

Pattern Recognition

Instructor:

Dr. Sadaf Yasmin

Associate professor

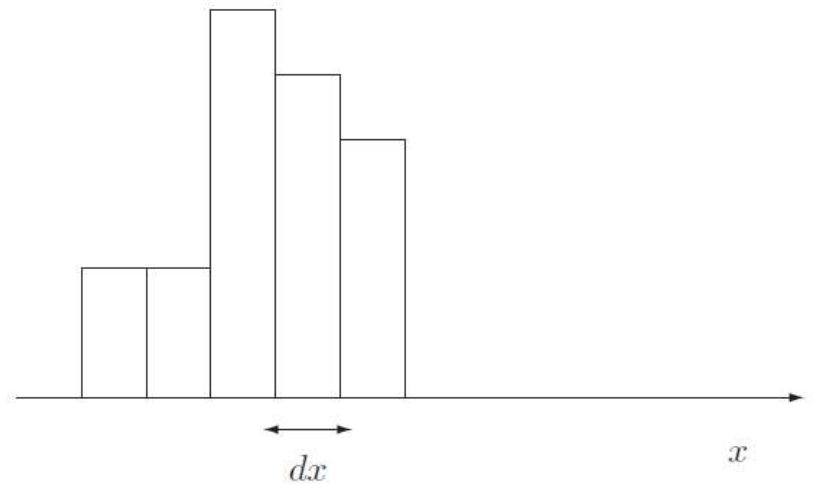
COMSATS University Islamabad Attock Campus

Histogram

Histogram Density Estimation

- The histogram method is perhaps the oldest method of density estimation.
- It is the classical method by which a probability density is constructed from a set of samples
- In one dimension, the real line is partitioned into a number of equal-sized cells and the estimate of the density at a point x is taken to be

$$\hat{p}(x) = \frac{n_j}{\sum_j^N n_j dx}$$



Histogram Density Estimation

Here's the idea step by step:

- **Divide the data range into bins**
 - Suppose your data lies between a minimum and maximum value. You split this range into equal-width intervals (bins).
- **Count the data points in each bin**
 - For each bin, count how many sample points fall into it.
- **Convert counts into probabilities**
 - The frequency in each bin is divided by the total number of samples and by the bin width, so that the estimated histogram integrates to 1 (making it a proper probability density estimate).

Histogram Density Estimation

$$\hat{f}(x) = \frac{\text{Number of samples in bin}}{n \cdot h}$$

where

- n = total number of samples
- h = bin width

Histogram Density Estimation

- For a multidimensional observation space

$$\hat{p}(\mathbf{x}) = \frac{n_j}{\sum_j n_j dV}$$

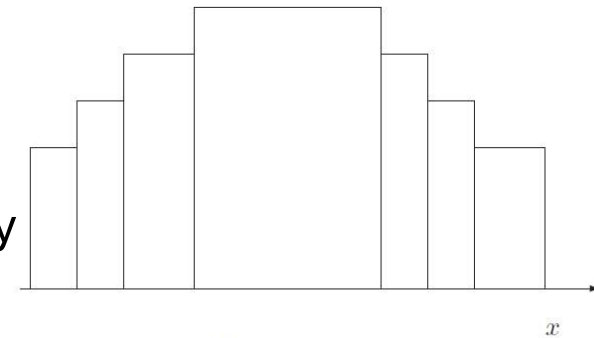
where dV is the volume of bin j

Problems with the basic Histogram Approach

- Curse of Dimensionality
- Density estimate is discontinuous and falls abruptly to zero at the boundaries of the region

Data-adaptive Histograms

- The bin widths are not fixed in advance, but instead adapt to the data distribution
- Better captures local structure of the distribution
- **Dense regions** → bins are narrower (to capture detail)
- **Sparse regions** → bins are wider (to avoid noisy estimates)
- **Equal-Frequency Binning**
 - Instead of equal widths, each bin contains approximately the same number of data points.
 - Example: In a dataset of 1000 values, if we choose 10 bins, each bin will contain about 100 samples (bin widths will vary).



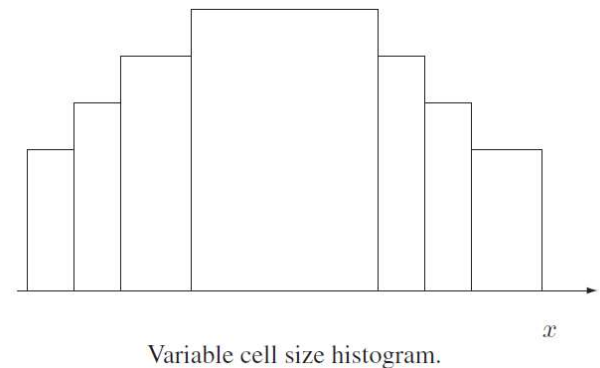
Variable cell size histogram.

Data-adaptive Histograms

Example:

Suppose we have 1,000 exam scores, but most students scored between 50–70.

- A fixed-width histogram might waste bins between 0–30 and 90–100 (few data).
- A data-adaptive histogram would assign **narrow bins around 50–70** (to capture details) and **wider bins elsewhere**.

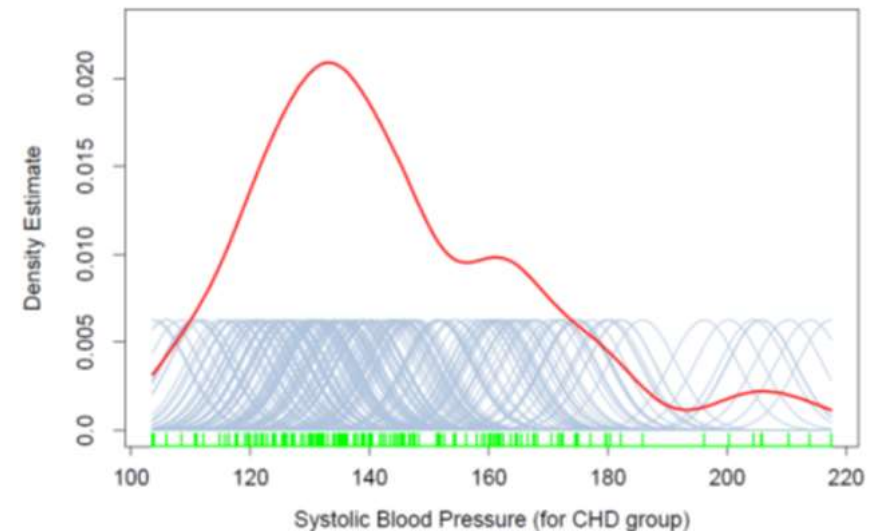


Kernal Method

Kernal Density Estimation

- Use regions centered on the datapoints
 - Allow the regions to overlap.
 - Let each individual region contribute a total density of $1/N$
 - Use regions with soft edges to avoid discontinuities

Density Estimation



Kernel Density Estimation

Kernel (Parzen) Method

- Instead of **fixed bins**, we place a **smooth kernel (like a bell curve)** on each data point.
- Then we **sum all those small curves** to get a smooth overall shape.

Formula:

$$\hat{f}(x) = \frac{1}{nh} \sum_{i=1}^n K\left(\frac{x - x_i}{h}\right)$$

Here:

- h = bandwidth (controls smoothness)
- K = kernel function (e.g., Gaussian)

Result: Smooth, continuous curve.

Kernel Functions

1. Gaussian Kernel (Normal Kernel)

$$K(u) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}u^2}$$

Shape: Smooth bell curve

Range: $-\infty < u < \infty$ (infinite support)

Properties:

- Most widely used in practice
- Always smooth and differentiable
- Good for continuous data

Kernel Functions

2. Epanechnikov Kernel

$$K(u) = \begin{cases} \frac{3}{4}(1 - u^2), & |u| \leq 1 \\ 0, & |u| > 1 \end{cases}$$

Shape: Parabolic (inverted "U")

Range: $-1 \leq u \leq 1$ (finite support)

Properties:

- Theoretically *most efficient* (minimizes mean square error)

Kernel Functions

3. Uniform (Rectangular) Kernel

$$K(u) = \begin{cases} \frac{1}{2}, & |u| \leq 1 \\ 0, & |u| > 1 \end{cases}$$

Shape: Flat box

Range: $-1 \leq u \leq 1$

Properties:

- Simplest kernel (counts all points equally within window)

Kernel Functions

Meaning of u

In the kernel density formula:

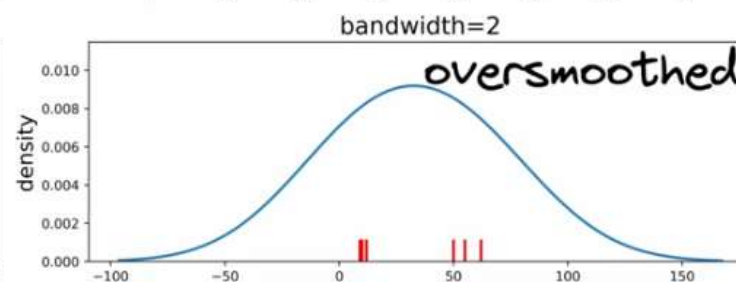
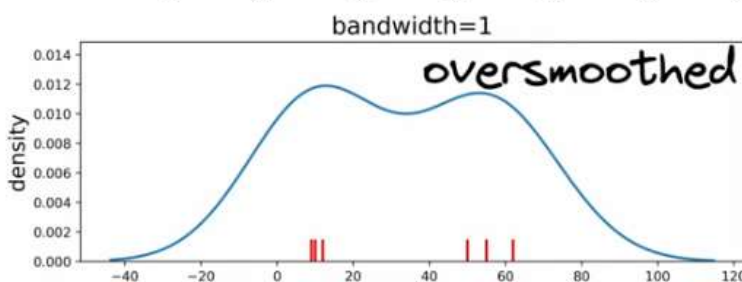
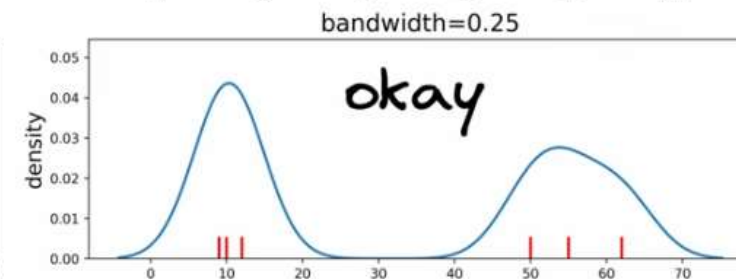
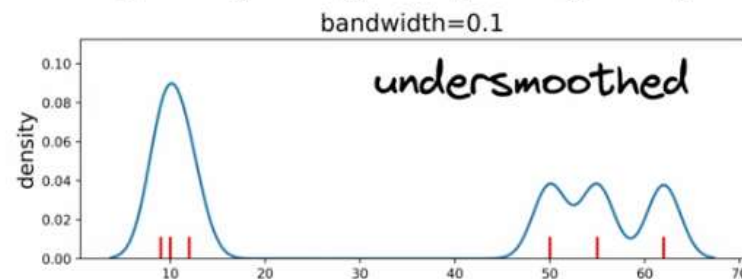
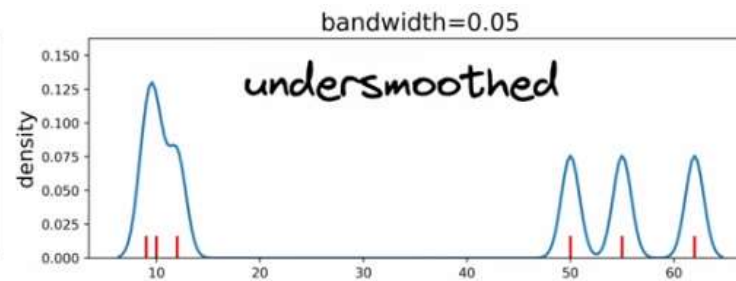
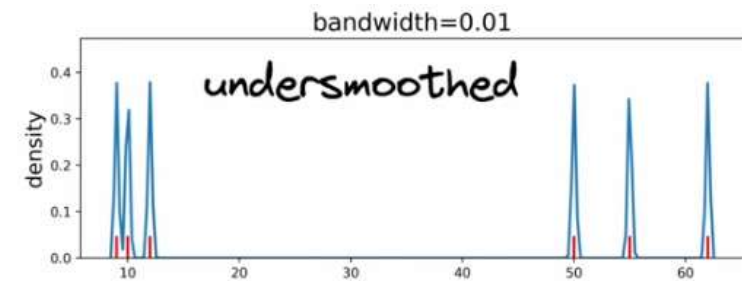
$$\hat{f}(x) = \frac{1}{nh} \sum_{i=1}^n K\left(\frac{x - x_i}{h}\right)$$

the u is defined as:

$$u = \frac{x - x_i}{h}$$

- the **Gaussian kernel** is almost always the default because it performs well in most cases.
- The **bandwidth (h)** usually matters *more* than the kernel choice itself.

Choice of bandwidth



Curse of Dimensionality

In kernel density estimation (KDE) or kernel regression, we compute

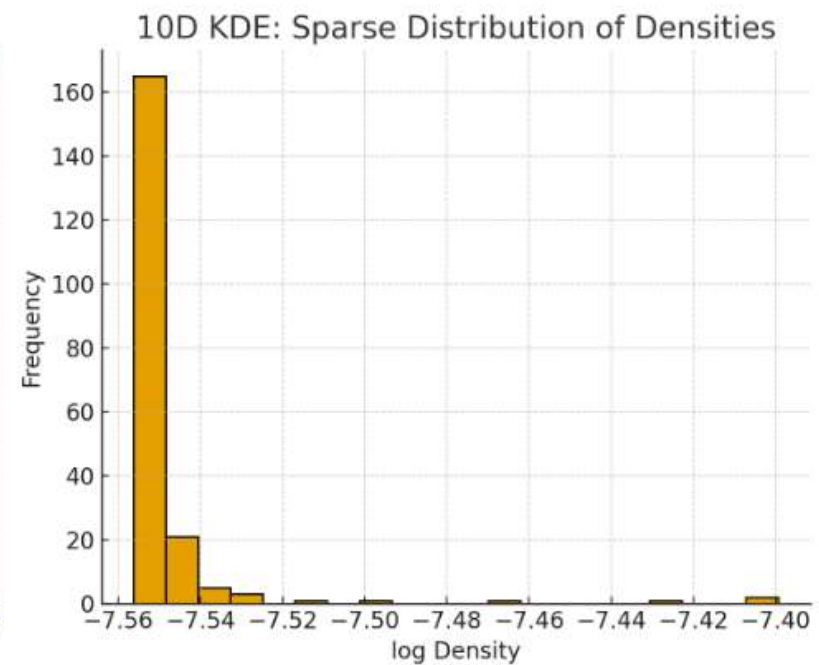
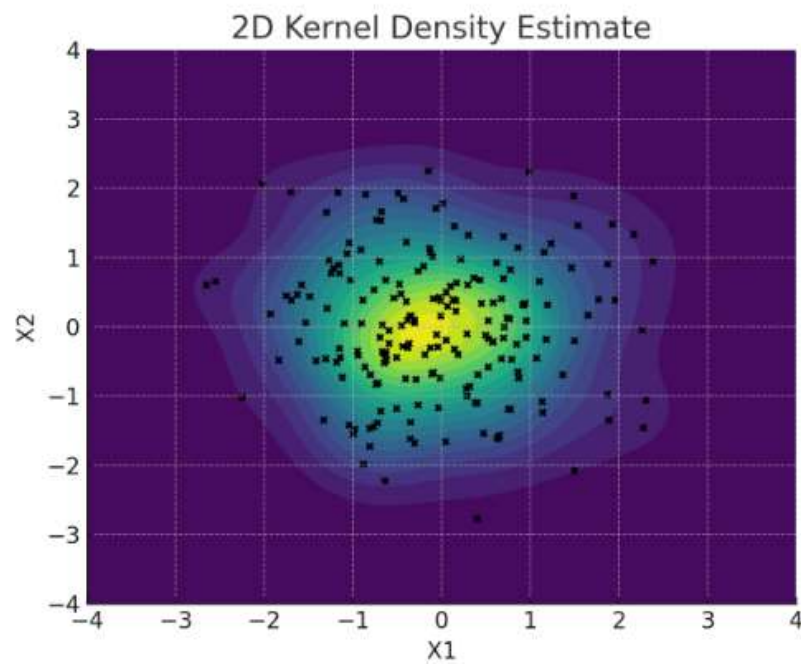
$$\hat{f}(x) = \frac{1}{nh^d} \sum_{i=1}^n K\left(\frac{x - x_i}{h}\right)$$

Here h is the **bandwidth** and d is the dimension.

As d increases:

- The factor h^d in the denominator explodes.
- You must use a larger h to get enough data in the neighborhood → this smooths too much → **loss of detail**.
- Or you keep h small → very few points contribute → **high variance**.

Curse of Dimensionality



Instead of a full d -D kernel, use a product of 1D kernels:

$$K(x) = \prod_{j=1}^d K_j(x_j)$$

This assumes independence between dimensions \rightarrow reduces computational burden,

Instead of a full d -D kernel, use a product of 1D kernels:

$$K(x) = \prod_{j=1}^d K_j(x_j)$$

This assumes independence between dimensions \rightarrow reduces computational burden,

OR

Dimensionality reduction first

Use methods like:

- **PCA (Principal Component Analysis)**
- **t-SNE, UMAP**
- **Autoencoders**

to project data into a lower-dimensional manifold where kernel estimation is more reliable.