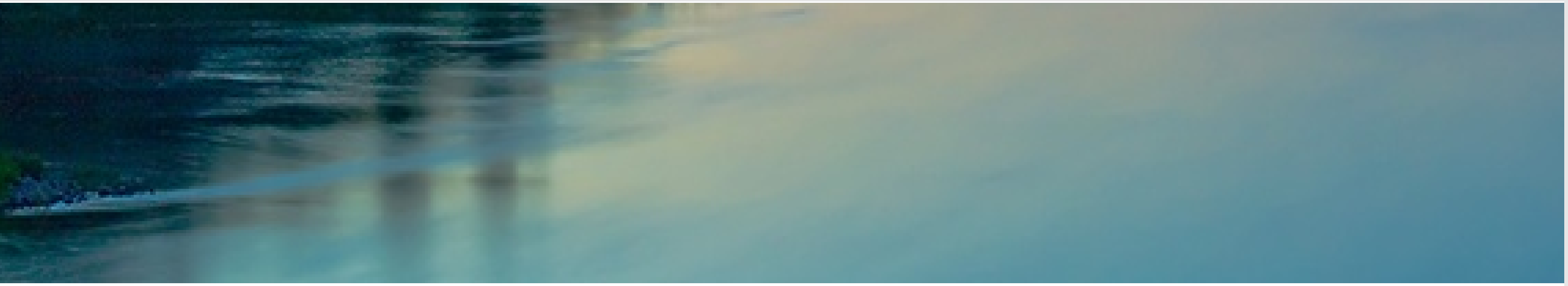


Neural Networks

Prof. Dr. Rudolf Kruse

Computational Intelligence Group
Faculty for Computer Science
kruse@iws.cs.uni-magdeburg.de

Supervised Learning / Support Vector Machines

Training data: Expression profiles of patients with known diagnosis

The known diagnosis gives us a structure within the data, which we want to generalize for future data.

Learning/Training: Derive a decision rule from the training data which separates the two classes.

Ability for generalization: How useful is the decision rule when it comes to diagnosing patients in the future?

Aim: Find a decision rule with high ability for generalization!

Learning from Examples

Given: $X = \{x_i, y_i\}_{i=1}^{n}$, training data of patients with known diagnosis

consists of:

$$x_i \in \mathbb{R}^g \text{ (points, expression profiles)}$$

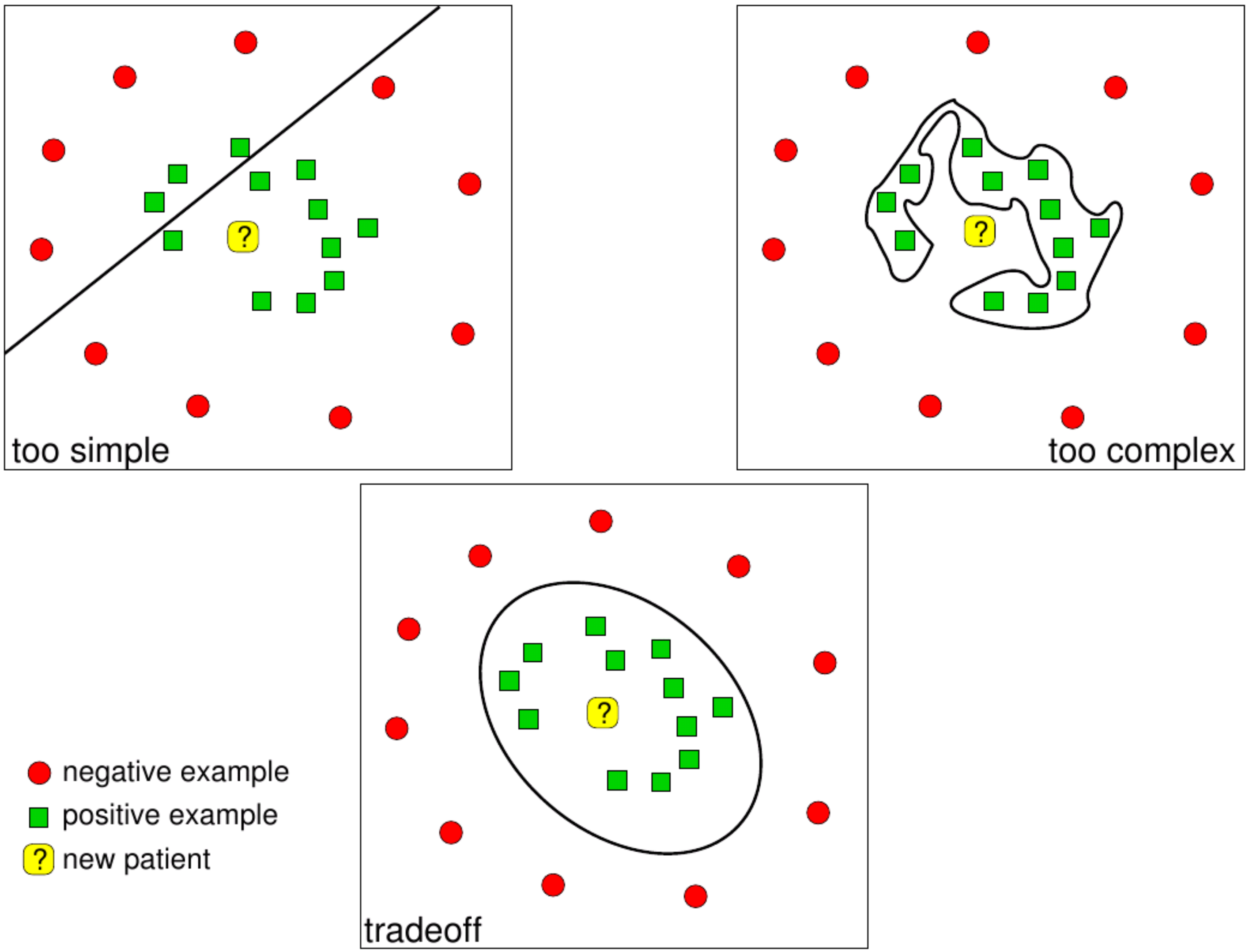
$$y_i \in \{+1, -1\} \text{ (classes, 2 kinds of cancer)}$$

Decision function:

$$f_X : \mathbb{R}^g \to \{+1, -1\}$$

$$\text{diagnosis} = f_X(\text{new patient})$$

Underfitting / Overfitting



Linear Separation of Training Data

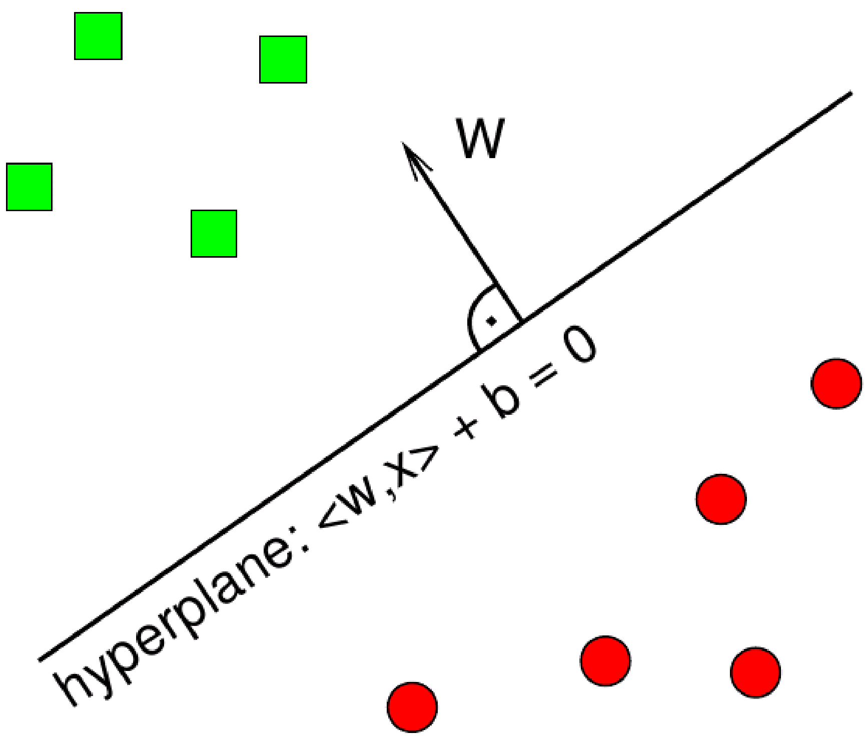
Begin with linear separation and increase the complexity in a second step with a kernel function.

A separating hyperplane is defined by

- a normal vector w and
- an offset b:

Hyperplane $\mathcal{H} = \{x | \langle w, x \rangle + b = 0\}$

$\langle \cdot, \cdot \rangle$ is called the inner product or scalar product.

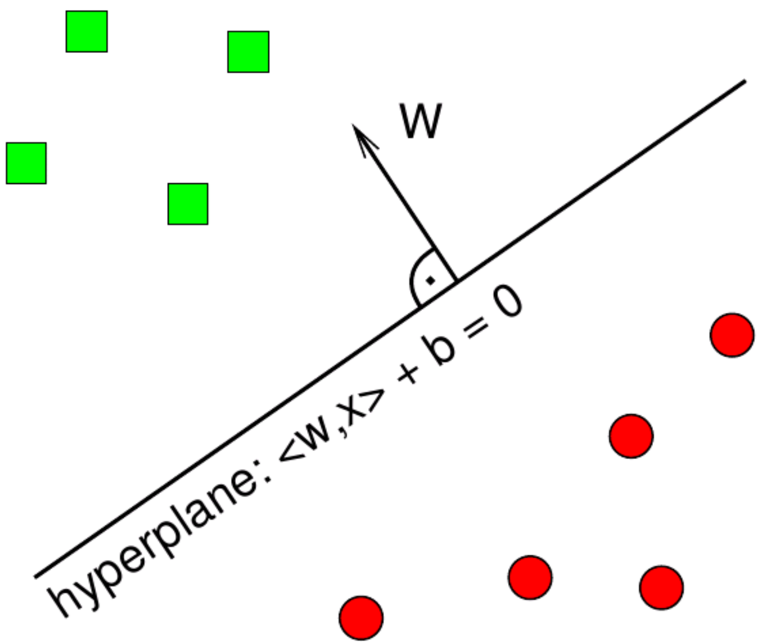


Training: Choose w and b in such a way that the hyperplane separates the training data.

Prediction: Which side of the hyperplane is the new point located on?

Points on the side that the normal vector points at are diagnosed as POSITIVE.

Points on the other side are diagnosed as NEGATIVE.



Motivation

Origin in Statistical Learning Theory; class of optimal classifiers

Core problem of Statistical Learning Theory: Ability for generalization.
When does a low training error lead to a low real error?

Binary Class Problem:
Classification $\equiv$ mapping function $f(x, u) : \; x \to y \in \{+1, -1\}$
x: sample from one of the two classes
u: parameter vector of the classifier

Learning sample with l observations $x_1, x_2, \ldots, x_l$
along with their class affiliation $y_1, y_2, \ldots, y_l$
$\to$ the empirical risk (error rate) for a given training dataset:

$$R_{emp}(u) \;=\; \frac{1}{2l} \sum_{i=1}^{l} |y_i - f(x_i, u)| \;\in [0, 1]$$

A lot of classifiers do minimize the empirical risk, e.g. Neural Networks.

Motivation

Expected value of classification error (expected risk):

$$R(u) = E\{R_{test}(u)\} = E\{\frac{1}{2}|y - f(x,u)|\} = \int \frac{1}{2}|y - f(x,u)|p(x,y)\, dxdy$$

$p(x, y)$: Distribution density of all possible samples x along with their class affiliation y (Can't evaluate this expression directly as $p(x, y)$ is not available.)

Optimal sample classification:
Search for deterministic mapping function $f(x, u) : x \to y \in \{+1, -1\}$ that minimizes the expected risk.

Core question of sample classification:

> How close do we get to the *real* error after we saw l training samples? How well can we estimate the real risk $R(u)$ from the empirical risk $R_{emp}(u)$? (Structural Risk Minimization instead of Empirical Risk Minimization)

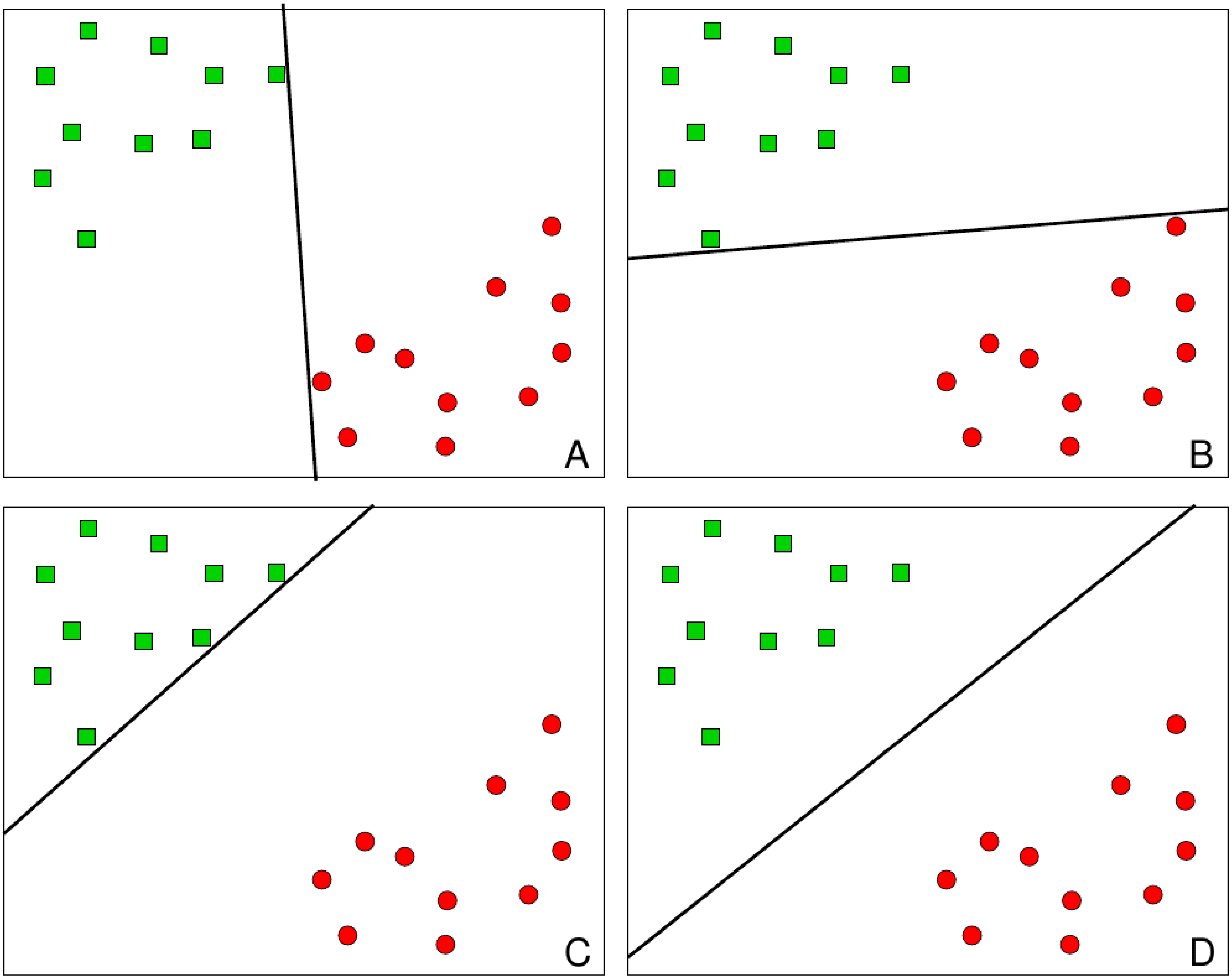
The answer is given by Learning Theory of Vapnik-Chervonenkis $\to$ SVMs

SVMs for linear separable classes

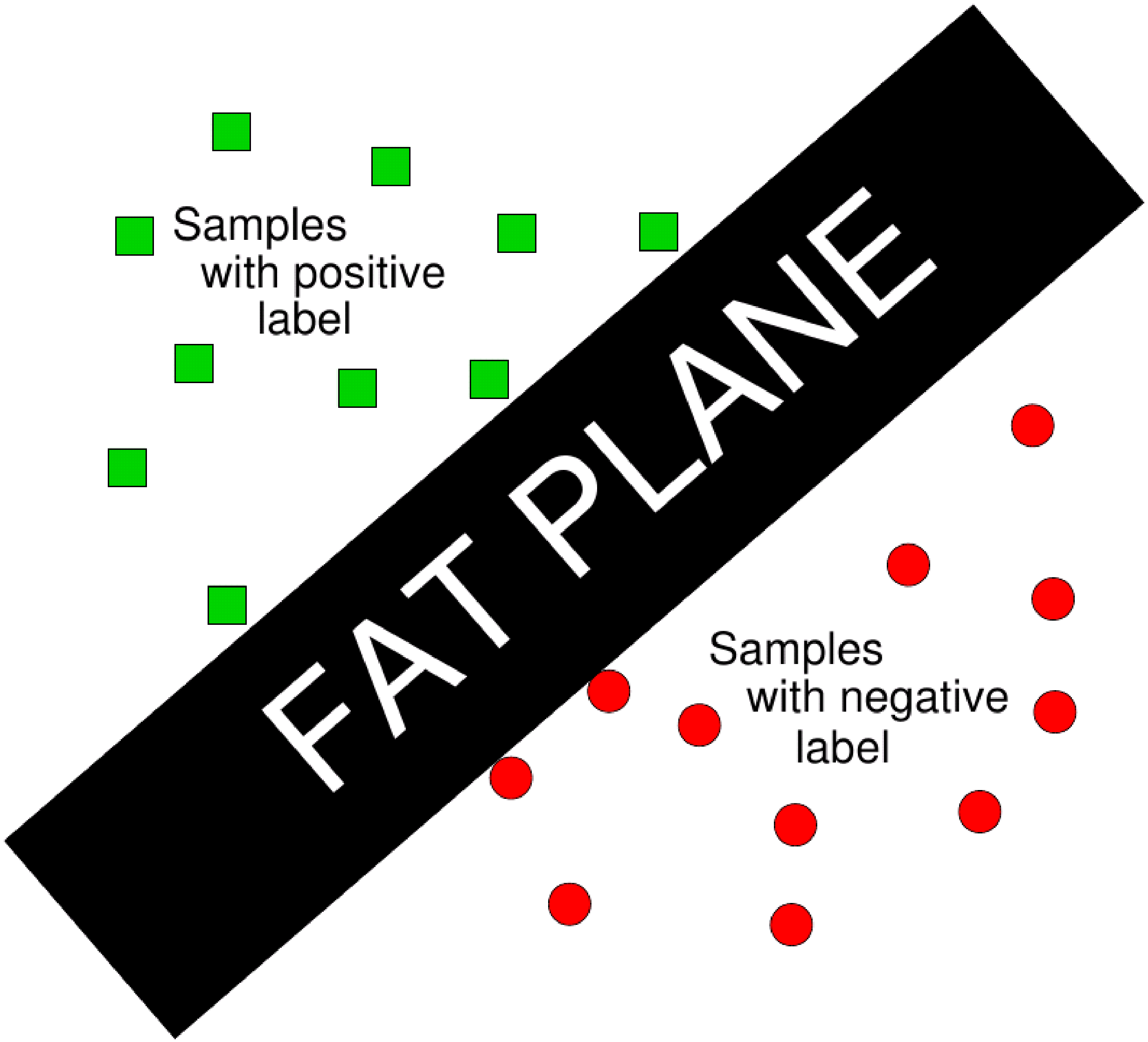
Previous solution:

- General hyperplane: $wx + b = 0$
- Classification: $\operatorname{sgn}(wx + b)$
- Training, e.g. by perceptron-algorithm
  (iterative learning, correction after every misclassification; no unique solution)

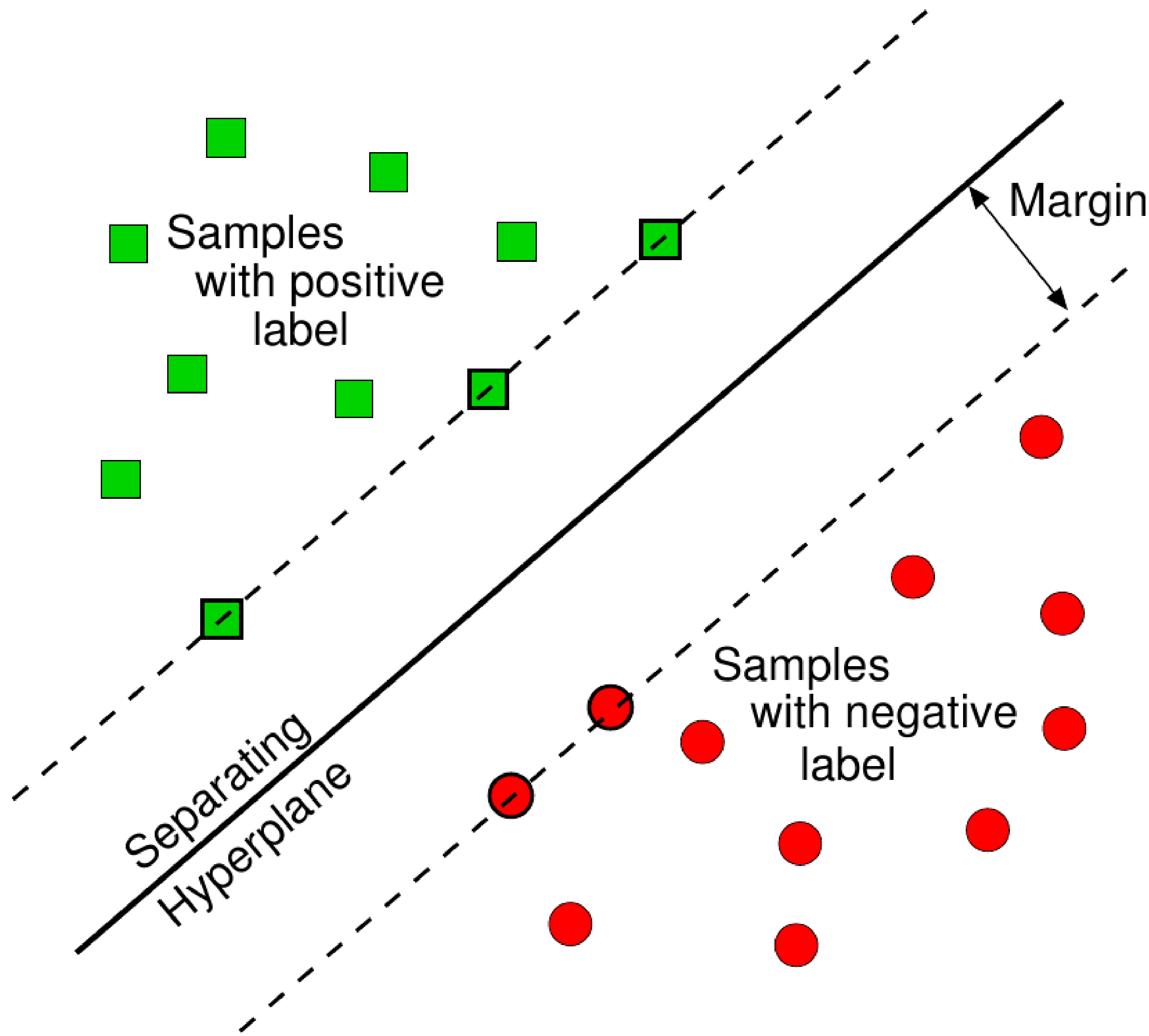
Which hyperplane is the best - and why?



No exact cut, but a . . .

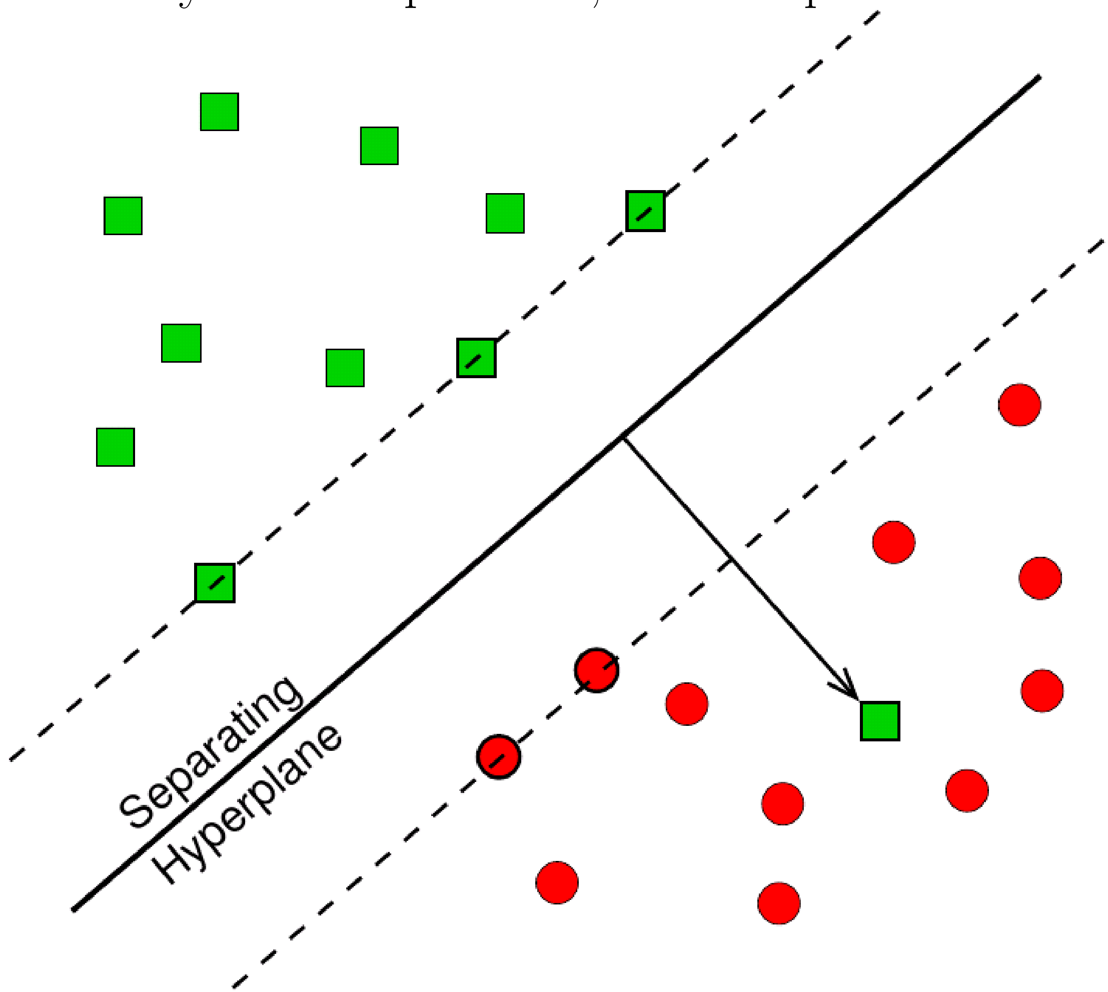


Separate the training data with maximal separation margin



Separate the training data with maximal separation margin

Try linear separation, but accept errors:

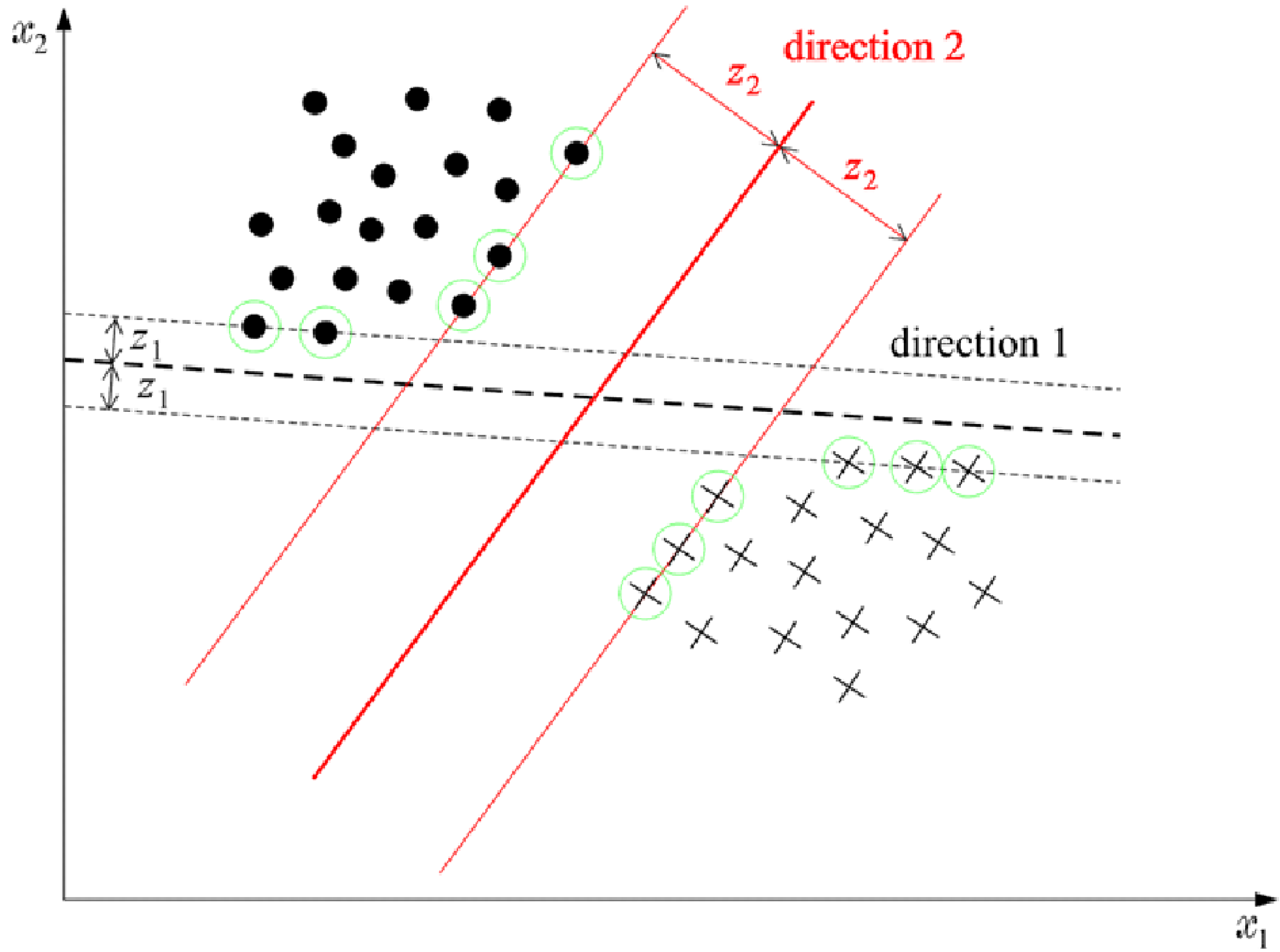


Penalty for errors: Distance to hyperplane times error weight C

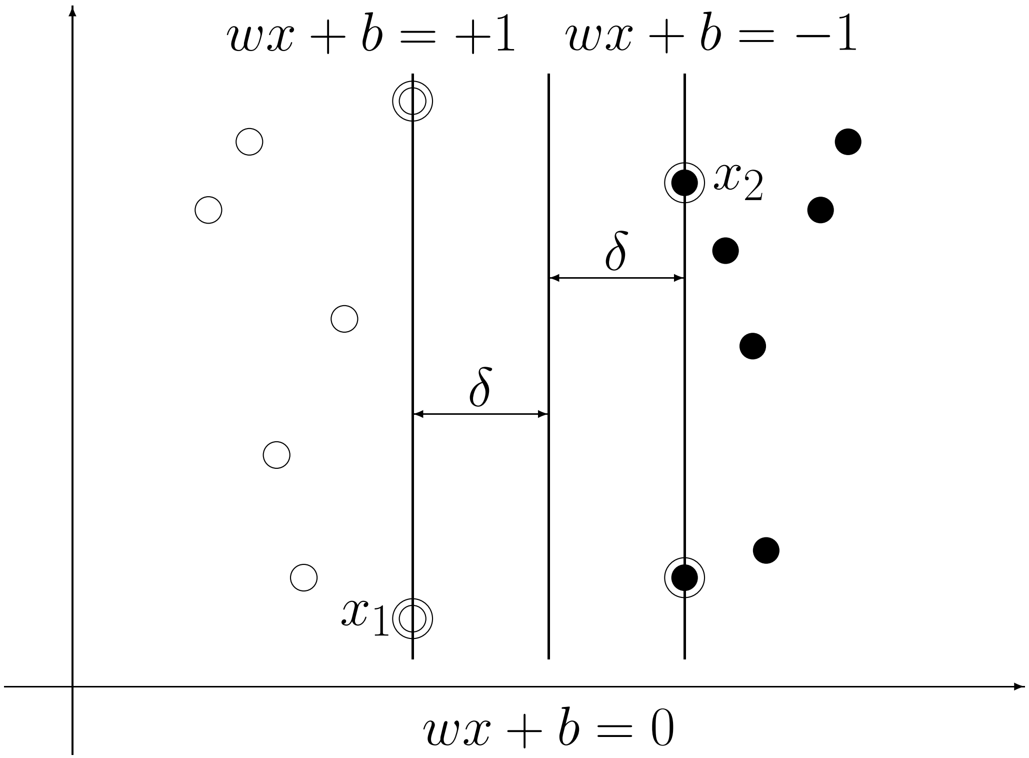
SVMs for linearly separable classes

- With SVMs we are searching for a separating hyperplane with maximal margin. *Optimum:* The hyperplane with the highest 2δ of all possible separating hyperplanes.
- This is intuitively meaningful
(At constant intra-class scattering, the confidence of right classification is growing with increasing inter-class distance)
- SVMs are theoretically justified by Statistical Learning Theory.

Large-Margin Classifier: Separation line 2 is better than 1



SVMs for linearly separable classes



Training samples are classified correctly, if:

$$y_i(wx_i+b)>0$$

Invariance of this expression towards a positive scaling leads to:

$$y_i(wx_i+b)\geq 1$$

with canonical hyperplanes:

$$\begin{cases} wx_i+b=+1; \text{(class with } y_i=+1) \\ wx_i+b=-1; \text{(class with } y_i=-1) \end{cases}$$

The distance between the canonical hyperplanes results from projecting x_1-x_2 to the unit length normal vector $\frac{w}{||w||}$:

$$2\delta = \frac{2}{||w||}; \quad \text{d.h. } \delta = \frac{1}{||w||}$$

$\rightarrow$ maximizing $\delta \equiv$ minimizing $||w||^2$

SVMs for linearly separable classes

Optimal separating plane by minimizing a quadratic function to linear constraints:

> Primal Optimization Problem:
> minimize: $J(w, b) \;=\; \frac{1}{2}||w||^2$
> to the constraints $\forall i\ [y_i(wx_i + b) \geq 1],\ \ i = 1, 2, \ldots, l$

Introducing a Lagrange-Function:

$$L(w, b, \alpha) \;=\; \frac{1}{2}||w||^2 - \sum_{i=1}^{l} \alpha_i [y_i(wx_i + b) - 1]; \quad \alpha_i \geq 0$$

leads to the *dual problem*:
maximize $L(w, b, \alpha)$ with respect to α, under the constraints:

$$\frac{\partial L(w,b,\alpha)}{\partial w} \;=\; 0 \quad \Longrightarrow \quad w \;=\; \sum_{i=1}^{l} \alpha_i y_i x_i$$

$$\frac{\partial L(w,b,\alpha)}{\partial b} \;=\; 0 \quad \Longrightarrow \quad \sum_{i=1}^{l} \alpha_i y_i \;=\; 0$$

SVMs for linearly separable classes

Insert this terms in $L(w, b, \alpha)$:

$$\begin{aligned}
L(w, b, \alpha) &= \frac{1}{2}||w||^2 - \sum_{i=1}^{l} \alpha_i [y_i(wx_i + b) - 1] \\
&= \frac{1}{2} w \cdot w - w \cdot \sum_{i=1}^{l} \alpha_i y_i x_i - b \cdot \sum_{i=1}^{l} \alpha_i y_i + \sum_{i=1}^{l} \alpha_i \\
&= \frac{1}{2} w \cdot w - w \cdot w + \sum_{i=1}^{l} \alpha_i \\
&= -\frac{1}{2} w \cdot w + \sum_{i=1}^{l} \alpha_i \\
&= -\frac{1}{2} \sum_{i=1}^{l} \sum_{j=1}^{l} y_i y_j \alpha_i \alpha_j x_i x_j + \sum_{i=1}^{l} \alpha_i
\end{aligned}$$

SVMs for linearly separable classes

Dual Optimization Problem:

maximize: $L'(\alpha) \; = \; \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i=1}^{l} \sum_{j=1}^{l} y_i y_j \alpha_i \alpha_j x_i x_j$

to the constraints $\alpha_i \geq 0$ and $\sum_{i=1}^{l} y_i \alpha_i = 0$

This optimization problem can be solved numerically with the help of standard quadratic programming techniques.

SVMs for linearly separable classes

Solution of the optimization problem:

$$
\begin{aligned}
w^* &= \sum_{i=1}^{l} \alpha_i y_i x_i = \sum_{x_i \in SV} \alpha_i y_i x_i \\
b^* &= -\frac{1}{2} \cdot w^* \cdot (x_p + x_m) \\
&\quad \text{for arbitrary } x_p \in SV,\ y_p = +1,\ \text{und } x_m \in SV,\ y_m = -1
\end{aligned}
$$

where

$$SV = \{x_i \mid \alpha_i > 0,\ i = 1, 2, \ldots, l\}$$

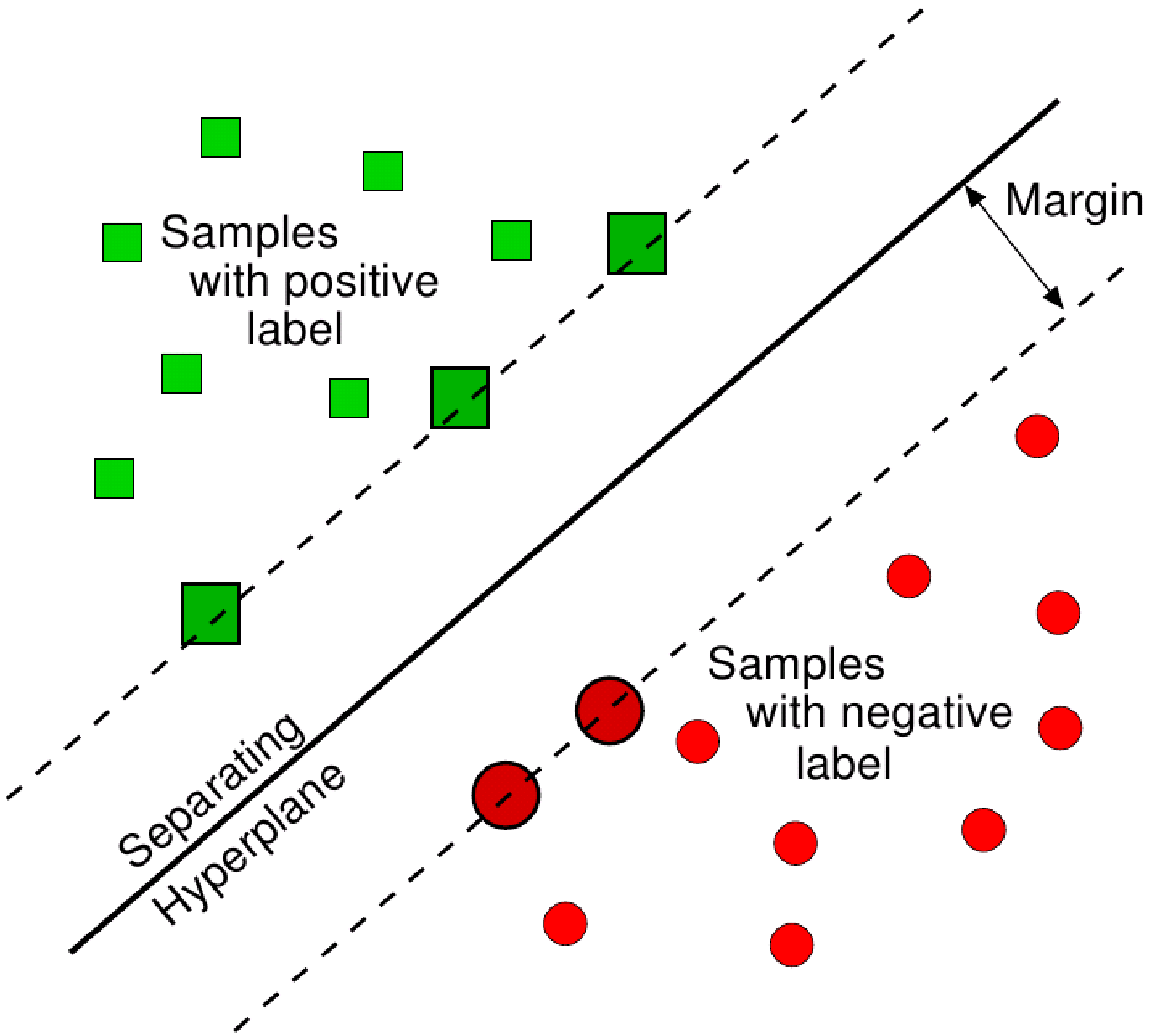
is the set of all support vectors.

Classification rule:

$$\operatorname{sgn}(w^* x + b^*) = \operatorname{sgn}[(\sum_{x_i \in SV} \alpha_i y_i x_i) x + b^*]$$

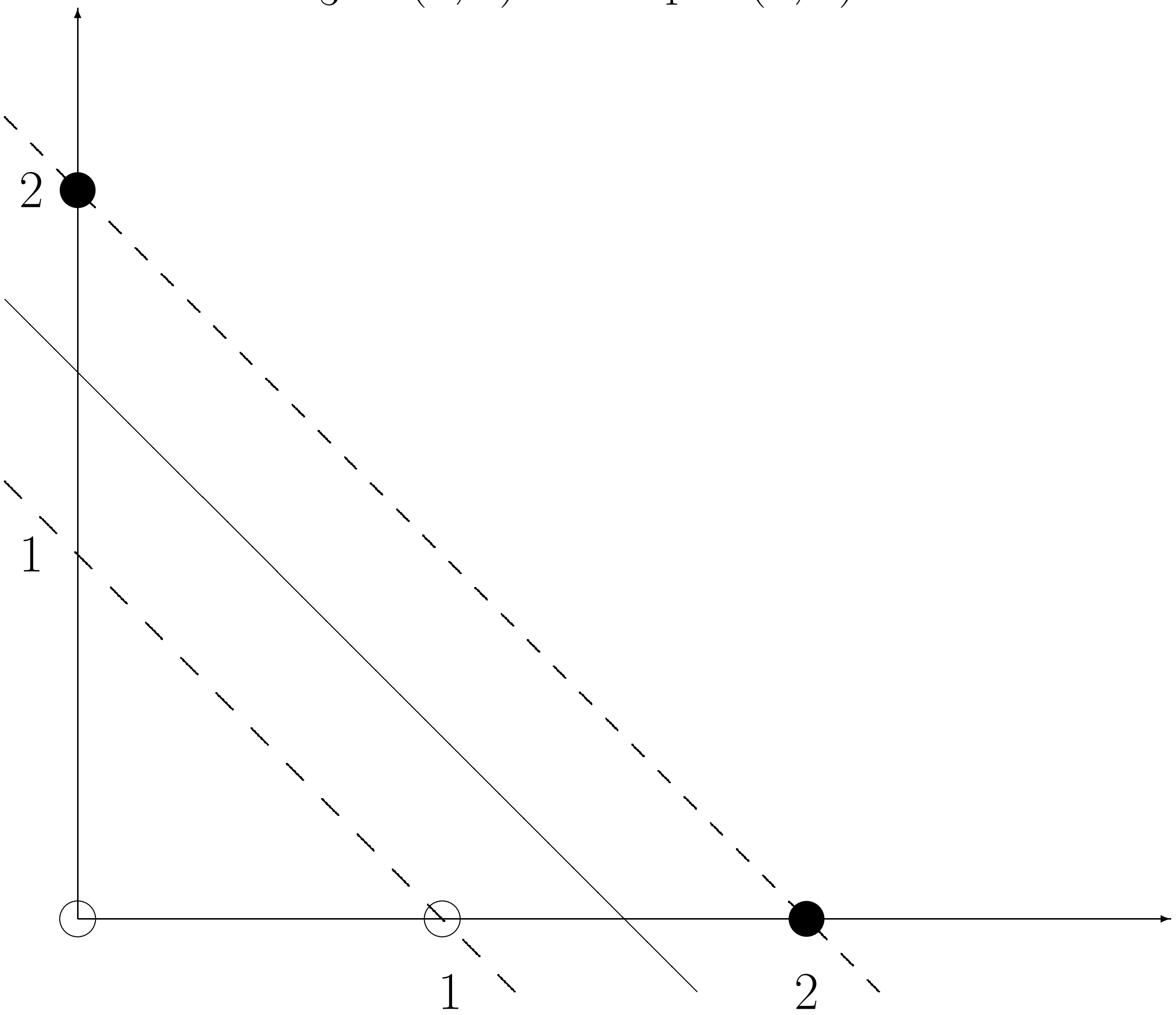
The classification only depends on the support vectors!

Example: Support Vectors



SVMs for linearly separable classes

Example: class +1 contains $x_1 = (0, 0)$ and $x_2 = (1, 0)$;
class -1 contains $x_3 = (2, 0)$ and $x_4 = (0, 2)$



SVMs for linearly separable classes

The Dual Optimization Problem is:

$$\boxed{\begin{array}{l} \text{maximize: } L'(\alpha) \;=\; (\alpha_1 + \alpha_2 + \alpha_3 + \alpha_4) - \frac{1}{2}(\alpha_2^2 - 4\alpha_2\alpha_3 + 4\alpha_3^2 + 4\alpha_4^2) \\ \text{to the constraints } \alpha_i \geq 0 \text{ and } \alpha_1 + \alpha_2 - \alpha_3 - \alpha_4 = 0 \end{array}}$$

Solution:

$$\begin{aligned}
&\alpha_1 = 0, \quad \alpha_2 = 1, \quad \alpha_3 = \tfrac{3}{4}, \quad \alpha_4 = \tfrac{1}{4} \\
&SV \;=\; \{(1,0),\ (2,0),\ (0,2)\} \\
&w^* \;=\; 1 \cdot (1,0) - \tfrac{3}{4} \cdot (2,0) - \tfrac{1}{4} \cdot (0,2) \;=\; (-\tfrac{1}{2}, -\tfrac{1}{2}) \\
&b^* \;=\; -\tfrac{1}{2} \cdot (-\tfrac{1}{2}, -\tfrac{1}{2}) \cdot ((1,0) + (2,0)) \;=\; \tfrac{3}{4}
\end{aligned}$$

Optimal separation line: $x + y = \frac{3}{2}$

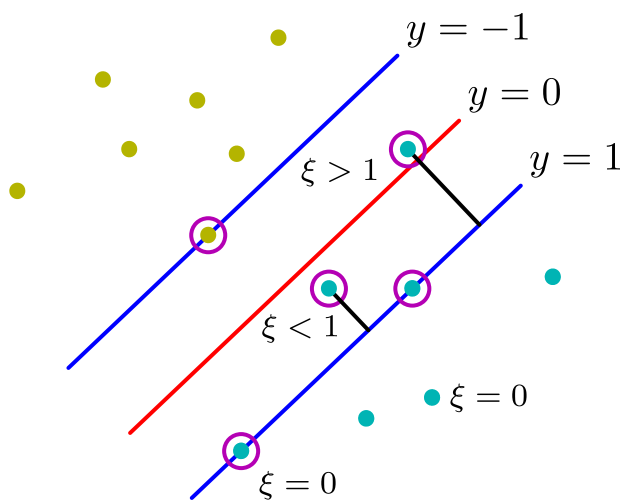
SVMs for linearly separable classes

Observations:

- For the Support Vectors holds: $\alpha_i > 0$
- For all training samples outside the margin holds: $\alpha_i = 0$
- Support Vectors form a *sparse* representation of the sample; They are sufficient for classification.
- The solution is the global optima and unique
- The optimization procedure only requires scalar products $x_i x_j$

SVMs for non-linearly separable classes

In this example there is no separating line such as $\forall i \; [y_i(wx_i + b) \geq 1]$



All three cases can be interpreted as: $y_i(wx_i + b) \; \geq \; 1 - \xi_i$

A) $\xi_i = 0$
B) $0 < \xi_i \leq 1$
C) $\xi_i > 1$

Three possible cases:

A) Vectors beyond the margin, which are correctly classified, i.e.

$$y_i(wx_i + b) \; \geq \; 1$$

B) Vectors within the margin, which are correctly classified, i.e.

$$0 \; \leq \; y_i(wx_i + b) \; < \; 1$$

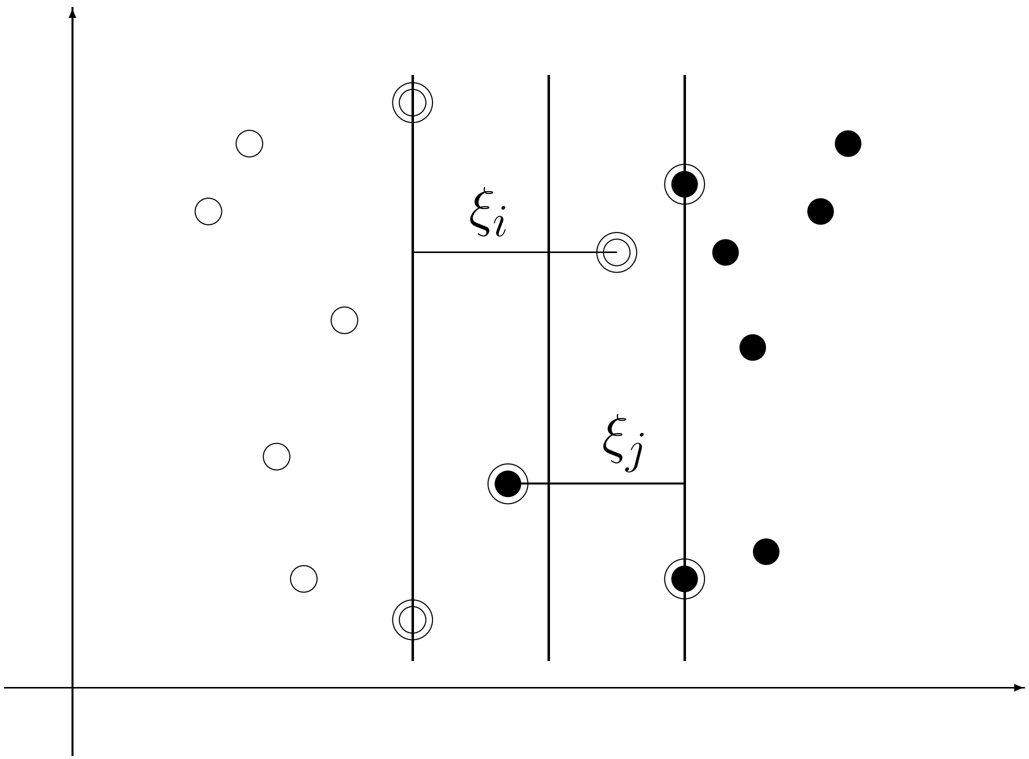
C) Vectors that are not correctly classified, i.e.

$$y_i(wx_i + b) \; < \; 0$$

Motivation for generalization:

- Previous approach gives no solution for classes that are non-lin. separable.
- Improvement of the generalization on outliers within the margin

Soft-Margin SVM: Introduce “slack”-Variables



SVMs for non-linearly separable classes

Penalty for outliers via "slack"-Variables

Primale Optimization Problem:

minimize: $J(w, b, \xi) \; = \; \frac{1}{2}||w||^2 + C \sum_{i=1}^{l} \xi_i$

to the constraints $\forall i \; [y_i(wx_i + b) \geq 1 - \xi_i, \;\; \xi_i \geq 0]$

Dual Optimization Problem:

maximize: $L'(\alpha) \; = \; \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i=1}^{l} \sum_{j=1}^{l} y_i y_j \alpha_i \alpha_j x_i x_j$

to the constraints $0 \leq \alpha_i \leq C$ and $\sum_{i=1}^{l} y_i \alpha_i = 0$

(Neither slack-Variables nor Lagrange-Multiplier occur in the dual optimization problem.)

The only difference compared to the linear separable case: Constant C in the constraints.

SVMs for non-linearly separable classes

Solution of the optimization problem:

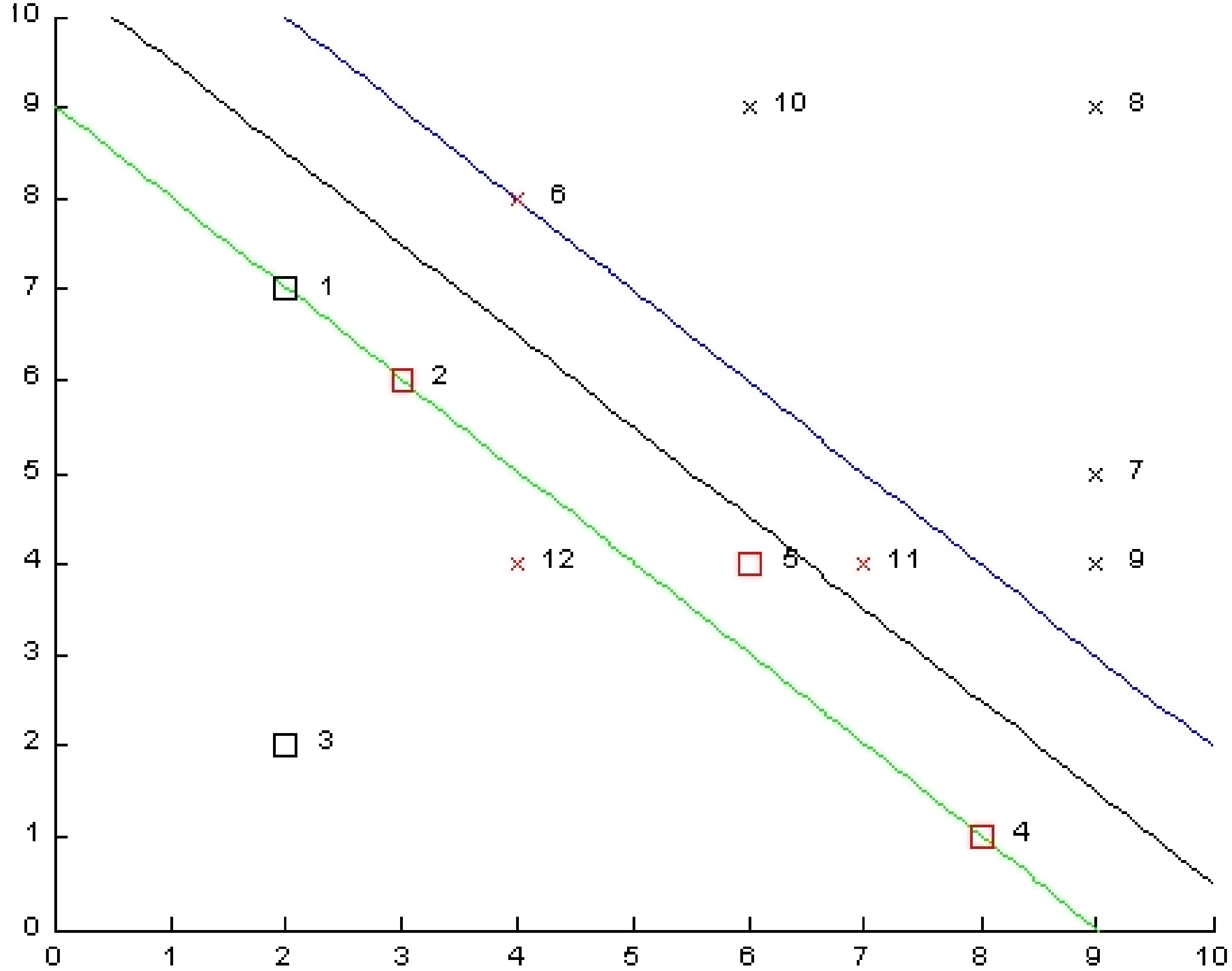
$$
\begin{aligned}
w^* &= \sum_{i=1}^{l} \alpha_i y_i x_i = \sum_{x_i \in SV} \alpha_i y_i x_i \\
b^* &= y_k(1 - \xi_k) - w^* x_k; \quad k = \arg\max_i \alpha_i
\end{aligned}
$$

where

$$
SV = \{x_i \mid \alpha_i > 0,\ i = 1, 2, \ldots, l\}
$$

describes the set of all Support Vectors.

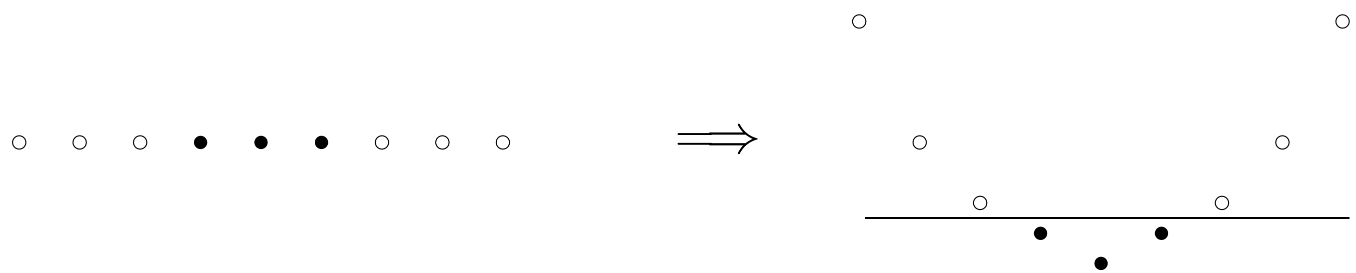
Example: non-linearly separable classes



Non-linear SVMs

Non-linear class boundaries $\rightarrow$ low precision

Example: Transformation $\Psi(x) = (x, x^2) \rightarrow C_1$ and C_2 linearly separable



Idea:

Transformation of attributes $x \in \Re^n$ in a higher dimensional space $\Re^m$, $m > n$ by

$$\Psi : \quad \Re^n \longrightarrow \Re^m$$

and search for an optimal linear separating hyperplane in this space.

Transformation Ψ increases linear separability.

Separating hyperplane in $\Re^m \equiv$ non-linear separating plane in $\Re^n$

Non-linear SVMs

Problem: High dimensionality of the attribute space $\Re^m$
E.g. Polynomes of p-th degree over $\Re^n \to \Re^m$, $m = O(n^p)$

Trick with kernel function:

Originally in $\Re^n$: only scalar products $x_i x_j$ required
new in $\Re^m$: only scalar products $\Psi(x_i)\Psi(x_j)$ required

Solution
No need to compute $\Psi(x_i)\Psi(x_j)$, but express them at reduced complexity with the kernel function

$$K(x_i, x_j) = \Psi(x_i)\Psi(x_j)$$

Non-linear SVMs

Example: For the transformation $\Psi : \Re^2 \longrightarrow \Re^6$

$$\Psi((y_1, y_2)) = (y_1^2,\ y_2^2,\ \sqrt{2}y_1,\ \sqrt{2}y_2,\ \sqrt{2}y_1y_2,\ 1)$$

the kernel function computes

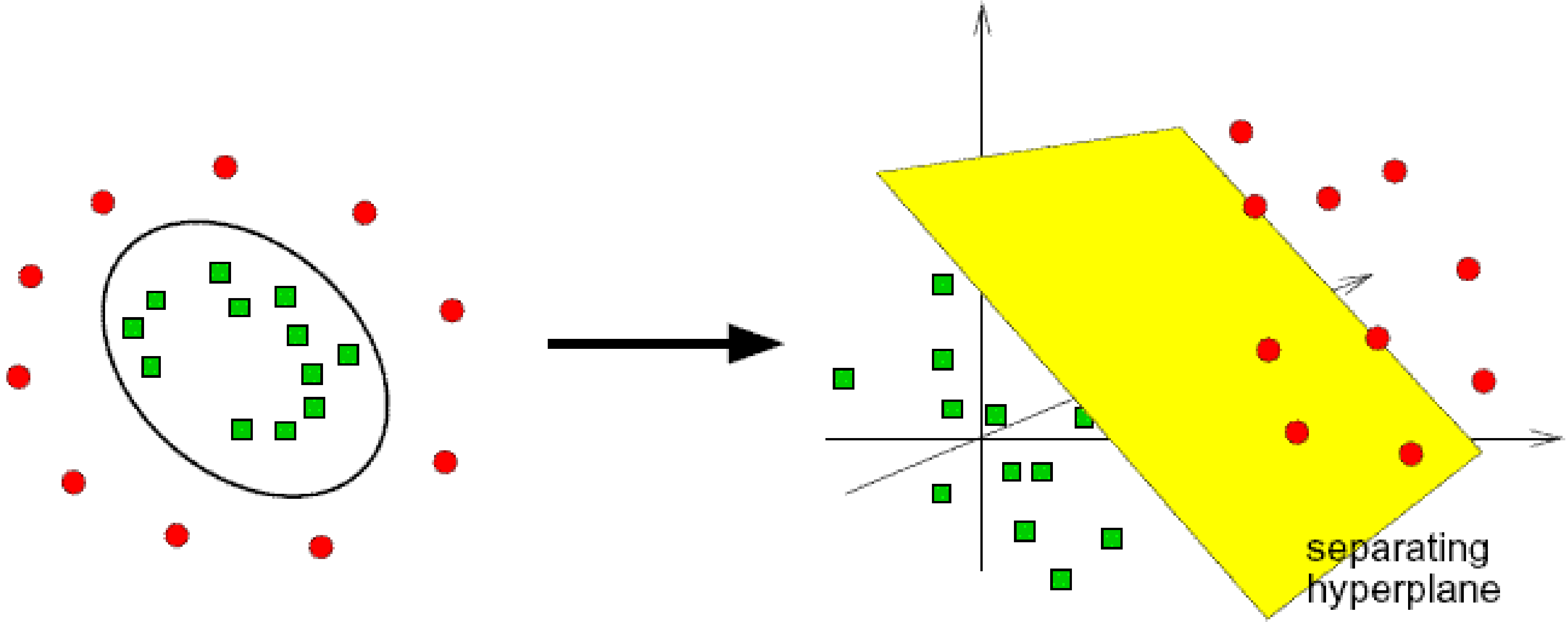
$$\begin{aligned}
K(x_i, x_j) &= (x_i x_j + 1)^2 \\
&= ((y_{i1}, y_{i2}) \cdot (y_{j1}, y_{j2}) + 1)^2 \\
&= (y_{i1}y_{j1} + y_{i2}y_{j2} + 1)^2 \\
&= (y_{i1}^2, y_{i2}^2, \sqrt{2}y_{i1}, \sqrt{2}y_{i2}, \sqrt{2}y_{i1}y_{i2}, 1) \\
&\quad \cdot (y_{j1}^2, y_{j2}^2, \sqrt{2}y_{j1}, \sqrt{2}y_{j2}, \sqrt{2}y_{j1}y_{j2}, 1) \\
&= \Psi(x_i)\Psi(x_j)
\end{aligned}$$

the scalar product in the new attribute space $\Re^6$

Non-linear SVMs

Example: $\Psi : \Re^2 \longrightarrow \Re^3$

$$\Psi((y_1, y_2)) = (y_1^2, \sqrt{2}y_1y_2,\ y_2^2)$$



The kernel function

$$K(x_i, x_j) = (x_i x_j)^2 = \Psi(x_i)\Psi(x_j)$$

computes the scalar product in the new attribute space $\Re^3$. It is possible to compute the scalar product of $\Psi(x_i)$ and $\Psi(x_j)$ without applying the function Ψ.

Nonlinear SVMs

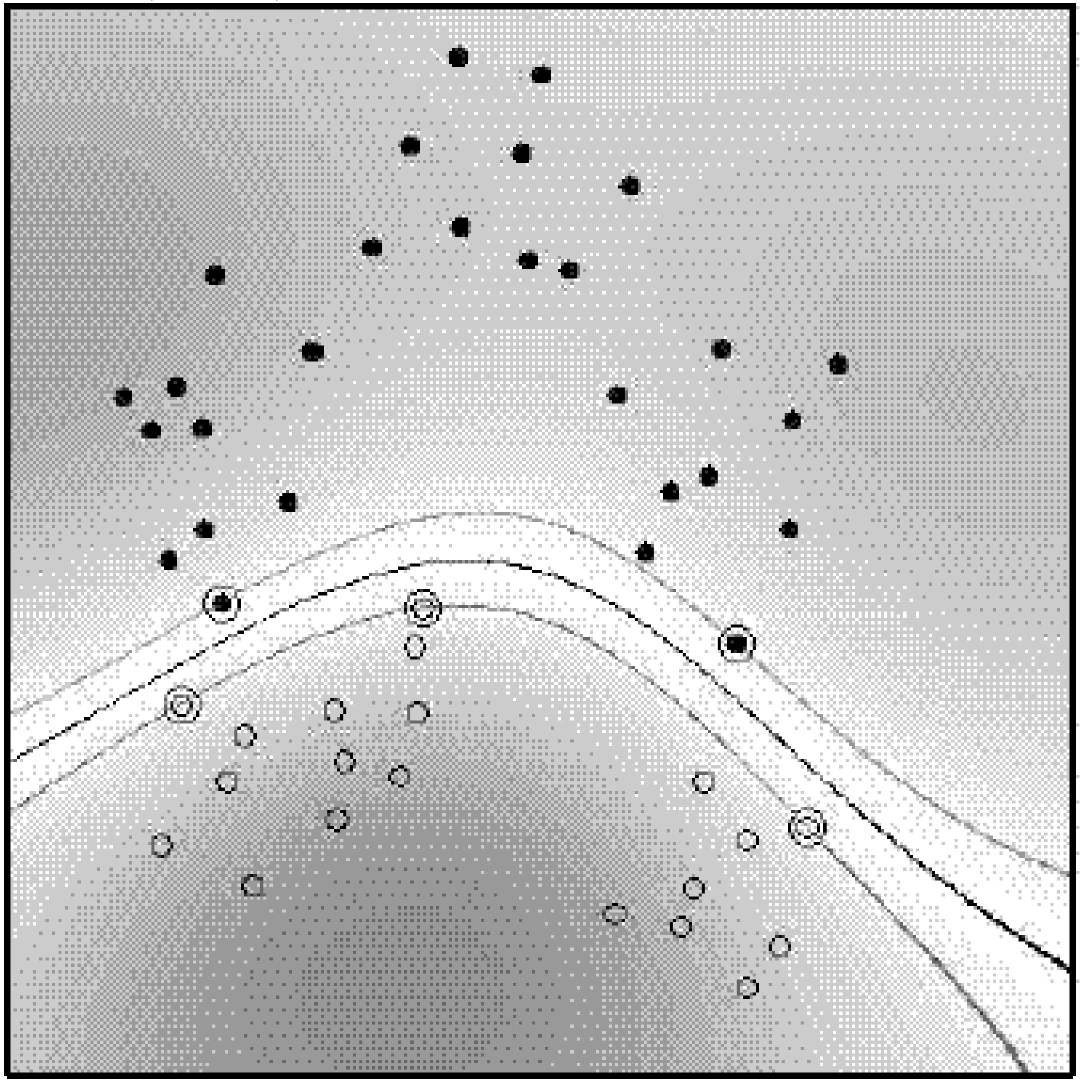
Commonly used kernel functions:

$$
\begin{array}{lrcl}
\text{Polynomial-Kernel:} & K(x_i, x_j) & = & (x_i x_j)^d \\
\text{Gauss-Kernel:} & K(x_i, x_j) & = & e^{-\frac{||x_i - x_j||^2}{c}} \\
\text{Sigmoid-Kernel:} & K(x_i, x_j) & = & \tanh(\beta_1 x_i x_j + \beta_2)
\end{array}
$$

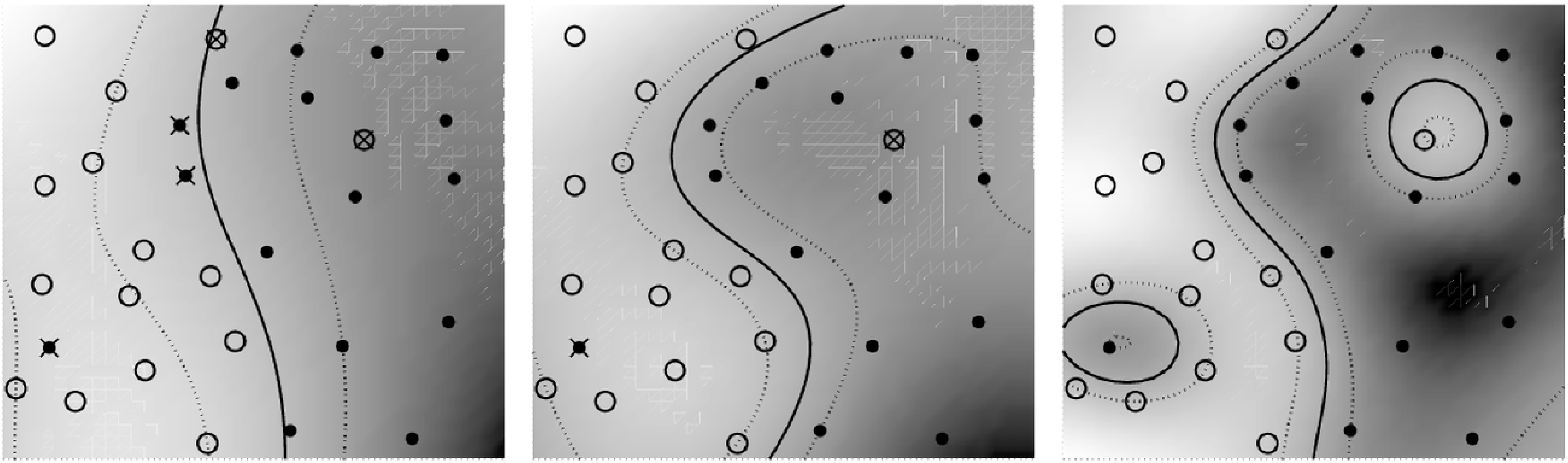
Linear combination of valid kernels $\rightarrow$ new kernel functions

We do not need to know what the new attribute space $\Re^m$ looks like. The only thing we need is the kernel function as a measure for similarity.

Example: Gauss-Kernel ($c = 1$). The Support Vectors are tagged by an extra circle.



Example: Gauss-Kernel ($c = 1$) for Soft-Margin SVM.



Final Remarks

Advantages of SVMs:

- According to current knowledge SVMs yield very good classification results; in some tasks they are considered to be the top-performer.
- Sparse representation of the solution by Support Vectors
- Easily practicable: few parameters, no need for a-priori-knowledge
- Geometrically intuitive operation
- Theoretical statements about results: global optima, ability for generalization

Disadvantages of SVMs

- Learning process is slow and in need of intense memory
- "Tuning SVMs remains a black art: selecting a specific kernel and parameters is usually done in a try-and-see manner"

Final Remarks

- List of SVM-implementations at
  `http://www.kernel-machines.org/software`
- The most common one is LIBSVM:
  `http://www.csie.ntu.edu.tw/~cjlin/libsvm/`