

# Shape-Adaptive Kernel Density Estimation

L.E.N. Baakman

September 7, 2017

## Abstract

Kernel density estimation has gained popularity in the past few years. Generally the 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 of the point whose density is estimated. We compare the performance of the shape adaptive kernels on simulated datasets with known density fields.

Results

Conclusion

## 1 Introduction

Estimating densities with kernels has been fairly popular of late; 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  $\mathbf{x}_1, \dots, \mathbf{x}_N$ , 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 the different symmetric kernel density estimation methods.

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

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

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 fixed-width methods is that the peakedness 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 of the fixed-width methods by allowing the width of the kernel to vary per data point. For example the estimator introduced by Breiman, Meisel, and Purcell 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:

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

In this equation  $K_G$  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 Equation (1) with (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

$$\gamma_i = \left( \frac{\tilde{f}(\mathbf{x}_i)}{\text{GM}(\tilde{f}(\mathbf{x}_0), \dots, \tilde{f}(\mathbf{x}_N))} \right)^{-\beta}, \quad (3)$$

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

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

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  $K_E$ , i.e.  $\sqrt{5}$ . There are two advantages to using this kernel, firstly it is computationally much simpler than the Gaussian kernel, in part due to its finite base, and secondly it is optimal in the sense of the Mean Integrated Square Error (MISE) [3]. One downside of this kernel is that it is not continuously differentiable. This is irrelevant when computing the pilot densities, however for the final densities it is a trade off between a continuously differentiable density estimate and a density estimator that has a low computational complexity. Wilkinson and Meijer [10] compute the global bandwidth according to

$$h = \sigma \cdot N^{-1/(d+4)} \left( \frac{8(d+4) \cdot (2\sqrt{\pi})^d}{c_d} \right)^{\frac{1}{d+4}}, \quad (6)$$

where  $\sigma$  represents the square root of the average of the variances of the different dimensions. The final densities are estimated with Equation (4), using the general and local bandwidths estimated with Equation (6) and (3), respectively.

Ferdosi et al. [4] consider the application of density estimation on large datasets, i.e. sets with more than 50 000 points with the dimension of the data points ranging 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 with

$$h_l = \frac{P_{80}(l) - P_{20}(l)}{\log N}, \quad l = 1, \dots, d, \quad (7)$$

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

Although the widths of the kernels of the discussed adaptive-width methods are sensitive to the data, the shapes of the kernels depend only on its definition. 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 conclusion 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 also referred to as the shape-adaptive Modified Breiman Estimator (saMBE). For its lower computational complexity we use the method introduced by Ferdosi et al. [4], defined in Equation (7), to compute the general bandwidth. Pilot densities are computed according to Equation (1), with an Epanechnikov kernel. Since using  $\beta = 1/2$  in Equation (3) results in a final density approximation with a lower mean squared error than using  $\beta = 1/d$ . We use the first when computing the local bandwidths. The final density estimate is computed according to:

$$\hat{f}(\mathbf{x}) = \frac{1}{N} \sum_{i=1}^N \frac{1}{\det(\mathbf{H}_i)} K_{\mathcal{E}}(\mathbf{H}_i^{-1}(\mathbf{x} - \mathbf{x}_i)). \quad (8)$$

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 (8) 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

$$s = h \cdot \gamma_i \left( \prod_{l=1}^d \lambda_l \right)^{-\frac{1}{d}} \quad (9)$$

where  $\lambda_1, \dots, \lambda_d$  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

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

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 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 basis 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 (9) 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 can recover simple density distributions in comparison to an existing method. We distinguish two types of datasets: 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.

To quantify the performance of the estimators we use the mean squared error (MSE):

$$\text{MSE}(\hat{f}(\bullet)) = \frac{1}{N} \sum_{j=1}^N (\hat{f}(\mathbf{x}_j) - f(\mathbf{x}_j))^2.$$

### 3.1 Datasets with a Single Gaussian

Figure 1 shows a scatter plot representation of the datasets containing a single Gaussian distribution 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. The Gaussian component in  $S_4$  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 advantage 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 set  $M_1$ . Dataset  $M_3$  embeds four non-overlapping Gaussians, with eigenspheres with notably differencing radii, in the uniform random background. The last dataset,  $M_4$ , is a variation on  $M_3$ , created with method that was used for the definition of dataset  $S_2$  from  $S_1$ .

We expect to find hardly any difference in performance between the classifiers 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. 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  $x = x$ . The colors of the points in these plot correspond to the colors of the elements of the datasets in Tables 1 and 2.

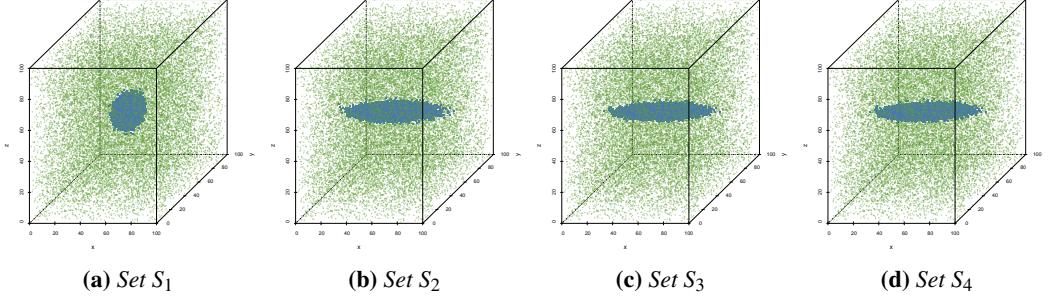
Section 4.1 presents the results of the datasets that contain a single Gaussian, in Section 4.2 the results of the datasets that consist of noise and multiple Gaussian distributions are presented.

### 4.1 Datasets with a Single Gaussian

This section compares the performance of the Modified Breiman Estimator and a shape-adaptive variant on datasets that contain one Gaussian, i.e. dataset  $S_1$ ,  $S_2$ ,  $S_3$ , and  $S_4$ . 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 result in Figure 3 where hardly any difference is visible between Figures 3a to 3d and Figures 3e to 3h, respectively.

Comparing Figure 3a with Figure 3e we find hardly any difference between the results of the two estimators, saMBE overshoots some densities more than MBE, but otherwise the results seem identical, which fits with the small difference in mean square error. Reviewing the mean squared errors

**Before Final Version:** Remove ticks and labels.



**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.

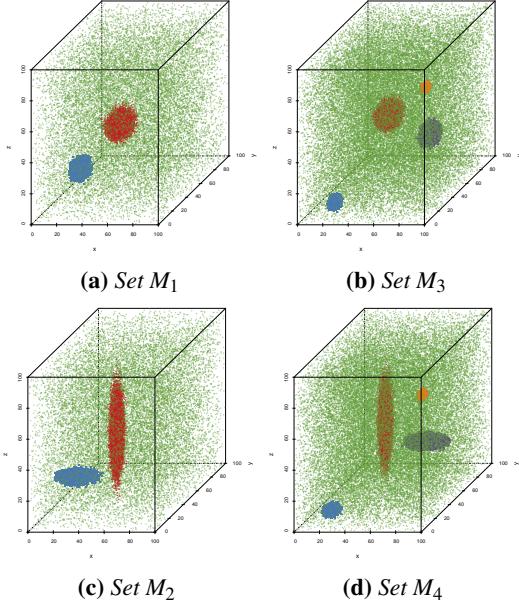
Set	Component	Number	Distribution
$S_1$	■ Trivariate Gaussian	$4.0 \times 10^4$	$\mathcal{N}([50, 50, 50], \text{diag}(11))$
	● Uniform random background	$2.0 \times 10^4$	$\mathcal{U}([0, 0, 0], [100, 100, 100])$
$S_2$	■ Trivariate Gaussian	$4.0 \times 10^4$	$\mathcal{N}([50, 50, 50], \text{diag}([11, \sqrt{11}, \sqrt{11}]))$
	● Uniform random background	$2.0 \times 10^4$	$\mathcal{U}([0, 0, 0], [100, 100, 100])$
$S_3$	■ Trivariate Gaussian	$4.0 \times 10^4$	$\mathcal{N}([50, 50, 50], \text{diag}([11, 2 * \sqrt{11}, 1/2\sqrt{11}]))$
	● Uniform random background	$2.0 \times 10^4$	$\mathcal{U}([0, 0, 0], [100, 100, 100])$
$S_4$	■ Trivariate Gaussian	$4.0 \times 10^4$	$\mathcal{N}([50, 50, 50], \text{diag}([11^2, 11, 1]))$
	● Uniform random background	$2.0 \times 10^4$	$\mathcal{U}([0, 0, 0], [100, 100, 100])$

**Table 1:** The datasets containing a single Gaussian distribution embed 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  $x_1, \dots, x_d$  on the diagonal is represented as  $\text{diag}([x_1, \dots, x_d])$ , 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.

Set	Component	Number	Distribution
$M_1$	■ Trivariate Gaussian 1	$2.0 \times 10^4$	$\mathcal{N}([25, 25, 25], \text{diag}(5))$
	▲ Trivariate Gaussian 2	$2.0 \times 10^4$	$\mathcal{N}([45, 45, 45], \text{diag}(11))$
	● Uniform random background	$2.0 \times 10^4$	$\mathcal{U}([0, 0, 0], [100, 100, 100])$
$M_2$	■ Trivariate Gaussian 1	$2.0 \times 10^4$	$\mathcal{N}([25, 25, 25], \text{diag}([5^2, \sqrt{5}, \sqrt{5}]))$
	▲ Trivariate Gaussian 2	$2.0 \times 10^4$	$\mathcal{N}([45, 45, 45], \text{diag}([\sqrt{11}, \sqrt{11}, 11^2]))$
	● Uniform random background	$2.0 \times 10^4$	$\mathcal{U}([0, 0, 0], [100, 100, 100])$
$M_3$	■ Trivariate Gaussian 1	$2.0 \times 10^4$	$\mathcal{N}([24, 10, 10], \text{diag}(2))$
	▲ Trivariate Gaussian 2	$2.0 \times 10^4$	$\mathcal{N}([33, 70, 40], \text{diag}(10))$
	◆ Trivariate Gaussian 3	$2.0 \times 10^4$	$\mathcal{N}([90, 20, 80], \text{diag}(1))$
	* Trivariate Gaussian 4	$2.0 \times 10^4$	$\mathcal{N}([60, 80, 23], \text{diag}(5))$
	● Uniform random background	$4.0 \times 10^4$	$\mathcal{U}([0, 0, 0], [100, 100, 100])$
$M_4$	■ Trivariate Gaussian 1	$2.0 \times 10^4$	$\mathcal{N}([24, 10, 10], \text{diag}([4, \sqrt{2}, \sqrt{2}]))$
	▲ Trivariate Gaussian 2	$2.0 \times 10^4$	$\mathcal{N}([33, 70, 40], \text{diag}([\sqrt{10}, \sqrt{10}, 100]))$
	◆ Trivariate Gaussian 3	$2.0 \times 10^4$	$\mathcal{N}([90, 20, 80], \text{diag}(1))$
	* Trivariate Gaussian 4	$2.0 \times 10^4$	$\mathcal{N}([60, 80, 23], \text{diag}([25, \sqrt{5}, \sqrt{5}]))$
	● Uniform random background	$4.0 \times 10^4$	$\mathcal{U}([0, 0, 0], [100, 100, 100])$

**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.



**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.

Set	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.

of the components of this dataset we find that MBE slightly outperforms saMBE on both datasets.

Figures 3b and 3f confirm what the MSE already told us, there is hardly any difference in performance between the two estimators. There is no difference within the estimators between components.

Based on the differences between Figures 3c and 3g we can at best conclude that the shape-adaptive estimator overestimates the densities slightly more than the fixed-shape estimator. The differences between estimators within components are not significantly large.

Figures 3d and 3h supports the MSE in that there hardly any difference in estimated densities between the two estimators on dataset  $S_4$ . Furthermore

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

within components the differences between the estimators are also negligible.

We have found no direct correlation between the length of largest minor axis of the eigenellipse and the performance of the estimators, e.g. the MSE of  $S_3$  is lower than that of  $S_2$ . Comparing the performance of both estimators on between dataset  $S_1$  and  $S_4$  suggest that lengthening the major axes has a negative influence on the performance of the estimator.

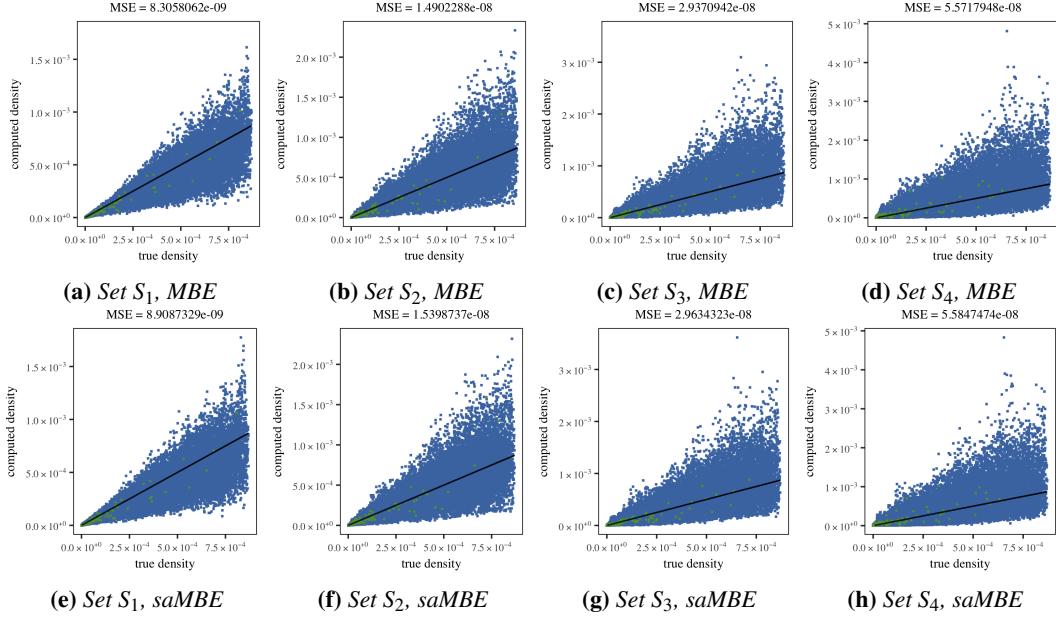
## 4.2 Datasets with Multiple Gaussians

In this section we present the results of the two estimators on dataset  $M_1, M_2, M_3, M_4$ .

Comparing Figure 4a with Figure 4c we find that both estimators underestimate the density and that the densities estimated by the saMBE are spread out more than those estimated by MBE. In spite of this the difference in mean squared error between the two estimators is small enough to be insignificant. The same holds for the mean squared error of the individual components.

Figures 4b and 4d show the same general trend as Figures 4a and 4c: both estimators underestimate, the shape-adaptive estimator less so than the symmetric estimator, but the differences between the two estimators are small. The differences in MSE within the difference components between the estimators are negligible. Comparing the performance of the estimators between datasets  $M_1$  and  $M_2$  we find that the performance of both estimators hardly suffers from the elongation of the Gaussians.

Figures 5a and 5c clearly show that both estimators significantly underestimate the true density, saMBE more so than MBE. Comparing the mean squared error of the different components we find that both estimators performed worst on the densest component, and best on the component with the highest value on the diagonal of its covariance. There is no significant difference between the esti-



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

mators within the different components.

Figures 5b and 5d shows the same underestimating of densities as the plot of the plots associated with datasets  $M_3$ . Compared to densities estimated for that dataset the range of densities estimated by both estimators for dataset  $M_4$  is greater. The difference in mean squared error within both the complete set and its components between the two estimators is negligible. Contrary to our expectations both estimators perform better on the elongated dataset, i.e.  $M_3$ , than on the spherical set.

In general we have found that the number of Gaussian distributions embed in the noise negatively influences the performance of both estimators. Furthermore the denser a Gaussian distribution is, the more difficulty the estimators have with correctly approximating the density of the points sampled from it.

By comparing the mean squared error of the different components of the datasets we have also found that both estimators are better at estimating the density of points sample from uniform random noise than points sampled from a Gaussian distribution.

## 5 Discussion

This section first discusses the results of the density estimation of dataset with a single Gaussian, in Sec-

tion 5.1, Section 5.2 discusses the results of applying the density estimators on datasets with multiple Gaussian components.

### 5.1 Datasets with a Single Gaussian

The difference in results between the Modified Breiman Estimator and its shape adaptive variant on dataset  $S_1$ ,  $S_2$ ,  $S_3$ ,  $S_4$  raise several questions. This section attempts to answer them.

Why are the differences between mbe and sambe so small? What do the scaling factors indicate?

### 5.2 Datasets with Multiple Gaussians

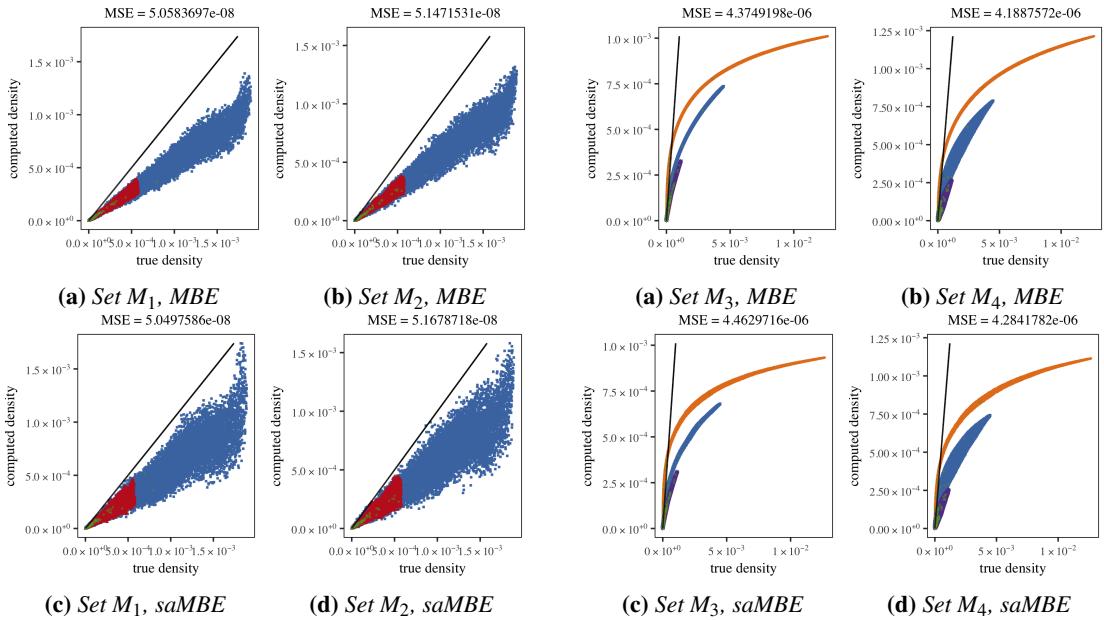
What does this section do?

Waarom is de error op denser Gaussians hoger dan op minder dense gaussians? Local bandwidth die niet helemaal goed zit?

SAMBE zou visueel een kleinere error moeten hebben, maar gemiddeld komt MBE beter uit de bus? Op welke punten doet SAMBE het significant beter / slechter dan MBE?

Waarom is de error op baakman3 kleiner dan op ferdosi 3?

General Discussion



**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.

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

## 6 Conclusion

Further research: If the k-nearest neighbor is more than some distance  $x$  away from  $x_i$  the identity matrix, or better, move between the fixed-width kernel and the shape-adaptive kernel based on the distance of the k-nearest neighbor.

Determine the optimal value of k.

## 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 Sjölund. “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 Probability. Springer-Science+Business Media, B.V., 1986.

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