A Comparison Report on Different Density Estimator

Yuli Song

DATA GENERATING PROCESSES AND PRELIMINARY EXPERIMENTS

The goal of the experiment and competitors chosen

Kernel density estimation is a useful statistical tool to estimate the probability density function of a random variable. Often shortened to KDE, it's a non-parametric technique that can create a smooth curve given a set of data. Besides KDE, there are also many other density estimators. Therefore, one interesting topic is to compare the performance of KDE and other methods. In order to investigate in this topic, several one-shot experiments and Monte Carlo simulation are designed to explore (a) the performance of multiple density estimator methods, and (b) the effect of different sample size in Monte Carlo simulation.

The most common competitor of KDE is histogram. There're also some other methods such as mixture models, k-nearest neighbor and smoothing spline and so on. In this report, Gaussian finite mixture model(mixture model), smoothing spline ANOVA models(smoothing spline) and histogram is used for comparison. Code for this report is available here.

Generate experiment data

When selecting the adequate data sets, both data generated from common distributions and some famous data sets like Old Faithful Geyser Data are considered. However, as the original distribution is known when generating the machine-generated data and machine-generated data provides strong guarantees against privacy violations, data generated from some distributions is used in this experiment. To generate the experiment data, we consider three 1D scenarios first:

1. Data generated from $Normal(0,1)$ distribution. This distribution is selected as it's a good representative of symmetric, unimodal, and asymptotic distributions.

2. Data generated from $Gamma(2,\frac{1}{3})$ distribution. This distribution represent those distributions that are skewed and have domain restrictions.

3. Data generated from $0.7*Normal(-10,1)+0.3*Normal(5,0.5)$ distribution. This distribution has good property of multiple modes.

For the cases above, they are used inside both the one-shot experiment and the Monte Carlo simulation. It's also a good practice to expand the domain to 2D space. Note that when the domain increases to 2D, there are much more interesting scenarios than that in 1D. However, due to page limit, only two scenarios are chosen here and 2D scenarios are only considered inside the one-shot experiment:

1. Data is simulated on an annulus by using uniform distribution. Source code is edited here. This distribution is chosen as it doesn't have a generally accepted peak.

2. Data is simulated on an $Normal(\mu_1,\Sigma_1)+Normal(\mu_2,\Sigma_2)$ with $\mu_1=\mu_2=(10,10)^T$ and $\Sigma_1=\Sigma_2=\begin{pmatrix}1 & 0\\ 0 & 1\end{pmatrix}$. This distribution is chosen as it has two peaks inside 3D space.

Moreover, as generating enough data avoids selection bias and guarantees randomness in samples, it's important to generate large data sets. However, as it can be computationally expensive to generate too much data, eventually, 1000 samples are generated in each of the one-shot experiment.

Conduct the one-shot experiment and comment

To compare different models' performance under the same level, it's important to make sure that each model nearly reaches its optimal performance and hence some important parameters should be tuned for each model.

The 1D histogram is computed and plotted using hist() function, and the density of the histogram is mainly influenced by the breaking points. However, R's default algorithm for calculating histogram breaking points is Sturges' formula, which implicitly bases the bin sizes on the range of the data and may perform poorly if the data are not normally distributed. Therefore, to find the bin number that can optimize the performance of histogram, cross-validation estimator of risk is used and R function cv.hist.fun() is available on Wasserman's from the All of Statistics website.

The 1D KDE is computed using density() function. The choice of kernel is not crucial but the choice of bandwidth is important. Also, R's default algorithm for selecting bandwidth is Silverman's rule of thumb. While this rule of thumb is easy to compute, it should be used with caution as it can yield widely inaccurate estimates when the density is not close to being normal and it has a strong tendency to oversmooth. Eventually, the plug-in bandwidth introduced by Sheather and Jones as it has a convergence rate much faster than the cross-validation selectors and it generally performs extremely well.

The 2D KDE and histogram are computed using kde2d() and hist2d() function separately. However, it's really hard to check the goodness of default parameters. Therefore, no change is made to the parameter part. This is also one reason why no Monte Carlo simulation is conducted for 2D domain.

The mixture model is computed using densityMclust() function. When fitting the model, the function will directly calculate BIC for each model fitted with different hyperparameters and select the optimal model, there's no need for further hyperparameter optimization.

The smoothing spline models is computed using ssden() function. However, the author of this package only mentions that the model complexity is largely determined by the number of smoothing parameters and didn't mention further about parameter tuning. Therefore, the default parameter is used here.

To conduct the one-shot experiment, several packages are used:

1. mclust packageTo estimate probability densities using Gaussian finite mixture model.

2. mvtnorm packageTo generate two dimensional gaussian distribution.

3. MASS packageTo compute two dimensional kernel density estimation.

4. gss packageTo estimate probability densities using smoothing spline ANOVA models.

5. gplots package: To compute and plot two dimensional histogram.

Eventually, by running the code, several plots are captured for all scenarios (See appendix). For the 1D case, it's easy to find that all methods can recover the original curve well. The result can be explored further:

1. Inside the Figure 1, smoothing spline model and mixture model exactly produce the bell-shape distribution, but smoothing spline model is more close to the original model. Considering mixture model is composed of many Gaussian distributions and it only have one Gaussian model here, this is a reasonable result. KDE model produces a nearly bell-shape distribution and the model is smooth. Histogram produces a roughly bell-shape distribution but the model shape is discrete.

2. Inside the Figure 2, KDE model is most close to the original distribution and smoothing spline model is the second most close. Histogram is still not preferred as it's not smooth. However, KDE model, smoothing spline model and mixture model also have one problem - their domain is outside the range of the original one.

3. Inside the Figure 3, mixture model is most close to the original distribution and it exactly produces the two-mode distribution. But other methods also perform quite well. Another finding is that the KDE model trained from the default parameter can't recover the real model well while the KDE model trained from the adjusted parameter can. It justifies our decision to change the hyperparameter.

For the 2D case, it's easy to find that some methods can recover the original curve well while some not. Note that smoothing spline model is not compared here due to the challenges to code for drawing 3D plots of it. The result can be explored further:

1. Figure 4 is the scatter plot of 2D data generated and contour plot of mixture model. The result shows that mixture model tends to undersmooth the data.

2. Inside the Figure 5, KDE model performs good and it is most close to the original distribution. Histogram and mixture model doesn't perform well as they both undersmooth the data and generate too much peaks. Especially for histogram, the peaks generated are quite discrete and sharp.

3. Inside the Figure 6, all models perform good. However, they all have some problems. For KDE model, it tends to oversmooth the data. For histogram, its model is not smooth. For mixture model, it tends to undersmooth the data.

STRENGTHS AND WEAKNESSES OF DIFFERENT METHODS

KDE, histogram and smoothing spline model are all non-parametric models while mixture model are semi-parametric models. Compared with parametric models, they make only weak, general prior assumptions about the data and are very flexible. Therefore, even the data doesn't satisfy any assumption regarding parametric families, they can still construct a density that reasonably fits the data. This conclusion can be verified by the one-shot experiment above. However, they are very expensive in memory and CPU and they don't have much opportunity to incorporate prior knowledge into the model built.

For KDE, among four methods adopted, its computational cost is relatively low, which can be checked by measuring the execution time. Also, its overall performance is the best within the one-shot experiment above as it can always produce smooth model that fits the original model well. However, it's domain is usually larger than that of the data with restricted domain.

For histogram, it's easy to find that it can't produce smooth and continuous model and it's really hard to use one simple function to represent the model generated by histogram. However, the domain of the model generated by histogram is exactly the same as the original model. Also, its computational cost is low.

For mixture model, it performs well in most of the one-shot experiment, especially for those models composed of normal distribution. Also, it can always produce smooth model. And the model trained by mixture model can be expressed in simple format. However, it's domain is usually larger than that of the data with restricted domain and tends to undersmooth the model. And its computational cost is quite high.

For smoothing spline model, it performs well in most of the one-shot experiment and it can always produce smooth model. However, it's domain is usually larger than that of the data with restricted domain. And its computational cost is quite high.

MONTE CARLO SIMULATION STUDY

Explain what experiment will be conducted next

The further experiment will be conducted is the Monte Carlo simulation on KDE, histogram, mixture model and smoothing spline model under all the scenarios designed before. To check the performance of each model and the influence of sample size $n = 250$, 500, and 1000, mean and standard deviation of integrated squared error(ISE) will be calculated for Monte Carlo simulation.

Report the ISE and discuss what happens when sample size increases

By running the code, the mean value and standard deviation of ISE under different cases can be obtained:

Value	Dist	KDE			Histogram		
		250	500	1000	250	500	1000
Mean	Norm Dist	0.00330655	0.001856879	0.001114913	0.01240743	0.007784372	0.004902357
	Gamma Dist	0.001623677	0.001008014	0.0006296534	0.004633285	0.002939214	0.001778537
	Mix Model	0.001173814	0.0006751656	0.0003913412	0.002911143	0.001713241	0.00119162
Sd	Norm Dist	0.00227001	0.001188117	0.000675403	0.008996144	0.005580659	0.002062684
	Gamma Dist	0.0007094382	0.0004059617	0.0002323864	0.006736886	0.001968476	0.0009768748
	Mix Model	0.001851897	0.0010681	0.000613399	0.1347564	0.0853433	0.0453686

Table 1. Mean and Sd of ISE under KDE and Histogram.

Value	Dist	Mixture Model			Smooth Spline Model		
		250	500	1000	250	500	1000
Mean	Norm Dist	0.001055327	0.0004914699	0.0002539465	0.001876304	0.001013376	0.0005855561
	Gamma Dist	0.003698371	0.002122818	0.001354644	0.001906122	0.001244318	0.0008339185
	Mix Model	9.050128e-5	6.053535e-5	3.624322e-05	0.002837427	0.001638725	0.0009512964
Sd	Norm Dist	0.001320288	0.000552482	0.000269173	0.001520139	0.0008141249	0.0004678311
	Gamma Dist	0.00156573	0.0007910385	0.000468714	0.0009451824	0.0005409505	0.0003333594
	Mix Model	0.00162663	0.001151384	0.0006057706	0.001718881	0.0009530741	0.0005065663

Table 2. Mean and Sd of ISE under Mixture Model and Smooth Spline Model.

From the result, it's easy to observe that mean of ISE decreases with sample size increases under all scenarios.One reasonable explanation is that the data is less likely to have selection bias when sample size increases and the density estimator performs better on data that can recover the original distributions better. The standard deviation of ISE shows that it also decreases with sample size increases under all scenarios. It is because larger data set generated tends to recover the original distribution more consistently under the impact of random seed.

Another interesting finding from the mean and variance is that it's KDE that performs the best and the most stable under distributions composed of non-normal distributions while it's mixture model that performs the best and the most stable under distributions composed of normal distributions. Histogram performs the worst and the most inconstant in all scenarios.

Also, KDE and histogram perform the best in mixture distribution and perform the most consistent in gamma distribution while the mixture model performs the best in mixture normal distributions and then normal distributions. But mixture model performs the most consistent in normal distributions and then mixture normal distributions. Smooth spline model performs better and more consistent in unimodal distributions.

CONCLUSION OF THE REPORT

In conclusion, compared with histogram, mixture model and smoothing spline model, KDE have both advantages of (a)It can recover the original data well, (b) It's a simple and fast calculation algorithm. And it's very suitable for modelling the non-normal distribution, which is a big class in real life. However, when coding for KDE, tuning its parameter, especially the bandwidth parameter, is very important. If the inadequate parameter is chosen, the model may perform not as good as other methods.

REFERENCES

[1] Jing Lei. *Cross-Validation With Confidence*. *Journal of the American Statistical Association*. 532:1978-1997, 2020.

[2] Gu, Chong; Qiu, Chunfu. *Smoothing Spline Density Estimation: Theory*. *Ann. Statist*. 21 (1993), no. 1, 217–234

[3] Luca Scrucca, Michael Fop, T. Brendan Murphy and Adrian E. Raftery *mclust 5: Clustering, Classification and Density Estimation Using Gaussian Finite Mixture Models*. *The R Journal*. Vol. 8/1, Aug. 2016

[4] Project 8 (Nonparametric density estimation)
`https://www.stat.tamu.edu/ suhasini/teaching613/Project8(Jaehong).pdf`

[5] Notes for Nonparametric Statistics
`https://bookdown.org/egarpor/NP-UC3M/kde-i-bwd.htmlkde-i-bwd-comp`

[6] Density Estimation
`http://www2.stat.duke.edu/ wjang/teaching/S05-293/lecture/ch6.pdf`

[7] Mixture density estimation
`http://www.svcl.ucsd.edu/courses/ece271A/handouts/mixtures.pdf`

Appendix

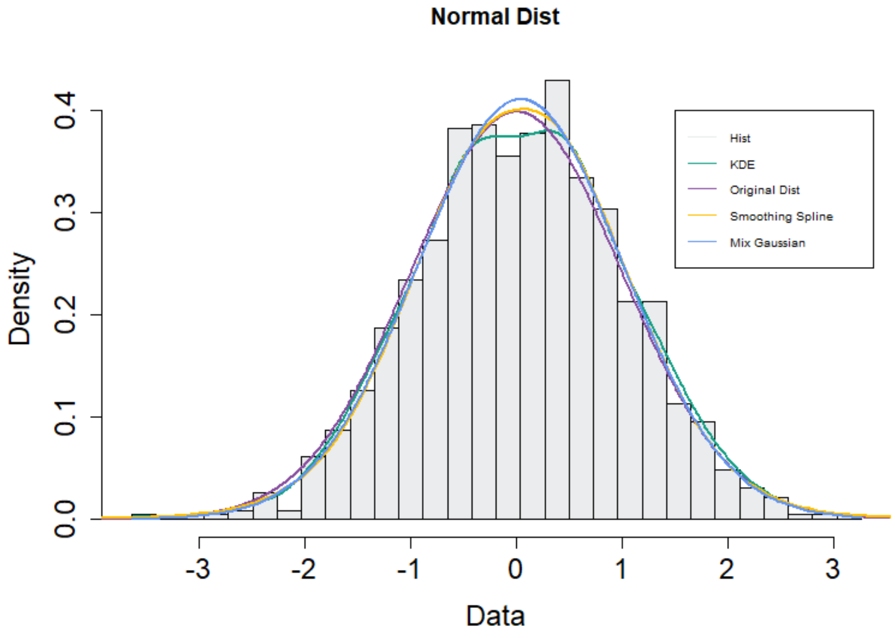


Figure 1. Density estimation of normal distribution.

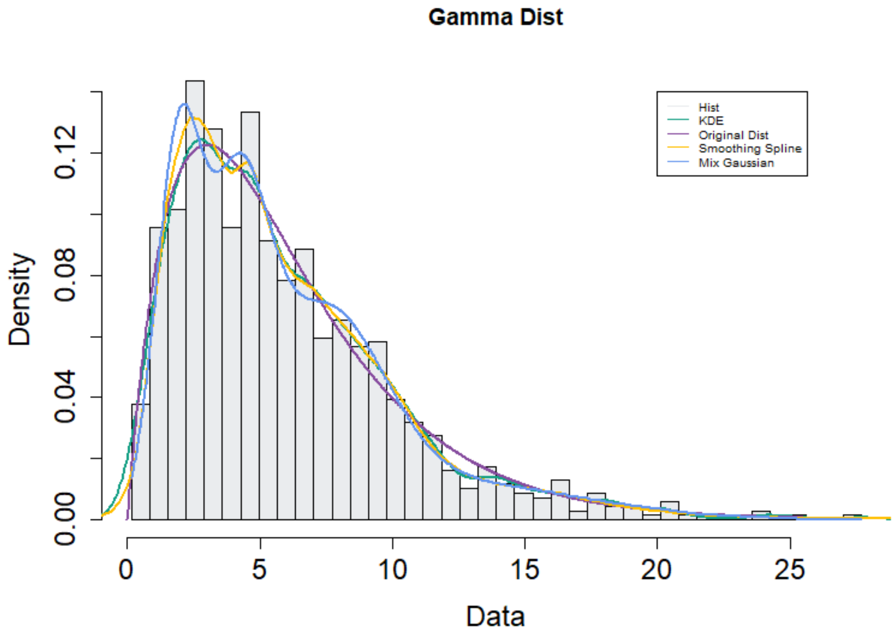


Figure 2. Density estimation of gamma distribution.

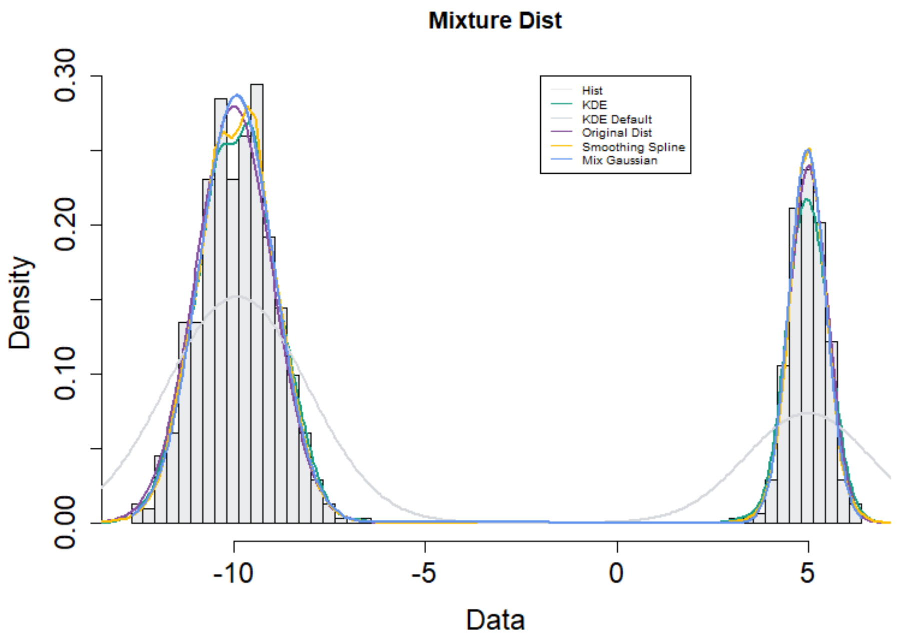
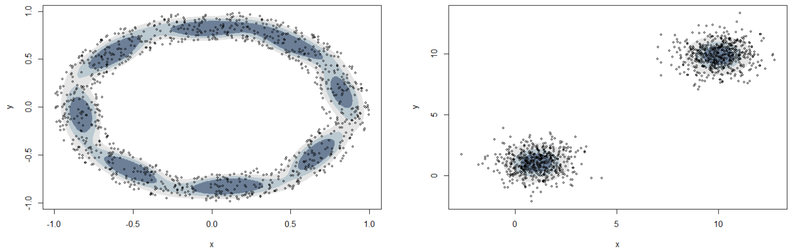


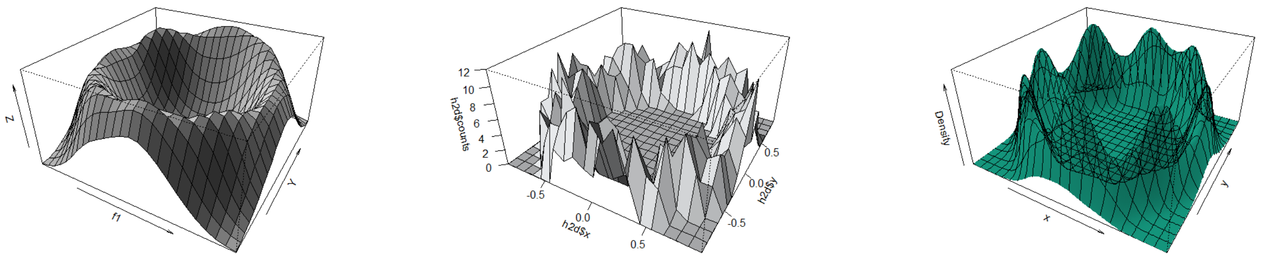
Figure 3. Density estimation of mixture distribution.



(a) Data on annulus

(b) Data from multimodal distribution

Figure 4. Scatter plot of 2D data generated and contour plot of mixture model

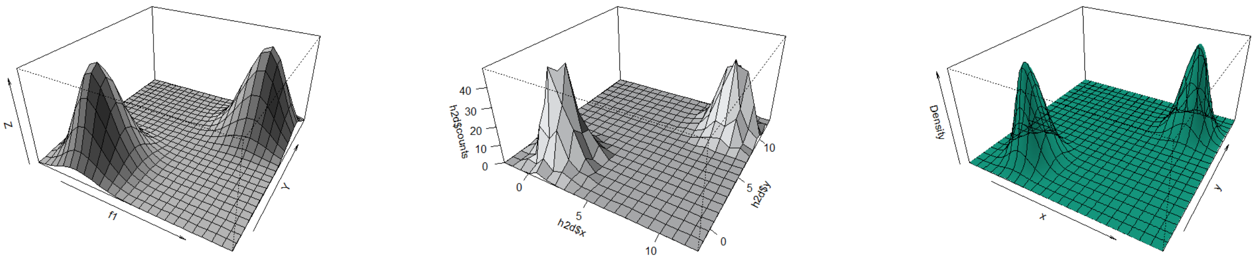


(a) KDE

(b) Histogram

(c) Mixture

Figure 5. Density estimation of data on annulus



(a) KDE **(b)** Histogram **(c)** Mixture

Figure 6. Density estimation of data on 2D two-peak normal distribution

Value	Dist	KDE			Histogram		
		250	500	1000	250	500	1000
Mean	Norm Dist	0.00330655	0.001856879	0.001114913	0.01240743	0.007784372	0.004902357
	Gamma Dist	0.001623677	0.001008014	0.0006296534	0.004633285	0.002939214	0.001778537
	Mix Model	0.001173814	0.0006751656	0.0003913412	0.002911143	0.001713241	0.00119162
Sd	Norm Dist	0.00227001	0.001188117	0.000675403	0.008996144	0.005580659	0.002062684
	Gamma Dist	0.0007094382	0.0004059617	0.0002323864	0.006736886	0.001968476	0.0009768748
	Mix Model	0.001851897	0.0010681	0.000613399	0.1347564	0.0853433	0.0453686

Table 3. Mean and Sd of ISE under KDE and Histogram.

Value	Dist	Mixture Model			Smooth Spline Model		
		250	500	1000	250	500	1000
Mean	Norm Dist	0.001055327	0.0004914699	0.0002539465	0.001876304	0.001013376	0.0005855561
	Gamma Dist	0.003698371	0.002122818	0.001354644	0.001906122	0.001244318	0.0008339185
	Mix Model	9.050128e-5	6.053535e-5	3.624322e-05	0.002837427	0.001638725	0.0009512964
Sd	Norm Dist	0.001320288	0.000552482	0.000269173	0.001520139	0.0008141249	0.0004678311
	Gamma Dist	0.00156573	0.0007910385	0.000468714	0.0009451824	0.0005409505	0.0003333594
	Mix Model	0.00162663	0.001151384	0.0006057706	0.001718881	0.0009530741	0.0005065663

Table 4. Mean and Sd of ISE under Mixture Model and Smooth Spline Model.