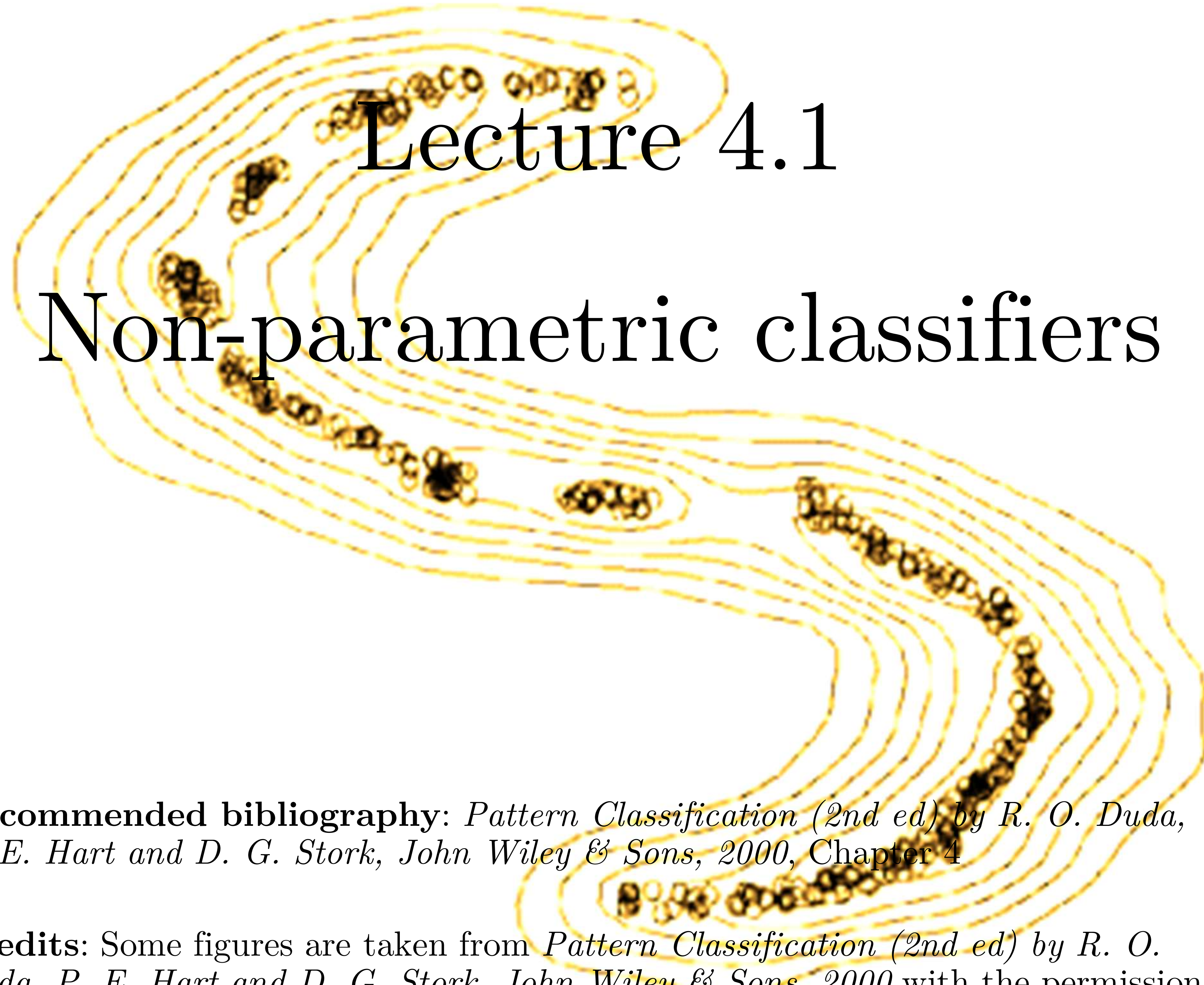


# Lecture 4.1

# Non-parametric classifiers

**Recommended bibliography**: *Pattern Classification (2nd ed) by R. O. Duda, P. E. Hart and D. G. Stork, John Wiley & Sons, 2000*, Chapter 4

**Credits**: Some figures are taken from *Pattern Classification (2nd ed) by R. O. Duda, P. E. Hart and D. G. Stork, John Wiley & Sons, 2000* with the permission of the authors

# Course overview

Pre-processing → Feature selection → Classification

| Techniques |
| --- |
| PCA (lecture 3)<br>MDA (lecture 3) |

| | Density function for classes | Classifier |
| --- | --- | --- |
| Supervised | Parametrized Gaussian | Linear discriminant<br>Quadratic discriminant[1]<br>(lecture 2) |
| Supervised | Non-parametric | K-nearest neighbours[2]<br>Fuzzy logic<br>(lecture 4) |
| Supervised | Unknown | SVM[3]<br>Neural networks[3]<br>Decision trees[4]<br>(lecture 4) |
| Unsupervised | Parametrized Gaussian | K-means<br>(lecture 6) |
| Unsupervised | Unknown | Clustering<br>(lecture 6) |

1. Useful only if covariance matrices are not rank deficient.
2. Useful with the number of features is very large, even larger that the number of training vectors.
3. Imposes a structure to the classifier irrespective of the training data base.
4. Useful when non-numeric features are present.

# CONTENTS

# 1 NON-PARAMETRIC ESTIMATION OF THE PDF

If we cannot assume a model for $f_{\mathbf{x}}(\mathbf{x}|\omega_i)$ we have to resort to non-parametric estimators, like the histogram. A precise definition follows:

- The probability that $\mathbf{x}$ is in region $R$ is:

$$P = \int_R f_{\mathbf{x}}(\mathbf{x}')d\mathbf{x}'$$

- If we have $n$ independent observations $D = \{\mathbf{x}_1,\ldots, \mathbf{x}_n\}$, the probability of having $k$ among the $n$ vectors in the region is:

$$P_k = \binom{n}{k} P^k (1-P)^{n-k} \qquad \Rightarrow \qquad E\{k\} = nP$$

The ML estimator of $P$ is: $\hat{P} = k / n$

If $f_{\mathbf{x}}(\mathbf{x})$ is continuous and $R$ is small enough that $f_{\mathbf{x}}(\mathbf{x})$ does not change:

$$\int_R f_{\mathbf{x}}(\mathbf{x}')d\mathbf{x}' \cong f_{\mathbf{x}}(\mathbf{x})\ V_R$$

Combining both expressions we get the histogram:

$$f_{\mathbf{x}}(\mathbf{x}) = \frac{\int_R f_{\mathbf{x}}(\mathbf{x}')d\mathbf{x}'}{\int_R d\mathbf{x}'} \cong \frac{k/n}{V_R}$$

# CONVERGENCE CONDITIONS

The histogram is averaging values in a region, and hence it is a distorted versión of $f_{\mathbf{x}}(\mathbf{x})$.

To reduce the effect we are interested in having $V_R \to 0$, which implies $k \to 0$ if the number of samples is finite.

How can we guarantee the convergence of

$$f_n(\mathbf{x}) = \frac{k_n / n}{V_{R,n}}$$

when $n \to \infty$?

How shall we design $V_{R,n}$?

Having $\lim_{n\to\infty} f_n(\mathbf{x}) = f(\mathbf{x})$ implies:

$$\boxed{\begin{aligned} &\lim_{n\to\infty} V_{R,n} = 0 \\ &\lim_{n\to\infty} k_n = \infty \\ &\lim_{n\to\infty} \frac{k_n}{n} = 0 \end{aligned}}$$

We can guarantee the three conditions in two ways:

1. Parzen windows: taking $V_{R,n}$ as a function of $n$.
2. k-nearest neighbors: adapting $V_{R,n}$ in every region of the domain of $\mathbf{x}$ in such a way that $k$ grows at a smaller rate than $n$.

# 2 PARZEN WINDOWS

Assume that region $R$ is defined by a function $\varphi(\mathbf{x})$ enclosing a hiper-volume $V_{R,n}$ around $\mathbf{x}$. A measure of the number of vectors inside this region is:

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

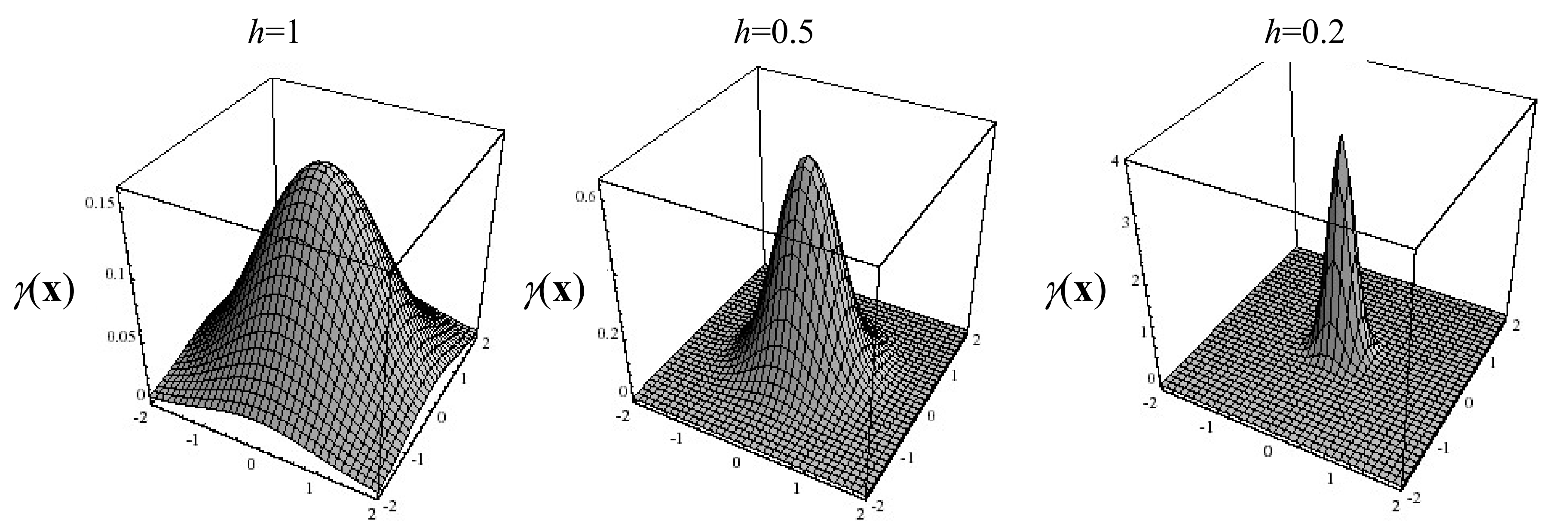
The estimation of the pdf is given by:

$$f_n(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n} \frac{1}{V_{R,n}} \varphi\left(\frac{\mathbf{x}-\mathbf{x}_i}{h_n}\right) = \frac{1}{n}\sum_{i=1}^{n} \gamma_n(\mathbf{x}-\mathbf{x}_i)$$

The condition for $\varphi(\mathbf{x})$ such that $f_n(\mathbf{x})$ be a pdf are:

$$\varphi(\mathbf{x}) \geq 0$$
$$\int \varphi(\mathbf{x}) d\mathbf{x} = 1$$
$$V_n \propto h_n^d$$

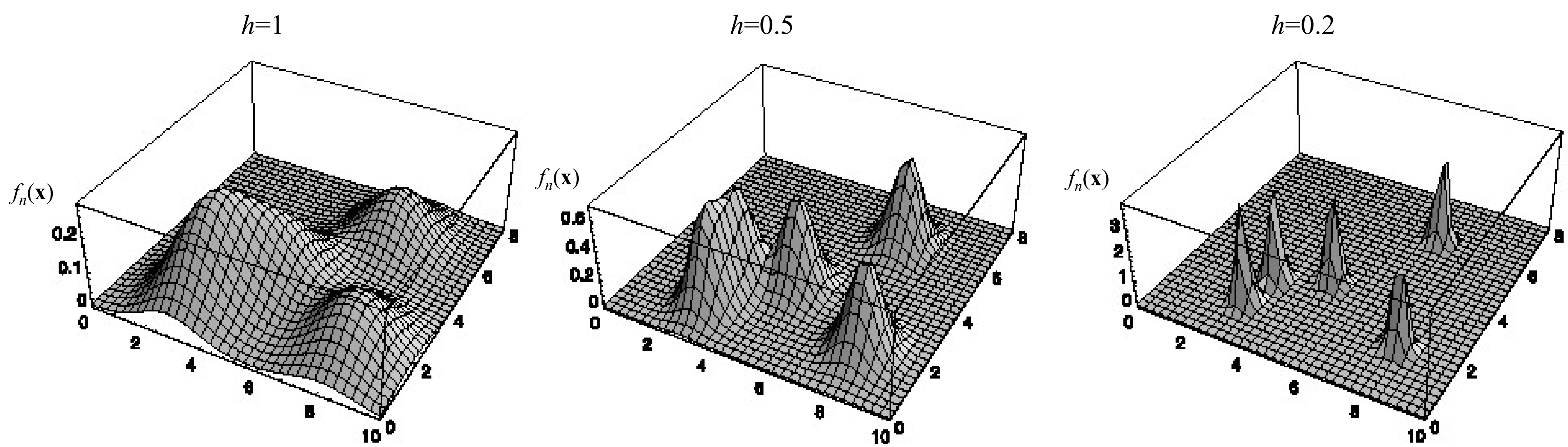
Example 1:




Will give a very much averaged estimation of the pdf

Will give a noisy estimation of the pdf

Estimation of $f_n(\mathbf{x})$ obtained with $n = 5$ vectors, for three different values of $h$:




Usually radial basis functions are used for $\varphi(\mathbf{x})$. These are defined as functions for which:

$$\|\mathbf{x}_1\| = \|\mathbf{x}_2\| \quad \Rightarrow \quad \varphi(\mathbf{x}_1) = \varphi(\mathbf{x}_2)$$

# MEAN AND VARIANCE OF THE HISTOGRAM

## Mean

$$\bar{f}_n(\mathbf{x}) = E\left\{f_n(\mathbf{x})\right\} = \frac{1}{n}\sum_{i=1}^{n} E\left\{\frac{1}{V_{R,n}}\varphi\left(\frac{\mathbf{x}-\mathbf{x}_i}{h_n}\right)\right\} =$$

$$= \int \frac{1}{V_{R,n}}\varphi\left(\frac{\mathbf{x}-\mathbf{v}}{h_n}\right) f(\mathbf{v})d\mathbf{v} = \int \gamma_n\left(\mathbf{x}-\mathbf{v}\right) f(\mathbf{v})d\mathbf{v}$$

It is a convolution of the window with the true pdf

## Interpretation

If $V_{R,n} \to 0$, then $\varphi(\mathbf{x}) \to \delta(\mathbf{x})$ (a Dirac delta function) and the estimator is non-biased, but the number of vectors in each region tends to zero and the estimation will not be good $\Rightarrow$ we have to evaluate the variance.

## Variance

$$\sigma_n^2(\mathbf{x}) = \sum_{i=1}^{n} E\left\{\left(\frac{1}{nV_{R,n}}\varphi\left(\frac{\mathbf{x}-\mathbf{x}_i}{h_n}\right) - \frac{1}{n}\bar{f}_n(\mathbf{x})\right)^2\right\} =$$

$$= nE\left\{\frac{1}{n^2V_{R,n}^2}\varphi^2\left(\frac{\mathbf{x}-\mathbf{x}_i}{h_n}\right)\right\} - \frac{1}{n}\bar{f}_n^2(\mathbf{x}) =$$

$$= \frac{1}{nV_{R,n}}\int \frac{1}{V_{R,n}}\varphi^2\left(\frac{\mathbf{x}-\mathbf{v}}{h_n}\right) f(\mathbf{v})d\mathbf{v} - \frac{1}{n}\bar{f}_n^2(\mathbf{x})$$

Upper bound: $$\sigma_n^2(\mathbf{x}) \le \frac{\sup\left(\varphi(.)\right)\bar{f}(\mathbf{x})}{nV_{R,n}}$$

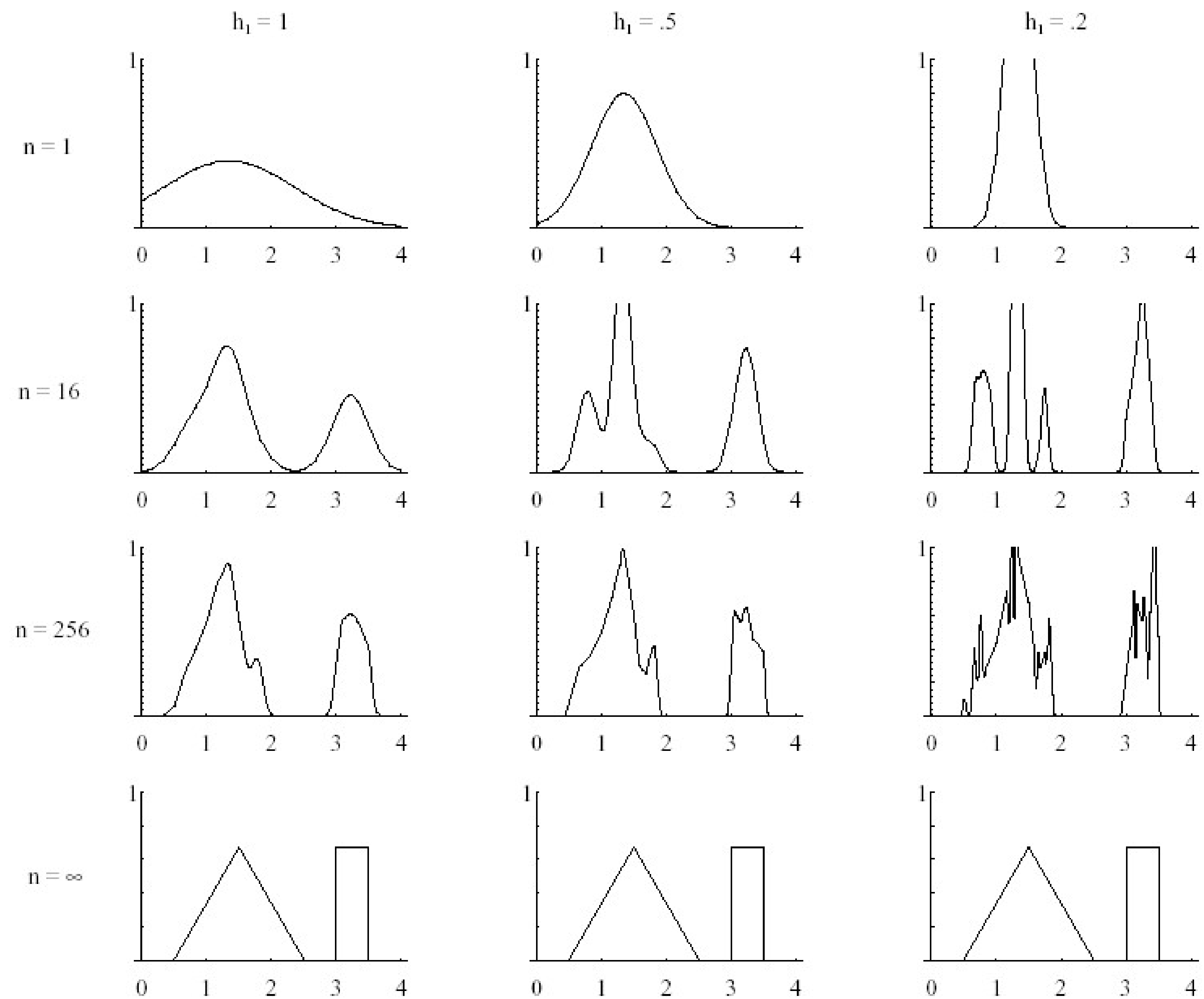
## Interpretation

Given $n$, a low variance implies a large $V_{R,n}$ $\Rightarrow$ large bias

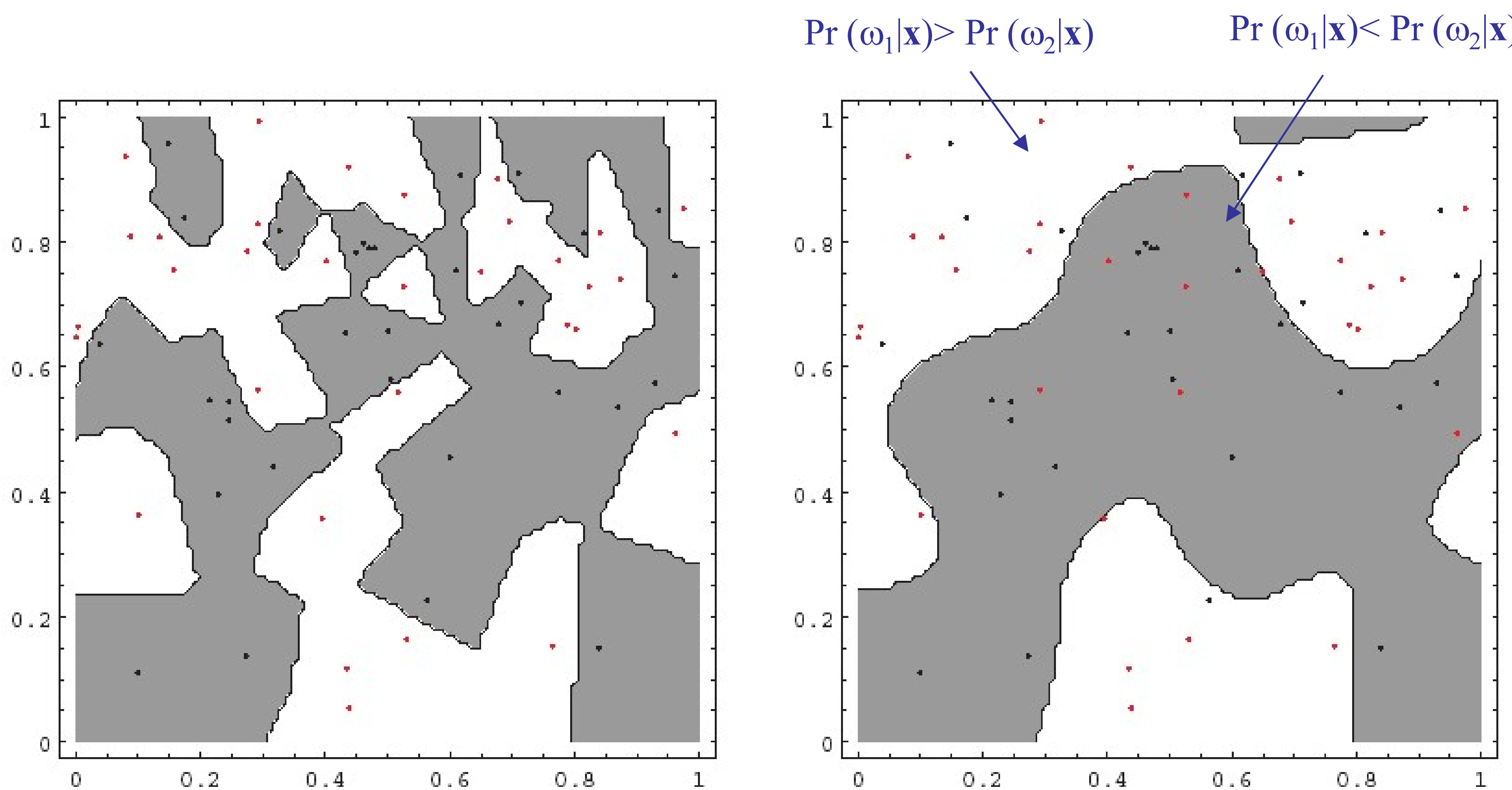
Variance can be small if $V_{R,n}$ is large, when $n \to \infty$.

Example 2:

Gaussian window



Example 3: Decision boundaries for a two classes problem using Parzen Gaussian windows for a small (left) and large value (right) of $h$.

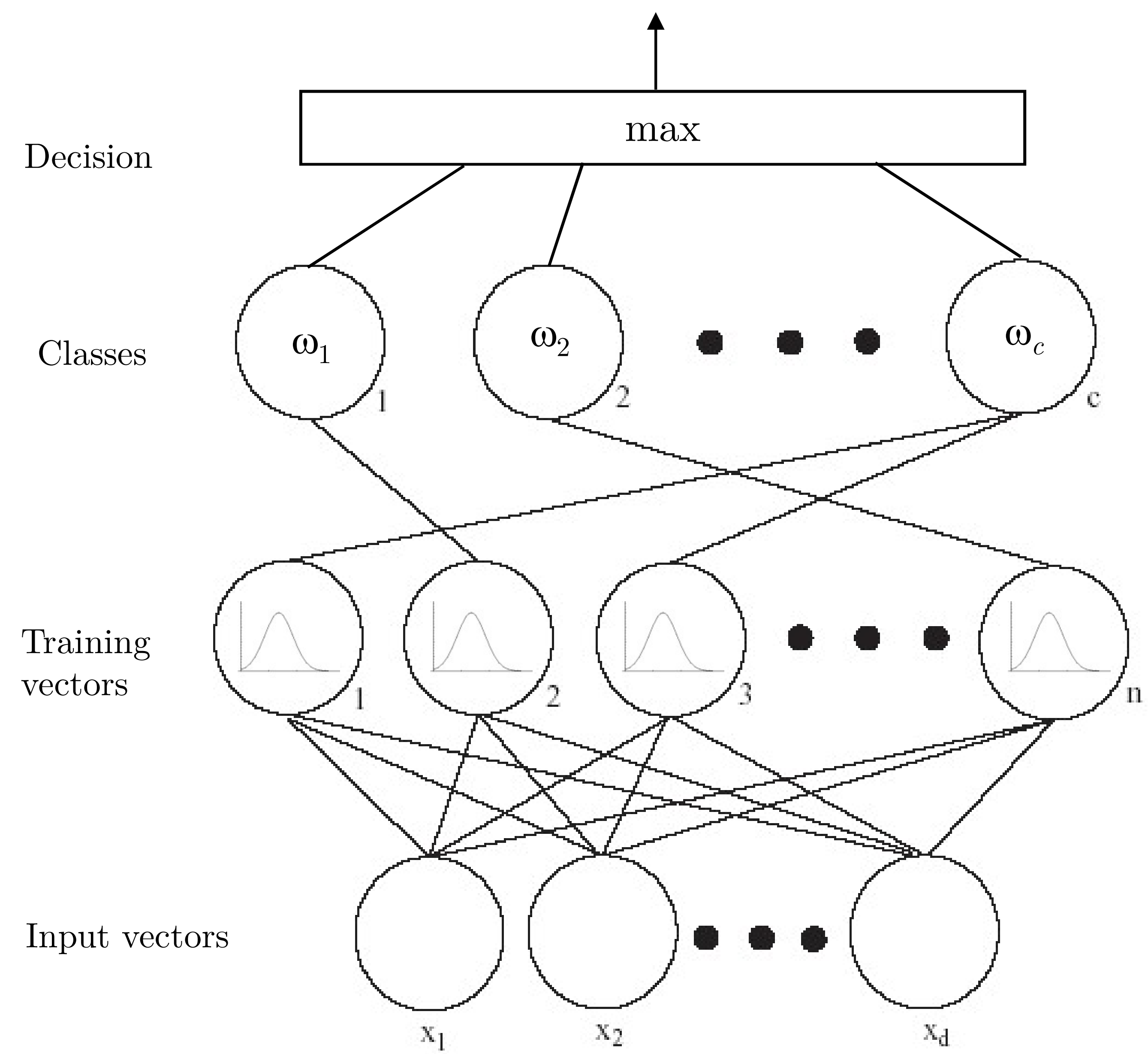



Observation: The optimum size of the window possibly depends on the region under analysis: should be larger where the density of data is small...

# 3 NEURAL PROBABILISTIC CLASSIFIER

A classifier based on the estimation of the pdf using Parzen windows can be built using a neural network-like structure that estimates the pdf and implements a Bayesian rule.

We have $n$ training vectors of dimension $d$: $\mathbf{x}_i$

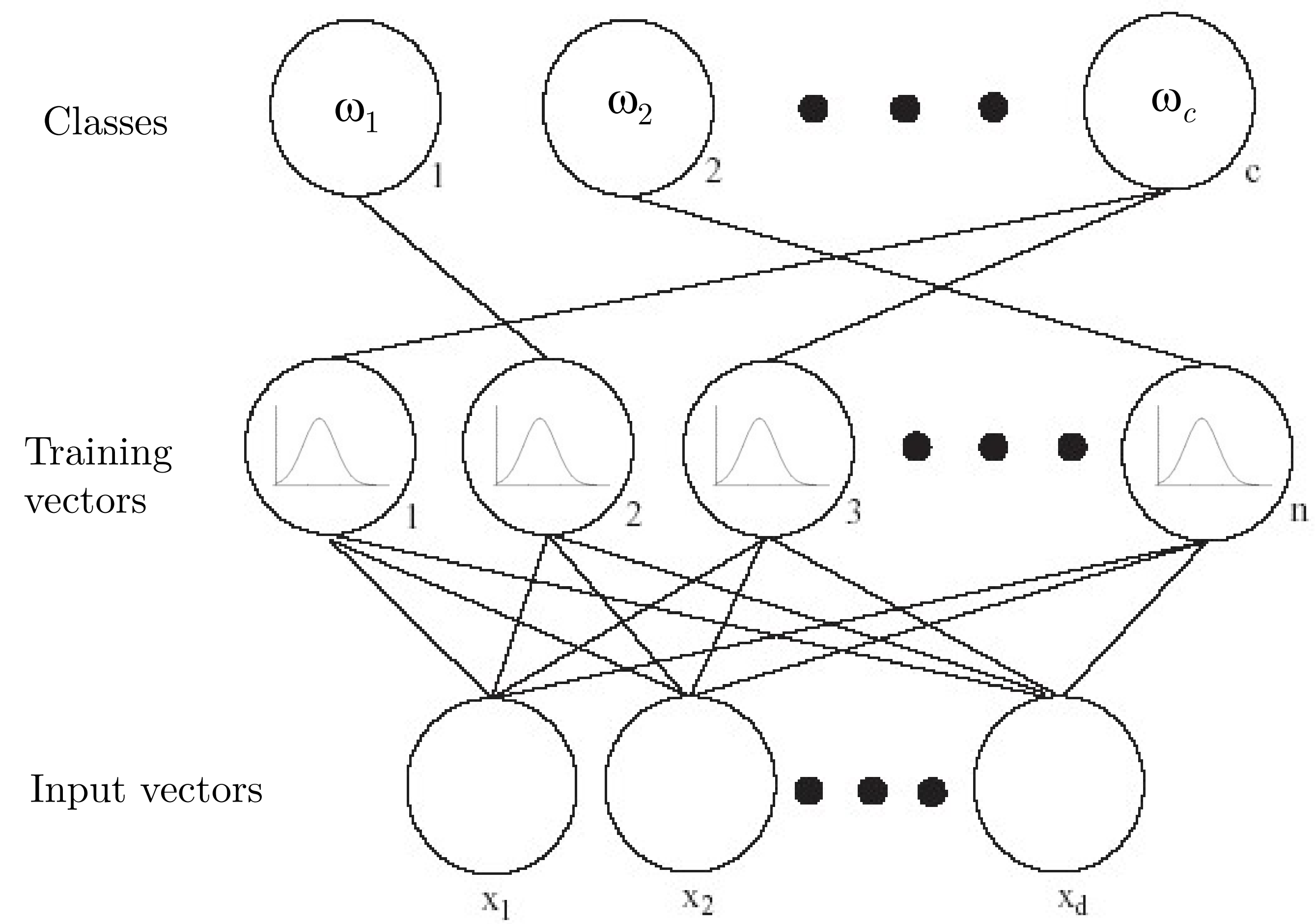


## Training phase

**3**. An edge is set between the block associated to the vector $\mathbf{x}_i$ and its associated class

**2**. The pattern block $i$ weights its inputs with the weights $\mathbf{w}_i = \mathbf{x}_i$

**1**. Training vectors are normalized.



## Decision phase

Links can be weighted with a priori probabilities if known

**4**. The decision on the class is taken by maximizing $g_j(\mathbf{x})$

**3**. Each block sums its inputs and generate a likelihood function $\propto f_n(\mathbf{x}|\omega_j)$

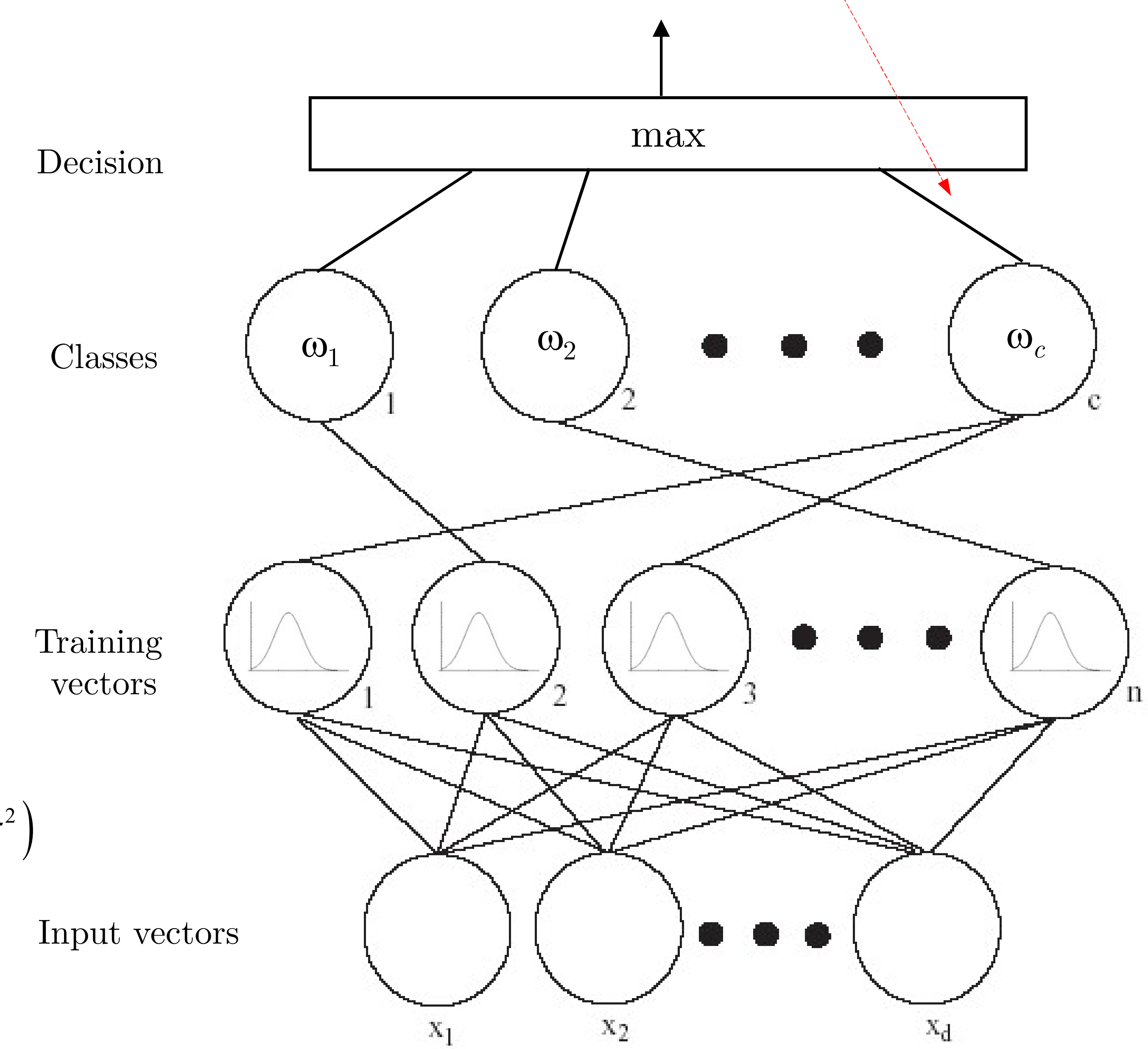
**2**. The pattern blocks compute an activity factor:

$$\varphi\left(\frac{\mathbf{x}-\mathbf{w}_i}{h_n}\right) \propto \exp\left(-\left(\mathbf{x}-\mathbf{w}_i\right)^T\left(\mathbf{x}-\mathbf{w}_i\right)/2\sigma^2\right) =$$

$$= \exp\left(-\left(\mathbf{x}^T\mathbf{x}+\mathbf{w}_i^T\mathbf{w}_i-2\mathbf{x}^T\mathbf{w}_i\right)/2\sigma^2\right) =$$

$$= \left\{\mathbf{x}^T\mathbf{x}=\mathbf{w}_i^T\mathbf{w}_i=1\right\} = \exp\left(\left(\mathbf{x}^T\mathbf{w}_i-1\right)/\sigma^2\right)$$

**1**. The vector $\mathbf{x}$ to classify is normalized.



# 4 K-NEAREST NEIGHBORS ESTIMATION

Instead of looking for the best window (in shape and size), the volumen of the cell is increased or decreased as a function of the training data:

To estimate $f(\mathbf{x})$ we enlarge the volumen $V_R(\mathbf{x})$ around $\mathbf{x}$ until $k_n$ vectors are included: the $k_n$ – nearest neighbors.

$$f_n(\mathbf{x}) = \frac{k_n / n}{V_R(\mathbf{x})}$$

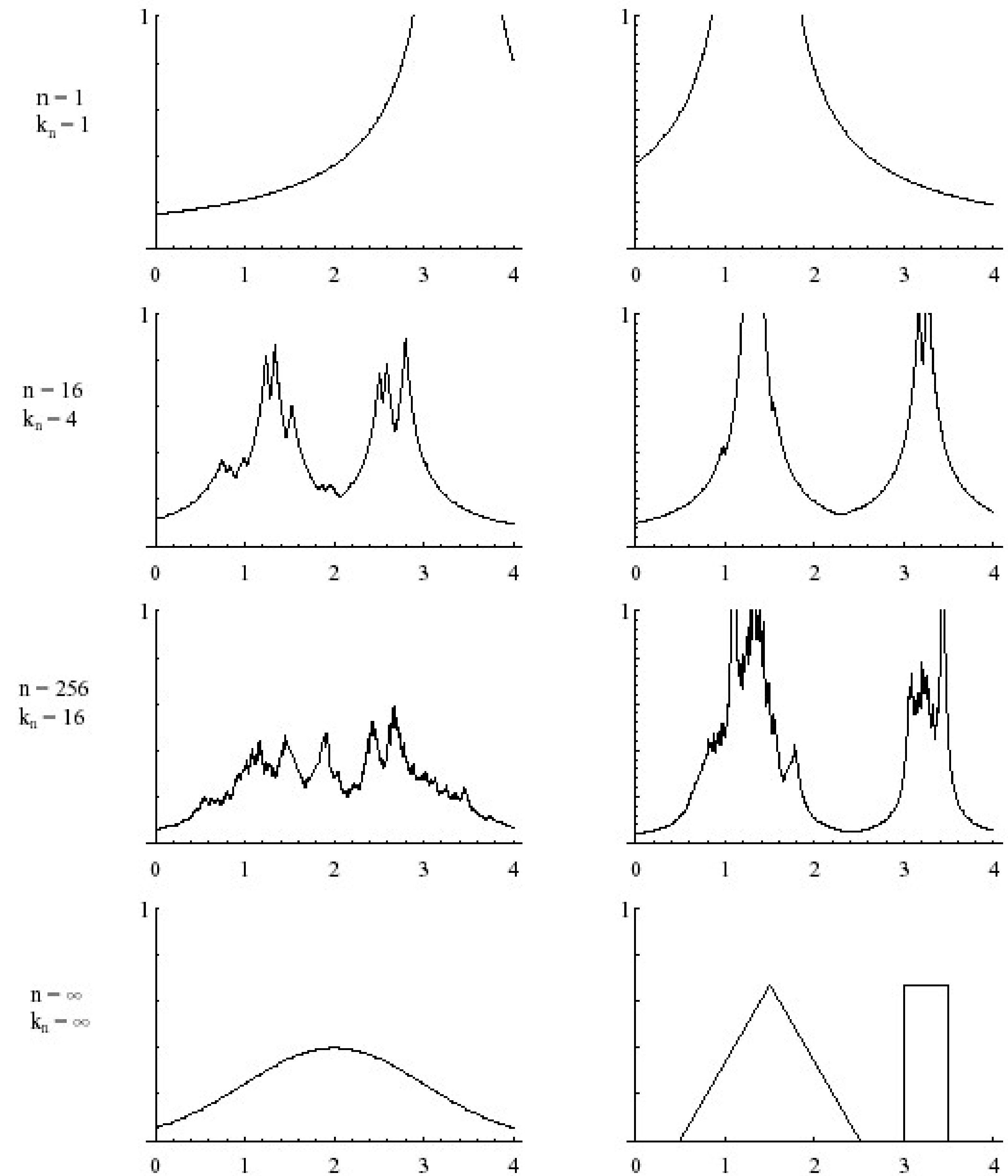
Convergence is guaranteed if $\lim_{n\to\infty} \frac{k_n}{n} = 0$

Example 4:

$$k_n \leq \sqrt{n} \qquad\qquad k_n \leq \ln(n)$$

Example 5:

k-nearest-neighbors estimations. Compare them with those in slide 13



A posteriori probabilities $\Pr(\omega_i|\mathbf{x})$ can be computed as:

$$\Pr\left(\omega_i \mid \mathbf{x}\right) = \frac{f_{\mathbf{x}}(\mathbf{x}\mid\omega_i)\Pr(\omega_i)}{\sum_{i=1}^{c} f_{\mathbf{x}}(\mathbf{x}\mid\omega_i)\Pr(\omega_i)} = \frac{f_{\mathbf{x}}(\mathbf{x}\mid\omega_i)\Pr(\omega_i)}{\sum_{i=1}^{c} f_{\mathbf{x}}(\mathbf{x},\omega_i)} = \frac{k_i / nV_R(\mathbf{x})}{k / nV_R(\mathbf{x})} = \frac{k_i}{k}$$

$$f_{\mathbf{x}}(\mathbf{x}\mid\omega_i)\Pr(\omega_i) \simeq \frac{k_i / n_i}{V_R(\mathbf{x})}\frac{n_i}{n} = \frac{k_i}{nV_R(\mathbf{x})}$$

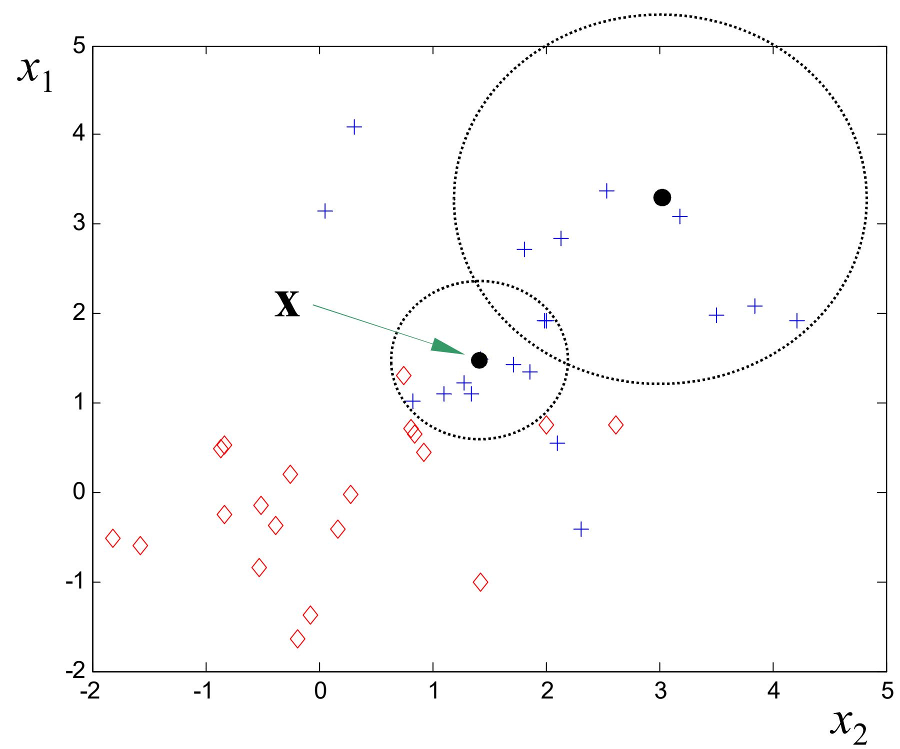
A fraction of data belonging to class $\omega_i$ among the $k$ neighbours of $\mathbf{x}$. It is the discriminant $g_i(\mathbf{x})$

# 5 K-NEAREST NEIGHBORS RULE

Classification rule. Using the previous equation we can classify a vector **x** with the following rule (which is optimum if $n$ is large):

Select the class most represented in a region around **x**.

Example 6:




$k = 8$

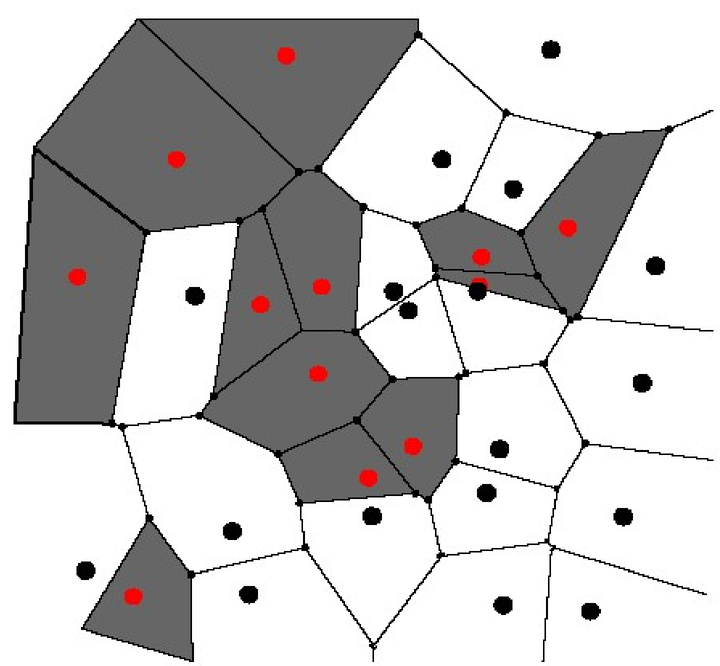
Selected class for vector **x** is '+'

The size of the region is not constant in the whole domain

- Parzen: Select the class with more weighted vectors in the window centered in $\mathbf{x}_i$

$$\varphi\left(\frac{\mathbf{x}-\mathbf{x}_i}{h_n}\right)$$

- K-nearest-neighbors: Find the window containing $k$ neighbours around $\mathbf{x}$. The most represented class is the chosen one.



**Decision regions for 1-nearest**

We can obtain reasonable performance if the class is chosen from the one associated to the nearest training vector $\mathbf{x}$ ("1-nearest").

# 6 DISTANCE

Properties:

| | |
|---|---|
| Non-negativity | $D(\mathbf{x},\mathbf{y}) \geq 0$ |
| Reflexivity | $D(\mathbf{x},\mathbf{y}) = 0 \quad \text{iff} \;\; \mathbf{x} = \mathbf{y}$ |
| Symmetry | $D(\mathbf{x},\mathbf{y}) = D(\mathbf{y},\mathbf{x})$ |
| Triangular inequality | $D(\mathbf{x},\mathbf{y}) + D(\mathbf{y},\mathbf{t}) \geq D(\mathbf{x},\mathbf{t})$ |

The chosen distance depence on each problem.

Example 7:

$$D(\mathbf{x},\mathbf{y}) = \left( \sum_{i=1}^{d} \left| x_i - y_i \right|^p \right)^{1/p}$$

$p = 1$

$p = 2$

$p = \infty$

## Example 7: Crime prediction in San Francisco

- Problem definition

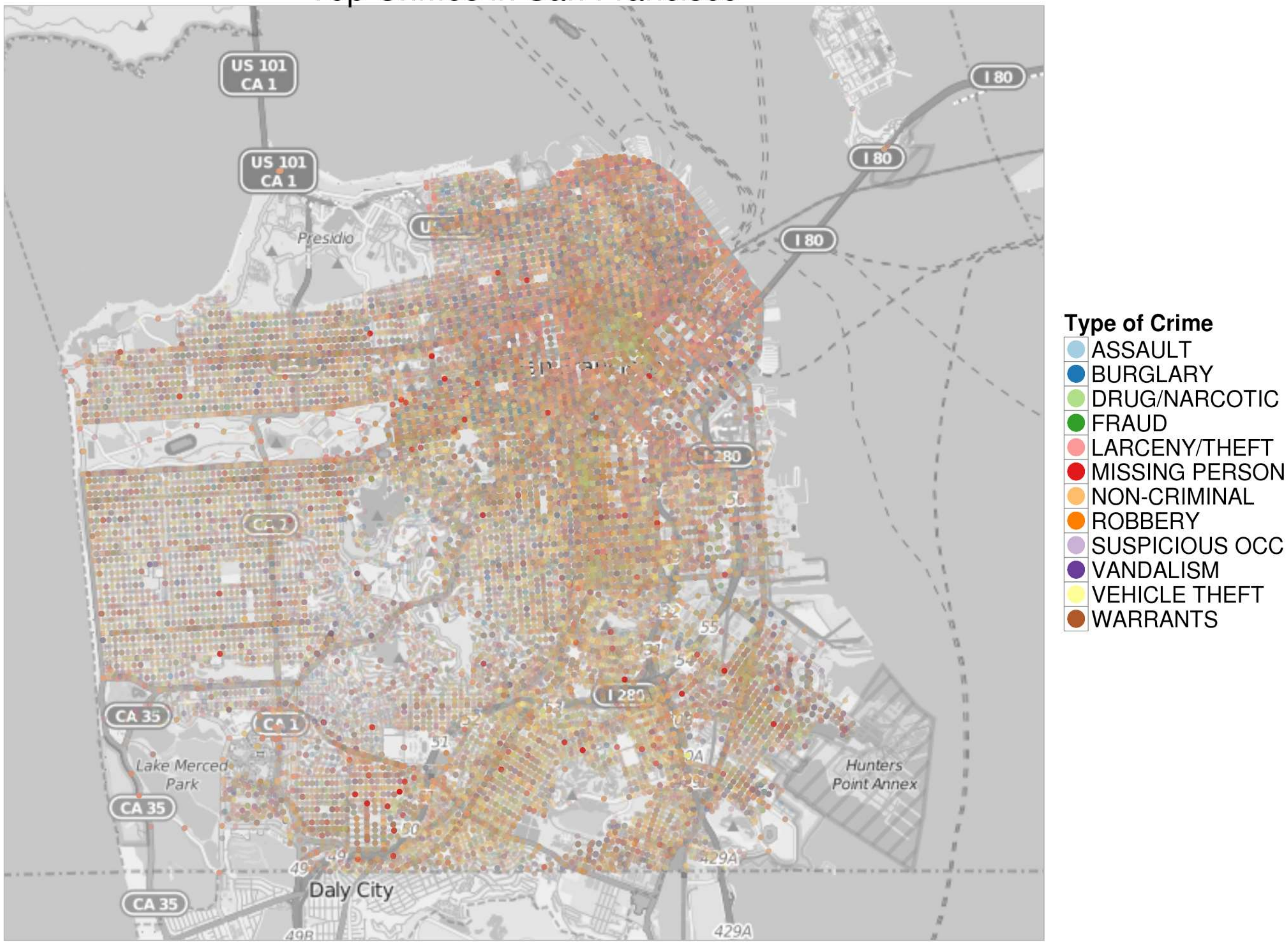
Predict crimes in San Francisco among 38 types from a labeled data base.

- Data base

800.000 crimes recorded between 2003 and 2015. Each crime is characterized by:

- **Dates** - timestamp of the crime incident
- **Category** - category of the crime incident (only in train.csv). **This is the target variable you are going to predict.**
- **Descript** - detailed description of the crime incident (only in train.csv)
- **DayOfWeek** - the day of the week
- **PdDistrict** - name of the Police Department District
- **Resolution** - how the crime incident was resolved (only in train.csv)
- **Address** - the approximate street address of the crime incident
- **X** - Longitude
- **Y** - Latitude

Top Crimes in San Francisco



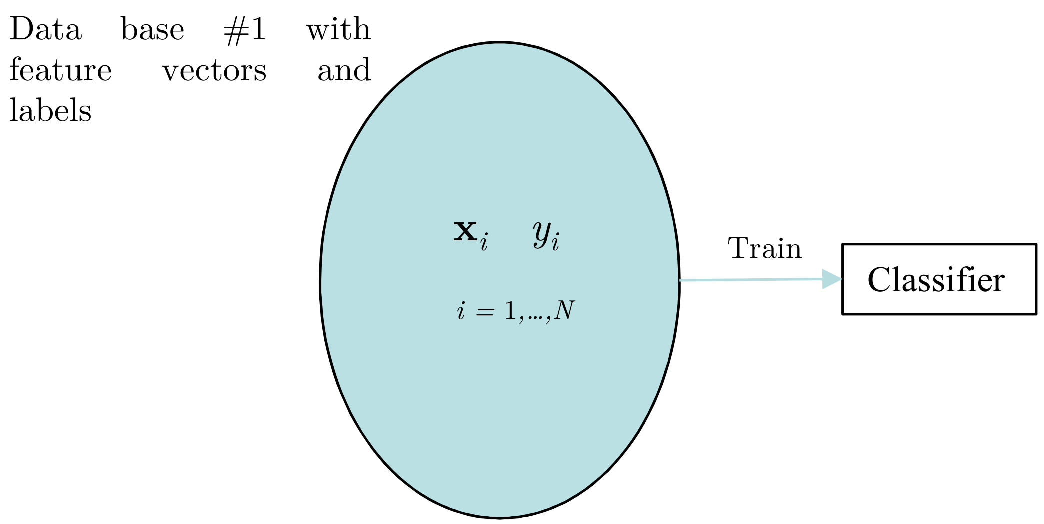
| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 2003-01-07 07:52:00 | WARRANTS | WARRANT ARREST | Tuesday | SOUTHERN | ARREST, BOOKED | 5TH ST / SHIPLEY ST | -122.402843 | 37.779829 |
| 2003-01-07 04:49:00 | WARRANTS | ENROUTE TO OUTSIDE JURISDICTION | Tuesday | TENDERLOIN | ARREST, BOOKED | CYRIL MAGNIN STORTH ST / EDDY ST | -122.408495 | 37.784452 |
| 2003-01-07 03:52:00 | WARRANTS | WARRANT ARREST | Tuesday | NORTHERN | ARREST, BOOKED | OFARRELL ST / LARKIN ST | -122.417904 | 37.785167 |
| 2003-01-07 03:34:00 | WARRANTS | WARRANT ARREST | Tuesday | NORTHERN | ARREST, BOOKED | DIVISADERO ST / LOMBARD ST | -122.442650 | 37.798999 |
| 2003-01-07 01:22:00 | WARRANTS | WARRANT ARREST | Tuesday | SOUTHERN | ARREST, BOOKED | 900 Block of MARKET ST | -122.409537 | 37.782691 |
| 2003-01-06 23:30:00 | WARRANTS | ENROUTE TO OUTSIDE JURISDICTION | Monday | BAYVIEW | ARREST, BOOKED | REVERE AV / INGALLS ST | -122.384557 | 37.728487 |
| 2003-01-06 23:14:00 | WARRANTS | WARRANT ARREST | Monday | CENTRAL | ARREST, BOOKED | BUSH ST / HYDE ST | -122.417019 | 37.789110 |
| 2003-01-06 22:45:00 | WARRANTS | WARRANT ARREST | Monday | SOUTHERN | ARREST, BOOKED | 800 Block of BRYANT ST | -122.403405 | 37.775421 |
| 2003-01-06 22:45:00 | WARRANTS | ENROUTE TO OUTSIDE JURISDICTION | Monday | SOUTHERN | ARREST, BOOKED | 800 Block of BRYANT ST | -122.403405 | 37.775421 |
| 2003-01-06 22:19:00 | WARRANTS | ENROUTE TO OUTSIDE JURISDICTION | Monday | NORTHERN | ARREST, BOOKED | GEARY ST / POLK ST | -122.419740 | 37.785893 |
| 2003-01-06 21:54:00 | WARRANTS | ENROUTE TO OUTSIDE JURISDICTION | Monday | NORTHERN | ARREST, BOOKED | SUTTER ST / POLK ST | -122.420120 | 37.787757 |

- Evaluation: multiclass logarithmic loss (better when lower)

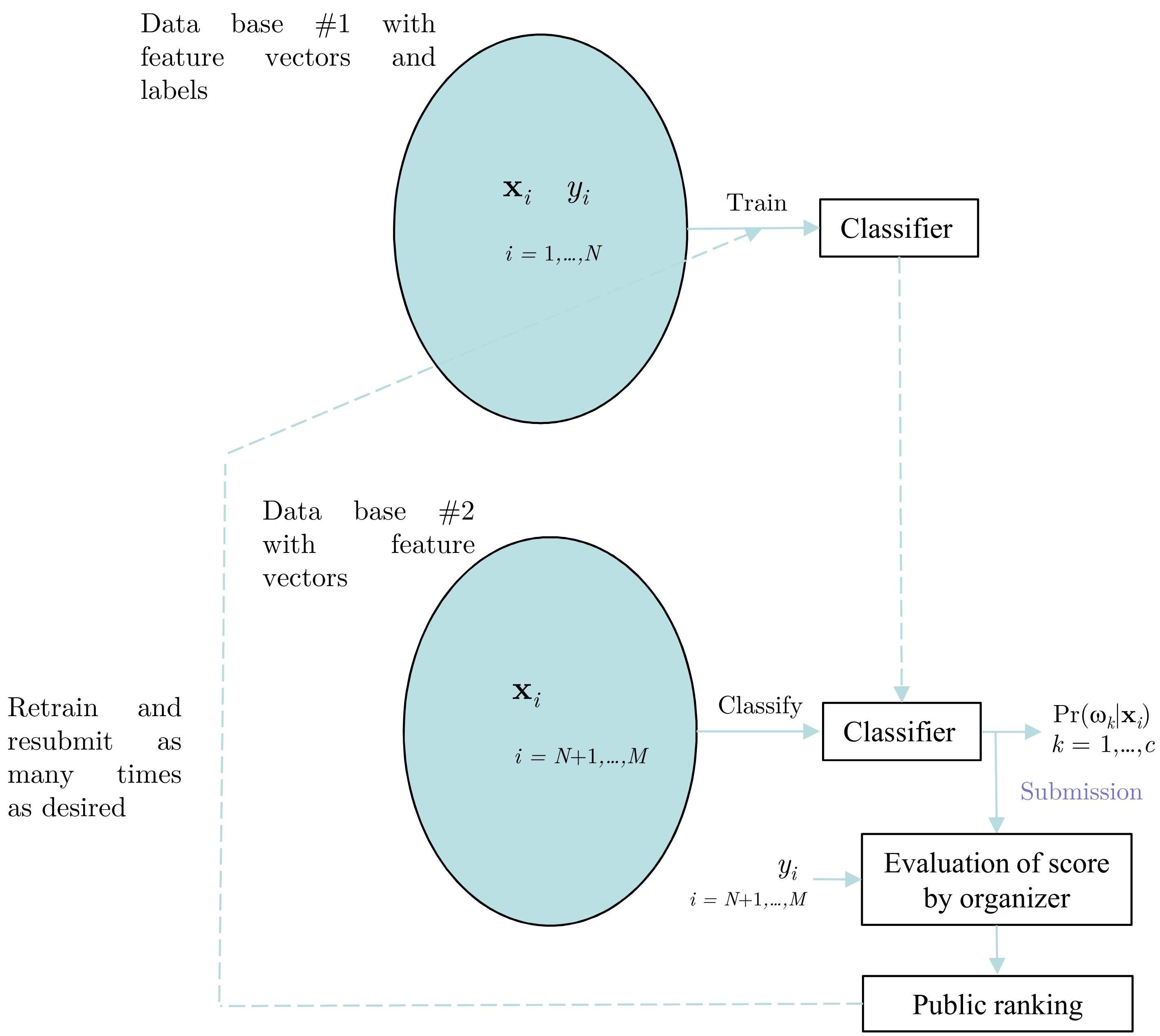
$$\text{logloss} = -\frac{1}{N}\sum_{i=1}^{N_{test}}\sum_{j=1}^{c} y_{ij}\log \Pr\left(\omega_j \middle| \mathbf{x}_i\right) \qquad y_{ij} \in \{0,1\}$$

Validation using a fraction of the data base.

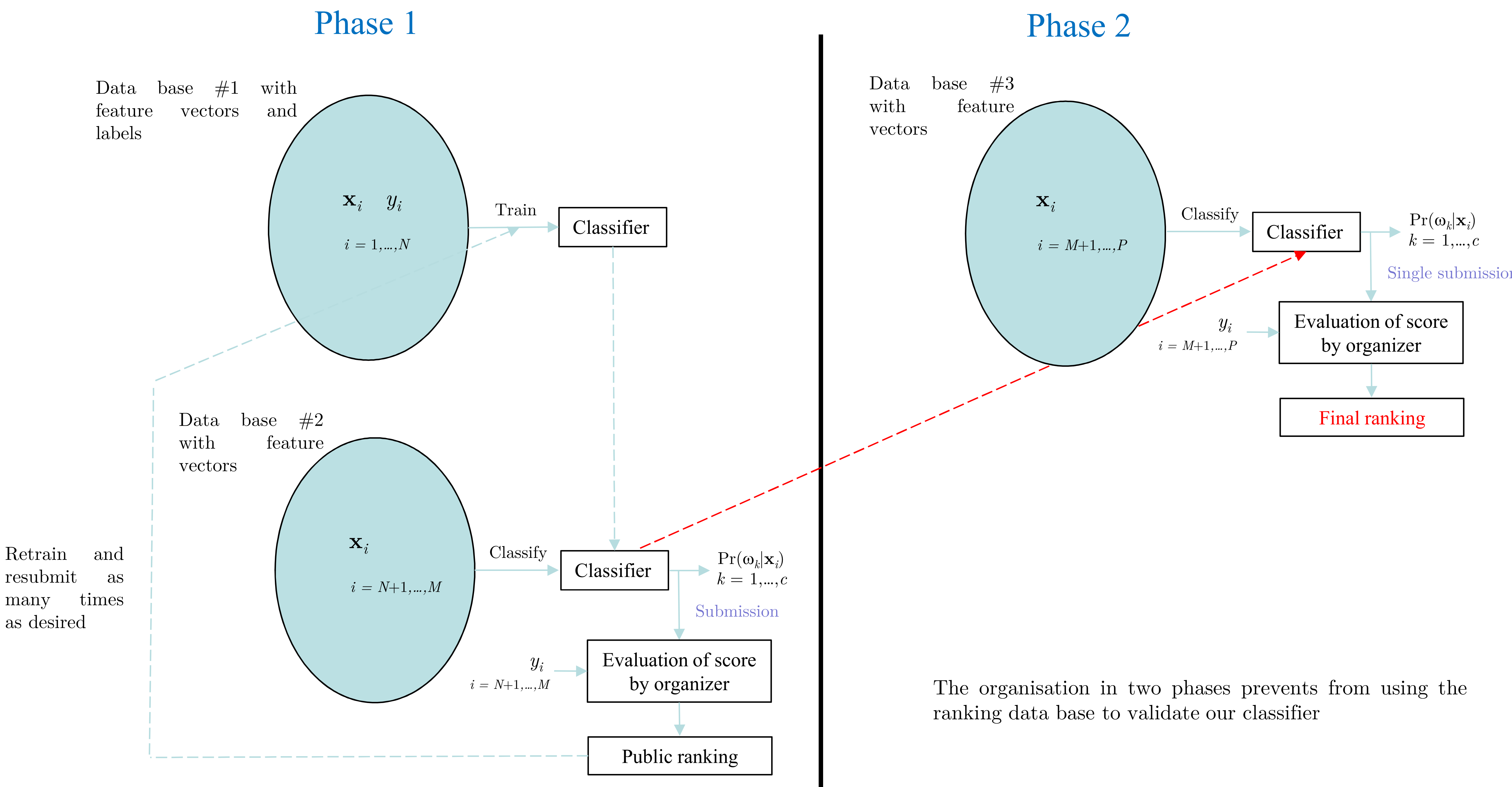
# Organization of a machine learning competition



# Organization of a machine learning competition

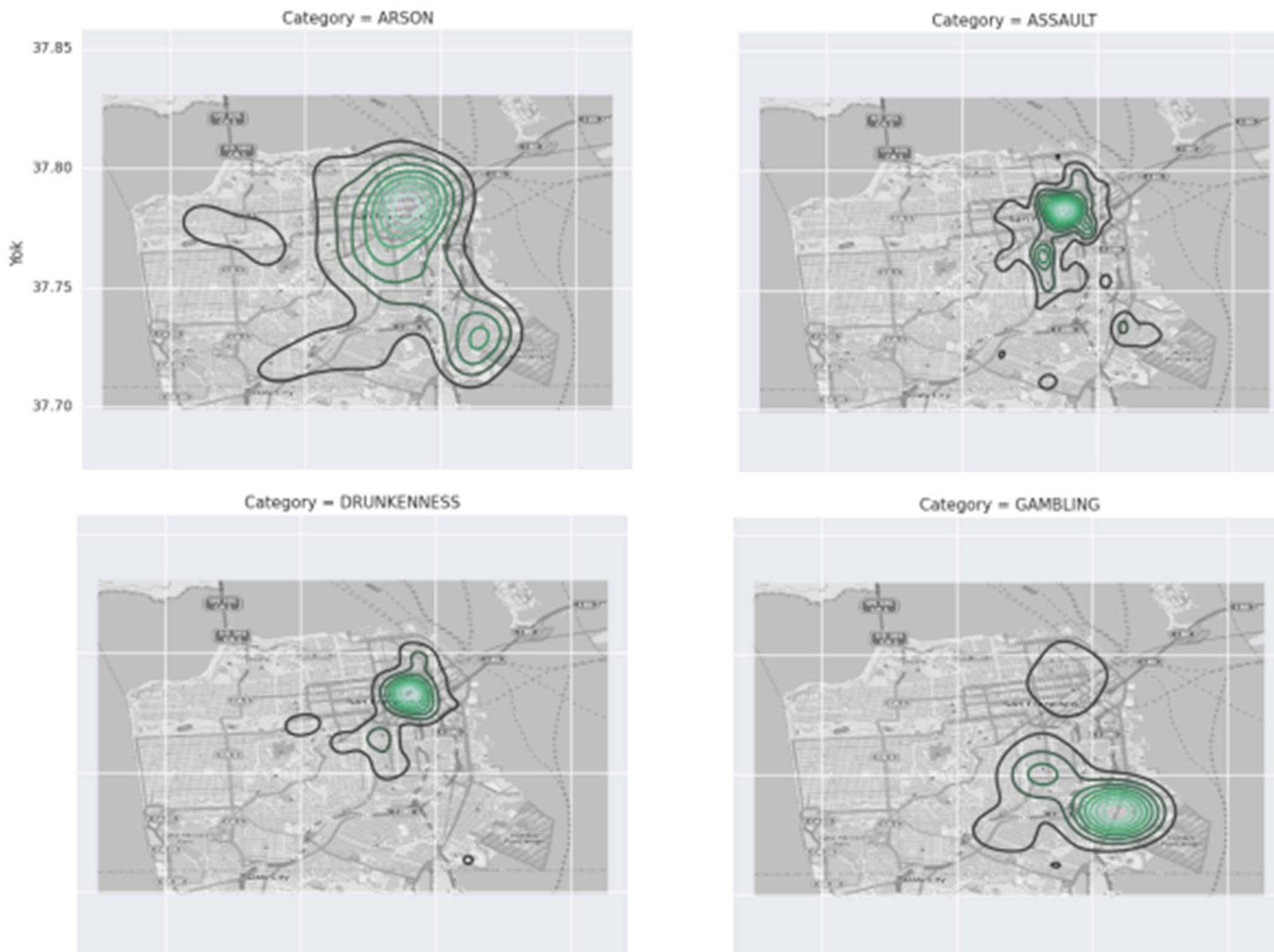


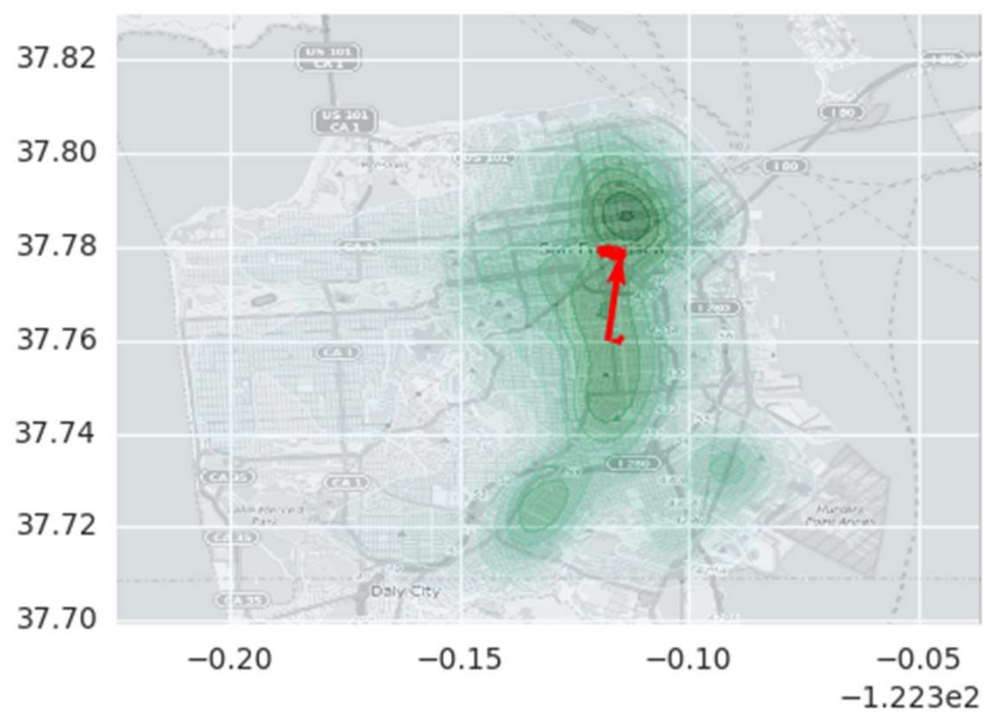
# Organization of a machine learning competition

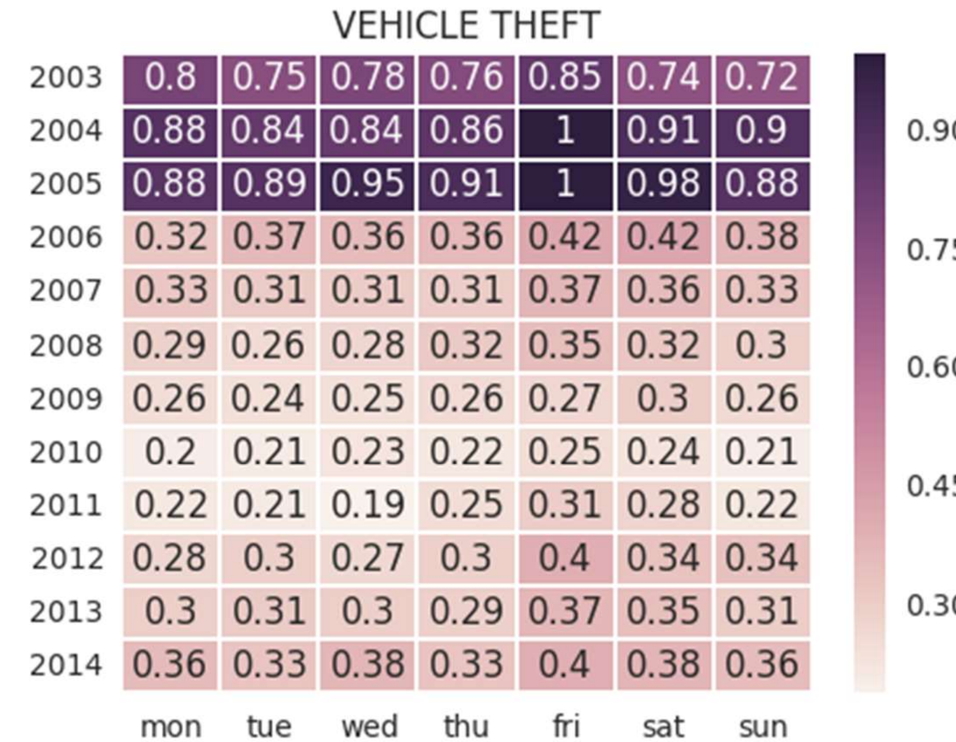


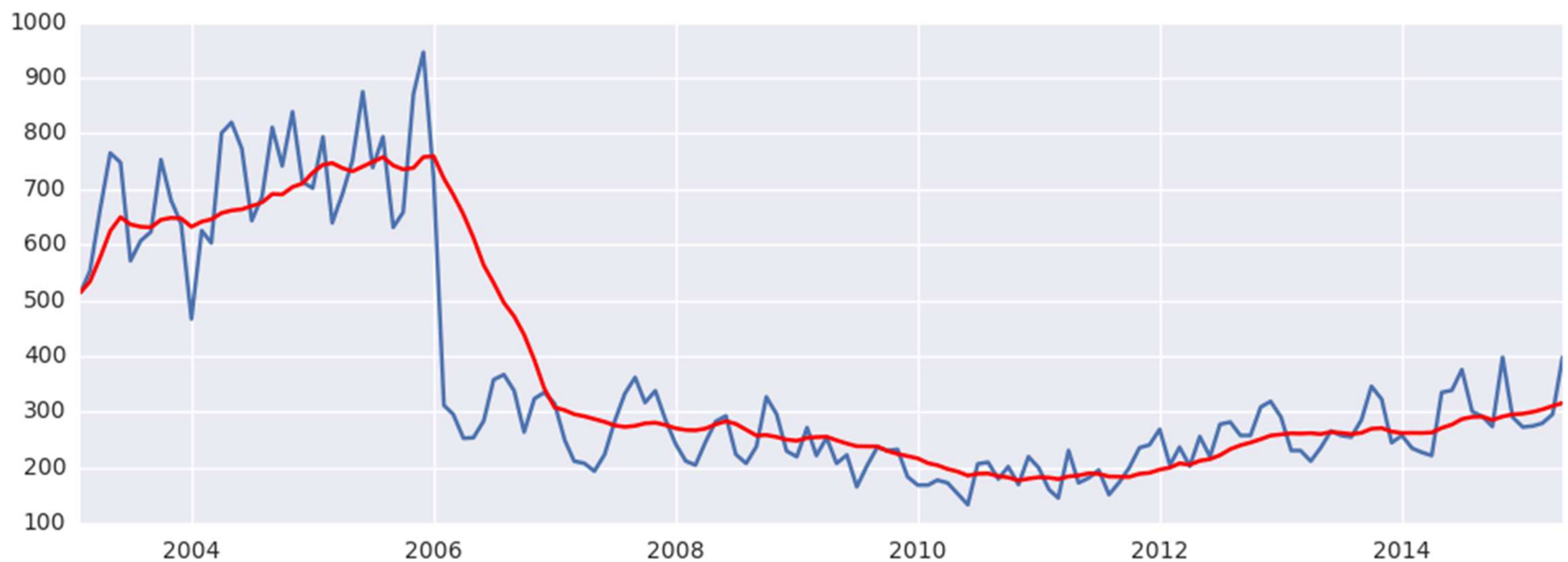

The organisation in two phases prevents from using the ranking data base to validate our classifier

- Estimation of $f(\mathbf{x}|\omega_j)$ for 4 classes…

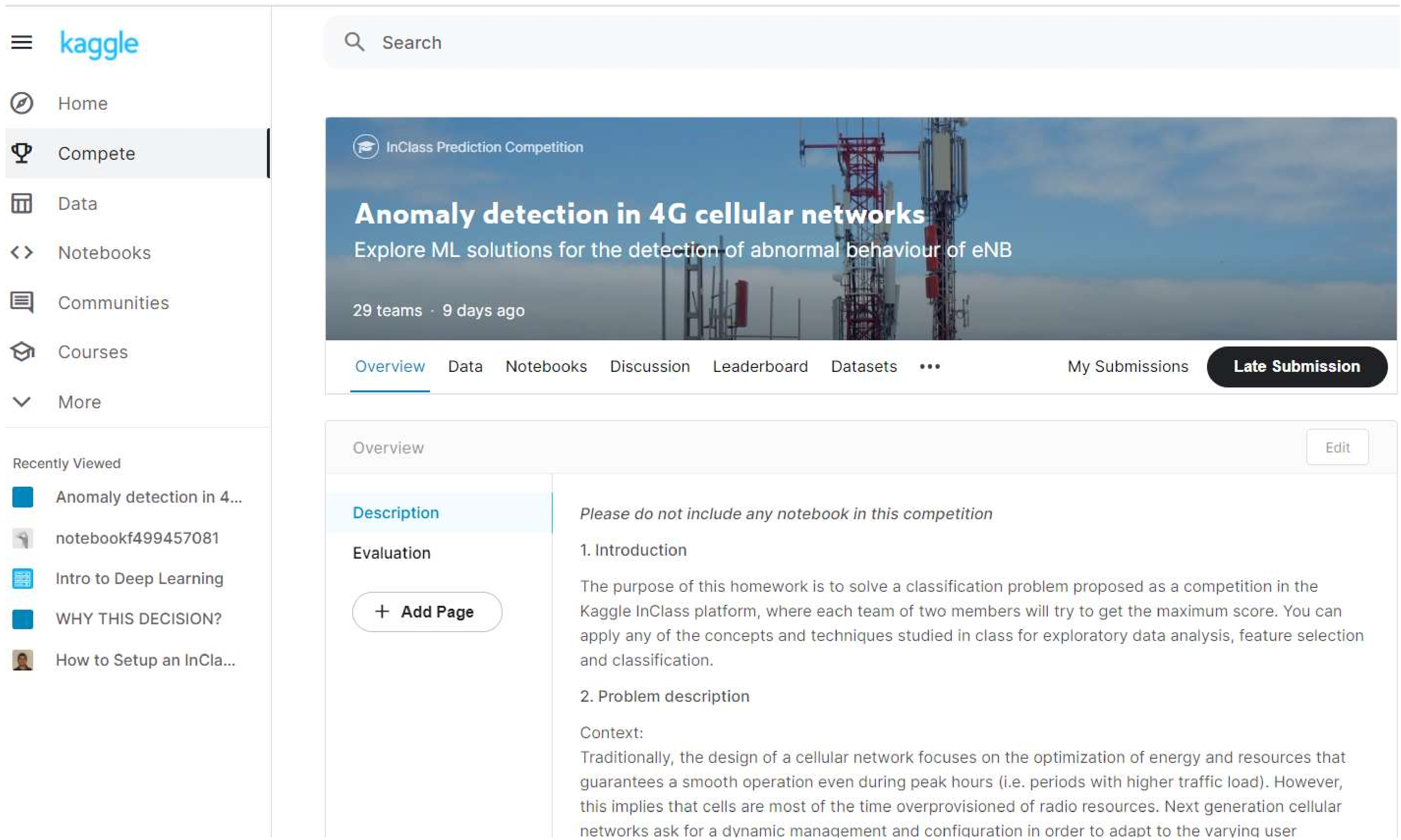



37.82
37.80
37.78
37.76
37.74
37.72
37.70
−0.20
−0.15
−0.10
−0.05
−1.223e2


VEHICLE THEFT
2003 0.8 0.75 0.78 0.76 0.85 0.74 0.72
2004 0.88 0.84 0.84 0.86 1 0.91 0.9
2005 0.88 0.89 0.95 0.91 1 0.98 0.88
2006 0.32 0.37 0.36 0.36 0.42 0.42 0.38
2007 0.33 0.31 0.31 0.31 0.37 0.36 0.33
2008 0.29 0.26 0.28 0.32 0.35 0.32 0.3
2009 0.26 0.24 0.25 0.26 0.27 0.3 0.26
2010 0.2 0.21 0.23 0.22 0.25 0.24 0.21
2011 0.22 0.21 0.19 0.25 0.31 0.28 0.22
2012 0.28 0.3 0.27 0.3 0.4 0.34 0.34
2013 0.3 0.31 0.3 0.29 0.37 0.35 0.31
2014 0.36 0.33 0.38 0.33 0.4 0.38 0.36
mon tue wed thu fri sat sun
0.90
0.75
0.60
0.45
0.30


1000
900
800
700
600
500
400
300
200
100
2004
2006
2008
2010
2012
2014

# One recent MLEARN proposed competition



# 7 CONCLUSIONS

Two non-parametric classifiers:

1. Parzen
2. k-nearest neighbors

The use of 1-nearest neighbor yields a **Pr(*error*)** (for large data base) equal to twice the Bayesian classifier, with a low complexity.

These are the only suitable methods when dimensionality of vectors is large compared to the number of training vectors.