

Skript zur Vorlesung
Knowledge Discovery in Databases
im Wintersemester 2010/2011

Kapitel 4: Regression

Vorlesung+Übungen:
PD Dr. Peer Kröger, Dr. Arthur Zimek

Skript © 2010 Johannes Aßfalg, Christian Böhm, Karsten Borgwardt, Martin Ester, Eshref Januzaj, Karin Kailing, Peer Kröger, Jörg Sander, Matthias Schubert, Arthur Zimek

[http://www.dbs.ifi.lmu.de/cms/Knowledge_Discovery_in_Databases_I_\(KDD_I\)](http://www.dbs.ifi.lmu.de/cms/Knowledge_Discovery_in_Databases_I_(KDD_I))

Klassifikation: Jedes Objekt o hat eine Klasse $C_i \in \{C_1, \dots, C_k\}$

Klassifikator: $O \rightarrow C$

C ist diskret !!

Regression: Jedem Objekt o ist ein Zielvariable $Y \in \mathcal{R}$ zugeordnet.

Regression: $O \rightarrow \mathcal{R}$

Aufgabe der Regression ist die Vorhersage eines kontinuierlichen Wertes.

Zum Beispiel :

Vorhersage des erwarteten Absatzes eines Produkts

oder

empfohlene Menge an Düngemittel für einen bestimmten Bodentyp .

lineare Regression

gesuchte Vorhersage Variable Y verhält sich linear.

multiple Regression

lineare Regression, bei der Y von einem Vektor abhängt.

nicht-lineare Regression

allgemeiner Fall, die beschriebene Regressionsfunktion muss nicht-linear sein.

z.B. Logistic Regression, Poisson Regression

Gegeben: Objekt ist durch Zufallsvariable X beschrieben. Trainingsobjekte haben zusätzlich noch Ausprägungen der Zielvariable Y .

Annahme: $Y = \alpha + \beta \cdot X$

Die beobachteten Werte von Y weichen mit konstanter Varianz von der Linie ab.

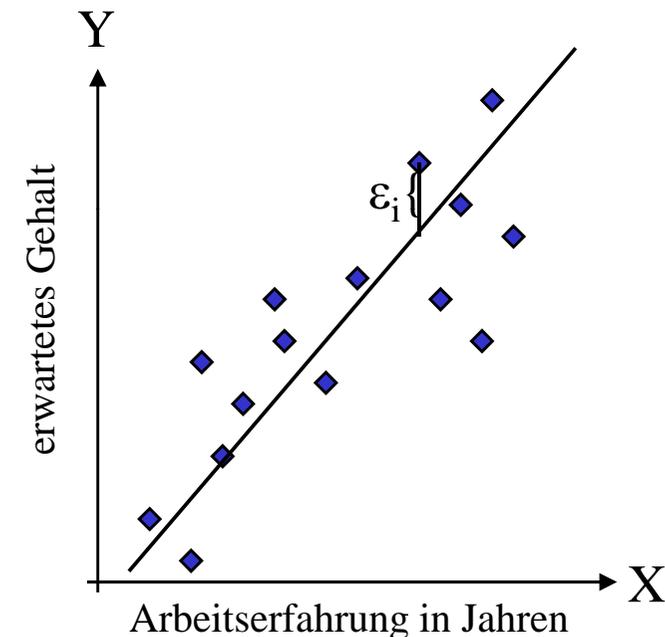
Gesucht: Linie auf der Erwartungswerte liegen.

Lösung: Minimiere quadratischen Fehler.
(Least Squares Methode)

D.h. bestimme α und β , so daß

$$L = \sum_{i=1}^s \varepsilon^2 = \sum_{i=1}^s (Y_i - \alpha - \beta \cdot X_i)^2 \text{ minimal wird.}$$

ε_i ist der Abstand von der angenommenen Regressionsgerade.



Lösungsansatz:

Die partiellen Ableitungen nach β und α ergeben folgende Abschätzungen.

Steigung:
$$\beta = \frac{\sum_{i=1}^s (x_i - E(x))(y_i - E(y))}{\sum_{i=1}^s (x_i - E(x))^2}$$

Y-Achsenabschnitt: $\alpha = E(Y) - \beta \cdot E(X)$

Multiple Regression:

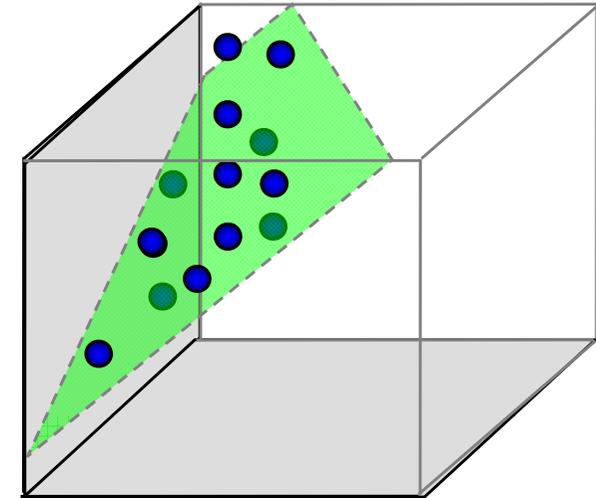
Objekt wird durch d -dimensionalen Featurevektor beschrieben.

Annahme: Y sei von einer Linear-Kombination abhängig.

$$Y = \alpha + \sum_{i=1}^d \beta_i \cdot X_i$$

Lösung: minimiere quadratischen Fehler
(Least Squares Methode)

$$L = \sum_{i=1}^s \varepsilon^2 = \sum_{i=1}^s \left(Y_i - \alpha - \sum_{k=1}^d (\beta_k \cdot X_{i,k}) \right)^2$$



Achtung: Es ergibt sich eine Regressionshyperebene !!

Gegeben: Zufallsvariable X und Zielvariable Y .

Annahme: Y hängt nicht-linear von X ab.

Beispiel: polynomielle Regression

Annahme:

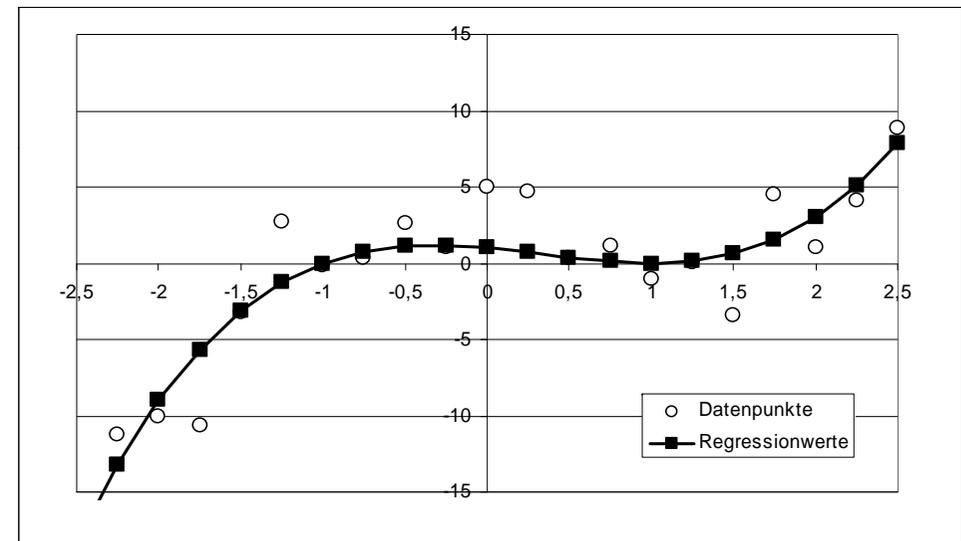
$$Y = \alpha + \beta_1 \cdot X + \beta_2 X^2 + \beta_3 X^3$$

Lösung: Definiere neue Variablen.

$$X_1 = X, \quad X_2 = X^2, \quad X_3 = X^3$$

Löse lineare multiple Regression

$$Y = \alpha + \beta_1 \cdot X_1 + \beta_2 X_2 + \beta_3 X_3$$



ε -insensitive Fehlerfunktion

bis jetzt müssen alle Punkte auf Regressionsgerade liegen.

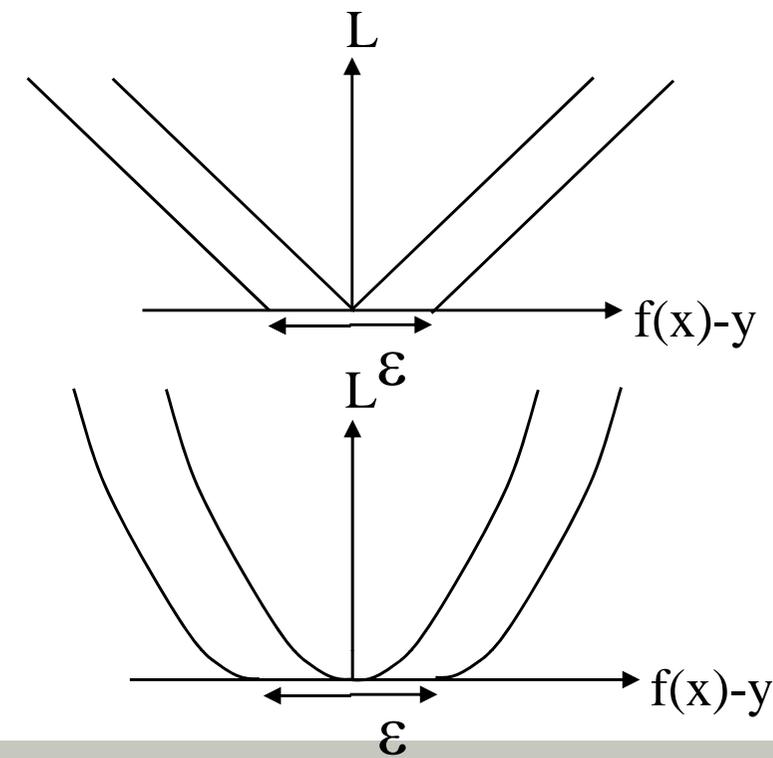
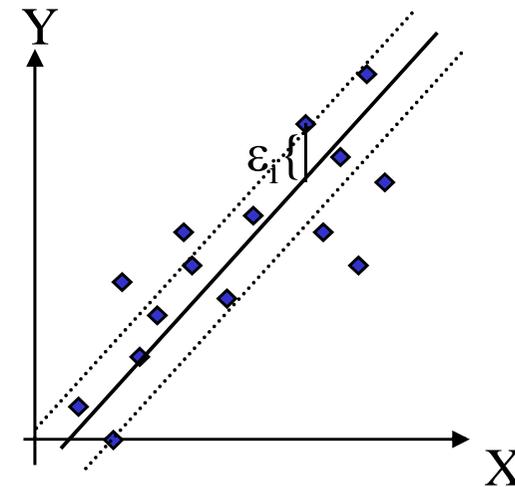
Aber: Häufig ist Abstand ε von der Linie noch akzeptabel

=> definiere Rand mit Größe ε , in dem der Fehler toleriert wird.

$$L^\varepsilon(x, y, f) = |y - f(x)|_\varepsilon = \max(0, |y - f(x)| - \varepsilon)$$

oder als quadratischer Fehler

$$L_2^\varepsilon(x, y, f) = |y - f(x)|_2^\varepsilon$$



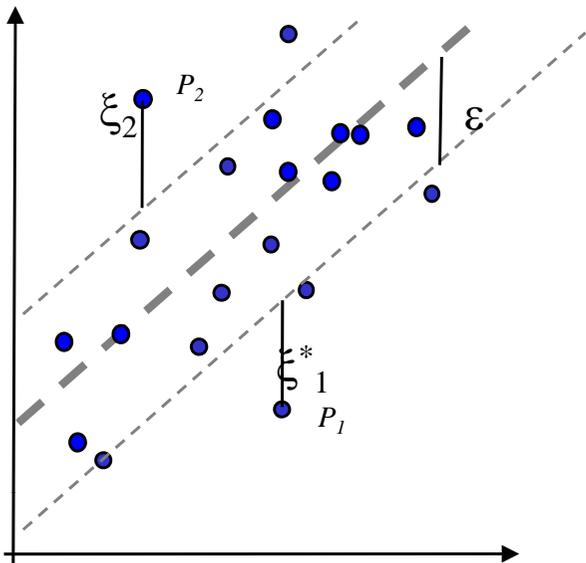
Auf Basis der ε -insensitiven Fehlerfunktion kann man jetzt ein Optimierungsproblem ähnlich zu dem der SVMs definieren

Primäres OP :

$$\text{minimiere } J(\vec{w}, b, \vec{\xi}) = \|\vec{w}\|^2 + C \cdot \sum_{i=1}^n \xi_i^2 + \xi_i^{*2}$$

unter Nebenbedingung für $\forall i \in [1..n]$ sei $y_i - (\langle \vec{w}, \vec{x}_i \rangle + b) \leq \varepsilon + \xi_i$

$$(\langle \vec{w}, \vec{x}_i \rangle + b) - y_i \leq \varepsilon + \xi_i^* \quad \text{und} \quad \xi_i, \xi_i^* \geq 0$$



- 2 Typen von Slack-Variablen für überhalb und unterhalb des Zielwertes y
- Beachte: $\xi_i, \xi_i^* = 0$, da Objekt entweder überhalb oder unterhalb der Regressionsgerade liegt.

Überführt in eine Form mit Lagrange Multiplikatoren:

Duales OP: maximiere

$$L(\vec{\alpha}) = \sum_{i=1}^n y_i (\alpha_i^* - \alpha_i) - \varepsilon \sum_{i=1}^n (\alpha_i^* - \alpha_i) - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n (\alpha_i^* - \alpha_i) \cdot (\alpha_j^* - \alpha_j) \cdot \left(\langle \vec{x}_i, \vec{x}_j \rangle + \frac{1}{C} \delta_{i,j} \right)$$

mit Bedingung $\sum_{i=1}^n (\alpha_i^* - \alpha_i) = 0$ und $0 \leq \alpha_i, 0 \leq \alpha_i^*, i=1 \dots n$

Beachte das hierbei gilt: $\alpha_i \alpha_i^* = 0$ und $\xi_i \xi_i^* = 0$

Verallgemeinertes Problem mit Kernelfunktion:

Duales OP: maximiere $L(\vec{\alpha}) = \sum_{i=1}^n y_i \alpha_i - \varepsilon \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \cdot \alpha_j \cdot K(\vec{x}_i, \vec{x}_j)$

mit Bedingung $\sum_{i=1}^n \alpha_i = 0$ und $-C \leq \alpha_i \leq C, i=1 \dots n$

Anmerkungen:

- Über Kernel lässt sich bequem nicht lineare Regression realisieren
- Training mit den gleichen Lösungsverfahren wie bei SVMs für die Klassifikation
- Es gibt weitere Varianten: z.B. Ridge-Regression.
Hierbei ist $\varepsilon = 0$.
- D.h. es handelt sich um Least Squares Regression mit einer Einschränkung der Gewichte.

Ridge Regression :

$$\text{minimiere } J(\vec{w}, b, \vec{\xi}) = \lambda \|\vec{w}\|^2 + C \cdot \sum_{i=1}^n \xi_i^2$$

unter Nebenbedingung für $\forall i \in [1..n]$ sei $y_i - \left(\langle \vec{w}, \vec{x}_i \rangle + b \right) = \xi_i$, $i = 1, \dots, n$

- Regression löst ein ähnliches Problem wie Klassifikation. Vorhersage: kontinuierlicher Werte.
- Regressionsgeraden können häufig analytisch bestimmt werden.
- Weiterführende Verfahren mit Kernelfunktionen.
- Anmerkung: Logistische Regression Anwendung von Regression auf Klassifikation.
(Klasse A: $Y = 1$, Klasse B: $Y = 0$)