The Support Vector Machine

Matthias Rebel
Dozent: Peter Kolb
Universität Potsdam 2009

Überblick

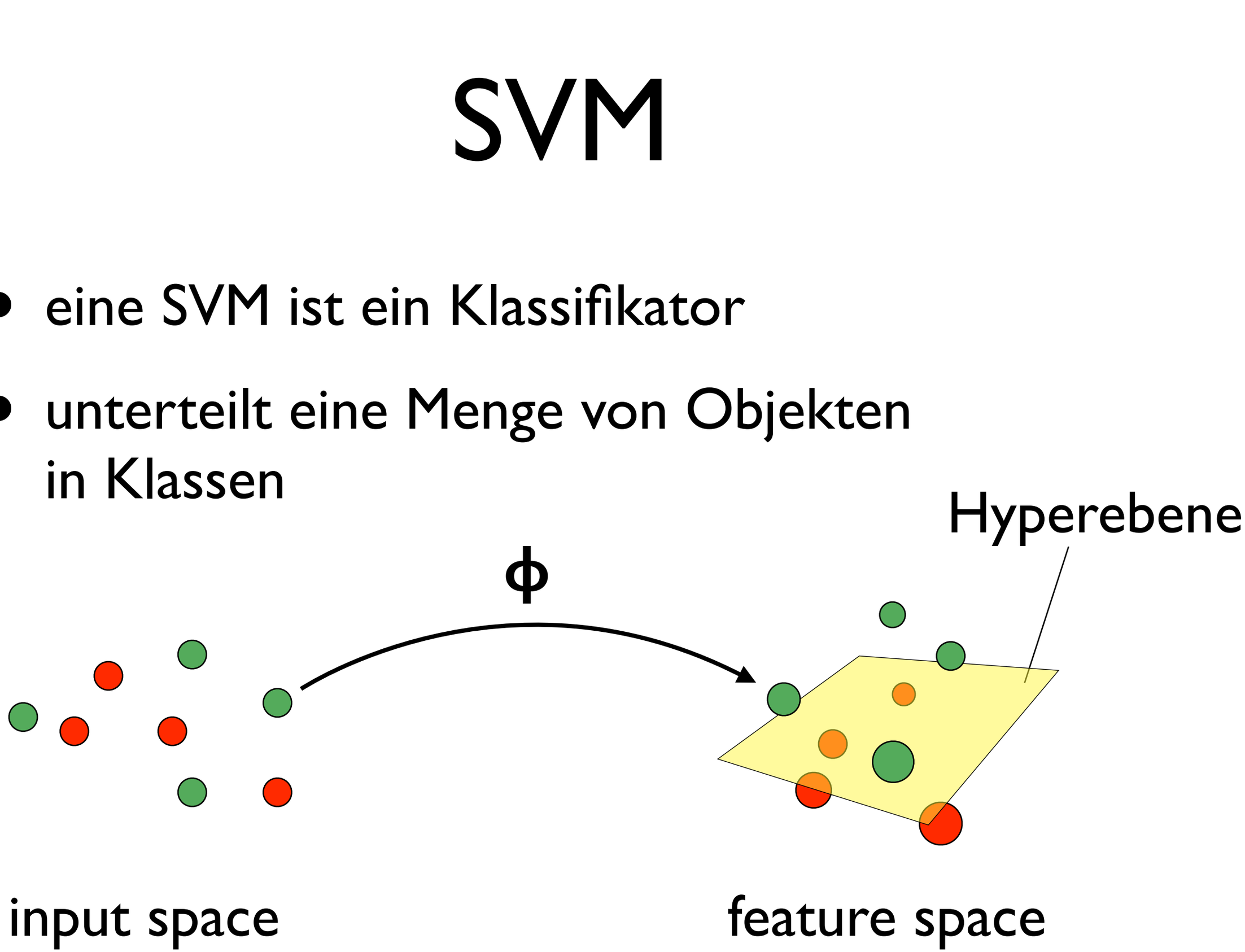
1. Was ist eine Support Vector Machine
2. Problem - Lösung
3. Vorgehensweise, Aufbau von Vektoren
3. Praktisches Beispiel: RTE_challenge
4. Aussichten, Verwendung
5. Klassen, Module, Systeme / Links

Support Vector Machine

- Übersetzung: Stützvektormaschine
- ein rein mathematisches Verfahren der Mustererkennung
- aus dem Bereich: *maschinelles* Lernen
- zur Klassifizierung / Regression

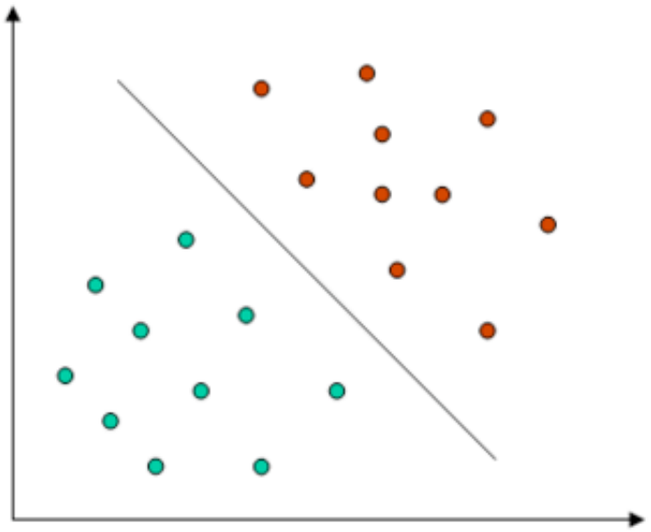
SVM

- eine SVM ist ein Klassifikator
- unterteilt eine Menge von Objekten in Klassen

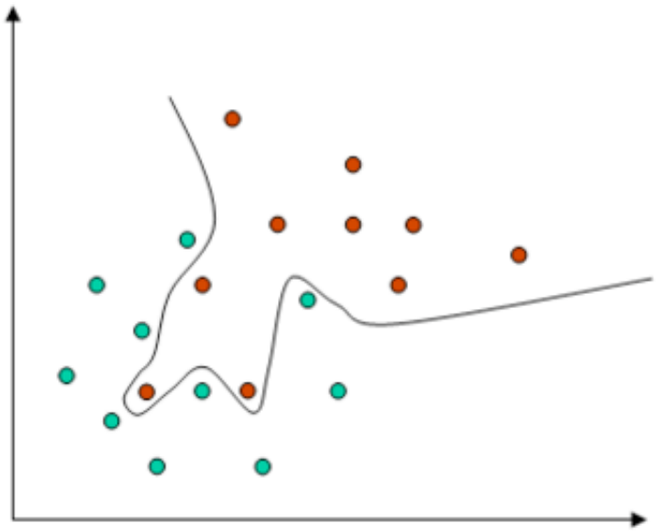


Problem

Die meisten Daten sind nicht linear trennbar.



linear trennbar



nicht linear trennbar

Die Hyperebene kann nicht gebogen werden.

Quelle der Grafik: Wikipedia, Support Vektor Machine

Lösung

- Überführung des Vektorraums
  und der enthaltenen Trainingsvektoren
  in einen höher dimensionalen Raum

- dann sind sie linear trennbar > Hyperebene
- nach der Rücktransformation in niedrig-dimensionalen Raum > Hyperfläche
- eine Hyperfläche trennt die Klassen

Kernel-Trick

- linear vs non-linear
- die Berechnung ist sehr aufwendig, daher ...

- kann mit Kernelfunktionen eine Trennfläche beschrieben werden > ohne die Hin- und Rücktransformation umzusetzen

Kernel +

- löst das Problem mit mehreren Dimensionen zu arbeiten
  - trotz infiniter Dimensionen effizient rechnen > time / space

- Welche Kernelfunktion ist die Richtige?
  - it depends ... data, goals etc. > grid.py

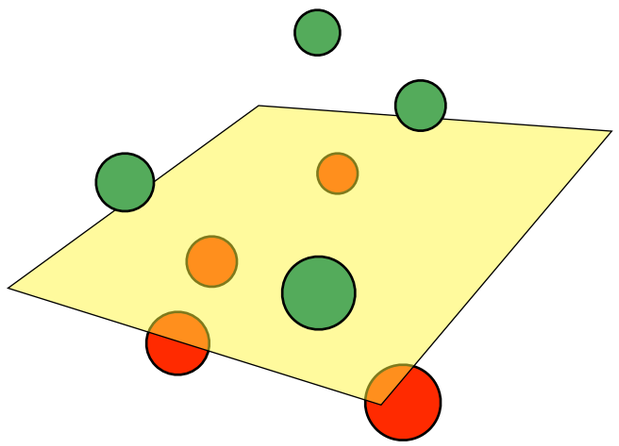
Ausgangssituation

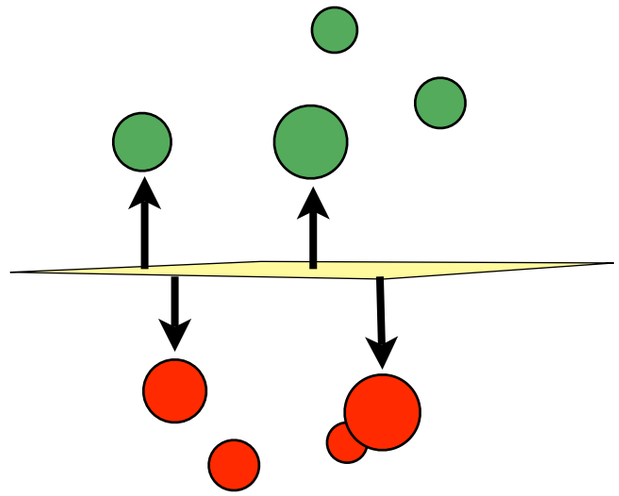
- Menge von Trainingsobjekten mit Info über Klassenzugehörigkeit
- jedes Objekt wird durch einen Vektor im Vektorraum repräsentiert
- die SVM erzeugt in diesem Raum eine Hyperebene zwischen den Klassen

Hyperebene

- ist von denen ihr am nächsten liegenden Vektoren abhängig - den Stützvektoren
  (nur diese beschreiben die Hyperebene)

- Abstand zur Hyperebene wird maximiert

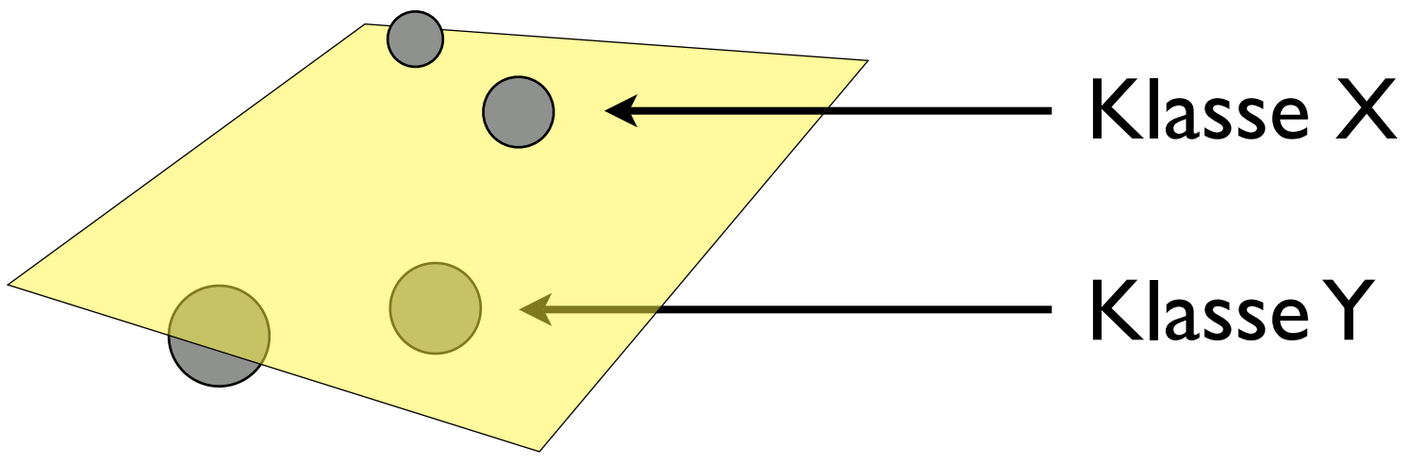




Klassifikation

- aus den Berechnungen entstandenes SVM-Model kann nun zur Klassifikation von Objekten verwendet werden

- wir kennen nicht die Klasse eines Objektes, aber wir kennen sein Vektor ...



Aufbau von Vektoren

M = Merkmal, L = Label, V = Vektor, n = Dimensionen

- binär

	L	M1	M2	M3	M4	M5	...	Mn
V1	0	0	1	1	0	0	...	
V2	1	0	0	1	0	0	...	

- multinär

	L	M1	M2	M3	M4	...	Mn
V1	3	0.12	1.44	2.0	23	...	
V2	5	0.33	0.0	1.0	62	...	

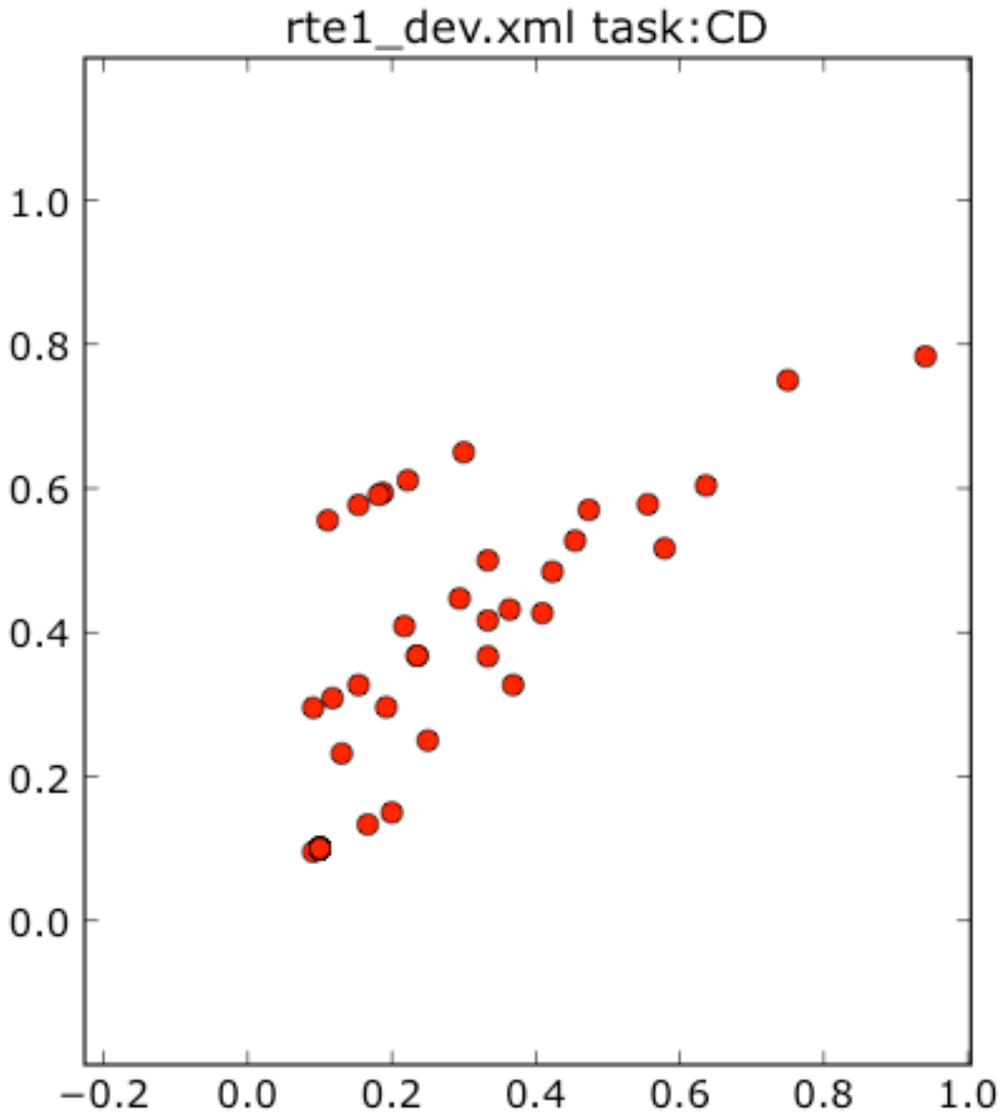
Praktisches Beispiel

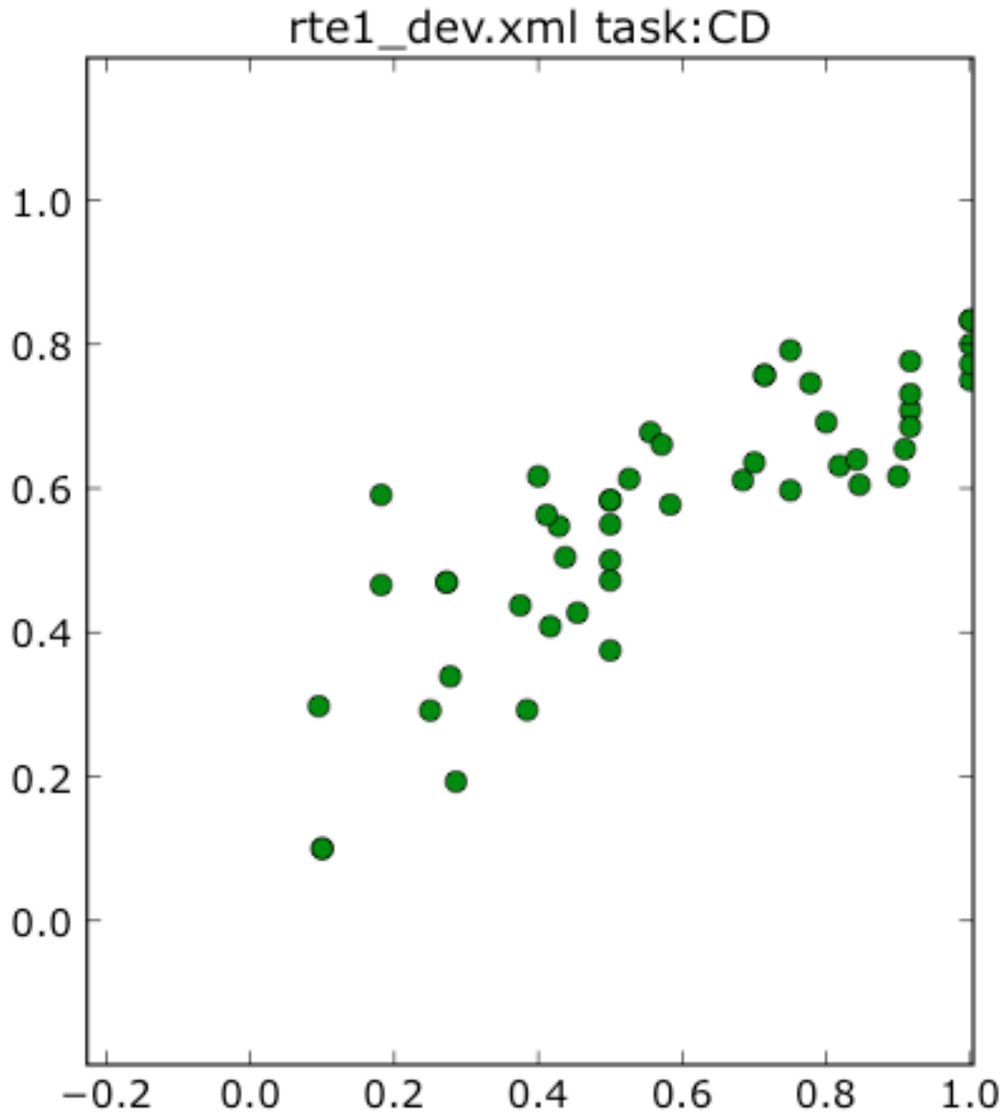
- Vektorform für ein Paar (Text, Hypothese)
  [1 M1:0.24 M2:0.5 M3:0.785 ... M5:0]
- v1 = Verhältnis der Länge von T und H, len(H) / len(T)
- v2 = Anzahl der gleichen Wörter in T und H / len(H)
- v3 = maxSim für jedes $w \in H$ aus allen Paaren (wH, wT)
- ...
- v5 = 1, wenn len(v2) == len(H), andernfalls 0

Darstellung der Vektoren in 2D

Trainingsdaten vom rte-task CD

- X Paare, False: rot, True: grün / Reduzierung von 5D auf 2D

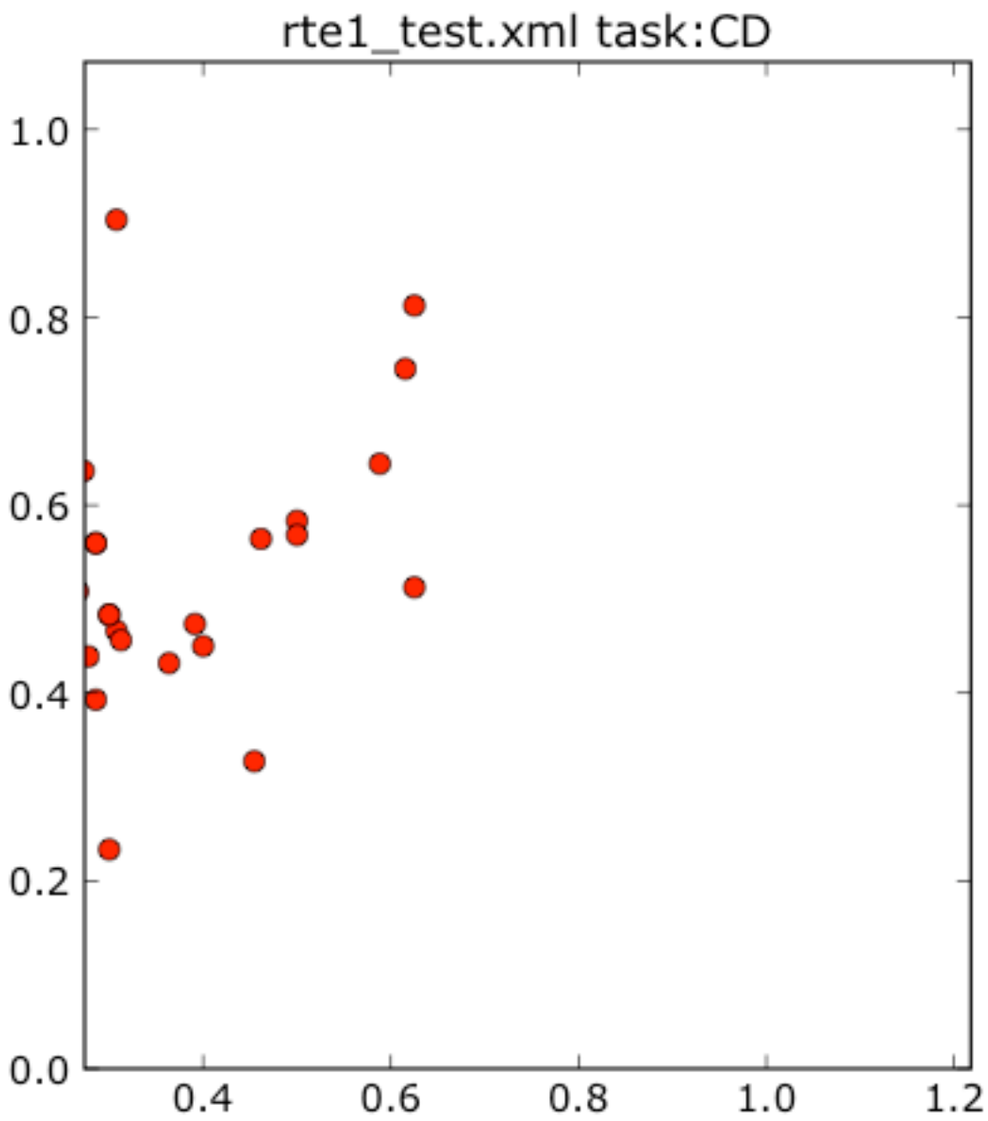


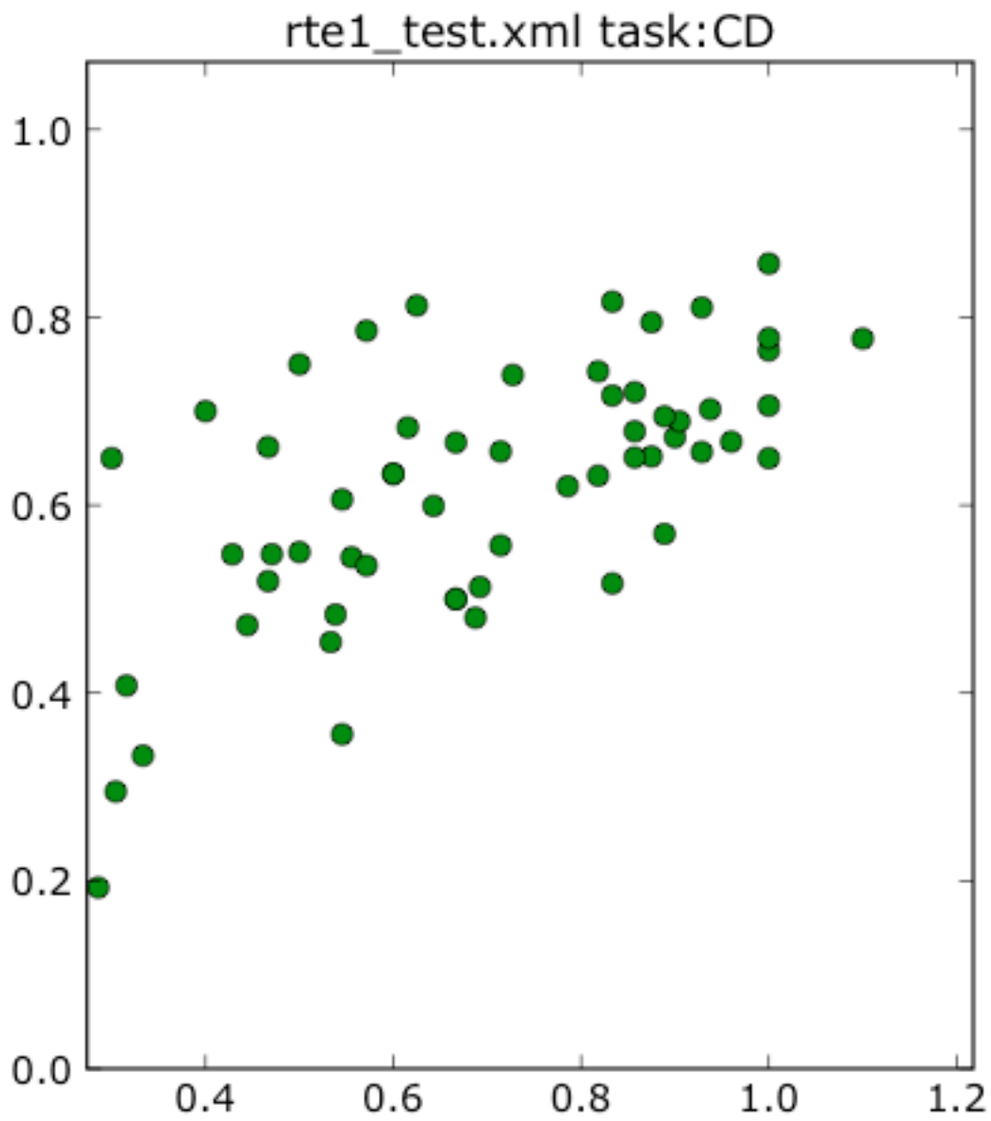


Darstellung der Vektoren in 2D

Trainingsdaten vom rte-task CD

- X Paare, False: rot, True: grün / Reduzierung von 5D auf 2D





Aussichten

- nächster Schritt: SVM an Testdaten überprüfen > Zuweisung der Labels durch das SVM-Model (Klassifizierung) ...
- SVMs können mehr als zwei Klassen behandeln/verarbeiten (one vs all / one vs one)
- nach der Installation einfach zu bedienen ;)
  > trainingsdaten, testdaten (oder neue Daten)
  > svm.train, svm.evaluate

Systeme, Bibliotheken

- LIBSVM - A Library for SVMs - sehr gut!
  (Interfaces to Python, R, C, Perl, Ruby ...)
- SVMlight - SVM in C (Bindings für Perl, Java)
- Torch - Machine Learning Biblio. in C++
- kernlab - Machine Learning Biblio. in R
- weka - Data Mining Software in Java ...

Quellen, weitere Links

- Wikipedia (de, eng)
- http://www.support-vector.net/icml-tutorial.pdf
- http://svms.org/tutorials/
- http://pyml.sourceforge.net/howto.html
- http://www.cs.waikato.ac.nz/~ml/weka/

EOF