



Swiss Federal Institute of Technology Zurich

Seminar for
Statistics

Department of Mathematics

Bachelor Thesis

Winter 2019

Nicolas Trutmann

Comparison of EM-algorithm and MLE using Cholesky decomposition

Submission Date: placeholder

Advisor: placeholder

Abstract

The intent of this work is to compare The EM algorithm to a MLE approach in the case of multivariate normal mixture models using the Cholesky decomposition. The EM algorithm is widely used in statistics and is proven to converge, however in pathological cases convergence slows down considerably.

methods(not done)

results(not done)

Contents

1	Introduction to normal mixture models	1
1.1	Definitions	1
1.2	The EM-algorithm in sketch	1
1.3	choice of notation	2
1.4	Models of Covariance Matrices	3
1.5	problems of EM	5
2	The <code>norMmix</code> Package	9
2.1	finer details of <code>norMmix</code> package	10
3	Comparing Algorithms	11
3.1	On The Development of <code>norMmix</code>	11
3.2	General Setup	12
3.3	Findings	13
4	Discussion	15
	Bibliography	16

List of Figures

1.1	200 EM steps	7
-----	------------------------	---

List of Tables

1.1	Table of Parameters	4
-----	-------------------------------	-------------------

Chapter 1

Introduction to normal mixture models

1.1 Definitions

A good and thorough introductory book is the work of [McLachlan and Peel \(2000\)](#) and the reader is encouraged to study it to learn in depth about normal mixtures and their clustering. We will here give a short overview of normal mixtures to fix notation and nomenclature. The motivating idea behind mixture models is, that in real world examples a sample might be suspected to arise from more than one population. The original example of this, by Karl Pearson, who fitted two normal distributions with different means and variances. In his paper Pearson analyzed measurements of forehead to body length of crabs sampled from the bay of Naples. His mixture model-based approach suggested, that the crabs were evolving into two new subspecies.

here make clear we restrict to multivariate normal case.

Let $\mu \in \mathbb{R}^p$, $\Sigma \in \mathbb{R}^{p \times p}$ symmetric positive definite and $\phi(-; \mu, \Sigma)$ be the normal distribution with mean μ and covariance matrix Σ .

$\mathbf{Y}_1, \dots, \mathbf{Y}_n$

Definition 1.1.0.1. Suppose we have a random sample $\mathbf{Y}_1, \dots, \mathbf{Y}_n$ with probability density function $\mathbf{Y}_j \sim f(y_j)$ on \mathbb{R}^p . We assume that the density $f(y_j)$ of \mathbf{Y}_j can be written in the form

$$f(y_j) = \sum_{k=1}^K \pi_k \phi_k(y_j; \mu, \Sigma)$$

The π_k are called the component densities of the mixture and the ϕ_k mixture components.

here small note about restricted cases. give ref to 1.4??

1.2 The EM-algorithm in sketch

With this definition we immediately face the problem of how to fit these mixture components to given data. A popular algorithm to solve this problem is the **Expectation-Maximization** algorithm, abbreviated as EM-algorithm.

27 We give here a sketch of the EM-algorithm in the case of all normal mixture components,
 28 since it is the scope of this thesis and simplifies it considerably.

29 Suppose we have a p -dimensional dataset of n samples x_1, \dots, x_n , onto which we would
 30 like to fit K normal distributions ϕ_k , $k \in 1, \dots, K$. We introduce a further explaining
 31 variable \mathbf{Z} in $\text{Mat}^{n \times K}$, with entries in $[0, 1]$ which represent the expectation that observation
 32 i belongs to component k .

33 The EM-algorithm is a two step, iterative process consisting of an 'e'-step and an 'm'-step.
 34 In the e-step the expectation of component membership is updated.

$$\tau_i(y_j; \Psi) = \phi_i(y_j; \mu_i, \Sigma_i) / \sum_{k=1}^K \phi_k(y_j; \mu_k, \Sigma_k)$$

35 and in the m-step given the component membership information we update the component
 36 means and covariances by weighted versions of the usual estimators.

$$\mu_i = \sum_{j=1}^n \tau_{ij} y_j / \sum_{j=1}^n \tau_{ij}$$

37

$$\Sigma_i = \sum_{j=1}^n \tau_{ij} (y_j - \mu_i)(y_j - \mu_i)^\top / \sum_{j=1}^n \tau_{ij}$$

38 here note about initialization methods.

39 1.3 choice of notation

40 The classification of models in this paper relies heavily on the work of [Celeux and Govaert](#)
 41 (1995), however, out of necessity for clarity, we break with their notation. So as to not
 42 confuse the reader we describe here in depth the differences in notation between [Celeux](#)
 43 [and Govaert \(1995\)](#) and ours.

44 The basis of classification in [Celeux and Govaert \(1995\)](#) is the decomposition of a symmet-
 45 ric matrix into an orthogonal and a diagonal component. A symmetric positive definite
 46 matrix Σ can be decomposed as follows

$$\Sigma = \lambda \mathbf{D} \mathbf{A} \mathbf{D}^\top$$

47 with \mathbf{D} an orthogonal matrix and \mathbf{A} a diagonal matrix and $\lambda = \sqrt[p]{\det(\Sigma)}$ the p -th root
 48 of the determinant of Σ .

49 This decomposition has an appealing geometric interpretation, with \mathbf{D} as the *orientation*
 50 of the distribution, \mathbf{A} the *shape*, and λ the *volume*. The problem of notation comes from
 51 standard conventions in linear algebra, where the letters A and D are usually occupied by
 52 arbitrary and diagonal matrices respectively. Furthermore, we intend to apply a variant of
 53 the Cholesky decomposition to Σ , the $\alpha \mathbf{LDL}^\top$ decomposition. This obviously raises some
 54 conflicts in notation.

55 Therefore we, from here on, when referring to the decomposition as described by [Celeux](#)
 56 [and Govaert \(1995\)](#), will use the following modification of notation:

$$\begin{aligned}
D &\mapsto Q \\
A &\mapsto \Lambda \\
\lambda &\mapsto \alpha \\
\Sigma &= \lambda D A D^\top = \alpha Q \Lambda Q^\top
\end{aligned}$$

These were chosen according to general conventions of linear algebra. Q is usually chosen for orthonormal matrices; Λ is often a choice for diagonal matrices eigenvectors and α was somewhat arbitrarily chosen.

1.4 Models of Covariance Matrices

make clear that the models can not be translated one to one to ldl model There is however an issue with the Cholesky decomposition. For 10 out of 14 cases as defined by Celeux and Govaert (1995), there exists a canonical translation of decompositions. The 6 diagonal cases need no translation; the eigen and Cholesky decomposition are equal to identity. For the non-diagonal cases note that for a given sym. pos. def. matrix Σ we have decompositions:

$$\Sigma = \alpha Q \Lambda Q^\top \quad \Sigma = \alpha L D L^\top$$

Since in both cases the bracketing matrices Q and L have determinant 1 the determinant of Σ falls entirely on α . Therefore α , in these particular decompositions, is equal for both. Celeux & Grovaert vary Σ by either varying or holding fixed the volume (α/α_k), shape (Λ/Λ_k) and orientation (Q/Q_k). These 3 times 2 cases would yield the 8 out of 14 cases of non-diagonal cases. However there is no canonical transform for either variable orientation and fixed shape or fixed orientation and variable shape. The reason for this is that in the $L D L^\top$ decomposition the lower diagonal matrix L holds some of the shape of the matrix, which in the eigendecomposition is in the Λ matrix. In fact, L is orthogonal if and only if $L = \text{Id}_{n \times n}$. Therefore we can only decompose matrices where either both or neither shape and orientation vary. See table 1.1.

While we could in theory construct the cases $L D_k L^\top$ and $L_k D L^\top$, however they do not correspond to the desired geometric intent behind the differentiation of models and are therefore not included.

make nice table(maybe sideways to account for parameter list)

Model	Σ_k C&G	volume	shape	orientation	parameters	LDL^\top	parameters	count
EII	$\alpha \mathbf{I}$	equal	equal	-	α	same as C&G		1
VII	$\alpha_k \mathbf{I}$	variable	equal	-	α_k			K
E EI	$\alpha \mathbf{\Lambda}$	equal	equal	coordinate axes	α, λ_i			$1 + (p-1)$
VEI	$\alpha_k \mathbf{\Lambda}$	variable	equal	coordinate axes	α_k, λ_i			$K + (p-1)$
E VI	$\alpha \mathbf{\Lambda}_k$	equal	variable	coordinate axes	$\alpha, \lambda_{i,k}$			$1 + K(p-1)$
V VI	$\alpha_k \mathbf{\Lambda}_k$	variable	variable	coordinate axes	$\alpha_k, \lambda_{i,k}$			$K + K(p-1)$
EEE	$\alpha \mathbf{Q} \mathbf{\Lambda} \mathbf{Q}^\top$	equal	equal	equal	$\alpha, \lambda_i, q_{i,j}$	αLDL^\top	$\lambda, d_i, l_{i,j}$	$1 + (p-1) + \frac{p(p-1)}{2}$
E VE	$\alpha \mathbf{Q} \mathbf{\Lambda}_k \mathbf{Q}^\top$	equal	variable	equal	$\alpha, \lambda_{i,k}, q_{i,j}$	doesn't exist		$1 + K(p-1) + \frac{p(p-1)}{2}$
VEE	$\alpha_k \mathbf{Q} \mathbf{\Lambda} \mathbf{Q}^\top$	variable	equal	equal	$\alpha_k, \lambda_i, q_{i,j}$	$\alpha_k LDL^\top$	$\lambda_k, d_i, l_{i,j}$	$K + p + \frac{p^2(p-1)}{2}$
V VE	$\alpha_k \mathbf{Q} \mathbf{\Lambda}_k \mathbf{Q}^\top$	variable	variable	equal	$\alpha_k, \lambda_{i,k}, q_{i,j}$			$K + K(p-1) + \frac{p(p-1)}{2}$
E EV	$\alpha \mathbf{Q}_k \mathbf{\Lambda} \mathbf{Q}_k^\top$	equal	equal	variable	$\alpha, \lambda_i, q_{i,j,k}$	don't exist		$1 + (p-1) + K \frac{p(p-1)}{2}$
VE V	$\alpha_k \mathbf{Q}_k \mathbf{\Lambda} \mathbf{Q}_k^\top$	variable	equal	variable	$\alpha_k, \lambda_i, q_{i,j,k}$			$K + (p-1) + K \frac{p(p-1)}{2}$
E VV	$\alpha \mathbf{Q}_k \mathbf{\Lambda}_k \mathbf{Q}_k^\top$	equal	variable	variable	$\alpha, \lambda_i, q_{i,j,k}$	$\alpha \mathbf{L}_k \mathbf{D}_k \mathbf{L}_k^\top$	$\lambda, d_{i,k}, l_{i,j,k} \quad j > i$	$1 + pK + K \frac{p(p-1)}{2}$
V VV	$\alpha_k \mathbf{Q}_k \mathbf{\Lambda}_k \mathbf{Q}_k^\top$	variable	variable	variable	$\alpha_k, \lambda_i, q_{i,j,k}$	$\alpha_k \mathbf{L}_k \mathbf{D}_k \mathbf{L}_k^\top$	$\lambda_k, d_{i,k}, l_{i,j,k} \quad j > i$	$K + pK + K \frac{p(p-1)}{2}$

Table 1.1: Table of Parameters

1.5 problems of EM

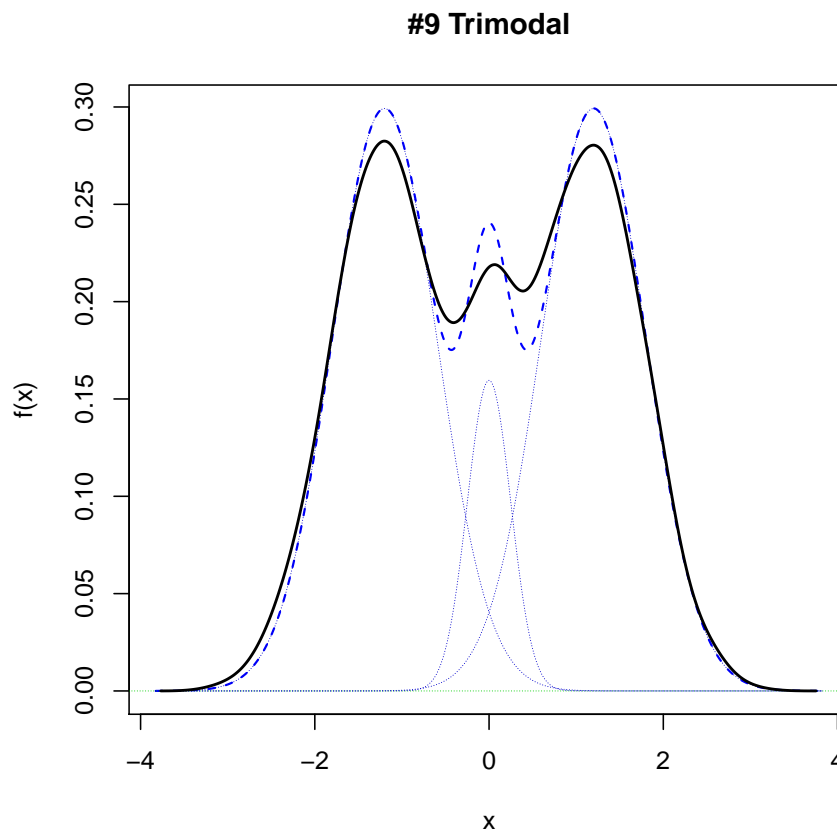
The EM-algorithm has stalling problems especially close to a local optimum. In their seminal work, [Dempster, Laird, and Rubin \(1977\)](#), have proven that the EM-algorithm converges under mild regularity conditions. However, convergence does not guarantee fast convergence. In fact, a lot of the work, that has gone into the research around the EM-algorithm has been concerned with speeding up convergence, see [McLachlan and Peel \(2000\)](#)[section 2.17]. In common software implementations, The concern here is that a slowing in convergence might be mistaken for actual convergence.

This phenomenon is not infrequent and in difficult mixtures quite visible. To illustrate let us look at a particular mixture taken from [Marron and Wand \(1992\)](#) and the `nor1mix` package from CRAN.

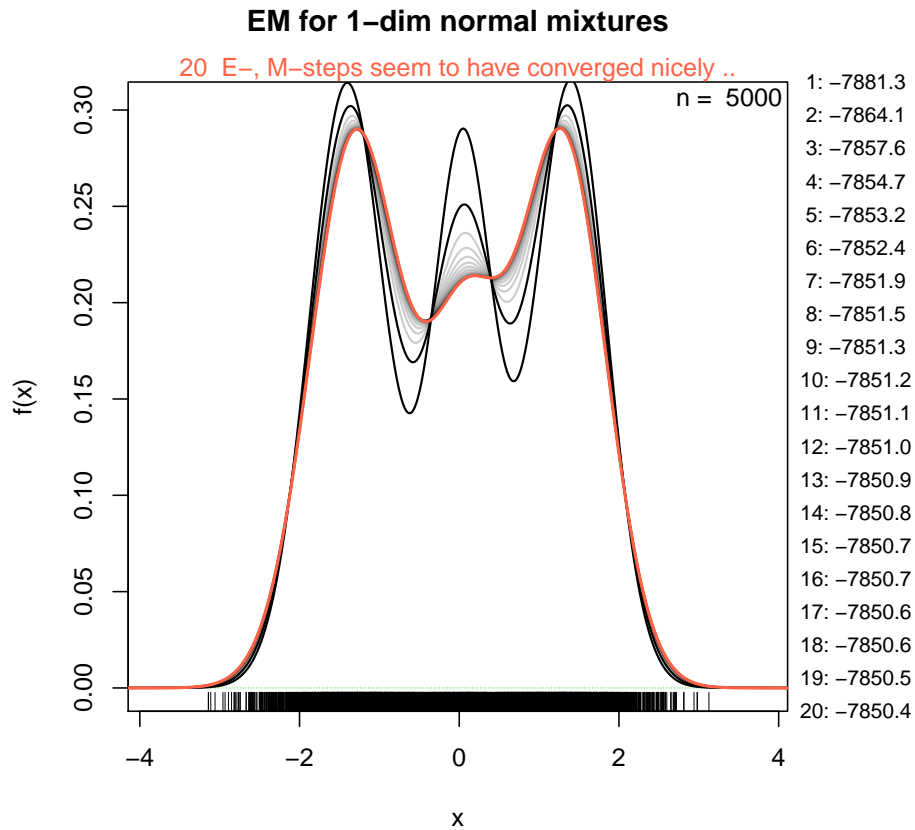
```
> library("nor1mix")
> MW.nm9 ## Trimodal mixture

'Normal Mixture' object      ``#9 Trimodal``
      mu sigma    w
[1,] -1.2  0.60 0.45
[2,]  1.2  0.60 0.45
[3,]  0.0  0.25 0.10
```

show an example using `nor1mix`



94 then an illustration of MW examples of pathological cases



95

96 here we see how change in loglik seems to stagnate. However, this does not stay that way,
 97 if we let EM run a bit further.

98 to conclude example show part of mixest that shows it takes 1200 iterations to converge

99 In fact, it seems that the previous solution is a saddle point in the likelihood function,
 100 where EM has chronic problems continuing improvements.

101 give 2D demonstration.

102 maybe show Marr Wand's examples of 'difficult' mixtures

103 give conclusion recapping the just demonstrated, and lead in for next chapter

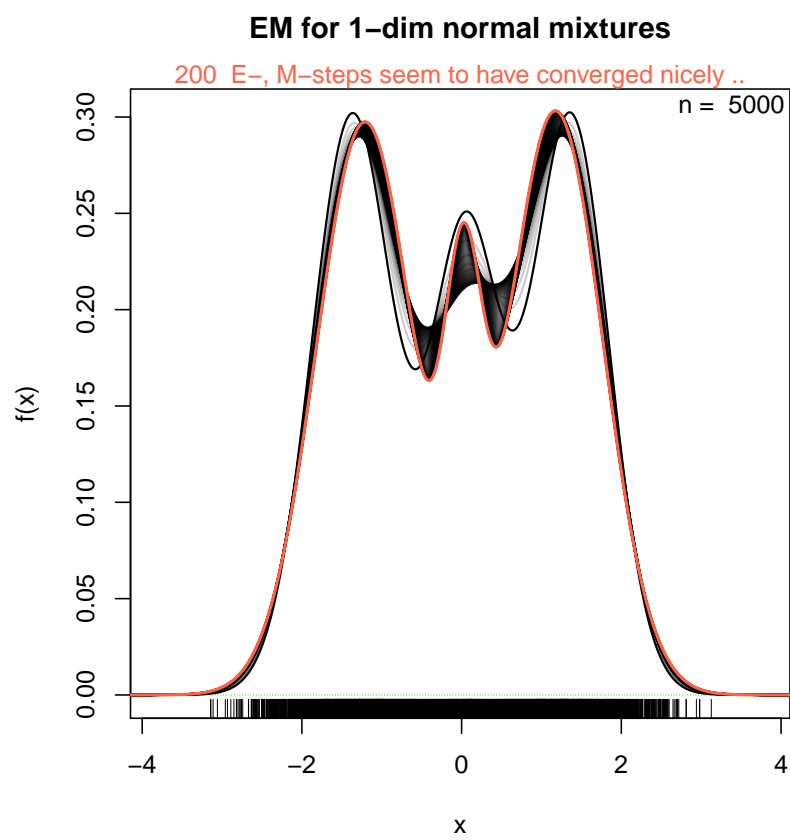


Figure 1.1: 200 EM steps

Chapter 2

The norMmix Package

For this thesis, an R package was developed that implements the algorithm that fits multivariate normal mixtures to given data. There is a lot of unused code still in the package. These were at one point implemented used and discarded. They are still included for demonstration. The norMmix package is constructed around the `norMmix` object, that codifies a normal Multivariate mixture model, and the `llnorMmix()` function, that calculates the log-likelihood given a model and data.

The package contains the following functionality:

`norMmix` `norMmix()` is the 'init-method' for `norMmix` objects. There exist `is.norMmix`, `rnorMmix` and `dnorMmix` functions.

parametrization The main functions that handle reparametrization of models from and to LDL^T decomposition are `nMm2par` and `par2nMm`, which are inverse to each other.

MLE The function `norMmixMLE` marries the main components of this package. It initializes a model and parametrizes it for use with `optim`

fitting Using `norMmixMLE`, the function `fitnMm` allows fitting of multiple models and components. Functions analyzing the output of this are also provided, e.g. BIC and `pring` methods.

misc There are also various methods of generics, like `logLik`, `print`, BIC, AIC and `nobs` as well as various `print` methods.

example objects Following the paper of [Marron and Wand \(1992\)](#) various example objects are provided and used for study. They follow the naming convention: MW + dimension + number. for example `MW213` for the 13th model of dimension 2.

relies on `optim()` generic optimizer. maximizes `llnormix` by varying model parameters.

since `mclust` is one of the more popular packages implementing the EM algo, we employ a lot of functions from `mclust`, to keep things around EM as similar as possible.

Conceptually, at the start of a fitting algo, e.g. EM we need to initialize a mixture object. thereafter the paths diverge. at the heart of `norMmix`'s functionality lie the functions: `llnorMmix` and `nMm2par` which are in turn employed by `norMmixMLE` to funnel a mixture object into `optim` and give `optim` a function to optimize.

also relies on `mixtools` package for random generating function `rnorMmix` using `rmvnorm`.

135 2.1 finer details of norMmix package

136 about Cholesky decomp as ldlt. has advantages: fast, parametrically parsimonious, can
137 easily compute loglikelihood

138 maybe reread section in McLachlan about accelerating EM algo

139 not possible to sensibly compare normal mixtures except maybe a strange sorting algorithm
140 using mahalanobis distance or Kullback-Leibler distance or similar (Hellinger), but not
141 numerically sensible to integrate over potentially high-dimensional spaces.

142 So comparison of algos done through throwing difficult mixtures and non-mixtures at it
143 and hoping that norMmix finds better solutions than EM. So the criteria for "better fit"
144 are 1. better log-likelihood 2. correct model, where EM fails.

Chapter 3

Comparing Algorithms

3.1 On The Development of norMmix

general list of (not necessarily mathematical) dead-ends in the development life of the norMmix package. argue why this is in this section?? because, as a BScT, the learning is as much part of the research as the results.

here on why logit doesnt work

One dead-end was the parametrization of the weights of a mixture using the `logit` function.

```
> logit <- function(e) {
+   stopifnot(is.numeric(e) ,all(e >= 0), all.equal(sum(e),1))
+   qlogis(e[-1L])
+ }
> logitinv <- function(e) {
+   if (length(e)==0) {return(c(1))}
+   stopifnot(is.numeric(e))
+   e<- plogis(e)
+   sp. <- sum(e)
+   w <- c((1-sp.), e)
+ }
```

This uses the logistical function `logis` to transform to reduce the number of weights from K to $K - 1$. Much like `clr1`, given a list of weights `logit` will transform them and `logitinv` will correctly reverse the transformation. However, unlike `clr1`, it will not transform an arbitrary list of length $K - 1$ into a valid weight parameter. For example:

```
> w <- runif(7); ret <- logitinv(w)
> ret

[1] -3.3177307  0.6364078  0.6511485  0.6018082  0.6783043  0.5517875  0.6631182
[8]  0.5351562
```

The issue here is that the last line of `logitinv`, which is necessary to sum to one, but results in a negative value in `ret[1]` which is not a valid weight. The underlying issue is that not every tuple in \mathbb{R}^{K-1} is a result of `logit`.

161 The option to use `logit` is still an argument to `norMmixMLE` by specifying `trafo="logit"`,
 162 but it shouldn't be used.

163 on `forcePositive` and previous m-step, lead to decision to use `mclust` msteps.

164 Another issue during development cropped up during fitting of high dimensional data. We
 165 studied the dataset `SMI.12` from the package `copula`:

```
> data(SMI.12, package="copula")
> str(SMI.12)

num [1:141, 1:20] 16.1 15.7 15.7 16.1 16.6 ...
- attr(*, "dimnames")=List of 2
..$ : chr [1:141] "2011-09-09" "2011-09-12" "2011-09-13" "2011-09-14" ...
..$ : chr [1:20] "ABBN" "ATLN" "ADEN" "CSGN" ...
```

166 A consequence of high dimensions is that matrix multiplication is no longer very stable.
 167 As a result, the covariance matrices produced by our own implementation of the EM-
 168 algorithms m-step (`mstep.nMm`) were not positive definite. In the case of `SMI.12`, several
 169 covariance matrices are degenerate, which results in cancellation error with near-zero en-
 170 tries. We attempted to correct this with the function `forcePositive`, which simply tries
 171 to set \mathbf{D} in \mathbf{LDL}^\top greater than zero. This didn't resolve the issue, since a non-negligible
 172 part of the numerical error was in the \mathbf{L} matrix and the resultant covariance matrix was
 173 still not positive definite.

174 We eventually resolved this issue by abandoning our own implementation and using the
 175 functions from the `Mclust` package. Not only were these numerically stable they were also
 176 able to differentiate between models, whereas ours would assume VVV for every fit.

177 testing of `mvtnorm` as proof that `ldlt` is in fact faster parametrization

178 mention, that there may be faster ways to apply `backsolve`. quote knuth about premature
 179 optimization.

180 3.2 General Setup

181 display abilities of `norMmix` on its own. can find correct models

182 Mention, that `mclust` doesn't depend on `seed`(double check) and therefore `norMmix` has
 183 advantage of 'confidence intervals'. We can run 50 simulations and see if there might be
 184 more sensible clusters.

185 maybe apply to `MW[0-9]` objects?

186 not sure

187 as in Raftery2002, Benaglia2009, Roeder 1997, maybe compare to MISE of various forms.
 188 They all did and see it as adequate method for comparing accuracy of algorithm.

189 also wanted is accuracy of model selection. generate from model and then compare fitted
 190 to original. either by `acc-model==fit-model` and `acc-k==fit-k` or `acc-ll - fit-ll`.

191 **3.3 Findings**

¹⁹² **Chapter 4**

¹⁹³ **Discussion**

Bibliography

- Celeux, G. and G. Govaert (1995). Gaussian parsimonious clustering models. *Pattern Recognition* 28(5), 781 – 793.
- Dempster, A. P., N. M. Laird, and D. B. Rubin (1977). Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society* 39(1), 1–38.
- Marron, J. S. and M. P. Wand (1992). Exact mean integrated squared error. *The Annals of Statistics* 20(2), 712–736.
- McLachlan, G. and D. Peel (2000). *Finite Mixture Models* (1 ed.). Wiley Series in Probability and Statistics. Wiley-Interscience.

Declaration of Originality

The signed declaration of originality is a component of every semester paper, Bachelor's thesis, Master's thesis and any other degree paper undertaken during the course of studies, including the respective electronic versions.

Lecturers may also require a declaration of originality for other written papers compiled for their courses.

I hereby confirm that I am the sole author of the written work here enclosed and that I have compiled it in my own words. Parts excepted are corrections of form and content by the supervisor.

Title of work (in block letters):

Authored by (in block letters):

For papers written by groups the names of all authors are required.

Name(s):

First name(s):

Master	Student

With my signature I confirm that

- I have committed none of the forms of plagiarism described in the Citation etiquette information sheet.
- I have documented all methods, data and processes truthfully.
- I have not manipulated any data.
- I have mentioned all persons who were significant facilitators of the work.
- I am aware that the work may be screened electronically for plagiarism.
- I have understood and followed the guidelines in the document *Scientific Works in Mathematics*.

Place, date:

Signature(s):

Zurich August 19th 2009	bla

For papers written by groups the names of all authors are required. Their signatures collectively guarantee the entire content of the written paper.