Shape-Adaptive Kernel Density Estimation

L.E.N. Baakman

September 13, 2017

Abstract

Kernel density estimation has gained popularity in the past few years. 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. We also found some differences in performance dependent on the distance to the mean of Gaussian distributions with low values on the diagonal of the covariance matrix. In conclusion shape-adaptive kernels are a promising idea that warrants further research.

1 Introduction

2 Method

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 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.

3.1 Datasets with a Single Gaussian

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

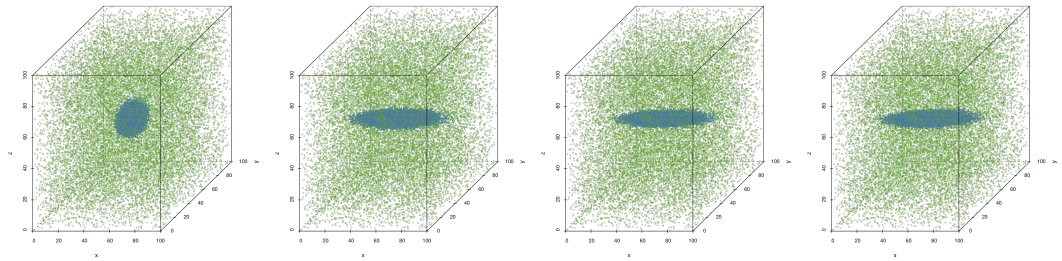
The Gaussian components of these datasets progress from a sphere, i.e. dataset S_1, to an increasingly more elongated ellipsoid. This makes it possible to investigate the influence of how strongly elongated the distribution is, on the density estimate. The first dataset is a simple spherical Gaussian 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

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.*

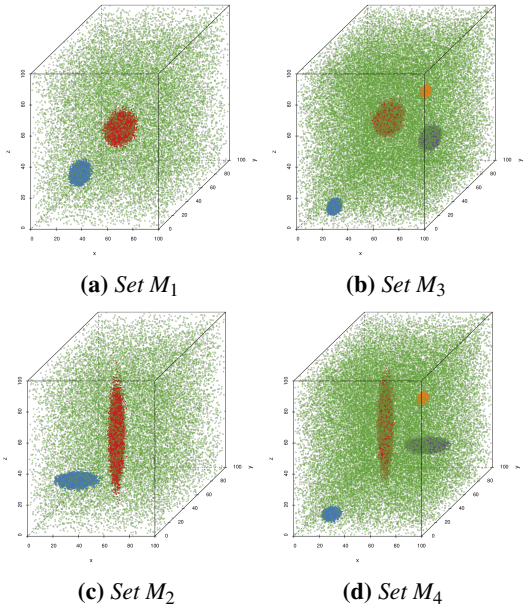
Set		Component	Number	Distribution
S_1	•	Trivariate Gaussian	4.0×10^4	$\mathcal{N}([50,50,50],\text{diag}(11))$
	•	Uniform random background	2.0×10^4	$\mathcal{U}([0,0,0],[100,100,100])$
S_2	•	Trivariate Gaussian	4.0×10^4	$\mathcal{N}([50,50,50],\text{diag}([11,\sqrt{11},\sqrt{11}]))$
	•	Uniform random background	2.0×10^4	$\mathcal{U}([0,0,0],[100,100,100])$
S_3	•	Trivariate Gaussian	4.0×10^4	$\mathcal{N}([50,50,50],\text{diag}([11,2*\sqrt{11},{}^1\!/_2\sqrt{11}]))$
	•	Uniform random background	2.0×10^4	$\mathcal{U}([0,0,0],[100,100,100])$
S_4	•	Trivariate Gaussian	4.0×10^4	$\mathcal{N}([50,50,50],\text{diag}([11^2,11,1]))$
	•	Uniform random background	2.0×10^4	$\mathcal{U}([0,0,0],[100,100,100])$

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.*

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

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.



(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.*

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. [1] 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.

	Estimator	
Set	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}

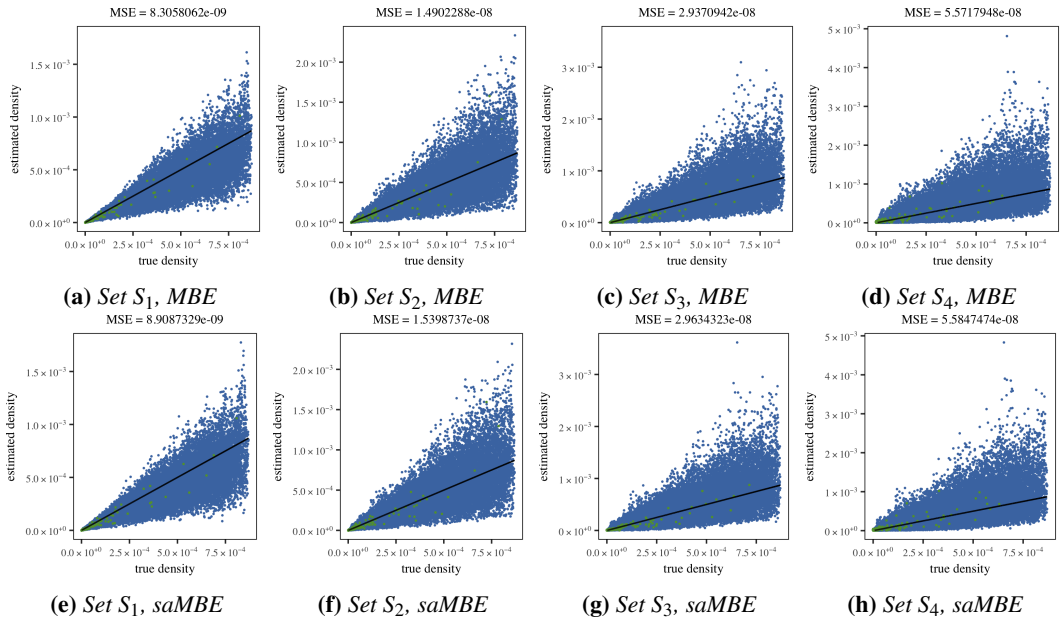
Table 3: *Performance of the Modified Breiman Estimator with fixed-shaped and shape-adaptive kernels on the datasets with a single Gaussian.*

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 Figure 3(a) with Figure 3(e) 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 of the components of this dataset we find that MBE



(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.*

slightly outperforms saMBE on both components.

Figures 3(b) and 3(f) confirm the conclusion drawn from the MSE, there is hardly any difference in performance between the two estimators. Nor is there any difference between them within components.

Based on the differences between Figures 3(c) and 3(g) 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 significant.

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

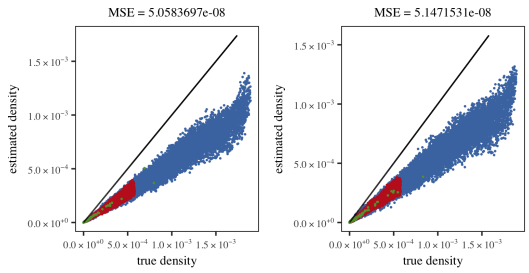
Comparing Figure 4(a) with Figure 4(c) we find

	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}

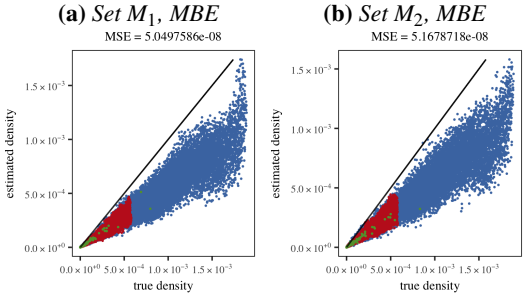
Table 4: *Performance of the symmetric and the shape-adaptive Modified Breiman Estimator on the datasets containing multiple Gaussian distributions.*

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 4(b) and 4(d) show the same general trend as Figures 4(a) and 4(c): 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.

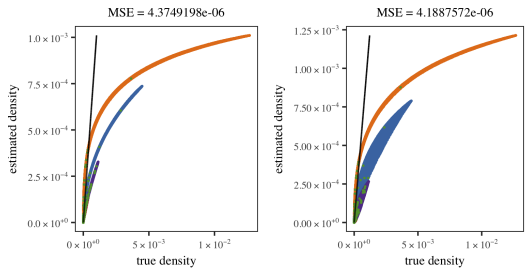


(a) *Set M_1, MBE* **(b)** *Set M_2, MBE*

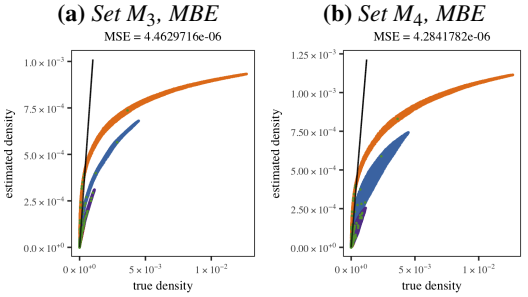


(c) *Set M_1, saMBE* **(d)** *Set M_2, saMBE*

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.*

Figures 5(a) and 5(c) 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 estimators within the different components.

Figures 5(b) and 5(d) 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

6 Conclusion

References

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