CS215 Assignment-3

Abhi Jain - 23b0903
Anushka Singhal - 23b0928
Sabil Ahmad - 23b1057

August 2024

Contents

1 Finding optimal bandwidth

1.1 Task 1: Deriving the cross validation estimator for histogram estimator

We will prove that for histogram estimator, with m bins, the cross validation estimator can be written as:

$$\hat{J}(h) = \int \hat{f}(x)^2 dx - \frac{2}{n}\sum_{i=1}^{n} \hat{f}_{-i}(X_i) = \frac{2}{(n-1)h} - \frac{n+1}{(n-1)h}\sum_{j=1}^{m} \hat{p}_j^2$$

(a)

To Prove:

$$\int \hat{f}(x)^2 dx = \frac{1}{n^2 h}\sum_{i=1}^{m} v_i^2$$

Proof We know

$$\hat{f}(x) = \sum_{i=1}^{m} \frac{\hat{p}_i}{h} I[x \in B_i]$$

$$\implies \hat{f}(x)^2 = \sum_{i=1}^{m} \frac{\hat{p}_i^2}{h^2} I[x \in B_i] + \sum_{i=1}^{m} \sum_{j=1, j\neq i}^{m} \frac{\hat{p}_i \hat{p}_j}{h^2} I[x \in B_i] I[x \in B_j]$$

Since, we know x can occur in exactly one bin. Therefore,

$$\hat{f}(x)^2 = \sum_{i=1}^{m} \frac{\hat{p}_i^2}{h^2} I[x \in B_i]$$

$$\begin{aligned}
\implies \int \hat{f}(x)^2 dx &= \int \sum_{i=1}^{m} \frac{\hat{p}_i^2}{h^2} I[x \in B_i] dx \\
&= \sum_{i=1}^{m} \int_{x \in B_i} \frac{\hat{p}_i^2}{h^2} dx \\
&= \sum_{i=1}^{m} \frac{\hat{p}_i^2}{h^2} \cdot h = \sum_{i=1}^{m} \frac{\hat{p}_i^2}{h} \\
\implies \int \hat{f}(x)^2 dx &= \frac{1}{n^2 h}\sum_{i=1}^{m} v_i^2
\end{aligned}$$

(b)

To Prove:

$$\sum_{i=1}^{n} \hat{f}_{-i}(X_i) = \frac{1}{h(n-1)}\sum_{j=1}^{m} (v_j^2 - v_j)$$

Proof:

$$\begin{aligned}
\hat{f}_{-i}(X_i) &= \sum_{j=1}^{m} \left(\frac{\hat{p}'_j}{h} I[X_i \in B_j] \right) \\
&= \sum_{j=1}^{m} \left(\frac{v_j - 1}{(n-1)h} I[X_i \in B_j] \right) \\
\implies \sum_{i=1}^{n} \hat{f}_{-i}(X_i) &= \sum_{i=1}^{n} \sum_{j=1}^{m} \left(\frac{v_j - 1}{(n-1)h} I[X_i \in B_j] \right) \\
&= \sum_{j=1}^{m} \left(\frac{v_j - 1}{(n-1)h} \cdot \sum_{i=1}^{n} I[X_i \in B_j] \right) \\
&= \sum_{j=1}^{m} \left(\frac{v_j - 1}{(n-1)h} \cdot v_j \right) \\
\implies \sum_{i=1}^{n} \hat{f}_{-i}(X_i) &= \frac{1}{h(n-1)} \sum_{j=1}^{m} (v_j^2 - v_j)
\end{aligned}$$

1.2 Task 2: Implementation of histogram estimator and Cross-Validation estimator in Python

We filtered the dataset to consider only the first 1500 data points and restricted our analysis to objects that are less than 4 Mpc away from Earth.

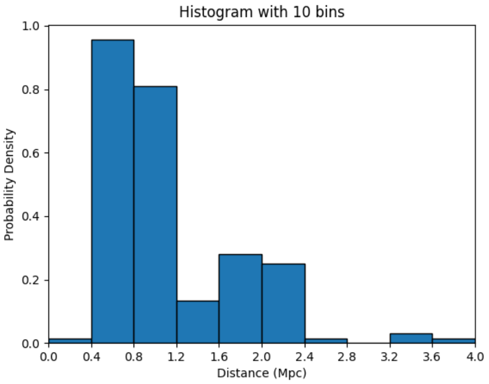


Figure 1: Histogram of Extragalactic Objects with 10 Bins

Part (a): Histogram with 10 Bins

A histogram of the filtered data was plotted using 10 bins. The estimated probabilities for each bin were calculated as follows:

```
0.006, 0.382, 0.324, 0.053, 0.112, 0.100, 0.006, 0.000, 0.012, 0.006
```

The histogram is displayed in Figure 1.

Part (b): Model Fit Analysis

The probability distribution is an undersmoothed, indicating that the histogram may not fully capture the underlying data distribution's complexity.

Part (c): Cross-Validation Score

The cross-validation score was calculated for a range of bin widths (1 to 1000). The corresponding plot is illustrated in Figure 2, saved as 'crossvalidation.png'.

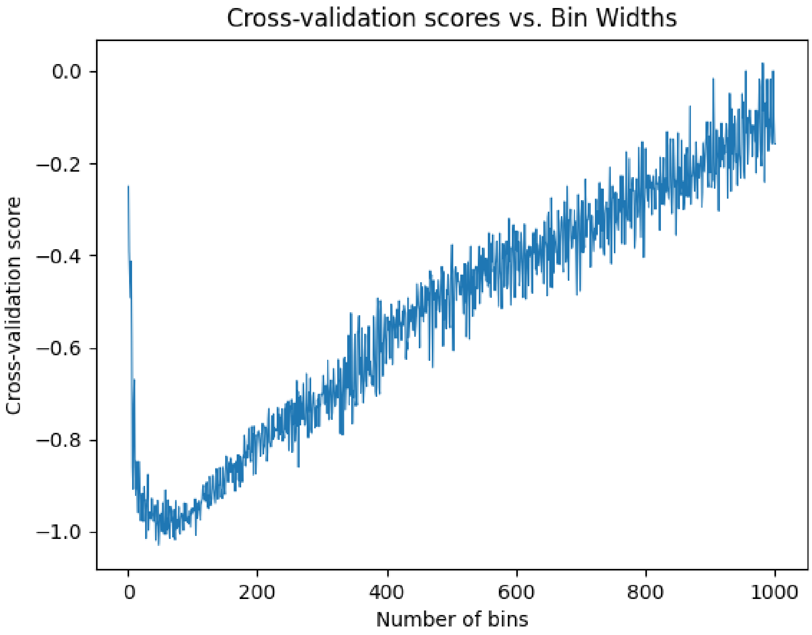


Figure 2: Cross-Validation Score vs. Number of Bins

Part (d): Optimal Bin Width

The optimal values identified were:

```
Optimal number of bins: 48
Optimal bin width: 0.083
```

Part (e): Optimal Histogram Comparison

The histogram was plotted using the optimal bin width. The resulting histogram is compared with the previous one, as shown in Figure 3, saved as ‘optimalhistogram.png‘.

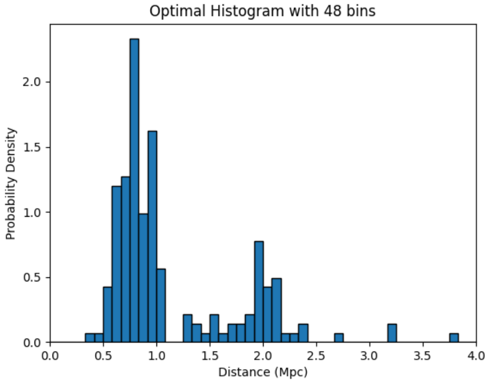


Figure 3: Histogram of Extragalactic Objects with Optimal Bin Width

2 Detecting Anomalous Transactions using KDE

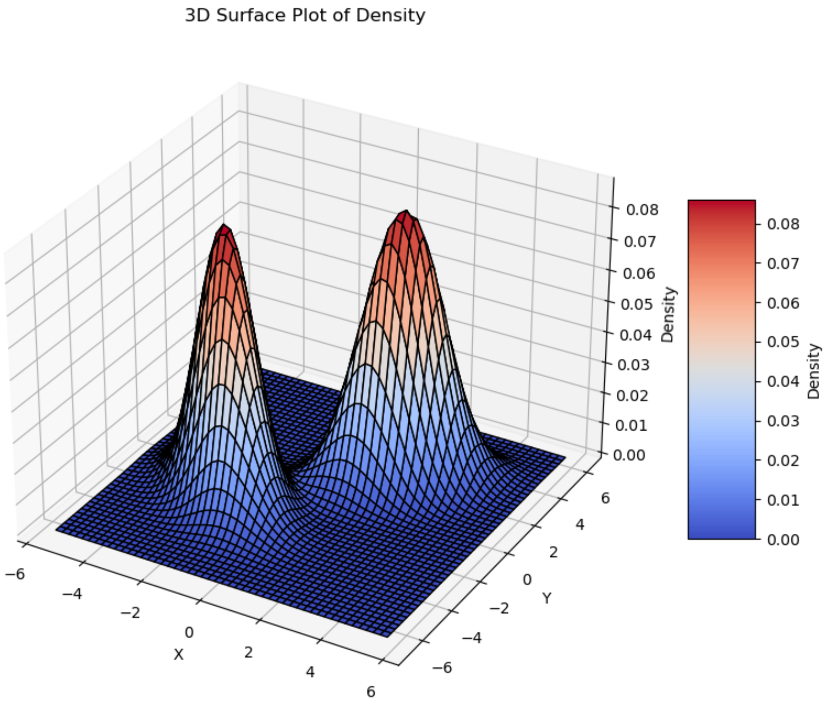


Figure 4: Transaction Data Probability Distribution

The above plot was obtained on estimating the kernel density of 2500 uniformly distributed points in the region $(x_{min}, x_{max}) * (y_{min}, y_{max})$ where x and y are the parameters of the 2D latent representation of the transaction data.

Clearly, the above data has **two modes**(assuming mode here implies peaks with vey high probability density in the whole data) in the dataset indicating the most likely of the transaction parameters.

3 Higher-Order Regression

3.1 Task 1

To prove: In a simple linear regression model the point $(\bar{x}, \bar{y})$ lies exactly on the least squares regression line. Let the dataset be $\mathcal{D} = \{(x_i, y_i), (x_2, y_2), \ldots, (x_N, y_N)\}$ and the line be $y = mx + c$.

Therefore, the square error is given by

$$E = \sum_{i=1}^{N} (y_i - mx_i - c)^2$$

Now, $\frac{\partial E}{\partial c} = 0$ and $\frac{\partial E}{\partial m} = 0$. Therefore, partial derivative w.r.t c gives:

$$(-2) \cdot \sum_{i=1}^{N} (y_i - mx_i - c) = 0$$

$$\implies \sum_{i=1}^{N} y_i = m \sum_{i=1}^{N} x_i + Nc$$

Dividing by N on both sides give

$$\implies \bar{y} = m\bar{x} + c$$

Therefore, $(\bar{x}, \bar{y})$ lies exactly on the least squares regression line

3.2 Task 2

Consider the simple linear regression model:

$$Y = \beta_0 + \beta_1 x + \epsilon,$$

We redefine the regressor variable as $z = x - \bar{x}$. The new model becomes:

$$Y = \beta_0^* + \beta_1^* z + \epsilon.$$

Aim: To find the least squares estimates for β_0^* and β_1^* and examine how they relate to the original estimates, $\hat{\beta}_0$ and $\hat{\beta}_1$.

For the original model, the least squares estimates minimize the sum of squared residuals:

$$\sum (Y_i - \hat{\beta}_0 - \hat{\beta}_1 x_i)^2.$$

The solutions for the estimates are:

$$\hat{\beta}_1 = \frac{\sum (x_i - \bar{x})(Y_i - \bar{Y})}{\sum (x_i - \bar{x})^2},$$
$$\hat{\beta}_0 = \bar{Y} - \hat{\beta}_1 \bar{x},$$

Similarly, the least squares estimates for the slope and intercept in this new model are:

$$\hat{\beta}_1^* = \frac{\sum(z_i - \bar{z})(Y_i - \bar{Y})}{\sum(z_i - \bar{z})^2} = \frac{\sum z_i(Y_i - \bar{Y})}{\sum z_i^2}$$

Since, $\bar{z} = 0$ and $z_i = x_i - \bar{x}$. Therefore, it simplifies to the same slope as the original model.

$$\hat{\beta}_1^* = \hat{\beta}_1.$$

Hence, the intercept in the new model is:

$$\hat{\beta}_0^* = \bar{Y} - \hat{\beta}_1^* \bar{z} = \bar{Y}$$

Relationship between the Two Models

From the Task 1, we know:

$$\bar{Y} = \hat{\beta}_1 \bar{x} + \hat{\beta}_0$$

Thus, the intercepts of the two models are related by:

$$\hat{\beta}_0^* = \hat{\beta}_0 + \hat{\beta}_1 \bar{x}.$$

The slopes remain identical:

$$\hat{\beta}_1^* = \hat{\beta}_1.$$

In the new model, $\hat{\beta}_0^*$ represents the expected value of Y when $z = 0$, which corresponds to $x = \bar{x}$. In the original model, $\hat{\beta}_0$ represents the expected value of Y when $x = 0$.

3.3 Task 3: OLS Implementation, SS_R and R^2 Calculation

Part (a): OLS Implementation and Predictions

Submitted Files `3_weights.pkl`,`3_predictions.csv` and `3.ipynb`.

Part (b): Plots for Underfit, Correct Fit, and Overfit

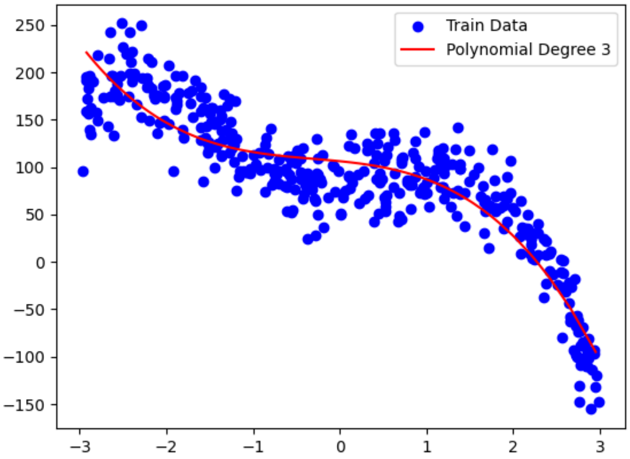


Figure 5: Underfitting (Degree = 3)

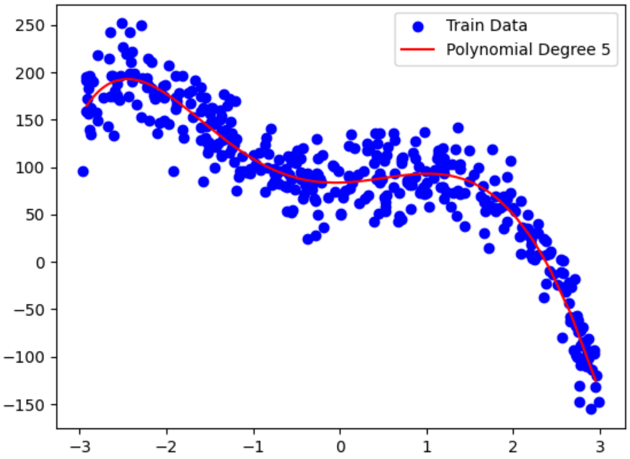


Figure 6: Correct Fit (Degree = 5)

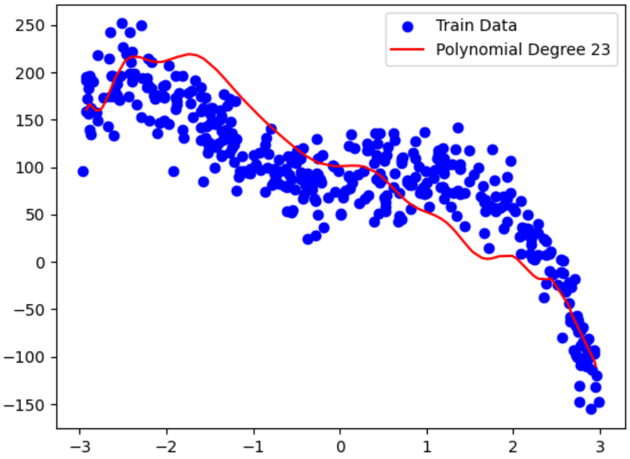


Figure 7: Overfitting (Degree = 23)

Part (c): SS_R and R^2 Metrics

Reported metrics for different models:

- **Degree 3**: $SS_R = 396409.76$, $R^2 = 0.8416$
- **Degree 5**: $SS_R = 235760.16$, $R^2 = 0.9058$
- **Degree 23**: $SS_R = 825924.75$, $R^2 = 0.6700$

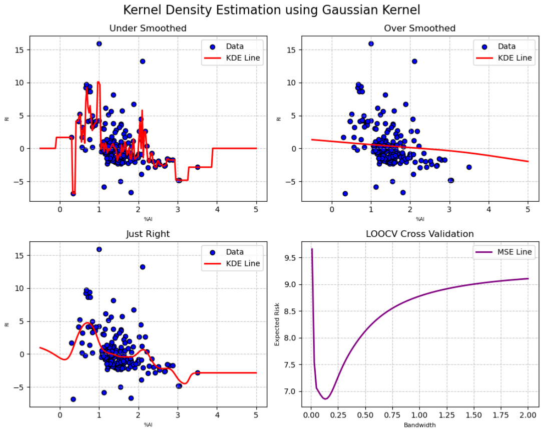
4 Non-parametric regression

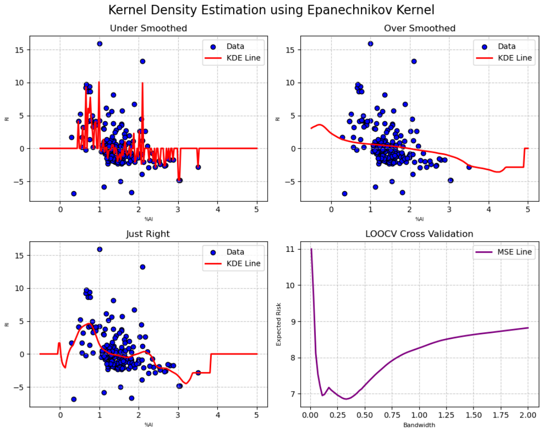
4.4 Report

2(a) Figures

Below are the KDE Estimation for the following Kernels :-

- **Gaussian Kernel**
- **Epanechnikov Kernel**





2(b) Optimal Values

The optimal bandwidth obtained for the two kernels are as follows:-
Gaussian Kernel = 0.130
Epanechnikov Kernel = 0.331

3 Differences and Similarities

The optimal bandwidth for the Gaussian kernel came out to be 0.130 and with the Epanechnikov kernel to be 0.331 using LOOCV cross-validation. The expected value of risk for the gaussian were 6.854 and with the epanechnikov kernel were 6.851. Clearly, this difference between the values is minimal this can be explained by the fact that kernels exhibit similar behaviour asymptotically, indicating that the choice of kernel here doesn't matter on the overall performance of the model however the choice of bandwidth leads to huge blowups in the risk on either side of the optimal value indicating the high dependence on the choice of bandwidht. Further it can be seen the gaussian is better at smoothening out the function as it produces a smooth curve compared to the epanechnikov where there are clearly bumps in between as seen in the bandwidth vs risk plot in the graphs above.

5 Multivariate Insights Unlocked!

The analysis and results, all are in the jupyter notebook `5.ipynb`.