May 2016

Distance Density Analysis and Multivariate Mode Detection

Immanuel Torben Lampe
University of Wisconsin-Milwaukee

DISTANCE DENSITY ANALYSIS AND MULTIVARIATE MODE DETECTION

by

Immanuel T. Lampe

A Thesis Submitted in

Partial Fulfillment of the

Requirements for the Degree of

MASTER OF SCIENCE

in Mathematics

at

The University of Wisconsin-Milwaukee

May 2016

ABSTRACT

DISTANCE DENSITY ANALYSIS AND MULTIVARIATE MODE DETECTION

by

Immanuel T. Lampe

The University of Wisconsin-Milwaukee, 2016
Under the supervision of Professor Daniel Gervini

Finding the mode of the distribution for a sample of points is a very interesting task. In one dimensional problems this can easily be done by estimating the kernel density. Unfortunately this method does not work well in higher dimensions. This thesis presents a new approach to solve this problem.

A method is presented which finds the mode by analyzing the distribution of the distances between each point and the rest of the sample. The idea is that if the i-th sample point, x_i, is in a high-density region, most of these distances should be small, whereas if x_i is an outlier, most of these distances should be large. By running simulations for different distributions this thesis shows that the new method works better than the existing ones in higher dimensions.

TABLE OF CONTENTS

LIST OF FIGURES

1 Introduction

Given a sample of points in $\mathbb{R}^p$ with $p \geq 1$, one important problem is to find the mode of the distribution, or more generally, regions of the space where points tend to accumulate (if there is more than one mode). In one-dimensional problems one can simply estimate the density function nonparametrically (using for example kernel smoothers), but in high dimensions kernel smoothers tend not to work well. Another problem when $p > 3$ is that the data cannot be visualized that easily, therefore it is hard to get a first intuition about the location of the mode.

In this thesis we propose the following way to detect accumulation points in $\mathbb{R}^p$: Given a sample $x_1, ..., x_n$, for each point x_i, consider the distances between that point and the other points in the sample, $\{||x_i - x_j|| : i \neq j\}$. The idea is that if x_i is in a high-density region, most of these distances should be small, whereas if x_i is an outlier, most of these distances should be large. For all other points the distances should be somewhere in between. So a kernel-density estimator $\hat{f}_i$ of these distances is computable for each x_i. If x_i is in a high-density region one would expect the mode of $\hat{f}_i(t), t_i$, to be small, but the peak $\hat{f}_i(t_i)$ to be large, since most distance values would be concentrated around t_i.

The goal of this thesis is to explore, by simulation, how well this works as a mode-detection method. Concretely, for a given sample $x_1, ..., x_n$ with density $f_0(x)$, we estimate the mode $\theta = argmax f_0(x)$ as the point x_i with smallest t_i. The existing method consists of first obtaining a multivariate kernel density estimator of $f_0(x)$, $\hat{f}_0(x)$, and then estimating θ as the point x_i with largest $\hat{f}_0(x_i)$.

In this thesis we run simulations with a number of distributions, normal and non-normal (skewed, like Gammas, where the mode is not the mean), and for different dimensions $p \geq 1$ to see which of the two estimators is closer to the true mode in each case.

The thesis begins with the introduction of the densities used, then it is explained how the kernel density works and how the optimal bandwidth is chosen. Next the new method is introduced and the simulation results are presented. Finally, concluding remarks about the two mode detection methods are made.

2 Important Density functions

As discussed in the introduction, the two different mode finding methods implemented in this thesis will be tried out for different density functions. In the following the notation $x = (x_1, ..., x_d)^T$ is used.

The first important density is the d-variate standard normal one:

$$f(x) = (2\pi)^{-d/2} \exp\left(-\frac{1}{2}x^T x\right)$$

The normal distribution is radially syymetric around 0. Setting a counterpoint to this, we also consider a fairly skewed distribution: The independent d-variate gamma distribution with parameters $\alpha = 3$ and $\beta = 1$. The density is given by

$$f(x) = \left(\frac{1}{2}\right)^d \prod_{i=1}^{d} x_i^2 \exp\left(-x_i\right)$$

The following figure displays these density functions for the case $d = 2$.

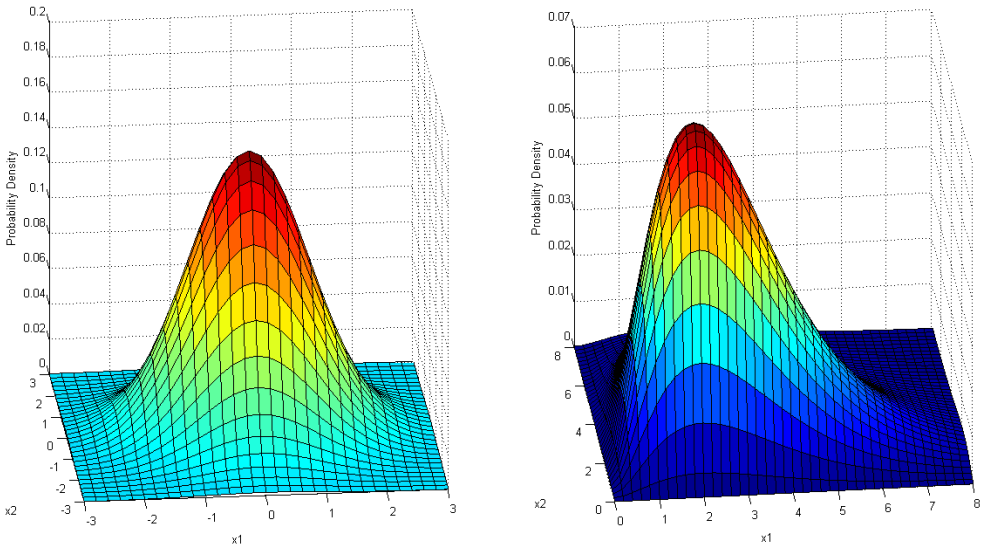


Figure 1: Standard normal and gamma distribution

Finally, in order to verify how the two different methods work for a d-dimensional distribution where the marginal densities are not equal, the d-variate normal distribution with idenpendent but not identically distributed marginals is used:

$$f(x) = (2\pi)^{-d/2} \left|\Sigma_d\right|^{-1/2} \exp\left(-\frac{1}{2} x^T \Sigma^{-1} x\right),$$

where

$$\Sigma_d = \begin{bmatrix} 1 + 0 \cdot \frac{5-1}{d-1} = 1 & 0 & \dots & 0 \\ 0 & 1 + 1 \cdot \frac{5-1}{d-1} & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \dots & 0 & 1 + (d-1) \cdot \frac{5-1}{d-1} = 5 \end{bmatrix}$$

Using this Σ_d for any fixed dimension d keeps the range of the variances equal to 4 while the step size($\frac{5-1}{d-1}$) is adjusted according to the given dimension.

3 The Kernel density estimator

This chapter briefly summarizes how a kernel density estimator is defined. Furthermore the importance of the bandwidth selection will be discussed. Throughout this chapter we will let $X_1, ..., X_n$ denote a $d-$variate random sample having (unknown) density f.

We will use the notation $X_i = (X_{i1}, ..., X_{id})^T$ to denote the components of X_i and a generic vector $x \in \mathbb{R}^d$ will have the representation $x = (x_1, ..., x_d)^T$. Finally the $d \times d$ identity matrix will be denoted by I_d.

3.1 The Gaussian kernel

Kernel density estimator is a nonparametric approach to estimating the probability density of a random variable from a given sample $X_1, ..., X_n$. The most general from is

$$\hat{f}(x; H) = n^{-1} \sum_{i=1}^{n} K_H(x - X_i),$$

with

$$K_H(x) = |H|^{-1/2} K(H^{-1/2}x).$$

In this case H is a symmetric positive definite $d \times d$ matrix. Usually H is referred to as the Bandwidth matrix and K_H as the kernel of the estimator. There are several kernel choices. One of the most common and the one I decided to use in my master thesis is the standard $d-$variate normal density

$$K(x) = (2\pi)^{-d/2} \exp\left(-\frac{1}{2}x^T x\right).$$

It is often referred to as the Gaussian Kernel. In this thesis $H \in \mathbb{D}$ is always assumed, where $\mathbb{D}$ is the subclass of diagoal positive definite $d \times d$ matrices. Then

$$\hat{f}(x; H) = n^{-1} \left(\prod_{l=1}^{d} h_l\right)^{-1} \sum_{i=1}^{n} K_H(x - X_i). \tag{1}$$

3.2 Bandwidth selection

Formula (1) shows that the choice of the bandwidth has an influence on the kernel density estimator $\hat{f}$. To underline the importance of the bandwidth selection, a sample of size 100 from a two-dimensional standard normal variable was created. Then two kernel density estimates with

$$H_1 = \begin{bmatrix} 0.4135^2 & 0 \\ 0 & 0.4135^2 \end{bmatrix}, H_2 = \begin{bmatrix} 0.2^2 & 0 \\ 0 & 0.2^2 \end{bmatrix}$$

were computed. The following figure compares the true density function with the resulting kernel densities if H_1 and H_2 are used.

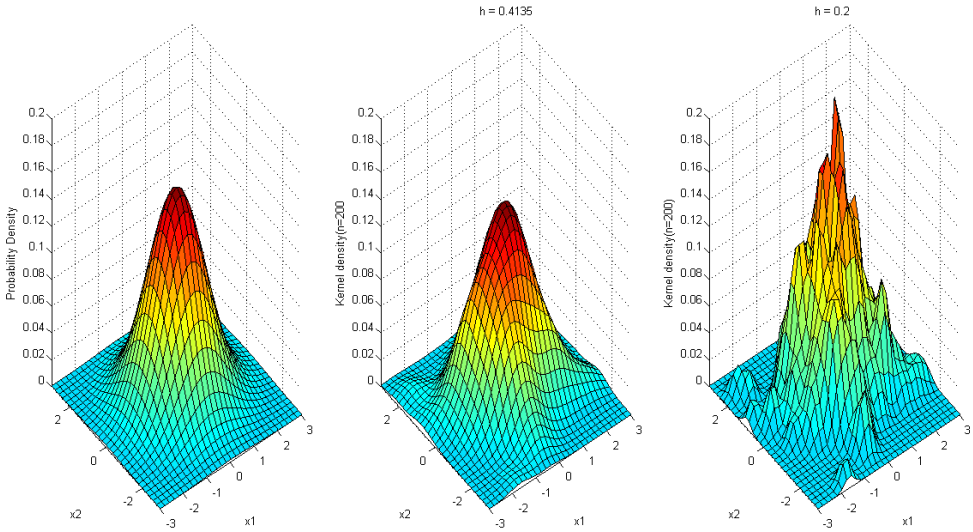


Figure 2: Importance of bandwidth choice

Obviously bandwidth selection influences the quality of the calculated kernel densities. It can be stated that the choice H_1 produces an estimator closer to the true density function.

The question motivated by this example iss: what is the optimal bandwidth choice? Before we are able to answer this, we have to ask ourselves what optimal bandwidth means.

The main goal of the kernel density estimator $\hat{f}$ should be to minimize the distance to the true but unknown density f. Usually (see Wand and Jones [1995, p.19]) the mean integrated square error (MISE) is chosen as a measure of closeness between the kernel density and the true density:

$$MISE(f(x;h)) = \int E(f(x;h) - f(x))^2 dx.$$

For simplicity the Taylor's formula is used to approximate the MISE. This is called the asymptotic mean integrated square error (AMISE). If the same bandwidth is used in every dimension, which means

$$H = \begin{bmatrix} h^2 & 0 & \dots & 0 \\ 0 & \ddots & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \dots & 0 & h^2 \end{bmatrix},$$

Wand and Jones [1995, p.99] showed that the bandwidth which minimizes the AMISE is given by

$$h_{AMISE} = \left(\frac{dR(K)}{\mu_2(K)^2 n \int \{\nabla^2 f(x)\}^2 dx} \right)^{1/(d+4)}, \tag{2}$$

where

$$R(K) = \int K(x)^2 dx,$$

$$\mu_2(K) = \int x_i^2 K(x) dx$$

and

$$\nabla^2 f(x) = \sum_{i=1}^{d} \frac{\mathrm{d}^2}{\mathrm{d}{x_i}^2} f(x).$$

As explained above, we decided to use the multivariate Gaussian-Kernel, for which it is easy to verify that

$$\mu_2(K) = 1.$$

The calculation for $R(K)$ is a little bit more complex:

$$\begin{aligned} R(K) &= \int K(x)^2 dx = \int_{\mathbb{R}^d} (2\pi)^{-d} \exp(-x^T x) dx \\ &= \int_{\mathbb{R}^d} (2\pi)^{-d} \exp\left(-x_1^2 - ... - x_d^2\right) dx \\ &= (2\pi)^{-d} \left(\int_{\mathbb{R}} \exp\left(-x_1^2\right) dx_1 \right)^d = (2\pi)^{-d} \left(\sqrt{\pi}\right)^d \\ &= (4\pi)^{(-1/2)d}. \end{aligned}$$

Then

$$h_{AMISE} = \left(\frac{d(4\pi)^{-d/2}}{n \int \{\nabla^2 f(x)\}^2 dx} \right)^{1/(d+4)}. \tag{3}$$

This expression depends on the underlying density f, so the next task is to compute $\int \{\nabla^2 f(x)\}^2 dx$ for the different multivariate density functions used in this thesis:

a) d-dimensional standard normal distribution

and

b) d-dimensional gamma distribution with parameters $\alpha = 3$ and $\beta = 1$.

We will start off with the calculation for case (a).

To recall the density is given by:

$$f(x) = (2\pi)^{-d/2} \exp\left(-\frac{1}{2}x^T x\right), x \in \mathbb{R}^d$$

Therefore

$$\frac{\mathrm{d}^2}{\mathrm{d}{x_i}^2} f(x) = \left(\frac{1}{\sqrt{2\pi}}\right)^d \exp\left(\frac{-x_i^2}{2}\right)(x_i^2 - 1) \prod_{\substack{j=1 \\ j \neq i}}^{d} \exp\left(\frac{-x_j^2}{2}\right).$$

Furthermore

$$\left[\sum_{i=1}^{d} \frac{\mathrm{d}^2}{\mathrm{d}{x_i}^2} f(x)\right]^2 = \sum_{i=1}^{d} \left(\frac{\mathrm{d}^2}{\mathrm{d}{x_i}^2} f(x)\right)^2 + \sum_{i=1}^{d} \sum_{\substack{j=1 \\ j \neq i}}^{d} \left(\frac{\mathrm{d}^2}{\mathrm{d}{x_i}^2} f(x)\right)\left(\frac{\mathrm{d}^2}{\mathrm{d}{x_j}^2} f(x)\right).$$

For a fixed $i \in \{1, ..., d\}$,

$$\begin{aligned}\int_{\mathbb{R}^d} \left(\frac{\mathrm{d}^2}{\mathrm{d}{x_i}^2} f(x)\right)^2 dx &= \left(\frac{1}{\sqrt{2\pi}}\right)^{2d} \int_{\mathbb{R}} \exp\left(-x_i^2\right)\left(x_i^2 - 1\right)^2 dx_i \left[\int_{\mathbb{R}} \exp\left(-x_j^2\right) dx_j\right]^{d-1} \\ &= \left(\frac{1}{\sqrt{2\pi}}\right)^{2d} \left(\frac{3\sqrt{\pi}}{4}\left(\sqrt{\pi}\right)^{d-1}\right).\end{aligned}$$

For fixed $j, i \in \{1, \ldots, d\}$ with $j \neq i$,

$$\begin{aligned}
&\int_{\mathbb{R}^d} \left(\frac{\mathrm{d}^2}{\mathrm{d}{x_i}^2} f(x)\right) \left(\frac{\mathrm{d}^2}{\mathrm{d}{x_j}^2} f(x)\right) dx \\
&= \left(\frac{1}{\sqrt{2\pi}}\right)^{2d} \left[\int_{\mathbb{R}} \exp\left(\frac{-x_i^2}{2}\right) (x_i^2 - 1) \exp\left(\frac{-x_i^2}{2}\right) dx_i\right]^2 \left[\int_{\mathbb{R}} \exp\left(-x_j^2\right) dx_j\right]^{d-2} \\
&= \left(\frac{1}{\sqrt{2\pi}}\right)^{2d} \left(\frac{\sqrt{\pi}}{2}\right)^2 \left(\sqrt{\pi}\right)^{(d-2)}.
\end{aligned}$$

Both integrals are independent of i and j. Hence

$$\begin{aligned}
&\int_{\mathbb{R}^d} \left[\sum_{i=1}^{d} \frac{\mathrm{d}^2}{\mathrm{d}{x_i}^2} f\right]^2 dx = d \int_{\mathbb{R}^d} \left(\frac{\mathrm{d}^2}{\mathrm{d}{x_i}^2} f\right)^2 dx \\
&+ d(d-1) \int_{\mathbb{R}^d} \left(\frac{\mathrm{d}^2}{\mathrm{d}{x_i}^2} f\right) \left(\frac{\mathrm{d}^2}{\mathrm{d}{x_j}^2} f\right) dx \\
&= \left(\frac{1}{\sqrt{2\pi}}\right)^{2d} \left[d \left(\frac{3\sqrt{\pi}}{4}\right) \left(\sqrt{\pi}\right)^{d-1} + d(d-1)\frac{\pi}{4} \left(\sqrt{\pi}\right)^{(d-2)}\right] \\
&= \left(2\sqrt{\pi}\right)^{-d} (d/2 + d^2/4).
\end{aligned}$$

The optimal bandwidth for the standard normal distribution then becomes

$$\begin{aligned}
h_{AMISE} &= \left(\frac{d(4\pi)^{-d/2}}{\int \{\nabla^2 f(x)\}^2 dx n}\right)^{1/(d+4)} \\
&= \left(\frac{d(4\pi)^{-d/2}}{\left(2\sqrt{\pi}\right)^{-d} (d/2 + d^2/4) n}\right)^{1/(d+4)} = \left(\frac{4}{(d+2)n}\right)^{1/(d+4)}.
\end{aligned}$$

To conclude, the bandwidth matrix is given by:

$$H_{AMISE} = \{4/(d+2)\}^{2/(d+4)} \cdot I_d \cdot n^{-2/(d+4)} \tag{4}$$

Now scenario (b) will be analyzed. Here, for $x \in (0,\infty)^d$,

$$\begin{aligned} f(x) &= \left(\frac{1}{\beta^\alpha \Gamma(\alpha)}\right)^d \prod_{i=1}^d x_i^{\alpha-1} \exp\left(-x_i/\beta\right) \\ &\underset{\beta=1}{\overset{\alpha=3}{=}} \left(\frac{1}{1^3\Gamma(3)}\right)^d \prod_{i=1}^d x_i^{3-1} \exp\left(-x_i/1\right) \\ &= \left(\frac{1}{2}\right)^d \prod_{i=1}^d x_i^2 \exp\left(-x_i\right). \end{aligned}$$

Accordingly,

$$\frac{\mathrm{d}^2}{\mathrm{d}{x_i}^2} f(x) = \left(\frac{1}{2}\right)^d \exp(-x_i)(x_i^2 - 4x_i + 2) \prod_{\substack{j=1 \\ j\neq i}}^d x_j^2 \exp\left(-x_j\right).$$

With the same argument as before,

$$\begin{aligned} &\int_{(0,\infty)^d} \left[\sum_{i=1}^d \frac{\mathrm{d}^2}{\mathrm{d}{x_i}^2} f(x)\right]^2 dx = d \int_{(0,\infty)^d} \left(\frac{\mathrm{d}^2}{\mathrm{d}{x_i}^2} f(x)\right)^2 dx \\ &+ d(d-1) \int_{(0,\infty)^d} \left(\frac{\mathrm{d}^2}{\mathrm{d}{x_i}^2} f(x)\right) \left(\frac{\mathrm{d}^2}{\mathrm{d}{x_j}^2} f(x)\right) dx. \end{aligned}$$

We therefore have to estimate

$$\int_{(0,\infty)^d} \left(\frac{\mathrm{d}^2}{\mathrm{d}{x_i}^2} f(x) \right)^2 dx := \circledast$$

and

$$\int_{(0,\infty)^d} \left(\frac{\mathrm{d}^2}{\mathrm{d}{x_i}^2} f(x) \right) \left(\frac{\mathrm{d}^2}{\mathrm{d}{x_j}^2} f(x) \right) dx := \circledast \circledast \, .$$

Applying basic integration laws gives

$$\begin{aligned} \circledast &= \left(\frac{1}{2} \right)^{2d} \int_0^\infty \exp(-2x_i)(x_i^2 - 4x_i + 2)^2 dx_i \left(\int_0^\infty x_j^4 \exp\left(-2x_j\right) dx_j \right)^{d-1} \\ &= \left(\frac{1}{2} \right)^{2d} \frac{3}{4} \left(\frac{3}{4} \right)^{d-1} \\ &= \left(\frac{1}{2} \right)^{2d} \left(\frac{3}{4} \right)^{d} \end{aligned}$$

and

$$\begin{aligned} \circledast\circledast &= \left(\frac{1}{2} \right)^{2d} \left(\int_0^\infty \exp(-2x_i) \left(x_i^2 - 4x_i + 2 \right) x_i^2 dx_i \right)^2 \left(\int_0^\infty x_j^4 \exp\left(-2x_j\right) dx_j \right)^{d-2} \\ &= \left(\frac{1}{2} \right)^{2d} \left(\frac{-1}{4} \right)^2 \left(\frac{3}{4} \right)^{d-2} . \end{aligned}$$

Then

$$\begin{aligned}\int_0^\infty \left[\sum_{i=1}^{d} \frac{\mathrm{d}^2}{\mathrm{d}{x_i}^2} f(x)\right]^2 dx &= d\left(\frac{1}{2}\right)^{2d}\left(\frac{3}{4}\right)^{d} + d(d-1)\left(\frac{1}{2}\right)^{2d}\left(\frac{-1}{4}\right)^{2}\left(\frac{3}{4}\right)^{d-2} \\ &= \left(\frac{1}{2}\right)^{2d}\left(\frac{3}{4}\right)^{d-2} d\left(\left(\frac{3}{4}\right)^{2} + (d-1)\frac{1}{16}\right) \\ &= \frac{1}{16}\left(\frac{1}{2}\right)^{2d}\left(\frac{3}{4}\right)^{d-2} d\left(9 + (d-1)\right) \\ &= \frac{1}{16}\left(\frac{1}{2}\right)^{2d}\left(\frac{3}{4}\right)^{d-2}\left(8d + d^2\right).\end{aligned}$$

Finally the optimal bandwidth for the kernel density estimator if a gamma distribution with $\alpha = 3$ and $\beta = 1$ is used can be found by combining the result from above and (3).

If the density of a d-dimensional variable is not the same in each dimension, certainly at least the diagonal elements of the bandwidth matrix should differ from one to another.

In a similar way (4) was derived, it can be shown that for the d-variate normal distribution with mean μ and covariance matrix Σ the bandwidth matrix which minimizes the $AMISE$ is

$$H_{AMISE} = \{4/(d+2)\}^{2/(d+4)} \Sigma n^{-2/(d+4)}. \tag{5}$$

It can easily be verified that this formula is the more general case of (4). Furthermore in this thesis the interest is always on d-dimensonal variables whose components are indepedent of each other. Hence

$$\Sigma = \begin{bmatrix} \sigma_1^2 & 0 & \dots & 0 \\ 0 & \ddots & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \dots & 0 & \sigma_d^2 \end{bmatrix}.$$

The resulting H_{AMISE} is a diagonal matrix and (1) can still be applied for the implementation.

4 Implementation

This chapter explains the basic ideas of the programs implemented in this thesis. The Matlab code of these programs can be found in the Appendix. The first two functions use the kernel density of the sample as the mode finding method, while the last one uses the new approach.

For the first function it is assumed that a sample of size n is generated from a d-variate normal distribution with covariance matrix Σ:

Algorithm 1 mode finding using kd for normal distributed sample

1: **procedure kd1**(A, Σ)
2: **Step 1:** Calculate optimal bandwidth according to (5)
3: **Step 2:** Estimate kernel density for sample points using (1)
4: **Step 3:** Find maximum value of kernel density
5: **Step 4:** Find maximizing sample point c
6: **Step 5:** Return c

In step two the kernel density estimator is only computed at the given sample points. The actual maximum could be computed but it would be more time consuming. Also, for comparision with the new method, we want to identify the model sample point, so we do not look at points outside the sample.

For the second function it is assumed that a sample of size n is generated from a d-variate gamma distribution with parameters $\alpha = 3$ and $\beta = 1$. The only part that changes compared to the first function is the selection of the bandwidth:

Algorithm 2 mode finding using kd for gamma distributed sample

1: **procedure kd2**(A)
2: **Step 1:** Calculate optimal bandwidth according to (3)
3: **Step 2:** Estimate kernel density for sample points using (1)
4: **Step 3:** Find maximum value of kernel density
5: **Step 4:** Find maximizing sample point c
6: **Step 5:** Return c

The next function is the one which uses the new approach.

The input of the next function is any sample of size n, $(X_1, ..., X_n)$, which means that it works independently of the underlying density. As pointed out in the introduction, the main idea behind the new approach is to compute a kernel density estimator for each sample point X_i, representing the density function of the distances to the other sample points. Then the maximum of each of those kernel densities is calculated. Finally, the mode is approximated by the sample point which has its kernel density peak furthest to the left. If this criteria does not lead to a unique choice, the sample point with the highest peak out of those candidates is chosen.

The pseudo code which would implement those ideas is given by:

Algorithm 3 mode finding using distance density estimator

procedure dda(x)
    Step 1: for $i = 1 : n$ **do**
        Calculate euclidean distances $||X_i - X_j||^2$ for any $j \neq i$
    Step 2: for $i = 1 : n$ **do**
        find max and min of $||X_i - X_j||^2$
    Step 3: find total $\max(t_+)$ and $\min(t_-)$ of distances
    Step 4: create vector t of 100 evenly spaced points between t_- and t_+
    Step 5: for $i = 1 : n$ **do**
        calculate distances as in step 1
        fit kernel density to those distances and t
    Step 6: Find maximum of each kernel density
    Step 7: Find sample point with maximum furthest to the left
    Step 7: If more than one candidate, choose the one with highest peak
    Step 8: Return sample point

5 Simulation Results

In this chapter we present the results of the different simulations, carried out in Matlab.

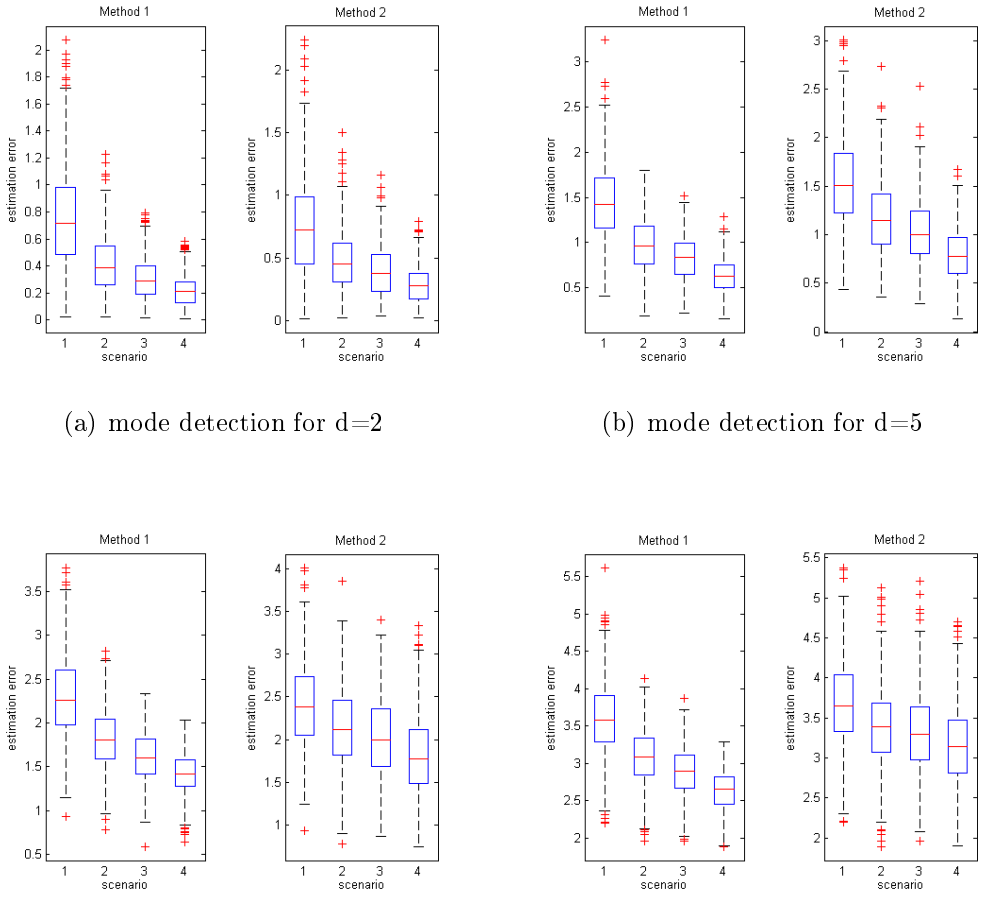
The simulated models were the following: For the three distributions mentioned in the first sections, we use the dimension parameter $d \in \{2, 5, 10, 20, 40, 80\}$. For each of those dimensions, the sample sizes used were $n \in \{10, 50, 100, 300\}$. Finally, for any fixed d and n, 500 simulations were performed.

To estimate the error of the two mode detection methods the Euclidean distance between the estimated and the true mode was used. The results were summarized by boxplots,where the new method was labeled as Method 1 and the old method was labeled as Method 2. Furthermore the different scenarios represent the different sample sizes, where the labels $\{1, 2, 3, 4\}$ correspond to $\{10, 50, 100, 300\}$.

The computation time of the two methodsis also compared at the end.

5.1 Normal distribution

This section summarizes the results for the simulations for the normal distribution is used. First the standard normal distribution will be analyzed. The following figure compares the results for $d = 2$, $d = 5$, $d = 10$ and $d = 20$:



(a) mode detection for d=2

(b) mode detection for d=5

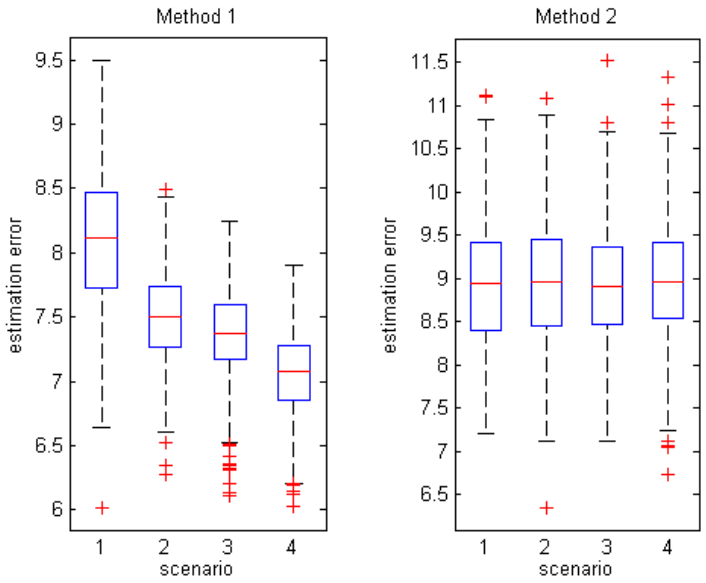
(c) mode detection for d=10

(d) mode detection for d=20

Figure 3: Simulation results standard normal distribution(i)

First of all it can be verified that a higher dimension leads to a higher estimation error for both methods. Furthermore there is only one case in which the old method works better. This is the scnerario where $d = 2$ and $n = 10$. Both methods show an improvement for increasing n. It can be recognized that the new method improves quicker in any dimension. The rate of improvement of the old method is influenced by the dimension. With increasing dimensions ($d = 10$, $d = 20$) the improvement flattens. In contrast to this, the rate of improvement of the new mode estimation method is independent of the dimension.

The results for $d = 40$ are fairly similar to the interpretations already mentioned, therefore it will just be referred to the figure in the Appendix. The following graphic displays the results for $d = 80$:

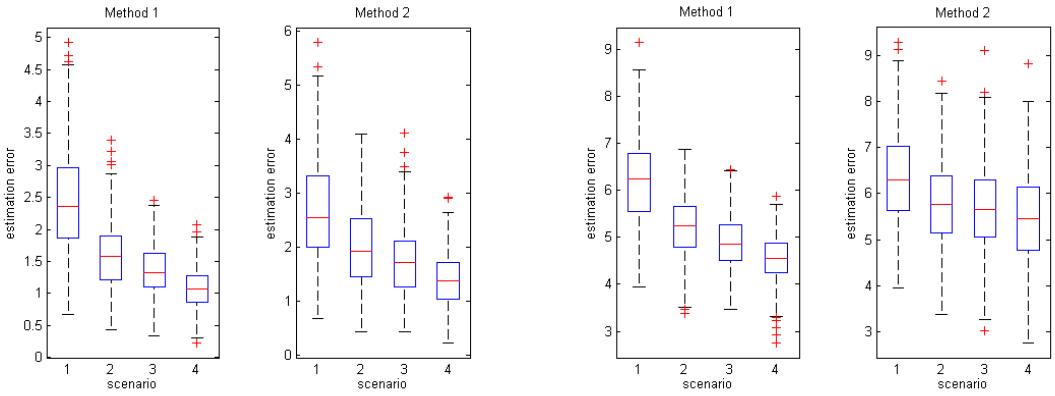


(a) mode detection for d=80

Figure 4: Simulation results standard normal distribution(ii)

This figure is very interesting, because it shows that in a high dimension the old approach does not work well at all anymore. For increasing n there is no improvement in estimation error. In contrast, the new method leads to a smaller mean error and variation of the error if a larger sample size is used.

The next simulations use the normal distribution with different variances in each dimension. The following figure shows the summarizing boxplots for $d = 5$ and $d = 20$:



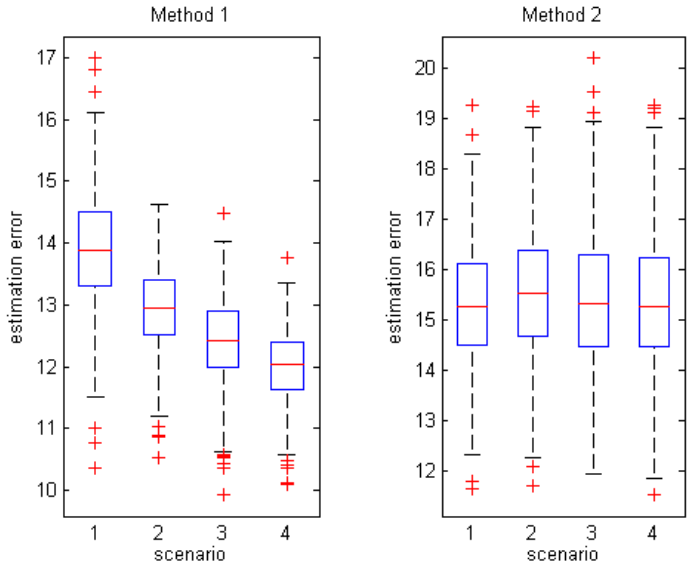
(a) mode detection for d=5 (b) mode detection for d=20

Figure 5: Simulation results normal distribution(i)

We see that the error patterns are very similar to those observed in the standard normal case. Those results holf for the other dimensions (see Appendix).

Appart from that, the main difference is that compared to the standard normal case the error is higher in any dimension, independently of the method chosen. For example, if $d = 5$ and $n = 10$, the mean error for the old method was approximately 1.5 for the standard normal distribution and approximately 2.5 now. For the new method the values are 1.4 and 2.4 respectively.

As before for $d = 80$ the old method does not work well at all:

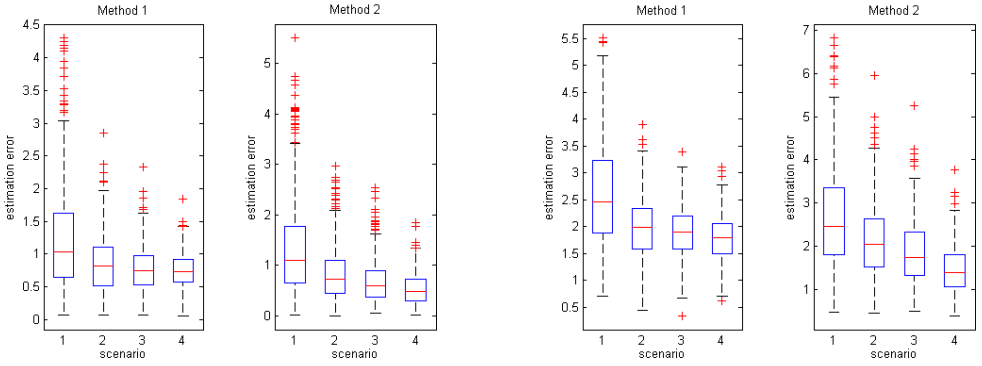


(a) mode detection for d=80

Figure 6: Simulation results normal distribution(ii)

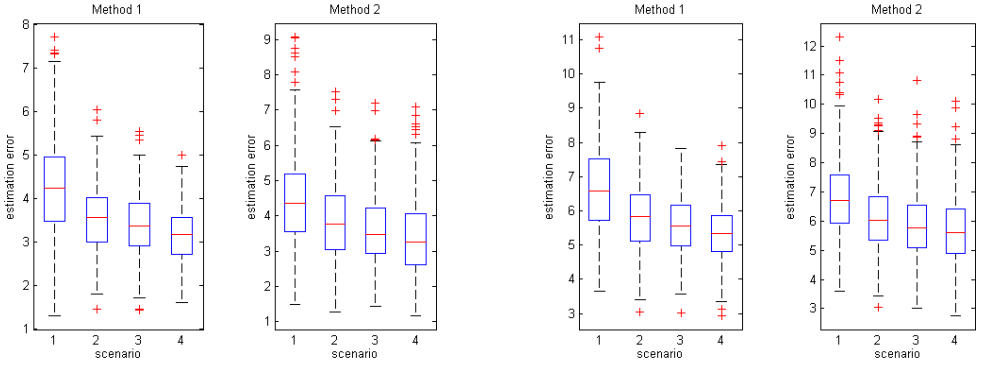
5.2 Gamma distribution

This section summarizes the results of the two mode detection method using a skewed distribution, the gamma distribution with parameters $\alpha = 3$ and $\beta = 1$. The following boxplots show the results for $d \in \{2, 5, 10, 20\}$:



(a) mode detection for d=2

(b) mode detection for d=5

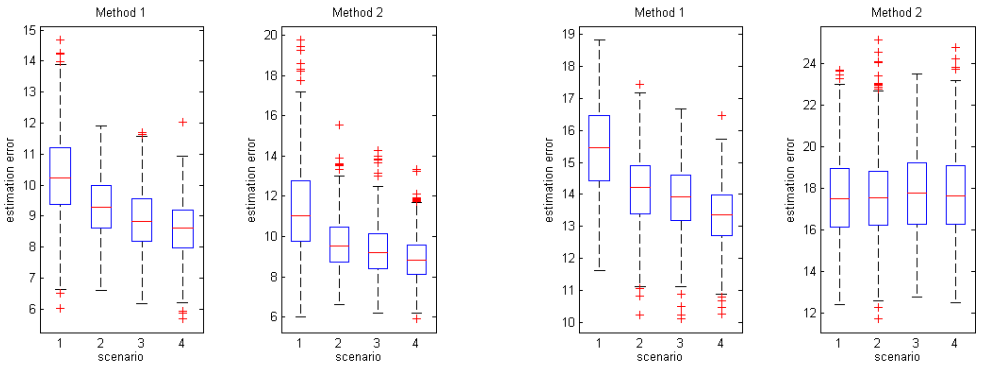


(c) mode detection for d=10

(d) mode detection for d=20

Figure 7: Simulation results gamma distribution(i)

First of all, it can be observed that for the low dimensions ($d \in \{2, 5\}$) the old approach shows better results. Next to that it is very interesting to see that an increasing number of observations still leads to a better performance of both methods, but there is a significant difference in this development. While for the normal distribution the new method showed some large improvement for increasing n, this is now not the case anymore. The improvement is a lot more flat. For example if the case $d = 5$ is analyzed, the mean error decreases from 2.5 ($n = 10$) to 2 ($n = 50$), this is a change of 20%. For the standard normal distribution it decreased from 1.4 to 1. This decrease is equal to 35%. The same observation can be made for any other dimension. As in the normal case the rate of decrease flattens for $d > 5$ if the old method is used. Now the cases $d = 40$ and $d = 80$ are analyzed:



(a) mode detection for d=40 (b) mode detection for d=80

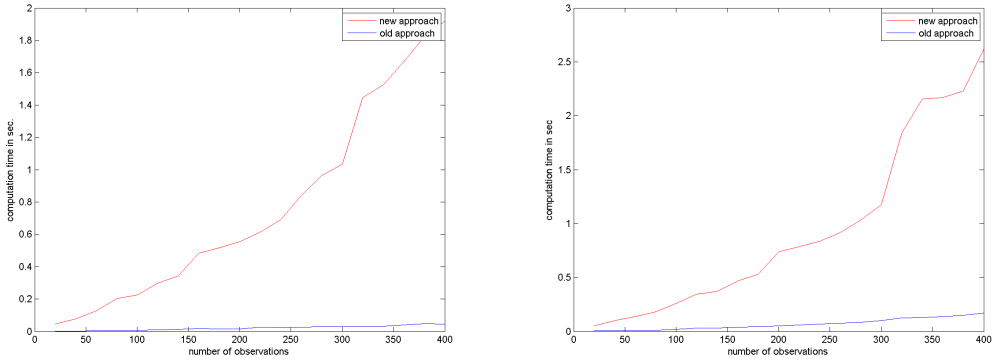
Figure 8: Simulation results gamma distribution(ii)

It can be recognized that for $d = 80$, the old approach does not show any improvement if the sample size is increased, whereas using the new approach there still is an improvement. But as for the lower dimension scenarios, a higher number of observations does not influence the performance of the new method in the same way as for the normal distribution.

To summarize there is definitely a difference in the performance of the two methods. Especially if the sample size and the dimension increase the new approach shows better results for all distributions although less remarkable for the skewed density.

5.3 Computational time

The following plots show the computational times for $d \in \{10, 80\}$.



(a) Computation time for d=10

(b) Computation time for d=80

Figure 9: Computational times

First of all it is observable that both methods work efficiently. Even in the most advanced type of calculation ($d = 80, n = 300$), the computational time is less than 3 seconds for the new approach and less than 0.25 seconds for the old approach. Furthermore it can be seen that the new approach needs more calculation time. The reason is that a kernel density is performed for each sample point separately. This explains why an increasing number of observations has a larger impact on the computational time than the increase of dimension from $d = 10$ to $d = 80$.

6 Summary

In this thesis a new mode finding method was proposed. In contrast to the existing method, the new function does not find the mode by calculating the kernel density of a given sample. Instead, the distances between the sample points are used.

There were two main expectations on this new method: accuracy and time effectiveness. Both of those aspects were analyzed in the previous chapters. Simulation results for the gamma distribution with parameters $\alpha = 3$ and $\beta = 1$ and the normal distribution were presented.

For the standard normal distribution the new method showed a noticeable accuracy increase for larger sample sizes independently of the dimension. For higher dimensions ($d > 5$) this is not true for the old method. The reason for those results is that a kernel density does not approximate a density nicely in high dimensions anymore, therefore the calculated maximum of the kernel density does not have to be close to the true maximum of the density function.

For the normal distribution with different variances in each dimension, the same results are observable. For the normal distribution it can be concluded that the new method definitely outperforms the old one.

Then the gamma distribution was analyzed. The new approach still works better than the old one in dimensions larger than 5. But there is an important difference. While for the normal densities the accuracy of the new function increased quickly with an increasing sample size, now this trend decelerates. One possible reason is that the written program (*dda.m*) uses the Matlab function *ksdensity* to estimate the kernel density of the distances between the sample points. *ksdensity* does not use cross-validation or any other efficient bandwidth selection methods, instead it uses the optimal bandwidth for normal densities, and the distances between sample points are not distributed normally. This problem may be worse for the gamma data than for the normal data.

Resulting from this observation, one improvement for the function *dda.m* would be not to use *ksdensity*. Instead a kernel density approach which calculates the bandwidth without assuming any density should be included in the program.

Next to the accuracy, the simulations done make a statement about the time effectiveness possible. Even though the old approach uses less computation time the new one still works very fast. Even with $d = 80$ and $n = 300$ the computation time is less than three seconds.

Summarizing this thesis, it can be stated that a new mode finding method was successfully created and implemented. Even when compared to the optimal-bandwidth kernel density estimator, the new method was superior. Specially if the number of dimension is high ($d > 5$) the new method outperforms the old one. It should still be recognized that there is room for improvement, as noted before in the density estimation method used by the new approach, especially in the bandwidth choice.

References

[1] Wand, M. P. and Jones, M. C. (1995). *Kernel Smoothing*. Monographs on Statistics and Applied Probability 60, Chapman & Hall/CRC.

Appendices

A Matlab Functions

A.1 kd1.m

```
function [c,f,t] = kd1(A,Sigma)
% Kernel Density Estimator 1
%
% Input
%   A     (n x d)     Data vectors
&   Sigma (d * d)     Covariane Matrix
% Output
%   c     (d x 1)     Estimated mode
%   f     (n x 100)   kernel densities (evaluated on t)
%   t     (1 x 100)   Grid of points where f is evaluated
                     (here points are sample points)

[n,d] = size(A);
% Estimation of optimal bandwidth
D = transpose(sqrt((diag((4/(d+2)) ^(2/(d+4)) * Sigma * n^(-2/(d+4))))));

t = A;
f = zeros(1,n);
for i = 1:n
    C = bsxfun(@minus,A,t(i,:));
    L = bsxfun(@rdivide,C,D);
    f(i) = sum(exp(-1/2*sum((L.^2),2)));
end

f = f./(n*prod(D)) * (1/sqrt(2*pi))^d
[M,I] = max(f);
c = t(I,:)
```

A.2 kd2.m

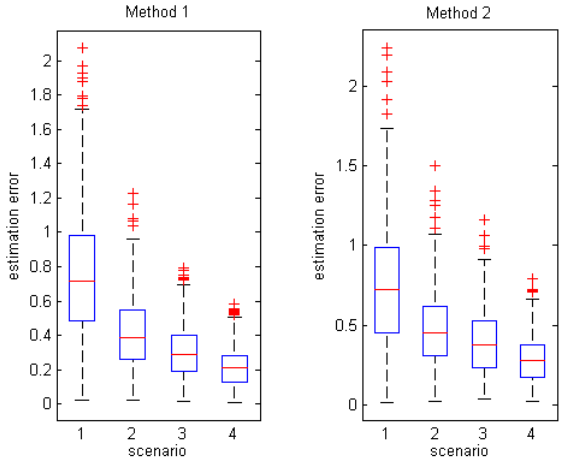
```
function [c,f,t] = kd2(A)
% Kernel Density Estimator 2
%
% Input
%   A     (n x d)     Data vectors
% Output
%   c     (d x 1)     Estimated mode
%   f     (n x 100)   kernel densities (evaluated on t)
%   t     (1 x 100)   Grid of points where f is evaluated
                     (here points are sample points)

[n,d] = size(A);

% Estimation of optimal bandwidth
temp = (1/16) * (1/2)^(2*d) * (3/4)^(d-2) * (8*d+d^2)
h = ((d * (4*pi)^(-d/2))/(n * temp))^(1/(d+4))

t = A;
f = zeros(1,n);
for i = 1:n
    C = bsxfun(@minus,A,t(i,:));
    L = C / h;
    f(i) = sum(exp(-1/2*sum((L.^2),2)));
end

f = f/(n*h^d) * (1/sqrt(2*pi))^d;
[M,I] = max(f);
c = t(I,:)
```

A.3 dda.m

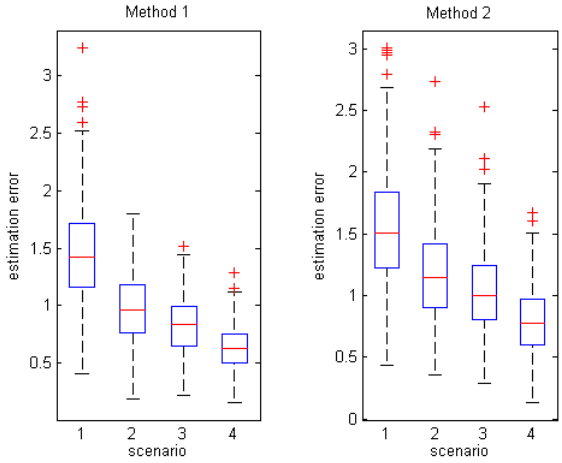
```
function [c,f,t] = dda(x)
% Distance Density Analysis
%
% Input
%   x   (n x d)     Data vectors
% Output
%   c   (d x 1)     Estimated mode
%   f   (n x 100)   Distance densities (evaluated on t)
%   t   (1 x 100)   Grid of equispace points where f is evaluated
                    (endpoints are min and max of sample interdistances)

n = size(x,1);
mini = zeros(n,1);
maxi = zeros(n,1);
for i = 1:n
    di = sum((x(1:n~=i,:)-ones(n-1,1)*x(i,:)).^2,2);
    mini(i) = min(di);
    maxi(i) = max(di);
end
a = min(mini);
b = max(maxi);

t = linspace(a,b,100);
f = zeros(n,100);
for i = 1:n
    di = sum((x(1:n~=i,:)-ones(n-1,1)*x(i,:)).^2,2);
    f(i,:) = ksdensity(di,t);
end

[C,rowmaxarray]=max(transpose(f));
mini = min(rowmaxarray);
index = rowmaxarray(1,:)==mini;
q = find(index);
[o,p]=max(C(q)); % if more than one find highest peak
c = x(q(p),:);
```

B Boxplots

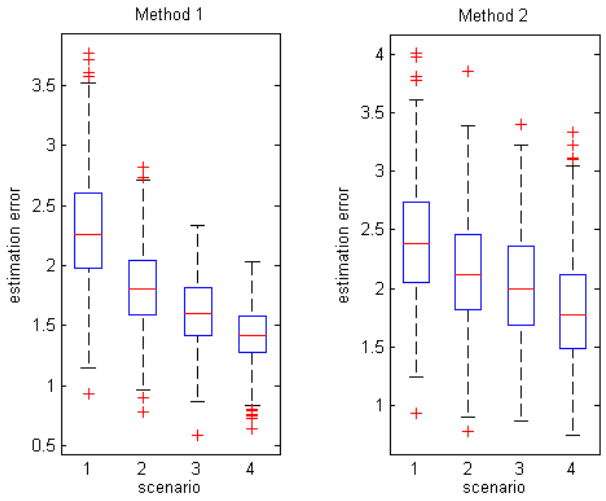
B.1 Standard Normal distribution



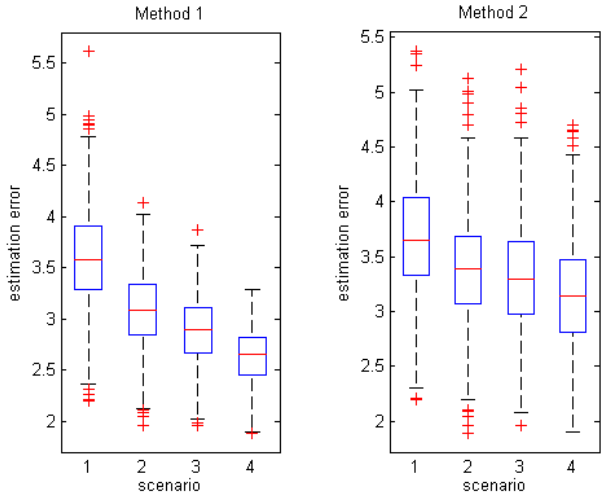
d = 2



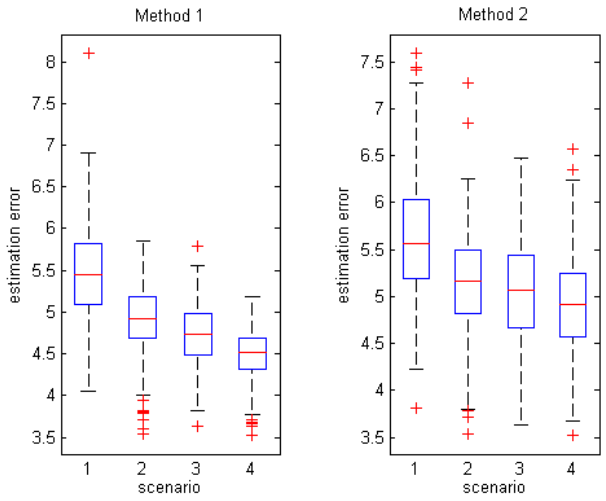
d = 5


Method 1
Method 2
estimation error
scenario
1
2
3
4
0.5
1
1.5
2
2.5
3
3.5
4

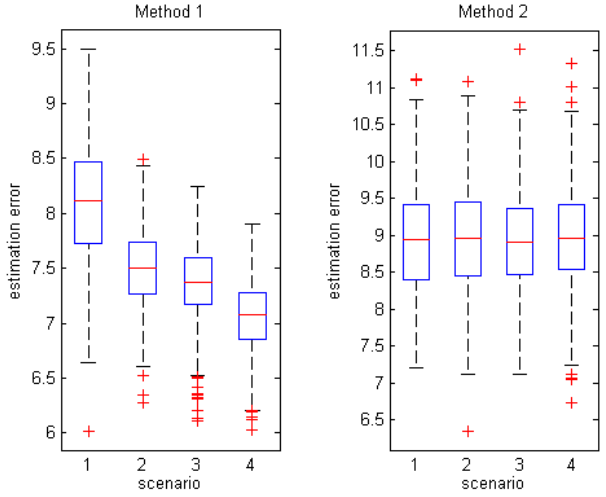
d = 10


Method 1
Method 2
estimation error
scenario
1
2
3
4
2
2.5
3
3.5
4
4.5
5
5.5

d = 20

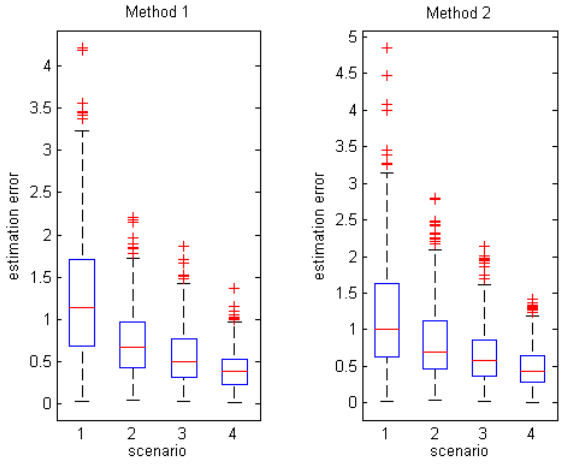

Method 1
Method 2
estimation error
scenario
1
2
3
4

d = 40

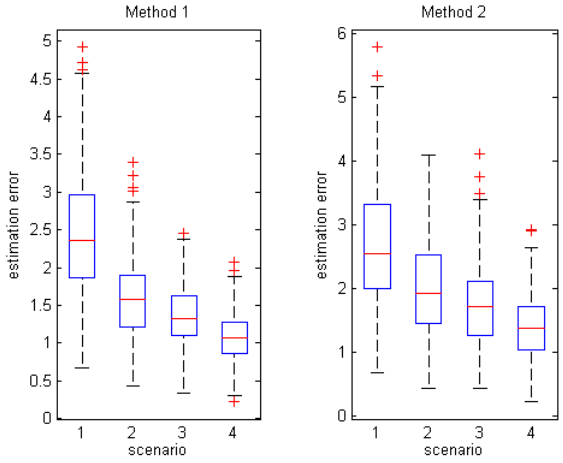

Method 1
Method 2
estimation error
scenario
1
2
3
4

d = 80

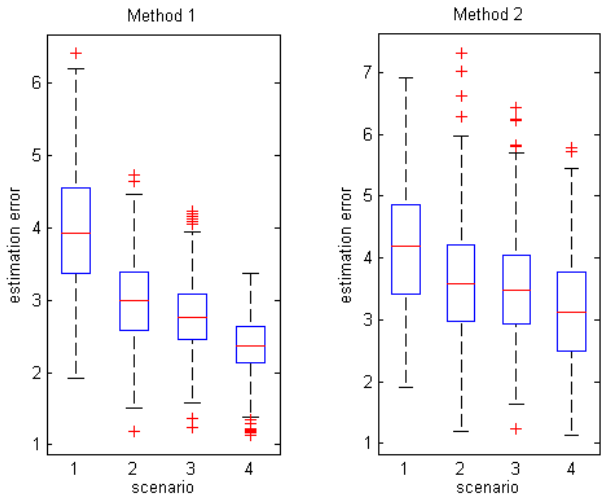
B.2 Normal distribution



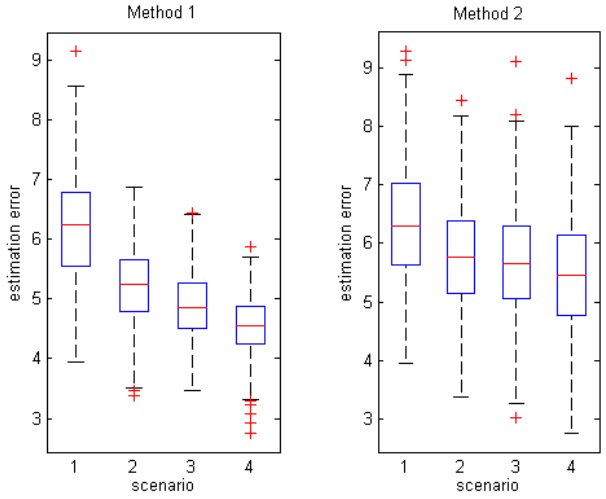
d = 2



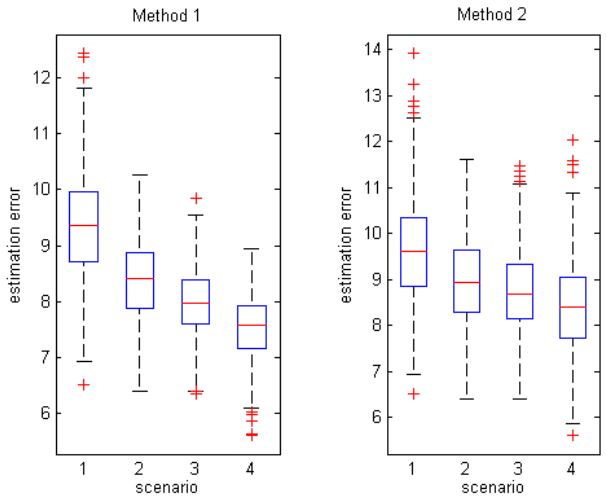
d = 5


Method 1
Method 2
estimation error
scenario
1
2
3
4
5
6
7

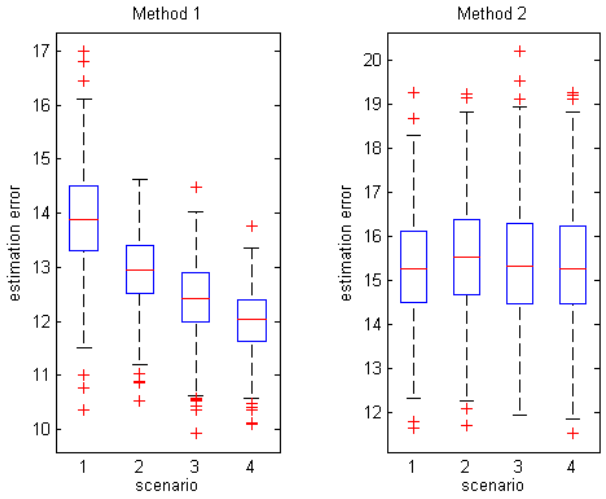
d = 10


Method 1
Method 2
estimation error
scenario
1
2
3
4
5
6
7
8
9

d = 20

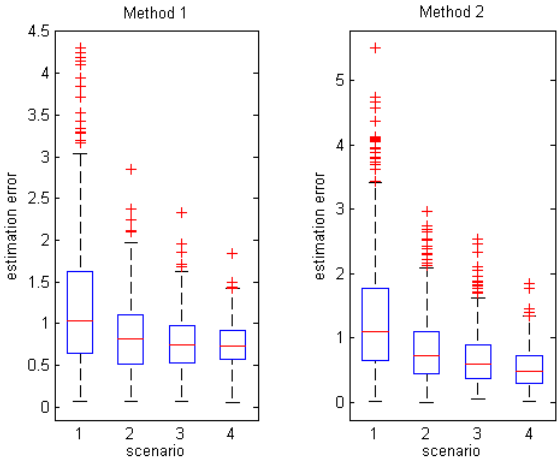

Method 1
Method 2
estimation error
scenario
1
2
3
4

d = 40

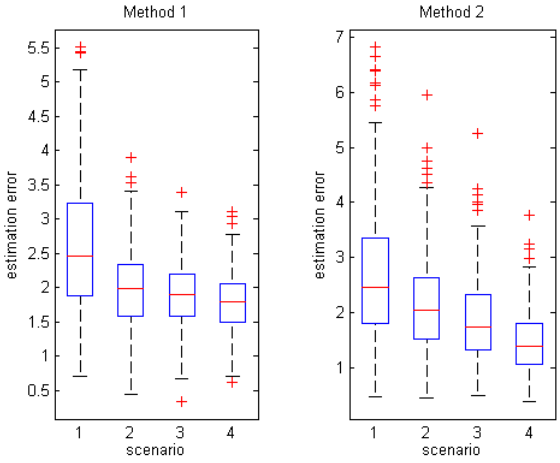

Method 1
Method 2
estimation error
scenario
1
2
3
4

d = 80

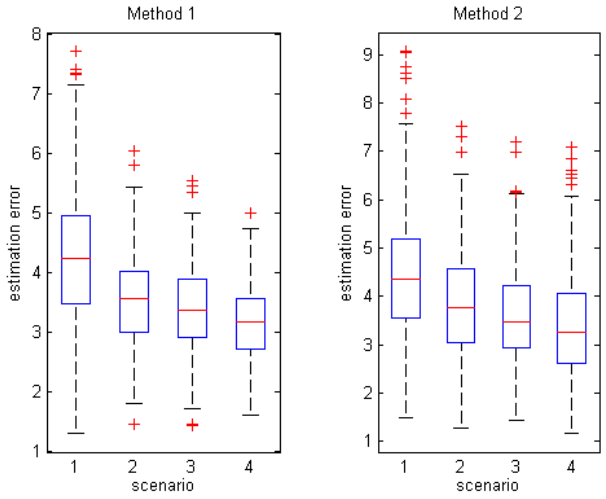
B.3 Gamma distribution



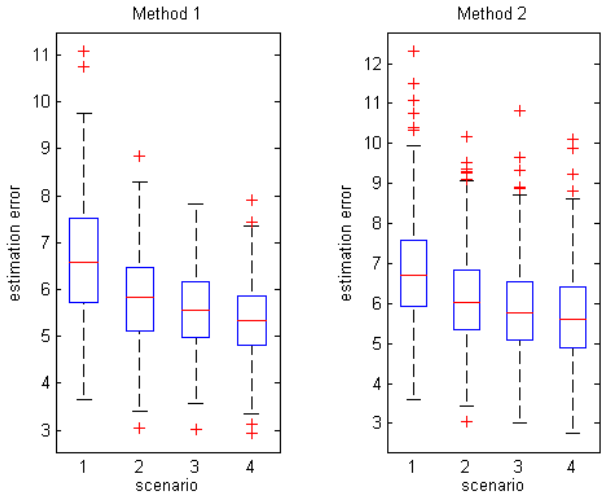
d = 2



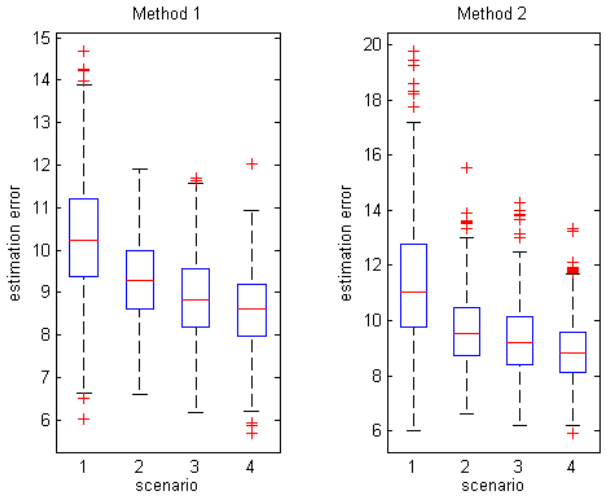
d = 5


Method 1
Method 2
estimation error
scenario
1
2
3
4

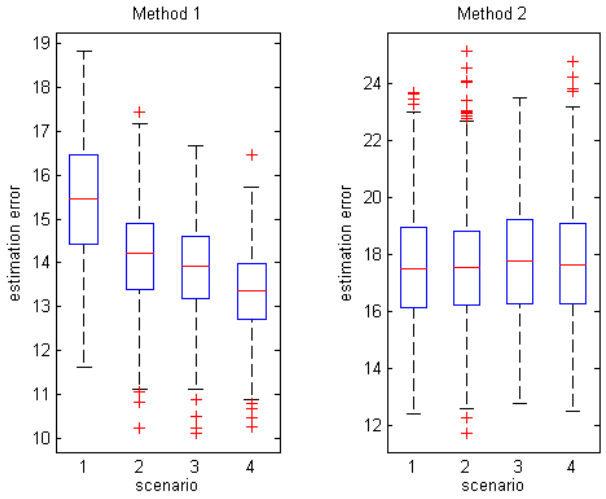
d = 10


Method 1
Method 2
estimation error
scenario
1
2
3
4

d = 20


Method 1
Method 2
estimation error
scenario
1
2
3
4

d = 40


Method 1
Method 2
estimation error
scenario
1
2
3
4

d = 80