

LEARNING & LINEAR CLASSIFIERS

V. Hlaváč

Czech Technical University, Faculty of Electrical Engineering
Department of Cybernetics, Center for Machine Perception
121 35 Praha 2, Karlovo nám. 13, Czech Republic
hlavac@fel.cvut.cz, <http://cmp.felk.cvut.cz>

LECTURE PLAN

- ◆ A classifier, linear classifier.
- ◆ Learning in pattern recognition.
- ◆ Perceptron algorithm.
- ◆ Optimal separating plane with the Kozinec algorithm.

CLASSIFIER

Analyzed object is represented by

X – space of observations

K – set of hidden states

Aim of the classification is to determine a relation between X and K , i.e. to find a function $f: X \rightarrow K$.

Classifier $q: X \rightarrow J$ maps observations $X^n \rightarrow$ set of class indices J , $J = 1, \dots, |K|$.

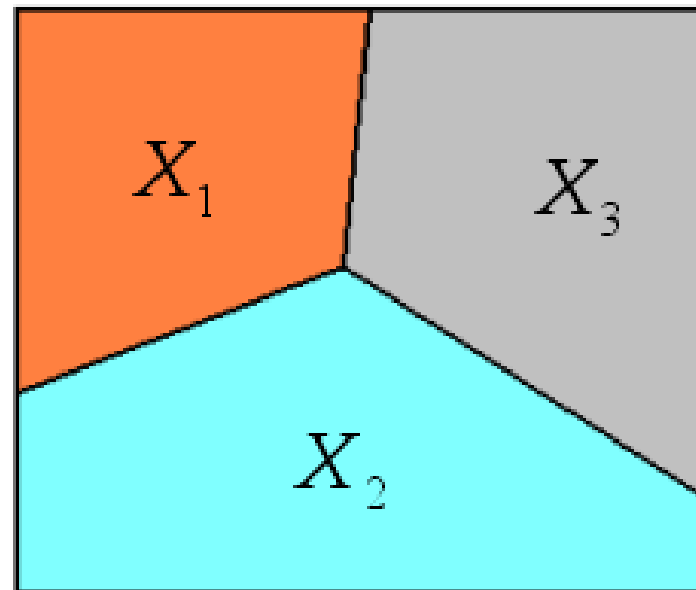
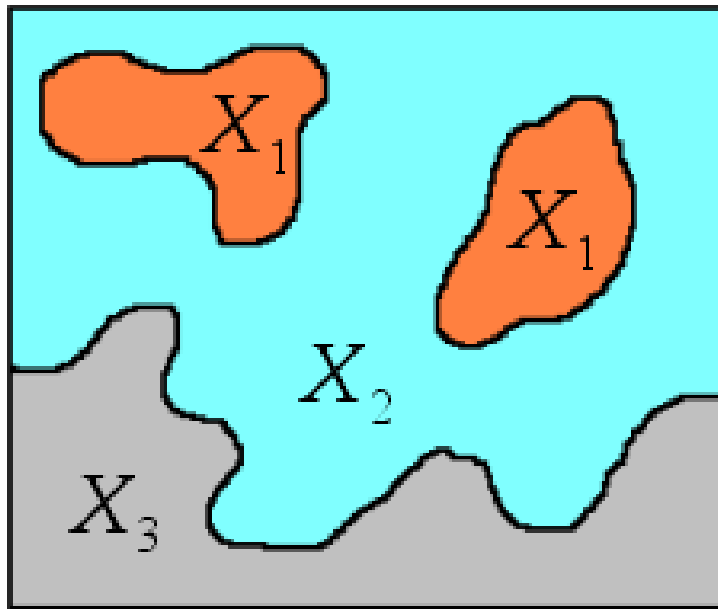
Mutual exclusion of classes

$$X = X_1 \cup X_2 \cup \dots \cup X_{|K|},$$

$$X_i \cap X_j = \emptyset, i \neq j, i, j = 1 \dots |K|.$$

CLASSIFIER, ILLUSTRATION

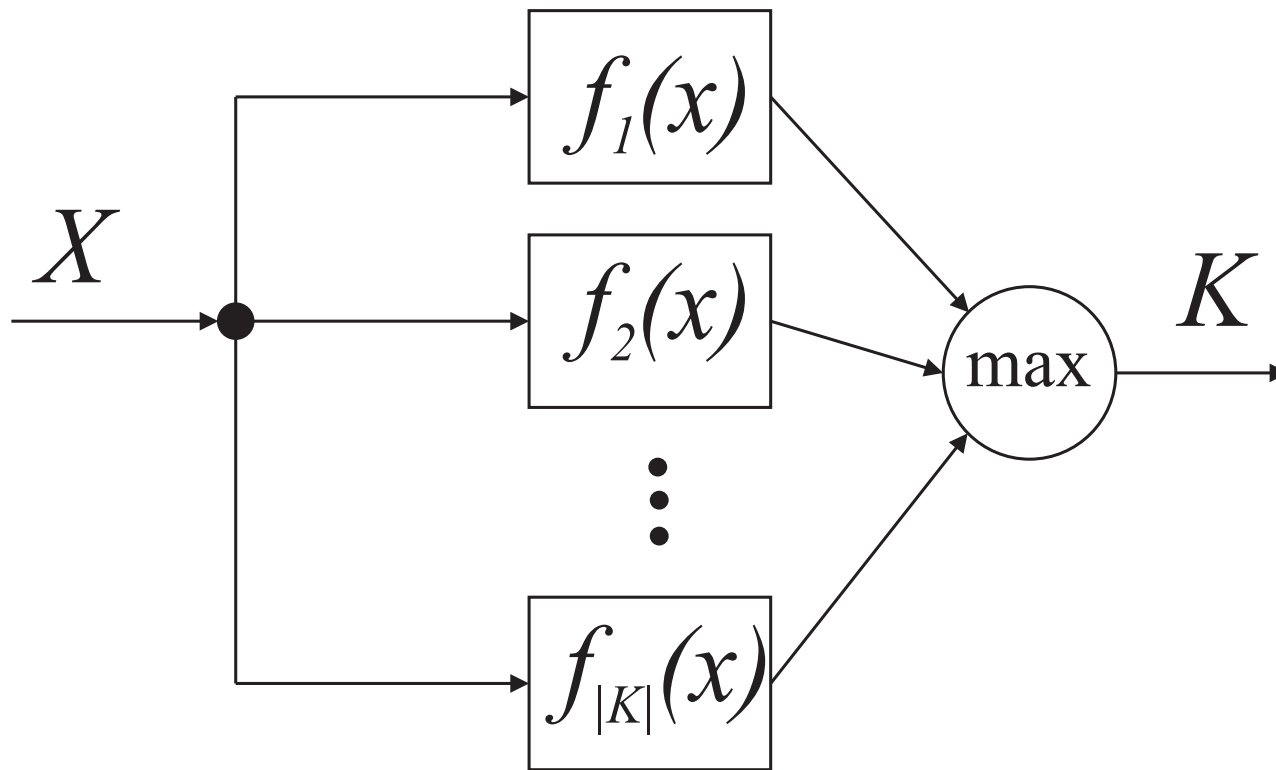
- ◆ A classifier partitions observation space X into class-labelled regions.
- ◆ Classification determines to which region an observation vector x belongs.
- ◆ Borders between regions are called decision boundaries.



RECOGNITION (DECISION) STRATEGY

Discriminant functions $f_i(x)$ should have the property:

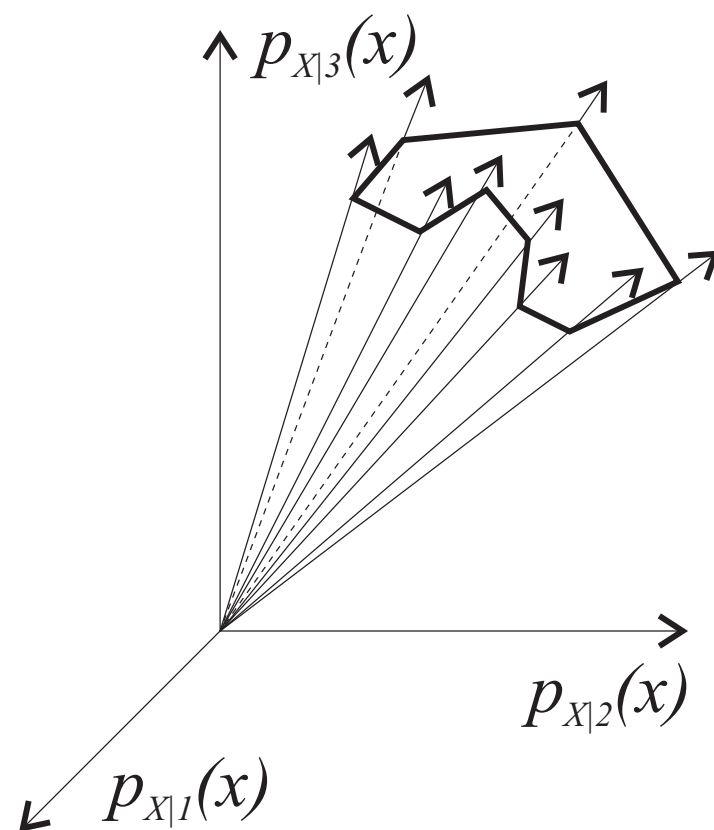
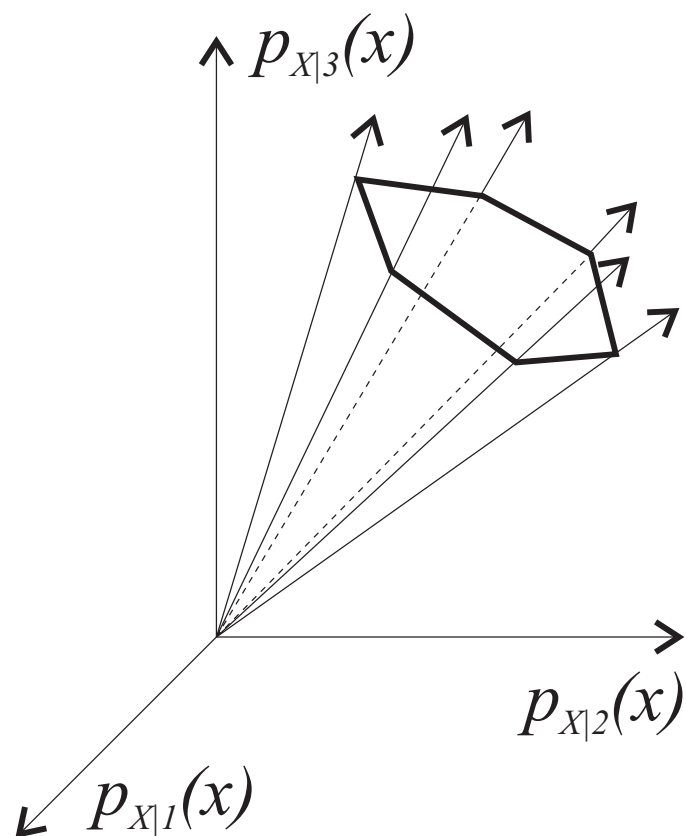
$$f_i(x) > f_j(x) \text{ for } x \in \text{class } i, i \neq j.$$



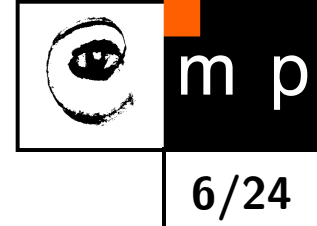
Strategy $j = \underset{j}{\operatorname{argmax}} f_j(x)$

WHY ARE LINEAR CLASSIFIERS IMPORTANT? (1)

Theoretical importance – Bayesian decision rule decomposes the space of probabilities into convex cones.



WHY ARE LINEAR CLASSIFIERS IMPORTANT? (2)



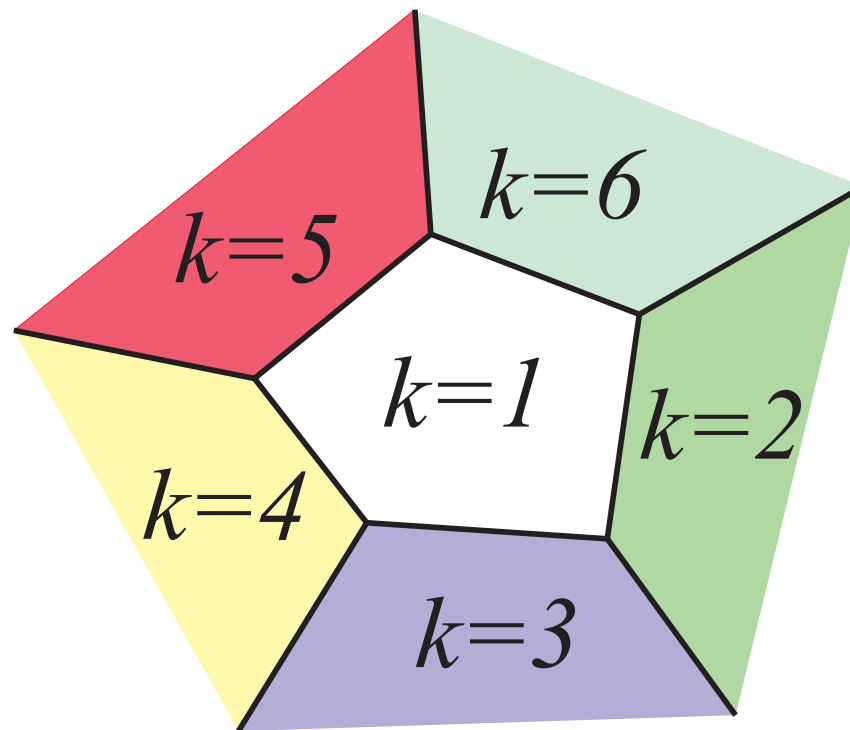
- ◆ For some statistical models, the Bayesian or non-Bayesian strategy is implemented by a linear discriminant function.

You should know an example!?

- ◆ Capacity (VC dimension) of linear strategies in an n -dimensional space is $n + 1$. Thus, the learning task is well-posed, i.e., strategy tuned on a finite training multiset does not differ much from correct strategy found for a statistical model.
- ◆ There are efficient learning algorithms for linear classifiers.
- ◆ Some non-linear discriminant functions can be implemented as linear after the feature space transformation.

LINEAR DISCRIMINANT FUNCTION $q(x)$

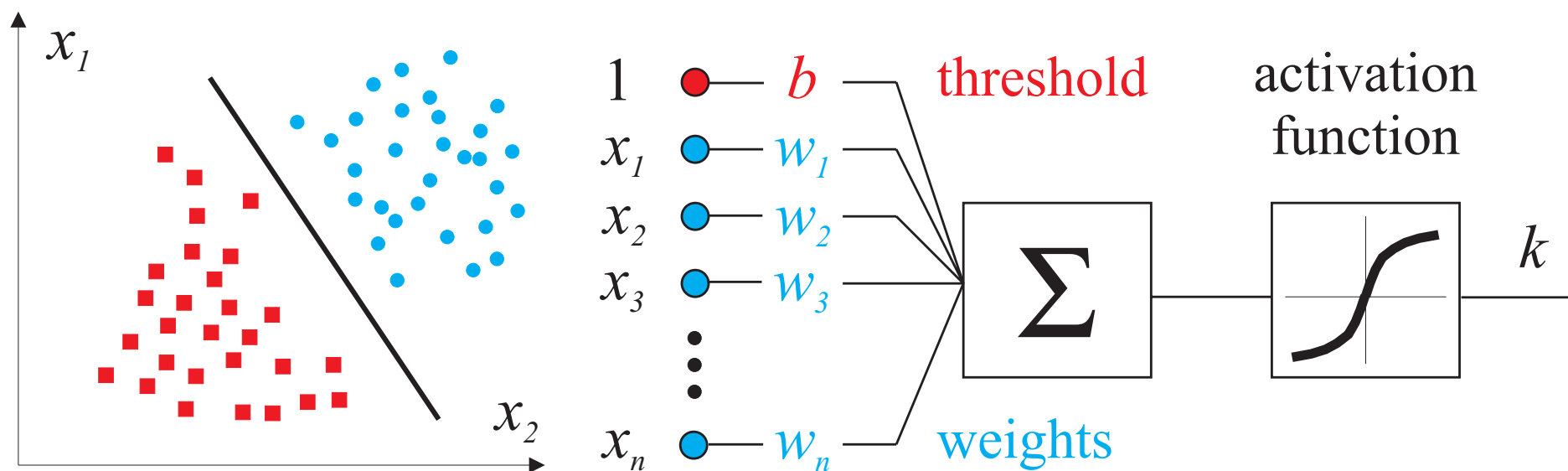
- ◆ $f_j(x) = \langle w_j, x \rangle + b_j$, where $\langle \rangle$ denotes a scalar product.
- ◆ A strategy $j = \operatorname{argmax}_j f_j(x)$ divides X into $|K|$ convex regions.



DICHOTOMY, TWO CLASSES ONLY

$|K| = 2$, i.e. two hidden states (typically also classes)

$$q(x) = \begin{cases} k = 1, & \text{if } \langle w, x \rangle + b \geq 0, \\ k = 2, & \text{if } \langle w, x \rangle + b < 0. \end{cases}$$



Perceptron by F. Rosenblatt 1957

The **aim of learning** is to estimate classifier parameters w_i, b_i for $\forall i$.

The **learning algorithms** differ by

◆ The **character of training set**

1. **Finite set** consisting of individual observations and hidden states, i.e., $\{(x_1, k_1) \dots (x_L, k_L)\}$.
2. **Infinite sets** described by Gaussian distributions.

◆ **Learning task formulations.**

Empirical risk minimization:

- ◆ True risk is approximated by

$$R_{\text{emp}}(q(x, \Theta)) = \frac{1}{L} \sum_{i=1}^L W(q(x_i, \Theta), k_i), \text{ where } W \text{ is a penalty.}$$

- ◆ Learning is based on the empirical minimization principle

$$\Theta^* = \underset{\Theta}{\operatorname{argmin}} R_{\text{emp}}(q(x, \Theta)).$$

- ◆ Examples of learning algorithms: Perceptron, Back-propagation.

Structural risk minimization:

- ◆ True risk is approximated by guaranteed risk (a regularizer securing upper bound of risk is added to empirical risk, Vapnik-Chervonenkis theory of learning).

- ◆ Example: Support Vector Machine (SVM).

PERCEPTRON LEARNING:

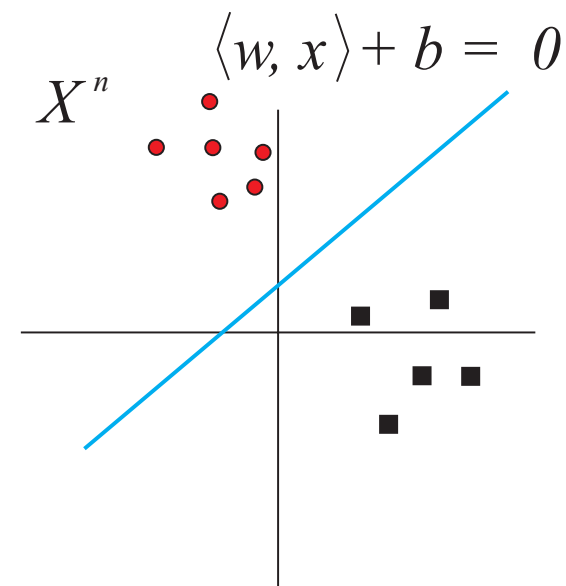
Task formulation

Input: $T = \{(x_1, k_1) \dots (x_L, k_L)\}$, $k_i \in \{1, 2\}$,
 $i = 1, \dots, L$, $\dim(x_i) = n$.

Output: a weight vector w , offset b
 for $\forall j \in \{1, \dots, L\}$ satisfying:

$$\langle w, x_j \rangle + b \geq 0 \text{ for } k = 1,$$

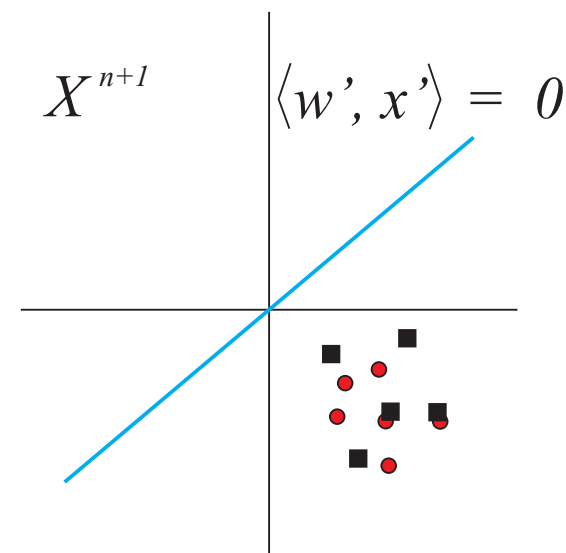
$$\langle w, x_j \rangle + b < 0 \text{ for } k = 2.$$



The task can be formally transcribed to a single inequality $\langle w', x'_j \rangle \geq 0$ by embedding it into $n + 1$ dimensional space, where $w' = [w \quad b]$,

$$x' = \begin{cases} [x & 1] & \text{for } k = 1, \\ -[x & 1] & \text{for } k = 2. \end{cases}$$

We drop the primes and go back to w, x notation.



PERCEPTRON LEARNING: THE ALGORITHM 1957

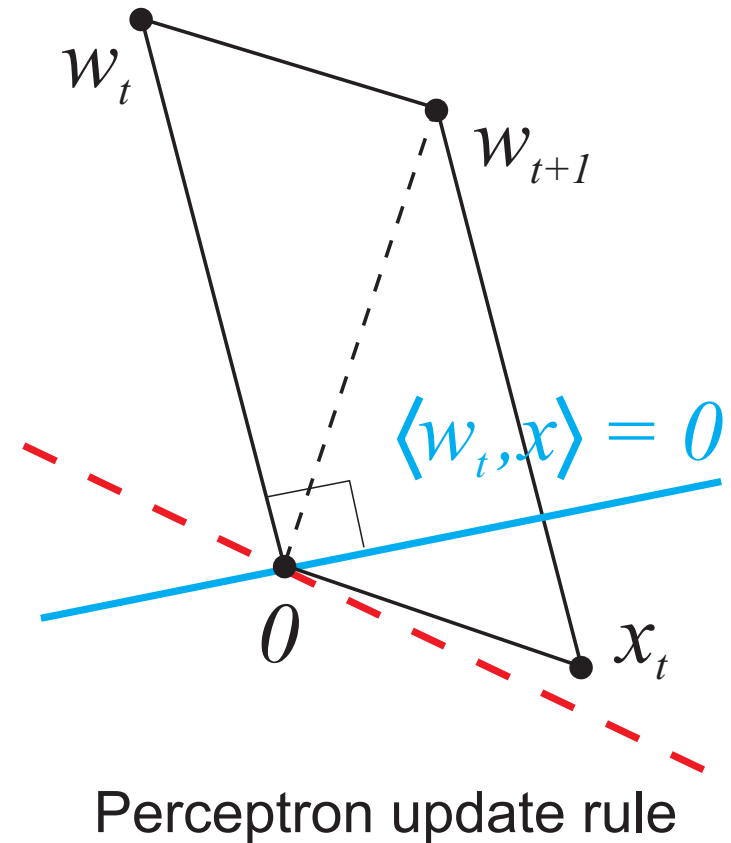
Input: $T = \{x_1, x_2, \dots, x_L\}$.

Output: a weight vector w .

Perceptron algorithm

(F. Rosenblatt):

1. $w_1 = 0$.
2. A wrongly classified observation x_j is sought, i.e., $\langle w_t, x_j \rangle < 0$, $j \in \{1, \dots, L\}$.
3. If there is no misclassified observation then the algorithm terminates otherwise
 $w_{t+1} = w_t + x_j$.
4. Goto 2.



NOVIKOFF THEOREM, 1962

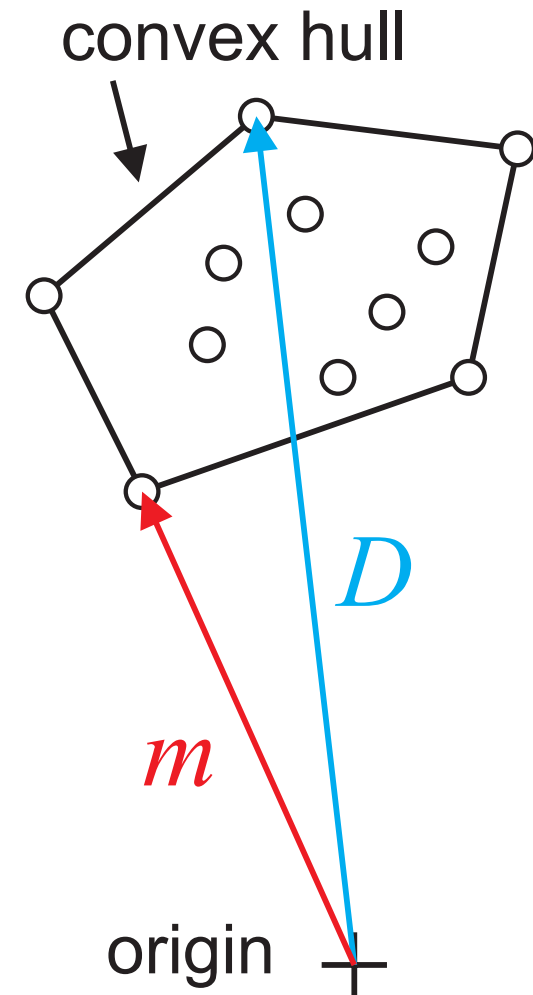
Let \overline{X} denotes the convex hull of points (set of observations) X .

Let $D = \max_i |x_i|$, $m = \min_{x \in \overline{X}} |x_i| > 0$.

Theorem:

If the data are linearly separable then there exists a number $t^* \leq \frac{D^2}{m^2}$, such that the vector w_{t^*} satisfies the inequality

$$\langle w_{t^*}, x_j \rangle > 0, \quad \forall j \in \{1, \dots, L\}.$$



-
- ◆ What if the data is not separable?
 - ◆ How to terminate perceptron learning?

IDEA OF THE NOVIKOFF THEOREM PROOF

Let express bounds for $|w_t|^2$:

Upper bound:

$$\begin{aligned}
 |w_{t+1}|^2 &= |w_t + x_t|^2 = |w_t|^2 + 2 \underbrace{\langle x_t, w_t \rangle}_{\leq 0} + |x_t|^2 \\
 &\leq |w_t|^2 + |x_t|^2 \leq |w_t|^2 + D^2 .
 \end{aligned}$$

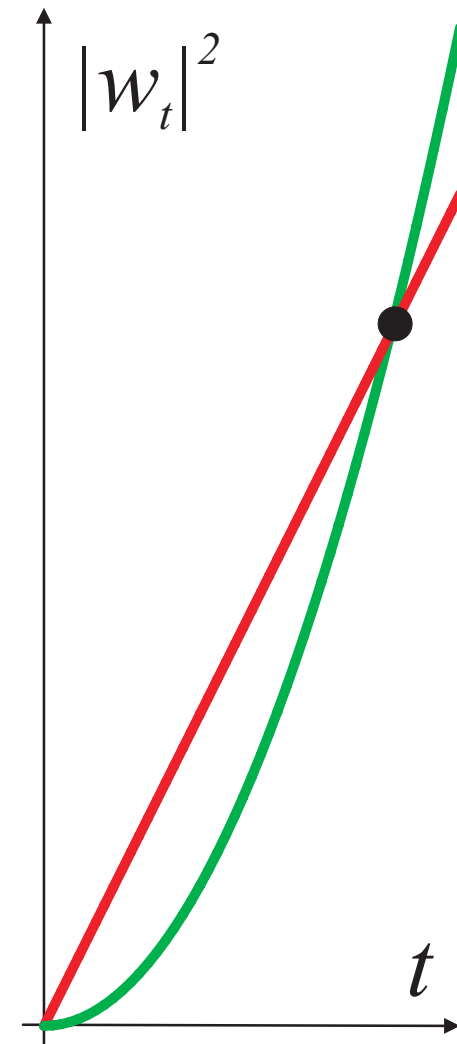
$$|w_0|^2 = 0, |w_1|^2 \leq D^2, |w_2|^2 \leq 2D^2, \dots$$

$$\dots, |w_{t+1}|^2 \leq t D^2, \dots$$

Lower bound: is given analogically

$$|w_{t+1}|^2 > t^2 m^2 .$$

Solution: $t^2 m^2 \leq t D^2 \Rightarrow t \leq \frac{D^2}{m^2} .$



PERCEPTRON LEARNING

as an Optimization problem (1)

Perceptron algorithm, batch version, handling non-separability, another perspective:

- ◆ Input: $T = \{x_1, x_2, \dots, x_L\}$.
- ◆ Output: a weight vector w minimising

$$J(w) = |\{x \in X : \langle w_t, x \rangle < 0\}|$$

or, equivalently

$$J(w) = \sum_{x \in X : \langle w_t, x \rangle < 0} 1.$$

What would the most common optimization method, the [gradient descent](#), perform?

$$w_t = w - \eta \nabla J(w).$$

The gradient of $J(w)$ is either 0 or undefined. Gradient minimization cannot proceed.

PERCEPTRON LEARNING

as an Optimization problem (2)

Let us redefine the cost function:

$$J_p(w) = \sum_{x \in X : \langle w, x \rangle < 0} \langle w, x \rangle .$$

$$\nabla J_p(w) = \frac{\partial J}{\partial w} = \sum_{x \in X : \langle w, x \rangle < 0} x .$$

- ◆ The Perceptron algorithm is a gradient descent method for $J_p(w)$.
- ◆ Learning by the empirical risk minimization is just an instance of an **optimization problem**.
- ◆ Either gradient minimization (backpropagation in neural networks) or convex (quadratic) minimization (called convex programming in mathematical literature) is used.

PERCEPTRON ALGORITHM

Classifier learning, non-separable case, batch version

Input: $T = \{x_1, x_2, \dots, x_L\}$.

Output: a weight vector w^* .

1. $w_1 = 0$, $E = |T| = L$, $w^* = 0$.
2. Find all misclassified observations $X^- = \{x \in X : \langle w_t, x \rangle < 0\}$.
3. if $|X^-| < E$ then $E = |X^-|$; $w^* = w_t$, $t_{lu} = t$.
4. if $tc(w^*, t, t_{lu})$ then terminate else $w_{t+1} = w_t + \eta_t \sum_{x \in X^-} x$.
5. Goto 2.

-
- ◆ The algorithm converges with probability 1 to the optimal solution.
 - ◆ Convergence rate not known (to me).
 - ◆ Termination condition $tc(\cdot)$ is a complex function of the quality of the best solution, time since the last update $t - t_{lu}$ and requirements on the solution.

ALTERNATIVE TRAINING ALGORITHM KOZINEC (1973)

Input: $T = \{x_1, x_2, \dots, x_L\}$.

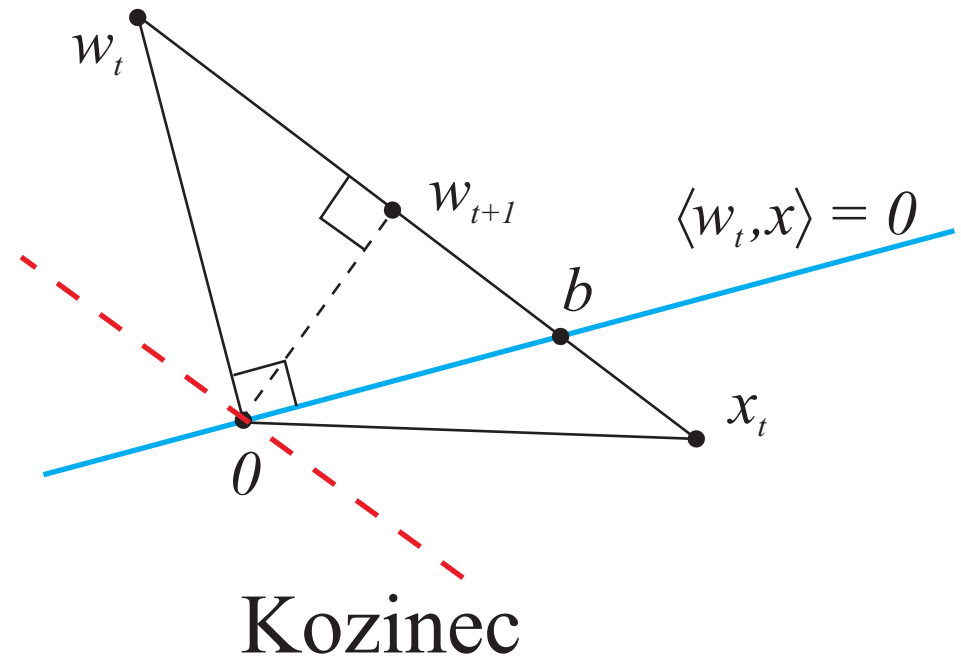
Output: a weight vector w^* .

1. $w_1 = x_j$, i.e., any observation.
2. A wrongly classified observation x_t is sought, i.e., $\langle w_t, x^j \rangle < b$, $j \in J$.
3. If there is no wrongly classified observation then the algorithm finishes otherwise

$$w_{t+1} = (1 - k) \cdot w_t + x_t \cdot k, \quad k \in \mathbb{R}.$$

$$\text{where } k = \underset{k}{\operatorname{argmin}} |(1 - k) \cdot w_t + x_t \cdot k|.$$

4. Goto 2.



OPTIMAL SEPARATING PLANE and THE CLOSEST POINT TO THE CONVEX HULL

The problem of optimal separation by a hyperplane

$$w^* = \operatorname{argmax}_w \min_j \left\langle \frac{w}{|w|}, x_j \right\rangle \quad (1)$$

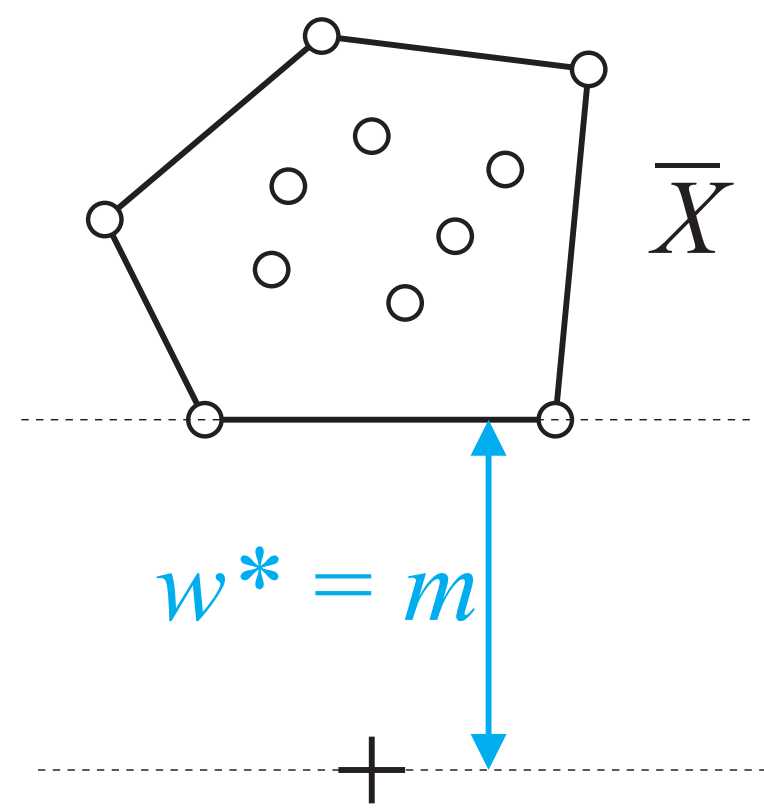
can be converted to seek for the closest point to a convex hull (denoted by the overline)

$$x^* = \operatorname{argmin}_{x \in \overline{X}} |x|.$$

It holds that x^* solves also the problem (1).

Recall that the classifier that maximizes separation minimizes the structural risk R_{str} .

CONVEX HULL, ILLUSTRATION



$$\min_j \left\langle \frac{w}{|w|}, x_j \right\rangle \leq m \leq |w|, w \in \bar{X}.$$

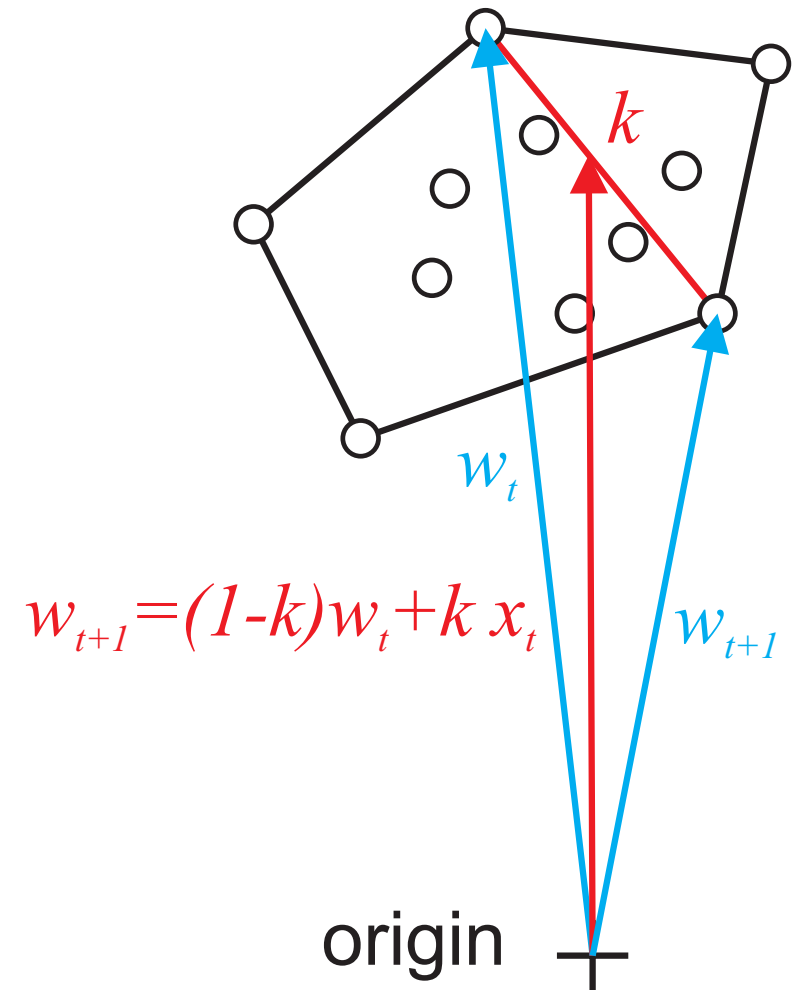
lower bound

upper bound

ϵ -SOLUTION

- ◆ The aim is to speed up the algorithm.
- ◆ The allowed uncertainty ϵ is introduced.

$$|w^t| - \min_j \left\langle \frac{w^t}{|w^t|}, x_j \right\rangle \leq \epsilon$$

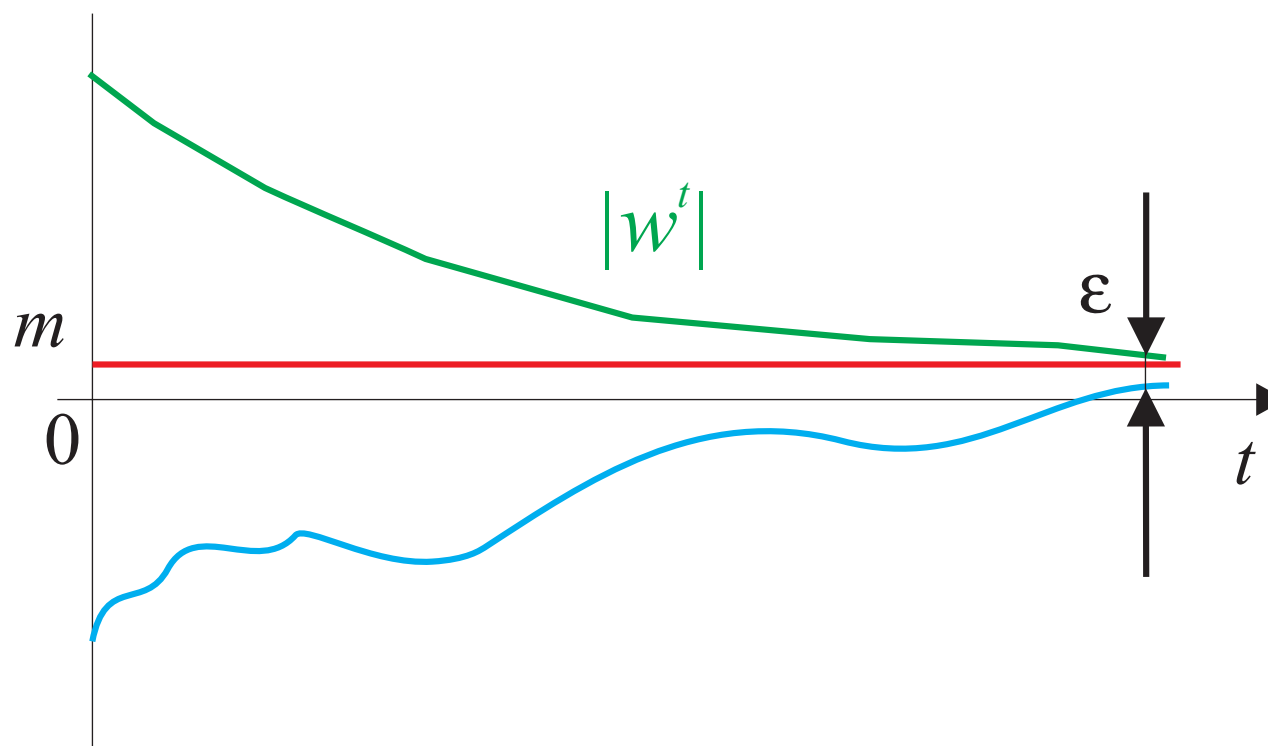


KOZINEC and ε -SOLUTION

The second step of Kozinec algorithm is modified to:

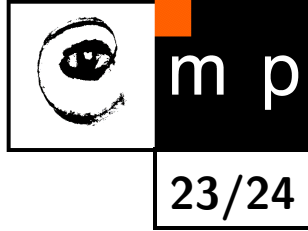
A wrongly classified observation x_t is sought, i.e.,

$$|w^t| - \min_j \left\langle \frac{w^t}{|w^t|}, x_t \right\rangle \geq \varepsilon$$



LEARNING TASK FORMULATION

For infinite training sets

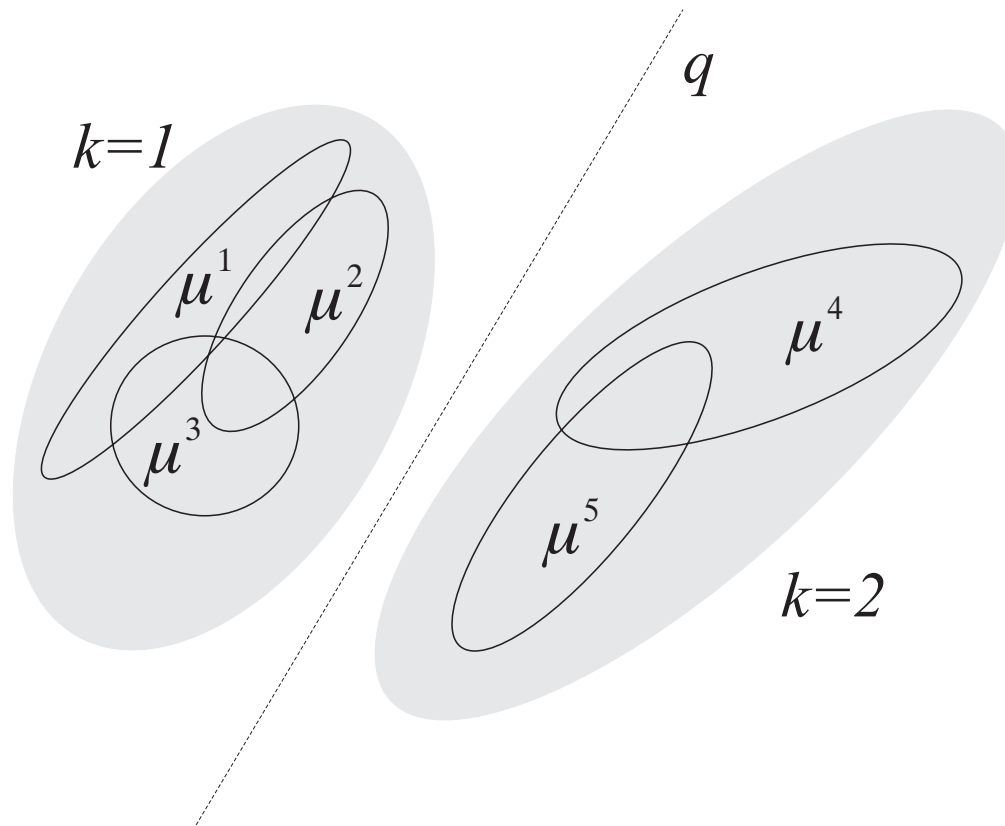


Generalized Anderson task by M.I. Schlesinger (1972) solves a quadratic optimization task.

- ◆ Solves learning problem for a linear classifier and two hidden states only.
- ◆ It is assumed that a class-conditional distribution $p_{X|K}(x | k)$ corresponding to both hidden states are multi-dimensional Gaussian distributions.
- ◆ The mathematical expectation μ_k and the covariance matrix σ_k , $k = 1, 2$, of these probability distributions are not known.
- ◆ The Generalized Anderson task is the extension of Anderson-Bahadur task (1962) which solved the problem for one Gaussian describing each of two classes.

GAndersonT ILLUSTRATED IN 2D SPACE

Illustration of the statistical model, i.e., mixture of Gaussians.



Unknown are weights of the Gaussian components.