Vorlesung Wissensentdeckung in Datenbanken

SVM — Anwendungen

Kristian Kersting, (Katharina Morik), Claus Weihs

LS 8 Informatik
Computergestützte Statistik
Technische Universität Dortmund

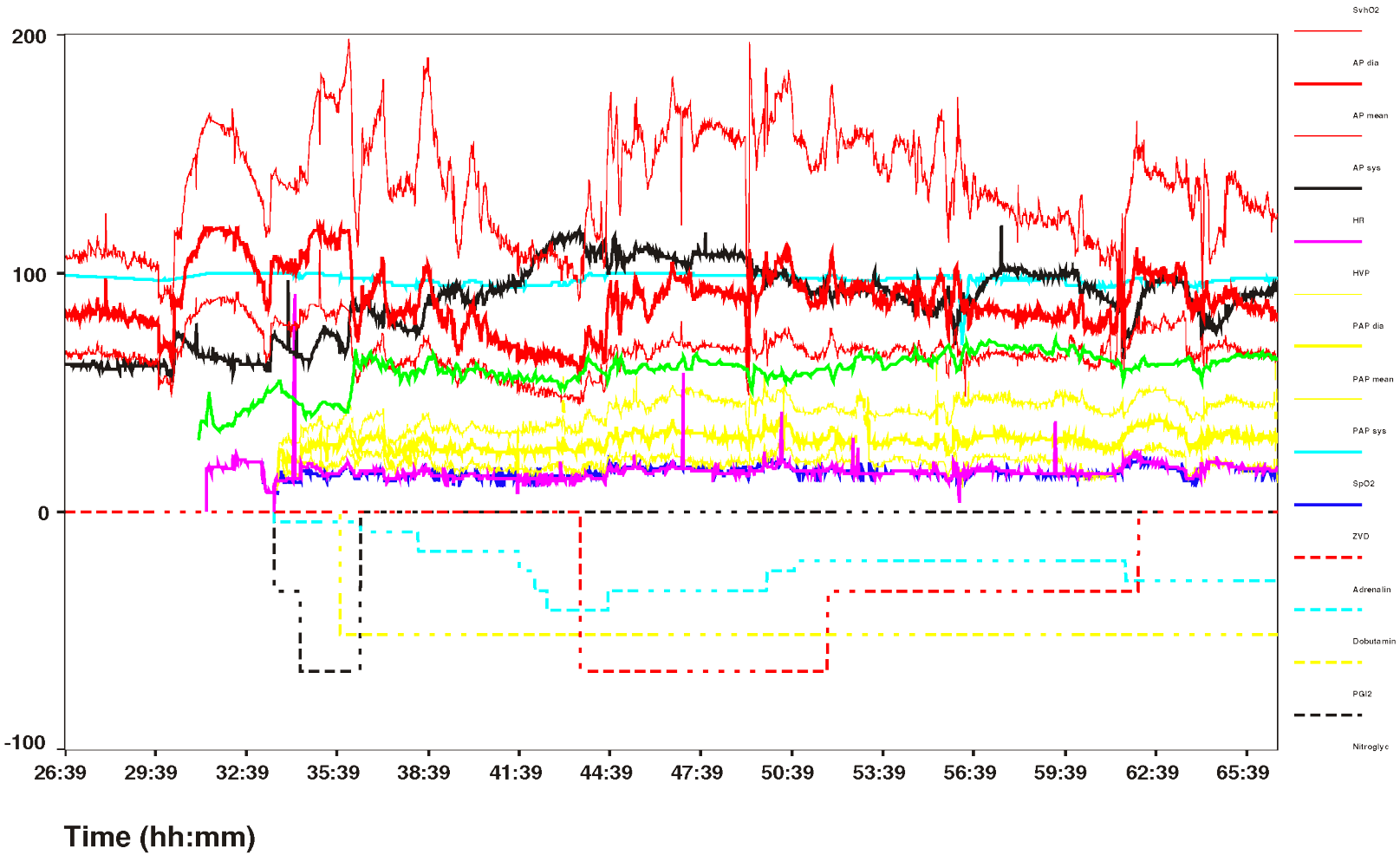
05.06.2014

Gliederung

Fallstudie Intensivmedizin

- Städtische Kliniken Dortmund, Intensivmedizin 16 Betten, Prof. Dr. Michael Imhoff (Ruhr-Universität Bochum)
- Hämodynamisches Monitoring (Vitalwerte) mittels minütliche Messungen
  - Diastolischer, systolischer, mittlerer arterieller Druck
  - Diastolischer, systolischer, mittlerer pulmonarer Druck
  - Herzrate
  - Zentralvenöser Druck
- Therapie, zuvergebene Medikamente:
  - Dobutamine, adrenaline, glycerol trinitrate, noradrenaline, dopamine, nifedipine

Patient G.C.,männlich, 60 Jahre alt - Hemihepatektomie rechts



Wann wird wleches der 6 ($>$ 0) Medikamente gegeben?

- Das ist ein Mehrklassenproblem. Wir haben 6 verschieden Medikamente.
- Daher wandeln wir es in mehrere 2-Klassen-Probleme um:
  - Für jedes Medikament entscheide, ob es gegeben werden soll oder nicht.
  - Positive Beispiele: alle Minuten, in denen das Medikament gegeben wurde
  - Negative Beispiele: alle Minuten, in denen das Medikament nicht gegeben wurde

Ergebnis: Gewichte der Vitalwerte $\vec{\beta}$, so dass positive und negative Beispiele maximal getrennt werden (SVM).

Achtung!

- Die Klassen sind jeweils schief-verteilt: sehr viel mehr negative als positive Beispiele
- Führe unterschiedliche Kostenfaktoren C_+ und C_- für die positiven und negativen Beispiele ein und setze sie so, dass

$$\frac{C_+}{C_-} = \frac{\text{Zahl der negativen Beispiele}}{Zahlderpositiven Beispiele}$$

SVM Optimierungsproblem mit klassenspezifischen Kosten

Sei $C \in \mathbb{R}$ mit $C > 0$ fest. Minimiere

$$||\vec{\beta}||^2 + C_+ \sum_{y_i>0} \xi_i + C_- \sum_{y_i<0} \xi_i$$

unter den Nebenbedingungen

$$\begin{aligned} y_i(\left\langle \vec{x}_i\,,\,\vec{\beta} \right\rangle + \beta_0) &\geq 1 - \xi_i \quad \forall\, i = 1, \ldots, N \\ \xi_i &\geq 0 \quad \text{für allle } i \end{aligned}$$

Gewichtete Vitalwerte für Glyceroltrinitrat

$$f(\vec{x}) = \left[\begin{pmatrix} 0.014 \\ 0.019 \\ -0.001 \\ -0.015 \\ -0.016 \\ 0.026 \\ 0.134 \\ -0.177 \\ \vdots \end{pmatrix} \begin{pmatrix} artsys = 174.00 \\ artdia = 86.00 \\ artmn = 121.00 \\ cvp = 8.00 \\ hr = 79.00 \\ papsys = 26.00 \\ papdia = 13.00 \\ papmn = 15.00 \\ \vdots \end{pmatrix} - 4.368 \right]$$

Wie wird ein Medikament dosiert ? Sollen wir die Dosis erhöhen, gleich lassen oder erniedrigen?

Jedes Medikament hat einen Dosierungsschritt. Für Glyceroltrinitrat ist es 1, für *Suprarenin* (adrenalin) 0,01. Die Dosis wird um einen Schritt erhöht oder gesenkt.

Wieder haben wir ein Mehrklassenproblem und wandeln es in mehrere 2 Klassenprobleme umwandeln: für jedes Medikament und jede Richtung (increase, decrease, equal):

- Positive Beispiele: alle Minuten, in denen die Dosierung in der betreffenden Richtung geändert wurde
- Negative Beispiele: alle Minuten, in denen die Dosierung nicht in der betreffenden Richtung geändert wurde.

Ergebnis fürs Steigern von Dobutamine

Vektor $\vec{\beta}$ für p Attribute

ARTEREN:	-0.05108108119
SUPRA :	0.0089280753865797 3
DOBUTREX :	-0.100650806786886
WEIGHT :	-0.0393531801046265
AGE :	-0.00378828681071417
ARTSYS :	-0.323407537252192
ARTDIA :	-0.0394565333019493
ARTMN :	-0.180425080906375
HR :	-0.10010405264306
PAPSYS :	-0.0252641188531731
PAPDIA :	0.0454843337112765
PAPMN :	0.00429504963736522
PULS :	-0.0313501236399881

Anwendung der gelernten SVM für Dobutamin

- Patientwerte
  pat46, artmn 95, min. 2231
  ...
  pat46, artmn 90, min. 2619

- Gelernte Gewichte β_i:
  $artmn - 0,18$
  ...

Wir berechnen also jetzt die Entscheidungsfunktion der gelernten SVM:

$$svm\_calc = \sum_{i=1}^{p} \beta_i x_i$$

$$decision = sign(svm\_calc + \beta_0)$$

Zum Beispiel

- $svm\_calc(pat46, dobutrex, up, min.2231, 39)$
- $svm\_calc(pat46, dobutrex, up, min.2619, 25)$

Da $\beta_0 = -26$, erhöhen wir Dobutamin in der 2231-ten Minute, aber nicht in der 2619-ten Minute.

Steigern von Glyceroltrinitrat (nitro)

$$f(x) = \left[\begin{pmatrix} 0.014 \\ 0.019 \\ -0.001 \\ -0.015 \\ -0.016 \\ 0.026 \\ 0.134 \\ -0.177 \\ -9.543 \\ -1.047 \\ -0.185 \\ 0.542 \\ -0.017 \\ 2.391 \\ 0.033 \\ 0.334 \\ 0.784 \\ 0.015 \end{pmatrix} \begin{pmatrix} artsys = 174.00 \\ artdia = 86.00 \\ artmn = 121.00 \\ cvp = 8.00 \\ hr = 79.00 \\ papsys = 26.00 \\ papdia = 13.00 \\ papmn = 15.00 \\ nifedipine = 0 \\ noradrenaline = 0 \\ dobutamie = 0 \\ dopamie = 0 \\ glyceroltrinitrate = 0 \\ adrenaline = 0 \\ age = 77.91 \\ emergency = 0 \\ bsa = 1.79 \\ broca = 1.02 \end{pmatrix} - 4.368 \right]$$

Evaluierung

- Blind test über 95 noch nicht gesehener Patientendaten.
  - Experte stimmte überein mit tatsächlichen Medikamentengaben in 52 Fällen
  - SVM Ergebnis stimmte überein mit tatsächlichen Medikamentengaben in 58 Fällen

Konfusionmatrizen: SVM und Experte (Experte steht in den Klammern)

Dobutamine	*Actual up*	*Actual equal*	*Actual down*
Predicted up	**10 (9)**	12 (8)	0 (0)
Predicted equal	7 (9)	**35 (31)**	9 (9)
Predicted down	2 (1)	7 (15)	**13 (12)**

Was wissen wir jetzt?

- Anwendung der SVM für die Medikamentenverordnung
- Asymmetrische Kostenparameter

Vorhersage Abverkauf

Gegeben Verkaufsdaten von 50 Artikeln in 20 Läden über 104 Wochen

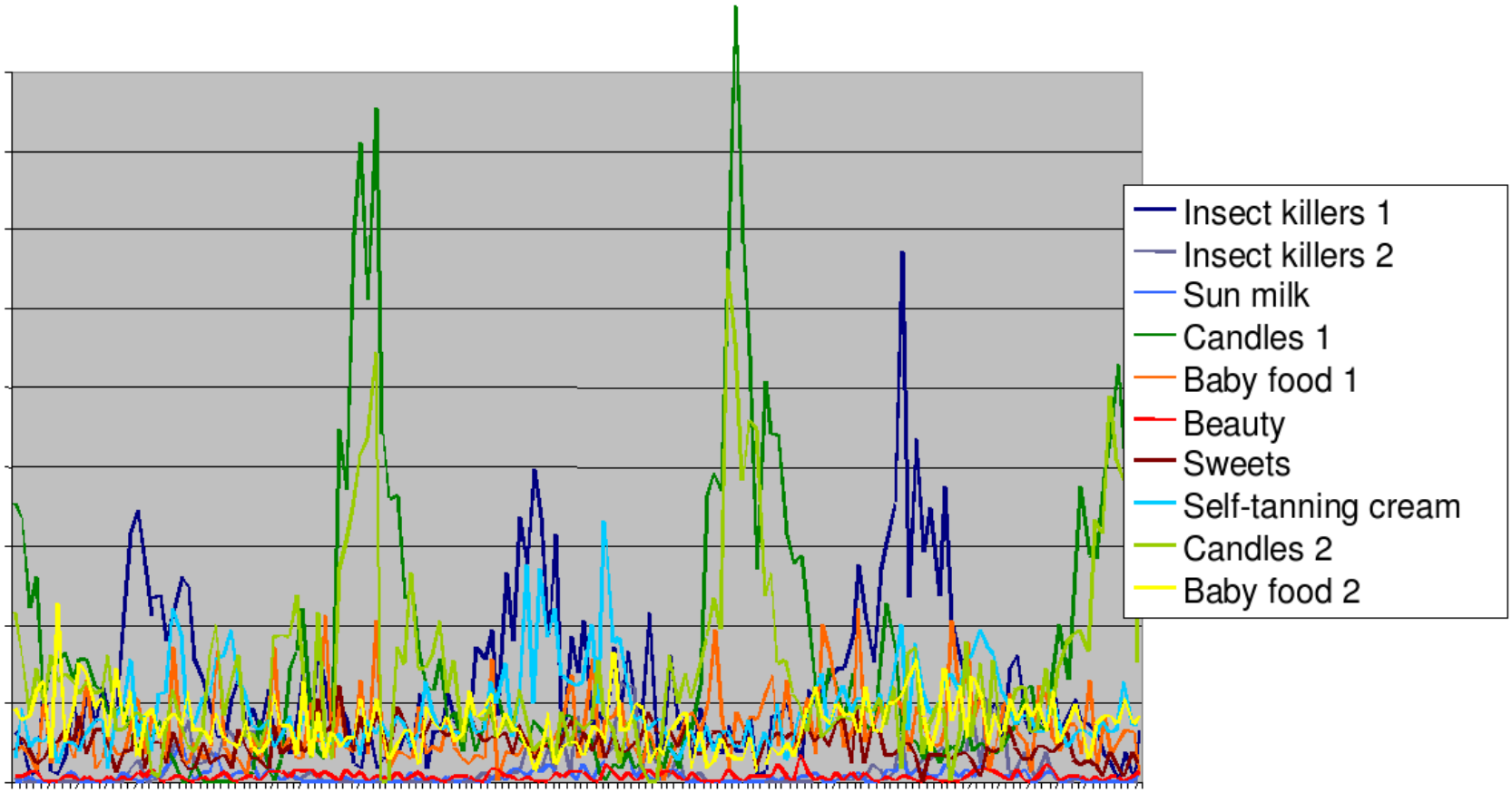
Vorhersage Verkäufe eines Artikels, so dass

- Die Vorhersage niemals den Verkauf unterschätzt,
- Die Vorhersage überschätzt weniger als eine Faustregel.

Beobachtung 90% der Artikel werden weniger als 10 mal pro Woche verkauft.

Anforderung Vorhersagehorizont von mehr als 4 Wochen.

Abverkauf Drogerieartikel



Verkaufsdaten – multivariate Zeitreihen

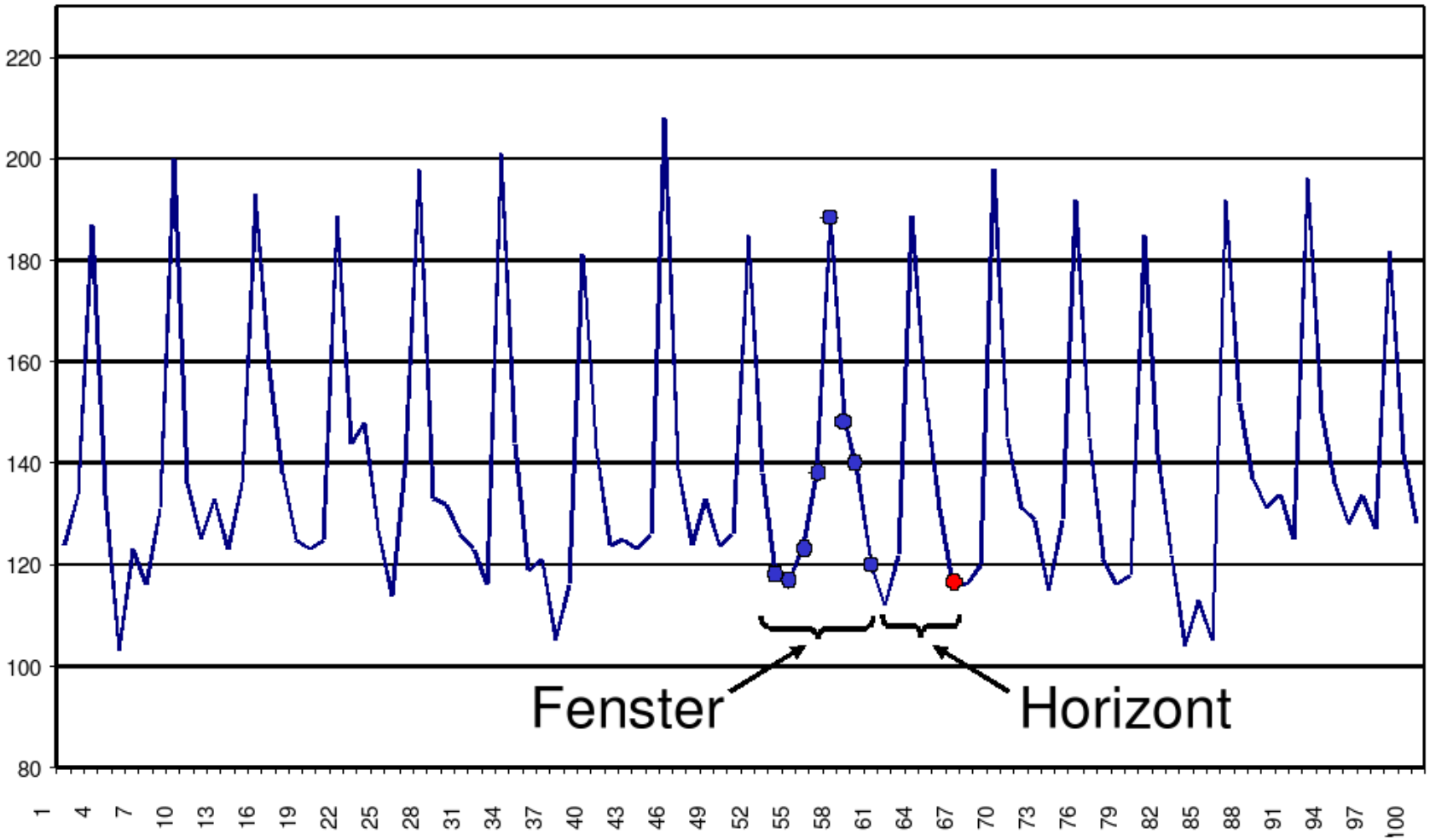
Shop	Week	Item1	...	Item50
Dm1	1	4	...	12
Dm1	...	...	...	...
Dm1	104	9	...	16
Dm2	1	3	...	19
...	...	...	...	...
Dm20	104	12	...	16

Vorverarbeitung: Umformung in univariates Problem und Windowing

- Univariat: Wir lernen eine SVM pro Produkt und pro Shop.
- Problem: eine Zeitreihe ist nur 1 Beispiel! Das ist für das Lernen zu wenig.
- Lösung (Windowing: Viele Vektoren aus einer Reihe gewinnen durch Fenster der Breite (Anzahl Zeitpunkte) w, bewege Fenster um m Zeitpunkte weiter.

Shop_Item_Window	Week	Sale	Week	Sale
Dm1_Item1_1	1	4...	5	7
Dm1_Item1_2	2	4...	6	8
...	...	...	...	...
Dm1_Item1_100	100	6...	104	9
...	...	...	...	...
Dm20_Item50_100	100	12...	104	16

Beispiel: Prognose von Zeitreihen



Houston, we have a problem!!!

Aber wir haben noch ein weiteres Problem. Bisher haben wir SVMs für Klassifikationsprobleme betrachtet.

Es liegt aber ein Regressionsproblem vor!!!

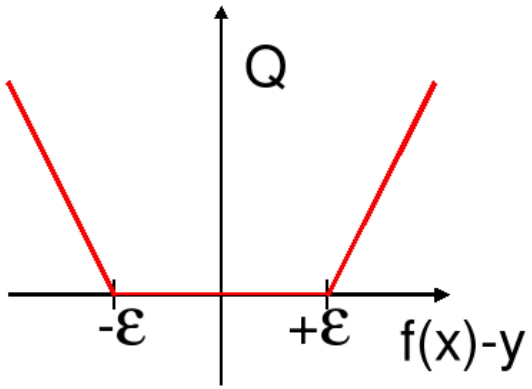
Support Vector Regression (SVR)

Durch Einführung einer anderen *Loss-Funktion* läßt sich die SVM zur Regression nutzen. Sei $\varepsilon \in \mathbb{R}_{>0}$ und

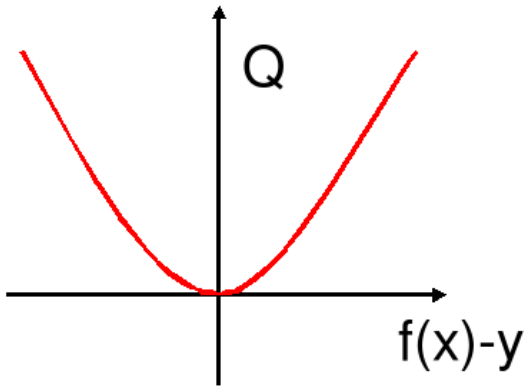
$$L_k(y, f(\vec{x}, \alpha)) = \begin{cases} 0 & \text{, falls } y - f(\vec{x}, \alpha) \leq \varepsilon \\ (y - f(\vec{x}, \alpha) - \varepsilon)^k & \text{, sonst} \end{cases}$$

Die *Loss-Funktion* L_1 gibt den Abstand der Funktion f von den Trainingsdaten an, alternativ quadratische Loss-Funktion L_2 (betrachten wir hier aber nicht):

lineare Verlustfunktion

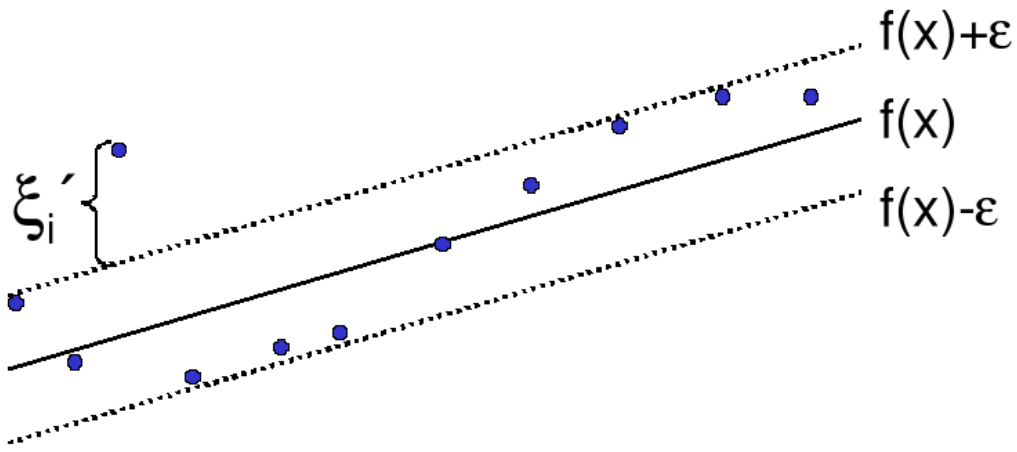


quadratische Verlustfunktion



SVMs für Regression

Die ξ_i bzw. ξ_i' geben für jedes Beispiel Schranken an, innerhalb derer der vorhergesagte Funktionswert für jedes Beispiel liegen soll:



Bei der Lösung des Optimierungsproblems mit Lagrange (Herleitung hier nicht gezeigt) führt dies zu *zwei* α-Werten je Beispiel!

SVMs für Regression

Dadurch ergibt sich das folgende Optimierungsproblem:

Regressions-SVM

Minimiere

$$\|\vec{\beta}\|^2 + C\left(\sum_{i=1}^{N}\xi_i + \sum_{i=1}^{N}\xi_i'\right)$$

unter den Nebenbedingungen

$$\begin{aligned} f(\vec{x_i}) &= \left\langle \vec{\beta}, \vec{x_i} \right\rangle + \beta_0 \leq y_i + \epsilon + \xi_i' \\ f(\vec{x_i}) &= \left\langle \vec{\beta}, \vec{x_i} \right\rangle + \beta_0 \geq y_i - \epsilon - \xi_i \end{aligned}$$

SVMs für Regression

Das duale Problem enthält für jedes $\vec{x}_i$ je zwei α-Werte α_i und α'_i, je einen für ξ_i und ξ'_i, d.h.

Duales Problem für die Regressions-SVM

Maximiere

$$L_D(\vec{\alpha}, \vec{\alpha}') = \sum_{i=1}^{N} y_i (\alpha'_i - \alpha_i) - \epsilon \sum_{i=1}^{N} y_i (\alpha'_i - \alpha_i) - \frac{1}{2} \sum_{i,j=1}^{n} y_i (\alpha'_i - \alpha_i) (\alpha'_j - \alpha_j) K(\vec{x}_i, \vec{x}_j)$$

unter den Nebenbedingungen

$$0 \leq \alpha_i, \alpha'_i \leq C \forall\, i = 1, \ldots, N \quad \text{und} \quad \sum_{i=1}^{N} \alpha'_i = \sum_{i=1}^{N} \alpha_i$$

SVR für Abverkaufsprognose

- Für jeden Laden und jeden Artikel, wende die SVM an. Die gelernte Regressionsfunktion wird zur Vorhersage genutzt.
- Asymmetrische Verlustfunktion :
  - Unterschätzung wird mit 20 multipliziert, d.h. 3 Verkäufe zu wenig vorhergesagt – 60 Verlust
  - Überschätzung zählt unverändert, d.h. 3 Verkäufe zu viel vorhergesagt – 3 Verlust
- Das können wir wieder durch asymmetrische Kostenparameter ausdrücken, jetzt allerdings laufen die "Kosten"-Summen über alle Bespiele.

(Diplomarbeit Stefan Rüping 1999)

Vergleich mit Exponential Smoothing

Der durchschnittliche Verlust, der nicht auf $[0, 1]$ normiert ist.

Horizont	SVM	exp. smoothing
1	56.764	52.40
2	57.044	59.04
3	57.855	65.62
4	58.670	71.21
8	60.286	88.44
13	59.475	102.24

Unter exponential smoothing versteht man

$$\begin{aligned} s_0 &= x_0 \\ s_{t+1} &= \gamma x_t + (1-\gamma) s_t, \quad t > 0 \end{aligned}$$

Der Glättungsparameter γ wurde auf einer holdout-Menge optimal bestimmt.

Was wissen wir jetzt?

- Idee der Regressions-SVM (SVR)
- Anwendung der SVR für die Verkaufsvorhersage
- Umwandlung multivariater Zeitreihen in mehrere univariate
- Gewinnung vieler Vektoren durch gleitende Fenster (Windowing)
- Asymmetrische Verlustfunktion (loss function)

World Wide Web

- Seit 1993 wächst die Anzahl der Dokumente – 12,9 Milliarden Seiten (geschätzt für 2005)
- Ständig wechselnder Inhalt ohne Kontrolle, Pflege
  - Neue URLs
  - Neue Inhalte
  - URLs verschwinden
  - Inhalte werden verschoben oder gelöscht
- Verweisstruktur der Seiten untereinander
- Verschiedene Sprachen
- Unstrukturierte Daten

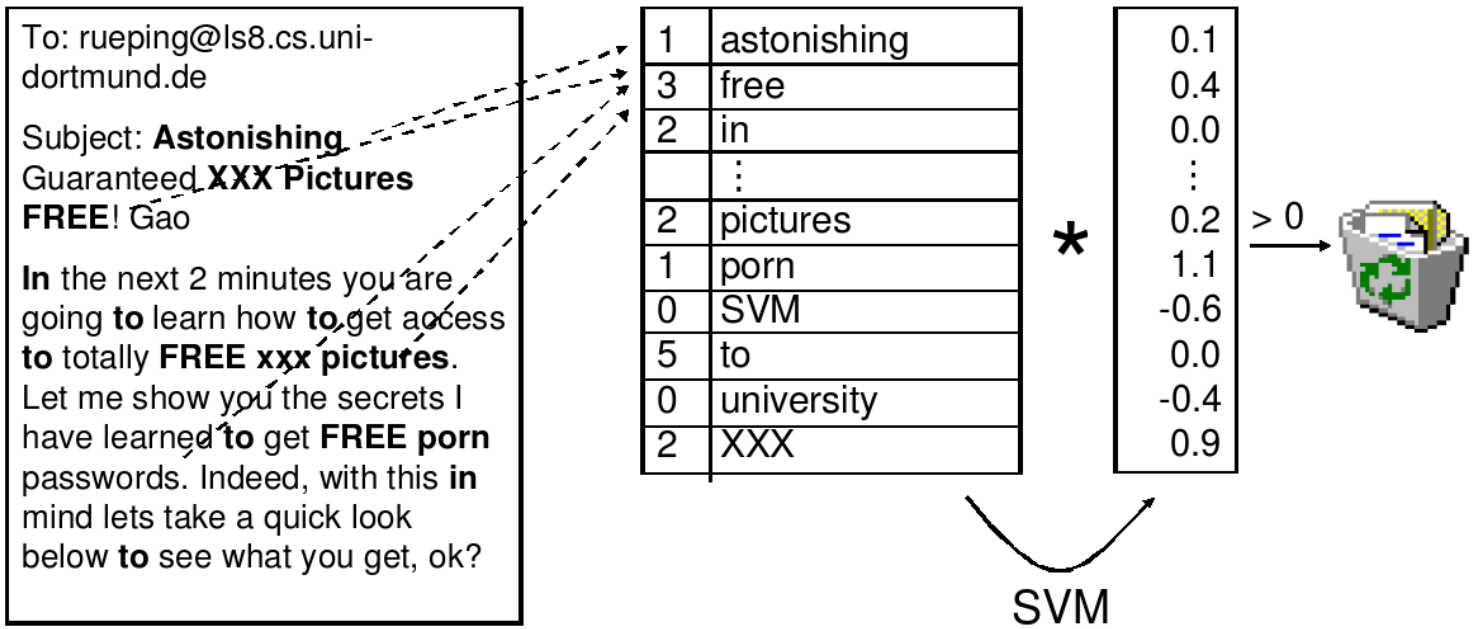
Interessante Aufgaben, in denen Data Mining zum Einsatz kommt / kommen kann

- Indexierung möglichst vieler Seiten (Google)
- Suche nach Dokumenten, ranking der Ergebnisse z.B. nach Häufigkeit der Verweise auf das Dokument (PageLink – Google)
- Kategorisierung (Klassifikation) der Seiten manuell (Yahoo), automatisch
- Strukturierung von Dokumentkollektionen (Clustering)
- Personalisierung:
  - Navigation durch das Web an Benutzer anpassen
  - Ranking der Suchergebnisse an Benutzer anpassen
- Extraktion von Fakten aus Texten

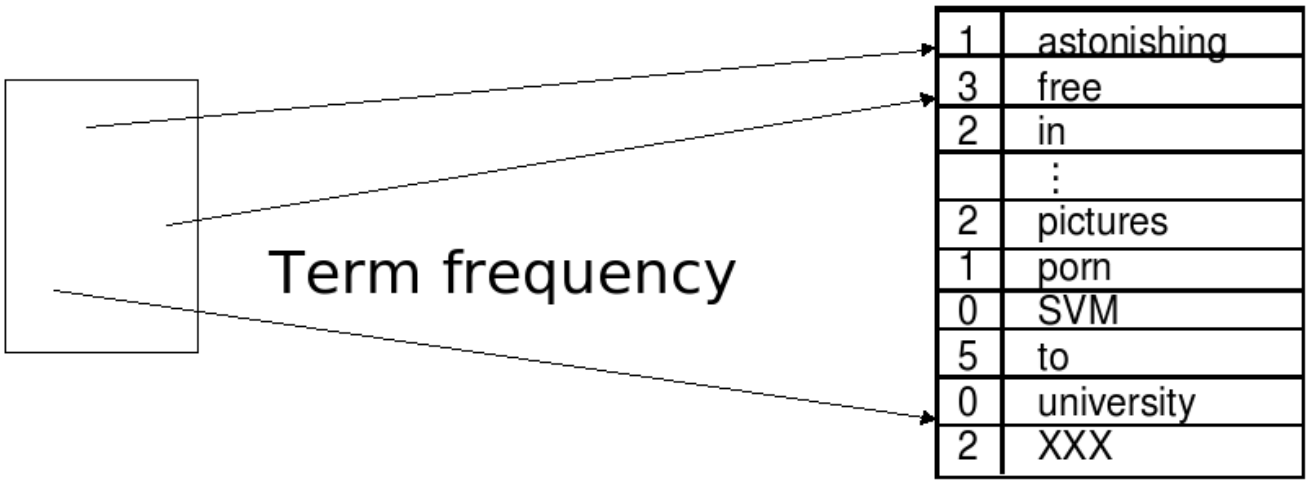
Textklassifikation

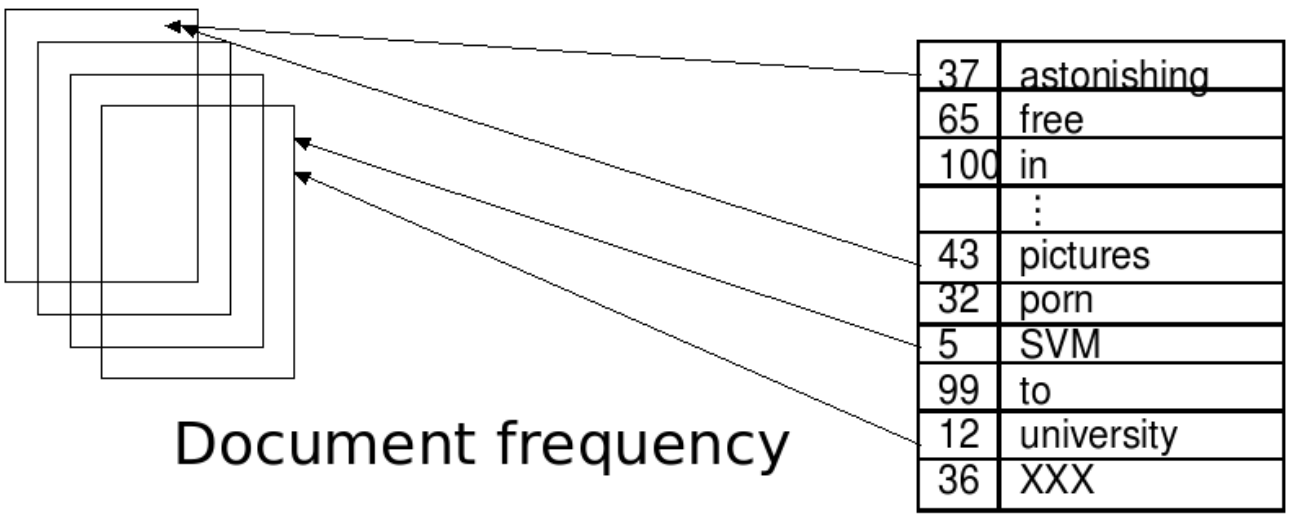
- Ein Dokument besteht aus einer Menge von Termen (Wörtern)
  - Bag of words: Vektor, dessen Komponenten die Häufigkeit eines Wortes im Dokument angeben.
- Für alle Dokumente gibt es eine Termliste mit Verweis auf die Dokumente.
  - Anzahl der Dokumente, in denen das Wort vorkommt.
- Wir wollen die Klasse eine Dokuments vorhersagen (Thema, Sprache, etc.).

Beispiel zur Klassifikation



Texte als Daten





TFIDF

- Term Frequenz: wie häufig kommt ein Wort w_i in einem Dokument d vor? $TF(w_i, d)$
- Dokumentenfrequenz: in wie vielen Dokumenten einer Kollektion D kommt ein Wort w_i vor? $DF(w_i)$
- Inverse Dokumentenfrequenz:

$$IDF(D, w_i) = \log \frac{|D|}{DF(w_i)}$$

- Bewährte Repräsentation:

$$TFIDF(w_i, D) = \frac{TF(w_i, d)IDF(w_i, D)}{\sqrt{\sum_j [TF(w_j, d)IDF(w_j, D)]^2}}$$

Eigenschaften der Textklassifikation 1

Hochdimensionaler Merkmalsraum

- Reuters Datensatz mit 9603 Dokumenten: verschiedene Wörter

$$V = 27658$$

Das unterliegt sogar physikalischen Gesetzmäßigkeiten:

- Heaps'sches Gesetz: Anzahl aller Wörter

$$V = ks^{\beta}$$

  wobei V die Zahl der unterschiedlichen Wörter in einem Text mit s vielen Wörtern ist. k und β sind freie Parameter. Für englische Texte oft $10 < k < 100$ und $0.4 < \beta < 0.6$ (gelernt aus Daten):
- Beispiel:
  - Konkatenieren von 10 000 Dokumenten mit je 50 Wörtern zu einem,
  - $k = 15$ und $\beta = 0,5$
  - ergibt $V = 35000 \rightarrow$ stimmt!

Eigenschaften der Textklassifikation 2

Heterogener Wortgebrauch

- Dokumente der selben Klasse haben manchmal nur Stoppwörter gemeinsam!
- Es gibt keine relevanten Terme, die in allen positiven Beispielen vorkommen.
- Familienähnlichkeit (Wittgenstein): A und B haben ähnliche Nasen, B und C haben ähnliche Ohren und Stirn, A und C haben ähnliche Augen.

Eigenschaften der Textklassifikation 3

Redundanz der Merkmale

- Ein Dokument enthält mehrere eine Klasse anzeigende Wörter.
- Experiment:
  - Ranking der Wörter nach ihrer Korrelation mit der Klasse.
  - Trainieren von Naive Bayes für Merkmale von Rang

1 - 200	(90% precision/recall)
201 - 500	(75%)
601 - 1000	(63%)
1001- 2000	(59%)
2001- 4000	(57%)
4001- 9947	(51%) – zufällige Klassifikation (22%)

Eigenschaften der Textklassifikation 4

Dünn besetzte Vektoren

- Reuters Dokumente durchschnittlich 152 Wörter lang
  - mit 74 verschiedenen Wörtern
  - also meist bei etwa 78 Wörtern 0
- Euklidsche Länge der Vektoren klein!

Eigenschaften der Textklassifikation 5

Zipfs Gesetz: Verteilung von Wörtern in Dokumentkollektionen ist ziemlich stabil.

- Ranking der Wörter nach Häufigkeit (r)
- Häufigkeit des häufigsten Wortes (max)
- $\frac{1}{r} max$ häufig kommt ein Wort des Rangs r vor.

Daher: Plausibilität guter Textklassifikation durch SVM

Eigentlich sprechen einge dieser Eigenschaften nicht für ein einfaches Klassifikationsaufgabe. Aber,

- SVM iteriert nicht über alle Attribute eines Beispiels, sondern hängt ab von der Euklidschen Länge der Vektoren.
- Wortvektoren sind spärlich besetzt, also ist die Euklidsche Länge klein.
- Folglich ist die SVM nicht bedroht durch die hohe Dimension der Wortvektoren.

Plausibilität guter Textklassifikation durch SVM: Etwas Formaler

- R sei Radius des Balles, der die Daten enthält. Dokumente werden auf einheitliche Länge normiert, so dass $R = 1$
- Margin sei δ, so dass großes δ kleinem $\frac{R^2}{\delta^2}$ entspricht.

Reuters	$\frac{R^2}{\delta^2}$	$\sum_{i=1}^{n} \xi$
Earn	1143	0
acquisition	1848	0
money-fx	1489	27
grain	585	0
crude	810	4

Reuters	$\frac{R^2}{\delta^2}$	$\sum_{i=1}^{n} \xi$
trade	869	9
interest	2082	33
ship	458	0
wheat	405	2
corn	378	0

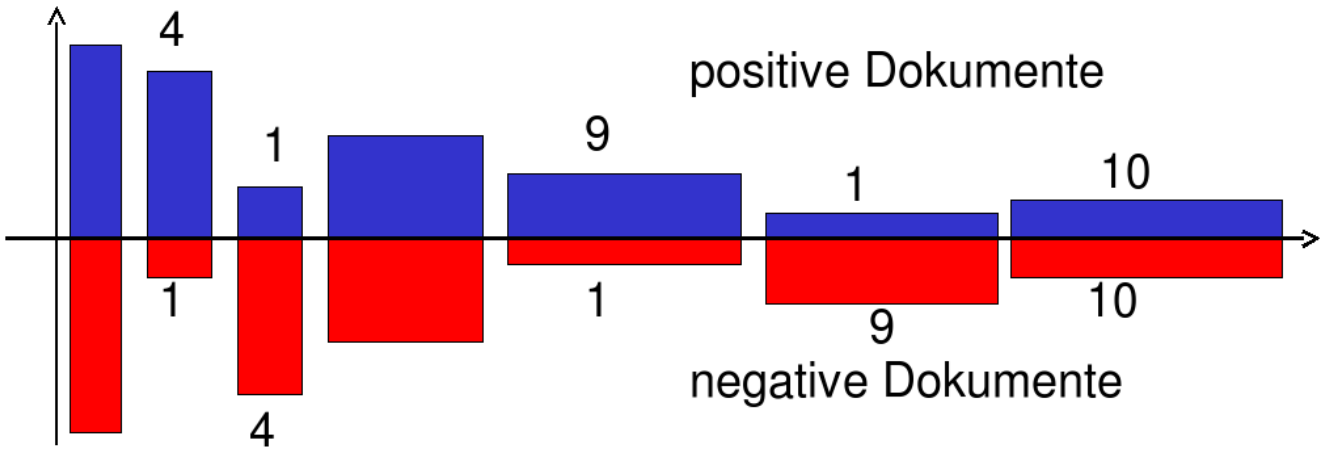
TCat Modell – Prototyp

- Hochdimensionaler Raum: $V = 11100$ Wörter im Lexikon
- Dünn besetzt: Jedes Dokument hat nur 50 Wörter, also mindestens 11050 Nullen
- Redundanz: Es gibt 4 mittelhäufige und 9 seltene Wörter, die die Klasse anzeigen
- Verteilung der Worthäufigkeit nach Zipf/Mandelbrot.
- Linear separierbar mit $\beta_0 = 0$, $\sum_{i=1}^{11100} \beta_i x_i$

$$\beta_i = \begin{cases} 0,23 & \text{für mittelhäufige Wörter in } \textit{POS}, \\ -0,23 & \text{für mittelhäufige Wörter in } \textit{NEG}, \\ 0,04 & \text{für seltene Wörter in } \textit{POS}, \\ -0,04 & \text{für seltene Wörter in } \textit{NEG}, \\ 0 & \text{sonst} \end{cases}$$

TCat im Bild

- 20 aus 100 Stoppwörtern, 5 aus 600 mittelhäufigen und 10 aus seltenen Wörtern kommen in *POS*- und *NEG*-Dokumenten vor; 4 aus 200 mittelhäufigen Wörtern in *POS*, 1 in *NEG*, 9 aus 3000 seltenen Wörtern in *POS*, 1 in *NEG* (Es müssen nicht immer die selben Wörter sein!)



TCat

The TCat concept

$$TCat\left(\left[p_1 : n_1 : f_1\right], \ldots, \left[p_s : n_s : f_s\right]\right)$$

describes a binary classification task with s sets of disjoint features. The i-th set includes f_i features. Each positive example contains p_i occurences of features from the respective set and each negative example contains n_i occurrences. The same feature can occur multiple times in one document. (Joachims 2002)

TCat zum Bild

7 disjunkte Wortmengen; bei einem zur Klasse gehörigen Dokument kommt 20 mal eines der 100 Wörter der ersten Wortmenge vor, 4 mal eines der 200 Wörter der zweiten Wortmenge, ...; bei einem nicht zur Klasse gehörigen Dokument gibt es 20 Auftreten von Wörtern aus der ersten Wortmenge,... Es sind also nicht bestimmte Wörter, die die Klassenzugehörigkeit anzeigen!

$$TCat(\underbrace{[20:20:100]}_{\text{sehr häufig}}\underbrace{[4:1:200][1:4:200][5:5:600]}_{\text{mittel häufig}}\underbrace{[9:1:3000][1:9:3000][10:10:4000]}_{\textit{selten}})$$

Lernbarkeit von TCat durch SVM

(Joachims 2002) Der erwartete Fehler einer SVM ist nach oben beschränkt durch:

$$\frac{R^2}{n+1}\frac{a+2b+c}{ac-b^2}$$

$$a = \sum_{i=1}^{s} \frac{p_i^2}{f_i}$$

$$b = \sum_{i=1}^{s} \frac{p_i^2 n_i}{f_i}$$

$$c = \sum_{i=1}^{s} \frac{n_i^2}{f_i}$$

$$R^2 = \sum_{r=1}^{d} \left(\frac{v}{(r+k)^{\phi}} \right)^2$$

Es gibt l Wörter, s Merkmalsmengen, für einige i: $p_i \neq n_i$ und die Termhäufigkeit befolgt Zipfs Gesetz. Wähle d so, dass:

$$\sum_{r=1}^{d} \frac{v}{(r+k)^{\phi}} = l$$

Das heisst im Wesentlichen, dass SVMs sich sehr gut zur Textklassifikation eignen. Wir können unterschiedliche Textklassen immer linear separieren.

Ergebnisse zur Textklassifikation

Reuters	$\frac{R^2}{\delta^2}$	$\sum_{i=1}^{n} \xi_i$
earn	1143	0
acq	1848	0
money-fx	1489	27
grain	585	0
crude	810	4
trade	869	9
interest	2082	33
ship	458	0
wheat	405	2
corn	378	0

WebKB	$\frac{R^2}{\delta^2}$	$\sum_{i=1}^{n} \xi_i$
course	519	0
faculty	1636	0
project	741	0
student	1588	0

Ohsumed	$\frac{R^2}{\delta^2}$	$\sum_{i=1}^{n} \xi_i$
Pathology	11614	0
Cardiovasc.	4387	0
Neoplasms	2868	0
Nervous Sys.	3303	0
Immunologic	2556	0

	model $\mathcal{E}(Err^n(h_{SVM}))$	experiment $Err^n_{test}(h_{SVM})$
WebKB “course”	11.2%	4.4%
Reuters “earn”	1.5%	1.3%
Ohsumed “pathology”	94.5%	23.1%

Was wissen Sie jetzt?

- Die automatische Klassifikation von Texten ist durch das WWW besonders wichtig geworden.
- Texte können als Wortvektoren mit TFIDF dargestellt werden. Die Formel für TFIDF können Sie auch!
- Textkollektionen haben bzgl. der Klassifikation die Eigenschaften: hochdimensional, dünn besetzt, heterogen, redundant, Zipfs Gesetz.
- Sie sind mit breitem margin linear trennbar.
- Das TCat-Modell kann zur Beschränkung des erwarteten Fehlers eingesetzt werden. Die Definition von TCat kennen Sie mindestens, besser wäre noch die Fehlerschranke zu kennen.

Thorsten Joachims “The Maximum-Margin Approach to Learning Text Classifiers Kluwer", 2001

Was wissen Sie jetzt insgesamt über SVMs ?

- Support Vektor Maschinen (SVM) maximieren die Margin zwischen Klassen
- SVMs können als mathematische Opimierungsprobleme formuliert werden (primal, dual)
- Können z.B. mittels koordiniertem Gradientenabstieg trainiert werden
- Können mit nicht-linearen und nicht–linear-separierbaren Daten umgehen (Strafferem, Kerne)
- Kerne berechnen das Skalarprodukt implizit in einem hochdimensionalen Raum
- SVMs können auch zur Regression benutzt werden
- SVMs haben viele Einsatzgebiete und sind insbesondere zur Klassifikation von hochdimensionalen Daten wie z.B. Text geeignet