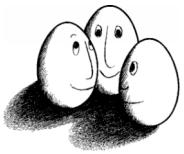


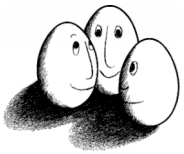
Kernfunktionen

Wie funktioniert der Kern-Trick?

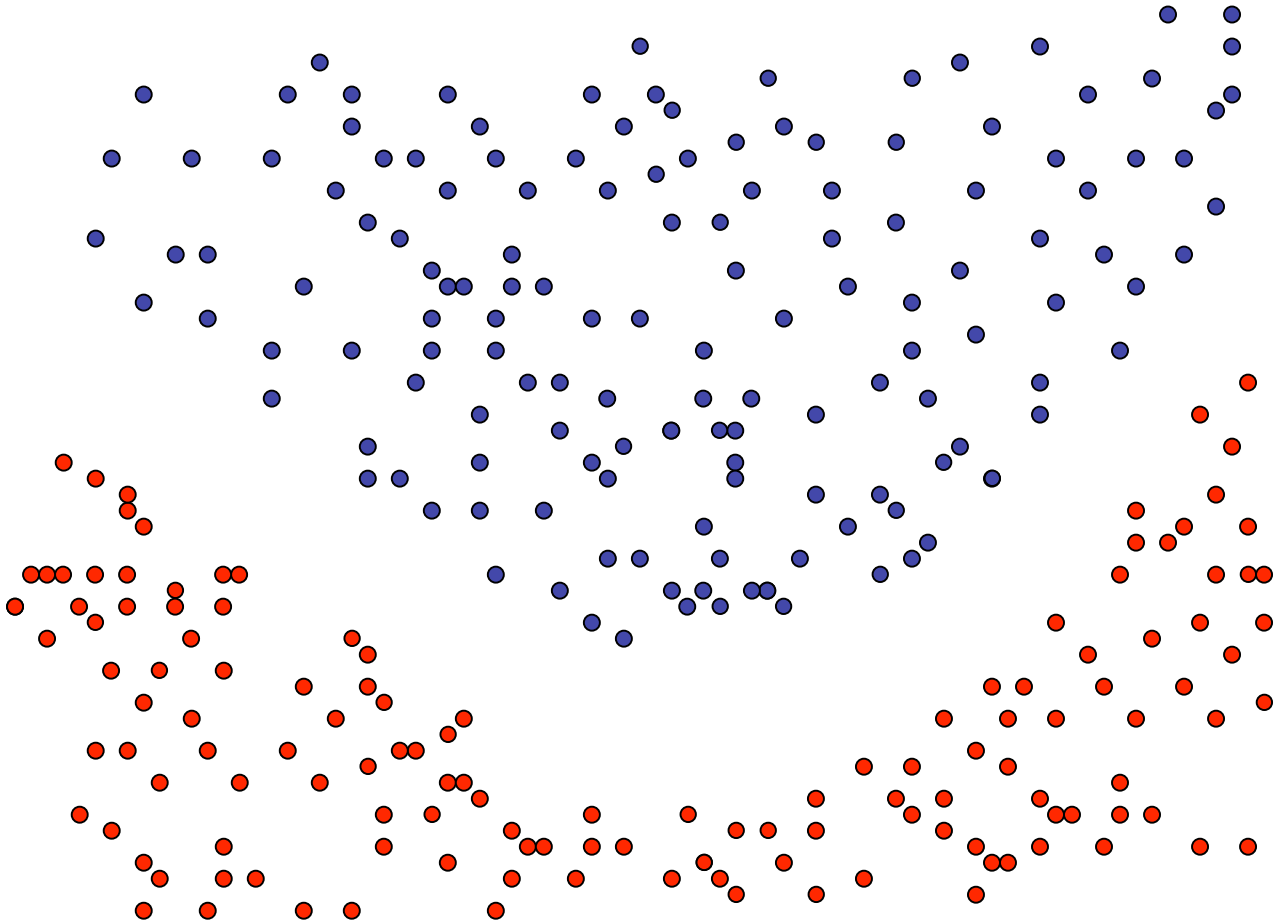
Wann funktioniert der Kern-Trick?

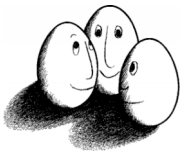
Warum?

Prof. Dr. Katharina Morik



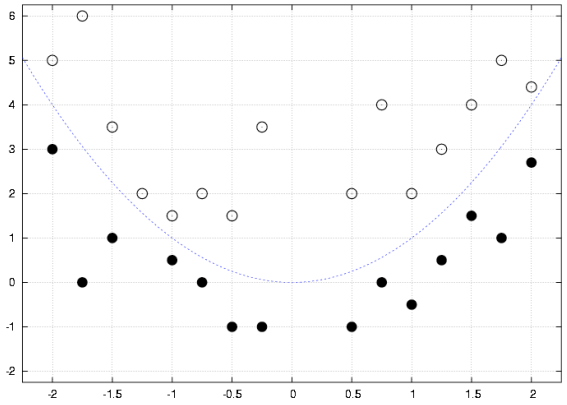
Nicht-lineare Daten



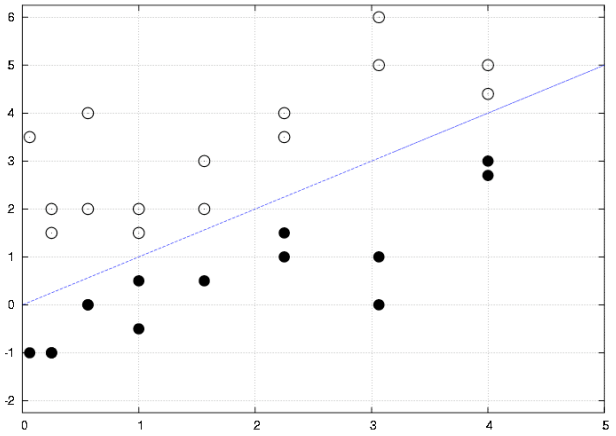


Nicht-lineare Daten

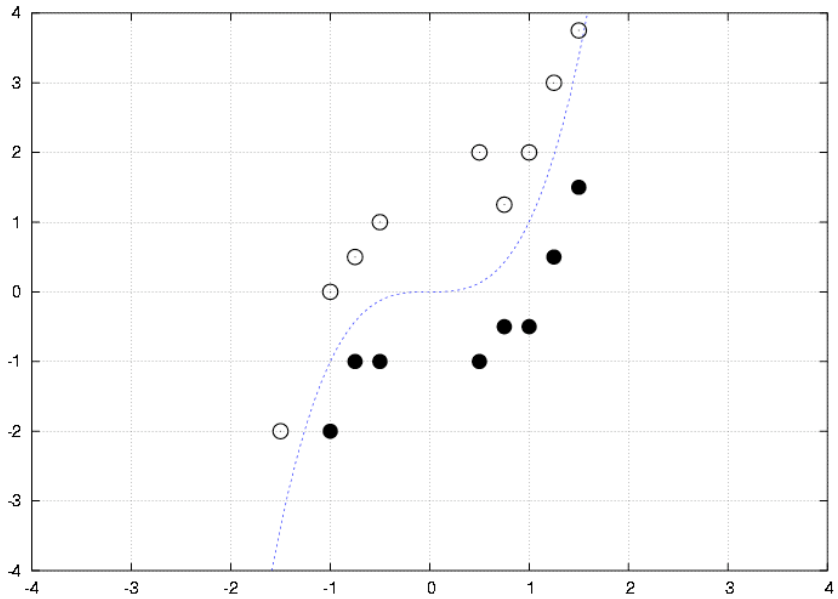
- Neue SVM-Theorie entwickeln? (Neeee!)
- Lineare SVM benutzen? („*If all you've got is a hammer, every problem looks like a nail*")
- Transformation in lineares Problem!

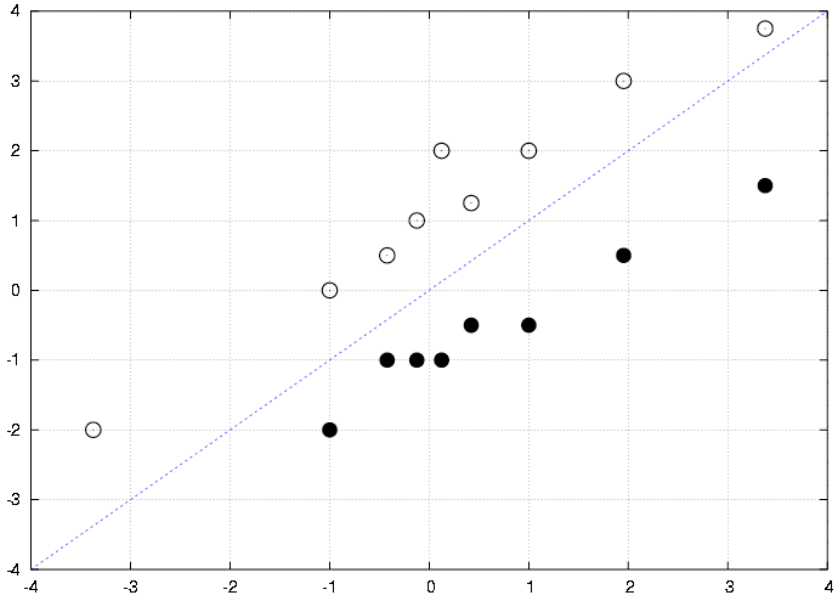


$$\Phi(x_1,x_2) = (x_1^2,x_2)$$

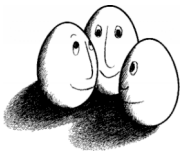


Beispiel





$$\Phi(x_1,x_2)=(x_1^3, x_2)$$

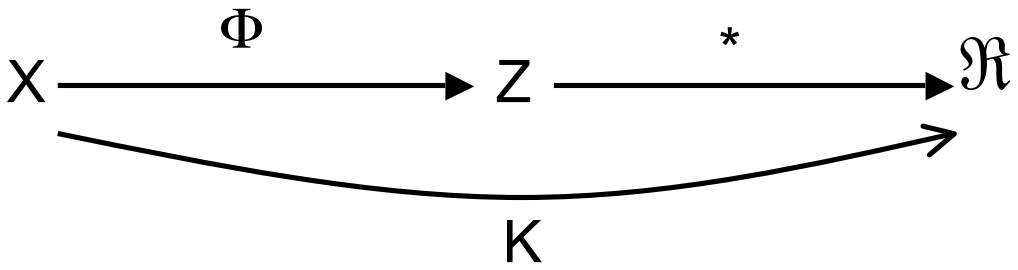


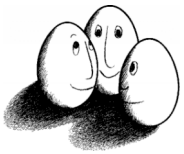
Kernfunktionen

- Erinnerung: $$L(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} y_i y_j \alpha_i \alpha_j (x_i * x_j)$$

$$f(x) = \sum \alpha_i y_i (x_i * x) + b$$

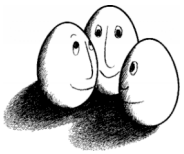
- SVM hängt von x nur über Skalarprodukt x*x‘ ab.
- Ersetze Transformation Φ und Skalarprodukt * durch Kernfunktion $K(x_1,x_2) = \Phi(x_1)*\Phi(x_2)$





Der Kern-Trick

- K(x,x‘)=x*x‘
- Φ: X → $\mathcal{H}$
- Das Skalarprodukt der Vektoren im Merkmalsraum $\mathcal{H}$ entspricht dem Wert der Kernfunktion über den Beispielen.
- Welche Funktionen machen k(x,x‘)= Φ(x)* Φ(x‘) wahr?



Polynome

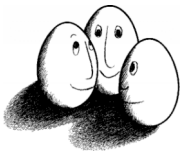
- Ein Monomial hat nur einen Term: x^d
- Alle Produkte von d Vektorkomponenten ergeben Merkmalsraum $\mathcal{H}$.

  $\Phi_2: \Re^2 \rightarrow \mathcal{H}^3$

  $$([x]_1, [x]_2) \rightarrow ([x]_1^2, [x]_2^2, [x]_1[x]_2)$$

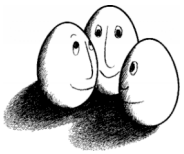
- Sei N die Anzahl der Dimensionen (Buchstaben), d Dimensionen werden daraus verwendet (Länge d. Wortes), dann ist die Anzahl verschiedener Monomials (Wörter):

$$\binom{d+N-1}{d} = \frac{(d+N-1)!}{d!(N-1)!}$$



Der Trick

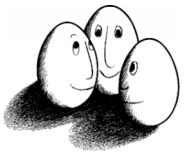
- Der Merkmalsraum ist sehr groß.
- Abbildung der Beispiele in den Merkmalsraum und dann Berechnen der Skalarprodukte zwischen den transformierten Beispielen ist sehr ineffizient.
- Eine Kernfunktion, die auf die Beispiele direkt angewandt dasselbe Ergebnis liefert, wäre effizient!
- Diese Kernfunktion ist hier $(x*x')^2$



Skalarprodukt im Polynomraum

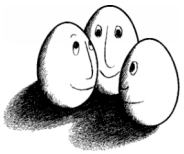
- Geordnete Monomials:
  $\Phi_2: \Re^2 \rightarrow \mathcal{H}^4$
  $$\Phi_2(x): ([x]_1, [x]_2) \rightarrow ([x]_1^2, [x]_2^2, [x]_1[x]_2, [x]_2[x]_1)$$
- Skalarprodukt im Merkmalsraum:
  $$\begin{aligned} \Phi_2(x) * \Phi_2(x') &= [x]_1^2[x']_1^2 + [x]_2^2[x']_2^2 + 2[x]_1[x]_2[x']_1[x']_2 \\ &= (x * x')^2 \end{aligned}$$

- Dies gilt allgemein für alle geordneten Produkte d-ten Grades der Komponenten von x: $\Phi_d(x)^* \Phi_d(x') = (x^*x')^d$



Beweis

$$
\begin{aligned}
k(x,x') &= \Phi_d(x) * \Phi_d(x') = (x * x')^d \\
\Phi_d(x) * \Phi_d(x') &= \sum_{j_1=1}^{N} \ldots \sum_{j_d=1}^{N} [x]_{j_1} \cdot \ldots \cdot [x]_{j_d} \cdot [x']_{j_1} \cdot \ldots [x']_{j_d} \\
&= \sum_{j_1=1}^{N} [x]_{j_1} \cdot [x']_{j_1} \ldots \sum_{j_d=1}^{N} [x]_{j_d} \cdot [x']_{j_d} \\
&= \left(\sum_{j=1}^{N} [x]_j \cdot [x']_j \right)^d \\
&= (x * x')^d
\end{aligned}
$$

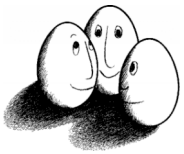


Randbemerkung

- Gerade wurden Reihenfolgen unterschieden.
- Üblicherweise ist Φ_d aber ohne Doppelte.
- Eine Komponente, die Doppelte enthalten würde, wird durch die Wurzel skaliert.

$$\Phi_2(x) = \left([x]_1^2, [x]_2^2, \sqrt{2}[x]_1[x]_2 \right)$$

- Die genaue Form von Φ_d ist aber egal: beide ergeben dieselbe Kernfunktion $k(x,x') = (x * x')^d$



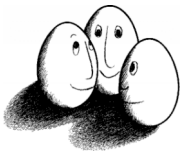
Verallgemeinerung

- Die Gram Matrix K für eine Funktion k:$X^2 \rightarrow \Re$ und Beobachtungen $x_1,..., x_m$ ist eine m×m Matrix
  $K_{ij}:=k(x_i, x_j)$
- Eine (reellwertige) positiv definite Matrix ist eine m×m Matrix K, für die für alle c_i in $\Re$ gilt

$$\sum_{i,j} c_i c_j K_{ij} \geq 0$$

- Einfaches Beispiel: 1) gelingt nicht, 2) gelingt

$$K = \begin{pmatrix} 0 & -1 \\ -1 & 0 \end{pmatrix}$$ ci, cj = 1 $$K = \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix}$$ ci, cj = 1

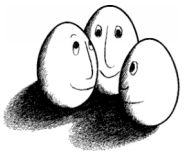


Kernfunktion Definition

- Eine Funktion k über X×X, die sich für alle x_i in X als positiv definite Gram Matrix mit symmetrischer Funktion k darstellen lässt, heißt Kernfunktion oder reproduzierender Kern oder Kovarianzfunktion.

- Bei Hilbert: Eine Funktion k, die zu dem Operator T_k führt durch $(T_k f)(x) = \int_x k(x, x') f(x') dx'$

  heißt Kern von T_k.

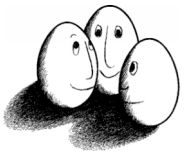


Konstruieren in 2 Richtungen

- Gegeben eine Kernfunktion k, konstruiere einen Merkmalsraum, in den Φ abbildet.
  Jede Kernfunktion kann als Skalarprodukt in einem anderen Raum betrachtet werden.
- Gegeben ein Merkmalsraum Φ mit Skalarprodukt, konstruiere eine Kernfunktion $k(x,x') = \Phi(x)^* \Phi(x')$.
  Gelingt, weil für alle c_i in $\Re$ und x_i in X gilt:

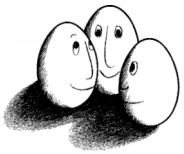
$$\sum_{i,j} c_i c_j k(x_i, x_j) = \sum_i c_i \Phi(x_i) * \sum_j c_j \Phi(x_j)$$

$$= \left\| \sum_i c_i \Phi(x_i) \right\|^2 \geq 0$$

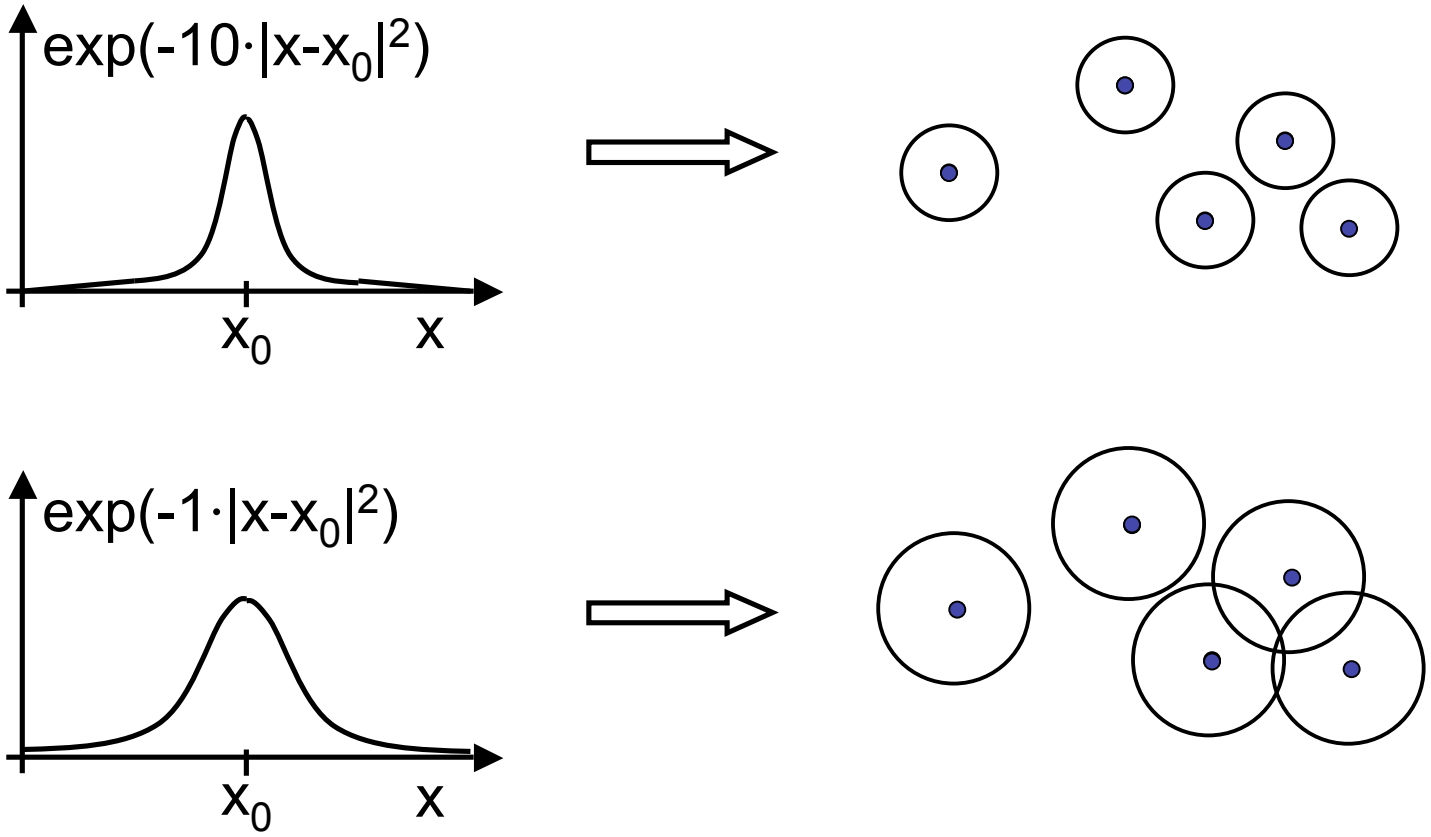


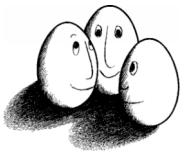
Kernfunktionen praktisch

- Angabe von Φ nicht nötig, einzige Bedingung: Kernmatrix $(K(x_i,x_j))_{i,j=1...n}$ muss positiv definit sein.
- Polynom: $K(x,x') = (x*x')^d$
- Radial-Basisfunktion: $K(x,x') = \exp(-\gamma||x-x'||^2)$
- Neuronale Netze: $K(x,x') = \tanh(\alpha \cdot x*x'+b)$
- Konstruktion von Spezialkernen durch Summen und Produkte von Kernfunktionen, Multiplikation mit positiver Zahl, Weglassen von Attributen



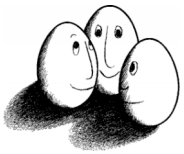
RBF-Kernfunktion





Eigenschaften von RBF-Kernen

- Allgemeine Form: k(x, x‘)=f(d(x,x‘))
- Funktion f kann außer Gauss z.B. auch B-Spline sein.
- Als Metrik d(x,x‘) wird auch gewählt $\|x - x'\| = \sqrt{(x - x') * (x - x')}$
- Bei $||x-x'||^2$ oder ||x-x‘|| ist die RBF-Kernfunktion invariant bezüglich Drehung und Verschiebung.
- Das bedeutet, dass das Lernergebnis unabhängig von dem Koordinatensystem unserer Daten ist.



Mercer Bedingung

- Es gibt eine Abbildung Φ und eine Kernfunktion

$$k(x,x') = \sum_i \Phi([x]) * \Phi([x'])$$

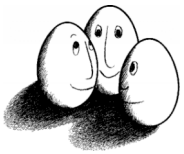
gdw. für jedes g(x) mit finitem

$$\int g(x)^2 dx$$

gilt:

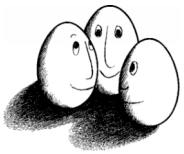
$$\int k(x,x') g(x) g(x') dx\, dx' \geq 0$$

- Wenn die Mercer Bedingung nicht gilt, könnte die Hesse Matrix über den Beispielen indefinit werden.



Was wissen Sie jetzt?

- Kernfunktionen berechnen das Skalarprodukt der Beobachtungen in einem Merkmalsraum, ohne tatsächlich erst in den Merkmalsraum abzubilden. $k(x,x') = \Phi(x) * \Phi(x')$
- Polykern und RBF-Kern als Beispiele.
- Der Kern-Trick: k(x,x‘) lässt sich allein aus x*x‘ berechnen.
- Eine Funktion $X \times X$, die sich für alle x_i in X als positiv definite Gram Matrix mit symmetrischer Funktion k darstellen lässt, heißt Kernfunktion.
- Die Mercer Bedingung prüft, ob es sich um eine Kernfunktion handelt, also die Matrix positiv definit ist.

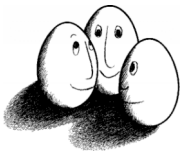


Duales weiches Optimierungsproblem

- Maximiere

$$L(\alpha) = \sum_{i=1}^{m} \alpha - \sum_{i=1}^{m} \sum_{j=1}^{m} y_i y_j \alpha_i \alpha_j x_i * x_j$$

$$u.d.Bedingungen \sum_{i=1}^{m} y_i \alpha_i = 0, \forall i : 0 \le \alpha_i \le C$$



Optimierungsproblem mit Kern

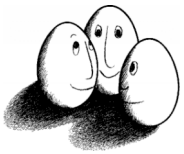
- Erst minimierten wir w, dann maximierten wir das duale Problem, jetzt minimieren wir das duale Problem, indem wir alles mit –1 multiplizieren...
- Minimiere L'(α) mit

$$L'(\alpha) = \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m} y_i y_j K(x_i, x_j)\alpha_i\alpha_j - \sum_{i=1}^{m}\alpha_i$$

unter den Nebenbedingungen

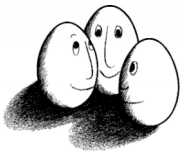
$$0 \le \alpha_i \le C$$

$$\sum_{i=1}^{m} y_i\alpha_i = 0$$



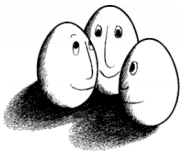
Chunking

- Beispiele x_i mit $\alpha_i = 0$ können aus der Matrix gestrichen werden.
  - Finde alle diese Beispiele, lösche sie.
  - Löse das Optimierungsproblem für die verbleibenden.
- Iteratives Vorgehen:
  - Löse das Optimierungsproblem für die $\alpha_i \neq 0$ aus dem vorigen Schritt und einige Beispiele, die die KKT-Bedingungen verletzen.
- Osuna, Freund, Girosi (1997): feste Matrixgröße.



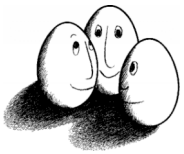
Algorithmus für das Optimierungsproblem

- Berechnen wir L'(α) durch Gradientensuche!
  - Naiver Ansatz berechnet Gradienten an einem Startpunkt und sucht in angegebener Richtung bis kleinster Wert gefunden ist. Dabei wird immer die Nebenbedingung eingehalten. Bei m Beispielen hat α m Komponenten, nach denen es optimiert werden muss. Alle Komponenten von α auf einmal optimieren? m^2 Terme!
  - Eine Komponente von α ändern? Nebenbedingung verletzt.



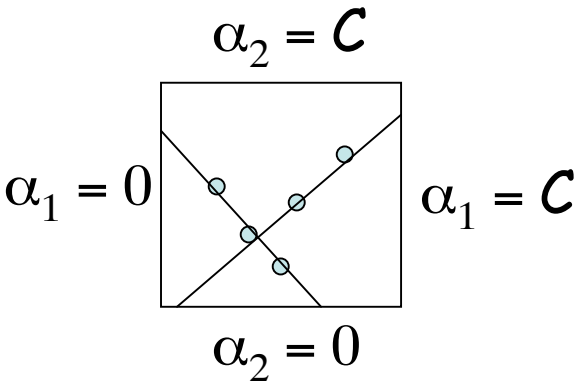
Sequential Minimal Optimization

- Zwei Komponenten α_1, α_2 im Bereich[0,C]x[0,C] verändern!
  - Optimieren von zwei α_i
  - Auswahl der α_i



KKT-Bedingungen einfach

- Notwendige und hinreichende Bedingungen an die Lösung des Optimierungsproblems: für alle i
  - $\alpha_i = 0$ gdw. $y_i f(x_i) \geq 1$
  - $\alpha_i = C$ gdw. $y_i f(x_i) \leq 1$
  - $0 < \alpha_i < C$ gdw. $y_i f(x_i) = 1$

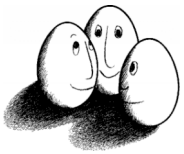


$y_1 = y_2$
$\alpha_1 - \alpha_2$

$y_1 \neq y_2$
$\alpha_1 + \alpha_2$

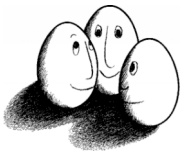
$$0 \leq \alpha_i \leq C$$

$$\sum_{i=1}^{m} y_i \alpha_i = 0$$



α_2 optimieren

- Maximum der Funktion L'(α) entlang der Geraden $g(\alpha_2)=s\ \alpha_2 + \alpha_1$ mit $s= y_2/y_1$
- Wenn $y_1=y_2$ ist s=1, also steigt die Gerade.
  Sonst s=-1, also fällt die Gerade.
- Schnittpunkte der Geraden mit dem Bereich[0,C]x[0,C]:
  - Falls s steigt: max(0; $\alpha_2 + \alpha_1 - C$) und min(C; $\alpha_2 + \alpha_1$)
  - Sonst: max(0; $\alpha_2 - \alpha_1$) und min(C; $\alpha_2 - \alpha_1 + C$)
  - Optimales α_2 ist höchstens max-Term,
    mindestens min-Term.



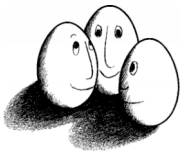
SMO

- Berechne α_2 und gib die Schnittpunkte max, min der Diagonalen mit der Box an.
- 2.Ableitung von L‘ entlang der Diagonalen
  $\eta = k(x_1, x_1) + k(x_2, x_2) - 2k(x_1, x_2)$
- Wenn η>0, wird das Minimum für α_2 ausgerechnet, wobei E der Fehler f(x)- y ist:

$$\alpha_2^{neu} = \alpha_2 + \frac{y_2(E_1 - E_2)}{\eta}$$

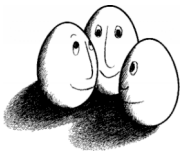
- Beschneiden, so dass min ≤ $\alpha_2^{neu‘}$ ≤ max
- Berechnen

$$\alpha_1^{neu} = \alpha_1 + y_1 y_2 (\alpha_2 - \alpha_2^{neu'})$$



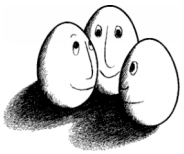
Randbemerkung

- Ein nicht positives η kann dann auftreten, wenn
  - zwei Beispiele genau gleich aussehen oder
  - die Kernfunktion nicht der Mercer-Bedingung gehorcht.



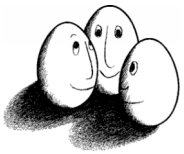
Algorithmus

- Äußere Schleife -- α_1 wählen
  1. Alle Beispiele durchgehen:
     welche verletzen KKT-Bedingungen?
  2. Non-bound Beispiele suchen (α_1 weder 0 noch C):
     welche verletzen KKT-Bedingungen?
     Verändern bis alle non-bound Beispiele KKT-Bedingungen erfüllen!
  3. Goto 1
- Innere Schleife -- α_2 wählen
  1. Wenn $E_1 > 0$, Beispiel mit kleinem E_2 suchen,
     wenn $E_1 < 0$, Beispiel mit großem E_2 suchen.
  2. L‘(α) ausrechnen
  3. b ausrechnen



Satz von Osuna

- Der Algorithmus konvergiert, solange an jedem Schritt 2 Lagrange Multiplikatoren optimiert werden und mindestens einer davon verletzte vorher die KKT-Bedingungen.



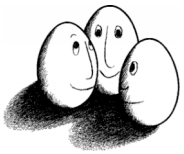
Was wissen Sie jetzt?

- Das Optimierungsproblem wird durch Optimieren je zweier Lagrange-Multiplikatoren gelöst, äußere Schleife wählt ersten, innere zweiten Multiplikator.
- Sei $\alpha=(\alpha_1,\ldots,\alpha_m)$ eine Lösung des Optimierungsproblems. Wir wählen zum update:

$$\hat{\alpha}_2 = \alpha_2 + \frac{y_2\left(\left(f(x_1)-y_1\right)-\left(f(x_2)-y_2\right)\right)}{K(x_1,x_1)-2K(x_1,x_2)+K(x_2,x_2)}$$

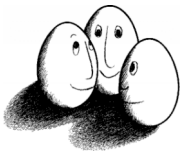
- Optimales $\hat{\alpha}_1 = \alpha_1 + y_1 y_2(\alpha_2 - \hat{\alpha}_2)$

- Prinzip des Optimierens: Nullsetzen der ersten Ableitung...
- Der Algorithmus konvergiert, wenn vorher ein α KKT-Bedingung verletzte.



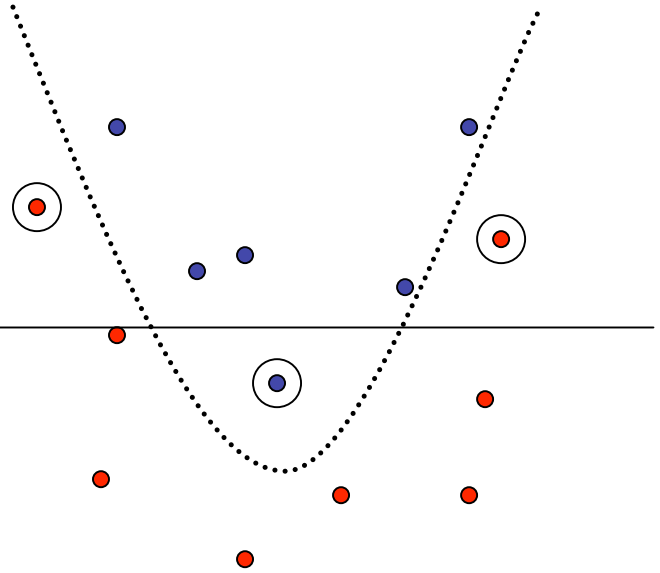
Was ist gutes Lernen?

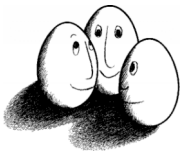
- Fauler Botaniker:
  "klar ist das ein Baum – ist ja grün."
  - Übergeneralisierung
  - Wenig Kapazität
  - Bias
- Botaniker mit fotografischem Gedächtnis:
  "nein, dies ist kein Baum, er hat 15 267 Blätter und kein anderer hatte genau so viele."
  - Overfitting
  - Viel Kapazität
  - Varianz
- Kontrolle der Kapazität!



Bias-Varianz-Problem

- Zu kleiner Hypothesenraum: Zielfunktion nicht gut genug approximierbar (Bias)
- Zu großer Hypothesenraum: Zuviel Einfluss zufälliger Abweichungen (Varianz)
- Lösung: Minimiere obere Schranke des Fehlers:
  $R(\alpha) \leq_{\eta} R_{emp}(\alpha) + Var(\alpha)$

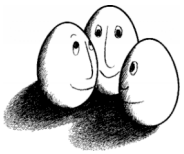




Risikoschranke nach Vapnik

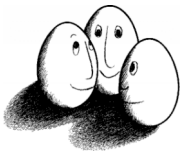
- Gegeben eine unbekannte Wahrscheinlichkeitsverteilung P(x,y) nach der Daten gezogen werden. Die Abbildungen x→ f(x, α) werden dadurch gelernt, dass α bestimmt wird. Mit einer Wahrscheinlichkeit 1-μ ist das Risiko R(α) nach dem Sehen von l Beispielen beschränkt:

$$R(\alpha) \leq R_{emp}(\alpha) + \underbrace{\sqrt{\frac{\eta\left(\log(2l/\eta)+1\right) - \log\left(\mu/4\right)}{l}}}_{\text{VC confidence}}$$



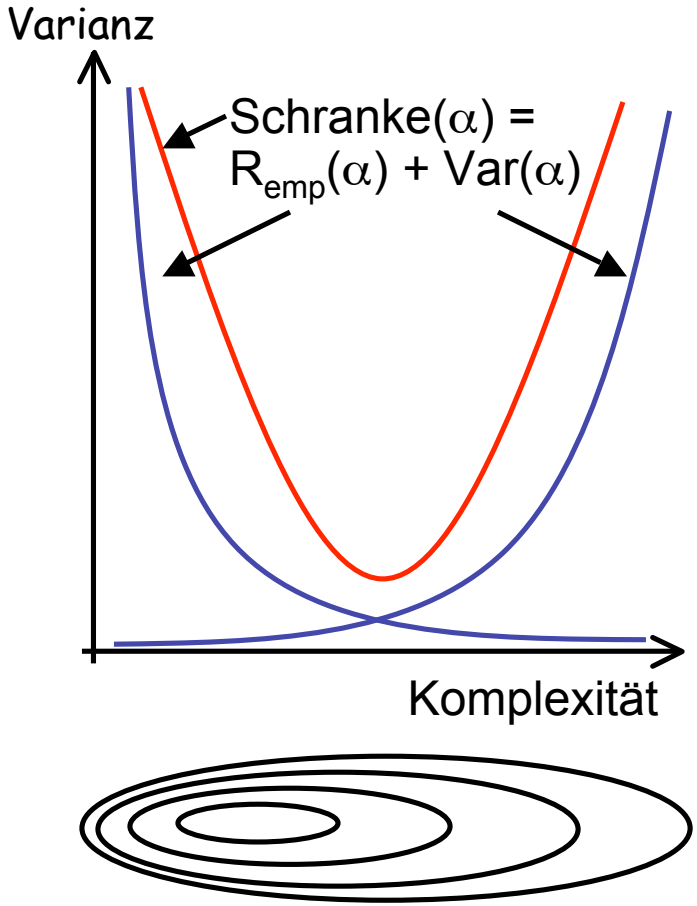
Strukturelle Risikoschranke

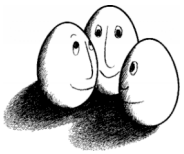
- Unabhängig von einer Verteilungsannahme. Alles, was die Schranke braucht, ist, dass Trainings- und Testdaten gemäß der selben Wahrscheinlichkeits- verteilung gezogen werden.
- Das tatsächliche Risiko können wir nicht berechnen.
- Die rechte Seite der Ungleichung können wir berechnen, sobald wir η kennen.
- Gegeben eine Menge Hypothesen für $f(x,\alpha)$, wähle immer die mit dem niedrigsten Wert für die rechte Seite der Schranke (R_{emp} oder VC confidence niedrig).



Strukturelle Risikominimierung

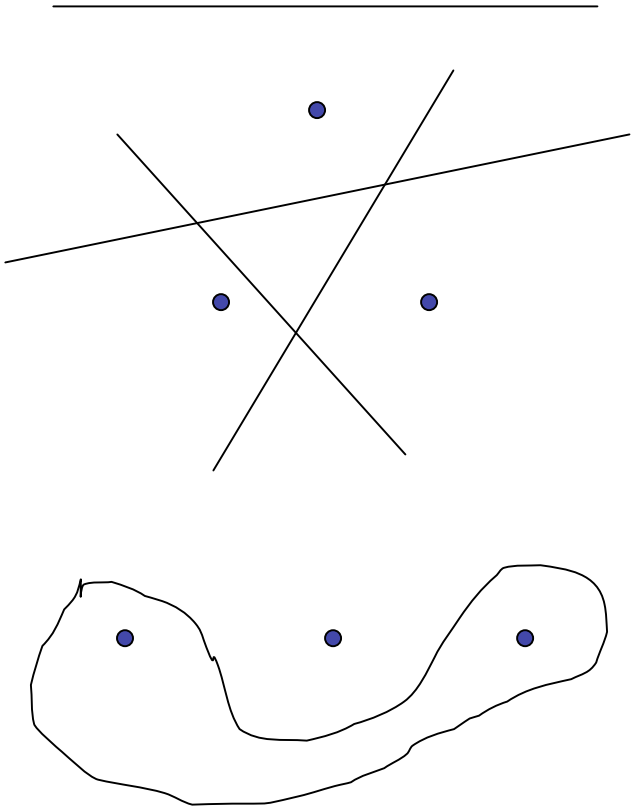
1. Ordne die Hypothesen in Teilmenge gemäß ihrer Komplexität
2. Wähle in jeder Teilmenge die Hypothese mit dem geringsten empirischen Fehler
3. Wähle insgesamt die Hypothese mit minimaler Risikoschranke

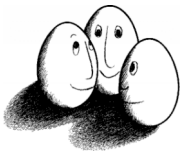




Vapnik-Chervonenkis-Dimension

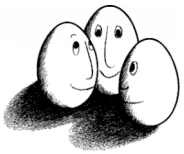
- Definition: Eine Menge H von Hypothesen *zerschmettert* eine Menge E von Beispielen, wenn jede Teilmenge von E durch ein h∈H abgetrennt werden kann.
- Definition: Die *VC-Dimension* einer Menge von Hypothesen
  H ist die maximale Anzahl von Beispielen E, die von H zerschmettert wird.
- Eine Menge von 3 Punkten kann von geraden Linien zerschmettert werden, keine Menge von 4 Punkten kann von geraden Linien zerschmettert werden.





ACHTUNG

- Für eine Klasse von Lernaufgaben gibt es mindestens eine Menge E, die zerschmettert werden kann – NICHT jede Menge E kann zerschmettert werden!
- Zum Beweis der VC Dimension n muss man also zeigen:
  - Es gibt eine Menge E aus n Punkten, die von H zerschmettert werden kann. VCdim(H)≥n
  - Es kann keine Menge E' aus n+1 Punkten geben, die von H zerschmettert werden könnte. VCdim(H)≤n

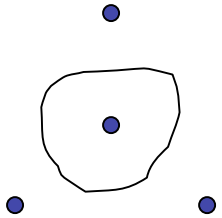


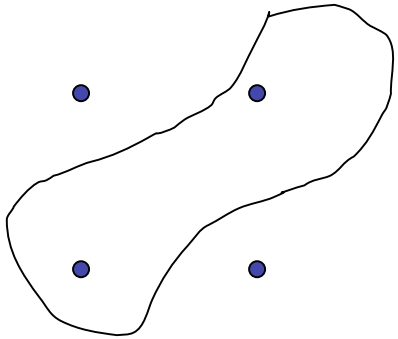
VC-Dimension von Hyperebenen

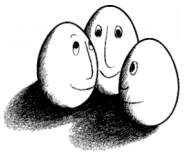
Satz: Die VC-Dimension der Hyperebenen im R^n ist n+1.

Beweis:

- VCdim(R^n) ≥ n+1: Wähle $x_0 = 0$ und x_i = (0,...,0,1,0,...0). Für eine beliebige Teilmenge A von $(x_0,...,x_n)$ setze $y_i = 1$, falls $x_i \in A$ und $y_i = -1$ sonst.
  Definiere $w = \sum y_k x_k$ und $b = y_0/2$. Dann gilt $wx_0+b = y_0/2$ und $wx_i+b = y_i+y_0/2$. Also: wx+b trennt A.
- VCdim(R^n) ≤ n+1: Zurückführen auf die beiden Fälle rechts.

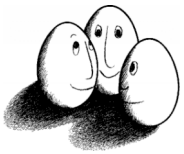






VCdim misst Kapazität

- Eine Funktion mit nur 1 Parameter kann unendliche VCdim haben: H kann Mengen von n Punkten zerschmettern, egal wie groß n ist.
- H kann unendliche VCdim haben und trotzdem kann ich eine kleine Zahl von Punkten finden, die H nicht zerschmettern kann.
- VCdim ist also nicht groß, wenn die Anzahl der Parameter bei der Klasse von Funktionen H groß ist.



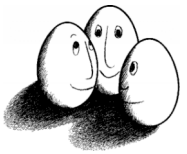
VC-Dim. und Anzahl der Parameter

- Setze $f_\alpha(x) = \cos(\alpha x)$ und $x_i = 10^{-i}$, $i=1\ldots\ell$, beliebiges ℓ. Wähle $y_i \in \{-1,1\}$. Dann gilt für $\alpha = \pi(\sum {}^1/_2(1-y_i)10^i)$:

$$\alpha x_k = \pi\left(\sum_{i=1}^{l} \tfrac{1}{2}(1-y_i)10^i\right)10^{-k} = \pi\left(\sum_{i=1}^{l} \tfrac{1}{2}(1-y_i)10^{i-k}\right)$$

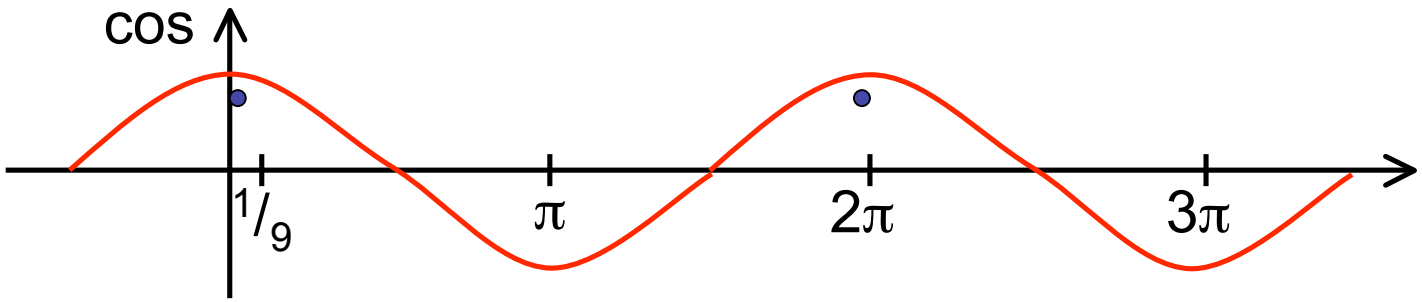
$$= \pi\left(\underbrace{\sum_{i=1}^{k-1} \tfrac{1}{2}(1-y_i)10^{i-k}}_{0 \le \sum\ldots \le 10^{-1}+10^{-2}+\ldots = 1/9} + \tfrac{1}{2}(1-y_k) + \underbrace{\sum_{i=k+1}^{l} \tfrac{1}{2}(1-y_i)10^{i-k}}_{\text{Vielfaches von 2}}\right)$$

(geometrische Reihe)



VC-Dim. und Anzahl der Parameter

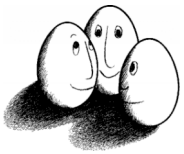
⇒ $\cos(\alpha x_k)=\cos(\pi z)$ mit $z\in[0,{}^1/_9]$ für $y_k=1$ und $z\in[1,{}^{10}/_9]$ für $y_k=-1$



⇒ $\cos(\alpha x)$ zerschmettert $x_1,...x_\ell$

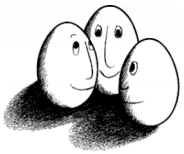
⇒ $\cos(\alpha x)$ hat unendliche VC-Dimension

⇒ Die VC-Dimension ist unabhängig von der Anzahl der Parameter!



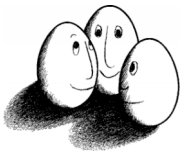
VC-Dimension der SVM

- Gegeben seien Beispiele $x_1,...,x_l \in \Re^n$ mit $||x_i|| < D$ für alle i. Für die VC-Dimension der durch den Vektor w gegebenen optimalen Hyperebene h gilt:
  $$VCdim(h) \leq \min\{D^2 ||w||^2, n\}+1$$
- Die Komplexität einer SVM ist nicht nur durch die Struktur der Daten beschränkt (Fluch der hohen Dimension), sondern auch durch die Struktur der Lösung!



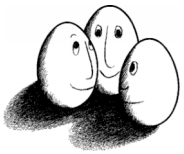
Zusicherungen

- Strukturelle Risikominimierung garantiert, dass die einfachste Hypothese gewählt wird, die noch an die Daten anpassbar ist.
- Strukturelle Risikominimierung kontrolliert die Kapazität des Lernens (weder fauler noch fotografischer Botaniker).
- Die Strukturen von Klassen von Funktionen werden durch die VCdim ausgedrückt. Große VCdim → große VC-confidence.



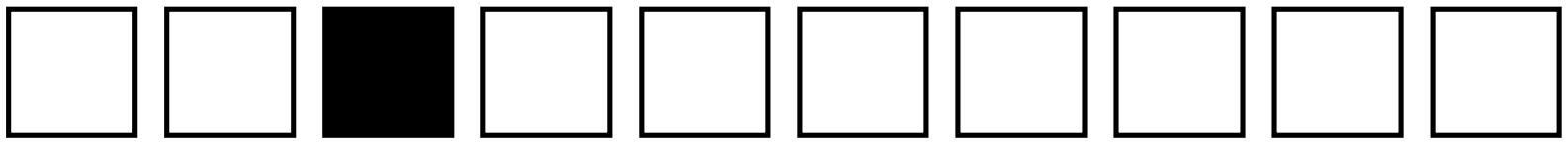
Was wissen wir jetzt?

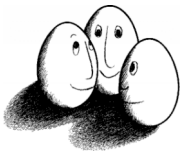
- Kernfunktionen – eine Transformation, die man nicht erst durchführen und dann mit ihr rechnen muss, sondern bei der nur das Skalarprodukt gerechnet wird.
- Idee der strukturellen Risikominimierung:
  - obere Schranke für das Risiko
  - Schrittweise Steigerung der Komplexität
- Formalisierung der Komplexität: VC-Dimension
- SRM als Prinzip der SVM
- Garantie für die Korrektheit der Lernstrategie



Performanzschätzer

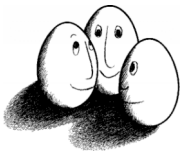
- Welches erwartete Risiko R(α) erreicht SVM?
- R(α) selbst nicht berechenbar
- Trainingsfehler (zu optimistisch – Overfitting)
- Obere Schranke mittels VC-Dimension (zu locker)
- Kreuzvalidierung / Leave-One-Out-Schätzer (ineffizient)





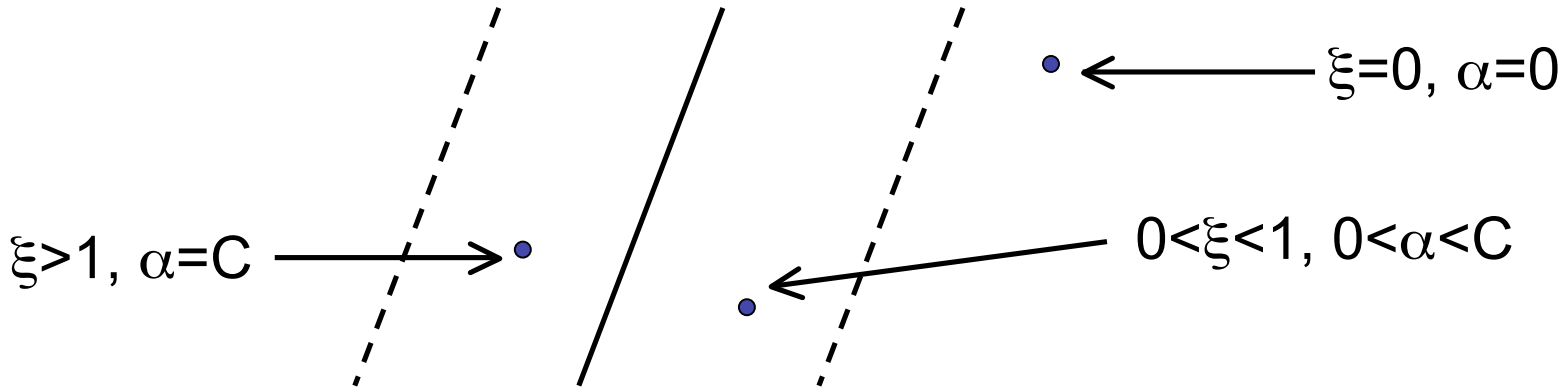
Performanzschätzer II

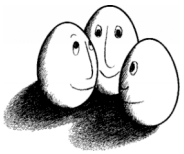
- Satz: Der Leave-One-Out-Fehler einer SVM ist beschränkt durch $R_{l1o} \leq |SV| / n$
- Beweis: Falsch klassifizierte Beispiele werden Stützvektoren. Also: Nicht-Stützvektoren werden korrekt klassifiziert. Weglassen eines Nicht-Stützvektors ändert die Hyperebene nicht, daher wird es auch beim l1o-Test richtig klassifiziert.



Performanzschätzer III

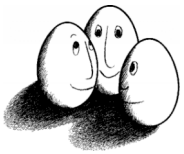
- Satz: Der Leave-One-Out-Fehler einer SVM ist beschränkt durch
  $R_{l1o} \leq |\{i : (2\alpha_i D^2+\xi_i)\geq 1\}| / n$
  (D = Radius des Umkreises um die Beispiele im transformierten Raum).
- Beweis: Betrachte folgende drei Fälle:





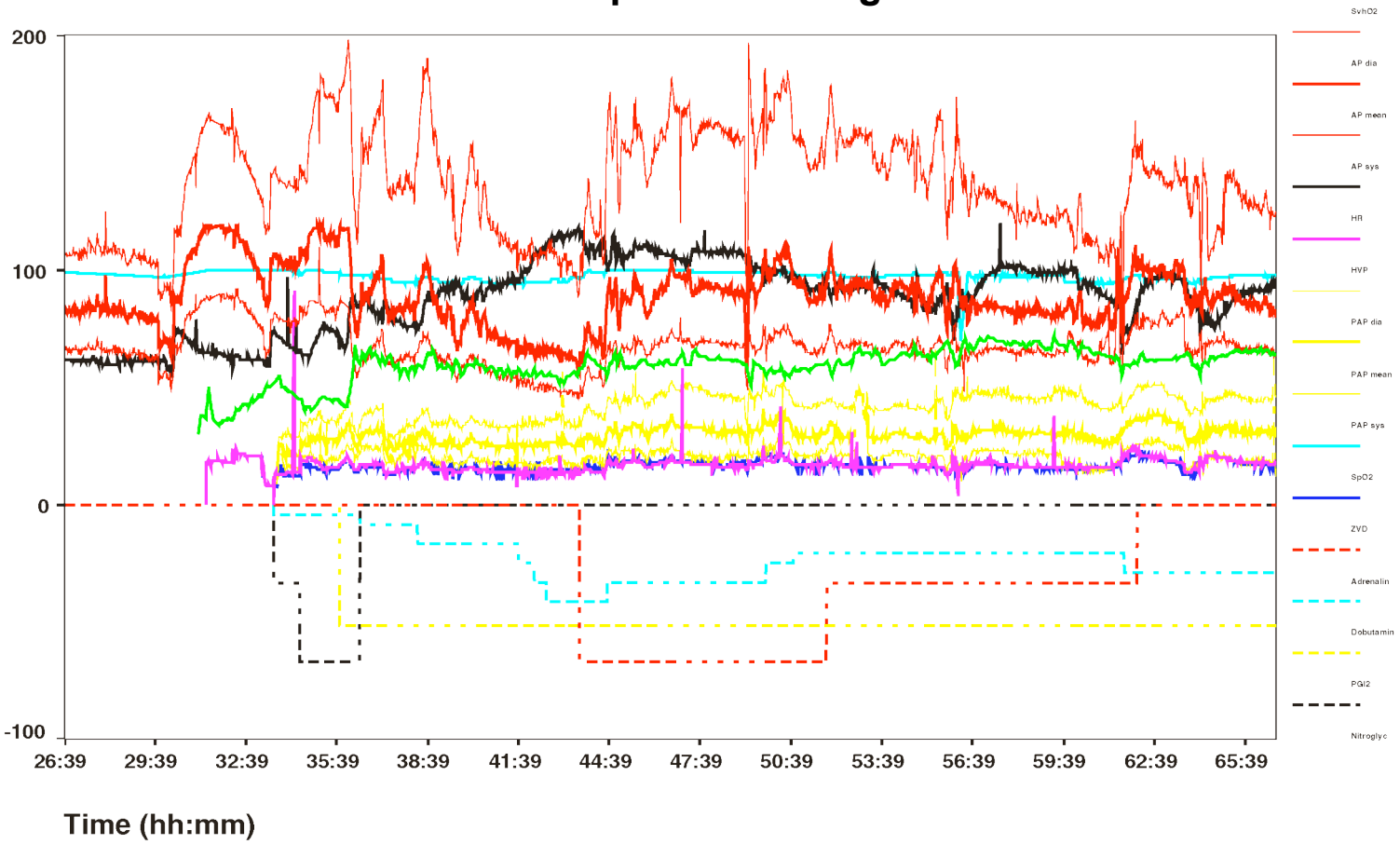
Fallstudie Intensivmedizin

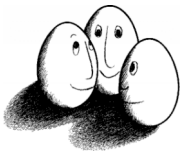
- Städtische Kliniken Dortmund, Intensivmedizin 16 Betten, Priv.-Doz. Dr. Michael Imhoff
- Hämodynamisches Monitoring, minütliche Messungen
  - Diastolischer, systolischer, mittlerer arterieller Druck
  - Diastolischer, systolischer, mittlerer pulmonarer Druck
  - Herzrate
  - Zentralvenöser Druck
- Therapeutie, Medikamente:
  - Dobutamine, adrenaline, glycerol trinitrate, noradrenaline, dopamine, nifedipine



Patient G.C., male, 60 years old

Hemihepatektomie right



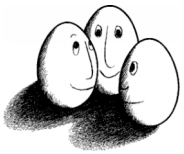


Wann wird Medikament gegeben?

- Mehrklassenproblem in mehrere 2Klassen-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

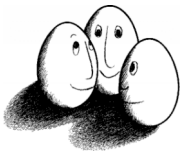
Ergebnis: Gewichte der Vitalwerte so dass positive und negative Beispiele maximal getrennt werden (SVM).



Beispiel: Intensivmedizin

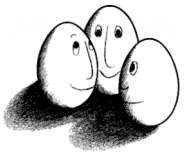
$$f(x) = \left[\begin{pmatrix} 0.014 \\ 0.019 \\ -0.001 \\ -0.015 \\ -0.016 \\ 0.026 \\ 0.134 \\ -0.177 \\ \vdots \end{pmatrix} \begin{pmatrix} artsys = 174.00 \\ artdia = 86.00 \\ artmn = 121.00 \\ cvp = 8.00 \\ hr = 79.00 \\ papsys = 26.00 \\ papdia = 13.00 \\ papmn = 15.00 \\ \vdots \end{pmatrix}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

- Vitalzeichen von Intensivpatienten
- Hohe Genauigkeit
- Verständlichkeit?



Wie wird Medikament dosiert ?

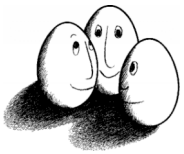
- Mehrklassenproblem in mehrere 2Klassenprobleme umwandeln: für jedes Medikament und jede Richtung (increase, decrease, equal), 2 Mengen von Patienten-daten:
  - Positive Beispiele: alle Minuten, in denen die Dosierung in der betreffenden Richtung geändert wurde
  - Negative Beispiele: alle Minuten, in denen die Dosierung nicht in der betreffenden Richtung geändert wurde.



Steigern von Dobutamine

ARTEREN: -0.05108108119
SUPRA: 0.0089280753865797 3
DOBUTREX: -0.100650806786886
WEIGHT: -0.039353180104626 5
AGE: -0.00378828681071417
ARTSYS: -0.323407537252192
ARTDIA: -0.039456533301949 3
ARTMN: -0.180425080906375
HR: -0.10010405264306
PAPSYS: -0.025264118853173 1
PAPDIA: 0.045484333711276 5
PAPMN: 0.0042950496373652 2
PULS: -0.031350123639988 1

Vektor w für k Attribute



Anwendung des Gelernten

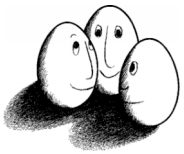
- Patientwerte
  pat46, artmn 95, min. 2231
  ...
  pat46, artmn 90, min. 2619

- Gelernte Gewichte für Dobutamin
  artmn -0,18
  ...

$$svm\_calc = \sum_{i=1}^{k} w_i x_i \qquad decision = sign(svm\_calc + b)$$

svm_calc (pat46, dobutrex, up,min.2231,39) svm_calc (pat46, dobutrex, up,min.2619, 25)

b=-26, i.e. increase in minute 2231,
not increase in minute 2619.

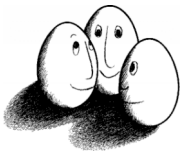


Steigern von Glyceroltrinitrat

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

Jedes Medikament hat einen Dosierungsschritt.
Für Glyceroltrinitrat ist es 1, für Suprarenin (adrenalin) 0.01.
Die Dosis wird um einen Schritt erhöht oder gesenkt.

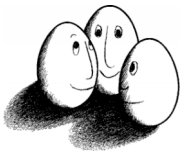
Vorhersage:
pred_interv(pat49, min.32,nitro, 1.0)



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

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)**

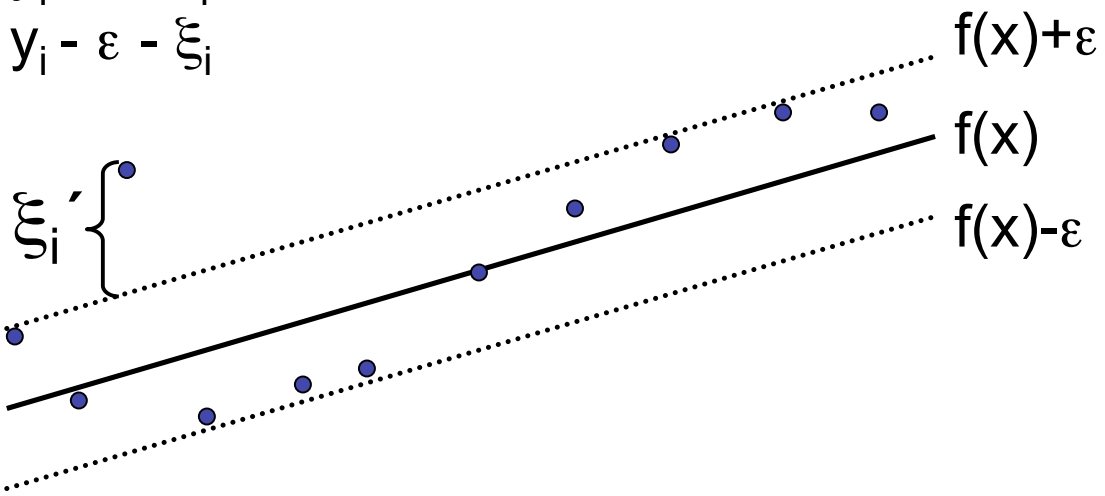


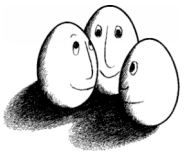
SVMs für Regression

- Minimiere $\|w\|^2 + C\left(\sum_{i=1}^{n} \xi_i + \sum_{i=1}^{n} \xi_i'\right)$
- so dass für alle i gilt:

  $f(x_i) = w^*x_i+b \leq y_i +\varepsilon +\xi_i'$ und

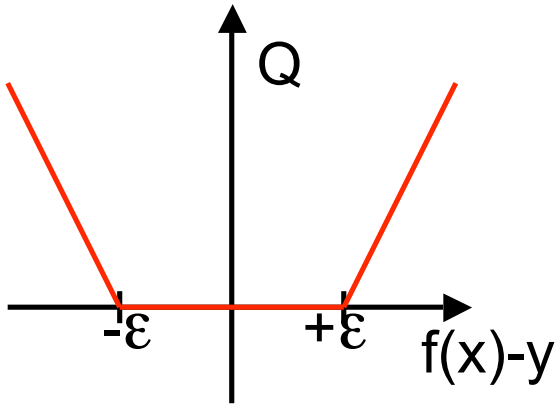
  $f(x_i) = w^*x_i+b \geq y_i - \varepsilon - \xi_i$



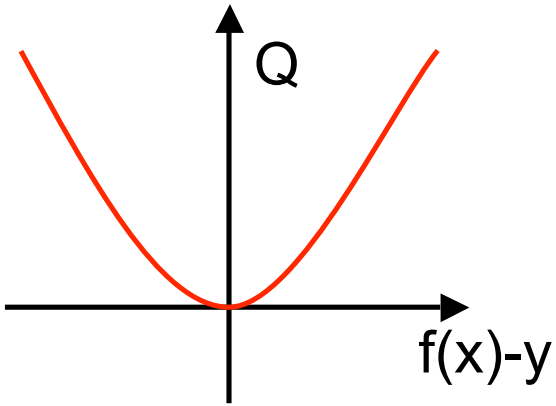


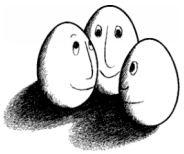
Verlustfunktion

lineare Verlustfunktion



quadratische Verlustfunktion



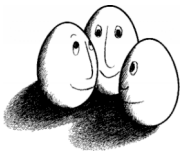


Duales Optimierungsproblem

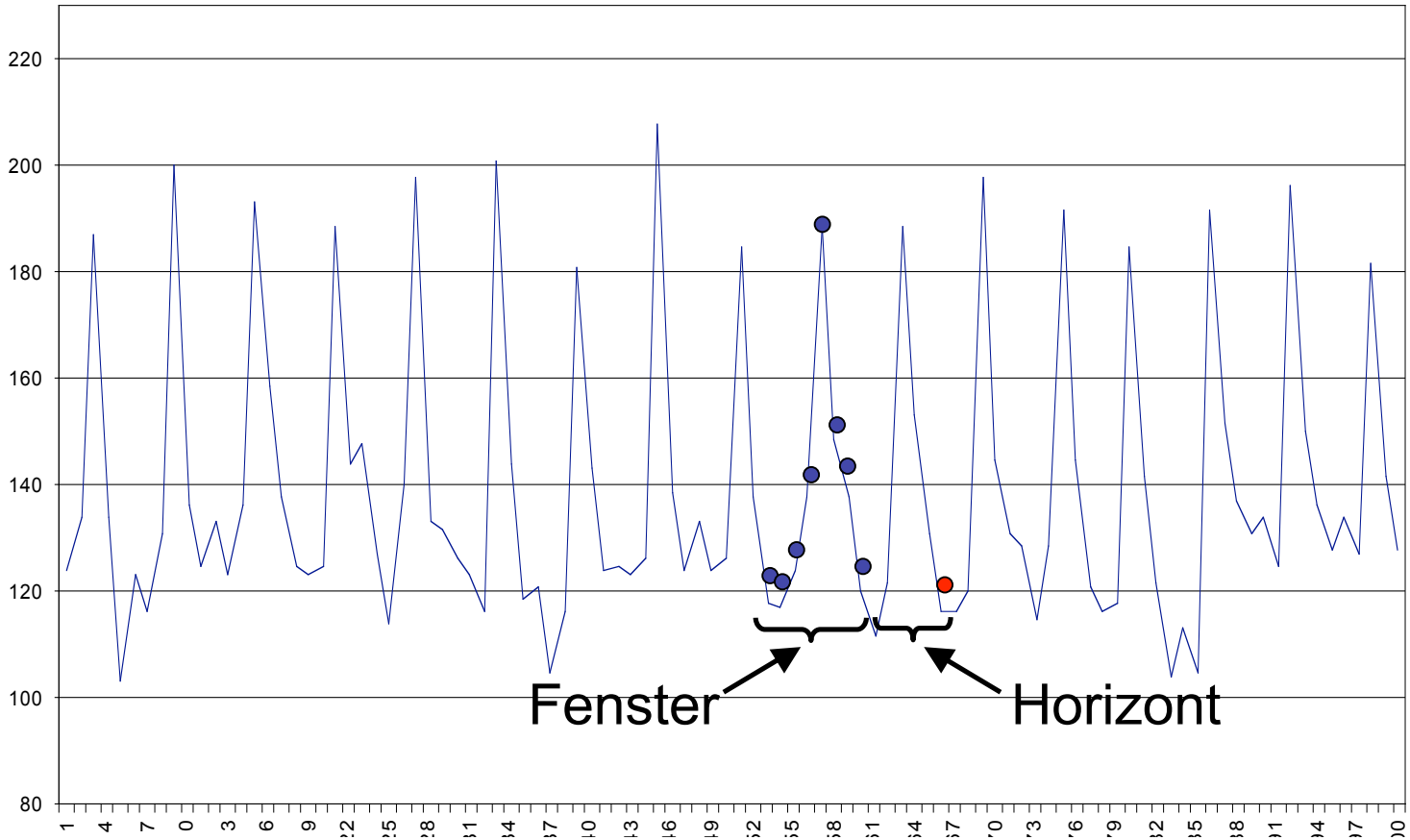
- Maximiere

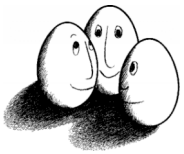
$$W(\alpha) = \sum_{i=1}^{n} y_i(\alpha_i' - \alpha_i) - \varepsilon \sum_{i=1}^{n} (\alpha_i' + \alpha_i) - \frac{1}{2} \sum_{i,j=1}^{n} (\alpha_i' - \alpha_i)(\alpha_j' - \alpha_j) K(x_i, x_j)$$

- unter $0 \leq \alpha_i, \alpha_i' \leq C$ für alle i und
  $\sum \alpha_i' = \sum \alpha_i$
- Mit $y_i \in \{-1,+1\}$, $\varepsilon=0$ und $\alpha_i=0$ für $y_i=1$ und $\alpha_i'=0$ für $y_i=-1$ erhält man die Klassifikations-SVM!



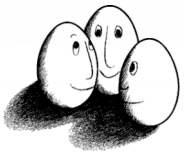
Beispiel: Prognose von Zeitreihen



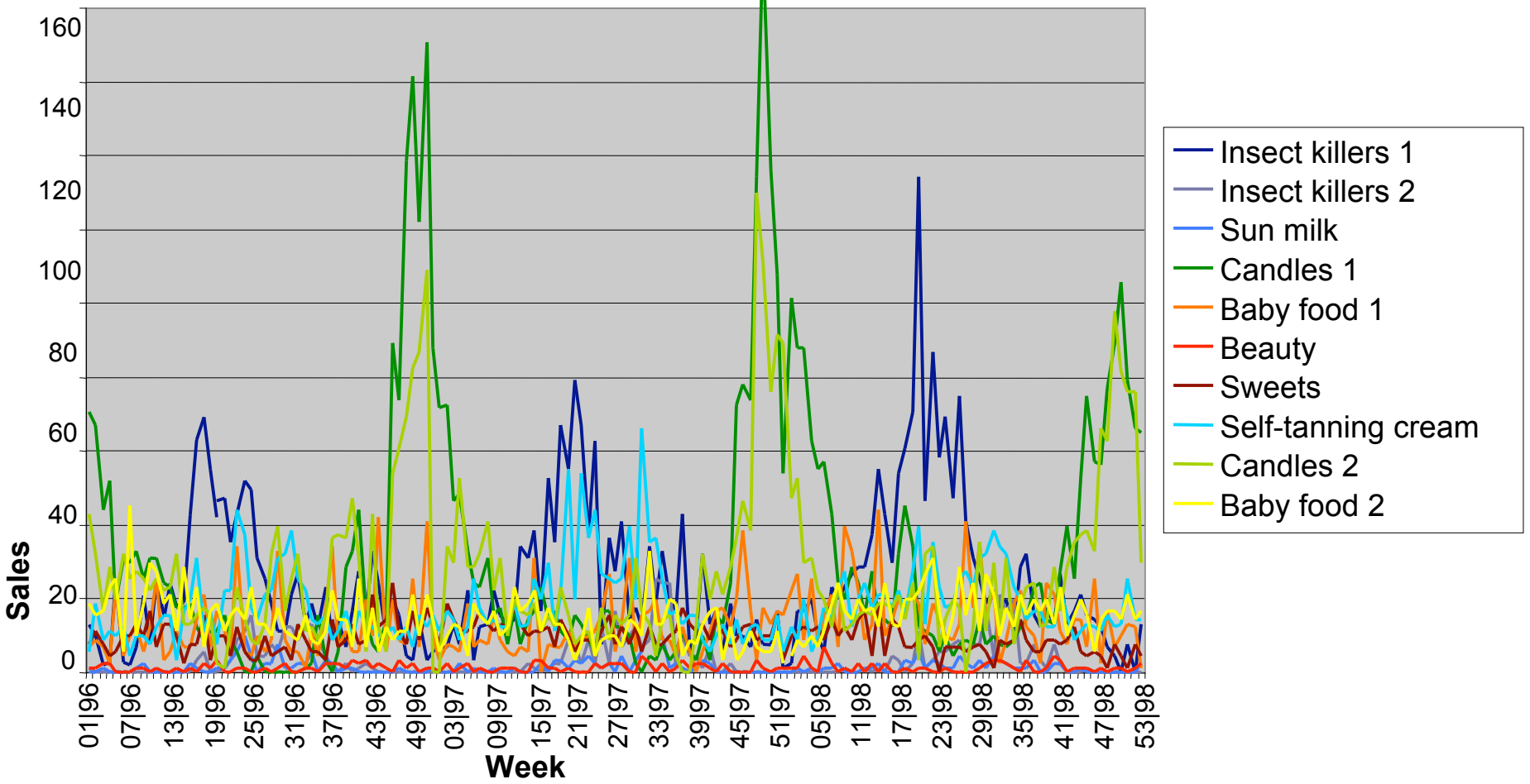


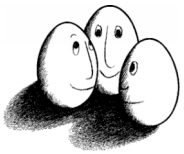
Prognose von Zeitreihen

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



Abverkauf Drogerieartikel





Vorhersage Abverkauf

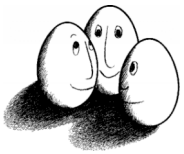
Gegeben Verkaufsdaten von 50 Artikeln in 20 Läden über 104 Wochen

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.

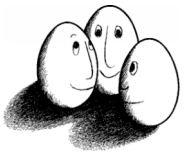
Anforderung: Vorhersagehorizont von mehr als 4 Wochen.



Verkaufsdaten

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

LE_{DB1}: I: T_1 A_1 ... A_{50}; Menge multivariater Zeitreihen



Vorverarbeitung

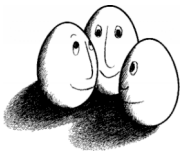
- Multivariat nach univariat

$L_{E1'}$: i:t_1 a_1 ... t_k a_k

For all shops for all items:
Create view Univariate as
Select shop, week, item$_i$
Where shop="dm$_j$"
From Source;

- Multiples Lernen

Dm1_Item1	1	4 ...	104	9
...				
Dm1_Item50	1	12...	104	16
....				
Dm20_Item50	1	14...	104	16



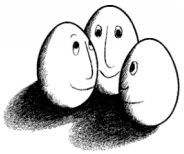
Vorverarbeitung II

- Viele Vektoren aus einer Reihe gewinnen durch Fenster

L_{H5} i:t_1 a_1 ... t_w a_w

bewege Fenster der Größe w um m Zeitpunkte

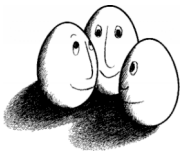
Dm1_Item1_1	1	4...	5	7
Dm1_Item1_2	2	4...	6	8
...				
Dm1_Item1_100	100	6...	104	9
...				
...				
Dm20_Item50_100	100	12...	104	16



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

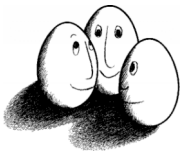
(Stefan Rüping 1999)



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



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