Maximum Likelihood Cross Validation for Bandwith Selection in Kernel Density Estimation

Théo Bacqueyrisse and Maria Bracque Vendrell

Semester 1 - 2023

Contents

1 Introduction

Kernel Density Estimation (KDE) is a non-parametric method used for estimating probability density functions from observed data. Cross-validation is one of the most used techniques for bandwidth selection in KDE since it ensures robustness and prevents overfitting.

In a nutshell, cross-validation is a statistical technique for assessing and comparing learning algorithms by dividing the data into two subsets: a training set and a testing set. The training set is used to train a model, and the testing one is used to validate the model. What is relevant about the cross-validation technique is that there are successive crossover between training and validation sets in each round ensuring that each data point has a chance of being validated against.

The choice of bandwidth influences the smoothness of the estimated density and, consequently, the model's ability to capture underlying patterns without being overly sensitive to noise. This is why, it is important to select the correct bandwidth when doing kernel density estimation. In the case of KDE, as previously mentioned, cross-validation leads to a data-driven choice of bandwidth h decreasing the risk of overfitting and ensuring the generalization of the model.

There are different cross-validation methods, but in this paper, we are going to focus on Maximum Likelihood Cross-Validation (MLCV). We are going to apply this technique to footballers *heights* data. We obtained a bandwidth of around 1.15 with the normal scale rule and of around 1.55 with the MLCV method.

Thus, we are going to first write a short literature review to have an overview of the development of ML cross-validation and its field. Then, we will have a look at the methodology behind it, and how it works. And finally, we will apply this method to real-life data.

2 Literature Review

The basic idea of kernel density estimation is to estimate the density function at a point x using neighboring observations (Zambom and Dias 2012). As we previously mentioned, a very important part of this estimation is choosing the adequate bandwidth h, as it significantly influences the smoothness and accuracy of the estimated density. The methods used for selecting this bandwidth can be divided into two main categories: classical and plug-in methods. On one hand, plug-in methods automate the bandwidth selection process by using initial estimates to guide the choice of bandwidth. On the other hand, classical methods, including cross-validation, Mallow's Cp, and Akaike's Information Criterion (AIC), are more or less natural extensions of methods used in parametric modeling (Loader 1999).

Maximum likelihood cross-validation (MLCV) is considered a classical method. It was first introduced by Hobbema, Hermans, and Van den Broeck in 1971 and further developed by Duin in 1976. Without going into the mathematical specifications, we will see them in the next section, its objective is to optimize the bandwidth parameter in a statistical model by maximizing the likelihood of the observed data given the model. The likelihood function represents the probability of observing the given data under the assumed statistical model. In the case of KDE, the likelihood function quantifies how well the chosen bandwidth parameter explains the observed data. MLCV incorporates a cross-validation framework to assess the performance of different bandwidth values. The dataset is typically divided into training and validation sets, and the likelihood is evaluated for each candidate bandwidth. The bandwidth that maximizes the average likelihood across all validation sets is considered optimal.

Concerning the performance of MLVC compared to other cross-validation methods, Horne and Garton 2006 found that in their simulations, likelihood cross-validation generally performed better than least squares cross-validation, producing

estimates with better fit and less variability. It was especially beneficial for sample sizes less than 50.

Van Es in 1991 also observed that the performances of Maximum Likelihood cross-validation could suffer from heavy or long-tailed distributions. This means that normally distributed variables are more likely to be better estimated with this method (Van Es 1991).

Now, let us move to the next section where we are going to discuss the methodological aspect of KDE and MLCV.

3 Methodological sheet

When performing an estimation of the density of a sample $(X_1, ..., X_n)$, a natural method is to use Kernel Density Estimation, which allows the estimate to be stable and smoother than a histogram estimator.

The formula for the Kernel Density estimate is the following :

$$f_{nh}(x) = \frac{1}{nh} \sum_{j=1}^{n} K(\frac{x - X_j}{h})$$

where K is the kernel function used, and h is the bandwidth used to control the width of the kernel function, thus controlling the smoothness of the estimation. A large bandwidth will result in a very smooth estimator, but that will possibly fail to capture some variations in the data. A small bandwidth will result in an estimator very well fitted to the data, but possibly lacking smoothness, which makes the estimation hard to generalize to new data.

When using this estimator, choices have to be made for the Kernel function, and the bandwidth to use for computation. In general, the choice of kernel function has a small impact on the quality of the estimator, even if the Epanechnikov kernel function gives the best results in terms of Asymptotic Mean Integrated Squared Error (AMISE), defined as :

$$K(u) = \frac{3}{4} \cdot \left(1 - u^2\right)$$

However, the choice of bandwidth is crucial to determine the quality of the estimator, as an optimal bandwidth will result in a smooth estimator able to capture variations of the data well enough.

This is why, in Kernel Density Estimation, the use of cross-validation for bandwidth selection is very popular. Cross-validation techniques are suited to this type of problem, as they will divide the data into training and testing samples iteratively, to assess the model performance, and arrive at an optimal solution suited to predict the known data, but also efficient for new data generalization.

In our project, we will focus on the Maximum Likelihood cross-validation method for bandwidth selection. This method is known to be efficient on large datasets, even if computationally expensive depending on the size of the data. It however can perform poorly on data with heavy or long tail distributions. We will then later work with a dataset that is suited to these properties.

The idea of Maximum Likelihood is to maximize the following function :

$$\prod_{i=1}^{n} \hat{f_{-i,h}}(X_i)$$

with respect to h, which is the likelihood of the sample $(X_1, ..., X_n)$ with the Leave-One-Out estimator. The Leave-One-Out estimator has the following formula :

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

This means that we do not use the i^{th} observation of the data. Indeed, maximizing the likelihood with the full data would result in an evident optimal solution at h = 0.

Maximizing this quantity is, as we know by passing to the log, equivalent to maximizing :

$$MLCV(h) = \frac{1}{n}\sum_{i=1}^{n} log(\hat{f_{-i,h}}(X_i))$$

which is the log likelihood of the sample with the Leave-One-Out estimator. If we replace in this equation the value of the Leave-One-Out estimator, we obtain the value :

$$MLCV(h) = \frac{1}{n}\sum_{i=1}^{n}[log(\sum_{j\neq i} K(\frac{X_j - X_i}{h})) - log((n-1)h]$$

This is the quantity we want to maximize with respect to h.

Maximizing $MLCV(h)$ with respect to h leads to finding an estimator that minimizes the Kullback-Leibler information distance, defined for two continuous variables p and q, by :

$$D_{KL}(p \parallel q) = \int p(x) \cdot \log\left(\frac{p(x)}{q(x)}\right) dx$$

This relative distance measures the similarity between two probability distributions. So, taking $h^* = argmax_h MLCV(h)$ ensures that the estimator $\hat{f}_h(x)$ and the actual density $f(x)$ are as close as possible by the Kullback-Leibler distance.

4 Real-Life Application

4.1 Data

The dataset we used is from Kaggle and it is called "Football Players' Transfer Fee Prediction Dataset". It was generated by scraping data from Transfermarkt using Selenium and BeautifulSoup on June 10, 2023. The data is composed of 10 754 football players and contains their characteristics such as their height, their

age, their name, their position, their team, the number of times a player appears on the field, the number of goals, the number of assists to goals, and the number of yellow cards he had.

The objective of this section is to estimate the density of the *height* variable.

4.2 Estimation using MLCV

First, we have a look at the summary statistics for the variable *height* which is our variable of interest. We obtain the following results:

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
156.0	176.0	181.2	181.2	186.0	206.0

Table 1: Summary Statistics for the variable *height*

Following, there is the boxplot for a more visual representation of the summary statistics for *height*.

Boxplot of the variable *height*

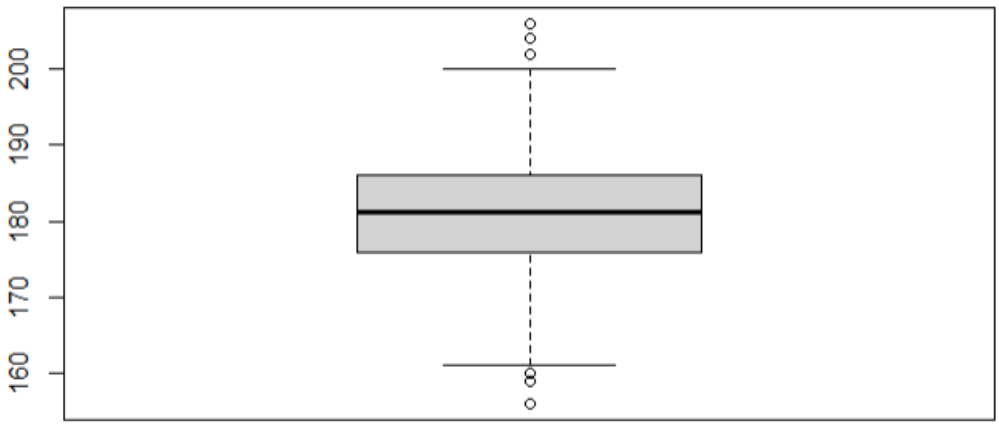

Figure 1: Boxplot of the variable *height*

We notice that the mean is close to the median and that the boxplot is symmetric. This suggests that the variable *height* has a relatively symmetric distribution.

Before starting the estimation, let us have a look at the histogram distribution.

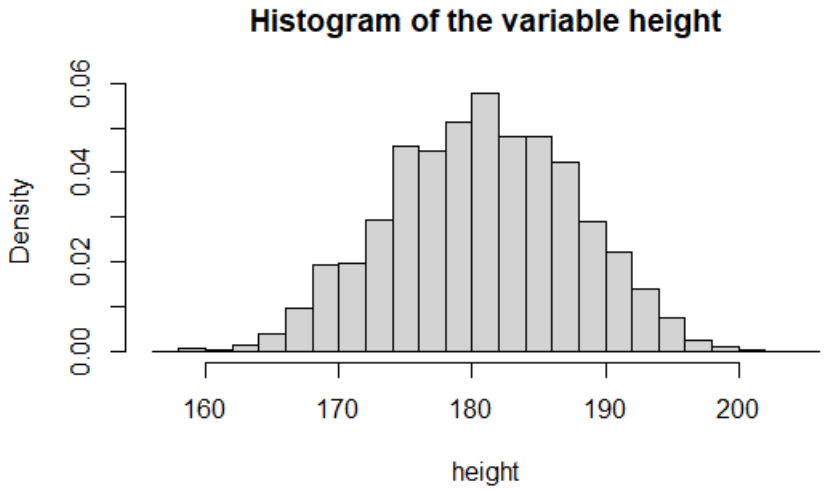

Figure 2: Histogram of the variable *height*

This histogram shows us that the variable *height* seems to be close to normally distributed, which means that the use of maximum likelihood cross-validation for bandwidth selection may be relevant. Indeed, as we said before, this technique is appropriate for distributions without heavy tails and performs well with large datasets.

Now, we will estimate the density of our variable using Kernel Density Estimation. The goal is to first perform a bandwidth selection and then apply the kernel density estimation

First, let us use the Normal Scale Rule of thumb for bandwidth selection, using the following formula :

$$h_{nsr} = 1.059 * \sigma * n^{-\frac{1}{5}}$$

with sigma being the standard deviation of the *height* variable in the data. We use this 1-dimensional formula since we only work with one variable.

In R, computing this formula yields: $h_{nsr} \approx 1.153$.
This result will serve as a benchmark to compare the results of our maximum likelihood cross-validation bandwidth selection.

We used the *np* R package that allows us to compute h using different techniques. Using the *npudensbw* function, setting the *bwmethod* parameter to *cv.ml* and using an Epanechnikov kernel, we obtained: $h_{mlcv} \approx 1.552$.

Now that we have our computed bandwidth, we can plot the Kernel Density Estimations obtained with the function *bkde* of the package *KernSmooth* of R. We obtain the following plot :

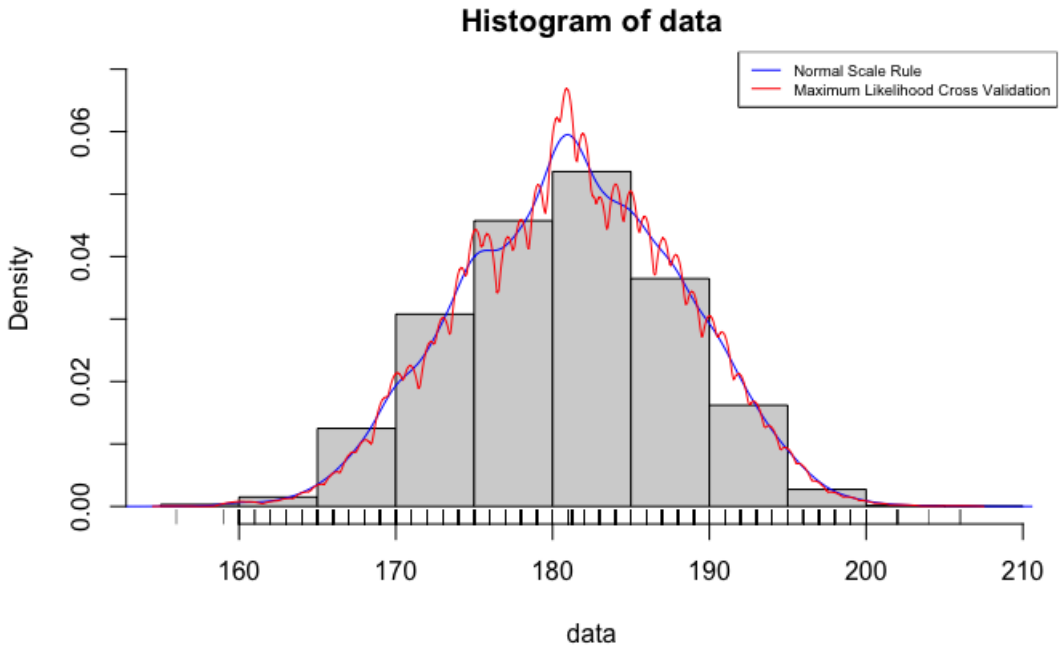

Figure 3: Kernel Density Estimations of the variable *height*

We can see in this plot that the bandwidth computed with maximum likelihood cross-validation yields a density that is less smooth than with the Normal Scale Rule. It however enables us to capture specific noises of the data. For example, we see that a drop in density is seen around the value 177, which the Normal Scale rule density does not quite capture.

The oscillations in the estimation can be explained for two reasons:

- First, since our variable is only discrete and takes integer values. This characteristic can naturally lead to fluctuations in the density between these values.
- Second, with over 10,000 observations, our dataset is quite extensive. This large size could allow the estimation process to accurately capture the actual behavior of the data, which may be reflected in the oscillations.

With the many oscillations in this estimation, it could be possible that the data was overfitted, but we think it is rather due to the nature of the data with discrete values as explained before.

5 Conclusion

To sum up, this report explored the use of maximum likelihood cross-validation (MLCV) for kernel density estimation (KDE) in analyzing football players' heights.

We started with an overview of KDE and MLCV in which we highlighted the importance of bandwidth selection. We also compared MLCV to other cross-validation methods and found that the MLCV generally performs better than least squares cross-validation. The methodological section outlined the mathematical foundations of KDE and MLCV.

We then applied the concepts described to real-life data of football players' characteristics, focusing on height density estimation. We started with a preliminary data exploration with some summary statistics, a boxplot, and a histogram for the variable *height* and found that this variable seemed to have a distribution close to a normal distribution. As a benchmark, we decided to use the Normal Scale Rule of thumb, and we found $h = 1.153$. Then, with the MLCV method, we found $h_{mlcv} = 1.552$ by using the *np* package on R and an Epanechnikov kernel.

In summary, the MLCV method introduced more oscillations in the density estimation, likely capturing specific features overlooked by the Normal Scale Rule. We attributed these oscillations to the discrete nature of the height variable and the large dataset size.

Bibliography

Broniatowski, Michel. 1993. "Cross Validation Methods In Kernel Non Parametric Density Estimation: A Survey." Ffhal-03664840f, *Annales de l'ISUP*.

Guidoum, Arsalane Chouaib. 2020. "Kernel Estimator and Bandwidth Selection for Density and its Derivatives: The kedd Package," arXiv: 2012.06102 [stat.CO].

Horne, Jon S, and Edward O Garton. 2006. "Likelihood Cross-Validation Versus Least Squares CrossValidation for Choosing the Smoothing Parameter in Kernel Home-Range Analysis." *Journal of Wildlife Management* 70 (3): 641–648. %5E1%5E.

Huynh, Khang. 2023. "Football Players' Transfer Fee Prediction Dataset." Dataset for predicting football players' transfer fees, available on Kaggle. Accessed December 12, 2023. %5E1%5E.

Loader, Clive R. 1999. "Bandwidth selection: classical or plug-in?" *The Annals of Statistics* 27 (2): 415–438. https://doi.org/10.1214/aos/1018031201. https://doi.org/10.1214/aos/1018031201.

Niranjan Pramanik, Ph.D. 2019. "Kernel Density Estimation: Kernel Construction and Bandwidth Optimization using Maximum Likelihood Cross Validation." Accessed December 12, 2023. %5E1%5E.

Van Es, Bert. 1991. "Likelihood cross-validation bandwidth selection for nonparametric kernel density estimators." *Journal of Nonparametric Statistics* 1:83–110.

Zambom, Adriano Zanin, and Ronaldo Dias. 2012. "A Review of Kernel Density Estimation with Applications to Econometrics," arXiv: 1212.2812 [stat.ME].

Appendices

Appendix 1: R Code