KDE Hyperparameter Selection

Liel Gutman 206779829 Yaniv Rotics 314861543

March 2023

1 Abstract

The problem we were trying to solve in our project is the selection of hyperparameters in Kernel Density Estimation, specifically the bandwidth and kernel function. To achieve this, we used several known distributions and 4 different datasets and evaluated the performance of KDE using different bandwidths and kernel functions. We also compared the results to a baseline using a histogram. We found a strong connection between the bandwidth and the accuracy of the estimation, and our solution using grid search cross-validation mostly achieved good results, as well as rules-of-thumbs such as Scott and Silverman. Overall, both methods should be used and compared to a histogram in order to see the smoothness of the estimation.

2 Problem description

Kernel Density Estimation (KDE) is a non-parametric technique used to estimate the probability density function (PDF) of a continuous random variable. In simpler terms, it allows us to estimate the distribution of a dataset without assuming a specific underlying distribution. KDE works by placing a kernel function at each data point and summing the contributions of all the kernels to obtain a smooth estimate of the PDF. The bandwidth parameter in KDE determines the width of the kernel and controls the smoothness of the estimated density function. A small bandwidth will result in a more wiggly (less smooth) estimate, while a large bandwidth will result in a smoother estimate that may over smooth the data. Therefore, selecting the appropriate bandwidth is crucial for obtaining an accurate and informative estimate. The choice of kernel function also plays a significant role in KDE. Commonly used kernel functions include Gaussian, Epanechnikov, and more. The kernel function determines the shape of the kernel and how it weights the contribution of each data point to the density estimate.

The problem in KDE is selecting the optimal bandwidth and kernel function for a given dataset. Selecting sub-optimal values can result in a biased or over-smoothed estimate of the PDF. The bandwidth selection problem is a classic trade-off between bias and variance. A small bandwidth will result in a high variance estimate with low bias, while a large bandwidth will result in a low variance estimate with high bias. Similarly, the kernel function selection problem is a trade-off between smoothness and sensitivity to the data. Therefore, in our project, we aim to improve the selection of the bandwidth and kernel function for KDE. Our goal is to develop a systematic approach to selecting the optimal hyperparameters for KDE that is applicable to a wide range of datasets and underlying distributions.

3 Solution overview

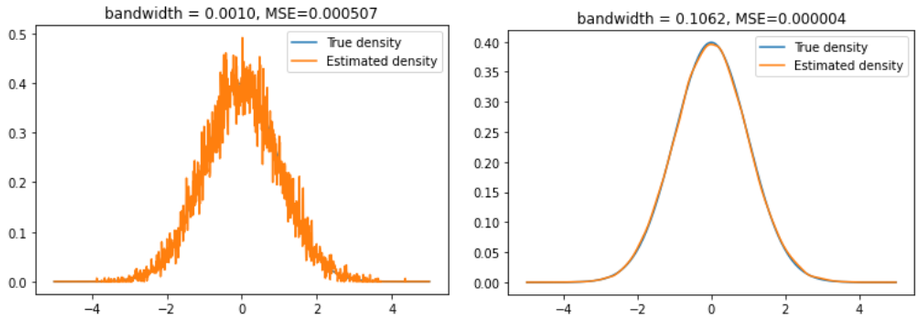
Our proposed solution for selecting the best hyperparameters for KDE includes evaluating different bandwidths including suggested rules-of-thumb, and kernel functions, as well as implementing AutoML using Sklearn's grid search with cross-validation. To begin, we will generate synthetic data from several known distributions. For each distribution, we will compare the different estimations to the true density. This will give us a visual representation of the impact of different hyperparameters on the performance of the KDE model. To evaluate the performance quantitatively, we will use two metrics: mean squared error (MSE) and total log-likelihood. MSE measures the average squared difference between the estimated density and the true density, while total log-likelihood sums the logs of the estimated densities. Using both metrics will hopefully give us a comprehensive understanding of the performance of the KDE model. In addition, we will use the "Scott"[1] and "Silverman"[2] rules-of-thumb for bandwidth selection, which provides simple and widely used heuristics for selecting bandwidth values. We will compare the performance of these rules-of-thumb to the other methods we are testing. To automate the process of finding the best bandwidth, we will use grid search with cross-validation, which is an AutoML method. Grid search involves searching over a hyperparameter space to find the optimal hyperparameters for the model. Cross-validation involves partitioning the data into multiple subsets and using each subset as both a training set and a validation set, to ensure that the model is not overfitting to the data. By using this method, we can explore a large hyperparameter space and estimate the optimal hyperparameters for the KDE model. All estimations will be compared to a baseline using a histogram. Aditionally, we will evaluate the performance of selected methods on 4 different datasets with unknown distributions, and visualize the estimations alongside a histogram. Overall, our proposed solution is a rigorous and systematic approach to selecting the best hyperparameters for KDE, and includes both visual and quantitative evaluations, as well as methods for automation of the hyperparameter selection.

4 Experimental evaluation

Bandwidth selection:

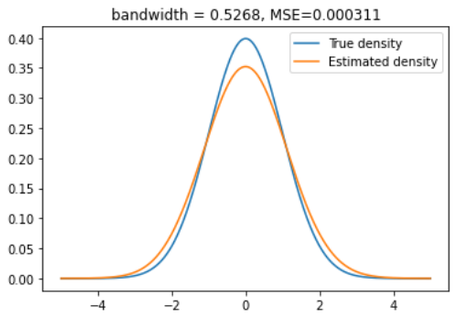
First we generated synthetic data from known distributions, and checked the performance of KDE models with different bandwidths, using MSE (the mean squared error between the true density and the estimated density).

Standard normal distribution:

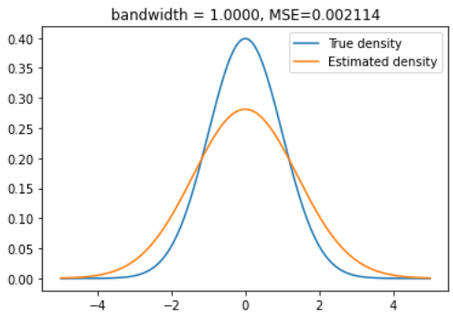


(a) bw = 0.001

(b) bw = 0.106



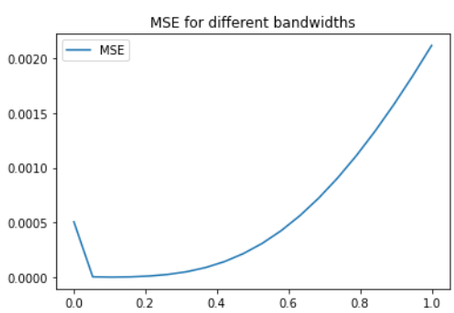
(c) bw = 0.52



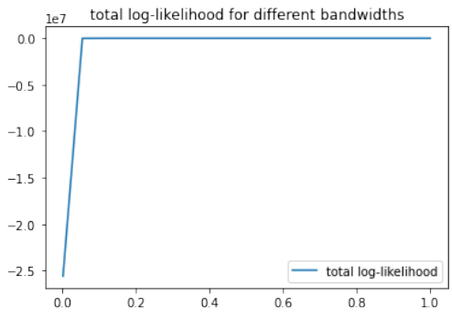
(d) bw = 1.00

Figure 1: Standard normal distribution

As can be seen in Figure 1, there is a strong impact of the bandwidth value on the performance of the KDE model. For bandwidth values that are too low, the KDE model is under-smoothed, and for too-high values the model is over-smoothed. For the standard normal distribution, we got an optimal bandwidth value of around 0.1. In Figure 2(a) you can see the MSE values for selected bandwidths. In addition, we computed the total log-likelihood for the estimated densities which can be seed in Figure 2(b). The KDE model computes the log of the densities, and summing those results gives us this metric, which could indicate the likelihood of the data. A maximum point for this metric is usually achieved around the same area as the MSE metric, therefore it might be a useful metric to use for data with unknown distributions.



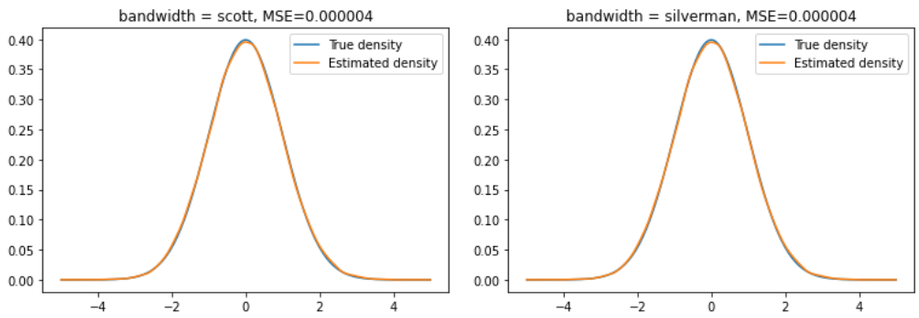
(a) MSE



(b) log-likelihood

Figure 2

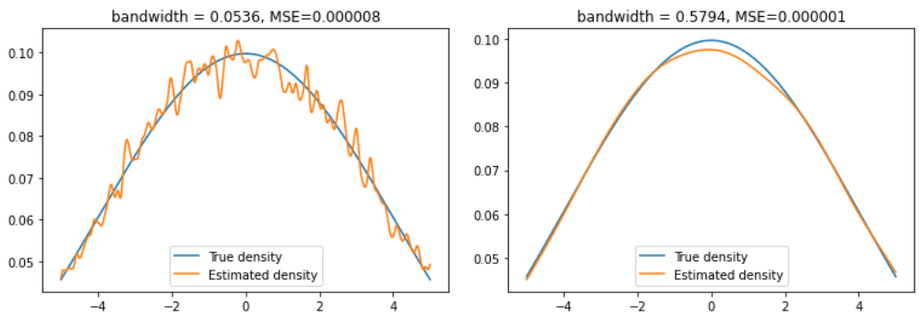
We also tried using known rules-of-thumbs, including “Scott” and “Silverman”. Both achieved good results for the normal standard distribution, but there is no significant difference between Scott and Silverman as can be seen in figure 3.



(a) scott (b) silverman

Figure 3

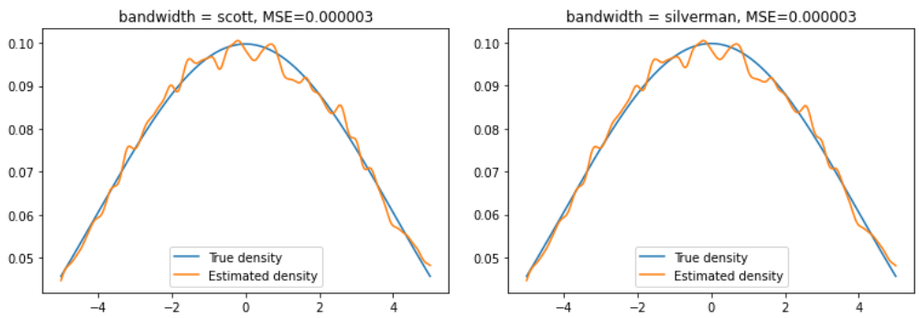
Non-standard normal distribution (std = 4):



(a) bw = 0.0536 (b) bw = 0.579

Figure 4: Non-standard normal distribution (std = 4)

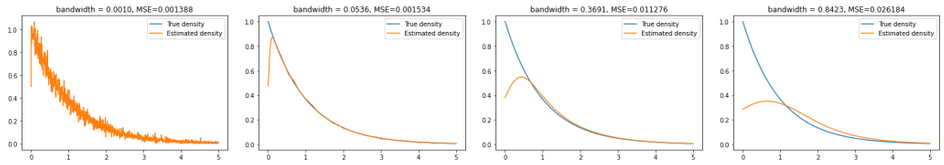
For this distribution, as seen in Figure 4, the optimal bandwidth value was around 0.5 (more graphs can be seen in the notebook). As for Scott and Silverman, as can be seen in Figure 5, these rules-of-thumb achieved under-smoothed estimates for these distributions (but still with a relatively low MSE result). Possibly these rules-of-thumb are more optimal for a standard distribution and we might want to try other methods for some cases.



(a) Scott (b) Silverman

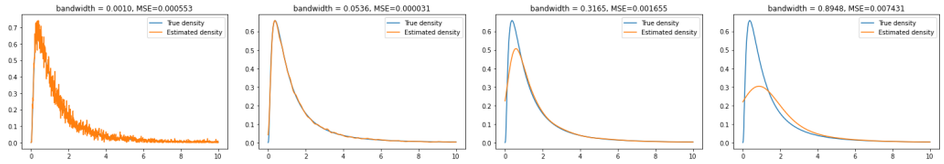
Figure 5

Additionally, we tried the same methods for the exponential and log-normal distributions. Some of the estimations can be seen in Figure 6 and Figure 7.



(a) bw = 0.001 (b) bw = 0.05 (c) bw = 0.36 (d) bw = 0.84

Figure 6: exponential distribution

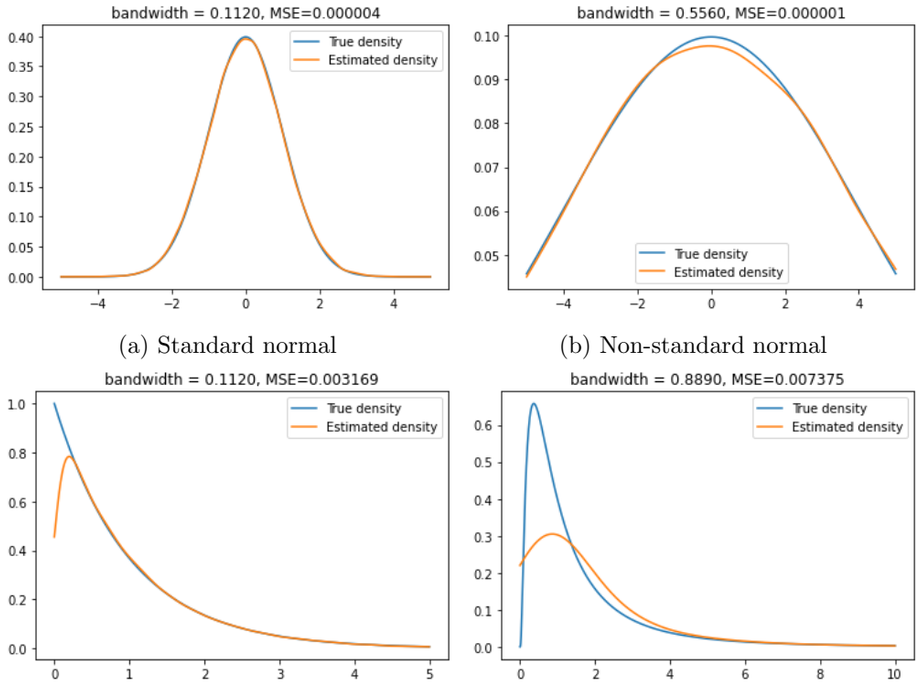


(a) bw = 0.001 (b) bw = 0.05 (c) bw = 0.31 (d) bw = 0.84

Figure 7: log-normal distribution

For the exponential and log-normal distributions, the minimum MSE was around 0.001, even though the estimation is under-smoothed. Scott and Silverman achieved estimations not as good as the normal distribution, but still relatively good (plots available in the notebook). For most bandwidth values in these distributions, the KDE model failed to estimate the density at low values, so possibly KDE has some limitations when used on distributions with high skewness.

We need a method that could be applied to different and unknown distributions, that could automate the process of bandwidth selection. We used grid search cross-validation, which performs a search over a hyperparameter space and estimates the performance on a subset of the data using the total log-likelihood with k-fold cross-validation. The limitation of this method is the long runtime, which increases with a bigger hyperparameter space and a bigger k value.
As can be seen in Figure 8, this method proves helpful for normal-distributed data, as it achieves good results that are not under-smoothed (but could probably be improved with a bigger hyperparameter space). As for the exponential distribution, the estimation is less accurate but not under-smoothed, and compared to the estimations observed before this is a reasonable estimation. For the log-normal distribution, this method resulted in over-smoothing, from which we can conclude that a larger hyperparameter space should be tried and possibly with a higher k-value. Overall, this method could be very helpful for automating the bandwidth selection for certain types of distributions.



(a) Standard normal

(b) Non-standard normal

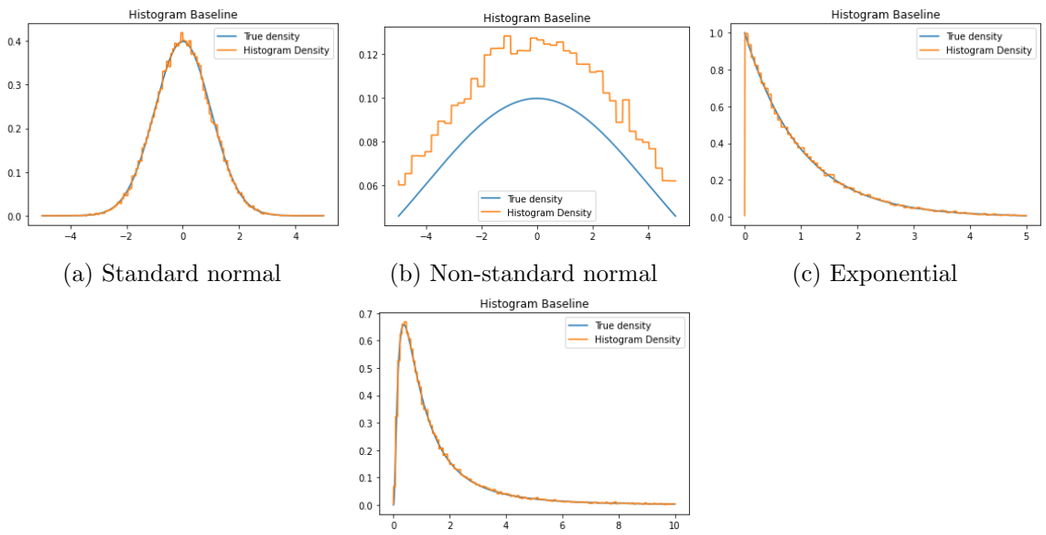
(c) Exponential

(d) Log-normal

Figure 8: grid search cross-validation

Comparison to baseline - Histogram

For each one of the distributions discussed above, we also applied a Histogram model to estimate the PDF and compare it to the real density. We used "auto" for the number of bins. The results can be seen in Figure 9.



(a) Standard normal

(b) Non-standard normal

(c) Exponential

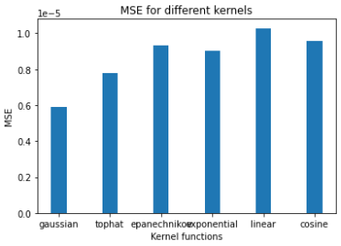
(d) Log-normal

Figure 9: Comparison to baseline - Histogram

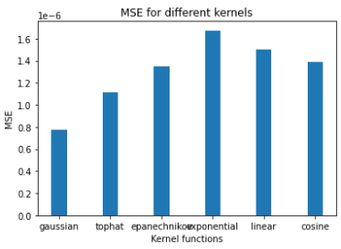
The histogram model achieved reasonable results for all distributions except for the non-standard normal distribution which estimated PDF values relatively higher than the real density. For all distributions, the baseline didn't achieve a better result than the best KDE model, except for the exponential distribution (This is because of the KDE model not estimating properly values close to 0). Mostly the KDE model proves to be much better than the histogram model when using the appropriate bandwidth, since it provides an estimation much closer to the real density and much smoother.

Kernel selection:

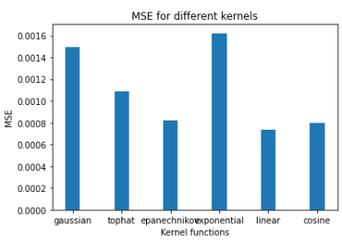
There are multiple possible kernels available for KDE, including “gaussian”, “tophat”, “epanechnikov”, “exponential”, “linear”, and “cosine”. We wanted to check the impact of the kernel function selection on KDE performance. We ran KDE with different kernels on each distribution discussed above. In Figure 10 there are the MSE results.



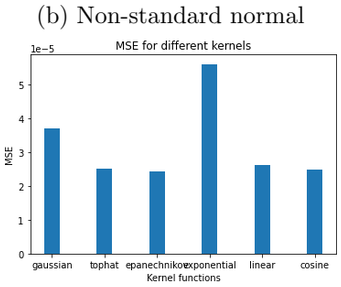
(a) Standard Normal Distribution



(b) Non-standard normal



(c) Exponential

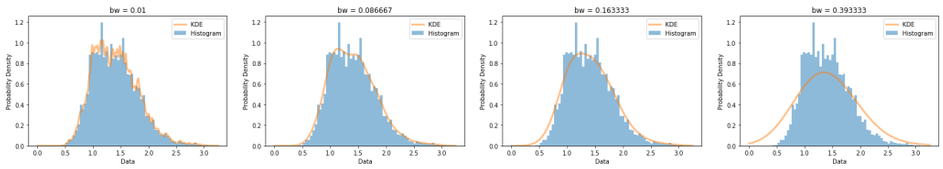


(d) log-normal

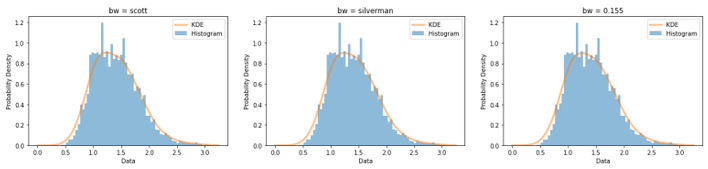
Figure 10

While there wasn’t a single kernel that achieved the minimal MSE in all distributions, the gaussian kernel achieved the most smoothed results (can be seen in the estimation graphs in the notebook). From these observations, we can infer that the differences between the estimations with different kernel functions don’t have a significant impact on the estimation as much as the bandwidth selection, however some are smoother than others and specifically the gaussian kernel would be a good choice for most cases.

After observing the performance of KDE with different methods for the bandwidth selection, we wanted to test our solution for real data from unknown distributions. We started by running KDE on each dataset for different bandwidth values, including scott and silverman, and the bandwidth values received by grid search cross-validation. We graphed the results alongside the histogram for this dataset. For the ”avocado prices dataset” (AveragePrice column) as seen in Figure 11, “Scott”, “Silverman” and grid search cross-validation achieved seemingly good results. For a bandwidth value of around 0.01 the KDE is very close to the data, which seems like overfitting. For higher bandwidth values the estimation is more smooth and still close to the data.



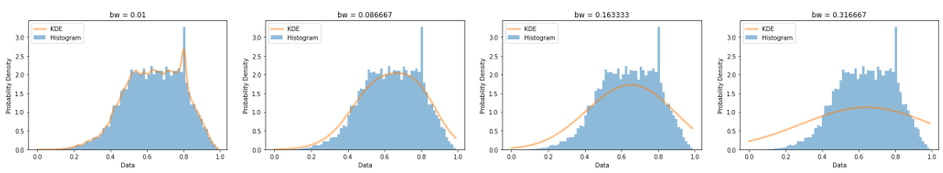
(a) bw = 0.01 (b) bw = 0.087 (c) bw = 0.016 (d) bw = 0.39



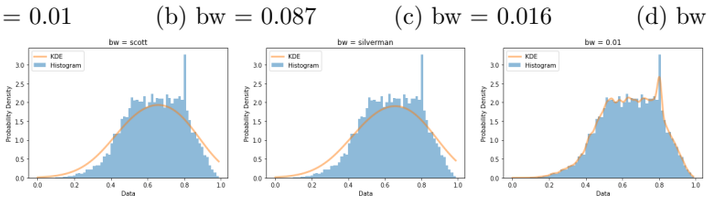
(e) Scott (f) Silverman (g) gridsearch

Figure 11: Avocado prices dataset (AveragePrice column)

For the "Spotify Songs" dataset, as seen in Figure 12, "Scott" and "Silverman" achieved smooth results but perhaps a bit over-smoothed. Grid search cross-validation achieved an estimation that is closer to the data with a bandwidth of around 0.01, but a little under smoothed.



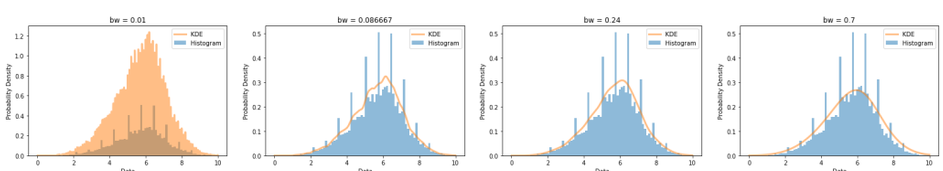
(a) bw = 0.01 (b) bw = 0.087 (c) bw = 0.016 (d) bw = 0.31



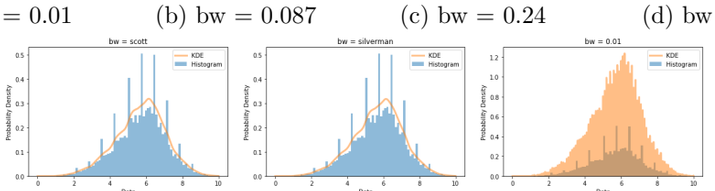
(e) Scott (f) Silverman (g) grid search cv

Figure 12: Spotify Songs dataset (Danceability column)

For the "IMDB action movie ratings" dataset, as seen in Figure 13, "Scott" and "Silverman" achieved seemingly good results with an estimation that is somewhat close to the data, not under-smoothed and not over-smoothed. The grid search cross-validation with a bandwidth of 0.01 achieved bad results that don't seem to be very close to the data.



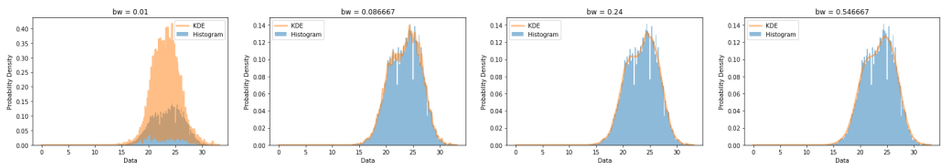
(a) bw = 0.01 (b) bw = 0.087 (c) bw = 0.24 (d) bw = 0.7



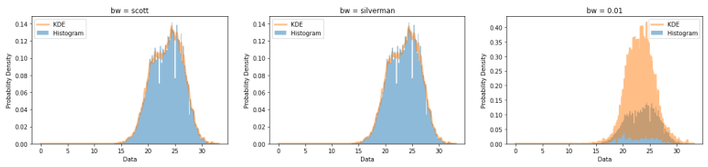
(e) Scott (f) Silverman (g) grid search cv

Figure 13: Action movie ratings IMDB dataset (Rating column)

For the "NASA nearest earth objects" dataset, as seen in Figure 14, grid search cross-validation didn't achieve a good result, while Scott and Silverman achieved better results but still seems under-smoothed.



(a) bw = 0.01 (b) bw = 0.087 (c) bw = 0.24 (d) bw = 0.54



(e) Scott (f) Silverman (g) grid search cv

Figure 14: NASA - Nearest Earth Objects dataset

Additionally to showing the histogram alongside the KDE models with different bandwidths, we split the data to train and test sets, created an estimation model based on the train set and used it to estimate the PDF for the test set. We used the histogram estimator, and KDE with 3 different bandwidth values - Scott, Silverman and the value received from GridSearchCV. Below are the total log-likelihood results for each model, for each dataset.

Total Log-Likelihood values for different models:

dataset	Histogram	KDE (Scott)	KDE (Silverman)	KDE (GridSearchCV)
Avocado prices	−83992.93	−3541.79	−3557.43	−3540.76
Spotify Songs	−171620.13	6358.78	6162.25	7984.17
movie ratings	−132630.66	−20349.98	−20349.503	−4072.16
NASA dataset	−480961.38	−89570.12	−89576.40	−64903.92

Table 1: Table caption

The histogram model achieved a very low total-likelihood value for all datasets. The KDE using grid search cross-validation achieved the maximal total log-likelihood (since this is the metric it is using to estimate its performance), but it's important to mention that this does not indicate a better model, and it's also important to compare the models to the histograms and observe the smoothness of the estimation.

5 Related work

6 Conclusion

In conclusion, our project aimed to improve the selection of hyperparameters in the KDE process. We found that the bandwidth value plays a crucial role in the accuracy of the KDE model's estimation. Additionally, the "Scott" and "Silverman" rules-of-thumb were helpful in achieving good results, particularly for the standard normal distribution, but might generate an under-smoothed estimation for other distributions. On the other hand, our solution using Grid Search cross-validation also achieved good results for most distributions. However, KDE models struggled more with the estimation for very skewed distributions such as exponential or log-normal. Regarding the kernel function selection, we found that there wasn't a significant difference, and the Gaussian kernel usually got more smooth estimations that seem closer to the true density, making it a suitable choice.

When we tested the bandwidth selection methods on four different datasets, we found that the "Scott" and "Silverman" rules-of-thumbs achieved good results. The grid search cross-validation sometimes achieved good results, but other times the estimation was under-smoothed and too far from the histogram estimated density. The main limitation of using KDE with a dataset where we don't know the actual underlying density is that we don't have an accurate way to estimate the performance of the model. Therefore, we recommend trying both grid search cross-validation and the recommended rules-of-thumb as methods of bandwidth selection and comparing them both to a histogram to see the smoothness of the estimation. No method is necessarily better than the other, and it might depend on the dataset and distribution. It is important to mention that the grid search cross-validation is a method that takes longer to run and may be unsuitable if runtime is an important factor.

References

1. Scott, D. (1979). "On optimal and data-based histograms". Link: `https://academic.oup.com/biomet/article-abstract/66/3/605/232642?redirectedFrom=fulltext&login=false`

2. Silverman, B.W. (1986). Density Estimation for Statistics and Data Analysis. Link: `https://archive.org/details/densityestimatio00silv_0/page/45`

3. Avocado Prices Dataset, Kaggle. Link: `https://www.kaggle.com/datasets/neuromusic/avocado-prices`.

4. Dataset of Songs in Spotify, Kaggle. Link: `https://www.kaggle.com/datasets/mrmorj/dataset-of-songs-in-spotify`.

5. IMDb Movie Dataset, Kaggle. Link: `https://www.kaggle.com/datasets/rajugc/imdb-movies-dataset-`

6. NASA - Nearest Earth Objects Dataset, Kaggle. Link: `https://www.kaggle.com/datasets/sameepvani/nasa-nearest-earth-objects?select=neo.csv`.