



# Vorlesung Maschinelles Lernen

## SVM – Anwendungen, Textkategorisierung

Katharina Morik

LS 8 Informatik  
Technische Universität Dortmund

22.11.2011



# Gliederung

- 1 Anwendungen
- 2 Web Mining
  - Information Retrieval
- 3 Textklassifikation
- 4 Verwendung des Modells zur Textklassifikation für zeitgestempelte Daten



## Fallstudie Intensivmedizin

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





## Wann wird Medikament gegeben?

- Mehrklassenproblem in mehrere 2-Klassen-Probleme umwandeln:
  - 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

Parameter: Kosten falscher Positiver = Kosten falscher Negativer

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



## Beispiel: Intensivmedizin

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



## Wie wird ein Medikament dosiert ?

- Mehrklassenproblem in mehrere 2 Klassenprobleme umwandeln: für jedes Medikament und jede Richtung (increase, decrease, equal), 2 Mengen von Patienten-daten:
  - 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.



## Steigern von Dobutamine

Vektor  $\vec{\beta}$  für  $p$  Attribute

<i>ARTEREN</i> :	-0.05108108119
<i>SUPRA</i> :	0.00892807538657973
<i>DOBUTREX</i> :	-0.100650806786886
<i>WEIGHT</i> :	-0.0393531801046265
<i>AGE</i> :	-0.00378828681071417
<i>ARTSYS</i> :	-0.323407537252192
<i>ARTDIA</i> :	-0.0394565333019493
<i>ARTMN</i> :	-0.180425080906375
<i>HR</i> :	-0.10010405264306
<i>PAPSYS</i> :	-0.0252641188531731
<i>PAPDIA</i> :	0.0454843337112765
<i>PAPMN</i> :	0.00429504963736522
<i>PULS</i> :	-0.0313501236399881



## Anwendung des Gelernten für Dobutamin

- Patientwerte  
pat46, artmn 95, min. 2231  
...  
pat46, artmn 90, min. 2619
- Gelernte Gewichte  $\beta_i$ :  
 $artmn - 0,18$   
...

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

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

- $svm\_calc(pat46, dobutrex, up, min.2231, 39)$
- $svm\_calc(pat46, dobutrex, up, min.2619, 25)$
- $\beta_0 = -26$ , i.e. increase in minute 2231, not increase in minute 2619.



## Steigern von Glyceroltrinitrat (nitro)

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

- Jedes Medikament hat einen Dosierungsschritt. Für Glyceroltrinitrat ist es 1, für *Suprarenin* (adrenalin) 0,01. Die Dosis wird um einen Schritt erhöht oder gesenkt.
- Vorhersage: *pred\_interv* (*pat*49, *min*.32, *nitro*, 1, 0)



## 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

<i>Dobutamine</i>	<i>Actual up</i>	<i>Actual equal</i>	<i>Actual down</i>
<i>Predicted up</i>	<b>10 (9)</b>	12 (8)	0 (0)
<i>Predicted equal</i>	7 (9)	<b>35 (31)</b>	9 (9)
<i>Predicted down</i>	2 (1)	7 (15)	<b>13 (12)</b>



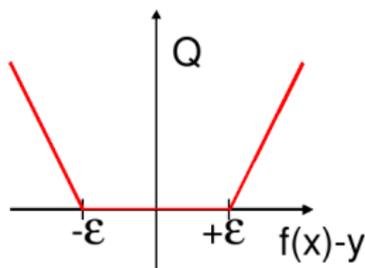
## SVMs für Regression

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

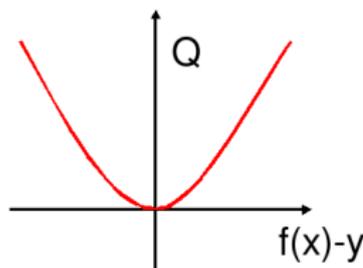
$$L_k(y, f(\vec{x}, \alpha)) = \begin{cases} 0 & \text{, falls } y - f(\vec{x}, \alpha) \leq \varepsilon \\ (y - f(\vec{x}, \alpha) - \varepsilon)^k & \text{, sonst} \end{cases}$$

Die *Loss-Funktion*  $L_1$  gibt den Abstand der Funktion  $f$  von den Trainingsdaten an, alternativ quadratische Loss-Funktion  $L_2$ :

lineare Verlustfunktion



quadratische Verlustfunktion





# SVMs für Regression

Dadurch ergibt sich das 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

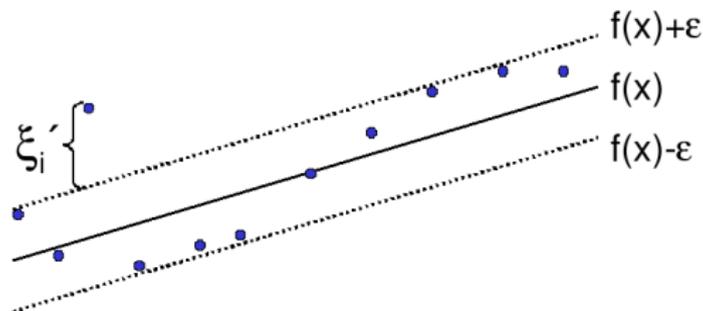
$$f(\vec{x}_i) = \langle \vec{\beta}, \vec{x}_i \rangle + \beta_0 \leq y_i + \epsilon + \xi_i$$

$$f(\vec{x}_i) = \langle \vec{\beta}, \vec{x}_i \rangle + \beta_0 \geq y_i - \epsilon - \xi'_i$$



## SVMs für Regression

Die  $\xi_i$  bzw.  $\xi'_i$  geben für jedes Beispiel Schranken an, innerhalb derer der vorhergesagte Funktionswert für jedes Beispiel liegen soll:



Bei der Lösung des Optimierungsproblems mit Lagrange führt dies zu *zwei*  $\alpha$ -Werten je Beispiel!



## SVMs für Regression

Das duale Problem enthält für jedes  $\vec{x}_i$  je zwei  $\alpha$ -Werte  $\alpha_i$  und  $\alpha'_i$ , je einen für  $\xi_i$  und  $\xi'_i$ , d.h.

### Duales Problem für die Regressions-SVM

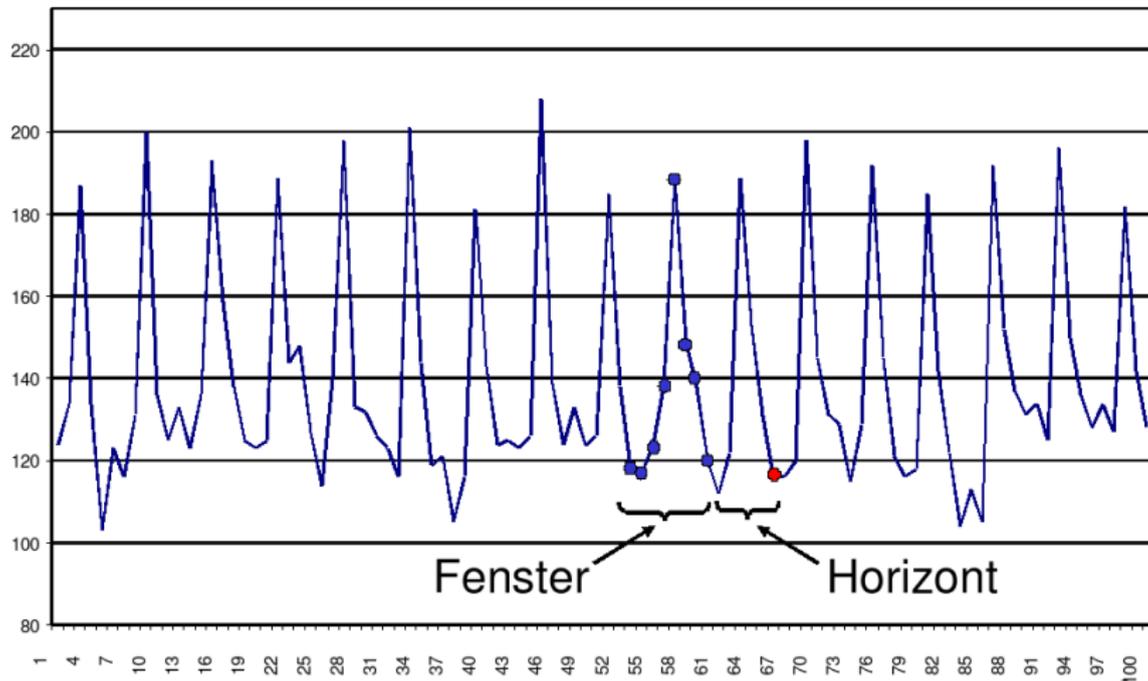
Maximiere

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

unter den Nebenbedingungen

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

# Beispiel: Prognose von Zeitreihen

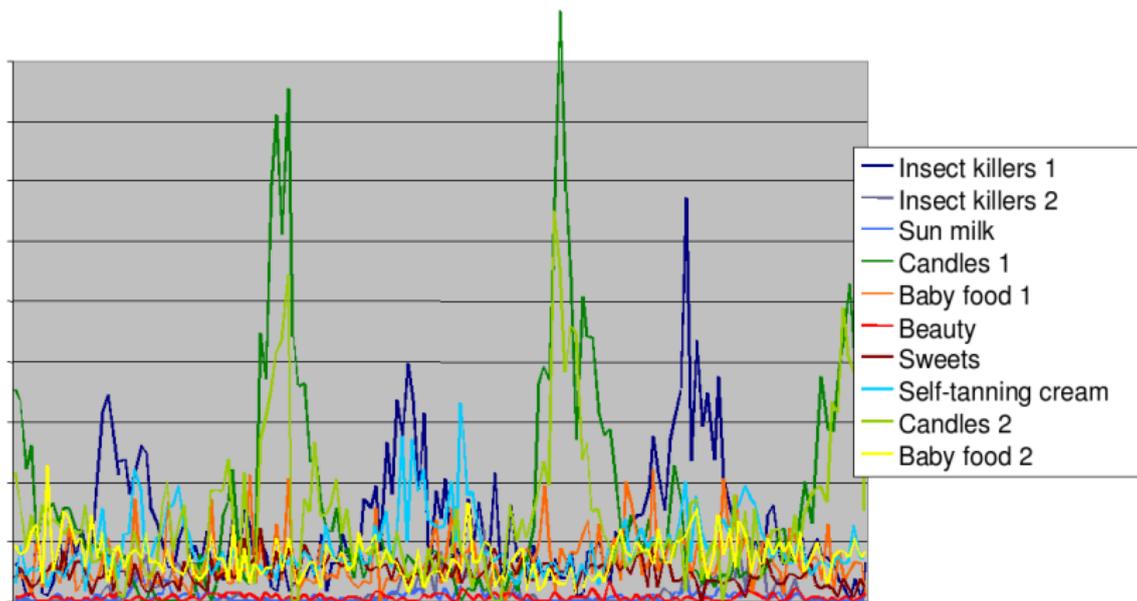




# Prognose von Zeitreihen

- Trend
- Zyklen
- Besondere Ereignisse (Weihnachten, Werbung, ...)
- Wieviel vergangene Beobachtungen?
- Ausreißer

# Abverkauf Drogerieartikel





## 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.



# 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: multivariat nach univariat

### Quasi-SQL:

For all shops for all  
items: Create view

Univariate as

Select shop, week,  
 $item_i$

Where shop="dm<sub>j</sub>"

From Source;

- Multiples  
Lernen für alle  
univariaten  
Zeitreihen

Shop_Item	Week	Sale	Week	Sale
Dm1_Item1	1	4...	104	9
...				
Dm1_Item50	1	12...	104	16
...				
Dm20_Item50	1	14...	104	16



## Vorverarbeitung II

- Problem: eine Zeitreihe ist nur 1 Beispiel!
- Das ist für das Lernen zu wenig.
- Lösung: 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



## SVM im Regressionfall

- Multiples Lernen:  
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

(Diplomarbeit Stefan Rüping 1999)



## Vergleich mit Exponential Smoothing

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

Verlust, nicht normiert auf  $[0, 1]$ !



# Was wissen wir jetzt?

- Anwendung der SVM für die Medikamentenverordnung
- Idee der Regressions-SVM
- Anwendung der SVM für die Verkaufsvorhersage
  - Umwandlung multivariater Zeitreihen in mehrere univariate
  - Gewinnung vieler Vektoren durch gleitende Fenster
  - Asymmetrische Verlustfunktion



## 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



## Aufgaben

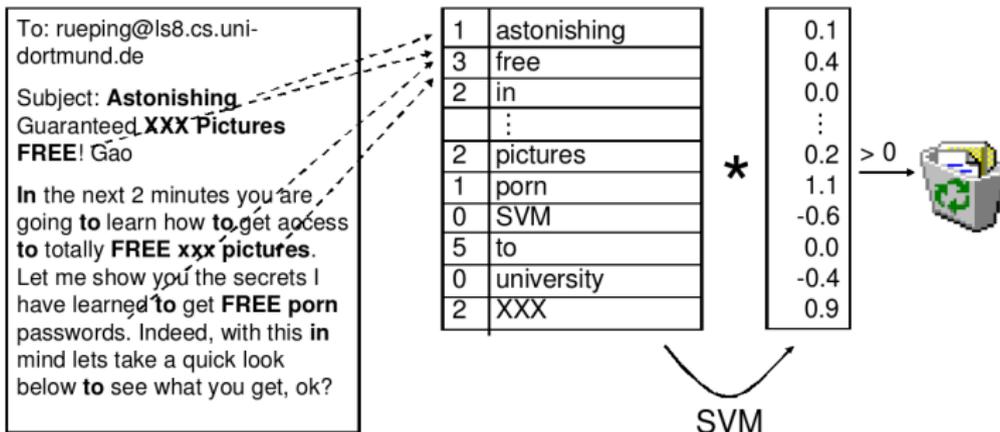
- Indexierung möglichst vieler Seiten (Google)
- Suche nach Dokumenten, ranking der Ergebnisse z.B. nach Häufigkeit der Verweise auf das Dokument (PageLink – Google)
- Kategorisierung (Klassifikation) der Seiten manuell (Yahoo), automatisch
- Strukturierung von Dokumentkollektionen (Clustering)
- Personalisierung: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



# Information Retrieval

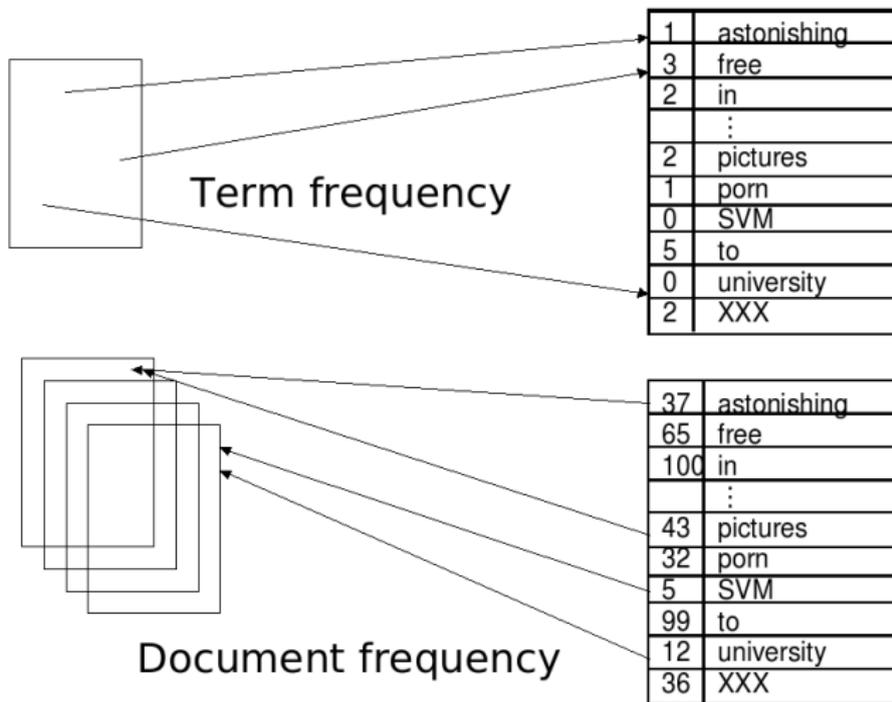
- Ein Dokument besteht aus einer Menge von Termen (Wörtern)
  - Bag of words: Vektor, dessen Komponenten die Häufigkeit eines Wortes im Dokument angeben.
- Für alle Dokumente gibt es eine Termliste mit Verweis auf die Dokumente.
  - Anzahl der Dokumente, in denen das Wort vorkommt.

# 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:

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



# Textklassifikation

- Thorsten Joachims "The Maximum-Margin Approach to Learning Text Classifiers Kluwer", 2001
- Modell der Textklassifikation TCat
- Verbindung zur SVM-Theorie

→ theoretisch begründete Performanzabschätzung



# Eigenschaften der Textklassifikation 1

- Hochdimensionaler Merkmalsraum

- Reuters Datensatz mit 9603 Dokumenten: verschiedene Wörter

$$V = 27658$$

- Heapes Gesetz: Anzahl aller Wörter

$$({}_s)V = ks^\beta$$

- 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 die 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
- Euklidische Länge der Vektoren klein!



## Eigenschaften der Textklassifikation 5

- Zipfs Gesetz: Verteilung von Wörtern in Dokumentkolektionen 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.
- Generalisierte Verteilung von Häufigkeit nach Rang (Mandelbrot):  $v$  ist Größe der Dokumentkolektion in Wortvorkommen

$$\frac{v}{(k + r)^\phi}$$



## Plausibilität guter Textklassifikation durch SVM

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

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



## TCat Modell – Prototyp

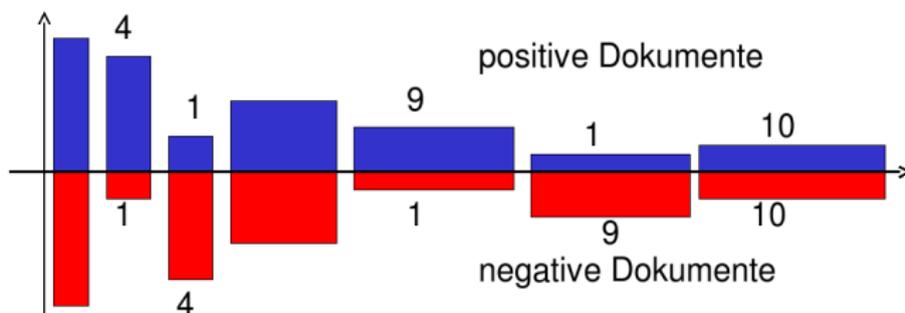
- Hochdimensionaler Raum:  $V = 11100$  Wörter im Lexikon
- Dünn besetzt: Jedes Dokument hat nur 50 Wörter, also mindestens 11050 Nullen
- Redundanz: Es gibt 4 mittelhäufige und 9 seltene Wörter, die die Klasse anzeigen
- Verteilung der Worthäufigkeit nach Zipf/Mandelbrot.
- Linear separierbar mit  $\beta_0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

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



## 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*

$$TCat ([p_1 : n_1 : f_1], \dots, [p_s : n_s : f_s])$$

*describes a binary classification task with  $s$  sets of disjoint features. The  $i$ -th set includes  $f_i$  features. Each positive example contains  $p_i$  occurrences 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!

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



## 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:

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



## 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.



## Verwendung des TCat Modells für zeitgestempelte Daten

Und jetzt wenden wir das Gelernte auf ein Gebiet fernab von Texten an!



# Lokale Muster

- Lokale Muster beschreiben seltene Ereignisse.
- Gegeben ein Datensatz, für den ein globales Modell bestimmt wurde, weichen lokale Muster davon ab.
  - Lokale Muster beschreiben Daten mit einer internen Struktur, z.B. Redundanz, Heterogenität



# Zeit-gestempelte Daten

- Zeit-gestempelte Daten können transformiert werden in:
  - Eine Menge von Ereignissen,
  - Zeitintervalle,
  - Zeitreihen.



# Klassische Methoden

- Zeitreihenanalyse für Vorhersage, Trend und Zyklus Erkennung
- Indexing und clustering von Zeitreihen (time warping)
- Segmentierung (motif detection)
- Entdeckung von Episoden
  - frequent sets,
  - chain logic programs (grammars)
- Regression



## Beispielrepräsentation

- Die Beispielrepräsentation  $X$  bestimmt die Anwendbarkeit der Methoden: welche Variablen, was sind Beispiele?
- Bedeutung der Repräsentation lange unterschätzt.
- Suche nach guter Repräsentation ist aufwändig.
- Transformieren der Rohdaten in die Repräsentation auch.



## Einige Repräsentationen für zeitgestempelte Daten

- Schnappschuss: ignoriere Zeit, nimm nur den aktuellen Zustand. (So war es bei der Intensivmedizin-Anwendung.)
- Ereignisse mit Zeitintervallen: aggregiere Zeitpunkte zu Intervallen, wende frequent set mining an. (Das machen wir in dieser Vorlesung nicht.)
- Generierte Merkmale: hier: transformiere Zeitinformation in Häufigkeitsmerkmale!



## Häufigkeitsmerkmale für Zeitaspekte

- Term frequency: wie oft änderte Attribut  $A$  seinen Wert  $a_i$  für ein Objekt  $c_j$ .

$$tf(a_i, c_j) = \| \{x \in \text{timepoints} \mid a_i \text{ of } c_j \text{ changed} \} \|$$

- Document frequency: in wie vielen Objekten  $c_j$  änderte Attribut  $A$  seinen Wert  $a_i$ .

$$df(a_i) = \| \{c_j \in C \mid a_i \text{ of } c_j \text{ changed} \} \|$$

- TF/IDF:

$$tfidf(a_i) = tf(a_i, c_j) \log \frac{\|C\|}{df(a_i)}$$



# Fallstudie SwissLife

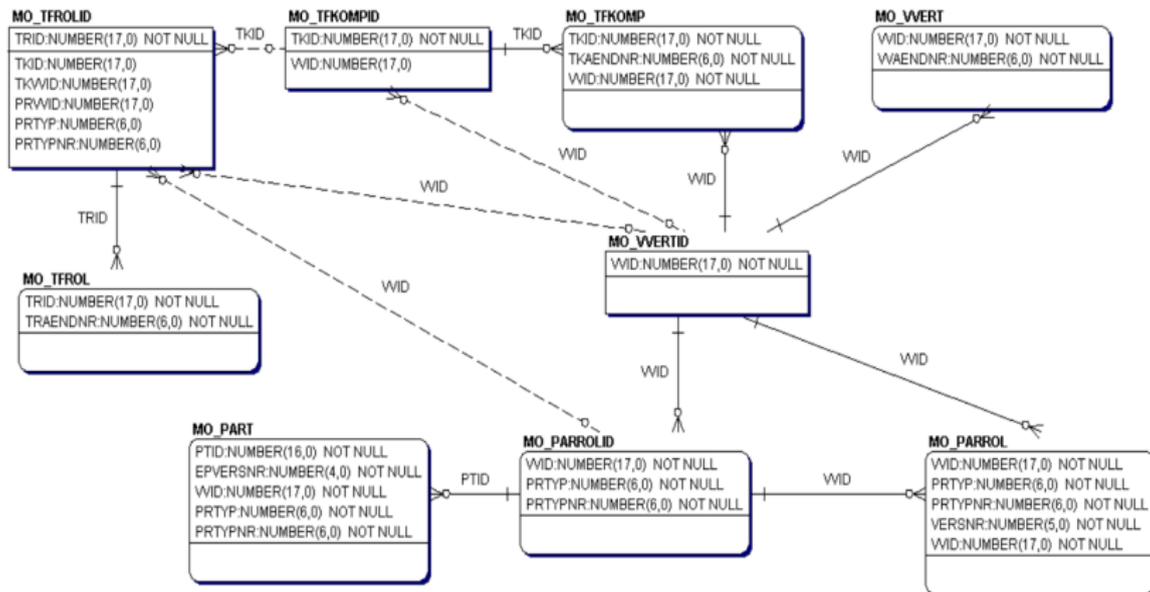
- Lokale Muster
  - Seltenes Ereignis der Kündigung
  - Lokales Muster weicht ab vom generellen Modell
  - Interne Struktur in lokalen Mustern
- Zeit-gestempelte Daten
  - Schnappschuss
  - Zeitintervall
  - Generierte Merkmale: *TFIDF*



## Lokale Muster in Versicherungsdaten

- Nur 7.7% der Verträge enden vorzeitig (customer churn).
- Für einige Attribute weicht die likelihood in der churn-Klasse von der globalen ab.
- Interne Struktur:
  - Überlappung: häufige Mengen in churn Verträgen sind auch häufig in fortgesetzten Verträgen.
  - Redundanz: in jedem Vertrag gibt es mehrere Attribute, die auf Fortsetzung oder Kündigung hinweisen.
  - Heterogenität: Es gibt gekündigte Verträge, die nicht ein einziges Attribut gemeinsam haben.

# Database



# Contract Table

VVID	VVAENDNR	VVWIVON	VVWIBIS	VVAENDAT	VVAENDART	...
16423	1	1946	1998	1946	1000	
16423	2	1998	1998	1998	27	
16423	3	1998	1998	1998	4	
16423	4	1998	1998	1998	54	
16423	5	1998	1998	1998	4	
16423	6	1998	9999	1998	61	
5016	1	1997	1999	1997	33	
5016	2	1999	2001	1999	33	
5016	3	2001	2001	2001	33	
5016	4	2001	2001	2001	33	
5016	5	2001	2002	2001	81	
5016	6	2002	9999	2001	94	
...	...	...	...	...	...	...



# Datensatz

- Tabellen enthalten Informationen über
  - 217586 Komponenten and
  - 163745 Kunden
- Attribute:
  - 14 Attributes ausgewählt
  - Eines der Attribute gibt den Grund an für einen Wechsel.  
Es gibt 121 Gründe. Daraus werden 121 Boolean Attribute.
  - 134 Attribute mit *TFIDF* Werten.



## Erste Experimente

- Bei SwissLife wurde die Abweichung der Wahrscheinlichkeit bestimmter Attributwerte in gekündigten und fortgesetzten Verträgen festgestellt anhand der Schnappschussrepräsentation → keine operationale Vorhersage.

# Calculating Term Frequency

VVID	...	VVSTACD	VVPRFIN	VVPRZA	VVINKZWEI	VVBEG	VVEND	VVINKPRL	...
16423		4	1	2	2	1946	1998	295.29	
16423		4	1	2	2	1946	1998	295.29	
16423		4	5	2	0	1946	2028	0	
16423		5	3	2	0	1946	2028	0	
16423		4	1	2	2	1946	1998	295.29	
16423		5	3	2	0	1946	1998	0	

3	VVSTACD
4	VVPRFIN
0	VVPRZA
3	VVINKZWEI
0	VVBEG
2	VVEND
3	VVINKPRL



## Experimente mit der TFIDF Repräsentation

- Vergleich der originalen Repräsentation und der TFIDF
  - 10fold cross validation
    - Apriori mit Konklusion 'churn'
    - Entscheidungsbaumlerner J4.8
    - Naive Bayes
    - mySVM mit linearem Kern
  - F-measure balanciert precision und recall gleich.

Alle Lernalgorithmen werden besser mit der *TFIDF*-Repräsentation.



## Resultate (F-measure)

<b>Lerner</b>	<b>TF/IDF repr.</b>	<b>Original repr.</b>
Apriori	63.35	30.24
J4.8	99.22	81.21
Naive Bayes	51.8	45.41
mySVM	97.95	16.06



# Erklärung?

- TF/IDF stammt aus Lernen über Texten.
- Dazu gibt es eine Theorie – TCat.
- Können wir die auch hier einsetzen??



# Datenbeschreibung im TCat Modell

$$TCat(\underbrace{[2 : 0 : 2], [1 : 4 : 3]}_{\text{high frequency}}, \underbrace{[3 : 1 : 3], [0 : 1 : 4]}_{\text{medium frequency}}, \underbrace{[1 : 0 : 19], [0 : 1 : 64]}_{\text{low frequency}}, \underbrace{[1 : 1 : 39]}_{\text{rest}})$$

$[1 : 4 : 3]$  : Aus der Menge von 3 Merkmale finden wir ein Auftreten in positiven und 4 in negativen Beispielen.



# Learnability of TCat

## Error bound (Joachims 2002)

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

$$a = \sum_{i=1}^s \frac{p_i^2}{f_i} = 5.41$$

$$b = \sum_{i=1}^s \frac{p_i^2 n_i}{f_i} = 2.326$$

$$c = \sum_{i=1}^s \frac{n_i^2}{f_i} = 5.952$$

$$R^2 = \sum_{r=1}^d \left( \frac{c}{(r+k)^\phi} \right)^2 \leq 37$$

Nach 1000 Beispielen erwarteter Fehler  $\leq 2.2\%$

Tatsächlicher Fehler 2.05%



## Experimente zu lokalen Mustern

- Durch TCat-Konzepte Daten künstlich generieren.
- Lokale Muster als seltene Ereignisse mit interner Struktur.



## Lokale Muster: Verzernte Verteilung

- 10 000 Beispiele mit 100 Attributen
- SVM runs mit 10 fold cross validation

<i>Repr.</i>	<i>Targetconcept :</i>	<b>Verzerrung:</b>
TF/IDF	1. change of a particular attribute	50%, 25%,
Boolean	2. frequency of changes	12.5%, 6.25%



## Lokale Muster: Strukturen

- 10 000 Beispiele mit 100 Attributen
- 20 Attribute wechseln pro Beispiel (dünn besetzt)
- Variieren:
  - Heterogenität:  $\frac{f_i}{p_i}$  Beispiele der selben Klasse haben kein gemeinsames Attribut 4, 5, 10, 20
  - Redundanz:  $\frac{p_i}{f_i}$  oder  $\frac{n_i}{f_i}$  für die Redundanz innerhalb einer Klasse 0.5, 0.2, 0.1
  - Überlappung: einige Attribute sind häufig in beiden Klassen 0.25, 0.66



## Resultate

- Für alle Kombinationen ohne Überlappung sind die Lernergebnisse 100% in Boolean und im TF/IDF- Format.
- Mehr Überlappung verschlechtert das Lernen bei Boolean auf 68.57% F-measure.
- Für alle Kombinationen (auch mit großer Überlappung) erreicht das Lernen mit TF/IDF Daten 100% precision und recall.



## Navigation im Raum der Beispiele

- Zunehmende Größe des Datensatzes zeitgestempelter Daten: Schnappschuss < Intervalle < Boolean < TF/IDF
- TF/IDF ist günstig für lokale Muster, wenn diese Redundanz, Heterogenität als Eigenschaft aufweisen.
- Berechnung des TCat Modells für gegebene Daten implementiert → Fehlerschranke angebbbar.



## Was wissen Sie jetzt?

- Lokale Muster haben manchmal die typische TCat-Struktur.
- Sie haben gesehen, wie manche zeitgestempelte Datenbanken in TCat-Modelle transformiert werden können.
- Die Lernbarkeit mit linearer SVM der so transformierten Daten können Sie ausrechnen.