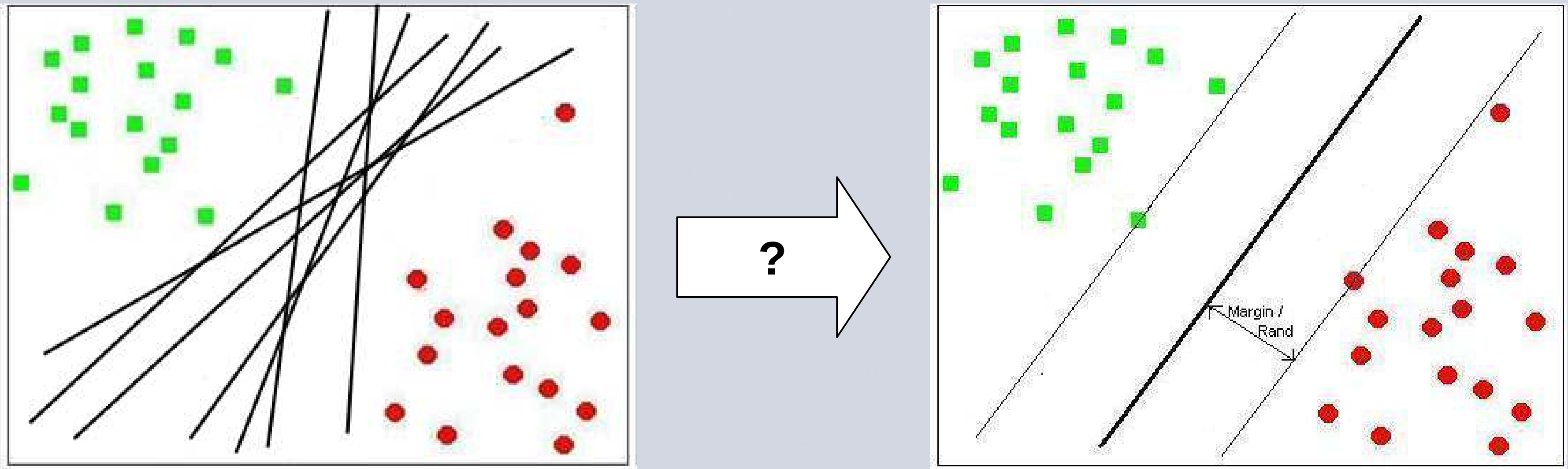
Klassifikation mit Support Vector Machines (SVM)



Seminar Digitale Nachrichtentechnik WS-2009

Johannes Leimig

Agenda

1. Grundlagen
2. Lineare SVM
   2.1 Hard-Margin
   2.2 Soft-Margin
3. Nichtlineare SVM
4. Anwendungen
5. Zusammenfassung

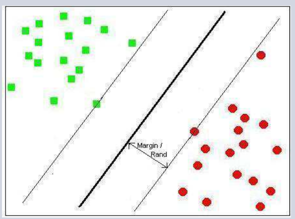
Grundlagen

- **Definition**
  - Support Vector Machines (SVM) sind Klassifikatoren, die eine Menge von Objekten so in Klassen unterteilten, dass um die Klassengrenzen herum ein möglichst breiter Bereich frei von Objekten bleibt
  - SVM sind angelernte Klassifikatoren

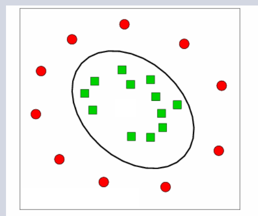
- **Motivation**
  - Klassifikation von Objekten zweier unterschiedlicher Klassen

- **Aufgabenstellung**

Lineare Klassifikation



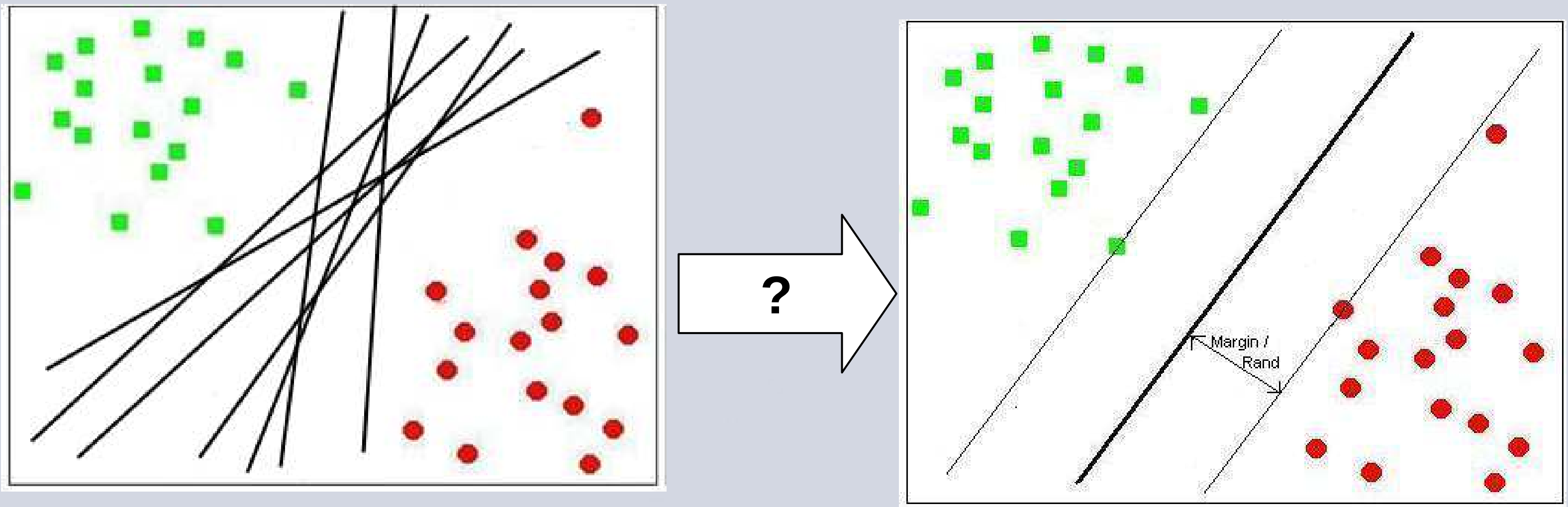
Nichtlineare Klassifikation



2.1 Lineare SVM (Hard-Margin)

Eindeutig linear trennbare Klassen (Hard-Margin)

Problemstellung: Wie kann man diese Daten am besten klassifizieren?



Idee: Diejenige Trennlinie ist am besten, die den größten Abstand zu den benachbarten Punkten der verschiedenen Klassen hat

Agenda

1. Grundlagen
2. **Lineare SVM**
3. Nichtlineare SVM
4. Anwen-dungen
5. Zusammen-fassung

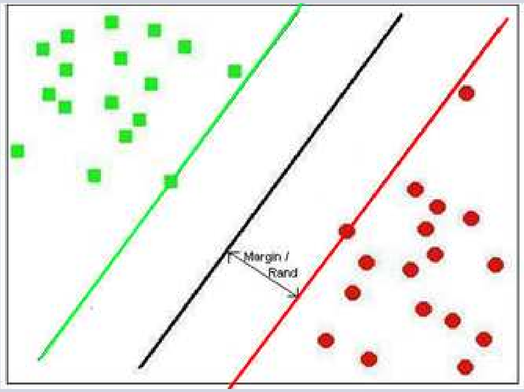
2.1 Lineare SVM (Hard-Margin)

- **Notation:**
  - Zwei-Klassen-Problem: pos./neg. Klasse $y_i = \pm 1$
  - Lerndatensatz: $x_i \in R^d$, $y_i \in \{-1,1\}$, $i \in 1,.., n$
  - Ebenengleichung
    - H: $w \cdot x_i + b = 0$
    - w : Normalenvektor
    - b : Verschiebung der Ebene
  - Lagrange Funktion
    - Verknüpfung von Haupt- und Nebenbedingung

$$L(\alpha_i, x_i) = z(x_i) - \sum_{i=1}^{n} \alpha_i g(x_i)$$

2.1 Lineare SVM (Hard-Margin)

- **Einführung der Hyperebene**



$H_1: w \cdot x_1 + b = 1$
$H_2: w \cdot x_2 + b = -1$
$H_0: w \cdot x_i + b = 0$

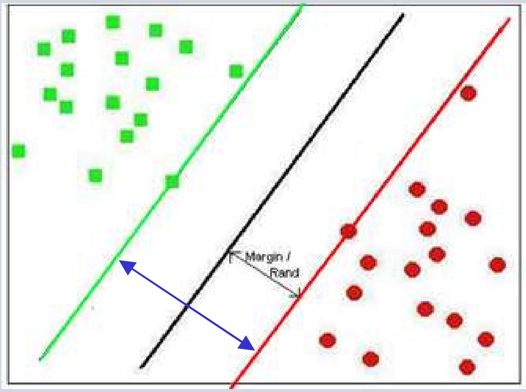
- $w \cdot x_i + b \geq 1$, für $y_i = +1$ *(1)*
  naheste positive Beispiele liegen auf
  $H_1: w \cdot x_1 + b = 1$
- $w \cdot x_i + b \leq -1$ für $y_i = -1$ *(2)*
  naheste negative Beispiele liegen auf
  $H_2: w \cdot x_2 + b = -1$
- Zusammenfassen *(1)* und *(2)*:

$\rightarrow y_i \cdot (w \cdot x_i + b) - 1 \geq 0$, für alle i *(Nebenbedingung)*

Agenda
1. Grundlagen
2. **Lineare SVM**
3. Nichtlineare SVM
4. Anwen-dungen
5. Zusammen-fassung

2.1 Lineare SVM (Hard-Margin)

- **Berechnung des maximalen Rands**



$H_1: w \cdot x_1 + b = 1$

$H_2: w \cdot x_2 + b = -1$

$H_0: w \cdot x_i + b = 0$

- Abstand zwischen H_1 und H_2 ergibt sich durch $H_1 – H_2$:
  - Breite $= 2 / ||w||$ → $||w||$ minimieren damit maximaler Rand (Maximum-Margin) entsteht
  - $y_i \cdot (w \cdot x_i + b) - 1 \geq 0$, Nebenbedingung muss zusätzlich erfüllt werden

→ Anwendung der Lagrange-Funktion

2.1 Lineare SVM (Hard-Margin)

- **Lagrange-Funktion**

  Formulierung des Primärproblems

  - Für jede Nebenbedingung wird ein Lagrange-operator (α) eingeführt

$$L(w,b,\alpha)=\frac{1}{2}\|w\|^2-\sum_{i=1}^{n}\alpha_i y_i(w\cdot x_i+b)+\sum_{i=1}^{n}\alpha_i$$

  - L ist bezüglich w und b zu minimieren und gegenüber α_i zu maximieren

Wir setzen

$$\frac{\partial}{\partial b}L(w,b,\alpha)=0 \quad \text{und} \quad \frac{\partial}{\partial w}L(w,b,\alpha)=0,$$

und erhalten

$$\sum_{i=1}^{n}\alpha_i y_i=0 \quad \text{und} \quad w=\sum_{i=1}^{n}\alpha_i y_i x_i \quad (1)$$

Agenda

1. Grundlagen
2. **Lineare SVM**
3. Nichtlineare SVM
4. Anwen-dungen
5. Zusammen-fassung

2.1 Lineare SVM (Hard-Margin)

- **Lagrange-Funktion**

  Formulierung des Dualproblems

$$L_D(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j (x_i \cdot x_j)$$

- L_D bezüglich α_i maximieren
- Die Lösung des Dualproblems lässt sich mit Hilfe der Methoden der quadratischen Optimierung lösen.

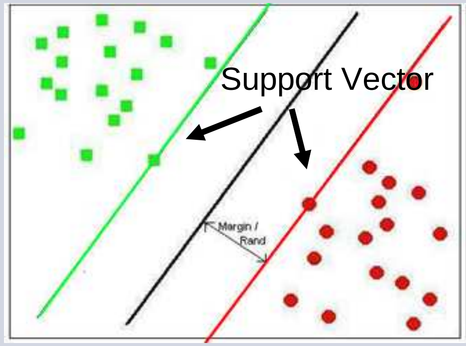
2.1 Lineare SVM (Hard-Margin)

- **Karush-Kuhn-Tucker-Bedingung**

Diese besagt: Eine optimale Lösung gilt wenn:

$$w = \sum_{i=1}^{n} \alpha_i y_i x_i \quad (1)$$

- $\alpha_i(y_i \cdot (w \cdot x_i + b) - 1) = 0$
- Da $\alpha_i \neq 0$ wegen *(1)* $\rightarrow y_i \cdot (w \cdot x_i + b) - 1 = 0$
- $\rightarrow$ Lösung wird nur durch Punkte bestimmt die auf dem Rand zwischen der Hyperebene liegen (Support Vector oder sog. Stützvektor)



- Gesuchter Normalenvektor

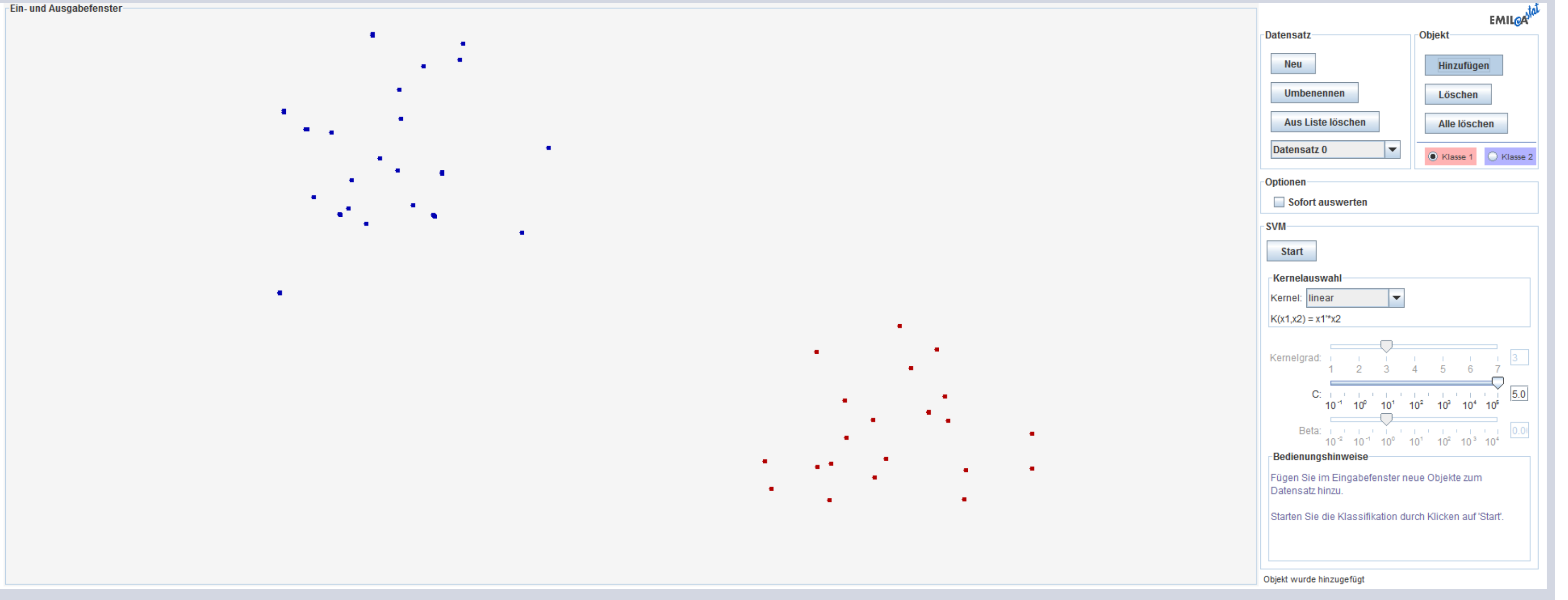
$$w = \sum_{i=1}^{NSV} \alpha_i y_i x_i$$

- Berechne b durch Einsetzen eines Stützvektors in H_0
- Entscheidungsfunktion $y_i = \mathrm{sgn}((w \cdot x_i) + b)$

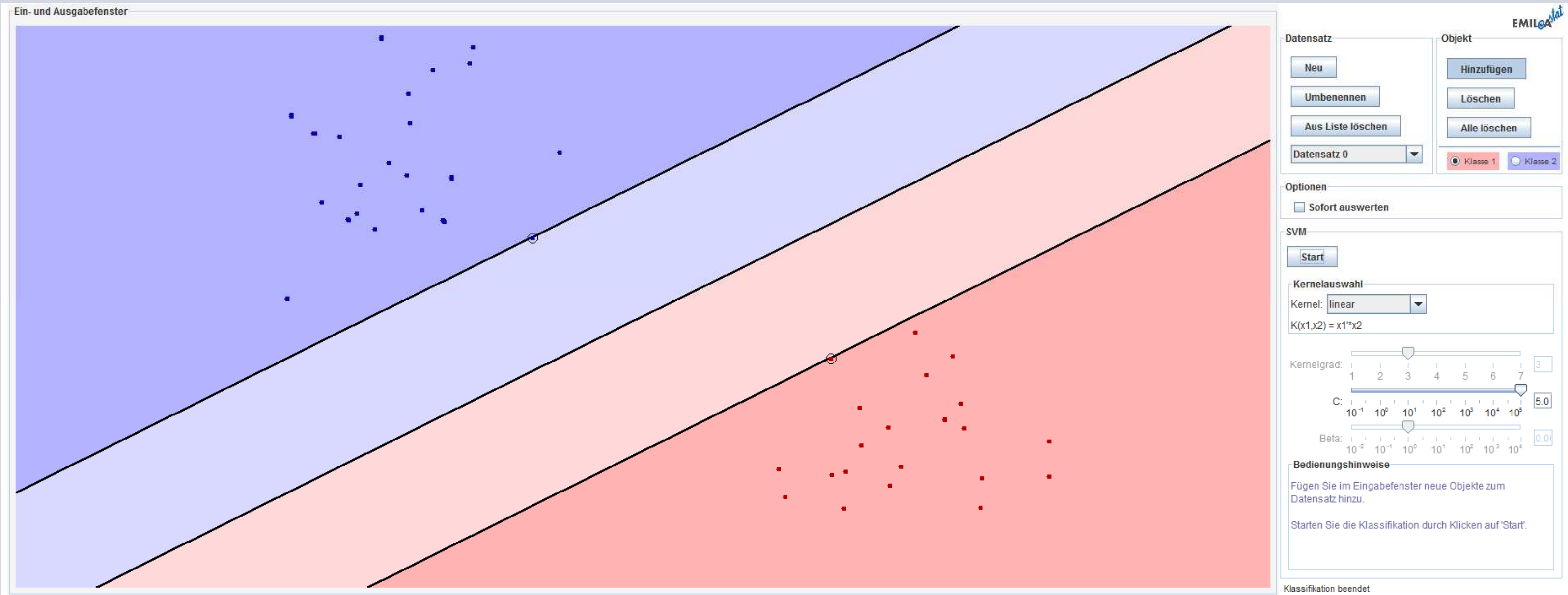
2.1 Lineare SVM (Hard-Margin)

Agenda

1. Grundlagen
2. **Lineare SVM**
3. Nichtlineare SVM
4. Anwen-dungen
5. Zusammen-fassung

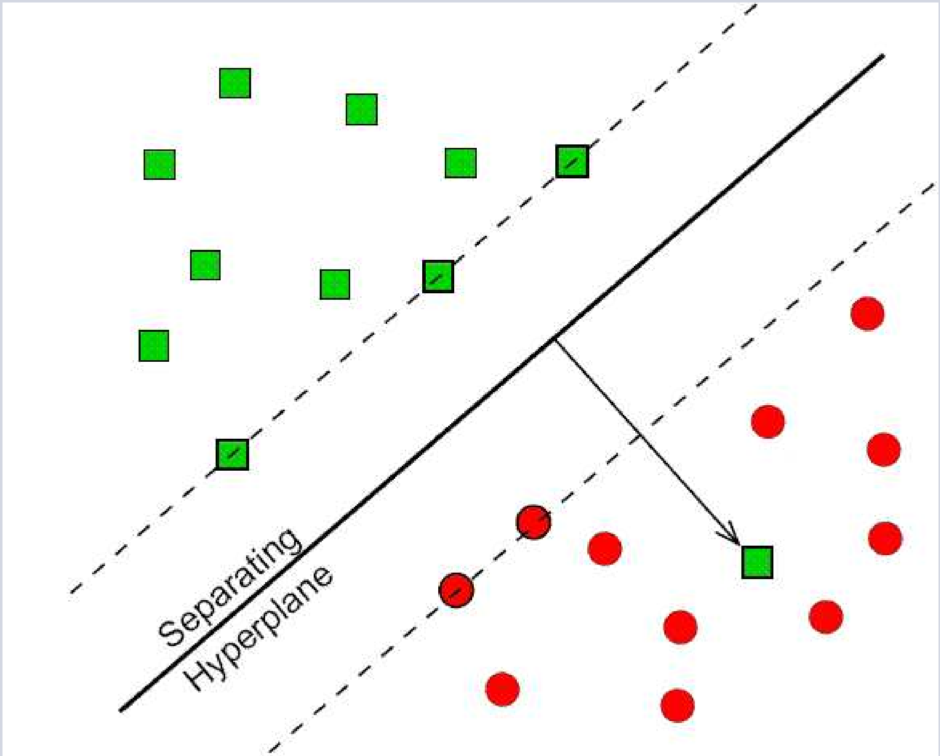


2.1 Lineare SVM (Hard-Margin)



2.2 Lineare SVM (Soft-Margin)

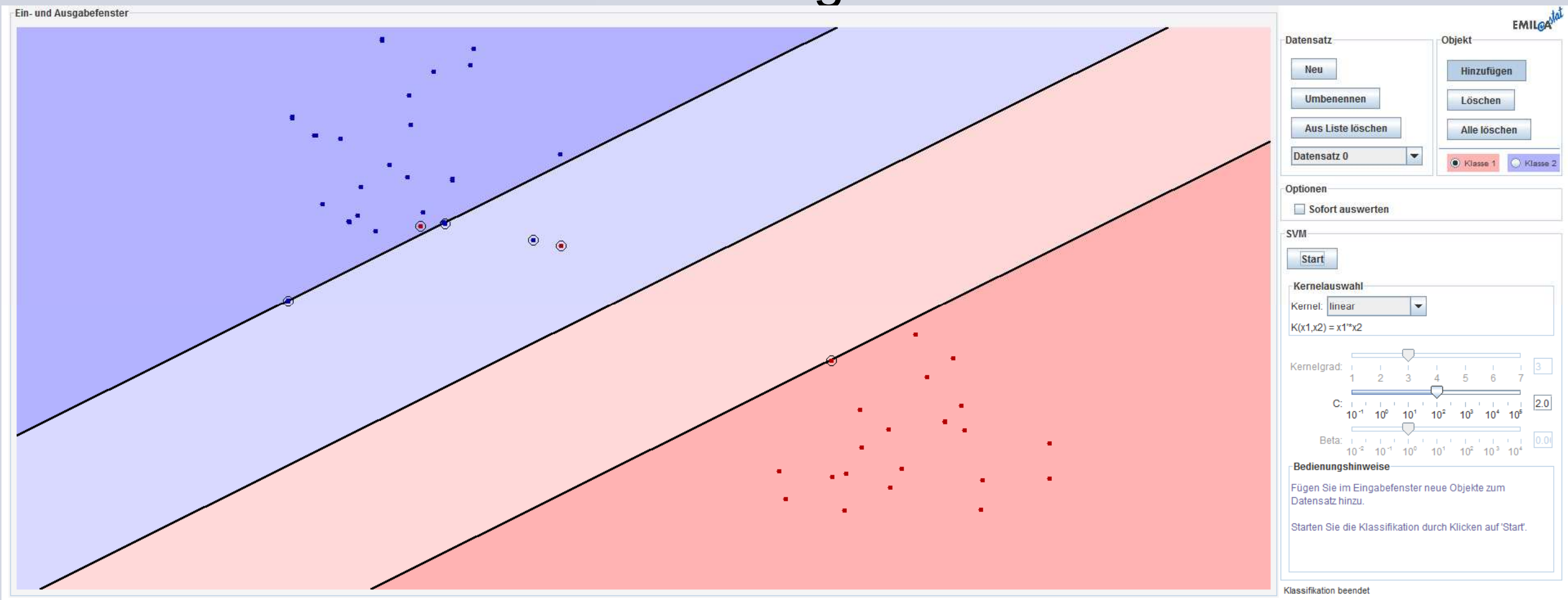
Nicht eindeutig linear trennbare Klassen (Soft-Margin)



- Gewichtungskonstante $C \geq \alpha_i \geq 0$

2.2 Lineare SVM (Soft-Margin)

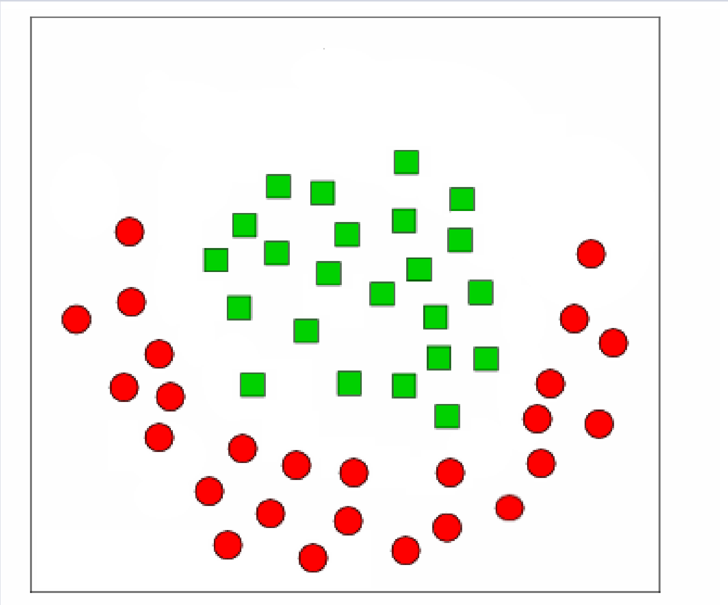
- Simulation des Soft-Margin Verfahrens



- Ergebnis:
  - C muss so gewählt werden das ein Kompromiss zwischen Fehlertoleranz und Randbreite der beiden zu trennenden Klassen getroffen wird

3. Nichtlineare SVM

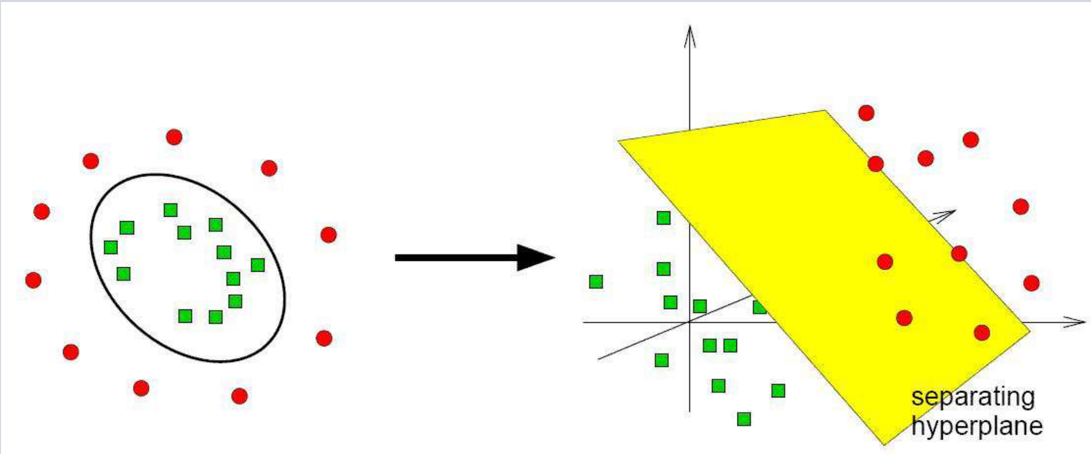
- Klassifikationsproblem



- Klassifikationsproblem nicht mehr linear lösbar → Wie kann dieses Problem gelöst werden?

3. Nichtlineare SVM

- Transformation in einen mehrdimensionalen Raum (sog. Feature Space) in dem die Daten trennbar sind



- Problem: Sehr hoher Rechenaufwand bei der Transformation, deshalb Anwendung des Kerntricks

3. Nichtlineare SVM

- **Kerntrick**
  - Erinnerung Dualproblem

$$L_D(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j (x_i \cdot x_j)$$

  - Dualproblem nur von Scalar-Produkt abhängig
  - Ersetzen des Scalar-Produktes durch Kern

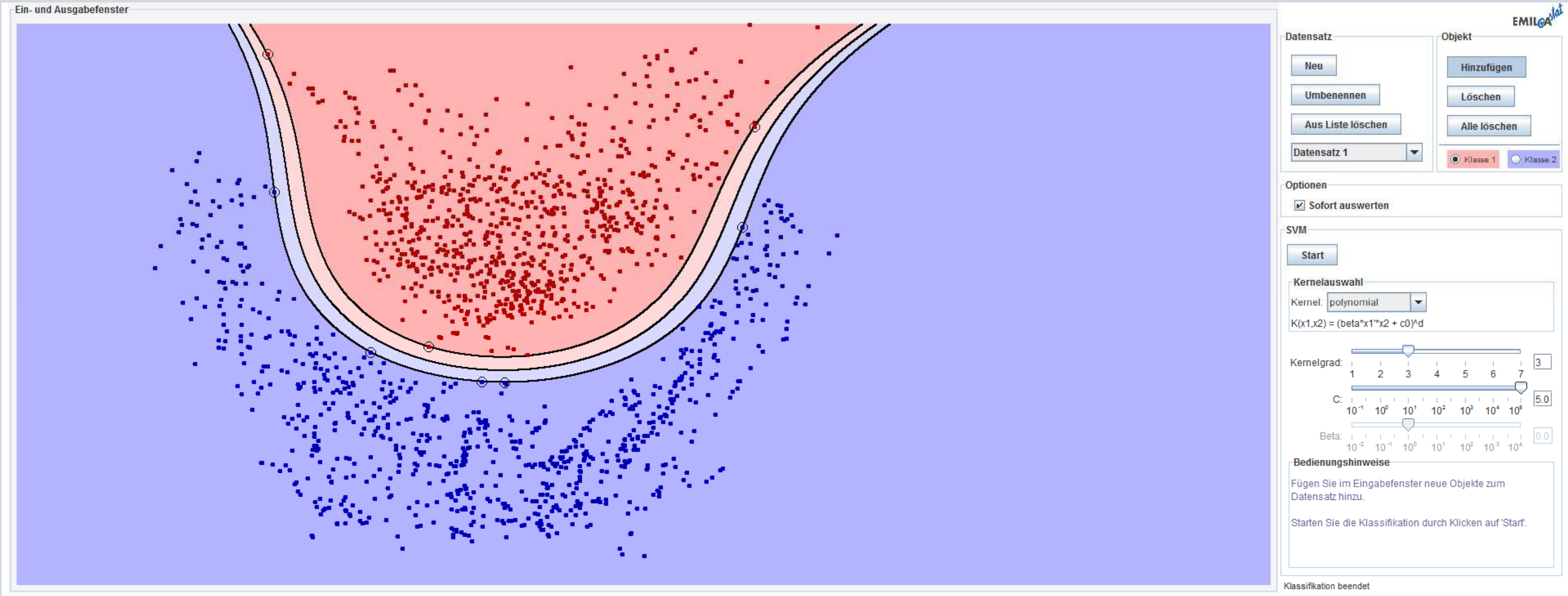
$$L_D(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j K(x_i \cdot x_j)$$

  - Mögliche Kernfunktionen

    - Lineare Kern: $K(x_i, x_j) = \langle x_i, x_j \rangle$
    - Polynome Kern: $K(x_i, x_j) = (\langle x_i, x_j \rangle + d_0)^n$
    - Gauß-Kern: $K(x_i, x_j) = \exp(-|x_i - x_j|^2 / 2\sigma^2)$

3. Nichtlineare SVM

- Simulation von nichtlinearen SVM



Anwendungen

- **Medizintechnik**
- **Sprachverarbeitung**
- **Bildverarbeitung**

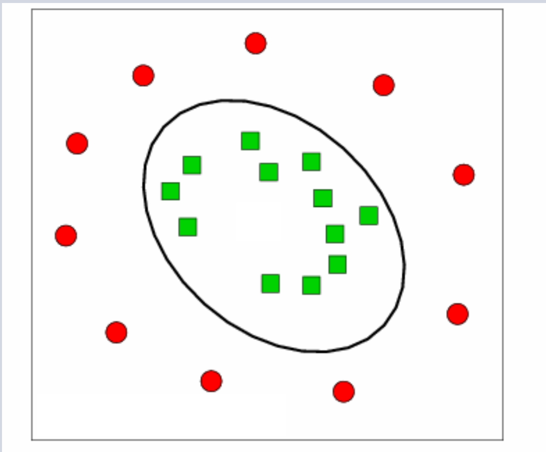
Zusammenfassung

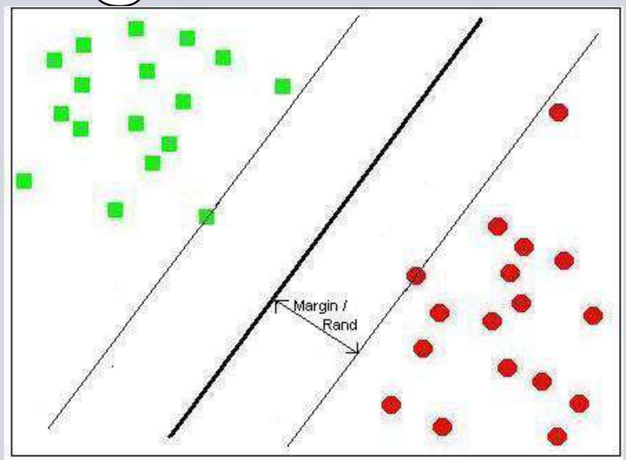
- **Vorteile von SVM**
  - Klassifikation kann schnell und eindeutig mit geringen Restfehler durchgeführt werden
  - Hohe Genauigkeit und geringe Fehlerwahrscheinlichkeit bei großen Rand
  - Verwendung von relativ einfachen mathematischen Funktionen
  - MatlabToolbox verfügbar

- **Nachteile von SVM**
  - Lernvorgang ist zeitaufwendig
  - Bei neuen Eingangsdaten muss die SVM neu angelernt werden
  - Empirische Ermittlung der Kernfunktion und die Gewichtungskonstante C
  - Gefahr der zu genauen Anpassung der Trainingsdaten durch die Wahl eines zu großen Gewichtungsfaktors

Danke für die Aufmerksamkeit!

Noch Fragen?





Seminar Digitale Nachrichtentechnik
Support Vector Machines (SVM)
Johannes Leimig

Literaturverzeichnis

[1] Kuhlenschmidt B.: „Support Vector Machines”, Ausgewählte Kapitel des Softcomputing, Universität Münster, Lehrstuhl Informatik, 2008.

[2] Böhmer M.: “Präsentation zu Support Vector Machines”, Universität Dortmund, Lehrstuhl Informatik, 2007.

[3] Burges C.: “A Tutorial on Support Vector Machines for Pattern Recognition”, Data Mining and Knowledge Discovery 2, pages 121-167, 1998.

[4] Burkhardt H.: “Vorlesung zu Support Vector Machines”, Universität Freiburg, Lehrstuhl Informatik, 2003.

[5] http://www.ml.inf.ethz.ch/education/lectures_and_seminars/annex_estat/Classifier/JSupportVectorApplet.html., Simulation von Support Vector Machines, aufgerufen am 02.12.2009.

[6] Fischer J.: “Support Vector Machines”, Seminar Statistische Lerntheorie und ihre Anwendungen, Universität Ulm, Lehrstuhl Informatik, 2007.

[7] Schleif F.: „Machine Learning – Support Vector Machines –Teil 1“, Universität Leipzig, Medizinische Fakultät, 2007.

[8] Eichhorn J.: “Grundlagen von Support Vector Machines und ihre Anwendung in der Bildverarbeitung”, Max-Plank-Institut für biologische Kybernetik, Tübingen, 2004.

[9] Sluzhivoy A.: „Effiziente Methoden der Objekterkennung mit Support Vector Machines“, Diplomarbeit, Universität Duisburg-Essen, Fachbereich Mathematik, 2008.