Unsupervised Kernel Regression

Stefan Klanke

9. Mai 2006

Inhalt

- Nichtlineare Dimensionsreduktion mittels UKR (Unüberwachte KernRegression, 2005)
- Anknüpfungspunkte Datamining I: PCA + Hauptkurven
- Benötigte Zutaten
  - Klassische Kernregression
  - Kerndichteschätzung
- Wir arbeiten uns von unten nach oben

Inhalt

- Nichtlineare Dimensionsreduktion mittels UKR (Unüberwachte KernRegression, 2005)
- Anknüpfungspunkte Datamining I: PCA + Hauptkurven
- Benötigte Zutaten
  - Klassische Kernregression
  - Kerndichteschätzung
- Wir arbeiten uns von unten nach oben

Inhalt

- Nichtlineare Dimensionsreduktion mittels UKR (Unüberwachte KernRegression, 2005)
- Anknüpfungspunkte Datamining I: PCA + Hauptkurven
- Benötigte Zutaten
  - Klassische Kernregression
  - Kerndichteschätzung
- Wir arbeiten uns von unten nach oben

Inhalt

- Nichtlineare Dimensionsreduktion mittels UKR (Unüberwachte KernRegression, 2005)
- Anknüpfungspunkte Datamining I: PCA + Hauptkurven
- Benötigte Zutaten
  - Klassische Kernregression
  - Kerndichteschätzung
- Wir arbeiten uns von unten nach oben

Inhalt

- Nichtlineare Dimensionsreduktion mittels UKR (Unüberwachte KernRegression, 2005)
- Anknüpfungspunkte Datamining I: PCA + Hauptkurven
- Benötigte Zutaten
  - Klassische Kernregression
  - Kerndichteschätzung
- Wir arbeiten uns von unten nach oben

Dichteschätzung

- Gegeben: Ein Datensatz x_i, $i = 1 \ldots N$.
- Gesucht: Die Wahrscheinlichkeitsdichte $p(x)$, nach der die Daten verteilt sind.
- Zwei Klassen von Verfahren
  - **Parametrische Verfahren** legen eine bestimmte Form der Verteilung zugrunde und optimieren dann deren Parameter z.B. Annahme x ist normalverteilt $\rightarrow$ bestimme "passenden" Mittelwert und Standardabweichung.
  - **Nicht parametrische Verfahren** machen keine Annahmen über die Verteilung. Ihre Form wird von den Daten selbst bestimmt. Dabei gibt es meistens nur eine "Stellschraube". Beispiel: Histogramm (Datamining I)

Dichteschätzung

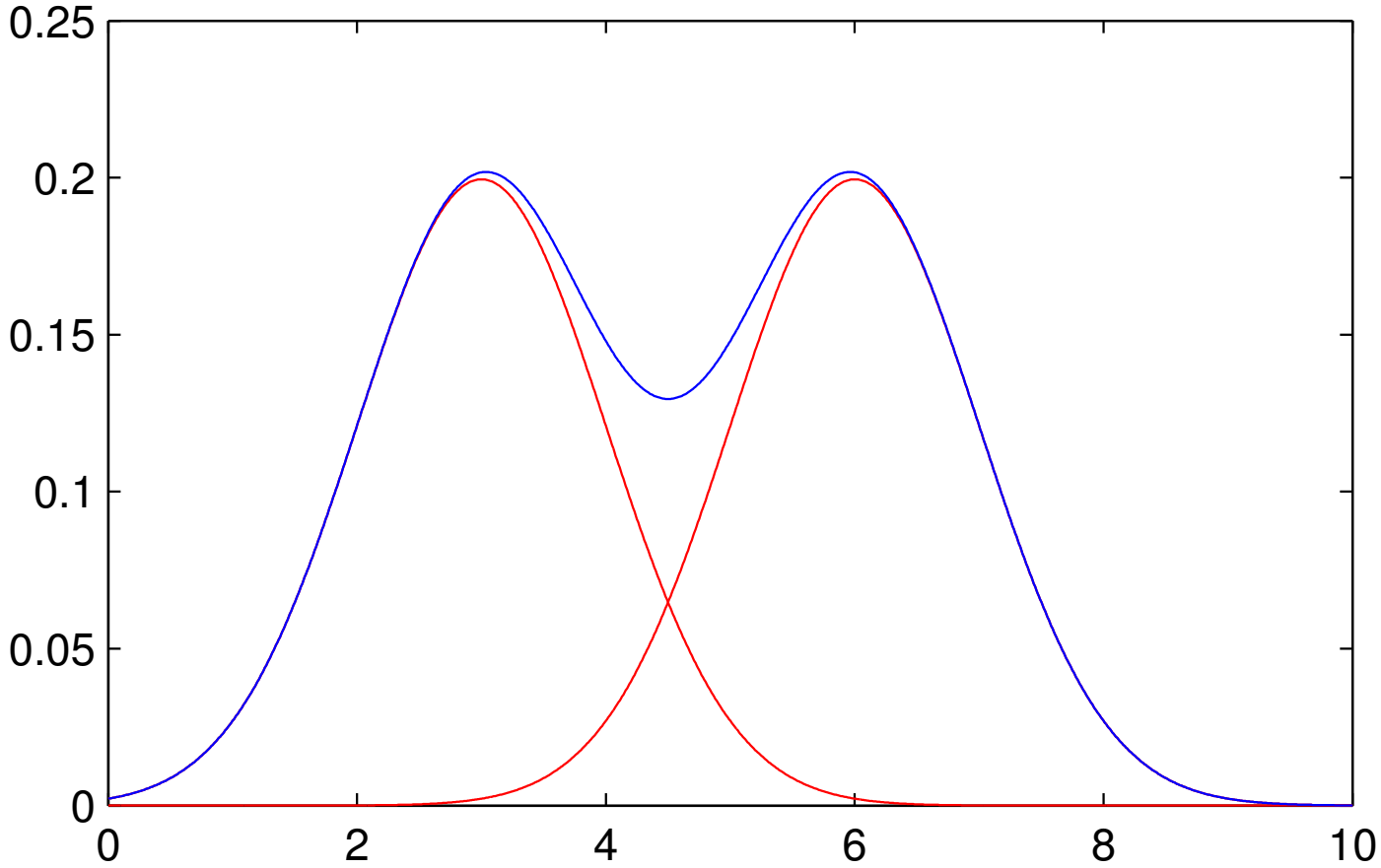
- Gegeben: Ein Datensatz x_i, $i = 1 \ldots N$.
- Gesucht: Die Wahrscheinlichkeitsdichte $p(x)$, nach der die Daten verteilt sind.
- Zwei Klassen von Verfahren
  - **Parametrische Verfahren** legen eine bestimmte Form der Verteilung zugrunde und optimieren dann deren Parameter z.B. Annahme x ist normalverteilt $\rightarrow$ bestimme "passenden" Mittelwert und Standardabweichung.
  - **Nicht parametrische Verfahren** machen keine Annahmen über die Verteilung. Ihre Form wird von den Daten selbst bestimmt. Dabei gibt es meistens nur eine "Stellschraube". Beispiel: Histogramm (Datamining I)

Dichteschätzung

- Gegeben: Ein Datensatz x_i, $i = 1 \ldots N$.
- Gesucht: Die Wahrscheinlichkeitsdichte $p(x)$, nach der die Daten verteilt sind.
- Zwei Klassen von Verfahren
  - **Parametrische Verfahren** legen eine bestimmte Form der Verteilung zugrunde und optimieren dann deren Parameter z.B. Annahme x ist normalverteilt $\rightarrow$ bestimme "passenden" Mittelwert und Standardabweichung.
  - **Nicht parametrische Verfahren** machen keine Annahmen über die Verteilung. Ihre Form wird von den Daten selbst bestimmt. Dabei gibt es meistens nur eine "Stellschraube". Beispiel: Histogramm (Datamining I)

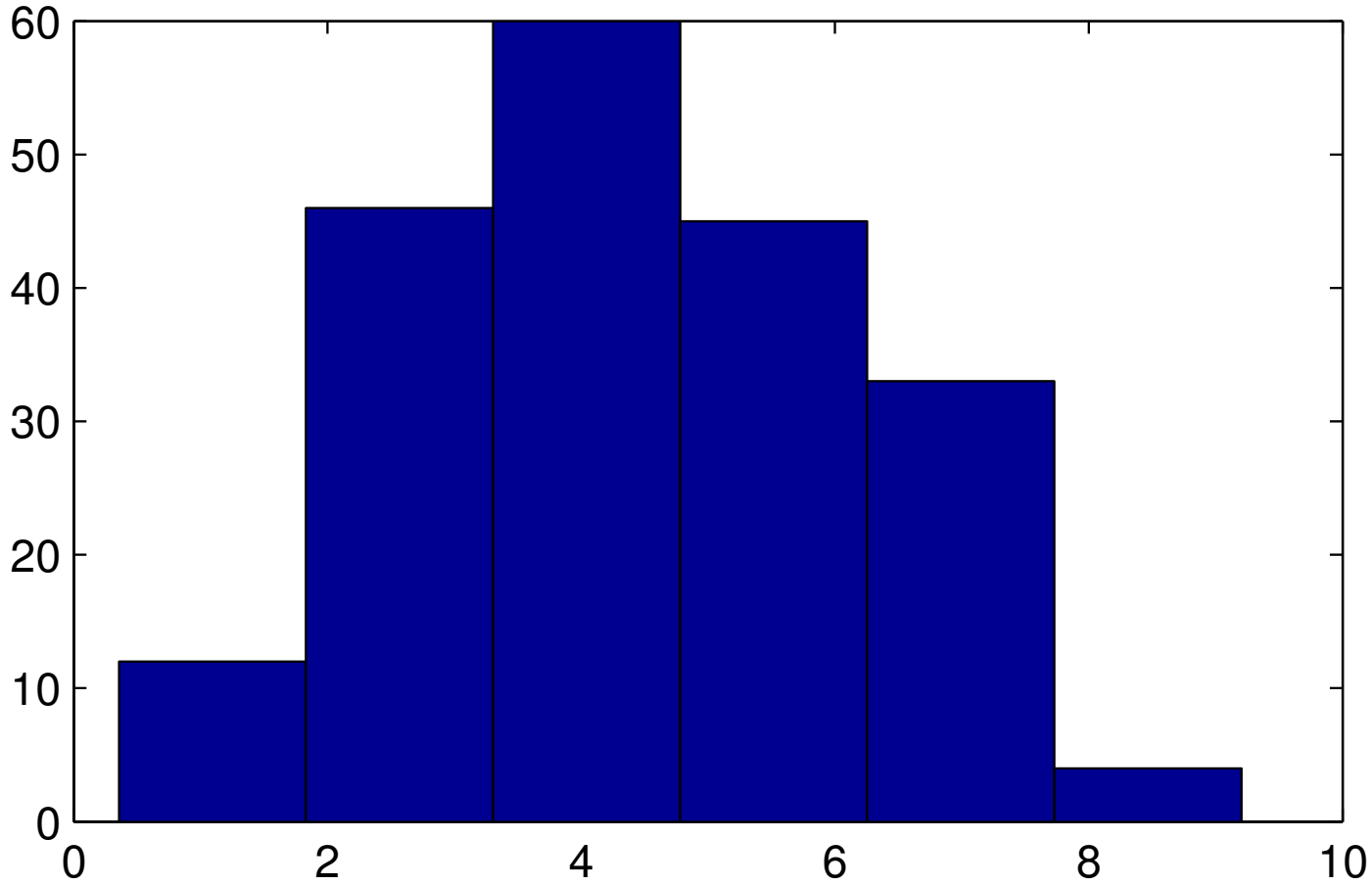
Dichteschätzung

- Beispiel: 200 Datenpunkte gemäß dieser Verteilung



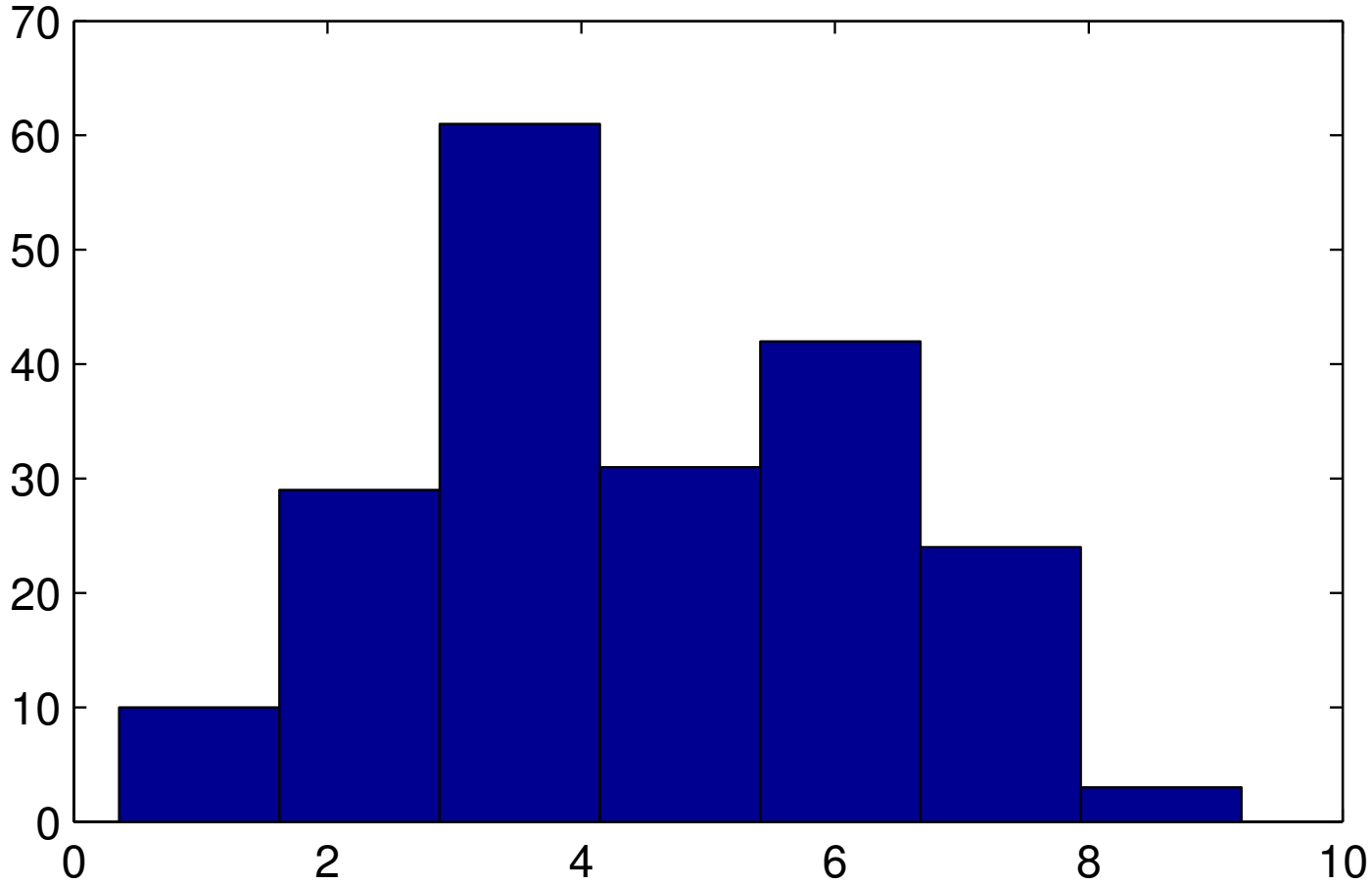
Dichteschätzung: Histogramm

- Histogramm mit 6 Balken



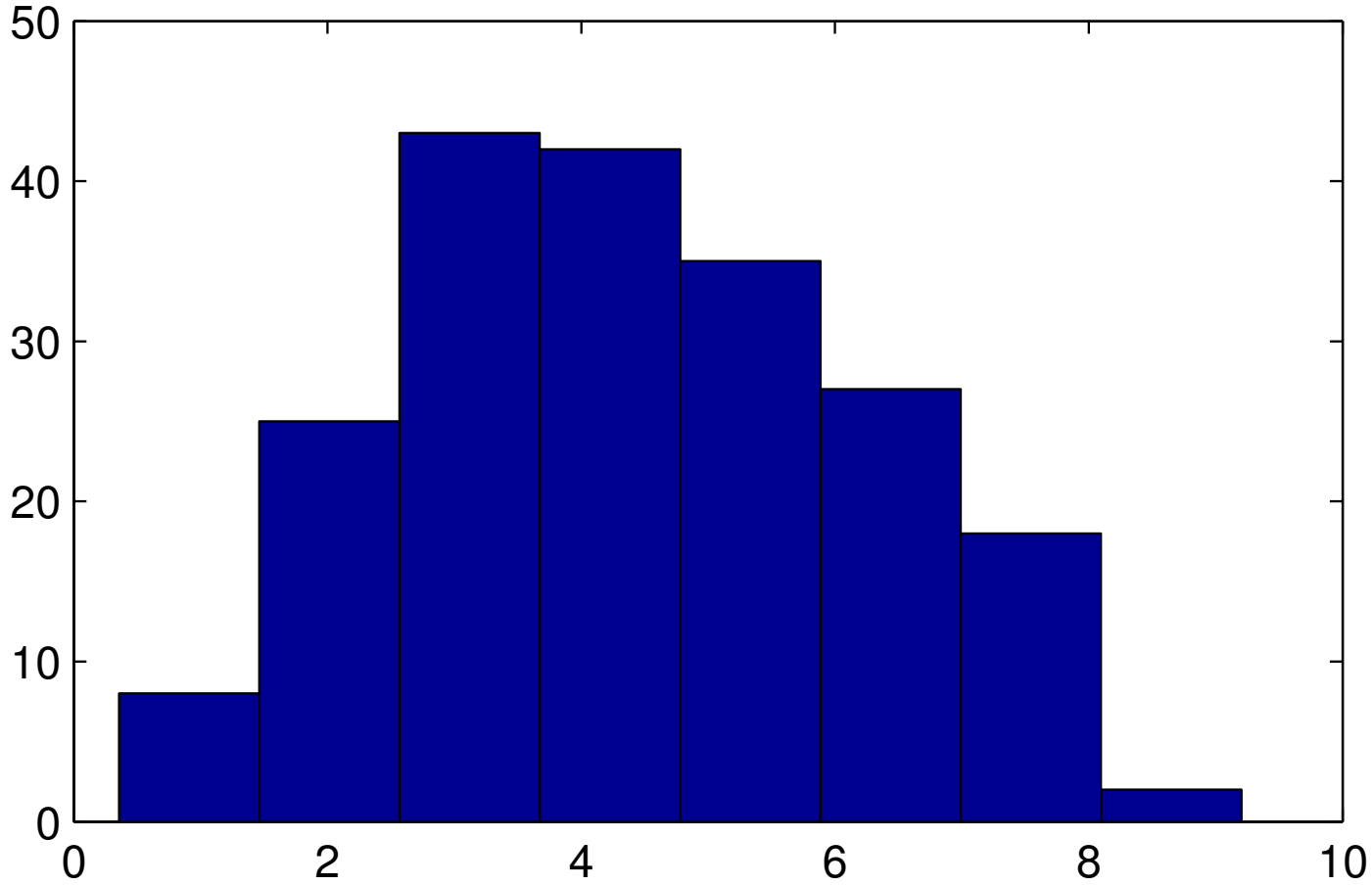
Dichteschätzung: Histogramm

- Histogramm mit 7 Balken



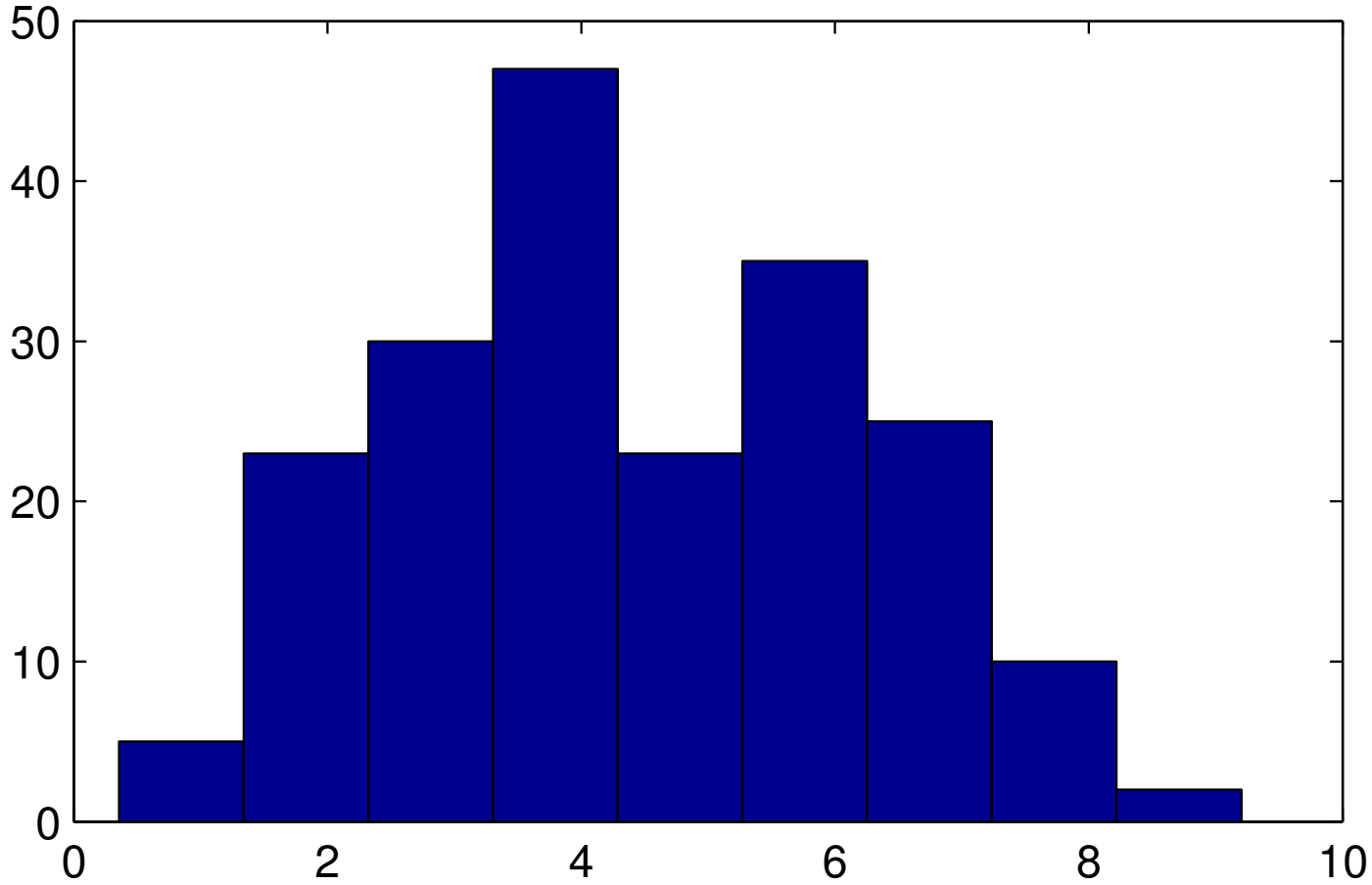
Dichteschätzung: Histogramm

- Histogramm mit 8 Balken



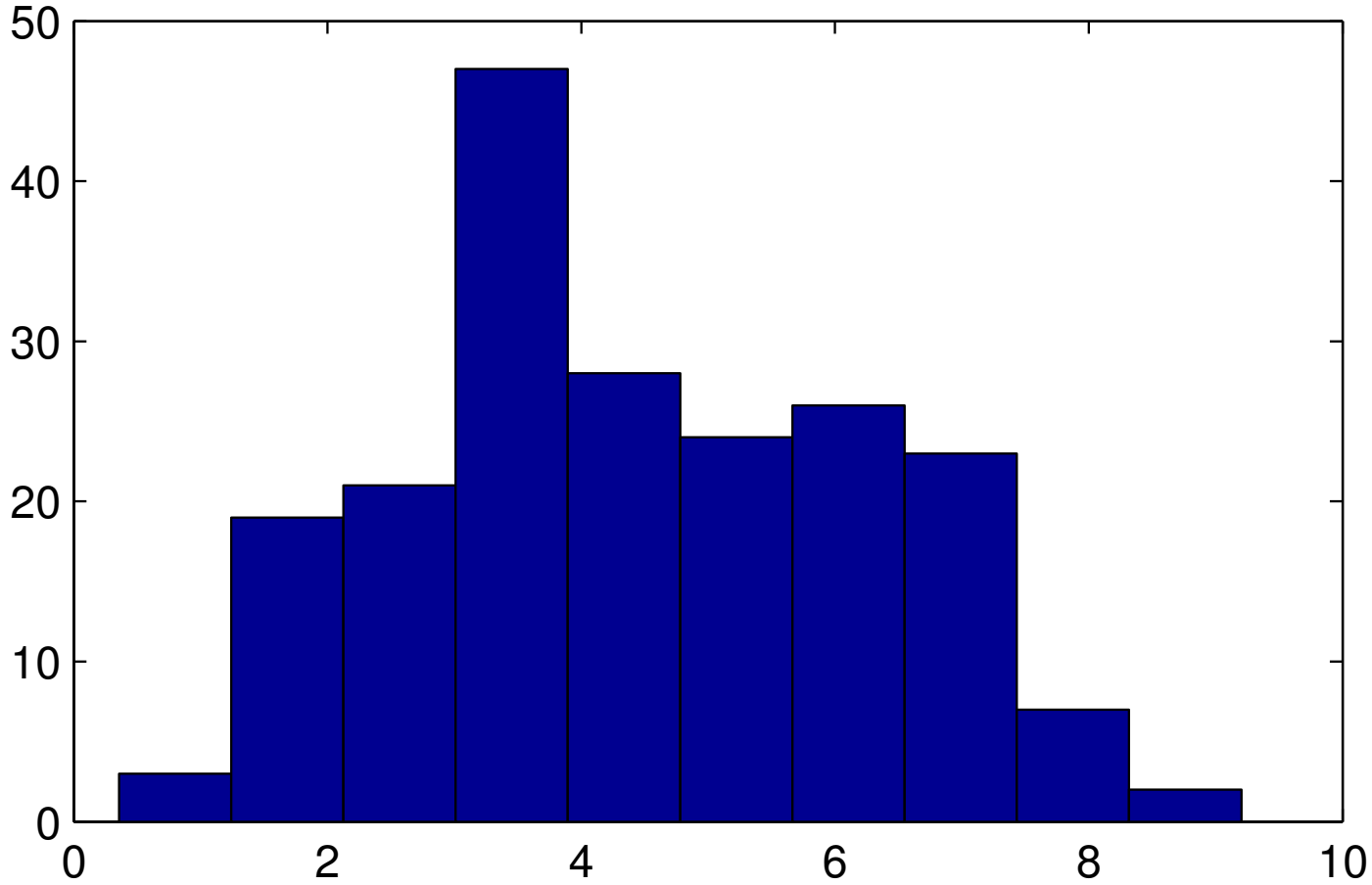
Dichteschätzung: Histogramm

- Histogramm mit 9 Balken



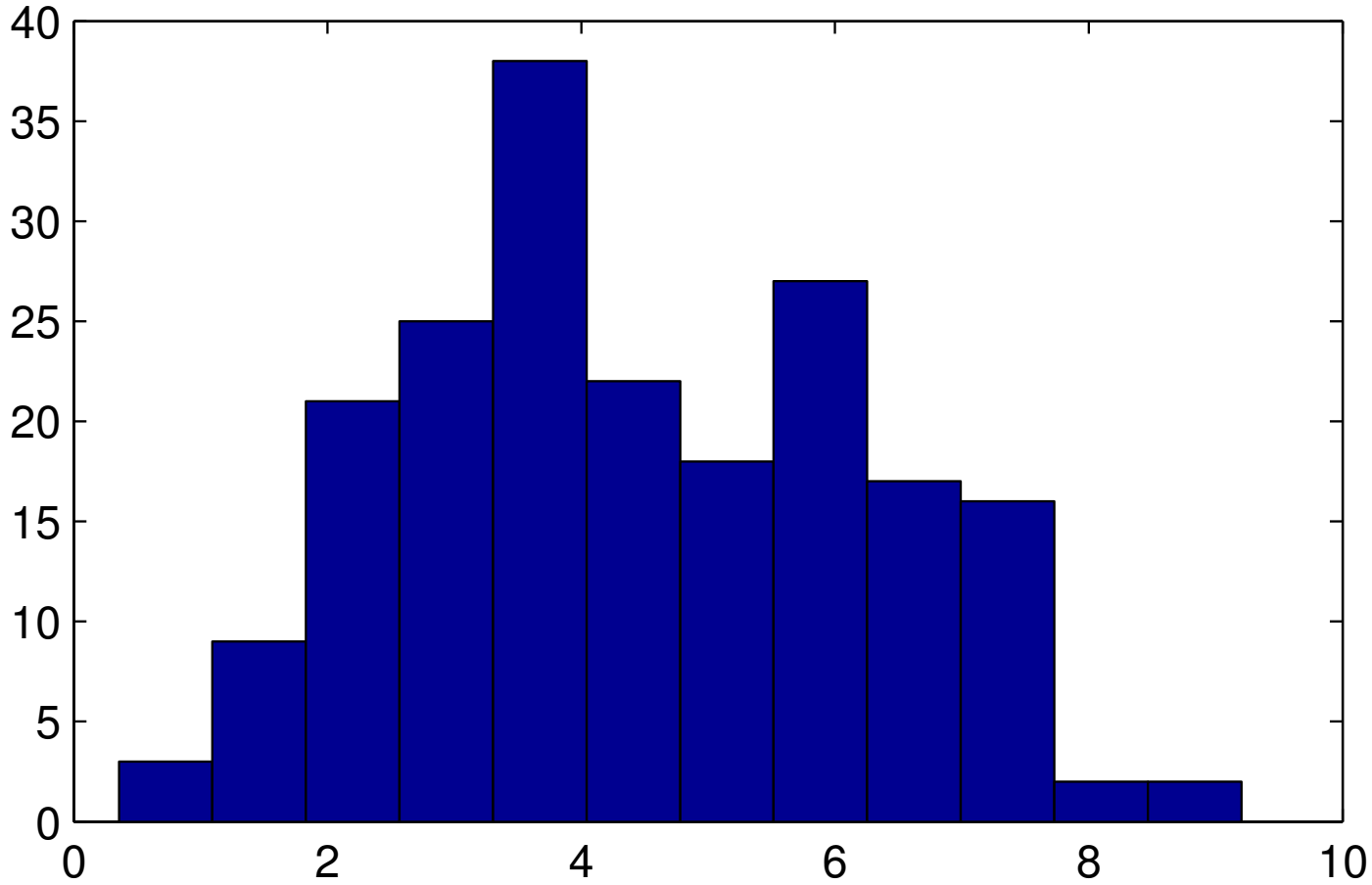
Dichteschätzung: Histogramm

- Histogramm mit 10 Balken



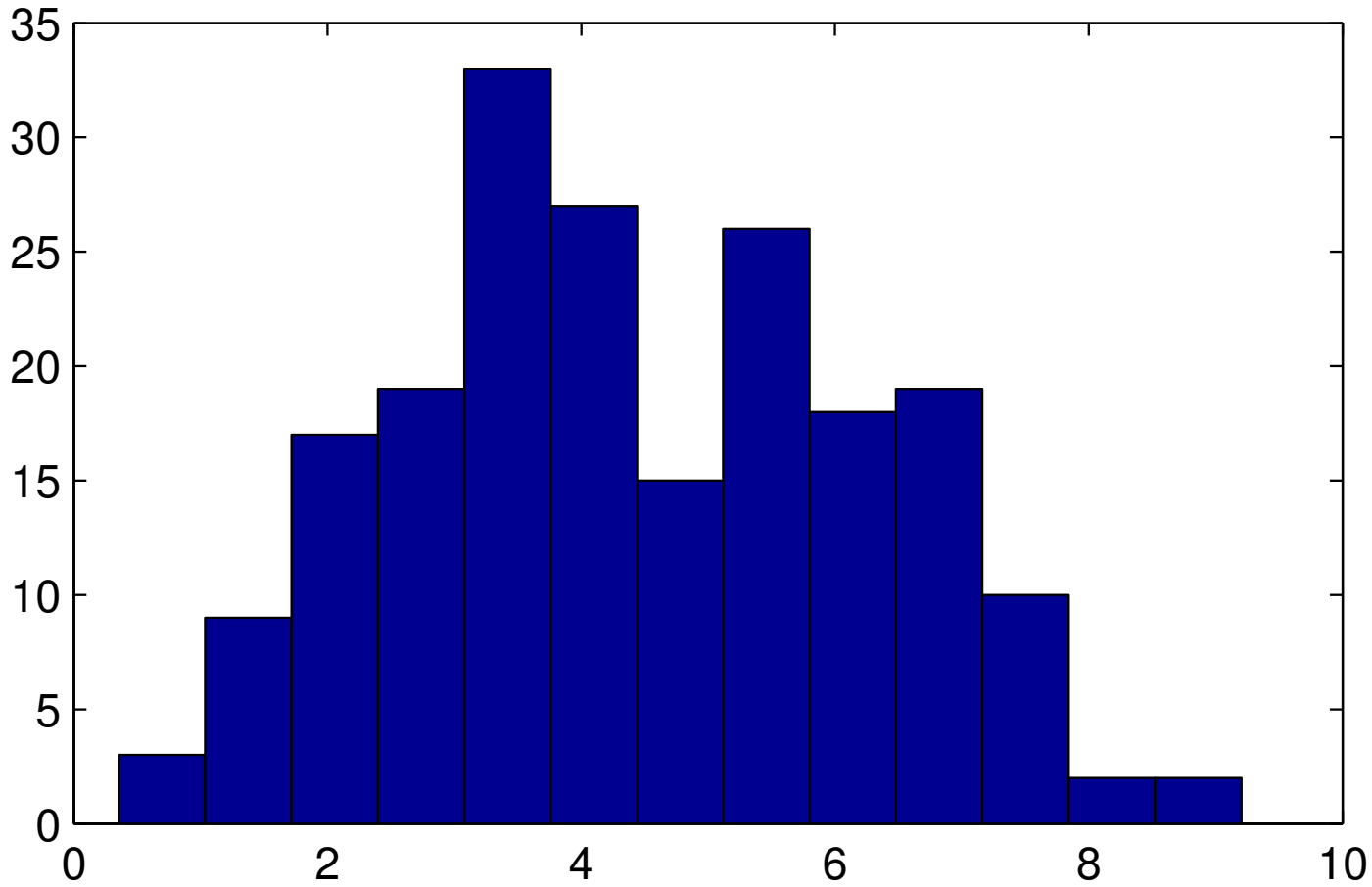
Dichteschätzung: Histogramm

- Histogramm mit 12 Balken



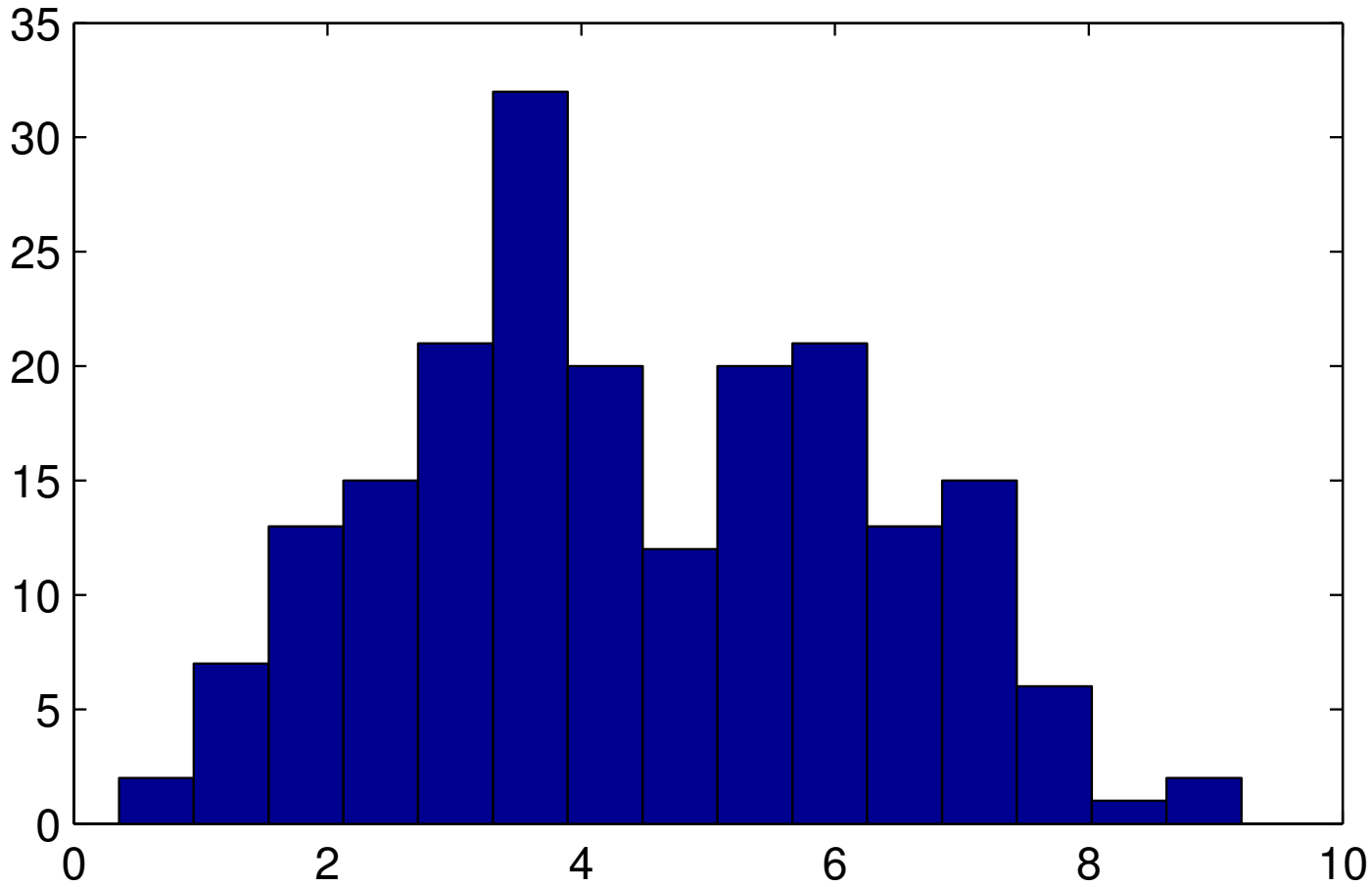
Dichteschätzung: Histogramm

- Histogramm mit 13 Balken



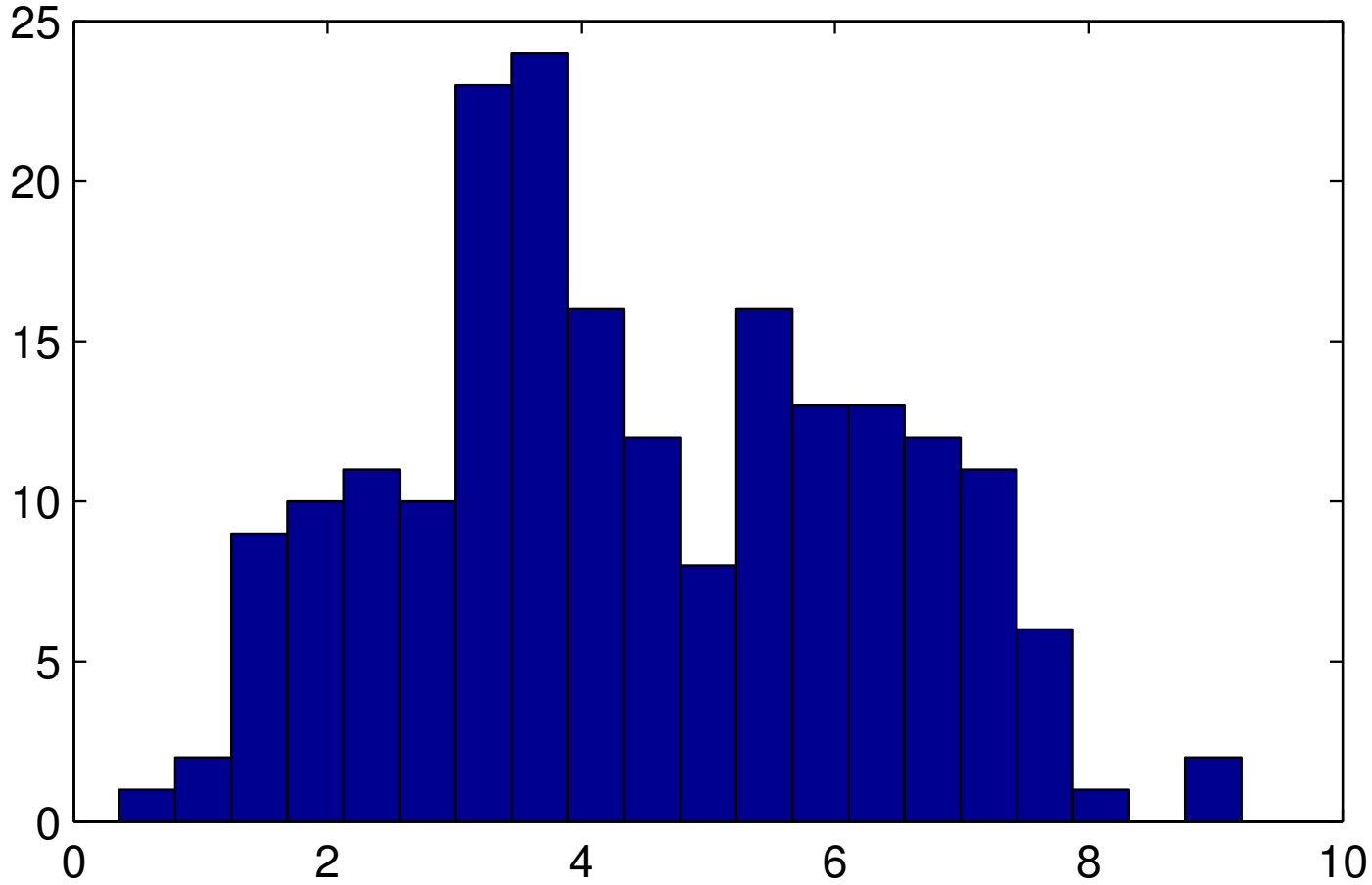
Dichteschätzung: Histogramm

- Histogramm mit 15 Balken



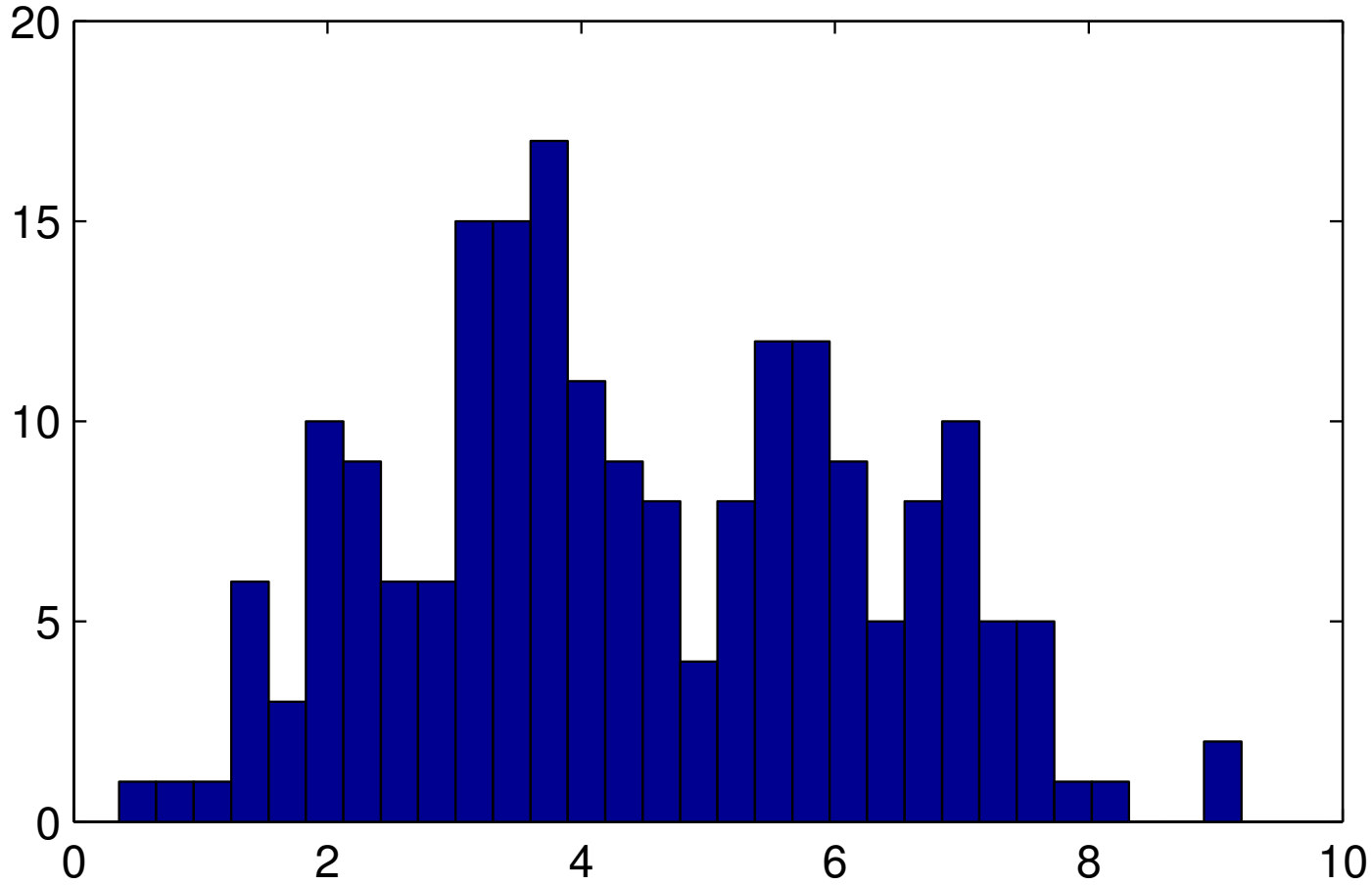
Dichteschätzung: Histogramm

- Histogramm mit 20 Balken



Dichteschätzung: Histogramm

- Histogramm mit 30 Balken



Dichteschätzung

- Beim Histogramm zählt man, wieviele Daten in einen bestimmten Bereich ("Bin") fallen.
- Modifikation: Zähle an jeder x-Koordinate, wie viele Daten in der Nachbarschaft liegen. Alternatives Bild: Schiebe ein Fenster der Breite h über die x-Achse und zähle, wieviele Daten das Fenster freilegt.
- Dies lässt sich wie folgt formalisieren:

$$K_U(v) = \begin{cases} 0 & \text{falls } |v| > h \\ \frac{1}{2h} & \text{sonst} \end{cases} \tag{1}$$

$$p(x) = \frac{1}{N} \sum_i K_U(x - x_i) \tag{2}$$

Rosenblatt (1956), Parzen (1962)

Dichteschätzung

- Beim Histogramm zählt man, wieviele Daten in einen bestimmten Bereich ("Bin") fallen.
- Modifikation: Zähle an jeder x-Koordinate, wie viele Daten in der Nachbarschaft liegen. Alternatives Bild: Schiebe ein Fenster der Breite h über die x-Achse und zähle, wieviele Daten das Fenster freilegt.
- Dies lässt sich wie folgt formalisieren:

$$K_U(v) = \begin{cases} 0 & \text{falls } |v| > h \\ \frac{1}{2h} & \text{sonst} \end{cases} \tag{1}$$

$$p(x) = \frac{1}{N} \sum_i K_U(x - x_i) \tag{2}$$

Rosenblatt (1956), Parzen (1962)

Dichteschätzung

- Beim Histogramm zählt man, wieviele Daten in einen bestimmten Bereich ("Bin") fallen.
- Modifikation: Zähle an jeder x-Koordinate, wie viele Daten in der Nachbarschaft liegen. Alternatives Bild: Schiebe ein Fenster der Breite h über die x-Achse und zähle, wieviele Daten das Fenster freilegt.
- Dies lässt sich wie folgt formalisieren:

$$K_U(v) = \begin{cases} 0 & \text{falls } |v| > h \\ \frac{1}{2h} & \text{sonst} \end{cases} \tag{1}$$

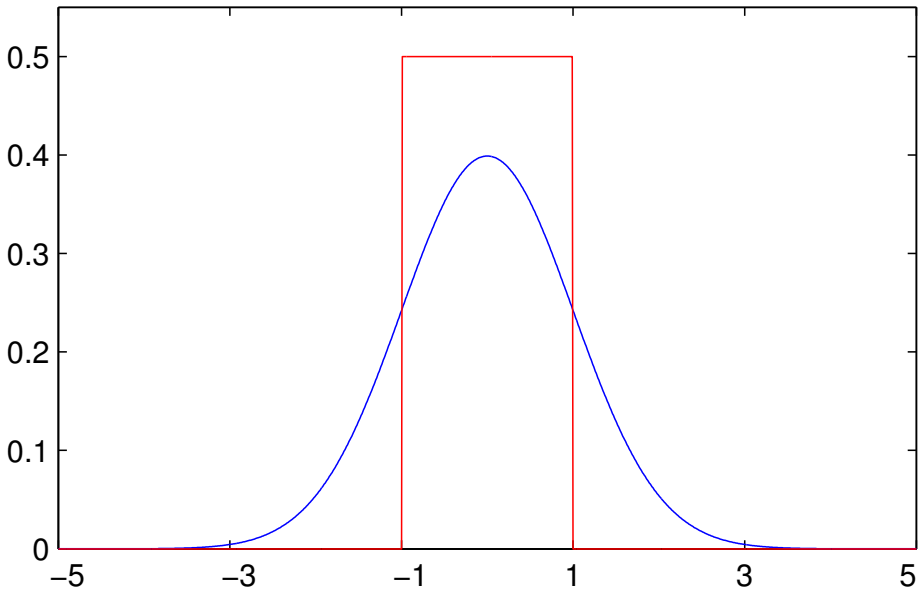
$$p(x) = \frac{1}{N} \sum_i K_U(x - x_i) \tag{2}$$

Rosenblatt (1956), Parzen (1962)

Kerndichteschätzung

- $K_U(v)$ ist die sog. “uniforme Kernfunktion”
- Lässt sich verallgemeinern, z.B. Gausskern:

$$K_G(v) = \frac{1}{\sqrt{2\pi h^2}} e^{-\frac{v^2}{2h^2}} \tag{3}$$



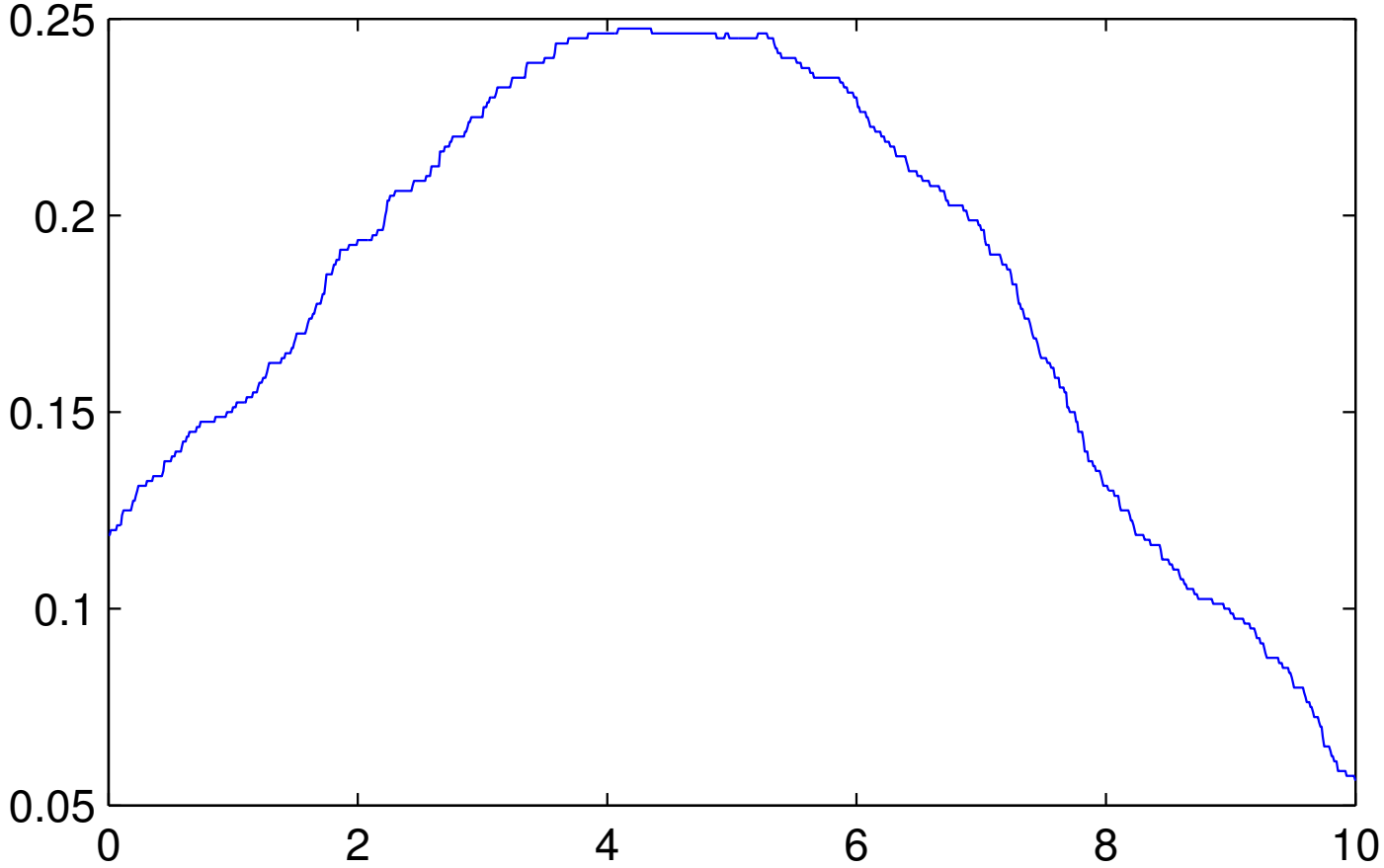
Oft Bedingungen

$$\int K(v)dv = 1$$

$$\int vK(v)dv = 0$$

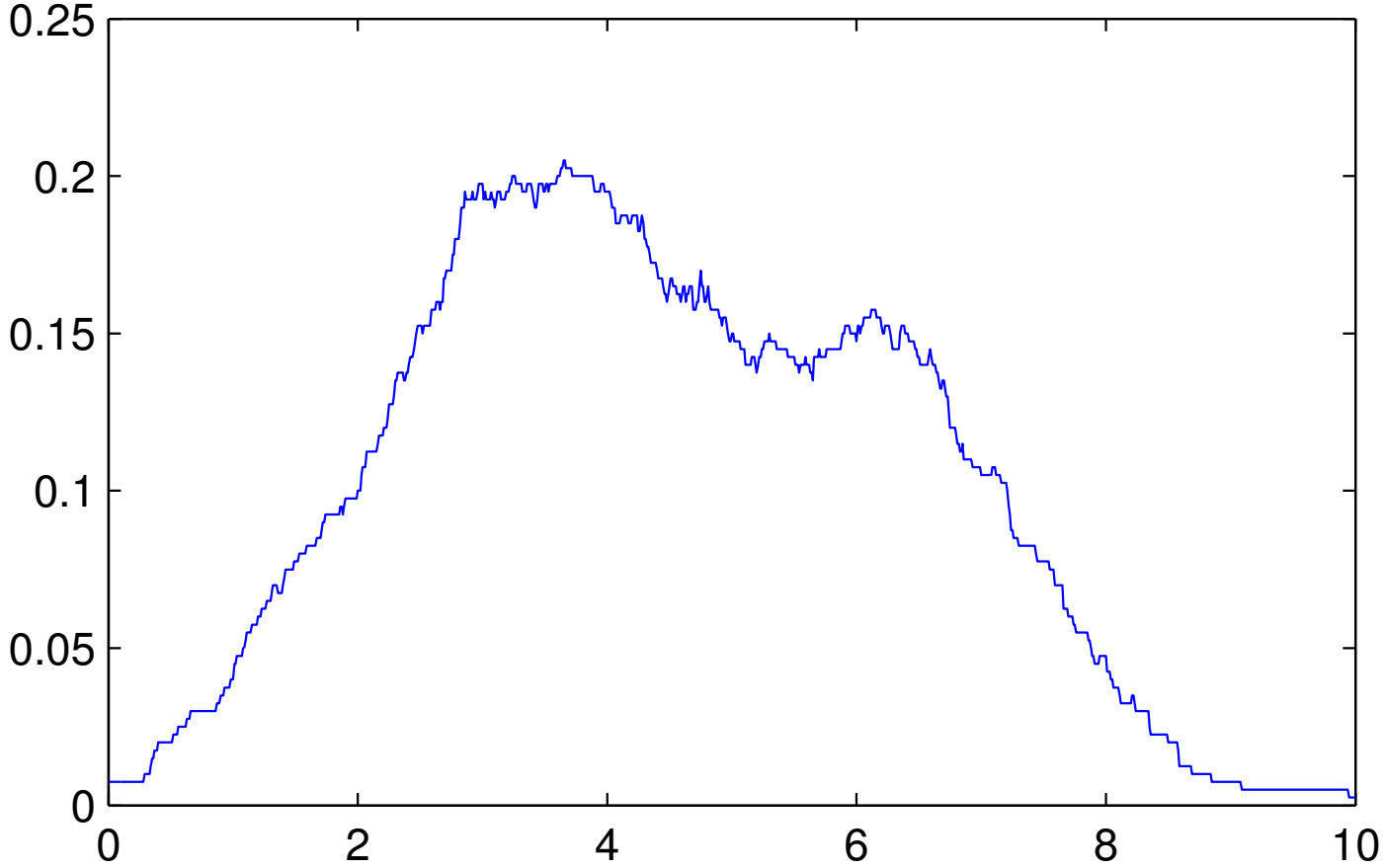
Kerndichteschätzung: uniformer Kern

- Bandbreite h=2



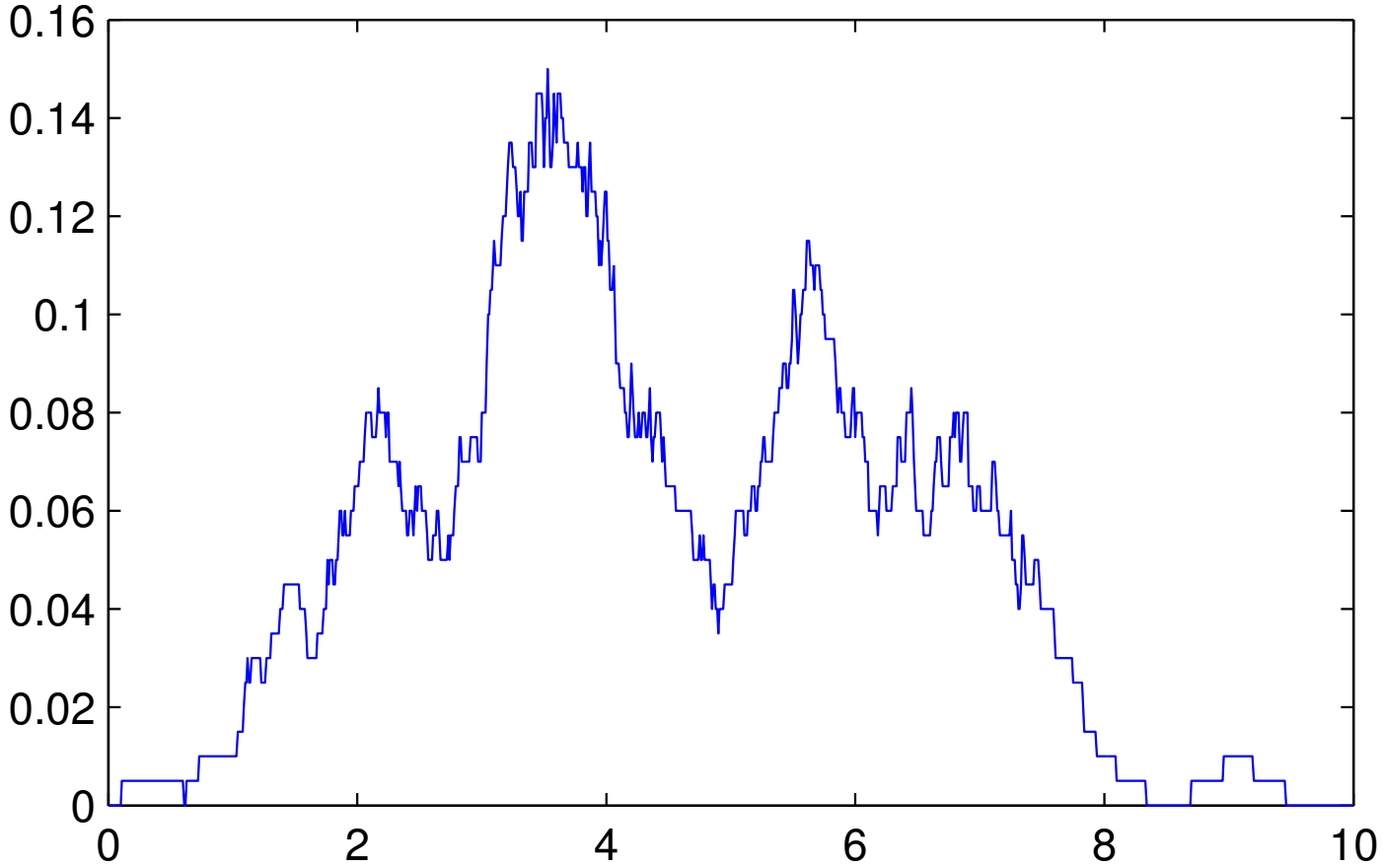
Kerndichteschätzung: uniformer Kern

- Bandbreite h=1



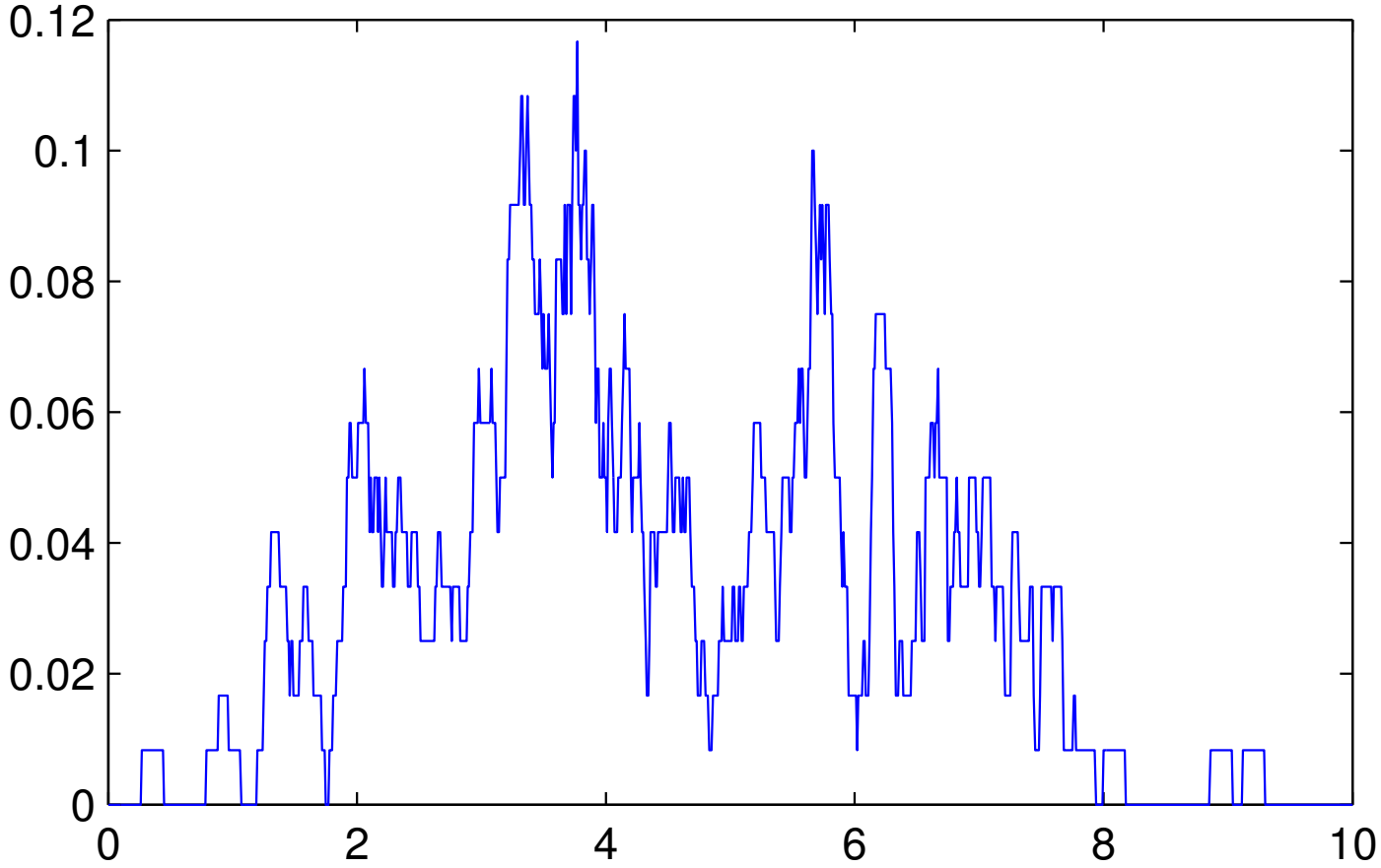
Kerndichteschätzung: uniformer Kern

- Bandbreite h=0.5



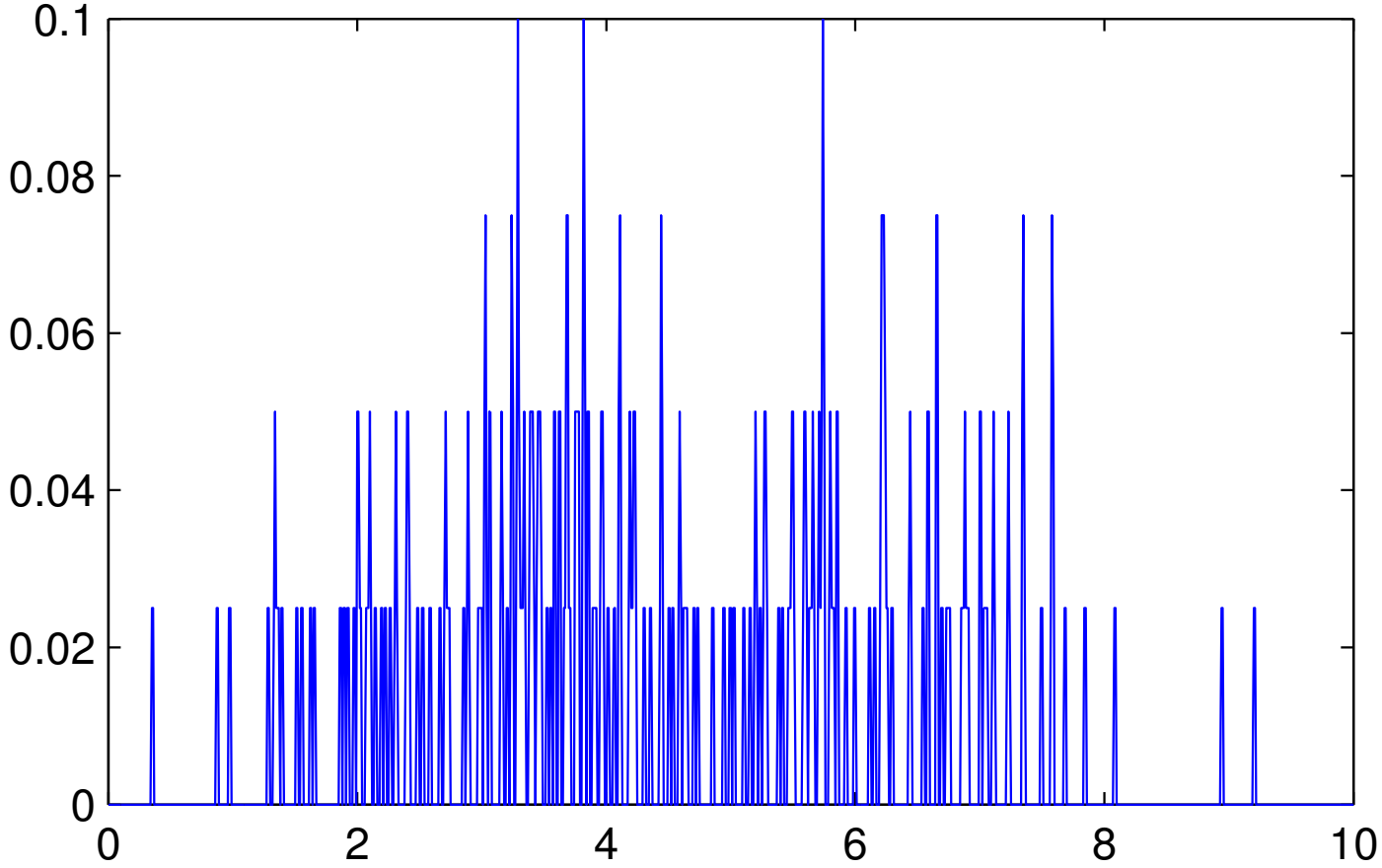
Kerndichteschätzung: uniformer Kern

- Bandbreite h=0.3



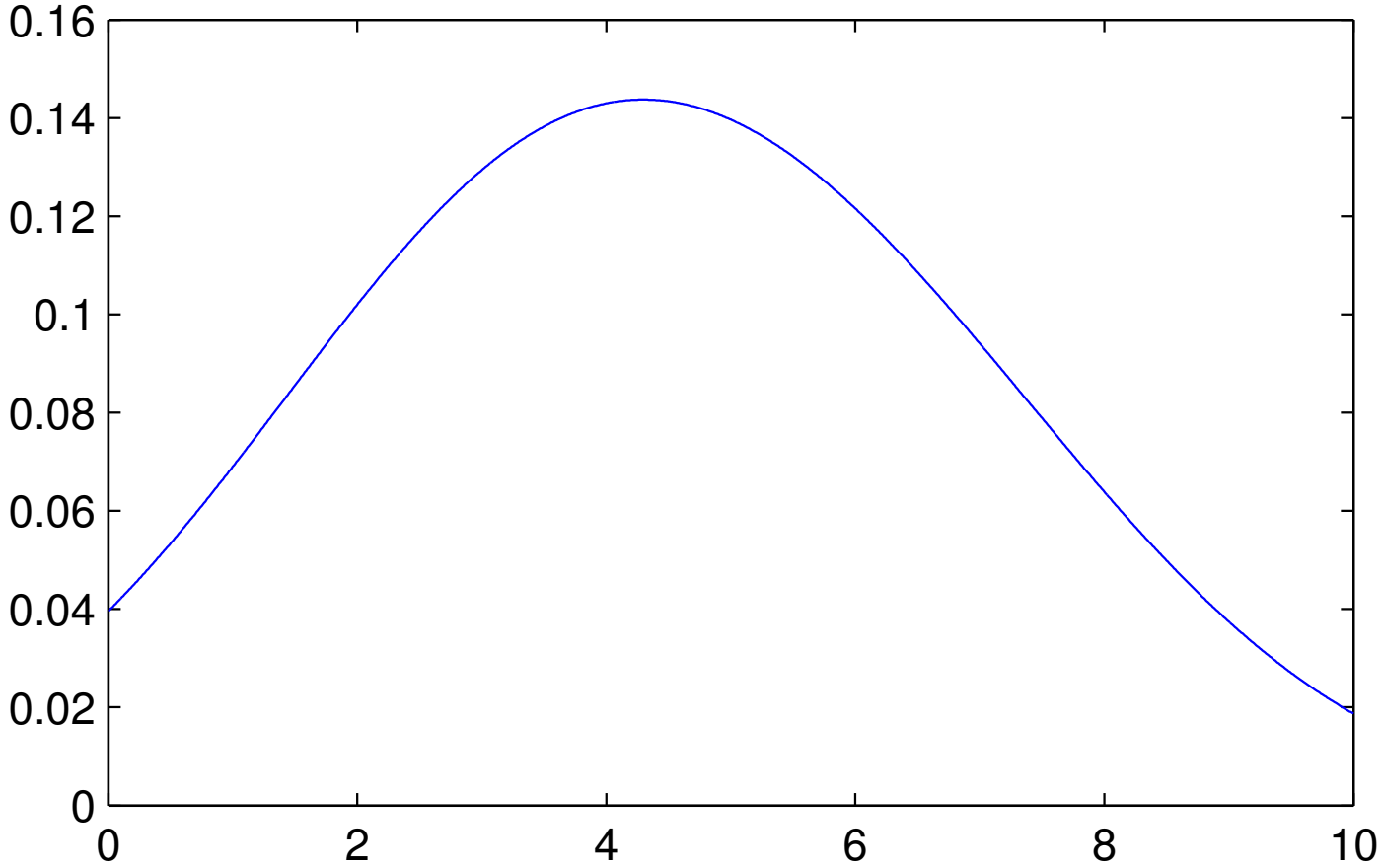
Kerndichteschätzung: uniformer Kern

- Bandbreite h=0.1



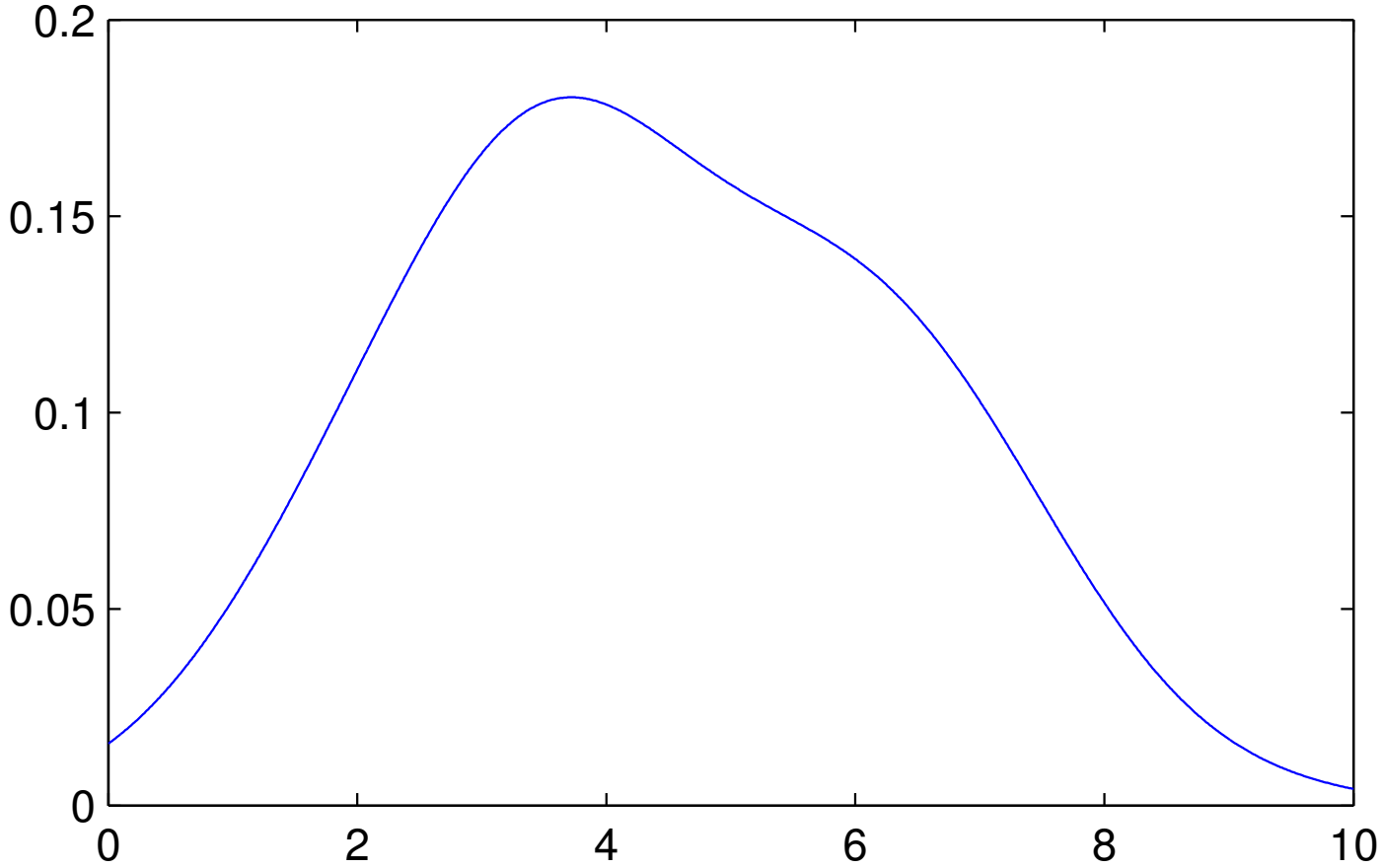
Kerndichteschätzung: Gausskern

- Bandbreite h=2



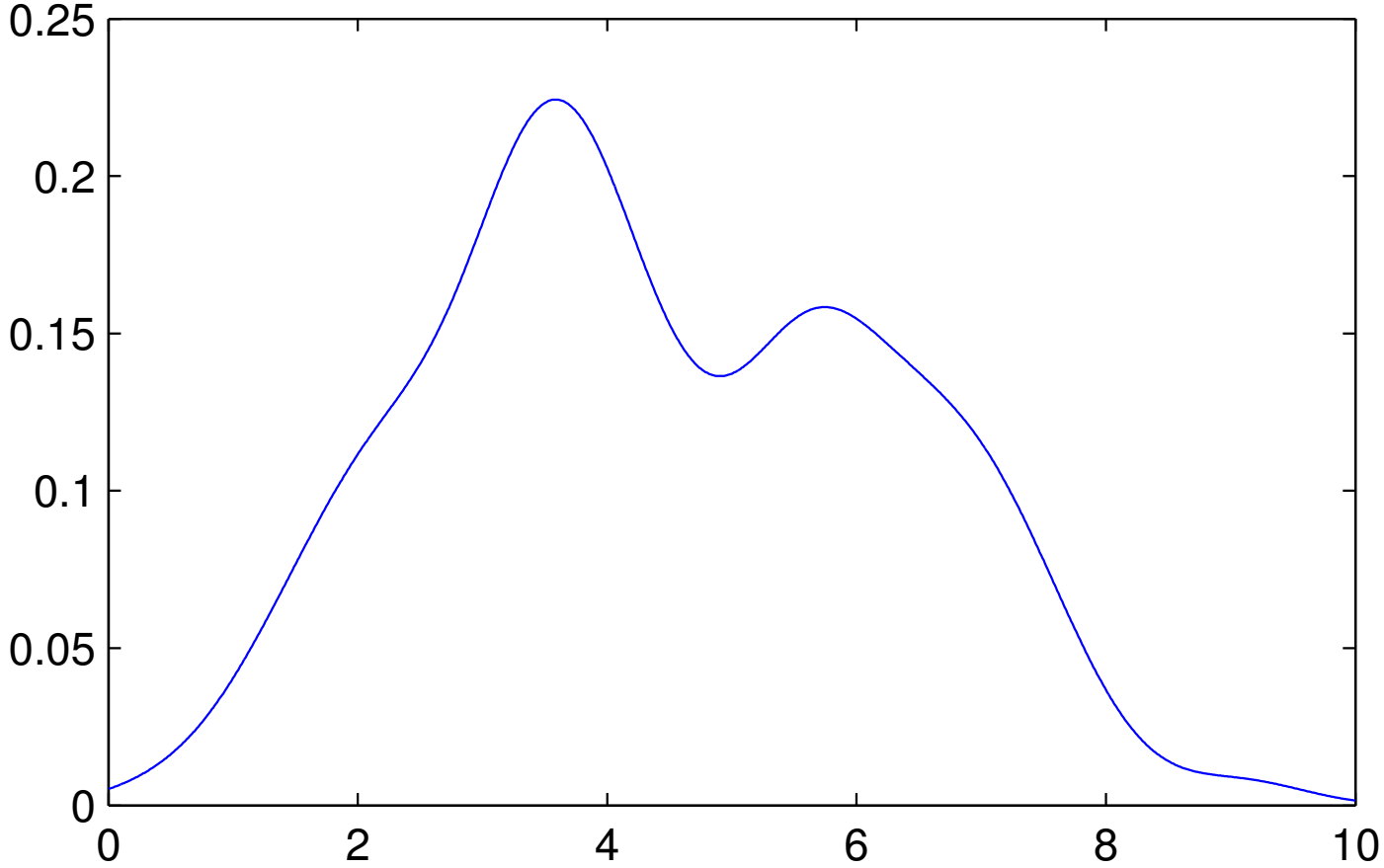
Kerndichteschätzung: Gausskern

- Bandbreite h=1



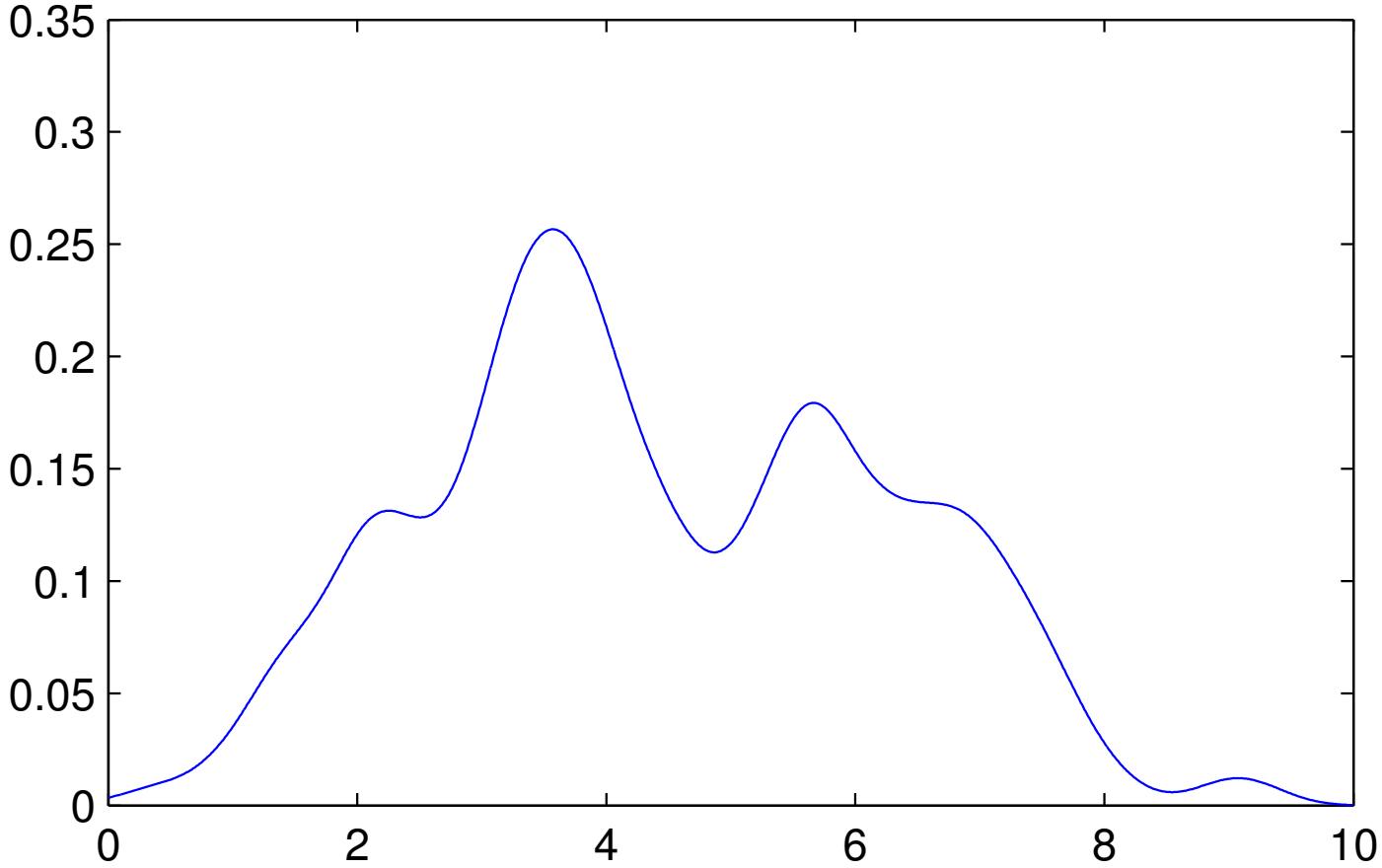
Kerndichteschätzung: Gausskern

- Bandbreite h=0.5



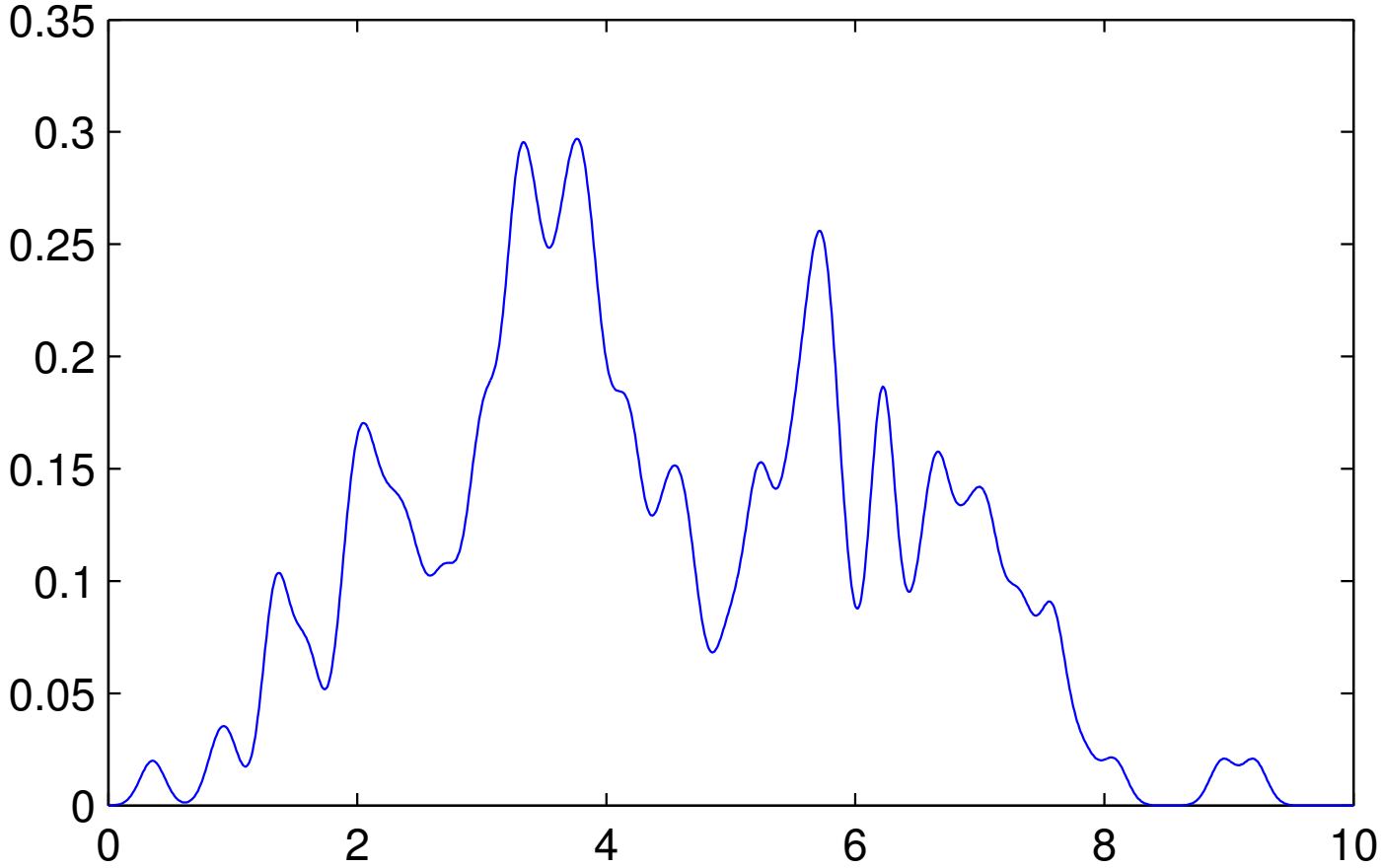
Kerndichteschätzung: Gausskern

- Bandbreite h=0.3



Kerndichteschätzung: Gausskern

- Bandbreite h=0.1



Kerndichteschätzung

- Wie bestimmt man die Bandbreite ?
- Sehr viele Methoden + Kriterien
- Für uns interessant: Likelihood-Crossvalidation
- Log-Likelihood der Daten ist definiert als

$$L = \log \prod_i p(x_i) = \sum_i \log p(x_i) \tag{4}$$

- Würde mit kleinen Bandbreiten unbegrenzt wachsen
- Verwende Leave-One-Out Kreuzvalidierung

$$L_{cv} = \sum_i \log p_{-i}(x_i) \quad , \quad p_{-i}(x) = \frac{1}{N-1} \sum_{k \neq i} K(x - x_k) \tag{5}$$

Kerndichteschätzung

- Wie bestimmt man die Bandbreite ?
- Sehr viele Methoden + Kriterien
- Für uns interessant: Likelihood-Crossvalidation
- Log-Likelihood der Daten ist definiert als

$$L = \log \prod_i p(x_i) = \sum_i \log p(x_i) \tag{4}$$

- Würde mit kleinen Bandbreiten unbegrenzt wachsen
- Verwende Leave-One-Out Kreuzvalidierung

$$L_{cv} = \sum_i \log p_{-i}(x_i) \quad , \quad p_{-i}(x) = \frac{1}{N-1} \sum_{k \neq i} K(x - x_k) \tag{5}$$

Kerndichteschätzung

- Wie bestimmt man die Bandbreite ?
- Sehr viele Methoden + Kriterien
- Für uns interessant: Likelihood-Crossvalidation
- Log-Likelihood der Daten ist definiert als

$$L = \log \prod_i p(x_i) = \sum_i \log p(x_i) \tag{4}$$

- Würde mit kleinen Bandbreiten unbegrenzt wachsen
- Verwende Leave-One-Out Kreuzvalidierung

$$L_{cv} = \sum_i \log p_{-i}(x_i) \quad , \quad p_{-i}(x) = \frac{1}{N-1} \sum_{k \neq i} K(x - x_k) \tag{5}$$

Kerndichteschätzung

- Wie bestimmt man die Bandbreite ?
- Sehr viele Methoden + Kriterien
- Für uns interessant: Likelihood-Crossvalidation
- Log-Likelihood der Daten ist definiert als

$$L = \log \prod_i p(x_i) = \sum_i \log p(x_i) \tag{4}$$

- Würde mit kleinen Bandbreiten unbegrenzt wachsen
- Verwende Leave-One-Out Kreuzvalidierung

$$L_{cv} = \sum_i \log p_{-i}(x_i) \quad , \quad p_{-i}(x) = \frac{1}{N-1} \sum_{k \neq i} K(x - x_k) \tag{5}$$

Kerndichteschätzung

- Wie bestimmt man die Bandbreite ?
- Sehr viele Methoden + Kriterien
- Für uns interessant: Likelihood-Crossvalidation
- Log-Likelihood der Daten ist definiert als

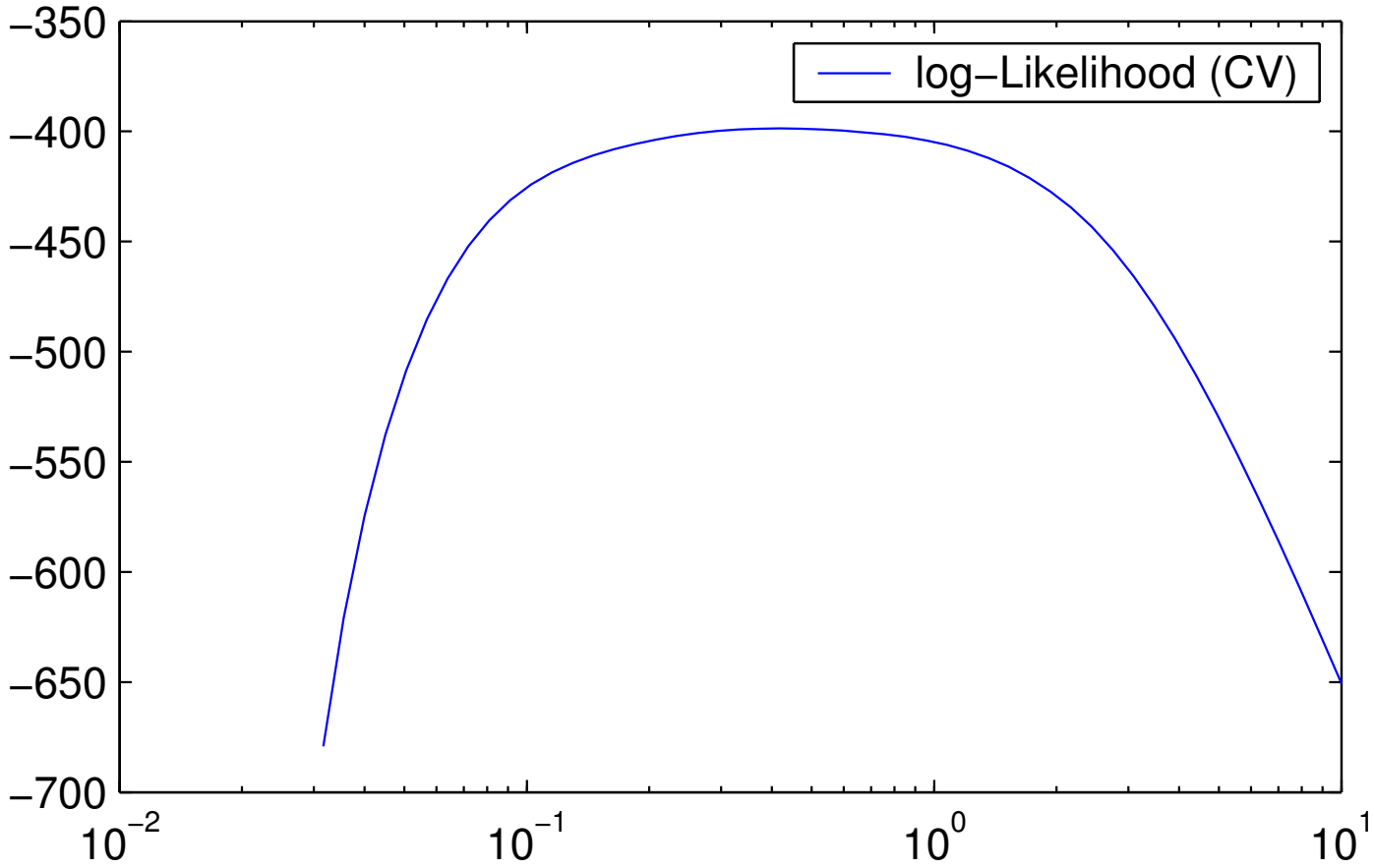
$$L = \log \prod_i p(x_i) = \sum_i \log p(x_i) \tag{4}$$

- Würde mit kleinen Bandbreiten unbegrenzt wachsen
- Verwende Leave-One-Out Kreuzvalidierung

$$L_{cv} = \sum_i \log p_{-i}(x_i) \quad , \quad p_{-i}(x) = \frac{1}{N-1} \sum_{k \neq i} K(x - x_k) \tag{5}$$

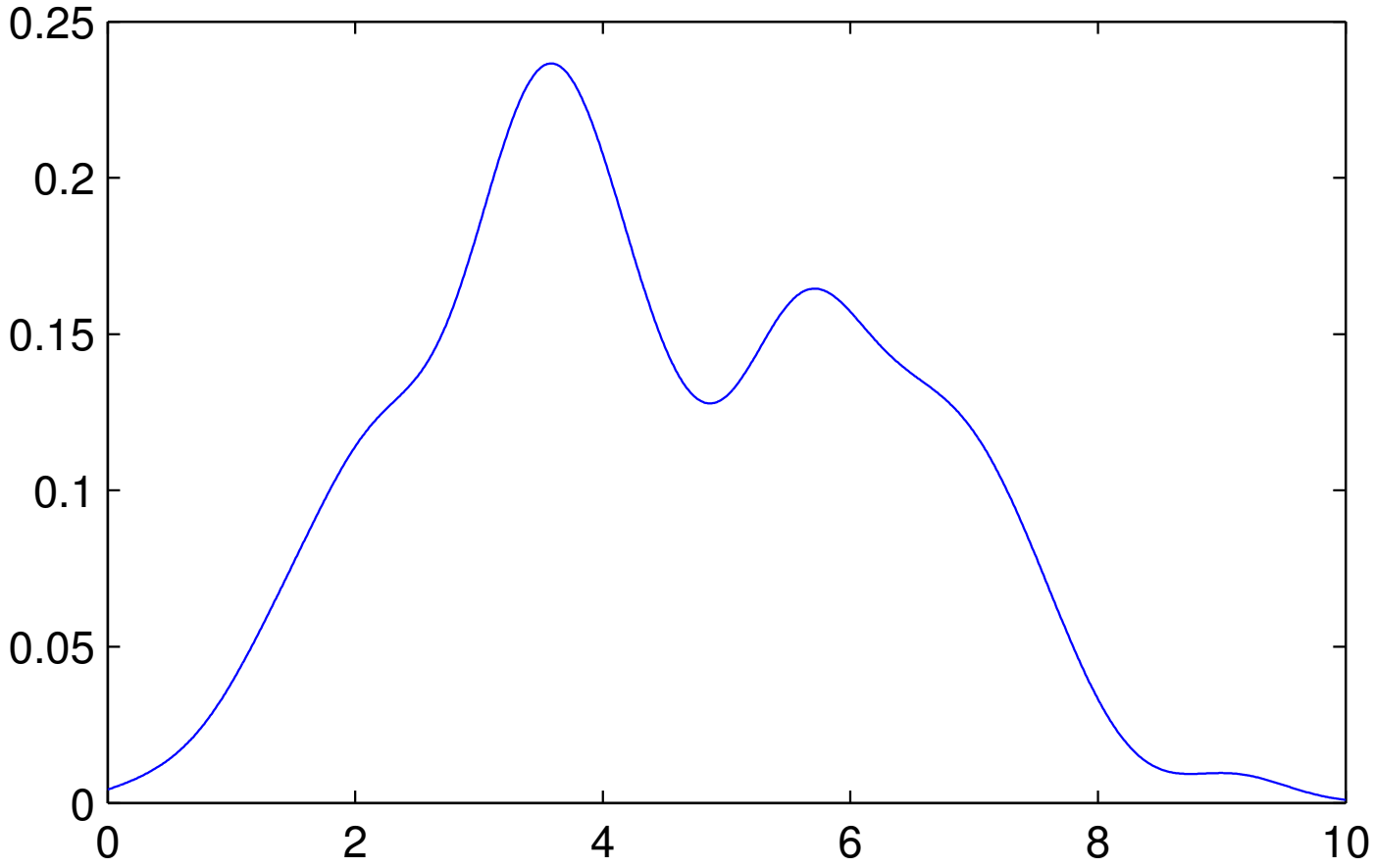
Kerndichteschätzung

- kreuzvalidierte Likelihood als Funktion der Bandbreite



Kerndichteschätzung

- Maximum bei h=0.4192 (Gausskern)



Kernregression

- Gegeben: Ein Satz von Ein-/Ausgabedaten (x_i, y_i), $i = 1 \ldots N$
- Gesucht: Eine glatte Funktion $y = f(x)$.
- Fasse die Daten x_i und y_i als Realisationen von Zufallsvariablen X und Y auf. Dann ist die Regressionsfunktion gegeben durch den bedingten Erwartungswert

$$f(x) = E(Y|X = x)$$

- Kennt man die gemeinsame Verteilung $p(x, y)$ der beiden Zufallsvariablen, so kann man $f(x)$ angeben als

$$f(x) = \int y p(y|x) dy \tag{6}$$

$$= \int y \frac{p(x, y)}{p(x)} dy \tag{7}$$

$$= \int y \frac{p(x, y)}{\int p(x, y') dy'} dy \tag{8}$$

Kernregression

- Gegeben: Ein Satz von Ein-/Ausgabedaten (x_i, y_i), $i = 1 \ldots N$
- Gesucht: Eine glatte Funktion $y = f(x)$.
- Fasse die Daten x_i und y_i als Realisationen von Zufallsvariablen X und Y auf. Dann ist die Regressionsfunktion gegeben durch den bedingten Erwartungswert

$$f(x) = E(Y|X = x)$$

- Kennt man die gemeinsame Verteilung $p(x, y)$ der beiden Zufallsvariablen, so kann man $f(x)$ angeben als

$$f(x) = \int y p(y|x) dy \tag{6}$$

$$= \int y \frac{p(x, y)}{p(x)} dy \tag{7}$$

$$= \int y \frac{p(x, y)}{\int p(x, y') dy'} dy \tag{8}$$

Kernregression

- Gegeben: Ein Satz von Ein-/Ausgabedaten (x_i, y_i), $i = 1 \ldots N$
- Gesucht: Eine glatte Funktion $y = f(x)$.
- Fasse die Daten x_i und y_i als Realisationen von Zufallsvariablen X und Y auf. Dann ist die Regressionsfunktion gegeben durch den bedingten Erwartungswert

$$f(x) = E(Y|X = x)$$

- Kennt man die gemeinsame Verteilung $p(x, y)$ der beiden Zufallsvariablen, so kann man $f(x)$ angeben als

$$f(x) = \int y p(y|x) dy \tag{6}$$

$$= \int y \frac{p(x, y)}{p(x)} dy \tag{7}$$

$$= \int y \frac{p(x, y)}{\int p(x, y') dy'} dy \tag{8}$$

Kernregression

- Im Allgemeinen ist die Verteilung nicht bekannt, muss also geschätzt werden. Benutze Produkt aus zwei Kernen:

$$p(x, y) = \frac{1}{N} \sum_i K_X(x - x_i) K_Y(y - y_i) \tag{9}$$

- Man erhält damit durch Marginalisierung zusätzlich

$$p(x) = \int \frac{1}{N} \sum_i K_X(x - x_i) K_Y(y - y_i) dy \tag{10}$$

$$= \frac{1}{N} \sum_i K_X(x - x_i) \int K_Y(y - y_i) dy \tag{11}$$

$$= \frac{1}{N} \sum_i K_X(x - x_i) \tag{12}$$

Kernregression

- Im Allgemeinen ist die Verteilung nicht bekannt, muss also geschätzt werden. Benutze Produkt aus zwei Kernen:

$$p(x, y) = \frac{1}{N} \sum_i K_X(x - x_i) K_Y(y - y_i) \tag{9}$$

- Man erhält damit durch Marginalisierung zusätzlich

$$\begin{aligned} p(x) &= \int \frac{1}{N} \sum_i K_X(x - x_i) K_Y(y - y_i) dy && (10) \\ &= \frac{1}{N} \sum_i K_X(x - x_i) \int K_Y(y - y_i) dy && (11) \\ &= \frac{1}{N} \sum_i K_X(x - x_i) && (12) \end{aligned}$$

Kernregression

- Einsetzen in (7) ergibt

$$f(x) = \int y \frac{\frac{1}{N}\sum_i K_X(x-x_i)K_Y(y-y_i)}{\frac{1}{N}\sum_j K_X(x-x_j)} dy \quad (13)$$

$$= \sum_i \frac{K_X(x-x_i)}{\sum_j K_X(x-x_j)} \int y K_Y(y-y_i) dy \quad (14)$$

- Um das Integral zu lösen, substituiere $z = y - y_i$ und benutze

$$\int K(z)dz = 1 \quad , \quad \int zK(z) = 0$$

$$\int yK_Y(y-y_i)dy = \int (z+y_i)K_Y(z)dz = y_i \quad (15)$$

Kernregression

- Einsetzen in (7) ergibt

$$f(x) = \int y \frac{\frac{1}{N}\sum_i K_X(x-x_i)K_Y(y-y_i)}{\frac{1}{N}\sum_j K_X(x-x_j)} dy \quad (13)$$

$$= \sum_i \frac{K_X(x-x_i)}{\sum_j K_X(x-x_j)} \int y K_Y(y-y_i) dy \quad (14)$$

- Um das Integral zu lösen, substituiere $z = y - y_i$ und benutze

$$\int K(z)dz = 1 \quad , \quad \int zK(z) = 0$$

$$\int y K_Y(y-y_i)dy = \int (z+y_i)K_Y(z)dz = y_i \quad (15)$$

Kernregression

- Insgesamt erhält man den von Nadaraya (1964) und Watson (1964) vorgeschlagenen Ausdruck:

$$f(x) = \sum_i y_i \frac{K_X(x - x_i)}{\sum_j K_X(x - x_j)} \tag{16}$$

- $f(x)$ ist eine Konvexkombination der y_i (lokal gewichtetes Mittel)
- $f(x)$ hängt nicht mehr von $K_Y(\cdot)$ ab, d.h. wir haben nur noch einen Kern $K_X(x) = K(x)$.
- Einziger Parameter: Bandbreite des Kerns.

Kernregression

- Insgesamt erhält man den von Nadaraya (1964) und Watson (1964) vorgeschlagenen Ausdruck:

$$f(x) = \sum_i y_i \frac{K_X(x - x_i)}{\sum_j K_X(x - x_j)} \tag{16}$$

- $f(x)$ ist eine Konvexkombination der y_i (lokal gewichtetes Mittel)
- $f(x)$ hängt nicht mehr von $K_Y(\cdot)$ ab, d.h. wir haben nur noch einen Kern $K_X(x) = K(x)$.
- Einziger Parameter: Bandbreite des Kerns.

Kernregression

- Insgesamt erhält man den von Nadaraya (1964) und Watson (1964) vorgeschlagenen Ausdruck:

$$f(x) = \sum_i y_i \frac{K_X(x - x_i)}{\sum_j K_X(x - x_j)} \tag{16}$$

- $f(x)$ ist eine Konvexkombination der y_i
  (lokal gewichtetes Mittel)
- $f(x)$ hängt nicht mehr von $K_Y(\cdot)$ ab, d.h. wir haben nur noch einen Kern $K_X(x) = K(x)$.
- Einziger Parameter: Bandbreite des Kerns.

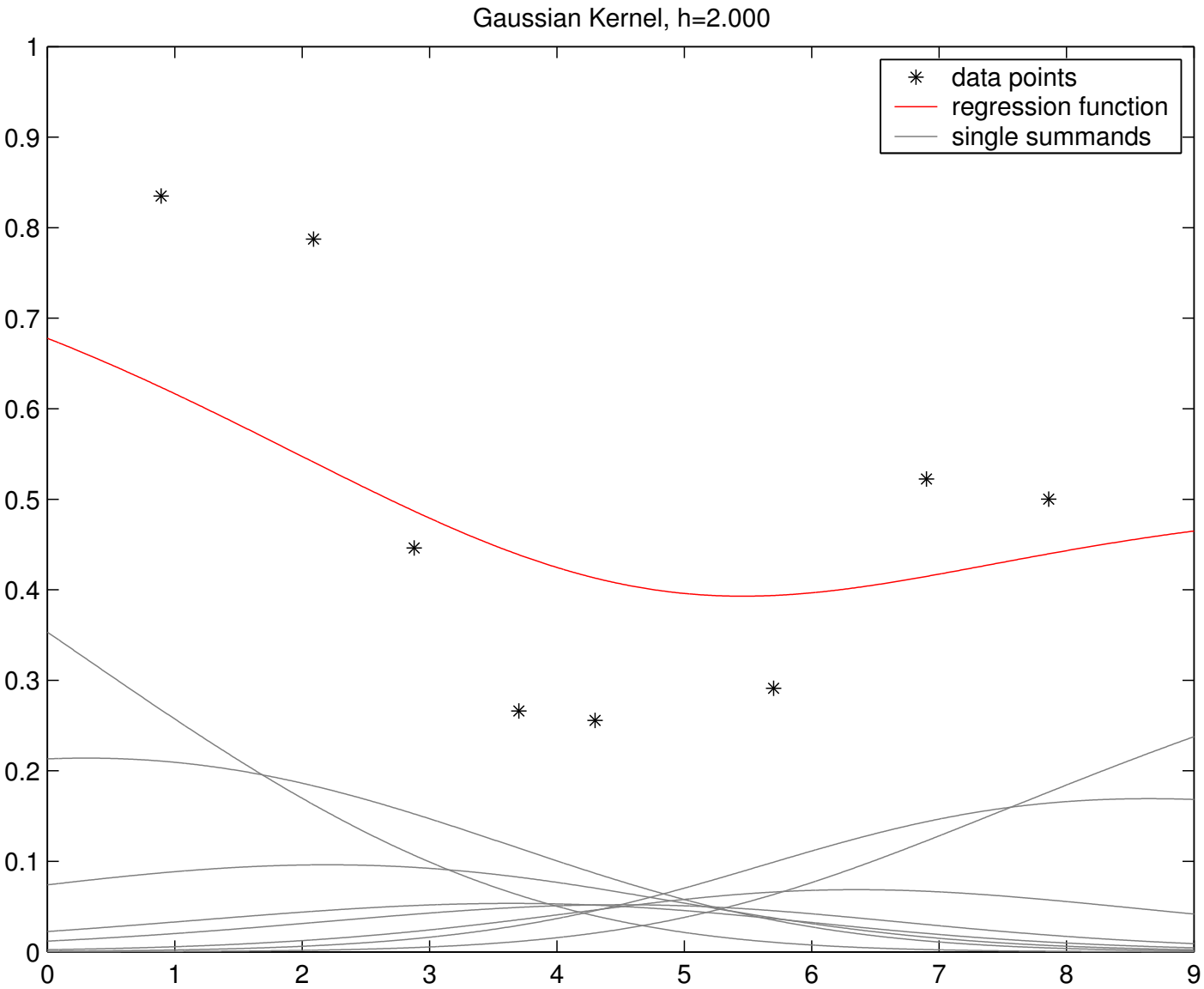
Kernregression

- Insgesamt erhält man den von Nadaraya (1964) und Watson (1964) vorgeschlagenen Ausdruck:

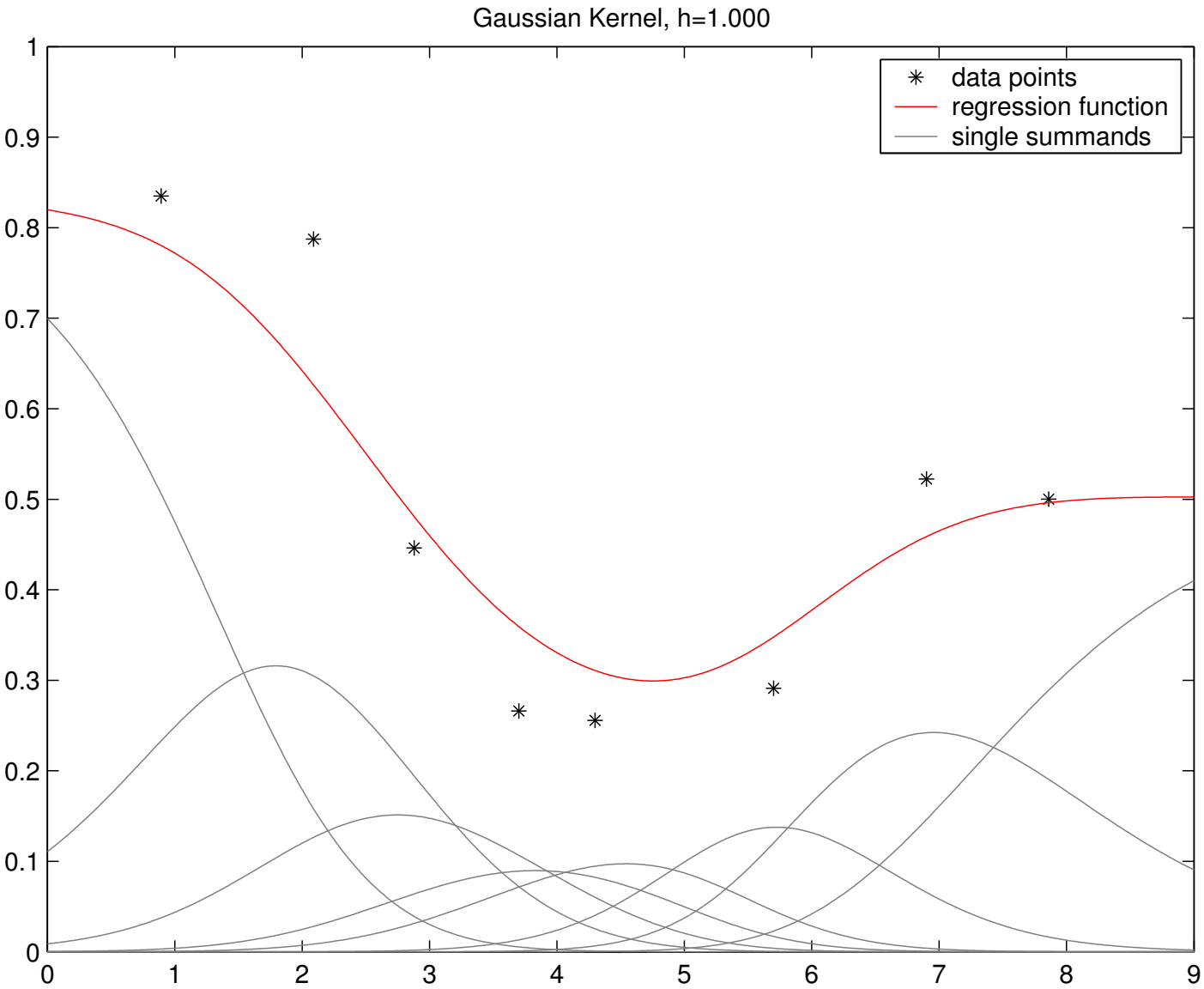
$$f(x) = \sum_i y_i \frac{K_X(x - x_i)}{\sum_j K_X(x - x_j)} \tag{16}$$

- $f(x)$ ist eine Konvexkombination der y_i
  (lokal gewichtetes Mittel)
- $f(x)$ hängt nicht mehr von $K_Y(\cdot)$ ab, d.h. wir haben nur noch einen Kern $K_X(x) = K(x)$.
- Einziger Parameter: Bandbreite des Kerns.

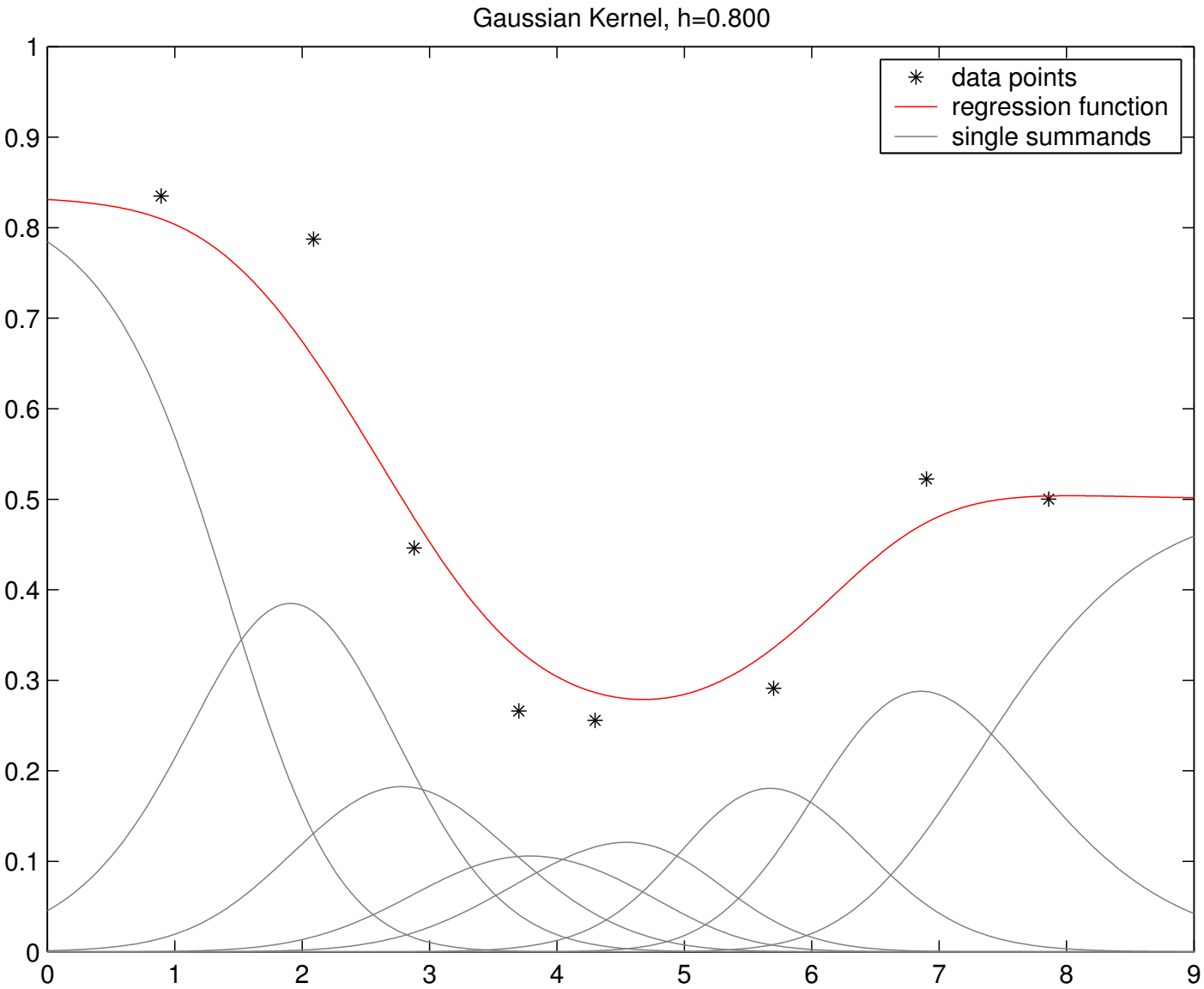
Kernregression



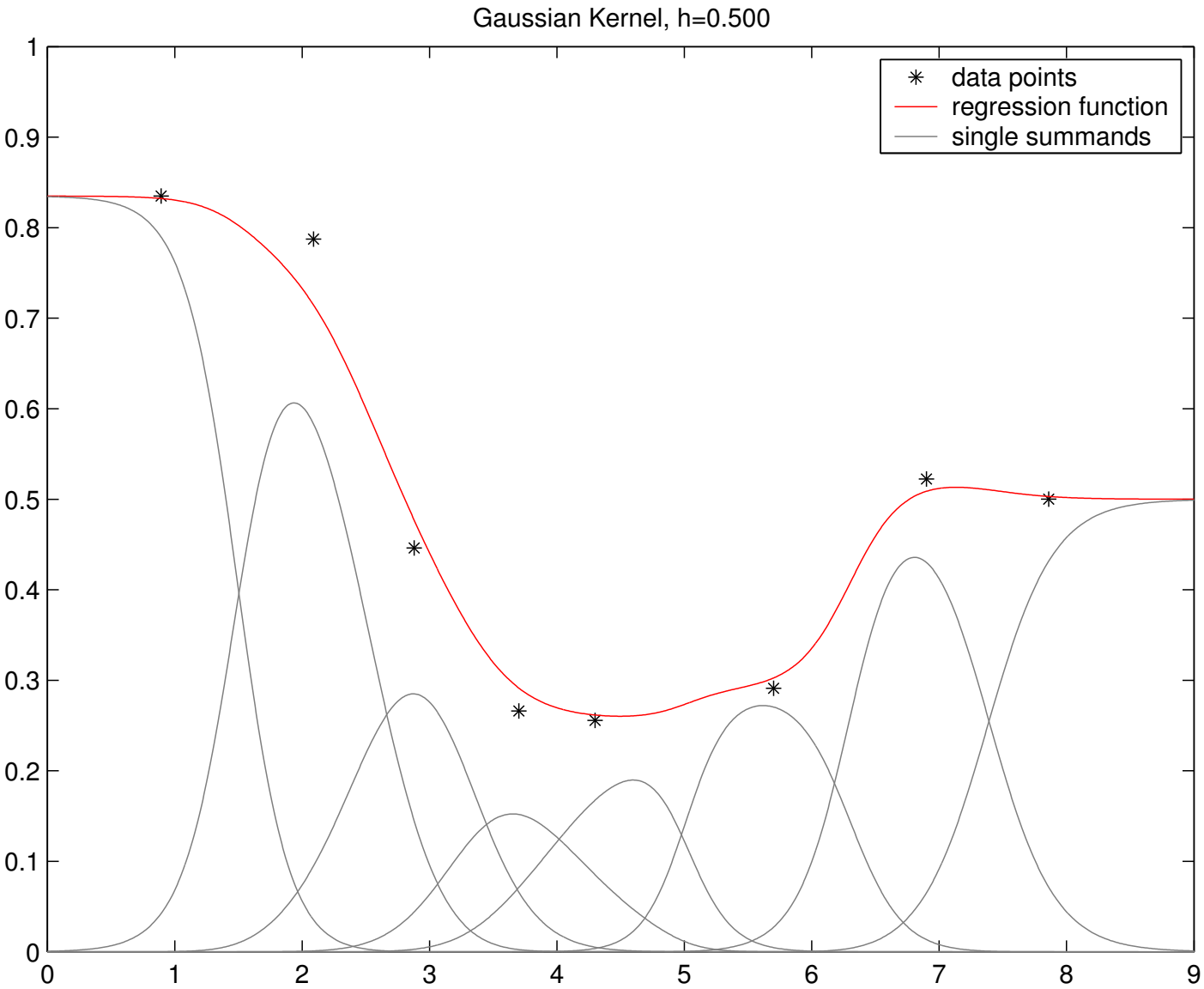
Kernregression



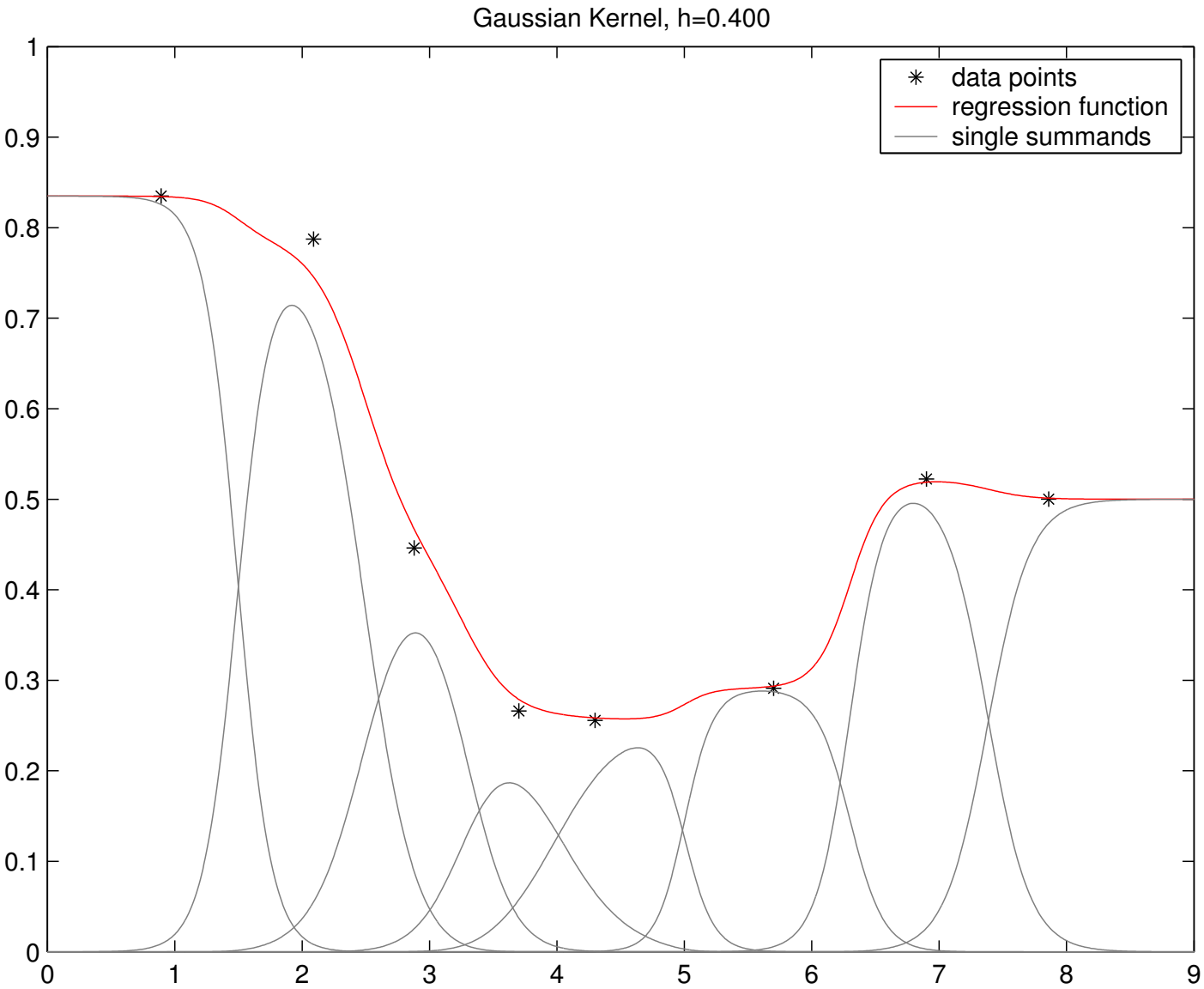
Kernregression



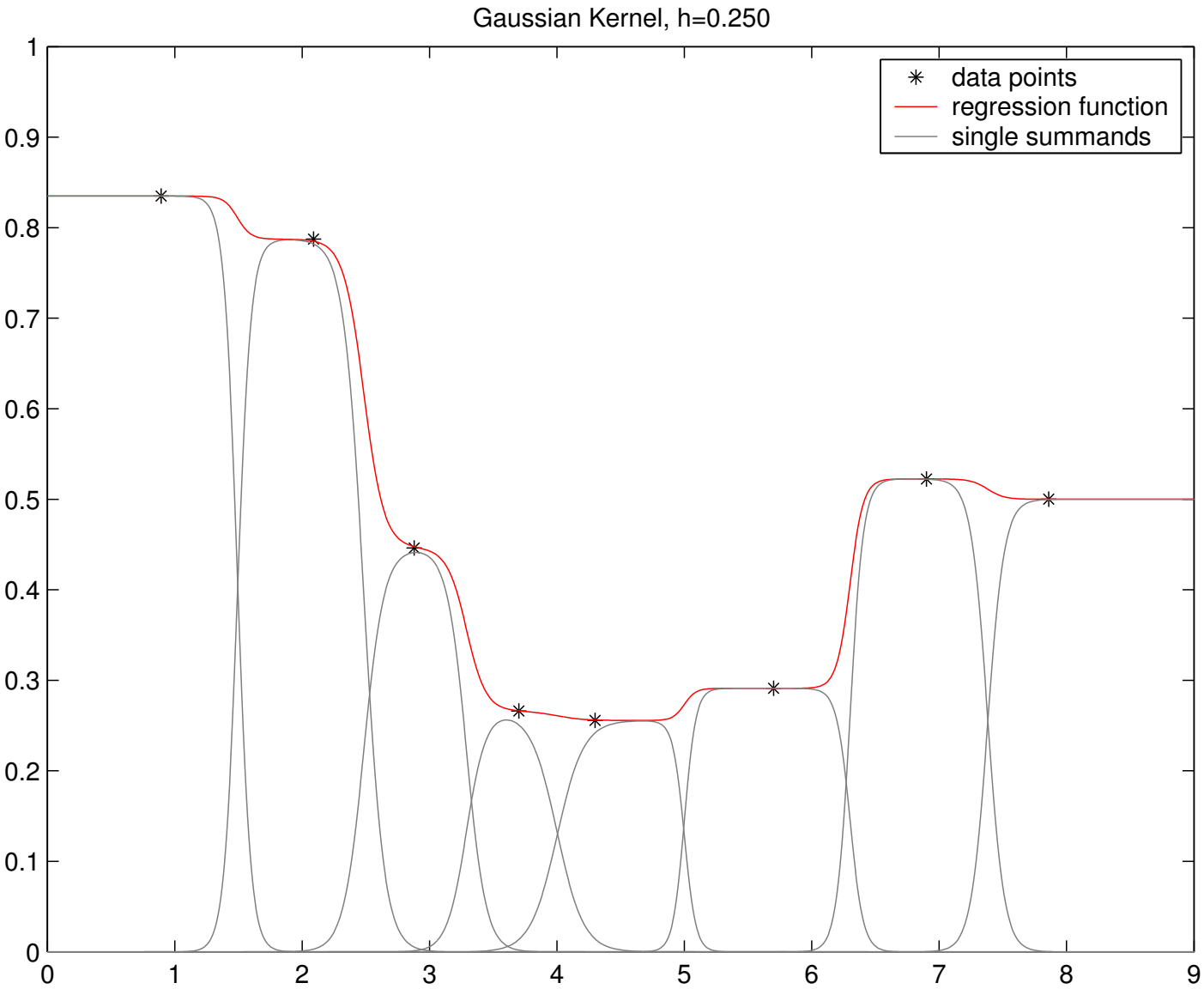
Kernregression



Kernregression



Kernregression

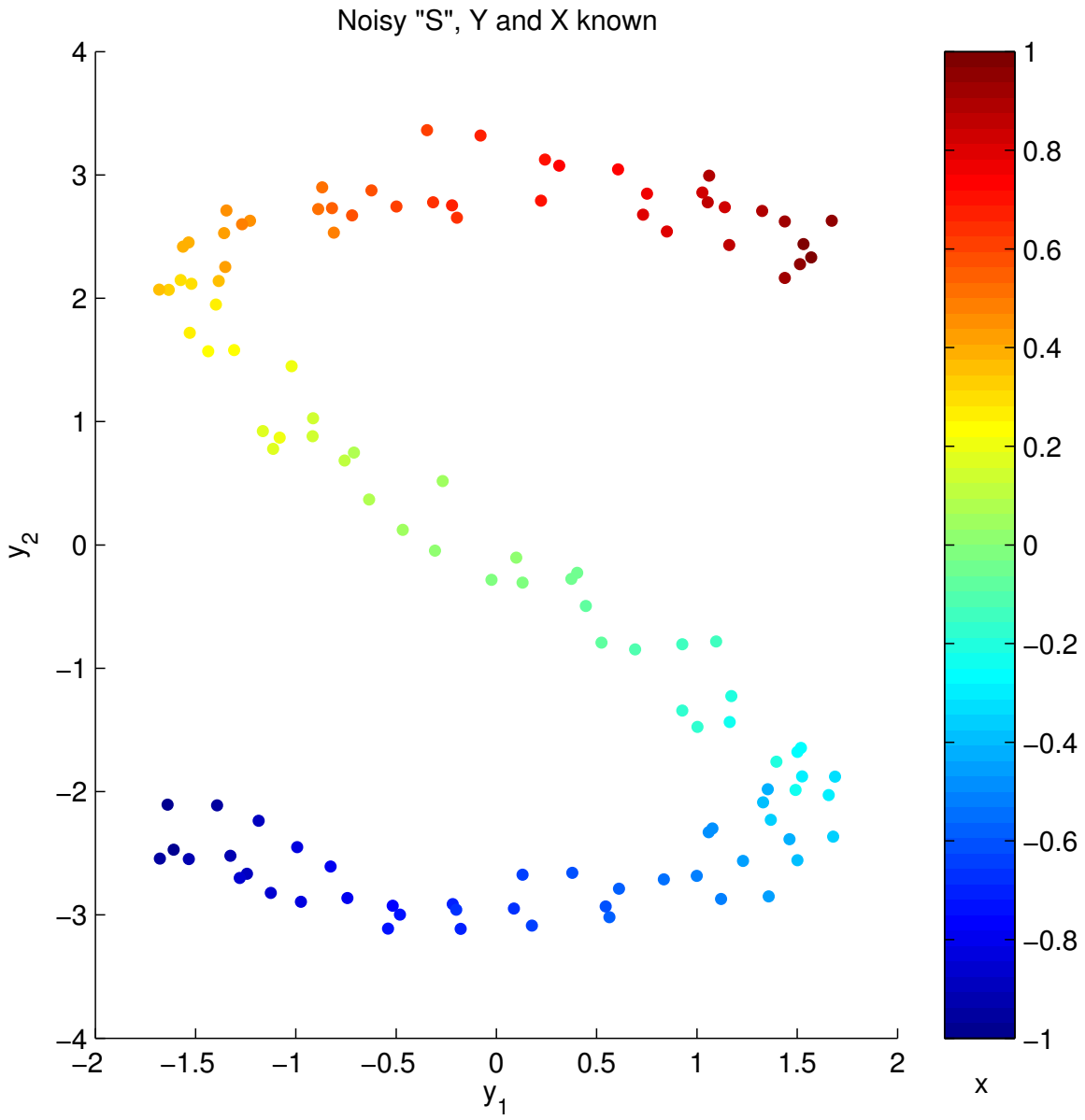


Kernregression

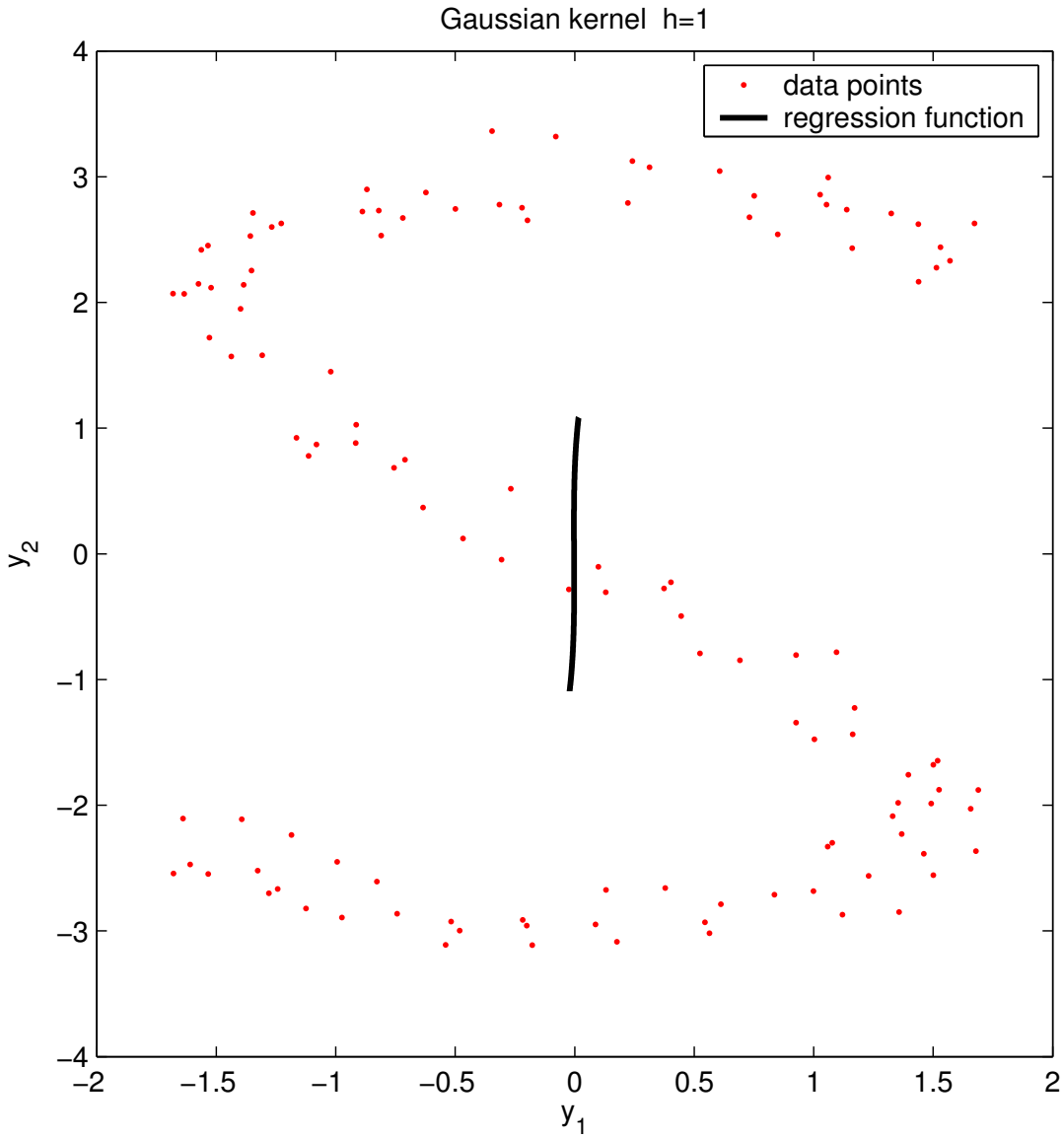
- Je größer die Bandbreite, desto glatter die Funktion

- Der Nadaraya-Watson-Schätzer funktioniert natürlich auch mehrdimensional (sowohl x als auch y).

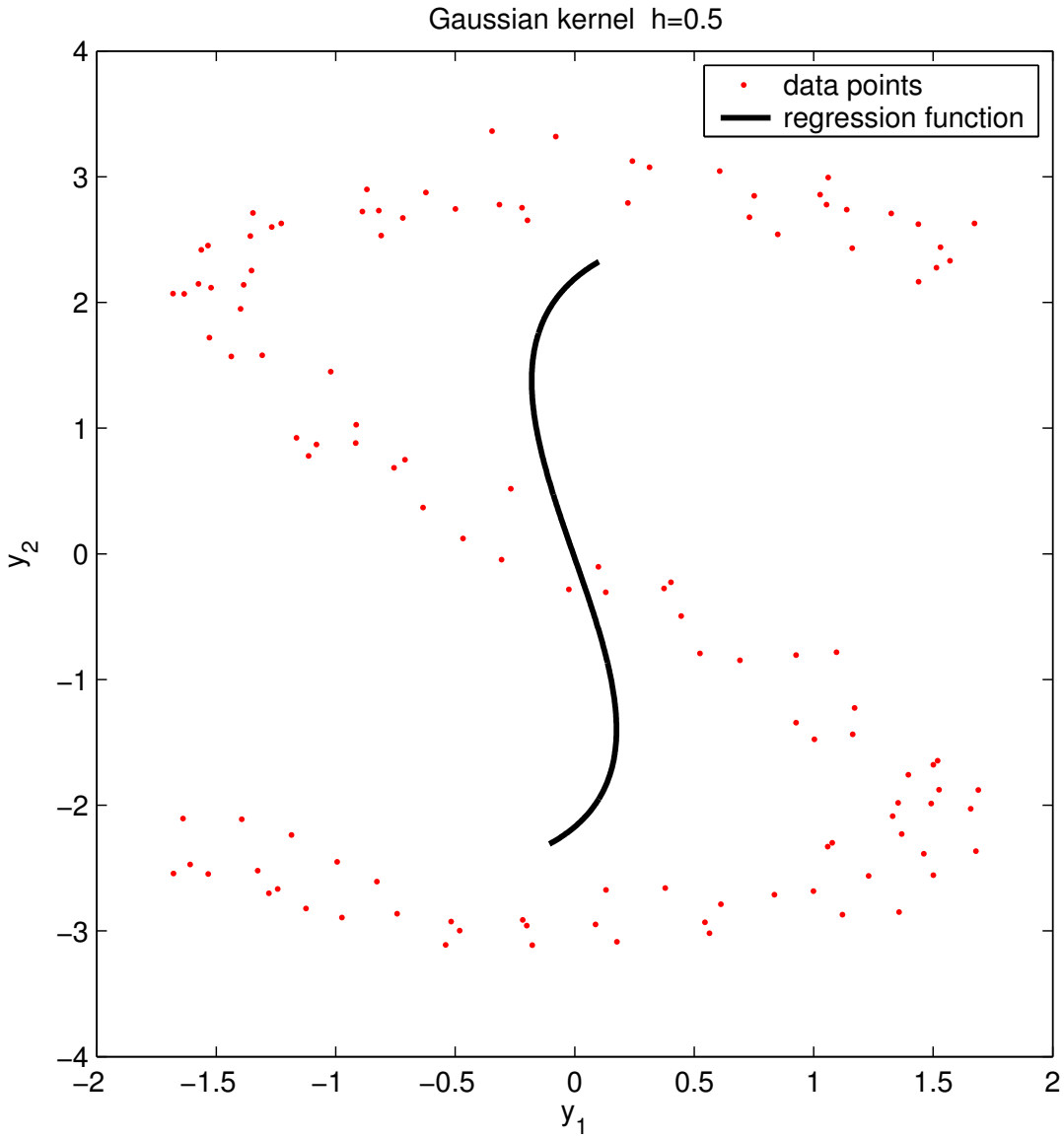
Kernregression 1D $\longrightarrow$ 2D



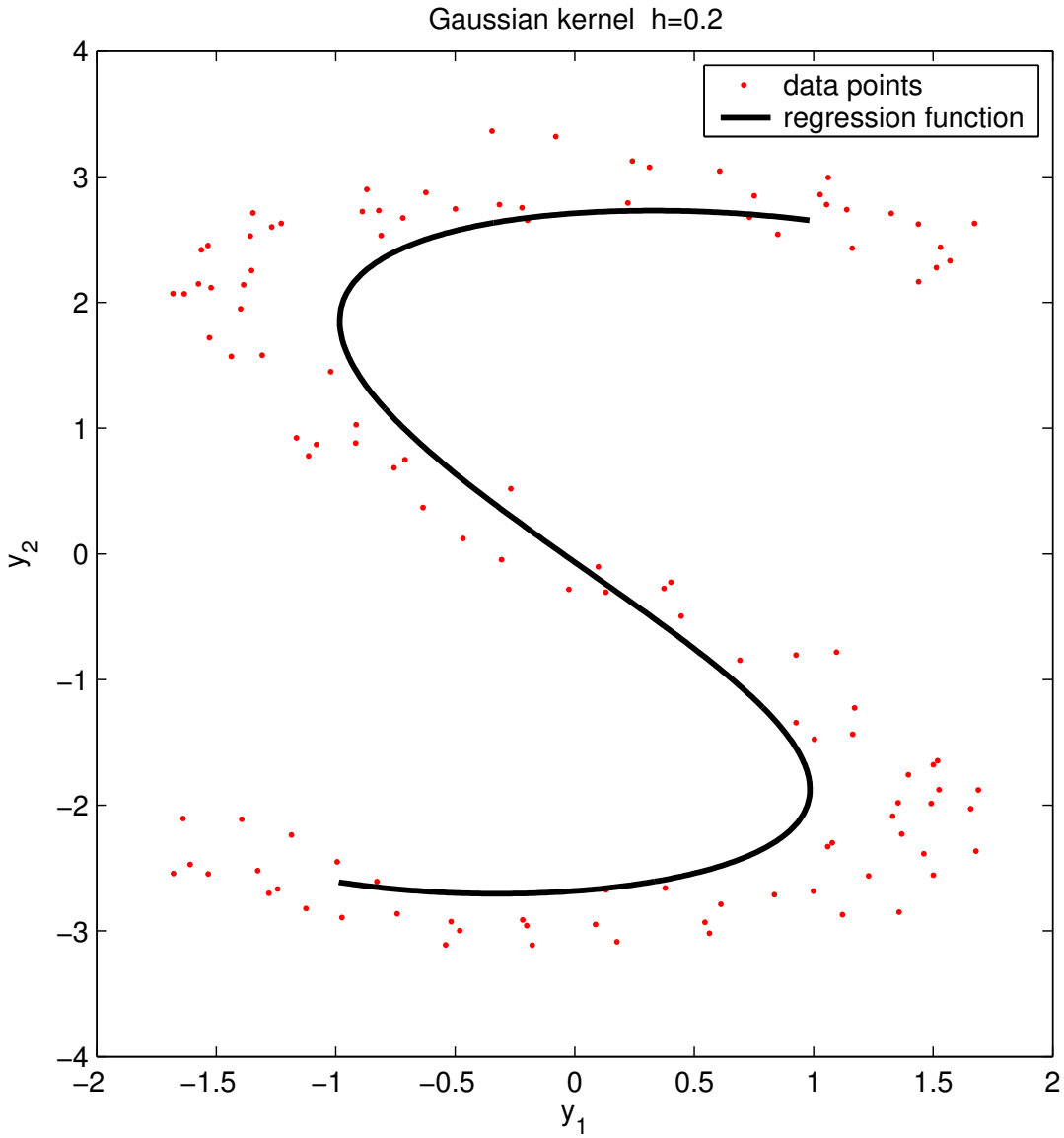
Kernregression



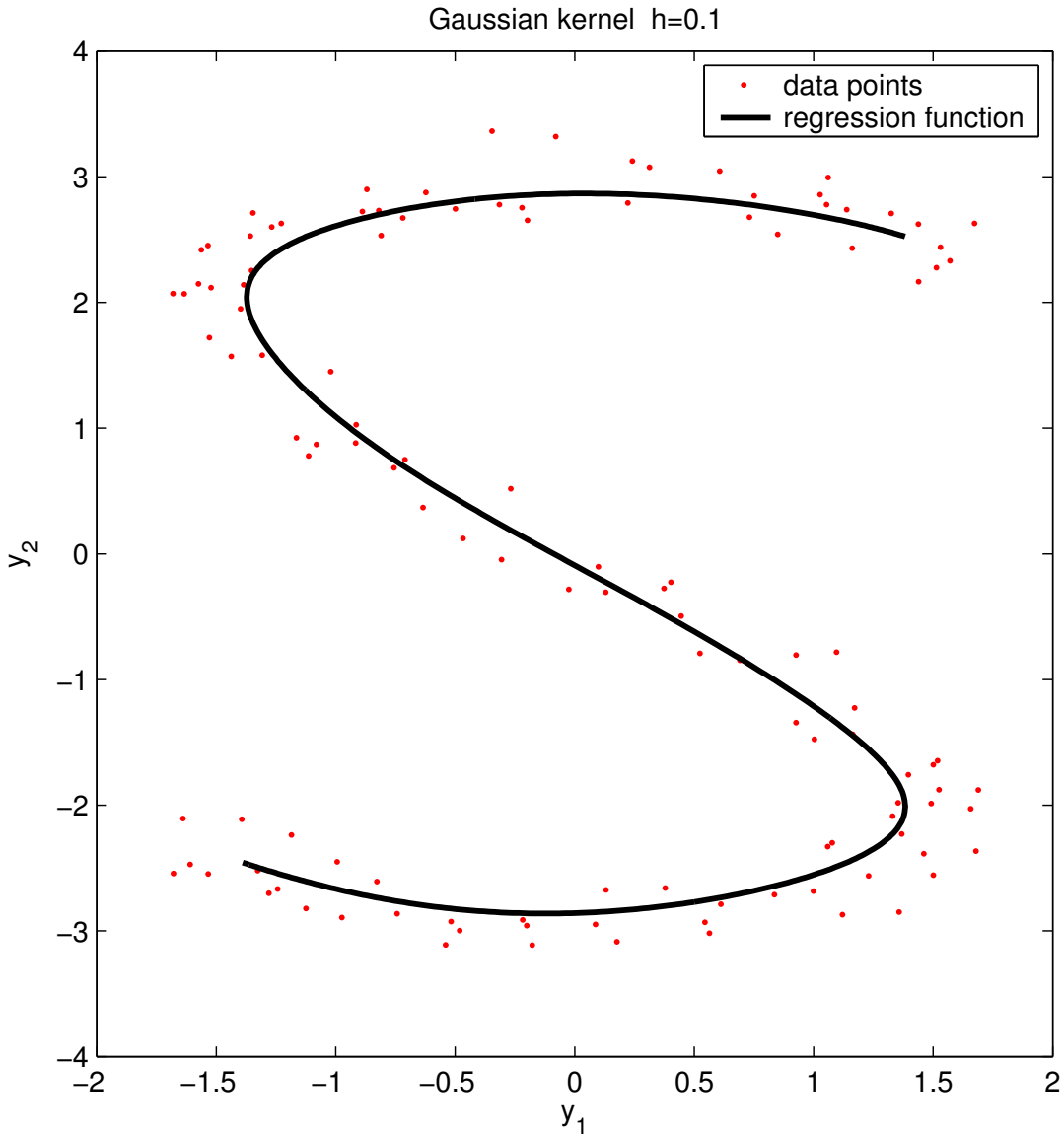
Kernregression



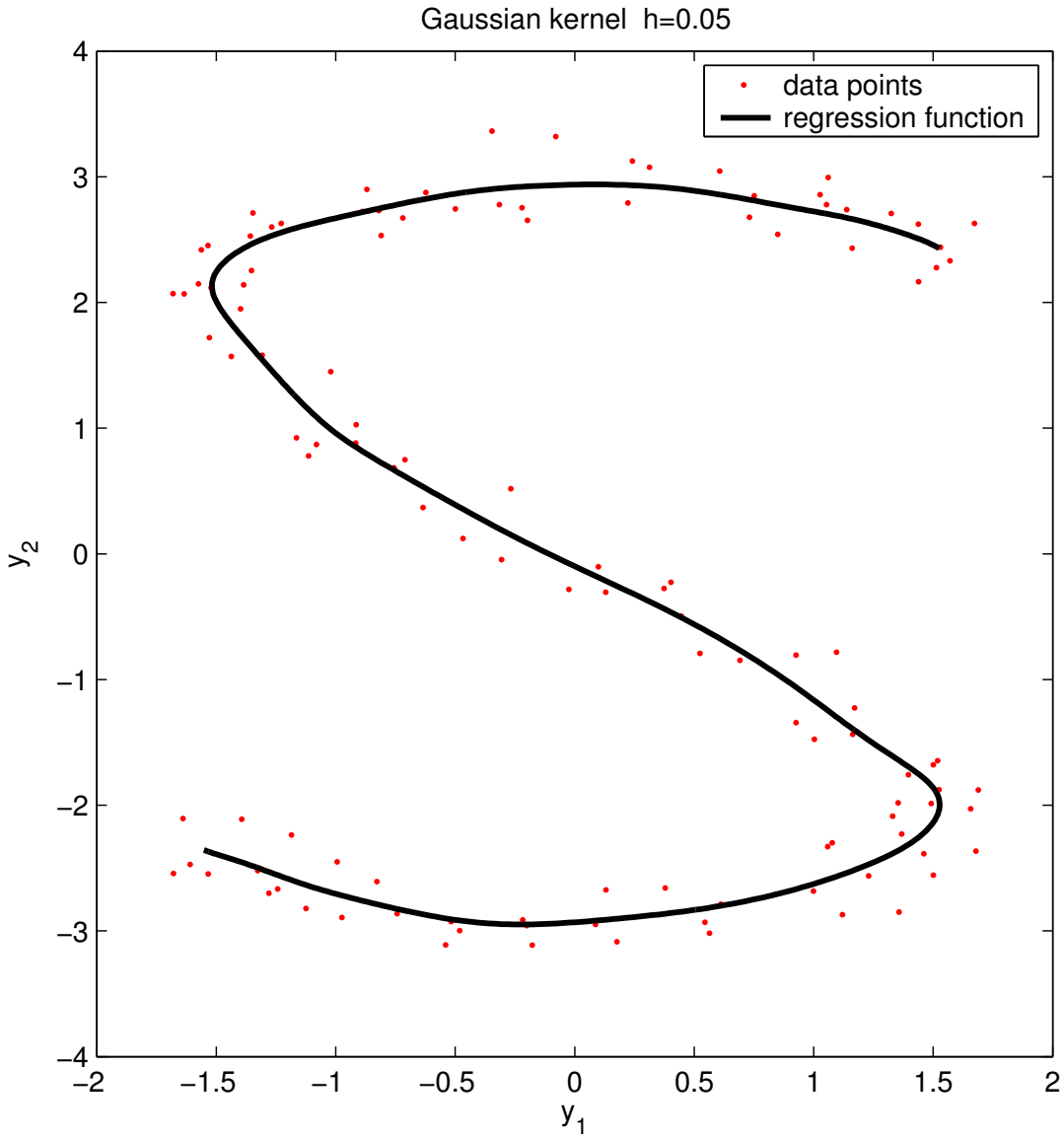
Kernregression



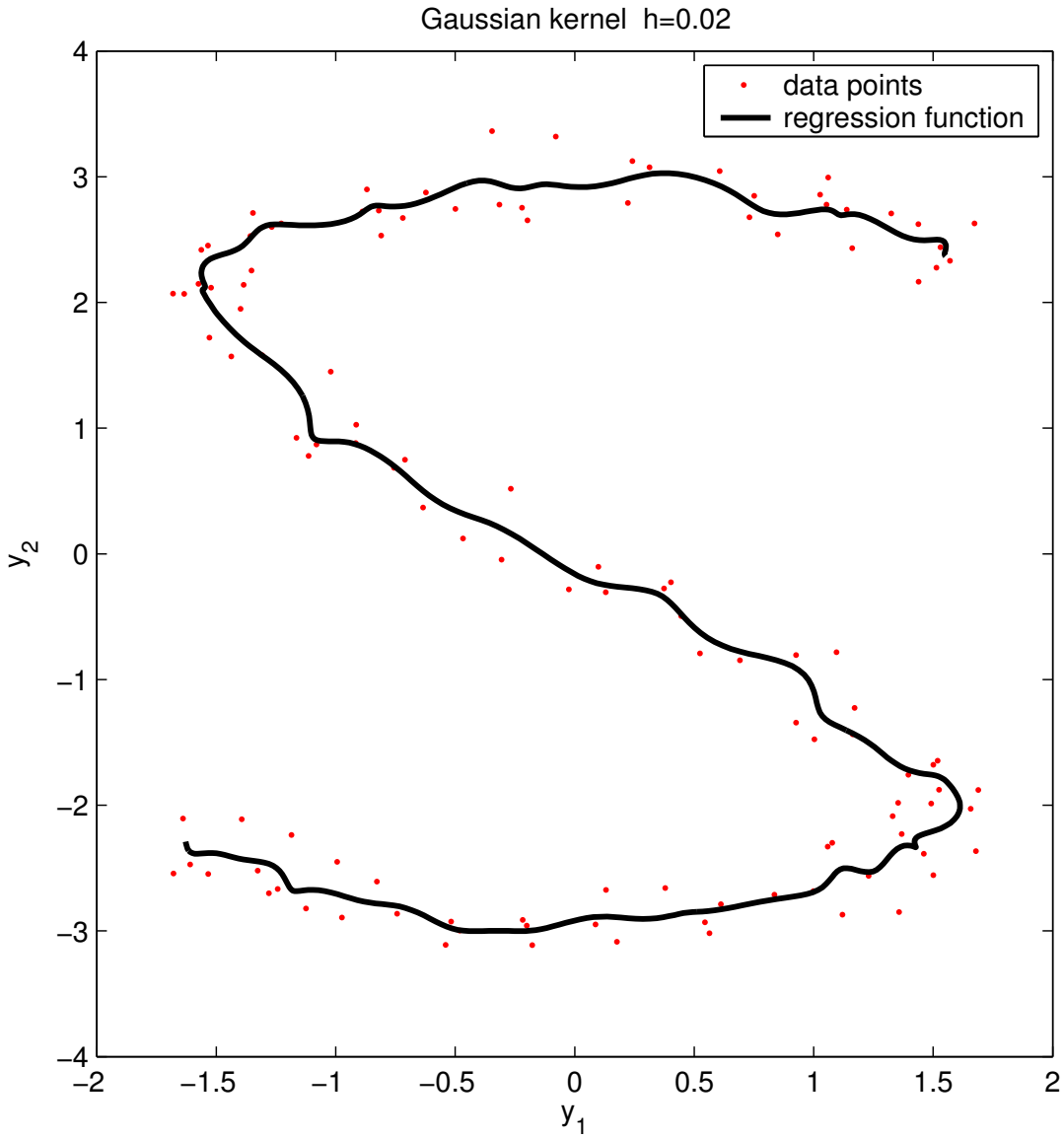
Kernregression



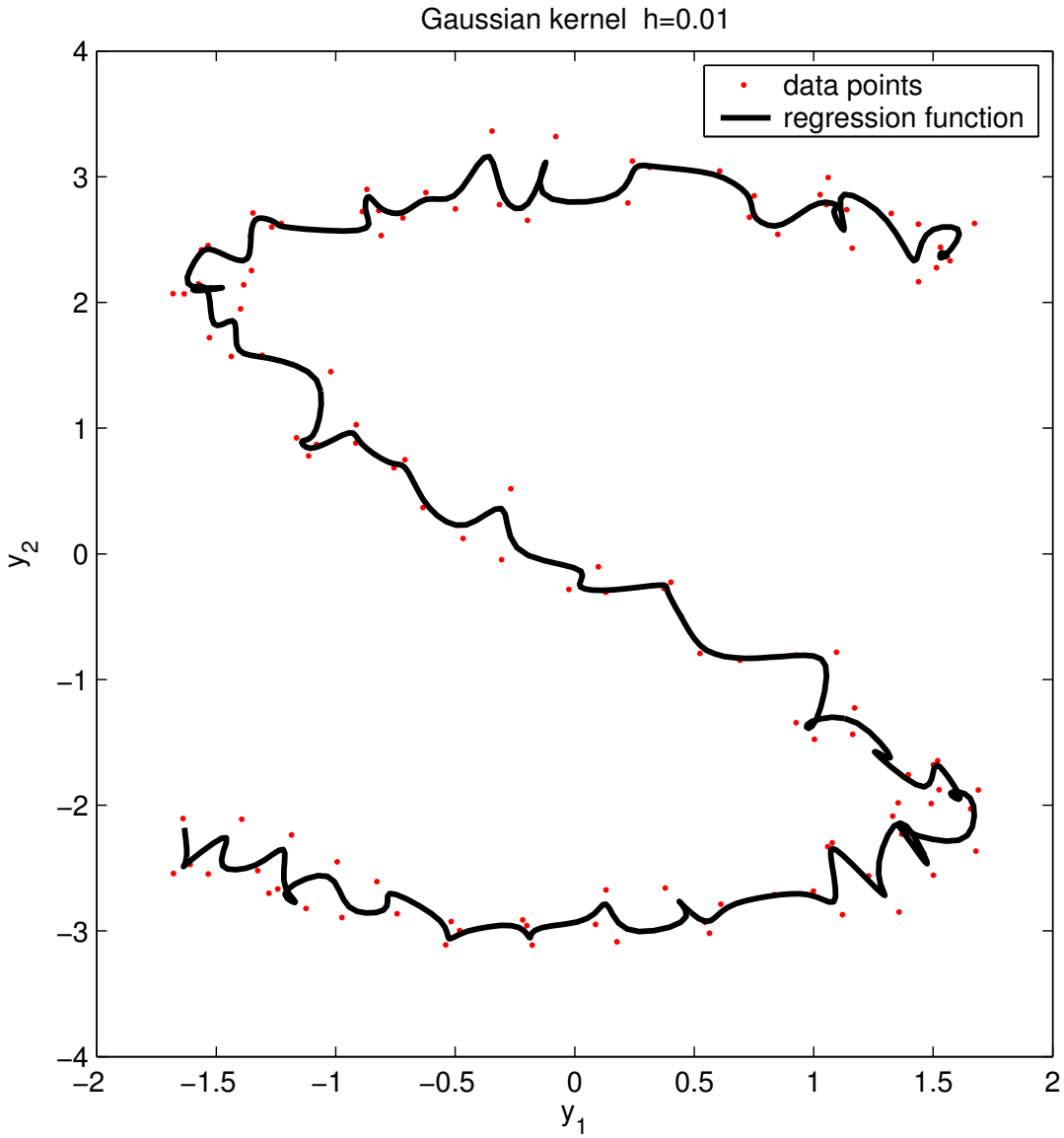
Kernregression



Kernregression



Kernregression



Kernregression

- Welche Bandbreite ist “richtig” ?
- Naives Fehlermaß: Mittlerer quadr. Abstand zw. $\mathbf{y}_i$ und $\mathbf{f}(\mathbf{x}_i)$, geht für kleine Bandbreite h gegen 0.
- Besser: Leave-One-Out Cross-Validation

$$E_{CV} = \frac{1}{N} \sum_i \|\mathbf{y}_i - \mathbf{f}_{-i}(\mathbf{x}_i)\|^2$$

$$\mathbf{f}_{-i}(\mathbf{x}) = \sum_{k \neq i} \mathbf{y}_k \frac{K(\mathbf{x} - \mathbf{x}_k)}{\sum_j K(\mathbf{x} - \mathbf{x}_j)}$$

Kernregression

- Welche Bandbreite ist “richtig” ?
- Naives Fehlermaß: Mittlerer quadr. Abstand zw. $\mathbf{y}_i$ und $\mathbf{f}(\mathbf{x}_i)$, geht für kleine Bandbreite h gegen 0.
- Besser: Leave-One-Out Cross-Validation

$$E_{CV} = \frac{1}{N} \sum_i \|\mathbf{y}_i - \mathbf{f}_{-i}(\mathbf{x}_i)\|^2$$

$$\mathbf{f}_{-i}(\mathbf{x}) = \sum_{k \neq i} \mathbf{y}_k \frac{K(\mathbf{x} - \mathbf{x}_k)}{\sum_j K(\mathbf{x} - \mathbf{x}_j)}$$

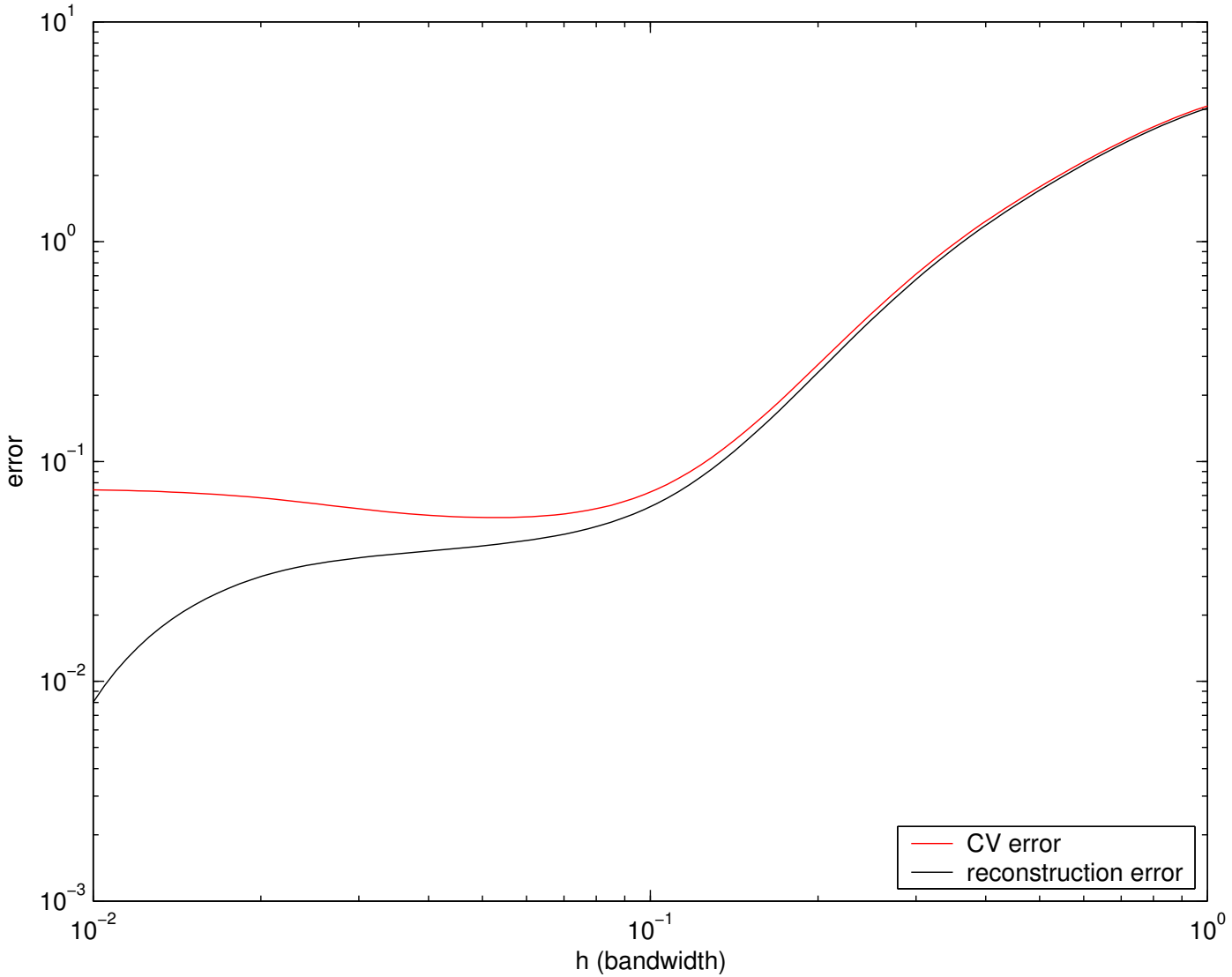
Kernregression

- Welche Bandbreite ist “richtig” ?
- Naives Fehlermaß: Mittlerer quadr. Abstand zw. $\mathbf{y}_i$ und $\mathbf{f}(\mathbf{x}_i)$, geht für kleine Bandbreite h gegen 0.
- Besser: Leave-One-Out Cross-Validation

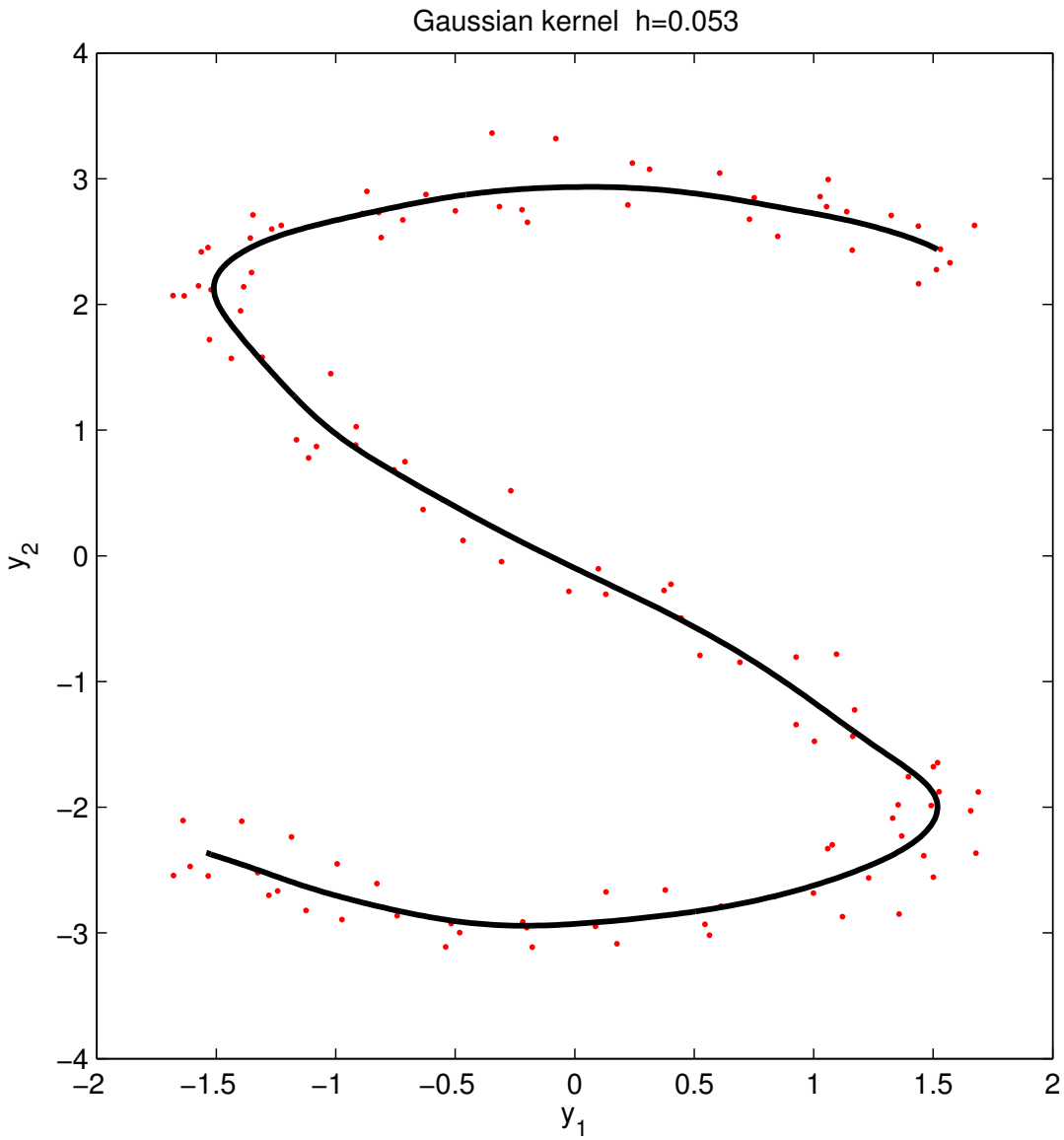
$$E_{CV} = \frac{1}{N} \sum_i \|\mathbf{y}_i - \mathbf{f}_{-i}(\mathbf{x}_i)\|^2$$

$$\mathbf{f}_{-i}(\mathbf{x}) = \sum_{k \neq i} \mathbf{y}_k \frac{K(\mathbf{x} - \mathbf{x}_k)}{\sum_j K(\mathbf{x} - \mathbf{x}_j)}$$

Kernregression



Kernregression



Unsupervised Kernel Regression

- Unüberwachtes Problem: Nur die $\mathbf{y}_i$ sind bekannt.
- Finde verborgene Struktur, latente Variablen $\mathbf{x}_i$.
  $\longrightarrow$ Dimensionsreduktion
- UKR-Idee:
  - Benutze den Nadaraya-Watson Kernregressionsschätzer, aber fasse die unbekannten $\mathbf{x}_i$ als Parameter auf

$$\mathbf{f}(\mathbf{x}) = \sum_i \mathbf{y}_i \frac{K(\mathbf{x} - \mathbf{x}_i)}{\sum_j K(\mathbf{x} - \mathbf{x}_j)}$$

  - Skalierung der $\mathbf{x}_i$ ist frei, d.h. Bandbreite h kann hier fest auf 1 gesetzt werden.
  - Minimiere CV-Fehler als Funktion der $\mathbf{x}_i$, z.B. durch zufällige Initialisierung und Gradientenabstieg

Unsupervised Kernel Regression

- Unüberwachtes Problem: Nur die $\mathbf{y}_i$ sind bekannt.
- Finde verborgene Struktur, latente Variablen $\mathbf{x}_i$.
  $\longrightarrow$ Dimensionsreduktion
- UKR-Idee:
  - Benutze den Nadaraya-Watson Kernregressionsschätzer, aber fasse die unbekannten $\mathbf{x}_i$ als Parameter auf

$$\mathbf{f}(\mathbf{x}) = \sum_i \mathbf{y}_i \frac{K(\mathbf{x} - \mathbf{x}_i)}{\sum_j K(\mathbf{x} - \mathbf{x}_j)}$$

  - Skalierung der $\mathbf{x}_i$ ist frei, d.h. Bandbreite h kann hier fest auf 1 gesetzt werden.
  - Minimiere CV-Fehler als Funktion der $\mathbf{x}_i$, z.B. durch zufällige Initialisierung und Gradientenabstieg

Unsupervised Kernel Regression

- Unüberwachtes Problem: Nur die $\mathbf{y}_i$ sind bekannt.
- Finde verborgene Struktur, latente Variablen $\mathbf{x}_i$.
  $\longrightarrow$ Dimensionsreduktion
- UKR-Idee:
  - Benutze den Nadaraya-Watson Kernregressionsschätzer, aber fasse die unbekannten $\mathbf{x}_i$ als Parameter auf

$$\mathbf{f}(\mathbf{x}) = \sum_i \mathbf{y}_i \frac{K(\mathbf{x} - \mathbf{x}_i)}{\sum_j K(\mathbf{x} - \mathbf{x}_j)}$$

  - Skalierung der $\mathbf{x}_i$ ist frei, d.h. Bandbreite h kann hier fest auf 1 gesetzt werden.
  - Minimiere CV-Fehler als Funktion der $\mathbf{x}_i$, z.B. durch zufällige Initialisierung und Gradientenabstieg

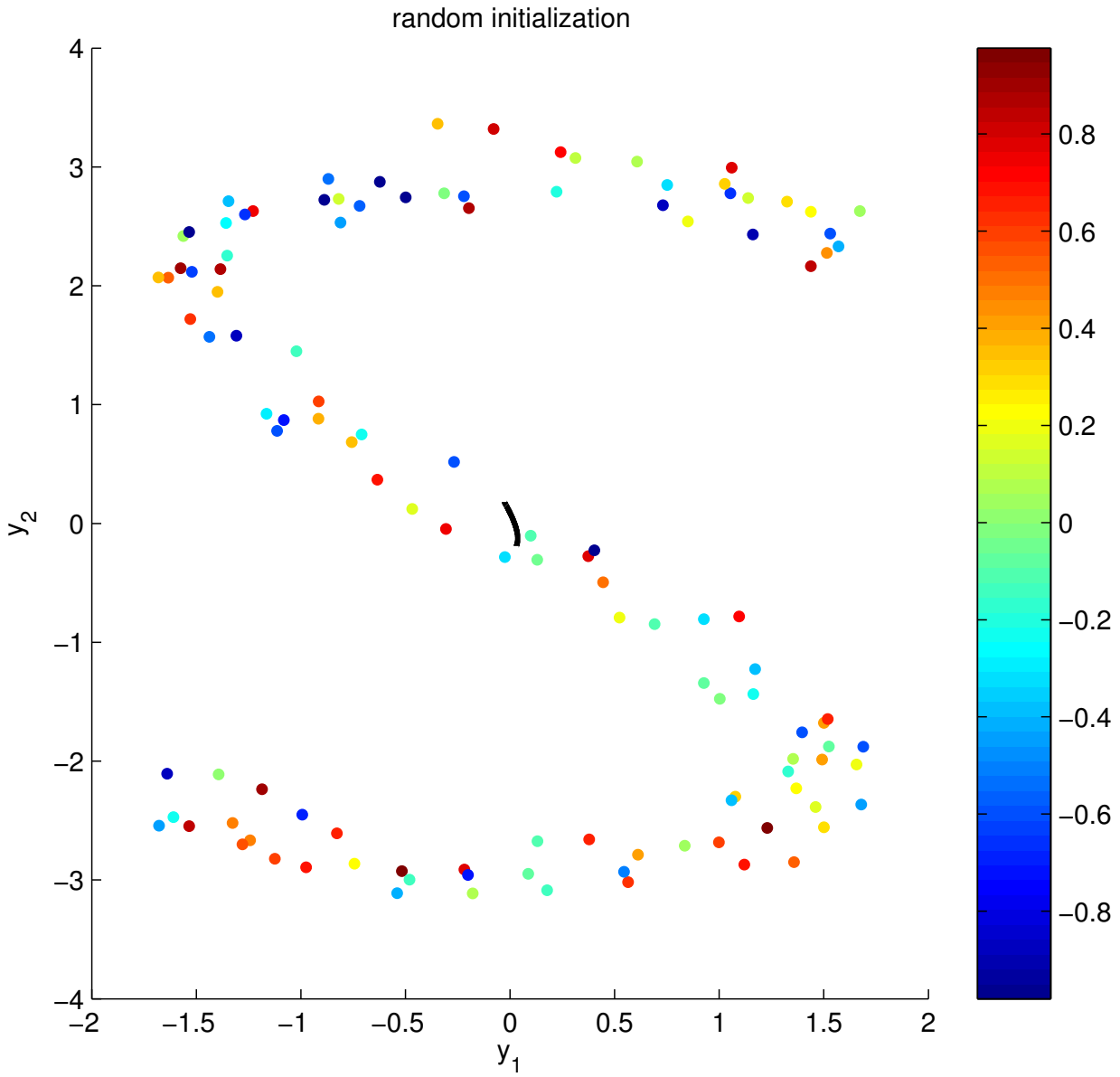
Unsupervised Kernel Regression

- Unüberwachtes Problem: Nur die $\mathbf{y}_i$ sind bekannt.
- Finde verborgene Struktur, latente Variablen $\mathbf{x}_i$.
  $\longrightarrow$ Dimensionsreduktion
- UKR-Idee:
  - Benutze den Nadaraya-Watson Kernregressionsschätzer, aber fasse die unbekannten $\mathbf{x}_i$ als Parameter auf

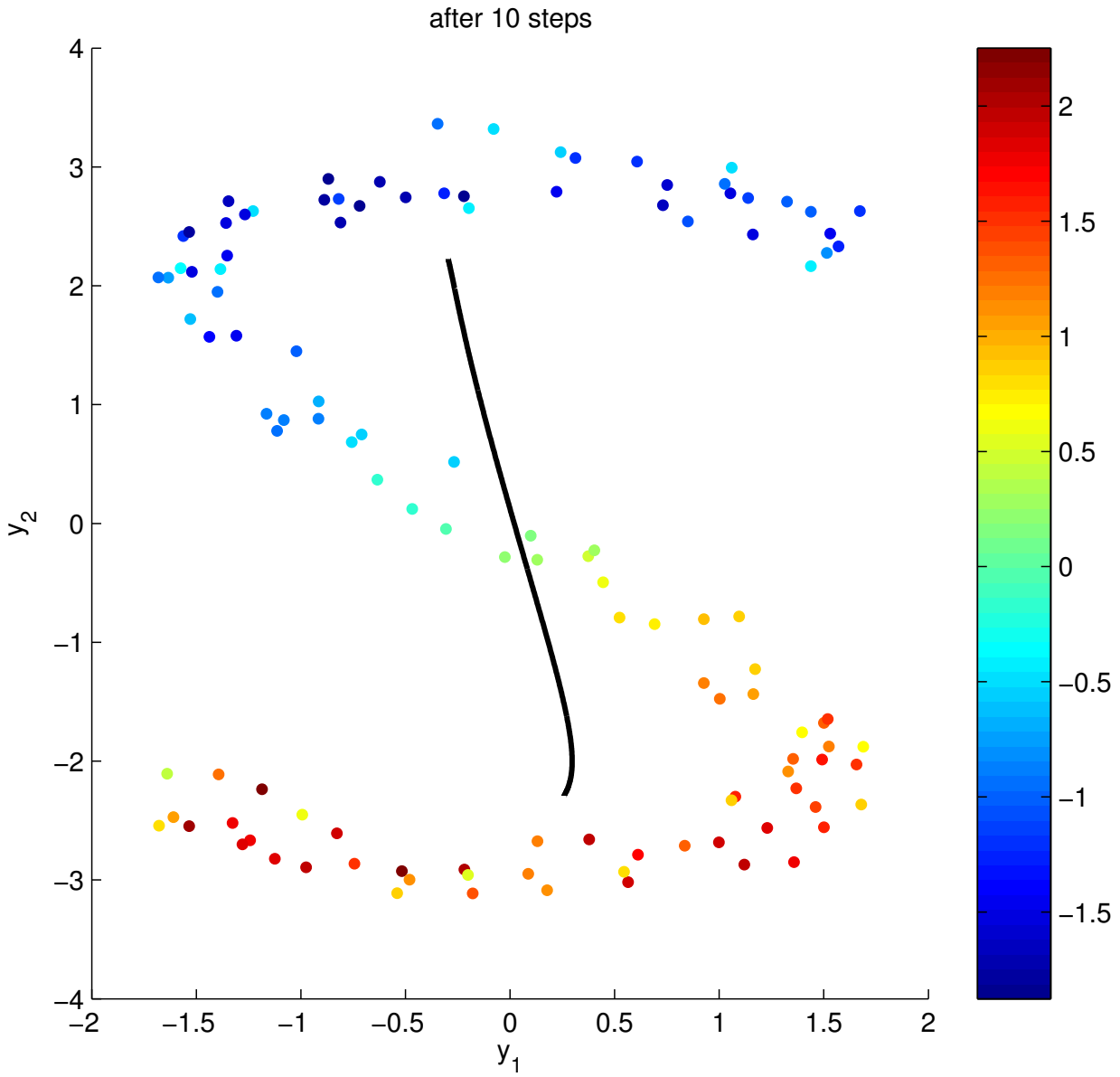
$$\mathbf{f}(\mathbf{x}) = \sum_i \mathbf{y}_i \frac{K(\mathbf{x} - \mathbf{x}_i)}{\sum_j K(\mathbf{x} - \mathbf{x}_j)}$$

  - Skalierung der $\mathbf{x}_i$ ist frei, d.h. Bandbreite h kann hier fest auf 1 gesetzt werden.
  - Minimiere CV-Fehler als Funktion der $\mathbf{x}_i$, z.B. durch zufällige Initialisierung und Gradientenabstieg

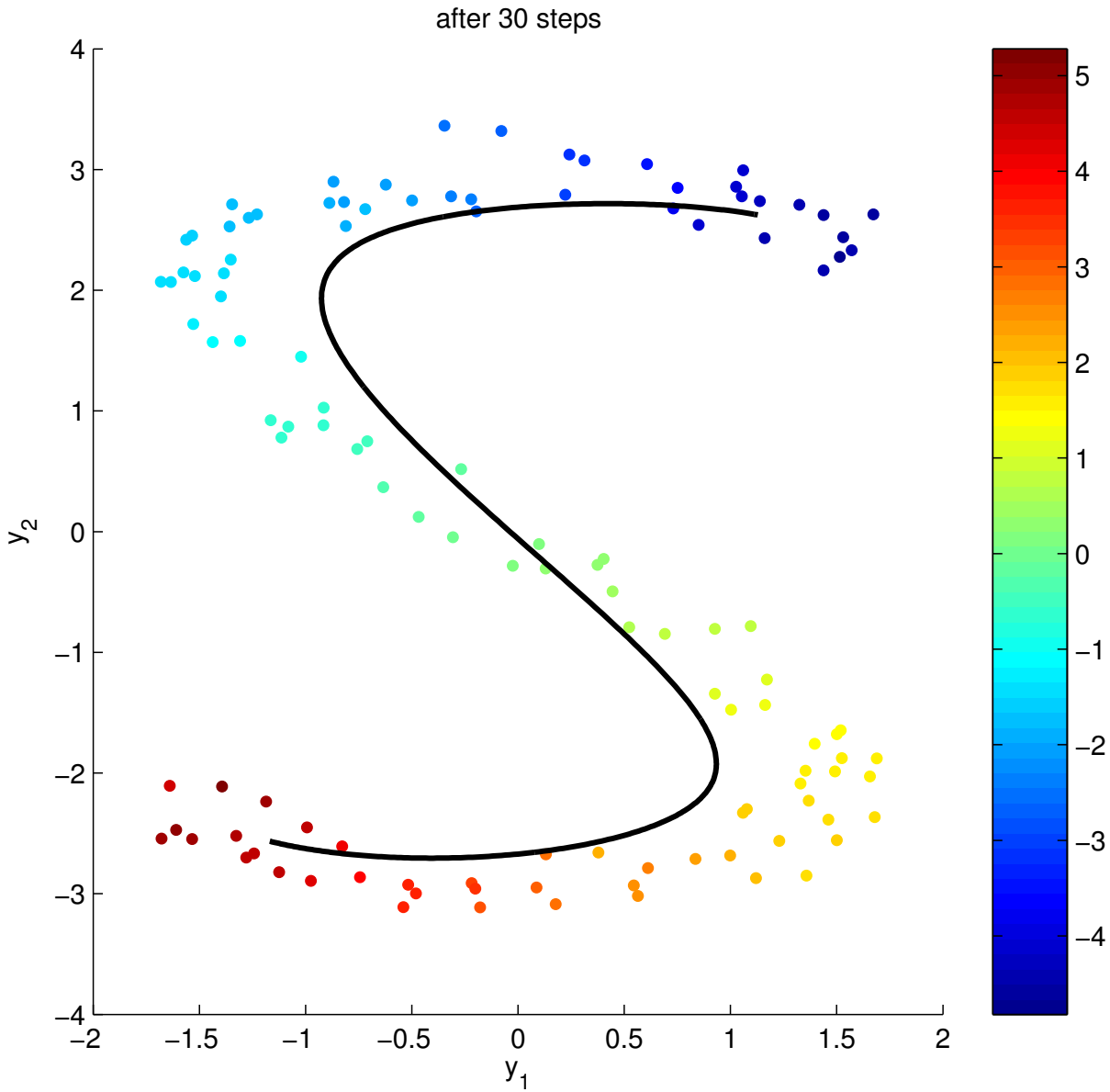
Unsupervised Kernel Regression



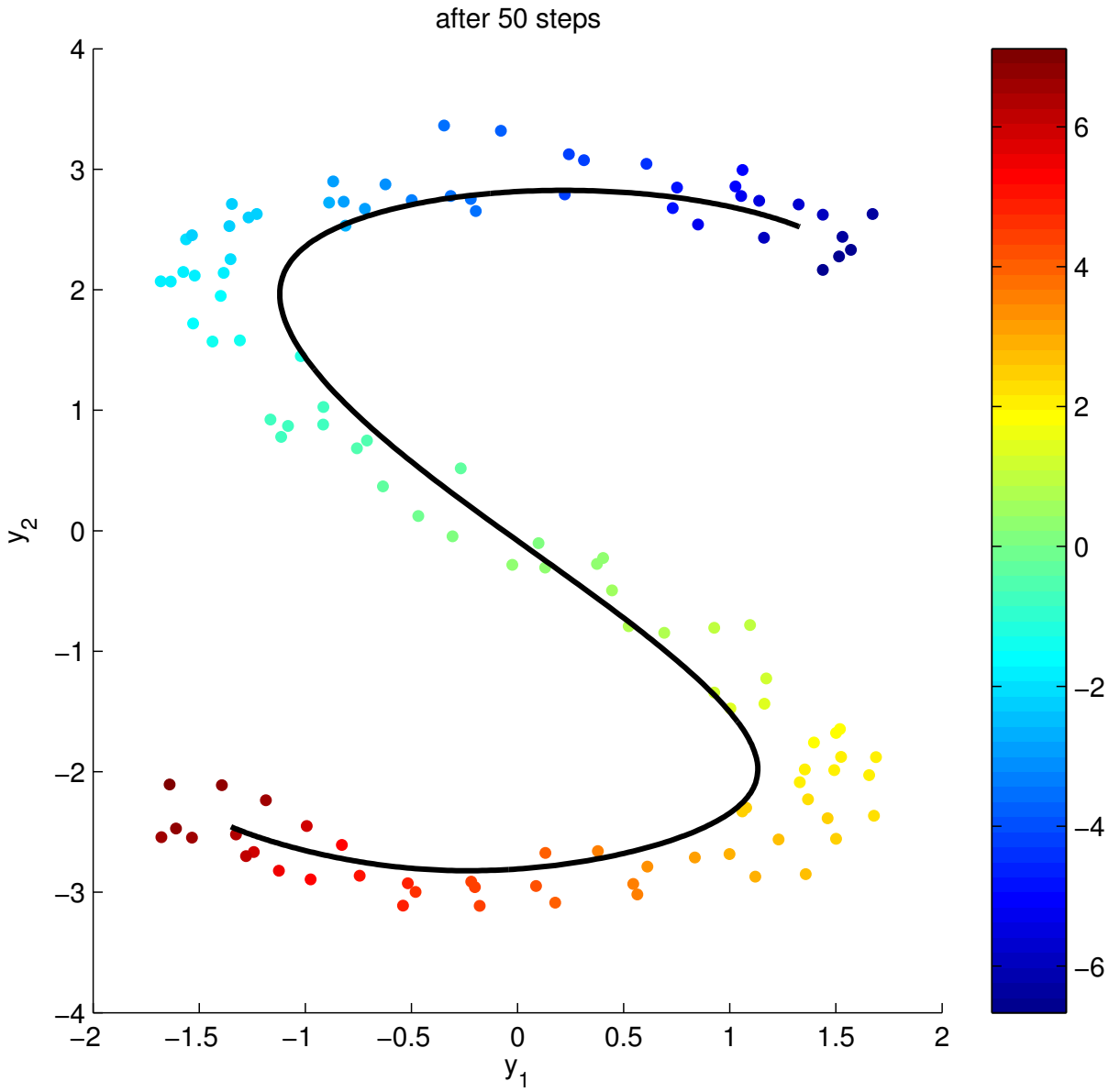
Unsupervised Kernel Regression



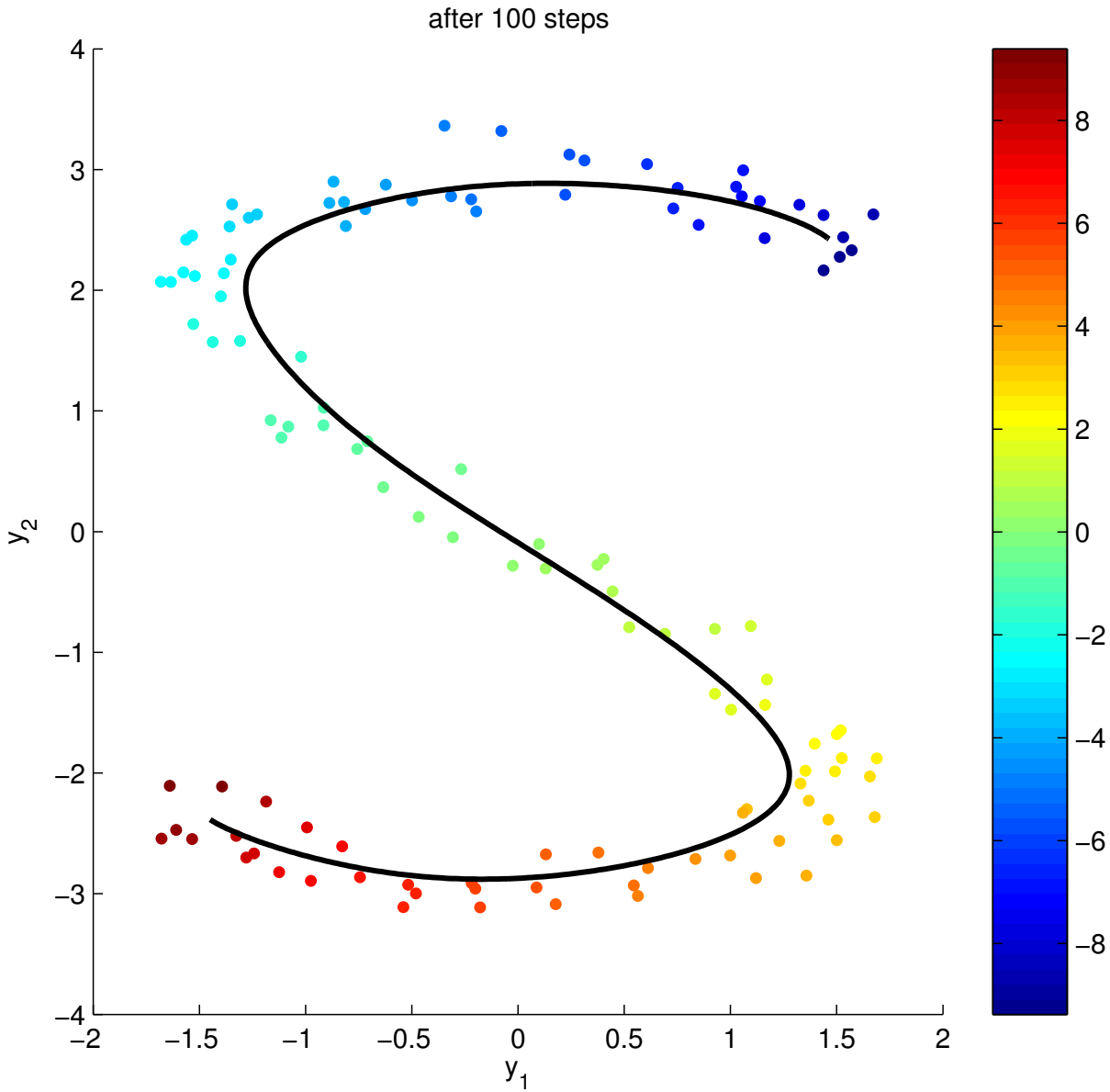
Unsupervised Kernel Regression



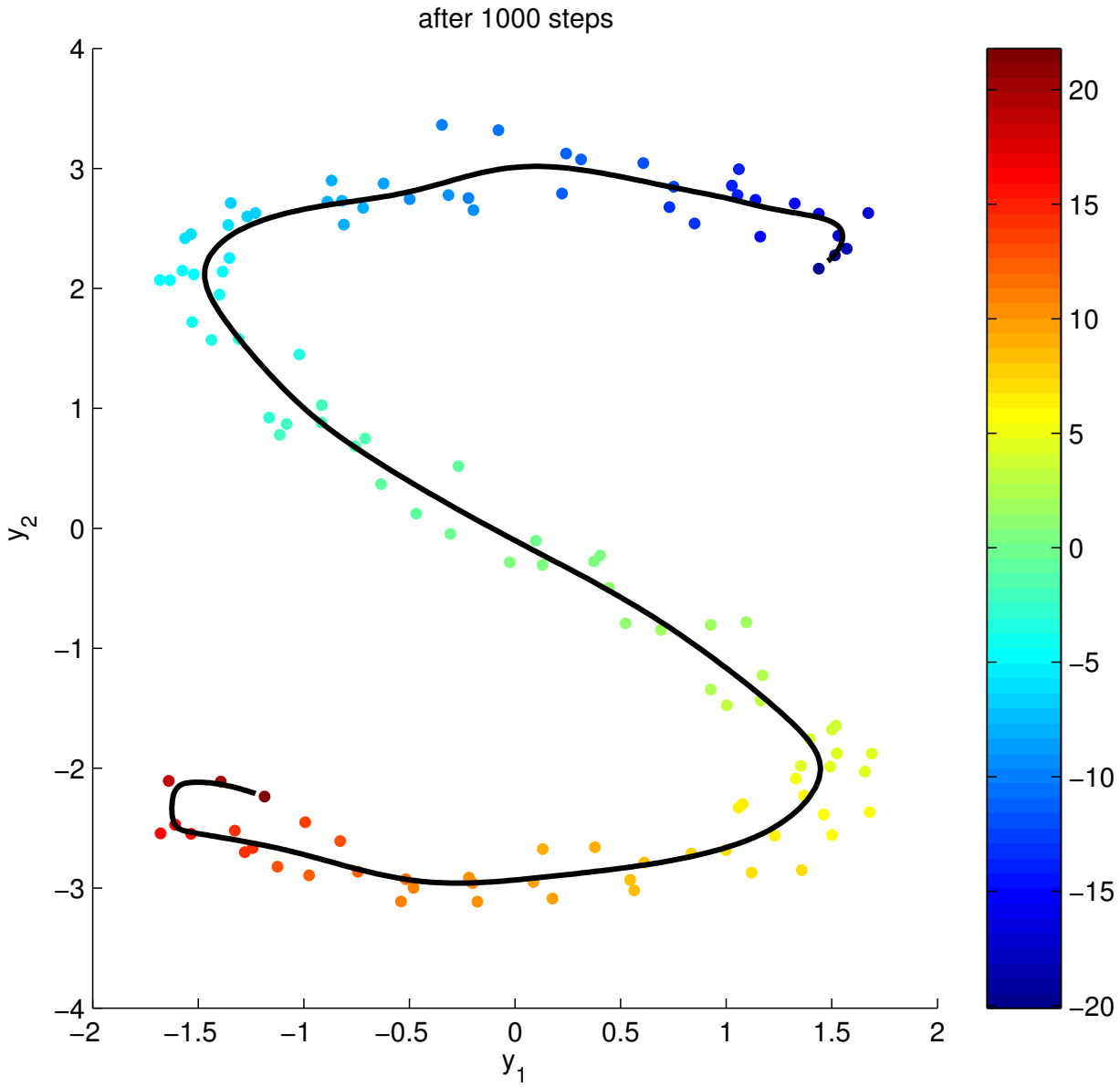
Unsupervised Kernel Regression



Unsupervised Kernel Regression



Unsupervised Kernel Regression



Unsupervised Kernel Regression

- Vorteil gegenüber PCA: nichtlineare Mannigfaltigkeiten behandelbar (dafür aber deutlich aufwendiger)
- Hauptkurven aus Datamining I: Optimierung schwierig, daher Vereinfachung auf diskretes Gitter, das vorher spezifiziert werden muss. Außerdem ist ein Glattheitsterm erforderlich.
- UKR hat mit LOO-CV eine eingebaute **automatische** Komplexitätskontrolle.
- Kein Gitter etc.
- Einzige Spezifikation: Wahl des Kerns (nicht entscheidend für Qualität des Modells)
- Nachteil: Nichtlineare Optimierung recht aufwendig, viele lokale Minima

Unsupervised Kernel Regression

- Vorteil gegenüber PCA: nichtlineare Mannigfaltigkeiten behandelbar (dafür aber deutlich aufwendiger)
- Hauptkurven aus Datamining I: Optimierung schwierig, daher Vereinfachung auf diskretes Gitter, das vorher spezifiziert werden muss. Außerdem ist ein Glattheitsterm erforderlich.
- UKR hat mit LOO-CV eine eingebaute **automatische** Komplexitätskontrolle.
- Kein Gitter etc.
- Einzige Spezifikation: Wahl des Kerns (nicht entscheidend für Qualität des Modells)
- Nachteil: Nichtlineare Optimierung recht aufwendig, viele lokale Minima

Unsupervised Kernel Regression

- Vorteil gegenüber PCA: nichtlineare Mannigfaltigkeiten behandelbar (dafür aber deutlich aufwendiger)
- Hauptkurven aus Datamining I: Optimierung schwierig, daher Vereinfachung auf diskretes Gitter, das vorher spezifiziert werden muss. Außerdem ist ein Glattheitsterm erforderlich.
- UKR hat mit LOO-CV eine eingebaute **automatische** Komplexitätskontrolle.
- Kein Gitter etc.
- Einzige Spezifikation: Wahl des Kerns (nicht entscheidend für Qualität des Modells)
- Nachteil: Nichtlineare Optimierung recht aufwendig, viele lokale Minima

Unsupervised Kernel Regression

- Vorteil gegenüber PCA: nichtlineare Mannigfaltigkeiten behandelbar (dafür aber deutlich aufwendiger)
- Hauptkurven aus Datamining I: Optimierung schwierig, daher Vereinfachung auf diskretes Gitter, das vorher spezifiziert werden muss. Außerdem ist ein Glattheitsterm erforderlich.
- UKR hat mit LOO-CV eine eingebaute **automatische** Komplexitätskontrolle.
- Kein Gitter etc.
- Einzige Spezifikation: Wahl des Kerns (nicht entscheidend für Qualität des Modells)
- Nachteil: Nichtlineare Optimierung recht aufwendig, viele lokale Minima

Unsupervised Kernel Regression

- Vorteil gegenüber PCA: nichtlineare Mannigfaltigkeiten behandelbar (dafür aber deutlich aufwendiger)
- Hauptkurven aus Datamining I: Optimierung schwierig, daher Vereinfachung auf diskretes Gitter, das vorher spezifiziert werden muss. Außerdem ist ein Glattheitsterm erforderlich.
- UKR hat mit LOO-CV eine eingebaute **automatische** Komplexitätskontrolle.
- Kein Gitter etc.
- Einzige Spezifikation: Wahl des Kerns (nicht entscheidend für Qualität des Modells)
- Nachteil: Nichtlineare Optimierung recht aufwendig, viele lokale Minima

(Unüberwachte) Kernregression

- Nützlich zur Effizienzsteigerung: Kernfunktionen mit endlichem Träger
- Epanechnikov-Kernel

$$K_E(x) \propto [1 - x^2]_+$$

- Quartic-Kernel

$$K_Q(x) \propto K_E(x)^2$$

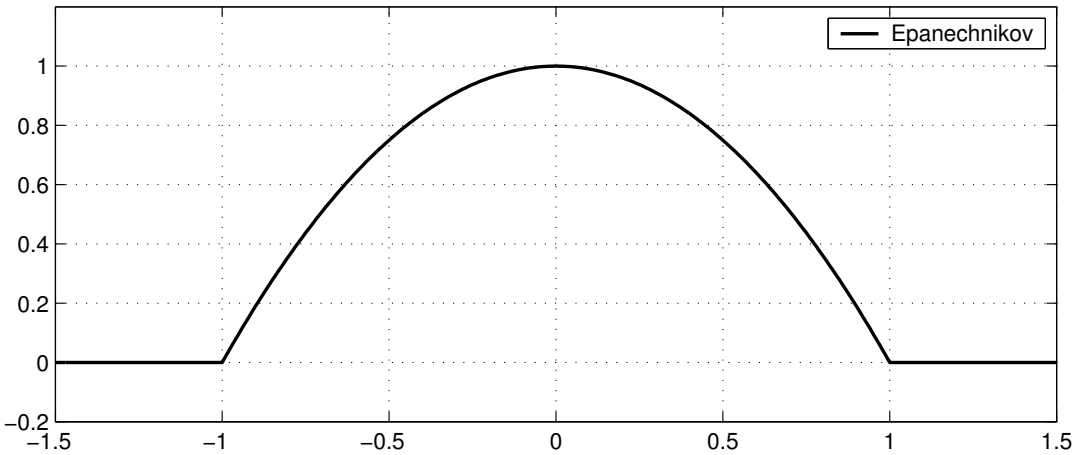
(Unüberwachte) Kernregression

- Nützlich zur Effizienzsteigerung:
  Kernfunktionen mit endlichem Träger
- Epanechnikov-Kernel

$$K_E(x) \propto [1 - x^2]_+$$

- Quartic-Kernel

$$K_Q(x) \propto K_E(x)^2$$



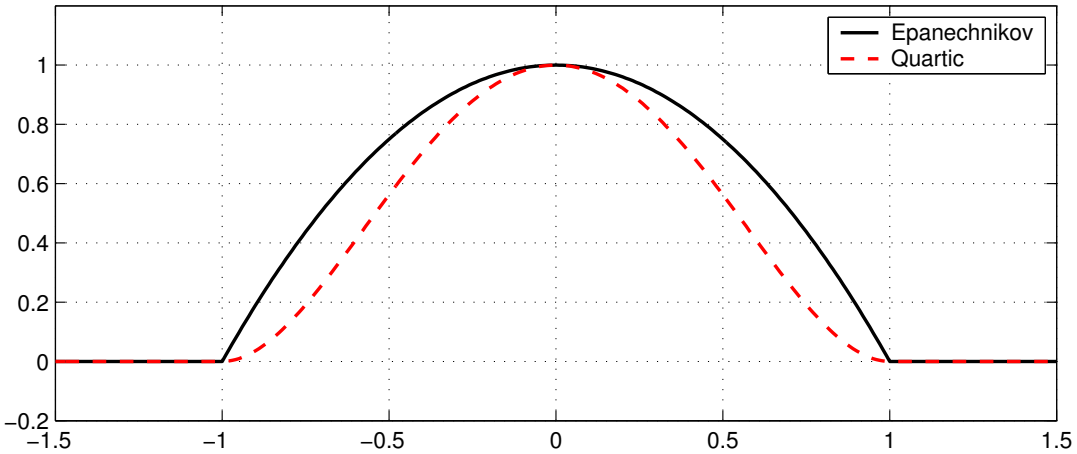
(Unüberwachte) Kernregression

- Nützlich zur Effizienzsteigerung:
  Kernfunktionen mit endlichem Träger
- Epanechnikov-Kernel

$$K_E(x) \propto [1 - x^2]_+$$

- Quartic-Kernel

$$K_Q(x) \propto K_E(x)^2$$



Unsupervised Kernel Regression

- In Kombination mit anderen Methoden können sehr schwierige Probleme “geknackt” werden.
- Es gibt eine Reihe sog. “nonlinear spectral embedding methods[1]”, die (ohne Regressionsfunktion) relativ effizient einen hochdimensionalen Datensatz in einen niedrig dimensionalen Raum einbetten können.
- Der Erfolg der Methoden hängt von einem zu spezifizierenden Nachbarschaftsparameter (im Datenraum der $\mathbf{y}_i$) ab, aber die Methoden bieten selbst i.A. kein Maß für den Erfolg.
- UKR lässt sich mit deren Resultat initialisieren, mehrere Versuche (mit anderer Nachbarschaft) lassen sich per CV-Fehler vergleichen.

[1] Locally Linear Embedding (Roweis u. Saul, 2000), Isomap (Tenenbaum, de Silva u. Langford, 2000), Laplacian Eigenmaps (Belkin u. Niyogi, 2003), Semidefinite Embedding (Weinberger u. Saul, 2004)

Unsupervised Kernel Regression

- In Kombination mit anderen Methoden können sehr schwierige Probleme “geknackt” werden.
- Es gibt eine Reihe sog. “nonlinear spectral embedding methods[1]”, die (ohne Regressionsfunktion) relativ effizient einen hochdimensionalen Datensatz in einen niedrig dimensionalen Raum einbetten können.
- Der Erfolg der Methoden hängt von einem zu spezifizierenden Nachbarschaftsparameter (im Datenraum der $\mathbf{y}_i$) ab, aber die Methoden bieten selbst i.A. kein Maß für den Erfolg.
- UKR lässt sich mit deren Resultat initialisieren, mehrere Versuche (mit anderer Nachbarschaft) lassen sich per CV-Fehler vergleichen.

[1] Locally Linear Embedding (Roweis u. Saul, 2000), Isomap (Tenenbaum, de Silva u. Langford, 2000), Laplacian Eigenmaps (Belkin u. Niyogi, 2003), Semidefinite Embedding (Weinberger u. Saul, 2004)

Unsupervised Kernel Regression

- In Kombination mit anderen Methoden können sehr schwierige Probleme "geknackt" werden.
- Es gibt eine Reihe sog. "nonlinear spectral embedding methods[1]", die (ohne Regressionsfunktion) relativ effizient einen hochdimensionalen Datensatz in einen niedrig dimensionalen Raum einbetten können.
- Der Erfolg der Methoden hängt von einem zu spezifizierenden Nachbarschaftsparameter (im Datenraum der $\mathbf{y}_i$) ab, aber die Methoden bieten selbst i.A. kein Maß für den Erfolg.
- UKR lässt sich mit deren Resultat initialisieren, mehrere Versuche (mit anderer Nachbarschaft) lassen sich per CV-Fehler vergleichen.

[1] Locally Linear Embedding (Roweis u. Saul, 2000), Isomap (Tenenbaum, de Silva u. Langford, 2000), Laplacian Eigenmaps (Belkin u. Niyogi, 2003), Semidefinite Embedding (Weinberger u. Saul, 2004)

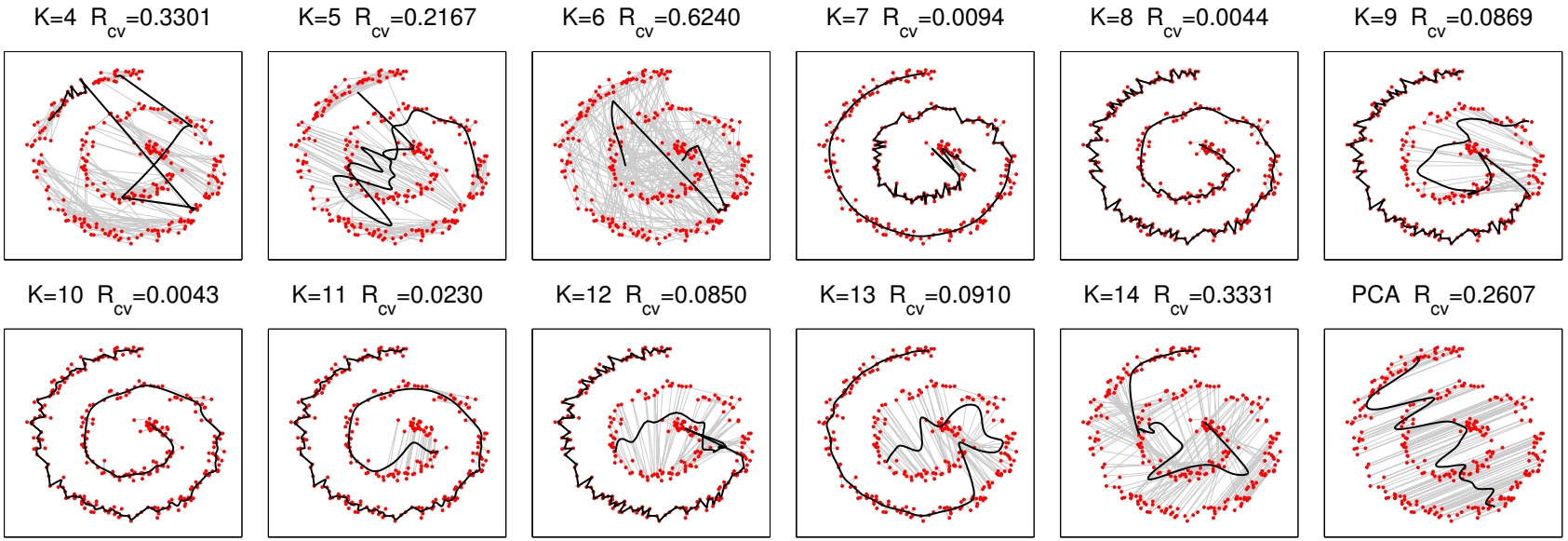
Unsupervised Kernel Regression

- In Kombination mit anderen Methoden können sehr schwierige Probleme "geknackt" werden.
- Es gibt eine Reihe sog. "nonlinear spectral embedding methods[1]", die (ohne Regressionsfunktion) relativ effizient einen hochdimensionalen Datensatz in einen niedrig dimensionalen Raum einbetten können.
- Der Erfolg der Methoden hängt von einem zu spezifizierenden Nachbarschaftsparameter (im Datenraum der $\mathbf{y}_i$) ab, aber die Methoden bieten selbst i.A. kein Maß für den Erfolg.
- UKR lässt sich mit deren Resultat initialisieren, mehrere Versuche (mit anderer Nachbarschaft) lassen sich per CV-Fehler vergleichen.

[1] Locally Linear Embedding (Roweis u. Saul, 2000), Isomap (Tenenbaum, de Silva u. Langford, 2000), Laplacian Eigenmaps (Belkin u. Niyogi, 2003), Semidefinite Embedding (Weinberger u. Saul, 2004)

Unsupervised Kernel Regression

- "Noisy Spiral" mit LLE & PCA-Initialisierung
- Gute Lösungen weisen kleine CV-Fehler auf



UKR zur Visualisierung

- USPS Digit "2": 731 Datenvektoren der Dimension 16x16
- Projektion in 2D-Raum, dort Abtastung von $\mathbf{f}(x)$
  $\longrightarrow$ 2D-Karte handgeschriebener Ziffern
- Dichte im latenten Raum als Intensität verwendbar