# Shape-Adaptive Kernel Density Estimation

L.E.N. Baakman

October 4, 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.

Say something about high-anisotropy

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:

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

In this equation $K_{\mathcal{G}}$ is used to represent a Gaussian kernel, and $\alpha$ is a multiplicative constant. The values of both $\alpha$ 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 $\gamma_i$ 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 $\beta$ 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 $\beta$. 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 $\beta$ 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 global 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:

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

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 $\Sigma$, the unbiased covariance matrix of the local neighborhood $C_{\mathbf{x}_i}$.

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

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

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

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 $\Sigma$ is nonsingular we also need $k > d$, therefore

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

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 our implementation of the 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 Experiments

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

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

where $\lambda_1, \cdots, \lambda_d$ are the eigenvalues of the bandwidth matrix, to express how anisotropic a kernel is. We use two different types of datasets: datasets consisting of a single Gaussian distribution and noise, defined in Section 3.1 and datasets containing multiple Gaussian distributions embedded in an uniform

distribution as background, these sets are presented in Section 3.2.

### 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 and nor it should do much steering. 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.

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. Furthermore how well the shape of the kernels is adapted to the local neighborhood is investigated with their anisotropy. All two-dimensional 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, we have ensured that the domain 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 the two-dimensional plots illustrates the line all points would lie on if a perfect estimator was used. The colors of the points in these plot correspond to the colors of the elements of the datasets in Tables 1 and 2.

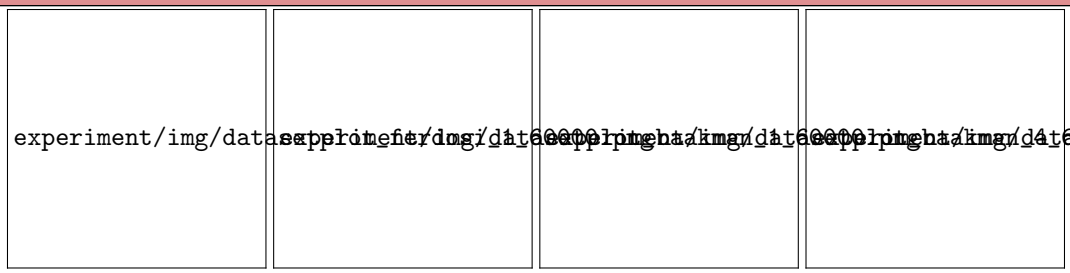
Section 4.1 presents the results of the datasets that contain a single Gaussian, in Section 4.2 the results of datasets that consist of noise and multiple Gaussian distributions are presented.

### 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 component. 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 consistently 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(c) and 3(g), and

**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.*

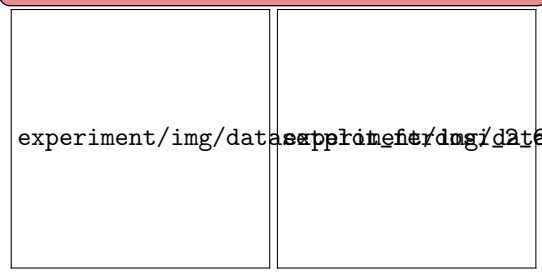
| | | Component | Number | Distribution |
|---|---|---|---|---|
| $S_1$ | • | Trivariate Gaussian | 40000 | $\mathcal{N}([50,50,50],\text{diag}(11))$ |
| | • | Uniform random background | 20000 | $\mathcal{U}([0,0,0],[100,100,100])$ |
| $S_2$ | • | Trivariate Gaussian | 40000 | $\mathcal{N}([50,50,50],\text{diag}([11^2,\sqrt{11},\sqrt{11}]))$ |
| | • | 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* $\mu$ *and covariance matrix* $\Sigma$*. 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.*

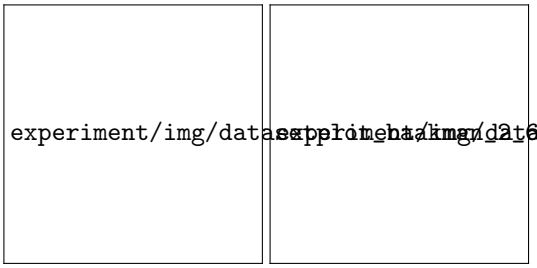
| | | Component | Number | Distribution |
|---|---|---|---|---|
| $M_1$ | • | Trivariate Gaussian 1 | 20000 | $\mathcal{N}([25,25,25],\text{diag}(5))$ |
| | • | Trivariate Gaussian 2 | 20000 | $\mathcal{N}([45,45,45],\text{diag}(11))$ |
| | • | Uniform random background | 20000 | $\mathcal{U}([0,0,0],[100,100,100])$ |
| $M_2$ | • | 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.*

**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.*

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

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

Figures 3(d) to 3(f) and 3(h), respectively. Comparing the plots associated with dataset $S_1$ we find that the shape-adaptive estimator tends to overestimate densities more than the symmetric estimator, if the Gaussian component is spherical. Based on Figures 3(c) and 3(f) the same holds for dataset $S_3$. Comparing the performance within datasets between the two components showed no marked differences 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. Furthermore as the anisotropy of the Gaussian component increases, the anisotropy of the kernels associated with points sampled from the uniform random background rises. Reviewing these statistics of the components of the datasets reveals that the increase in average anisotropy is primarily caused by an increase in anisotropy of kernels of points sampled from the Gaussian component. The mean anisotropy of the noise component stays relatively constant. Furthermore as the Gaussian component is more anisotropic the variation in anisotropy of the kernels increases.

| | | | • Gaussian | | • Background | |
|---|---|---|---|---|---|---|
| | M | SD | M | SD | M | SD |
| $S_1$ | 1.482 | 0.5207 | 1.292 | 0.1356 | 1.864 | 0.7499 |
| $S_2$ | 1.573 | 0.5531 | 1.407 | 0.2892 | 1.908 | 0.7656 |
| $S_3$ | 1.641 | 0.5857 | 1.506 | 0.4035 | 1.914 | 0.7716 |
| $S_4$ | 1.801 | 0.6982 | 1.737 | 0.6384 | 1.929 | 0.7901 |

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

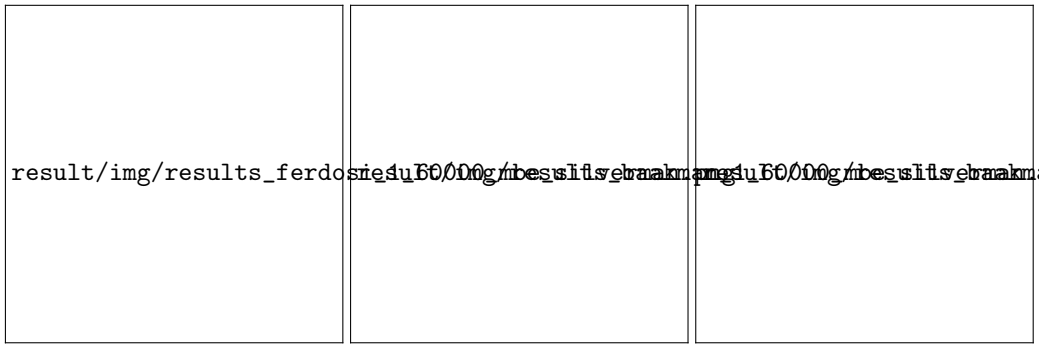
| | Estimator | |
|---|---|---|
| Set | MBE | saMBE |
| $M_1$ | $5.058 \times 10^{-8}$ | $5.050 \times 10^{-8}$ |
| $M_2$ | $5.147 \times 10^{-8}$ | $5.168 \times 10^{-8}$ |
| $M_3$ | $4.375 \times 10^{-6}$ | $4.463 \times 10^{-6}$ |
| $M_4$ | $4.189 \times 10^{-6}$ | $4.284 \times 10^{-6}$ |

**Table 5:** *Performance of the symmetric and the shape-adaptive Modified Breiman Estimator on dataset $M_1$ through $M_4$.*

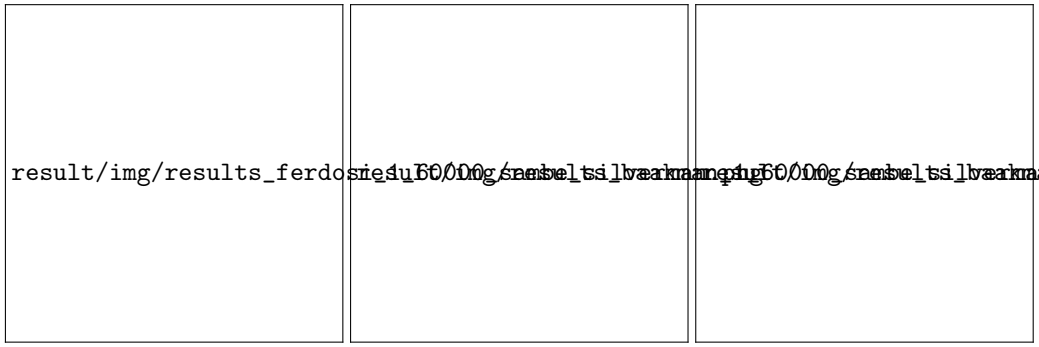
To summarize; in spite of differences in anisotropy of the used kernels we have observed very few differences between the two estimators. Using shape-adaptive kernels did not yield the expected gain in performance. We did find the expected influence of the anisotropy of the Gaussian components on the shape of the kernels. Furthermore the kernels associated with points sampled from the background are more anisotropic than those belonging to points drawn from the Gaussian distribution.

## 4.2 Datasets with Multiple Gaussians

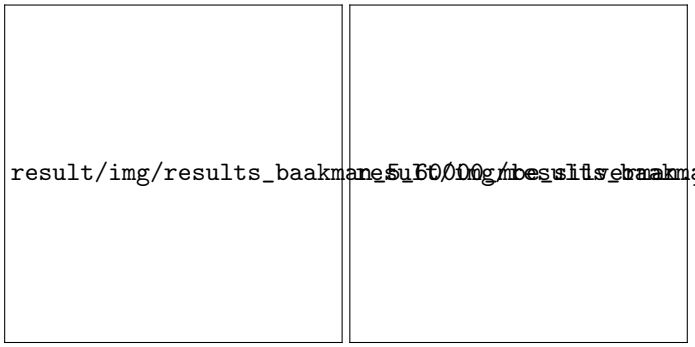
In this section we present the results of the two estimators on dataset $M_1$, $M_2$, $M_3$, and $M_4$. Based on the small differences between the mean squared errors of the estimators in Table 5 they perform com-




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




**(d)** *Set $S_1$, saMBE* **(e)** *Set $S_2$, saMBE* **(f)** *Set $S_3$, saMBE*




**(g)** *Set $S_4$, MBE* **(h)** *Set $S_4$, saMBE*

**Figure 3:** *The density as estimated by MBE and saMBE as a function of the known density of datasets $S_1$ through $S_4$.*

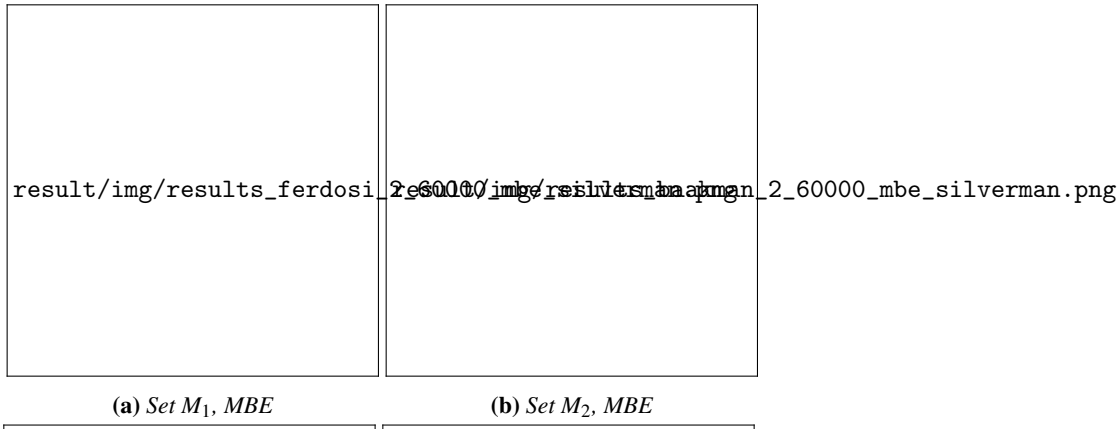
parably. Comparing the MSE between components and estimators within data sets yields no differences. However within data sets the differences in mean squared errors between components are quite large. Within dataset $M_1$ and $M_2$ both estimators perform significantly better on the component with higher values on the diagonal of its covariance matrix, e.g. 'Trivariate Gaussian 2'. In data set $M_3$ and $M_4$ both estimators show a negative correlation between the eigenvalues of the covariance matrix of that component and the MSE of points sampled from that component. Additionally, contrary to our expectation, the symmetric estimator performed better on dataset $M_4$, than on the symmetric set it was derived from.

Figure 4 shows the estimated density as a function of the known density for both estimators for the datasets with two Gaussian components. These plots show that irrespective of the type of kernel used the density is underestimated, more so by MBE than by saMBE. Furthermore using shape-adaptive kernels results in a larger variation in estimated densities than the use of symmetric kernels. Comparing the results of saMBE with those of MBE in Figure 5 suggest that saMBE hardly underestimates the densities of the most anisotropic component, i.e. 'Trivariate Gaussian 2', in dataset $M_1$ and $M_2$. However the difference in mean squared error of this component between MBE and saMBE is pretty small. The large difference in standard deviation of the squared error between estimators on this component suggests that the seemingly better performance of the shape-adaptive estimator is due to its higher spread of estimated densities. Furthermore the mean squared errors of the individual components of dataset $M_1$ and $M_4$ show that both estimators perform worse on components with covariance matrices that have low eigenvalues, e.g. 'Trivariate Gaussian 1' in $M_1$ and $M_4$. The plots in Figure 5 confirm the large difference in performance between datasets with two and four Gaussian components observed in Table 5. Moreover they show that both MBE and saMBE underestimate densities, especially on the points whose known density is high. In Figure 5 we also observe the larger spread of densities estimated by saMBE, compared to 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 $M_4$. The mean and standard deviation of the anisotropy of the kernels used for datasets with anisotropic components is slightly higher than for dataset $M_1$ and $M_3$. 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 component. The kernels used for dataset $M_2$ show the same positive correlation between the variation of the anisotropy of the components and the anisotropy of the associated kernels, as 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 component with the lowest value on the diagonal of its covariance matrix, '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 points drawn from that component in dataset $M_3$ and $M_4$ is $3.109 \times 10^{-15}$.

To summarize the results of the datasets with multiple Gaussian components: we have found that the differences in performance between the two estimators are very small. Although saMBE tends to underestimate densities less than MBE, the later performs slightly better on all datasets. Furthermore both estimators show a negative correlation between the values on the diagonal of the covariance matrix of the Gaussian component and their performance on that component. Regarding the anisotropy of the kernels we have found only a small increase in anisotropy between datasets with spherical kernels and dataset $M_2$ and $M_4$. Zeroing in on the components we have observed that the anisotropy of kernels associated with points drawn from the noise is strongest. Lastly a positive correlation between the mean anisotropy of the kernels and the anisotropy of the Gaussian the associated points were drawn from was found.

In conclusion for all datasets we have only 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 means of the anisotropies of the kernels associated with the points drawn from the Gaussian components showed a positive correlation with the anisotropy of the compo-



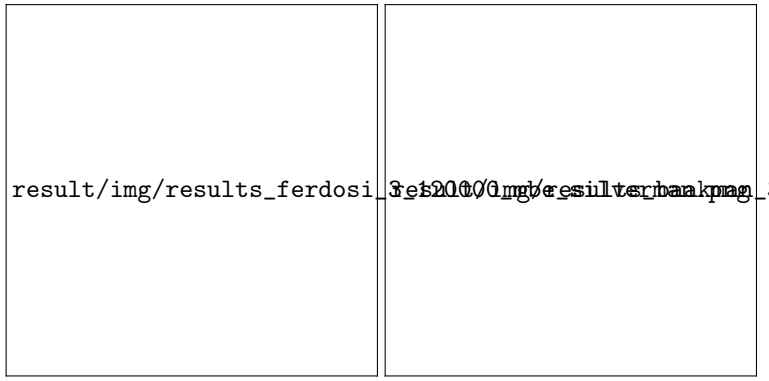

**(a)** *Set $M_1$, MBE* **(b)** *Set $M_2$, MBE*

**(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.*

| | | | • Gaussian 1 | | • Gaussian 2 | | • Gaussian 3 | | • Gaussian 4 | | • Background | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | M | SD | M | SD | M | SD | M | SD | M | SD | M | SD |
| $M_1$ | 1.504 | 0.5310 | 1.320 | 0.1750 | 1.305 | 0.1428 | | | | | 1.890 | 0.7587 |
| $M_2$ | 1.615 | 0.5702 | 1.407 | 0.2782 | 1.491 | 0.3453 | | | | | 1.948 | 0.7826 |
| $M_3$ | 1.460 | 0.5507 | 1.294 | 0.1890 | 1.266 | 0.1301 | 1.292 | 0.2103 | 1.276 | 0.1655 | 1.820 | 0.8112 |
| $M_4$ | 1.532 | 0.5714 | 1.315 | 0.2191 | 1.487 | 0.3393 | 1.292 | 0.2103 | 1.396 | 0.2851 | 1.855 | 0.8195 |

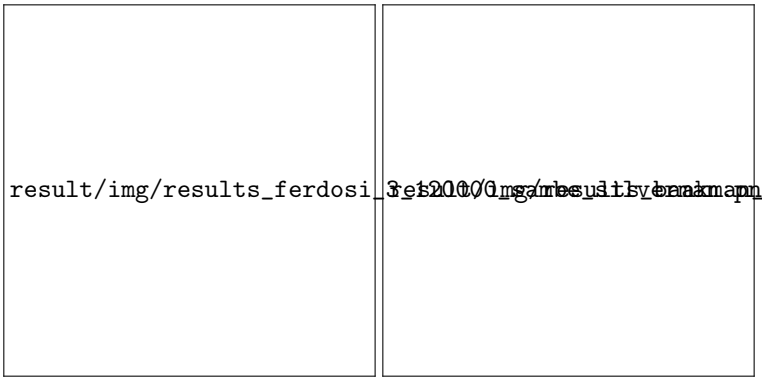
**Table 6:** *The mean (*M*) and standard deviation (*SD*) of the anisotropy of the kernels used for points from the datasets with multiple Gaussian components, for each component separately and for the full dataset.*




**(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 as a function of the true density for dataset $M_3$ and $M_4$, for both MBE and saMBE.*

nent the point was sampled from. It should be noted that this relation has not been observed between the anisotropy of the Gaussian components in the datasets with a single Gaussian and the anisotropy of the kernels of points sampled from the uniform random background.

# 5 Discussion

This section discusses the previously presented results. We first consider the lack of difference in performance between the two estimators. The section after that is concerned with the anisotropy of the kernels used by the shape-adaptive estimator. 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 difference in performance between the two estimators is minimal.

Plotting the densities 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 reveal some differences between the estimators. 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 $M_2$. Reviewing the raw data shows that saMBE underestimates densities less than MBE on points near the mean of 'Trivariate Gaussian 1'. The kernels in this neighborhood are all slightly anisotropic, which has allowed the shape-adaptive estimator to use more data points, to better approximate the local densities. Thus showing that estimating densities with shape-adaptive kernels can be 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 where the differences in estimated densities between the two estimators are largest 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 than the number of points 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, instead of the global neighborhood.


**(a)** *Dataset $M_1$*


**(b)** *Dataset $M_2$*

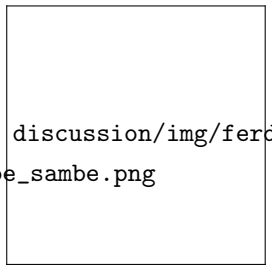
**Figure 6:** *Plots of the densities estimated by saMBE as a function of those estimated by MBE for dataset (a) $M_1$ and (b) $M_2$.*

The plots in Figure 8 emphasize the points in dataset $S_1$ and $S_2$ where the absolute error of MBE is smaller than that of saMBE. These plots show that the shape-adaptive kernels outperform symmetric kernels near the borders of the datasets. We expect that this boundary effect is due to the strong anisotropy of the local neighborhood of the points near the limits of the datasets. Consequently the domain of the shape-adaptive kernels extends less outside of the boundaries of the dataset than the domains of the symmetric kernels. This results in less underestimation 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 the density estimated by MBE. In dataset $S_1$ the two estimators give a different result on all points. In dataset $S_2$ the estimators agree on the density of 13.3 % of the points, this increases to 35.4 % in dataset $S_4$. As the Gaussian component becomes more anisotropic the number

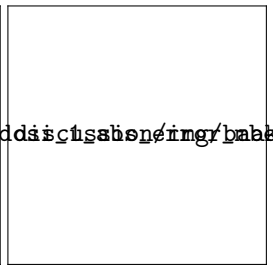

**(a)** *Dataset* $M_3$


**(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)* $M_4$*.*

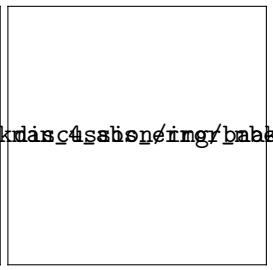


**(a)** *Dataset* $S_1$



**(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.*

of points whose local neighborhood consists only of uniform noise increases. On average the covariance matrix of neighborhoods that contain primarily points sampled from the noise component should be scalar. Consequently as the anisotropy of the Gaussian component increases more shape-adaptive kernels take on a shape that is near-symmetric. This results in points were both estimators give the same approximated density.

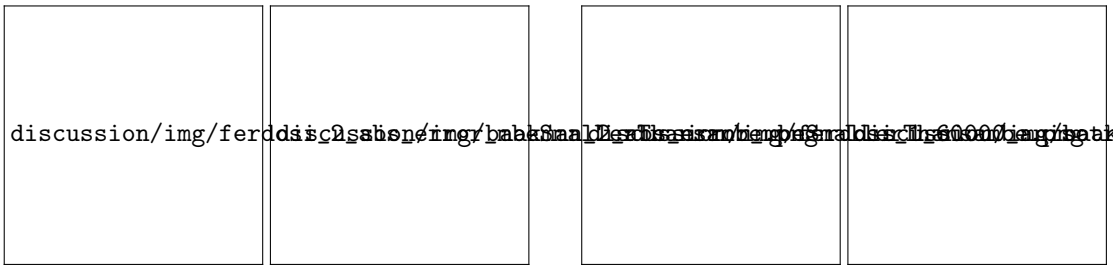
The points where using fixed-shape kernels results in a smaller error in datasets $M_1$ through $M_4$ are emphasized in Figure 9. We contribute the boundary effect in these datasets to the same cause as the boundary effect in the datasets with a single Gaussian component. Interestingly the points in 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 the smaller distance between the means of the Gaussian components and the range of the uniform random background in $M_3$ and $M_4$. To test this we define dataset $M'_3$, which replaces the uniform random background of $M_3$ with $\mathcal{U}\big([-20,-20,-20],[120,120,120]\big)$. We adjust the number of points sampled from this component to ensure that its density is equal to that of the noise component of $M_3$. The overall mean squared error of both estimators is slightly smaller for dataset $M'_3$ than 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. As the shape defined by the emphasized points is now a cubical instead of spherical.

To conclude we have found that shape-adaptive kernels definitely improve performance in some cases, i.e. 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 shows that generally, using fixed-shape kernels is slightly advantageous.

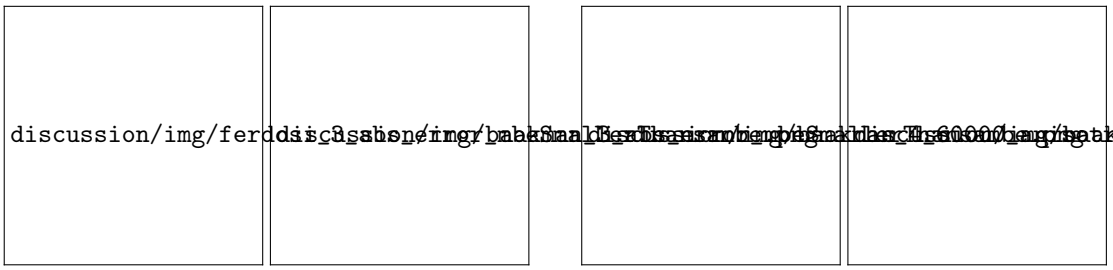
## 5.2 Kernel Anisotropy

In Section 4 some small differences between the anisotropies of the different kernels were observed. This section attempts to find an explanation for this effect.

Figure 11 shows the datasets with a single Gaus-



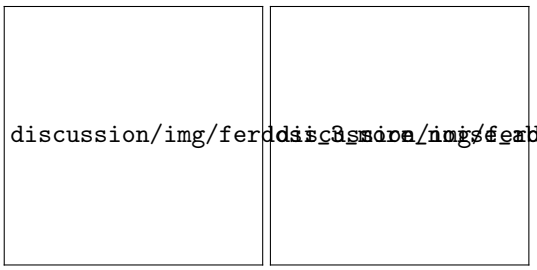
**(a)** *Dataset* $M_1$ **(b)** *Dataset* $M_2$ **(a)** $S_1$ *[2.784, 4.877]* **(b)** $S_2$ *[2.956, 4.877]*



**(c)** *Dataset* $M_3$ **(d)** *Dataset* $M_4$ **(c)** $S_3$ *[3.106, 5.987]* **(d)** $S_4$ *[3.546, 8.192]*

**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.*

**Figure 11:** *Scatter plot of data set (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 sampled from. The range of the anisotropy of the kernels of the emphasized points is shown below each plot.*

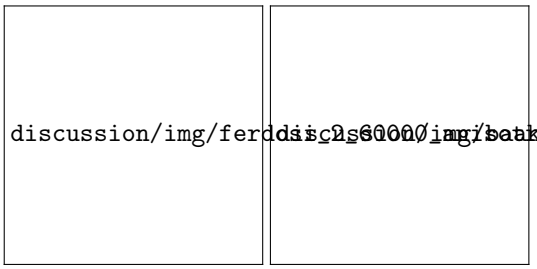


**(a)** *Absolute Error* **(b)** *Anisotropy*

**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.*

sian with points whose anisotropy lie in the $90^{\text{th}}$ percentile emphasized. Hardly any differences are visible between the plots of dataset $S_1$ and $S_2$. 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. This shows that the kernels in data set $S_2$ are influenced by the increase in anisotropy compared to the anisotropy of the Gaussian component used in data set $S_1$. Furthermore a shell of points sampled from the noise with kernels whose anisotropy is relatively high surrounds the Gaussian component. It is quite likely that the shape of these kernels is influenced by that 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 low and thus possibly insufficient to represent the shape of the Gaussian. In dataset $S_3$ and $S_4$ a larger percentage of the points with a kernel whose anisotropy is relatively high is sampled from the Gaussian component, 21.9 % and 42.1 %, respectively. Therefore we tentatively conclude that as the anisotropy of the Gaussian component increases, the anisotropy of the kernels of the points near that component increases as well.

Figure 12 emphasizes the points with the most anisotropic kernels in data set $M_1$ through $M_4$. In the plot associated with data set $M_1$ we observe the same shells of points with highly anisotropic kernels around the Gaussian components as in data set $S_1$. Another similarity between these two data sets is that very few points sampled from the Gaussian component have a kernel with high anisotropy; 1.45 % and 0.117 % of the points with a high



**(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 data set (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 is shown below each plot.*

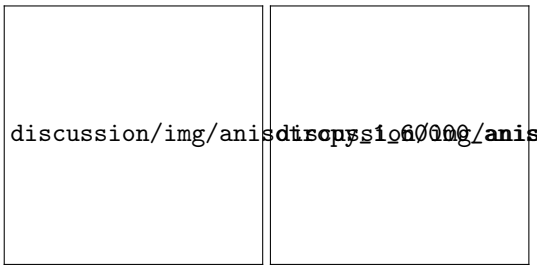
anisotropy are sampled from the first and second Gaussian component, respectively. We contribute the difference in the number of highly anisotropic kernels associated with data points sampled from the two Gaussian components to the difference in the volume of the their eigenspheres. Comparing Figures 12(a) and 12(b) we find that the increase in anisotropy of the components 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, 'Trivariate Gaussian 1' and 3, are associated with respectively 1.40 % and 2.22 % of the highly anisotropic kernels. Whereas no point sampled from 'Trivariate Gaussian 2' or 4 has a kernel with high anisotropy. Comparing the mean kernel anisotropy in Table 6 we find that relative to the other Gaussian components in $M_3$, 'Trivariate Gaussian 1' and 3 have a relatively low mean anisotropy. However their standard deviations are relatively high, suggesting that in these components some kernels are extremely anisotropic, whereas others are near isotropic. We contribute this difference within the points sampled from these Gaussian components to differences in the physical densities of the neighborhoods around the means, as we 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 the volume of their eigenspheres is relatively low and secondly they are placed near the boundaries of the dataset. Figure 10(b) shows that the distance to the limits of the background does 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, even though they are farther away from the boundary of the dataset. The first explanation fits with the observations from Section 4 that components with eigensphere with a larger volume, have kernels that 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 other components.

| | Estimator | |
|---|---|---|
| | MBE | saMBE |
| $A_1$ | $1.240 \times 10^{-6}$ | $1.297 \times 10^{-6}$ |
| $A_2$ | $1.713 \times 10^{-8}$ | $1.941 \times 10^{-8}$ |

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

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 matrix detecting fine, random structures in the noise, that give the impression of anisotropy in the data where there is none.

In Section 4 we observed that both the volume of the eigensphere and the anisotropy of the Gaussian component influenced the anisotropy of the kernels. To test which factor is responsible for the difference in anisotropy we introduce two new datasets: $A_1$ and $A_2$. To create dataset $A_1$ the covariance matrix used in $S_1$ was replaced by $\mathrm{diag}([4,3,1])$, dataset $A_2$ is the same as $A_1$ but uses $3 \cdot \mathrm{diag}([4,3,1]$. Consequently the Gaussian components of $A_1$ and $A_2$ have the same anisotropy, but the volumes of the eigenspheres of their covariance matrices differ. 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 data set with higher values on the diagonal of the covariance matrix of the Gaussian component. Figure 13 illustrates the




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

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

influence of the volume of the eigensphere of the Gaussian component on the anisotropy of the kernels near that component. Of the points with a highly anisotropic kernel in data set $A_1$ 6.23 % is sampled from the Gaussian component, in dataset $A_2$ this is 4.16 %. Although the difference is small, it illustrates that the volume of the eigensphere of a component influences the anisotropy of the kernels.

In closing we have found that near Gaussian components with a high volume eigensphere 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 increasing the size of the local neighborhood will decrease the number of detected fine structures. Which results in more isotropic kernels for points that lie in the uniform background. Furthermore increasing $k$ might also allow the kernels to adapt their shape to nearby Gaussian components whose eigensphere has a high volume. Multiplying the $k$ computed in Equation (9) by ten, resulted in saMBE outperforming MBE on dataset $S_4$. Furthermore the increased $k$ also improved the performance of saMBE on dataset $M_3$, $M_4$ and the datasets with a single Gaussian component. Whereas the performance of saMBE on dataset $M_1$ and $M_2$ suffered. This shows that blindly increasing the size of the local neighborhood does not consistently improve the performance of the estimator, but that some adaptive increase of $k$ is required.

Another solution to the too anisotropic kernels in the noise might be to 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 challenge with this approach is detecting the isotropy of the local neighborhood. Since the obvious solution, the covariance matrix, has proven sensitive to fine structures within the uniform random background. One possible alternative to the covariance matrix is the Hessian matrix. A potential issue with this approach is that it uses second order derivatives. Firstly this may lead to it detecting more fine local structures in the background. Secondly the used Epanechnikov kernel would have to be replaced with a kernel that has second order differentiability.

Finally the correlation between the performance of the estimators and the volume of the eigensphere of the Gaussian component suggests that the method used to compute the local bandwidths is far from ideal. One possible solution, at the cost of increased computational complexity, might be to use the method proposed by Breiman, Meisel, and Purcell.

# 6 Conclusion

We have found that the shape-adaptive Modified Breiman Estimator gives results comparable to those of the symmetric version of the estimator. Anisotropic kernels have proven advantageous near the borders of the data sets. However this positive effect is negated by the number of points where the kernel is too isotropic, or too anisotropic for its local neighborhood. Kernels that are too anisotropic for their neighborhood occur mostly in the uniform random noise, due to the local neighborhood being sensitive to spurious, fine structures in the background, where the data is isotropic. Too isotropic kernels occur mostly with points near the mean of Gaussian components whose covariance matrix has a large eigensphere.

Both cases show that the estimator has problems with detecting the shape of the local neighborhood. One way of addressing this problem might be to (adaptively) increase the size of the local neighborhood. Another possible improvement could be to decide how anisotropic the kernel should be, based on its local neighborhood.

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.