Machine Learning

Non-parametric Algorithms: k-NN Classifier and Parzen Window

Indian Institute of Information Technology
Sri City, Chittoor

Agenda

- Parzen Window
- Defining R_n (region in the feature-space)
- Two different approaches - fixed volume vs. fixed number of samples in a variable volume
- Example 3D hypercube
- The window function and estimation
- Critical parameters of the Parzen-window technique: window width and kernel
- Selecting Window
- Selecting Kernel

Introduction

Probability density estimation is one of the oldest problems in statistics and machine learning.

There are two approaches, viz.,

1. Parametric, and
2. Non-parametric.

Parametric Density Estimation

- We assume a parametric class (the form) from which the data is drawn. For eg., Gaussian distribution.
- Then we try to estimate the parameters of the Gaussian distribution ie., mean and covariance matrix.

- For doing the parameter estimation we use the training examples.

We study about this Later.

Non-parametric Methods

No assumption is made about the form of the distribution.

Depends totally on the data set.

Non-parametric Methods

1. Parzen Window based
2. Nearest Neighbors based

Parzen window

- The Parzen-window method (also known as Parzen-Rosenblatt window method) is a widely used non-parametric approach to estimate probability density p(x), for a specific point x.
- Notation: The estimate of p(x) when we use dataset of size n is denoted by $p_n(x)$.
- It doesn't require any knowledge or assumption about the underlying distribution.
- A popular application of the Parzen-window technique is to estimate the class-conditional densities (or also often called 'likelihoods').
- Likelihoods, $p(x \mid \omega_i)$ in a supervised pattern classification problem from the training dataset (where p(x) refers to the probability density that the sample x belongs to the particular class ω_i).

Credits: "Kernel density estimation via the Parzen-Rosenblatt window method"
By Sebastian Raschka

Where would this method be useful?

- Imagine that we are about to design a Bayes classifier for solving a statistical pattern classification task using Bayes' rule:

$$P(\omega_i \mid \boldsymbol{x}) = \frac{p(\boldsymbol{x} \mid w_i) \cdot P(\omega_i)}{p(\boldsymbol{x})}$$

$$\Rightarrow posterior\ probability = \frac{likelihood \cdot prior\ probability}{evidence}$$

- If the parameters of the the class-conditional densities (also called likelihoods) are known, it is pretty easy to design the classifier.
- Imagine we are about to design a classifier for a pattern classification task where the parameters of the underlying sample distribution are not known.
- Therefore, we wouldn't need the knowledge about the whole range of the distribution; it would be sufficient to know the probability of the particular point, which we want to classify, in order to make the decision.

Credits: "Kernel density estimation via the Parzen-Rosenblatt window method"
By Sebastian Raschka

Parzen Window

- In parzen window we are going to see how we can estimate this probability from the training sample.

- However, the only problem of this approach would be that we would seldom have exact values - if we consider the histogram of the frequencies for a arbitrary training dataset.

- Therefore, we define a certain region (i.e., the Parzen-window) around the particular value to make the estimate.

[1] *Parzen, Emanuel*. On Estimation of a Probability Density Function and Mode. The Annals of Mathematical Statistics 33 (1962), no. 3, 1065 1076.
[2] *Rosenblatt, Murray*. Remarks on Some Nonparametric Estimates of a Density Function. The Annals of Mathematical Statistics 27 (1956), no. 3, 832–837.

The most fundamental technique

- The probability P that a vector x will fall in a region R is given by

$$P = \int_{\mathcal{R}} p(\mathbf{x}')\, d\mathbf{x}'.$$

- $p(x) \approx \frac{\left(\int_R\ p(x')dx'\right)}{V}$ where V is the volume of the region R

Defining the Region R_n

- The basis of this approach is to count how many data-points fall within a specified region R_n (or "window"). Our intuition tells us, that (based on the observation), the probability that one sample falls into this region is:

$$P = \frac{\#\ of\ samples\ in\ R}{Total\ samples}$$

- The probability of observing k points out of n in a Region R_n : we consider **a binomial distribution:**

$$P_k = \binom{n}{k} \cdot P^k \cdot (1-P)^{n-k}$$

- In a binomial distribution, the probability peaks sharply at the mean

Credits: "Kernel density estimation via the Parzen-Rosenblatt window method"
By Sebastian Raschka

Defining the Region Rn

- Let p(x) be the probability density at x. Let over the small region R it is uniformly distributed.

$$P = \int_R p(x')\,dx' = p(x) \cdot V$$

where V is the volume of the region R, and if we rearrange those terms, so that we arrive at the following equation:

$$\frac{k}{n} = p(x) \cdot V$$

$$\Rightarrow p(x) = \frac{k/n}{V}$$

- This simple equation above (i.e, the "probability estimate") lets us calculate the probability density of a point x by counting how many points k fall in a defined region (or volume).

Credits: Sebastian Raschka

To estimate the density at $\mathbf{x}$, we form a sequence of regions $\mathcal{R}_1, \mathcal{R}_2, ...$, containing $\mathbf{x}$ — the first region to be used with one sample, the second with two, and so on. Let V_n be the volume of $\mathcal{R}_n$, k_n be the number of samples falling in $\mathcal{R}_n$, and $p_n(\mathbf{x})$ be the nth estimate for $p(\mathbf{x})$:

$$p_n(\mathbf{x}) = \frac{k_n/n}{V_n}.$$

If $p_n(\mathbf{x})$ is to converge to $p(\mathbf{x})$, three conditions appear to be required:

- $\lim_{n\to\infty} V_n = 0$
- $\lim_{n\to\infty} k_n = \infty$
- $\lim_{n\to\infty} k_n/n = 0.$

Theoretically it can be shown that

- V_n can be reduced as n increases: Like $V_n = 1/\sqrt{n}$. Starting with $V_1 = 1$.
- Or, k_n can be increased as n increases: Like $k_n = \sqrt{n}$. Here the volume of the region is increased to fit the k_n points exactly.

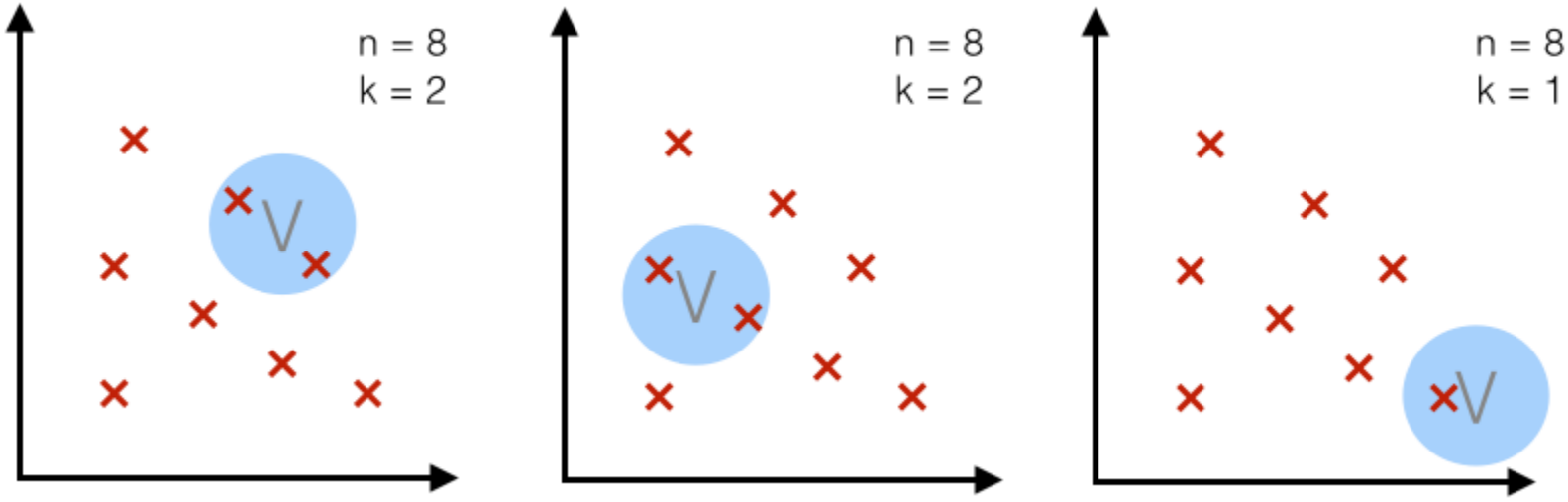
- Both these approaches are shown to converge to the true density *asymptotically*.

In Practice: Two approaches followed are-

Two different approaches - fixed volume vs. fixed number of samples in a variable volume

Case 1 - fixed volume:

- For a particular number n (= number of total points), we use volume V of a fixed size and observe how many points k fall into the region.

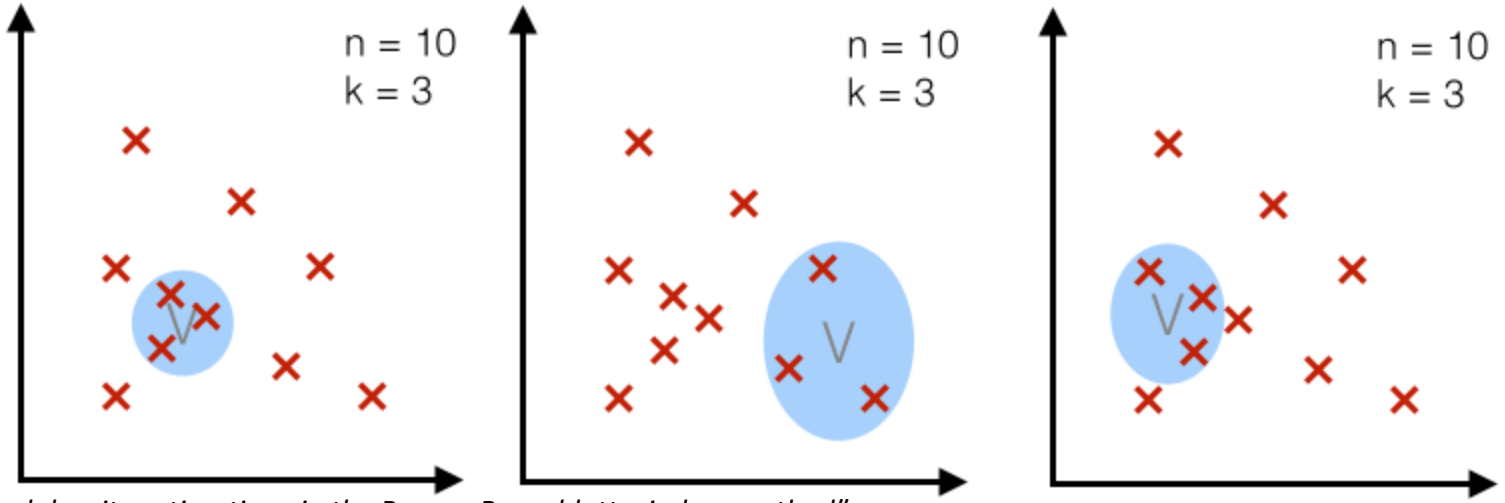


Credits: "Kernel density estimation via the Parzen-Rosenblatt window method"
By Sebastian Raschka

Two different approaches - fixed volume vs. fixed number of samples in a variable volume

Case 2 - fixed *k*:

- For a particular number n (= number of total points), we use a fixed number k (number of points that fall inside the region or volume) and adjust the volume accordingly..



Credits: "Kernel density estimation via the Parzen-Rosenblatt window method"
By Sebastian Raschka

- The second approach, fixed k, is nothing but the k-Nearest Neighbor Classifier.
- We will see about this, in detail later.

Example 3D-hypercubes

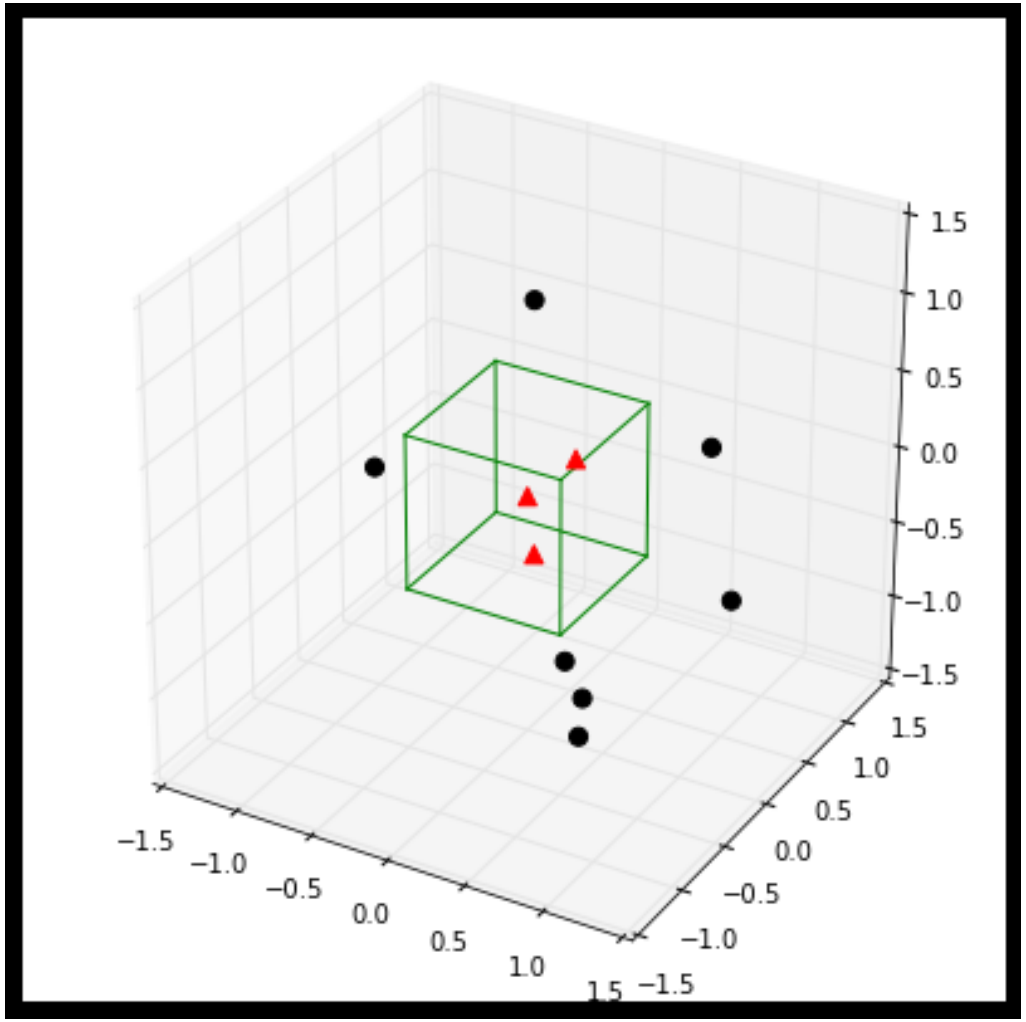
- To illustrate this with an example and a set of equations, let us assume this region R_n is a hypercube.

- The volume of this hypercube is defined by $V_n = h_n^d$, where h_n is the length of the hypercube, and d is the number of dimensions.

- For an 2D-hypercube with length 1, for example, this would be $V_1 = 1^2$ and for a 3D hypercube $V_1 = 1^3$, respectively.

Credits: "Kernel density estimation via the Parzen-Rosenblatt window method"
By Sebastian Raschka

Example: A typical 3-dimensional unit hypercube ($h_1 = 1$) representing the region R_1, and 10 sample points, where 3 of them lie within the hypercube (red triangles), and the other 7 outside (black dots).



Credits: "Kernel density estimation via the Parzen-Rosenblatt window method"
By Sebastian Raschka

- Each point falling within the window (hyper-cube) contributes to the density.
- Points falling outside will not.
- We can formalize this in to a window function.

The window function

- Once we visualized the region R_1 like above, it is easy and intuitive to count how many samples fall within this region, and how many lie outside.
- To approach this problem more mathematically, we would use the following equation to count the samples k_n within this hypercube, where φ is our so-called window function.

$$\phi(\boldsymbol{u}) = \begin{cases} 1 & |u_j| \leq 1/2\,; \qquad j = 1, \ldots, d \\ 0 & otherwise \end{cases}$$

for a hypercube of unit length 1 centered at the coordinate system's origin.

- If we extend on this concept, we can define a more general equation that applies to hypercubes of any length h_n that are centered at *x*:

$$k_n = \sum_{i=1}^{n} \phi\left(\frac{\boldsymbol{x} - \boldsymbol{x}_i}{h_n} \right)$$

$$\text{where } \boldsymbol{u} = \left(\frac{\boldsymbol{x} - \boldsymbol{x}_i}{h_n} \right)$$

Credits: "Kernel density estimation via the Parzen-Rosenblatt window method" By Sebastian Raschka

Parzen-window estimation

- we can now formulate the Parzen-window estimation with a hypercube kernel as follows:

$$p_n(\boldsymbol{x}) = \frac{1}{n} \sum_{i=1}^{n} \frac{1}{h^d} \phi \left[\frac{\boldsymbol{x} - \boldsymbol{x}_i}{h_n} \right]$$

where

$$h^d = V_n \quad \text{and} \quad \phi \left[\frac{\boldsymbol{x} - \boldsymbol{x}_i}{h_n} \right] = k$$

- And applying this to our unit-hypercube example above, for which 3 out of 10 samples fall inside the hypercube (into region *R*), we can calculate the probability *p(**x**)* that ***x*** samples fall within region *R* as follows:

$$\boldsymbol{x} = \frac{k/n}{h^d} = \frac{3/10}{1^3} = \frac{3}{10} = 0.3$$

Credits: "Kernel density estimation via the Parzen-Rosenblatt window method" By Sebastian Raschka

An important observation

- A point falling slightly outside the window will not contribute to the density.
- This is incorrect.
- Also, intuitively, very near point to x should contribute more to the density than far point (even though both are within the window).

An important observation

- Each point in the training set should contribute to density.
  - Near contributes more than far.
- This gives smooth estimates
- This avoids selecting the window width problem.
- Empty window problem can be avoided.

An important Observation

- For p(x), let near-by points contribute more, and far-away points less.
- For example one can use a Gaussian function to do this.

An important Observation

- Assume that a Gaussian N(0,1) is kept at each of the data points. For example, let x_i be the data point.
- Let the dataset size is n.
- Then contribution of x_i to $p(x)$ will be
-

$$\varphi(x - x_i) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}(x-x_i)^2}$$

- If x is a multivariate data point, say with d dimensions, then

$$\phi(x - x_i) = \frac{1}{(2\pi)^{d/2}} \exp\left[-\frac{1}{2} \|x - x_i\|^2 \right]$$

The Gaussian Kernel

- The estimation of $p(x)$ when we have n data points is:

$$p_n(x) = \frac{1}{n}\sum_{i=1}^{n}\phi(x - x_i) = \frac{1}{n \cdot (2\pi)^{d/2}}\sum_{i=1}^{n}\exp\left[-\frac{1}{2}\|x - x_i\|^2\right]$$

- This approach is called "Parzen-window density estimation using the Gaussian Kernel"

Note, division by the volume (of the hypercube !) V is not appearing here. Since the contribution of each data point to density is itself density.

- Classification example

In classifiers based on Parzen-window estimation:

- We estimate the densities for each category and classify a test point by the label corresponding to the maximum posterior

- The decision region for a Parzen-window classifier depends upon the choice of window function as illustrated in the following figure.

In the next Lecture we will see…

In Practice, K-NNC is the most used classifer (which is nothing but a non-parametric density estimation based method only!!)

Thank You:
Question?