SoftwareX 18 (2022) 101017

Contents lists available at ScienceDirect

SoftwareX

journal homepage: www.elsevier.com/locate/softx

Original software publication

MATLAB tool for probability density assessment and nonparametric estimation

Jenny Farmer [a], Donald J. Jacobs [a,b,*]

[a] Department of Physics and Optical Science, University of North Carolina at Charlotte, Charlotte, NC 28223, USA
[b] Center for Biomedical Engineering and Science, University of North Carolina at Charlotte, Charlotte, NC 28223, USA

ARTICLE INFO

Article history:
Received 7 April 2020
Received in revised form 18 September 2021
Accepted 11 February 2022

Keywords:
Nonparametric density estimation
Maximum entropy
Single order statistics
MATLAB
R

ABSTRACT

A MATLAB function is presented for nonparametric probability density estimation, based on an iterative method that employs the principle of maximum entropy and characteristic properties of single order statistics. Featuring a robust and adaptive design, the method is well-suited for high throughput applications. The implementation comprises a MATLAB interface and underlying C++ code with extensible components that can be easily integrated into third party software. The functionality includes plotting capabilities and model independent diagnostics featuring the scaled quantile residual that is invariant to sample size, distribution, and estimation method.

Code metadata

Current code version	v2.1
Permanent link to code/repository used of this code version	https://github.com/ElsevierSoftwareX/SOFTX_2020_166
Legal Code License	GPL
Code versioning system used	git
Software code languages, tools, and services used	C++, MATLAB
Compilation requirements, operating environments & dependencies	MATLAB Install MingGW C/C++ compiler for Windows as a MATLAB Add-on to compile MEX
Support email for questions	jfarmer@carolina.rr.com

1. Motivation and significance

1.1. Background

Estimating the probability density function (PDF) from a given sample of random variables is a ubiquitous task in diverse applications across disciplines such as economics [1,2], engineering [3], physics [4,5], biology [6–8], and statistics [9]. As machine learning and the analysis of big data become increasingly important in many scientific fields [10–13], readily accessible software that automates nonparametric estimation and model-free error checking is needed. This paper is based on a C++ implementation of nonparametric PDF estimation, called PDFE, which has several differentiating characteristics over kernel density methodology. The details of PDFE have been previously published [14], and the software is also available on the Comprehensive R Archive Network (CRAN) [15]. The focus of this article is the MATLAB interface to PDFE, but the method is briefly summarized here.

PDFE constructs a PDF based on the principle of maximum entropy. In a traditional maximum entropy method (MEM), a set of characteristic functions relating to various moments is introduced, and the coefficients to these functions are optimized to match the predicted moments with the empirical moments. The form of the density function is

$$\boldsymbol{p}(\boldsymbol{v}) = \sum_{j=1}^{D} \exp(\lambda_j \boldsymbol{g}_j(\boldsymbol{v})) \quad (1)$$

* Corresponding author at: Department of Physics and Optical Science, University of North Carolina at Charlotte, Charlotte, NC 28223, USA.
E-mail addresses: jfarmer6@uncc.edu (Jenny Farmer), djacobs1@uncc.edu (Donald J. Jacobs).

https://doi.org/10.1016/j.softx.2022.101017

where $\mathbf{g_j}(\boldsymbol{\nu})$ are bounded level functions and $\boldsymbol{\lambda_j}$ are Lagrange coefficients controlling the shape of the density function [16]. For a fixed number of coefficients, D, the form is parametric.

PDFE employs an expansion of orthogonal functions in the form of Eq. (1), where higher modes are successively added as needed, testing each trial PDF according to a scoring function. The algorithm employs a random search method to iteratively explore possible density functions by perturbing the Lagrange multipliers from their current values. The perturbation is Gaussian distributed where the standard deviation decreases as the final solution nears. Lagrange multipliers are updated when the score for the calculated density improves. As progress slows, more Lagrange multipliers to higher modes are considered.

1.2. Scoring and scaled quantile residual

A scoring function is employed to rate the quality of a trial PDF, where multiple scoring functions have been implemented and benchmarked previously [17]. From the trial PDF, a cumulative density function (CDF) is calculated on the range (0 1). This trial CDF is an accurate representation of the data when $\mathbf{CDF}(\mathbf{x})$ models sampled uniform random data (SURD). The inverse problem becomes assessing if $\mathbf{r_k} = \mathbf{CDF}(\mathbf{x_k})$ represents SURD. Due to its simplicity, and being one of the best performing scoring functions overall, PDFE employs an average quadratic z-score, defined as $\mathbf{z^2} = \frac{1}{\mathbf{N}}\sum_{\mathbf{k=1}}^{\mathbf{N}}(\mathbf{r_k}-\boldsymbol{\mu_k})^2/\boldsymbol{\sigma^2}$, where $\mathbf{k}$ is the sort ordered position in a sample size N, and $\boldsymbol{\mu_k}$ and $\boldsymbol{\sigma_k}$ are the mean and standard deviation from single order statistics [18] known to be $\boldsymbol{\mu_k} = \mathbf{k}/(\mathbf{N}+\mathbf{1})$ and $\boldsymbol{\sigma_k} = \mathbf{u_k}(\boldsymbol{\mu_k} - \mathbf{1})/\sqrt{\mathbf{N}+\mathbf{2}}$. Minimizing this function corresponds to minimizing the variance in single order statistics. To visually assess the quality of the estimate per position and identify the locations of potential errors, a scaled quantile residual (SQR) is defined as $\mathbf{SQR_k} = \sqrt{\mathbf{N}+\mathbf{2}}\,(\mathbf{r_k} - \boldsymbol{\mu_k})$. The scaling factor of $\sqrt{\mathbf{N}+\mathbf{2}}$ creates a sample-size-invariant metric for each position $\mathbf{k}$. It has been shown that, when plotted against position, $\mathbf{SQR_k}$ for SURD falls approximately within an oval shaped region [14,17].

Upon close examination of SURD, there is a slight asymmetry in the oval-shaped range of expected values, as seen in the scatter plot in Fig. 1. This plot was created by calculating the SQR for 10,000 trials of SURD with a sample size of $N = 100$. The horizontal dashed line at zero indicates the SQR for exactly uniform data, which highlights the skewed SQR ranges near the endpoints. This asymmetry originates from sort order statistics [18], where the probability of position k in a sample size of n random variables from a uniform distribution follows a beta distribution with parameters $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$, equal to $\mathbf{k}$ and $\mathbf{N}+\mathbf{1}-\mathbf{k}$ respectively. A few examples of these distributions for a sample size of 100 are shown in Fig. 1, coloured according to their positions on the SQR. The distributions are highly skewed near the boundaries, whereas the midpoint follows a Gaussian distribution with a much larger variance. This asymmetry in SQR affects estimates within the tails of probability densities for small sample sizes only, having virtually no effect when $N >> 100$.

The SQR plot is useful as an assessment tool for benchmarking density estimation quality because it is independent of sample size and the underlying distribution. Popular measures such as Kullback–Leibler [19] and Kolmogorov–Smirnov [20] require the true PDF for comparison. Furthermore, the SQR plot highlights low confidence or over-fitted regions within the estimated PDF.

1.3. A MATLAB tool for PDFE

The PDFanalyze MATLAB function is a flexible interface to PDFE for MATLAB users that includes a convenient plotting tool to visualize the estimate with publication-quality figures. Minimal usage requires a single MATLAB function call with a data sample input, similar to the native MATLAB kernel density estimation (KDE) function. Unlike standard KDE, default settings are data-driven, where resolution and boundary settings are automated with no user intervention. For high-throughput applications, density estimates can be returned with plotting suppressed. The SQR plot is available in multiple formats for a visual assessment of the quality of a proposed estimate. Additional parameters are available to accommodate greater control over difficult distributions with specific requirements and constraints. Given the previously demonstrated power and versatility of the SQR [17], the functionality has been further refined and extended for a more accurate and accessible implementation, taking into consideration the skewed ranges of the SQR plot.

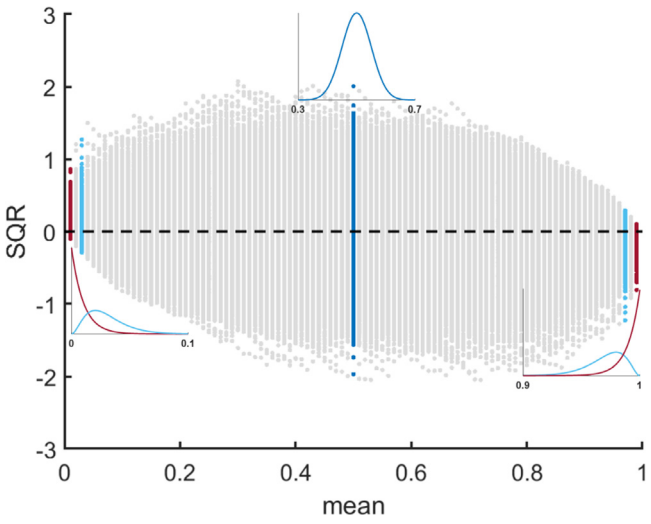

Fig. 1. Scatter plot of SQR for 10,000 trials of 100 samples showing skewness of range near the boundaries. The beta distributions shown for select positions quantify this asymmetry..

2. Software description

The PDF analysis package, depicted in Fig. 2, has three separate components: a high level MATLAB user interface for plotting and analysis (shown in grey), a low level core C++ class library for rapid estimation (blue), and C++ interface code providing communication between MATLAB and C++ (orange). Each component will be discussed in the following three subsections.

2.1. PDFE

The blue blocks in Fig. 2 depict nine C++ classes that comprise the underlying functionality of PDFE, and they can be compiled separately as an independent executable. The object-oriented class design allows for customizable programming for developers, as well as seamless integration into popular statistical software such as R and MATLAB. The ***OutputControl*** class directs error and informational messages to the appropriate console as specified by compiler directives in the header file. Support for command line, MATLAB, and R consoles is currently included, but additional output protocols can be customized by the developer. The ***InputParameters*** and ***WriteResults*** classes handle the input and output of PDFE respectively, allowing sample data and PDF results to either be passed to a calling function or handled with text files.

The ***InputData*** and ***ChebyChev*** classes analyse the random sample to identify outliers, determine the appropriate adaptive resolution scales, establish estimated boundaries, and transform

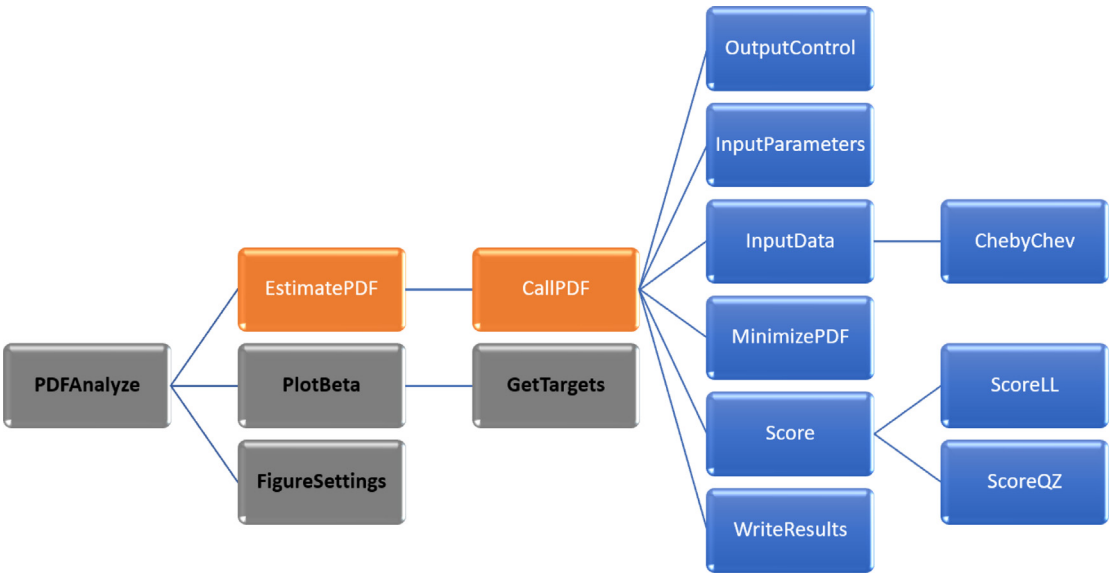

Fig. 2. Structure of source files for PDFE (blue), PDFAnalyze (grey), and interface files (orange). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

the data to a finite range. The methods provided within these classes collectively define automated data-driven parameters for PDF estimates. This intelligent analysis of the data allows for high-throughput estimation requiring no direction from the user. The core processing for the density estimate is contained within the ***MinimizePDF*** and ***Score*** classes. Additional scoring functions can be customized by implementing an instance of the ***Score*** virtual template class. The quadratic z-score is implemented in ***ScoreQZ*** as the default scoring method. ***ScoreLL***, based on log-likelihood, was the original scoring function implemented for PDFE and is included as an additional example.

2.2. MATLAB/C++ interface

The ***EstimatePDF*** and ***CallPDF*** classes, shown in orange in Fig. 2, define an interface for a high-level MATLAB function call to PDFE. The interface and PDFE classes are compiled together as a MATLAB executable (mex) file, creating a custom MATLAB function called ***EstimatePDF***. A random data sample is a required input to ***EstimatePDF***. The default values for all other parameters are tuned to produce optimal estimates across a wide range of testing but can be modified for customized results. For example, assuming infinite support, boundaries are set to exclude extreme outliers. The user may bypass this behaviour by specifying exact upper and/or lower limits.

Resolution and smoothness can be adjusted mainly through two parameters. Increasing the number of integration points or the SURD target increases resolution, accuracy, and processing time. Decreasing the SURD target creates faster and smoother estimates. The extent of the impact on these parameters is distribution-dependent, making experimentation necessary. Furthermore, the limits on Lagrange multipliers can be modified to achieve a semi-parametric fit by narrowing the range between these limits. A strictly parametric approach can be achieved by setting the two limits to the same value. For example, setting both minimum and maximum to 1 forces a uniform fit. Similarly, setting them both to 2 or 3 respectively yields exponential and Gaussian distributions. These advanced options allow a high level of intervention but are not necessary for most users.

2.3. PDFAnalyze

The four grey blocks in Fig. 2 represent MATLAB functions designed to control the plotting and analysis of PDF estimates. ***FigureSettings*** overrides MATLAB default parameters to improve appearance and resolution of all subsequent figures, creating high-quality plots to maximize the effectiveness of visualization. ***PlotBeta*** creates a shaded background based on the beta distributions that represent SURD probabilities by position, with darker shading representing higher probability. This background is automatically scaled by sample size on the y-axis, and optionally scaled by the range of the sample data along the x-axis. ***GetTargets*** is a high-precision numeric integrator for the beta distribution to obtain confidence contour lines for the SQR plot. These utility functions can be called independently, but PDFAnalyze orchestrates all these utilities. Detailed usage of PDFAnalyze is given in the MATLAB help documentation.

3. Illustrative examples

To illustrate the plots available in PDFAnalyze, consider the Burr distribution, with parameters $\mathbf{c} = \mathbf{k} = \mathbf{2}$. The Burr distribution features a long tail to the right and is often used to model household income [21]. For the purpose of comparison, the exact target PDF is shown in Fig. 3a and d for sample sizes of 100 and 3000 respectively. Fig. 3b and e are examples of the SQR plot type that reveal important features central to the analysis of a PDF. The blue scatter points represent the SQR for the estimate at each sample point that is returned from EstimatePDF. The grey shading and dashed lines indicate the relative probability and thresholds for the SQR by position. The dark inner dashed lines and the lighter outer dashed lines represent 50% and 98% confidence levels respectively, indicating the percentage of data expected within these enclosed ovals. Note that the 98% ovals are not symmetric about the x-axis for 100 samples (Fig. 3b), but that this asymmetry is not distinguishable for 3000 samples (Fig. 3e), because the SQR features automatically adapt according to the sample size. SQR values outside the 98% thresholds are shown in red, highlighting where the estimate may not fit well. Similarly, the 50% threshold depicts an area in which approximately half of the SQR values are expected to fall. The estimate is likely

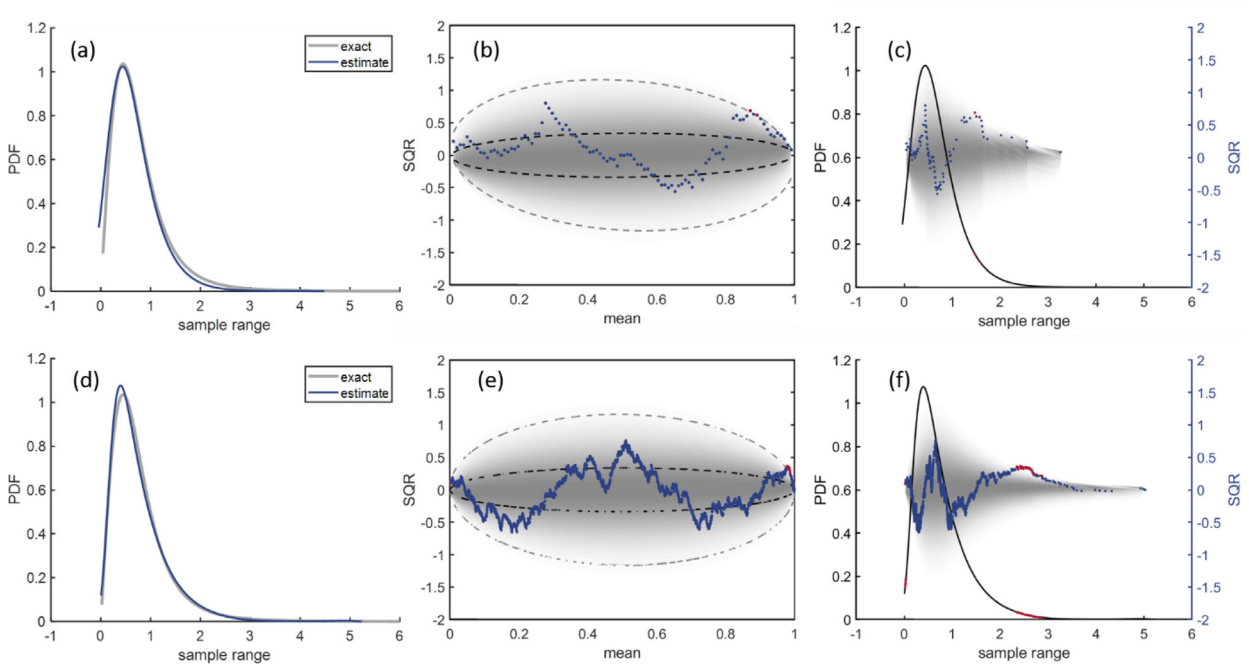

Fig. 3. Examples for plot types PDF (column 1), SQR (column 2), and Combined (column 3) showing the Burr distribution for sample sizes 100 (row 1) and 3000 (row 2).

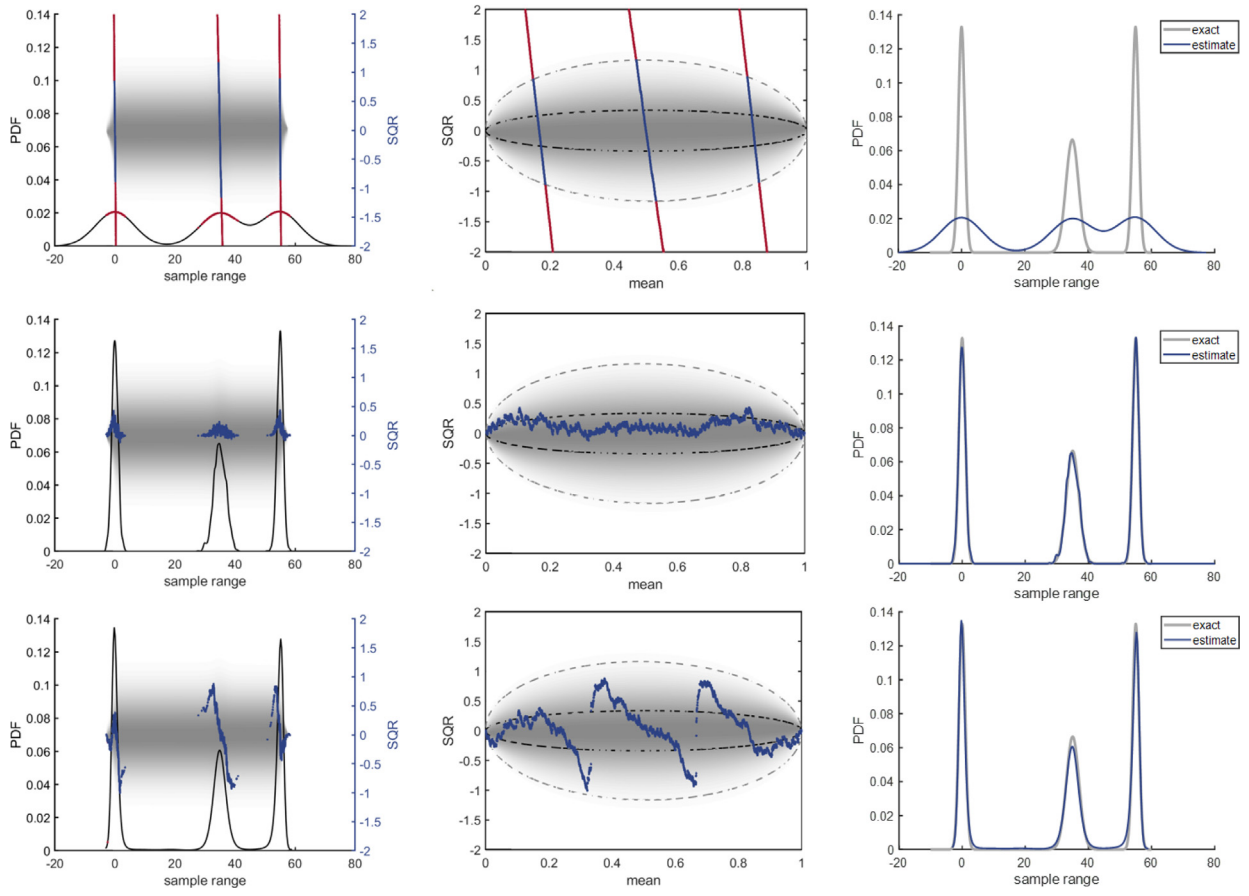

Fig. 4. Comparative examples between ksdensity (row 1), kde (row 2), and PDFE (row 3) demonstrating multi-resolution scales for a tri-modal distribution. Column 3 shows the estimated and exact PDFs, column 2 is the SQR, and column is the combination of both.

overfitting to the sample data when all points within the SQR plot fall entirely within this smaller oval.

Fig. 3c and f demonstrate a combined plot type, including the functionality of both the PDF and the SQR together. Although the

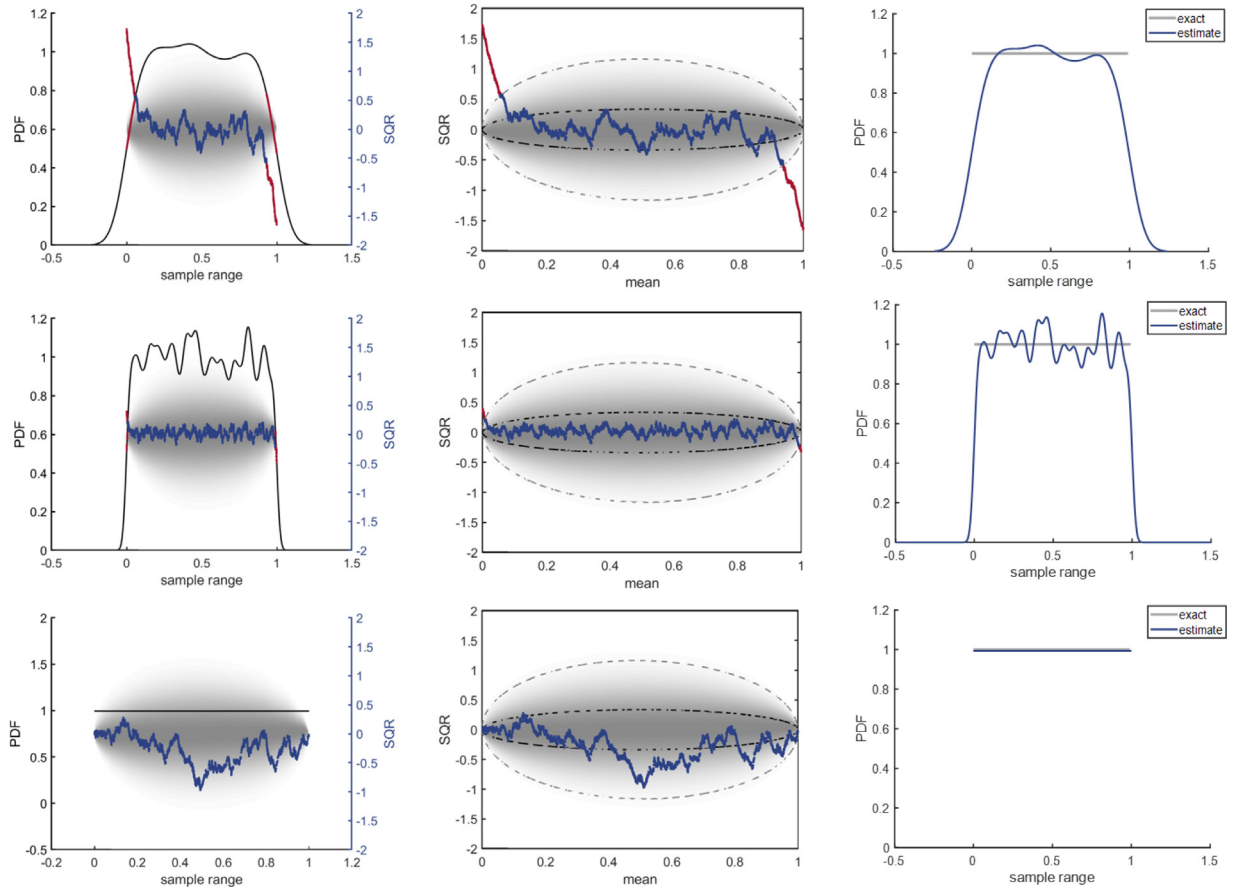

Fig. 5. Comparative examples between ksdensity (row 1), kde (row 2), and PDFE (row 3) demonstrating a fixed boundary uniform distribution. Column 3 shows the estimated and exact PDFs, column 2 is the SQR, and column is the combination of both.

dashed lines are not shown, the grey shading and red markers remain as visual guides. The shape of the shaded region will generally appear distorted because it is mapped to the sample data along the x-axis to align with the PDF. The sample data points falling outside the 98% threshold are also marked in red on the PDF, thus showing where these outliers fall along the estimate. Statistically, it is expected that 2% of SQR values will fall outside of this threshold, so red markings do not necessarily indicate a poor estimate, however, high numbers of outliers certainly warrant suspicion, as in Fig. 3e and f. The red areas fall predominately along the long tail of the Burr distribution, a result not uncommon in under-sampled regions. Although the fit to the tail appears quite good in Fig. 3d, the SQR shown in red alerts the user that the expected level of quality for the estimate has diminished. Often, as larger sample sizes yield an improved fit for the PDF, a greater number of red flags in the SQR appear. This counter-intuitive trend reflects a smaller error tolerance as the amount of sample data increases.

4. Impact

The SQR plots included in PDFAnalyze provide a means of visual assessment for any density estimation method. Comparing PDFE to alternative nonparametric methods illustrates its power and advantages. For example, MATLAB includes a standard KDE implementation function, ***ksdensity***. Although KDE has known weaknesses, such as boundary estimation and resolution, it is commonly used because it is fully nonparametric, produces fast results, and includes a simple interface. To mitigate problems with KDE, myriad variants have been developed [12,13,22–25]. One such refinement employs a linear diffusion process towards an adaptive method and has been shown to improve upon standard KDE, particularly in the case of widely separated multimodal densities [26]. This method is implemented as a MATLAB function, ***kde***, and is available through the MathWorks File Exchange network [27].

An example of a multimodal distribution is shown in Fig. 4, consisting of three separate Gaussian distributions. The top row of Fig. 4 shows the performance of ***ksdensity*** for 3000 sample data points on the multimodal distribution. The mean values of the three peaks are estimated well, but the standard deviations are spread much too widely. The SQR plot flags these errors starkly, showing data samples on either side of each mean far outside the 98% threshold. The middle row of Fig. 4 demonstrates the same distribution estimated by ***kde***, with much improved results in the estimate. Although the SQR plot has no red data points that indicate problem areas, nearly all the points are within the 50% threshold, indicating the sampled data is being overfit. The bottom row of Fig. 4 shows that PDFE yields an excellent estimate with the SQR not indicating systematic overfitting or underfitting to the sample data.

Another comparative example for the uniform distribution on the interval (0, 1) is shown in Fig. 5. Despite its simplicity, the finite boundaries notoriously challenge traditional KDE methods. The ***kde*** estimate is again suggestive of an overfit to the data and both KDE methods yield poor estimates near the boundaries, as expected. For the default KDE case, the red areas outside the oval in the SQR plot alerts the user to the inaccuracies and

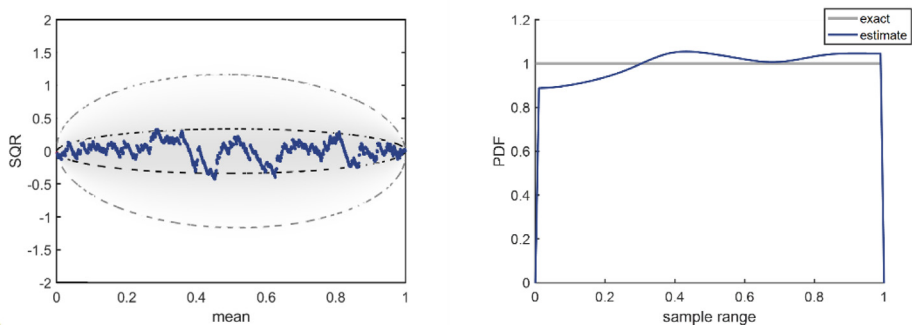

Fig. 6. SQR (left) and PDF (right) estimates for a uniform distribution with a boundary correction specified in ksdensity.

their location within the estimate. Armed with this information, the user can opt to apply a boundary correction term to KDE. Fig. 6 shows the resulting PDF and SQR plots for ***ksdensity*** with the BoundaryCorrection parameter set to 'reflective' for finite boundary support on (0, 1). The impact demonstrated in these representative examples is twofold. First, PDFE provides robust nonparametric estimates superior to KDE methods, without user intervention or expertise. Second, the SQR plots created by PDFAnalyze provide visual clues for evaluating the validity of any estimate without the need to know the exact PDF. In particular, the locations of errors and uncertainty are unambiguously highlighted in a consistent manner for any distribution, sample size, or PDF estimator.

5. Conclusions and future direction

The PDFAnalyze software package has been implemented in MATLAB providing a fully automated data-driven nonparametric PDF estimation tool as well as model-free density estimation diagnostics. PDFAnalyze is built upon stand-alone underlying C++ code called PDFE. The plotting and analysis tools included in PDFAnalyze identify potential regions of the estimate that over or under fit to the sample data. A parallelized version of PDFE is under development for improved accuracy and speed in general, and specifically to handle heavy tailed distributions exhibiting extreme statistics. Future work will extend the method to include multi-dimensional estimation.

CRediT authorship contribution statement

Jenny Farmer: Conceptualization, Methodology, Writing – original draft, Software, Visualization. **Donald J. Jacobs:** Supervision, Methodology, Writing – review & editing, Project administration.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] Mikosch T, de Vries CG. Heavy tails of OLS. J Econometrics 2013;172(2):205–21.
[2] Alemany R, Bolancé C, Guillén M. A nonparametric approach to calculating value-at-risk. Insurance Math Econom 2013;52(2):255–62.
[3] Weilong R, et al. A novel asymmetrical probability density function for modeling log-ratio SAR images. IEEE Geosci Remote Sens Lett 2016;13(3):369–73.
[4] Wang JP, Yun X, Chang SC. A new procedure modeling the probability distribution of earthquake size. Physica A 2014;413:385–93.
[5] Pressé S, et al. Nonadditive entropies yield probability distributions with biases not warranted by the data. Phys Rev Lett 2013;111(18):1–4.
[6] Lee M, Kang YS, Seok J. The estimation of probability distribution for factor variables with many categorical values. PLoS One 2018;13(8):e0202547-e0202547.
[7] Vanmourik S, Stigter H, Molenaar J. Prediction uncertainty assessment of a systems biology model requires a sample of the full probability distribution of its parameters. PeerJ 2014;2(1).
[8] Farmer J, et al. Statistical measures to quantify similarity between molecular dynamics simulation trajectories. Entropy 2017;19((12):646.
[9] Munkhammar J, Mattsson L, Ryden J. Polynomial probability distribution estimation using the method of moments. PLoS One 2017;12(4):e0174573.
[10] Tigani S, et al. Low complexity algorithm for probability density estimation applied in big data analysis. Int J Comput Appl 2014;101(7):1–5.
[11] Cavuoti S, et al. METAPHOR: A machine-learning-based method for the probability density estimation of photometric redshifts. Mon Not R Astron Soc 2017;465(2):1959–73.
[12] Sidibé A, et al. Big Data Framework for Abnormal Vessel Trajectories Detection using Adaptive Kernel Density Estimation. ACM; 2018, p. 43–6.
[13] Tang L, et al. A network kernel density estimation for linear features in space–time analysis of big trace data. Int J Geogr Inf Sci: Hum Dyn Mob Big Data Era 2016;30(9):1717–37.
[14] Farmer J, Jacobs D. High throughput nonparametric probability density estimation. PLoS ONE 2018;13(5):e0196937.
[15] Farmer J, Jacobs D. PDFEstimator: Nonparametric probability density estimator. R package version 3.0. 2021, https://CRAN.R-project.org/package=PDFEstimator.
[16] Jacobs DJ. Best probability density function from limited sampling. Entropy 2008;11:1001–24.
[17] Farmer J, et al. Universal sample size invariant measures for uncertainty quantification in density estimation. Entropy 2019;21(11):1120.
[18] Wilks SS. Order statistics. Bull Amer Math Soc 1948;54(1):6–50.
[19] Kullback S. The Kullback–Leibler distance. Amer Statist 1987;41:340–1.
[20] Massey FJ. The Kolmogorov–Smirnov test for goodness of fit. J Amer Statist Assoc 1951;46(253):68–78.
[21] De Capitani L, Nicolussi F, Zini A. Trivariate burr-III copula with applications to income data. METRON 2017;75(1):109–24.
[22] Malec P, Schienle M. Nonparametric kernel density estimation near the boundary. Comput Statist Data Anal 2014;72:57–76.
[23] Hazelton M. Kernel Smoothing. Wiley StatsRef: Statistics Reference Online; 2014.
[24] McCarthy MT, O'Callaghan CA. PeaKDEck: A kernel density estimator-based peak calling program for DNaseI-seq data. Bioinform 2014;30(9):1302–4.
[25] Ramachandran P, Perkins TJ. Adaptive bandwidth kernel density estimation for next-generation sequencing data. BMC Proc 2013;7(S7):1–10.
[26] Botev ZI, Grotowski JF, Kroese DP. Kernel density estimation via diffusion. Ann Statist 2010;38(5):2916–57.
[27] Botev Z. Kernel density estimator. 2020, https://www.mathworks.com/matlabcentral/fileexchange/14034-kernel-density-estimator, MATLAB File Central Exchange. Retrieved March 17, 2020.