

Assignment 3

CS215: Data Analysis and Interpretation

Shaik Awez Mehtab, 23B1080

Satyam Sinoliya, 23B0958

Vaibhav Singh, 23B1068

Solutions

SOLUTION 1

Finding optimal bandwidth

Part 1

(a)

We are tasked with proving the following equation for the histogram estimator:

$$\int \hat{f}(x)^2 dx = \frac{1}{n^2 h} \sum_{j=1}^m v_j^2 \quad (1.1)$$

where $\hat{f}(x)$ is the histogram estimator, v_j is the number of points in the j -th bin, n is the total number of points, h is the bin width, and m is the total number of bins.

The histogram estimator $\hat{f}(x)$ is given by:

$$\hat{f}(x) = \sum_{j=1}^m \frac{\hat{p}_j}{h} I_{[x \in B_j]} \quad (1.2)$$

where $\hat{p}_j = \frac{v_j}{n}$ is the estimated probability that a point falls in the j -th bin, and $I_{[x \in B_j]}$ is the indicator function, which is 1 if $x \in B_j$, and 0 otherwise.

Now, square $\hat{f}(x)$:

$$\hat{f}(x)^2 = \left(\sum_{j=1}^m \frac{\hat{p}_j}{h} I_{[x \in B_j]} \right)^2 \quad (1.3)$$

Since the bins B_j are non-overlapping, the cross terms vanish, and we are left with:

$$\hat{f}(x)^2 = \sum_{j=1}^m \left(\frac{\hat{p}_j}{h} \right)^2 I_{[x \in B_j]} \quad (1.4)$$

Integrating $\hat{f}(x)^2$ over the entire domain:

$$\int \hat{f}(x)^2 dx = \int \sum_{j=1}^m \left(\frac{\hat{p}_j}{h} \right)^2 I_{[x \in B_j]} dx \quad (1.5)$$

Because the bins B_j are disjoint, the integral breaks down into a sum of integrals over each bin:

$$\int \hat{f}(x)^2 dx = \sum_{j=1}^m \int_{B_j} \left(\frac{\hat{p}_j}{h} \right)^2 dx \quad (1.6)$$

The length of each bin is h and \hat{p}_j/h is independent of x , so the integral over each bin B_j is:

$$\int_{B_j} 1 dx = h \quad (1.7)$$

Thus, we get:

$$\int \hat{f}(x)^2 dx = \sum_{j=1}^m \left(\frac{\hat{p}_j}{h} \right)^2 h = \frac{1}{h} \sum_{j=1}^m \hat{p}_j^2 \quad (1.8)$$

Recall that $\hat{p}_j = \frac{v_j}{n}$, so we can substitute \hat{p}_j^2 as:

$$\hat{p}_j^2 = \left(\frac{v_j}{n}\right)^2 = \frac{v_j^2}{n^2} \quad (1.9)$$

Thus, the integral becomes:

$$\int \hat{f}(x)^2 dx = \frac{1}{h} \sum_{j=1}^m \frac{v_j^2}{n^2} \quad (1.10)$$

Factor out $\frac{1}{n^2 h}$:

$$\int \hat{f}(x)^2 dx = \frac{1}{n^2 h} \sum_{j=1}^m v_j^2 \quad (1.11)$$

Hence, proved.

(b)

Since $\hat{f}_{(-i)}$ is defined as the histogram estimator after removing i^{th} observation. We have

$$\hat{f}_{(-i)}(X_i) = \sum_{j=1}^m \frac{v_j - 1}{h(n-1)} \mathbb{1}_{[x \in B_j]} \quad (1.12)$$

Since the probability of $x \in B_j$ is $\frac{v_j}{n}$

$$\hat{f}_{(-i)}(X_i) = \sum_{j=1}^m \frac{v_j}{n} \frac{v_j - 1}{h(n-1)} \quad (1.13)$$

$$\sum_{i=1}^n \hat{f}_{(-i)}(X_i) = \sum_{i=1}^n \sum_{j=1}^m \frac{v_j}{n} \frac{v_j - 1}{h(n-1)} \quad (1.14)$$

$$= \frac{1}{(n-1)h} \sum_{j=1}^m (v_j^2 - v_j) \quad (1.15)$$

Hence Proved.

Part 2

(a) The estimated probabilities for all bins \hat{p}_j is given as:

$p_1 = 0.2059$	$p_2 = 0.4882$
$p_3 = 0.0471$	$p_4 = 0.0412$
$p_5 = 0.1353$	$p_6 = 0.0588$
$p_7 = 0.0059$	$p_8 = 0.0000$
$p_9 = 0.0118$	$p_{10} = 0.0059$

(b) The probability distribution is underfit. The binwidth is very large due to which the depiction of data is not quite smooth.

(c) The optimal binwidth h is 0.06835999999999999.

(d) The plot seems to give a better analysis of the distribution. It offers significantly more detail and resolution compared to the 10-bin histogram. Each bin covers a narrower range of data values, leading to a more refined visualization that can better capture small variations in the dataset

SOLUTION 2

Detecting Anomalous Transactions using KDE

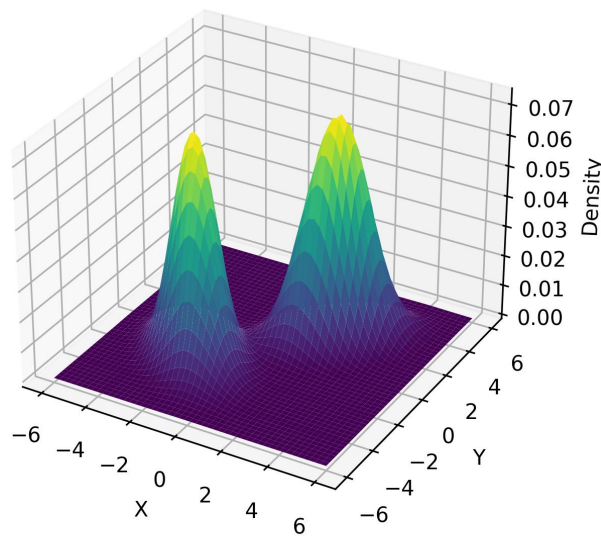


Figure 1.1: Distribution of transactions

As can be seen in the given figure, the resulting estimated distribution contains two modes.

SOLUTION 3

Higher-Order Regression

Part 1

Suppose our estimates for α and β are A and B respectively, then these values of A and B minimize

$$\sum_{i=1}^n (y_i - A - Bx_i)^2 \quad (1.1)$$

$$\Rightarrow \frac{\partial}{\partial A} \sum_{i=1}^n (y_i - A - Bx_i)^2 = 0 \quad (1.2)$$

$$\sum_{i=1}^n -2(y_i - A - Bx_i) = 0 \quad (1.3)$$

$$n\bar{y} - nA - nB\bar{x} = 0 \quad (1.4)$$

$$\bar{y} = A + B\bar{x} \quad (1.5)$$

Least square regression line is given by $y = A + Bx$. Thus by (1.5), (\bar{x}, \bar{y}) lies on the regression line.

Part 2

Suppose our estimates for β_0^* and β_1^* are A^* and B^* respectively, then A^* and B^* minimize $\sum_{i=1}^n (y_i - A^* - B^* z_i)^2$

$$\implies \frac{\partial}{\partial A^*} \sum_{i=1}^n (y_i - A^* - B^* z_i)^2 = 0 \quad \frac{\partial}{\partial B^*} \sum_{i=1}^n (y_i - A^* - B^* z_i)^2 = 0 \quad (1.6)$$

$$\sum_{i=1}^n -2(y_i - A^* - B^* z_i) = 0 \quad \sum_{i=1}^n -2z_i(y_i - A^* - B^* z_i) = 0 \quad (1.7)$$

$$n\bar{y} - nA^* - nB^*\bar{z} = 0 \quad \sum_{i=1}^n z_i y_i - A^* n\bar{z} - B^* \sum_{i=1}^n z_i^2 = 0 \quad (1.8)$$

$$\sum_{i=1}^n y_i z_i - n(\bar{y} - B^*\bar{z})\bar{z} - B^* \sum_{i=1}^n z_i^2 = 0 \quad (1.9)$$

$$B^* = \frac{\sum_{i=1}^n y_i z_i - n\bar{y}\bar{z}}{n\bar{z}^2 - \sum_{i=1}^n z_i^2} \quad A^* = \bar{y} - B^*\bar{z} \quad (1.10)$$

$$\text{Since, } z_i = x_i - \bar{x}. \bar{z} = \frac{\sum_{i=1}^n (x_i - \bar{x})}{n} = \frac{n\bar{x} - n\bar{x}}{n} = 0. \sum_{i=1}^n (x_i - \bar{x})^2 = \sum_{i=1}^n x_i^2 - n\bar{x}^2$$

$$B^* = \frac{\sum_{i=1}^n y_i (x_i - \bar{x}) - n\bar{y} \cdot 0}{n(0)^2 - (\sum_{i=1}^n x_i^2 - n\bar{x}^2)} \quad (1.11)$$

$$= \frac{\sum_{i=1}^n y_i x_i - n\bar{x}\bar{y}}{n\bar{x}^2 - \sum_{i=1}^n x_i^2} \quad (1.12)$$

This is same as B , least square estimate of β_1 i.e $B^* = B$. Also since $\bar{z} = 0$, we have $A^* = \bar{y}$ i.e $A^* = A + B\bar{x}$, where A is the least square estimate of β_0 .

Part 3

Let's restrict ourselves to single feature for simplicity. Suppose we have n data points

$$\{(x_1, y_1), \dots, (x_n, y_n)\} \quad (1.13)$$

and our OLS estimates for β_0, \dots, β_m be B_0, \dots, B_m . These must minimize

$$\sum_{i=1}^n (y_i - B_0 - B_1 x_i - \dots - B_m x_i^m)^2 \quad (1.14)$$

Partial differentiation w.r.t each B_i must be 0 which gives

$$\sum_{i=1}^n -2(y_i - B_0 - B_1 x_i - \dots - B_m x_i^m) = 0 \quad (1.15)$$

$$\sum_{i=1}^n -2x_i(y_i - B_0 - B_1 x_i - \dots - B_m x_i^m) = 0 \quad (1.16)$$

$$\vdots \quad (1.17)$$

$$\sum_{i=1}^n -2x_i^m(y_i - B_0 - B_1 x_i - \dots - B_m x_i^m) = 0 \quad (1.18)$$

Which give

$$\sum_{i=1}^n y_i = B_0 n + B_1 \sum_{i=1}^n x_i + \dots + B_m \sum_{i=1}^n x_i^m \quad (1.19)$$

$$\sum_{i=1}^n x_i y_i = B_0 \sum_{i=1}^n x_i + B_1 \sum_{i=1}^n x_i^2 + \dots + B_m \sum_{i=1}^n x_i^{m+1} \quad (1.20)$$

$$\vdots \quad (1.21)$$

$$\sum_{i=1}^n x_i^m y_i = \sum_{i=1}^n B_0 x_i^m + B_1 \sum_{i=1}^n x_i^{m+1} + \dots + B_m \sum_{i=1}^n x_i^{2m} \quad (1.22)$$

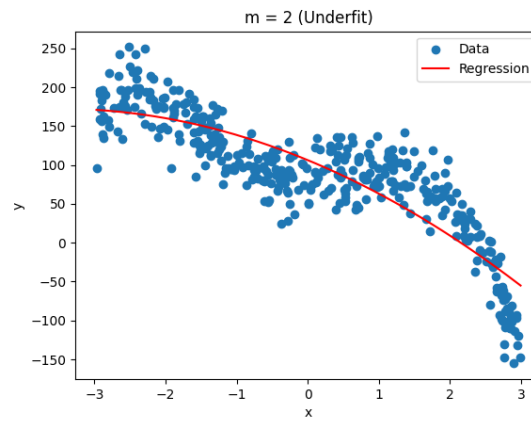


Figure 1.2: Underfit model

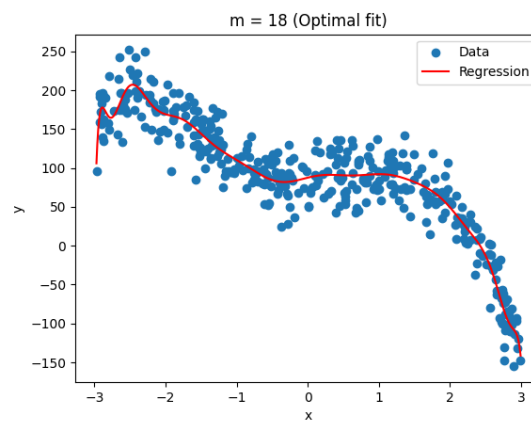


Figure 1.3: Optimal model

Taking $X = \begin{bmatrix} 1 & x_1 & \dots & x_1^m \\ 1 & x_2 & \dots & x_2^m \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & \dots & x_n^m \end{bmatrix}$, $Y = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix}$, $B = \begin{bmatrix} B_0 \\ \vdots \\ B_m \end{bmatrix}$ we have

$$X^T Y = X^T X B \quad (1.23)$$

$$B = (X^T X)^{-1} X^T Y \quad (1.24)$$

We'll use this in our code.

1. By the SSR graph above, $m = 18$ seems to be the optimal degree of the polynomial. Thus, polynomial regressor of degree 18 would be the best fit regressor for this data.
2. SS_R s values are too close so values close aren't too much of an underfit/overfit (they're almost similar) We see that SS_R at 2 and 21 have comparatively high SS_R s, so degree 2 can be considered as an underfit and degree 21 as an overfit. Here's the fit for underfit, optimal fit and overfit respectively

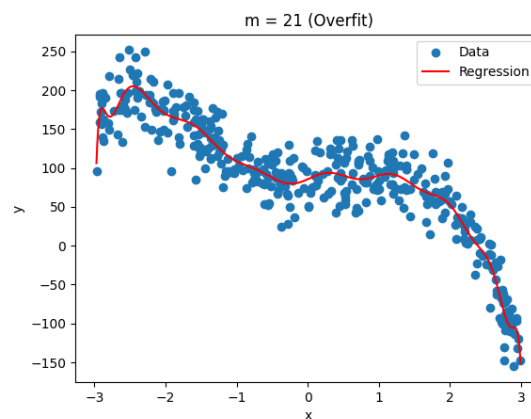


Figure 1.4: Overfit model

SOLUTION 4

Non-parametric regression

Report Bandwidth Corresponding to Minimum Estimated Risk

After running the Nadaraya-Watson kernel regression using the Epanechnikov and Gaussian kernel and performing cross-validation for bandwidth selection, the optimal bandwidth corresponding to the minimum estimated risk is:

Optimal Bandwidth of Gaussian kernel: 0.180

Optimal Bandwidth of Gaussian kernel: 0.164

Similarities and Differences Due to Choice of Different Kernel Functions

Similarities

- **General Functionality:** Both kernels assign weights to data points based on their distance from the query point, resulting in similar predictions in regions with high data density.
- **Smoothing:** As the bandwidth increases, all kernel functions produce smoother estimates. At very large bandwidths, all kernels oversmooth the data, giving too much influence to distant points.
- **Cross-validation Behavior:** Both kernels display a similar behavior during cross-validation, and the corresponding risk curves follow the same trend with bandwidth changes.

Differences

- **Shape of the Weights:**
 - **Epanechnikov Kernel:** This kernel assigns zero weight to points farther than the bandwidth due to its quadratic form, creating a more localized effect.
 - **Gaussian Kernel:** This kernel assigns non-zero weight to every point, regardless of distance, due to its exponential decay. It results in smoother estimates, but it is more sensitive to distant points.
- **Sensitivity to Outliers:**
 - **Epanechnikov Kernel:** This kernel is more resilient to outliers because they assign zero or reduced weight to distant points, decreasing the influence of outliers on the prediction.

- **Gaussian Kernel:** The Gaussian kernel is more prone to incorporating outliers, as it assigns non-zero weights even to far-away points, making it less resilient in the presence of outliers.

- **Plots**

- **Epanechnikov Kernel:** This kernel produces more precise and localized estimates, with a good balance between bias and variance when using the optimal bandwidth.
- **Gaussian Kernel:** The Gaussian kernel leads to smoother curves but gives undue influence to distant points, which can result in overfitting or oversmoothing depending on the bandwidth.

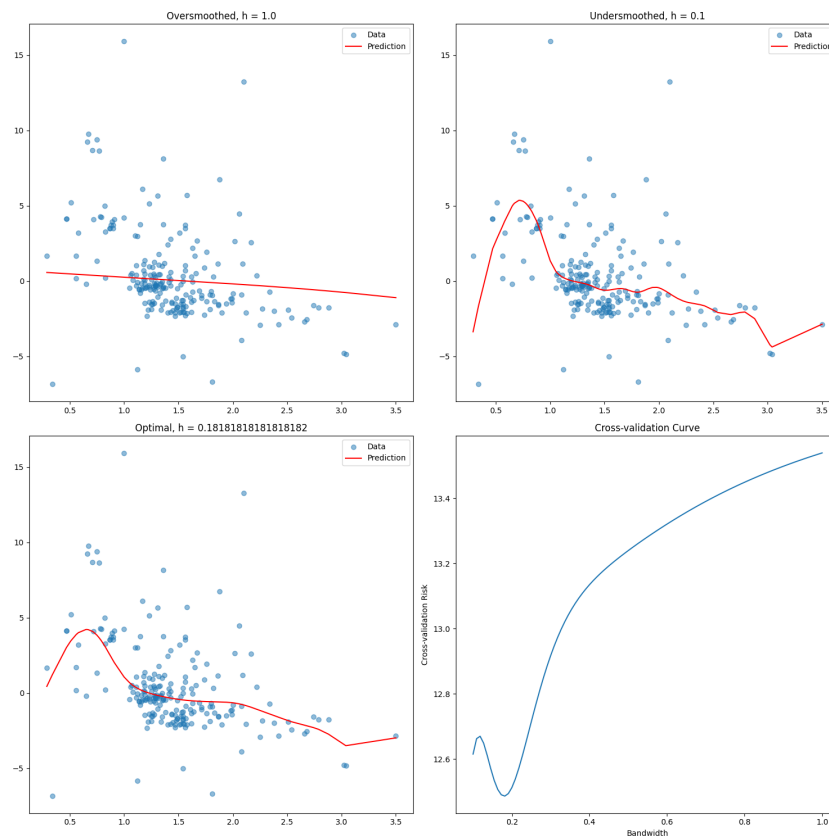


Figure 1.5: Oversmoothed, undersmoothed, optimal and cross validation curve of gaussian kernel

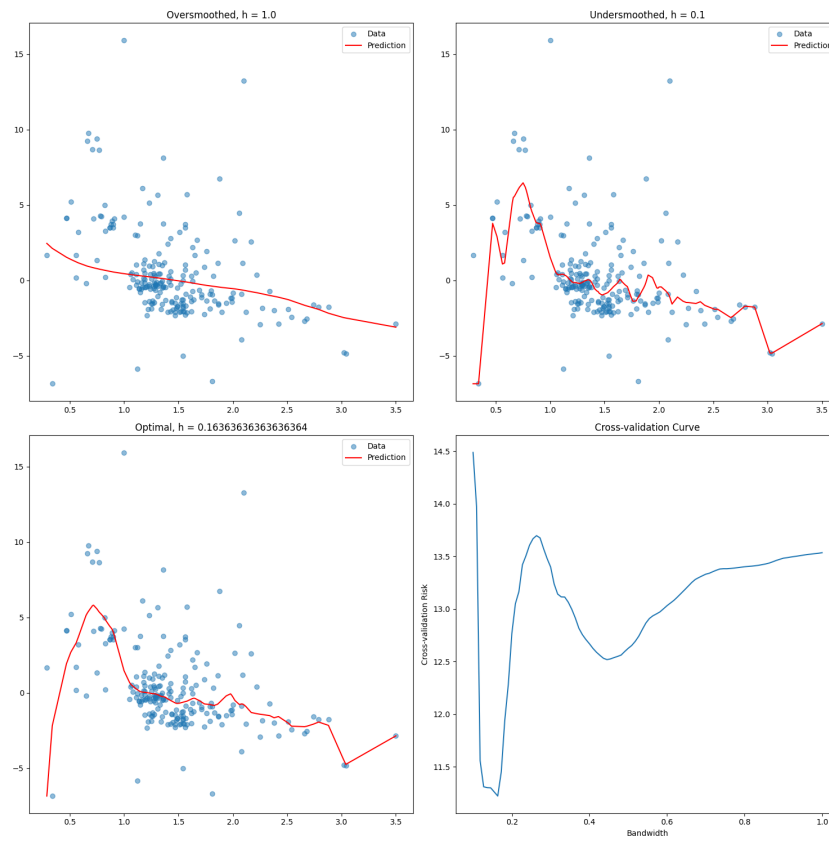


Figure 1.6: Oversmoothed, undersmoothed, optimal and cross validation curve of epanechnikov kernel