NON-PARAMETRIC

NAME – ABHIRUP BAG
ROLL – 13000122082
DEPARTMENT – CSE(B)
SEMESTER – 6
PAPER – PATTERN RECOGNITION(PEC-IT602D)

CONTENT

INTRODUCTION

- **What is Non-Parametric Estimation?**
  - ✓ *A statistical method that makes minimal assumptions about data distribution.*
  - ✓ *Unlike parametric methods, it does not assume a fixed functional form.*
- **Why is it Important?**
  - ✓ *Useful for analyzing real-world data with unknown or complex distributions.*
  - ✓ *Provides more flexibility in modeling.*

KEY CONCEPTS

- **No Fixed Parameters :** Unlike parametric methods (e.g., normal distribution with mean and variance), non-parametric methods do not assume a predefined shape for the data distribution.
- **Data-Driven Approach :** These methods rely on the structure of the observed data to make inferences.
- **More Flexible but More Data Needed :** Non-parametric methods adapt to data patterns but often require larger sample sizes to achieve similar accuracy as parametric methods.

COMMON NON-PARAMETRIC METHODS

- Decision Trees
- Support Vector Machines (SVM)
- K-Nearest Neighbour Estimation(KNN)
- Neural networks

K-NEAREST NEIGHBOR ESTIMATION (K-NN DENSITY ESTIMATION)

What is Nearest Neighbor Estimation?

- Determines the density of a point based on the distance to its k-nearest neighbor.
- Provides a local density estimate based on proximity to other data points.

Strengths & Weaknesses:

✓ Works well for high-dimensional data.

✗ Computationally expensive, especially for large datasets.

Formula :

$$\hat{f}(x) = \frac{k}{nV}$$

- V is the **volume** around xxx containing k nearest neighbors.

📌 **Example:** Used in **anomaly detection** (e.g., fraud detection in banking transactions).

▪ ALGORITHM OF K-NN DENSITY ESTIMATOR

1. Select a value for k.
2. Find the k-nearest neighbors of the query point x.
3. Compute the volume V of the region containing these neighbors.
4. Estimate the density using the formula above.

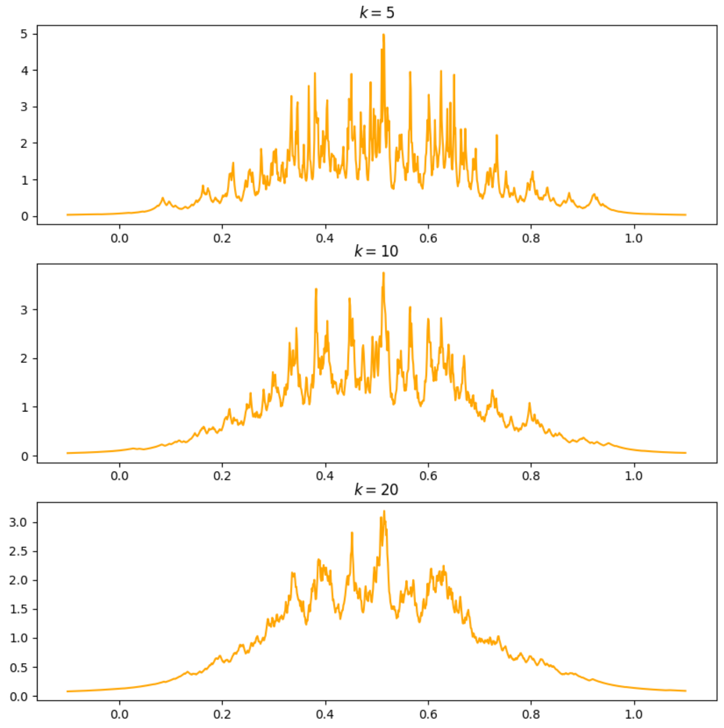
APPLICATIONS OF KNN

1. Anomaly detection (low-density regions indicate anomalies).
2. Probability estimation in machine learning.
3. Image processing (density-based segmentation).

Python Code to plot a K-NN DENSITY ESTIMATOR

```
# KNN Density esitmator
gaussian = norm(loc=0.5, scale=0.2)
X = gaussian.rvs(500)
grid = np.linspace(-0.1, 1.1, 1000)
k_set = [5, 10, 20]
fig, axes = plt.subplots(3, 1, figsize=(10, 10))
for i, ax in enumerate(axes.flat):
    K = k_set[i]
    p = np.zeros_like(grid)
    n = X.shape[0]
    for i, x in enumerate(grid):
        dists = np.abs(X-x)
        neighbours = dists.argsort()
        neighbour_K = neighbours[K]
        p[i] = (K/n) * 1/(2 * dists[neighbour_K])
    ax.plot(grid, p, color='orange')
    ax.set_title(f'$k={K}$')
plt.show()
```

Output



$k = 5$
5
4
3
2
1
0
0.0
0.2
0.4
0.6
0.8
1.0
$k = 10$
3
2
1
0
0.0
0.2
0.4
0.6
0.8
1.0
$k = 20$
3.0
2.5
2.0
1.5
1.0
0.5
0.0
0.0
0.2
0.4
0.6
0.8
1.0

SUPPORT VECTOR MACHINES (SVM)

What is Nearest Neighbor Estimation?

- SVM is a non-parametric supervised learning algorithm used for classification and regression.
- It works by finding the optimal separating hyperplane that maximizes the margin between different classes.

Mathematical Formulation

For a binary classification problem, given training data (x_i, y_i) where $y_i \in \{-1, 1\}$, the optimization problem is:

$$\min_{w,b} (1/2)\|\omega\|^2$$

subject to:

$$y_i\left(w^T + b\right) \geq 1$$

where: w is the weight vector defining the hyperplane. b is the bias term. The kernel trick allows transforming data into higher dimensions without explicitly computing the transformation.

TYPES OF SVM

- SVM
  - Linear SVM: Uses a linear hyperplane.
  - Non-linear SVM: Uses kernel functions like:
    - Gaussian (RBF) Kernel
    - Polynomial Kernel

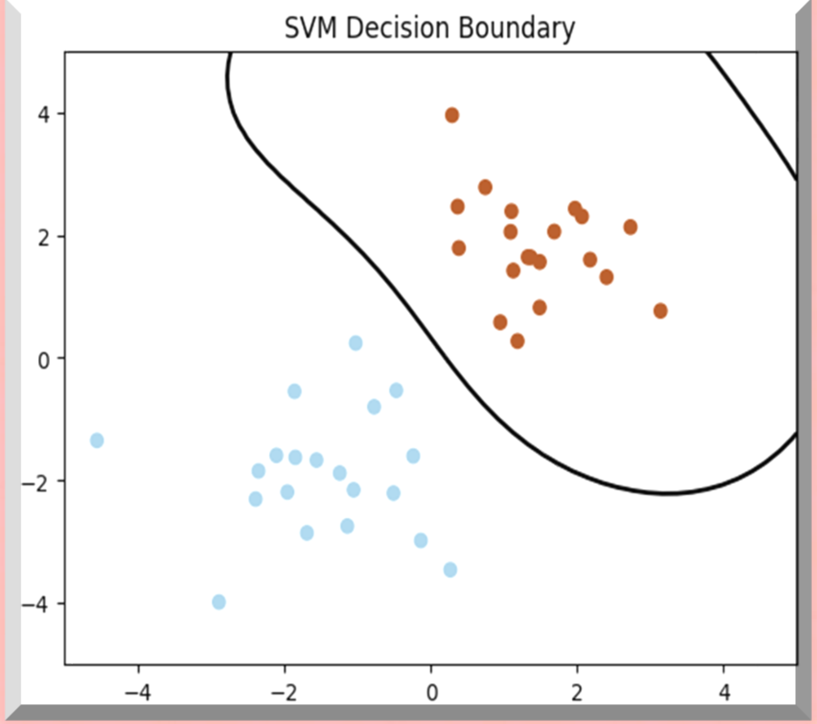
APPLICATIONS OF SVM

1. Text classification (e.g., spam detection).
2. Image recognition.
3. Medical diagnosis (e.g., cancer classification).

Example Code: SVM in Python

```python
import numpy as np
import matplotlib.pyplot as plt
from sklearn import svm

# Generate sample data
np.random.seed(0)
X = np.r_[np.random.randn(20, 2) - [2, 2], np.random.randn(20, 2) + [2, 2]]
Y = [0] * 20 + [1] * 20

# Train SVM model with RBF kernel
clf = svm.SVC(kernel='rbf', C=1.0, gamma='scale')
clf.fit(X, Y)

# Plot decision boundary
xx, yy = np.meshgrid(np.linspace(-5, 5, 50), np.linspace(-5, 5, 50))
Z = clf.decision_function(np.c_[xx.ravel(), yy.ravel()])
Z = Z.reshape(xx.shape)

plt.scatter(X[:, 0], X[:, 1], c=Y, cmap=plt.cm.Paired)
plt.contour(xx, yy, Z, levels=[0], linewidths=2, colors='k')
plt.title("SVM Decision Boundary")
plt.show()
```

Output



COMPARISON: KNN VS. SVM

Feature	KNN Density Estimation	SVM
Flexibility	Very flexible but sensitive to k	Controlled flexibility via kernel
Computation	Expensive for large datasets	Efficient after training
Type	Unsupervised	Supervised
Purpose	Density Estimation	Classification/Regression
Parameters	Number of neighbors (k), distance metric	Kernel, C (regularization), Gamma

Advantages & Disadvantages of Non-Parametric Estimation Methods

✅ **Advantages:**

✓ No assumption about underlying distribution.
✓ More **adaptive and flexible** to real-world data.
✓ Works well for **high-dimensional and irregular data.**

❌ **Disadvantages:**

✗ **More data needed** for accurate estimation.
✗ **Computationally expensive**, especially for large datasets.
✗ Choice of **bandwidth/smoothing parameters** affects results.

📌 **Key Takeaway:** While flexible, non-parametric methods require careful tuning for accurate results.

APPLICATIONS

- 📌 **Density Estimation:** Income distribution, species population density.
- 📌 **Trend Analysis:** Stock prices, climate data.
- 📌 **Classification/Regression:** Medical diagnosis (k-NN), real estate pricing.
- 📌 **Hypothesis Testing:** Non-parametric tests (Mann-Whitney U test).
- 📌 **Visual:** Real-world examples (e.g., KDE plot of income data).

CONCLUSION

- ✓ KNN Density Estimation is useful for probability estimation and anomaly detection.
- ✓ SVM is a powerful classification tool that works well with high-dimensional data using the kernel trick.
- ✓ Both methods are widely used in practical machine learning applications.

REFERENCES

- 📘 **"The Elements of Statistical Learning"** – Trevor Hastie, Robert Tibshirani, Jerome Friedman
- 📗 **"Pattern Recognition and Machine Learning"** – Christopher Bishop
- 📙 **"Nonparametric Statistical Methods"** – Myunghee Cho Paik
- 📕 **"All of Statistics: A Concise Course in Statistical Inference"** – Larry Wasserman
- 📓 **"Kernel Smoothing"** – M. P. Wand, M. C. Jones
- 📔 **"Learning with Kernels"** – Bernhard Schölkopf, Alexander J. Smola
- 📖 **"Support Vector Machines and Other Kernel-Based Learning Methods"** – Nello Cristianini, John Shawe-Taylor
- 📚 **"Understanding Machine Learning: From Theory to Algorithms"** – Shai Shalev-Shwartz, Shai Ben-David

Thank you

ABHIRUP BAG

abhirup7477@gmail.com

ROLL NO. : 13000122082