Part 5

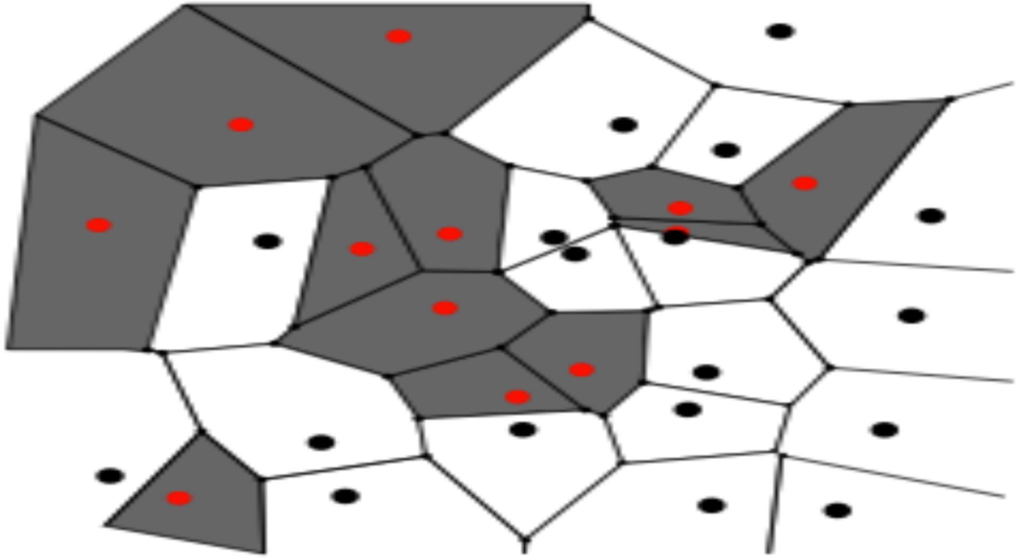
Elements of nonparametric techniques: the k-Nearest Neighbor (k-NN) classifier

Introduction

- So far (Part 4), we assumed that the forms of the probability density functions were known.
- However, this assumption cannot be done in some pattern recognition applications.
- In this chapter, we shall examine a nonparametric method that can be used with arbitrary distributions and without the assumption that the forms of the underlying densities are known.
- We will discuss the k-Nearest Neighbor (**k-NN**) pattern classifier which allows:
    - A direct estimation of the density function $p(\mathbf{x}|\omega_j)$
    - A direct estimation of the posterior probability $P(\omega_j|\mathbf{x})$

The k-Nearest Neighbor (k-NN) method

- Nonparametric classification is often associated with the notion of **prototype**.
- We can think of a prototype as a representative element from a class. The class label assigned to an example is based on the **similarity** of this example to one or more prototypes. Typically, similarity is defined in a geometrical sense, that is, based on a certain distance. The smaller the distance, the higher the similarity between x and the prototype.



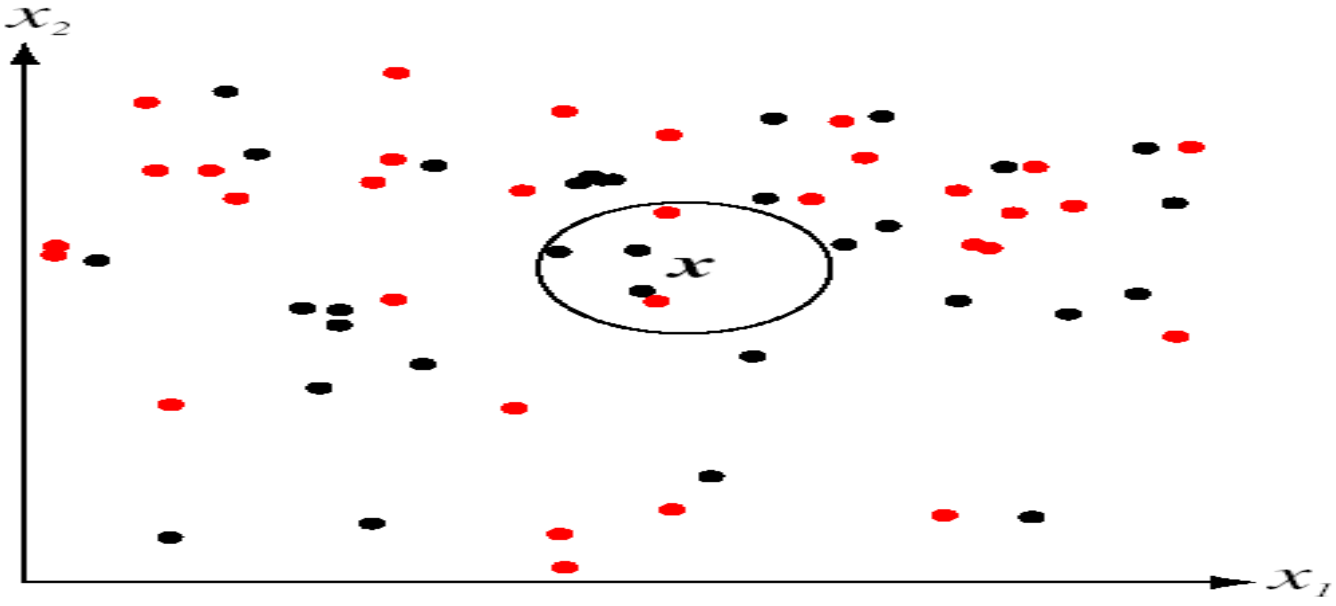
The k-Nearest Neighbor (k-NN) method

- k-nn is one of the most theoretically elegant and simple classification techniques [L. Kuncheva, 2004].
- Let D be a labeled **training set** containing n points, referred to as **prototypes**. The prototypes are labeled in the "c" classes.

$$\boldsymbol{D} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots.., \boldsymbol{x}_n]$$

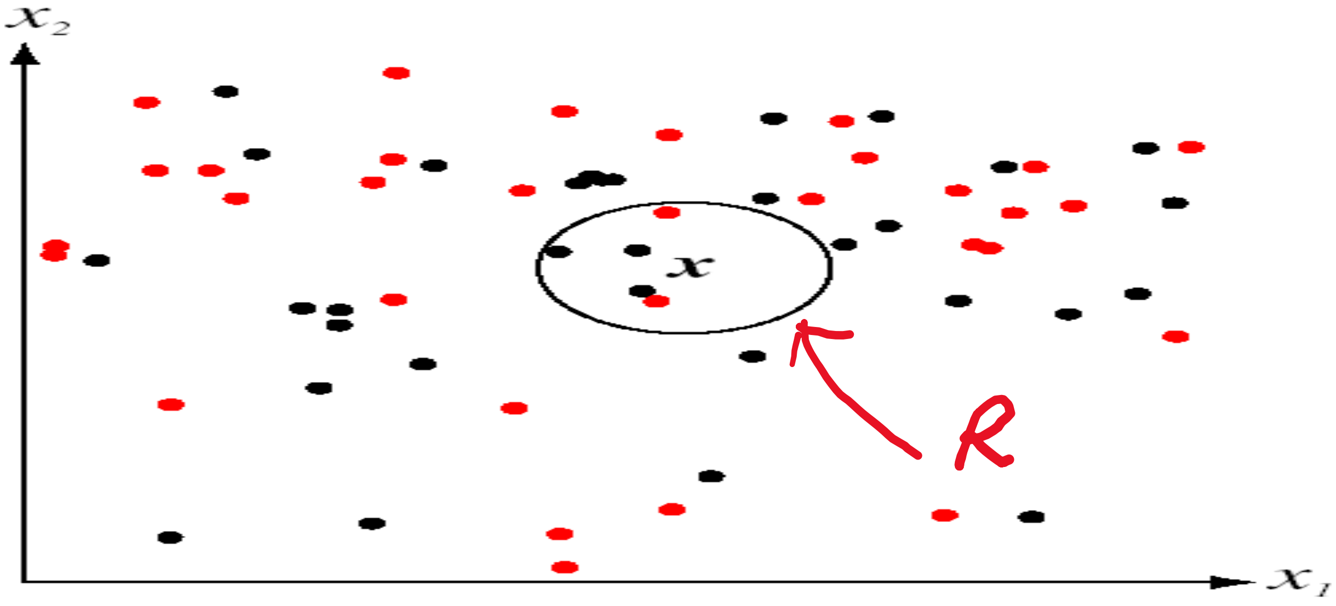
$$\boldsymbol{x}_i = (\boldsymbol{x}_{i1}, \boldsymbol{x}_{i2}, \ldots.., \boldsymbol{x}_{id}) \text{ i=1,..,n}$$

$\boldsymbol{x}_i$ belonging to one of the "c" classes ($\boldsymbol{x}_i$ ε ω_j $j=1,\ldots,c$)



The k-Nearest Neighbor (k-NN) method

- To classify an input ***x***, the **k nearest prototypes** are retrieved from D together with their class labels. The input x is labeled to the most represented class label amongst the k nearest neighbors.
  - In the figure below: ***x*** is the pattern to be classified
  - We consider a "region" *R* of the feature space containing the the **k nearest prototypes** of ***x***
  - We classify ***x*** as belonging to the most represented class label amongst the k nearest neighbors within the region *R*.



The k-Nearest Neighbor (k-NN) method

It can be shown (we see that later) that the k-NN method estimates the posterior probabilities as:

$$\hat{P}\ (\omega_i/\mathbf{x}) = \frac{k_i}{k}$$

- Where k_i is the number of nearest neighbors belonging to the class ω_i within the region *R*.
- *k* is the number of the **k nearest prototypes** of ***x*** within the “region” *R*

•The minimum error (Bayes) classifier using the approximations above will assign x to the class with the highest posterior probability, that is, the class most represented among the k nearest neighbors of x.

How to select the right value of «k» parameter

One simple rule of thumb (heuristic rule) is to select k as:

$$k = \sqrt{n}$$

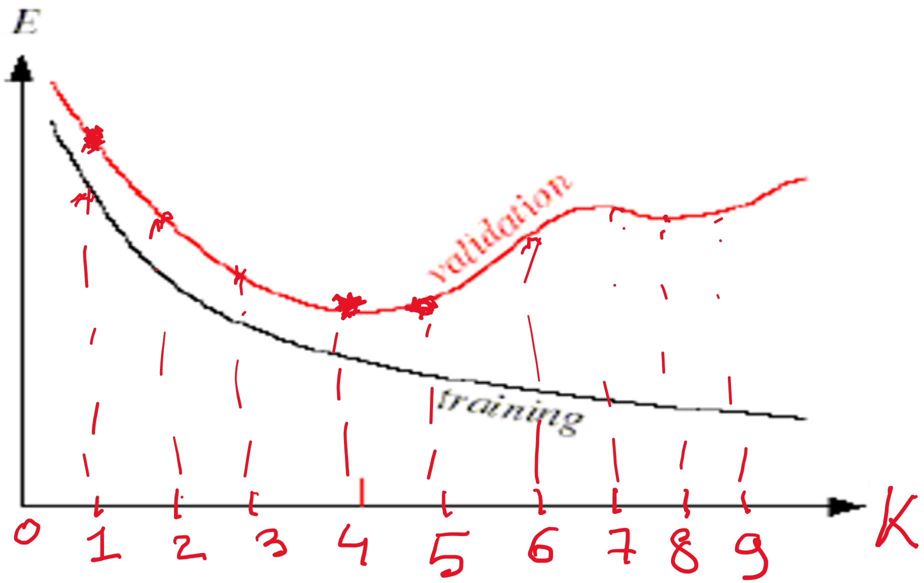
- n is the number of examples in our training set D.

- Note that this rule imposes that the number k of nearest neighbors of the pattern x that fall within the region R is smaller than the total number n of training set examples.

- In binary problems (two class) or problems with **even** number of clasess, it is helpful to choose k to be an odd number as this avoids **tie breaks**.

How to select the right value of «k» parameter

- An experimental method for selecting the value of the «k» parameter is «cross validation» (see Chapter 8).
- We subdivide our original data set D into three subsets : training set, validation set, and test set
- We use training set as the set containing the “prototypes”
- Simple method: we evaluate error E using different values of the «k» parameter with the validation set (more on this in Chapter 8)



The k-Nearest Neighbor (k-NN) method

- In the next slides, we see the theoretical concepts behind the k-NN method and how one can arrive to the approximation below:

$$\hat{P}\ (\omega_i/\mathbf{x})=\frac{k_i}{k}$$

Density estimation for the k-nn method

•The basic idea underlying many of the non-parametric methods is very simple, and it can be illustrated as follows.

•The probability ***P*** that a vector **x** will fall in a region $\mathcal{R}$ of the feature space is:

$$P = \int_{\Re} p(\mathbf{x}')d\mathbf{x}'$$

- Note that, if *R* is a small region, *P* can be regarded as a smoothed or averaged version of the density function p(x).
- Therefore, we can estimate this smoothed value of p(x) by estimating the probability P.

Density estimation

Suppose that n samples $\mathbf{x}_1,\ldots,\mathbf{x}_n$ are drawn independently and identically distributed (i.i.d.) according to the probability law p(**x**).

If we know that **k** samples of these **n** fall in *R*, then ***P*** can be estimated simply as ***P*** = k/n. In general, this can be proved as follows.

The probability that **k** of the **n** samples fall in *R* is given by the binomial law:

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

and we know that the expected value of the binomial law for k is :

$$\varepsilon(k) = nP$$

Density estimation

- If we consider the ratio **k/n** as argument of the binomial distribution, we can rewrite the expected value for k/n as:

$$\varepsilon(k/n) = P$$

- We know that the variance of the binomial law is *var*(k/n)=P(1-P)/n

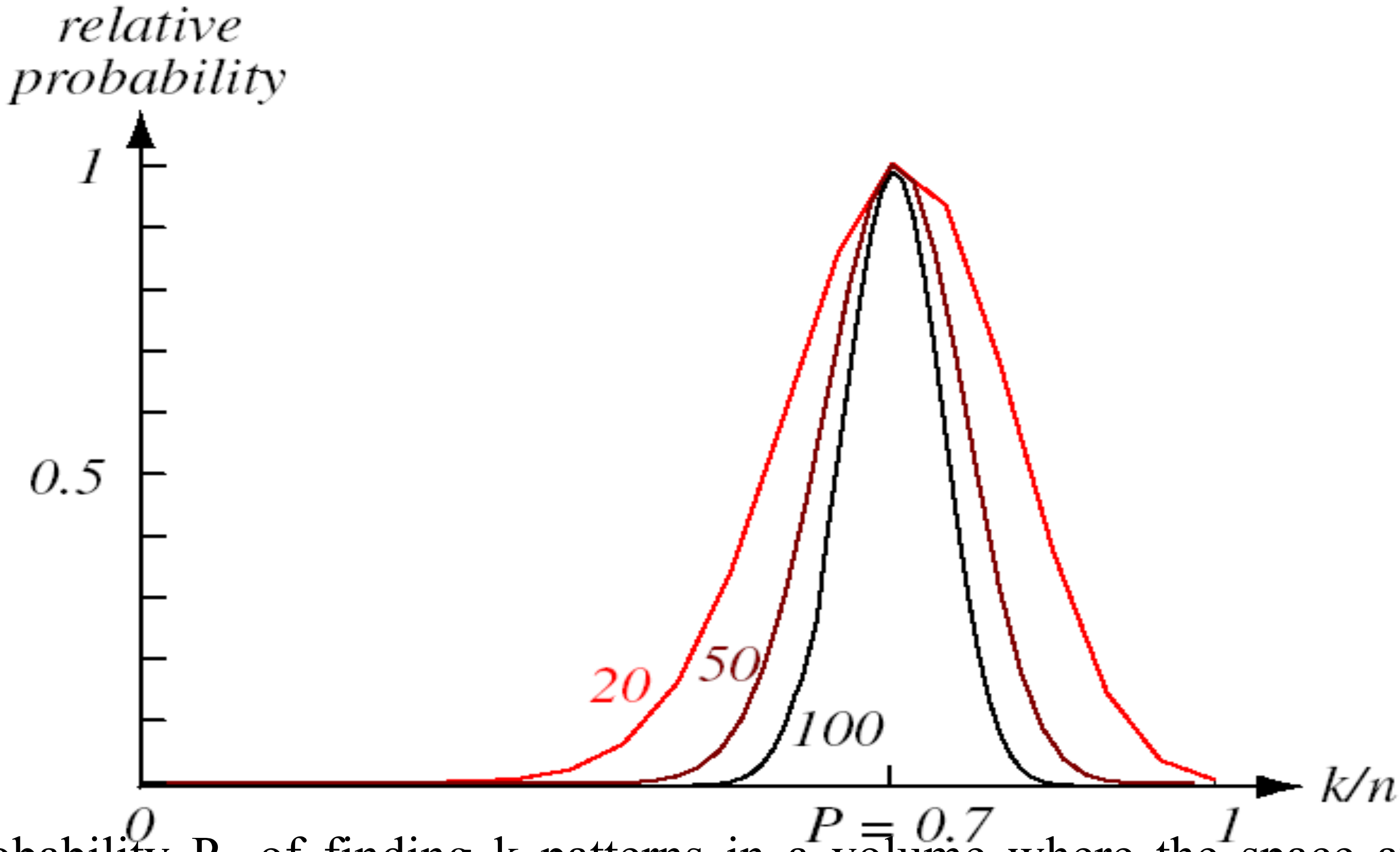
- If the number of samples n increases, the limit when it goes to infinity is:

$$n \to +\infty \;\; \to \;\; \varepsilon(k/n) = P \text{ and } \mathrm{var}(\mathrm{k/n}) = 0$$

- Therefore, we can say that k/n is an asymptotically unbiased estimator of P.
- We expect that the ratio k/n will be a very good estimate for the probability P, and hence for the smoothed density function if n is very large.

➢ Indeed, this estimate is especially accurate when n is very large (see next Figure).

Density estimation: binomial distribution



•The probability P_k of finding k patterns in a volume where the space averaged probability is P as a function of k/n. Each curve is labelled by the total number of patterns n. For large n, such binomial distributions peak strongly at k/n = P (here chosen to be 0.7).

Density estimation

- If we now assume that p(x) is continuous and that the region R is so small that p(x) does not vary appreciably within it, we can write:

$$\int_{\Re} p(\mathbf{x}')d\mathbf{x}' \cong p(\mathbf{x})V = P$$

- where x is a point within R and V is the volume enclosed by R.

- Combining previous equations, we arrive at the following obvious estimate for p(x):

$$\int_{\Re} p(\mathbf{x}')d\mathbf{x}' \cong p(\mathbf{x})V \cong k/n \;\rightarrow\; p(\mathbf{x}) \cong \frac{k/n}{V}$$

We assume that these two approximations can be made equal.

Density estimation

$$\int_{\Re} p(\mathbf{x}')d\mathbf{x}' \cong p(\mathbf{x})V \cong k/n \;\rightarrow\; p(\mathbf{x}) \cong \frac{k/n}{V}$$

- If we fix the volume V and take more and more training samples, the ratio k/n will converge (in probability) as desired, but we have only obtained an estimate of the space-averaged value of p(x):

$$\frac{P}{V} = \frac{\int_{\Re} p(\mathbf{x}')d\mathbf{x}'}{\int_{\Re} d\mathbf{x}'}$$

Issues:

✓ if we want to obtain p(x) rather than just an averaged version of it, we must be prepared to let V approach zero.

✓ From a practical standpoint, we note that the number of samples is always limited. Thus, the volume V can not be allowed to become arbitrarily small.

Density estimation

- In practical applications, the number of samples is always limited.
- Thus, the volume V can not be allowed to become arbitrarily small.
- ➢ Therefore, one will have to accept a certain amount of variance in the ratio k/n and a certain amount of averaging of the density p(x).

From a theoretical standpoint, it is however interesting to ask how these limitations can be circumvented if an unlimited number of samples would be available. We discuss that in the next slides.

Density estimation: convergence

To estimate the density at x, we form a sequence of regions R_1, R_2,... R_n, containing x — the first region to be used with one sample, the second with two, and so on. Let V_n be the volume of R_n, k_n be the number of samples falling in R_n, and $p_n(x)$ be the nth estimate for p(x):

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

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

$$\lim_{n\to\infty} V_n = 0$$

$$\lim_{n\to\infty} k_n = \infty$$

$$\lim_{n\to\infty} \frac{k_n}{n} = 0$$

Density estimation: convergence

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

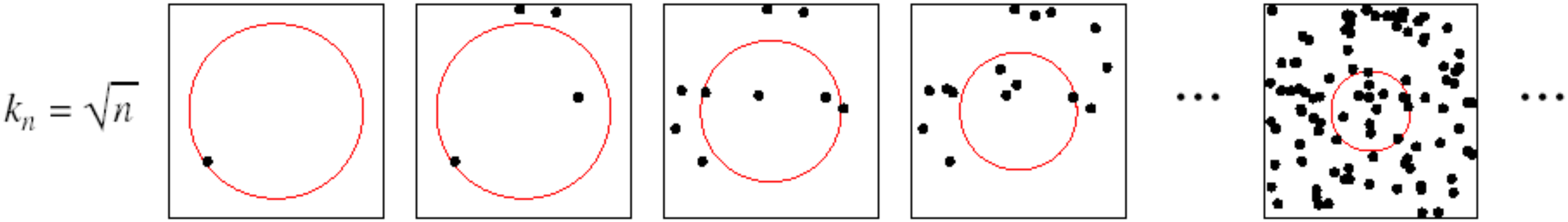
$$\lim_{n\to\infty} V_n = 0$$

$$\lim_{n\to\infty} k_n = \infty$$

$$\lim_{n\to\infty} \frac{k_n}{n} = 0$$

- ✓ The first condition assures us that the space averaged P/V will converge to p(x), provided that the regions shrink uniformly and that p(·) is continuous at x.
- ✓ The second condition, which only makes sense if $p(x) \neq 0$, assures us that the frequency ratio will converge (in probability) to the probability P.
- ✓ The third condition is clearly necessary if $p_n(x)$ is to converge at all. It also says that although a huge number of samples will eventually fall within the small region R_n, they will form a negligibly small fraction of the total number of samples.

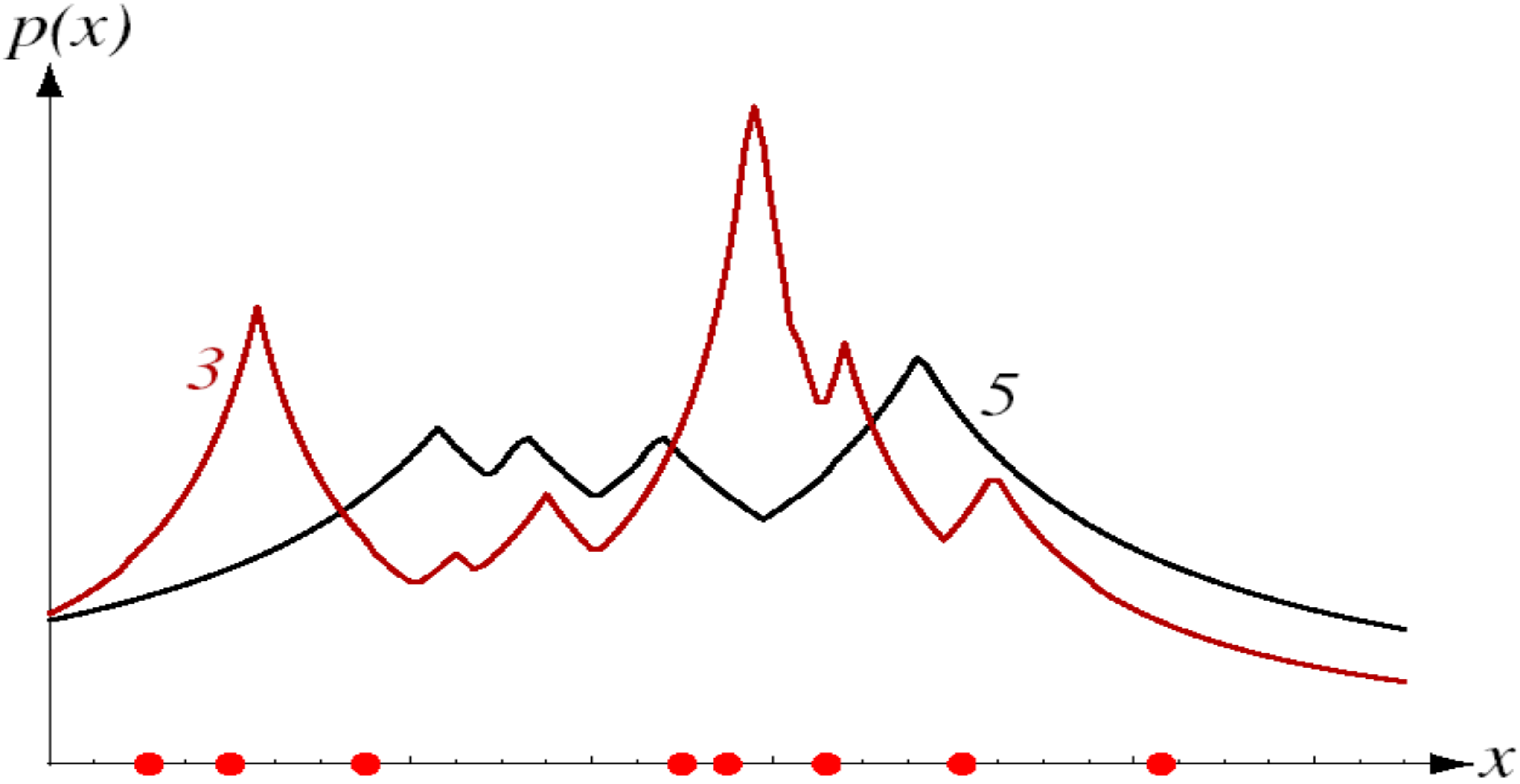
“k” nearest neighbor method: value of «k»



➢k Nearest-Neighbor non-parametric method

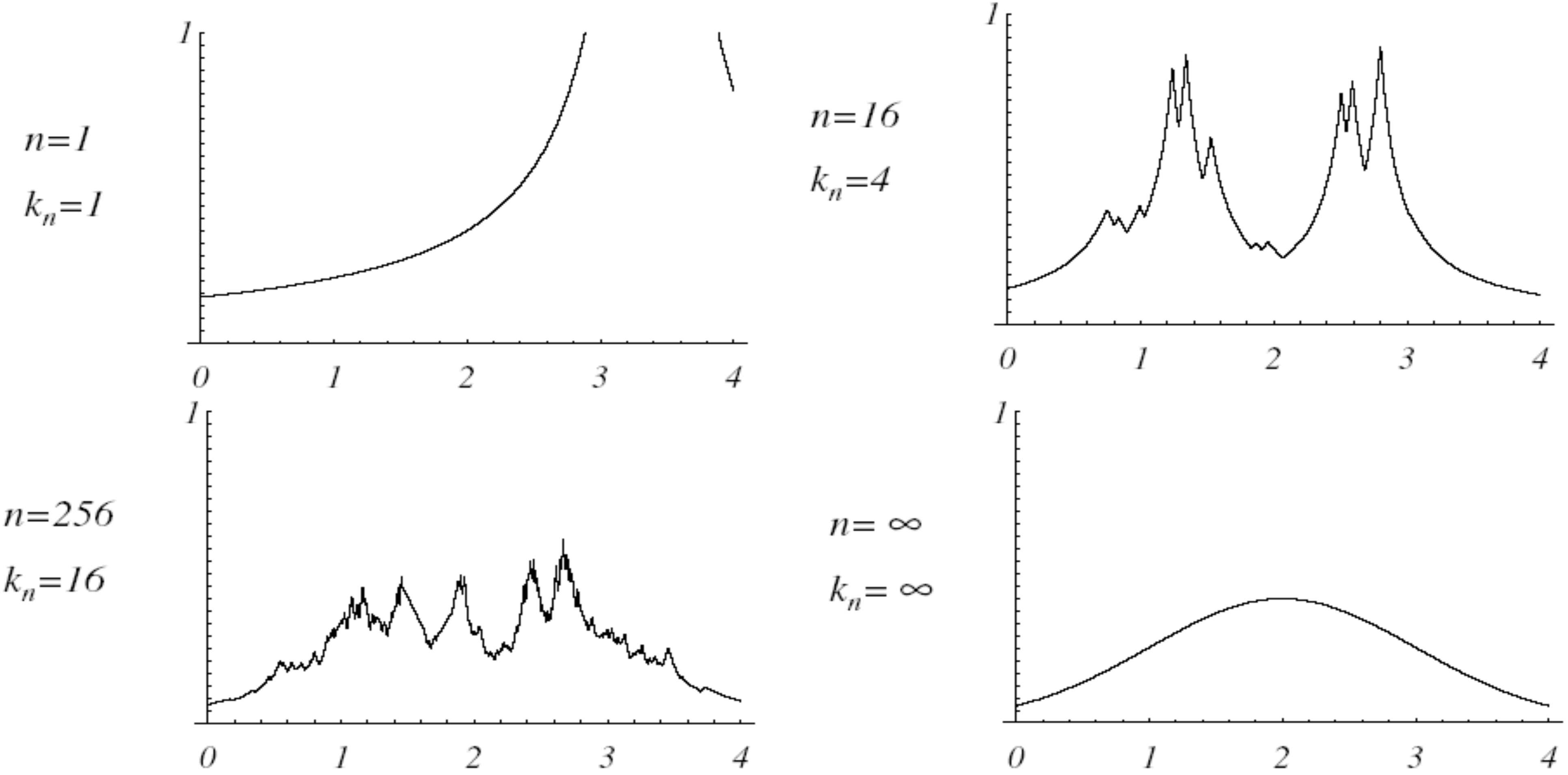
- This method specifies k_n as some function of n, such as $k_n = \sqrt{n}$. Here the volume V_n is grown until it encloses k_n neighbors of x.
- **Key concept:** the region (volume) is specified taking into account the number k of the samples which fall into it.

An extreme example of density estimation with k-nn



•Eight points in one dimension and the k-nearest-neighbor density estimates, for k = 3 and 5. Note especially that the discontinuities in the slopes in the estimates generally occur away fom the positions of the points themselves.

An example of density estimation with k-nn



•Several k-nearest-neighbor estimates of unidimensional densities: a Gaussian distribution. Notice how the finite n estimates can be quite “spiky.”

The k-Nearest Neighbor (k-NN) method

- In the previous slides, we have shown that:

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

- Now it is easy to show that the k-NN method estimates the posterior probabilities as :

$$P(\omega_i/\mathbf{x}) = \frac{k_i}{k}$$

where k_i is the number of nearest neighbors belonging to the class ω_i within the region *R.*

➢ See the detailed explanation in the next slides

The k-Nearest Neighbor (k-NN) method

- To arrive at this classification method, we fix k_n and n and allow for a variable V_n. Assuming Euclidean distance, let R be the region containing exactly k_n of the elements of the training set D. We know that the unconditional p.d.f. can be approximated as

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

- Denoting by k_i the number of elements in R from class ω_i, the class-conditional pdf for ω_i, i=1,...,c, can be approximated in R, as

$$p(\mathbf{x} / \omega_i) = \frac{k_i / n_i}{V_n}$$

The k-Nearest Neighbor (k-NN) method

- The posterior probabilities are obtained as

$$P_n(\omega_i \mid \mathbf{x}) = \frac{p_n(\mathbf{x}/\omega_i)P(\omega_i)}{p_n(x)} = \frac{\dfrac{k_i/n_i}{V}\dfrac{n_i}{n}}{\dfrac{k/n}{V}} = \frac{k_i}{k}$$

- The minimum error (Bayes) classifier using the approximations above will assign x to the class with the highest posterior probability, that is, the class most represented among the k nearest neighbors of x.

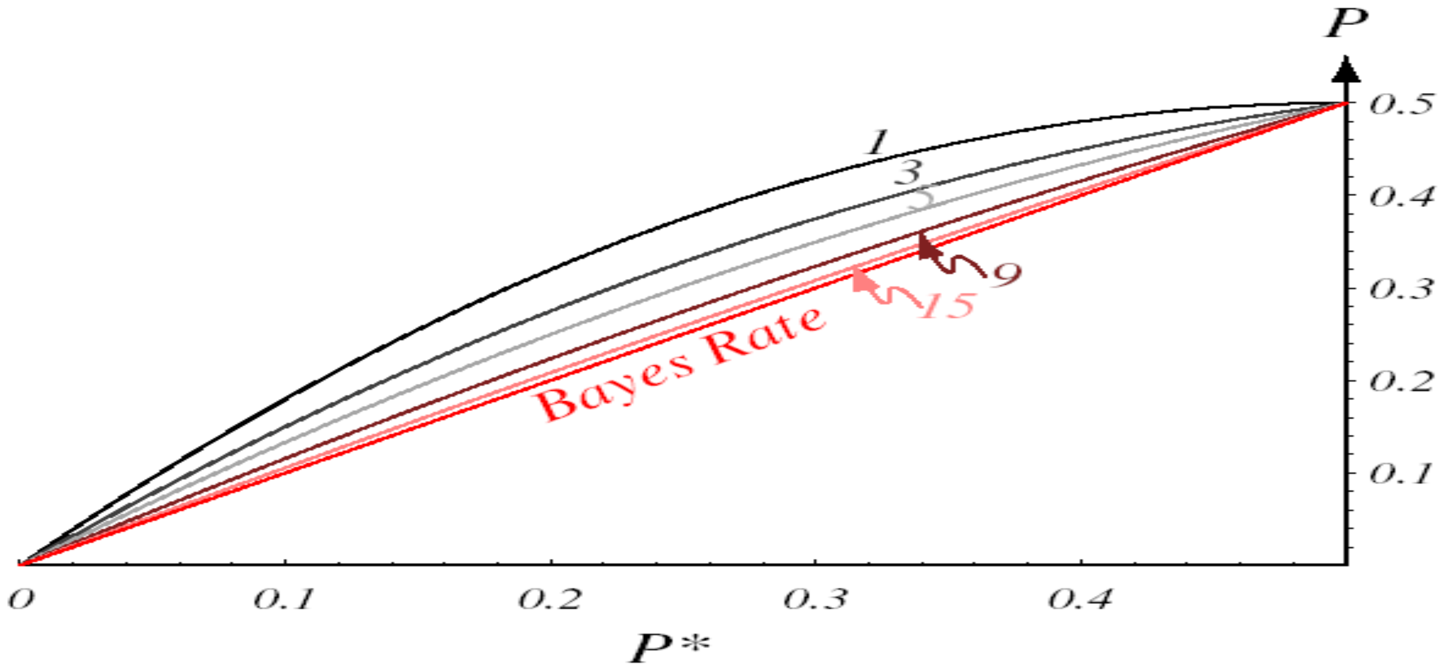
- The region R and the volume V, respectively, are specific for each x. The k-nn classification rule, however, assigns the class label using only the numbers k_i, so the winning label does not depend on V.

The k-Nearest Neighbor (k-NN) method

- k-nn is Bayes-optimal when:

$$\lim_{n\to\infty} k_n = \infty$$

$$\lim_{n\to\infty} \frac{k_n}{n} = 0$$



- The error-rate for the k-nearest-neighbor rule for a two-category problem. Each curve is labelled by k; when k=∞, the estimated probabilities match the true probabilities and thus the error rate is equal to the Bayes rate, i.e., $P = P_{Bayes}$.

The Nearest-Neighbor Rule

•Let be $D^n=\{x_1,..,x_n\}$ a training set of samples belonging to the "c" classes $\omega_1,..,\ \omega_c$ and be $\mathbf{x}'\ \varepsilon\ D^n$ the nearest sample to the unknown sample **x**.

➢Decision rule: we assign **x** to the class of $\mathbf{x}'$

- The nearest-neighbor rule is a sub-optimal procedure; its use will usually lead to an error rate greater than the minimum possible, the Bayes rate. However, with an infinite number of prototypes the error rate is never worse than twice the Bayes rate.

The Nearest-Neighbor Rule

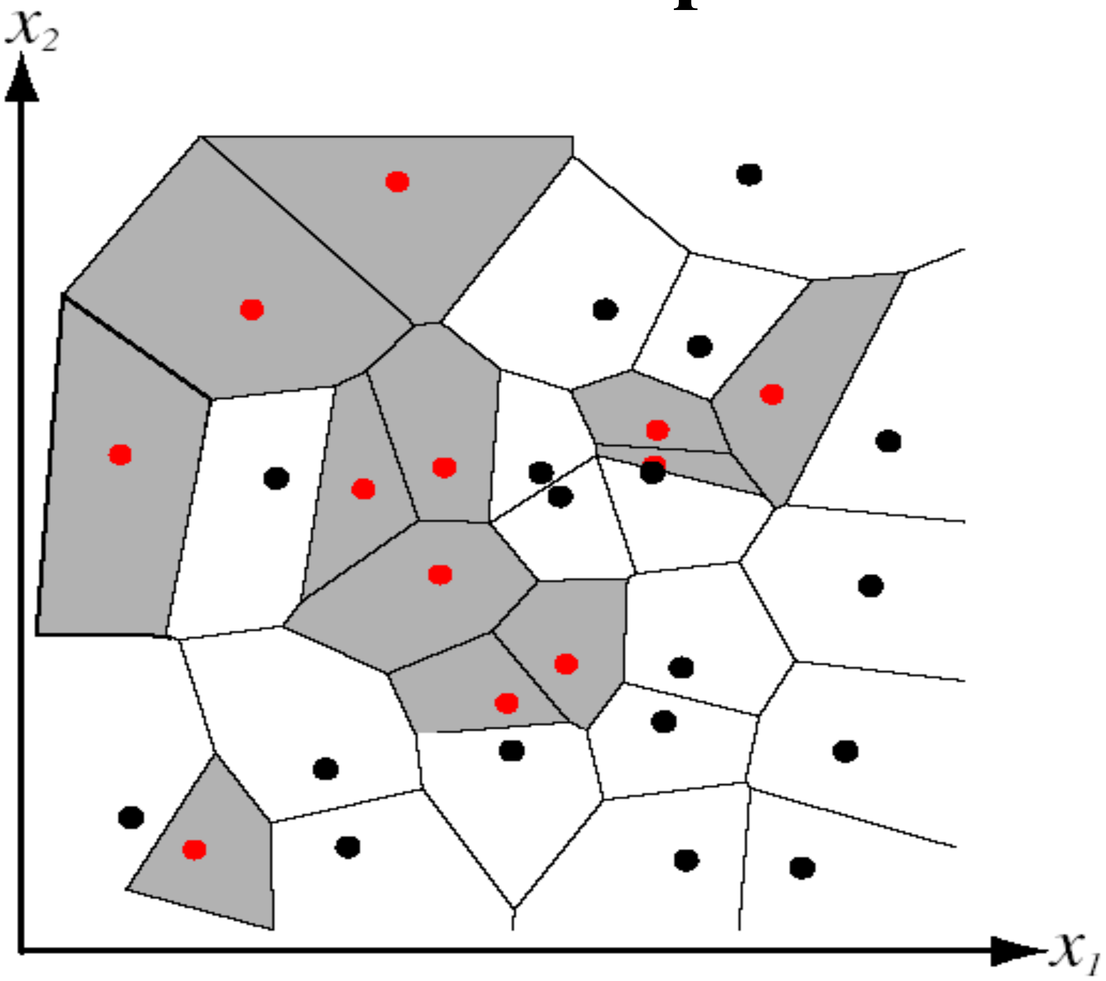
- Let be ω’ the true class of **x’**. The probability that ω’= ω_i is P(ω_i|**x**´)

- If “n” is very large, we can assume that **x**´ is very close to **x**, so that P(ω_i|**x**´)= P(ω_i|**x**); here the nearest-neighbour decision rule is in a good agreement with the MAP rule.

- If we specify ω_m(**x**) by $P(\omega_m \mid \mathbf{x}) = \max_i P(\omega_i \mid \mathbf{x})$
The MAP rule assigns the pattern to the class ω_m
- If **x**´ has been assigned to the class “j”, then we should assume that ω_m(**x’**)= ω_j

- This rule allows us to partition the feature space into cells consisting of all points closer to a given training point x′ than to any other training points. All points in such a cell are thus labelled by the category of the training point — a so-called **Voronoi tesselation** of the space (see next slide)

Voronoi tesselation: 2D example



•In two dimensions, the nearest-neighbor algorithm leads to a partitioning of the input space into Voronoi cells, each labelled by the category of the training point it contains.

The Nearest-Neighbor Rule: some remarks

- If $P(\omega_m|\mathbf{x})\cong 1$ the nearest-neighbor rule is close to the optimal rule, as it is very unlikely that the posterior probability changes abruptly.

- That is, when the minimum probability of error is small, the nearest-neighbor probability of error is also small.

- If $P(\omega_m|\mathbf{x})\cong 1/c$, the classes have the same probabilities, and the nearest-neighbor rule is likely suboptimal.

- In this case the error probability is about 1-1/c for both methods.

References

- Sections 4.1, 4.2, 4.3, 4.4, 4.5, Pattern Classification, R. O. Duda, P. E. Hart, D. G. Stork, John Wiley & Sons, 2000
- L. Kuncheva, Combining pattern classifiers, Wiley, 2004.