Naive Bayes für Regressionsprobleme

Vorhersage numerischer Werte mit dem Naive Bayes Algorithmus

Nils Knappmeier

Fachgebiet Knowledge Engineering
Fachbereich Informatik
Technische Universität Darmstadt

14.12.2005

Gliederung

Gliederung

Der Naive Bayes Algorithmus
Rahmenbedingungen

- Gegeben: Diskrete Attribute
- Gesucht: Zuordnung zu einer Kategorie
- Trainingsdaten enthalten möglichst zu jedem Zielwert mehrerer Beispiel

Der Naive Bayes Algorithmus

Lösungsmechanismus

- $p(K|A_1, A_2, \ldots, A_n) = \frac{p(A_1,A_2,\ldots,A_n|K)\cdot p(K)}{p(A_1,A_2,\ldots,A_n)}$
- Annahme: A_k sind unabhängig voneinander.
- Daher: Wähle k mit maximalem

$$\hat{P}(K|A_1, A_2, \ldots, A_n) = p(K) \cdot \prod_{i=1}^{n} p(A_i|k)$$

- $p(K)$ und $p(A_i|K)$ kann anhand der Trainingsdaten berechnet werden.

Neue Rahmenbedingungen

- Gegeben: Numerische und nominale Attribute
- Gesucht: Ein numerischer Zielwert
- Numerische Attribute können auch kontinuierlich sein
- Zielwert ist ebenfalls kontinuierlich
- Zielwert kommt möglicherweise nicht in den Trainingsdaten vor.

Allgemeine Vorgehensweise

- Erstellung einer Annäherungsfunktion für $p(Z|A)$
- Berechnung der Funktionwerte für ein diskretisiertes Intervall
- Berechnung des Zielwertes auf Basis der Funktionswerte
- Zielwert ist Durchschnitt oder Median der Annäherungsfunktion

Bayes Formel

- Angepasste Bayes-Formel

$$p(Z|A) = \frac{p(A|Z) \cdot p(Z)}{\int p(A|Z) \cdot p(Z) dZ}$$

- Nach Anwendung der Unabhängigkeitsannahme:

$$p(Z|A) = \frac{p(Z) \cdot \prod_{i=1}^{n} p(A_i|Z)}{\int p(Z) \cdot \prod_{i=1}^{n} p(A_i|Z) dZ}$$

- Zu ermitteln: Annäherungen an $p(A_i|Z)$ und $p(Z)$

Bayes Formel

- Angepasste Bayes-Formel

$$p(Z|A) = \frac{p(A|Z) \cdot p(Z)}{\int p(A|Z) \cdot p(Z) dZ}$$

- Nach Anwendung der Unabhängigkeitsannahme:

$$p(Z|A) = \frac{p(Z) \cdot \prod_{i=1}^{n} p(A_i|Z)}{\int p(Z) \cdot \prod_{i=1}^{n} p(A_i|Z) dZ}$$

- Zu ermitteln: Annäherungen an $p(A_i|Z)$ und $p(Z)$

Im Paper wird die Bayes Formel folgendermaßen dargestellt:

$$p(Z|A) = \frac{p(A|Z) \cdot p(Z)}{\int p(A|Z) \cdot p(Z) dZ}$$

und kann folgendermaßen **hergeleitet** werden
$p(Z|A_1, A_2, \ldots, A_n) = \frac{p(A_1, A_2, \ldots, A_n|Z) \cdot p(Z)}{p(A_1, a_2, \ldots, A_n)}$ mit
$p(A_1, A_2, \ldots, A_n) = \sum_{z \in \textbf{Zielwerte}} p(A_1, A_2, \ldots, A_n|Z = z) \cdot p(Z = z)$
wobei die Summe aufgrund der kontinuierlichen Funktion durch ein Integral ersetzt wurde.
Der Nenner ist in gewisser Weise notwendig: Später wird der Durchschnitt oder der Median berechnet. Wenn wir den Nenner nicht ausrechnen, müssen wir später beim Durchschnitt durch die Gesamtsumme aller $p(Z|A)$ teilen. Was in diesem Fall dem Nenner entspricht. Beim Median müssen wir ausrechnen, wieviel die Hälfte der Summe ist, also auch den Nenner ausrechnen.

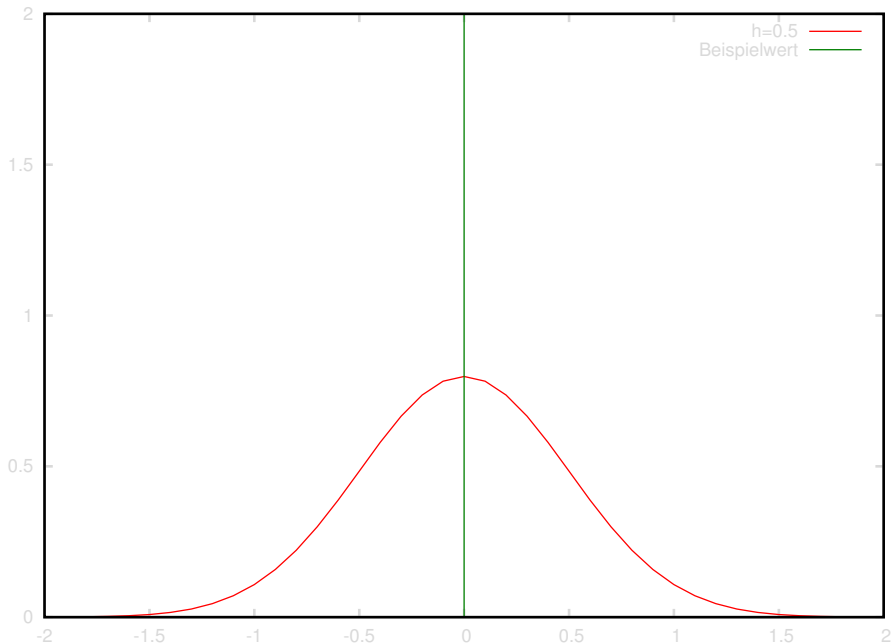
Gliederung

Interpolation durch Gauss-Kernfunktionen

Wie leite ich eine kontinuierliche Wahrscheinlichkeitsfunktion aus Beispielwerten ab?

b sei nunr der Zielwert Wert und b_j der Wert der Beispielwert j.

$$\frac{1}{h}K\left(\frac{b-b_i}{h}\right) \quad \text{mit} \quad K(x) = (2\pi)^{-\frac{1}{2}} \cdot e^{-\frac{x^2}{2}}$$

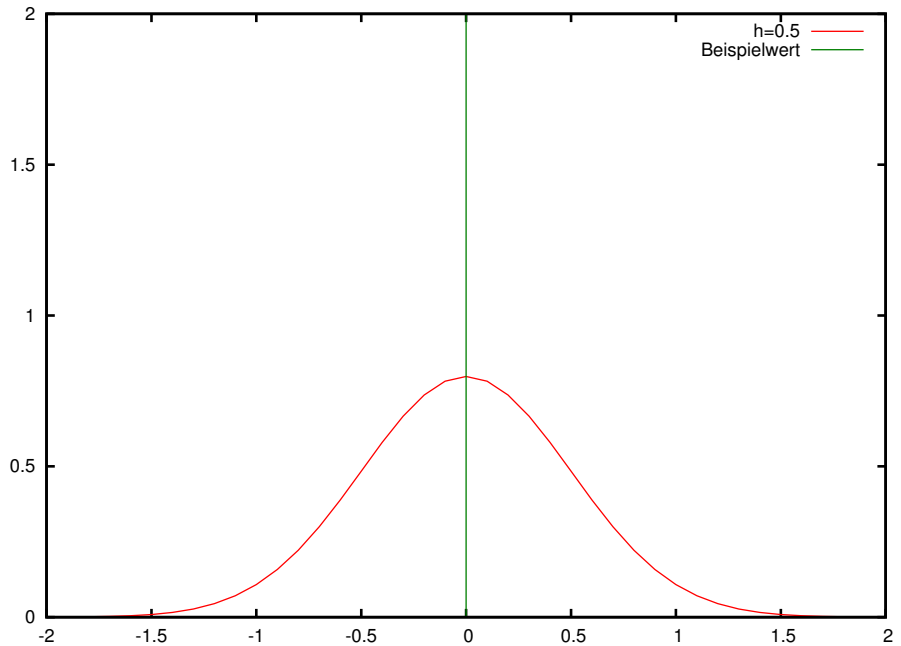


Interpolation durch Gauss-Kernfunktionen

Wie leite ich eine kontinuierliche Wahrscheinlichkeitsfunktion aus Beispielwerten ab?

b sei nunr der Zielwert Wert und b_j der Wert der Beispielwert j.

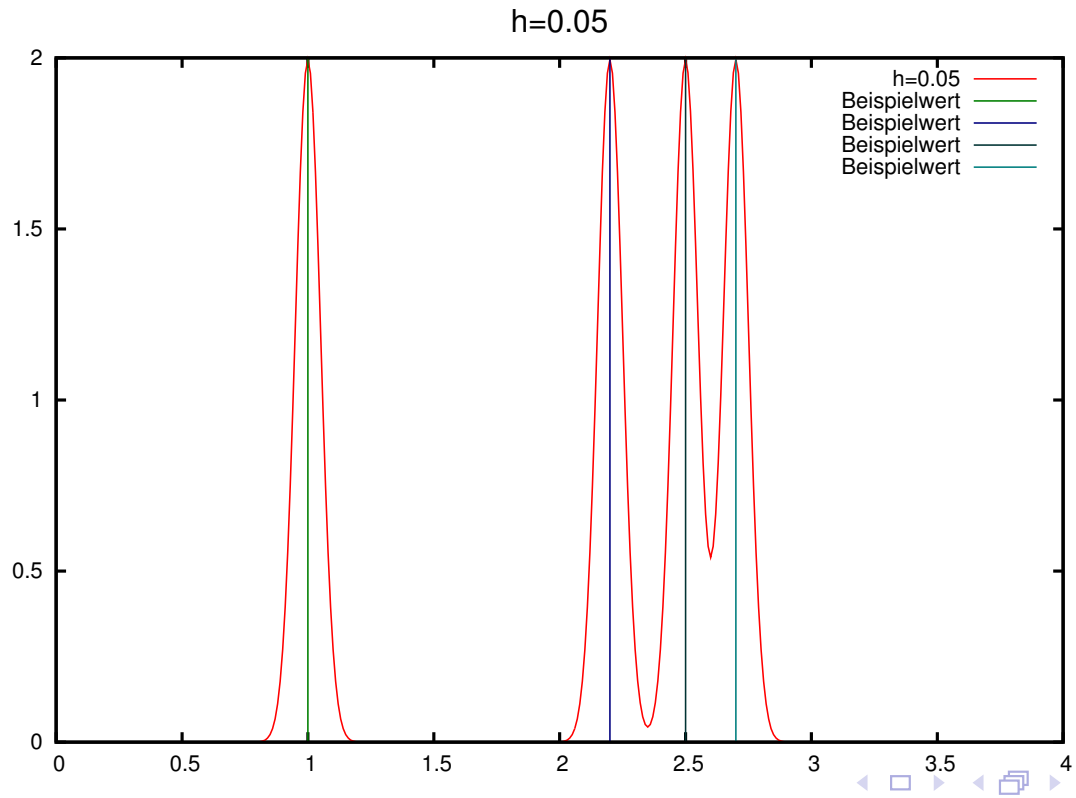
$$\frac{1}{h}K\left(\frac{b-b_i}{h}\right) \quad \text{mit} \quad K(x) = (2\pi)^{-\frac{1}{2}} \cdot e^{-\frac{x^2}{2}}$$



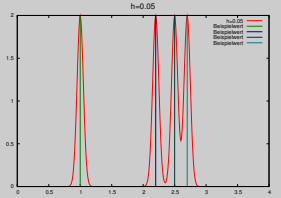
Interpolation durch Gauss-Kernfunktionen

$$\frac{1}{n \cdot h} \sum_{i=1}^{n} K\left(\frac{b - b_i}{h}\right)$$

h ist zu klein



$\frac{1}{n \cdot h} \sum_{i=1}^{n} K\left(\frac{b - b_i}{h}\right)$ h ist zu klein

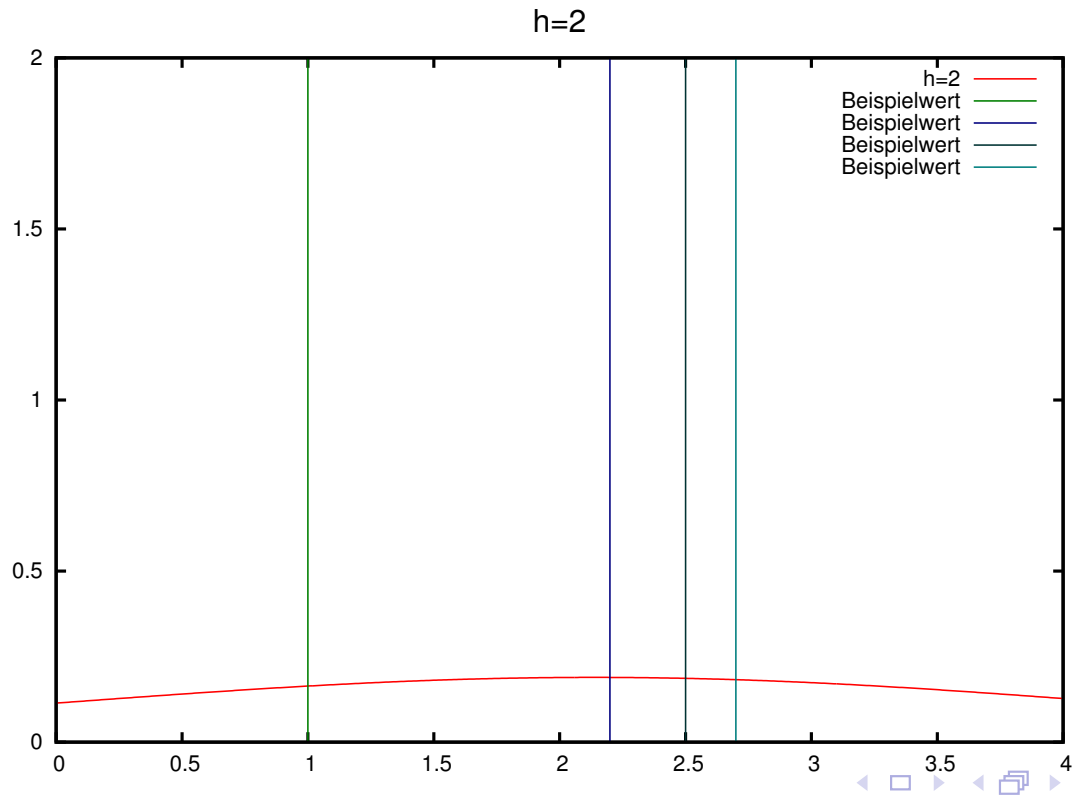


Diese Funktion ist nicht das, was wir wollen, weil hier die Wahrscheinlichkeiten nur auf ganze kleine Bereiche aufgeteilt sind. Wir nehmen aber an, dass die anderen Bereiche auch potentiell legitime Werte sind.

Interpolation durch Gauss-Kernfunktionen

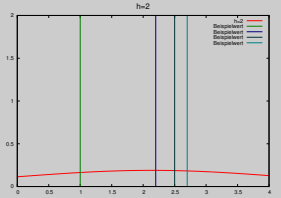
$$\frac{1}{n \cdot h} \sum_{i=1}^{n} K\left(\frac{b - b_i}{h}\right)$$

h ist zu groß



Interpolation durch Gauss-Kernfunktionen

$\frac{1}{n \cdot h} \sum_{i=1}^{n} K\left(\frac{b - b_i}{h}\right)$ h ist zu groß



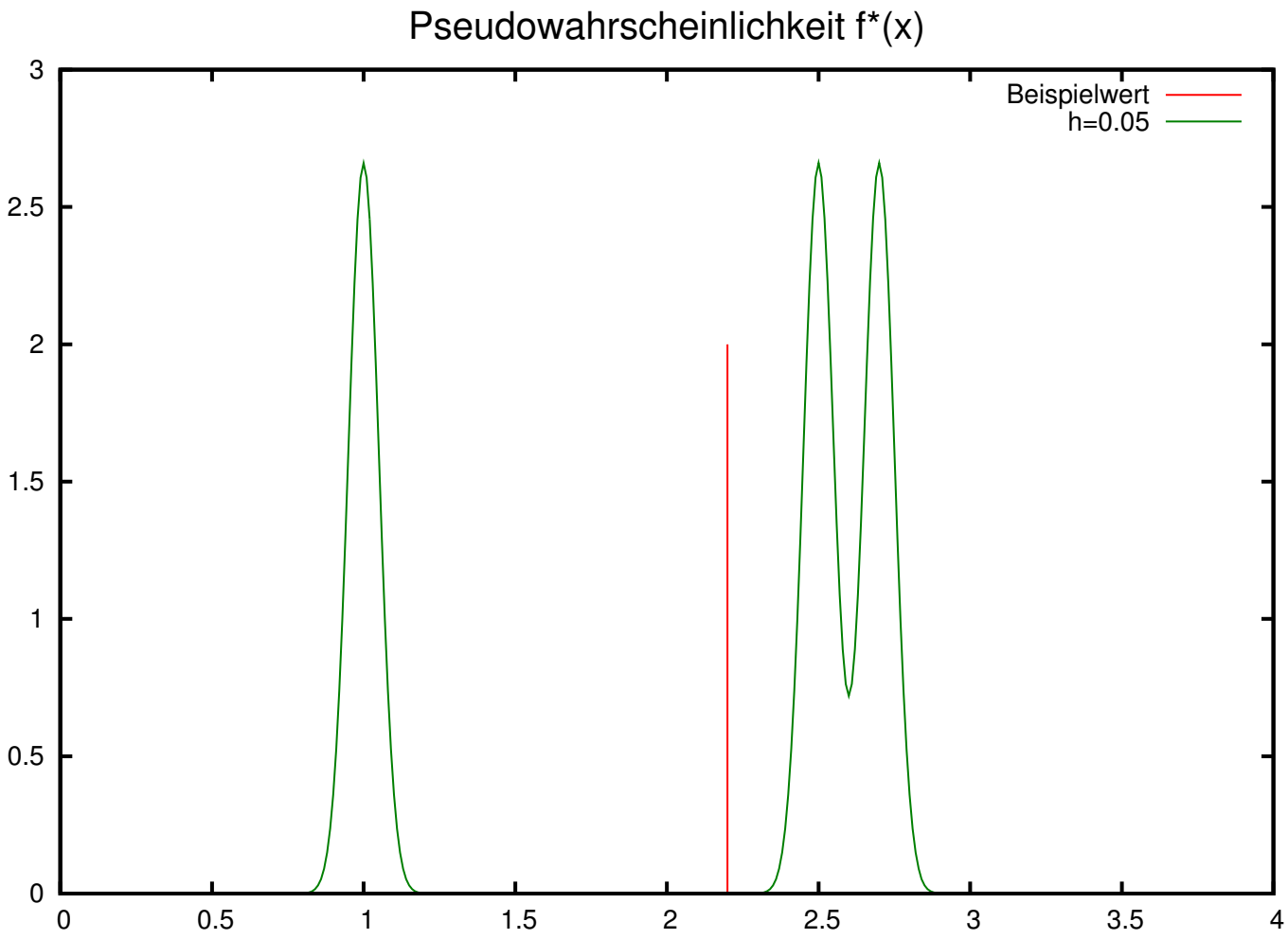
In dieser Funktion spielt das Zentrum der Gauss-Kurven kaum noch eine Rolle, weil die Kurven sich fast komplett überlagern.

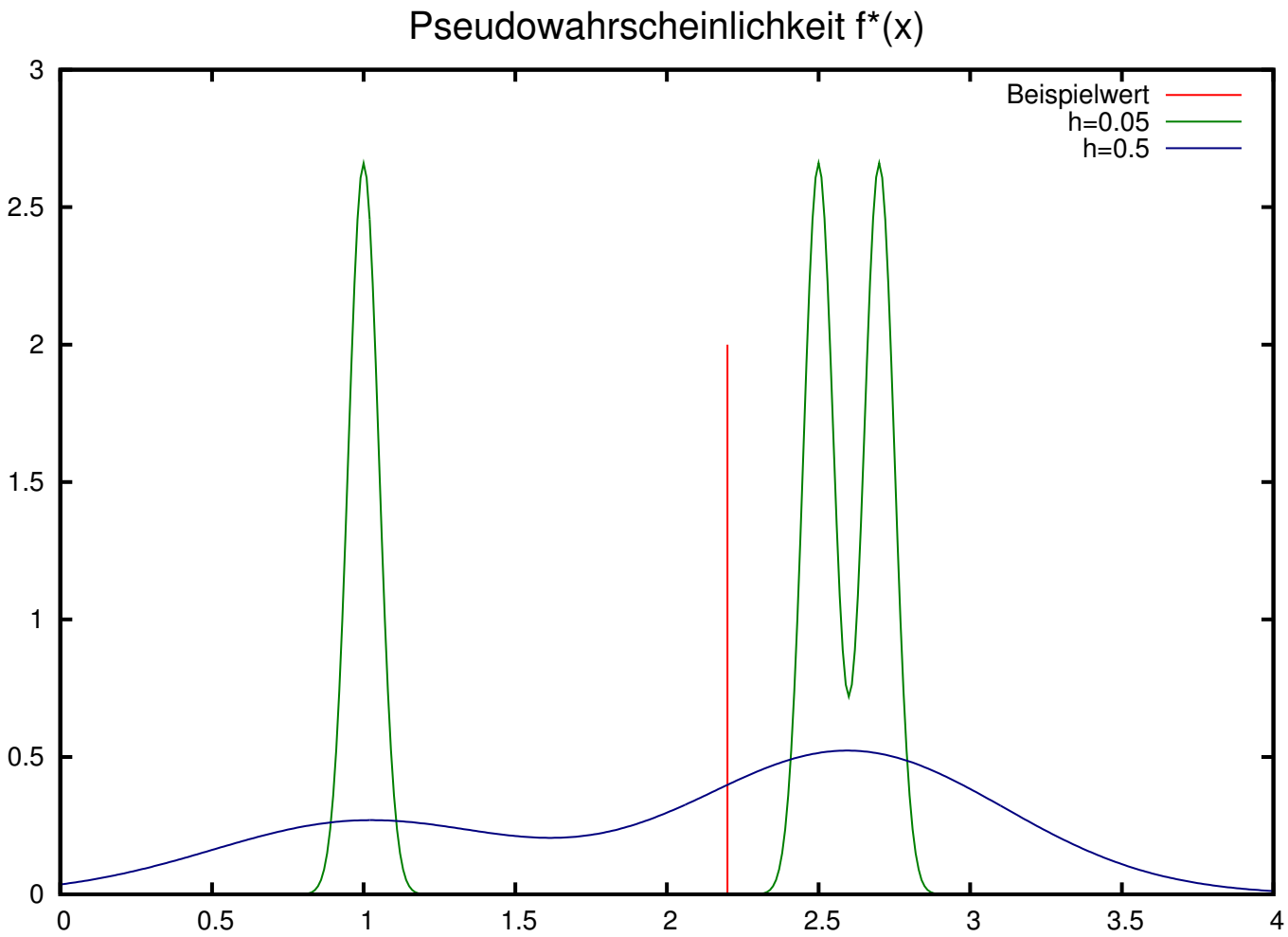
Die richtige Wahl des h

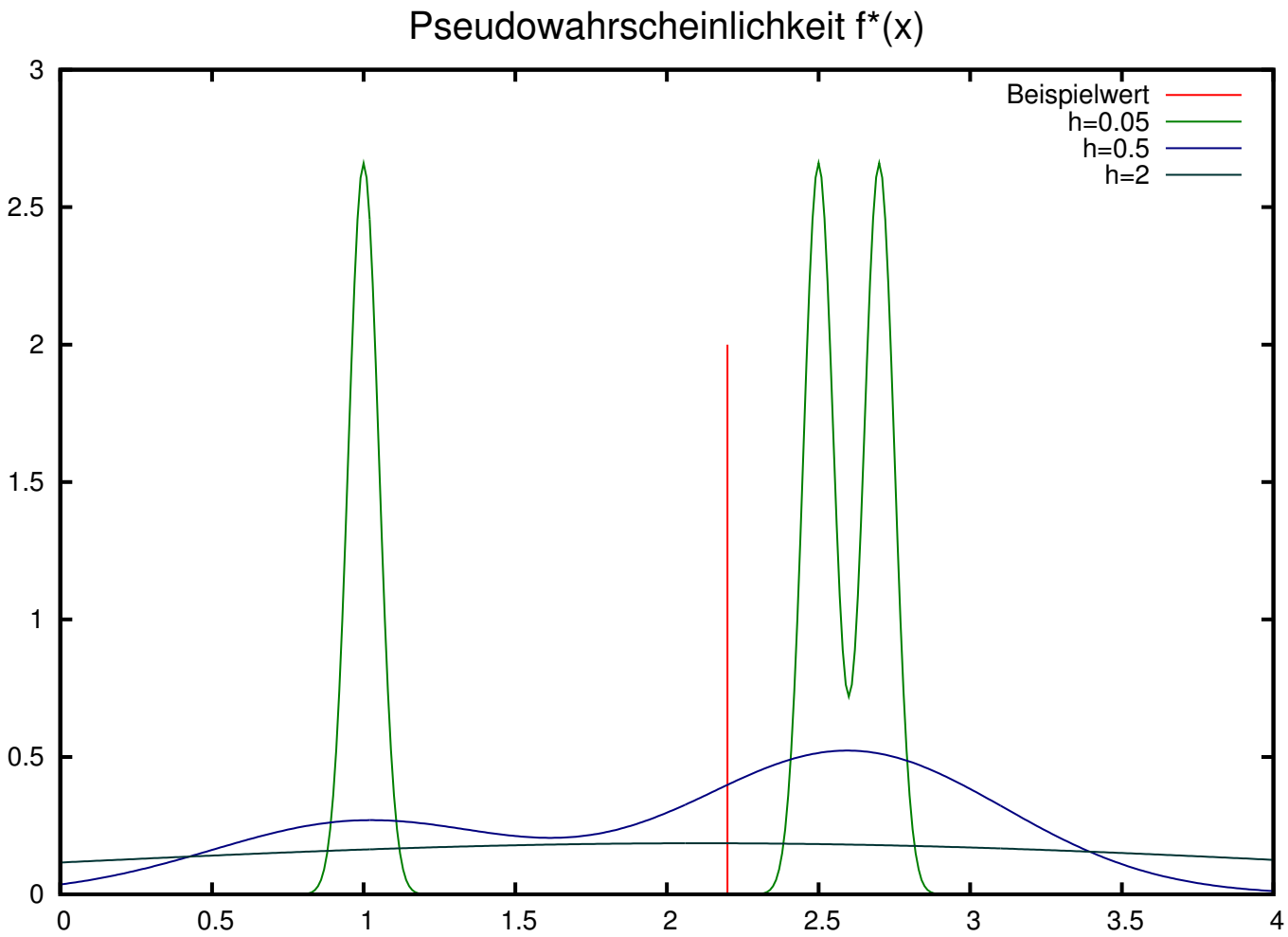
Leave-One-Out-Cross-Validation

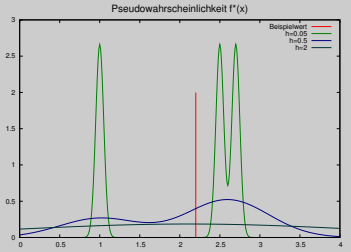
- Intuitiv: Maximale Wahrscheinlichkeit bei den Beispielwerten
- Problem: $h \to 0$
- Lösung: Maximierung einer Pseudo-Wahrscheinlichkeit, bei der alle Kernel ausser dem über b_i berücksichtigt werden.

$$f_i^*(b_i) = \frac{1}{(n-1)h} \sum_{j=1;i\neq j}^{n} K\left(\frac{b_i - b_j}{h}\right)$$


Pseudowahrscheinlichkeit f*(x)
Beispielwert
h=0.05
3
2.5
2
1.5
1
0.5
0
0
0.5
1
1.5
2
2.5
3
3.5
4


Pseudowahrscheinlichkeit f*(x)
Beispielwert
h=0.05
h=0.5
3
2.5
2
1.5
1
0.5
0
0
0.5
1
1.5
2
2.5
3
3.5
4


Pseudowahrscheinlichkeit f*(x)
Beispielwert
h=0.05
h=0.5
h=2
3
2.5
2
1.5
1
0.5
0
0
0.5
1
1.5
2
2.5
3
3.5
4



- Bei kleinen *h* ist die Wahrscheinlichkeit am Beispiel klein.
- Bei sehr großen *h* allerdings auch.
- Der optimale Werte liegt irgendwo dazwischen.
- (Der Durchschnitt der Pseudo-Funktionen müsste gleich dem Durchschnitt der echten Funktionen sein.)

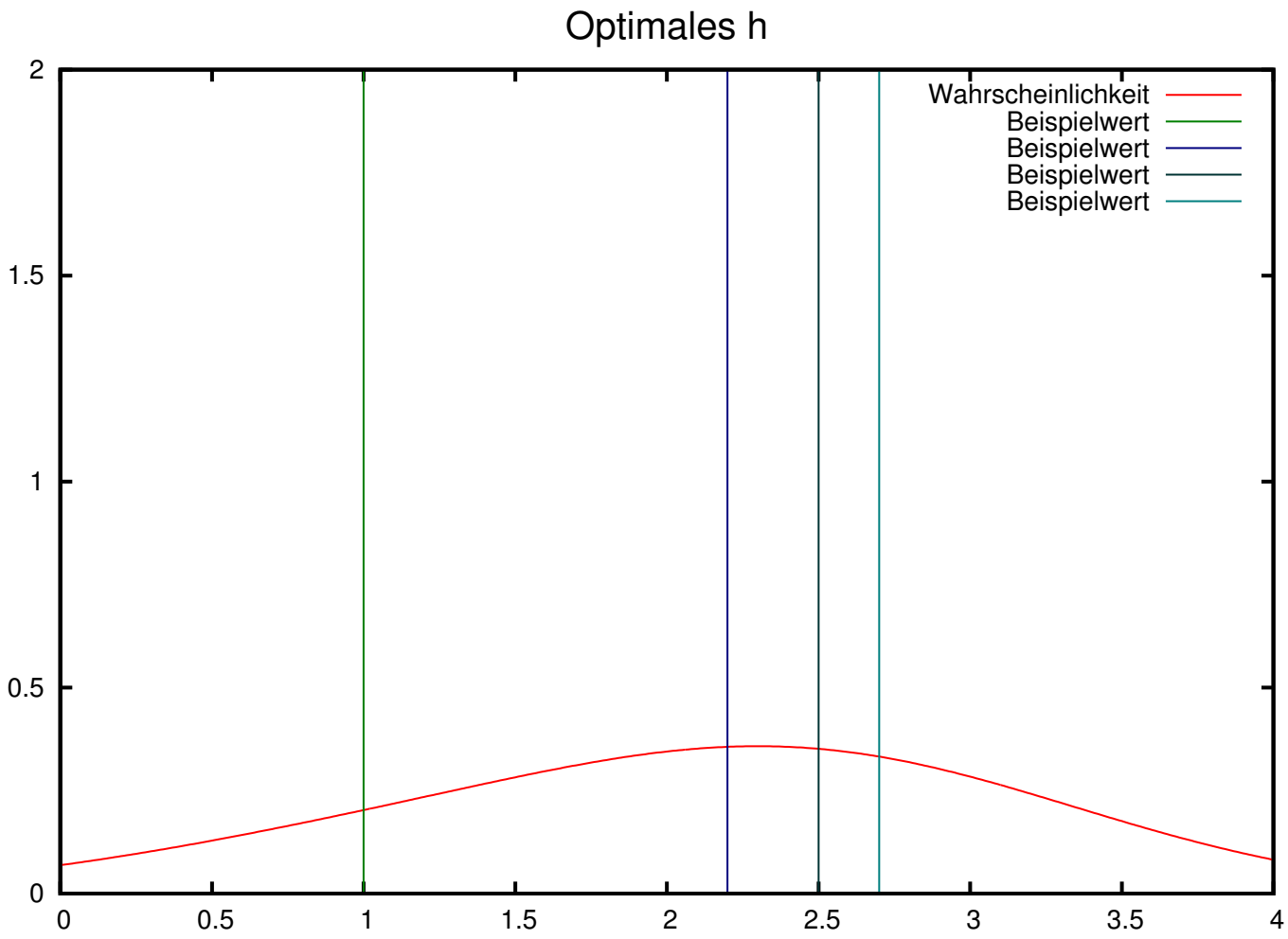
Exkurs: Die richtige Wahl des h (2)

Leave-One-Out-Cross-Validation

Maximierung der Wahrscheinlichkeit über alle i

$$h_{CV} = \arg\max_h \left\{ \frac{1}{n} \sum_{i=1}^{n} \log f^*(b_i) \right\}$$

Vorgehen: Ausprobieren von Werten für h über einem festgelegten Intervall.


Optimales h
Wahrscheinlichkeit
Beispielwert
Beispielwert
Beispielwert
Beispielwert
2
1.5
1
0.5
0
0
0.5
1
1.5
2
2.5
3
3.5
4

Gliederung

$p(A_i|Z)$ für numerische Attribute

$$p(Z|A) = \frac{p(Z) \cdot \prod_{i=1}^{n} p(A_i|Z)}{\int p(Z) \cdot \prod_{i=1}^{n} p(A_i|Z) dZ}$$

- Gesucht: $p(A_i|Z) =: p(B|Z) = \frac{p(B,Z)}{p(Z)}$
- $\hat{p}(B = b, Z = z)$ durch zweidimensionale Gauss-Interpolation
- $\hat{p}(Z)$: Gauss Interpolation über alle z

$$p(Z|A) = \frac{p(Z) \cdot \prod_{i=1}^{n} p(A_i|Z)}{\int p(Z) \cdot \prod_{i=1}^{n} p(A_i|Z) dZ}$$

- Gesucht: $p(A_i|Z) =: p(B|Z) = \frac{p(B,Z)}{p(Z)}$
- $\hat{p}(B = b, Z = z)$ durch zweidimensionale Gauss-Interpolation
- $\hat{p}(Z)$: Gauss Interpolation über alle z

Im ersten Item ist $p(Z)$ eigentlich $\int p(B,Z)db$. Hier vereinfacht dargestellt. Im Grunde haben wir hier schon $\hat{p}(Z)$ für den späteren Teil. Es wird nicht ersichtlich, ob dort auch das selbe h_z benutzt wird. Zur Erinnerung: Die Formel vom Beginn des Exkurs war

$$\frac{1}{n \cdot h} \sum_{i=1}^{n} K\left(\frac{b - b_i}{h}\right)$$

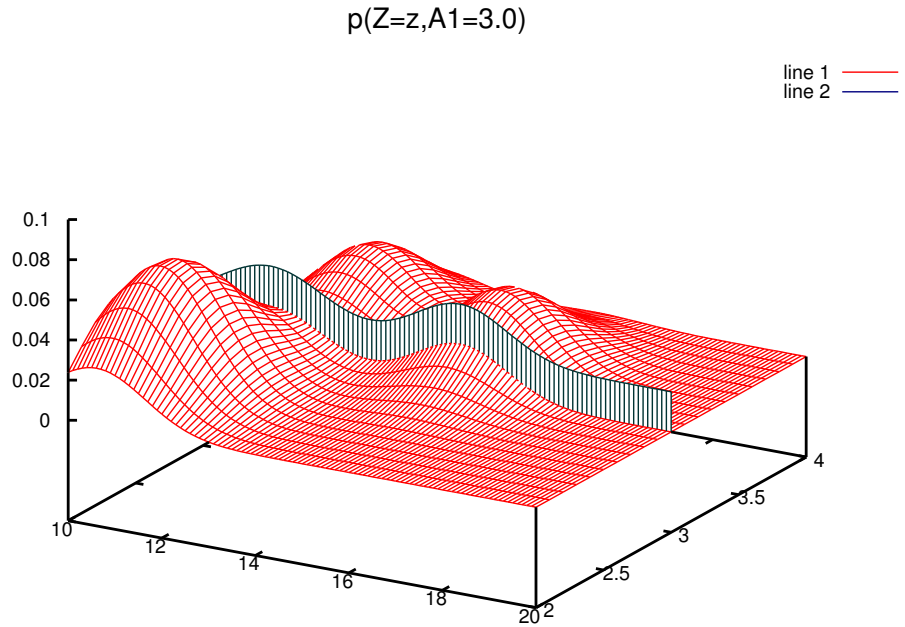
Die Formel auf dieser Folie ist lediglich die zweidimensonale Variante davon. Im Paper wird nicht explizit gesagt, ob das h_z für $\hat{p}(Z)$ das gleiche ist wie für den zweidimensonalen Part.

$\hat{p}(Z = z) = \frac{1}{nh_z} \sum_{i=1}^{n} K\left(\frac{k - z_i}{h_z}\right)$

Beispiel: Lebensdauer von Ameisen

Trainingsdaten: $Z \in [10, 20]$ Monate, $A_1 \in [2.0, 4.0]$ mm und $A_2 \in \{\text{rot,schwarz}\}$

Z	A_1	A_2
10.5	2.3	rot
15.5	3.3	schwarz
11.2	2.6	rot
12.1	3.5	schwarz
?	3.0	rot

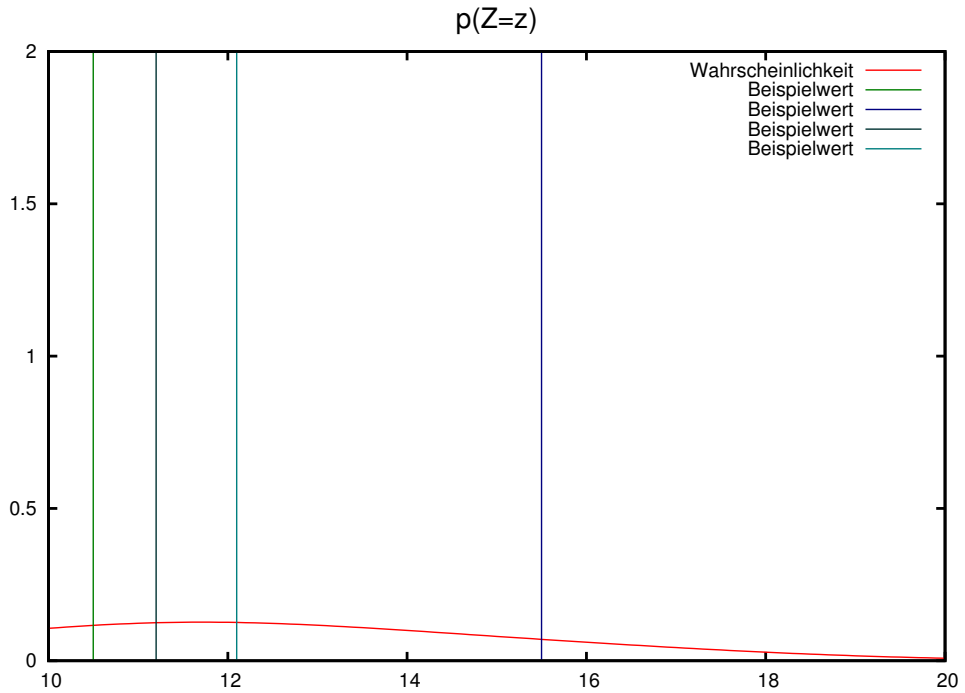


Berechnung von $p(A_1 = 3.0|Z = z)$ für alle $z \in \{10, 10.1, 10.2, ..., 20\}$

Beispiel: Lebensdauer von Ameisen

Trainingsdaten: $Z \in [10, 20]$ Monate, $A_1 \in [2.0, 4.0]$ mm und $A_2 \in \{\text{rot,schwarz}\}$

Z	A_1	A_2
10.5	2.3	rot
15.5	3.3	schwarz
11.2	2.6	rot
12.1	3.5	schwarz
?	3.0	rot



Berechnung von $p(A_1 = 3.0|Z = z)$ für alle $z \in \{10, 10.1, 10.2, ..., 20\}$

$p(A_i|Z)$ für nominale Attribute

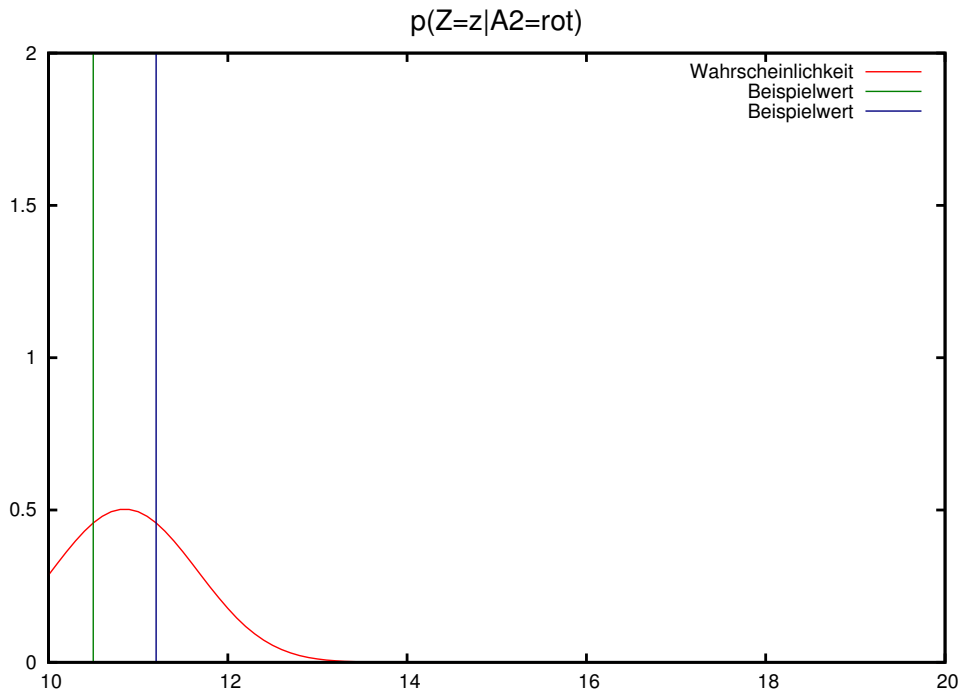
$$p(Z|A) = \frac{p(Z) \cdot \prod_{i=1}^{n} p(A_i|Z)}{\int p(Z) \cdot \prod_{i=1}^{n} p(A_i|Z) dZ}$$

- Nach Bayes:
  $p(A_i|Z) =: p(B = b|Z = z) = \frac{p(B=b)\ p(Z=z|B=b)}{\sum_{b \in \mathbf{Kat_B}} p(B=b)\ p(Z=z|B=b)}$
- $\hat{p}(Z = z|B = b)$: Gauss-Interpolation über alle Zielwerte von Beispielen mit $A_i = b$.
- $\hat{p}(B = b)$ ist der prozentuale Anteil der Beispiele mit $A_i = b$ an der Gesamtzahl der Beispiele.

$$\frac{p(A_2 = rot)\, p(Z = z|A_2 = rot)}{p(A_2 = rot)p(Z = z|B = rot) + p(A_2 = schw)p(Z = z|A_2 = schw)}$$

Z	A_1	A_2
10.5	2.3	rot
15.5	3.3	schwarz
11.2	2.6	rot
12.1	3.5	schwarz
?	3.0	rot

$p(A_2 = rot) = 0.5$

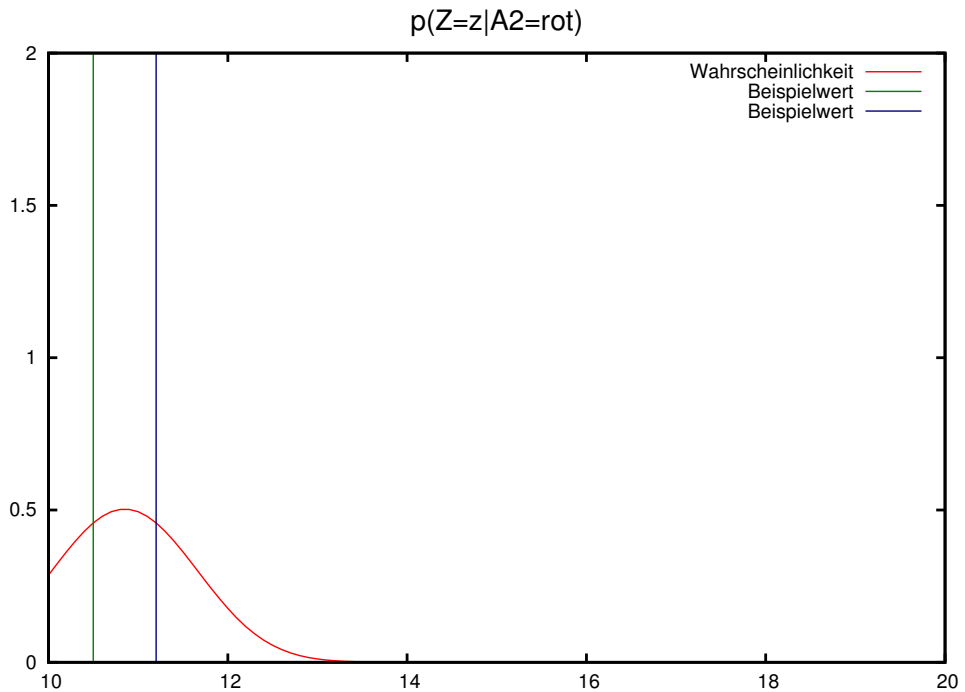


Berechnung von $p(A_2 = rot|Z = z)$ für alle $k \in \{10, 10.1, 10.2, ..., 20\}$

$$\frac{p(A_2 = rot)\, p(Z = z|A_2 = rot)}{p(A_2 = rot)p(Z = z|B = rot) + p(A_2 = schw)p(Z = z|A_2 = schw)}$$

Z	A_1	A_2
10.5	2.3	rot
15.5	3.3	schwarz
11.2	2.6	rot
12.1	3.5	schwarz
?	3.0	rot

$p(A_2 = rot) = 0.5$



Berechnung von $p(A_2 = rot|Z = z)$ für alle $k \in \{10, 10.1, 10.2, ..., 20\}$

Berechnung von $p(z)$

$$p(Z|A) = \frac{p(Z) \cdot \prod_{i=1}^{n} p(A_i|Z)}{\int p(Z) \cdot \prod_{i=1}^{n} p(A_i|Z) dZ}$$

- Es fehlt noch: $p(Z)$
- $\hat{p}(z)$: Gauss-Interpolation über die Zielwerte aller Objekte in den Trainingsdaten

Berechnung von $p(Z = z)$ für alle $z \in \{10, 10.1, 10.2, ..., 20\}$

$$p(Z|A) = \frac{p(Z) \cdot \prod_{i=1}^{n} p(A_i|Z)}{\int p(Z) \cdot \prod_{i=1}^{n} p(A_i|Z) dZ}$$

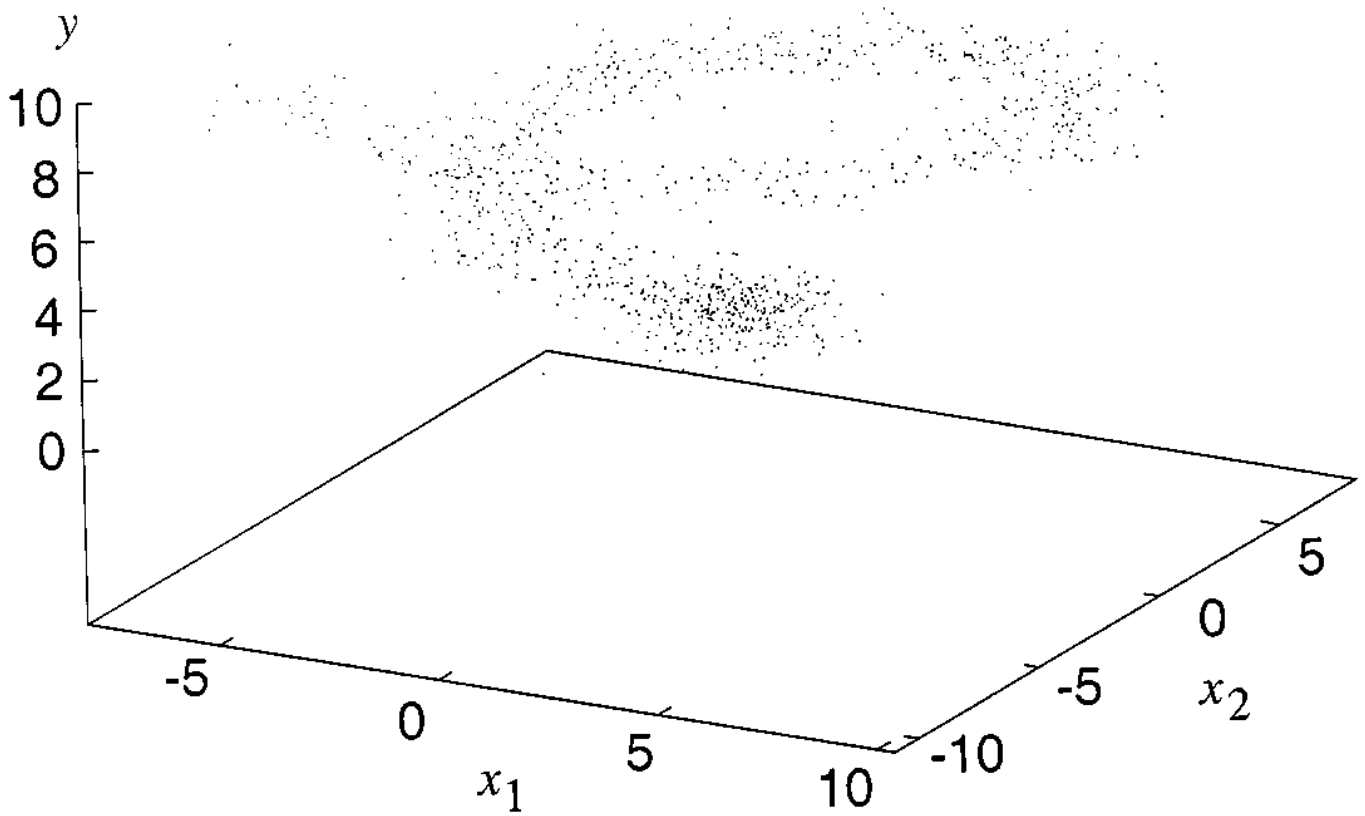
- $p(Z = z|A)$ ist nun bekannt für das gesamte Intervall
- Entweder: Minimierung des quadratischen Fehlers
- Zielwert ist der Erwartungswert
- Oder: Minimierung des absoluten Fehlers
- Zielwert ist der Median

Gliederung

Vergleiche mit anderen Algorithmen

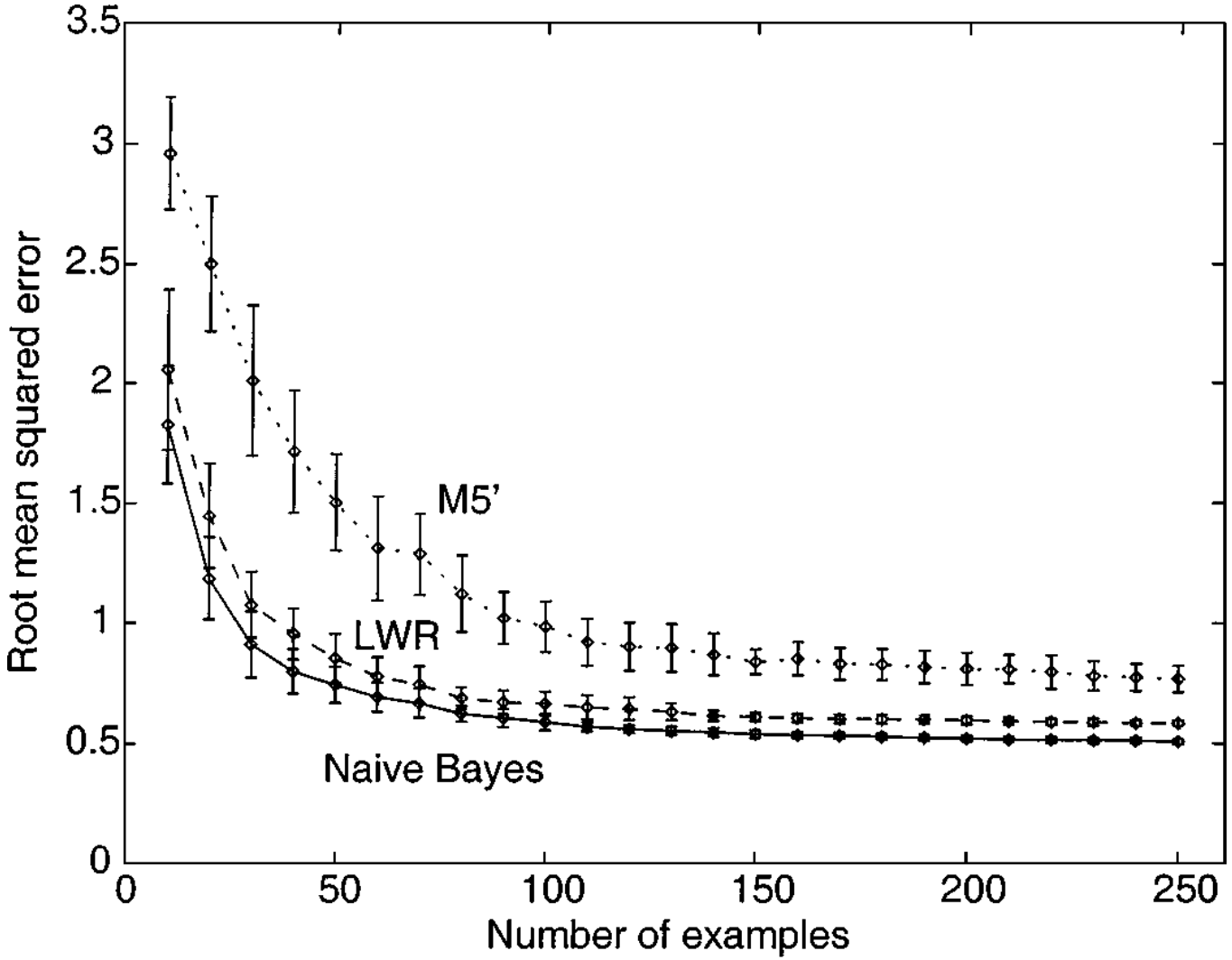
- Locally Weighted Linear Regression (LWR)
- Linear Regression (LR)
- Model Tree Prediction (M5')

3D-Spirale $(x1, x2) \rightarrow y$



Auswertung

Vergleich mit dem Model Tree Inducer M5'



Vergleich mit anderen Lernalgorithmen

Auswahl aus 41 Datensätzen

Datensatz	Instanzen	Fehlend	Numerisch	Nominal
Schlvote	38	0.4 %	4	1
EchoMonths	130	7.5 %	6	3
BreastTumor	286	0.3 %	1	8
Meta	528	4.3 %	19	2

Fehler bei verschiedenen Algorithmen

Durchschnittlicher Quadratischer Fehler

Datensatz	Naive Bayes	LR	LWR	M5'
Schlvote	95.92±7.2	114.23±3.6 •	118.81±6.6 •	94.00±10.2
EchoMonths	78.53±1.5	68.25±1.4 ◦	68.04±1.1 ◦	71.01±0.7 ◦
BreastTumor	100.96±1.2	97.43±1.2 ◦	103.05±1.2 •	97.29±0.6 ◦
Meta	160.49±17.4	202.18±11.8 •	160.29±10.4	150.68±32.2

Durschnittlicher Absoluter Fehler

Datensatz	Naive Bayes	LR	LWR	M5'
Schlvote	90.13±7.0	112.43±4.3 •	114.23±7.1 •	89.78±7.4
EchoMonths	72.34±1.5	65.42±1.5 ◦	64.30±1.2 ◦	67.95±0.7 ◦
BreastTumor	104.26±0.9	99.29±1.6 ◦	106.25±1.4 •	99.91±0.6 ◦
Meta	78.44±3.7	146.42±4.5 •	104.90±3.6 •	79.00±8.5

Fehler bei verschiedenen Algorithmen

Durchschnittlicher Quadratischer Fehler

Datensatz	Naive Bayes	LR	LWR	M5'
Schlvote	95.92±7.2	114.23±3.6 ●	118.81±6.6 ●	94.00±10.2
EchoMonths	78.53±1.5	68.25±1.4 ○	68.04±1.1 ○	71.01±0.7 ○
BreastTumor	100.96±1.2	97.43±1.2 ○	103.05±1.2 ●	97.29±0.6 ○
Meta	160.49±17.4	202.18±11.8 ●	160.29±10.4	150.68±32.2

Durschnittlicher Absoluter Fehler

Datensatz	Naive Bayes	LR	LWR	M5'
Schlvote	90.13±7.0	112.43±4.3 ●	114.23±7.1 ●	89.78±7.4
EchoMonths	72.34±1.5	65.42±1.5 ○	64.30±1.2 ○	67.95±0.7 ○
BreastTumor	104.26±0.9	99.29±1.6 ○	106.25±1.4 ●	99.91±0.6 ○
Meta	78.44±3.7	146.42±4.5 ●	104.90±3.6 ●	79.00±8.5

- heisst: Wesentlich schlechter! *circle* heisst: Wesentlich besser

Direktvergleich zwischen den Algorithmen

Verlierer ⇓	Gewinner Naive Bayes	LR	LWR	M5'
Naive Bayes		18	20	23
LR	8		13	15
LWR	6	10		15
M5'	3	4	6	

Durchschnittlicher Quadratischer Fehler

Verlierer ⇓	Gewinner Naive Bayes	LR	LWR	M5'
Naive Bayes		13	19	22
LR	13		17	16
LWR	6	9		19
M5'	5	5	8	

Durschnittlicher Absoluter Fehler

Direktvergleich zwischen den Algorithmen

	Gewinner			
Verlierer ⇓	Naive Bayes	LR	LWR	M5'
Naive Bayes		18	20	23
LR	8		13	15
LWR	6	10		15
M5'	3	4	6	

Durchschnittlicher Quadratischer Fehler

	Gewinner			
Verlierer ⇓	Naive Bayes	LR	LWR	M5'
Naive Bayes		13	19	22
LR	13		17	16
LWR	6	9		19
M5'	5	5	8	

Durschnittlicher Absoluter Fehler

Vergleich mit Naive Bayes für Klassifikation

Liegt der Fehler in der Ableitung des Naive Bayes?

- Modellierung eines Klassifikationsproblems als Regression:
  - Ein Regressionsproblem pro Zielklasse
- Tests auf Standard-Klassifikations-Datensätzen
- Ergebnis: Besseres Abschneiden als der normale Naive Bayes

Vergleich mit einer modifizierten Version des M5'

Liegt es an der Unabhängigkeitsannahme?

- M5'Independent: Mit Unabhängigkeitsannahme
- Direktvergleiche:
  - M5' gegen M5'Independent
  - M5' gegen Naive Bayes
- Ergebnis: M5'Independent schneidet im Vergleich genauso schlecht gegen M5' ab wie Naive Bayes

Fazit

Für Regressionsprobleme gilt: Unabhängigkeit der Attribute

- erfüllt: Naive Bayes funktioniert gut
- nicht-erfüllt: Andere Algorithmnen schneiden besser ab

Vielen Dank für eure Aufmerksamkeit!

Quellen

- Technical Notes: Naive Bayes for Regression; E. Frank, L.Trigg, G.Holmes, I.H.Witten; Machine Learning 41, 5-25, 2000
- Retrofitting Decision Tree Classifiers Using Kernel Density Estimation; P.Smyth, A.Gray, U.M.Fayyad (Appendix: Univariate Bandwidth Selection for Kernel Density Estimation)
- Naive Bayes zur Klassifikation: http://www.ke.informatik.tu-darmstadt.de/lehre/ws05/mldm/bayes.pdf