Shape-Adaptive Kernel Density Estimation

L.E.N. Baakman

September 15, 2017

Abstract

Kernel density estimation is a popular method to approximate probability densities in numerous fields. Generally these methods use symmetric kernels, even though the data of which the density is estimated are not necessarily spread equally in all dimensions. To account for this asymmetric distribution of data we propose the use of shape adaptive kernels: kernels whose shape changes to fit the spread of the data in the local neighborhood. We compare the performance of the shape adaptive kernels with that of an estimator that uses a symmetric kernel on simulated datasets with known density fields. No significant differences in performance between the symmetric and the shape-adaptive estimator were found. Although the former outperformed the latter on points near the boundary of the datasets. In conclusion shape-adaptive kernels are a promising idea that warrants further research.

1 Introduction

Kernel density estimation is a popular method to approximate probability densities; in the medical field it has been used to predict dose-volume histograms, which are instrumental in the determination of radiation doses [7]. Ecologists have applied it to explore the habitats of seabirds [6]. Ferdosi et al. [4] have described it as "a critical first step in making progress in many areas of astronomy." Within this discipline density estimation is, among other things, used to estimate the density of the cosmic density field, which is required for the reconstruction of the large-scale structure of the universe.

Formally the aim of density estimation is to find the probability density $f(\mathbf{x})$ in the d-dimensional Euclidean space underlying N points $\mathbf{x}_1, \ldots, \mathbf{x}_N$, that have been selected independently from $f(\mathbf{x})$.

Kernel density estimation methods approximate $f(\mathbf{x})$ by placing bumps, referred to as kernels, on the different observations, and summing these bumps to arrive at a final density estimate. This paper is concerned with a method to make the shape of the kernels adaptive to their local neighborhood. Before introducing the process used to determine the form of the kernel we first review different symmetric kernel density estimation methods.

The simplest of which is the Parzen approach [8]. It approximates the density of some pattern $\mathbf{x}$ according to:

$$\hat{f}(\mathbf{x}) = \frac{1}{N}\sum_{i=1}^{N} h^{-d} K\left(\frac{\mathbf{x}-\mathbf{x}_i}{h}\right). \tag{1}$$

The shape of the used bumps is determined by the kernel function $K(\bullet)$, their width by the bandwidth h. The Parzen approach requires the kernel to be a probability density function, i.e. $K(\mathbf{x}) \geq 0$ and $\int K(\mathbf{x}) = 1$ [9]. The bandwidth directly influences the result of the density estimation process; a too small bandwidth results in a density estimate with spurious fine structures, whereas kernels that are too wide can oversmooth the density estimate. Kernel estimators, such as the Parzen approach, that use kernels of the same width for all $\mathbf{x}_i$, are called fixed-width estimators.

One downside of these methods is that the height of the peak of the kernel is not data-responsive. Consequently in low density regions the density estimate will have peaks at the few sample points and be too low elsewhere. Whereas in areas with high density the Parzen estimate is spread out, as the sample points are more densely packed together [2]. Adaptive-width methods address this disadvantage by allowing the width of the kernel to vary per data point. For example the estimator introduced by Breiman, Meisel, and Purcell [2] uses the distance between $\mathbf{x}_i$ and the k-nearest neighbor of $\mathbf{x}_i$, denoted by $D_{i,k}$, to determine the width of the kernel:

$$\hat{f}(\mathbf{x}) = \frac{1}{N}\sum_{i=1}^{N} (\alpha \cdot D_{i,k})^{-d} K_{\mathcal{G}}\left(\frac{\mathbf{x}-\mathbf{x}_i}{\alpha \cdot D_{i,k}}\right). \tag{2}$$

In this equation $K_{\mathcal{G}}$ is used to represent a Gaussian kernel, and α is a multiplicative constant. The values of both α and k can be determined with a minimization algorithm on a goodness of fit statistic. Comparing Equations (1) and (2) one finds that the bandwidth h of the Parzen estimator is defined as $\alpha \cdot D_{i,k}$ in Equation (2). The factor $D_{i,k}$ depends on the local neighborhood of $\mathbf{x}_i$, in low density regions this factor is large, and the kernel spreads out due to its high bandwidth. In areas with relatively many data points the converse occurs.

Silverman [9] shows that the minimization procedure used by Breiman, Meisel, and Purcell implicitly uses a k-NN pilot estimate. If pilot estimates, denoted by $\tilde{f}(\bullet)$, are used explicitly, the density estimation process becomes:

(i) Compute pilot densities with some estimator that ensures that $\forall i\ \tilde{f}(\mathbf{x}_i) > 0$.

(ii) Define local bandwidths γ_i as

$$\gamma_i = \left(\frac{\tilde{f}(\mathbf{x}_i)}{\mathrm{GM}\left(\tilde{f}(\mathbf{x}_1), \ldots, \tilde{f}(\mathbf{x}_N)\right)} \right)^{-\beta}, \quad (3)$$

where GM denotes the geometric mean and the sensitivity parameter β must lie in the range $[0, 1]$.

(iii) Compute the adaptive kernel estimate as

$$\hat{f}(\mathbf{x}) = \frac{1}{N} \sum_{i=1}^{N} (h \cdot \gamma_i)^{-d} K\left(\frac{\mathbf{x} - \mathbf{x}_i}{h \cdot \gamma_i} \right) \quad (4)$$

with K integrating to unity.

Since the pilot densities computed in step (i) do not need to be sensitive to the fine details of the pilot estimate a convenient method, e.g. the Parzen approach, can be used to estimate them [9]. The local bandwidths, computed in step (ii), depend on the exponent β. The higher this value is the more sensitive the local bandwidths are to variations in the pilot densities. Choosing $\beta = 0$ reduces Equation (4) to a fixed-width method. In the literature two values of β are prevalent. Breiman, Meisel, and Purcell [2] argue that choosing $\beta = {}^1/_d$ ensures that the number of observations covered by the kernel will be approximately the same in all areas of the data. Whereas Silverman [9] favors $\beta = {}^1/_2$ independent of the dimension of the data, as this value results in a bias that can be shown to be of a smaller order than that of the fixed-width kernel estimate.

One disadvantage of the Breiman estimator is its computational complexity. This is partially due to the use of a Gaussian kernel. Because of the infinite base of this kernel an exponential function has to be evaluated N times to estimate the density of one data point. Wilkinson and Meijer [10] address this in their Modified Breiman Estimator (MBE) by replacing the Gaussian kernel with a spherical Epanechnikov kernel in both the computation of the pilot densities and in the final density estimate. This kernel is defined as

$$K_{\mathcal{E}}(\mathbf{x}) = \begin{cases} \frac{d+2}{2c_d}(1 - \mathbf{x} \cdot \mathbf{x}) & \text{if } \mathbf{x} \cdot \mathbf{x} < 1 \\ 0 & \text{otherwise} \end{cases} \quad (5)$$

where c_d denotes the volume of the d-dimensional unit sphere [3]. It should be noted that the kernel defined in Equation (5) does not have unit variance. This can be corrected by multiplying the bandwidth, h, with the square root of the variance of $K_{\mathcal{E}}$, i.e. $\sqrt{5}$. There are two advantages to using this kernel, firstly it is computationally much simpler than the Gaussian kernel, partially due to its finite base, and secondly it is optimal in the sense of the Mean Integrated Square Error (MISE) [3]. A downside of this kernel is that it is not continuously differentiable. This is irrelevant when computing the pilot densities, however for the final densities one has to choose between a continuously differentiable density estimate and a density estimator that has a low computational complexity.

Ferdosi et al. [4] consider the application of density estimation on large datasets, i.e. sets with more than 50 000 points, where the dimension of the data points ranges from ten to hundreds of elements. They use the MBE, but introduce a computationally less complex method to estimate the bandwidth. First an intermediate bandwidth for each dimension l of the data is computed according to

$$h_l = \frac{P_{80}(l) - P_{20}(l)}{\log N}, l = 1, \ldots, d, \quad (6)$$

where $P_{20}(l)$ and $P_{80}(l)$ are the twentieth and eightieth percentile of the data in dimension l, respectively. The global bandwidth, h, is defined as the minimum of these intermediate bandwidths.

Although the widths of the kernels of the discussed adaptive-width methods are sensitive to the data, the shape of a kernel depends only on its definition, and is thus the same for all $\mathbf{x}_i$. To further increase the responsiveness of the estimator to the data we propose the use of shape-adaptive kernels; not only the width but also the shape of these kernels is steered by the local neighborhood of the data.

A possible disadvantage of these shape-adaptive kernels is that in regions where the density of sample points is low, the number of data points is insufficient to reliably compute the shape of the kernel.

Therefore we let the amount of influence exerted by the local data on the shape of the kernel depend on the number of data points in the local neighborhood.

This paper is organized as follows. Section 2 introduces the proposed shape-adaptive kernels. The experiment used to investigate the performance of these kernels is discussed in Section 3, the results are presented in Section 4. They are discussed in Section 5, and the reached conclusions can be found in Section 6.

2 Method

We use shape adaptive kernels in combination with the Modified Breiman Estimator introduced by Wilkinson and Meijer [10], the resulting estimator is henceforth referred to as the shape-adaptive Modified Breiman Estimator (saMBE). For its low computational complexity we use the method defined in Equation (6) to compute the general bandwidth. Pilot densities are computed according to Equation (1), with an Epanechnikov kernel. Since using $\beta = {}^1/_2$ instead of $\beta = {}^1/_d$ results in a final density approximation with a lower mean squared error for most of our datasets we use the first. We compute the final density estimate according to:

$$\hat{f}(\mathbf{x}) = \frac{1}{N}\sum_{i=1}^{N} \frac{1}{\det(\mathbf{H}_i)} K_{\mathcal{E}}\left(\mathbf{H}_i^{-1}(\mathbf{x}-\mathbf{x}_i)\right). \qquad (7)$$

The shape of the kernel $K_{\mathcal{E}}(\bullet)$ is determined by the bandwidth matrix $\mathbf{H}_i$ [5]. If $\mathbf{H}_i = h \cdot \gamma_i \cdot \mathbb{I}_{d\times d}$, Equation (7) reduces to Equation (4).

For each data point $\mathbf{x}_i$ that is used in the density estimation of some pattern $\mathbf{x}_j$, the bandwidth matrix is determined according to these steps:

(i) Find $C_{\mathbf{x}_i}$, the k-nearest neighbors of $\mathbf{x}_i$.

(ii) Compute Σ, the unbiased covariance matrix of the local neighborhood $C_{\mathbf{x}_i}$.

(iii) Determine $\mathbf{H}_i$ by scaling Σ with

$$s = h \cdot \gamma_i \left(\prod_{l=1}^{d} \lambda_l\right)^{-\frac{1}{d}} \qquad (8)$$

where $\lambda_1, \cdots, \lambda_d$ are the eigenvalues of Σ.

Step (i) determines the local neighborhood of $\mathbf{x}_i$ with a k-nearest neighbors search in a KD-tree [1], with Euclidean distance as the distance metric. We follow Silverman's [9] recommendation of choosing $k = \sqrt{N}$. To ensure that Σ is nonsingular we also need $k > d$, therefore

$$k = \max\left(\left\lfloor\sqrt{N}\right\rceil, d\right) + 1. \qquad (9)$$

Using a KD-tree for the k-nearest neighbors search instead of the naive implementation, significantly improves the time complexity of finding $\mathbf{H}_i$. The downside of using a space partitioning tree is that $C_{\mathbf{x}_i}$ is an approximation of the actual neighborhood, as long as k is rather large the use of an approximation instead of the exact k-nearest neighbors should not impact the final kernel result too strongly. We use k-NN rather than a fixed-radius neighborhood to ensure that, independent of the sparsity of the data, the kernel shape is always based on a reasonable number of data points.

The basic shape of the kernel is determined in step (ii). The covariance matrix ensures that the major axis of the kernel has the same direction as the maximum variance of the data.

The scaling factor computed in step (iii) ensures that the kernels used in the density estimation of different patterns have a comparable domain. Equation (8) scales the bandwidth matrix in such a way that the volume of the ellipsoid defined by the eigenvectors and values of $\mathbf{H}_i$ is equal to that of the eigenellipsoid of the bandwidth matrix that is implicitly used in Equation (4).

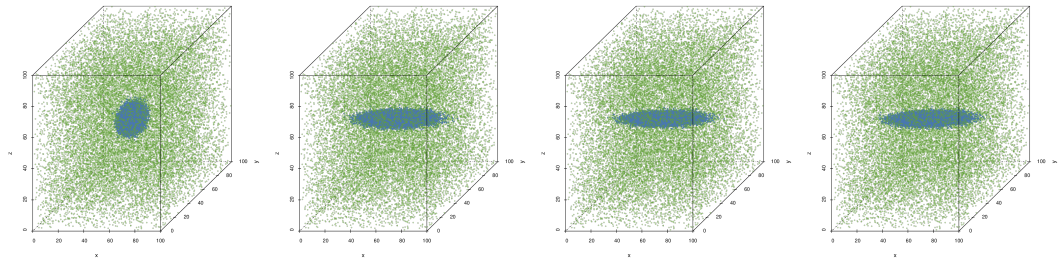
3 Experiment

We contrast the performance of the shape-adaptive and the symmetric Modified Breiman Estimator on simulated datasets with known density fields. This allows us to test how well the proposed method recovers simple density distributions in comparison to an existing method. The mean squared error (MSE) is used to quantify the performance of the estimators. We use

$$\frac{\max(\lambda_1, \cdots, \lambda_d)}{\min(\lambda_1, \cdots, \lambda_d)}$$

with $\lambda_1, \cdots, \lambda_d$ the eigenvalues of the bandwidth matrix, to express how anisotropic a kernel is. Two different types of datasets can be distinguished: datasets consisting of a single Gaussian distribution and noise, defined in Section 3.1 and datasets containing multiple Gaussian distributions embed in noise, these sets are presented in Section 3.2.

Before Final Version: Remove ticks and labels.



(a) *Set* S_1 **(b)** *Set* S_2 **(c)** *Set* S_3 **(d)** *Set* S_4

Figure 1: *Scatter plot representation of the datasets defined in Table 1. The used colors correspond to those associated with the different components in Table 1.*

		Component	Number	Distribution
S_1	•	Trivariate Gaussian	40000	$\mathcal{N}([50,50,50],\text{diag}(11))$
	•	Uniform random background	20000	$\mathcal{U}([0,0,0],[100,100,100])$
S_2	•	Trivariate Gaussian	40000	$\mathcal{N}([50,50,50],\text{diag}([11^2,\sqrt{11},\sqrt{11}]))$
	•	Uniform random background	20000	$\mathcal{U}([0,0,0],[100,100,100])$
S_3	•	Trivariate Gaussian	40000	$\mathcal{N}([50,50,50],\text{diag}([11,2*\sqrt{11},{}^1\!/_2\sqrt{11}]))$
	•	Uniform random background	20000	$\mathcal{U}([0,0,0],[100,100,100])$
S_4	•	Trivariate Gaussian	40000	$\mathcal{N}([50,50,50],\text{diag}([11^2,11,1]))$
	•	Uniform random background	20000	$\mathcal{U}([0,0,0],[100,100,100])$

Table 1: *The datasets containing a single Gaussian distribution embedded in uniform noise. The column 'Number' indicates for each component the number of patterns sampled from it.* $\mathcal{N}(\mu,\Sigma)$ *denotes a Gaussian distribution with mean* μ *and covariance matrix* Σ*. A diagonal matrix with the values* $x_1,\cdots,x_d$ *on the diagonal is represented as* $\text{diag}([x_1,\cdots,x_d])$*, a scalar matrix with x on the diagonal is shown as* $\text{diag}(x)$*.* $\mathcal{U}(a,b)$ *denotes a uniform distribution with its minimum and maximum set to a and b, respectively. The second column presents the symbol used to represent this component in plots throughout the paper.*

3.1 Datasets with a Single Gaussian

Figure 1 shows a scatter plot representation of the datasets defined in Table 1.

The Gaussian components of these datasets progress from a sphere, i.e. dataset S_1, to an increasingly more elongated ellipsoid. This makes it possible to investigate the influence of how strongly elongated the distribution is, on the density estimate. The first dataset is a simple spherical Gaussian distribution centered in a uniform random background. The covariance matrix of the Gaussian component in S_2 is created from S_1 by squaring one of the eigenvalues of the covariance matrix, and taking the square root of the other two eigenvalues, without changing the eigenvectors. The resulting covariance matrix defines an eigenellipse with the same volume as the one defined by S_1. The Gaussian component of dataset S_3 changes the shape of the eigenellipse of the Gaussian component by lengthening one of the minor axes, and shortening the other. In dataset S_4 the Gaussian component is spread out more along the y-axis and less along the z-axis, than the Gaussian component in dataset S_3.

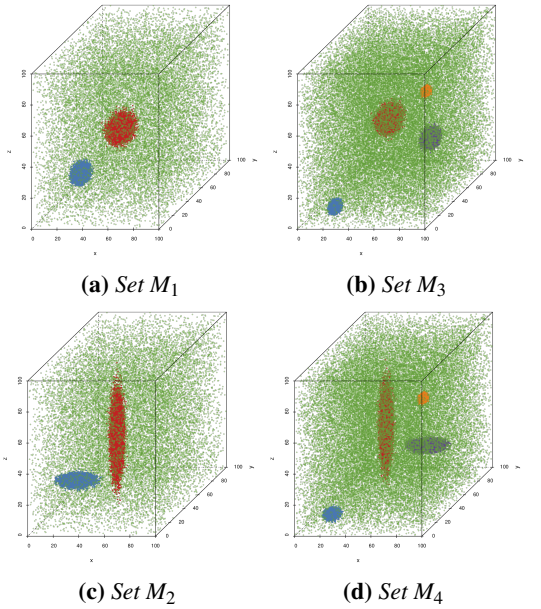
We expect the Modified Breiman Estimator and its shape-adaptive cousin to perform comparably on dataset S_1, since due to the symmetric shape of the Gaussian distribution no advantage should be gained by using a shape-adaptive kernel. As the Gaussian distribution is more and more elongated, the advantages of using saMBE should become more pronounced.

3.2 Datasets with Multiple Gaussians

Table 2 defines the datasets that consist of uniform random noise and multiple Gaussian distributions, a scatter plot representation of these sets is shown in Figure 2. Dataset M_1 consists of two Gaussian distributions, that are unlikely to overlap, embedded in noise. The first Gaussian component is significantly denser than the second. The procedure outlined in Section 3.1 for the creation of dataset S_2 was used to derive dataset M_2 from M_1. Dataset M_3 embeds four non-overlapping Gaussians, with eigenspheres with notably different radii, in the uniform random background. The last dataset, M_4, is a variation on M_3, created with the method that was used for the definition of dataset S_2 from S_1.

Due to the spherical nature of the Gaussian components we expect hardly any difference in performance between the estimators on dataset M_1 and M_3. Given the shape of the Gaussian distributions embedded in dataset M_2 and M_4 we hypothesize that saMBE outperforms MBE on these sets.

Before Final Version: Remove ticks and labels.



(a) *Set* M_1 **(b)** *Set* M_3 **(c)** *Set* M_2 **(d)** *Set* M_4

Figure 2: *Scatter plot representation of the datasets defined in Table 2, the colors used for the different components correspond to those in Table 2.*

Ferdosi et al. [4] found that the Modified Breiman Estimator resulted in lower integrated squared errors if fewer Gaussian distributions were present in the datasets. Since the presented datasets are comparable to those used by Ferdosi et al. we expect to find the same influence of the number of distributions on the error.

4 Results

This section presents the results of the experiments described in Section 3. We compare the performance of the two estimators on each dataset with the mean squared error and visually with plots. All plots associated with a single dataset have the same domain and range, to allow for easy comparison of the results within a dataset. The horizontal axis is used to represent the known densities, its range is such that each known density can be shown. The estimated densities are shown on the vertical axis, the length of these axes is such that they are long enough to represent every estimated density for that dataset, independent of the used estimator. The black line in each plot illustrates the line all points would lie on if a perfect estimator was used, i.e. the line $x = x$.

		Component	Number	Distribution
M_1	•	Trivariate Gaussian 1	20000	$\mathcal{N}([25,25,25],\text{diag}(5))$
	•	Trivariate Gaussian 2	20000	$\mathcal{N}([45,45,45],\text{diag}(11))$
	•	Uniform random background	20000	$\mathcal{U}([0,0,0],[100,100,100])$
M_2	•	Trivariate Gaussian 1	20000	$\mathcal{N}([25,25,25],\text{diag}([5^2,\sqrt{5},\sqrt{5}]))$
	•	Trivariate Gaussian 2	20000	$\mathcal{N}([45,45,45],\text{diag}([\sqrt{11},\sqrt{11},11^2]))$
	•	Uniform random background	20000	$\mathcal{U}([0,0,0],[100,100,100])$
M_3	•	Trivariate Gaussian 1	20000	$\mathcal{N}([24,10,10],\text{diag}(2))$
	•	Trivariate Gaussian 2	20000	$\mathcal{N}([33,70,40],\text{diag}(10))$
	•	Trivariate Gaussian 3	20000	$\mathcal{N}([90,20,80],\text{diag}(1))$
	•	Trivariate Gaussian 4	20000	$\mathcal{N}([60,80,23],\text{diag}(5))$
	•	Uniform random background	40000	$\mathcal{U}([0,0,0],[100,100,100])$
M_4	•	Trivariate Gaussian 1	20000	$\mathcal{N}([24,10,10],\text{diag}([4,\sqrt{2},\sqrt{2}]))$
	•	Trivariate Gaussian 2	20000	$\mathcal{N}([33,70,40],\text{diag}([\sqrt{10},\sqrt{10},100]))$
	•	Trivariate Gaussian 3	20000	$\mathcal{N}([90,20,80],\text{diag}(1))$
	•	Trivariate Gaussian 4	20000	$\mathcal{N}([60,80,23],\text{diag}([25,\sqrt{5},\sqrt{5}]))$
	•	Uniform random background	40000	$\mathcal{U}([0,0,0],[100,100,100])$

Table 2: *The datasets with multiple Gaussian distributions embedded in uniform noise. This table has the same structure and uses the same notation as Table 1.*

	Estimator	
	MBE	saMBE
S_1	8.306×10^{-9}	8.909×10^{-9}
S_2	1.490×10^{-8}	1.540×10^{-8}
S_3	2.937×10^{-8}	2.963×10^{-8}
S_4	5.572×10^{-8}	5.585×10^{-8}

Table 3: *Performance of the Modified Breiman Estimator with fixed-shaped and shape-adaptive kernels on the datasets with a single Gaussian.*

	• Gaussian		• Noise	
	M	SD	M	SD
S_1	1.48	0.521	1.29	0.136	1.86	0.750
S_2	1.57	0.553	1.41	0.289	1.91	0.766
S_3	1.64	0.586	1.51	0.403	1.91	0.772
S_4	1.80	0.698	1.74	0.638	1.93	0.790

Table 4: *The mean (*M*) and the standard deviation (*SD*) of the anisotropy of the kernels used for the datasets with a single Gaussian.*

The colors of the points in these plot correspond to the colors of the elements of the datasets in Tables 1 and 2.

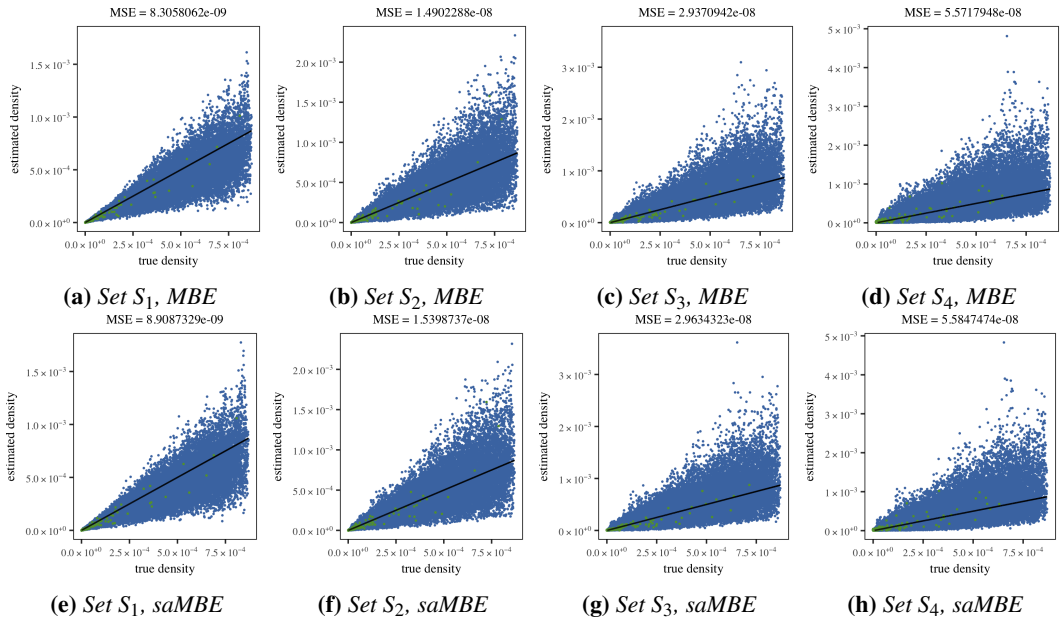
Section 4.1 presents the results of the datasets that contain a single Gaussian, in Section 4.2 the results of the datasets that consist of noise and multiple Gaussian distributions are presented.

4.1 Datasets with a Single Gaussian

This section compares the performance of the Modified Breiman Estimator with symmetric and shape-adaptive kernels on datasets that contain one Gaussian. Comparing the mean squared errors of the MBE with those of saMBE in Table 3 we find that the two estimators perform comparably, but that the fixed-shape estimator always gives a slightly lower mean squared error.

This is confirmed by the visualization of the results in Figure 3 where hardly any difference is visible between Figures 3(a) to 3(d), and Figures 3(e) to 3(h), respectively. Comparing the plots associated with dataset S_1 we find that the shape-adaptive estimators tends to overestimate densities more than the symmetric observation. Based on Figures 3(c) and 3(g) the same holds for dataset S_3. Comparing the performance within datasets between the two components showed no marked differences in performance between the estimators between components.

Table 4 presents the mean and the standard deviation of the anisotropy of the kernels used for the different datasets. Comparing the means we find a positive correlation between the anisotropy of the Gaussian component of the dataset and mean anisotropy of the kernels. The same positive correlation can be observed for the standard deviation. Reviewing these statistics of the components of the datasets reveals that the increase in average anisotropy is primarily caused by an increase in anisotropy op kernels of points sampled from the Gaussian component. The mean anisotropy of the noise component



(a) *Set S_1, MBE* **(b)** *Set S_2, MBE* **(c)** *Set S_3, MBE* **(d)** *Set S_4, MBE*

(e) *Set S_1, saMBE* **(f)** *Set S_2, saMBE* **(g)** *Set S_3, saMBE* **(h)** *Set S_4, saMBE*

Figure 3: *Plot of the density as estimated by (a)-(d) MBE and (e)-(h) saMBE as a function of the known density of the datasets with a single Gaussian.*

	Estimator	
Set	MBE	saMBE
M_1	5.058×10^{-8}	5.050×10^{-8}
M_2	5.147×10^{-8}	5.168×10^{-8}
M_3	4.375×10^{-6}	4.463×10^{-6}
M_4	4.189×10^{-6}	4.284×10^{-6}

Table 5: *Performance of the symmetric and the shape-adaptive Modified Breiman Estimator on the datasets containing multiple Gaussian distributions.*

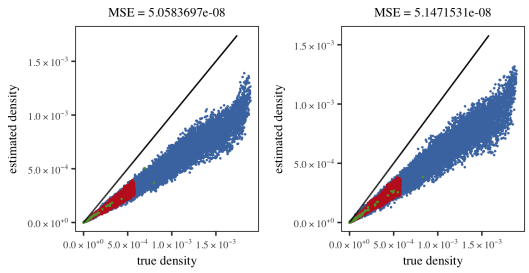
stays relatively constant. Furthermore as the Gaussian component is more anisotropic the variation in anisotropy of the kernels increases.

To summarize; in spite of differences in anisotropy of the used kernels we have observed very few differences between two estimators. Using shape-adaptive kernels did not yield the expected gain in performance. We did find the expected influence of the anisotropy of the Gaussian components on that of the kernels.

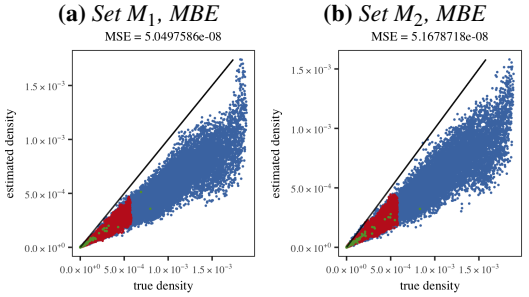
4.2 Datasets with Multiple Gaussians

In this section we present the results of the two estimators on dataset M_1, M_2, M_3, and M_4. Based on the small differences between the mean squared errors of the estimators in Table 5 the estimators perform comparably on these datasets. Comparing the MSE between components within the data sets between estimators yields no differences. However within data sets the difference in mean squared errors are quite large. Within dataset M_1 and M_2 both estimators perform significantly better on the more sparse component 'Trivariate Gaussian 2'. Both estimators show the same positive correlation between the density of the Gaussian components and the MSE between the components of dataset M_3 and M_4. Additionally contrary to our expectation both estimators performed slightly better on dataset M_4 than on the set it was derived from.

Figure 4 shows the estimated density as a function of the known density for both estimators and both datasets with two Gaussians. These plots show that both estimators underestimate the density, MBE more than saMBE. Furthermore the shape-adaptive estimator shows more variation in the densities it estimates than the symmetric estimator. Comparing Figures 4(a) and 4(b) with Figures 4(c) and 4(c) suggest that saMBE hardly underestimates the densities of the most anisotropic component, i.e. 'Trivariate Gaussian 2' in bot dataset M_1 and M_2. However the difference in mean squared error of this component between MBE and saMBE is pretty small, a large difference in standard deviation of the squared error suggests that the seemingly better performance of the shape-adaptive estimator in Figures 4(c) and 4(c) is mostly due to the higher spread of densities esti-

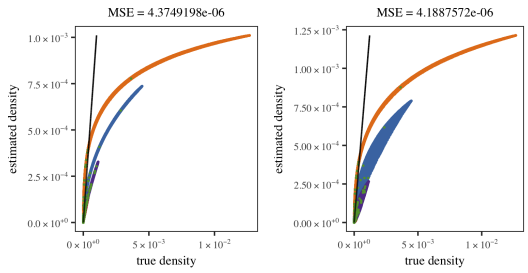


(a) *Set M_1, MBE* **(b)** *Set M_2, MBE*

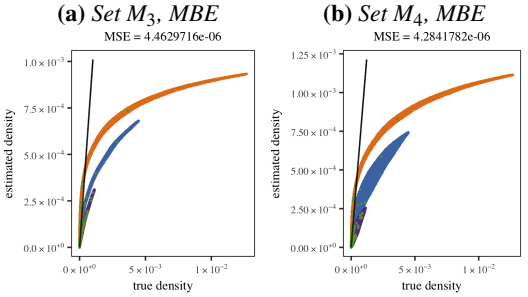


(c) *Set M_1, saMBE* **(d)** *Set M_2, saMBE*

Figure 4: *Plots of the true versus estimated density of datasets M_1 and M_2 for the shape-adaptive and the symmetric Modified Breiman Estimator.*



(a) *Set M_3, MBE* **(b)** *Set M_4, MBE*



(c) *Set M_3, saMBE* **(d)** *Set M_4, saMBE*

Figure 5: *The estimated density plotted as a function of the true density for datasets M_3 and M_4 for MBE and saMBE.*

mated by the shape-adaptive estimator. Furthermore the difference in mean squared error between components within estimators and dataset M_1 and M_2 shows that both estimators perform worse on the denser Gaussian component, i.e. 'Trivariate Gaussian 1'. The plots in Figure 5 confirm the large difference in performance of both estimators between datasets with two and four Gaussian components. Moreover they show that both MBE and saMBE underestimate densities, especially on the points with a high density. Figure 5 also shows that the spread of the densities estimated by saMBE is higher than of those estimated by MBE.

Table 6 presents the mean and standard deviation of the anisotropy of the kernels used for the points from dataset M_1 through M_4 and from their components. Although the anisotropy of the kernels used for the datasets with anisotropic Gaussian components is higher on average and more varied than the anisotropy of the kernels used for dataset M_1 and M_3, the differences are small. As in Table 4 the kernels associated with the points sampled from the component 'Uniform random background' have the highest anisotropy, and vary the most in how anisotropic they are. Contrasting the mean anisotropy of the kernels used for points drawn from the different components we find a positive correlation between the mean anisotropy of the kernel and the anisotropy of the Gaussian components. In the kernels used for dataset M_2 we observe the same positive correlation between the variation of the anisotropy of the kernels and the anisotropy of the associated kernel as we observed in dataset S_2, S_3, and S_4. Comparing the standard deviation of the anisotropy of the kernels used for the points sampled from the components of dataset M_4 does not reveal this relation: the densest component, Trivariate Gaussian 3, does not have the highest variation in kernel anisotropy. One thing that stands out when comparing dataset M_3 and M_4 in Table 6 is the lack of difference in the mean and standard deviation in the anisotropy of the kernels associated with the component 'Trivariate Gaussian 3'. Reviewing the raw data reveals that the largest difference in anisotropy between any point drawn from that component is 3.109×10^{-15}.

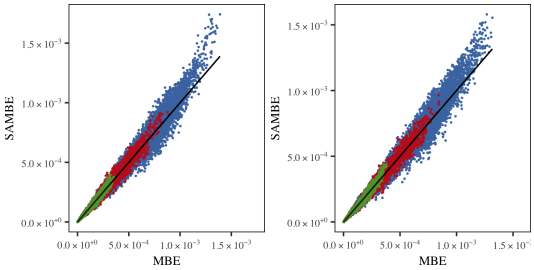
To summarize the results of the datasets with multiple Gaussian components: we have found that the differences in performance are very small between the two estimators, although saMBE underestimates less than MBE, the later performs slightly better. Furthermore both estimators show a positive correlation between the density of the Gaussian component and their performance on that component. Regarding the anisotropy of the kernels we have found only a small increase between dataset with spherical kernels and dataset M_2 and M_4. Zeroing in on the components we have observed that the anisotropy of

			Gaussian 1		Gaussian 2		Gaussian 3		Gaussian 4		Noise	
	M	SD	M	SD	M	SD	M	SD	M	SD	M	SD
M_1	1.50	0.531	1.32	0.175	1.30	0.143					1.89	0.759
M_2	1.61	0.570	1.41	0.278	1.49	0.345					1.95	0.783
M_3	1.46	0.551	1.29	0.189	1.27	0.130	1.29	0.210	1.28	0.165	1.82	0.811
M_4	1.53	0.571	1.31	0.219	1.49	0.339	1.29	0.210	1.40	0.285	1.85	0.820

Table 6: *The mean (*M*) and standard deviation (*SD*) of the anisotropy of the kernels used for points from the datasets with multiple Gaussians, split per component and for the full dataset.*

kernels associated with points drawn from the noise is strongest. Lastly a positive correlation between the mean anisotropy of the kernels of the points sampled from a component and the anisotropy of the Gaussian those points were drawn from was found.

In conclusion for all datasets we have found very small differences in mean squared error between the two estimators. Furthermore both estimators perform better on points drawn from Gaussian components with a smaller eigensphere. Concerning the anisotropy of the kernels we have observed that in all datasets this statistic is highest for the noise component. Comparing the mean anisotropy of the kernels of the points drawn from the Gaussian components showed a positive correlation with the anisotropy of the component the point was sampled from.



(a) *Dataset* M_1 **(b)** *Dataset* M_2

Figure 6: *Plots of the density estimated by saMBE as a function of those estimated by MBE for dataset (a)* M_1 *and (b)* M_2.

5 Discussion

This section discusses the results presented in Section 4. We consider the lack of difference in performance between the estimators in Section 5.1. The next section discusses the anisotropy of the kernels used by the shape-adaptive estimators. Finally we offer some directions research into solving the identified issues might take.

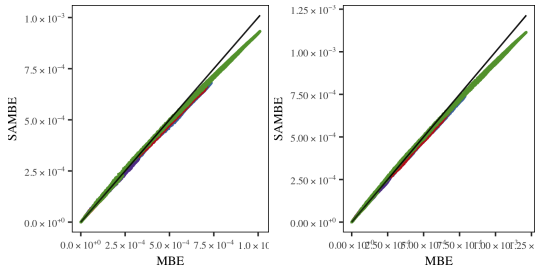
5.1 Performance

One of the most striking observations from Section 4 is that the differences in performance between the two estimators are minimal.

Plotting the densities as estimated by saMBE as a function of those estimated by MBE shows no interesting differences between the two estimators for dataset S_1 through S_4. However for dataset M_1 through M_4 these plots are interesting. As can be seen in Figure 6 using shape-adaptive kernels results in estimated densities that are generally higher than those estimated with a fixed-shape kernel for dataset M_1 and M_2. A review of the raw data shows that the points whose density is estimated by saMBE to be significantly higher than MBE, and nearer to the true density, can all be found near the mean of "Trivariate Gaussian 1". The kernels in this neighborhood are all slightly anisotropic, and have allowed the shape-adaptive estimator to use more data points, to better approximate the local densities. Thus showing that at least for some points estimating the density with shape-adaptive kernels is advantageous.

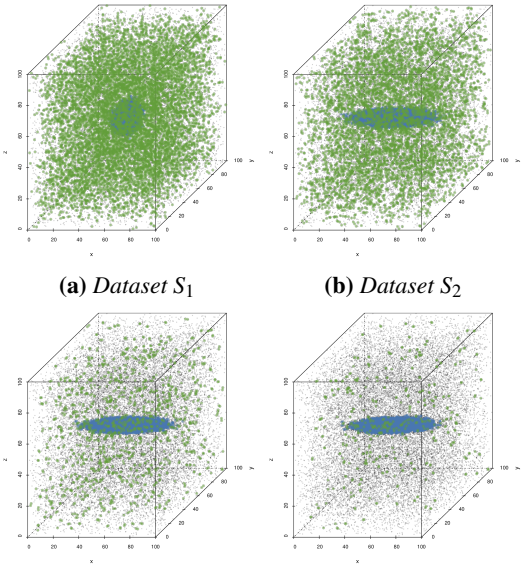
Figure 7 shows the opposite effect; the densities estimated by saMBE for dataset M_3 and M_4 are generally lower than those estimated by MBE. Reviewing the raw data shows that the points with the greatest differences between the two estimators lie near the mean of the 'Trivariate Gaussian 3' in both dataset M_3 and M_4. The number of points used in the density estimate by saMBE is consistently lower and than those used by the fixed-shape estimator. Given the relatively high anisotropy of the kernels in that area we expect that this is due to the kernels reflecting fine local structures, that are not relevant to the more global neighborhood.

The plots in Figure 8 emphasizes the points in dataset S_1 through S_2 where the absolute error of MBE is smaller than that of saMBE. These plots show that the shape-adaptive kernels outperforms the symmetric kernels near the borders of the datasets. We expect that the boundary effect is due



(a) *Dataset* M_3 **(b)** *Dataset* M_4

Figure 7: *Plots of the density estimated by saMBE as a function of those estimated by MBE for dataset (a)* M_3 *and (b)* M_4.

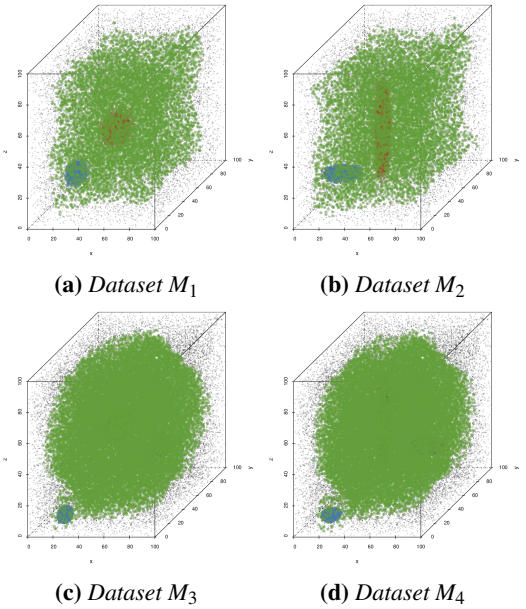


(a) *Dataset* S_1 **(b)** *Dataset* S_2

(c) *Dataset* S_3 **(d)** *Dataset* S_4

Figure 8: *Low opacity scatter plot of dataset (a)* S_1*, (b)* S_2*, (c)* S_3*, and(d)* S_4*, with an overlay of larger colored points where the absolute error of saMBE is larger than the absolute error of MBE.*

to the strong anisotropy of the local neighborhood of the points near the boundaries. After all these neighborhoods all point strongly towards the dataset. Consequently the domain of the shape-adaptive kernels extends less outside of the boundaries dataset than the domains of the symmetric kernels. This is results in less underestimating of densities near the boundary of the dataset if shape-adaptive kernels are used. Furthermore the strength of the boundary effect seems to increase as the Gaussian component of the dataset is more anisotropic. However the seemingly better performance of saMBE is due to an increase in the number of points where the density estimated by saMBE equals that estimated by MBE. In dataset S_1 the density estimates of the two estimators differ on all points, however this percentage increases as the anisotropy of the Gaussian component increases, to 35.4 % in dataset S_4. As the Gaussian component becomes more anisotropic there are more points whose local neighborhood consists only of uniform noise. On average the covariance matrix of neighborhoods that contain primarily points sampled from the noise component should be scalar. Consequently the shape-adaptive kernels of a large number of points sampled from the noise component are symmetric, resulting in points where the estimators give the same approximated density.

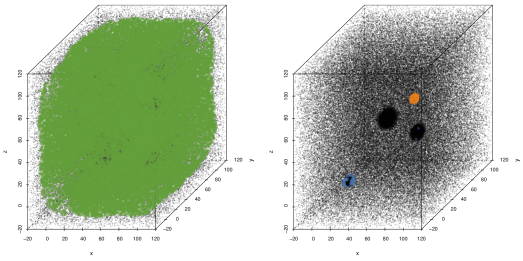


(a) *Dataset* M_1 **(b)** *Dataset* M_2

(c) *Dataset* M_3 **(d)** *Dataset* M_4

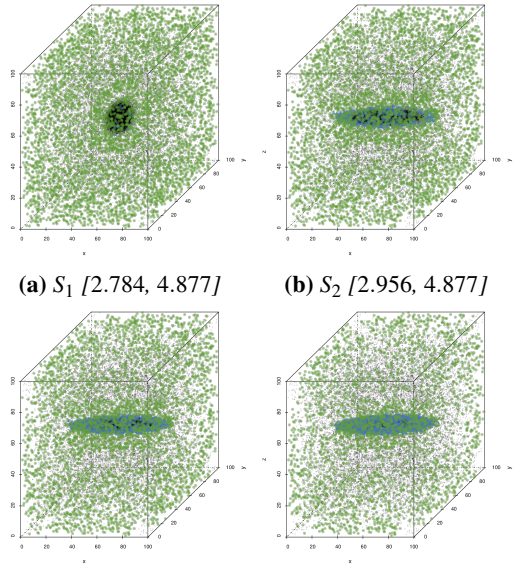
Figure 9: *Low opacity scatter plot of dataset (a)* M_1*, (b)* M_2*, (c)* M_3*, and (d)* M_4 *with an overlay of high opacity larger points where the absolute error of MBE is smaller than that of saMBE.*

The points where using symmetric kernels results in a smaller error in datasets M_1 through M_4 are emphasized in Figure 9. We contribute the boundary effect in these datasets to the same cause as the boundary effects in the datasets with a single Gaussian component. Interestingly the points dataset M_3 and M_4 where the absolute error of MBE is lower approximate a sphere, contrary to the cube they define for dataset M_1 and M_2. It is our expectation that this is caused by smaller distance between the means of the Gaussian components and the boundary of the dataset. To test this we define dataset M'_3, this dataset replaces the uniform random background of M_3 with $\mathcal{U}\big([-20,-20,-20],[120,120,120]\big)$. To



(a) *Absolute Error* **(b)** *Anisotropy*

Figure 10: *Low opacity scatter plot of dataset M'_3 with (a) points where the absolute error of MBE is lower than that of saMBE and (b) points sampled from the Gaussian components with kernels whose anisotropy falls in the 95^{th} percentile of the complete dataset emphasized.*



(a) S_1 *[2.784, 4.877]* **(b)** S_2 *[2.956, 4.877]*

(c) S_3 *[3.106, 5.987]* **(d)** S_4 *[3.546, 8.192]*

Figure 11: *Scatter plot of the data sets (a) S_1, (b) S_2, (c) S_3, and (d) S_4. The points with kernels whose anisotropy lies in the 90^{th} percentile are shown larger and in the color of the component they were drawn from. The range of the anisotropy of the kernels of the emphasized points in shown below each plot.*

ensure that the density of that components is equal to that of the noise of M_3 we adjust the number of points of the component. Compared to M_3, the overall mean squared error of both estimators has decreased slightly for M'_3, however the mean squared error of 'Trivariate Gaussian 1' and 3 shows a small increase. Figure 10(a) confirms that the spherical shape in Figures 9(c) and 9(d) is caused by the Gaussian near the boundary of the dataset.

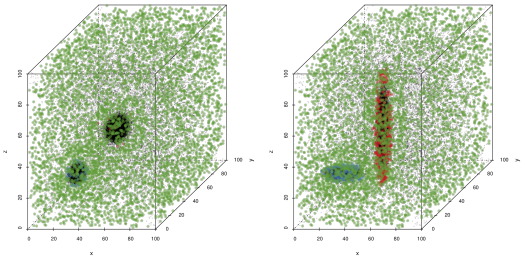
To conclude, by investigating the differences in performance between the two estimators on various datasets we have found that shape-adaptive kernels definitely improve performance in some cases, e.g. near the boundary of the datasets and near the mean of some Gaussian components. Unfortunately in other cases the anisotropic kernels are detrimental, the difference in mean squared error between the two estimators show that on these datasets using symmetric kernels is slightly advantageous.

5.2 Kernel Anisotropy

In Section 4 some differences in anisotropy of the kernels were observed, however these differences were relatively small. This section investigates the anisotropy of the kernels.

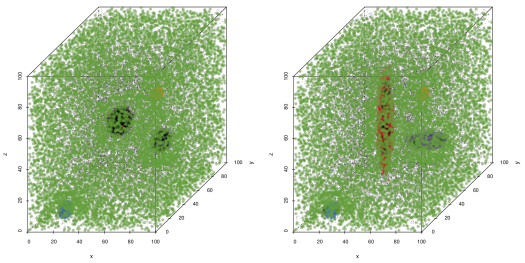
Figure 11 shows the datasets with a single Gaussian with points whose anisotropy lies in the 90^{th} percentile emphasized. In this figure hardly any difference is visible between Figures 11(a) and 11(b). In Figures 11(a) and 11(b) 0.518 % and 11.7 %, respectively of the emphasized points are sampled from the Gaussian component of the datasets. Illustrating that the kernels in dataset S_2 are influenced by the anisotropy of the Gaussian component of that dataset. Furthermore a shell of points with kernels with relative high anisotropy, sampled from the noise, seems to surround the Gaussian component, it is quite likely that the shape of the kernel of these points is influenced by the Gaussian component. We expect that nearer to the mean of the Gaussian component fewer kernels are influenced by its anisotropy as the physical density of points is quite high in that area. Consequently the volume of the local neighborhood is quite small, and is therefore not representative of the shape of the Gaussian component. In dataset S_3 and S_4 the number of points with a kernel with a high anisotropy sampled from the Gaussian components is higher than in dataset S_1 and S_2, to be exact 21.9 % and 42.1 %, respectively. Thus as the anisotropy of the Gaussian component increases, the anisotropy of the kernels of the points near that component increase.

Figure 12 emphasizes the points with the most anisotropic kernels in the dataset M_1 through M_4. In the plot associated with dataset M_1 we observe the same shells of high anisotropy kernels of points sampled from the noise around the Gaussian components as in dataset S_1. Another similarity between these two datasets is that very few points sampled from the Gaussian component have a ker-



(a) M_1 *[2.216, 5.678]*

(b) M_2 *[2.463, 7.135]*



(c) M_3 *[2.139, 8.856]*

(d) M_4 *[2.316, 8.856]*

Figure 12: *Scatter plot of datasets (a) M_1, (b) M_2, (c) M_3, and (d) M_4. The points that have an anisotropy in the 90^{th} percentile are shown larger and in the color of the component they were drawn from. The range of the anisotropy of the kernels of the emphasized points in shown below each plot.*

nel with high anisotropy; 1.45 % and 0.117 % of the first and second Gaussian component, respectively. We contribute the difference in the number of highly anisotropic kernels associated with data points sampled from the two Gaussian components to the difference in density between the two Gaussian components. Comparing Figures 12(a) and 12(b) we find that the increase in anisotropy of the kernel has causes 4.33 % and 8.66 % of the kernels with high anisotropy to be associated with a point sampled from 'Trivariate Gaussian 1' and 2, respectively. Interestingly in dataset M_3, two of the four spherical Gaussians, i.e. 'Trivariate Gaussian ' are associated with respectively 1.40 % and 2.22 % of the highly anisotropic kernels. Comparing the mean kernel anisotropy in Table 6 we find that compared to the other Gaussian components in M_3, these two component have a relatively low mean. However their standard deviations are higher, suggesting that in these denser components some kernels are extremely anisotropic, whereas others are near to isotropic. We contribute this difference within the points sampled from a Gaussian component to difference in physical density of the data points nearer and father away from the mean as we

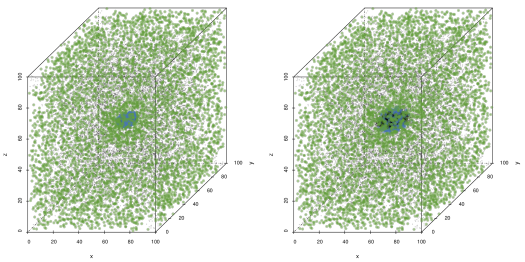
	Estimator	
	MBE	saMBE
A_1	1.240×10^{-6}	1.297×10^{-6}
A_2	4.713×10^{-8}	4.941×10^{-8}

Table 7: *Performance of the Modified Breiman Estimator with fixed-shaped and shape-adaptive kernels on the datasets A_1 and A_2.*

did for dataset S_1 and S_2. Component one and four of dataset M_3 differ from the other Gaussian components in two aspects: firstly they have a relatively dense and secondly they are placed near the boundaries of the dataset. Figure 10(b) shows that distance to the boundaries do not explain the difference in anisotropy of the kernel as in that figure the first and third component of M'_3 have more anisotropic kernels than the other components. The first explanation fits with the observations from Section 4 that components with a higher density have a higher anisotropy. In dataset M_4 we observe the same effect as in M_3 but stronger; from the points with the most anisotropic kernels more are sampled from the two densest Gaussian component than from the others.

Figures 11 and 12 show that the overwhelming majority of the points with a relative highly anisotropic kernel are sampled from the 'Uniform random background'. We contribute this to the covariance detecting fine, random structures in the noise, that give the impression of anisotropy where there is none.

In Section 4 we observed that both the density and the anisotropy of the Gaussian component influenced the anisotropy of the kernels. To test if the difference was caused by the anisotropy or the density of the dataset we introduce two datasets, A_1 and A_2, with the same anisotropy but the same density. To create dataset A_1 the covarianceMatrix used in S_1 was replaced by $\mathrm{diag}([4,3,1]$, dataset A_2 is the same as set A_1 but uses $3 \cdot \mathrm{diag}([4,3,1]$. The mean squared errors of the two estimators on dataset A_1 and A_2 are shown in Table 7, clearly both estimators perform better on the dataset with higher values on the diagonal of the covariance matrix of the Gaussian component. Figure 13 illustrates the influence of the density of the Gaussian component on the anisotropy of the kernels near the Gaussian component. 6.23 % of the points with a highly anisotropic kernel in dataset A_1 is sampled from the Gaussian component, in dataset A_2 this is 4.16 %. Although the difference is small, this does illustrate that the density of a component, irrespective of its anisotropy, influences the



(a) A_1 **(b)** A_2

Figure 13: *Scatter plot of the data sets (a) A_1 and (b) A_2 with emphasized the points whose kernels with anisotropy in the 90^{th} percentile.*

anisotropy of the kernels.

We have found that near low density Gaussian components the kernels are too isotropic, whereas in the uniform random background fine structures are detected that result in too anisotropic kernels. Both of these problems might be solved by increasing the value of k. We expect that in a larger local neighborhood fewer fine structures are detected, resulting in more isotropic kernels for points that lie in the uniform background. Furthermore increasing the size of the local neighborhood might also allow the kernels to adapt to their shape to nearby low density Gaussian components. Increasing the k computed in Equation (9) resulted in saMBE outperforming MBE on dataset S_4. Furthermore the increased k also improved the performance on dataset M_3, M_4 and those with a single Gaussian, whereas the performance on dataset M_1 and M_2 decreased. This suggest that blindly increasing the size of the local neighborhood will not consistently improve the performance of the estimator, but that same adaptive increase of k is required.

Another solution to the too anisotropic kernels in the noise might be too only use shape-adaptive kernels if the neighborhood is sufficiently anisotropic. One might even consider using a kernel-shape that is somewhere between the isotropic and the fully anisotropic kernel depending on the anisotropy of the local neighborhood. A difficulty with this approach is detecting the isotropy of the local neighborhood. As the obvious solution, the covariance matrix, has proven sensitive to fine structures in the noise.

Finally the correlation between the performance of the estimators and the density of the Gaussian component suggests that the method used to compute the local bandwidths is far from ideal.

6 Conclusion

We have found that the shape-adaptive Modified Breiman Estimator gives results comparable to those of the symmetric Modified Breiman Estimator. Especially near the boundary of the datasets anisotropic kernels prove advantageous. Reviewing the raw data shows that using anisotropic kernel is definitely advantageous. However the number of points where the kernel is too isotropic, or too anisotropic for its local neighborhood negate this advantage. One of the main problems is that fine structures in the background result in kernels with a relatively high anisotropy. Whereas Gaussian components are surrounded by points with highly anisotropic kernels, the shape of components with a lower density is hardly detected. Resulting in too isotropic kernels.

These issues might be addressed by increasing the size of the local neighborhood or by deciding the strength. Another possible improvement could be to decide the strength of the anisotropy of the kernel based on the local neighborhood of the associated data point.

In conclusion shape-adaptive kernels are a promising idea that definitely warrants the research needed to work out the kinks identified in this paper.

References

[1] Jon Louis Bentley. “Multidimensional Binary Search Trees Used for Associative Searching”. In: *Commun. ACM* 18.9 (1975), pp. 509–517. URL: `http://doi.acm.org/10.1145/361002.361007`.

[2] L. Breiman, W. Meisel, and E. Purcell. “Variable Kernel Estimates of Multivariate Densities”. In: *Technometrics* 19.2 (1977), pp. 135–144.

[3] V.A. Epanechnikov. “Non-Parametric Estimation of a Multivariate Probability Density”. In: *Theory of Probability & Its Applications* 14.1 (1969), pp. 153–158.

[4] B.J. Ferdosi et al. “Comparison of Density Estimation Methods for Astronomical Datasets”. In: *Astronomy & Astrophysics* 531 (2011).

[5] Wolfgang Härdle et al. *Nonparametric and semiparametric models*. Springer Series in Statistics. Springer Science & Business Media, 2012.

[6] Kirsty J Lees, Andrew J Guerin, and Elizabeth A Masden. “Using kernel density estimation to explore habitat use by seabirds at a marine renewable wave energy test facility”. In: *Marine Policy* 63 (2016), pp. 35–44.

[7] Johanna Skarpman Munter and Jens Sjőlund. “Dose-volume histogram prediction using density estimation”. In: *Physics in Medicine and Biology* 60.17 (2015), p. 6923.

[8] E. Parzen. “On Estimation of a Probability Density Function and Mode”. In: *The Annals of Mathematical Statistics* 33.3 (1962), pp. 1065–1076.

[9] B.W. Silverman. *Density Estimation for Statistics and Data Analysis*. Monographs on Statistics and Applied Probablity. Springer-Science+Business Media, B.V., 1986.

[10] M.H.F. Wilkinson and B.C. Meijer. “DATAPLOT: A Graphical Display Package for Bacterial Morphometry and Fluorimetry Data”. In: *Computer Methods and Programs in Biomedicine* 47.1 (1995), pp. 35–49.