Aprenentatge i Reconeixement de Formes
Pattern Recognition and Machine Learning:

2. Distance-based Classification

Francesc J. Ferri

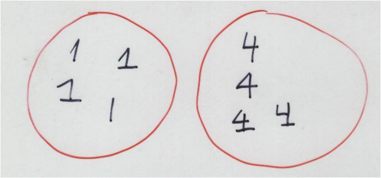
Dept. d'Informàtica. Universitat de València

December 2009

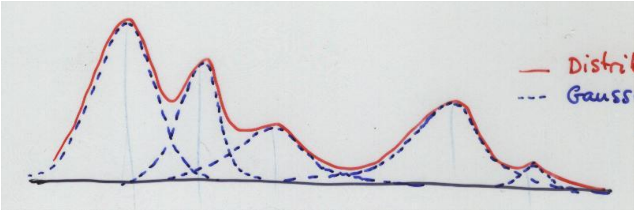
Summary of the talk

Nonparametric PR methods

- Nonparametric means no assumption about the underlying statistical structure of the problem is made.
- Either this is unknown or there is no practical/convenient way of using it.
- Nonparametric methods are closely related to multimodal problems.



- Multimodality can always be tackled by breaking the problem into smaller unimodal problems. (Semiparametric approaches: if these subproblems are solved by parametric methods).



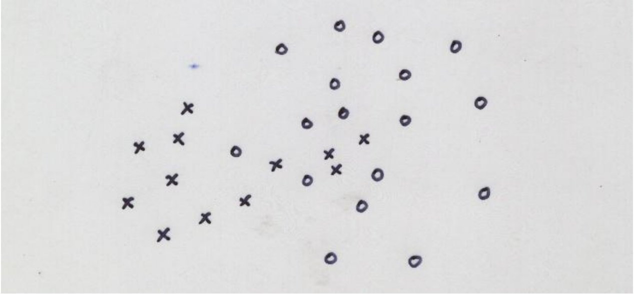
Distance-based methods

Inductive learning

If no a priori knowledge is assumed about the problem and its statistical structure, all information is only in the examples

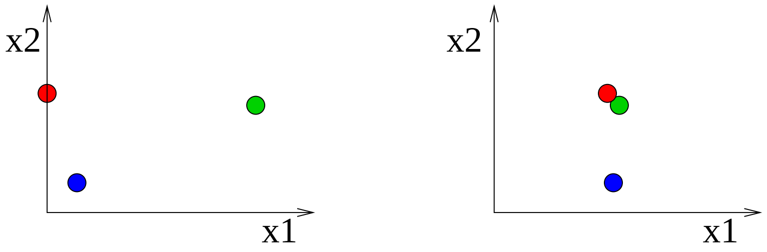
Two basic approaches

1. try to fit a particular learning model to data (e.g. GLD functions)
2. look for a measure of similarity or dissimilarity and corresponding classification criteria



Distances in representation spaces

What is imporant now is not the values of attributes/features but the relative position (distances) among different objects in the space.



Pros

Affinity, class membership, likeliness are easily interpretable and computable. Possibility of using more abstract spaces (e.g. the space of graphs/strings, etc.)

Cons

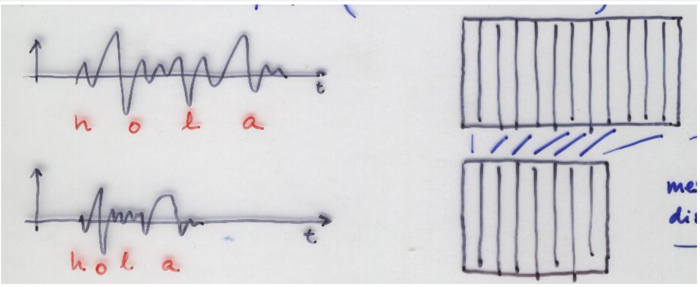
Dependence on changes of space (e.g. scale). Importance of the particular metric

Similarity/Dissimilarity

In order to apply distance based methods the only ingredient is a convenient measure of similarity/dissimilarity.

Families of distance measures

- euclidean (Minkowski in general)
- mahalanobis (data dependent)
- (feature) weighted distances
- (prototype) weighted distances
- local distances
- pseudo-distances
- abstract (dis)similarities



Minimum Distance Classifier

Let E a given representation space.

Let $P = \{(\mathbf{p_i}, w_i)\}_{i=1}^{c}$ a c-class training set (one prototype per class)

$\mathbf{p_i} \in E$, $\Omega = \{w_1, \ldots, w_c\}$

Let $d : E \times E \longrightarrow \mathbb{R}^{\geq 0}$ a distance measure.

Minimum Distance Classification Rule

$x \in w_i \iff d(\mathbf{x}, \mathbf{p_i}) < d(\mathbf{x}, \mathbf{p_j}) \; \forall j \neq i$

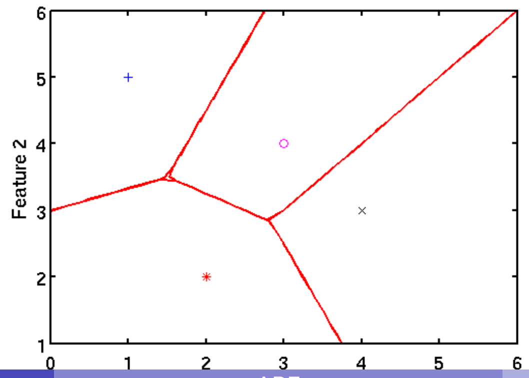
Minimum Distance Classifier

In the particular case of $\mathbb{R}^d$ and euclidean distance

$$||\mathbf{x} - \mathbf{p_i}|| = (\mathbf{x}^2 - 2\mathbf{p_i} \cdot \mathbf{x} + \mathbf{p_i}^2)^{\frac{1}{2}}$$

As squaring is monotonously increasing and eliminating the 2nd order term which does not depend on i, this is equivalent to considering **linear** discriminant functions:

$$d_i(\mathbf{x}) = (2\mathbf{p_i}) \cdot \mathbf{x} - \mathbf{p_i}^2$$



Minimum distance – Nearest mean

The minimum distance classifier is equivalent to the Nearest mean classifier when the c prototypes in the training set are taken as the corresponding means.

Also, the Bayes classifier for the normal case can be seen as the Minimum distance classifier by taking the (class-conditional) Mahalanobis distance to each prototype (mean).

Nearest Neighbor Classifier

Let E a given representation space.

Let $P = \bigcup_{i=1}^{c} P_i$ be the training set where

$P_i = \{(\mathbf{p_j^i}, w_i)\}_{j=1}^{N_i}$ (several prototypes per class)

Let $d : E \times E \longrightarrow \mathbb{R}^{\geq 0}$ a distance measure.

Nearest Neighbor Classification Rule

$x \in w_i \iff d(\mathbf{x}, P_i) < d(\mathbf{x}, P_j) \; \forall j \neq i$

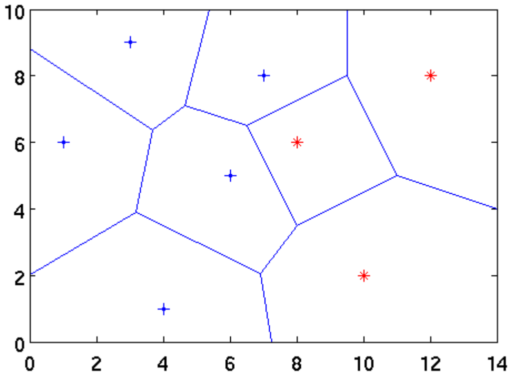
where a distance from a point to a set is defined as

$$d(x, P) = \min_{y \in P} d(x, y)$$

Nearest Neighbor Classifier

In the particular case of $\mathbb{R}^d$ and euclidean distance, with slightly more work it can be shown that this is equivalent to considering the following **piecewise linear** discriminant functions:

$$d_i(\mathbf{x}) = \max_{1 \leq k \leq N_i} [(2\mathbf{p_k^i})^T \cdot \mathbf{x} - \mathbf{p_k^i}^2]$$



Proximity structures. Basics

Given a set of prototypes (points) $P = \{p_i\}_{i=1}^N$ in a particular **metric** space, E.

Voronoi Polytope at p_i

$VP(p_i) = \{x \in E | d(x, p_i) \leq d(x, p_j) \; \forall j \neq i\}$

Voronoi Diagram of P

$$VD(P) = \bigcap_{p \in P} VP(p_i)$$

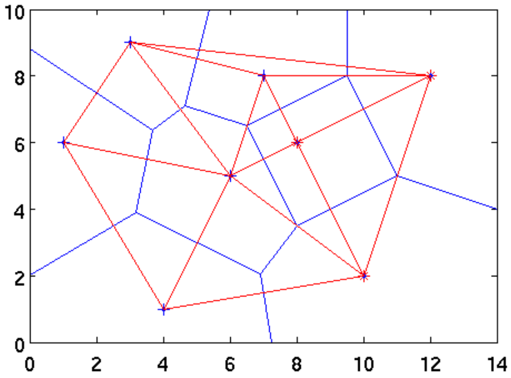
Proximity graphs. Basics

Delaunay Triangulation of P

Is an undirected graph, $DT = (P, A)$, where P are nodes and $(p_i, p_j) \in A \iff VP(p_i) \cap VP(p_j) \neq \emptyset$

Voronoi diagram (as a graph)

Is the dual graph of the Delaunay triangulation. Nodes are equidistant points to three (or more) prototypes (which need be the closest ones) and arcs connect those points that share some of the equidistant prototypes.



Other proximity graphs

Proximity graph

Nodes are prototypes (points). There is an arc between two prototypes if a certain geometric constraint related to distances is fulfilled.

Minimal Spanning Tree (MST)

Is a connected graph whose **total** distance is minimum.

Relative Neighborhood Graph (RNG) and Gabriel Graph (GG)

there is an arc between two points if no other point lies in a certain area depending on these points.

The MST, RNG, and GG are nested subgraphs of the Delaunay Triangulation (DT). That is

$$MST \subseteq RNG \subseteq GG \subseteq DT$$

k-Nearest Neighbors Classifier

Let E a given representation space.

Let $P = \bigcup_{i=1}^{c} P_i$ be the training set where

$P_i = \{(\mathbf{p_j^i}, w_i)\}_{j=1}^{N_i}$ (several prototypes per class)

Let $d : E \times E \longrightarrow \mathbb{R}^{\geq 0}$ a distance measure.

Let $V_k^P(\mathbf{x})$ a set containing the k closest prototypes to $\mathbf{x}$ in the set P.

k-Nearest Neighbors Classification Rule

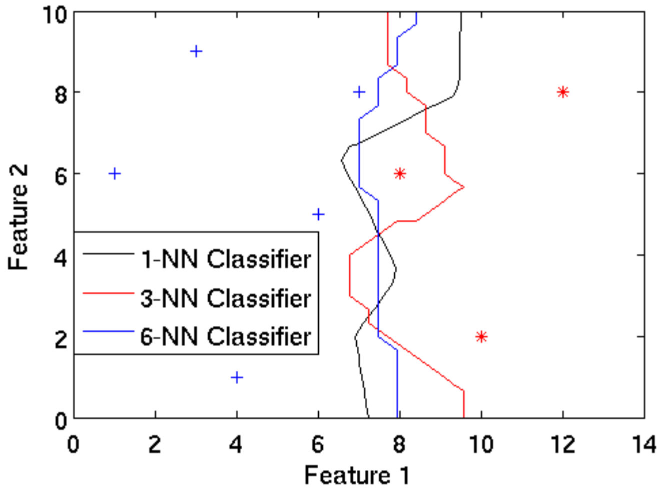
$x \in w_i \iff |V_k(\mathbf{x}) \cap P_i| > |V_k(\mathbf{x}) \cap P_i| \; \forall j \neq i$

Now the classification does not depend on comparing distance values but on a **voting**.

Nevertheless, the NN classifier corresponds to the 1-NN classifier.

k-Nearest Neighbors Classifier

It is considerably more difficult to see it but in the euclidean case the decision boundaries are also piecewise linear.



Classification power of nearest neighbors

Both NN and kNN are able to discriminate between any two classes using a piecewise linear discrimination boundary.

Question

Can any piecewise linear partition of the euclidean space be generated by a convenient 2-class prototype set and either the NN or the kNN classification rule?

Answer, yes. But not for any k.

Nearest Neighbors. Motivation

- Intuitively, using nearest neighbors will improve classification results for difficult (multimodal) problems.
- Using high k values diminishes the influence of non representative (or noisy or erroneous) prototypes.

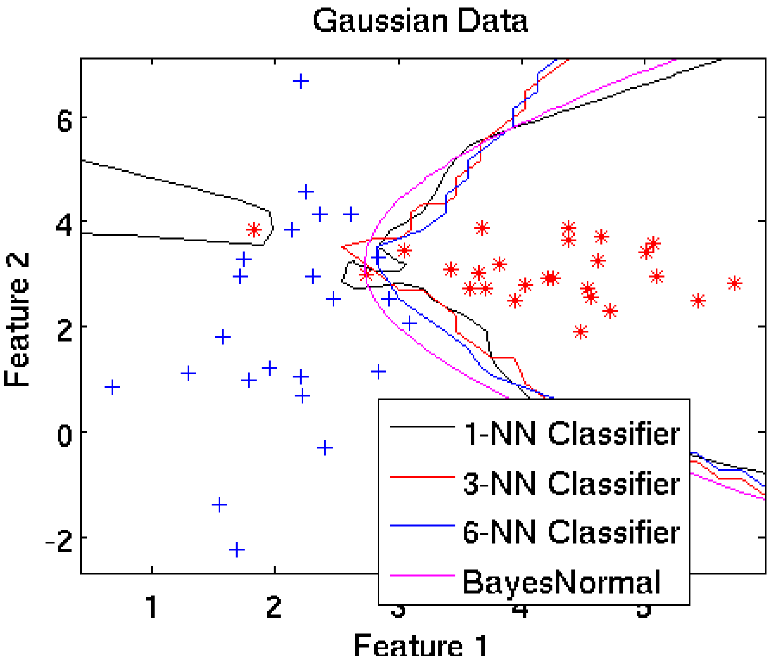
Question

Will higher values of k improve always the results? Even with only correct prototypes?

Question

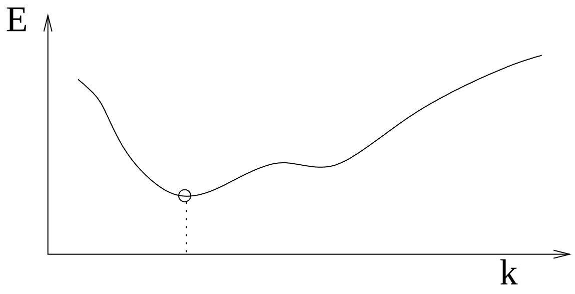
The more prototypes the better?

Performance of nearest neighbors



Nearest neighbors classification error

A typical error curve as a function of k



Question

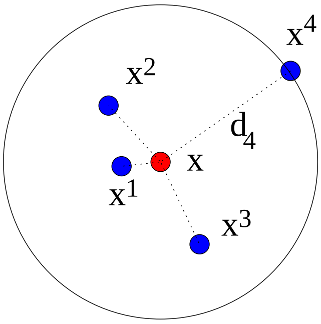
Why increasing k is worse?

Nearest Neighbors. The asymptotics.

Imagine we have infinitely many prototypes all correct (no noise, no errors).

Under some assumptions, it can be shown that the nearest neighbor of $\mathbf{x}$ is infinitely close to it (their distance tends to zero if the size of P tends to infinity).

The same holds for their k-th neighbor.



Convergence of the NN rule

Let $d_k(\mathbf{x})$ be the (asymptotic) distance from $\mathbf{x}$ to their k-th neighbor.

then $d_k(\mathbf{x}) \longrightarrow 0$ as $|P| \longrightarrow \infty$.

In another way,

$$\mathbf{x} \longrightarrow \mathbf{x^k}$$

The k-th neighbor tends to $\mathbf{x}$

This implies that (asymptotically)

$$p(\mathbf{x}|w_i) = p(\mathbf{x^k}|w_i)$$

Convergence of the NN rule

In the particular case $k = 1$ and $c = 2$ is easy to show that

Bayes risk

$$r^*(\mathbf{x}) = \min\{P(w_1|\mathbf{x}), P(w_2|\mathbf{x})\}$$

1-NN risk

$$r_1(\mathbf{x}) = P(w_1|\mathbf{x})P(w_2|\mathbf{x}) + P(w_2|\mathbf{x})P(w_1|\mathbf{x})$$

Relation to Bayes rate

$$r_1(\mathbf{x}) = 2r^*(\mathbf{x})(1 - r^*(\mathbf{x})) \leq 2r^*(\mathbf{x})$$

The overall rates

$$R_1 = 2R^*(1 - R^*) \leq 2R^*$$

Convergence of the k-NN rule

It is considerably more difficult to derive. But it is intuitively simple to see that

$$\frac{|V_k(\mathbf{x}) \cap P_i|}{k} \rightarrow P(w_i|\mathbf{x})$$

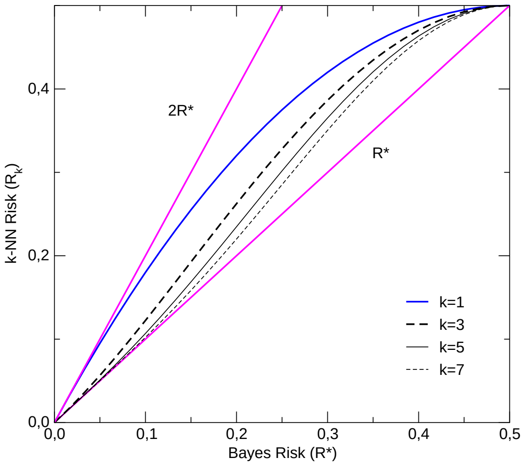
if

$$\begin{aligned} |P| &\rightarrow \infty \\ k &\rightarrow \infty \\ k/|P| &\rightarrow 0 \end{aligned}$$

Which in fact implies that the k-NN rule for (convenient) infinite k tends to the Bayes rule!

Asymptotic bounds to the *k*-NN rule

The following asymptotic bounds can be analitically derived



Summarizing

The *k*-NN rule is **asymptotically** optimal.

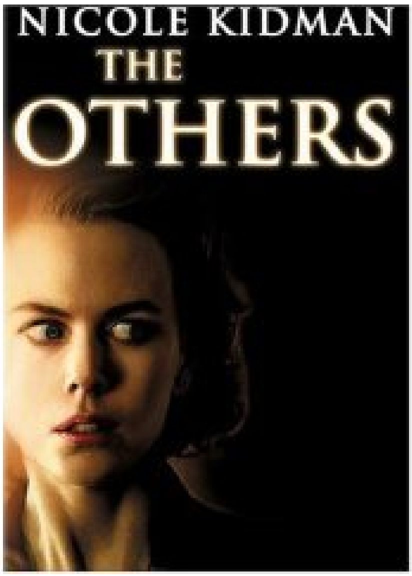
Great!

Question

What if $|P| \neq \infty$?

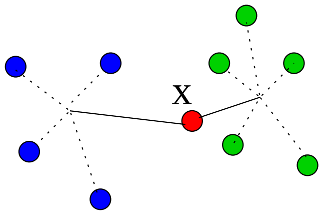
he he he ;-)

Extensions and “other” neighbors



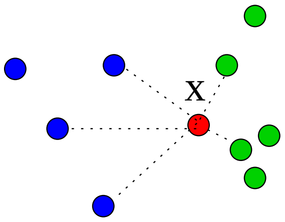
Alternative definitions

1. Averaged k-Nearest Neighbors

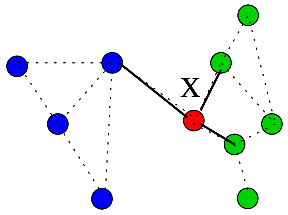


$$d(\mathbf{x}, P_i) = \frac{1}{|V_k(\mathbf{x} \cap P_i|} \sum_{\mathbf{p} \in V_k(\mathbf{x} \cap P_i} d(\mathbf{x}, \mathbf{p})$$

2. Nearest Centroid Neighbors



3. Graph neighbors



Nearest Neighbors with Reject

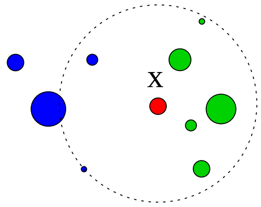
A threshold $\ell_j \leq k$ can be established for each class j in such a way that $\mathbf{x}$ is **rejected** if the majority class, i does not have at least ℓ_i neighbors.

This is called the (k, ℓ_i)-NN rule. The (k, ℓ)-NN rule is a particular case.

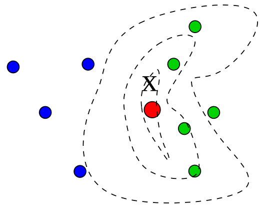
Weighted Nearest Neighbors

idea:

attach a different weight to each prototype and use the sum of weights instead of counting.



It is also possible to attach the weight to the distance



In all cases the problem is to "learn" the weights from data.

Edited Nearest Neighbors

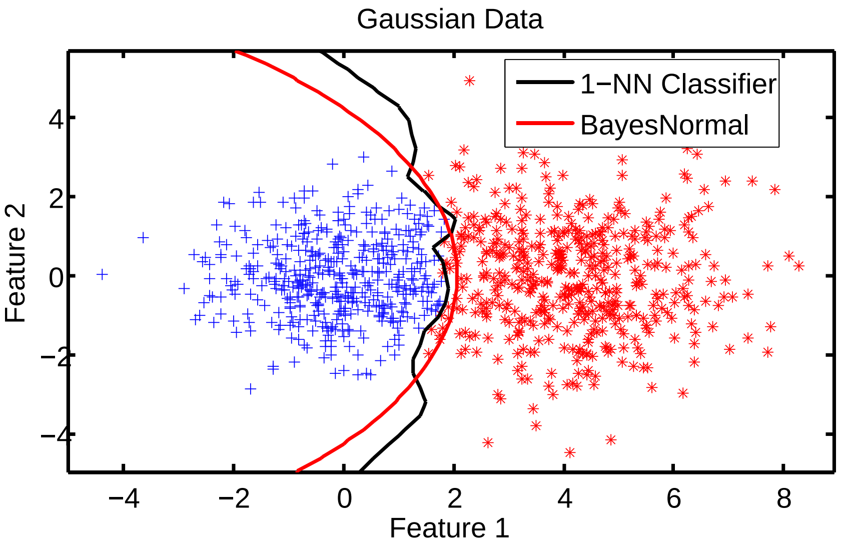
Given a particular training set, it is possible to **discard** some of them in such a way that the corresponding k-NN rule using this **edited** set gets better.

Under certain conditions, it is possible to edit (large enough) training sets in such a way that the plain 1-NN rule is Bayes optimal!!

Editing can be considered as an extreme case of weighting prototipes (0/1).

A number of different criteria and suboptimal algorithms have been proposed.

Editing



Condensed Nearest Neighbor Rule

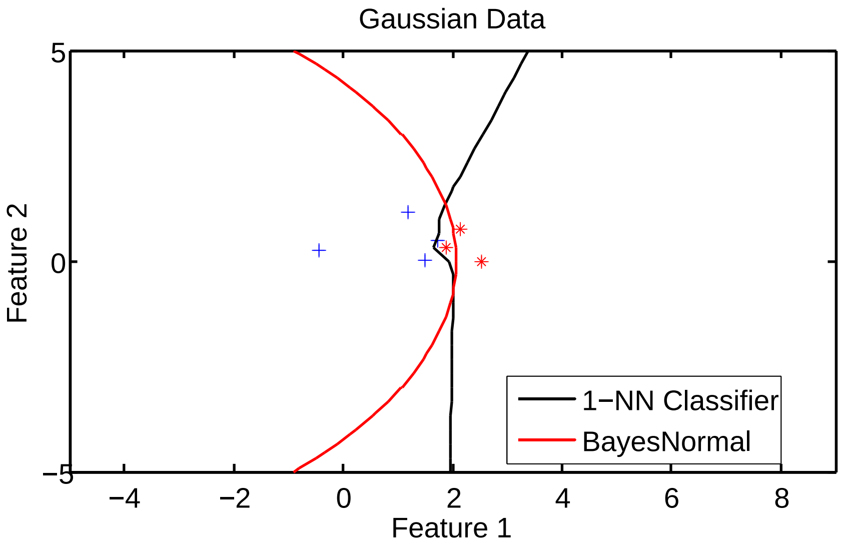
Once we are happy with a (k-) NN rule using a particular trainig set, one can consider the following problem.

Is there a subset of the training set which gives **the same** (or approximately the same) performance when used with the (k or k') NN rule?

Such subsets are known as condensed sets, and the corresponding classification rule is referred to as the condensed NN rule.

An extension of this problem is obtained when removing the constraint of the result being a subset of the training set

Condensing

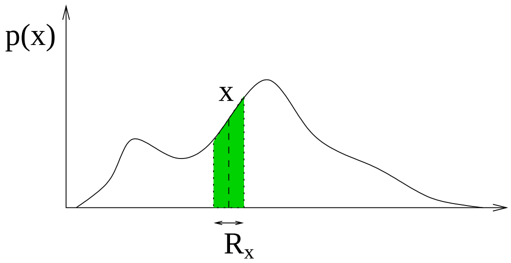


Neighbors and pdf estimates

It has already been argued that $\frac{|V_k(\mathbf{x}) \cap P_i|}{k} \longrightarrow P(w_i|\mathbf{x})$

In general, we can estimate the value of any $p(x)$ as

$$\hat{p}(x) = \int_{R_x} p(x')dx'$$

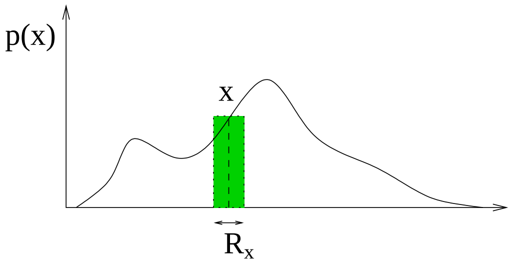


If there is an i.i.d. sample S_n of size n according to $p(x)$ we can estimate $\hat{p}(x)$ as $\frac{k}{n}$ where $k = |S_n \cap R_x|$

Neighbors and pdf estimates

Moreover, if p is smooth enough, we can approximate the above integral as $p(x) \cdot V$ where V is the **volume** of the region R. Then,

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



In fact,

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

where $V_n \rightarrow 0$, $k_n \rightarrow \infty$

Parzen Windows (or kernel estimates)

According to the previous sample-based estimate, it is possible either

- to fix k and compute estimates according to V (Nearest Neighbors)
- to fix V and compute estimates according to k (Parzen Windows)

The (fixed) volume or region is referred to as **window**. The easiest case corresponds to a d-dimensional hypercube of size h_n whose volume is h_n^d.

Let φ be the window function

$$\varphi(\mathbf{u}) = \begin{cases} 1 & |u_j| \leq \frac{1}{2}, \quad 1 \leq j \leq d \\ 0 & \text{otherwise} \end{cases}$$

Parzen Windows

The number of neighbors inside the “window” can be written as

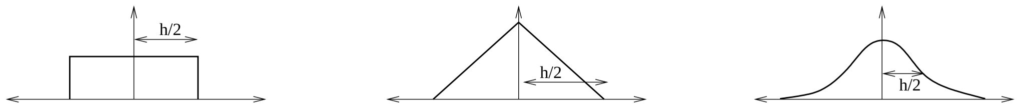
$$k_n = \sum_{i=1}^{n} \varphi(\frac{\mathbf{x} - \mathbf{p_i}}{h_n})$$

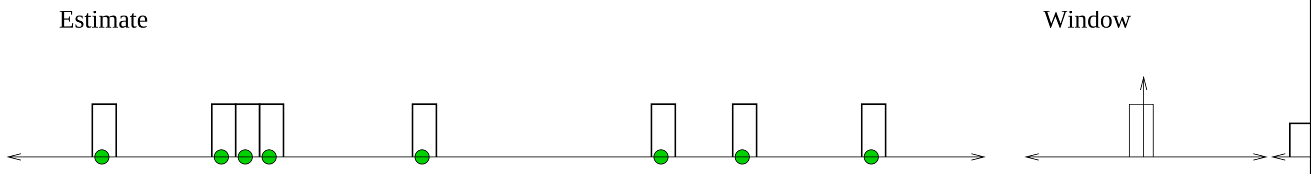
And the corresponding estimate

$$p_n(\mathbf{x}) = \frac{1}{n} \sum_{i=1}^{n} \frac{1}{V_n} \varphi(\frac{\mathbf{x} - \mathbf{p_i}}{h_n})$$

Parzen Windows

The same estimate can be generalized to a number of different window functions φ.




Estimate
Window