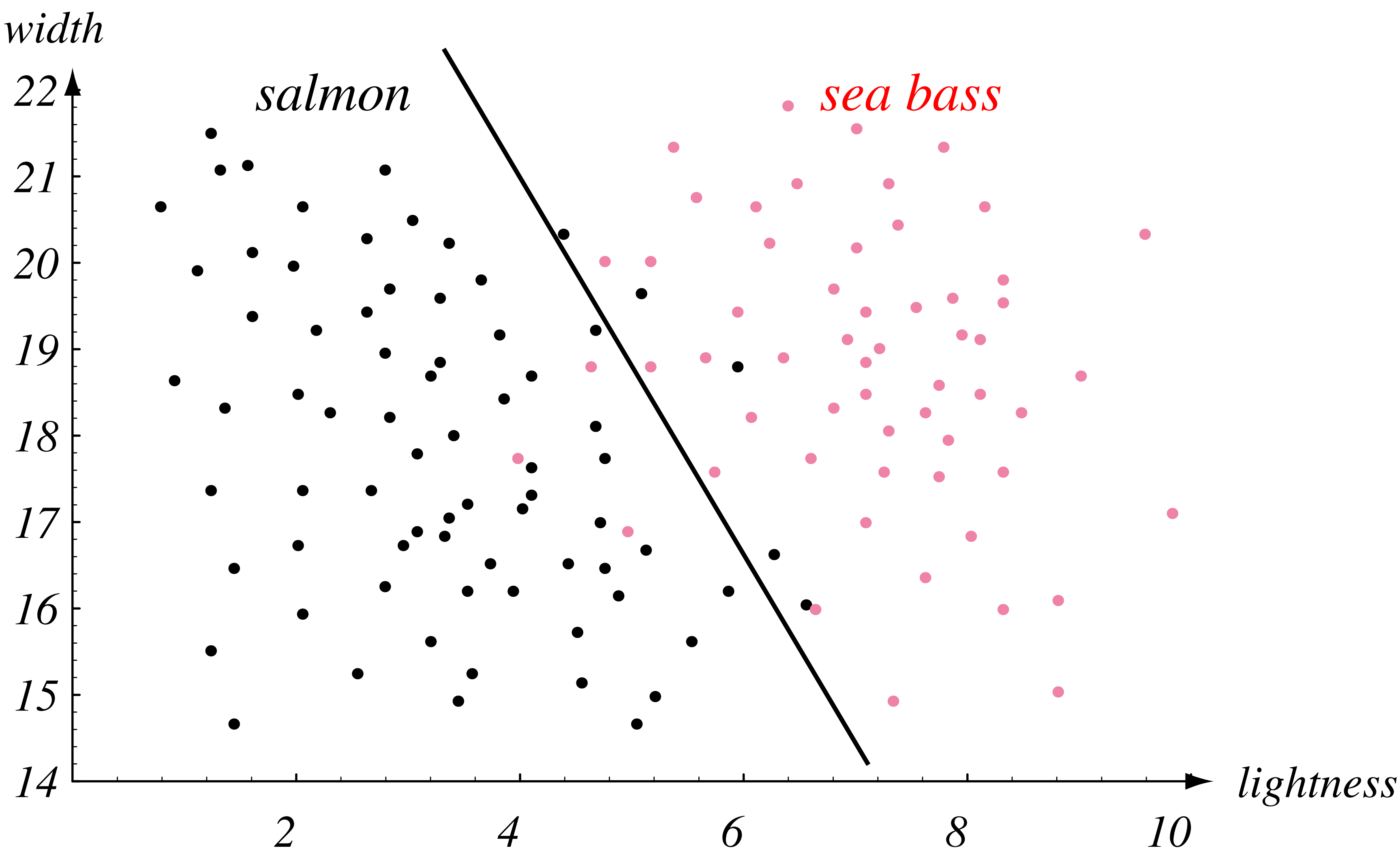
Machine Learning: Topic Chart

- **Core problems of pattern recognition**
- **Bayesian decision theory**
- ***Perceptrons and support vector machines***
- **Data clustering**
- **Dimension reduction**

Linear Classification



Generalized Linear Discriminant Functions

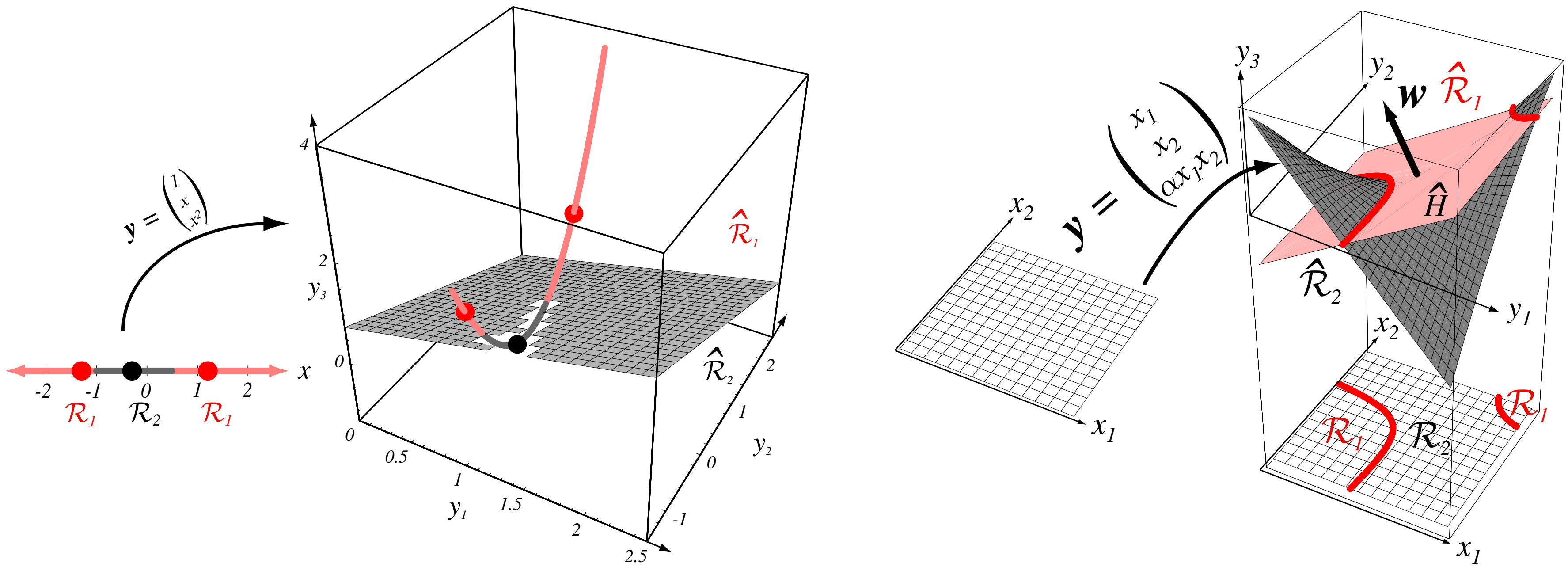
Linear Discriminant Functions can be written as

$$g(x) \;=\; w_0 + \sum_{1 \le i \le d} w_i x_i \;=\; (w_0, w)(1, x)^T \;=:\; a^T y\,.$$

with generalized coordinates $y = (1, x)^T,\ a = (w_0, w)^T$.
Note that the generalized separating hyperplanes contain the origin of the y-space!

Quadratic Discriminant Functions have the form

$$g(x) = w_0 + \sum_{i \le d} w_i x_i + \sum_{i \le d} \sum_{j \le d} w_{ij} x_i x_j\,.$$

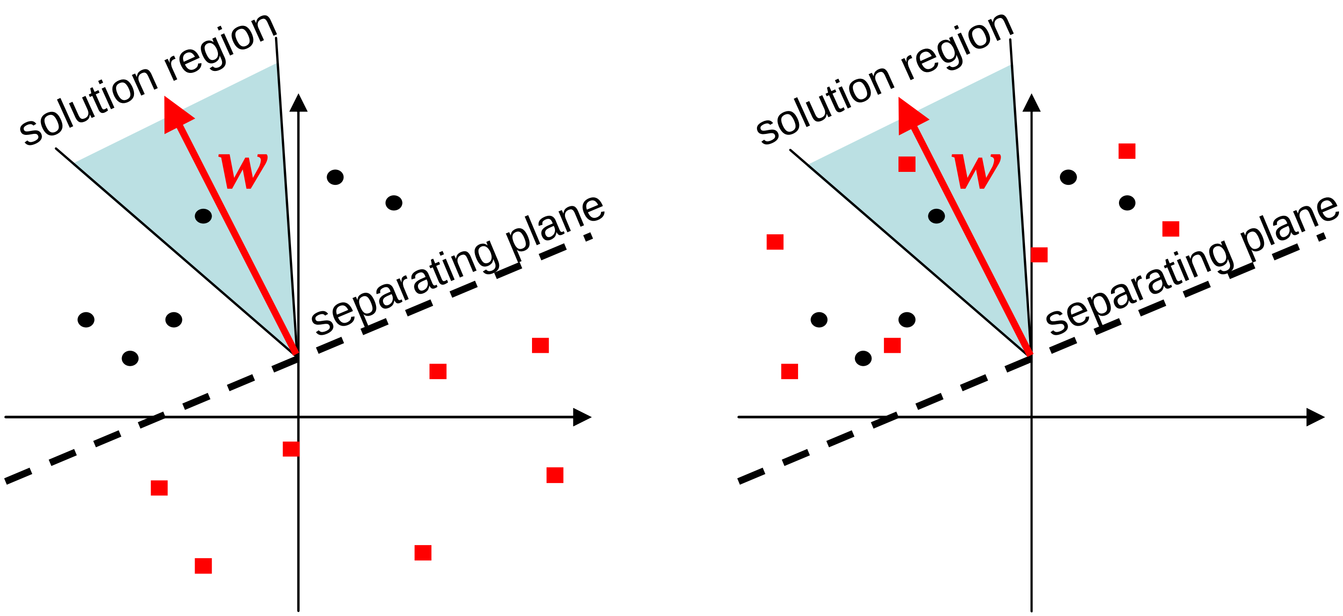


The quadratic map $y = (1, x, x^2)^T$ transforms a line in a parabola in three dimensions. A planar split in y-space corresponds to a partitioning in x-space which is not simply connected.

Linear Separable Two Class Case

Linear Separability:

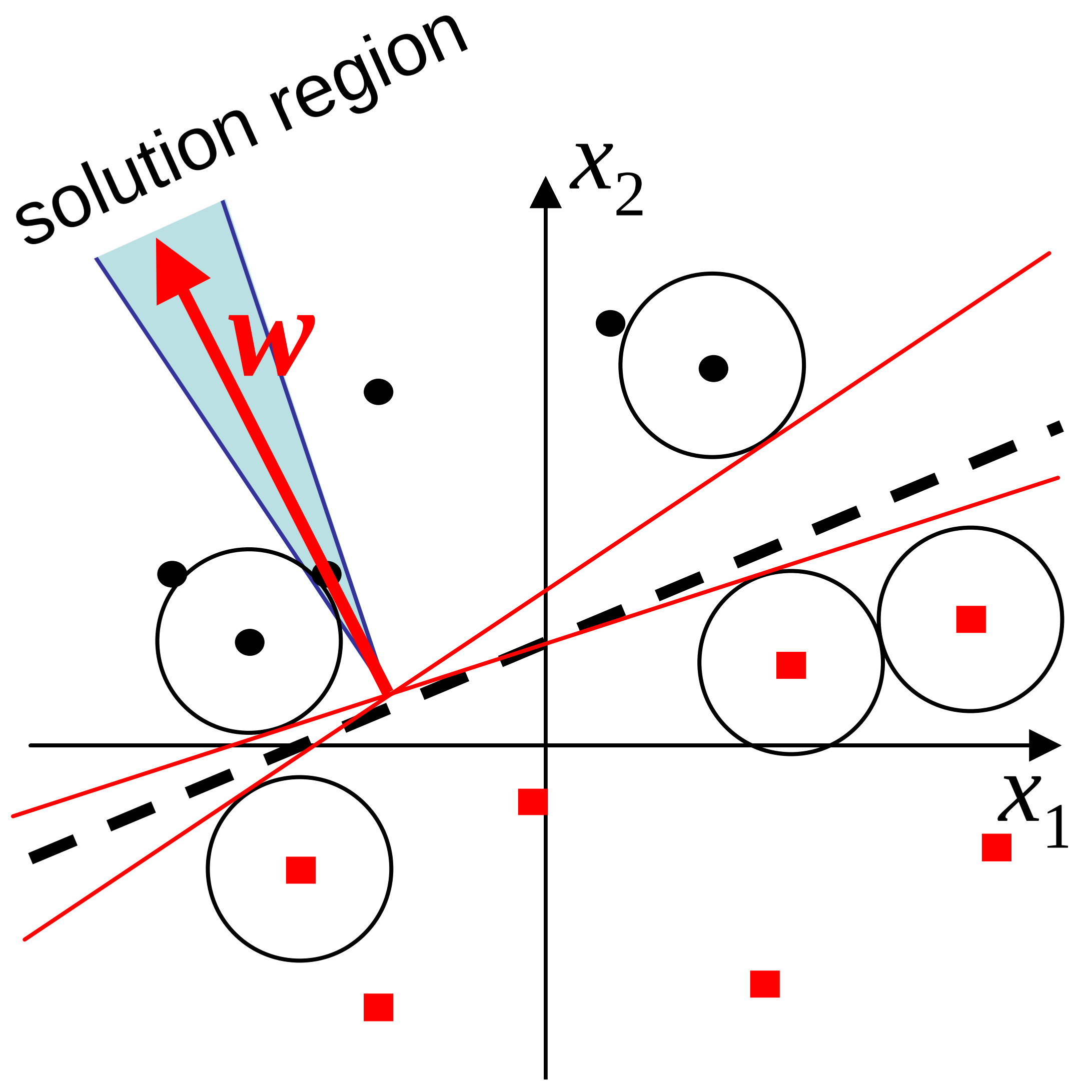
$$\exists (w_0, w)^T \text{ with } \begin{cases} w^T x_i + w_0 > 0 & \text{for } y_i = 1 \\ w^T x_i + w_0 < 0 & \text{for } y_i = 2 \end{cases}$$



Problem: The solution vector is not unique!

The Margin Idea in the Linear Separable Two Class Case

Idea: Introduce a *margin* m to classify data with a “safe distance” from the decision boundary, i.e., $z_i(w^T x_i + w_0) \geq m > 0.$ **Regularization of classifier!**



The Perceptron Criterion

(in generalized coordinates $y = (1, x)^T,\ a = (w_0, w)^T$)

Problem: Solve the inequalities $a^T y_i > 0, \forall i$

Criterion Functions: $J(a; y_1, \ldots, y_n) \ = \ldots$

- $\ldots$ number of misclassified samples: poor choice since J is piecewise constant! No gradient!
- $\ldots$ sum of violating projections.

Perceptron Criterion:

$$\boxed{J_p(a) = \sum_{y \in \mathcal{Y}} (-a^T y)}$$

$\mathcal{Y}$ is the set of misclassified samples.

Perceptron Rule: $\Rightarrow\ a(k+1) = a(k) + \eta(k) \sum_{y \in \mathcal{Y}} y$

Perceptron Algorithm (Batch Version)

Require: initialize $a, \theta, \eta(\cdot)$

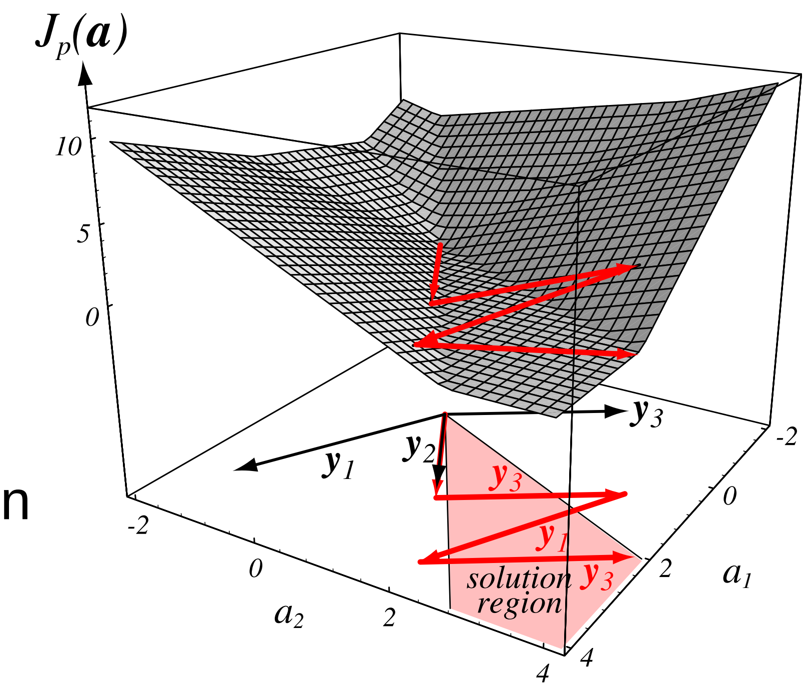
1: $k \leftarrow 0$

2: **repeat**

3: $\quad a \leftarrow a + \sum_{y \in \mathcal{Y}} \eta(k) y$

4: $\quad k \leftarrow k + 1$

5: **until** $|\eta(k) \sum_{y \in \mathcal{Y}} y| < \theta$



Fixed-Increment Single Sample Perceptron

Require: initialize $a, k \leftarrow 0$

1: **repeat**

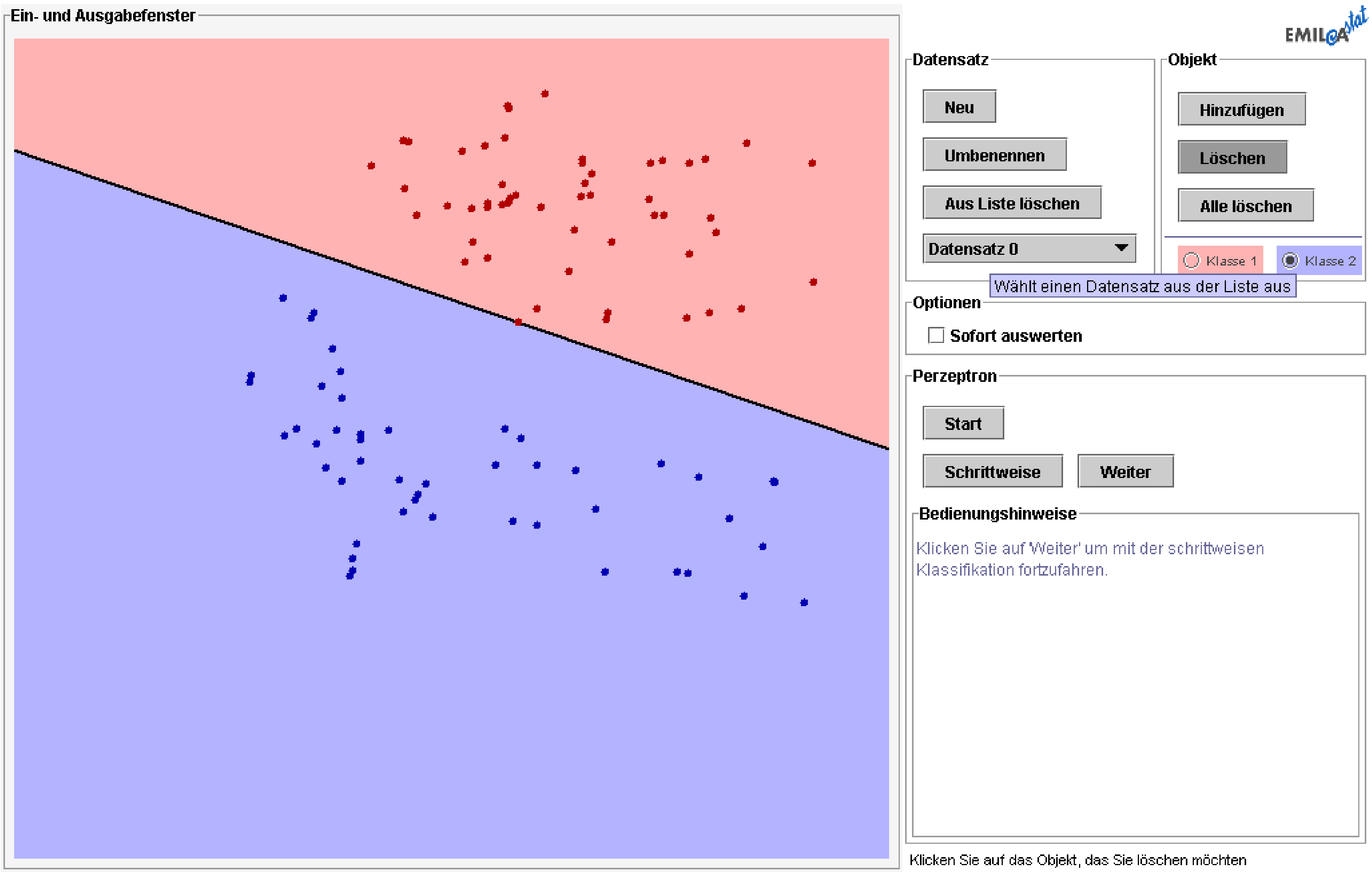
2: $\quad k \leftarrow (k+1)$ mod n

3: $\quad$ **if** y^k is misclassified by a **then**

4: $\quad\quad a \leftarrow a + y^k$

5: $\quad$ **end if**

6: **until** all patterns are correctly classified


Ein- und Ausgabefenster
EMIL@A stat
Datensatz
Neu
Umbenennen
Aus Liste löschen
Datensatz 0
Objekt
Hinzufügen
Löschen
Alle löschen
Klasse 1
Klasse 2
Wählt einen Datensatz aus der Liste aus
Optionen
Sofort auswerten
Perzeptron
Start
Schrittweise
Weiter
Bedienungshinweise
Klicken Sie auf 'Weiter' um mit der schrittweisen Klassifikation fortzufahren.
Klicken Sie auf das Objekt, das Sie löschen möchten

Perceptron Convergence

Theorem: If the training samples are linearly separable, then the sequence of weight vectors $a \leftarrow a + y^k$ will terminate at a solution vector.

Proof: Let $\hat{a}$ be a solution vector, i.e., $\hat{a}^T y_i > 0, \ \forall i$ and let $\alpha > 0$ be a scaling factor. Then it holds:

$$
\begin{aligned}
a(k+1) - \alpha\hat{a} &= a(k) - \alpha\hat{a} + y^k \\
\Rightarrow \ \|a(k+1) - \alpha\hat{a}\|^2 &= \|a(k) - \alpha\hat{a}\|^2 + 2\left(a(k) - \alpha\hat{a}\right)^T y^k + \|y^k\|^2
\end{aligned}
$$

Since y^k was misclassified the inequality $a^T(k)y^k \leq 0$ holds.

$$
\Rightarrow \ \|a(k+1) - \alpha\hat{a}\|^2 \leq \|a(k) - \alpha\hat{a}\|^2 - 2\underbrace{\alpha\hat{a}^T y^k}_{>0} + \|y^k\|^2
$$

$\hat{a}^T y^k$ dominates $\|y^k\|^2$ for sufficiently large α.

Defs.: $\beta^2 := \max_i \|y_i\|^2$, $\gamma := \min_i(\hat{a}^T y^i) > 0$

$$\begin{aligned} \Rightarrow \|a(k+1) - \alpha\hat{a}\|^2 &\leq \|a(k) - \alpha\hat{a}\|^2 - 2\alpha\gamma + \beta^2 \\ &= \|a(k) - \alpha\hat{a}\|^2 - \beta^2 \text{ for } \alpha = \beta^2/\gamma \end{aligned}$$

The algorithm converges since $\|a(k+1) - \alpha\hat{a}\|^2$ decreases at least by the constant β^2 and every error will be corrected. $\square$

Bound on the Number of Update Steps:

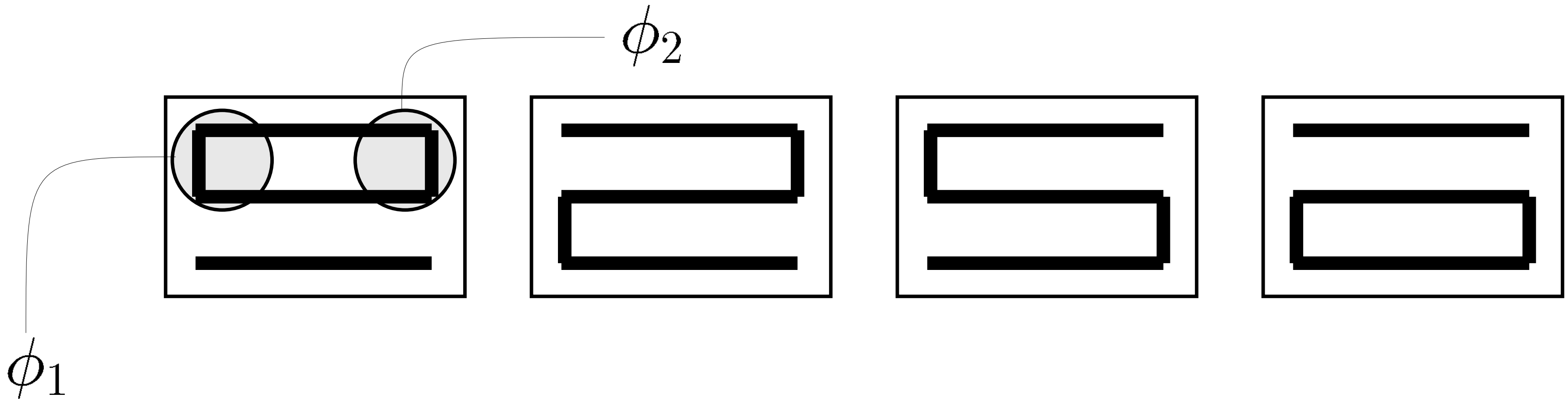
$$\begin{aligned} \|a(k+1) - \alpha\hat{a}\|^2 &\leq \|a(1) - \alpha\hat{a}\|^2 - k\beta^2 = 0 \\ \Rightarrow k_0 &= \frac{\|a(1) - \alpha\hat{a}\|^2}{\beta^2} \end{aligned}$$

Choose $a(1) = 0$ $\quad \Rightarrow k_0 = \dfrac{\alpha^2\|\hat{a}\|^2}{\beta^2} = \dfrac{\beta^2\|\hat{a}\|^2}{\gamma^2} = \dfrac{\max_i \|y_i\|^2\|\hat{a}\|^2}{\min_i(\hat{a}^T y^i)^2}$

Remark: Examples orthogonal to the solution vector are difficult to learn!

Limitations of Single-Layer Perceptrons

(Minsky & Pappert 1969)

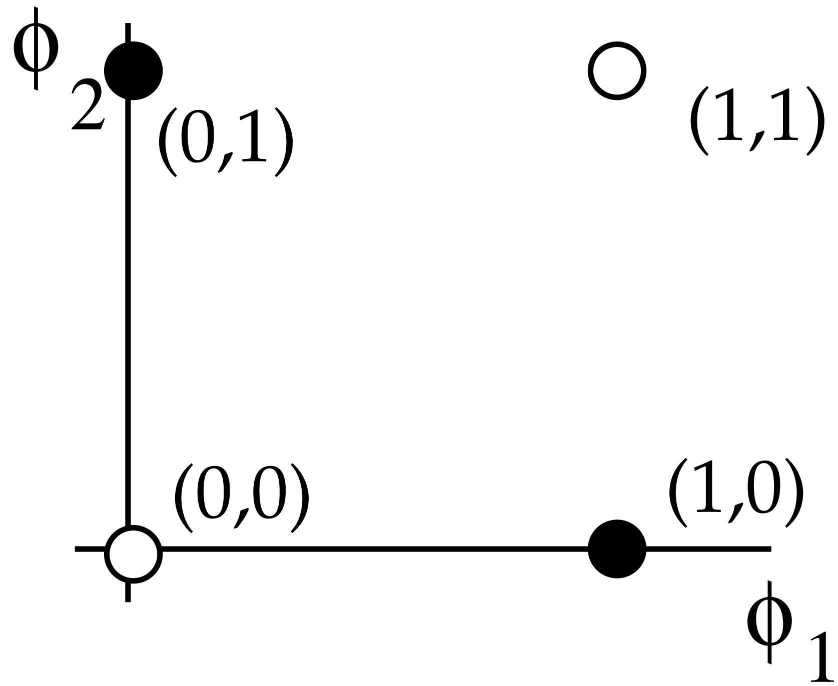


Theorem: A size limited perceptron cannot decide in all cases if parts of a figure are connected or separate.

Proof: The problem is reduced to the XOR problem which is not linearly separable.
ϕ_1, ϕ_2 are detectors which recognize vertical bars in the upper left and right corner.

Truth Table for the connectivity problem.

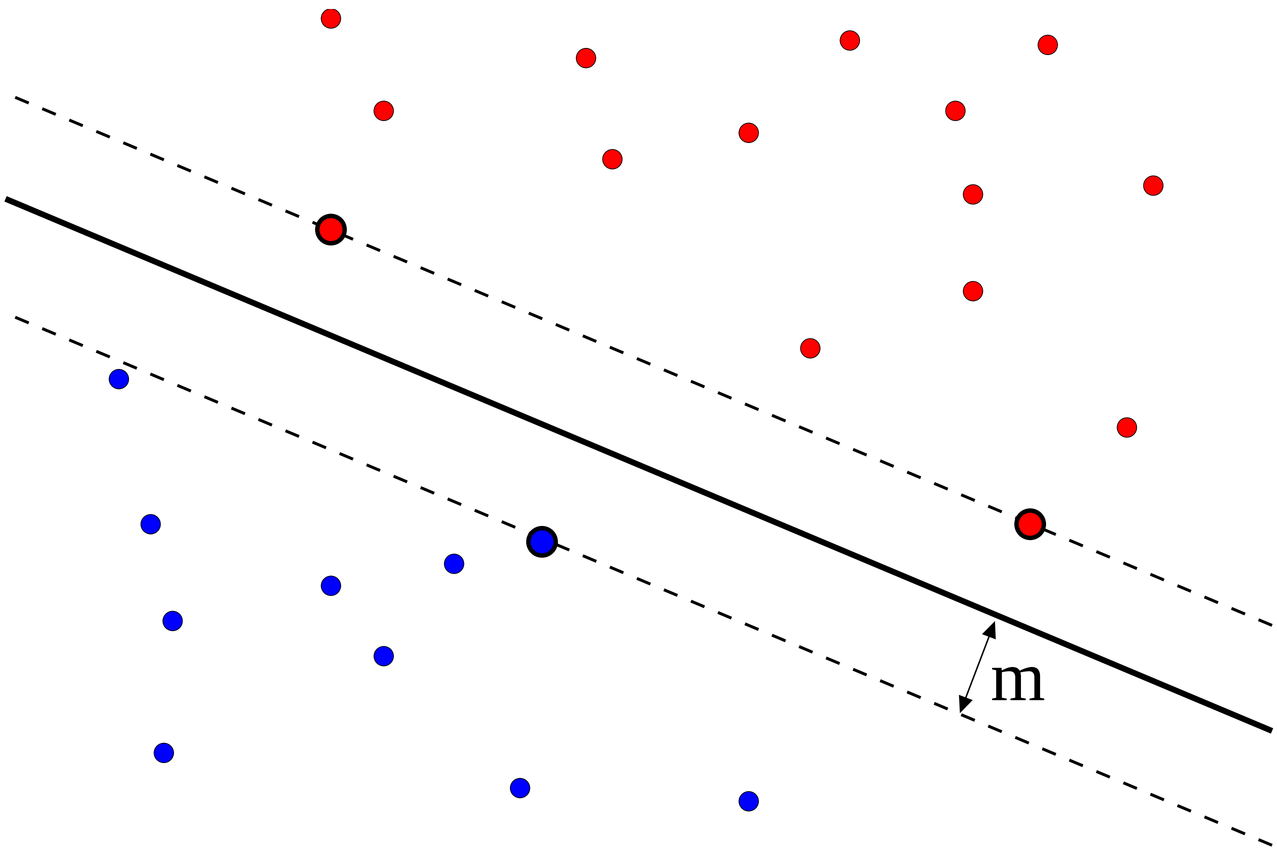
ϕ_1	ϕ_2	y
1	1	0
0	1	1
1	0	1
0	0	0



Simple (single layer) perceptrons can be trained efficiently since classification errors can be “blamed” to components of the weight vector $\mathbf{a}$ in a direct way. *Credit Assignment!*

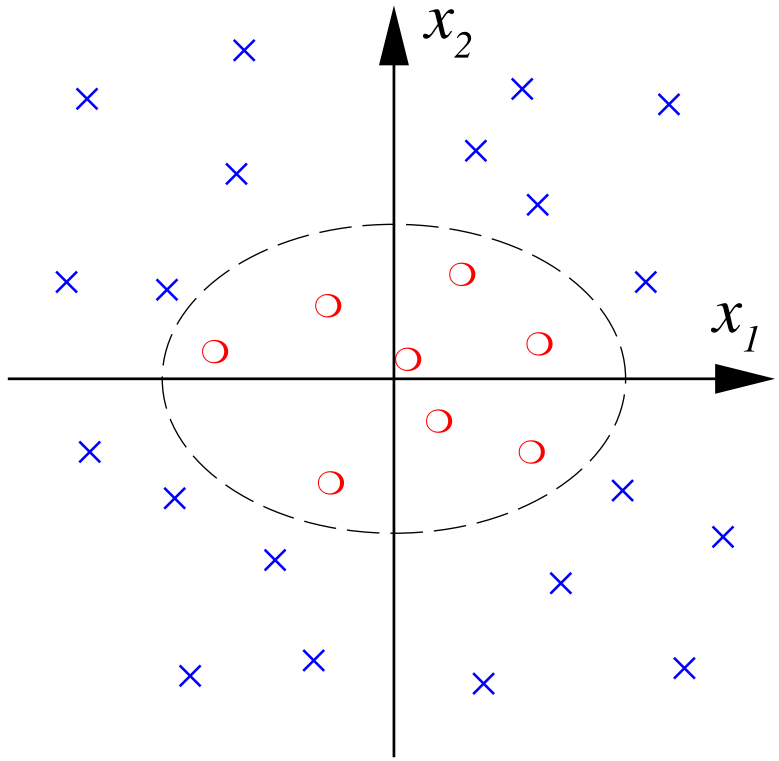
Support Vector Machine (SVM)

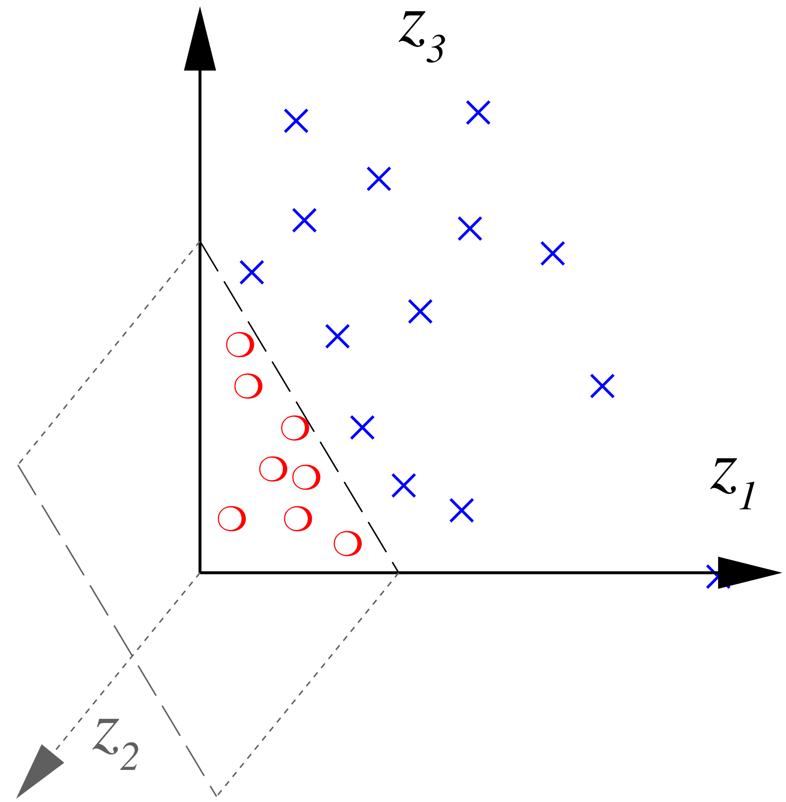
Extending the perceptron idea: use a **linear classifier with margin** and a **non-linear feature transformation**.



Nonlinear Transformation in Kernel Space

$$\Phi : \mathbb{R}^2 \to \mathbb{R}^3$$
$$(x_1, x_2) \mapsto (z_1, z_2, z_3) := (x_1^2, \sqrt{2}\, x_1 x_2, x_2^2)$$





Lagrangian Optimization Theory

Optimization under constraints (Primal Problem):

Given an optimization problem with domain $\Omega \subseteq \mathbb{R}^d$,

$$\begin{array}{rll} \text{minimize} & f(\mathbf{w}), & \mathbf{w} \in \Omega \\ \text{subject to} & g_i(\mathbf{w}) \leq 0, & i = 1, \ldots, k \\ & h_i(\mathbf{w}) = 0, & i = 1, \ldots, m \end{array}$$

The **generalized Lagrangian function** is defined as

$$L(\mathbf{w}, \boldsymbol{\alpha}, \boldsymbol{\beta}) = f(\mathbf{w}) + \sum_{i=1}^{k} \alpha_i g_i(\mathbf{w}) + \sum_{i=1}^{m} \beta_i h_i(\mathbf{w})$$

Kuhn-Tucker Conditions (1951)

Theorem: Given an optimization problem with convex domain $\Omega \subseteq \mathbb{R}^d$,

$$\begin{array}{rll} \text{minimize} & f(\mathbf{w}), & \mathbf{w} \in \Omega \\ \text{subject to} & g_i(\mathbf{w}) \le 0, & i = 1, \ldots, k \\ & h_i(\mathbf{w}) = 0, & i = 1, \ldots, m \end{array}$$

with $f \in C^1$ convex and g_i, h_i affine, necessary and sufficient conditions for a normal point $\mathbf{w}^*$ to be an optimum are the existence of $\boldsymbol{\alpha}^*$, $\boldsymbol{\beta}^*$ such that

$$\frac{\partial L(\mathbf{w}^*, \boldsymbol{\alpha}^*, \boldsymbol{\beta}^*)}{\partial \mathbf{w}} = 0 \qquad \frac{\partial L(\mathbf{w}^*, \boldsymbol{\alpha}^*, \boldsymbol{\beta}^*)}{\partial \boldsymbol{\beta}} = 0$$

$$\alpha_i^* g_i(\mathbf{w}^*) = 0, \quad g_i(\mathbf{w}^*) \le 0, \quad \alpha_i^* \ge 0, \quad i = 1, \ldots, k$$

Support Vector Machines (SVM)

Idea: linear classifier with margin and feature transformation.

Transformation from original feature space to nonlinear feature space.

$$\mathbf{y}_i = \phi(\mathbf{x}_i) \quad \text{e.g. Polynomial, Radial Basis Function, ...}$$

$$\phi : \mathbb{R}^d \to \mathbb{R}^e \text{ with } d \ll e$$

$$z_i = \begin{cases} +1 \\ -1 \end{cases} \text{if } \mathbf{x}_i \text{ in class } \begin{cases} y_1 \\ y_2 \end{cases}$$

Training vectors should be linearly separable after mapping!

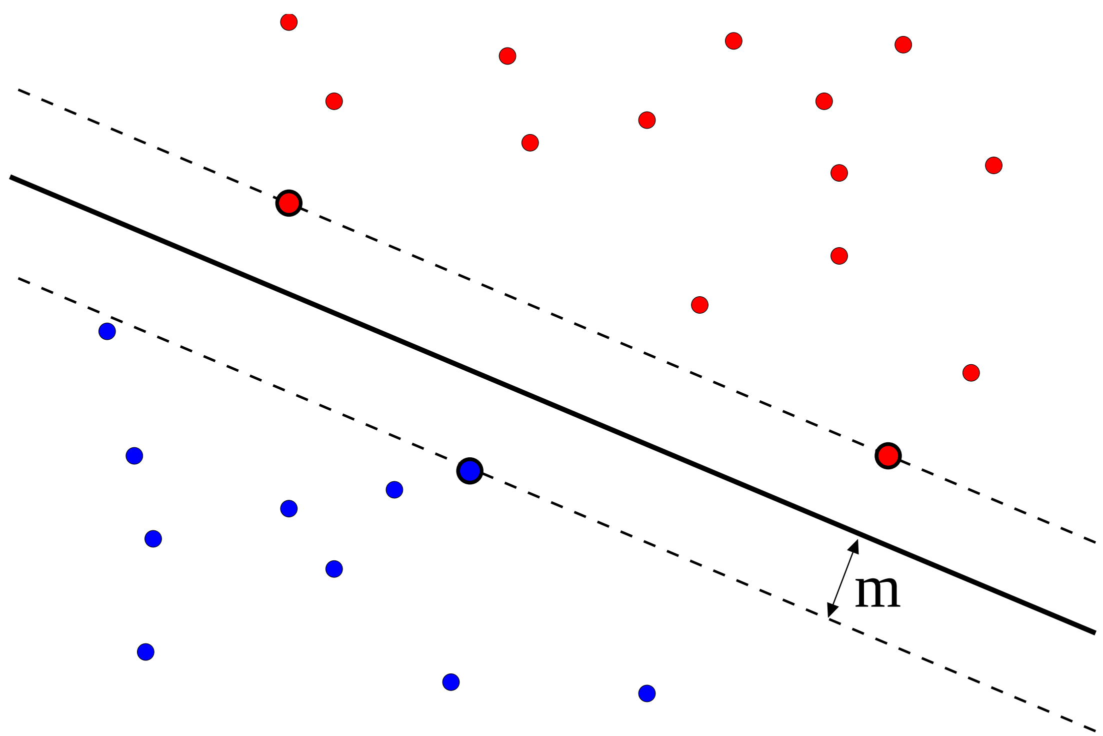
Linear discriminant function:

$$g(\mathbf{y}) = \mathbf{w}^\mathsf{T}\mathbf{y} + w_0$$

Support Vector Machine (SVM)

Find hyperplane that maximizes the **margin** m with

$$z_i\, g(\mathbf{y}_i) = z_i(\mathbf{w}^\mathsf{T}\mathbf{y} + w_0) \geq m \quad \text{for all } \mathbf{y}_i \in \mathcal{Y}$$



Vectors $\mathbf{y}_i$ with $z_i g(\mathbf{y}_i) = m$ are the **support vectors**.

Maximal Margin Classifier

Invariance: assume that the weight vector $\mathbf{w}$ is normalized ($\|\mathbf{w}\| = 1$) since a rescaling $(\mathbf{w}, w_0) \leftarrow (\lambda\mathbf{w}, \lambda w_0), m \leftarrow \lambda m$ does not change the problem.

Condition: $$z_i = \begin{cases} +1 & \mathbf{w}^\mathsf{T}\mathbf{y}_i + w_0 \geq m \\ -1 & \mathbf{w}^\mathsf{T}\mathbf{y}_i + w_0 \leq -m \end{cases} \qquad \forall i$$

Objective: maximize margin m s.t. joint condition $z_i \left(\mathbf{w}^\mathsf{T}\mathbf{y}_i + w_0\right) \geq m$ is met.

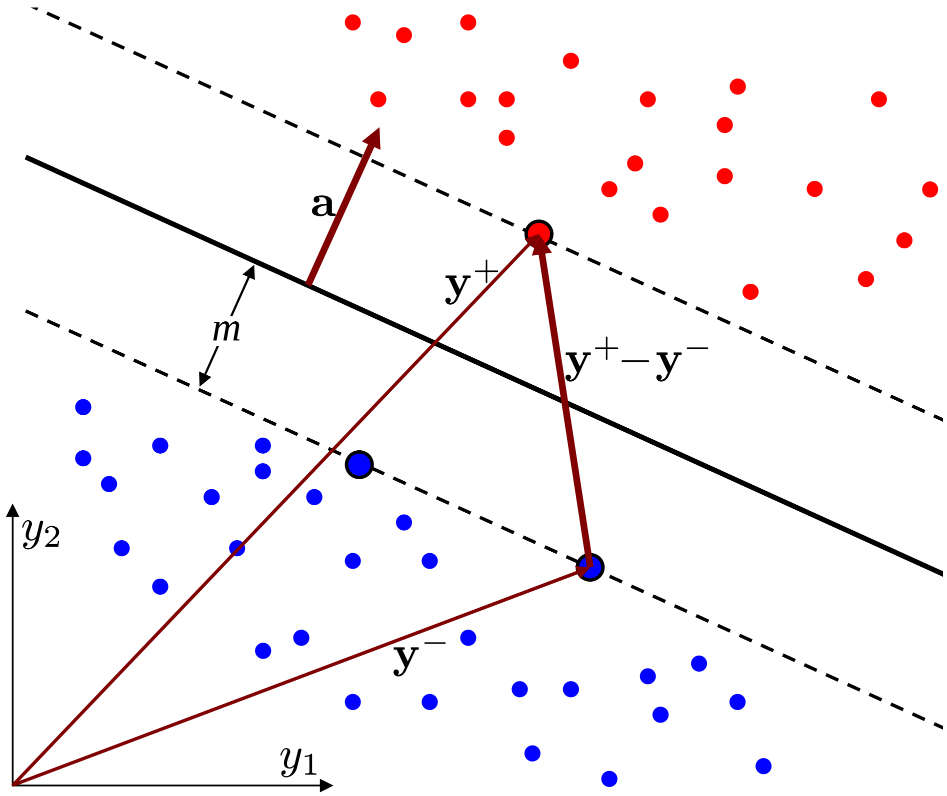
Learning problem: Find $\mathbf{w}$ with $||\mathbf{w}|| = 1$, such that the margin m is maximized.

$$\begin{array}{rl} \text{maximize} & m \\ \text{subject to} & \forall\, \mathbf{y}_i \in \mathcal{Y} : z_i(\mathbf{w}^\mathsf{T}\mathbf{y}_i + w_0) \geq m \end{array}$$

SVM Learning

What is the margin m?

Consider two points $\mathbf{y}^+, \mathbf{y}^-$ of class 1,2 which are located on both sides of the margin boundaries.



Transformation of objective:

rescaling $\mathbf{w} \leftarrow \frac{\mathbf{w}}{m}$, $w_0 \leftarrow \frac{w_0}{m}$ $\Rightarrow$

yields the constraints

$z_i(\mathbf{w}^\mathsf{T}\mathbf{y}_i + w_0) \geq 1$

Margin:

$$m = \frac{1}{2\|\mathbf{w}\|}(\mathbf{w}^\mathsf{T}\mathbf{y}^+ - \mathbf{w}^\mathsf{T}\mathbf{y}^-) = \frac{1}{\|\mathbf{w}\|}$$

$m = \frac{1}{\|\mathbf{w}\|}$ follows from inserting $\pm(\mathbf{w}^\mathsf{T}\mathbf{y}^\pm + w_0) = 1$

$\Rightarrow$ maximizing the margin corresponds to minimizing the norm $\|\mathbf{w}\|$ for margin $m = 1$.

SVM Lagrangian

Minimize $||\mathbf{w}||$ for a given margin $m = 1$

$$\begin{array}{rrcl} \text{minimize} & \mathcal{T}(\mathbf{w}) & = & \frac{1}{2}\mathbf{w}^\mathsf{T}\mathbf{w} \\ \text{subject to} & z_i(\mathbf{w}^\mathsf{T}\mathbf{y}_i + w_0) & \geq & 1 \end{array}$$

Generalized Lagrange Function:

$$L(\mathbf{w}, w_0, \boldsymbol{\alpha}) = \frac{1}{2}\mathbf{w}^\mathsf{T}\mathbf{w} - \sum_{i=1}^{n} \alpha_i \left[z_i(\mathbf{w}^\mathsf{T}\mathbf{y}_i + w_0) - 1 \right]$$

Functional and geometric margin: The problem formulation with margin $m = 1$ is called the *functional margin* setting; The original formulation refers to the *geometric margin*.

Stationarity of Lagrangian

Extremality condition:

$$\frac{\partial L(\mathbf{w}, w_0, \boldsymbol{\alpha})}{\partial \mathbf{w}} = \mathbf{w} - \sum_{i \leq n} \alpha_i z_i \mathbf{y}_i = 0 \quad \Rightarrow \quad \mathbf{w} = \sum_{i \leq n} \alpha_i z_i \mathbf{y}_i$$

$$\frac{\partial L(\mathbf{w}, w_0, \boldsymbol{\alpha})}{\partial w_0} = -\sum_{i \leq n} \alpha_i z_i = 0$$

Resubstituting $\frac{\partial L}{\partial \mathbf{w}} = 0, \frac{\partial L}{\partial w_0} = 0$ into the Lagrangian function $L(\mathbf{w}, w_0, \boldsymbol{\alpha})$

$$\begin{aligned}
L(\mathbf{w}, w_0, \boldsymbol{\alpha}) &= \frac{1}{2}\mathbf{w}^\mathsf{T}\mathbf{w} - \sum_{i \leq n} \alpha_i \left[z_i(\mathbf{w}^\mathsf{T}\mathbf{y}_i + w_0) - 1 \right] \\
&= \frac{1}{2} \sum_{i \leq n} \sum_{j \leq n} \alpha_i \alpha_j z_i z_j \mathbf{y}_i^\mathsf{T} \mathbf{y}_j - \sum_{i \leq n} \sum_{j \leq n} \alpha_i \alpha_j z_i z_j \mathbf{y}_i^\mathsf{T} \mathbf{y}_j + \sum_{i \leq n} \alpha_i \\
&= \sum_{i \leq n} \alpha_i - \frac{1}{2} \sum_{i \leq n} \sum_{j \leq n} \alpha_i \alpha_j z_i z_j \mathbf{y}_i^\mathsf{T} \mathbf{y}_j \qquad \text{(note the scalar product!)}
\end{aligned}$$

Dual Problem

The **Dual Problem** for support vector learning is

$$\text{maximize} \quad W(\boldsymbol{\alpha}) = \sum_{i=1}^{n} \alpha_i - \tfrac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} z_i z_j \alpha_i \alpha_j \mathbf{y}_i^{\mathsf{T}} \mathbf{y}_j$$

$$\text{subject to} \quad \forall i \; \alpha_i \geq 0 \quad \wedge \quad \sum_{i=1}^{n} z_i \alpha_i = 0$$

The optimal hyperplane $\mathbf{w}^*, w_0^*$ is given by

$$\mathbf{w}^* = \sum_{i=1}^{n} \alpha_i^* z_i \mathbf{y}_i, \quad w_0^* = -\frac{1}{2} \left(\min_{i:z_i=1} {\mathbf{w}^*}^{\mathsf{T}} \mathbf{y}_i + \max_{i:z_i=-1} {\mathbf{w}^*}^{\mathsf{T}} \mathbf{y}_i \right)$$

where $\boldsymbol{\alpha}^*$ are the optimal Lagrange multipliers maximizing the Dual Problem.

Support Vectors

The **Kuhn-Tucker Conditions** for the maximal margin SVM are

$$
\begin{aligned}
\alpha_i^*(z_i g^*(\mathbf{y}_i) - 1) &= 0, \qquad i = 1, \dots, n \\
\alpha_i^* &\geq 0, \qquad i = 1, \dots, n \\
z_i g^*(\mathbf{y}_i) - 1 &\geq 0, \qquad i = 1, \dots, n
\end{aligned}
$$

The first one is known as the **Kuhn-Tucker complementary condition**. The conditions imply

$$z_i g^*(\mathbf{y}_i) = 1 \;\Rightarrow\; \alpha_i^* \geq 0 \qquad \text{Support Vectors (SV)}$$

$$z_i g^*(\mathbf{y}_i) \neq 1 \;\Rightarrow\; \alpha_i^* = 0 \qquad \text{Non Support Vectors}$$

Optimal Decision Function

Sparsity:

$$\begin{aligned} g^*(\mathbf{y}) &= \mathbf{w}^{*\mathsf{T}}\mathbf{y} + w_0^* = \sum_{i=1}^{n} z_i \alpha_i^* \mathbf{y}_i{}^{\mathsf{T}}\mathbf{y} + w_0^* \\ &= \sum_{i \in \mathsf{SV}} z_i \alpha_i^* \mathbf{y}_i{}^{\mathsf{T}}\mathbf{y} + w_0^* \end{aligned}$$

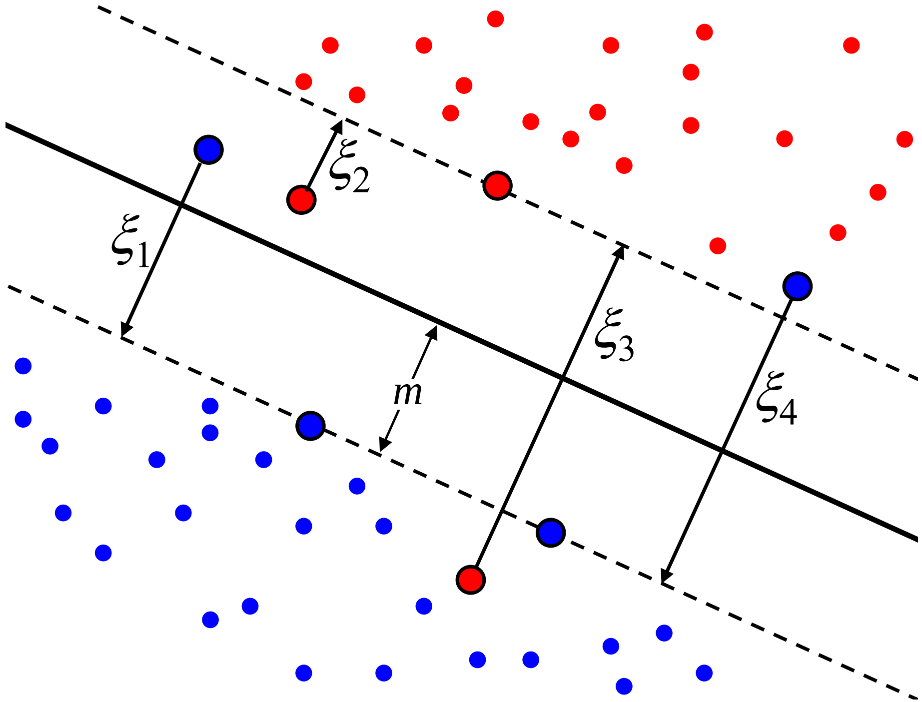
Remark: only few support vectors enter the sum to evaluate the decision function! $\Rightarrow$ efficiency and interpretability

Optimal margin: $\mathbf{w}^{\mathsf{T}}\mathbf{w} = \sum_{i \in \mathsf{SV}} \alpha_i^*$

Soft Margin SVM

For each trainings vector $\mathbf{y}_i \in \mathcal{Y}$ a **slack variable** ξ_i is introduced to measure the violation of the margin constraint.

Find hyperplane that maximizes the margin $z_i g^*(\mathbf{y}_i) \geq m(1-\xi_i)$



Vectors $\mathbf{y}_i$ with $z_i g^*(\mathbf{y}_i) = m(1-\xi_i)$ are called **support vectors**.

Learning the Soft Margin SVM

Slack variables are penalized by L_1 norm.

$$\begin{array}{rrcl} \text{minimize} & \mathcal{T}(\mathbf{w}, \boldsymbol{\xi}) & = & \frac{1}{2}\mathbf{w}^{\mathsf{T}}\mathbf{w} + C\sum_{i=1}^{n}\xi_i \\ \text{subject to} & z_i(\mathbf{w}^{\mathsf{T}}\mathbf{y}_i + w_0) & \geq & 1 - \xi_i \\ & \xi_i & \geq & 0 \end{array}$$

C controls the amount of constraint violations vs. margin maximization!

Lagrange function for soft margin SVM

$$\begin{aligned} L(\mathbf{w}, w_0, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\beta}) \;&=\; \frac{1}{2}\mathbf{w}^{\mathsf{T}}\mathbf{w} + C\sum_{i=1}^{n}\xi_i \\ &-\; \sum_{i=1}^{n}\alpha_i\left[z_i(\mathbf{w}^{\mathsf{T}}\mathbf{y}_i + w_0) - 1 + \xi_i\right] - \sum_{i=1}^{n}\beta_i\xi_i \end{aligned}$$

Stationarity of Primal Problem

Differentiation:

$$\frac{\partial L(\mathbf{w}, w_0, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\beta})}{\partial \mathbf{w}} = \mathbf{w} - \sum_{i=1}^{n} \alpha_i z_i \mathbf{y}_i = 0 \quad \Rightarrow \quad \mathbf{w} = \sum_{i=1}^{n} \alpha_i z_i \mathbf{y}_i$$

$$\frac{\partial L(\mathbf{w}, w_0, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\beta})}{\partial \xi_i} = C - \alpha_i - \beta_i = 0 \quad \frac{\partial L(\mathbf{w}, w_0, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\beta})}{\partial b} = -\sum_{i=1}^{n} \alpha_i z_i = 0$$

Resubstituting into the Lagrangian function $L(\mathbf{w}, w_0, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\beta})$

$$\begin{aligned} L(\mathbf{w}, w_0, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\beta}) &= \tfrac{1}{2}\mathbf{w}^{\mathsf{T}}\mathbf{w} + C \textstyle\sum_{i=1}^{n} \xi_i \\ &- \textstyle\sum_{i=1}^{n} \alpha_i \left[z_i(\mathbf{w}^{\mathsf{T}}\mathbf{y}_i + w_0) - 1 + \xi_i \right] - \textstyle\sum_{i=1}^{n} \beta_i \xi_i \end{aligned}$$

$$
\begin{aligned}
L(\mathbf{w}, w_0, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\beta}) &= \tfrac{1}{2} \textstyle\sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j z_i z_j \mathbf{y}_i^\mathsf{T} \mathbf{y}_j + C \sum_{i=1}^n \xi_i \\
&\quad - \textstyle\sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j z_i z_j \mathbf{y}_i^\mathsf{T} \mathbf{y}_j \\
&\quad + \textstyle\sum_{i=1}^n \alpha_i (1 - \xi_i) - \sum_{i=1}^n \beta_i \xi_i \\
&= \textstyle\sum_{i=1}^n \alpha_i - \tfrac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j z_i z_j \mathbf{y}_i^\mathsf{T} \mathbf{y}_j \\
&\quad + \textstyle\sum_{i=1}^n (\underbrace{C - \alpha_i - \beta_i}_{= \frac{\partial L}{\partial \xi_i} = 0}) \xi_i \\
&= \textstyle\sum_{i=1}^n \alpha_i - \tfrac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j z_i z_j \mathbf{y}_i^\mathsf{T} \mathbf{y}_j
\end{aligned}
$$

Constaints of the Dual Problem

The dual objective function is the same as for the maximal margin SVM. The only difference is the constraint

$$C - \alpha_i - \beta_i = 0$$

Together with $\beta_i \geq 0$ it implies

$$\alpha_i \leq C$$

The Kuhn-Tucker complementary conditions

$$\begin{aligned} \alpha_i(z_i(\mathbf{w}^\mathsf{T}\mathbf{y}_i + w_0) - 1 + \xi_i) &= 0, \qquad i = 1, \ldots, n \\ \xi_i(\alpha_i - C) &= 0, \qquad i = 1, \ldots, n \end{aligned}$$

imply that nonzero slack variables can only occur when $\alpha_i = C$.

Dual Problem of Soft Margin SVM

The **Dual Problem** for support vector learning is

$$\begin{array}{rl} \text{maximize} & W(\boldsymbol{\alpha}) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} z_i z_j \alpha_i \alpha_j \mathbf{y}_i^{\mathsf{T}} \mathbf{y}_j \\ \text{subject to} & \sum_{j=1}^{n} z_j \alpha_j = 0 \;\wedge\; \forall i \; C \;\geq \alpha_i \geq 0 \end{array}$$

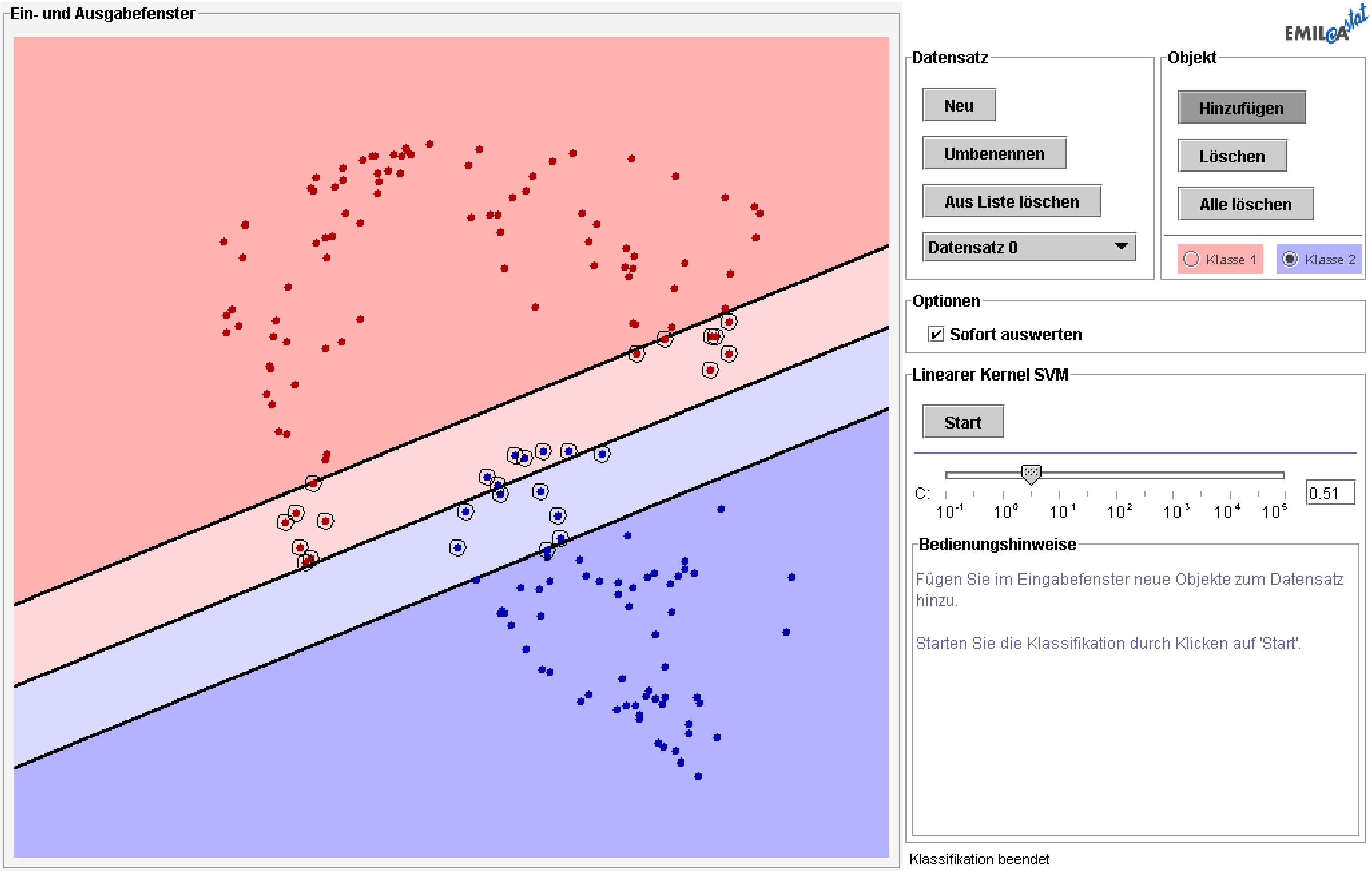
The optimal hyperplane $\mathbf{w}^*$ is given by

$$\mathbf{w}^* = \sum_{i=1}^{n} \alpha_i^* z_i \mathbf{y}_i$$

where $\boldsymbol{\alpha}^*$ are the optimal Lagrange multipliers maximizing the Dual Problem.

Only for **support vectors** it holds $\alpha_i^* > 0$

Applet HTML Page



Linear Programming Support Vector Machines

Idea: Minimize an estimate of the number of positive multipliers $\sum_{i=1}^{n} \alpha_i$ which improves bounds on the generalization error.

The **Lagrangian** for the LP-SVM is

$$\begin{aligned} \text{minimize} \quad & W(\boldsymbol{\alpha}, \xi) = \sum_{i=1}^{n} \alpha_i + C \sum_{i=1}^{n} \xi_i \\ \text{subject to} \quad & z_i \left[\sum_{j=1}^{n} \alpha_j \mathbf{y}_i^{\mathsf{T}} \mathbf{y}_j + w_0 \right] \geq 1 - \xi_i, \\ & \alpha_i \geq 0,\ \xi_i \geq 0,\ 1 \leq i \leq n \end{aligned}$$

Advantage: efficient LP solver can be used.

Disadvantage: theory is not as well understood as for standard SVMs.

Non–Linear SVMs

Feature extraction by non linear transformation $\mathbf{y} = \phi(\mathbf{x})$

Problem:

$$\mathbf{y}_i^{\mathsf{T}} \mathbf{y}_j = \phi^{\mathsf{T}}(\mathbf{x}_i)\phi(\mathbf{x}_j)$$

is the inner product in a high dimensional space.

A **kernel function** is defined by

$$\forall \mathbf{x}, \mathbf{z} \in \Omega : \quad K(\mathbf{x}, \mathbf{z}) = \phi^{\mathsf{T}}(\mathbf{x})\phi(\mathbf{z})$$

Using the kernel function the discriminant function becomes

$$f(\mathbf{x}) = \sum_{i=1}^{n} \alpha_i z_i K(\mathbf{x}_i, \mathbf{x})$$

Characterization of Kernels

For a symmetric matrix $K(\mathbf{x}_i, \mathbf{x}_j)|_{i,j=1}^{n}$ (Gram matrix) there exists an EV decomposition

$$K = V \Lambda V^{\mathsf{T}}$$

V: orthogonal matrix of eigenvectors $(v_{ti})|_{i=1}^{n}$
Λ: diagonal matrix of eigenvalues λ_t

Assume all eigenvalues are nonnegative and consider mapping

$$\phi : \mathbf{x}_i \to \left(\sqrt{\lambda_t} v_{ti} \right)_{t=1}^{n} \in \mathbb{R}^n, i = 1, \ldots, n$$

Then it follows

$$\phi^{\mathsf{T}}(\mathbf{x}_i)\phi(\mathbf{x}_j) = \sum_{t=1}^{n} \lambda_t v_{ti} v_{tj} = \left(V \Lambda V^{\mathsf{T}} \right)_{ij} = K_{ij} = K(\mathbf{x}_i, \mathbf{x}_j)$$

Positivity of Kernels

Theorem: Let Ω be a finite input space with $K(\mathbf{x}, \mathbf{z})$ a symmetric function on Ω. Then $K(\mathbf{x}, \mathbf{z})$ is a kernel function if and only if the matrix

$$K = (K(\mathbf{x}_i, \mathbf{x}_j))_{i,j=1}^n$$

is *positive semi-definite* (has only non-negative eigenvalues).

Extension to infinite dimensional Hilbert Spaces:

$$< \phi(\mathbf{x}), \phi(\mathbf{z}) >= \sum_{i=1}^{\infty} \lambda_i \phi_i(\mathbf{x}) \phi_i(\mathbf{z})$$

Mercer's Theorem

Theorem (Mercer): Let Ω be a compact subset of $\mathbb{R}^n$. Suppose K is a continous symmetric function such that the integral operator $T_k : L_2(X) \to L_2(X)$,

$$(T_k f)(\cdot) = \int_\Omega K(\cdot, \mathbf{x}) f(\mathbf{x}) d\mathbf{x},$$

is positive, that is

$$\int_{\Omega\times\Omega} K(\mathbf{x}, \mathbf{z}) f(\mathbf{x}) f(\mathbf{z}) d\mathbf{x} d\mathbf{z} > 0 \quad \forall\, f \in L_2(\Omega)$$

Then we can expand $K(\mathbf{x}, \mathbf{z})$ in a uniformly convergent series in terms of T_k's eigen-functions $\phi_j \in L_2(\Omega)$, with $||\phi_j||_{L_2} = 1$ and $\lambda_j > 0$.

Possible Kernels

Remark: Each kernel function, that hold Mercer's conditions describes an inner product in a high dimensional space. The kernel function replaces the inner product.

Possible Kernels:

$$a) \quad K(\mathbf{x}, \mathbf{z}) \;=\; \exp\left(-\frac{||\mathbf{x}-\mathbf{z}||^2}{2\sigma^2}\right) \qquad \text{(RBF Kernel)}$$

$$b) \quad K(\mathbf{x}, \mathbf{z}) \;=\; \tanh \kappa\mathbf{x}\mathbf{z} - b \qquad \text{(Sigmoid Kernel)}$$

$$c) \quad K(\mathbf{x}, \mathbf{z}) \;=\; (\mathbf{x}\mathbf{z})^d \qquad \text{(Polynomial Kernel)}$$

$$K(\mathbf{x}, \mathbf{z}) \;=\; (\mathbf{x}\mathbf{z}+1)^d$$

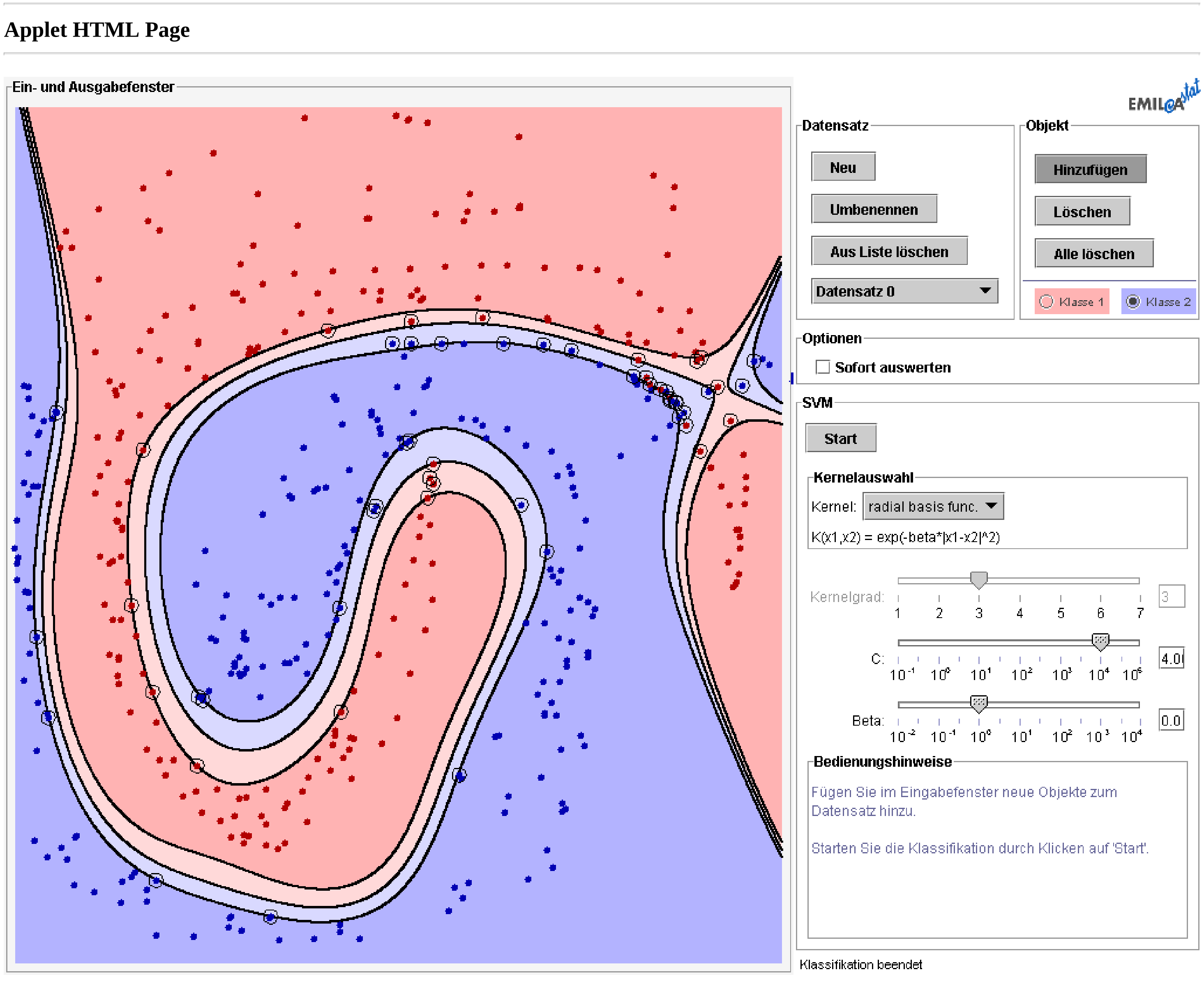
$$d) \quad K(\mathbf{x}, \mathbf{z}) \;:\; \text{string kernels for sequences}$$

Kernel Engineering

Kernel composition rules: Let K_1, K_2 be kernels over $\Omega \times \Omega, \Omega \subseteq \mathbb{R}^d$, $a \in \mathbb{R}^+, f(.)$ a real-vealued function $\phi : \Omega \to \mathbb{R}^e$ with K_3 a kernel over $\mathbb{R}^e \times \mathbb{R}^e$.

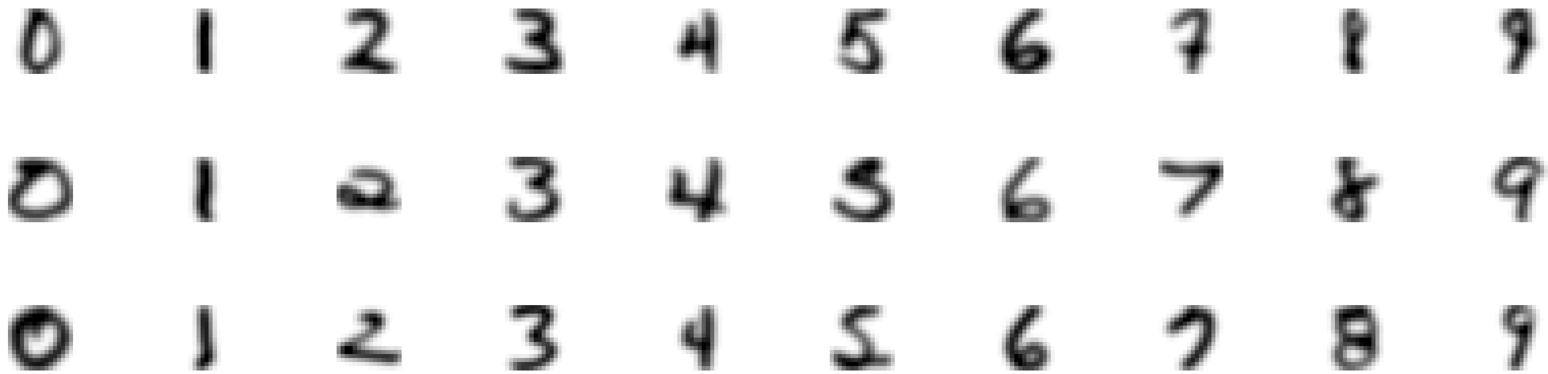
Then the following functions are kernels:

1. $K(\mathbf{x}, \mathbf{z}) = K_1(\mathbf{x}, \mathbf{z}) + K_2(\mathbf{x}, \mathbf{z})$,
2. $K(\mathbf{x}, \mathbf{z}) = aK_1(\mathbf{x}, \mathbf{z})$,
3. $K(\mathbf{x}, \mathbf{z}) = K_1(\mathbf{x}, \mathbf{z})K_2(\mathbf{x}, \mathbf{z})$,
4. $K(\mathbf{x}, \mathbf{z}) = f(\mathbf{x})f(\mathbf{z})$,
5. $K(\mathbf{x}, \mathbf{z}) = K_3(\phi(\mathbf{x}), \phi(\mathbf{z}))$,
6. $K(\mathbf{x}, \mathbf{z}) = p(K_1(\mathbf{x}, \mathbf{z}))$, ($p(x)$ is a polynomial with positive co-efficients)
7. $K(\mathbf{x}, \mathbf{z}) = \exp(K_1(\mathbf{x}, \mathbf{z}))$,


Applet HTML Page
Ein- und Ausgabefenster
Datensatz
Neu
Umbenennen
Aus Liste löschen
Datensatz 0
Objekt
Hinzufügen
Löschen
Alle löschen
Klasse 1
Klasse 2
Optionen
Sofort auswerten
SVM
Start
Kernelauswahl
Kernel: radial basis func.
K(x1,x2) = exp(-beta*|x1-x2|^2)
Kernelgrad:
1 2 3 4 5 6 7
C:
4.0
Beta:
0.0
Bedienungshinweise
Fügen Sie im Eingabefenster neue Objekte zum Datensatz hinzu.
Starten Sie die Klassifikation durch Klicken auf 'Start'.
Klassifikation beendet

Example: Hand Written Digit Recognition

- 7291 training images und 2007 test images (16x16 pixel, 256 gray values)



Classification method	test error
human classification	2.7 %
perceptron	5.9 %
support vector machines	4.0 %