# *Prognostic models designed with the linear separability principle*

**Leon Bobrowski**[1,2]
[1]Faculty of Computer Science, Białystok University of Technology
[2]Institute of Biocybernetics and Biomedical Engineering,
Polish Academy of Science, Poland
*e-mail*: l.bobrowski@pb.edu.pl

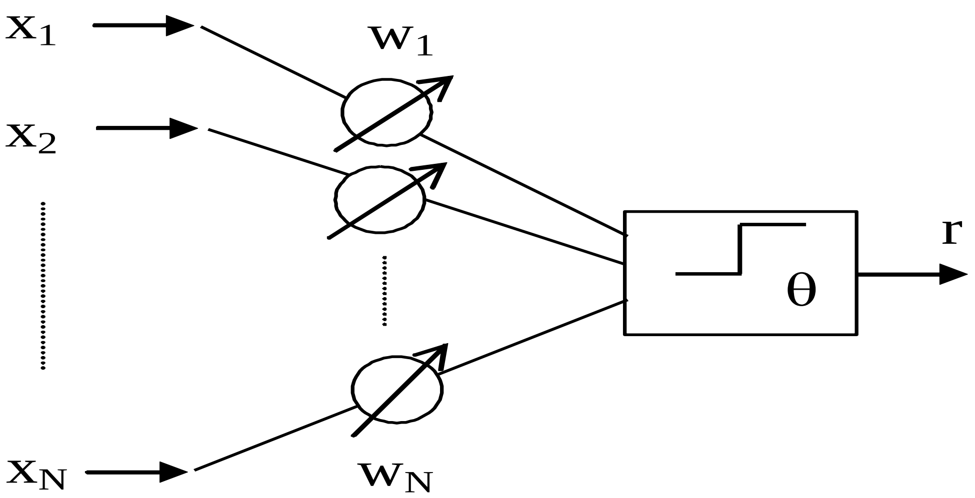
Seminar of the Department of Structural Methods of Knowledge Processing, Faculty of Mathematics and Information Science, Warsaw University of Technology

**March 3, 2016, Warsaw**

# Content

# I. Introduction

# Learning algorithms of the formal neuron $NF(\mathbf{w},\theta)$




$\mathbf{x}(n) = [x_1,...,x_N]T$ - *input vector* during the $n$-th learning step ($n = 1,2,3,4,..$)

$s(n)$ ($s(n) = 1$ or $s[n] = 0$) – *teacher's decision* during the $n$-th learning step

($\mathbf{x}(1)$, s(1)), ($\mathbf{x}(2)$, s(2)), ($\mathbf{x}(3)$, s(3)),… ***learning sequence*** $\{(\mathbf{x}(n), s(n)\}$

$\mathbf{x}(1)$, $\mathbf{x}(2)$, ($\mathbf{x}(3)$, ..... ***self - learning sequence*** $\{\mathbf{x}(n)\}$

# Error correction algorithm (*Perceptron*)

$\mathbf{w}(n) = [w_1,.......,w_N]^T$ – the *weight vector* of the formal neuron $NF(\mathbf{w},\theta)$ during the $n$-th learning step ($\mathbf{w}(n) \in R^N$)

$\theta(n)$ – the *threshold* of the formal neuron $NF(\mathbf{w},\theta)$ during the $n$-th step ($\theta(n) \in R1$)

$r(n)$ – *output* of the formal neuron $NF(\mathbf{w},\theta)$ during the $n$-th step ($r(n) = 1$ or $r(n) = 0$)

$$r(n) = r(\mathbf{w}(n),\theta(n);\mathbf{x}(n)) = \begin{cases} 1 & \text{if } \mathbf{w}(n)^T\mathbf{x}(n) \geq \theta(n) \\ 0 & \text{if } \mathbf{w}(n)^T\mathbf{x}(n) < \theta(n) \end{cases}$$

If $r(n) \neq s(n)$ (***error***), then the correction of the weight vector $\mathbf{w}(n)$ and the threshold $\theta(n)$ follows.

# *Error correction algorithm* (*Perceptron*)

$$\mathbf{w}(n+1) = \begin{cases} \mathbf{w}(n) + \mathbf{x}(n) & \textit{if} \ \ \mathrm{r}(n) = 0 \ \textit{and} \ \mathrm{s}(n) = 1 \\ \mathbf{w}(n) & \textit{if} \ \ \mathrm{r}(n) = \mathrm{s}(n) \\ \mathbf{w}(n) - \mathbf{x}(n) & \textit{if} \ \ \mathrm{r}(n) = 1 \ \textit{and} \ \mathrm{s}(n) = 0 \end{cases}$$

$$\theta(n+1) = \begin{cases} \theta(n) - 1 & \textit{if} \ \ \mathrm{r}(n) = 0 \ \textit{and} \ \mathrm{s}(n) = 1 \\ \theta(n) & \textit{if} \ \ \mathrm{r}(n) = \mathrm{s}(n) \\ \theta(n) + 1 & \textit{if} \ \ \mathrm{r}(n) = 1 \ \textit{and} \ \mathrm{s}(n) = 0 \end{cases}$$

*or*

$$\mathbf{w}(n+1) = \begin{cases} \mathbf{w}(n) + \mathbf{x}(n) & \textit{if} \ \ \mathbf{w}(n)^{\mathrm{T}}\mathbf{x}(n) < \theta(n) \ \textit{and} \ \mathrm{s}(n) = 1 \\ \mathbf{w}(n) & \textit{if} \ \ \mathrm{r}(n) = \mathrm{s}(n) \\ \mathbf{w}(n) - \mathbf{x}(n) & \textit{if} \ \ \mathbf{w}(n)^{\mathrm{T}}\mathbf{x}(n) \geq \theta(n) \ \textit{and} \ \mathrm{s}(n) = 0 \end{cases}$$

$$\theta(n+1) = \begin{cases} \theta(n) - 1 & \textit{if} \ \ \mathbf{w}(n)^{\mathrm{T}}\mathbf{x}(n) < \theta(n) \ \textit{and} \ \mathrm{s}(n) = 1 \\ \theta(n) & \textit{if} \ \ \mathrm{r}(n) = \mathrm{s}(n) \\ \theta(n) + 1 & \textit{if} \ \ \mathbf{w}(n)^{\mathrm{T}}\mathbf{x}(n) \geq \theta(n) \ \textit{and} \ \mathrm{s}(n) = 0 \end{cases}$$

# ***Feature vectors*** $\mathbf{x}_j[n]$

(Terminology of *pattern recognition*)

$$\mathbf{x}_j[n] = [x_{j1},\ldots\ldots, x_{jn}]^T$$

where $x_{ji} \in R^1$, or $x_{i_{ji}} \in \{0,1\}$, $j = 1,\ldots, m$, $i = 1,\ldots,n$.
The $n$-dimensional *feature vector* $\mathbf{x}_j[n]$ ($\mathbf{x}_j[n] \in F[n]$) represents the $j$-th object (patient) $O_j$ from a given database in the feature space $\boldsymbol{F}[n]$.
The component $x_{ji}$ of the vector $\mathbf{x}_j[n]$ is the numerical value of the $i$-th *feature* (measurement, diagnostic test) of the object $O_j$.

# ***Learning sets*** $G^+$ ***and*** $G^-$

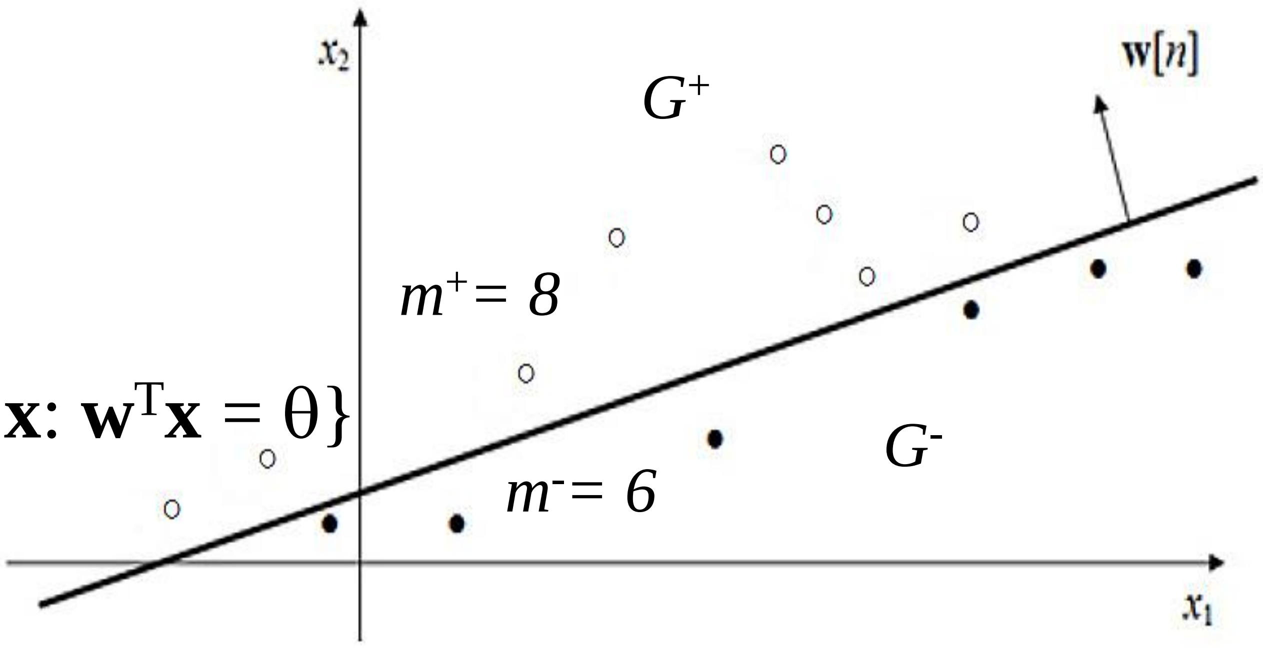
The learning set $G^+$ contains $m^+$ **positive precedents** (examples) $\mathbf{x}_j[n]$
The learning set $G^-$ contains $m^-$ **negative precedents** (examples) $\mathbf{x}_j[n]$

If the number $\boldsymbol{n}$ of features $x_i$ is greater than the number $\boldsymbol{m} = m^+ + m^-$ of feature vectors $\mathbf{x}_j[n]$, then each $\mathbf{x}_j[n]$ can be called *"a **long vector**"*. For example, genetic data sets are usually built from ***long vectors*** $\mathbf{x}_j[n]$.

# II. Linear separability of the learning sets

# Linearly separable learning sets $G^+$ and $G^-$

The concept of ***linear separabilty*** of multidimensional data sets is linked to the origins of methods of *neural networks* and *pattern recognition* (*Perceptron* theory)




$$H(\mathbf{w},\theta) = \{\mathbf{x}\colon \mathbf{w}^T\mathbf{x} = \theta\}$$

**If the learning sets $G^+$ and $G^-$ are linearly separable, then the error correction algorithm converges in a finite number of steps.**

# *Linearly separable* learning sets $G^+$ and $G^-$

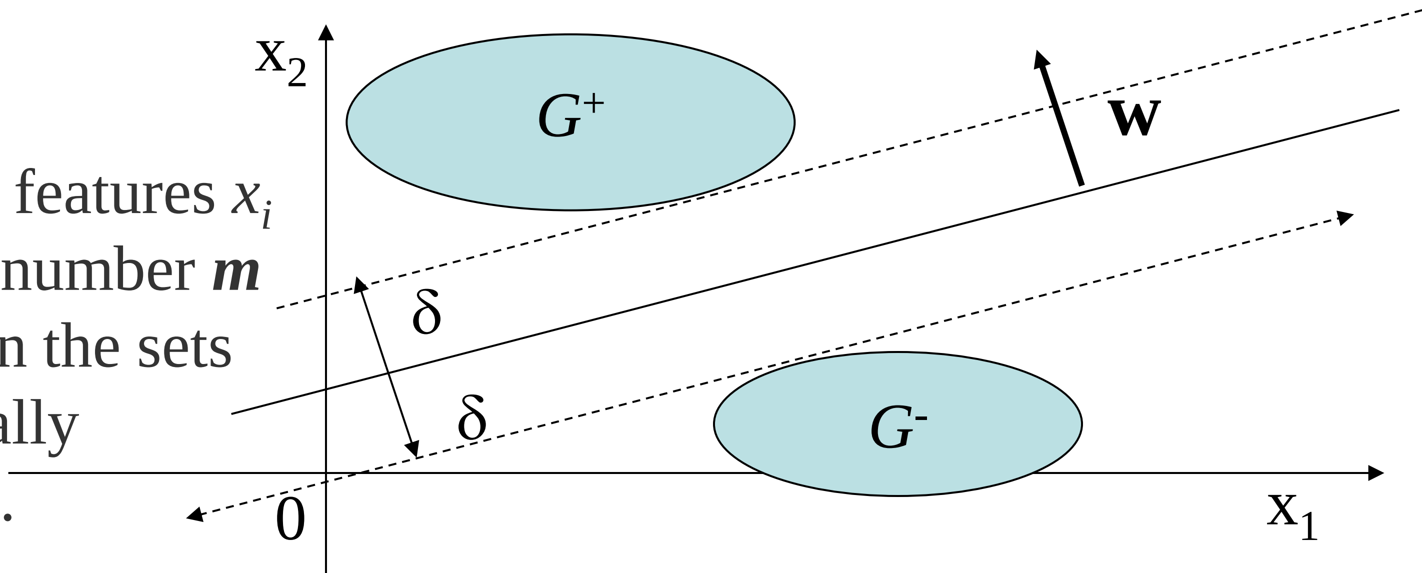
Data set $G^+$ can be exactly separated from the set $G^-$ by some ***hyperplane*** $H(\mathbf{w},\theta) = \{\mathbf{x}: \mathbf{w}^T\mathbf{x} = \theta\}$:

$$(\exists \mathbf{w}, \theta)\ \ (\forall \mathbf{x}_j \in G^+)\quad \mathbf{w}^T\mathbf{x}_j - \theta > 0$$
$$\textbf{\textit{and}}\ \ (\forall \mathbf{x}_j \in G^-)\quad \mathbf{w}^T\mathbf{x}_j - \theta < 0, \quad \textit{or}$$

$$(\exists \mathbf{w}, \theta)\ \ (\forall \mathbf{x}_j \in G^+)\quad \mathbf{w}^T\mathbf{x}_j - \theta \geq 1$$
$$\textbf{\textit{and}}\ \ (\forall \mathbf{x}_j \in G^-)\quad \mathbf{w}^T\mathbf{x}_j - \theta \leq -1$$

$\mathbf{w}[n] = [x_1,..., x_n]^T$ is the ***weight vector***, $\theta$ is the ***threshold*** ($\theta \in R^1$)
$\delta = 1/\ \|\mathbf{w}\|$ is the ***positive marigin***

***REMARK***:
If the number ***n*** of features $x_i$ is greater than the number ***m*** of elements $\mathbf{x}_j$, then the sets $G^+$ and $G^-$ are usually ***linearly separable***.



# *Linearly separable* data sets $G^+$ and $G^-$

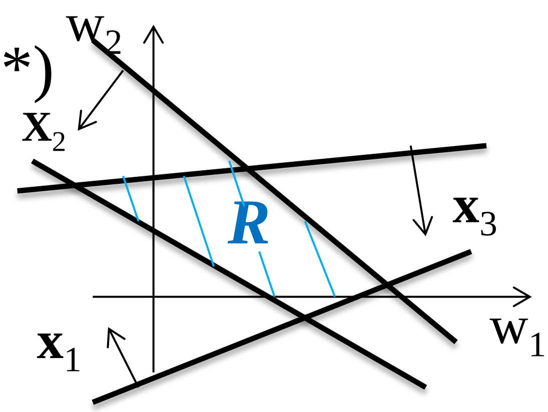
Data set $G^+$ can be exactly separated from the set $G^-$ by the ***hyperplane*** $H(\mathbf{w},\theta) = \{\mathbf{x}: \mathbf{w}^T\mathbf{x} = \theta\}$ if the parameters $\mathbf{w}$ and $\theta$ fulfill the below inequlities:

$$(\forall \mathbf{x}_j \in G^+) \quad \mathbf{w}^T\mathbf{x}_j - \theta \geq 1 \quad (*)$$

$$\textit{and} \quad (\forall \mathbf{x}_j \in G^-) \quad \mathbf{w}^T\mathbf{x}_j - \theta \leq -1$$

$\mathbf{w} = [x_1,\ldots, x_N]^T$ is the ***weight vector***, $\theta$ is the ***threshold*** ($\theta \in R^1$)

***R*** **– the solution region of the linear inequalities (*)**




$$\boldsymbol{R} = \{(\mathbf{w}, \theta): (\forall \mathbf{x}_j \in G^+) \quad (\mathbf{x}_j)^T\mathbf{w} - \theta \geq 1$$
$$\textit{and} \quad (\forall \mathbf{x}_j \in G^-) \quad (\mathbf{x}_j)^T\mathbf{w} - \theta \leq -1\}$$

The solution region ***R*** is nonempty if and only if the learning sets $G^+$ and $G^-$ are linearly separable. The nonempty set ***R*** is a **convex polyhedron** in the parameter space.

# Linear separability of the learning sets $G^+$ and $G^-$

$$(\exists \mathbf{w}, \theta)\ \ (\forall \mathbf{x}_j \in G^+)\quad \mathbf{w}^T\mathbf{x}_j > \theta$$
$$\textbf{\textit{and}}\quad (\forall \mathbf{x}_j \in G^-)\quad \mathbf{w}^T\mathbf{x}_j < \theta$$

•*Theorem*: It the learning sets $G^+$ and $G^-$ are linearly separable, and the matrix $\mathbf{A}$ is nonsingular ($\mathbf{A}^{-1}$ exists), then the sets $R^+ = \{\mathbf{r}_j : \mathbf{r}_j = \mathbf{A}\,\mathbf{x}_j + \mathbf{b}$ and $\mathbf{x}_j \in G^+\}$ and $R^- = \{\mathbf{r}_j : \mathbf{r}_j = \mathbf{A}\,\mathbf{x}_j + \mathbf{b}$ and $\mathbf{x}_j \in G^-\}$ are also linearly separable.

*Proof*:

**I.** $\mathbf{r}_j = \mathbf{A}\,\mathbf{x}_j$, where $\mathbf{A}^{-1}$ exists

*if* $\mathbf{v} = (\mathbf{A}^{-1})^T\mathbf{w}$, *then* $\mathbf{v}^T\mathbf{r}_j = ((\mathbf{A}^{-1})^T\mathbf{w})^T(\mathbf{A}\,\mathbf{x}_j) = \mathbf{w}^T(\mathbf{A}^{-1}\mathbf{A})\mathbf{x}_j = \mathbf{w}^T\mathbf{x}_j$

**II.** $\mathbf{r}_j = \mathbf{A}\,\mathbf{x}_j + \mathbf{b}$, where $\mathbf{A}^{-1}$ exists

*if* $\mathbf{v}' = (\mathbf{A}^{-1})^T\mathbf{w}$, *then* $(\mathbf{v}')^T\mathbf{r}_j = ((\mathbf{A}^{-1})^T\mathbf{w})^T(\mathbf{A}\,\mathbf{x}_j + \mathbf{b}) =$

$= \mathbf{w}^T(\mathbf{A}^{-1}\mathbf{A})\mathbf{x}_j + \mathbf{w}^T(\mathbf{A}^{-1}\mathbf{A})\mathbf{b} = \mathbf{w}^T\mathbf{x}_j + \Delta\theta$, where $\Delta\theta = \mathbf{w}^T\mathbf{b}$

# Linear separability of the learning sets $G^+$ and $G^-$

## TASKS:

1. Detect linear separability of the learning sets $G^+$ and $G^-$ in the space $F[n$
2. Find a separating hyperplane $H(\mathbf{w}^*,\theta^*)$ in a given feature space $F[n]$
3. Find a ″good″ feature subspace $F^*[n_k]\subset F[n]$ (*feature subset selection*)

## METHODS:

*A*. Discriminant analysis based on the Fisher′s criterion function

$$\mathbf{w}^* = \mathbf{\Sigma}^{-1}(\boldsymbol{\mu}^+ - \boldsymbol{\mu}^-), \text{ where } \mathbf{\Sigma} \text{ is the covariance matrix}$$

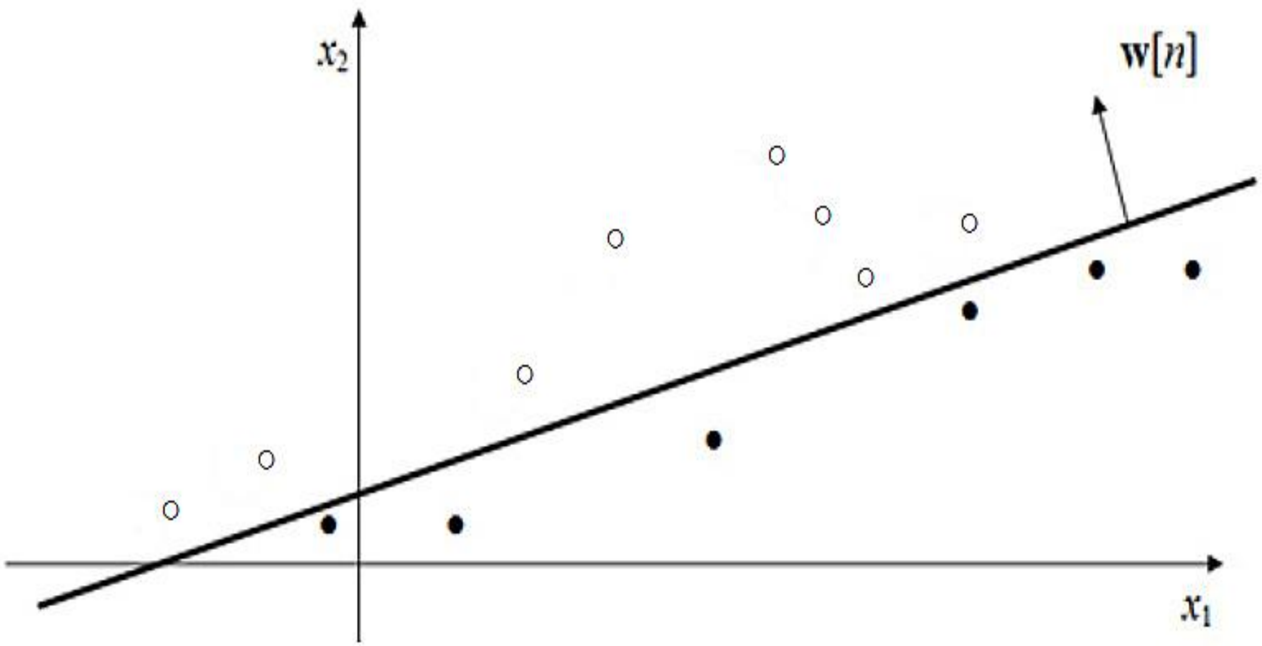
*B*. Singular Value Decomposition (*SVD*)

*C*. Support Vector Machines (*SVM*)

- quadratic programming is used for finding the minimum of the *SVM* criterion functio
- *SVM* is the most popular and successful method in bioinformatics

*D*. Convex and piecewise linear (*CPL*) criterion functions

- the basis exchange algorithms (*linear programming*) allow to find efficiently the minimum of the *CPL* criterion function
- *perceptron criterion function* belongs to the *CPL* family




*Example*: Each of the two features $x_1$ and $x_2$ individually has a very low discriminative power. But, the discriminative power of the set $\{x_1, x_2\}$ of the two features is very high.

# The beginnings of neural networks

1943 **McCulloch** and **Pitts** introduce a model for the neuron (*formal neuron*)

1949 **Hebb** postulates Learning-Paradigm (*reinforcement only for active neurons*)

1958 **Rosenblatt** develops the perceptron model (*single-layer perceptron*)

1962 **Rosenblatt** proves the Perceptron-Convergence-Theorem (*error correction* algorithm)

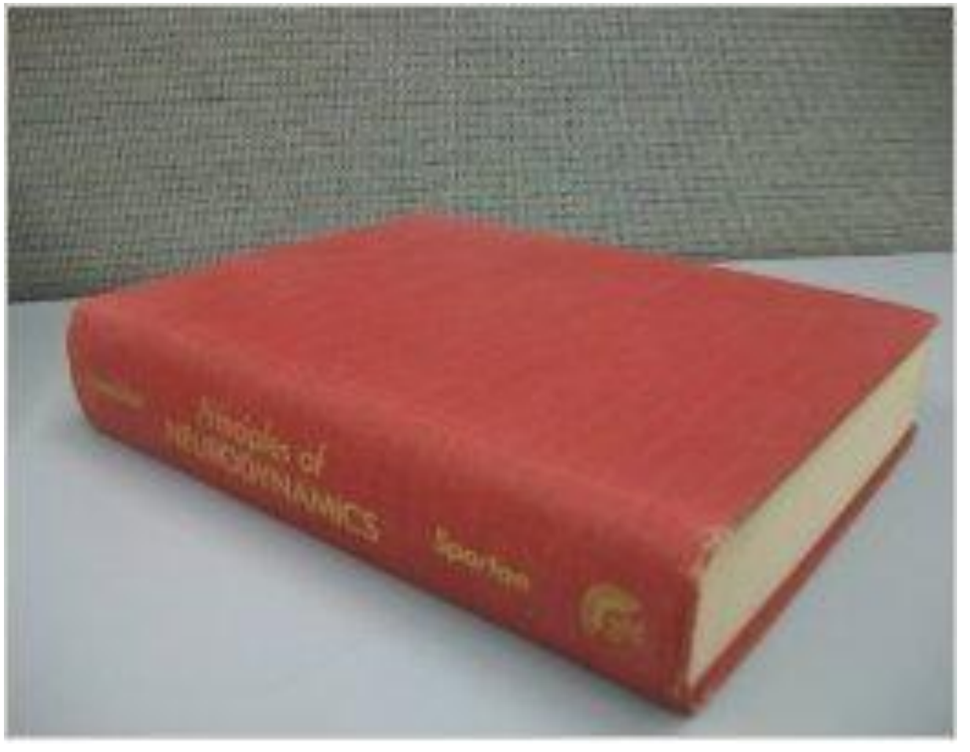
1969 **Minsky** & **Papert** publish a book regarding the limits of perceptrons (*XOR problem*)

1986 **Rumelhart** & **McClelland** present the Multilayer Perceptron (*back propagation* alghorithm)

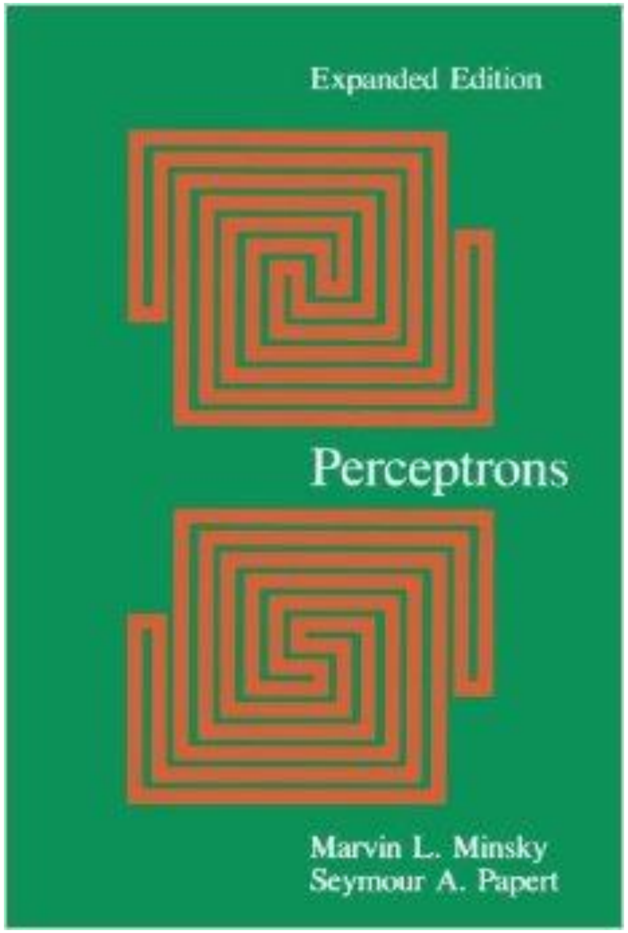
**Frank Rosenblatt**

***Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms*****,**

Spartan Books, Washington, 1962



**Marvin Minsky** and **Seymour Papert**
*Perceptrons*. Cambridge, MA: MIT Press, 1969

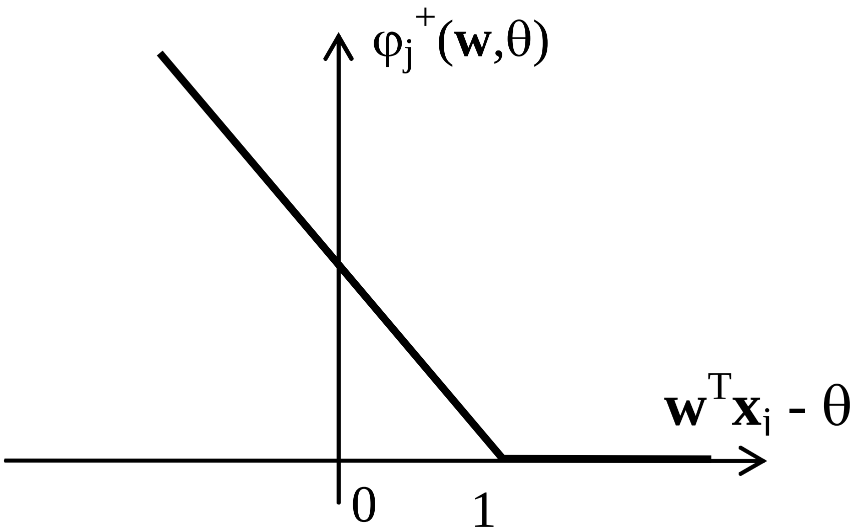


# III. Perceptron criterion functions

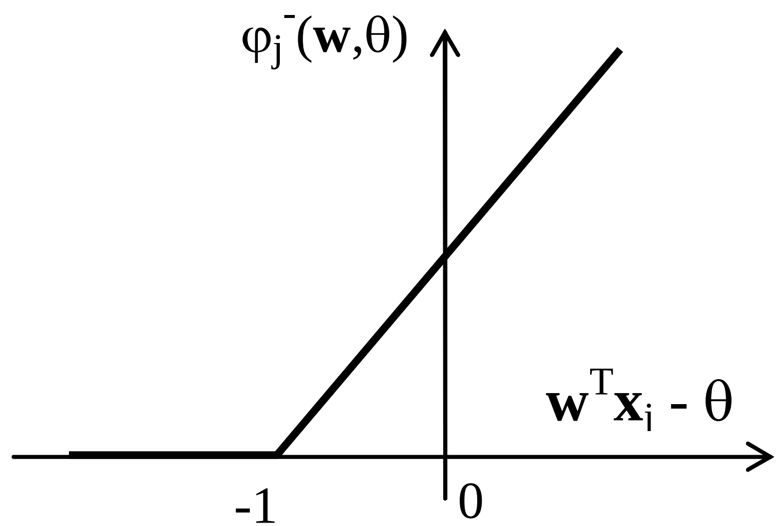
# Perceptron penalty functions

## (convex and piecewise linear functions – *CPL*)

$(\forall \mathbf{x}_j \in G^+)$




$(\forall \mathbf{x}_j \in G^-)$



# Perceptron penalty functions

$\varphi_j^+(\mathbf{w},\theta)$ and $\varphi_j^-(\mathbf{w},\theta)$

$(\forall \mathbf{x}_j \in G^+)$

$$\varphi_j^+(\mathbf{w},\theta) = \begin{cases} 1 + \theta - \mathbf{w}^T\mathbf{x}_j & \textbf{\textit{if}} \quad \mathbf{w}^T\mathbf{x}_j - \theta < 1 \\ 0 & \textbf{\textit{if}} \quad \mathbf{w}^T\mathbf{x}_j - \theta \geq 1 \end{cases}$$

and $(\forall \mathbf{x}_j \in G^-)$

$$\varphi_j^-(\mathbf{w},\theta) = \begin{cases} 1 - \theta + \mathbf{w}^T\mathbf{x}_j & \textbf{\textit{if}} \quad \mathbf{w}^T\mathbf{x}_j - \theta > -1 \\ 0 & \textbf{\textit{if}} \quad \mathbf{w}^T\mathbf{x}_j - \theta \leq -1 \end{cases}$$

# Perceptron criterion function $\Phi_p(\mathbf{w},\theta)$

.

$$\Phi_p(\mathbf{w},\theta) = \sum_{\mathbf{x}_j \in G+} \alpha_j \varphi_j^+(\mathbf{w},\theta) + \sum_{\mathbf{x}_j \in G-} \alpha_j \varphi_j^-(\mathbf{w},\theta)$$

where the nonnegative parameters $\alpha_j$ determine relative importance (***prices***) of particular feature vectors (patients) $\mathbf{x}_j$.

***Standard prices***: $\alpha_j = 1/(2m^+)$ for $\mathbf{x}_j \in G^+$,

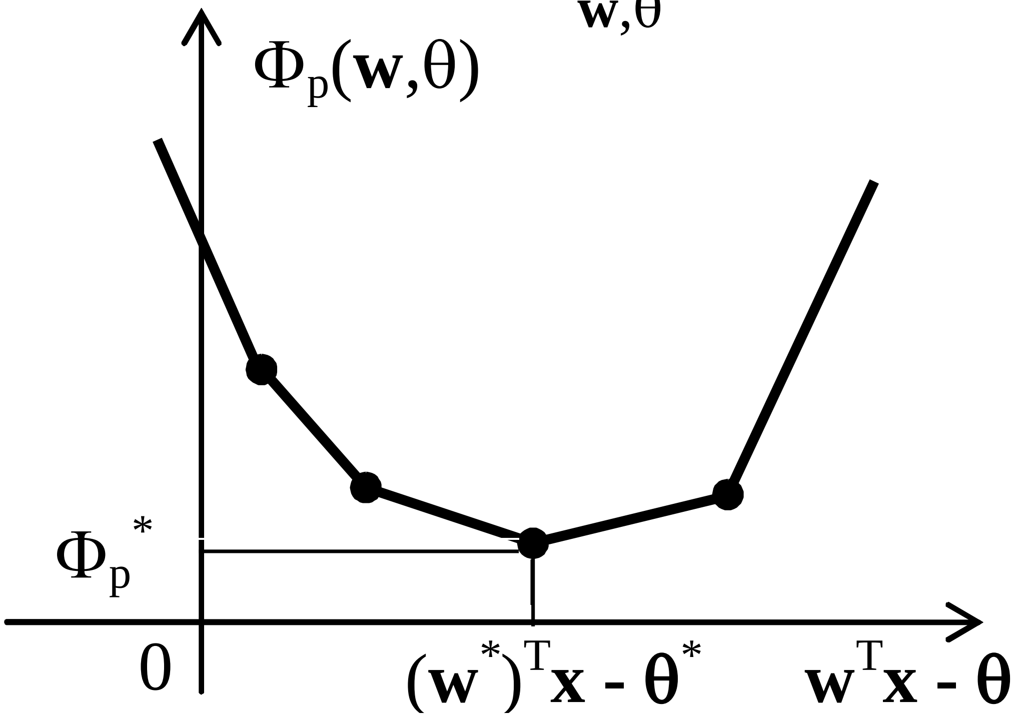
$\alpha_j = 1/(2m^-)$ for $\mathbf{x}_j \in G^-$,

where $m^+$ is the number of elements $\mathbf{x}_j$ in the set $G^+$, and $m^-$ is the number of elements $\mathbf{x}_j$ in the set $G^-$

$\Phi_p(\mathbf{w},\theta)$ is the *convex and piecewise linear* (*CPL*) function.

# The minimal value $\Phi_p^*$ of the perceptron criterion function $\Phi_p(\mathbf{w},\theta)$

$$\Phi_p^* = \Phi_p(\mathbf{w}^*,\theta^*) = \min_{\mathbf{w},\theta} \Phi_p(\mathbf{w},\theta)$$



# Perceptron criterion function $\Phi(\mathbf{w},\theta)$

Minimisation task:

$$\Phi^* = \Phi(\mathbf{w}^*,\theta^*) = \min_{\mathbf{w},\theta} \Phi(\mathbf{w},\theta)$$

The *basis exchange algorithms*, which are similar to the linear programming, allow to find in an efficient manner the optimal parameters ($\mathbf{w}^*,\theta^*$) and the minimal value $\Phi$ * of the criterion function $\Phi(\mathbf{w},\theta)$**,** even in the case of large, multidimensional data sets G+ and G-.

L. Bobrowski and W.Niemiro, "A method of synthesis of linear discriminant function in the case of nonseparabilty". *Pattern Recognition* **17,** pp.205-210,1984.

# BASIS EXCHANGE ALGORITHMS
# (*Gaus – Jordan transformation*)

Hyperplanes $h_j^+$ and $h_j^-$ in the (dual) **parameter space**:

$$(\forall\ \mathbf{x}_j \in G^+)\quad h_j^+ = \{\mathbf{w}: \mathbf{x}_j^T\mathbf{w} = 1\}$$

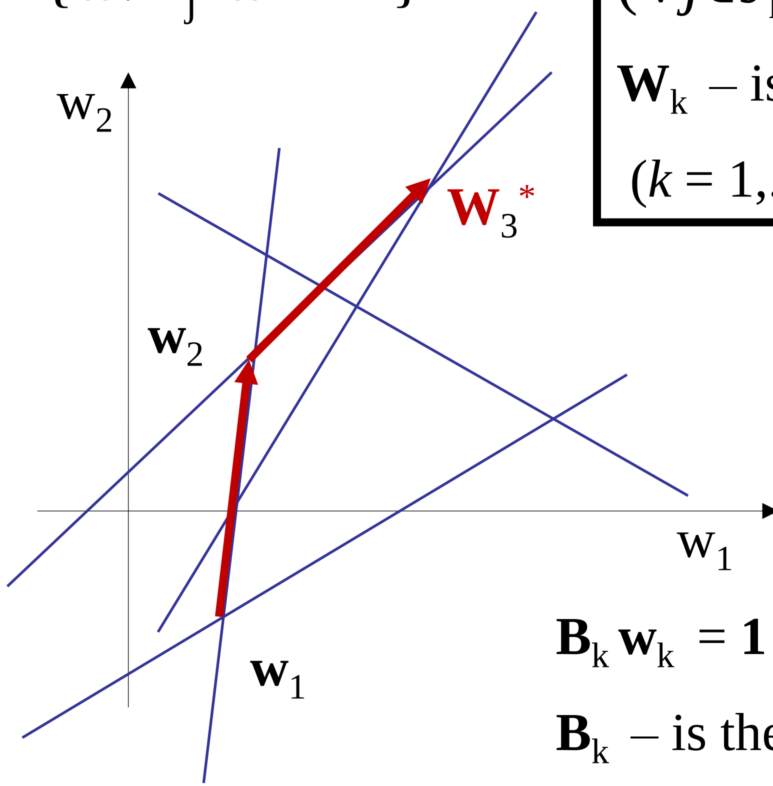
$$(\forall\ \mathbf{x}_j \in G^-)\quad h_j^- = \{\mathbf{w}: \mathbf{x}_j^T\mathbf{w} = -1\}$$

$(\forall j \in J_k^+)\ \ \mathbf{x}_j^T\mathbf{w}_k = 1$

$(\forall j \in J_k^-)\ \ \mathbf{x}_j^T\mathbf{w}_k = -1$

$\mathbf{W}_k$ – is the $k$-th **vertex**

$(k = 1,\ldots\ldots.,k_0)$




Equivalent matrix form:

$\mathbf{B}_k\,\mathbf{w}_k = \mathbf{1}$ *or* $\mathbf{w}_k = \mathbf{B}_k^{-1}\mathbf{1}$

$\mathbf{B}_k$ – is the $k$-th **basis**

($\mathbf{B}_k$ – nonsingular $n$x$n$ matrix )

# The minimal value $\Phi_p^*$ as the measure of nonseparability of the leaarning sets $G^+$ and $G^-$

• *Remark* 1 (*detection of linear separabilty*): The minimal value $\Phi_{\mathbf{p}}^*$ of the standardized criterion function $\Phi_p(\mathbf{w},\theta)$ is contained in the interval [0,1]

$$0 \leq \Phi_{\mathbf{p}}^* \leq 1$$

$\Phi_{\mathbf{p}}^* = 0$ if and only if the learning sets $G^+$ and $G^-$ are **linearly separable**.

• *Remark* 2 (the *positive monotonocity property*): Neglecting of arbitrary feature vector $\mathbf{x}_j$ from the learning sets can not increase the value of $\Phi_p^*$ (the value $\Phi_p^*$ usually ***decreases***)

• *Remark* 3 (the *negative monotonocity property*): Neglecting of arbitrary feature $x_i$ from vectors $\mathbf{x}_j$ belonging to set $G^+$ or $G^-$ can not decrease the value of $\Phi_p^*$ (the value $\Phi_p^*$ usually ***increases***)

• *Remark* 4 (the *invariancy property*): The minimal value $\Phi_p^*$ of the perceptron criterion function $\Phi_p(\mathbf{w},\theta)$ does not depend on linear, nonsingular transformations of feature vectors $\mathbf{x}_j$ :

*if* $(\forall \mathbf{x}_j \in G^+ \cup G^-)$ $\mathbf{y}_j = \boldsymbol{A}\, \mathbf{x}_j$, *where* $\boldsymbol{A}^{-1}$ *exists*, *then* $\Phi_y^* = \Phi_x^*$

*Lemma*: (the *invariancy property*): The minimal value $\Phi^*$ of the perceptron criterion function $\Phi(\mathbf{w},\theta)$ does not depend on affine, nonsingular transformations of feature vectors $\mathbf{x}_j$ :

*if* $(\forall \mathbf{x}_j \in G^+ \cup G^-)$ $\mathbf{y}_j = \boldsymbol{A}\,\mathbf{x}_j + \mathbf{b}$, *where* $\boldsymbol{A}^{-1}$ *exists*, *then* $\Phi_y^* = \Phi_x^*$

*Proof*:

*if* $\mathbf{y}_j = \boldsymbol{A}\,\mathbf{x}_j$ *and* $\mathbf{w}' = (\boldsymbol{A}^{-1})^T\mathbf{w}$, *then*

$$(\mathbf{w}')^T\mathbf{y}_j = ((\boldsymbol{A}^{-1})^T\mathbf{w})^T(\boldsymbol{A}\,\mathbf{x}_j) = \mathbf{w}^T(\boldsymbol{A}^{-1}\boldsymbol{A})\mathbf{x}_j = \mathbf{w}^T\mathbf{x}_j$$

*if* $\mathbf{y}_j = \mathbf{x}_j + \mathbf{b}$ *and* $\mathbf{w}' = \mathbf{w}$, *and* $\theta' = \theta - \mathbf{w}^T\mathbf{b}$, *then*

$$(\mathbf{w}')^T\mathbf{y}_j + \theta' = \mathbf{w}^T(\mathbf{x}_j + \mathbf{b}) + \theta' = \mathbf{w}^T\mathbf{x}_j + \mathbf{w}^T\mathbf{b} + \theta - \mathbf{w}^T\mathbf{b} = \mathbf{w}^T\mathbf{x}_j + \theta$$

*if* $\mathbf{y}_j = \boldsymbol{A}\,\mathbf{x}_j + \mathbf{b}$, *and* $\mathbf{w}' = (\boldsymbol{A}^{-1})^T\mathbf{w}$, *and* $\theta' = \theta - (\mathbf{w}')^T\mathbf{b}$, *then*

$$(\mathbf{w}')^T\mathbf{y}_j + \theta' = ((\boldsymbol{A}^{-1})^T\mathbf{w})^T(\boldsymbol{A}\mathbf{x}_j + \mathbf{b}) + \theta - ((\boldsymbol{A}^{-1})^T\mathbf{w})^T\mathbf{b} =$$

$$= (\mathbf{w})^T\mathbf{x}_j + \theta$$

# IV. Relaxed linear separability (RLS) method of feature subset selection

# Modified *CPL* criterion function $\Psi_\lambda(\mathbf{w},\theta)$ with feature costs

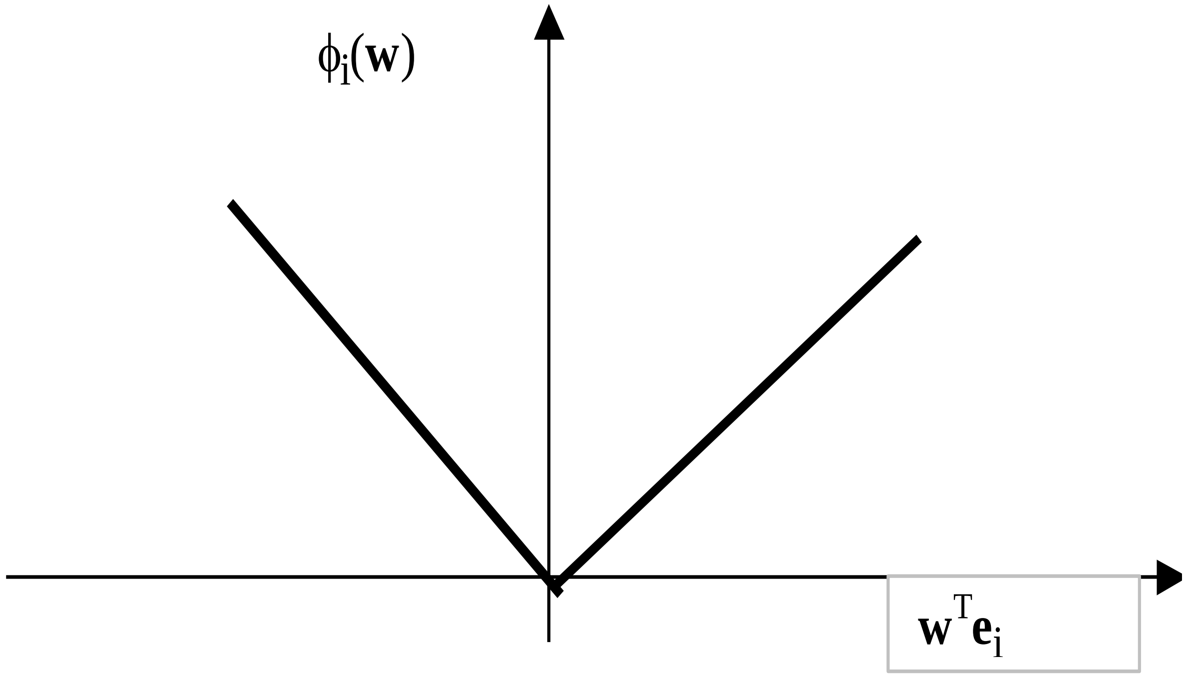
$$\Psi_\lambda(\mathbf{w},\theta) = \Phi_p(\mathbf{w},\theta) + \lambda \sum_{i\in I} \gamma_i \phi_i(\mathbf{w}) =$$

$$= \Phi_p(\mathbf{w},\theta) + \lambda \sum_{i\in I} \gamma_i |w_i|$$

where $\Phi_p(\mathbf{w},\theta)$ is the perceptron criterion function, $\phi_i(\mathbf{w})$ are additional penalty functions ($i = 1,\ldots\ldots,n$), $\gamma_i$ are the *costs* of particular features $x_i$ ($\gamma_i > 0$),

$\lambda$ ($\lambda \geq 0$) is the *cost level* ($\lambda \geq 0$), and $I = \{1,\ldots\ldots,n\}$ .

# Additional penalty functions $\phi_i(\mathbf{w})$ reflecting feature $x_i$ costs



# Additional penalty functions $\phi_i(\mathbf{w})$ reflecting feature $x_i$ costs

$$\phi_i(\mathbf{w}) = |w_i| = \begin{cases} -(\mathbf{e}_i)^T \mathbf{w} & \textbf{\textit{if}} \quad (\mathbf{e}_i)^T \mathbf{w} < 0 \\ (\mathbf{e}_i)^T \mathbf{w} & \textbf{\textit{if}} \quad (\mathbf{e}_i)^T \mathbf{w} \geq 0 \end{cases}$$

where $\mathbf{e}_i = [0,...,0,1,0,...,0]^T$ are the unit vectors ($i = 1,...,n$)

## Modified criterion function $\Psi_\lambda(w,\theta)$ with feature costs

$$\Psi_\lambda(\mathbf{w},\theta) = \Phi(\mathbf{w},\theta) + \lambda \sum_{i\in I} \gamma_i \, \phi_i(\mathbf{w}) =$$

$$= \Phi(\mathbf{w}, \theta) + \lambda \sum_{i\in I} \gamma_i \, |w_i|$$

The regularization component $\lambda \, \Sigma \, \gamma_i \, |w_i|$ used in the modified criterion function $\Psi_\lambda(\mathbf{w})$ is similar to that used in the ***Lasso*** method developed in the framework of the regression analysis for the purpose of *model selection*. The main difference between the *Lasso* and the *RLS* methods is in the types of the basic criterion functions. The basic criterion function used in the Lasso method is the *residual least squared* type. The *perceptron criterion function* $\Phi(\mathbf{w},\theta)$ is the basic function used in the *RLS* method. This difference affects, inter alia, the computational techniques used to minimize of the criterion functions. The criterion function $\Psi_\lambda(\mathbf{w},\theta)$ similarly to the function $\Phi(\mathbf{w},\theta)$ is convex and piecewise-linear (*CPL*).

# Relaxed Linear Separability (*RLS*) method of feature subsets selection

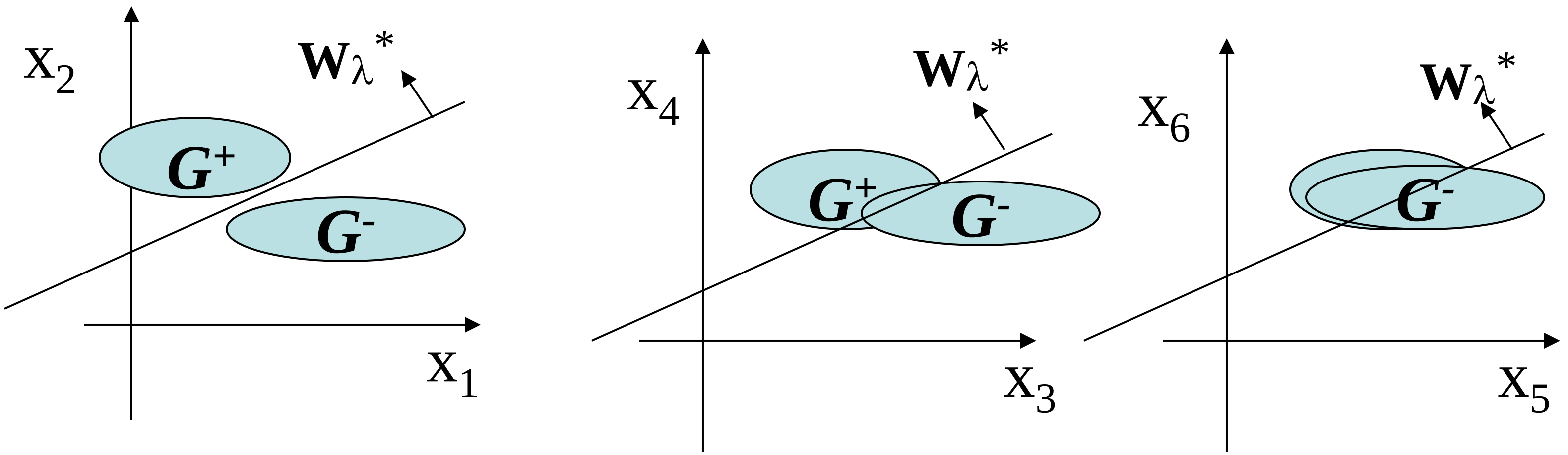
The *RLS* method is aimed at reduction of the maximum number of redundant features $x_i$ under the condition that the linear separability of the learning sets $G^+$ and $G^-$ is sufficiently preserved. This method is based on reapeated minimization of the modified criterion function $\Psi_\lambda(\mathbf{w},\theta)$.


Bobrowski L. and Łukaszuk T.: Relaxed linear separability (RLS) approach to feature (gene) subset selection, *Selected Works in Bioinformatics*, Xuhua Xia (Ed.), *INTECH* 2011, pp.103-118.

Bobrowski L., Łukaszuk T: Feature selection based on relaxed linear separability, *Biocybernetics and Biomedcal Engineering* 2009 (Volume 29, Number 2, pp. 43-59)

*FEATURE SELECTION BASED ON*
***RELAXED LINEAR SEPARABILITY***

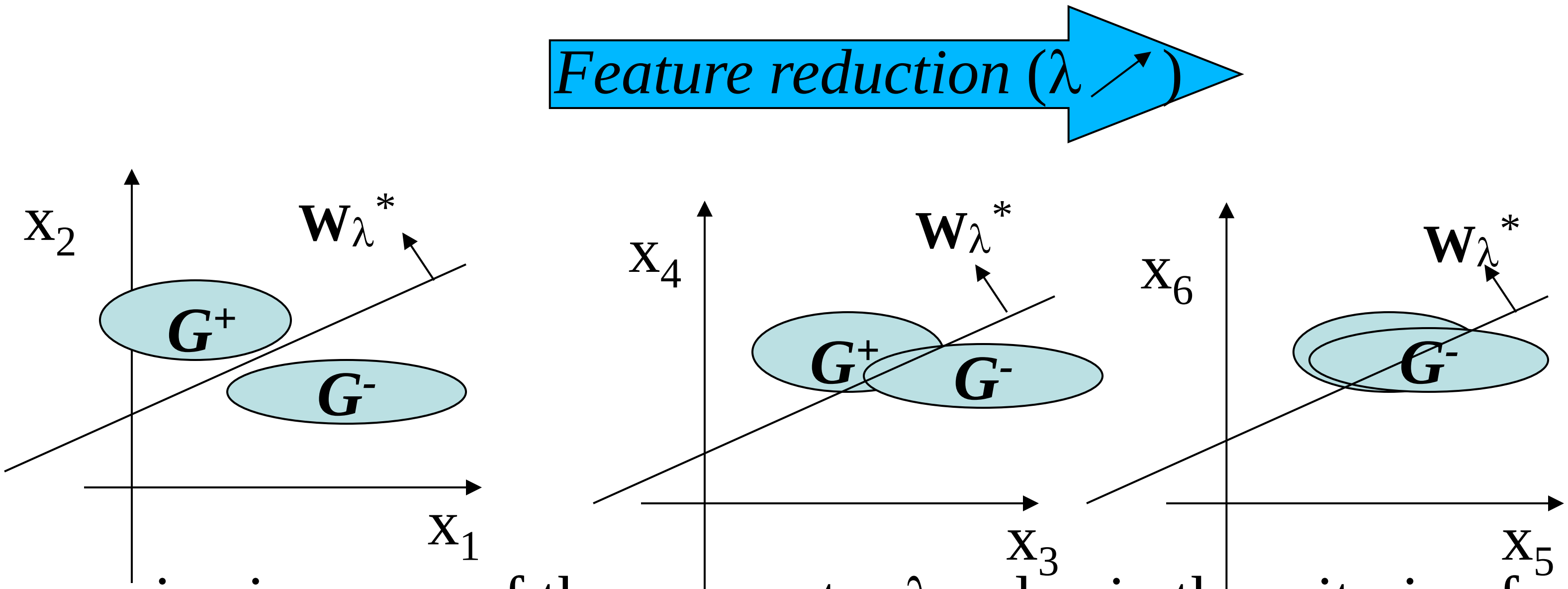
*Feature reduction* ($\lambda \nearrow$)




$\Psi_p(\mathbf{v}^*) = 0$

We are searching for such minimal feature subset which assures sufficiently high degree of linear separability of learning sets $G^+$ and $G^-$ The maximal number of unecessary features $x_i$ should be removed from data sets. The remaining features may indicate for the ***differential pattern*** (differential feature subset) of a given disease.

# *FEATURE SELECTION BASED ON*
# ***RELAXED LINEAR SEPARABILITY***




Successive increase of the parameter $\lambda$ value in the criterion function $\Psi_\lambda(\mathbf{w},\theta)$ allows to generate the descended sequence of ***feature subspaces*** $F_k[n_k]$:

$$F[n] \supset F_1[n_1] \supset \ldots \supset F_{k'}[n_{k'}]$$

where $n_k > n_{k+1}$.

The sequence of the feature subspaces $F_k[n_k]$ is generated in a deterministic manner in accordance with the relaxed linear separability method. The feature subspace $F_{k'}[n_{k'}]$ is determined by the stop criterion.

# *EATURE SELECTION BASED ON RELAXED LINEAR SEPARABILITY*

*Example*: Four dimensional feature space ($n$ = 4)

$\{x_1, x_2, x_3, x_4\}$

$\{x_1, x_2, x_3\}$ $\{x_1, x_2, x_4\}$ $\{x_1, x_3, x_4\}$ $\{x_2, x_3, x_4\}$

$\{x_1, x_2\}$ $\{x_1, x_4\}$ $\{x_2, x_4\}$

$\{x_2\}$ $\{x_4\}$

*Remark*:The number of feature subsets grows rapidly as $2^n$ -1 with the dimensionality $n$ of feature space.

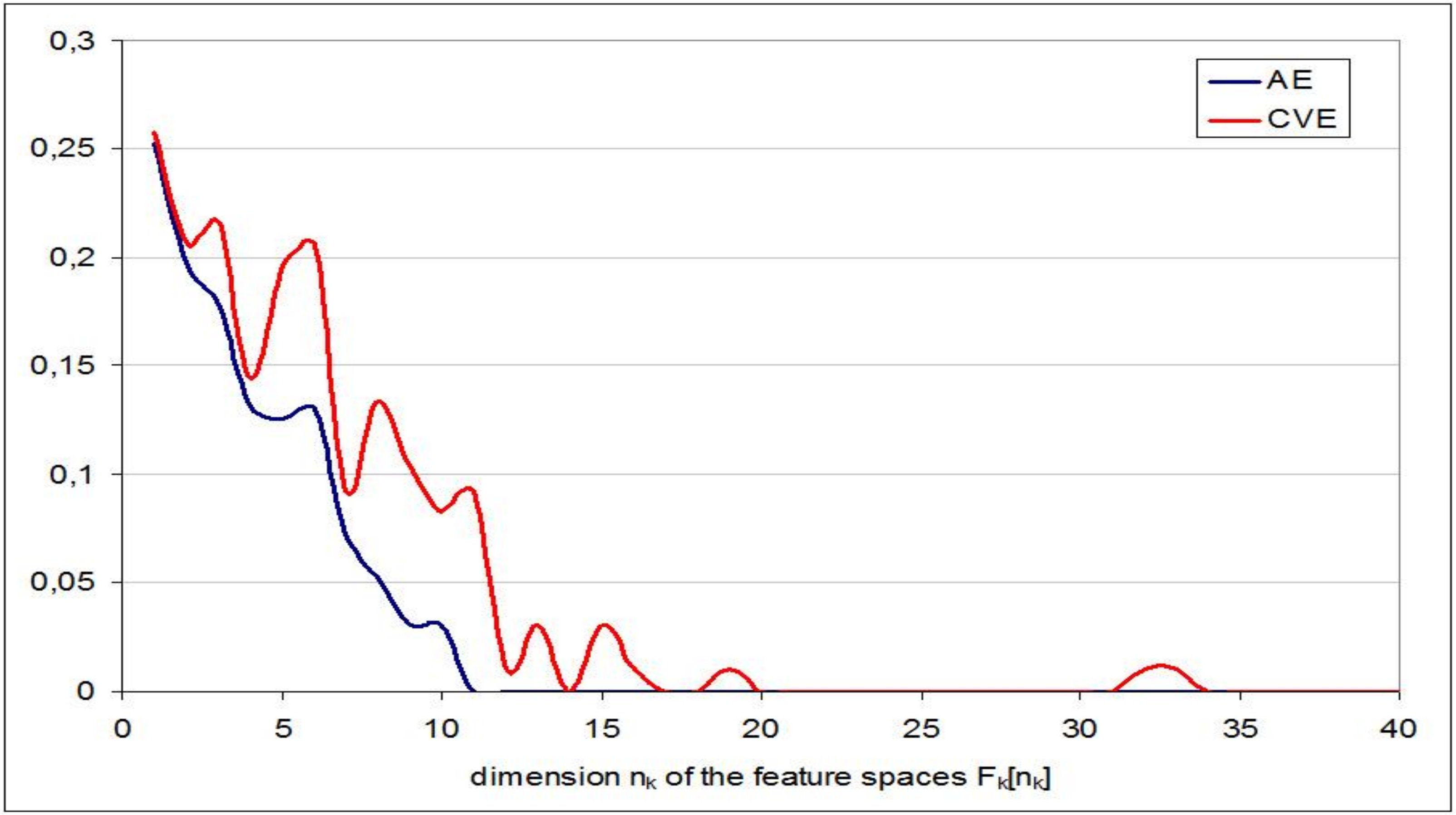
# *Example* 1: **Results of the *RLS* feature selection**

The ***Breast cancer*** data set (van’t Veer et al., 2002) describes the patients tested for the presence of breast cancer. The data contains **97** patient samples, **46** of which are from patients who had developed distance metastases within 5 years, the rest **51** samples are from patients who remained healthy.

The number of genes in this data set is equal to **24481**.

van’t Veer, L. J., et al. (2002). Gene expression profiling predicts clinical outcome of breast cancer, *Nature*, 415(6871), pp. 530–536

# *Example* 1: **Results of the *RLS* feature selection**




The apparent error (AE) and the cross-validation error (CVE) in different feature subspaces $F_k[n_k]$ of the *Breast cancer* data set.

Bobrowski L. ,Łukaszuk T. (2011) Relaxed Linear Separability (*RLS*) Approach to Feature (Gene) Subset Selection, in: *Bioinformatics*, INTECH

## *EXAMPLE* 1*: **FEATURE SUBSET SELECTION BASED ON THE RELAXED LINEAR SEPARABILITY (RLS) METHOD***

- High efficiency of the *CPL* procedures allows, among others, to use the *RLS* method for selection of optimal gene subsets which are characterized by a high discriminative power. For example, the *RLS* method were applied to the ***Breast Cancer*** data set which contains descriptions of **46** cancer and **51** non-cancer patients. Each patient was characterized in this set by $n$ = **24481** genes. The *RLS* method allowed to select the optimal subset of $n_1$ = **12** genes and such linear combination of these genes (*linear key*), which allows to correctly distinguish cancer from non-cancer patients in this set – with 100% accuracy.
- This example demonstrates the ability to use data mining based on the *CPL* criterion functions also when the number of features $n$ is very high and many times greater than the number of objects $m$.
- Bobrowski L. and Łukaszuk T.: Relaxed linear separability (RLS) approach to feature (gene) subset selection, Selected Works in Bioinformatics, Xuhua Xia (Ed.), INTECH 2011, pp.103-118.

# Modified *CPL* criterion function $\Psi_\lambda(\mathbf{w},\theta)$ with equal feature costs

$$\Psi_\lambda(\mathbf{w},\theta) = \Phi_p(\mathbf{w},\theta) + \lambda \sum_{i\in\{1,\ldots,n\}} \phi_i(\mathbf{w}) =$$

$$= \Phi_p(\mathbf{w},\theta) + \lambda \sum_{i\in\{1,\ldots,n\}} |w_i|$$

where $\Phi_p(\mathbf{w},\theta)$ is the perceptron criterion function, and

$\lambda$ ($\lambda \geq 0$) is the *cost level* ($\lambda \geq 0$).

The *CPL* optimal weight vector $\mathbf{w}_\lambda^* = [w_{\lambda 1}^*,\ldots,w_{\lambda n}^*]^T$:

$$(\forall(\mathbf{w},\theta)) \quad \boldsymbol{\Psi}_\lambda(\mathbf{w},\theta) \geq \Psi_\lambda(\mathbf{w}_\lambda^*,\theta_\lambda^*)$$

**The *CPL* optimal weight vector $\mathbf{w}_\lambda^* = [w_{\lambda 1}^*, \ldots, w_{\lambda N}^*]^T$ in the case of linearly separable learning sets $G^+$ and $G^-$**

$$\sum_{i \in \{1,\ldots,N\}} |w_{\lambda i}^*| = \{min\, (\sum_{i \in \{1,\ldots,N\}} |w_i|): \mathbf{w} \in \boldsymbol{R}\}$$

*Remark*: The *CPL* optimal vertex $\mathbf{v}_k^* = [-\theta_k^*, \mathbf{w}_k^*]^T$ of the set $\boldsymbol{R}$ is characterised by the lowest $L_1$ length of the weight vector $\mathbf{w}_k^*$.

## The *SVM* optimal weight vector $\mathbf{w}_{SVM}^* = [w_1^*,\ldots,w_n^*]^T$ in the case of linearly separable learning sets $G^+$ and $G^-$

$$(\mathbf{w}_{SVM}^*)^T\mathbf{w}_{SVM}^* = \{min\ (\mathbf{w}^T\mathbf{w}) \colon \mathbf{w} \in \boldsymbol{R}\}$$

*Remark*: The *SVM* optimal vector $\mathbf{w}_{SVM}^*$ is characterized by the lowest Euclidean $L_2$ norm, in contrast to the $L_1$ norm used in the *CPL* solution $\mathbf{w}_{CPL}^*$.

# *CPL* crtiterion function approach versus Support Vector Machines (*SVM*) in data mining

1. The history of the *CPL* approach could be dated back to the beginning of the neural networks theory (*perceptron* criterion function).
2. *SVM* method is based on the quadratic programming and the *CPL method* - on the linear programming. We develop basis exchange algorithms which are similar to the linear programming. These algorithms allow to find the minimum of single *CPL* criterion functions efficiently, even in case of large, multidimensional data sets.
3. The *CPL* method can be also used to design a variety of data mining tools, such as hierarchical neural networks, **ranked regression models**, prognostic models with censored data, multivariate decision trees or visualising transformations.
4. *CPL* approach allows to integrate the designing data mining tools with the **feature selection** process.

Bobrowski L. and Łukaszuk T.: Relaxed linear separability (RLS) approach to feature (gene) subset selection, *Selected Works in Bioinformatics*, Xuhua Xia (Ed.), *INTECH* 2011, pp.103-118.

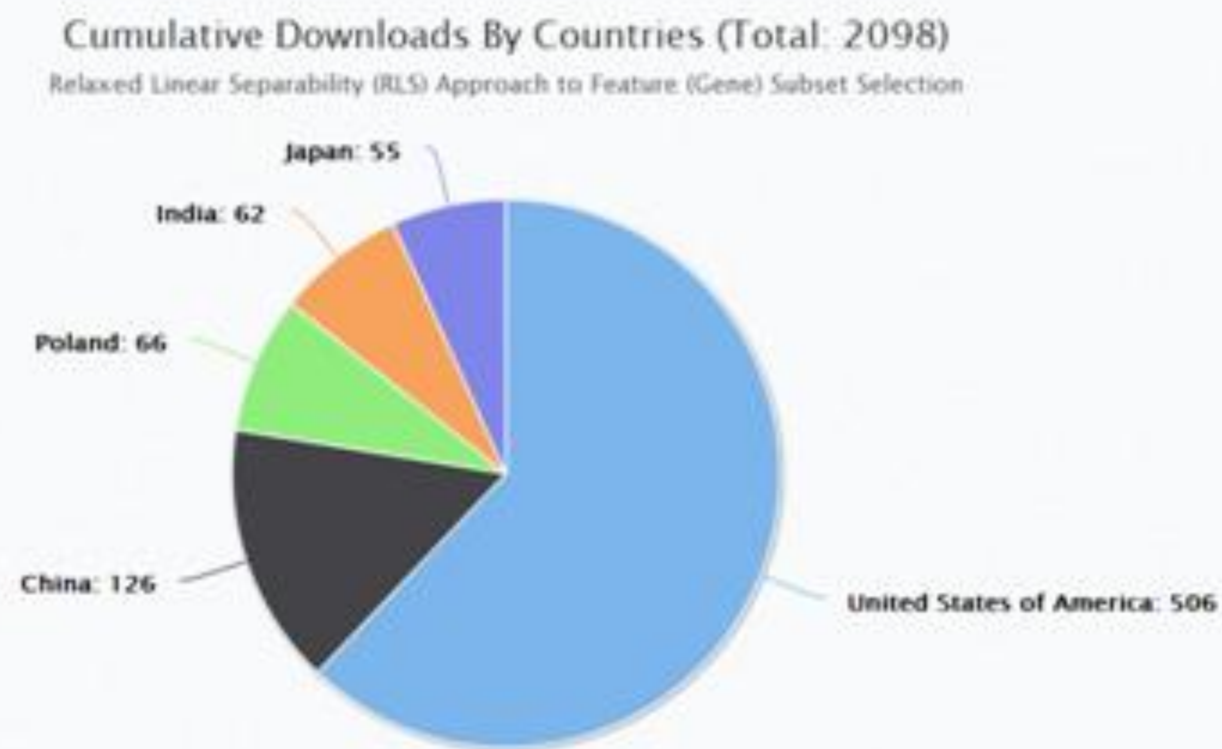



Fig. 3. Pie chart shows the download share by TOP 5 countries from which this publication was accessed.

# V. Interval regression models

# Linear prognostic models

$$T(\mathbf{x}[n]) = \mathbf{w}[n]^{T}\mathbf{x}[n] + w_0 = \qquad (1)$$
$$= w_1x_1 + \ldots. + w_nx_n + w_0 =$$
$$= \mathbf{v}[n+1]^{T}\mathbf{y}[n+1]$$

$T(\mathbf{x}[n])$ - the prognostic model of an unknown survival time $T$ of the patient $O$ represented by the feature vector $\mathbf{x}[n] = [x_1,\ldots,x_n]^T$, where $\mathbf{w}[n] = [w_1,\ldots,w_n]^T$ is the *weight* vector ($\mathbf{w}[n] \in R^n$), $w_0$ is the *threshold* ($w_0 \in R$

$\mathbf{y}[n+1] = [1, \mathbf{x}[n]^T]^T$ is the *augmented* feature vector,
$\mathbf{v}[n+1] = [-w_0, \mathbf{w}[n]^T]^T$ is the *augmented* weight vector.

The parameters $\mathbf{w}[n]$ and $\theta$ of the model (1) are estimated on the basis of a given *data set C*.

# *Learning data set in classical regression*

In the **classical regression** *additional knowledge* about feature vectors $\mathbf{x}_j[n]$ is provided by the accompanying values $t_j$ of the *dependent variable* $T$, where $t_j \in R^1$. The learning sets $C_m$ have the below form:

$$C_m = \{\mathbf{x}_j[n], t_j\} \qquad (j \in \{1,\ldots..,m\}) \tag{2}$$

The parameters $\mathbf{w}[n]$ and θ can be estimated through minimization of the *mean squared error* (*MSE*) or the *mean absolute error* (*MAE*)

$$MSE(\mathbf{w}[n], \mathrm{w}_0) = \sum_{j=1,\ldots,m} (T(\mathbf{x}_j[n]) - t_j)^2 = \sum_{j=1,\ldots,m} (\mathbf{w}[n]^T\mathbf{x}_j[n] + \mathrm{w}_0 - t_j)^2 \rightarrow min$$

$$MAE(\mathbf{w}[n], \mathrm{w}_0) = \sum_{j=1,\ldots,m} |T(\mathbf{x}_j[n]) - t_j| = \sum_{j=1,\ldots,m} |\mathbf{w}[n]^T\mathbf{x}_j[n] + \mathrm{w}_0 - t_j| \rightarrow min$$

# *Least squares estimation in classical regression*

In the **classical regression** *additional knowledge* about feature vectors $\mathbf{y}_j[n+1] = [1,\mathbf{x}_j[n]^T]^T$ is provided by the accompanying values $t_j$ of the *dependent variable T*, where $t_j \in R^1$. The learning sets $C_m$ have the below form:

$$C_m = \{\mathbf{y}_j[n+1], t_j\} \qquad (j \in \{1,\ldots..,m\})$$

The optimal parameters $\mathbf{v}^*[n+1] = [-\,\theta^*, \mathbf{w}^*[n]^T]^T$ of the model $T(\mathbf{y}[n+1]) = \mathbf{v}[n+1]^T\mathbf{y}[n+1]$ are often estimated through minimization of the *mean squared error* (*MSE*):

$$MSE(\mathbf{v}[n+1]) = \sum_{j=1,\ldots,m} (\mathbf{v}[n+1]^T\mathbf{y}_j[n+1] - t_j)^2 = \rightarrow min$$

$$\mathbf{v}^*[n+1] = (\boldsymbol{Y}^T \boldsymbol{Y})^{-1} \boldsymbol{Y}^T \mathbf{t}$$

where $\mathbf{t} = [t_1,\ldots, t_m]^T$ and $\boldsymbol{Y}^T = [\mathbf{y}_1[n+1], \ldots, \mathbf{y}_m[n+1]]$.

# *Learning data set in the interval regression*

In the case of the **interval regression** *additional knowledge* about feature vectors $\mathbf{x}_j[n]$ is given in the form of *intervals* $[t_j^-, t_j^+]$:

$$C_m' = \{\mathbf{x}_j[n], [t_j^-, t_j^+]\} \qquad (\forall j \in \{1,\ldots..,m\}\ t_j^- < t_j^+) \qquad (3)$$

The parameters $\mathbf{w}[n]$ and $w_0$ of the interval regression model (1) can be estimated through the postulated inequalites:

$$(\forall j \in \{1,\ldots,m\}) \qquad t_j^- < T(\mathbf{x}_j[n]) < t_j^+ \qquad (4)$$

or

$$(\forall j \in \{1,\ldots.,m\}) \quad t_j^- < \mathbf{w}[n]^T\mathbf{x}_j[n] + w_0 < t_j^+ \qquad (5)$$

The interval regression model can also be estimated on the basis of the classical learning set $C_m$ (2) by using a small positive *margin* $\varepsilon$ ($\varepsilon > 0$):

$$(\forall\ j \in \{1,\ldots..,m\}) \quad t_t - \varepsilon < \mathbf{w}[n]^T\mathbf{x}_j[n] + w_0 < t_j + \varepsilon \qquad (6)$$

# *Estimation of interval regression parameters*

$$(\forall j \in \{1,\ldots, m\}) \qquad t_j^- < \mathbf{w}[n]^T\mathbf{x}_j[n] + w_0 < t_j^+ \qquad (9)$$

$$\textit{or} \quad t_j^- < \mathbf{w}'[n+1]^T\mathbf{x}'[n+1] < t_j^+$$

*Problem*: How to estimate the parameters $\mathbf{w}'[n+1] = [\mathbf{w}[n]^T, w_0]^T$ on the basis of the learning set $C_m' = \{\mathbf{x}_j[n], [t_j^-, t_j^+]\}$ (3)?

1. Method of the *Expectation Maximization* (*EM*)

2. Method based on the *linear separability* exploration through the minimization of the convex and piecewise linear (*CPL*) criterion function

# The censored survival times $T_j$

The censored survival times $T_j$ can be represented by intervals $[t_j^-, t_j^+]$ and by the *indicators of censoring* $\delta_j$ of ($\delta_j \in \{-1, 0, 1\}$):

$$\begin{array}{llll} \textbf{\textit{if}} \ \delta_j = 0 & \textbf{\textit{then}} \ T_j \in [t_j^-, t_j^+] & (t_j^- < T_j < t_j^+) & \\ \textbf{\textit{if}} \ \delta_j = -1 & \textbf{\textit{then}} \ T_j \in (-\infty, t_j^+] & (T_j < t_j^+) & \text{- } \textbf{\textit{left censoring}} \\ \textbf{\textit{if}} \ \delta_j = 1 & \textbf{\textit{then}} \ T_j \in [t_j^-, +\infty) & (t_j^- < T_j) & \text{- } \textbf{\textit{right censoring}} \end{array} \tag{10}$$

The rules (10) allow to introduce the below set of the postulated linear inequalities:

$(\forall j \in \{1, \ldots, m\})$

$$\begin{array}{llll} \textbf{\textit{if}} \ \delta_j = 0 & \textbf{\textit{then}} & t_j^- < \mathbf{w}[n]^T \mathbf{x}_j[n] + w_0 < t_j^+ & \\ \textbf{\textit{if}} \ \delta_j = -1 & \textbf{\textit{then}} & \mathbf{w}[n]^T \mathbf{x}_j[n] + w_0 < t_j^+ & \text{- } \textbf{\textit{left censoring}} \\ \textbf{\textit{if}} \ \delta_j = 1 & \textbf{\textit{then}} & t_j^- < \mathbf{w}[n]^T \mathbf{x}_j[n] + w_0 & \text{- } \textbf{\textit{right censoring}} \end{array} \tag{11}$$

## Augmented feature vectors $\mathbf{z}_j^+[n+2]$ and $\mathbf{z}_j^-[n+2]$

$(\forall j \in \{1,\ldots, m\})$ (12)

**if** $(\delta_j \geq 0)$ **then** $\mathbf{z}_j^+[n+2] = [\mathbf{x}_j[n]^T, 1, -t_j^-]^T$, **else** $\mathbf{z}_j^+[n+2] = \mathbf{0}$

**if** $(\delta_j \leq 0)$ **then** $\mathbf{z}_j^-[n+2] = [\mathbf{x}_j[n]^T, 1, -t_j^+]^T$, **else** $\mathbf{z}_j^-[n+2] = \mathbf{0}$

***and***

$$\mathbf{v}[n+2] = [v_1,\ldots,v_{n+2}]^T = [\mathbf{w}[n]^T, w_0, \beta]^T$$

where $\mathbf{v}[n+2] \in R^{n+2}$ and $\beta$ is the interval parameter ($\beta \in R^1$).

## The positive set $\mathbf{Z}^+[n+2]$ and the negative set $\mathbf{Z}^-[n+2]$

The positive set $\mathbf{Z}^+[n+2]$ and the negative set $Z^-[n+2]$ are composed of such $(n + 2)$ - dimensional vectors $\mathbf{z}_j^+[n+2]$ $(j \in J^+)$ and $\mathbf{z}_j^+[n+2]$ ] $(j \in J^+)$ that are different from zero :

$$\mathbf{Z}^+[n+2] = \{\mathbf{z}_j^+[n+2]: j \in J^+\} \quad and \tag{13}$$

$$\mathbf{Z}^-[n+2] = \{\mathbf{z}_j^-[n+2]\ ]: j \in J^-\}$$

## *Linear separability of the sets* $\mathbf{Z}^{+}[n+2]$ *and* $\mathbf{Z}^{+}[n+2]$

We are examining the possibility of separating the sets $\mathbf{Z}^{+}[n+2]$ and $\mathbf{Z}^{-}[n+2]$ by the hyperplane $H(\mathbf{v}'[n+2],0)$ in the $(n + 2)$ – dimensional feature space $F[n+2]$.

*Definition*: The sets $Z^{+}$ and $Z^{-}$ (12) are *linearly separable* if and only if the below conditions are fulfilled:

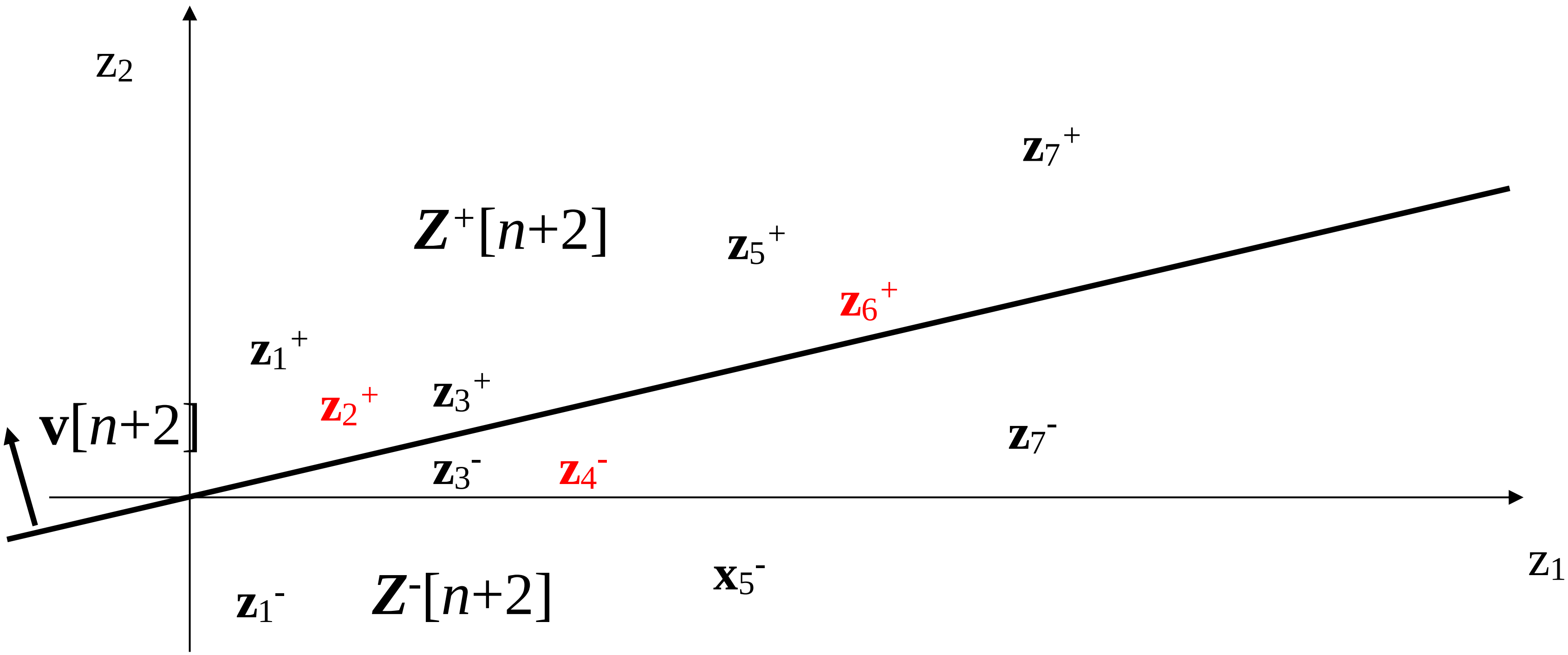
$$(\exists\ \mathbf{v}'[n+2] = [\mathbf{w}'[n+1]^{T}, \beta']^{T} \tag{14}$$

$$(\forall j \in \{1,\ldots., m\}) \qquad \mathbf{v}'[n+2]^{T}\mathbf{z}_{j}^{+}[n+2] \geq 1$$

$$\textit{and} \qquad \mathbf{v}'[n+2]^{T}\mathbf{z}_{j}^{-}[n+2] \leq -1$$

# *Linear separability of the sets* $\mathbf{Z}^{+}[n+2]$ *and* $\mathbf{Z}^{+}[n+2]$

If the inequalities (13) hold, then all the elements $\mathbf{z}_j^{+}[n+2]$ of the set $\mathbf{Z}^{+}[n+2]$ (12) can be situated on the positive side of the hyperplane $H(\mathbf{v}'[n+2],0)$ and all the elements $\mathbf{x}_j^{-}[n+2]$ of the set $R^{-}$ can be situated on the negative side of this hyperplane.




$[-\infty, \mathbf{z}_2^{+}[n+2]]$, $[-\infty, \mathbf{z}_6^{+}[n+2]]$ - the ***left censored*** observations

$[\mathbf{z}_4^{+}[n+2], +\infty]$ - the ***right censored*** observation

# Minimisation of the *CPL* criterion function $\Phi(\mathbf{w})$

The basis exchange algorithms which are similar to the linear programming, allow to find the minimum of the function $\Phi(\mathbf{v})$ in an efficient manner, even in the case of large, multidimensional data sets $\boldsymbol{Z}^{+}$ and $\boldsymbol{Z}^{-}$ (13):

$$\Phi^{*} = \Phi(\mathbf{v}^{*}) = \min_{\mathbf{v}} \Phi(\mathbf{v}) \geq 0 \qquad (15)$$

The optimal parameter vector $\mathbf{v}^{*}[n+2] = [\mathbf{w}^{*}[n]^{\mathrm{T}}, \mathrm{w}_0^{*}, \beta^{*}]^{\mathrm{T}}$ can be used in the definition of the optimal prognostic model (1)

$$T^{*}(\mathbf{x}[n]) = (\mathbf{w}^{*}[n] / \beta^{*})^{\mathrm{T}}\mathbf{x}[n] + \mathrm{w}_0^{*} / \beta^{*} \qquad (16)$$

# *Example* 2: **Prognostic model selection on the *Breast Cancer* survival data set**

**<u>Data set:</u>**

The ***Breast cancer*** data set (van't Veer et al., 2002, van de Vijver et al., 2002) consists of patient samples from **primary invasive breast carcinomas**.

Number of patients: **295**

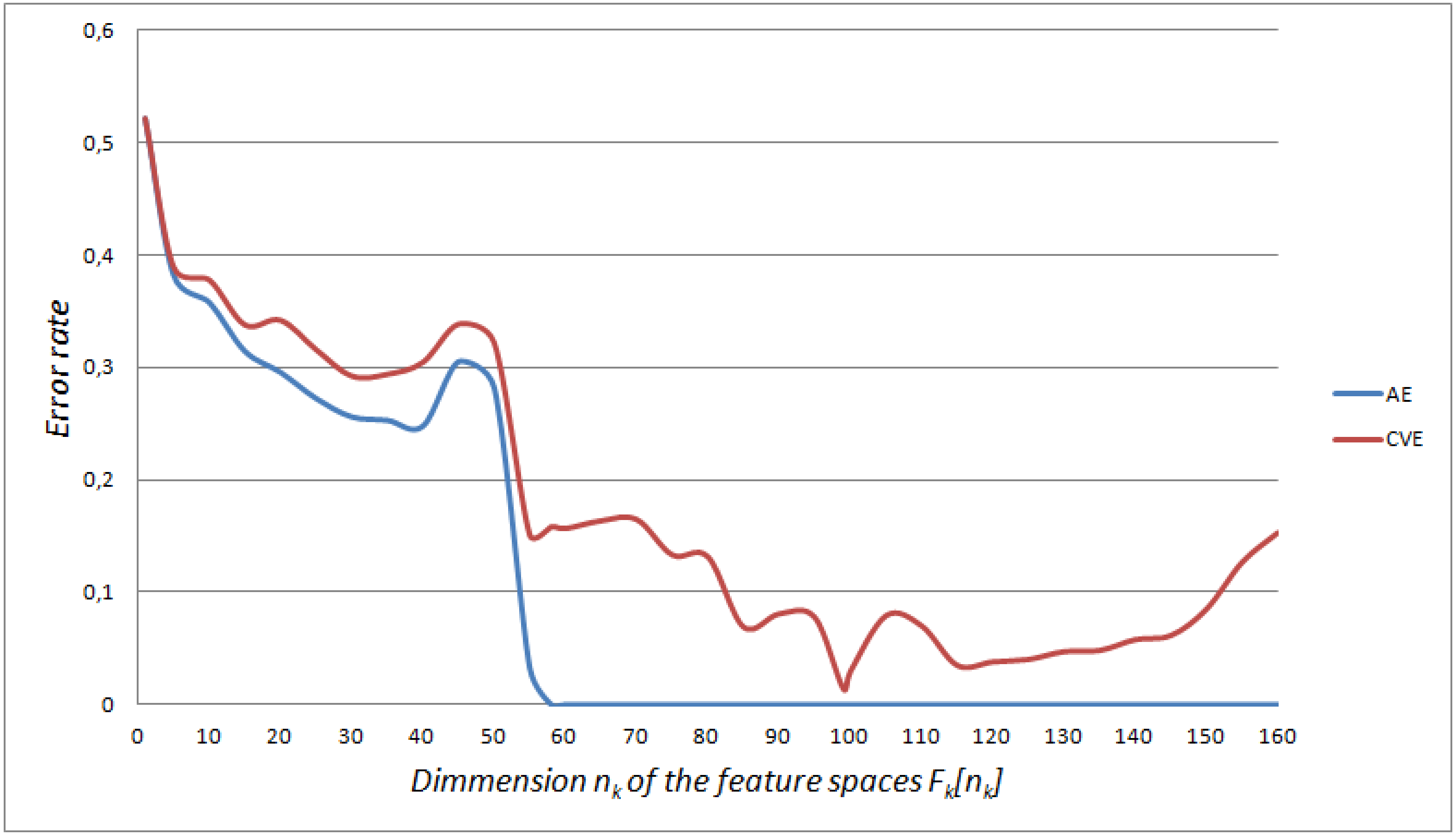
Number of features (genes): **4919**

Each patients has a **specified time value** measured from start of observation until death or censoring. 216 patients (73%) were still alive at the final follow-up visit (*censored observations*).

van't Veer, L. J., et al. (2002). Gene expression profiling predicts clinical outcome of breast cancer, *Nature*, 415(6871), pp. 530–536

Vijver M.J. van de, et al. (2002). A gene-expression signature as a predictor of survival in breast cancer, N Engl J Med, 347:1999-2009.

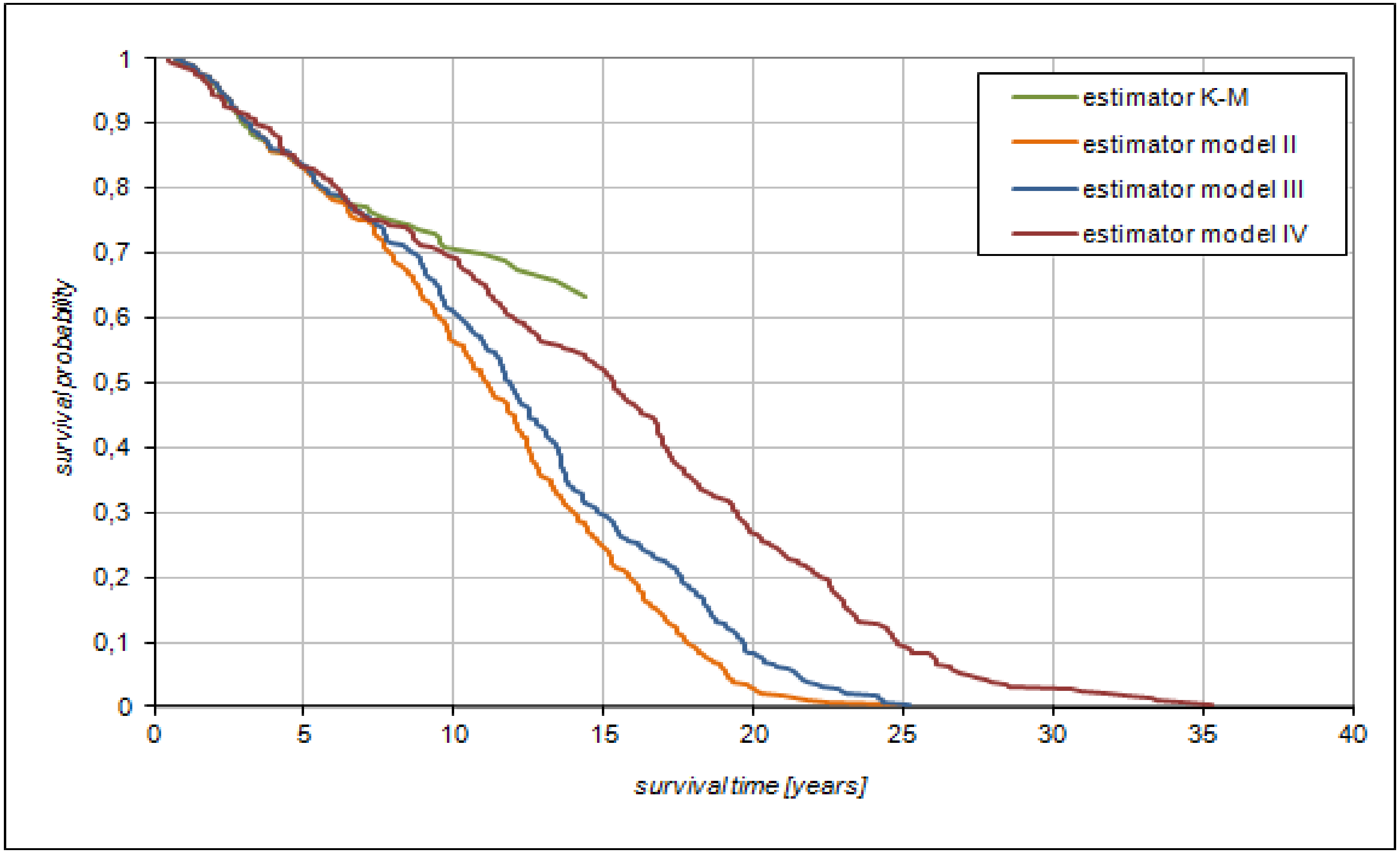
# *Example* 2: **Results of the *RLS* feature selection**




The apparent error (AE) and the cross-validation error (CVE) in different feature subspaces $F_k[n_k]$ of the *Breast cancer* data set.

Bobrowski L, Łukaszuk T.: Prognostic Modeling with High Dimensional and Censored Data, pp. 178 – 193 in: *Advances in Data Mining*, P. Perner (Ed.), Springer, Berlin 2012

# Survival probability




K-M –Kaplan-Meier estimator
model II - $T^*(\mathbf{x}[160])$ maximal margin
model III - $T^*(\mathbf{x}[99])$ minimal *CVE*
model IV - $T^*(\mathbf{x}[58])$ second *RLS* stop criterion

# Remarks

The *CPL* criterion functions allows to combine the feature subset selection process with a search for the optimal parameters of the designed prognostic models (model selection). Such procedure gives possibility for designing regression models also on the basis of such high-dimensional data as genetic data sets with censored values of dependent variable. This novel approach was presented for the first time at the conference *ICDM* 2012 (*Industrial Conference on Data Mining*) in Berlin (L. Bobrowski, T. Łukaszuk, Prognostic Modeling with High Dimensional and Censored Data, pp. 178 – 193 in: *Advances in Data Mining*, P. Perner (Ed.), Springer, Berlin 2012). The article was honored by the *Best Paper Aword*: http://www.data-mining-forum.de/paper_award_2012.php

# Industrial Conference on Data Mining

**Berlin 2012**

Best Paper Award 2012

Sculpture "*Everything is possible*"

# VI. Ranked regression models

# *Learning data set in the ranked regression*

In the case of the **ranked regression** some *additional knowledge* about feature vectors $\mathbf{x}_j[n]$ is given in the form of *ranked* relationship "$\mathbf{x}_j[n] \prec \mathbf{x}_k[n]$" inside selected pairs $\{\mathbf{x}_j[n], \mathbf{x}_k[n]\}$, where $(j,k) \in I_p$. In this case, the learning data set $C_m''$ can have the below form:

$$C_m'' = \{\mathbf{x}_j[n], "\mathbf{x}_j[n] \prec \mathbf{x}_k[n]"\} \tag{17}$$

where $j \in \{1,\ldots..,m\}$ and $(j,k) \in I_p$.

The linear transformation $y = \mathbf{w}[n]^T\mathbf{x}[n]$ constitutes the *ranked regression model* if it preserves the below implications for a possibly large number of the *ranked* relations "$\mathbf{x}_j[n] \prec \mathbf{x}_k[n]$":

$$(\mathbf{x}_j[n] \prec \mathbf{x}_k[n]) \Rightarrow (\mathbf{w}[n]^T\mathbf{x}_j[n] < \mathbf{w}[n]^T\mathbf{x}_k[n]) \tag{18}$$

# Ranked linear transformations

Linear transformations $y = \mathbf{w}^T\mathbf{x}$ of $n$-dimensional feature vectors $\mathbf{x}_j$ ($\mathbf{x}_j \in R^n$) on the points $y_j$ on the line $R^1$ ($y_j \in R^1$):

$$(\forall j \in \{1,\ldots, m\}) \qquad y_j = \mathbf{w}[n]^T\mathbf{x}_j[n] \tag{19}$$

where $\mathbf{w}[n] = [w_1,\ldots, w_n]^T$ is the parameter vector.

*Definition* 2: The line $y = \mathbf{w}[n]^T\mathbf{x}[n]$ constitutes the *ranked risk model* if it preserves the below implications for possibly large number of the *ranked relations* "$O_j \prec O_k$" (3):

$$(O_j \textit{ is less risky than } O_k) \Rightarrow (\mathbf{w}[n]^T\mathbf{x}_j[n] < \mathbf{w}[n]^T\mathbf{x}_k[n]) \tag{20}$$

# Ranked linear transformation - *Example*

ranked relations

$\mathbf{x}_4 \prec \mathbf{x}_1$

$\mathbf{x}_1 \prec \mathbf{x}_5$

$\mathbf{x}_5 \prec \mathbf{x}_3$

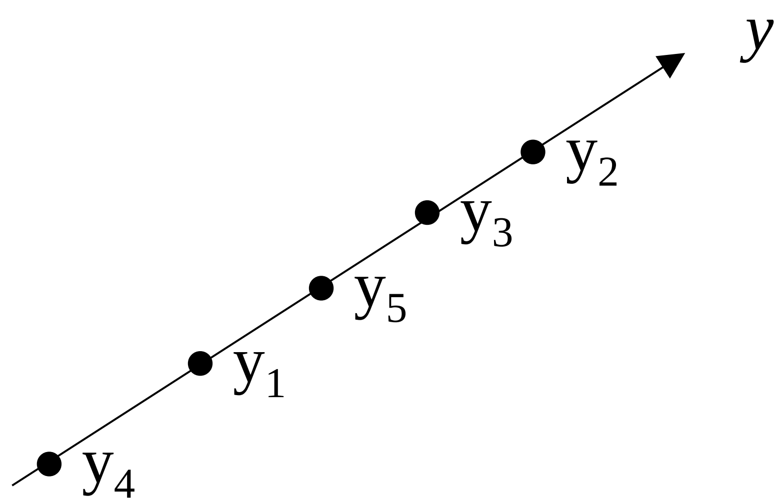
$\mathbf{x}_3 \prec \mathbf{x}_2$

ranked linear transformation

$\Rightarrow$

$y_j = \mathbf{w}^T \mathbf{x}_j$

a ″trend″ in data




L. Bobrowski, “Ranked modelling with feature selection based on the *CPL* criterion functions”, in: *Machine Learning and Data Mining in Pattern Recognition*, Eds. P. Perner et al., *Lecture Notes in Computer Science*, vol. 3587, Springer Verlag, Berlin 2005

# Ranked relations in survival data

*Definition* 1: If the ***real survival time*** $T_j$ of the *j*-th patient $O_j$ is greater than the time $T_k$ of the *k*-th patient $O_k$, then the **ordinal relation** "$O_j \prec O_k$" ("$O_j$ *is less risky than* $O_k$") takes place.

$$(T_j > T_k) \Rightarrow (O_j \prec O_k) \qquad (21)$$

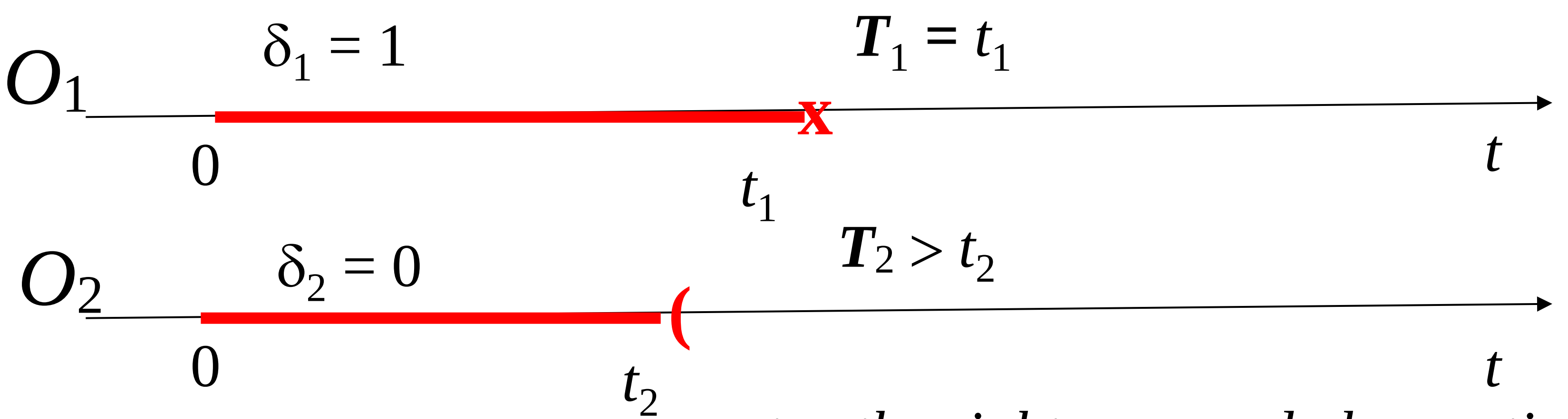
or

$$(\delta_k = 1 \textit{ and } t_j > t_k) \Rightarrow (O_j \prec O_k) \qquad (22)$$

# **Survival data** (cont.)

The ***real survival time $T_j$*** is the time interval between the entry of the *j*-th patient $O_j$ into the study and the failure (*event*, *death*), where

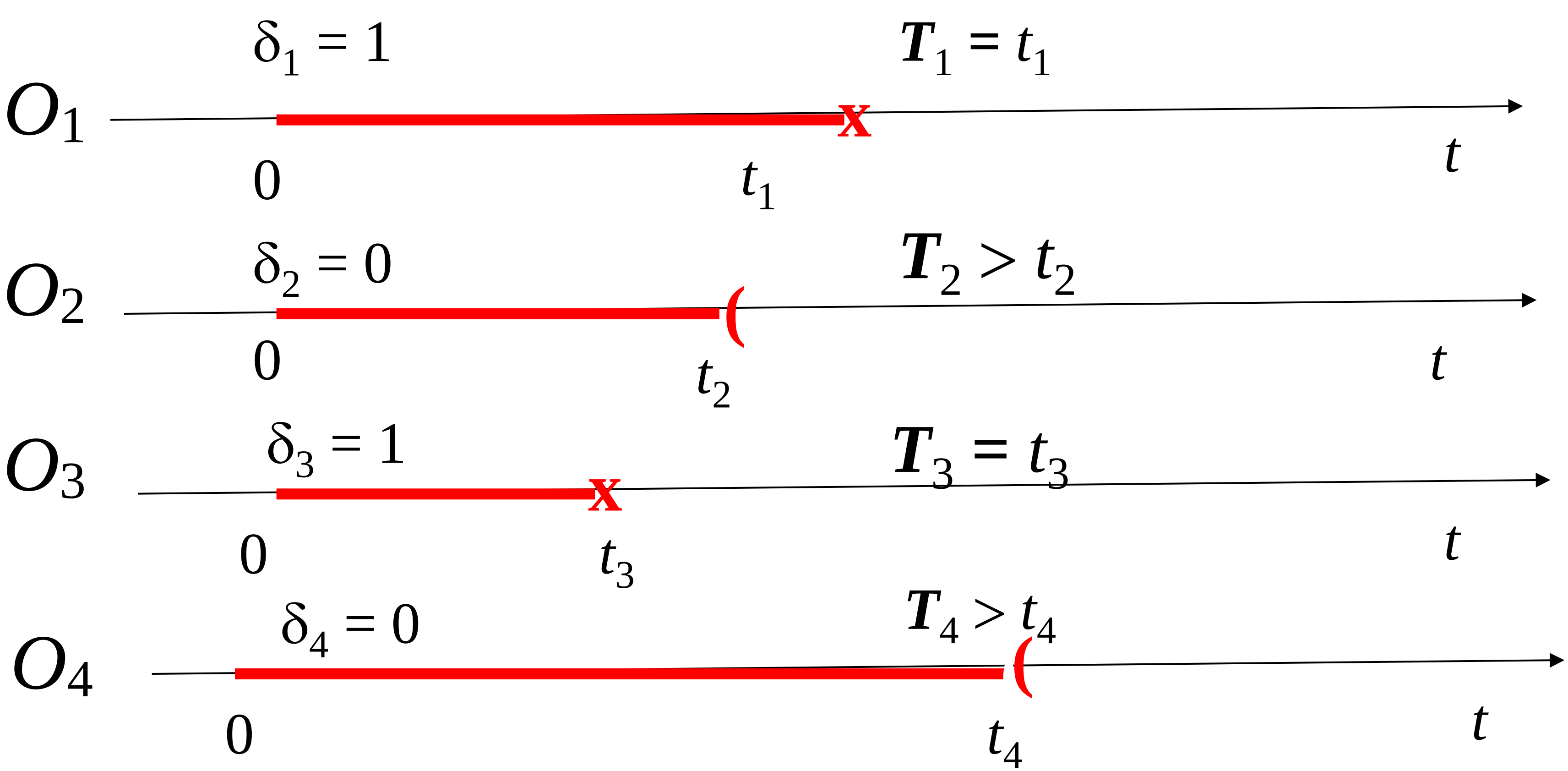
$$(\forall j \in \{1,\ldots.,m\}) \qquad \begin{array}{ll} T_j = t_j & \textit{if} \quad \delta_j = 1 \\ T_j > t_j & \textit{if} \quad \delta_j = 0 \end{array} \qquad (23)$$




$t_2$ - *the right censored observation*

# **Survival data** (cont.)

*Example* 1 (*the right censored observations $t_2$ and $t_4$*):

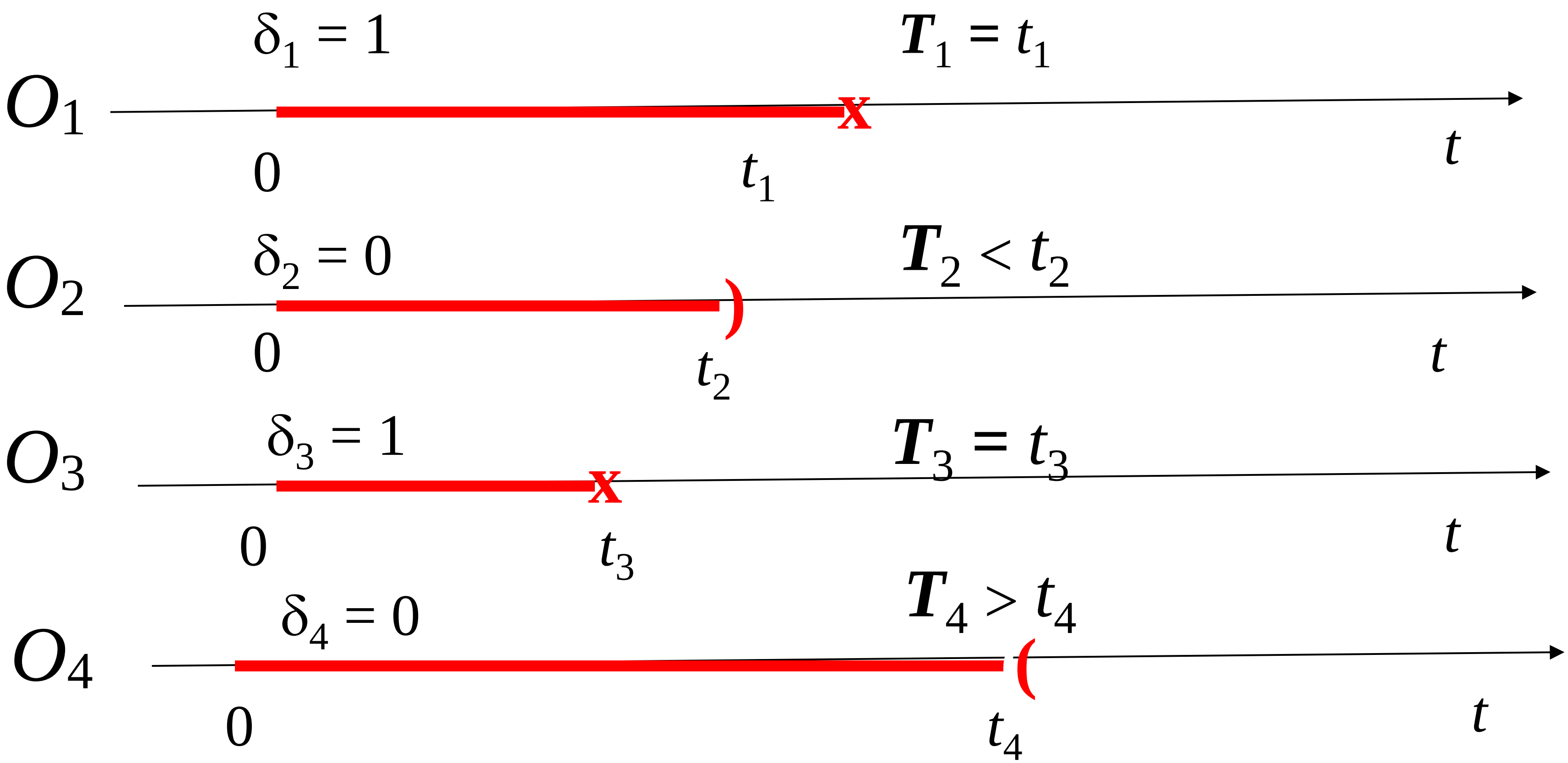



*Ranked relations:* $O_1 \prec O_3$; $O_2 \prec O_3$; $O_4 \prec O_1$; $O_4 \prec O_3$

*There are no ranked relations between patients* $O_2$ ***and*** $O_1$ *or* $O_2$ ***and*** $O_4$

# **Survival data** (cont.)

*Example* 2 (*the left censored observation* $t_2$ *and the right censored observation* $t_4$):




*Ranked relations:* $O_1 \prec O_2$, $O_1 \prec O_3$, $O_4 \prec O_1$, $O_4 \prec O_2$, $O_4 \prec O_3$.

*There is no ranked relation between patients* $O_2$ ***and*** $O_1$.

# Positive and negative sets
## of the differential vectors

The positive $G^+$ and the negative $G^-$ sets of the differential vectors $\mathbf{r}_{jj'} = \mathbf{x}_{j'} - \mathbf{x}_j$:

$$G^+ = \{\mathbf{r}_{jj'} = \mathbf{x}_{j'} - \mathbf{x}_j\text{:}\ \ j < j' \text{ and } \ O_j \prec O_{j'}\} \quad (24)$$
$$G^- = \{\mathbf{r}_{jj'} = \mathbf{x}_{j'} - \mathbf{x}_j\text{:}\ \ j < j' \text{ and } \ O_{j'} \prec O_j\}$$

We are examining the possibility of the sets $G^+$ and $G^-$ separation by a hyperplane $H(\mathbf{w})$ which passes through the origin $\mathbf{0}$ of the feature space:

$$H(\mathbf{w}) = \{\mathbf{x}\text{: } \mathbf{w}^T\mathbf{x} = 0\} \quad (25)$$

# Linear separability with the treshold equal to zero

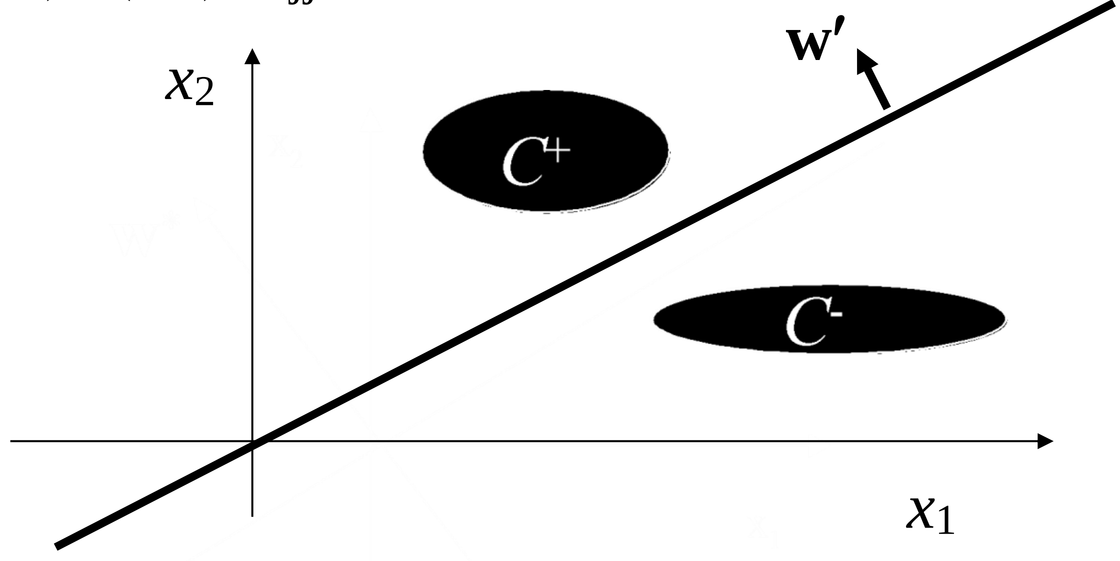
*Definition* 3: The sets $G^+$ and $G^-$ (7) are linearly separable with the threshold equal to zero if and only if there exists such a parameter vector $\mathbf{w}'$ that:

$$(\forall\, \mathbf{r}_{jj'} \in G^+) \quad (\mathbf{w}')^T \mathbf{r}_{jj'} > 0 \qquad (26)$$

$$(\forall\, \mathbf{r}_{jj'} \in G^-) \quad (\mathbf{w}')^T \mathbf{r}_{jj'} < 0$$

*or* $$(\exists \mathbf{w}')\, (\forall \mathbf{r}_{jj'} \in G^+) \;\; (\mathbf{w}')^T \mathbf{r}_{jj'} \geq 1 \qquad (27)$$

$$(\forall \mathbf{r}_{jj'} \in G^-) \;\; (\mathbf{w}')^T \mathbf{r}_{jj'} \leq -1$$



## *CPL* penalty functions $\varphi_{jj'}^{+}(\mathbf{w})$ and $\varphi_{jj'}^{+}(\mathbf{w})$

$(\forall \mathbf{r}_{jj'} \in G^{+})$

$$\varphi_{jj'}^{+}(\mathbf{w}) = \begin{cases} 1 - \mathbf{w}^{T}\mathbf{r}_{jj'} & \textbf{\textit{if}} \;\; \mathbf{w}^{T}\mathbf{r}_{jj'} < 1 \\ 0 & \textbf{\textit{if}} \;\; \mathbf{w}^{T}\mathbf{r}_{jj'} \geq 1 \end{cases} \tag{29}$$

*and*

$(\forall \mathbf{r}_{jj'} \in G^{-})$

$$\varphi_{jj'}^{-}(\mathbf{w}) = \begin{cases} 1 + \mathbf{w}^{T}\mathbf{r}_{jj'} & \textbf{\textit{if}} \;\; \mathbf{w}^{T}\mathbf{r}_{jj'} > -1 \\ 0 & \textbf{\textit{if}} \;\; \mathbf{w}^{T}\mathbf{r}_{jj'} \leq -1 \end{cases} \tag{30}$$

# *C*riterion function $\Phi(\mathbf{w})$

The criterion function $\Phi(\mathbf{w})$ is the weighted sum of the penalty functions $\varphi_{jj'}^{+}(\mathbf{w})$ and $\varphi_{jj'}^{-}(\mathbf{w})$

$$\Phi(\mathbf{w}) = \sum_{(j,j')\in I+} \gamma_{jj'}\,\varphi_{jj'}^{+}(\mathbf{w}) + \sum_{(j,j')\in I-} \gamma_{jj'}\,\varphi_{jj'}^{-}(\mathbf{w}) \qquad (31)$$

where $\gamma_{jj'}$ ($\gamma_{jj'} > 0$) is a positive parameter (*price*) related to the pair $\{\mathbf{x}_j,\mathbf{x}_{j'}\}$ ($j < j'$).

$I^+$ is the set of indices $(j, j')$ of the vectors $\mathbf{r}_{jj'}$ belonging to $G^+$.

$I^-$ is the set of indices $(j, j')$ of the vectors $\mathbf{r}_{jj'}$ belonging to $G^-$.

# **VII.** Diagnostic maps of the system Hepar

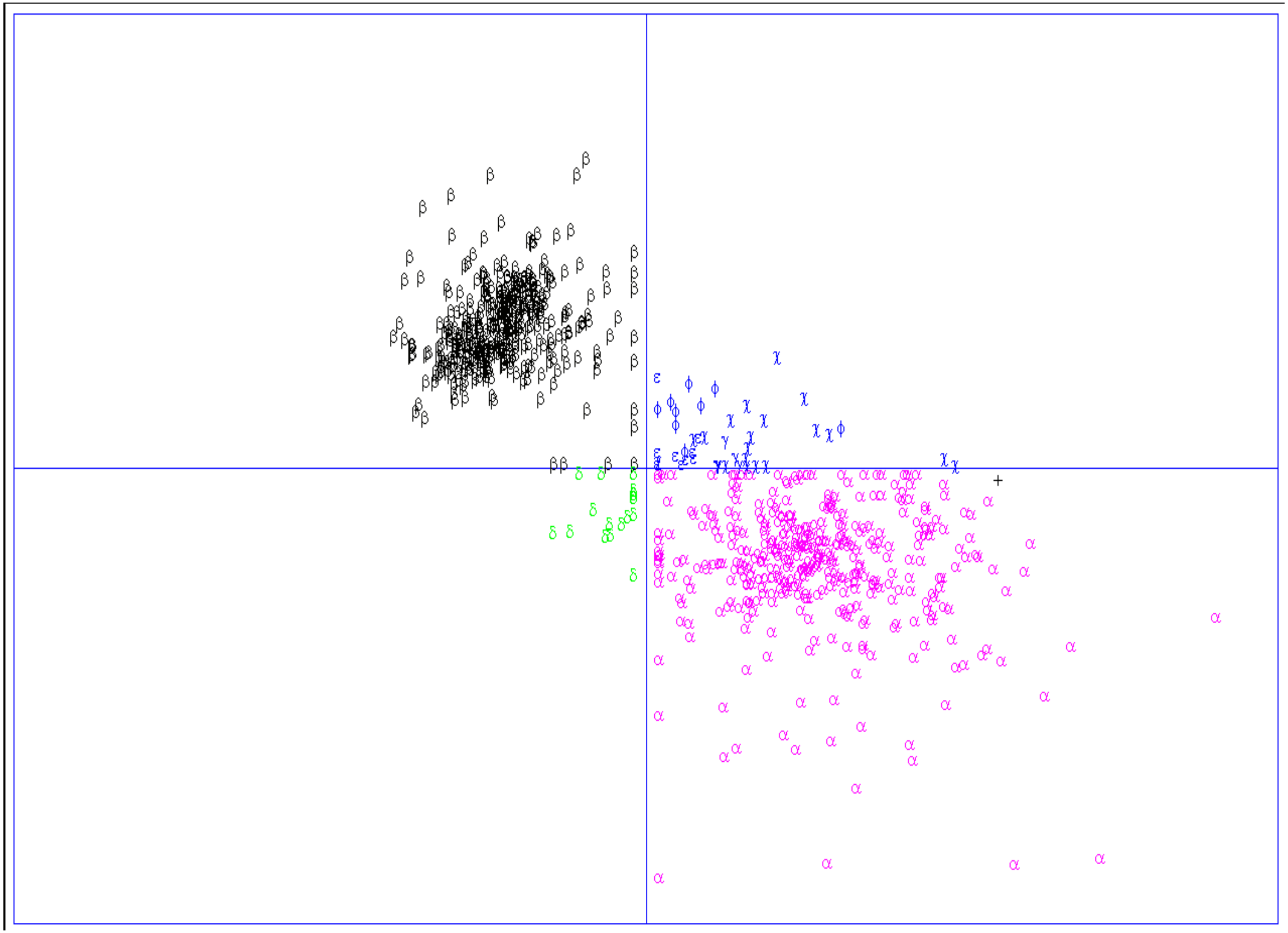
L. Bobrowski, H. Wasyluk, “Induction of Diagnostic Support Rules through Data Mapping - on the Example of the Hepar system”, pp. 3 – 14 in: *Biocybernetics and Biomedical Engineering*, Vol. 27, Nr 3, 2007

# Diagnostic maps of the system *Hepar*

The learning sets $C_k$ represent seven liver diseases $\omega_k$ listed below:

| | |
|---|---|
| $\omega_1$ – *Cirrhosis hepatis* | $C_1$ – 382 patients |
| $\omega_2$ – *Hepatitis chronica* | $C_2$ – 373 patients |
| $\omega_3$ – *Carcinoma* | $C_3$ – 20 patients |
| $\omega_4$ – *H-biopsy negative* | $C_4$ – 16 patients |
| $\omega_5$ – *Hepatitis acuta* | $C_5$ – 9 patients |
| $\omega_6$ – *Hepatitis subacuta* | $C_6$ – 9 patients |
| $\omega_7$ – *HBV-positive* | $C_7$ – 5 patients |
| | *TOTAL*: 814 |

Each patient $O_j$ from the sets $C_k$ has been represented by the feature vector $\mathbf{x}_j = [x_{j1},..,x_{jn}]^T$ of the dimensionality $n = 40$. Numerical results of both laboratory tests ($x_{ji} \in R^1$) as well as the patient symptoms ($x_{ji} \in \{0,1\}$) have been used in computations. The diagnostic maps resulted from the affine (linear) transformation of the 40 - dimensional feature vectors $\mathbf{x}_j$ on the visualizing plane.



The diagnostic map of the system *Hepar* with the below structure :
the upper-left quarter – $C_2$, the upper-right quarter – $C_3 \cup C_5 \cup C_6 \cup C_7$,
the lower-right quarter – $C_1$, the lower-left quarter – $C_4$

Tab. 1: Allocation of the feature vectors $\mathbf{x}_j[40]$ by the $K - NN$ rule with $K$ = 10.

| | Allocation *A* | Allocation *B* | Allocation *C* | Allocation *D* | Success rate (%) |
|---|---|---|---|---|---|
| Class *A* | **2** | 8 | 0 | 33 | 4.7 |
| Class *B* | 0 | **353** | 9 | 20 | 94.6 |
| Class *C* | 0 | 2 | **7** | 7 | 43.6 |
| Class *D* | 4 | 18 | 9 | **357** | 93.4 |
| TOTAL | | | | | **88.3** |

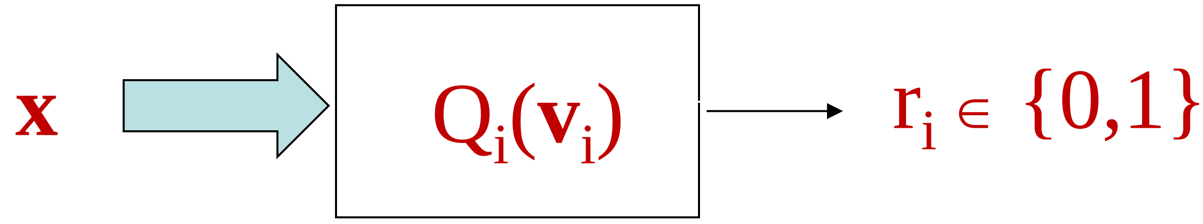
Tab. 2: Allocation of the transformed vectors $\mathbf{y}_j[2]$ on the diagnostic map by the $K - NN$ rule with $K$ = 10.

| | Allocation *A* | Allocation *B* | Allocation *C* | Allocation *D* | Success rate (%) |
|---|---|---|---|---|---|
| Class *A* | **31** | 2 | 1 | 9 | 72.1 |
| Class *B* | 0 | **369** | 1 | 3 | 98.9 |
| Class *C* | 0 | 3 | **11** | 2 | 68.8 |
| Class *D* | 6 | 2 | 3 | **371** | 97.1 |
| TOTAL | | | | | **96.0** |

# VIII. Linearization of the learning sets by ranked layers of binary classifiers

# BINARY CLASSIFIERS $Q_i(\mathbf{v}_i)$

## $(i = 1,\ldots\ldots,L)$




$\mathbf{x} = [x_1,\ldots, x_n]^T$ **-** the *input vector*

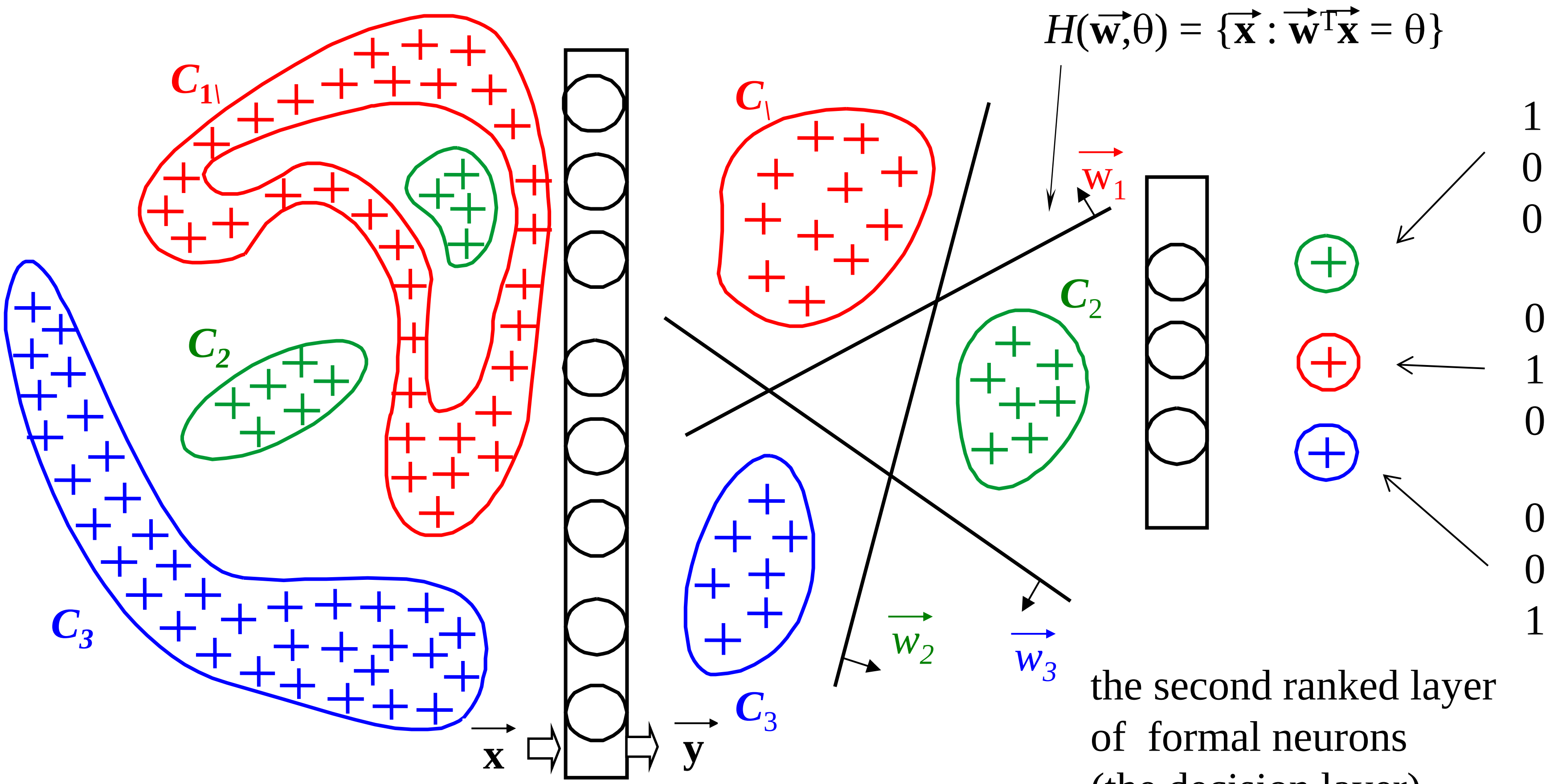
$r_i = r_i(\mathbf{v}_i;\mathbf{x})$ – the *binary output* $(r_i \in \{0,1\})$,

where $r_i(\mathbf{v}_i;\mathbf{x})$ is the *activation function*

$\mathbf{v} = [v_1,\ldots., v_{n'}]^T$ – *vector of parameters* $v_i$

$S_i' = \{\mathbf{x}: r_i(\mathbf{v}_i;\mathbf{x}) = 1\}$ – *activation field*

# RANKED LAYERS OF BINARY CLASSIFIERS



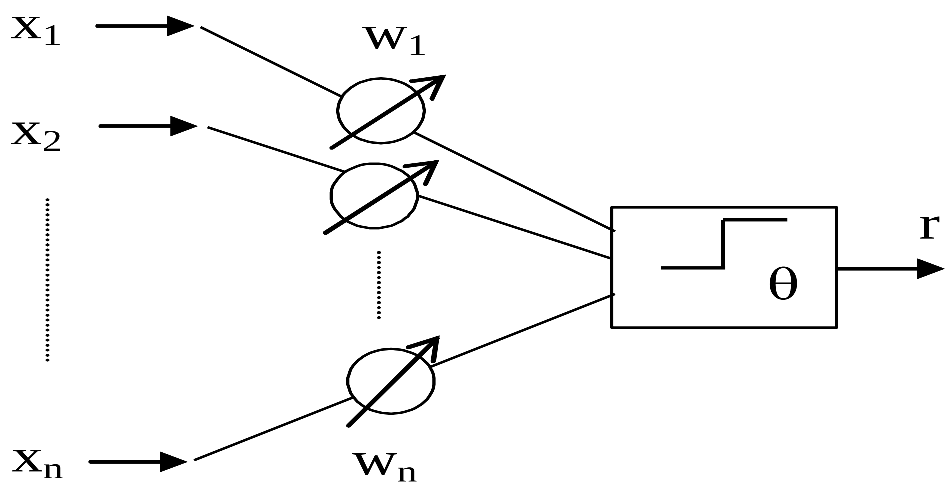

$\vec{y} = F(\vec{x})$

the first ranked layer of
binary classifiers

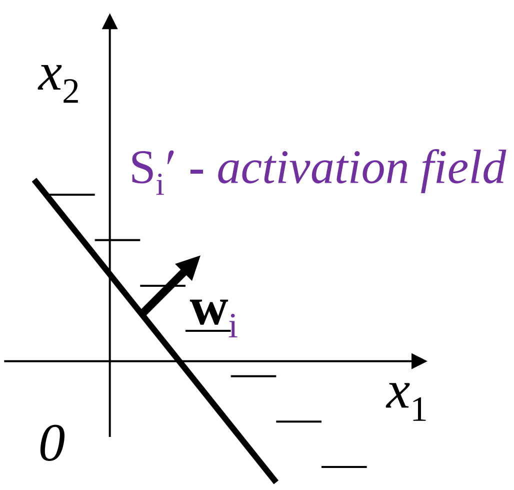
the second ranked layer
of formal neurons
(the decision layer)

# BINARY CLASSIFIERS

## Example 1: *Formal neurons NF*($\mathbf{w}_i$,$\theta_i$)




$$r = r(\mathbf{w},\theta;\mathbf{x}) = \begin{cases} 1 & \text{if} \quad \mathbf{w}^T\mathbf{x} \geq \theta \\ 0 & \text{if} \quad \mathbf{w}^T\mathbf{x} < \theta \end{cases}$$




where:

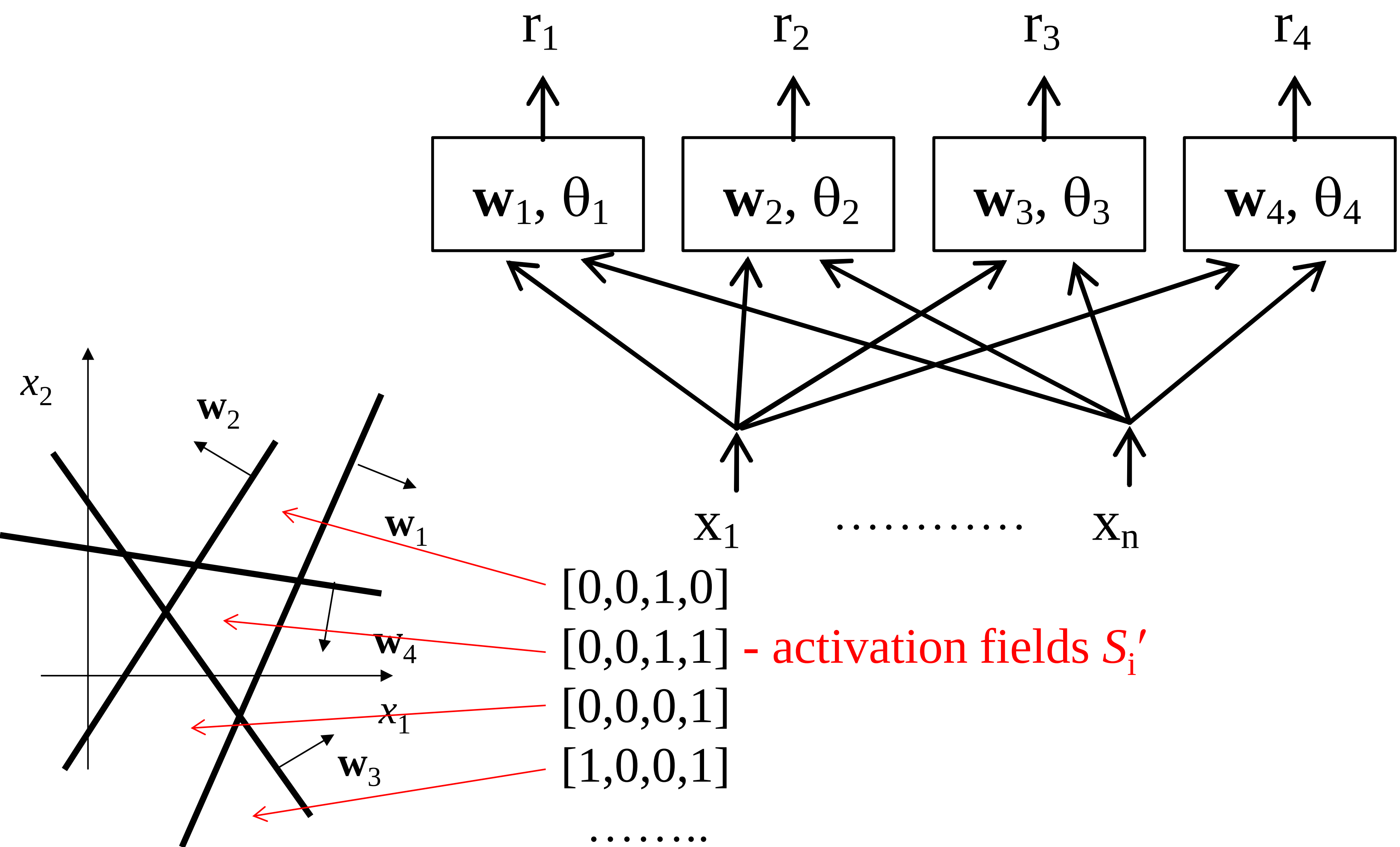
$\mathbf{x} = [x_1,....,x_n]^T$ - input (feature) vector

$\mathbf{w} = [w_1,....,w_n]^T$ - weigth vector ($\mathbf{w} \in R^n$)

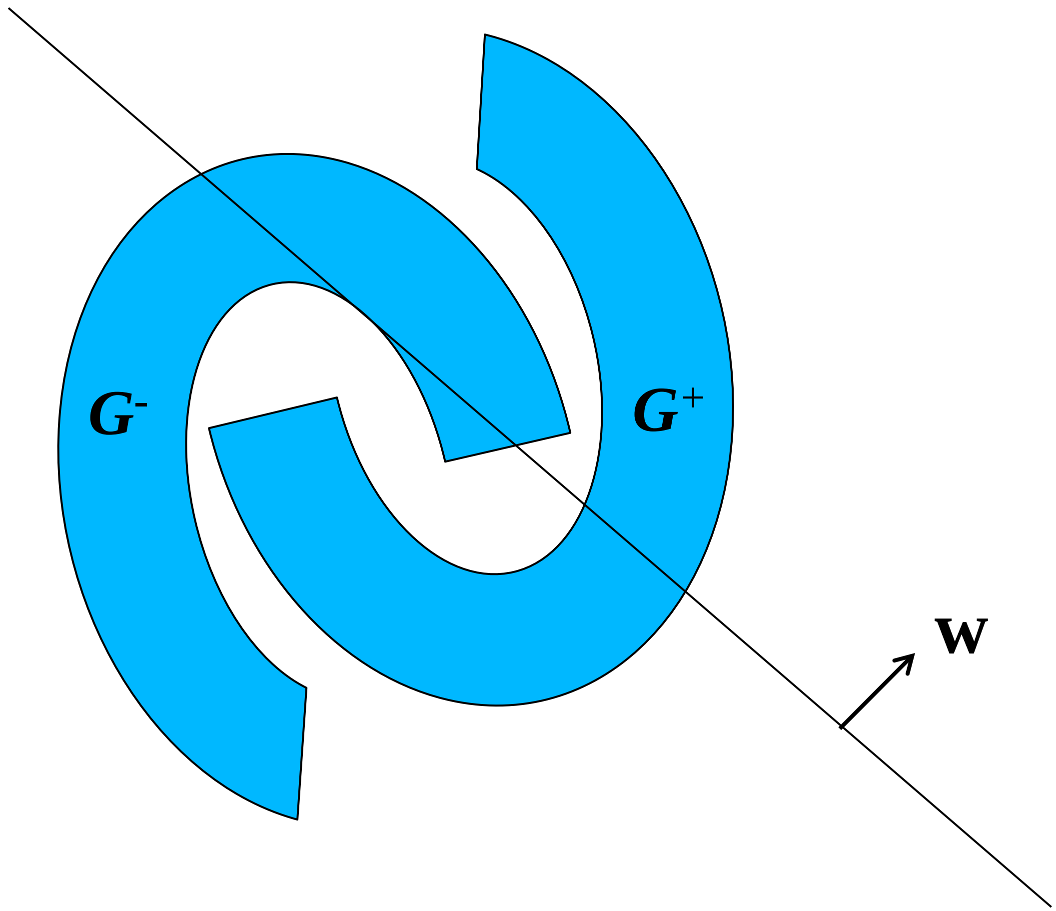
$\theta$ - threshold ($\theta \in R^1$)

# LAYERS OF BINARY CLASSIFIERS

## Example: *The layer of four formal neurons NF*($\mathbf{w}_i$,$\theta_i$)



# Learning sets $G^+$ and $G^-$ which are not *linearly separable*



# INDUCTION OF LINEAR SEPARABILTY BY A ***RANKED*** LAYER OF BINARY CLASSIFIERS $Q_i(\mathbf{v}_i)$

The $k$-th *transformed set* $D_k$ is obtained in result of the transformation of all feature vectors $\mathbf{x}_j(k)$ from the $k$-th learning set $C_k$:

$$D_k = \{\mathbf{r}_j(k) \colon (\forall j \in J_k) \ \ \mathbf{r}_j(k) = \mathbf{r}(\mathbf{V};\mathbf{x}_j(k))\}$$

*Theorem*: Transformation of feature vectors $\mathbf{x}_j(k)$ by a such layer of $L$ binary classifiers $Q_i(\mathbf{v}_i)$ which is ***ranked*** in respect to the separable learning sets $C_k$ results in linear separability of the transformed sets $D_k$:

$$(\forall k \in \{1,\ldots,K\}) \quad (\exists \mathbf{v}_k \in R^L) \ \ (\forall \mathbf{r}_j(k) \in D_k) \ \ \mathbf{v}_k^T \mathbf{r}_j(\mathrm{k}) > 0.$$
$$\textit{and} \ \ (\forall \mathbf{r}_j(i) \in D_i,\ i \neq k) \ \ \mathbf{v}_k^T \mathbf{r}_j(i) < 0$$

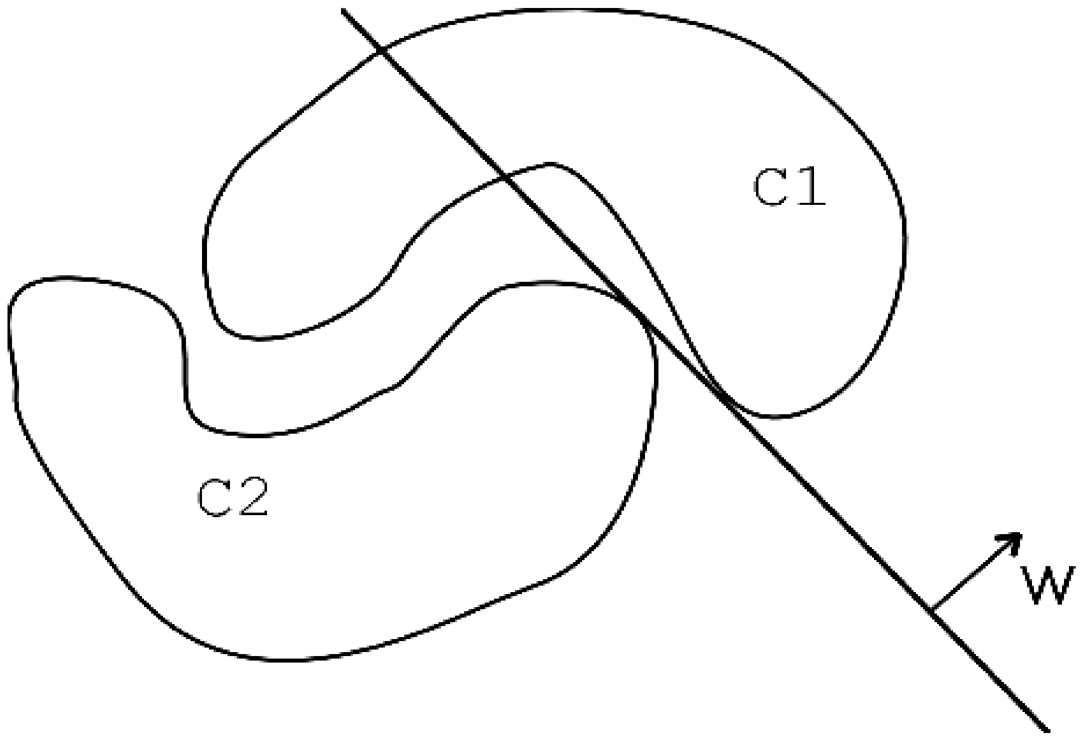
Ranked layer ***induces linear separability*** with the threshold $\theta_k$ equal to zero ($\theta_k = 0$) of the transformed learning sets $C_k$.

Ranked layer can be designed in result of sequence of ***admissible cuts*** of the learning sets $C_k$.

# RANKED LAYERS OF FORMAL NEURONS

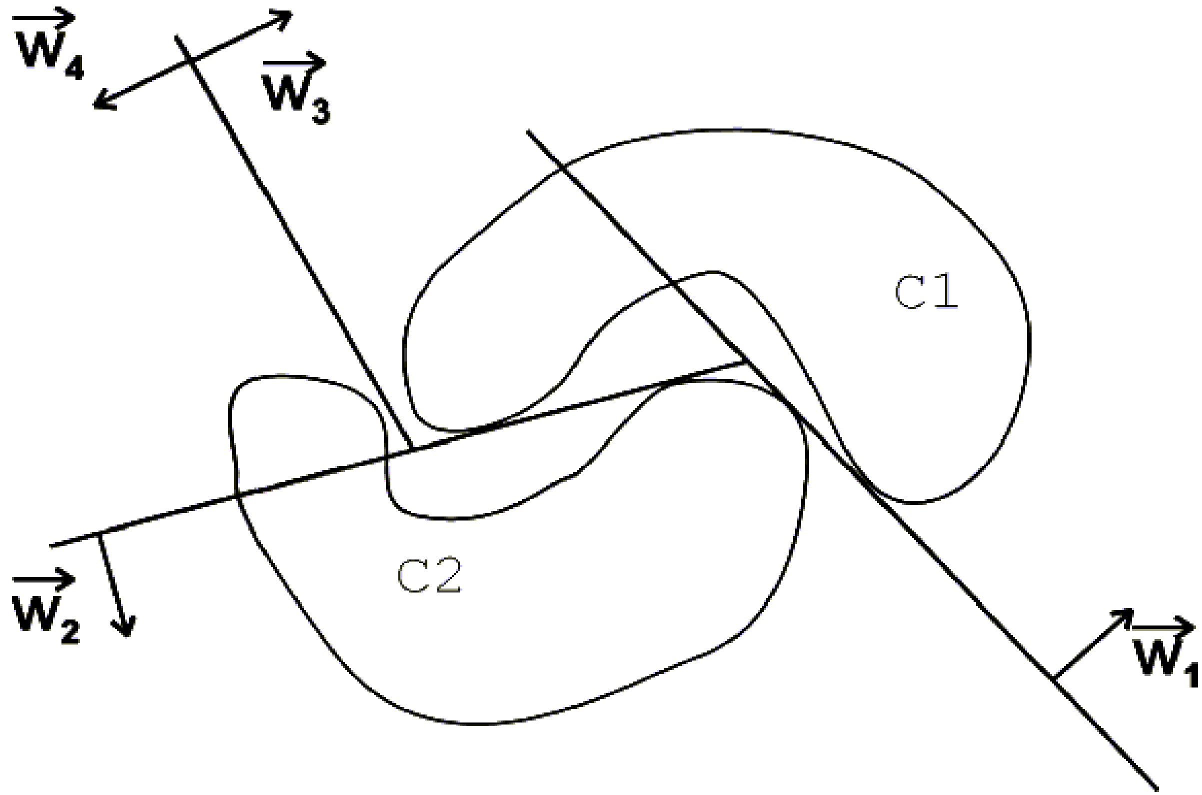
An example of an ***admissible cut*** by the hyperplane

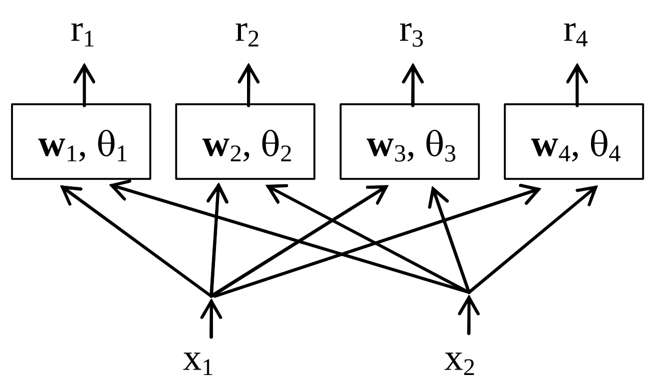
$H(\mathbf{w},\theta) = \{\mathbf{x}: \mathbf{w}^T\mathbf{x} = \theta\}$ (formal neuron $FN(\mathbf{w},\theta)$).




L. Bobrowski, *"Design of piecewise linear classifiers from formal neurons by some basis exchange technique"*
*Pattern Recognition*, **24**(9), pp. 863-870, 1991

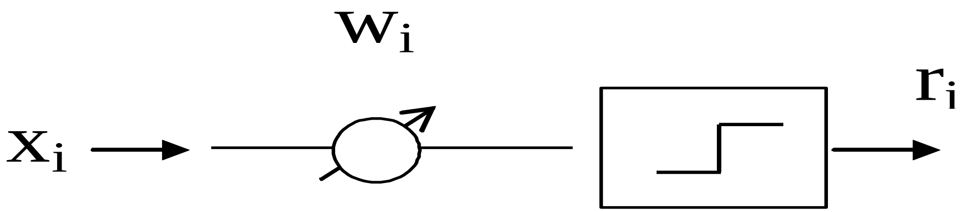
# RANKED LAYERS OF FORMAL NEURONS (II)



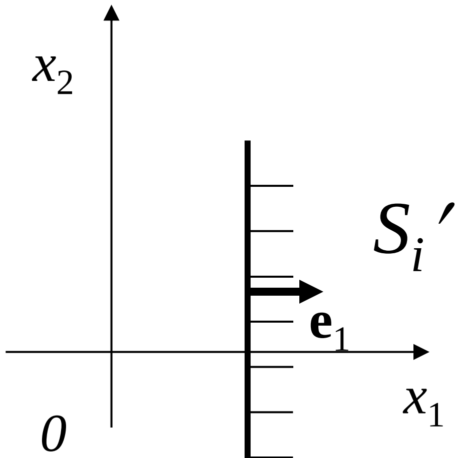
# BINARY CLASSIFIERS

## Example 2: *Logical elements LE*$(w_i, \theta_i)$




$$r = r(w_i, \theta_i; x_i) = \begin{cases} 1 & \textit{if} \quad w_i x_i \geq \theta_i \\ 0 & \textit{if} \quad w_i x_i < \theta_i \end{cases}$$

where $x_i$ is the i-th component of the feature vector $\mathbf{x}$

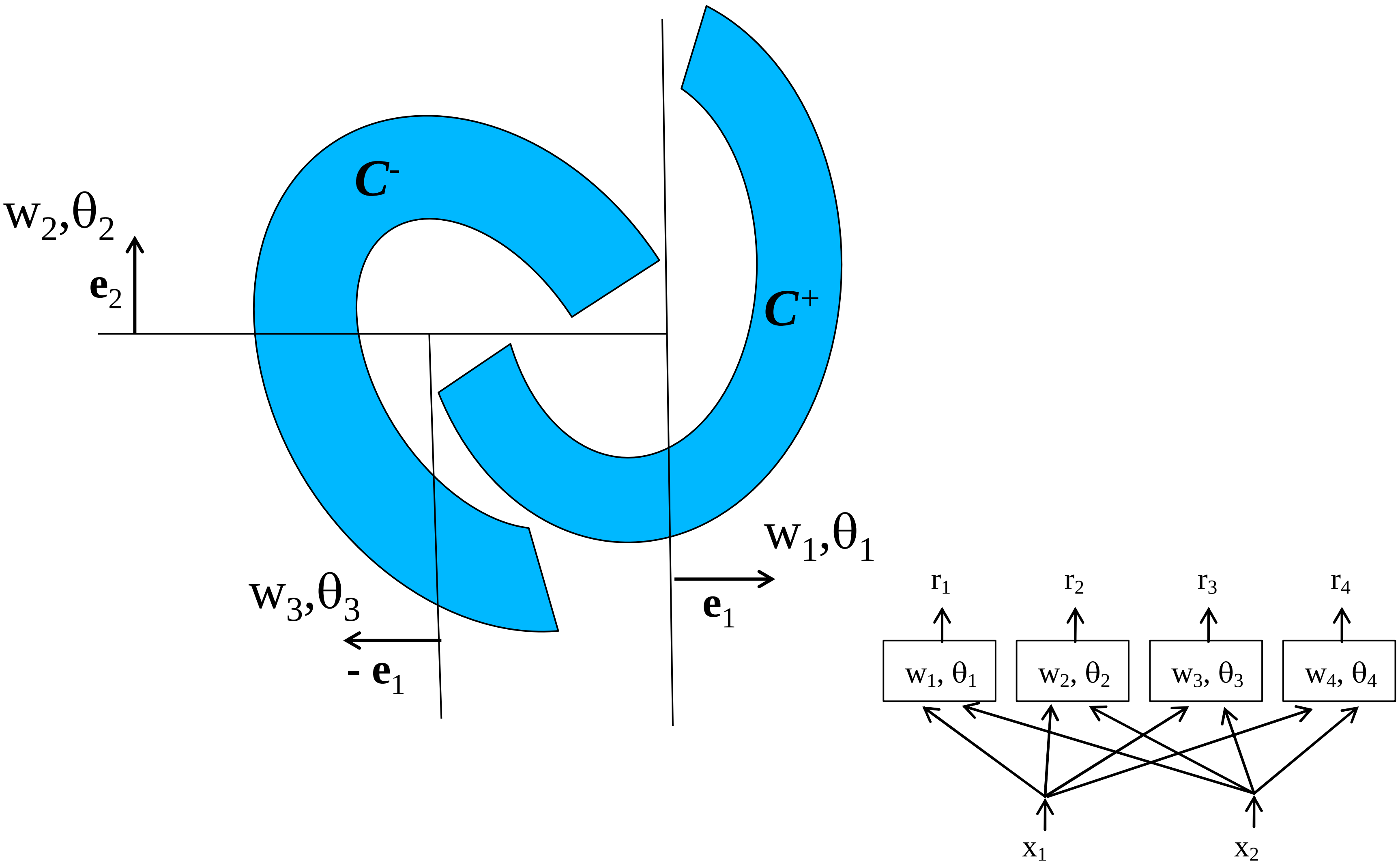



***Logical rules***

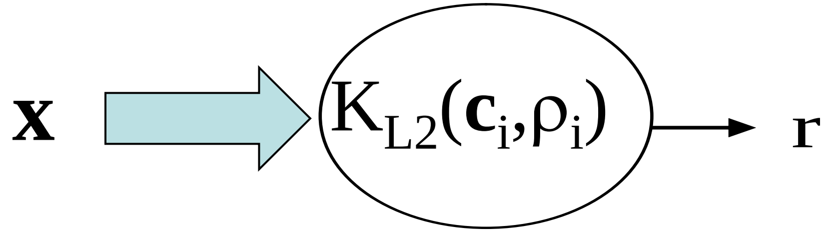
*I.* ***if*** $(x_i \geq a_i)$ ***then*** $r_i = 1$, ***else*** $r_i = 0$, *or*

*II.* ***if*** $(x_i < a_i)$ ***then*** $r_i = 1$, ***else*** $r_i = 0$

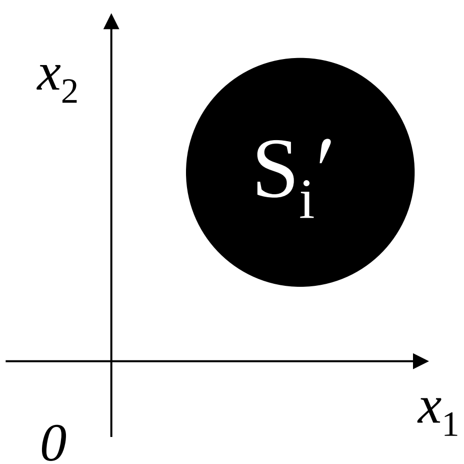
# RANKED LAYERS OF LOGICAL ELEMENTS $LE(w_i,\theta_i)$



# BINARY CLASSIFIERS
## Example 3: *Radial neurons* $RN(\mathbf{c}_i, \rho_i)$




$$r = r(\mathbf{c},\rho;\mathbf{x}) = \begin{cases} 1 & \textit{if} \quad (\mathbf{x}-\mathbf{c})^T(\mathbf{x}-\mathbf{c}) \leq \rho \\ 0 & \textit{if} \quad (\mathbf{x}-\mathbf{c})^T(\mathbf{x}-\mathbf{c}) > \rho \end{cases}$$



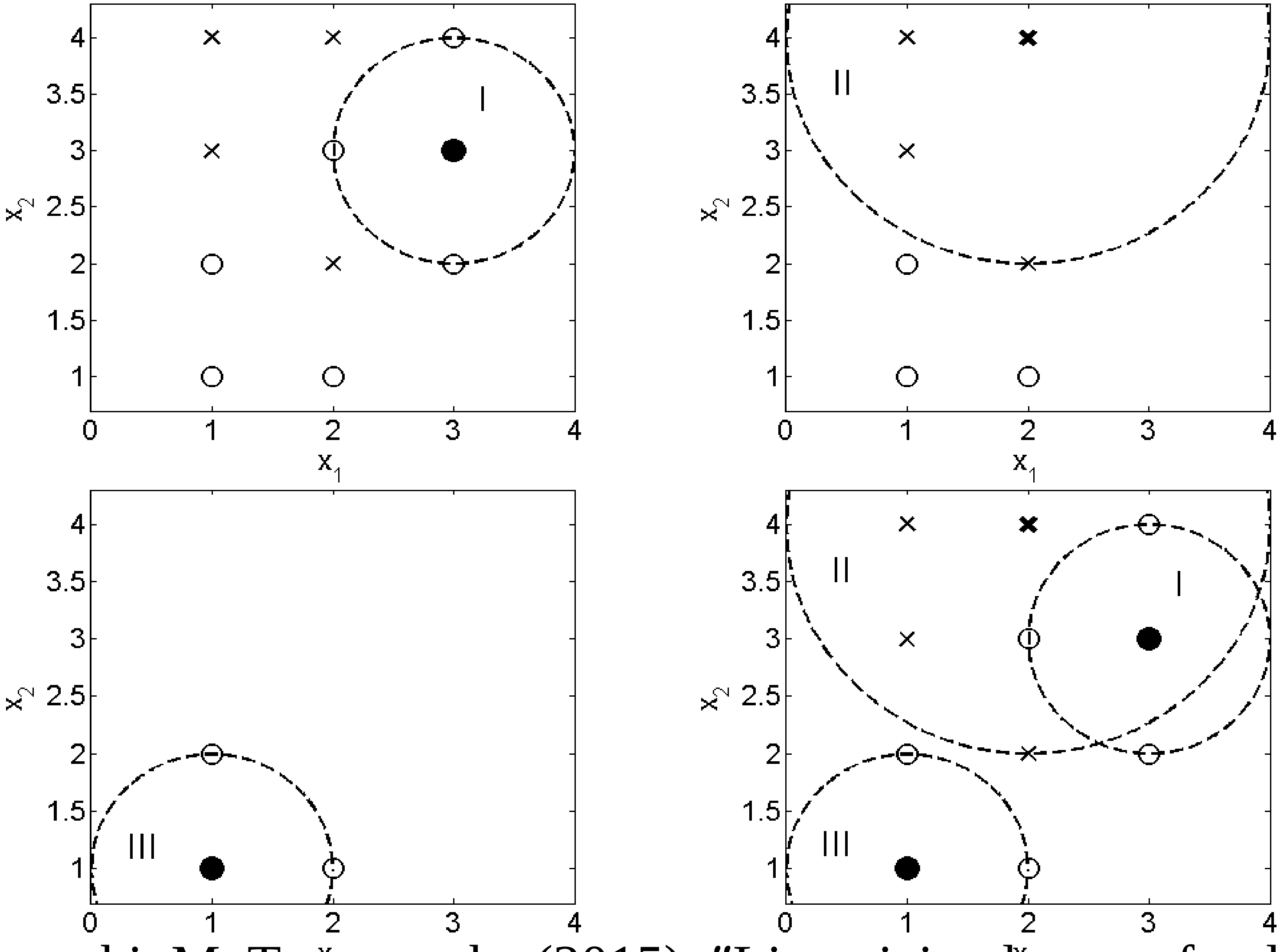

where:

$\mathbf{x} = [x_1,\ldots,x_n]^T$ - feature vector

$\mathbf{c} = [c_1,\ldots,c_n]^T$ - ball center

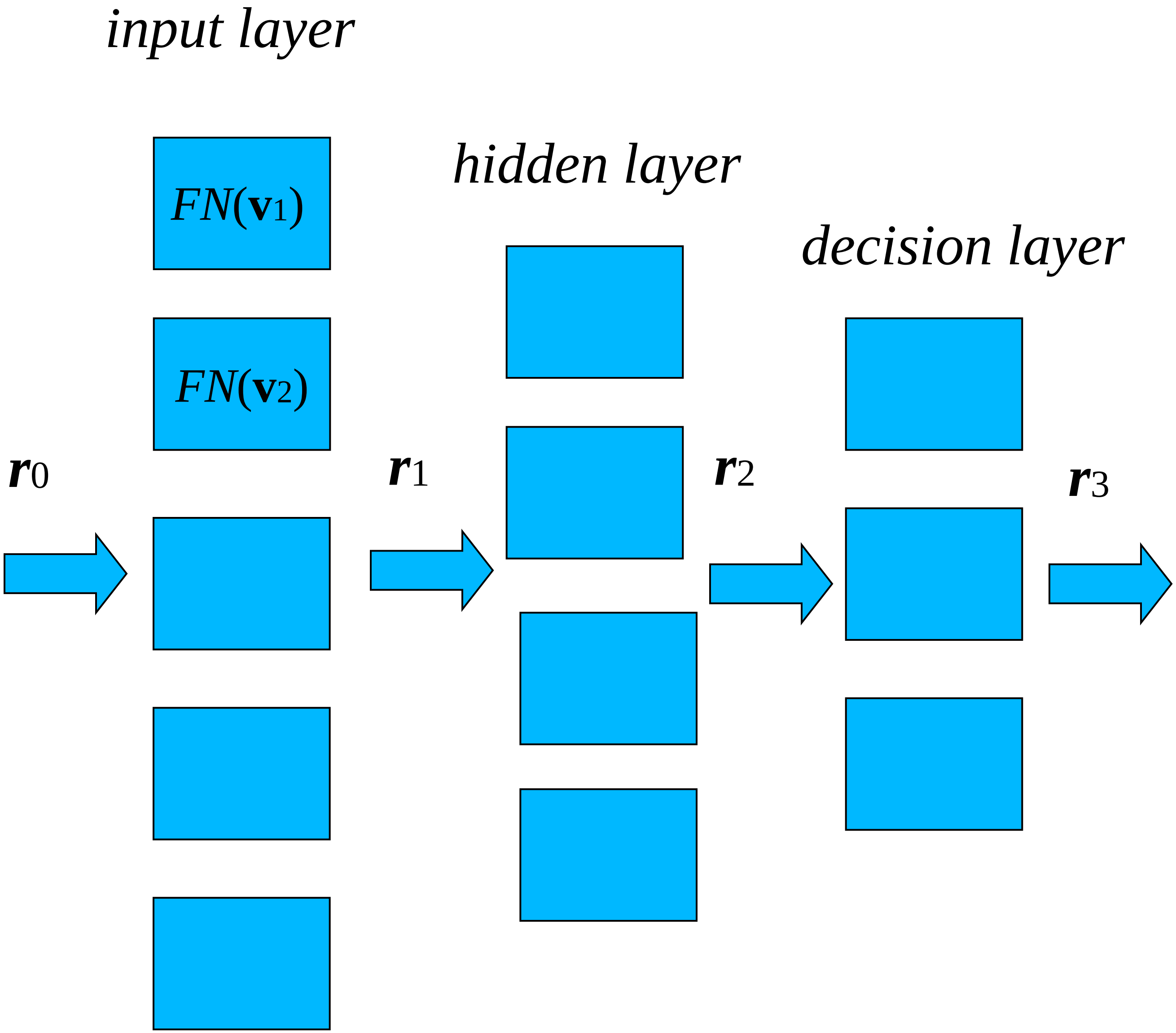
$\rho$ - ball radius

# Toy example data set



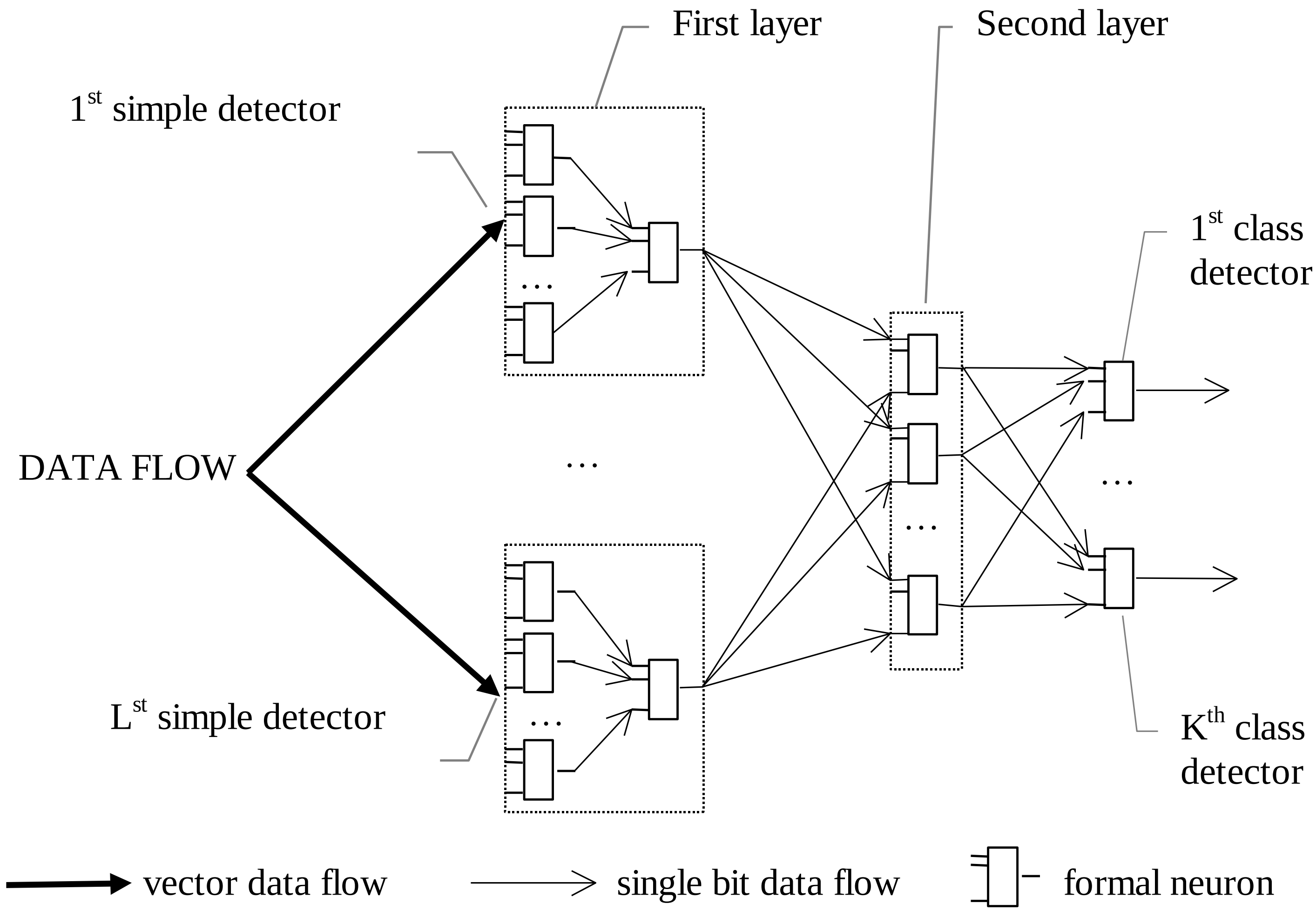

L. Bobrowski, M. Topczewska (2015), ″Linearizing layers of radial binary classifiers with movable centers ″,
 pp. 771 – 78 in: ***Pattern Anal Applic***, *18(4), 2015*

# *Designing hierarchical networks of binary classifiers*



input layer
hidden layer
decision layer
$FN(\mathbf{v}_1)$
$FN(\mathbf{v}_2)$
$\boldsymbol{r}_0$
$\boldsymbol{r}_1$
$\boldsymbol{r}_2$
$\boldsymbol{r}_3$

# HIERARCHICAL NEURAL NETWORKS



# ***Designing hierarchical networks of binary classifiers***

*Problem* 1: Choice of the network architecture.

How to fit network architecture to the problem?

How many layers should be in the network?

Which and how much should be the binary classifiers in each layer?

- the method of trial and error
- the ranked strategy or the dipolar strategy

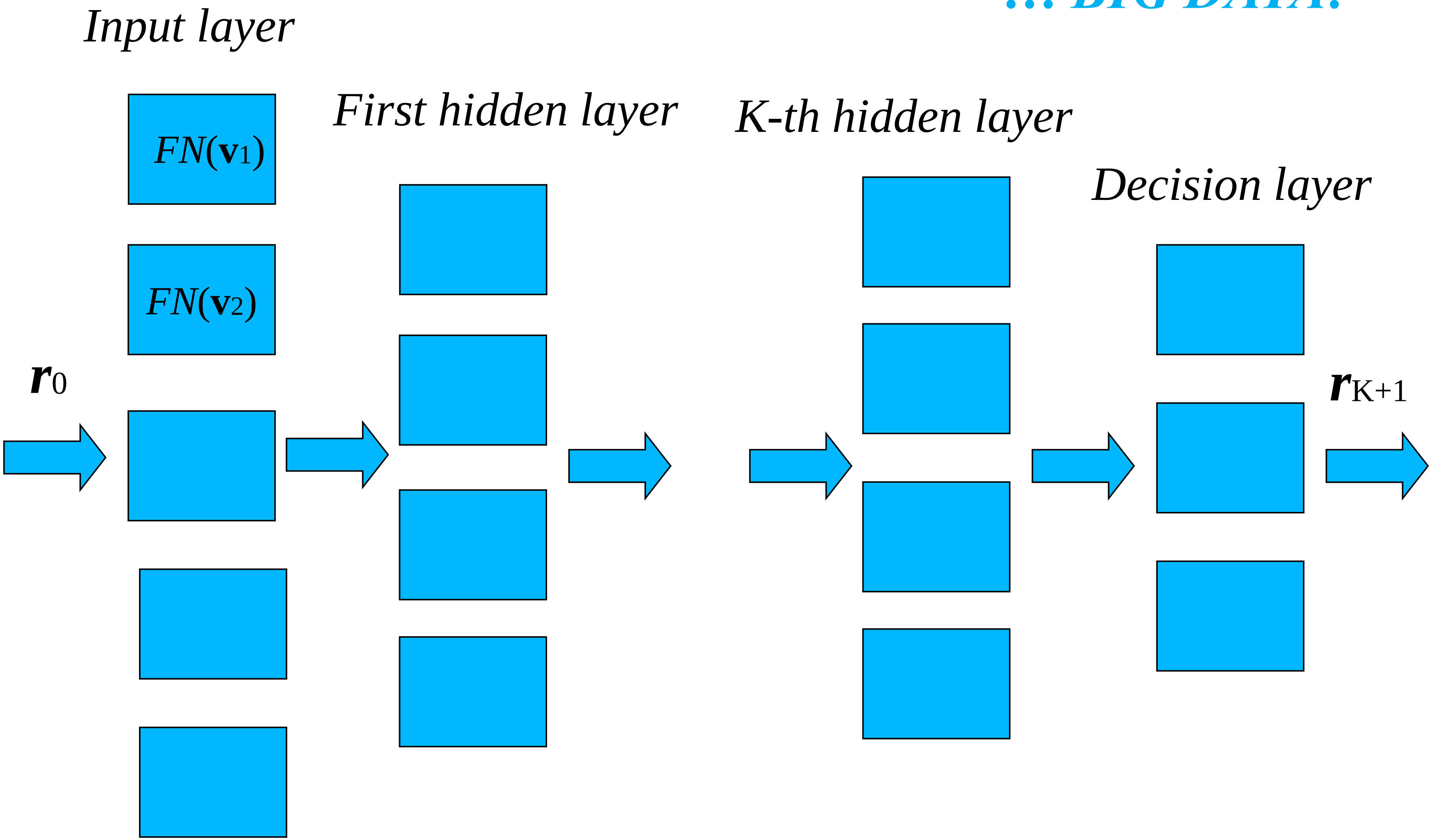
*Problem* 2: Choice of network parameters.

- *back-propagation* algorithm
- a modified error-correction algorithm with a fixed *decomposition rule* $s_k(\omega[n])$ of the teacher′s decision $\omega[n]$ aimed at the $k$-th element of the network during the $n$-th learning step ($\forall n = 1,2,\ldots$), where $\omega[n] \in \{1,\ldots,K\}$ and $s_k(\omega[n]) \in \{1, 0\}$
- minimization of convex and piecewise linear (CPL) criterion functions

# Deep Learning

*… moving beyond machine learning since 2006!*

*… BIG DATA!*



# References

Johnson, R. A., Wichern, D. W.: *Applied Multivariate Statistical Analysis*, Prentice-Hall, Inc., Englewood Cliffs, New York (1991)

Duda, O. R., Hart, P. E., and Stork, D. G.: *Pattern Classification*, J. Wiley, New York (2001)

Hand D., Smyth P. and Mannila H.: *Principles of data mining*, MIT Press, Cambridge (2001)

Bobrowski L.: *Eksploracja danych oparta na wypukłych i odcinkowo-liniowych funkcjach kryterialnych*, (*Data mining based on convex and piecewise linear* (*CPL*) *criterion functions*) (in Polish), Białystok University of Technology, 2005

Bobrowski L. and Łukaszuk T.: Relaxed linear separability (RLS) approach to feature (gene) subset selection, *Selected Works in Bioinformatics*, Xuhua Xia (Ed.), *INTECH* 2011, pp.103-118

Bobrowski L., Łukaszuk T.: Prognostic Modeling with High Dimensional and Censored Data, pp. 178 – 193 in: *Advances in Data Mining*, P. Perner (Ed.), Springer, Berlin 2012

# Thank you for your attention