
One-class Support Vector Machines

Seminar Wissensbasierte Systeme

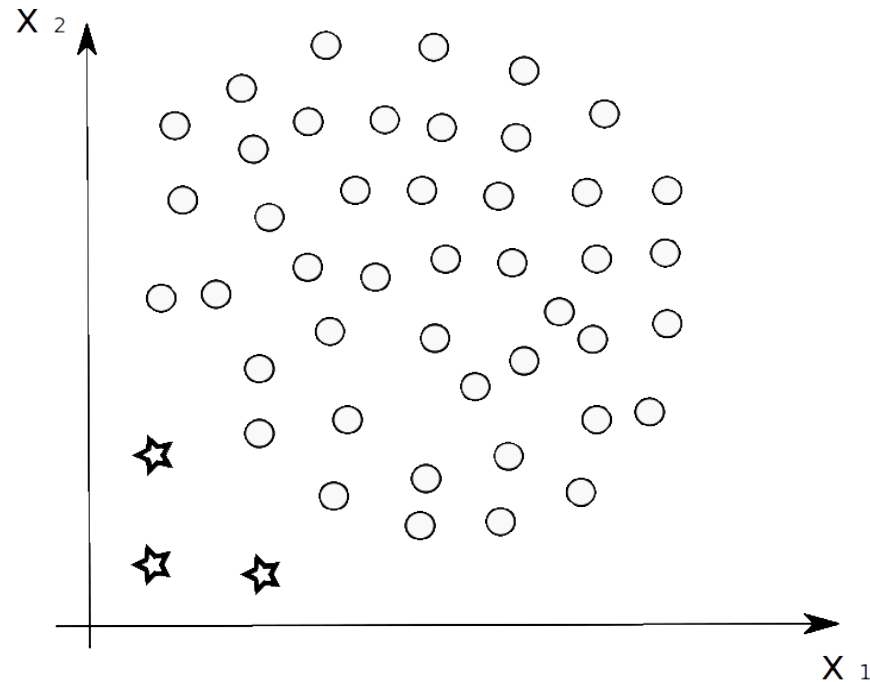
Dietrich Derksen

31. Januar 2014



Motivation

- One-class Support Vector Machines:
 - Detektion von Ausreißern (Systemfehlererkennung)
 - Klassifikation unterrepräsentierter Klassen
 - Vergleich von Datenmengen (Sprechererkennung)



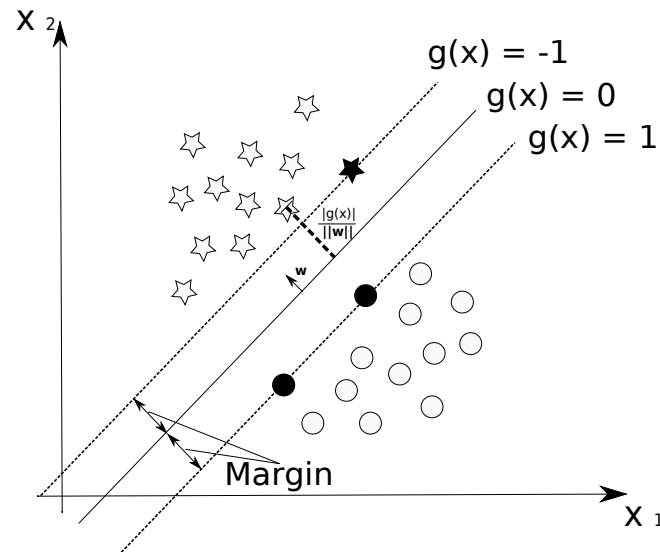
Überblick

- Support Vector Machines
 - Zweiklassenproblem mit linear separierbaren Trainingsdaten
 - Erweiterung auf das Mehrklassenproblem
 - Nicht linear trennbare Trainingsdaten
 - Soft Margin
 - Kernel Trick
- One-class Support Vector Machines
 - ν -Support Vector Classifier (ν -SVC)
 - SVDD
- Zusammenfassung



Support Vector Machines

- **Gegeben:** Trainingsdaten $x = [x_1, \dots, x_k]$ im mehrdimensionalen Vektorraum aus zwei verschiedenen Klassen (Abstand der Vektoren als Ähnlichkeitsmaß)
- **Ziel:** Lineare Diskriminante mit maximalem Abstand zu den am nächsten liegenden Merkmalsvektoren zu bestimmen



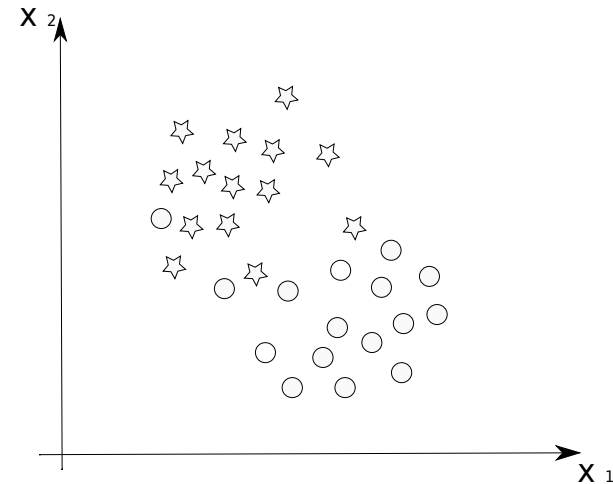
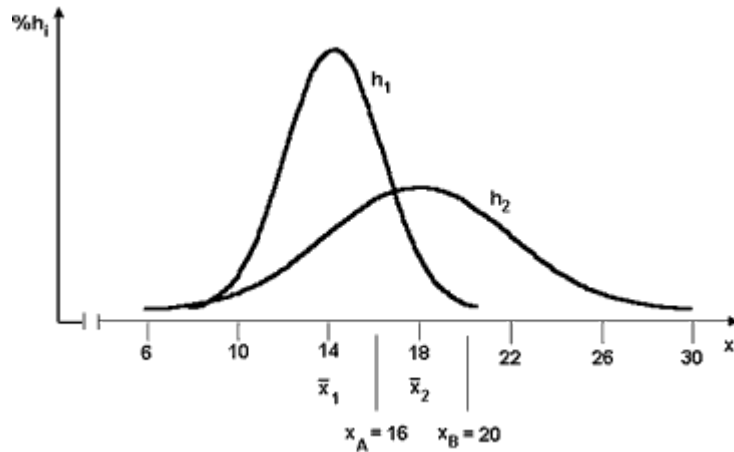
Mehrklassenproblem

- **One-versus-All:** K Diskriminanten zwischen der k -ten Klasse und allen anderen Klassen
 - Datum gehört zur k -ten Klasse wenn die Diskriminante für diese Klasse den maximalen Wert liefert
- **One-versus-One:** $K(K-1)/2$ Diskriminanten zwischen jeweils zwei Klassen
 - **Abstimmungsprinzip:**
 - Vergleich der $K(K-1)/2$ Diskriminanten
 - Klasse mit den meisten Stimmen gewinnt



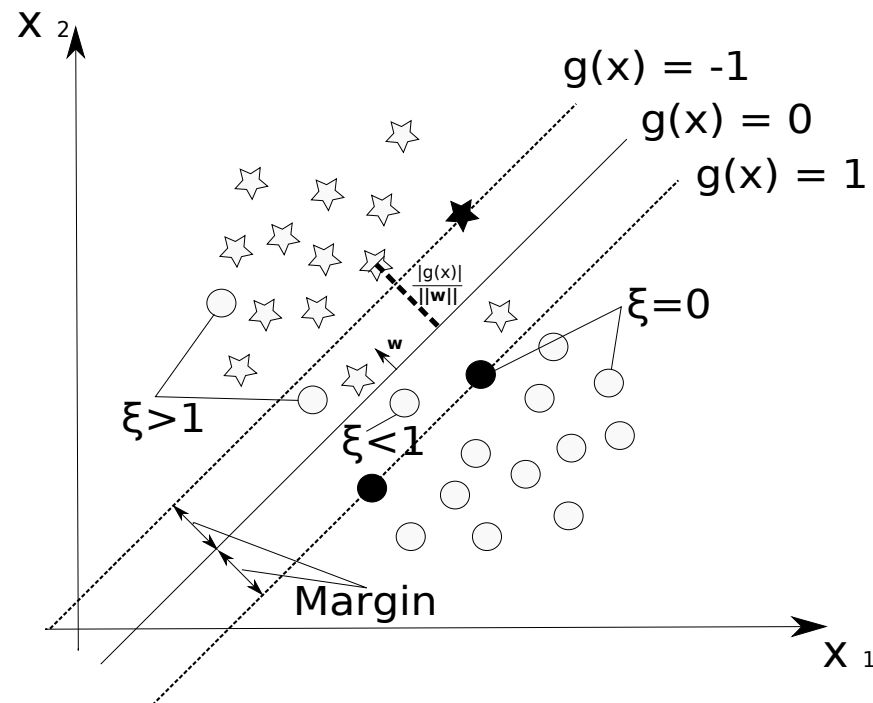
Nicht linear separierbare Trainingsdaten

- Trainingsdaten nur durch eine nichtlineare Hyperebene trennbar



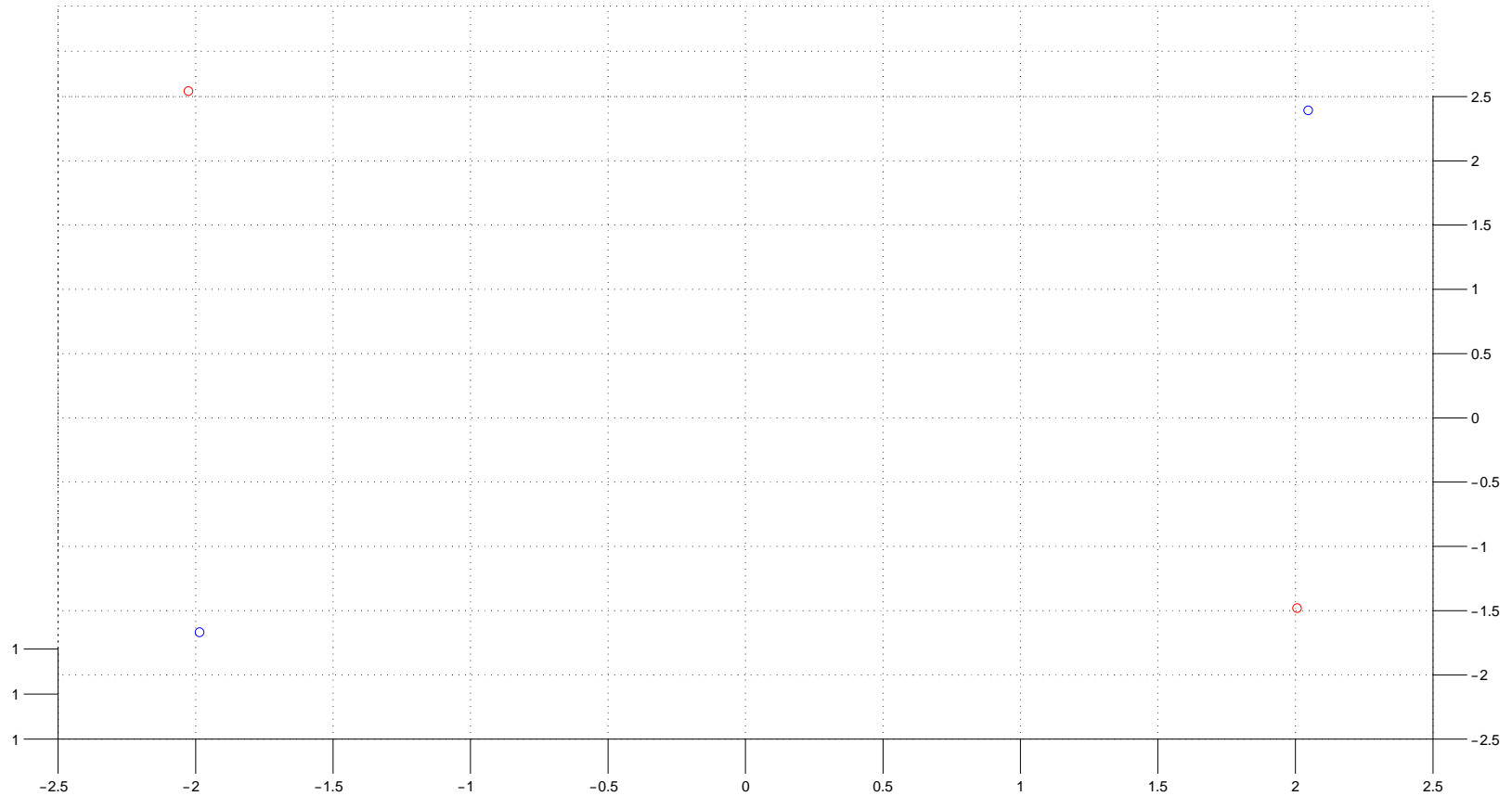
Soft Margin

- Einige Merkmalsvektoren können im Margin liegen
- Robuster gegen Overfitting



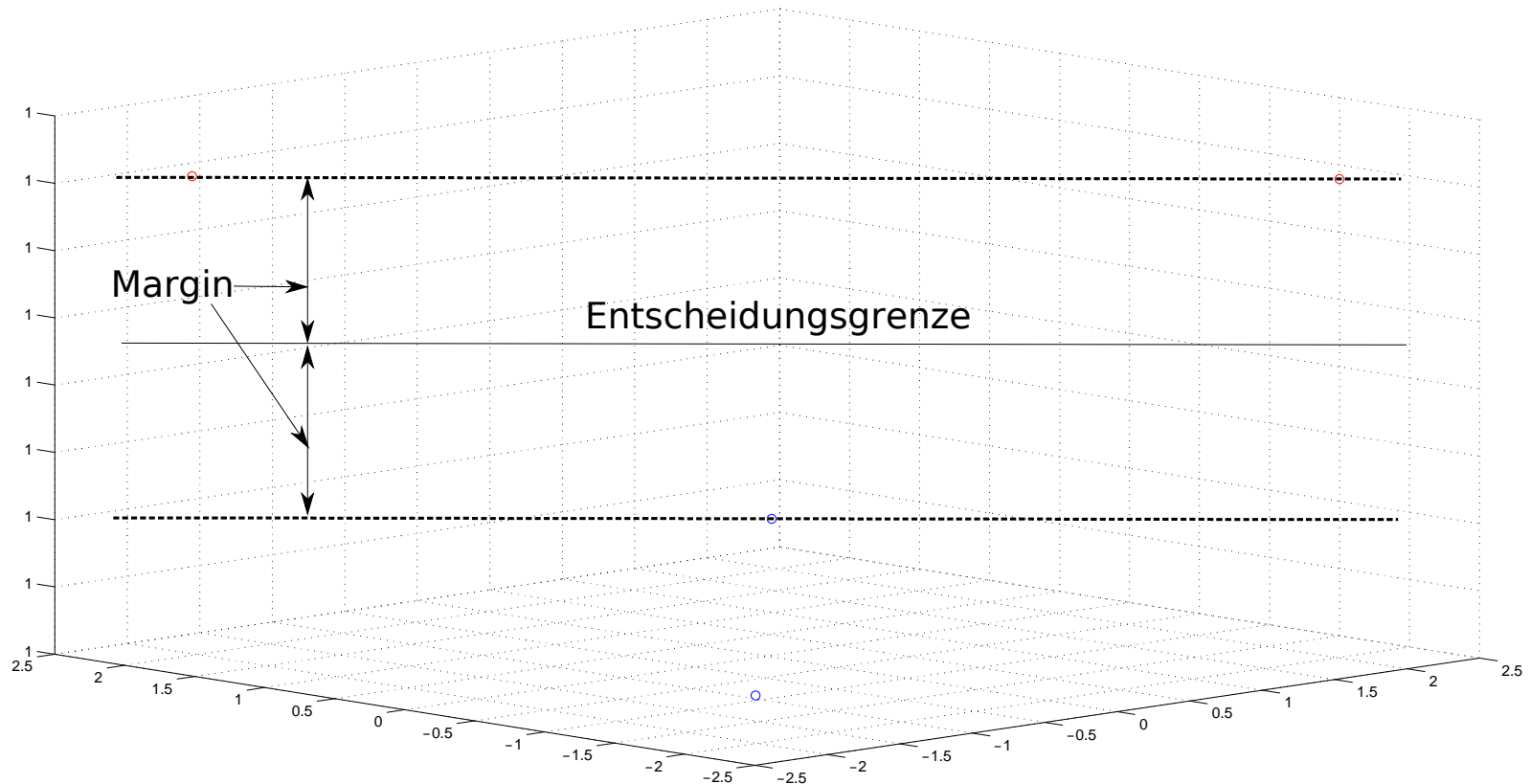
Kernel Trick

- Exclusive-Or Problem: Linear nicht trennbar



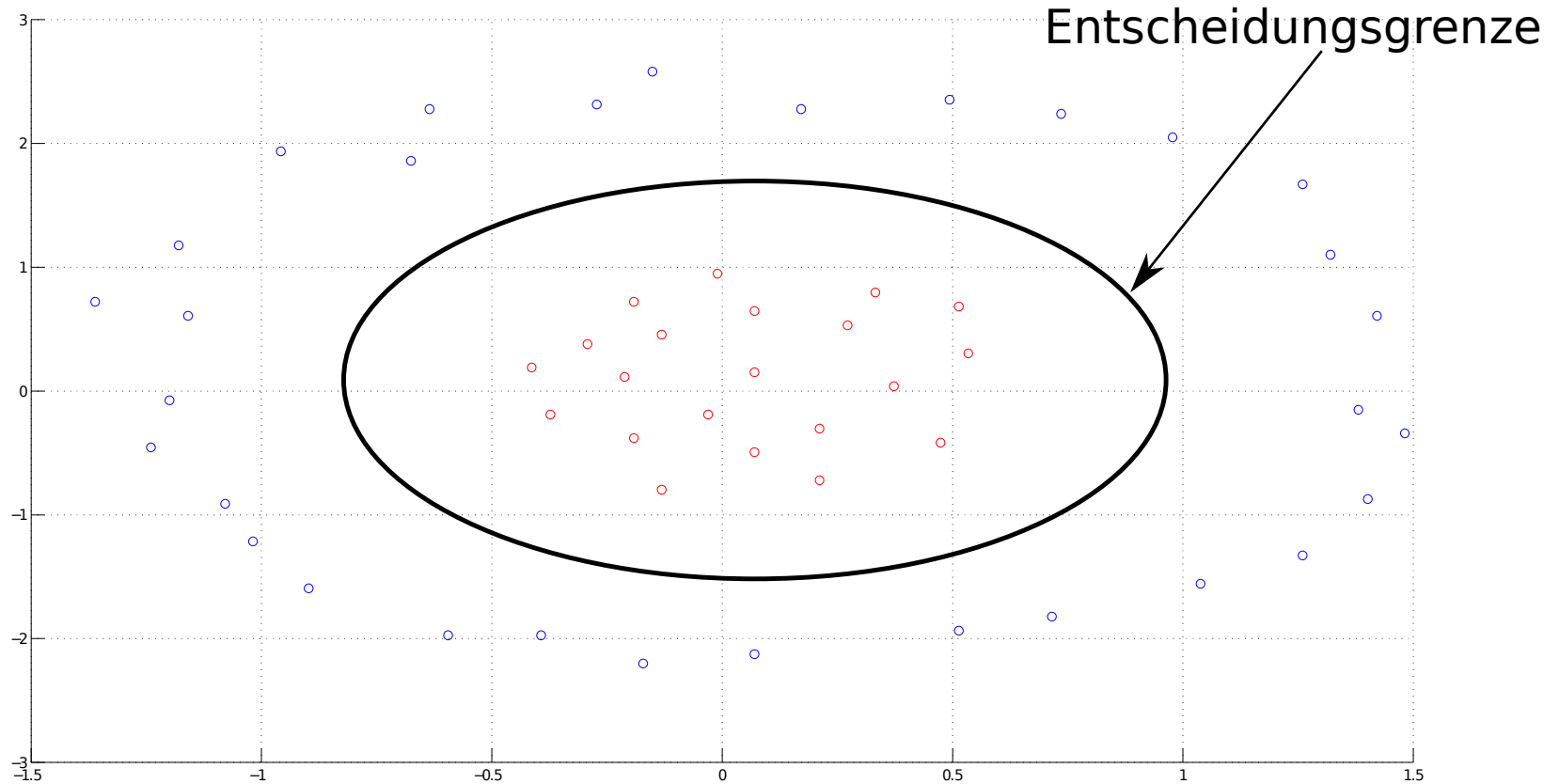
Kernel Trick (Exclusive-Or Problem)

- Exclusive-Or Problem: nach Transformation vom zweidimensionalen in den dreidimensionalen Vektorraum



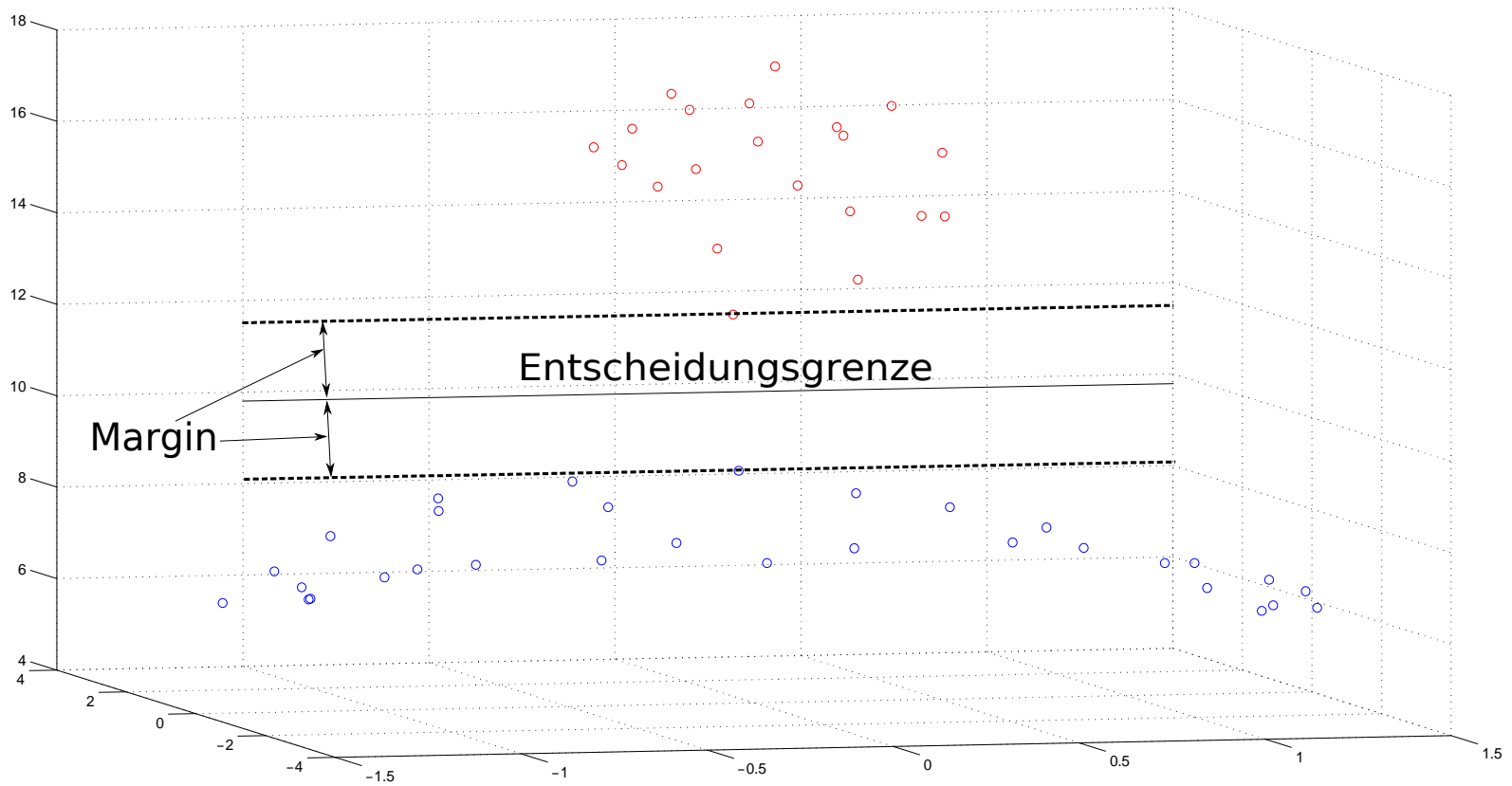
Kernel Trick

- Kreisförmig angeordnete Daten



Kernel Trick

- Transformation der kreisförmig angeordneten Daten in den dreidimensionalen Vektorraum



Kernel Trick

- Transformation: $\phi(x) = (x, x^2)$
- Kernel (von $D = 2$ nach $D = 3$):

$$\begin{aligned} k(x_n, x_m) &= (x_n^T x_m)^2 = \sum_{l=1}^N \sum_{k=1}^N x_{nl} x_{ml} x_{nk} x_{mk} = \\ &= \begin{pmatrix} x_{n1}^2 \\ \sqrt{2} x_{n1} x_{n2} \\ x_{n2}^2 \end{pmatrix}^T \begin{pmatrix} x_{m1}^2 \\ \sqrt{2} x_{m1} x_{m2} \\ x_{m2}^2 \end{pmatrix} = \phi(x_n)^T \phi(x_m) \end{aligned}$$



Kernel Funktionen

- Polynomialkernel:

$$k(x_n, x_m) = (a x_n^T x_m + c)^p$$

- Radial Basis Function-Kernel (RBF):

$$k(x_n, x_m) = e^{-\|x_n - x_m\|^2 / (2\sigma^2)}$$

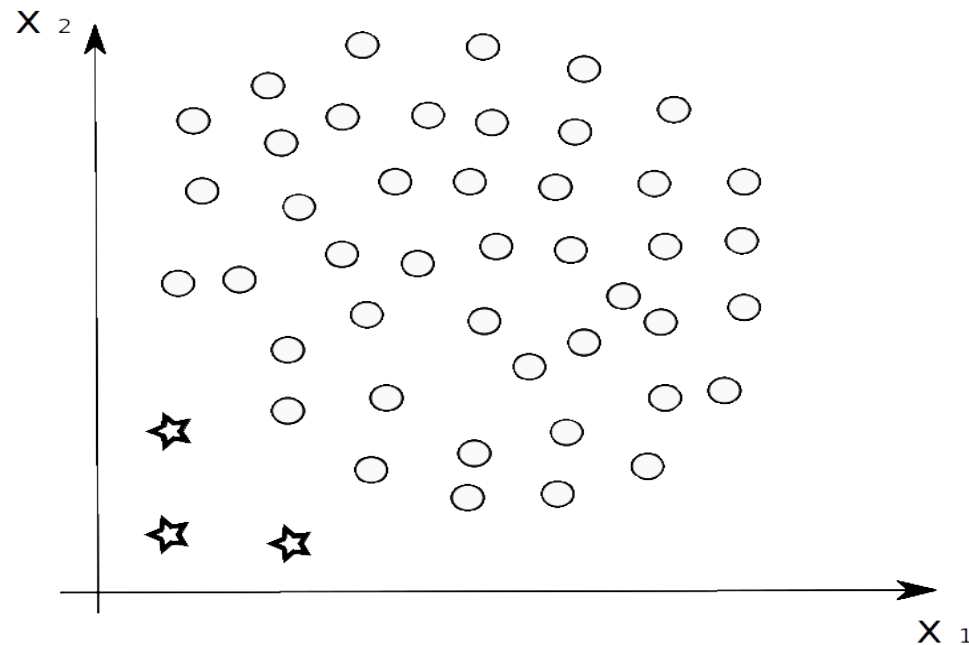
- Sigmoid Kernel:

$$k(x_n, x_m) = \tanh(a x_n^T x_m + c)$$



One-class Support Vector Machines

- Zwei Verfahren:
 - ν -Support Vector Classifier nach Schölkopf et al. (1999)
 - Support Vector Data Description (SVDD) nach Tax und Duin (2004)

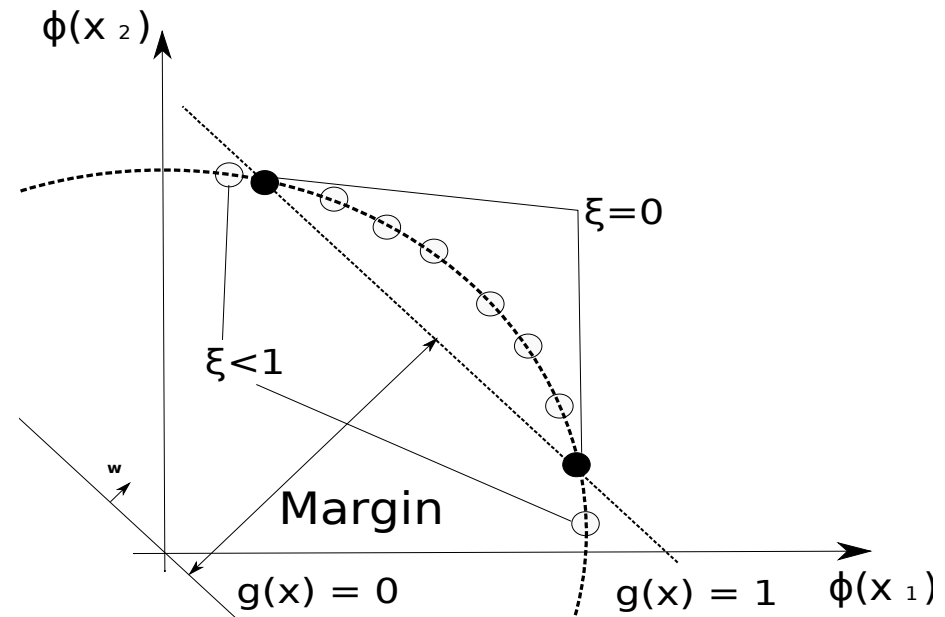


ν -SVC

- Maximiert den Abstand der Trainingsdaten zum Ursprung
- **Soft Margin:**
 - Erlaubt einige Ausreißer in den Trainingsdaten
 - Robuster gegenüber Overfitting

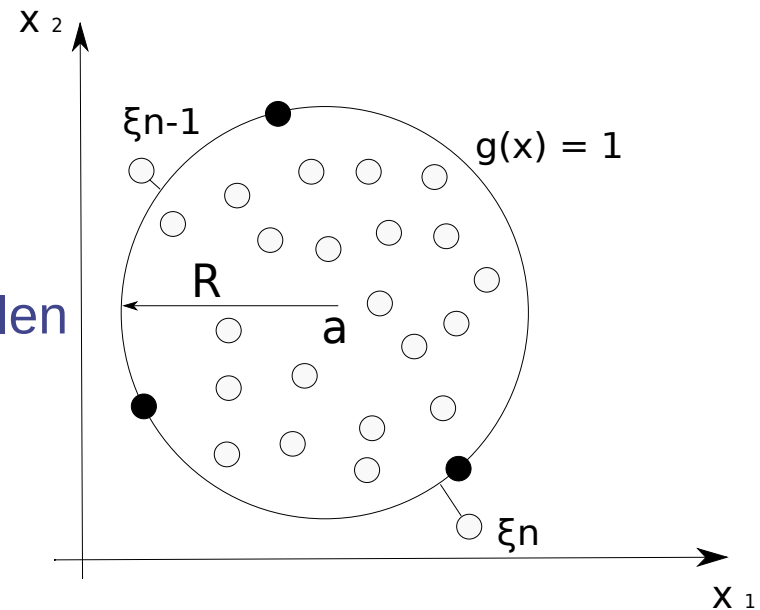
- **Kernel Trick:**

- Transformation der Trainingsdaten auf eine Hypersphäre für $k(x_n, x_m) = \text{const}$
- Vergrößert Abstand zum Ursprung



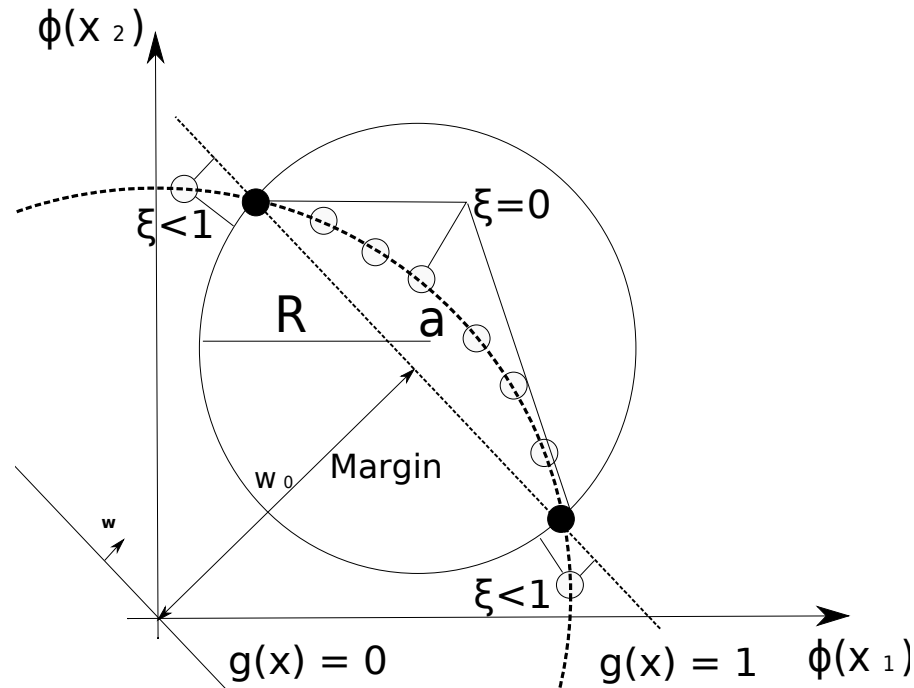
SVDD

- Minimiert das Volumen eines Hyperballs um die Trainingsdaten
- **Soft Margin:**
 - Erlaubt einige Ausreißer in den Trainingsdaten
 - Robuster gegenüber Overfitting
- **Kernel Trick:**
 - Beschreibung der Trainingsdaten durch Hyperball im höherdimensionalen Merkmalsraum entspricht einer Hyperwolke im Ursprungsraum



Vergleich SVDD und ν -SVC

- SVDD Lösung äquivalent mit ν -SVC wenn ein translationsinvarianter Kernel wie z.B. der RBF-Kernel verwendet wird



Zusammenfassung

- Support Vector Machines
- One-versus-One, One-versus-All
- Soft-Margin-SVM
- Kernel Trick
- One-class Support Vector Machines
 - ν -Support Vector Classifier (ν -SVC)
 - SVDD

