Kernel Density Estimation

Aiman Aminuddin Bin Azami

November 2, 2020

1 Introduction

The probability density function (PDF) is a relationship between observations and their probabilities. Knowing the PDF of a sample of data is useful as we can determine whether a certain observation is likely or unlikely. Whether, or not a certain observation is an anomaly and should be removed.Normally, It is unlikely that the PDF of random data sample is known.[2]

Hence, we can implement the Kernel Density Estimation (KDE) which is a non-parametric method used to estimate the PDF of a random variable. With just a finite set of data points, KDE generates a smooth curve from such a sample. This allows us to analyse and study the PDF in question.KDE is also known as **Parzel-Rosenblatt window** method in fields such as data science and econometrics. In machine learning, KDE is implemented in clustering problems.[1]

1.1 What is a Kernel?

A kernel is a PDF that is an **even** function. Hence, a kernel must have following properties:

1. $\int K(x)dx = 1$
2. $\forall$x,K(x) ≥ 0
3. $\forall$x,K(x) = K(-x)

1.2 Examples of Kernels

Here are examples of kernel functions:

1. Gaussian K(x) = $\frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}$
2. Uniform K(x) = $\frac{1}{2}\mathbb{1}$(-1 $\leq$ x $\leq$ 1)
3. Epanechnikov K(x) = $\frac{3}{4}\max\{\, 1 - x^2, 0 \,\}$

2 Description of Kernel Density Estimation

Let X be a random variable with continuous distribution F(X) and PDF f(x). We want to estimate f(x) from a random sample $\{X_1, X_2, ..., X_n\}$ The mathematical definition of KDE is the function

$$\hat{f}(x) = \frac{1}{n}\sum_{i=1}^{n} K_h(X_i - x) = \frac{1}{nh}\sum_{i=1}^{n} K\left(\frac{X_i - x}{h}\right) \qquad \text{h} > 0$$

K is the kernel function and the parameter h which is called the bandwidth that controls the amount of smoothing applied to estimate f(x). The kernel acts a weighting function on each data point of f(x). We sum up all the kernels at each data point to generate the KDE of f(x).

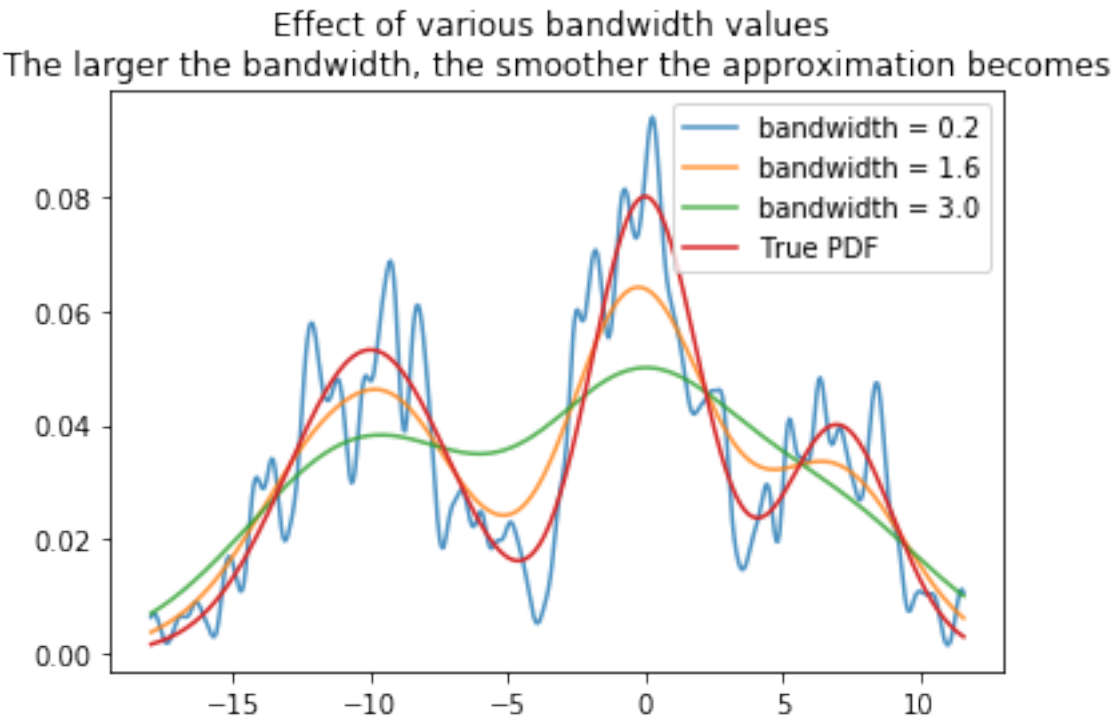

Figure 1: Effect of different bandwidth lengths on KDE estimation of unknown density function f(x).[5]

2.1 Effect of Different Bandwidth Lengths

From Figure 1, we see that h influences the shape of KDE. When h is too small (h = 0.2), our KDE looks very squiggly as a result of under-smoothing. As a result, many of these squiggly structures are caused by random noise. However, if h is too large (h = 3.0), over smoothing will occurred. Important structures eg. local minimums and maximums may be removed causing KDE to be a poor estimate of true PDF.

3 Mathematical Analysis of Kernel Density Estimate

In order to find the optimal KDE of unknown density f, we need to find the h that minimizes the Mean Square Error (MSE) of KDE, $\hat{f}$(x).From the lectures, we have already derive MSE as sum of variance and squared bias. By setting d = 1, we derive the following equation:

$$MSE(\theta) = Var(\theta) + (Bias(\theta))^2 \tag{1}$$

For simplicity, we shall first analyse the KDE of f on a single point by calculating the bias and variance term of KDE at x_0.Suppose that X1,X2,...,Xn are independent and identically distributed (IID) sample from an unknown density function f(x). In this problem, the parameter of interest is f which is the true density function.

3.1 Calculating Bias

The Bias term is as follows:

$$Bias(\theta) = E[\theta] - \theta \tag{2}$$

$$
\begin{aligned}
E[\hat{f}(x_0)] - f(x_0) &= E\left(\frac{1}{nh}\sum_{i=1}^{n} K\left(\frac{X_i - x_0}{h}\right)\right) - f(x_0) \\
&= \frac{1}{h}E\left(K\left(\frac{X_1 - x_0}{h}\right)\right) - f(x_0) \quad \text{property of Expectation and IID} \\
&= \frac{1}{h}\int K\left(\frac{x - x_0}{h}\right) f(x)dx - f(x_0) \\
&= \int K(y) f(hy + x_0)dy - f(x_0) \quad \text{let y} = \frac{x - x_0}{h} \iff \frac{x}{h} = \text{y} + \frac{x_0}{h} \iff \text{dy} = \frac{dx}{h} \\
&= \int K(y)[f(x_0) + hyf'(x_0) + \frac{h^2y^2}{2}f^{(2)}(x_0) + o(h^2)]dy - f(x_0) \\
&= [f(x_0) + o(h^2)]\int K(y)dy + hf'(x_0)\int yK(y)dy + \frac{h^2 f^{(2)}(x_0)}{2}\int y^2K(y)dy - f(x_0)
\end{aligned}
$$

To simplify the Bias term, We use Taylor's Expansion, assuming that f is at least 2 times differentiable, $f(hy + x_0) = f(x_0) + hyf'(x_0) + \frac{h^2y^2}{2}f^2(x_0) + o(h^2)$,$\lim_{h\to 0}\frac{o(h^2)}{h^2} = 0$. Additionally, from 1.1, $\int K(x)dx = 1$. Furthermore, we shall use the fact that K(x) is an even function and x is an odd function thus xK(x) is an odd function. Hence, $\int$ xK(x) dx = 0. We shall also define R = $\int$ y^2K(y) dy. Thus Bias term can be simplified to

$$E[\hat{f}(x)] - f(x_0) = \frac{h^2 f^{(2)}(x_0)}{2}R + o(h^2) \tag{3}$$

From (3), It is easy to see that the Bias term is dependent on the second derivative $f^{(2)}(x)$ or the concavity/convexity of f(x). Hence,if $f^{(2)}(x)$ is large, Bias term will also be large. Visually, it makes sense, as areas where $f(x)$ curves a lot, KDE will try to estimate f(x) by using a smooth curve hence making the estimate less curved thus more biased.

3.2 Calculating Variance

The Variance Term is as follows:

$$Var(\theta) = E[\theta^2] - (E[\theta])^2 \tag{4}$$

Here, we find an upper bound for Variance term in our analysis:

$$
\begin{aligned}
Var(\hat{f}(x_0)) &= Var\left(\frac{1}{nh}\sum_{i=1}^{n} K\left(\frac{X_i - x_0}{h}\right)\right) \\
&= \frac{1}{n^2h^2}Var\left(\sum_{i=1}^{n} K\left(\frac{X_i - x_0}{h}\right)\right) \quad \text{property of variance} \\
&\leq \frac{1}{nh^2}E\left[K^2\left(\frac{X_i - x_0}{h}\right)\right] \quad \text{Variance is non-negative} \\
&= \frac{1}{nh^2}\int K^2\left(\frac{x - x_0}{h}\right) f(x)dx \\
&= \frac{1}{nh}\int K^2(y) f(x_0 + h)dy \quad \text{Here we apply the same trick from 3.1}
\end{aligned}
$$

We use Taylor's expansion again, but this time $f(x_0 + h) = f(x_0) + hyf'(x_0) + o(h)$ where $\lim_{h\to 0}\frac{o(h)}{h} = 0$.

Since K is an even function,K^2 even and hence yK^2 is odd. Thus, the Variance term is reduced to:

$$\begin{aligned} Var(\hat{f}(x_0)) &= \frac{1}{nh}\int K^2\left(f(x_0) + hyf'(x_0) + o(h)\right)dy \\ &= \frac{1}{nh}\left(f(x_0) + o(h)\right)S^2 \quad \text{where } S^2 = \int K^2 dy \end{aligned}$$

Hence, it can be shown that, $Var(\hat{f}(x)) = \frac{f(x_0)S^2}{nh} + o(\frac{1}{nh})$.Additionally, from the Variance term, if $f(x_0)$ large, variance will be large as well.

3.3 Finding h^* that minimizes MSE

We can now compute MSE which is:

$$MSE(\hat{f}(x_0)) = \frac{1}{4}h^4|f^{(2)}(x_0)|^2R^2 + \frac{1}{nh}f(x_0)S^2 + o(h^4) + o(\frac{1}{nh}) \tag{5}$$

As, $o(h^4)$ appears as $0 = \left(\lim_{h^2\to 0}\frac{o(h^2)}{h^2}\right)^2 = \lim_{h^4\to 0}\frac{o(h^2)o(h^2)}{h^4} = \lim_{h^4\to 0}\frac{o(h^4)}{h^4}$ using properties of limits.

We now define the Asymptotic Mean Square Error (AMSE) $= \frac{1}{4}h^4|f^{(2)}(x_0)|^2R^2 + \frac{1}{nh}f(x_0)S^2$. We shall use AMSE to approximate MSE and take the first derivative of AMSE and setting it to 0 to obtain h^*:

$$\begin{aligned} \frac{\partial(AMSE)}{\partial h} = h^3|f^{(2)}(x_0)|^2R^2 - \frac{1}{nh^2}f(x_0)S^2 &= 0 \\ \therefore h^5|f^{(2)}(x_0)|^2R^2 - \frac{1}{n}f(x_0)S^2 &= 0 \\ \therefore h^* &= \left(\frac{f(x_0)S^2}{n|f^{(2)}(x_0)|^2R^2}\right)^{\frac{1}{5}} \end{aligned}$$

3.4 Finding h^* that minimizes Mean Integrated Square Error

From the above analysis, we only minimised MSE of function at x_0. Ideally, we would want to minimise the MSE of the entire function and obtain h^* that minimises this MSE. We can find such a h^* by minimising the Mean Integrated Square Error (MISE) of KDE: $\hat{f}(x)$. We definite MISE of KDE:

$$\begin{aligned} MISE(\hat{f}(x)) &= E\left[\left(\int \hat{f}(x) - f(x)\right)^2\right]dx \\ &= \int E\left[\left(\hat{f}(x) - f(x)\right)^2\right]dx \quad \text{By Fubini's Theorem} \\ &= \int MSE(\hat{f}(x))dx \end{aligned}$$

Side note: Since Expectation can be seen as a form of integral, we are able to applied Fubini's Theorem from Mathematical Analysis that allows for swapping of integral and Expectation. [6]

3.5 Finding h^* that minimizes MISE

We can then find h^* that minimizes MISE of KDE as follows:

$$\begin{aligned} MISE(\hat{f}(x)) &= \int \left(\frac{1}{4} h^4 [f^{(2)}(x)]^2 R^2 + \frac{1}{nh} f(x) S^2 + o(h^4) + o(\frac{1}{nh}) \right) dx \\ &= \frac{1}{4} h^4 R^2 \int |f^{(2)}(x)|^2 dx + \frac{S^2}{nh} \int f(x) dx + \left(o(h^4) + o(\frac{1}{nh}) \right) \int dx \\ &= \frac{1}{4} h^4 R^2 \int |f^{(2)}(x)|^2 dx + \frac{S^2}{nh} + L \left(o(h^4) + o(\frac{1}{nh}) \right) \quad \text{where L is the interval of integration} \end{aligned}$$

MISE can be approximated by the Asymptopical Mean Integrated Square Error (AMISE). We define AMISE:

$$AMISE(\hat{f}(x)) = \frac{1}{4} h^4 R^2 \int |f^{(2)}(x)|^2 dx + \frac{S^2}{nh} \tag{6}$$

3.6 Finding h^* that minimizes AMISE

Here, we shall find h^* that minimizes AMISE as follows:

$$\begin{aligned} \frac{\partial AMISE}{\partial h} = h^3 R^2 \int |f^{(2)}(x)|^2 dx - \frac{S^2}{nh^2} &= 0 \\ \therefore h^5 R^2 \int |f^{(2)}(x)|^2 dx - \frac{S^2}{n} &= 0 \\ \therefore h^* = \left[\frac{S^2}{nR^2 \int |f^{(2)}(x)|^2 dx} \right]^{\frac{1}{5}} & \end{aligned}$$

In theory, it is possible to obtain the explicit formula for the optimal bandwidth: h^* as as shown above. However, in practice we are unable to use h^* as $f(x)$ is unknown and h^* contains the $\int |f^{(2)}(x)|^2 dx$ term. In fact, finding the optimal h (bandwidth selection) is an unsolved statistic problem. Hence, approximation methods are needed to obtain optimal bandwidth.

4 Bandwidth Selection

4.1 Silverman's Rule-of-Thumb

Assuming that f is a normal density function and that the choice of kernel is the Gaussian Kernel. Define h_{rule} to be the optimal bandwidth.Then, the Silverman's Rule-of-Thumb bandwidth is

$$h_{rule} = \left(\frac{4\hat{\sigma}^5}{3n} \right)^{\frac{1}{5}} \approx 1.06 \hat{\sigma} n^{-\frac{1}{5}} \tag{7}$$

From Figure 2, a dataset that has a unimodal distribution that is close to normal distribution is generated from Python. We see that using the Silverman's Rule-of-Thumb Bandwidth allows KDE to estimate the unknown density function well. Of course, if we know that f is a normal density or approximately normal we would not need to use KDE to estimate f. From Silverman(1986), h_{rule} works well if f is a normal density function. Suppose the distribution in question is multimodal or heavily skewed. KDE gives a poor estimation of unknown density function f as over smoothing occurs. On the contrary, if h_{rule} is rewritten in terms of Interquartile Range(IQR) of the normal distribution in question:

$$h_{rule} = 0.79 \, (IQR) \, n^{-\frac{1}{5}} \tag{8}$$

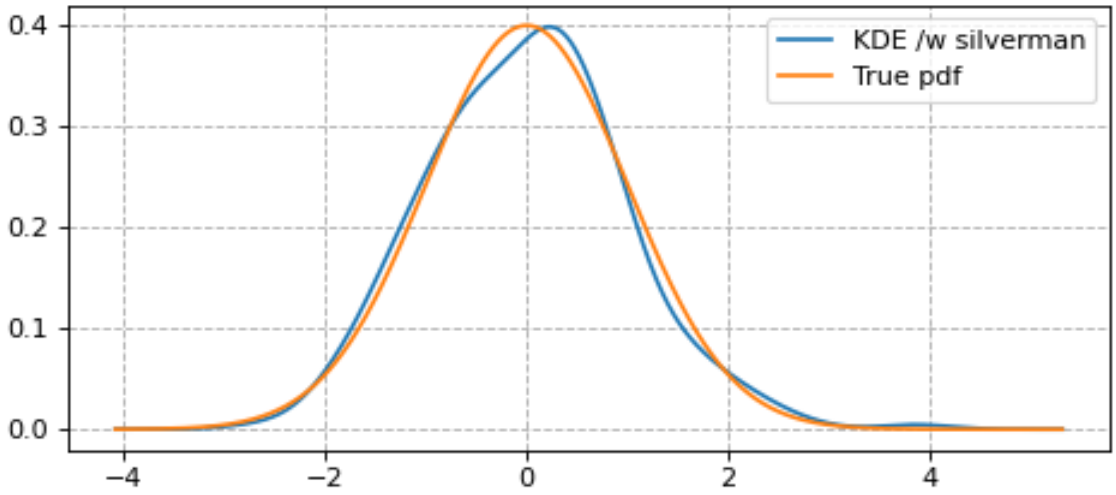

Figure 2: Estimating unknown density function using Silverman's Rule-of-Thumb Bandwidth[8]

The new h_{rule} improves the kernel density estimation of long-tailed and skewed distribution. However, if the distribution in question is bimodal , the estimation actually worsens as the KDE of f is more oversmoothed. We define $A = min(\sigma, \frac{IQR}{1.34})$ and let:

$$h_{rule} = 1.06An^{-\frac{1}{5}} \tag{9}$$

The new h_{rule} allow KDE to estimate unimodal densities well and will not perform too badly if the distribution is moderately bimodal. Consequently, h_{rule} can allow better KDE estimates if 1.06 is reduced. If f is a normal density function, we reduce 1.06 to 0.9 such that:

$$h_{rule} = 0.9An^{-\frac{1}{5}} \tag{10}$$

4.2 Maximum Likelihood Cross-Validation (MLCV)

Cross-Validation is an example of a model validation technique. The goal of Cross-Validation is to test the model's ability to predict new or unseen data that was not used in the model building process. This is done to flag problems such as overfitting. From Figure 3 (suppose we have 10 subsets of data), the steps involved in Cross-Validation includes:

1. Splitting your data set into smaller data subsets
2. Reserving one subset (validation set) and train your model using the remaining data set (training sets)
3. Testing your model on the reserved data set (unseen data)
4. Repeating Step 1-3 for each subset

MLCV is an example of a Leave-One-Out Cross-Validation bandwidth selector. This method was proposed by Hobbema, Hermans, and Van den Broeck (1971) and by Duin (1976). They proposed to choose the bandwidth that maximises the pseudo-likelihood function $\prod_{i=1}^{n} \hat{f}(x)$. However, from the definition of KDE, we can trivially choose $h = 0$ to maximise the pseudo-likelihood function. Firstly, we shall replace $\hat{f}(x)$ with $\hat{f}_{i,j}(x)$:

$$\hat{f}_{i,j}(X_i) = \frac{1}{(n-1)h} \sum_{i \neq j} K\left(\frac{X_i - X_j}{h}\right)$$

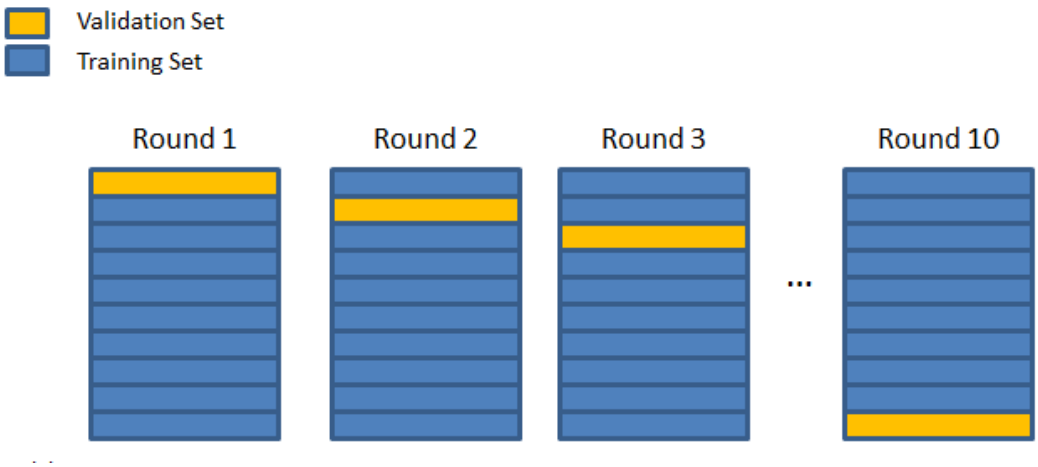

Figure 3: The essential steps of Cross-Validation[10]

From lectures, maximising the likelihood function is equivalent to maximising the log-likehood function.Here we shall define the natural log to be log. Thus, we want the optimal $h_{mlcv} = \arg\max_{h>0} \frac{1}{n} \log \left[\prod_{i=1}^{n} \hat{f}_{i,j}(X_i)\right]$ where:

$$\begin{aligned}
\frac{1}{n} \log \left[\prod_{i=1}^{n} \hat{f}_{i,j}(X_i)\right] &= \frac{1}{n} \sum_{i=1}^{n} \log \left[\hat{f}_{i,j}(X_i)\right] \\
&= \frac{1}{n} \sum_{i=1}^{n} \log \left[\frac{1}{(n-1)h} \sum_{i \neq j} K\left(\frac{X_i - X_j}{h}\right)\right] \\
&= \frac{1}{n} \sum_{i=1}^{n} \log \left[\sum_{i \neq j} K\left(\frac{X_i - X_j}{h}\right)\right] - \log\left[(n-1)h\right] \\
&= MLCV(h)
\end{aligned}$$

In Figure 4, we compare the KDE obtain from the inbuilt density function in Rstudios and KDE obtained using the h_{mclv} that maximises MCLV. The Kernel used is a Gaussian Kernel which is:

$$K(x) = \frac{1}{h\sqrt{2\pi}} e^{-0.5\left(\frac{x - x_i}{h}\right)^2}$$

We note that KDE obtained from using h_{mclv} is similar to KDE obtained using the inbuilt density function from Rstudios.

5 Extending KDE to Higher Dimensions

5.1 Multivariate Kernel Density Estimation

KDE can be extended to estimate multivariate density function in $\mathbb{R}^d$. Suppose again we have a random sample $\{\boldsymbol{X_1}, \boldsymbol{X_2}, ..., \boldsymbol{X_n}\}$ from unknown distribution F with multivariate density function $f(\boldsymbol{x})$ where $\boldsymbol{x} \in \mathbb{R}^d$. Then the Multivariate KDE of f is:

$$\hat{f}(\boldsymbol{x}) = \frac{1}{nh^d} K\left(\frac{\boldsymbol{X_i} - \boldsymbol{x_i}}{h}\right) \tag{11}$$

For the multivariate case, the Kernel function satisfy these conditions where $\boldsymbol{u} \in \mathbb{R}^d$:

1. $K(\boldsymbol{u}) \geq \boldsymbol{0}$

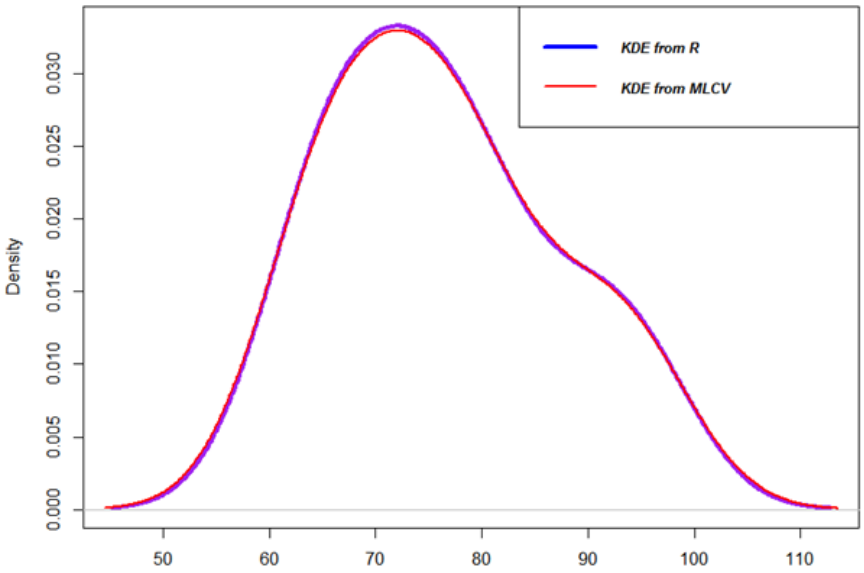

Figure 4: Comparison between inbuilt KDE function from Rstudios and MLCV[12]

2. $\int_{\mathbb{R}^d} K(\boldsymbol{u})d\boldsymbol{u} = 1$
3. $\int_{\mathbb{R}^d} \boldsymbol{u}K(\boldsymbol{u})d\boldsymbol{u} = \boldsymbol{0}$
4. $\int_{\mathbb{R}^d} \boldsymbol{u}\boldsymbol{u}^T K(\boldsymbol{u}) = I_d$

Additionally, the Kernel for the multivariate case is usually obtained as a product of Kernels from univariate case (K_{univ}) i.e.

$$K(\boldsymbol{u}) = \prod_{i=1}^{d} K_{univ}(u_i) \quad \boldsymbol{u} = (u_1, ...u_d) \tag{12}$$

We can also obtain the optimal bandwidth by first finding MISE and taking the first derivative of MISE and setting it to 0. However, we shall not go through the multivariate case due to space constraints.

5.2 Curse of Dimensionality

While Kernel Density Estimation is widely-used as non-parametric method in estimating unknown density functions in many fields, it is largely effected by the "Curse of Dimensionality". The "Curse of Dimensionality", which was coined by Richard Bellman, indicates that the number of samples needed to estimate an arbitrary function with a certain level of accuracy grows exponentially with respect to the number of input variables[14] i.e dimension d. According to a paper by Scott and Wand (1991), the quality of Multivariate KDE estimation start to deteriorate as dimension d starts to get larger and larger. Furthermore, in order, to maintain a certain level of accuracy, the sample size needs to increase as d increases.[15] Of course, there exists many dimensionality reduction algorithms such as the Fast Fourier Transform to mitigate this phenomena. However, this topic is too advanced and beyond the scope of MA4270.

6 Applications of Kernel Density Estimation

6.1 Clustering

In Machine Learning, the goal of Clustering is to group your data from your data set into disjoint groups known as clusters where data points in the same clusters are more similar to each other as compared to data points from other clusters. There are algorithms that utlizes the KDE to tackle Data Clustering problems in Machine Learning.

6.2 Mean Shift Algorithm

The objective of the Mean Shift Algorithm is to locate the maximas or modes of a density function from a given set of data points. For the multivariate KDE, this algorithm generates a smooth KDE surface. The algorithm will then iteratively "push" each data point uphill until all the data points are at the top of the nearest local maximum of the surface thus forming different data clusters. Other Clustering methods such as K-Means require the number of clusters to be generated to be pre-determined. This may be problematic as,in practice, it is often difficult to choose the right number of clusters to used. There are situations where we expect certain data points to cluster together due to some similar features. However, because of the predetermined number of clusters in K-Means, such data points could be seperated into different clusters. However, Mean Shift Algorithm avoids this problem by utilizing KDE and pushing data points to different modes of PDF to generate clusters. On the contrary, it is worth pointing out that it has a very large time complexity and thus computationally expensive. In fact, it is an $O(N^2)$ algorithm where N refers to the number of data points. If N is large (let say a billion), the mean shift algorithm will calculate $(10^9)^2$ operations.

6.3 Steps of Mean Shift Algorithm

Here, we shall give a brief procedure of the Mean Shift Algorithm:

1. For each data point $x \in D$, find the neighbouring points: $N(x)$ of x
2. For each data point $x \in D$, calculate:

$$m(x) = \frac{\sum_{x_i \in N(x)} K\left(x_i - x\right) x}{\sum_{x_i \in N(x)} K\left(x_i - x\right)} \tag{13}$$

3. For each data point $x \in D$, update x $\leftarrow$ m(x), (Note: $m(x) - x$ is known as mean shift) [19]
4. Repeat step 1 to 3 for n iterations or until point x do not move or almost not moving

The algorithm terminates when $m(x) = x$.[19] When this occur, we say that $m(x)$ converges. It is also worth noting that the sequence, $x, m(x), m(m(x)), ...$ is called the trajectory of x.[19]

7 Conclusion

To conclude, in this paper, we have discussed in great length on the properties of Kernel Density Estimation and have extended KDE to the multivariate case and touch on the application of KDE in Clustering Problems. While KDE may be useful, there are pitfalls that one might face such as dealing with the "Curse of Dimensionality" when dimension d is large. Hence, having multiple approaches might be ideal to tackling variety of machine learning problems.

8 References

1. Kernel Density Estimation: Wikipedia https://en.wikipedia.org/wiki/Kernel_density_estimation
2. A Gentle Introduction to Probability Density Estimation https://machinelearningmastery.com/probability-density-estimation/
3. Exploratory Data Analysis: Kernel Density Estimation Conceptual Foundations The Chemical Statistician https://chemicalstatistician.wordpress.com/2013/06/09/exploratory-data-analysis-kernel-density-estimation-in-r-on-ozone-pollution-data-in-new-york-and-ozonopolis/
4. Yen-Chi Chen (2017) Lecture 7: Kernel Density Estimation. Personal Collection of Yen-Chi Chen, University of Washington (2017). STAT/Q SCI 403: Introduction to Resampling Methods, Lecture 7: Kernel Density Estimation
5. Kernel Density Modality Test Github https://github.com/ciortanmadalina/modality_tests/blob/master/kernel_density.ipynb
6. Scott, D. W., and Sain, S. R. (n.d.). Multi-Dimensional Kernel Density Estimation.
7. Silverman, B.W. (1986). Density Estimation for Statistics and Data Analysis. London: Chapman Hall/CRC. p. 45-47
8. Bandwidth- KDEpy 1.0.7 documentation https://kdepy.readthedocs.io/en/latest/bandwidth.ht
9. Cross-Validation-Wikipedia https://en.wikipedia.org/wiki/Cross-validation_(statistics)
10. Cross Validation Explained: Evaluating Estimator Performance https://towardsdatascience.com/cross-validation-explained-evaluating-estimator-performance-e51e5430ff85
11. Guidoum, A. C. (2015, October 30). Kernel Estimator and Bandwidth Selection for Density and its Derivatives. Retrieved October 31, 2020, from https://cran.r-project.org/web/packages/kedd/vignettes/kedd.pdf
12. Kernel Construction and Bandwidth Optimization using Maximum Likelihood Cross Validation- Medium https://medium.com/analytics-vidhya/kernel-density-estimation-kernel-construction-and-bandwidth-optimization-using-maximum-b1dfce127073
13. Buhlmann, P., amp; Machler, M. (2016, October 12). Computational Statistics. Retrieved October 31, 2020, from https://stat.ethz.ch/lectures/ss19/comp-stats.php#literature
14. Bellman R.E. Adaptive Control Process. Princeton University Press, Princeton, NJ, 1961.
15. Crabbe, J. J. (2013, July). Handling the Curse of DImensionality in Multivariate Kernel Density Estimation [Scholarly project]. Retrieved October 31, 2020, from https://shareok.org/bitstream/handle/11244/11009/Crabbe_okstate_0664D_12854.pdf?sequence=1
16. Mark A. Davenport (2017) Lecture 19: Kernel Density Estimation K-means. Personal Collection of Mark A. Davenport, Georgia Institute of Technology (2017). ECE6254: Statistical Machine Learning Lecture 19: Lecture 19: Kernel Density Estimation K-means.
17. Mean Shift Clustering Overview- Atomic Spin https://spin.atomicobject.com/2015/05/26/mean-shift-clustering/
18. Meanshift Algorithm for the Rest of Us (Python) - http://jamesxli.blogspot.com/2012/03/on-mean-shift-and-k-means-clustering.html
19. Leow Wee Kheng. Mean Shift Tracking. Personal Collection of Leow Wee Kheng, National University of Singapore. CS4234 Computer Vision and Pattern Recognition Lecture: Mean Shift Tracking.