Tilted Beta Extremal Mixtures

Samuel Sekarski
Supervisor: Prof. Davison

June 2020

Contents

List of Figures

Chapter 1

Introduction

Modelling extreme events is becoming more and more important, mostly in order to assess risks (financial, ecological, structural, ...). Modelling of univariate extremes is well documented and explored, using techniques such as block maxima, threshold exceedances and point processes. However, things become, as usual, more complicated in higher dimensions. Multivariate extremes suffer from problems that affect univariate extremes less, such as the curse of dimensionality and sparsity.

The most primitive way to deal with multivariate extremes is to study each component as a univariate process. However, this is limiting, as we could easily imaging that there is interdependence of the components, which we lose by considering the components independently. Another reason, as is stated in [2], is that the combination of the individual processes might be of more interest than each process individually.

Methods analogous to block maxima and threshold analysis exist for multivariate cases and we can find models for extreme multivariate events, but we do not have a characterization for the class of all the models. Theorem 8.1 from [2] defines a family of bivariate extreme value distributions (and can be generalized to general multivariate case) that arise as the limiting distribution for componentwise block maxima.

Here is Theorem 8.1 restated (for a bivariate process) for completeness:

Theorem 1 *Let* $M_n^* = (\max_{i=1,\ldots,n}\{X_i\}/n, \max_{i=1,\ldots,n}\{Y_i\}/n)$ *be the vector of rescaled componentwise maxima, where* (X_i, Y_i) *are independent vectors with standard Fréchet marginal distributions. Then if*

$$\mathbb{P}\{M_n^* \leq (x,y)\} \xrightarrow{d} G(x,y),$$

where G *is a non-degenerate distribution function, then* G *has the form*

$$G(x,y) = \exp\{-V(x,y)\}, \quad x > 0, y > 0$$

where

$$V(x,y) = 2\int_0^1 \max\left(\frac{w}{x}, \frac{1-w}{y}\right) d\nu(w)$$

and ν is a distribution on $[0,1]$ *satisfying the mean constraint*

$$\int_0^1 w d\nu(w) = 1/2.$$

The problem is that we don't know how to characterize ν. An approach is to try and approximate the class arbitrarily well, using parametric subfamilies or nonparametric methods and another way is to use nonparametric methods.

Boldi and Davison [3] approached the problem by using a semi-parametric model based on mixtures of Dirichlet distributions that weakly approximates the class of limit distributions.

In this project we will try to use mixtures of beta distributions that have been tilted using Theorem 2 from Coles and Tawn [1] to satisfy the mean constraints.

In Section 2 we will discuss how it is possible to tilt a distribution for it to satisfy the mean constraints, how to sample from a tilted distribution, and provide examples of tilted densities and sampling therefrom. In Section 3 we will explore how to fit a tilted distribution to some data, using maximum likelihood, fit for some artificially generated data and fit from some real world data, and assess the quality of the fits.

Chapter 2

Multivariate extremes

2.1 Basic setup

We will restrict ourselves to the two-dimensional case but some theorems and results will be stated for arbitrary D dimensions. The D-simplex on which our considered distributions will be defined is the set

$$S_D := \left\{ x \in \mathbb{R}_+^D : \sum_{i=1}^{D} x_i = 1 \right\}$$

When $D = 2$, that means that we only need to define a distribution on $x_1 \in [0, 1]$, and $x_2 = 1 - x_1$ is completely determined by x_1. As mentioned in Section 1, we are going to consider distributions that are a mixture of K beta distributions:

$$\nu^*(x_1, x_2) = \prod_{k=1}^{K} \pi_k Beta(x_1; \alpha_k, \beta_k), \quad 0 < x_1 < 1,$$

with

$$\prod_{k=1}^{K} \pi_k = 1, \quad \pi_k \geq 0.$$

The mean of this distribution is

$$\mathbb{E}[X_1] = \sum_{k=1}^{K} \pi_k \frac{\alpha_k}{\alpha_k + \beta_k}, \quad (X_1, X_2) \sim \nu^*,$$

But in general this is not equal to $1/2$ and so this class of distributions does not satisfy Theorem 8.1. In section 2.2 we will see how to tilt a wide class of distributions to force the mean constraint $1/D$ to hold, and will apply it to our case.

2.2 Construction of angular distributions

The main tool for tilting distributions is Theorem 2 from the 1991 paper from Coles and Tawn [1], which we state again here for completeness:

Theorem 2 *If h^* is any positive function on S_D with finite first moments, then*

$$\nu(w) = (m^T w)^{-(D+1)} D^{-1} \left(\prod_{d=1}^{D} m_d \right) \nu^* \left(\frac{m_1 w_1}{m^T w}, \dots, \frac{m_D w_D}{m^T w} \right), \quad (w_1, \dots, w_D) \in S_D,$$

where

$$m_d = \int_{S_D} u_d \nu^*(u) du, \quad d = 1, \dots, D,$$

satisfies mean constraints $1/D$ and is therefore the density of a valid measure function ν.

To verify that the theorem holds, we need to verify that the Jacobian of the transformation

$$W_d = \frac{W_d^*/m_d}{\sum_{c=1}^{D} W_c^*/m_c}, \quad W_d^* = \frac{m_d W_d}{\sum_{c=1}^{D} m_c W_c}, \quad d = 1, \dots, D,$$

is $|\partial w^*/\partial w| = (m^T w)^{-D} \prod_{d=1}^{D} m_d$. To do this, we rewrite the first transformation as

$$w_s^* = \frac{m_d w_d}{m_D + \sum_{c=1}^{D-1} (m_c - m_D) w_c}, \quad d = 1, \dots, D-1.$$

by noting that $w = (w_1, \dots, w_D) \in S_D$. To simplify notation, we write $m^T w = m_D + \sum_{c=1}^{D-1} (m_c - m_D) w_c$. As such,

$$\begin{aligned} \partial w_d^*/\partial w_d &= m_d/(m^T w) - m_d w_d (m_d - m_D)/(m^T w)^2, \\ \partial w_d^*/\partial w_c &= -m_d w_d (m_c - m_D)/(m^T w)^2, \quad c \neq d. \end{aligned}$$

This defines a matrix that we can write as $A + ab^T$, where

$$\begin{aligned} A &= \operatorname{diag}(m_1, \dots, m_{D-1})/(m^T w), \\ a &= (m_1 w_1, \dots, m_{D-1} w_{D-1})^T, \\ b &= -(m_1 - m_D, \dots, m_{D-1} - m_D)^T/(m^T w)^2. \end{aligned}$$

Let's recall the determinant lemma: Let A be an invertible $p \times p$ matrix, and let a, b, be vectors of length p. Then $|A + ab^T| = |A|(1 + b^T A^{-1} a)$.

In our case, A is diagonal, so

$$|A| = (m^T w)^{-(D-1)} \prod_{c=1}^{D-1} m_c,$$

$$b^T A^{-1} a = -\frac{m^T w + m_D}{m^T w},$$

so

$$1 + b^T A^{-1} a = m_D / m^T w.$$

Therefore, $|\partial w^* / \partial w| = (m^T w)^{-D} \prod_{d=1}^{D} m_d$, so the variable $W = (W_1, \ldots, W_D)$ has the probability density function

$$f(w) = (m^T w)^{-D} \left(\prod_{d=1}^{D} m_d \right) \nu^* \left(\frac{m_1 w_1}{m^T w}, \ldots, \frac{m_D w_D}{m^T w} \right)$$

For every d, W_d^* / m_d has unit expectation, which leads to the following equality

$$1 = \mathbb{E}\left(\frac{W_d^*}{m_d} \right) = \mathbb{E}\left(\frac{W_d}{m^T W} \right) = \int_{S_{D-1}} w_d (m^T w)^{-(D+1)} \left(\prod_{d=1}^{D} m_d \right) \nu^* \left(\frac{m_1 w_1}{m^T w}, \ldots, \frac{m_D w_D}{m^T w} \right) \mathrm{d}w$$

by noting that $\sum_d w_d = 1$ and summing the previous equality over d, we get that

$$\nu(w) = D^{-1} (m^T w)^{-(D+1)} \left(\prod_{d=1}^{D} m_d \right) \nu^* \left(\frac{m_1 w_1}{m^T w}, \ldots, \frac{m_D w_D}{m^T w} \right).$$

which is a well defined density on S_{D-1} satisfying the mean constraint

$$\int_{S_{D-1}} w_d \nu(w) \mathrm{d}w = D^{-1}, \quad d = 1, \ldots, D.$$

Sampling from tilted density

As we have seen in the proof of theorem 2, to construct the appropriate density ν, we start with a density ν^*, apply a change of variable to it, then tilt the result by diving it by $D m^T w$, which is bounded, in order to sample from the tilted density, we can sample from the original density ν^*, transform the the samples, then apply the following Acceptance-Rejection step:

$$U \leq m_{\min} / m^T w$$

where $m_{\min} = \min_d m_d$

This is because a realisation W^* of ν^* that has been transformed has density f and not ν. But $m_{\min}/m^T W \leq 1$, so conditional on $W = w$ the event $U \leq m_{\min}/m^T w$ has probability $m_{\min}/m^T w$. Thus the marginal density of the W for which the event occurs is proportional to $f(w)/(m^T w)$ and has to be ν.

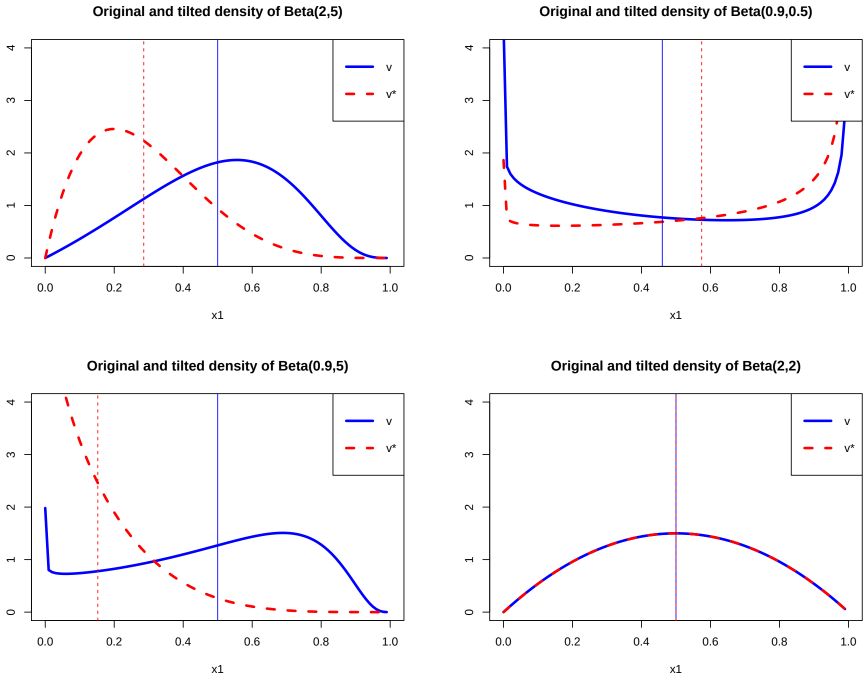


Figure 2.1: Graphs of beta distributions for various shape parameters, before and after tilting. The vertical lines represent the means. As we can see, although the original densities have various means, the tilted densities all have mean 0.5, and when the original distribution already has the correct mean, it is not tilted.

The acceptance probability is given by

$$\int \frac{m_{\min}}{m^T w} f(w) \mathrm{d}w = D m_{\min} \int \nu(w) \mathrm{d}w = D m_{\min}$$

Thus the number of accepted samples is proportional to m_{min}, so the algorithm is the most efficient when all the m_d are equal to $1/D$ which would mean our original density ν^* already satisfies the criteria and we would not have to run the algorithm at all.

Tests

Figure 2.2 some examples of tilted mixtures of beta distributions, sampled using the algorithm, and the theoretical tilted distributions using the formula.

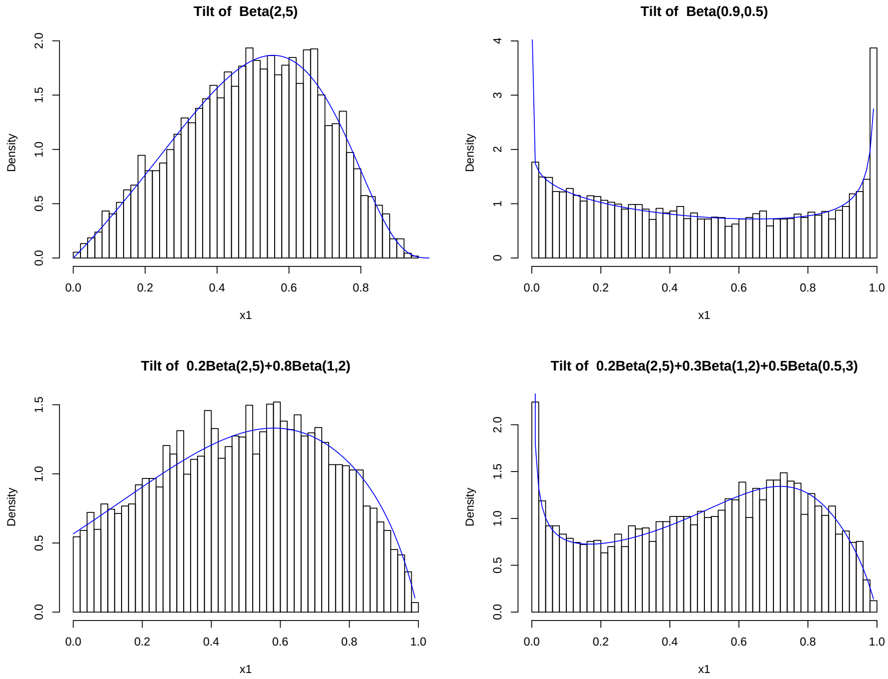


Figure 2.2: Histograms of 10^4 samples from various mixtures of beta distributions that have been tilted. The pdfs of the distributions are overlaid in blue. As we can see, the sampling algorithm for the tilted mixtures is correct.

2.3 Tilted mixtures are dense

To prove that the set of all tilted distributions are dense within the class of all distributions on the p-dimensional simplex S_p which satisfy the mean constraint $\mu = p^{-1}\mathbf{1}_p$, where $\mathbf{1}_p$ is the unit vector in $\mathbb{R}^p$.

Boldi and Davison prove in appendix A of [3] that the class of Dirichlet mixtures that satisfy the mean constraint are weakly dense. From there we may simply remark that a Dirichlet mixtures that satisfy the constraint is itself tilted and thus belongs to the class of all tilted distributions. The final step is to remark that if a subclass of the tilted distributions is already weakly dense, then the whole class is therefore weakly dense as well.

Chapter 3

Statistical aspects

3.1 Likelihood fitting

We use the generic R optimiser *optim* to minimise the negative log-likelihood function. In order to use it we first reparameterized the parameters, in order to express the S_2 constraint in a way that the optimiser can understand.

$$\pi_k = \exp(\eta_k) / \left\{ 1 + \sum_{i=2}^{K} \exp(\eta_i) \right\}, \quad k = 2, \ldots, K,$$

and

$$\pi_1 = 1 / \left\{ 1 + \sum_{i=2}^{K} \exp(\eta_i) \right\}.$$

We also reparametrize the α and β parameters as

$$\alpha_k = \exp(\xi_k), \quad \beta_k = \exp(\zeta_k), \quad k = 1, \ldots, K,$$

to ensure that the parameters can take any values in $\mathbb{R}$. There are $3K - 1$ parameters to estimate, which can then be transformed back into the original parameterization after estimation. Initially, we consider K to be a fixed parameter, and for given data we will try to fit with various values of K, including the real value, to see the accuracy, and if values of K that are close, but not correct give a "good enough" estimation (for example we could use a smaller K than the true K and still have a model that is accurate "enough" for certain purposes. Then we shall try to consider K as a parameter to be estimated as well, using methods such as step-AIC/BIC to select the model.

3.2 Numerical experiments

As a measure of the error between the true density f and the estimated density based on n samples f_n, we will use the integrated square error, defined as

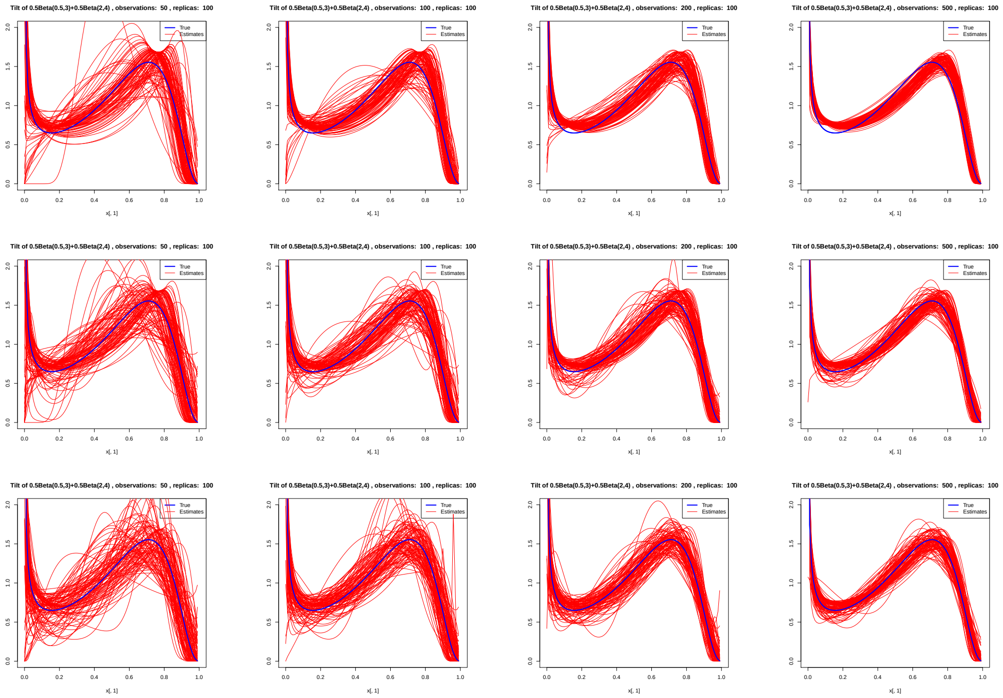


Figure 3.1: MLE fits to samples of size, from left to right, $n = 50$, $n = 100$, $n = 200$ and $n = 500$, with different number of betas in the mixture. From top to bottom: $K = 1$, $K = 2$ (which is the number that generated the data), and $K = 3$, using a *Nelder-Mead* optimiser each time, with a maximum of 500 iterations.

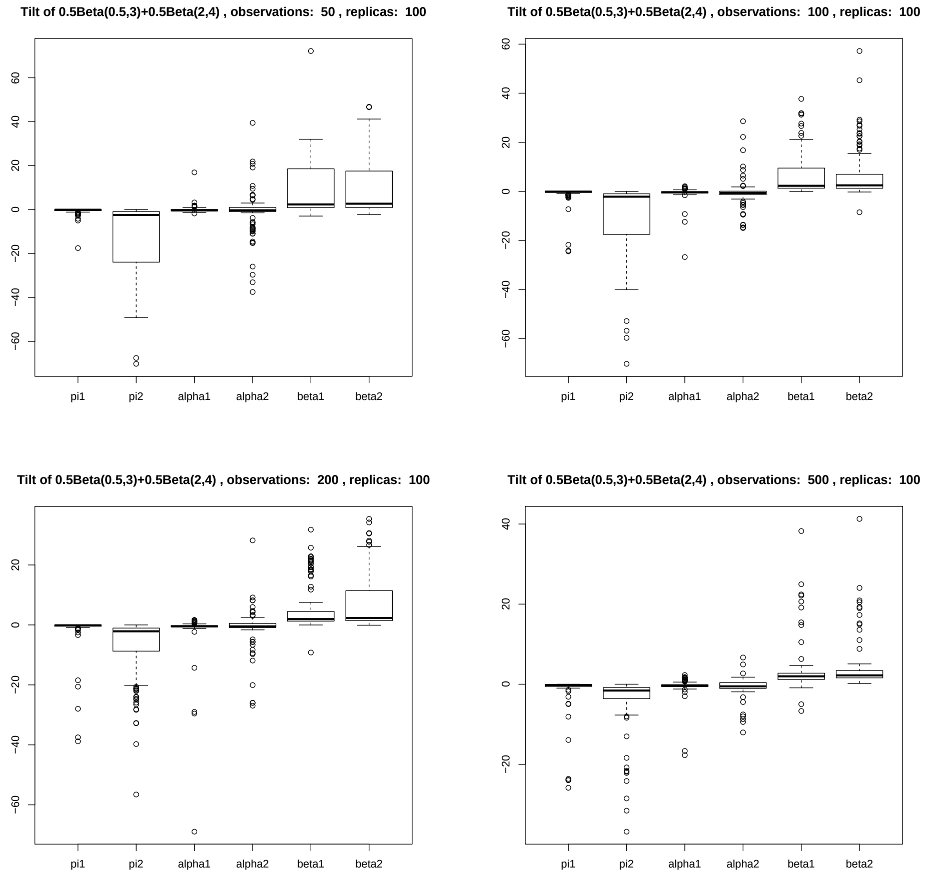
$$\mathrm{err}(f_n) := ||f_n - f||_2^2 = \int_0^1 (f_n(x) - f(x))^2 dx$$

In Section 3.2.1 we fitted a tilted beta mixture to data actually simulated from a tilted beta mixture, then in Section 3.2.2 we fitted a tilted beta mixture to data simulated from other distributions (i.e., not from a beta mixture).

For each test distribution, we simulate $R = 100$ samples of $n = 50$, $n = 100$, $n = 200$, and $n = 500$ independent observations. The Maximum Likelihood Estimation is done with ten different sets of initial parameters chosen uniformly at random between -0.5 and $+0.5$.

3.2.1 Data simulated from mixture

First we generated data using a tilted mix of two beta distributions, $0.5Beta(0.5, 3)+0.5Beta(2, 4)$, to test how well the maximum likelihood estimation would work. In Figure 3.1 we see that for samples with a high number of observations, even



Tilt of 0.5Beta(0.5,3)+0.5Beta(2,4) , observations: 50 , replicas: 100

Tilt of 0.5Beta(0.5,3)+0.5Beta(2,4) , observations: 100 , replicas: 100

Tilt of 0.5Beta(0.5,3)+0.5Beta(2,4) , observations: 200 , replicas: 100

Tilt of 0.5Beta(0.5,3)+0.5Beta(2,4) , observations: 500 , replicas: 100

Figure 3.2: Boxplot of log MLE estimates (Nelder-Mead, maxit 500), for $K = 2$ and sample of 50, 100, 200, 500 observations

estimating with a different number K of components, the results are good, which suggests that we could indeed use a step-AIC/BIC method, starting with a single tilted beta distribution, and keep estimating with more until the added accuracy is no longer significant. Looking at the case n=50, however, we see a lot of volatility and artefacts. Some of the estimations have clearly not converged in 500 iterations of the Nelder-Mead algorithm, even when the correct number K of betas is used.

Looking at the boxplots of the log parameter for $K = 2$, (Figure 3.2) we see that there are a lot of very large and very small values. This makes the analysis of the boxplots difficult. But looking at Figure 3.1 we can guess what is happening: as the algorithm fits relatively well with just just a single beta

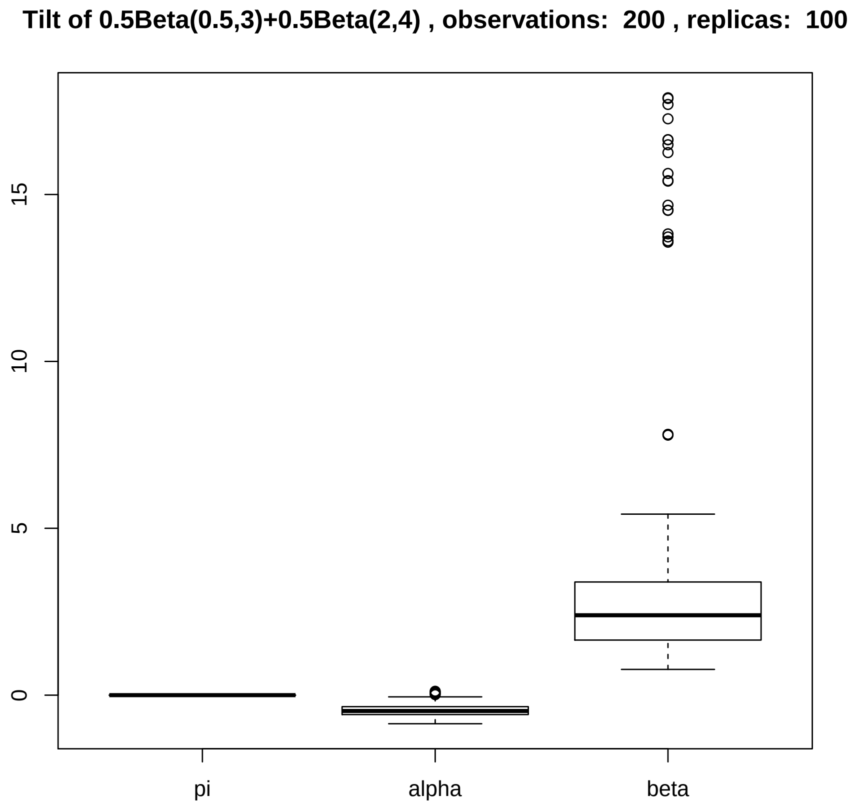


Figure 3.3: Boxplot of log MLE estimates (Nelder-Mead, maxit 500), for $K = 1$ and sample of 200 observations

distribution instead of two, all the cases with a very small π_i, is just the algorithm fitting one of the components very well, which sends the other component to insignificance. As for very small/large values for α_i and β_i, our hypothesis is that in the cases where the algorithm fitted one component really well, and the other component can take pretty much any value it wants, as it's contribution to the mixture has hardly any weight.

To try to confirm this, first we can look at a boxplot of one of the fits with $K = 1$, and indeed if we look at Figure 3.3 we do see that the parameters seem a lot better contained. The group of β outliers could be explained by the fact the original density we are working with increases asymptotically on the left side (the side controlled by the α parameters) so the relative importance of the

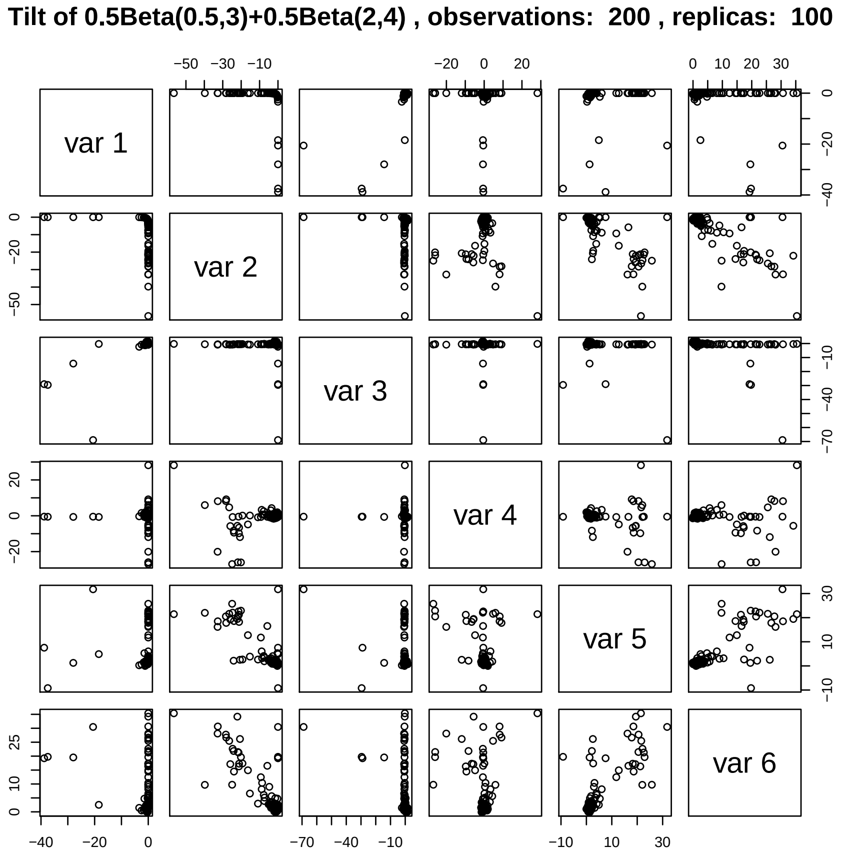


Figure 3.4: pairs plot of log MLE estimates (Nelder-Mead, maxit 500), for $K = 2$ and sample of 200 observations

right side (controlled by the β parameter) might be low.

Another check we can do is have a look at a pairs plot for $K = 2$, to see if low values of π_i correspond to very small/large values of α_i and β_i. In Figure 3.4, we have the pairs plot for $n = 200$ samples. Comparing Var2 (a π_i) vs Var6 (the corresponding β_i), we clearly see a clump down at the right corner, where stuff is happening according to plan, but also that as the weight of the component gets smaller, the value of the β parameter gets larger and more volatile. If we look at Var2 vs Var4 (the corresponding α_i), we also see a nice clump around 0 on the right side, where the weight of the component is still significant, and as the weight gets smaller, the α parameter gets more volatile, with a slight tendency for very small values.

K	n=50	n=100	n=200
1	0.029	0.017	0.008
2	0.040	0.018	0.011
3	0.036	0.022	0.011

Table 3.1: Integrated square error for the beta mixture fits, with $K = 1, 2, 3$ and $n = 50, 100, 200$.

To try and fight this phenomenon, we will try two things: the first is to use a constrained optimiser (L-BFGS-B) to keep all the log-parameters in a box of say, +/- 5, and the second is to first fit with a single component, and use those estimates, along with added random ones, as a starting point for a round of Nelder-Mead or BFGS (though, considering that the problem seems to be that the algorithms are overfitting one component in neglect of the other, this probably won't help, and method 1 might be the only course of action.).

Looking at the integrated squared error, summarized in Table 3.1, we see that it decreases the more observations we have in our sample, which seems logical, but oddly enough, using more parameters does not necessarily reduce the error.

3.2.2 Data simulated from other distributions

We will try to fit tilted mixtures to 2 other distributions. The first is simply the uniform distribution on $[0, 1]$ and the second is to a logit-normal distribution.

uniform distribution

In the case were the underlying distribution is a $U(0, 1)$, we expect to see a good fit with $K = 1$, as the uniform distribution is identical to a $Beta(1, 1)$ distribution. In Figure 3.5 we have ploted fits with 1 and 2 components. Unsurprisingly, there is basically no difference between using 1 or 2 components, which makes sense, as the data war generated from a beta with 1 component. Unsurprisingly either, the more observations in each sample, the less volatility there is in the fits.

What is more interesting is the general shape of the fits. We know that the Beta distribution will be a horizontal line if and only if both the α and β parameters are equal to 1. Even if they are a little off, the ends of the distribution will curl up to infinity or down to 0 (in fact, this probably only happens when a parameter get closer to 1 than the machine precision of the computer, and it rounds it to 1), and because it is very unlikely to get exactly 1, we see the ends going vertical on each en of the graphs, regardless of the number of observations.

We did not include the boxplots for these cases, as they show similar behaviours as the ones fitting actual tilted beta mixtures.

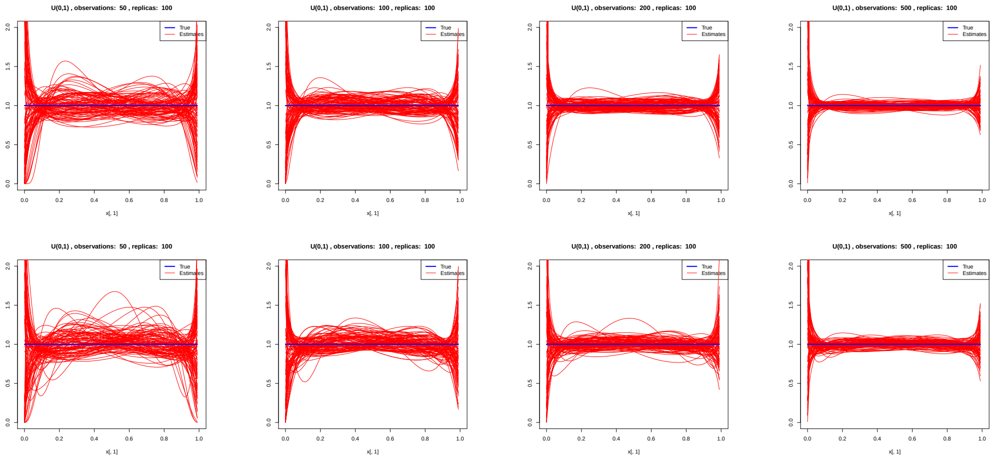


Figure 3.5: MLE fits to samples of size, from left to right, $n = 50$, $n = 100$, $n = 200$ and $n = 500$, with different number of betas in the mixture. From top to bottom: $K = 1$ and $K = 2$, using a *Nelder-Mead* optimiser each time, with a maximum of 500 iterations. The underlying distribution is a uniform distribution, which is also a $Beta(1, 1)$ distribution.

logit-normal distribution

Next we tried using a logit-normal distribution with parameters $\mu = -1.5$ and $\sigma = 1.5$ to generate data to fit. I thought this tilt would be fit better as the ends drop off to 0, similarly to a beta distribution with parameters approaching 1 from the top, but with a small dip in the center which I thought would force at least a second component to be fit as well. But looking at Figure 3.6 it ones again looks like there is not much difference between using one or more components in the mixture. Much as in the uniform case, unsurprisingly the fits get less variable the more observations there are in the samples. I calculated the mean AIC for the different cases, and the results are summarized in Table 3.2. The results were not as I had expected before doing the simulations, as I mentioned before, I expected there to be significantly smaller AIC with more than one component, however the results do match what I expected after looking at Figure 3.6. The AIC is always lowest for the case $K = 1$, and it increases roughly by 6 every time a component is added, which corresponds to the minimum log likelihood not changing but the penalty for the number of parameters increase by $2(K_{i+1} - K_i) = 6$. Unsurprisingly the AIC is better the more observations there are the in sample, as there is more information available.

In call cases, for this original distribution, we lose some of the observations after tilting, because of the acceptance rejection step. In fact, only about 60% percent of the data remains, which would also contribute to lowering the quality of the fits. As for the uniform case, we will omit the boxplots of the parameters estimates, as they are much the same as the in case with beta mixtures.

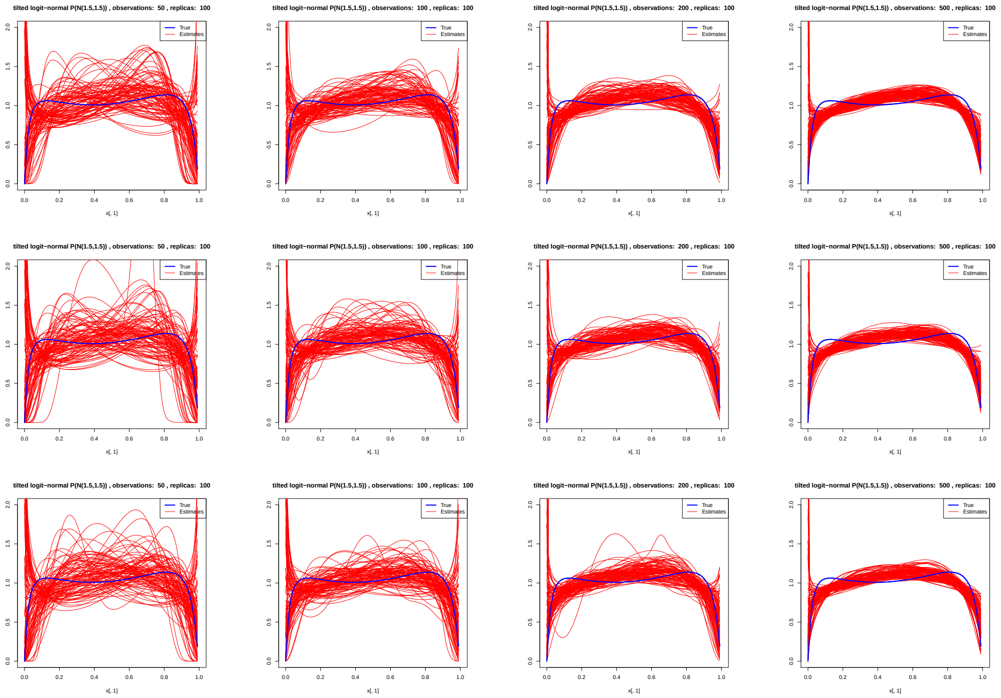


Figure 3.6: MLE fits to samples of size, from left to right, $n = 50$, $n = 100$, $n = 200$ and $n = 500$, with different number of betas in the mixture. From top to bottom: $K = 1$, $K = 2$ and $K = 3$, using a *Nelder-Mead* optimiser each time, with a maximum of 500 iterations. The underlying distribution is a tilted logit-normal distribution.

One thing that I noticed at the end of doing this part, is that many of the fits reached the maximum iteration of `Nelder-Mead`, before achieving converging, although this is less the case with observation-high samples. So I will try to up the amount of maximum iterations for `Nelder-Mead` from 500 to 1'000 in the next section and monitor the convergence of the fits.

3.3 Numerical example

3.3.1 The data

The data we will be exploring consists of 2 variables, daily maximum windspeed (mph) and the Fosberg fire weather index FFWI, for 20 locations in Southern California. The FFWI is an index to give some measure of the potential influence of the weather on wildland fires. It is calculated using temperature, relative humidity and windspeed, and calibrated to equal 100 when the windspeed is 30 mph and the air moisture content is 0. Any value of FFWI higher than 100

K	n=50	n=100	n=200	n=500
1	1.44	0.98	-0.48	-4.68
2	7.62	7.06	5.75	1.78
3	13.56	13.56	11.58	1.76

Table 3.2: AIC for fits to a tilted logit-normal model of parameters $\mu = -1.5$ and $\sigma = 1.5$.

is rounded down to 100. The data originated from the Hadley Centre (`https://www.metoffice.gov.uk/hadobs/hadisd/`) and were processed by Professor Ben Shaby to get them into the form they are now. Notably, the data went under some corrections for bad data, windspeed conversion from m/s to mph, calculation of the FFWI and they may have been homogenised to deal with instrument drift and other issues. The windspeed values may therefore be seem a bit odd, and to get windspeeds in m/s again, the windspeed would need to be divided by 2.23694. The processed data is available on Github (`https://github.com/Sekarski/MasterSemestreProject/data/had_ffwi_wind-SantaAna.RData`).

For this project we have chosen the 3 locations out of the 20 that have the most data point to study. These are locations 1,17 and 15, with respectively 15'400, 15'399 and 15'372 points, after removing missing or incomplete data points.

3.3.2 Making the data angular

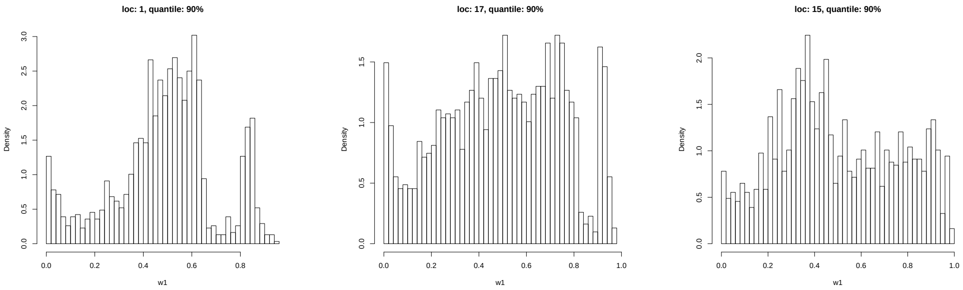


Figure 3.7: Histograms for the angular data above the 90% quantile for locations 1,17 and 15 respectively.

In order to fit our tilted mixtures, we first need to make the data angular. Let $X = (X_1, X_2)$ be the bivariate data at one of the locations, and $X_i = (x_{i1}, \ldots, x_{in})$ for $i = 1, 2$. We start by fitting a Generalized Pareto Distribution to the exceedances over thresholds u_i on each margins. For the thresholds, we chose the 90% quantiles for each margin. We get estimates $\hat{\sigma}_i$ and $\hat{\xi}_i$ and fitted distributions

$$\hat{F}_i(x) = n^{-1}\sum_{j=1}^{n} \mathrm{I}_{\{x_{ij}\leq x\}}(x)\mathrm{I}_{\{x\leq u_i\}}(x) + \left[1 - \frac{n_{u_i}}{n} \times \left\{1 + \hat{\xi}_i\frac{(x-u_i)}{\hat{\sigma}_i}\right\}_+^{-1/\hat{\xi}_i}\right]\mathrm{I}_{\{x>u_i\}}(x),$$

Were n_{u_i} is the number of observations above the threshold u_i. We then apply the transformation $Z_i = -1/\log \hat{F}_i(X_i)$ to bring X to the unit Fréchet scale, then tilt the data by setting

$$W_i = Z_i/(Z_1 + Z_2), \quad i = 1, 2.$$

Finally, we get the data on which we will do the fitting by selecting all the observation where $z_1j + z_2j$ are above the 90% quantile of $Z_1 + Z_2$. That is:

$$W = \{(w_{1j}, w_{2j}) : z_{1j} + z_{2j} > r_{90}\}$$

Where r_{90} is the 90% quantile. Figure 3.7 show what the distribution of W_1 looks like. All three locations have different shapes.

3.3.3 Fitting

We can see graphically the result of fitting tilted mixtures to locations 1,17 and 15 using 1,2, and 3 components in Figure 3.8. Visually there seems to be a big improvement between $K = 1$ and $K = 2$ but then hardly any improvement between 2 and 3 component fits, suggesting that there is no extra gain in using 3 component versus using only 2. Quantitatively, looking at Table 3.3, we see that a forward step-AIC method would also chose the models with $K = 2$. So it would seem that using 2 components is the way to go here. However there is an issue: the `Nelder-Mead` reached the maximum number of iterations, which is set at 1'000, before converging already in the cases $K = 2$, which means that the fact that the 3 component cases have higher AIC may just be that we didn't give it enough time, and as it has to fit more 3 extra parameters it is penalized more while not getting to it's peak Likelihood. We could just up the maximum number of iterations, but instead we are going to keep the maximum number of iterations the same, but switch from the `Nelder-Mead` optimiser to the `BFGS` optimizer, which is a little more aggressive in it's exploration of the parameter space. This means that it will take bigger steps towards convergence, but might be more unstable.

In Table 3.4 we can see the AIC value for mixtures with more components, using `BFGS` with a maximum of 1'000 iterations. For location 1, we do indeed see that using more components yields quite a bit better AIC value compared to $K = 2$ from Table 3.3 and a forward step-AIC method would chose $K = 6$. For locations 17 and 15 however, the AIC is smaller for $K = 3$, even using `BFGS` compared to $K = 2$ using `Nelder-Mead` and forward step-AIC methods would still keep $K = 2$. We can see that if we skip a few Ks the AIC start decreasing again, even massively between $K = 5$ and $K = 6$ for location 17. Looking at the graphs of the fit, in Figure 3.9, this is probably due to overfitting. As for

Loc	K	AIC	$\hat{\pi}_1$	$\hat{\alpha}_1$	$\hat{\beta}_1$	$\hat{\pi}_2$	$\hat{\alpha}_2$	$\hat{\beta}_2$	$\hat{\pi}_3$	$\hat{\alpha}_3$	$\hat{\beta}_3$
1	1	-458	1	1.15	5.83						
1	2	-711	0.94	15.42	2.4	0.06	0.91	0.95			
1	3	-705	0.94	13.67	2.41	0.03	0.89	3.5	0.03	2.55	2.53
17	1	-147	1	0.88	2.41						
17	2	-194	0.94	2.17	1.68	0.06	0.67	14.01			
17	3	-185	0.17	0.57	0.99	0.71	3.06	1.68	0.12	4.12	1.55
15	1	-79	1	1.4	1.29						
15	2	-116	0.067	1.26	0.67	0.93	8.22	0.97			
15	3	-110	0.91	5.17	1.01	0.07	2.13	0.95	0.02	1.48	7.63

Table 3.3: MLE estimates for locations 1,17 and 15, using K=1,2 and 3 components, with `Nelder-Mead` using maximum 1000 iterations.

loc	K=3	K=4	K=5	K=6	K=7
1	-1029	-1070	-1073	-1093	-1081
17	-188	-186	-198	-350	
15	-112	-106	-100	-94	

Table 3.4: AIC for fits using `BFGS` for various number of components

location 15, we see that the AIC increases by 6 for every added component, which means that the minimum negative log likelihood is not changing at all, while the penalty for parameter increases ($\text{AIC} = 2(3K-1) - 2\log(L)$)

For the rest of the analysis, we shall use the fits with 6, 2 and 2 components respectively for locations 1, 17 and 15.

Goodness of fit

To check the goodness of fit, we will do several statistical test. The first test we will do, it a Kolmogorov-Smirnov test to see if it is plausible that the data came from the tilted mixtures we selected above. The KS test will test the null hypothesis that this is the case, versus the alternative hypothesis that it does not come from the selected tilted mixture.

For location 17, the model is a tilt of $0.94Beta(2.08, 1.71)+0.06Beta(0.67, 16.33)$. The KS test gives us a p-value of 0.056. This is low, but depending on what level we chose this could be significant or not.

For location 15, the model is a tilt of $0.95Beta(7.77, 0.98)+0.05Beta(1.25, 1.04)$. The KS test gives a p-value of 0.059. This is low, but depending on what level we chose this could be significant or not.

For location 1. There is a problem with this model, in that it has means $m_1 \approx 0$ and $m_2 \approx 1$ such that it is impossible to sample from this distribution in `R` using the method in Section 2.2 because the value $m_{\min}/m^T w \approx 0$ and the acceptance rate is for all intent and purposes, zero. A KS test using the `ks.test()` command cannot be done. Switching back to the model were $K = 3$, which has the most significance increase in its AIC when adding a component, we get a p-value of 0.059 for the KS test. This is low, but depending on what level we chose this could be significant or not. A model with a higher K might be better.

The problem with the KS test in `R` is that it uses samples from the null distribution, which means that the p-value will vary for every sample (and sample sizes), making the value not particularly useful. I have set a seed in `R` for reproducibility. What might be better, is to do the test over and over so many times, than take the mean p-value. It may just be here that I took too many sample of the null distribution ($\approx 100\times$ as many as in the fitted data)

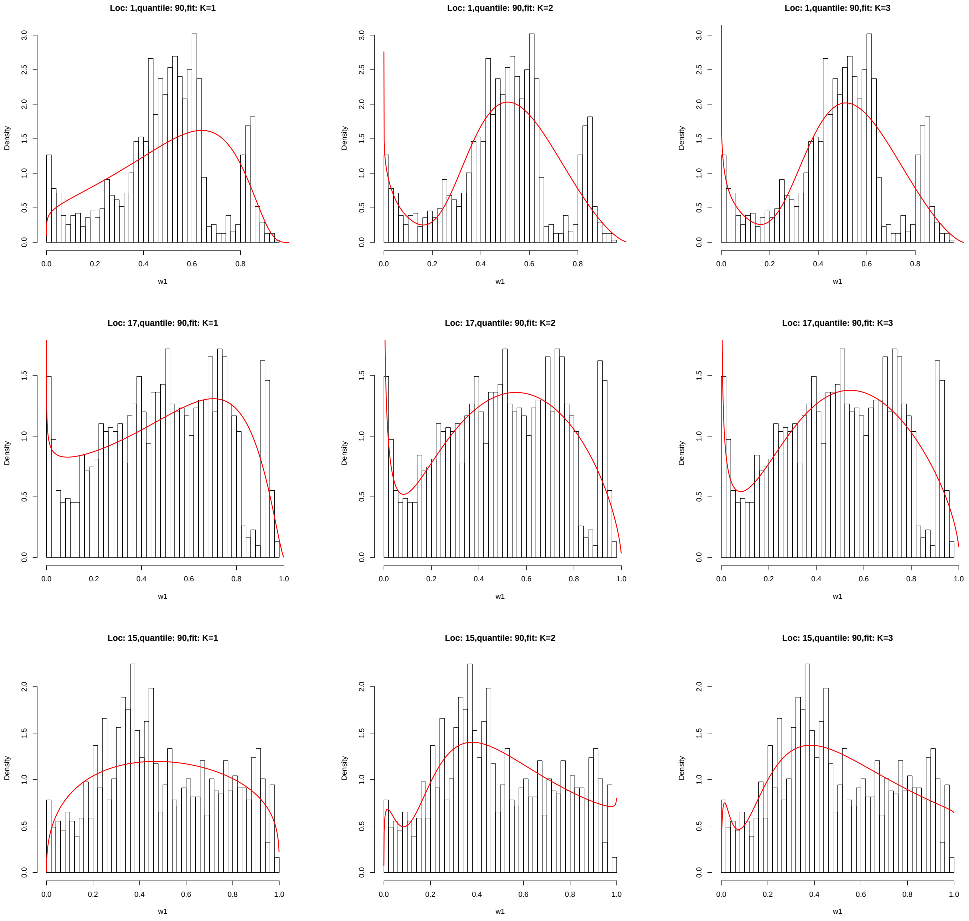


Figure 3.8: Fits to locations 1, 17 and 15, using from left to right $K = 1, 2, 3$. Using the `Nelder-Mead` optimizer, with a maximum of 1'000 iterations.

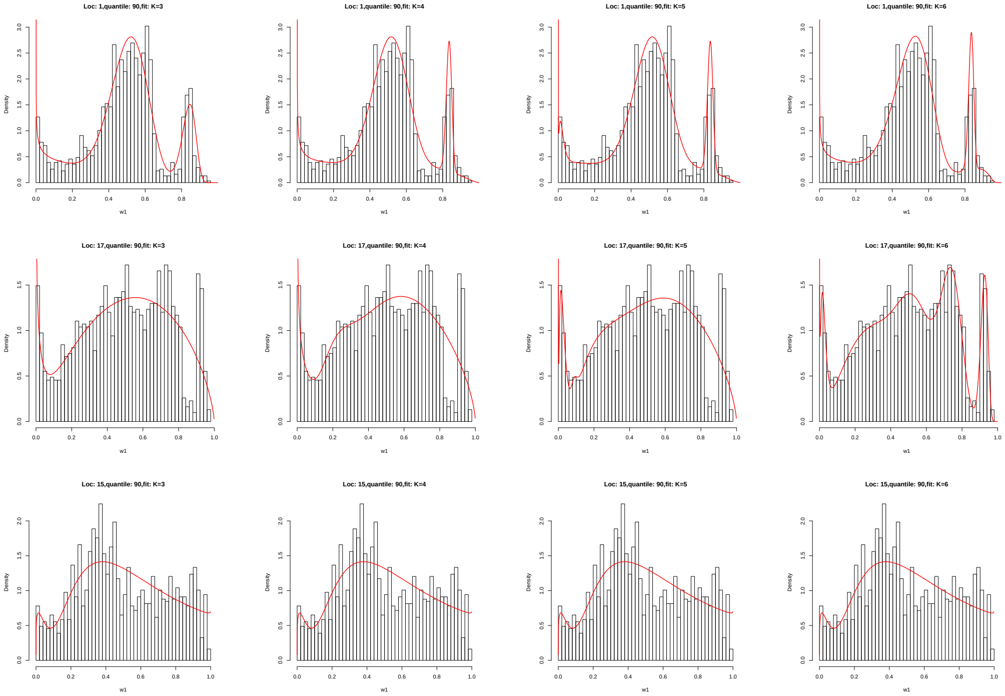


Figure 3.9: Fits to locations 1, 17 and 15, using from left to right $K = 3, 4, 5, 6$. Using the `BGFS` optimizer.

Chapter 4

Conclusions

This fits to the real data are not bad, but could be a lot better. There are a few problems that should be looked at more in depth, such as using a constrained optimizer to solve the maximum likelihood estimation, to prevent the parameters from exploding. When one of the parameters explodes, but not the other one, which was often the case in our simulations, then the optimizer is fitting a distribution that tends to resemble a Dirac delta ”distribution” in either 0 or 1, which can cause unforeseeable behaviour in the optimization algorithm as the computer is ill equipped to deal with infinite-like values. If both parameters explode, the fitted distribution tends to resemble a Dirac delta ”distribution” in 0.5, and the same sort of problems could occur. The question is, how to chose the appropriate bounds, that stop this behaviour, while still allowing the correct parameters to be found, if by chance they are very large. Another question, is if the optimizers are exploring the whole parameter space, or if they are getting stuck in local minimas, which might be the case when they fit only 1 component very well, then using more might actually yield a better result. This least also to another thing work asking, which is how to chose the initial value for the optimization algorithms. Currently we are just using 10 tuples of uniformly generated initial values within a bounded hypercube, and using the one that yield the best log likelihood. But there might be better ways to go about this.

We see also, that more of the time, a good enough fit is obtained by using just 1 or 2 components, and this is confirmed visually as well as by using a forward step-AIC method for selecting the number of parameters. There was just the case for location 1, where maybe a fit with $K = 6$ would be the optimal, but numerically there were some problems with this model, so that I couldn’t compute a KS test on it. But the model with $K = 3$ is pretty good too.

Bibliography

[1] S. Coles and J. A. Tawn, *Modelling Extreme Multivariate Events*, Journal of the Royal Statistical Society, 1991, pp. 381-382

[2] S. Coles, *An Introduction to Statistical Modelling of Extreme Values*, Springer Series in Statistics, 2001, p. 144

[3] M.-O. Boldi and A. C. Davison, *A mixture model for multivariate extremes*, Journal of the Royal Statistical Society Series B, 2007, pp.217-218

Code

All the `R` code, graphs, literature, used in this project and the LaTeX for this very document, can be found at `https://github.com/Sekarski/MasterSemestreProject`.