Assignment-3

Data Analysis and Interpretation

Nischal 23b1024

Pushpendra Uikey 23B1023

Nithin 23B0993

Professor: **Sunita Sarawagi**

October 16, 2024

1 Finding optimal bandwidth

Proof of $\int \hat{f}(x)^2\,dx$

1. **Define the Histogram Estimator:** The histogram estimator $\hat{f}(x)$ is given by:

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

where $\hat{p}_j = \frac{v_j}{n}$ is the proportion of data points in bin B_j, v_j is the number of points in bin B_j, h is the bin width, and $I[x \in B_j]$ is an indicator function that equals 1 if $x \in B_j$ and 0 otherwise.

2. **Calculate the Integral of $\hat{f}(x)^2$:** We are interested in calculating $\int \hat{f}(x)^2\,dx$. Expanding $\hat{f}(x)^2$, we have:

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

This can be written as:

$$\int \sum_{j=1}^{m} \sum_{k=1}^{m} \frac{\hat{p}_j}{h} \cdot \frac{\hat{p}_k}{h} I[x \in B_j] I[x \in B_k]\,dx$$

3. **Simplify the Integral:** Since a point x can belong to only one bin B_j, the product $I[x \in B_j]I[x \in B_k]$ is zero when $j \neq k$. Therefore, the integral simplifies to:

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

4. **Evaluate the Integral:** The integral $\int I[x \in B_j]\,dx$ gives the length of bin B_j, which is equal to h, the bin width. Thus, we have:

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

This simplifies to:

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

5. **Substitute for $\hat{p}_j$:** Substituting $\hat{p}_j = \frac{v_j}{n}$ into the equation, we get:

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

Final Result: Thus, the integral of the squared histogram estimator is:

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

Proof of the Second Term

To prove that

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

we'll follow these steps:

1. **Define $\hat{f}^{(-i)}(X_i)$:** Given that we remove the ith observation X_i, the histogram estimator becomes:

$$\hat{f}^{(-i)}(X_i) = \sum_{j=1}^{m} \frac{\hat{p}_j^{(-i)}}{h} I[X_i \in B_j],$$

where $\hat{p}_j^{(-i)}$ is the estimated probability of bin B_j after removing the ith observation.

2. **Change in $\hat{p}_j^{(-i)}$:** When removing the point X_i, the number of points in bin B_j becomes $v_j - I[X_i \in B_j]$. Therefore:

$$\hat{p}_j^{(-i)} = \frac{v_j - I[X_i \in B_j]}{n-1}.$$

3. **Substituting into the Histogram Estimator:** Substituting this into the estimator:

$$\hat{f}^{(-i)}(X_i) = \sum_{j=1}^{m} \frac{v_j - I[X_i \in B_j]}{(n-1)h} I[X_i \in B_j].$$

4. **Distributing the Terms:** We can separate the terms:

$$\hat{f}^{(-i)}(X_i) = \sum_{j=1}^{m} \frac{v_j}{(n-1)h} I[X_i \in B_j] - \sum_{j=1}^{m} \frac{I[X_i \in B_j]}{(n-1)h} I[X_i \in B_j].$$

5. **Evaluate Each Term:** The first term becomes:

$$\sum_{j=1}^{m} \frac{v_j}{(n-1)h} I[X_i \in B_j].$$

Since $I[X_i \in B_j] = 1$ for only the specific B_j that contains X_i, it simplifies to v_j for that j.

The second term is:

$$\sum_{j=1}^{m} \frac{I[X_i \in B_j]}{(n-1)h} I[X_i \in B_j] = \sum_{j=1}^{m} \frac{I[X_i \in B_j]}{(n-1)h},$$

which simplifies to $\frac{1}{(n-1)h}$ for that specific bin B_j since there is one indicator for the bin containing X_i.

6. **Summing Over All X_i:** Now, summing over all i from 1 to n:

$$\sum_{i=1}^{n} \hat{f}^{(-i)}(X_i) = \sum_{i=1}^{n} \left(\sum_{j=1}^{m} \frac{v_j}{(n-1)h} I[X_i \in B_j] - \sum_{j=1}^{m} \frac{1}{(n-1)h} I[X_i \in B_j] \right).$$

Since there are v_j points in each bin B_j, the sum over the indicators $I[X_i \in B_j]$ for each X_i counts the points in that bin.

7. **Final Form:** Thus, we have:

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

This leads to the result:

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

Part 2

Probability Distribution (10 Bins)

```
[0.20588235, 0.48823529, 0.04705882, 0.04117647, 0.13529412,
 0.05882353, 0.00588235, 0.00000000, 0.01176471, 0.00588235]
```

Distribution Assessment **Observation**: Distribution exhibits underfitting (ref: *10binhistogram.png*)

Reason: Large blocks oversimplify data patterns, missing crucial variations

Optimal Parameters

Parameter	**Value**
Optimal Bins	50
Bin Width	0.068

Table 1: Optimal histogram parameters from cross-validation

1.0.1 Comparative Analysis

- Highly smoothed view
- Risk of underfitting
- Oversimplified distribution

1.0.2 50-Bin Histogram (Optimal)

- Balanced detail level
- Captures true distribution
- Maintains data integrity

1.1 Key Differentiating Factors

- Granularity
- Smoothness
- Fit Quality
- Peak Representation

2 Detecting Anomalous Transactions using KDE

The resulting distribution contains **2** modes.

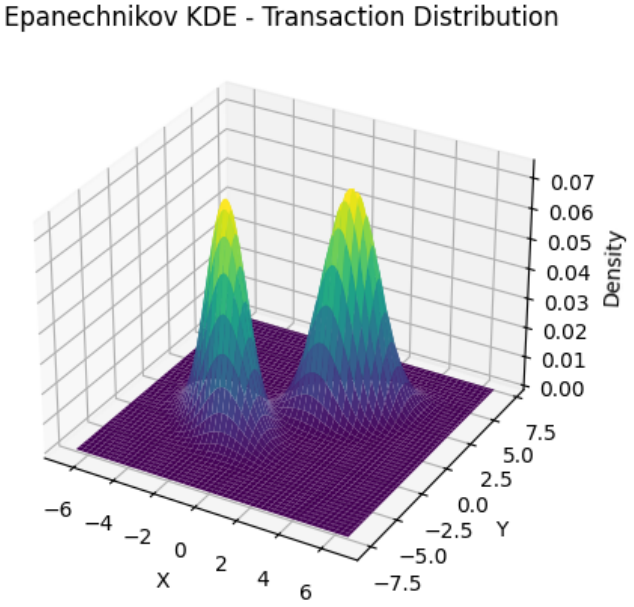

Figure 1: Distribution of Transactions

3 Higher-Order Regression

3.1 To prove that the point $(\bar{x}, \bar{y})$ lies on the least squares regression line, we start with the regression equation:

$$\hat{y} = b_0 + b_1 x.$$

Calculating $\hat{y}$ at $x = \bar{x}$:

$$\hat{y} = b_0 + b_1 \bar{x}.$$

Substituting b_0:

$$\hat{y} = \frac{\sum y - b_1(\sum x)}{n} + b_1 \bar{x}.$$

Expressing $\bar{y}$:

$$\bar{y} = \frac{\sum y}{n}.$$

Thus, we have:

$$\hat{y} = \bar{y} - b_1 \frac{\sum x}{n} + b_1 \bar{x}.$$

Since $\bar{x} = \frac{\sum x}{n}$, we substitute:

$$\hat{y} = \bar{y} - b_1 \frac{\sum x}{n} + b_1 \frac{\sum x}{n} = \bar{y}.$$

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

$$\hat{y} = \bar{y}.$$

3.2 When replacing x with $x - \bar{x}$, we define the new variable:

$$x' = x - \bar{x}.$$

The regression model becomes:

$$\hat{y} = b_0' + b_1'(x - \bar{x}).$$

Calculating b_1'

The slope b_1' remains unchanged:

$$b_1' = \frac{n\sum(x - \bar{x})y - \sum(x - \bar{x})\sum y}{n\sum(x - \bar{x})^2} = b_1.$$

Calculating b_0'

To find the new intercept b_0':

$$b_0' = \bar{y} - b_1'\bar{x}.$$

Since $b_1' = b_1$, we have:

$$b_0' = \bar{y} - b_1\bar{x}.$$

Comparison with Old Estimates

- Old estimates:

$$b_1 = \frac{n(\sum xy) - (\sum x)(\sum y)}{n(\sum x^2) - (\sum x)^2}$$

$$b_0 = \frac{\sum y - b_1(\sum x)}{n}$$

- New estimates:

$$b_1' = b_1$$

$$b_0' = \bar{y} - b_1\bar{x}.$$

Conclusion

- The slope b_1 remains unchanged: $b_1' = b_1$. - The intercept changes to $b_0' = \bar{y} - b_1\bar{x}$.

The new intercept reflects that the regression line is centered around the means of x and y.

4 Non-Parametric Regression

Similarities and differences due to choice of different Kernels

1. Epanechnikov Kernel (epanechnikov_kernel_regression.png):

Features: The Epanechnikov kernel is efficient, since it gives weights that decrease quadratically with distance from the target point. It would thus tend to smooth more the estimation: instead of considering some others, it focuses a neighborhood around the test point.

Plot Analysis:

- The regression line is quite smooth in those regions where the data is dense. Hence, the noise is minimized, and the trend in general appears to be followed fairly well.
- It depicts sharp cuts at the edges like the case with any bounded kernel such as Epanechnikov.

The smoothness is balanced with just the right amount, avoiding oversmoothing while still preserving good locality.

2. Gaussian Kernel (gaussian_kernel_regression.png)

Properties: The Gaussian kernel weights the points furthest from the target exponentially decreasingly. It applies positive weight to all the points but the influence of lower-weight points is much lesser. This makes it sensitive to over-smoothing unless engineered correctly.

Plot Interpretation:

- The Gaussian kernel produces a much smoother curve than the Epanechnikov kernel, perhaps even over-smoothing local variations.
- It does not respond sharply to noise, perhaps beneficial when dealing with noisy datasets but missing finer details in the data.

It's a trade-off between smoothness and local flexibility. The Gaussian kernel is suitable for applications where a smooth global trend is expected to be.

3. Triangular Kernel (triangular_kernel_regression.png)

It actually has all the characteristics: A triangular kernel has a linear decay in weights. Hence, it assigns relatively high weights to the nearby points but still includes some influence of points that are farther away. This is similar to Epanechnikov but with a linear drop instead of quadratic.

Plot Analysis:

- Like this kernel, the local flexibility is balanced with smoothness as well. The graph shows that it is more smooth than the Epanechnikov curve but captures the local fluctuations a bit better than the Gaussian kernel.

- It acts almost like the Epanechnikov in the sense that it responds to local fluctuations and gives the localized estimate.

Key Observations:

Smoothing Effect: Gaussian will probably smooth the data far more significantly than both the Epanechnikov and Triangular kernels would. This may be desirable on noisy datasets but could over-smooth fine features.

Local Sensitivity: The Epanechnikov and Triangular kernels are more locally sensitive to distributional changes than the Gaussian kernel.

Boundary Behavior: For the Epanechnikov and Triangular kernels, the sharp cutoff at the boundaries occurs due to compact support. For the Gaussian kernel, the derivative does extend out to infinity, which can cause the decision boundary to be worse closer to the edges of the dataset.

Kernel Bandwidths Corresponding to Minimum Estimated Risk

Gaussian Kernel:

Bandwidth corresponding to minimum estimated risk is

$$0.13584593503338627$$

Triangular Kernel:

Bandwidth corresponding to minimum estimated risk is

$$0.3461555677887769$$

Epanechnikov Kernel:

Bandwidth corresponding to minimum estimated risk is

$$0.3270365102655595$$