsymbol{\Sigma}_j^{-1}$ and $\boldsymbol{\mu}_j$ outside of the sum is due to them being independent of i.

With $n_j^{(m)} = \sum_{j=1}\gamma_{ij}^{(m)}$, this, when solving for $\boldsymbol{\mu}_j$ yields

$$\boldsymbol{\mu}_j^{(m+1)} \coloneqq \boldsymbol{\mu}_j = \frac{1}{\sum_{i=1}^{n}\gamma_{ij}^{(m)}}\sum_{i=1}^{n}\gamma_{ij}^{(m)}\boldsymbol{y}_i = \frac{1}{n_j^{(m)}}\sum_{i=1}^{n}\gamma_{ij}^{(m)}\boldsymbol{y}_i.$$

■

Similarly, for $\boldsymbol{\Sigma}_j^{(m+1)}$ we set up the score equation:

$$\begin{aligned}
0 &= \frac{dQ(\boldsymbol{\theta}|\boldsymbol{\theta}^{(m)})}{d\boldsymbol{\Sigma}_j} = \frac{d}{d\boldsymbol{\Sigma}_j}\left(\sum_{i=1}^{n}\sum_{l=1}^{p}\gamma_{ij}^{(m)}\log p(\boldsymbol{y}_i, Z_i = l|\boldsymbol{\theta})\right)\\
&=\sum_{i=1}^{n} -\frac{\gamma_{ij}^{(m)}}{2}\frac{d}{d\boldsymbol{\Sigma}_j}\left(\log\det\left(\boldsymbol{\Sigma}_j\right)+\left(\boldsymbol{y}_i-\boldsymbol{\mu}_j\right)^T\boldsymbol{\Sigma}_j^{-1}\left(\boldsymbol{y}_i-\boldsymbol{\mu}_j\right)+C_2\right)\\
&=\sum_{i=1}^{n} -\frac{\gamma_{ij}^{(m)}}{2}\frac{d}{d\boldsymbol{\Sigma}_j}\log\det\left(\boldsymbol{\Sigma}_j\right)+\sum_{i=1}^{n} -\frac{\gamma_{ij}^{(m)}}{2}\frac{d}{d\boldsymbol{\Sigma}_j}\left(\left(\boldsymbol{y}_i-\boldsymbol{\mu}_j\right)^T\boldsymbol{\Sigma}_j^{-1}\left(\boldsymbol{y}_i-\boldsymbol{\mu}_j\right)\right)
\end{aligned} \tag{24}$$

where C_2 is a constant, independent of $\boldsymbol{\Sigma}_j$, which vanishes after differentiation.

By using *Jacobi's formula* one can show that for some differentiable matrix $\boldsymbol{X}$ it holds that

$$\frac{d}{d\boldsymbol{X}}\det\left(\boldsymbol{X}\right) = \det\left(\boldsymbol{X}\right)\left(\boldsymbol{X}^{-1}\right)^T.$$

Using this together with the product rule we get that

$$\frac{d}{d\boldsymbol{\Sigma}_j}\log\det\left(\boldsymbol{\Sigma}_j\right) = \frac{\det\left(\boldsymbol{\Sigma}_j\right)\left(\boldsymbol{\Sigma}_j^{-1}\right)^T}{\det\left(\boldsymbol{\Sigma}_j\right)} = \left(\boldsymbol{\Sigma}_j^{-1}\right)^T = \boldsymbol{\Sigma}_j^{-1},$$

since $\boldsymbol{\Sigma}_j$ is symmetric.

It can also be shown by straightforward but long algebraic manipulations that for some vector $\boldsymbol{s}$ and matrix $\boldsymbol{X}$

$$\frac{d}{d\boldsymbol{X}} \boldsymbol{s}^T \boldsymbol{X}^{-1} \boldsymbol{s} = -\boldsymbol{X}^{-1} \boldsymbol{s}\boldsymbol{s}^T \boldsymbol{X}^{-1},$$

which gives us

$$\frac{d}{d\boldsymbol{\Sigma}_j} \left(\boldsymbol{y}_i - \boldsymbol{\mu}_j\right)^T \boldsymbol{\Sigma}_j^{-1} \left(\boldsymbol{y}_i - \boldsymbol{\mu}_j\right) = -\boldsymbol{\Sigma}_j^{-1} \left(\boldsymbol{y}_i - \boldsymbol{\mu}_j\right) \left(\boldsymbol{y}_i - \boldsymbol{\mu}_j\right)^T \boldsymbol{\Sigma}_j^{-1}.$$

These two results now allow us to express (24) as

$$\begin{aligned}
&\sum_{i=1}^{n} -\frac{\gamma_{ij}^{(m)}}{2} \boldsymbol{\Sigma}_j^{-1} + \sum_{i=1}^{n} \frac{\gamma_{ij}^{(m)}}{2} \boldsymbol{\Sigma}_j^{-1} \left(\boldsymbol{y}_i - \boldsymbol{\mu}_j\right) \left(\boldsymbol{y}_i - \boldsymbol{\mu}_j\right)^T \boldsymbol{\Sigma}_j^{-1} \\
&= -\frac{\boldsymbol{\Sigma}_j^{-1}}{2} \sum_{i=1}^{n} \gamma_{ij}^{(m)} + \frac{\boldsymbol{\Sigma}_j^{-1}}{2} \left(\sum_{i=1}^{n} \gamma_{ij}^{(m)} \left(\boldsymbol{y}_i - \boldsymbol{\mu}_j\right) \left(\boldsymbol{y}_i - \boldsymbol{\mu}_j\right)^T \right) \boldsymbol{\Sigma}_j^{-1}
\end{aligned}$$

Since $\boldsymbol{\Sigma}_j$ is independent of i we can move the two $\boldsymbol{\Sigma}_j^{-1}$-terms outside of the sum and thus after setting the equation equal to 0 we yield the equation

$$\boldsymbol{\Sigma}_j^{-1} \left(\sum_{i=1}^{n} \gamma_{ij}^{(m)} \left(\boldsymbol{y}_i - \boldsymbol{\mu}_j\right) \left(\boldsymbol{y}_i - \boldsymbol{\mu}_j\right)^T \right) \boldsymbol{\Sigma}_j^{-1} = \boldsymbol{\Sigma}_j^{-1} \sum_{i=1}^{n} \gamma_{ij}^{(m)}$$

which after multiplication with $\boldsymbol{\Sigma}_j$ from both sides and then solving for $\boldsymbol{\Sigma}_j$ becomes

$$\boldsymbol{\Sigma}_j^{(m+1)} \coloneqq \boldsymbol{\Sigma}_j = \frac{\sum_{i=1}^{n} \gamma_{ij}^{(m)} \left(\boldsymbol{y}_i - \boldsymbol{\mu}_j\right) \left(\boldsymbol{y}_i - \boldsymbol{\mu}_j\right)^T}{\sum_{i=1}^{n} \gamma_{ij}^{(m)}} = \frac{\sum_{i=1}^{n} \gamma_{ij}^{(m)} \left(\boldsymbol{y}_i - \boldsymbol{\mu}_j\right) \left(\boldsymbol{y}_i - \boldsymbol{\mu}_j\right)^T}{n_j^{(m)}}$$

■

Finding $w^{(m+1)}$ means we want to maximize

$$\sum_{i=1}^{n} \sum_{j=1}^{p} \gamma_{ij}^{(m)} \log w_j + C_3 \propto \sum_{i=1}^{n} \sum_{j=1}^{p} \gamma_{ij}^{(m)} \log w_j = \sum_{j=1}^{p} n_j^{(m)} \log w_j,$$

subject to the restriction $\sum_{j=1}^{p} w_j = 1$. We use the method of *Lagrange multipliers*, which gives the expression

$$0 = \frac{d}{dw_j} \left(\sum_{l=1}^{p} n_l^{(m)} \log w_l - \lambda \left(\sum_{l=1}^{p} w_l - 1 \right) \right) = \frac{n_j^{(m)}}{w_j} - \lambda,$$

which gives the solution $w_j = \frac{n_j^{(m)}}{\lambda}$. By choosing λ so that the restriction is fulfilled we get that

$$w_j^{(m+1)} \coloneqq w_j = \frac{n_j^{(m)}}{\sum_{l=1}^{p} n_l^{(m)}}.$$

■

6.3 Some details on the k-means clustering algorithm

The k-means clustering algorithm is an iterative algorithm that estimates the centeroids of k data clusters. If assuming $k = p = 2$, then the basic idea is the following:

Step 1. Two data points are randomly sampled from the data set as initial guesses for the cluster centeroids.

Step 2. Assigned each data point to its closest respective cluster centroid.

Step 3. Calculate a new centroid for each cluster based on the centerpoints of the allocated data in step 2.

Step 4. Allocated every point anew to the closest of the updated centerpoints.

Step 5. Repeated steps 3-4 until convergence (when no points switch allocation).

The k-means clustering algorithm dates back to 1967 and the EM algorithm can be seen as an expansion of it by instead of allocating every observation $\boldsymbol{y}_i$ to a specific cluster, we allocate a responsibility $\gamma_{ij}^{(m)}$ to each cluster for every $\boldsymbol{y}_i$.

More details can be found in the original paper [8].

6.4 Extreme initial values of covariance matrices

To see how the algorithms handle a completely inaccurate guess we initiate the algorithms with $\boldsymbol{\Sigma}_1^{(0)} = \begin{pmatrix} 100 & 30 \\ 30 & 100 \end{pmatrix}$ and $\boldsymbol{\Sigma}_2^{(0)} = \begin{pmatrix} 100 & -30 \\ -30 & 100 \end{pmatrix}$, $w_1 = w_2 = 0.5$ and $\boldsymbol{\mu}_j^{(0)}$ by k-means. Representative results are presented in Table 10. Because of many singular results in the simulations we only consider $R = 100$ data sets in this experiment, with $n = 1000$ and $\Delta = 0.5$. Yet again we come to the conclusion

Table 10: **Experiment 2** - Standard errors of parameter estimates for the regular EM algorithm and the approximative variant when initiated with the extreme initial guess $\boldsymbol{\Sigma}^{(0)}$ (disr.) and with the correct guess $\mathbb{I}_2$. See Table 18 for full results.

			Standard error of					
EM variant	$\boldsymbol{\Sigma}^{(0)}$	RM	$\hat{\mu}_{12}$	$\hat{\mu}_{21}$	$\hat{\Sigma}_{111}$	$\hat{\Sigma}_{222}$	$\hat{\Sigma}_{212}$	$\hat{w}$
Regular	disr.	5	0.045	0.051	0.064	0.061	0.051	0.016
Regular	$\mathbb{I}_2$	-	0.045	0.052	0.064	0.061	0.050	0.016
Appr.	disr.	4	0.046	0.053	0.071	0.068	0.053	0.016
Appr.	$\mathbb{I}_2$	-	0.045	0.053	0.071	0.067	0.051	0.016

that neither of the algorithms seem to struggle with faulty initial guesses for the covariance matrices, even when they are two orders of magnitude(!) from the true value, $\mathbb{I}_2$. The size of the errors are approximately the same as for the canonical example (compare with row 1 and 3 in Table 5).

However, we do note that with a larger covariance matrix, we also obtained a significantly higher number of undesired results. Even though we only simulated 100 data sets, 5 of these produced undesired estimates for the algorithms. Comparing this with the experiments in sections 4.2.1 and 4.2.2, where there were 1 or 2 undesired results on 1000 data sets, this is a significant increase – from 0.1 - 0.2% to 4 - 5%.

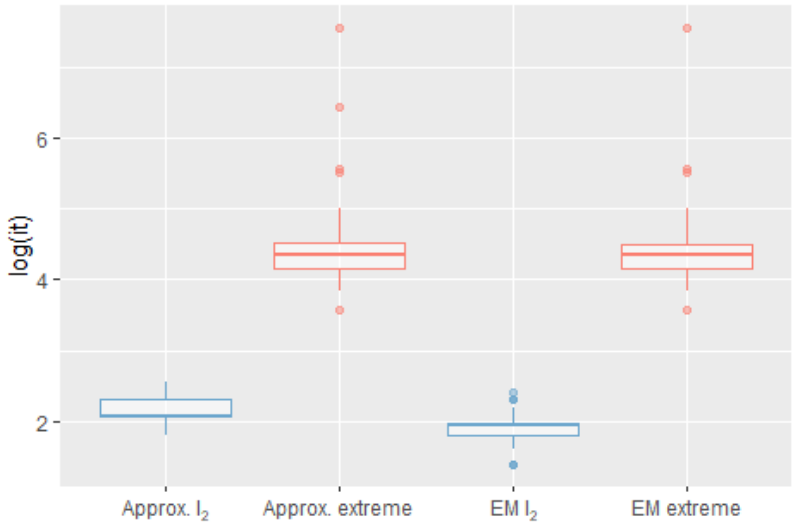

Figure 13: Boxplot of the natural logarithm of the number of iterations until convergence for the regular and approximative EM algorithm when initiated with extremely disruptive $\mathbf{\Sigma}_j^{(0)}$ and with the true value $\mathbb{I}_2$ respectively.

As seen in Figure 13 the extreme initiation dramatically increases the number of iterations, from a mean of 6.8 to almost 103 for the regular algorithm, and from 8.7 to over 156 for the approximative variant. We find that both algorithms are both equally sensitive to this extreme initial guess and even though it takes them longer to do so, they eventually converge to very similar results as with a good initial guess. The full results of iterations can be found in Table 19 in section 6.6 of the appendix.

6.5 Complimentary graphics

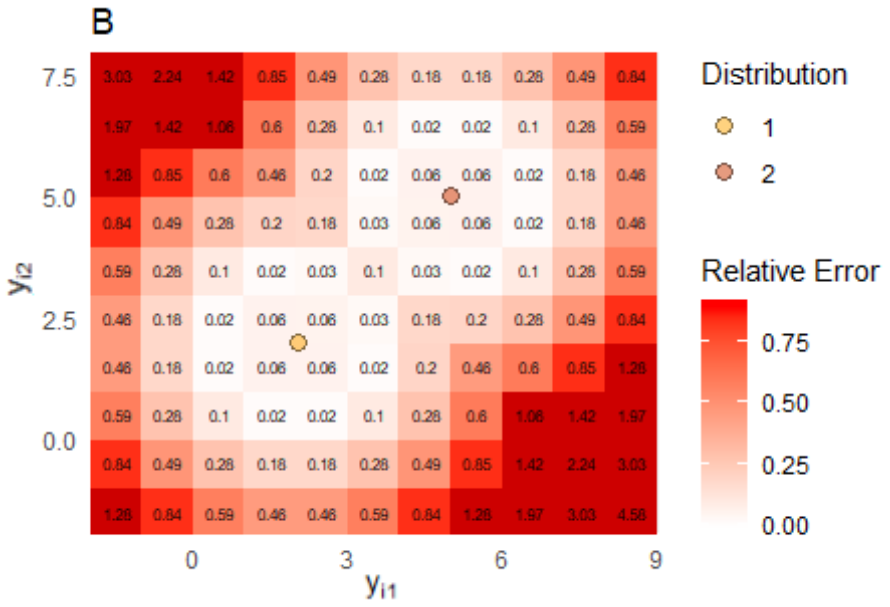

Figure 14: Visualization of relative error between Monte Carlo approximation and the point estimate approximation of $P(\boldsymbol{y}_i \in b_k)$ with $\Delta = 1$ based on a underlying Gaussian Mixture Model with parameters $w_1 = w_2 = 0.5$, $\boldsymbol{\mu}_1 = (2,2)^T$, $\boldsymbol{\mu}_2 = (5,5)^T$ and $\boldsymbol{\Sigma}_1 = \boldsymbol{\Sigma}_2 = \mathbb{I}_2$. The points indicate the two cluster centers $\boldsymbol{\mu}_1$ and $\boldsymbol{\mu}_2$. All relative errors larger than 1 are truncated to the same color to allow for a more interpretable visualization. The color scale of all values less than 1 is directly comparable to Figure 3.

6.6 Full tables

Table 11: **Experiment 1** - Standard errors of all parameter estimates for the regular EM algorithm and the approximative variant after removing RM undesired observations, with $R = 1000$, $n = 1000$ and for varying levels of Δ.

			Standard error of										
EM variant	Δ	RM	$\hat{\mu}_{11}$	$\hat{\mu}_{12}$	$\hat{\mu}_{21}$	$\hat{\mu}_{22}$	$\hat{\Sigma}_{111}$	$\hat{\Sigma}_{122}$	$\hat{\Sigma}_{112}$	$\hat{\Sigma}_{211}$	$\hat{\Sigma}_{222}$	$\hat{\Sigma}_{212}$	$\hat{w}$
Regular	-	-	0.045	0.046	0.045	0.047	0.064	0.064	0.047	0.066	0.066	0.048	0.016
Approx.	0.1	-	0.045	0.046	0.045	0.047	0.064	0.064	0.047	0.066	0.066	0.048	0.016
Approx.	0.5	-	0.077	0.077	0.081	0.076	0.147	0.137	0.140	0.140	0.164	0.125	0.017
Approx.	1	-	0.099	0.106	0.057	0.154	0.207	0.210	0.189	0.187	0.235	0.215	0.025
Approx.	0.5	1	0.046	0.047	0.046	0.048	0.069	0.069	0.049	0.07	0.071	0.049	0.016
Approx.	1	2	0.047	0.048	0.048	0.049	0.107	0.109	0.052	0.108	0.108	0.052	0.016

Table 12: **Experiment 1** - Standard errors of all parameter estimates from the regular EM algorithm and the approximative variant with the restriction $\mathbf{\Sigma}_1 = \mathbf{\Sigma}_2 = \mathbf{\Sigma}$ with $R = 1000$, $n = 1000$ and for varying levels of Δ.

			Standard error of							
EM variant	Δ	RM	$\hat{\mu}_{11}$	$\hat{\mu}_{12}$	$\hat{\mu}_{21}$	$\hat{\mu}_{22}$	$\hat{\Sigma}_{11}$	$\hat{\Sigma}_{22}$	$\hat{\Sigma}_{12}$	$\hat{w}$
Regular	-	-	0.046	0.045	0.045	0.045	0.045	0.047	0.032	0.016
Approx.	0.1	-	0.280	0.273	0.281	0.284	0.563	0.551	0.579	0.040
Approx.	0.5	-	0.284	0.256	0.348	0.250	0.543	0.530	0.562	0.040
Approx.	1	-	0.295	0.303	0.306	0.378	0.604	0.615	0.618	0.045
Approx.	0.1	20	0.046	0.044	0.045	0.045	0.045	0.047	0.032	0.016
Approx.	0.5	19	0.046	0.044	0.045	0.045	0.051	0.052	0.033	0.016
Approx.	1	23	0.048	0.045	0.046	0.046	0.095	0.097	0.034	0.016

Table 13: **Experiment 1** - Mean and median number of iterations for the regular EM algorithm and the approximative variant after removing RM undesired observations, with the restriction $\mathbf{\Sigma}_1 = \mathbf{\Sigma}_2 = \mathbf{\Sigma}$ and with $R = 1000$, $n = 1000$ and for varying levels of Δ.

			No. of iterations	
EM variant	Δ	RM	Mean	Median
Regular	-	-	4.4	4
Appr.	0.1	-	62.8	12
Appr.	0.5	-	51.6	7
Appr.	1	-	66.1	7
Appr.	0.1	20	38.3	12
Appr.	0.5	19	29.8	7
Appr.	1	23	24.6	7

Table 14: **Experiment 2** - Standard errors of all parameter estimates of the regular EM algorithm and the approximative variant after removing RM undesired observations. Both algorithms initialised with a disruptive $\boldsymbol{\mu}_j^{(0)}$ (disr.) and by a k-means generated $\boldsymbol{\mu}_j^{(0)}$ (k-m.) respectively, with $R = 1000$, $n = 1000$ and $\Delta = 0.5$.

			Standard error of										
EM variant	$\boldsymbol{\mu}^{(0)}$	RM	$\hat{\mu}_{11}$	$\hat{\mu}_{12}$	$\hat{\mu}_{21}$	$\hat{\mu}_{22}$	$\hat{\Sigma}_{111}$	$\hat{\Sigma}_{122}$	$\hat{\Sigma}_{112}$	$\hat{\Sigma}_{211}$	$\hat{\Sigma}_{222}$	$\hat{\Sigma}_{212}$	$\hat{w}$
Regular	disr.	-	0.045	0.046	0.045	0.047	0.064	0.064	0.047	0.066	0.066	0.048	0.016
Regular	k-m.	-	0.045	0.046	0.045	0.047	0.064	0.064	0.047	0.066	0.066	0.048	0.016
Appr.	disr.	-	0.046	0.047	0.046	0.048	0.069	0.069	0.049	0.070	0.071	0.049	0.016
Appr.	k-m.	-	0.077	0.077	0.081	0.076	0.147	0.137	0.140	0.140	0.164	0.125	0.017
Appr.	k-m.	1	0.046	0.047	0.046	0.048	0.069	0.069	0.049	0.070	0.071	0.049	0.016

Table 15: **Experiment 2** - Summary table for number of iterations of the regular and approximative EM algorithms after RM removed undesired observations. Both algorithms initiated with disruptive $\boldsymbol{\mu}^{(0)}$ (disr.) and by a k-means generated $\boldsymbol{\mu}^{(0)}$ (k-m.) respectively, with $R = 1000$, $n = 1000$ and $\Delta = 0.5$.

			No. of iterations		
EM variant	$\boldsymbol{\mu}^{(0)}$	RM	Mean	Median	Max
Regular	disr.	-	10.4	10	16
Regular	k-m.	-	6.6	7	11
Appr.	disr.	-	10.7	11	17
Appr.	k-m.	-	13.9	8	2468
Appr.	k-m.	1	11.5	8	2010
Appr.	k-m.	3	8.8	8	88

Table 16: **Experiment 2** - Standard errors of all parameter estimates of the regular EM algorithm and the approximative variant after removing RM undesired observations. Both algorithms initialised with disruptive $\boldsymbol{\Sigma}_j^{(0)}$ (disr.) and by $\boldsymbol{\Sigma}_1^{(0)} = \boldsymbol{\Sigma}_2^{(0)} = \mathbb{I}_2$ respectively, with $R = 1000$, $n = 1000$ and $\Delta = 0.5$.

		Standard error of										
EM variant	$\boldsymbol{\Sigma}^{(0)}$	$\hat{\mu}_{11}$	$\hat{\mu}_{12}$	$\hat{\mu}_{21}$	$\hat{\mu}_{22}$	$\hat{\Sigma}_{111}$	$\hat{\Sigma}_{122}$	$\hat{\Sigma}_{112}$	$\hat{\Sigma}_{211}$	$\hat{\Sigma}_{222}$	$\hat{\Sigma}_{212}$	$\hat{w}$
Regular	$\mathbb{I}_2$	0.045	0.046	0.045	0.047	0.064	0.064	0.047	0.066	0.066	0.048	0.016
Regular	disr.	0.045	0.046	0.045	0.047	0.064	0.064	0.047	0.066	0.066	0.048	0.016
Approx.	$\mathbb{I}_2$	0.046	0.047	0.046	0.048	0.069	0.069	0.049	0.070	0.071	0.049	0.016
Approx.	disr.	0.046	0.047	0.046	0.048	0.069	0.069	0.049	0.070	0.071	0.049	0.016

Table 17: **Experiment 2** - Summary table for number of iterations of the regular and approximative EM algorithms initiated by a disruptive $\mathbf{\Sigma}_j^{(0)}$ (disr.) and by $\mathbf{\Sigma}_1^{(0)} = \mathbf{\Sigma}_2^{(0)} = \mathbb{I}_2$ respectively, with $R = 1000$, $n = 1000$ and $\Delta = 0.5$.

			No. of iterations		
EM variant	$\mathbf{\Sigma}^{(0)}$	RM	Mean	Median	Max
Regular	disr.	-	11.7	12	17
Regular	$\mathbb{I}_2$	-	6.6	7	11
Appr.	disr.	-	13.9	12	1848
Appr.	$\mathbb{I}_2$	-	9.8	8	1122
Appr.	disr.	1	12.1	12	18
Appr.	$\mathbb{I}_2$	1	8.7	8	72

Table 18: **Experiment 2** - Standard errors of all parameter estimates of regular EM algorithm and the approximative variant after removing RM undesired observations. Both initialised with extreme $\mathbf{\Sigma}_j^{(0)}$ (extr.) and by $\mathbf{\Sigma}_1^{(0)} = \mathbf{\Sigma}_2^{(0)} = \mathbb{I}_2$ respectively, with $R = 100$, $n = 1000$ and $\Delta = 0.5$.

			Standard error of										
EM variant	$\mathbf{\Sigma}^{(0)}$	RM	$\hat{\mu}_{11}$	$\hat{\mu}_{12}$	$\hat{\mu}_{21}$	$\hat{\mu}_{22}$	$\hat{\Sigma}_{111}$	$\hat{\Sigma}_{122}$	$\hat{\Sigma}_{112}$	$\hat{\Sigma}_{211}$	$\hat{\Sigma}_{222}$	$\hat{\Sigma}_{212}$	$\hat{w}$
Regular	$\mathbb{I}_2$	-	0.045	0.045	0.052	0.048	0.064	0.062	0.050	0.073	0.061	0.050	0.016
Regular	extr.	-	0.427	0.445	0.384	0.419	0.915	0.819	0.885	0.641	1.168	0.867	0.081
Appr.	$\mathbb{I}_2$	-	0.045	0.045	0.053	0.048	0.071	0.065	0.051	0.078	0.067	0.051	0.016
Appr.	extr.	-	0.380	0.417	0.365	0.397	0.833	0.672	0.817	0.629	1.019	0.783	0.073
Regular	extr.	5	0.046	0.045	0.051	0.048	0.064	0.063	0.050	0.073	0.061	0.051	0.016
Appr.	extr.	4	0.046	0.046	0.053	0.049	0.070	0.065	0.052	0.078	0.068	0.052	0.016

Table 19: **Experiment 2** - Summary table for number of iterations for the regular and approximative EM algorithms initiated by extreme initial guess (extr.) and the correct guess $\mathbb{I}_2$.

			No. of iterations		
EM variant	$\mathbf{\Sigma}^{(0)}$	RM	Mean	Median	Max
Regular	extr.	-	155.3	78.5	1936
Regular	$\mathbb{I}_2$	-	6.8	7	11
Appr.	extr.	-	191.9	79	1913
Appr.	$\mathbb{I}_2$	-	8.7	8	13
Regular	extr.	5	102.9	78	1936
Appr.	extr.	4	156.3	77.5	1913

Table 20: **Experiment 3** - Standard errors of all parameter estimates for the regular EM algorithm and the approximative variant with varying levels of overlap and with $R = 1000$, $n = 1000$ and $\Delta = 0.5$.

		Standard error of										
EM variant	Step	$\hat{\mu}_{11}$	$\hat{\mu}_{12}$	$\hat{\mu}_{21}$	$\hat{\mu}_{22}$	$\hat{\Sigma}_{111}$	$\hat{\Sigma}_{122}$	$\hat{\Sigma}_{112}$	$\hat{\Sigma}_{211}$	$\hat{\Sigma}_{222}$	$\hat{\Sigma}_{212}$	$\hat{w}$
Regular	1	0.045	0.046	0.048	0.050	0.061	0.064	0.055	0.074	0.075	0.066	0.016
Regular	2	0.049	0.049	0.076	0.079	0.066	0.069	0.066	0.095	0.100	0.083	0.028
Regular	3	0.050	0.051	0.051	0.052	0.075	0.076	0.078	0.076	0.078	0.078	0.041
Approx.	1	0.046	0.046	0.048	0.050	0.065	0.068	0.056	0.078	0.080	0.068	0.016
Approx.	2	0.050	0.050	0.080	0.083	0.070	0.073	0.069	0.101	0.106	0.088	0.029
Approx.	3	0.052	0.052	0.052	0.053	0.081	0.082	0.082	0.082	0.084	0.082	0.044

Table 21: **Experiment 3** - Summary table for number of iterations of the regular and approximate EM algorithm respectively for varying levels of overlap with $R = 1000$, $n = 1000$ and $\Delta = 0.5$.

		No. of iterations		
EM variant	Step	Mean	Median	Max
Regular	1	14.4	15	26
Regular	2	63.2	62	172
Regular	3	59.7	59	125
Appr.	1	15.5	17	26
Appr.	2	61.4	59	158
Appr.	3	72.9	71	182

References

[1] Baker, M., "Cryo-electron microscopy shapes up", *Nature*, vol. 561, pp. 565-567, 2018.
Retrieved from: https://www.nature.com/articles/d41586-018-06791-6 [Accessed on May 6th, 2019].

[2] Scheres, S. H. W., "RELION: Implementation of a Bayesian approach to cryo-EM structure determination", *Journal of Structural Biology*, vol. 180, no. 3, pp. 519-530, 2012.
Retrieved from: DOI: 10.1016/j.jsb.2012.09.006.

[3] Jank, W., "The EM algorithm, Its Randomized Implementations for Global Optimization: Some Challenges and Opportunities for Operations Research", *Perspectives in Operations Research*. New York, NY: Springer, pp. 367-392, 2006.
Retrieved from: DOI: 10.1007/978-0-387-39934-8_21.

[4] Gupta, M. R. and Chen, Y., "Theory and use of EM Algorithm", *Foundations and Trends in Signal Processing*, vol. 4, no. 3: pp. 223-296, 2010.
Retrieved from DOI: 10.1561/2000000034.

[5] Dempster, A. P., Laird, N. M., and Rubin, D.B., "Maximum Likelihood from Incomplete Data via the EM Algorithm", *Journal of the Royal Statistical Society*, Series B, vol. 39, no. 1, pp. 1-38, 1977.
Retrieved from: https://www.jstor.org/stable/2984875.

[6] McLachlan, G. and Peel, D., *Finite Mixture Models*. Hoboken, NJ: John Wiley & Sons Inc, 2nd Edition, 2000.
Retrieved from: https://www.annualreviews.org/doi/pdf/10.1146/annurev-statistics-031017-100325.

[7] Sundberg, R. "An Iterative Method for Solution of the Likelihood Equations for Incomplete Data from Exponential Families", *Communications in Statistics - Simulation and Computation*, Series B5, vol. 1, pp. 55-64, 1976.
Retrieved from: http://staff.math.su.se/rolfs/Publikationer/CommStatB1976.pdf [Accessed on May 12th, 2019].

[8] MacQueen, J., "Some methods for classification and analysis of multivariate observations", *Proceedings of the fifth Berkeley Symposium on Mathematical Statistics and Probability*, pp. 281-297, 1967.
Retrieved from: https://projecteuclid.org/euclid.bsmsp/1200512974.