Model of Estimation of Distribution Density for Large Statistical Datasets

Nataliya Boyko, Yaroslav Rohan and Svitlana Honchar

Lviv Polytechnic National University, Profesorska Street 1, Lviv, 79013, Ukraine

Abstract

The distribution density function is a fundamental concept in statistics that provides a natural description of the distribution for any random variable and allows the identification of the corresponding probabilities by ratio. In this paper, we attempt to determine the 3D density distribution of galaxies in large surveys (such as the SDSS) in order to study the effect of the environment on galaxy evolution. We will also explore finding structures in large spaces, such as six-dimensional phase space, or even larger spaces in large astronomical databases (such as the SDSS database itself). This is why we are interested in accurate and efficient density estimators for astronomical data sets in several dimensions.

Keywords

Distribution density, statistics, k-nearest neighbors, adaptive Gaussian kernel density estimation, Sloan Digital Sky Survey, Smoothed Particle Hydrodynamics.

1. Introduction

Estimating density in datasets is a critical first step in progress in many areas of astronomy. For example, the galactic environment obviously plays an important role in its evolution, as observed in the ratio of color density and color concentration density. It is important to estimate the local density of galaxies for these relationships [1, 15, 20].

As another example, the reconstruction of a large-scale structure of the universe requires a proper assessment of the space field density. Even modeling requires density estimation: SPH is a method of creating a simulated astronomical structure by using astrophysical fluid dynamic calculations, which uses nuclear density estimation to solve hydrodynamic equations [3, 6, 8].

Density estimation is required not only for the analysis of spatial region structures, but also for structures in other spaces, such as the search for connected structures in six-dimensional phase space when modeling space structure formation or in three-dimensional phase space projections in satellite galaxy growth simulations [18, 22].

This paper examines the performance of four distribution density function evaluation methods [5, 11, 21]:

- k-nearest neighbors (KNN);
- 3D-implementation of adaptive estimation of Gaussian nucleus density called DEDICA;
- a modified version of the adaptive Braiman core density estimation, called the modified Braiman estimation (MBE);
- Delaunay tessellation field estimator (DTFE).

The first method is well known among astronomers and involves determining the density by counting the number of nearby neighbors to the issue under consideration [2, 16]. This method is commonly used in studies of the relationship between the environment and the properties of the galaxy. The second and third methods are both adaptive nucleus density estimators, where a nucleus whose size

MoMLeT+DS 2022: 4th International Workshop on Modern Machine Learning Technologies and Data Science, November, 25-26, 2022, Leiden-Lviv, The Netherlands-Ukraine
EMAIL: nataliya.i.boyko@lpnu.ua (N. Boyko); yaroslav.rohan.knm.2019@lpnu.ua (Ya. Rohan); svitlana.y.honchar@lpnu.ua (S. Honchar)
ORCID: 0000-0002-6962-9363 (N. Boyko); 0000-0002-9527-3145 (Ya. Rohan); 0000-0002-7420-962X (S. Honchar)

adapts to local conditions (usually isotropic), depending on certain criteria established before or iteratively during process estimation, is used to smooth the point distribution so that typically density can be estimated. The fourth method, like the first, uses the position of neighbors to estimate the local densities.

2. Overview of methods

2.1. Study of the object of study

The purpose of density estimation is to approximate the true probability density function (pdf) of a random process from the observed data set. There are two main families of density estimators: parametric and nonparametric. In parametric methods, the type of distribution (uniform, normal, Poisson, etc.) must be known (or guessed) in advance, while nonparametric methods do not require this information. The methods considered in this study belong to the second type [4, 10, 19].

First, we must distinguish between different types of design density. Starting from the input data set, which consists of a list of positions of points $r_i \in \mathbb{R}_d$, $i = 1, ..., N$ in the d-dimensional spatial region, we define two types of probability density as [22, 26]:

1. Point probabilities: probability densities $\acute{p}(r_i)$ in the initial positions of the points r_i;
2. Probability density field: probability density $\acute{p}(r)$ at arbitrary points in the spatial region $\mathbb{R}_d$. We often estimate the field density at Cartesian grid points, so we also talk about grid density.

In addition, the probability density must be converted to the physical density when comparing galaxies. This is because the parameter of interest is the quantification of the environment of individual galaxies, not the probability of finding the galaxy in a particular position. The latter is calculated using density estimators and can be converted to the first by multiplying by N, namely [6, 12, 27]:

1. Density of point numbers: $\acute{n}(r_i) = N\acute{p}(r_i)$;
2. Density field of numbers: $\acute{n}(r) = N\acute{p}(r)$.

2.2. Method k nearest neighbors

The KNN estimator is well known in astronomy, and its principle is to focus a spherical window on each point r and allow it to grow until it captures k samples (k nearest neighbors r). Then the estimate of the density KNN for a data set with N data points is determined in any $r \in \mathbb{R}^d$ (Formula 1) [7, 13, 25]:

$$\acute{p}(r) = \frac{1}{N}\frac{k}{V_d \delta_k^d}, \quad (1)$$

where δ_k is the distance of the k-th nearest neighbor from r and V_d is the volume of a unit sphere in d-dimensional space. The KNN approach uses a different window size for each point, so it adapts to the local density: when the density is high near r, the window will be small; but when the local density is low, the window will grow to a larger size.

The KNN approach can be a good solution for finding the "best" window size. However, this method suffers from a number of disadvantages. The obtained density estimate is not a proper probability density, because its integral differs in all spaces, and the tails fall out extremely slowly. The density field is very "prickly", and the calculated density is far from zero, even in large regions where no samples are observed due to heavy tails. In addition, it leads to gaps, even when the main distributions are continuous [14, 17, 23].

In astronomical works, it is typical that the sampling point is not considered its own neighbor. This is a conceptual problem because the point density will then disagree with the field density at the location of the sample point. In this paper, we take the sampling point as its own first neighbor, as in Silverman (1986), and use the average value of the estimated KNN densities with $k = 5$ and $k = 6$ when calculating either the density of the point or the grid. This is not exactly equivalent to the average density $k = 4$ and $k = 5$ KNN used in many astronomical works (for example, Baldry et al. 2006). While V in the denominator of equation (1) would be equal, k in the denominator is one higher by Silverman's definition [9, 21, 27].

2.3. Epanechnikov adaptive estimation of nucleus density

Braiman (1977) described the case of an adaptive (Gaussian) nucleus. This method begins by calculating the distance δ_i, k to the *k*-th nearest neighbor of each data point located on r_i, similar to the density estimator KNN. Instead of using this distance to calculate the KNN density estimate, it uses it to control the local core size (also known as bandwidth) in the adaptive core density estimate or in the Parzen estimate. To sample DN from N points with position vectors $r_i \in \mathbb{R}d$ ($i = 1, ..., N$) and the nucleus $K(r)$, the adaptive density of the nucleus $\acute{p}(r)$ is estimated by (Formula2) [17, 24, 28]:

$$\acute{p}(r) = \frac{1}{N}\sum_{i=1}^{N}\left(\alpha_k \delta_{i,k}\right)^{-d} K\left(\frac{r-r_i}{\alpha_k \delta_{i,k}}\right). \quad (2)$$

In his simulations, Braiman use a symmetric Gaussian kernel. Here k and α_k still need to be determined. For k or α_k, too small a result will be noisy, whereas if *k* and α_k are large, details are lost. The eigenvalues of the parameters for σ (width of the normal distribution), *k* and α_k were determined by optimizing certain eligibility criteria [11, 19].

Silverman (1986) argues that we can interpret this as the use of a "pilot estimate" of density. We can understand this by observing from equation (1) that (Formula 3):

$$\acute{p}_{kNN}(r_i) \propto \delta_{i,k}^{-d}. \quad (3)$$

Thus, the bandwidth at each location is proportional. Thus, the density estimate of the KNN pilot level is implicitly used to control the final density estimate. The effect is that in low-density regions δ_i, *k* will be large and the nucleus will expand; in high-density regions the opposite happens.

2.4. Fundamentals of the Modified Braiman Estimator (MBE)

Braiman's approach, which is used to find the correct parameter values, is computationally expensive because it need to run the estimator many times to find the optimal parameters. This is even more expensive because the kernel has endless support. This means that each data point contributes to the density at each position, so that O (N^2) is worth testing the parameters [12, 22].

We plan to apply the method to astronomical datasets that are very large (> 50,000 data points) and dimensional (10 to hundreds). For this reason, we use a rapid and scalable modification of the Braiman's method according to the principles of Wilkinson and Meyer (1995). Silverman (1986) noted that an implicit pilot estimate of KNN can be replaced by another estimate without significant changes in quality. Therefore, Wilkinson and Meyer used the core density estimator itself to evaluate the pilot. In addition, they replaced the infinite support of the Gaussian kernel with the finite support of the Epanechnikov kernel, which significantly increases the computational speed and is optimal in terms of the minimum mean integral square error. To increase the computational speed of the pilot estimation, the pilot density field is first calculated at the grid points, after which the pilot signal density for each data point is obtained by multiline interpolation. The method is also scalable: even when the number of data points grows very large, the calculation time remains limited by the number of grid points [13, 25].

In the modified version of equation (2) becomes (Formula 4):

$$\acute{p}(r) = \frac{1}{N}\sum_{i=1}^{N}(\sigma\lambda_i)^{-d} K_e\left(\frac{r-r_i}{\sigma\lambda_i}\right), \quad (4)$$

where K_e is the Epanechnikov core defined as (Formuls 5):

$$K_e(t) = \begin{cases} \frac{d+2}{2V_d}(1-t.t) if t*t<1 \\ 0\ otherwise \end{cases}, \quad (5)$$

where V_d is the volume of a unit sphere in d-dimensional space.

Estimation of density proceeds in two phases [11, 19].

Phase 1. Calculate the optimal width of the window experiment σ_{opt} with the percentage of data defined in equation (6) below. Determine the density of the pilot using equation (4) for $\sigma = \sigma_{opt}$ i $\lambda_i = 1$.

Phase 2. From the density of the pilot to calculate the local bandwidth parameters λ_i on (Formula 6)

$$\lambda_i = \left(\frac{\acute{p}_{pilot}(r_i)}{g}\right)^{-\alpha}. \quad (6)$$

Here g is the geometric mean value of the pilot density, and $\alpha = 1 / d$ is the sensitivity parameter. The value of $1 / d$ is chosen to be equivalent to the Braiman's method, although some authors prefer the value of 1/2 regardless of d. The final estimate of the density is given by the equation. (4) again, but now for $\sigma = \sigma_{opt}$ and λ_i, as given by equation (6).

Compared to the original Braiman's method, it should be noted that a fixed window width σ_{opt} is used for the pilot assessment, rather than a fixed value of k. During the second phase of the algorithm, we change the width of the window with the density at each data point using the local bandwidth parameter. Data points with a low pilot score get a large window and vice versa.

2.5. Adaptive estimation of Gaussian nucleus density (DEDICA)

Pisani proposed a kernel-based density estimation method for multivariate data, which is a continuation of his work for the universal case. Again, this is an adaptive kernel evaluator. The main differences of the MBE method are that the Gaussian core is used and that the optimal bandwidths are determined iteratively, minimizing the cross-checking estimate. The study uses a 3D density estimator DEDICA, which is an implementation of FORTRAN Pisani [3, 8, 19].

For a sample of D_N from N points with position vectors $r_i \in \mathbb{R}_d$, $(i = 1, ..., N)$ and the core width of the i-th point given σ_i, the adaptive estimate of the Gaussian core density is given (Formula 7):

$$\acute{p}(r) = \frac{1}{N}\sum_{i=1}^{N} K_N(|r_i - r|, \sigma_i), \tag{7}$$

where $K_n (t, \sigma)$ is the standard d-dimensional Gaussian nucleus (Formula 8):

$$K_n(t, \sigma) = \frac{1}{(2\pi\sigma^2)^{d/2}} exp\left[-\frac{t^2}{2\sigma^2}\right] \tag{8}$$

The kernel width σi is chosen by an iterative method that minimizes the local error of the integrated square. Procedure as follows:

1. The window width is initialized (Formula 9):

$$\sigma^{(0)} = 4\sigma_t, \quad \sigma_t = A(K)N^{-\frac{1}{d+4}}\sqrt{\frac{1}{d}\sum_{i=1}^{d} s_u^2}, \tag{9}$$

where s_u is the standard deviation of the l-th coordinate of the data, and $A (K)$ = 0.96 for the Gaussian kernel.

2. Iteratively perform the following steps for n = 1,2, ...:

(a) halve the width of the window: $\sigma (n) = \sigma (n - 1) / 2$;

(b) calculate the pilot estimate for (7) with fixed core sizes $\sigma i = \sigma (n)$;

(c) calculate the local throughput coefficients through (6) with $\alpha = 1/2$;

(d) calculate the adaptive core estimate for (7) with adaptive core sizes;

(e) calculate the cross-validation estimate (Pisani 1996, level (7)) (Formula 10):

$$M\left(\acute{p}_{ka}^{(n)}\right) = \frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N} K_n(|r_i - r_j|, (\left(\left(\sigma_i^{(n)}\right)^2 + \left(\sigma_j^{(n)}\right)^2\right)^{\frac{1}{2}}) - \frac{2}{N(N-1)}\sum_{i=1}^{N}\sum_{j\neq i} K_n(|r_i - r_j|, \sigma_j^{(n)}). \tag{10}$$

Minimizing the cross-checking estimate is equivalent to minimizing the integral square error between the true density and the calculated density [5, 14, 20].

3. Determine the iteration number $n = n_{opt}$, for which the cross-check estimate is minimized, and return the corresponding optimal window widths and the adaptive core density estimate at the sampling points.

The cross-checking procedure can be understood by looking at the behavior of different terms. As it decreases during the iteration, some conditions will continue to increase, while others will begin to decrease as the local windows size becomes much smaller than the interpoint distances. This is the point when the minimum is reached and the iteration stops [4, 29].

Although, as shown below, DEDICA gives good results in many cases, in some situations it fails. This can be attributed to some disadvantages of the method. First, the fixed core sizes $\sigma (n)$ used for the pilot estimates form a discrete series of values (determined by the choice of $\sigma (0)$). This range of values

may be too rough to find the optimal window width. Second, the method seeks to achieve what leads to a globally optimal result, which, however, may not be optimal in some regions [12, 19].

An extension to the DEDICA code was developed in this study to obtain the grid density, because the original code calculates only the point density. The optimal window widths of each point, calculated during the estimation of the point density, are used to obtain an adaptive estimation of the core density at each point of the grid r by (7) (Fig. 1).

Dataset	Component	Points	Distribution
1	Trivariate Gaussian 1	40 000	$M_1 = (50,50,50)$ $CM_1 = diag(30)$
	Uniform random noise	20 000	Uniform$(x,y,z) = [0,100]$
2	Trivariate Gaussian 1	20 000	$M_1 = (25,25,25)$ $CM_1 = diag(5)$
	Trivariate Gaussian 2	20 000	$M_2 = (65,65,65)$ $CM_2 = diag(20)$
	Uniform random noise	20 000	Uniform$(x,y,z) = [0,100]$
3	Trivariate Gaussian 1	20 000	$M_1 = (24,10,10)$ $CM_1 = diag(2)$
	Trivariate Gaussian 2	20 000	$M_2 = (33,70,40)$ $CM_2 = diag(10)$
	Trivariate Gaussian 3	20 000	$M_3 = (90,20,80)$ $CM_3 = diag(1)$
	Trivariate Gaussian 4	20 000	$M_4 = (60,80,23)$ $CM_4 = diag(5)$
	Uniform random noise	40 000	Uniform$(x,y,z) = [0,100]$
4	Wall-like structure	30 000	Uniform$(x,y) = [0,100]$,Gaussian$(z) = [M = 50,\ var = 5]$
	Filament-like structure	30 000	Uniform$(z) = [0,100]$,Gaussian$(x,y) = [M = 50,\ var = 5]$
5	Wall-like structure 1	20 000	Uniform$(x,z) = [0,100]$,Gaussian$(y) = [M = 10,\ var = 5]$
	Wall-like structure 2	20 000	Uniform$(x,y) = [0,100]$,Gaussian$(z) = [M = 50,\ var = 5]$
	Wall-like structure 3	20 000	Uniform$(x,z) = [0,100]$,Gaussian$(y) = [M = 50,\ var = 5]$
6	Log-normal	60 000	Log − normal$(x,y,z) = [M = 3,\ var = 4]$

Figure 1: Simulated data sets with known density distributions

2.6. Delaunay Tessellation Field Estimator (DTFE)

DTFE is a well-known in astronomy method of density fields reconstructing from a discrete set of scattered points. In this method, the Delaunay tessellation of points is first constructed. Then the density of a point is defined as inverse to the total volume V of the surrounding tetrahedra (in 3D) of each point multiplied by the normalization constant. To sample DN from N points with position vectors $\in \mathbb{R}_d$, ($i = 1, ..., N$) the DTFE density estimate is given (Formula 11):

$$\acute{p}(r_i) = \frac{1}{N}\frac{d+1}{V_i}, \tag{11}$$

where $V_i = \sum_{j=1}^{K} V_{tetra,j}$. Here $V_{tetra,\ j}$ is the volume of the j-th tetrahedron and K is the number of tetrahedra containing the point r_i.

In the next step, the density field is obtained by linear interpolation of point densities at the vertices of Delaunay tetrahedra to the full sample size.

2.7. Formulation of the problem

Galaxies are strongly influenced by their environment. Quantifying the density of a galaxy is a difficult but critical step in studying the properties of galaxies.

Therefore, the aim is to identify differences in density estimation methods and their application in astronomical problems. We study the effectiveness of four density estimation methods: *k*-nearest neighbors (KNN), adaptive Gaussian nucleus density estimation (DEDICA), a special case of adaptive Epanechnikov nucleus density estimation (MBE), and Delaunay tessellation field estimator (DTFE).

3. Analytical section

3.1. Dataset

Investigated the effectiveness of four density estimation methods on three classes of datasets: a series of simulated datasets with known density fields to test the ability to recover relatively simple density distributions of each method; astronomical data set with unknown but well-selected density field based on millennium simulations; and two different observed galaxy samples taken from SDSS.

3.2. Modulated data sets with known density fields

We start by constructing six simulated data sets with known density distributions (Fig. 1).

Data set 1 is the unimodal Gaussian distribution with added uniform noise.

Data set 2 contains two Gaussian distributions with the same number of points, but different covariance matrices (MC) and different centers, again with added uniform noise; this data set has the same number of points as 1.

Data set 3 contains four Gaussian distributions with equal number of points, but different KM and different centers, again with added uniform noise; this data set has twice as many points as data sets 1 and 2.

Data set 4 contains a wall and thread-like structure. The x and y coordinates of the wall-like structure are derived from the uniform distribution, and the z-coordinate is derived from the Gaussian distribution. The pod-like structure is created with a Gaussian distribution in the x and y coordinates and a uniform z-coordinate distribution.

Data set 5 contains three wall-like structures, where each wall is created with a uniform distribution in two dimensions and a Gaussian distribution in the third.

Data set 6 contains points derived from the lonormal distribution.

The representation of the scattering graphs of these data sets is shown in Fig. 2.

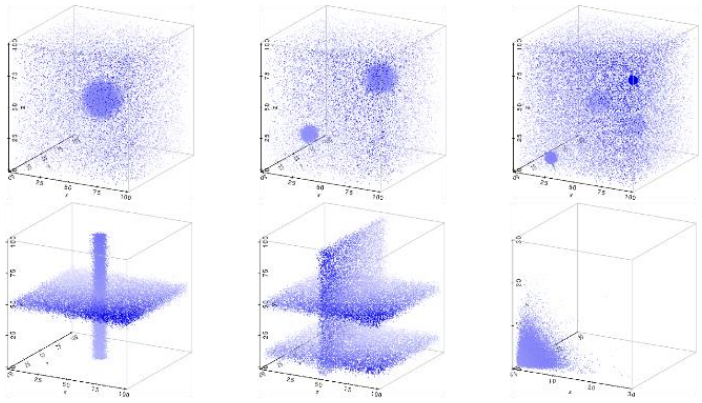

Figure 2: Graphical representations of scattered simulated data sets. Left to right, top to bottom: data sets 1-6

3.3. Astronomical data sets with fields of unknown density

Three astronomical data sets are used to test the performance of methods on astronomical data: semi-analytical model galaxies performed as a result of Millennium modeling, and two samples of galaxies performed with SDSS.

3.3.1. MSG data set

The first astronomical data set consists of a L-Galaxy "millimil" experimental model Millennium sample. Simulation Millennium is one of the largest simulations that has ever studied the evolution of the universe, after almost 2 × 1010 particles. It was created to predict the scale structure of the universe

and compare them with observational data and astrophysical theories. L-Galaxies are created by inhabiting halo trees extracted from the Millunium simulation with semi-analytical models according to the commandments of De Lucius and Blaisot. A much smaller “milliMillennium” simulator (“milliMil”) is used, which took a sample of only ~ 2 × 107 particles and associated L-galaxy data. This dataset is referred to as the MSG dataset, which contains 53,918 points. In the visual representation, the simulation output looks like a thin three-dimensional fabric of threads with fractal self-similarity and several layers of organization.

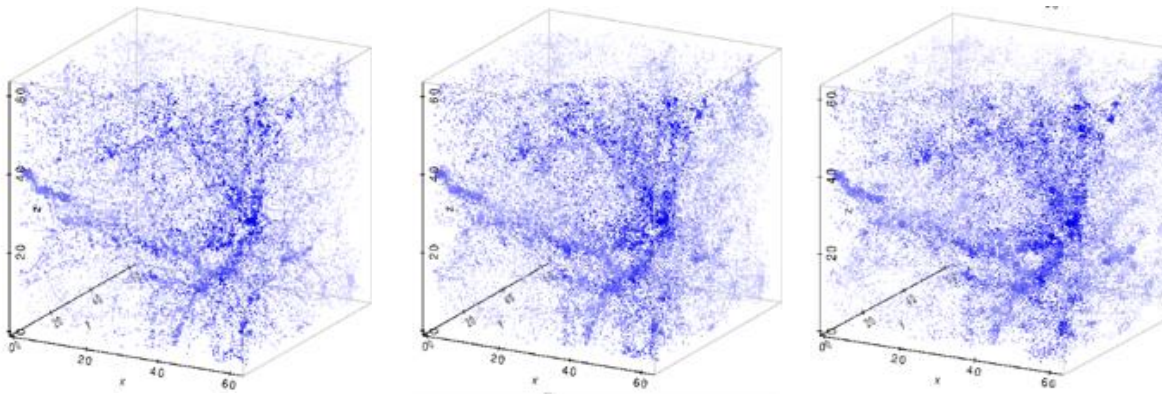

Figure 3: Representation of plot graphs of MSG data sets. Top to bottom: MSG data, MSG-DTFE data set, MSG-MBE data set

The goal is to use the MSG dataset complexity for testing the methods performance with a well-chosen but reasonably "astronomical" setting. Unfortunately, the true core density field of the MSG dataset is unknown. Therefore, MSG samples are downloaded to determine the "true density" for astronomical data. The MSG data density field is used to create new datasets, and their density is considered to be the true density of these datasets. The process of creating new datasets can be described as follows:

Step 1: Calculate the density field of the MSG data set using one of the density estimation methods.

Create a new data set that has a probability density function similar to MSG data as follows:

1. Generate a random position $r_i(x, y, z)$ within the original sample and a random value of p between zero and the maximum density of the sample field;
2. Interpolate the density P of the point $r_i(x, y, z)$ in the field obtained from step 1;
3. If p <P take the point $r_i(x, y, z)$ as a point in the new data set; P is the true density $r_i(x, y, z)$;
4. Repeat steps 2-3 until the required number of points is obtained.

Two such datasets were created, one was using DTFE (called the "MSG-DTFE dataset") and the other was using the MBE (called "MSG-MBE"), both of them was with the same number of points as the original MSG dataset. For the MSG-MBE data set, the true density P was interpolated from a grid of 2563 points, and for the MSG-DTFE data set from Delaunay tessellation. The representation of the scattering graphs of these three fields - the original MSG data set and the two derived data sets - is shown in Fig. 3. It should be noted that both obtained data sets look soothing, like the original MSG data set, although in some derived data sets there is a slight smoothing.

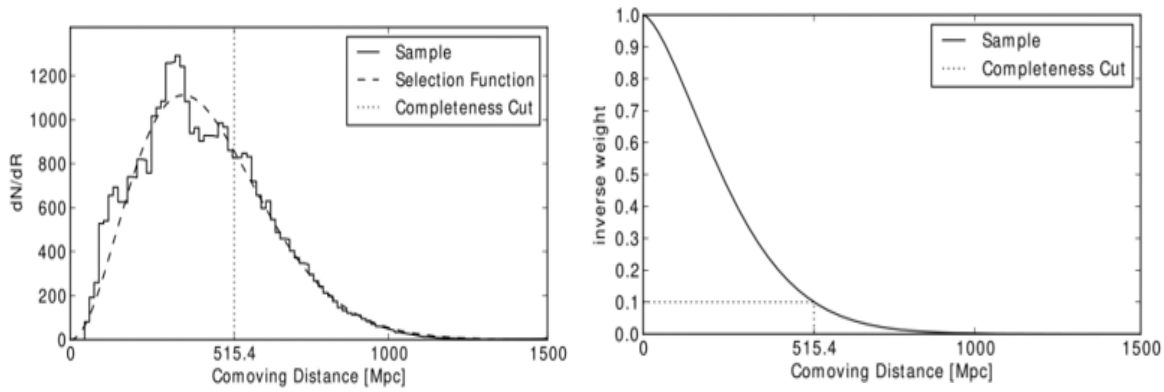

Figure 4: Above: distance distribution of SDSS spectroscopic data over a distance in the distance zone

Figure 4. Above: distance distribution of SDSS spectroscopic data over a distance in the distance zone, assuming a cosmology of concordance (Ωm = 0.28, $\Omega\Lambda$ = 0.72, h = 0.7). The dotted line approaches this distribution, assuming that galaxies perform the function of Schechter luminosity with an obvious limit value r <17.7. Bottom: The corresponding inverse weight based on the luminosity function. To extract a high level of redshift, a 10% level of completeness is selected (corresponding to R = 515 MPC, which is equivalent to z = 0.122).

Next, the field density generated by all density estimation methods is compared with the true densities obtained in the process described above.

3.3.2. SDSS data sets

Finally, to apply these density estimation techniques to the observed astronomical data, two samples of galaxies are extracted from DR7 in SDSS: a "cone" of galaxies over a relatively small solid angle in the sky but extended by redshift and a "z-shell" of galaxies over a small redshift interval

Spectroscopic redshift is used to calculate the distance R, which is then converted to Cartesian coordinates to estimate the density using flat cosmology with Ω_m = 0.28, Ω_Λ = 0.72, h_0 = 0.7.

Completeness correction is required when calculating the density according to SDSS, which we discuss before submitting samples. SDSS is selected by size, but is not (initially) limited to redshift. This means that the distance of the number of galaxies in the sample decreases with distance, because weak galaxies can no longer be detected, causing a low density for distant galaxies. To counteract this effect, weights are calculated for each distance, assuming the Schechter glow function, following the procedure of Martinezta Saar (2002). For this calculation, all SDSS galaxies with a spectroscopic distance between 50 and 2000 M_{ps} (corresponding to a redshift from 0.0117 to 0.530) and Petrosyansky r <17.7 are used. If galaxies follow Schechter's luminosity function, they must also follow the distribution of numbers (Formula 12):

$$\frac{dN}{dR} = \langle p(r) \rangle \Omega R^2 \Phi(R), \tag{12}$$

where $\langle \rho\ (r) \rangle$ average field density, Ω survey area and $\Phi\ (R)$ - selection function given (Formula 13)

$$\Phi(R) = e^{-(\frac{R}{R_c})^{\beta}}, \tag{13}$$

The best suitability of the equation (12) for the data (Ω = 2,444 sr) set $\langle \rho\ (r) \rangle$ = 0,013 Mpc-3, Rc = 299,8 Mpc and β = 1,5 and is shown in Fig. 4, above. The corresponding selection function is shown in Fig. 4, below. After the calculation, the density is adjusted by dividing by the value of the selection function by the distance of the galaxy.

It should be noted that due to the fiber masks used for SDSS spectroscopy, not all (bright) sources in dense media have spectroscopic redshifts. These sources are not included in the sample, and failure to correct this has led to a shift in low density in the densest regions.

The "primary" galaxies of 1939 were chosen within the rectangular boundary R_A = (185,190) and Dec = (9,12). The celestial coverage of our sample is 14.7 °.

To cut the sample of the galaxy, the lower limit of completeness was chosen (Fig. 4) - 10%; to limit the effect of long distances; incompleteness up to 90% does not cause unacceptably large errors when trying to estimate the density of galaxies. This corresponds to the distance R_{max} = 515 Mps (redshift 0.123).

To prevent edge effects and limit the effects of local motion, the lower distance limit is set at R_{min} = 50.0 Mps (corresponding to a redshift of 0.0117). This results in the final number of galaxies in the 1030 cone sample. Bulk densities were calculated using this sample with a limited size and redshift of 1030 galaxies.

The integration of equation (12) for the cone sample (Ω_{cone} = 0.00449 sr) is expected to have 2702 sources in the region, of which 692 should be detected. Instead, the cone sample has 1030 galaxies, which is 49% more than expected. A comparison with other regions of the same size shows that the cone pattern is indeed extremely dense. Therefore, the average field density of the "cone" sample was corrected by $\langle \rho_{cone}(r) \rangle$ = 0.0196 Mps-3.

The definition of σ_{opt} for MBE is insufficient for narrow cone-shaped samples. Problem cases of such samples are strong alignment of one axis (or planes) of the Cartesian coordinate system or alignment with one of the diagonals of space. The first leads to too small σ_{opt} value because one or two values of σ_l will be much smaller than the others (s), while the last leads to too high σ_{opt} because N does not reflect incomplete filling of the sample space. Therefore, a new definition of σ_{opt} was created for conical samples: first, the average distance to the nearest half of the galaxies is determined; then σ_{opt} is selected as the square root of the cross section of the cone at this distance.

	Integrated squared error				Generalized Kullback-Leibler divergence			
Dataset	MBE	DEDICA	DTFE	kNN	MBE	DEDICA	DTFE	kNN
1	2.23×10^{-7}	6.44×10^{-6}	1.54×10^{-5}	2.82×10^{-5}	5.61×10^{-2}	7.62×10^{-2}	1.83×10^{-1}	1.59×10^{-1}
2	3.04×10^{-6}	1.75×10^{-6}	5.85×10^{-5}	1.19×10^{-4}	4.53×10^{-2}	8.34×10^{-2}	1.90×10^{-1}	1.62×10^{-1}
3	4.74×10^{-6}	9.10×10^{-6}	1.99×10^{-4}	4.28×10^{-4}	3.90×10^{-2}	6.77×10^{-2}	1.62×10^{-1}	1.54×10^{-1}
4	2.35×10^{-6}	2.91×10^{-4}	1.12×10^{-5}	2.02×10^{-5}	6.22×10^{-2}	$1.33 \times 10^{+1}$	2.34×10^{-1}	1.79×10^{-1}
5	5.65×10^{-7}	5.38×10^{-7}	1.31×10^{-6}	2.13×10^{-6}	1.01×10^{-1}	9.12×10^{-2}	2.42×10^{-1}	2.12×10^{-1}
6	7.66×10^{-4}	7.94×10^{-5}	1.96×10^{-3}	3.71×10^{-3}	3.21×10^{-1}	6.32×10^{-2}	1.07×10^{-1}	1.43×10^{-1}
MSG-DTFE	1.68×10^{-3}	4.86×10^{-3}	1.24×10^{-3}	1.39×10^{-3}	6.50×10^{-1}	$2.18 \times 10^{+1}$	5.74×10^{-1}	5.73×10^{-1}
MSG-MBE	6.89×10^{-7}	5.88×10^{-4}	1.95×10^{-1}	1.71×10^{-4}	3.00×10^{-2}	$2.26 \times 10^{+1}$	1.25×10^{0}	3.08×10^{-1}

Figure 5: Performance of density estimators: simulated data sets and MSG

4. Experiments

4.1. Simulated data sets

Firstly, we studied the effectiveness of four density estimation methods on six simulated datasets and then on the two data sets obtained by MSG.

The effectiveness of the methods for artificial data sets in the upper rows of Fig. 5 is compared using ISE and GKLD indicators. True density is the parametric density calculated from the parameters used to create data sets. It is clear that the methods based on the adaptive core, MBE and DEDICA, perform much better than KNN or DTFE in restoring the density distributions of the input data. For all but the data set 6, the lonormal distribution, the performance of the MBE is better or approximately equal to the efficiency of DEDICA. It is worth noting that the MBE density was calculated by automatically selecting the core size, and the best MBE performance can be obtained by changing the anti-aliasing setting manually.

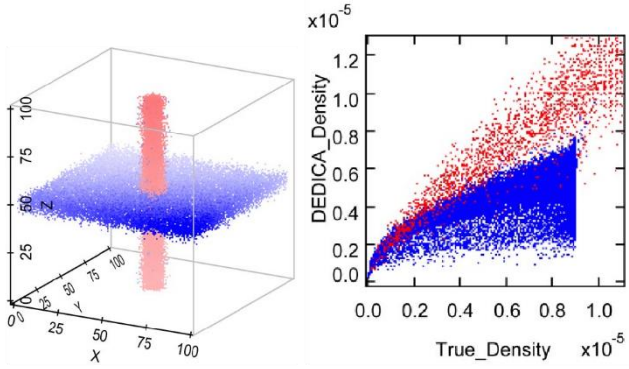

Figure 6: DEDICA performance for the data set 4

The thread is red and the wall is blue. Left: Spatial representation of the data set. Right: a comparison of true and DEDICA-formed density.

It can also be noted that DEDICA works very poorly for data set 4 (wall plus thread), where it is not possible to estimate the proper density. Studying the density of points in Fig. 6, it is seen that DEDICA underestimates the density in the wall. This is because the correct kernel size could not be selected during automatic (cross-validation) selection of the kernel size on this dataset. Similar behavior is also observed when considering MSG and SDSS datasets. In addition, the field produced by KNN is not normalized. For data sets from 1 to 6 fields on average are approximately 25-30% excessively dense. This is why KNN performs the worst in terms of integral square error on these datasets.

4.2. MSG datasets

The performance of the density estimators on the MSG datasets in the lower rows of Fig. 7 is compared. DTFE is expected to work best on the MSG-DTFE dataset, and MBE works best on the MSG-MBE dataset. Interestingly, KNN works in the same way as DTFE on the MSG-DTFE dataset. This is not a complete surprise, as DTFE and KNN are conceptually similar, as both use only points in close proximity to the current location for direct density estimation. Because of this, both can work better than core estimates, with strong gradients or even fundamental density gaps. Nevertheless, the MBE works in much the same way as DTFE and KNN on the MSG-DTFE dataset, suggesting that the MBE continues to operate even in spatially complex datasets.

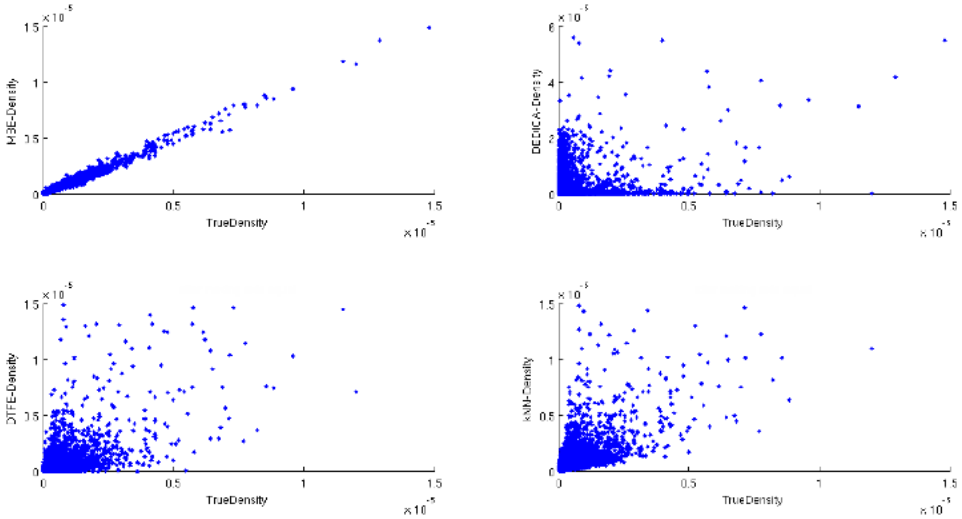

Figure 7: Diagram of the true and approximate density of the MSG-MBE data set field using MBE (upper left), DEDICA (upper right), DTFE (bottom left) and KNN (bottom right). Approximately 16,000 random grids are shown

The GKLD in Fig. 5 shows that DEDICA is unable to estimate the appropriate density for the samples from the Millennium dataset. For both samples, MSG DEDICA produces very different density distributions compared to the "true" distribution (MSG-MBE dataset Fig. 7). As mentioned above, a similar performance of DEDICA was observed on the simulated data set 4, which contains a filamentous structure. The MSG data set also contains an obvious filamentous structure. Again, it turns out that DEDICA's automatic kernel size selection (using cross-checking) failed to select the appropriate kernel size for such datasets (although it works quite well in Gaussian and lonormal cases).

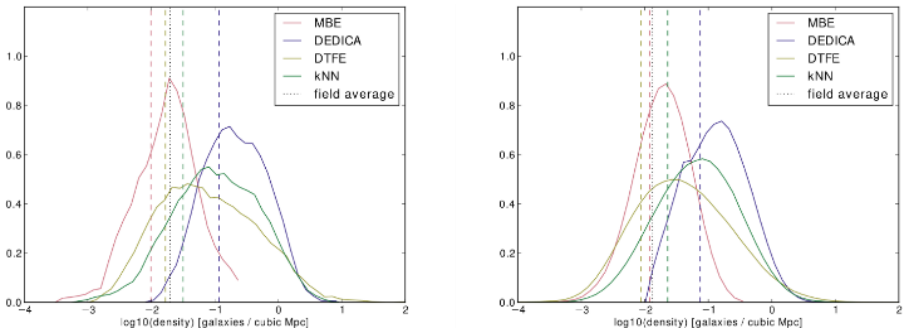

Figure 8: Normalized distribution of density values in the logical space for each estimator

The distribution is smooth and close to Gauss. A wider density range (DTFE, KNN) means that the estimator detects more clustering. More clustering leads to an increase in galaxies in regions of higher density, shifting the distribution peak to the right. The dotted line represents the measured average field density from the selection function. Left: sample "cone". Right: a sample of the "shell".

4.3. Density distributions

We are currently studying the application of our own density estimators to two observable data sets of galaxies with SDSS, "cone" and "shell" samples.

Starting with comparing the distributions of density values obtained by four different methods (Fig. 8). All four methods for estimating density give an approximately logical distribution of values for SDSS samples (as expected from previous research and theoretical ideas). Therefore, the analysis is performed with the logarithm of the density, or "standardized density", defined as (Formula 14):

$$p_s = \frac{p_l - \mu_l}{\sigma_l}, \tag{14}$$

where μ_l and σ_l are the mean and standard deviation of (almost) Gaussian density distributions. We give graphs of the distribution of logarithmic density in Fig. 8.

The true average galaxy density $\langle \rho(r) \rangle$ for the cone and shell samples is 0.0196 and 0.013 galaxies per cubic megaparsec, respectively. The average value of the calculated densities cannot be directly compared with this number, because it is averaged over the set of galaxies and $\langle \rho(r) \rangle$ over the field. High-density regions contain more galaxies, and therefore they have more weight in the average density of points. This weight is proportional to the density, and if a lonormal distribution of the estimated density is assumed, the average value of the calculated field density can be calculated as (Formula 15):

$$\langle \acute{p}(r) \rangle = e^{\ln 10 \mu_i - \frac{(\ln 10 \sigma_j)^2}{2}}. \tag{15}$$

For each estimator, the calculated value is shown in Fig. 7, as well as the known average field density. For the DTFE "cone" sample the best approximates the known average field density, closely following the MBE. For the "shell" sample, this order is reversed. DEDICA incorrectly represents the known average field density, and KNN is between them.

The distribution of the "shell" sample is smoother than that of the "cone" sample due to the larger number of data points. Even for the "shell" sample, the DEDICA density distribution is not equal, due to global optimization, which results in small window widths. The MBE density distribution peak reaches a slightly higher density for the "shell" sample. In addition to the difference in means and widths, differences in density methods are manifested in the tails of the calculated density distribution. DTFE creates high-density tails because it is sensitive to excessive density due to the local nature of the method. MBE creates a low density tail. And the distribution from KNN has stronger tails of high and lower class (compared to Gauss).

The DEDICA density distribution is offset by other distributions. Comparing the estimated average field density and the true average field density, it can be seen that the calculated values cannot represent the actual

density. This is due to DEDICA's sensitivity to excessive density: in the case of high-cluster data such as ours, they create very small cores, emphasizing the density field. Moving the positions of galaxies by 1 MP in the random direction, thereby slightly homogenizing the sample, removes this effect almost completely. However, despite the fact that the density of DEDICA galaxies is much higher than expected, it can still be used as a parameter describing the environment of galaxies, using it in a standardized form.

5. Conclusion

All four methods are applicable in astronomical problems; in general, a modified Braiman estimator is preferred. For artificial datasets, kernel-based methods outperform DTFE and KNN in terms of integral quadratic error and Kullbeck-Leibler discrepancy. Correct kernel sizing is crucial, and DEDICA is unable to properly estimate kernel size in more complex datasets such as Millennium and SDSS simulations.

From the artificial data sets, it can be concluded that the methods based on adaptive nuclei, MBE and DEDICA, are better in restoring the "true" density distributions than the KNN or DTFE methods. However, DEDICA obviously has difficulties with spatially complex distributions, which makes it unsuitable for use in problems related to the large-scale structure of the universe.

All the methods overestimate the density of dense regions, and DTFE has the largest deviation from the true density, because the density of DTFE is close to infinity if the volume of the surrounding tetrahedra is close to zero. On the other hand, all methods almost equally underestimate the density in areas of low density.

DTFE even produces zero density for points on the convex body of the data set. However, in astronomical conditions it is not always problematic. The convex body is the edge of the sample: physically outside the edge are galaxies that are not represented in our estimated densities. Therefore, all methods give a density lower than the unknown "true" densities in these regions. The zero values of the DTFE density estimator can be used as an implicit indication that the density estimation was not successful for these galaxies. With other methods, these galaxies are silently in a bucket for too low a density.

6. References

[1] A. Tsybakov, Introduction to Nonparametric Estimation. Springer: Berlin/Heidelberg, Germany, 2009.

[2] F. Ziegelmann, Nonparametric estimation of volatility functions: the local exponential estimator. Econom. Theory, Vol. 18, 2002, pp. 985–991.

[3] D. Kristensen, Nonparametric filtering of the realized spot volatility: A kernel-based approach. Econom. Theory, Vol. 26, 2010, pp. 60–93.

[4] Y. Zu, H. Boswijk, Estimating spot volatility with high-frequency financial data. J. Econom, Vol. 181, 2014, pp. 117–135.

[5] B. Maillot, Propriétés Asymptotiques de Quelques Estimateurs Non-Paramétriques Pour des Variables Vectorielles et Fonctionnelles. Ph.D. Thesis, Université du Littoral, Paris, France, 2008.

[6] C. Chesneau, J. Fadili, B. Maillot, Adaptive estimation of an additive regression function from weakly dependent data. J. Multivar. Anal., Vol. 133, 2015, pp. 77–94.

[7] N. Kunanets, O. Vasiuta, N. Boiko, Advanced Technologies of Big Data Research in Distributed Information Systems, in: International Scientific and Technical Conference on Computer Sciences and Information Technologies, Vol. 3, 2019, pp. 71–76,

[8] L. Crambes, A. Delsol, Y. Laksaci, Robust nonparametric estimation for functional data. J. Nonparametr. Stat., Vol. 20, 2008, pp. 573–598.

[9] A. Gheriballah, A. Laksaci, R. Rouane, Robust nonparametric estimation for spatial regression. J. Stat. Plan. Inference, Vol.140, 2010, pp. 1656–1670.

[10] P. S. Abril, R. Plant, The patent holder's dilemma: Buy, sell, or troll? Communications of the ACM, Vol. 50, 2007, pp. 36–44. doi:10.1145/1188913.1188915.

[11] S. Cohen, W. Nutt, Y. Sagic, Deciding equivalances among conjunctive aggregate queries. J. ACM, Vol. 54, 2007. doi:10.1145/1219092.1219093.

[12] S. W. Smith, An experiment in bibliographic mark-up: Parsing metadata for xml export, in: R. N. Smythe, A. Noble (Eds.), in: Proceedings of the 3rd. annual workshop on Librarians and Computers, Vol. 3, 2010, pp. 422–431. doi:99.9999/woot07-S422.

[13] N. Boyko, N. Tkachuk, Processing of medical different types of data using hadoop and Java mapreduce, in: CEUR Workshop Proceedings, Vol. 2753, 2020, pp. 405–414.

[14] M. V. Gundy, D. Balzarotti, G. Vigna, Catch me, if you can: Evading network signatures with web-based polymorphic worms, in: Proceedings of the first USENIX workshop on Offensive Technologies, WOOT '07, USENIX Association, Berkley, CA, 2007.
[15] D. A. Anisi, Optimal Motion Control of a Ground Vehicle, Master's thesis, Royal Institute of Technology (KTH), Stockholm, Sweden, 2003.
[16] H. Thornburg, Introduction to bayesian statistics, 2001. URL: http://ccrma.stanford.edu/jos/bayes/bayes.html.
[17] R. Ablamowicz, B. Fauser, Clifford: a maple 11 package for clifford algebra computations, version 11, 2007. URL: http://math.tntech.edu/rafal/cli11/index.html.
[18] Poker-Edge.Com, Stats and analysis, 2006. URL: http://www.poker-edge.com/stats.php.
[19] B. Obama, A more perfect union, Video, 2008. URL: http://video.google.com/videoplay?docid=6528042696351994555.
[20] D. Novak, Solder man, in: ACM SIGGRAPH 2003 Video Review on Animation theater Program: Part I, Vol. 145, 2003, p. 4. URL: http://video.google.com/videoplay?docid=6528042696351994555. doi:99.9999/woot07-S422.
[21] N. Lee, Interview with bill kinder. Comput. Entertain., Vol. 3, 2005. doi:10.1145/1057270.1057278.
[22] M. Saeedi, M. S. Zamani, M. Sedighi, A library-based synthesis methodology for reversible logic. Microelectron. J., Vol. 41, 2010, pp. 185–194.
[23] M. Saeedi, M. S. Zamani, M. Sedighi, Z. Sasanian, Synthesis of reversible circuit using cycle-based approach. J. Emerg. Technol. Comput. Syst., Vol. 6, 2010.
[24] M. Kirschmer, J. Voight, Algorithmic enumeration of ideal classes for quaternion orders. SIAM J. Comput, Vol. 39, 2010, pp. 1714–1747. URL: http://dx.doi.org/10.1137/080734467. doi:10.1137/080734467.
[25] TUG, Institutional members of the TEX users group, 2017. URL: http://www.tug.org/instmem.html.
[26] R Core Team, R: A language and environment for statistical computing, 2019. URL: https://www.R-project.org/.
[27] S. Anzaroot, A. McCallum, UMass citation field extraction dataset, 2013. URL: http://www.iesl.cs.umass.edu/data/data-umasscitationfield.
[28] N. Shakhovska, N. Boyko, P. Pukach, The Information Model of Cloud Data Warehouses, in: Advances in Intelligent Systems and Computing, Vol. 871, 2019, pp. 182–191.