Eindimensionale Kernglättung

Michelle Anell

Seminar Machine Learning
Mathematisches Institut

17. Mai 2019

Übersicht

Übersicht

Kernglättungsmethoden

- Klasse von flexiblen Regressionstechniken
  $\rightarrow$ Separate Anpassung unterschiedlicher Modelle an jedem Beobachtungspunkt x_0

Kernglättungsmethoden

- Klasse von flexiblen Regressionstechniken
  $\rightarrow$ Separate Anpassung unterschiedlicher Modelle an jedem Beobachtungspunkt x_0
- Form des funktionalen Zusammenhangs nicht vorgegeben, sondern aus Daten hergeleitet

Kernglättungsmethoden

- Klasse von flexiblen Regressionstechniken
  $\rightarrow$ Separate Anpassung unterschiedlicher Modelle an jedem Beobachtungspunkt x_0
- Form des funktionalen Zusammenhangs nicht vorgegeben, sondern aus Daten hergeleitet
- Verwendung von Beobachtungen nah am Zielpunkt x_0, s.d. resultierende geschätzte Funktion $\hat{f}(X)$ glatt ist
  $\rightarrow$ Approximation als gewichtete Summe der naheliegenden Beobachtungswerte berechnet

Kerne

- Lokalisierung durch *Kern* $K_\lambda(x_0, x_i)$
  $\rightarrow$ *Gewichtungsfunktion*: ordnet x_i Gewicht zu, basierend auf Distanz zu x_0
  $\rightarrow$ Nahe Punkte erhalten höhere Gewichte

Kerne

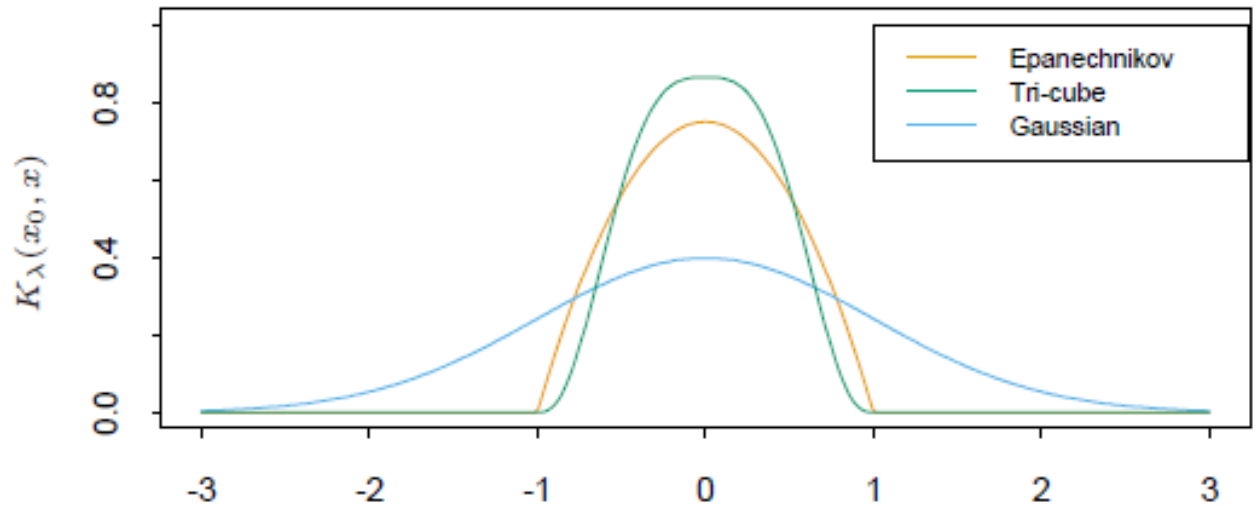
- Lokalisierung durch *Kern* $K_\lambda(x_0, x_i)$
  $\rightarrow$ *Gewichtungsfunktion*: ordnet x_i Gewicht zu, basierend auf Distanz zu x_0
  $\rightarrow$ Nahe Punkte erhalten höhere Gewichte
- Definiere

$$K_\lambda(x_0, x_i) = D\left(\frac{|x_i - x_0|}{h_\lambda(x_0)}\right)$$

  $\rightarrow$ $D(\cdot)$ symmetrisch um x_0, Wert sinkt mit steigender Distanz zwischen x_i und x_0
  $\rightarrow$ $h_\lambda(x_0)$ *Breitefunktion*: bestimmt Breite der betrachteten Umgebung bei x_0 in Abhängigkeit von Index λ

Verschiedene Kerne

- Gauss-Kern: $D(t) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}t^2}, \quad h_\lambda(x_0) = \sigma$
- Epanechnikov-Kern: $D(t) = \begin{cases} \frac{3}{4}(1-t^2), & \text{falls } |t| \leq 1 \\ 0, & \text{sonst} \end{cases}, \quad h_\lambda(x_0) = \lambda$
- Tricube-Kern: $D(t) = \begin{cases} (1-|t|^3)^3, & \text{falls } |t| \leq 1 \\ 0, & \text{sonst} \end{cases}, \quad h_\lambda(x_0) = \lambda$



Aus: The Elements of Statistical Learning [1]

K-Nächste-Nachbarn Durchschnitt

- Schätzung der Regressionsfunktion: $\hat{f}(x) = Ave(y_i|x_i \in N_k(x))$
  $\rightarrow N_k(x)$ Menge der k Punkte mit geringstem Abstand von x

K-Nächste-Nachbarn Durchschnitt

- Schätzung der Regressionsfunktion: $\hat{f}(x) = Ave(y_i|x_i \in N_k(x))$
  $\rightarrow N_k(x)$ Menge der k Punkte mit geringstem Abstand von x
- Kern: Umgebungsgröße k ersetzt λ, s.d.

$$h_k(x_0) = \left|x_0 - x_{[k]}\right|$$

mit $x_{[k]}$ k-nächstes x_i zu x_0 und

$$D(t) = \begin{cases} 1/k, & \text{falls } |t| \leq 1 \\ 0, & \text{sonst} \end{cases}$$

K-Nächste-Nachbarn Durchschnitt

- Schätzung der Regressionsfunktion: $\hat{f}(x) = Ave(y_i|x_i \in N_k(x))$
  $\rightarrow N_k(x)$ Menge der k Punkte mit geringstem Abstand von x
- Kern: Umgebungsgröße k ersetzt λ, s.d.

$$h_k(x_0) = \left|x_0 - x_{[k]}\right|$$

mit $x_{[k]}$ k-nächstes x_i zu x_0 und

$$D(t) = \begin{cases} 1/k, & \text{falls } |t| \leq 1 \\ 0, & \text{sonst} \end{cases}$$

- Nachteil:Unstetigkeiten, da gleiche Gewichte für alle Punkte, trotz unterschiedlicher Distanz zu x_0

Beispiel

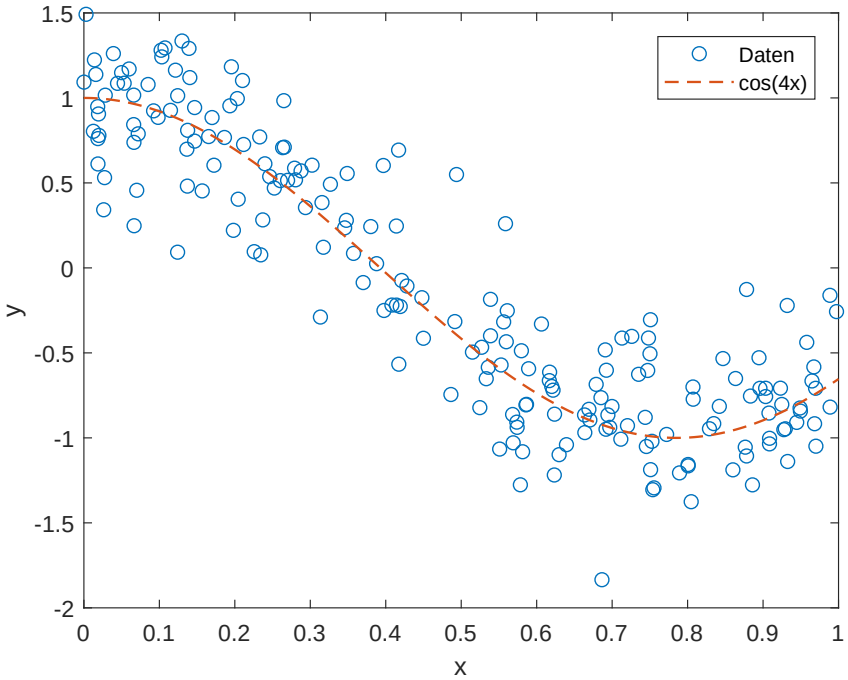
Wir betrachten $N = 200$ Paare (x_i, y_i) mit

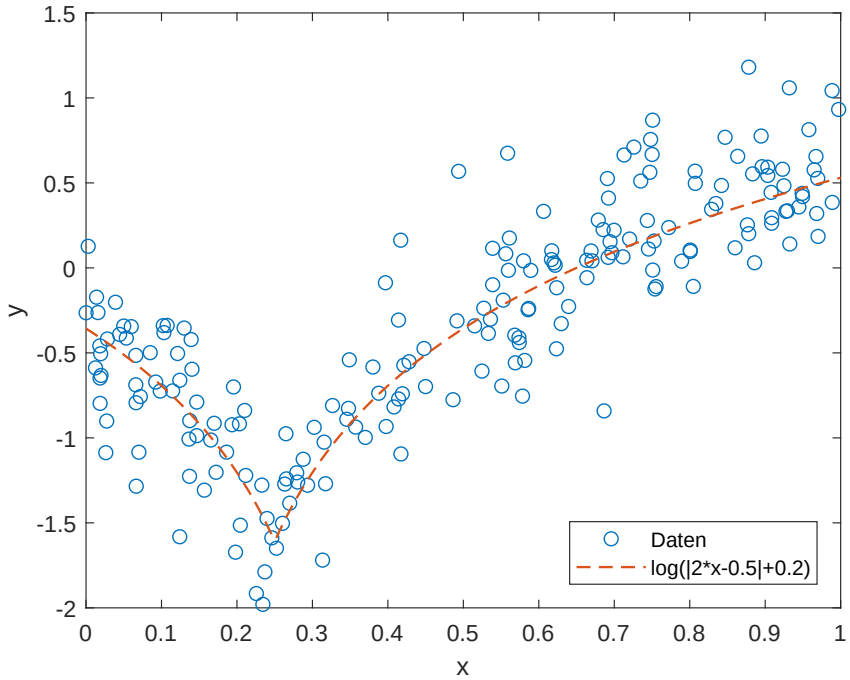
$$Y = f(X) + \varepsilon,$$

$$(1)\ f(X) = \cos(4X),$$

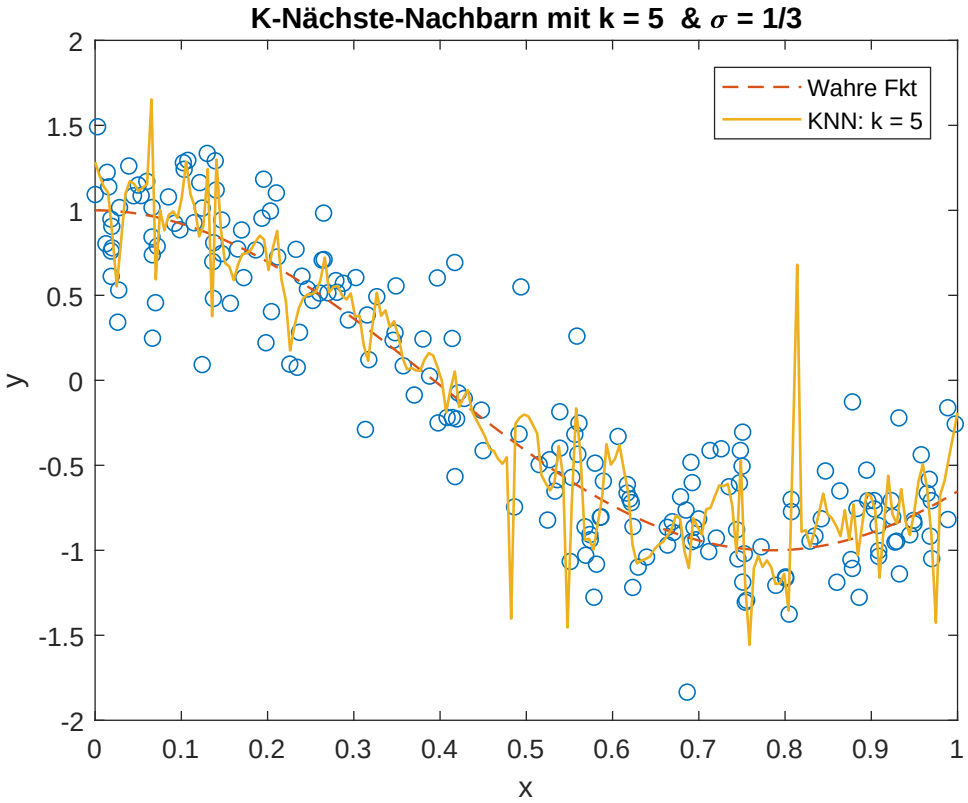
$$(2)\ f(X) = \log(|2X - 0.5| + 0.2),$$

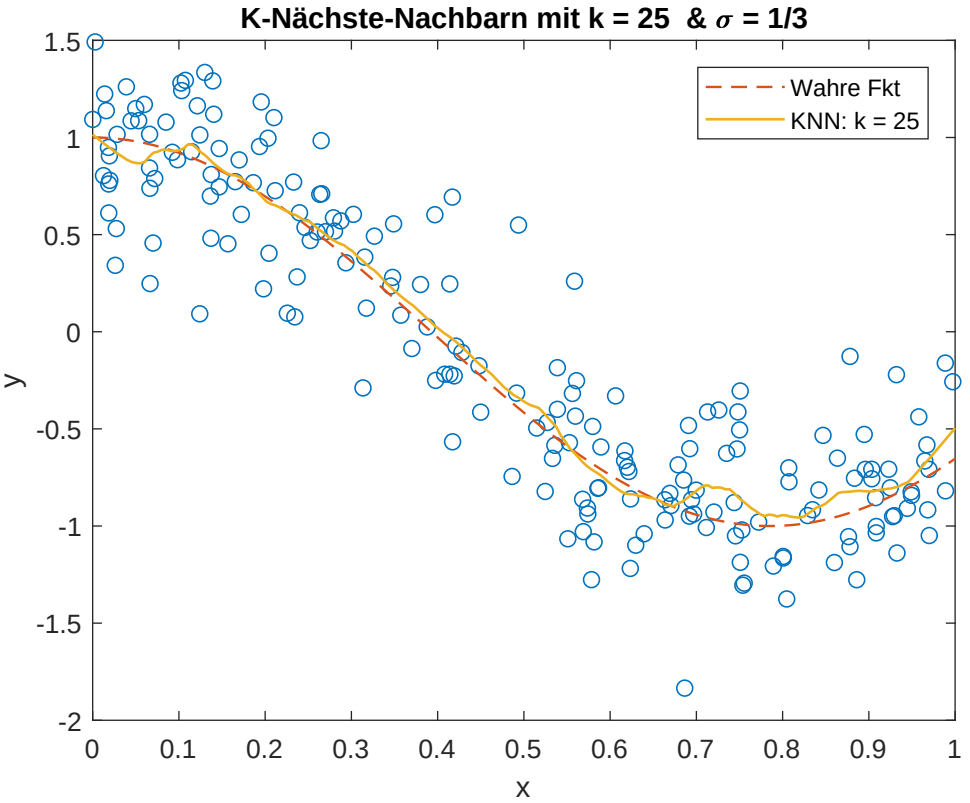
mit $X \sim U[0,1]$ und $\varepsilon \sim N(0, (1/3)^2)$



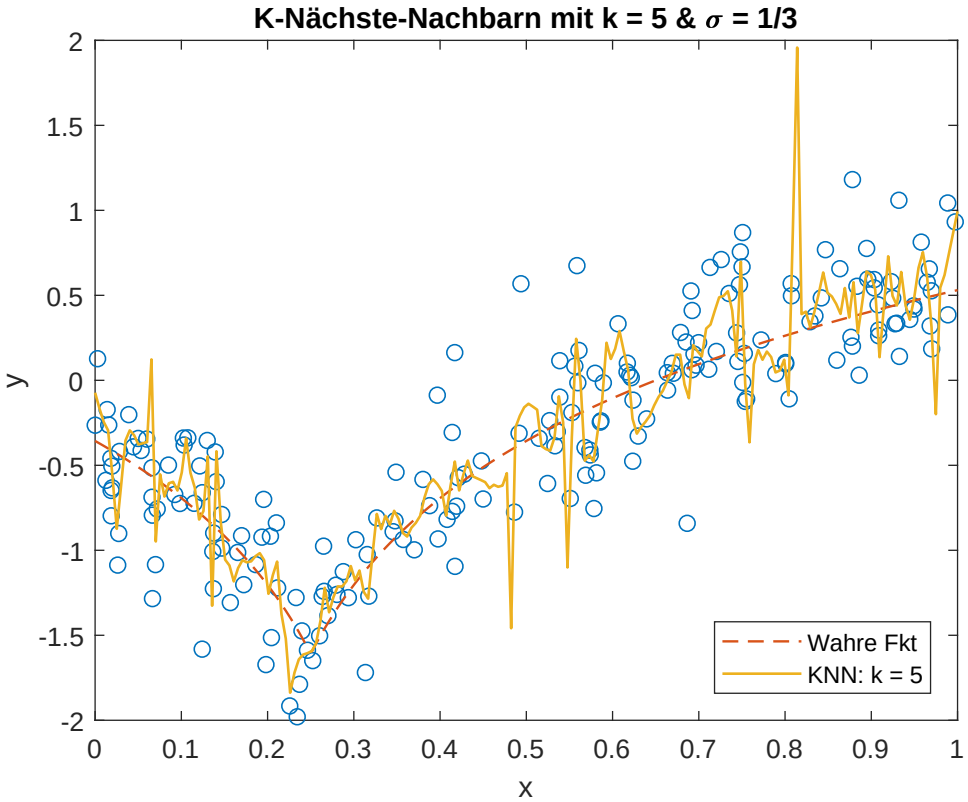


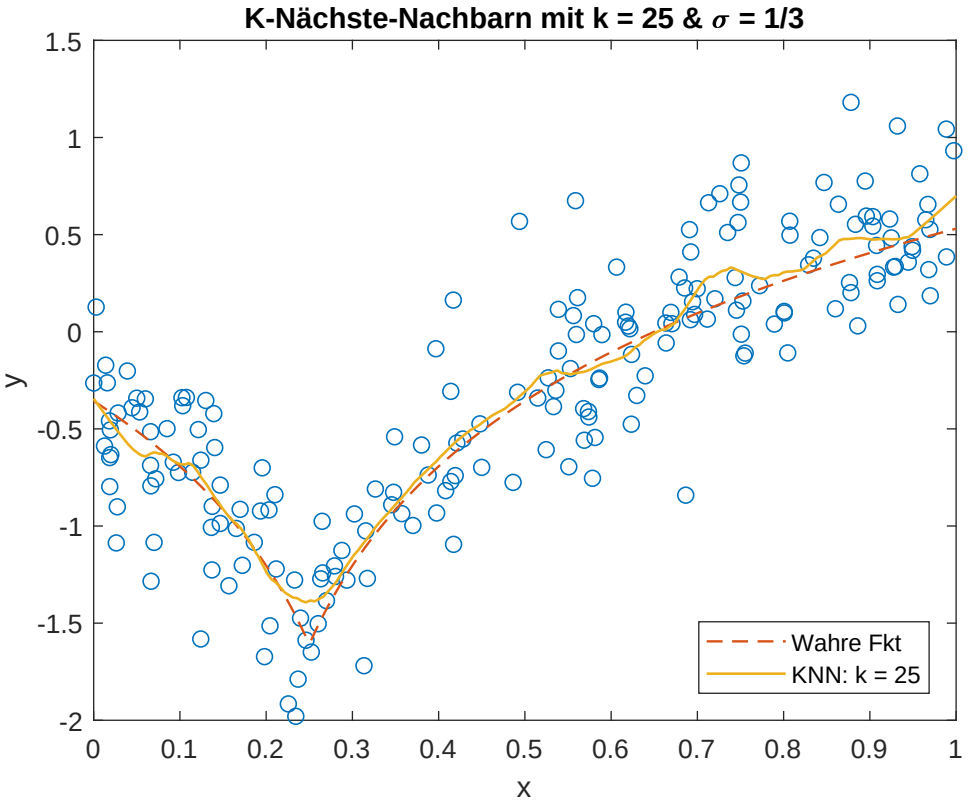
Schätzung mit K-Nächste-Nachbarn Durchschnitt





Schätzung mit K-Nächste-Nachbarn Durchschnitt





Nadaraya-Watson Durchschnitt

- Schätzt Funktion aus Beobachtungsdaten $(x_1, y_1), \ldots, (x_N, y_N)$ durch

$$\hat{f}(x_0) = \frac{\sum_{i=1}^{N} K_\lambda(x_0, x_i) y_i}{\sum_{i=1}^{N} K_\lambda(x_0, x_i)},$$

  mit $h_\lambda(x_0) = \lambda = konstant$

Nadaraya-Watson Durchschnitt

- Schätzt Funktion aus Beobachtungsdaten $(x_1, y_1), \ldots, (x_N, y_N)$ durch

$$\hat{f}(x_0) = \frac{\sum_{i=1}^{N} K_\lambda(x_0, x_i) y_i}{\sum_{i=1}^{N} K_\lambda(x_0, x_i)},$$

  mit $h_\lambda(x_0) = \lambda = konstant$
- Kern K_λ so konstruiert, dass Gewicht einer Beobachtung y_i immer kleiner wird, je größer der Abstand $|x_i - x_0|$ ist

Nadaraya-Watson Durchschnitt

- Schätzt Funktion aus Beobachtungsdaten $(x_1, y_1), \ldots, (x_N, y_N)$ durch

$$\hat{f}(x_0) = \frac{\sum_{i=1}^{N} K_\lambda(x_0, x_i) y_i}{\sum_{i=1}^{N} K_\lambda(x_0, x_i)},$$

  mit $h_\lambda(x_0) = \lambda = konstant$
- Kern K_λ so konstruiert, dass Gewicht einer Beobachtung y_i immer kleiner wird, je größer der Abstand $|x_i - x_0|$ ist
- Wenn wir Ziel nach rechts bewegen, betreten Punkte die Umgebung mit Gewicht 0, dann steigt Wert langsam an

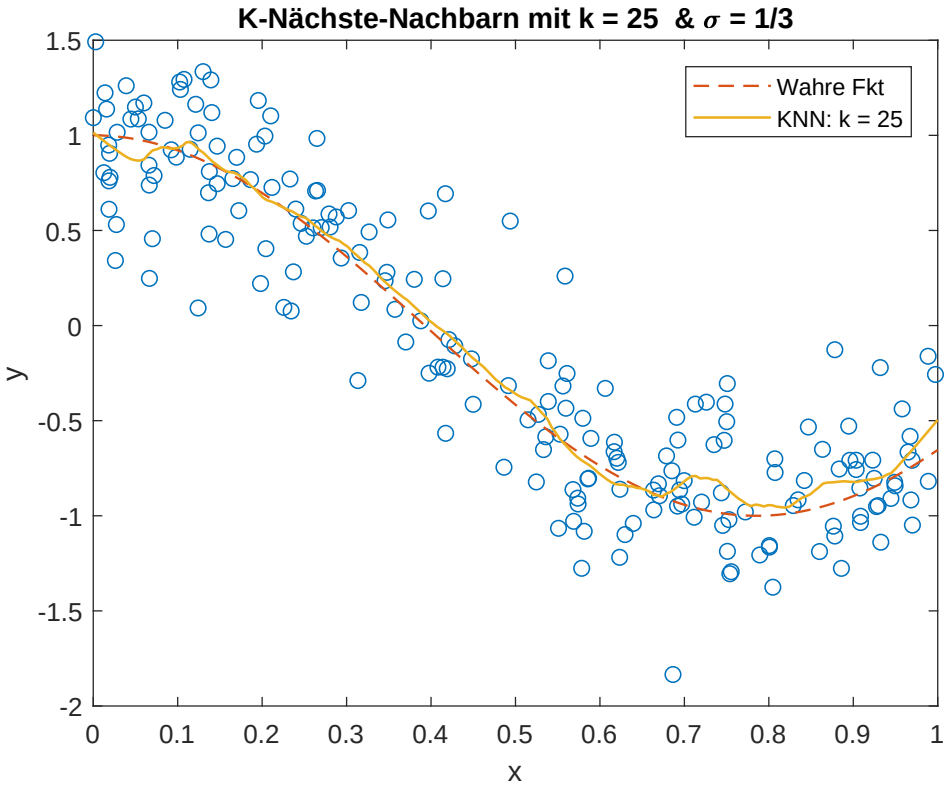
Nadaraya-Watson Durchschnitt

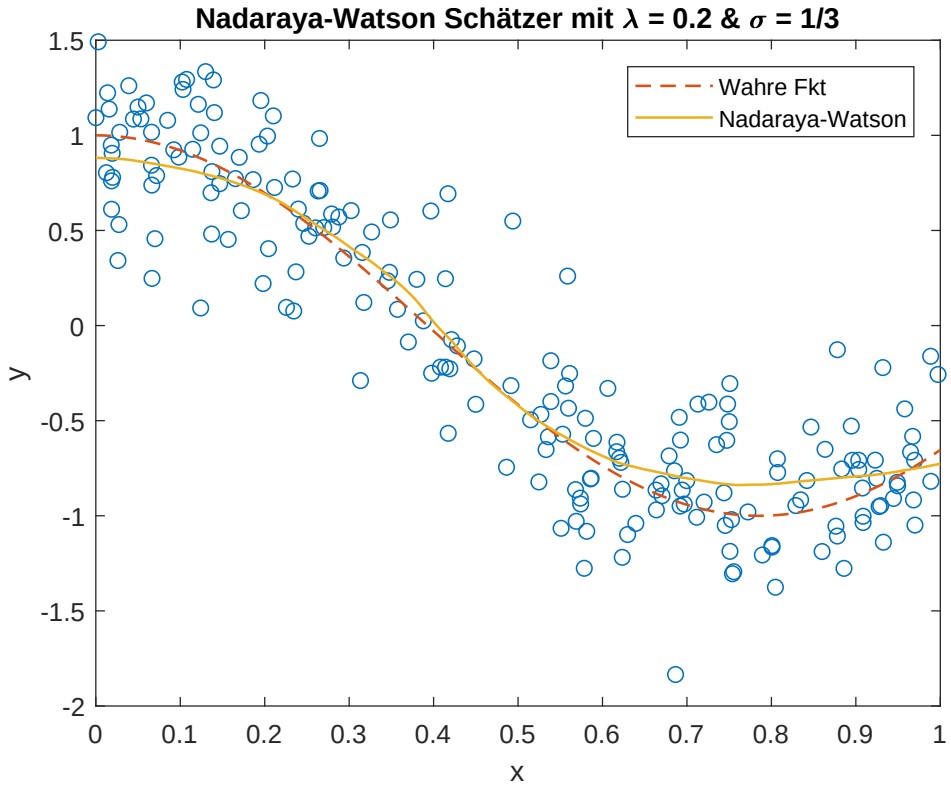
- Schätzt Funktion aus Beobachtungsdaten $(x_1, y_1), \ldots, (x_N, y_N)$ durch

$$\hat{f}(x_0) = \frac{\sum_{i=1}^{N} K_\lambda(x_0, x_i) y_i}{\sum_{i=1}^{N} K_\lambda(x_0, x_i)},$$

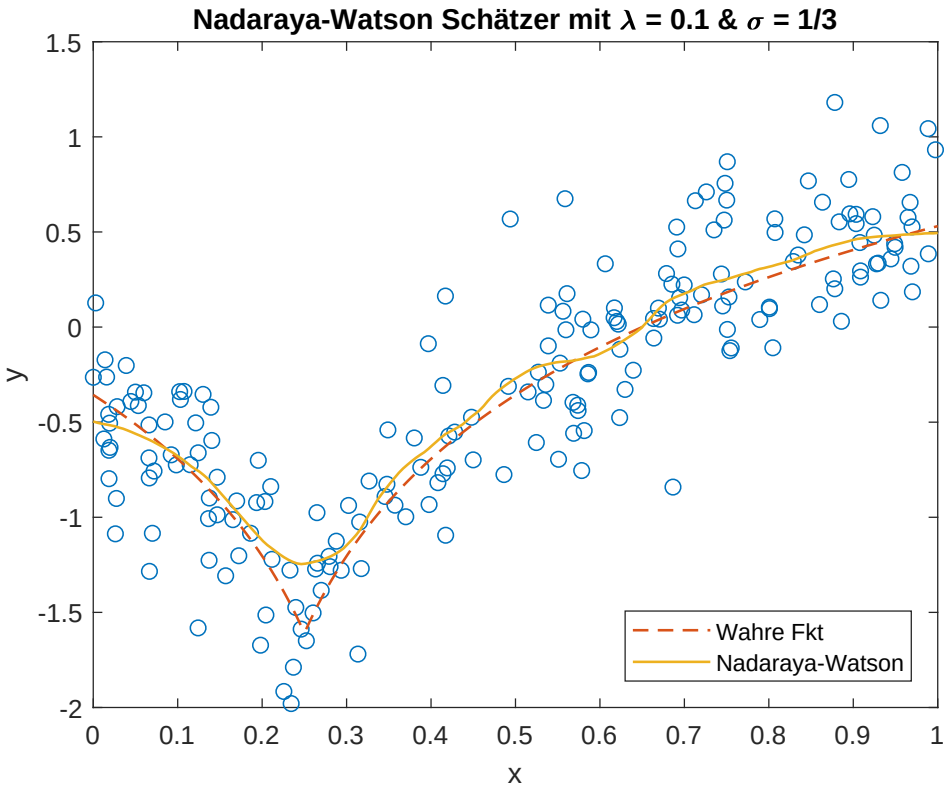
  mit $h_\lambda(x_0) = \lambda = konstant$
- Kern K_λ so konstruiert, dass Gewicht einer Beobachtung y_i immer kleiner wird, je größer der Abstand $|x_i - x_0|$ ist
- Wenn wir Ziel nach rechts bewegen, betreten Punkte die Umgebung mit Gewicht 0, dann steigt Wert langsam an
- Bandweite λ legt fest, in welchem Bereich um x_0 die Beobachtungen großes Gewicht haben: Einfluss auf Glattheit des Schätzers

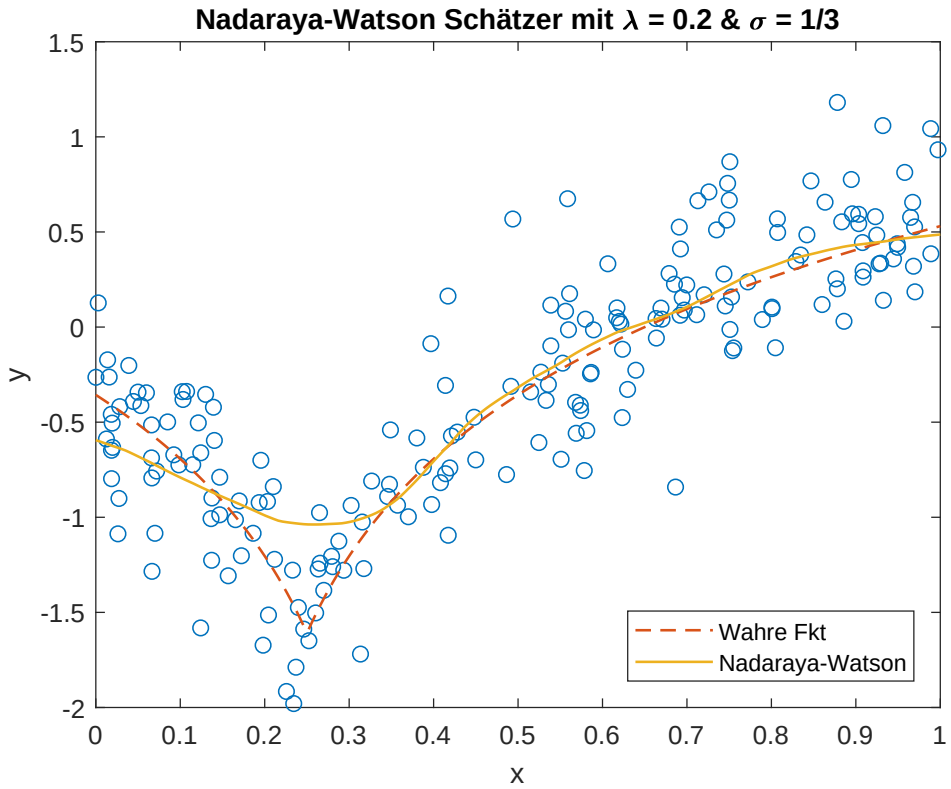
K-Nächste-Nachbarn vs. Nadaraya-Watson Schätzer



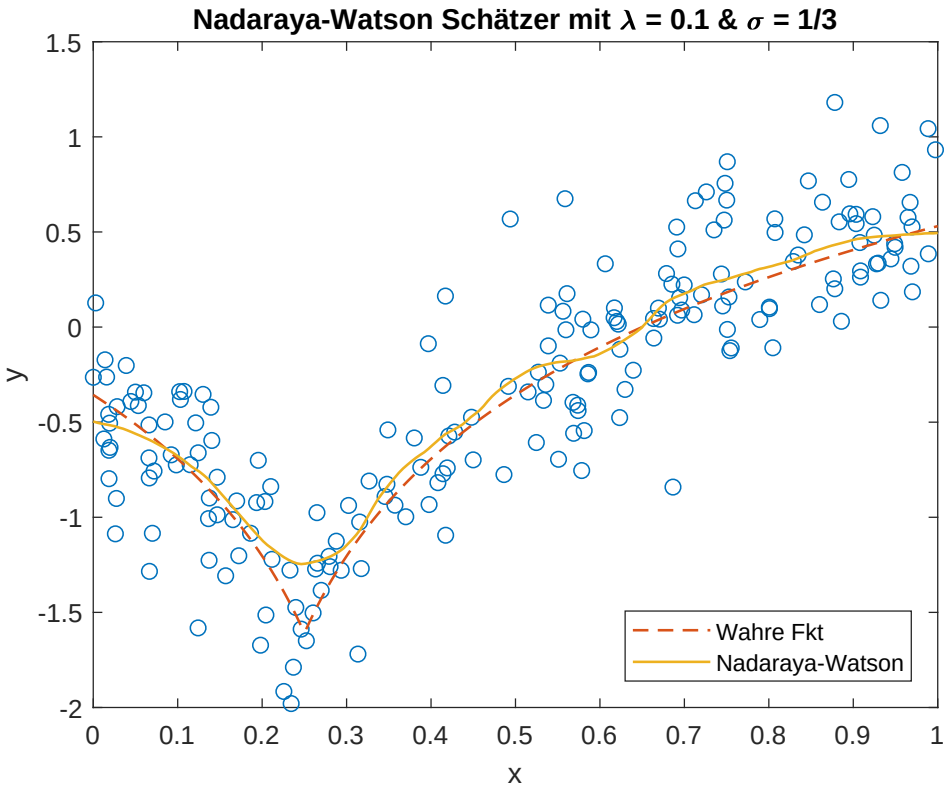


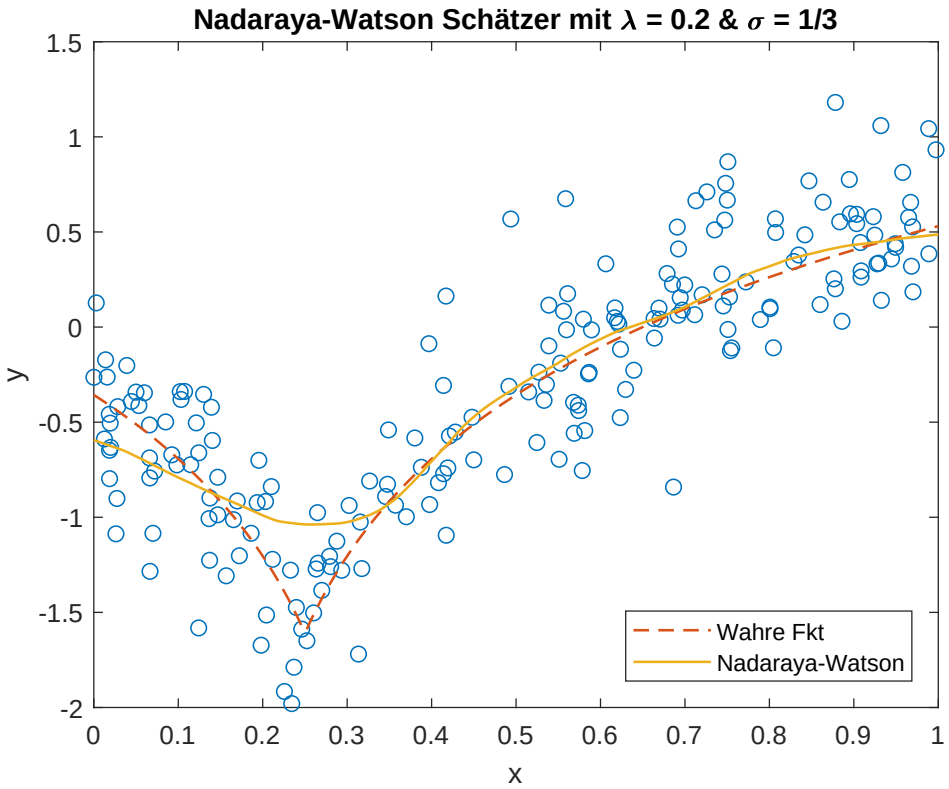
Nadaraya-Watson Schätzer mit verschiedenen λ





Nadaraya-Watson Schätzer mit verschiedenen λ





Bias an den Rändern

$\rightarrow$ Umgebungen enthalten weniger Punkte

Übersicht

Idee

- Lokal gewichtete Durchschnitte können an Rändern verzerrt sein
- Lösung: Lokale Anpassung gerader Linien, anstelle von Konstanten
  $\rightarrow$ Gewichtete lokale Anpassung einer Geraden an jedem Punkt x_0

Idee

- Lokal gewichtete Durchschnitte können an Rändern verzerrt sein
- Lösung: Lokale Anpassung gerader Linien, anstelle von Konstanten
  $\rightarrow$ Gewichtete lokale Anpassung einer Geraden an jedem Punkt x_0
- Entfernen Bias am Rand zu erster Ordnung
- Bias im Inneren, falls Beobachtungen nicht gleichmäßig verteilt sind kann auch entfernt werden

Berechnung

- Bestimme Koeffizienten $\hat{\alpha}(x_0), \hat{\beta}(x_0)$ für gegebenes x_0
- Lösen separates gewichtetes kleinste Quadrate Problem an jedem Zielpunkt x_0:

$$\min_{\alpha(x_0),\beta(x_0)} \sum_{i=1}^{N} K_\lambda(x_0, x_i) \Big[y_i - \alpha(x_0) - \beta(x_0) x_i \Big]^2$$

$\Rightarrow$ Schätzung: $\hat{f}(x_0) = \hat{\alpha}(x_0) + \hat{\beta}(x_0) x_0$

Berechnung

- Bestimme Koeffizienten $\hat{\alpha}(x_0), \hat{\beta}(x_0)$ für gegebenes x_0
- Lösen separates gewichtetes kleinste Quadrate Problem an jedem Zielpunkt x_0:

$$\min_{\alpha(x_0),\beta(x_0)} \sum_{i=1}^{N} K_\lambda(x_0, x_i) \Big[y_i - \alpha(x_0) - \beta(x_0) x_i \Big]^2$$

  $\Rightarrow$ Schätzung: $\hat{f}(x_0) = \hat{\alpha}(x_0) + \hat{\beta}(x_0) x_0$
- Passen gesamtes lineares Modell auf Daten in der Region an, aber verwenden es nur, um Anpassung am Punkt x_0 zu berechnen

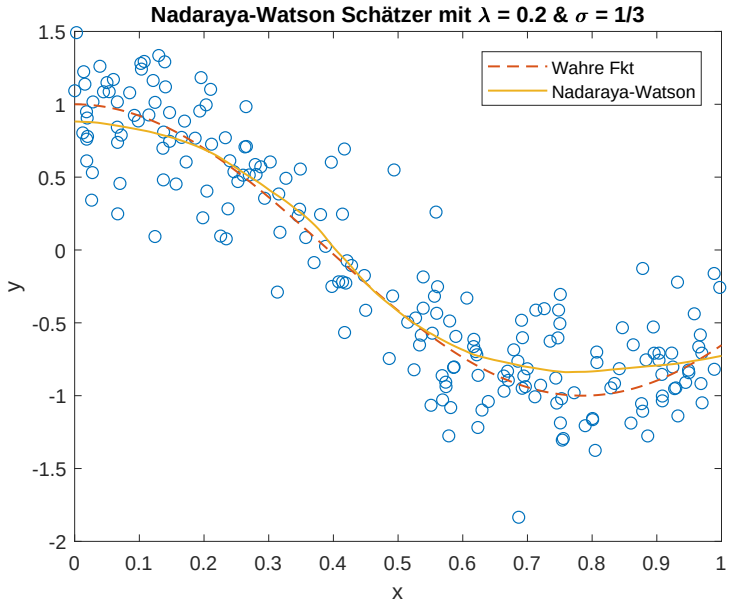
Berechnung

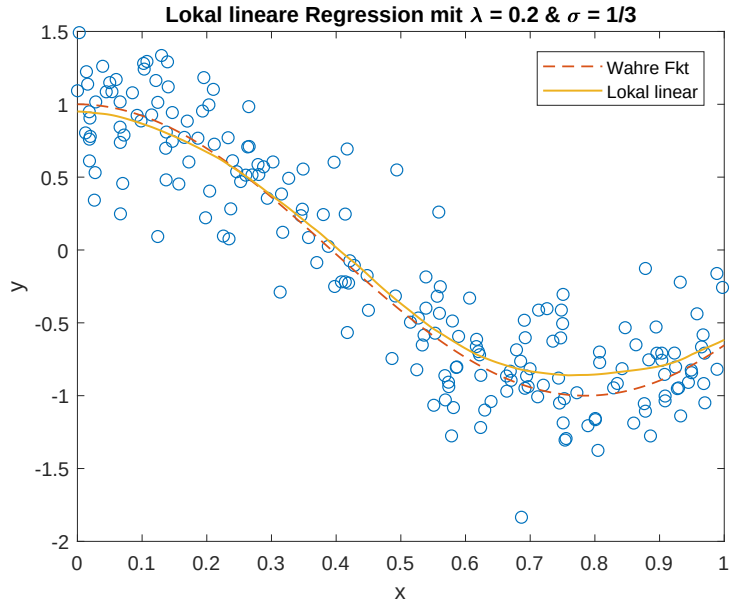
- Bestimme Koeffizienten $\hat{\alpha}(x_0), \hat{\beta}(x_0)$ für gegebenes x_0
- Lösen separates gewichtetes kleinste Quadrate Problem an jedem Zielpunkt x_0:

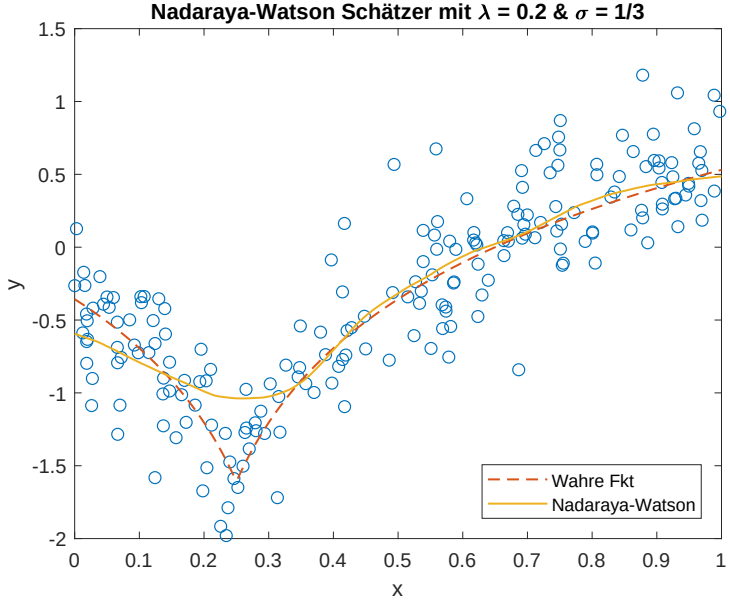
$$\min_{\alpha(x_0),\beta(x_0)} \sum_{i=1}^{N} K_\lambda(x_0, x_i) \Big[y_i - \alpha(x_0) - \beta(x_0) x_i \Big]^2$$

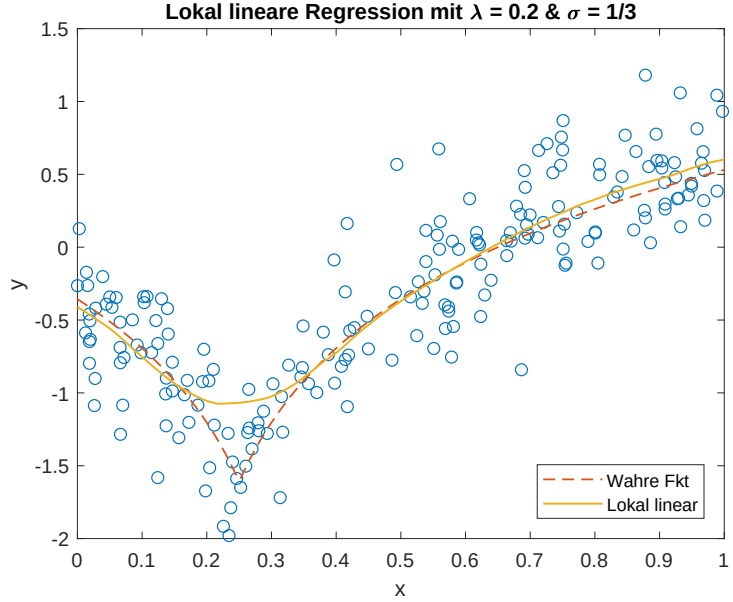
  $\Rightarrow$ Schätzung: $\hat{f}(x_0) = \hat{\alpha}(x_0) + \hat{\beta}(x_0) x_0$
- Passen gesamtes lineares Modell auf Daten in der Region an, aber verwenden es nur, um Anpassung am Punkt x_0 zu berechnen
- Verbesserung des Bias am Rand

Nadaraya-Watson Durchschnitt vs. Lokal lineare Regression









Bias-Korrektur

- Definiere:
  - Vektorwertige Funktion $b(x)^T = (1, x)$
  - B: $N \times 2$ *Regressionsmatrix* mit i-ter Zeile $b(x_i)^T$
  - $W(x_0)$: $N \times N$ Diagonalmatrix mit i-tem Diagonaleintrag $K_\lambda(x_0, x_i)$

Bias-Korrektur

- Definiere:
  - Vektorwertige Funktion $b(x)^T = (1, x)$
  - B: $N \times 2$ *Regressionsmatrix* mit i-ter Zeile $b(x_i)^T$
  - $W(x_0)$: $N \times N$ Diagonalmatrix mit i-tem Diagonaleintrag $K_\lambda(x_0, x_i)$
- Schätzung:

$$\hat{f}(x_0) \quad = \quad b(x_0)^T \left(B^T W(x_0) B \right)^{-1} B^T W(x_0) y$$

Bias-Korrektur

- Definiere:
  - Vektorwertige Funktion $b(x)^T = (1, x)$
  - B: $N \times 2$ *Regressionsmatrix* mit i-ter Zeile $b(x_i)^T$
  - $W(x_0)$: $N \times N$ Diagonalmatrix mit i-tem Diagonaleintrag $K_\lambda(x_0, x_i)$
- Schätzung:

$$\begin{aligned} \hat{f}(x_0) &= b(x_0)^T \left(B^T W(x_0) B\right)^{-1} B^T W(x_0) y \\ &= \sum_{i=1}^{N} l_i(x_0) y_i \end{aligned}$$

$\rightarrow$ Gewichte $l_i(x_0)$ kombinieren Kerne $K_\lambda(x_0, x_i)$ und kleinste Quadrate-Operationen

Bias-Korrektur

Taylorentwicklung von f um x_0:

$$E\left[\hat{f}(x_0)\right] = \sum_{i=1}^{N} l_i(x_0) f(x_i)$$

Bias-Korrektur

Taylorentwicklung von f um x_0:

$$\begin{aligned} E\left[\hat{f}(x_0)\right] &= \sum_{i=1}^{N} l_i(x_0) f(x_i) \\ &= f(x_0) \underbrace{\sum_{i=1}^{N} l_i(x_0)}_{=1} + f'(x_0) \underbrace{\sum_{i=1}^{N} (x_i - x_0) l_i(x_0)}_{=0} \\ &\quad + \frac{f''(x_0)}{2} \sum_{i=1}^{N} (x_i - x_0)^2 l_i(x_0) + R \end{aligned}$$

Bias-Korrektur

Taylorentwicklung von f um x_0:

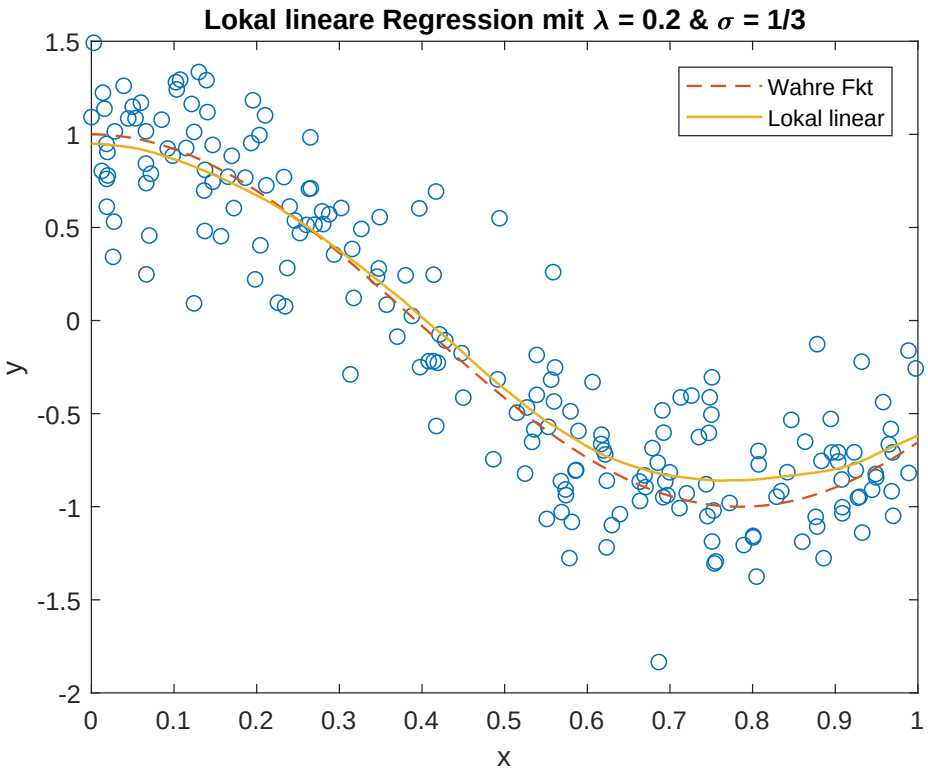
$$\begin{aligned} E\left[\hat{f}(x_0)\right] &= \sum_{i=1}^{N} l_i(x_0) f(x_i) \\ &= f(x_0) \underbrace{\sum_{i=1}^{N} l_i(x_0)}_{=1} + f'(x_0) \underbrace{\sum_{i=1}^{N} (x_i - x_0) l_i(x_0)}_{=0} \\ &\quad + \frac{f''(x_0)}{2} \sum_{i=1}^{N} (x_i - x_0)^2 l_i(x_0) + R \end{aligned}$$

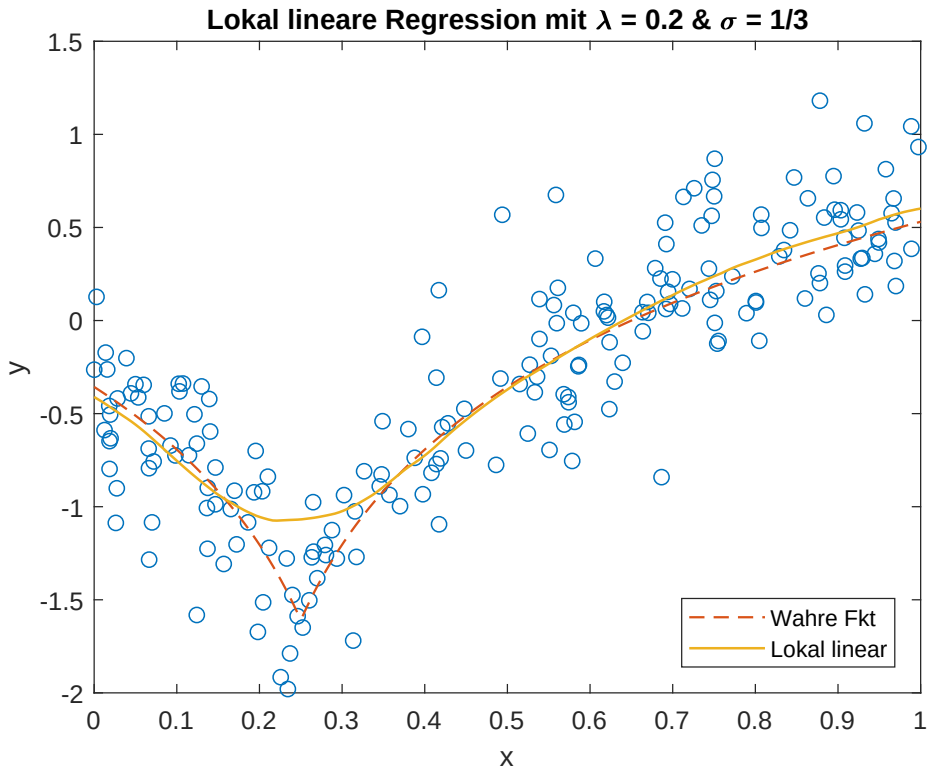
Bias:

$$E[\hat{f}(x_0)] - f(x_0) = \frac{f''(x_0)}{2} \sum_{i=1}^{N} (x_i - x_0)^2 l_i(x_0) + R$$

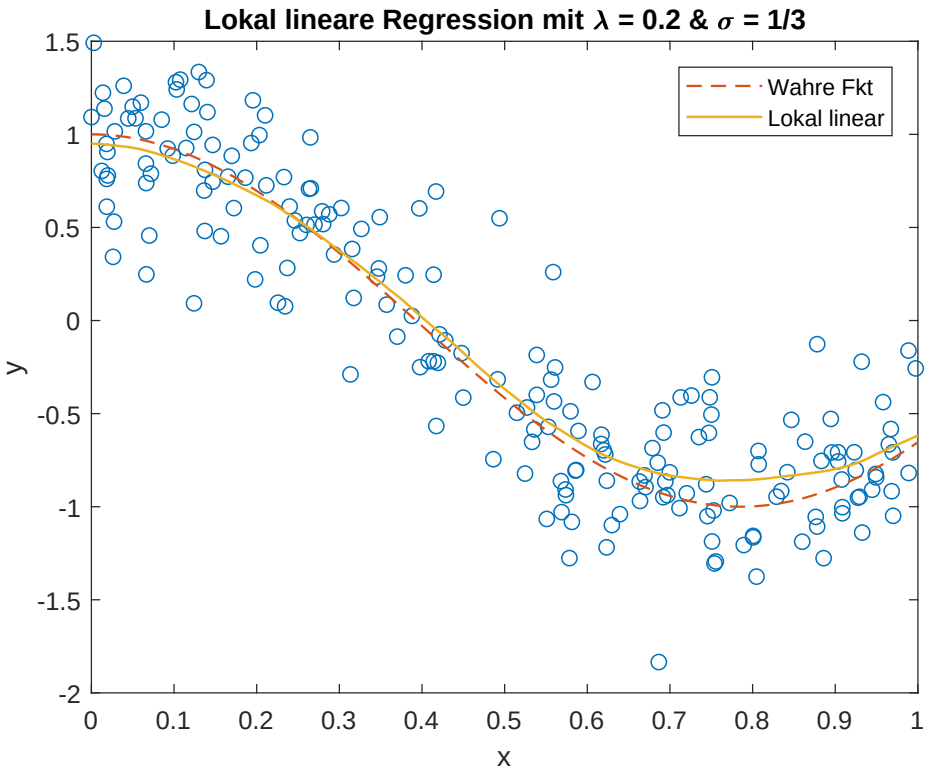
$\Rightarrow$ Nur abhängig von quadratischen Termen und Termen höherer Ordnung

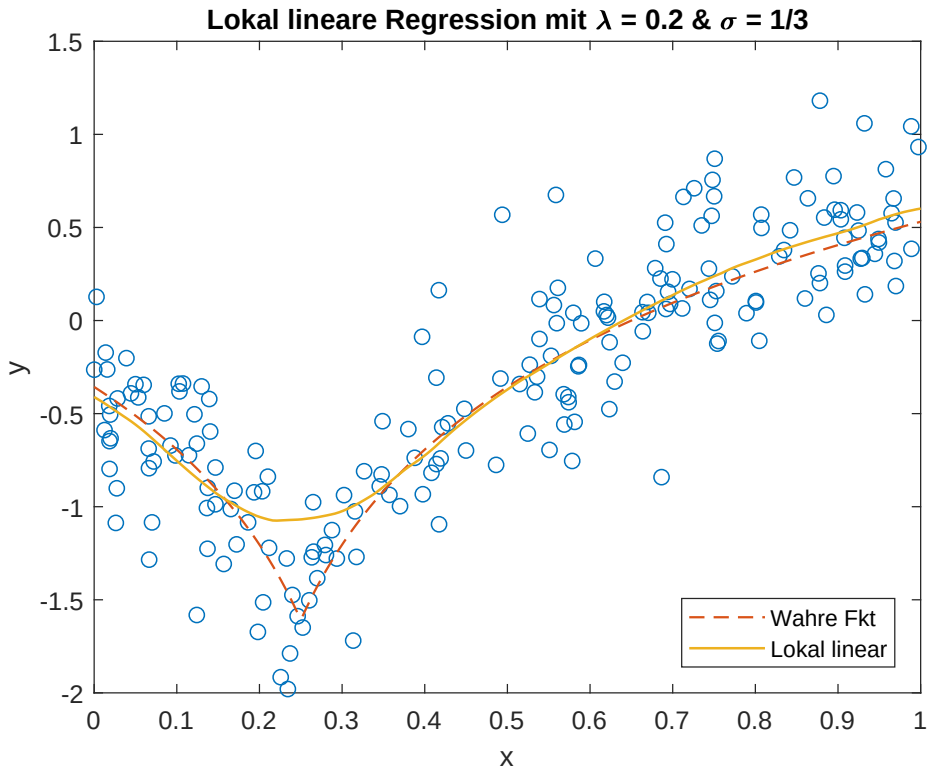
Nachteil lokal lineare Regression





Nachteil lokal lineare Regression





Lokal lineare Anpassungen oft verzerrt in gekrümmten Bereichen der wahren Funktion: *trimming the hills / filling the valleys*

Übersicht

Idee und Berechnung

- Gewichtete lokale Anpassung eines Polynoms vorgegebenen Grades $d > 1$ an jedem Punkt x_0

Idee und Berechnung

- Gewichtete lokale Anpassung eines Polynoms vorgegebenen Grades $d > 1$ an jedem Punkt x_0
- Bestimme Koeffizienten $\hat{\alpha}(x_0), \hat{\beta}_1(x_0), \ldots, \hat{\beta}_d(x_0)$ für gegebenes x_0
  $\rightarrow$ Löse gewichtetes kleinste Quadrate Problem

$$\min_{\alpha(x_0),\beta_j(x_0),j=1,\ldots,d} \sum_{i=1}^{N} K_\lambda(x_0, x_i) \left[y_i - \alpha(x_0) - \sum_{j=1}^{d} \beta_j(x_0) x_i^j \right]^2$$

$\Rightarrow$ Schätzung:

$$\hat{f}(x_0) = \hat{\alpha}(x_0) + \sum_{j=1}^{d} \hat{\beta}_j(x_0) x_0^j$$

Idee und Berechnung

- Gewichtete lokale Anpassung eines Polynoms vorgegebenen Grades $d > 1$ an jedem Punkt x_0
- Bestimme Koeffizienten $\hat{\alpha}(x_0), \hat{\beta}_1(x_0), \ldots, \hat{\beta}_d(x_0)$ für gegebenes x_0
  $\rightarrow$ Löse gewichtetes kleinste Quadrate Problem

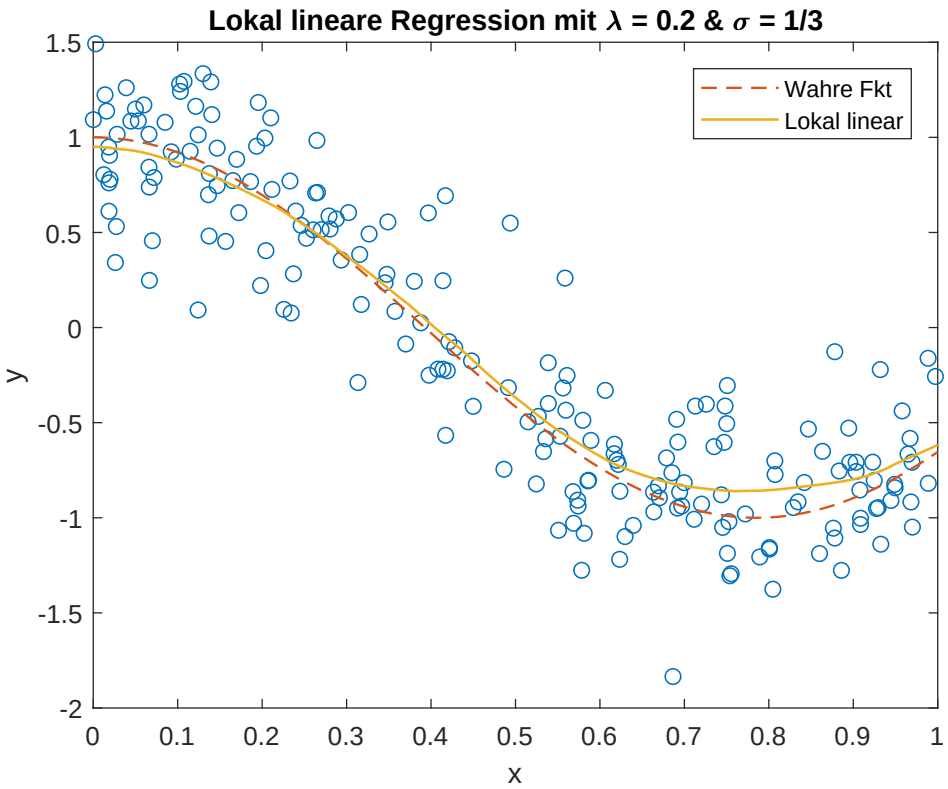
$$\min_{\alpha(x_0),\beta_j(x_0),j=1,\ldots,d} \sum_{i=1}^{N} K_\lambda(x_0, x_i) \left[y_i - \alpha(x_0) - \sum_{j=1}^{d} \beta_j(x_0) x_i^j \right]^2$$

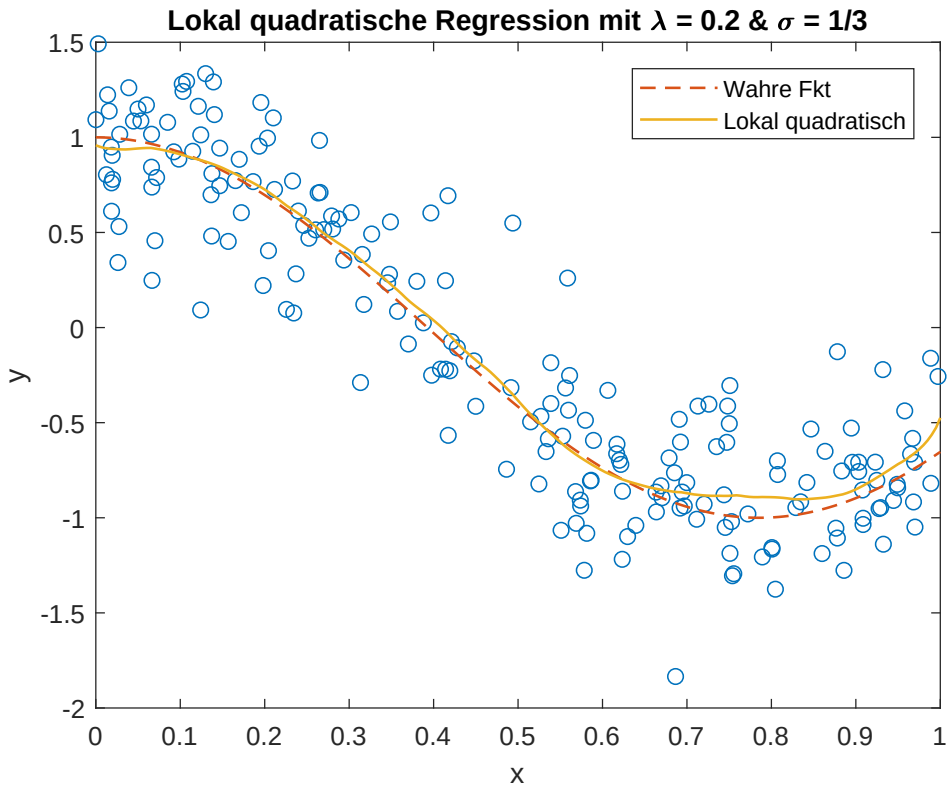
  $\Rightarrow$ Schätzung:

$$\hat{f}(x_0) = \hat{\alpha}(x_0) + \sum_{j=1}^{d} \hat{\beta}_j(x_0) x_0^j$$

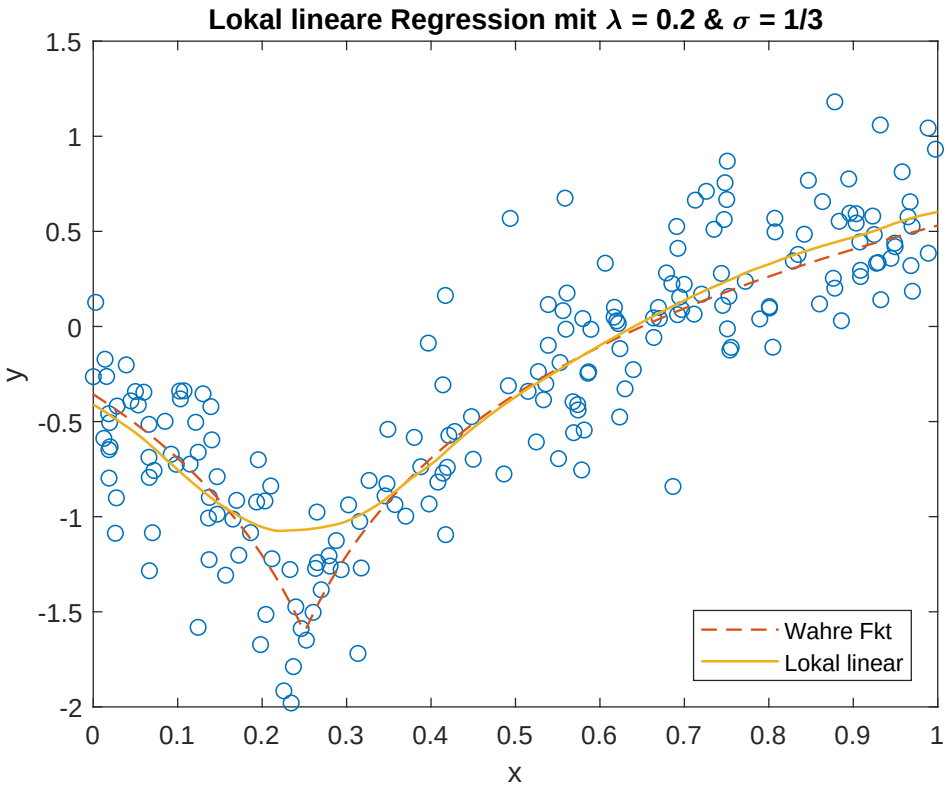
- Lokal quadratische Regression kann Bias im Inneren korrigieren
  $\rightarrow$ Taylorentwicklung: Bias enthält nur Komponenten vom Grad $d + 1$

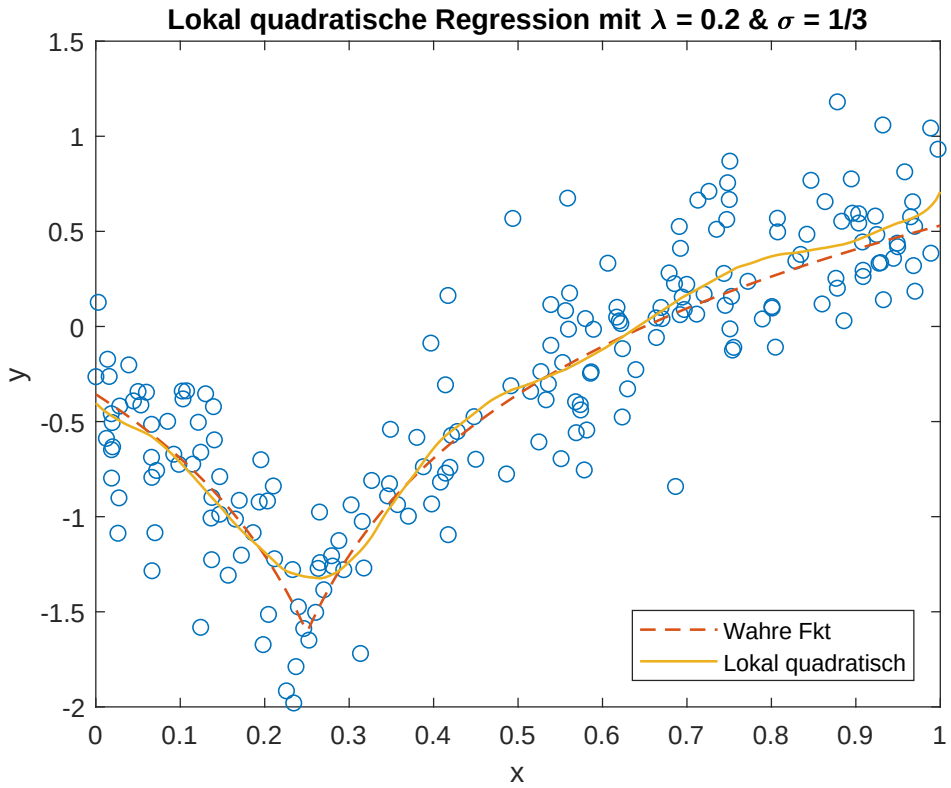
Lokal lineare vs. Lokal quadratische Regression



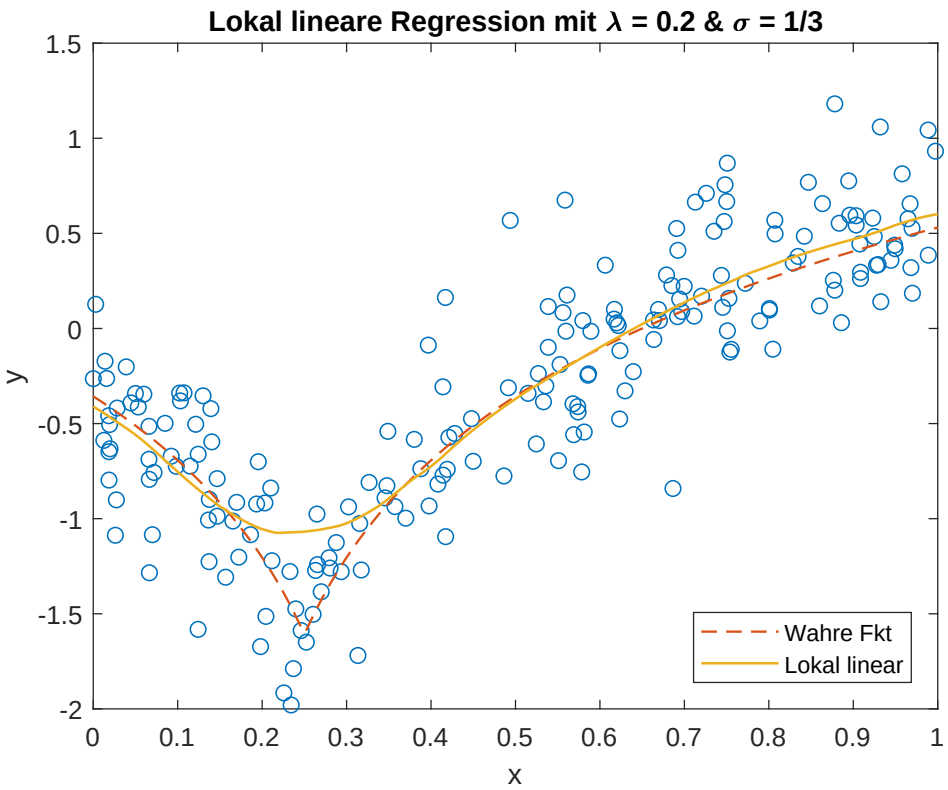


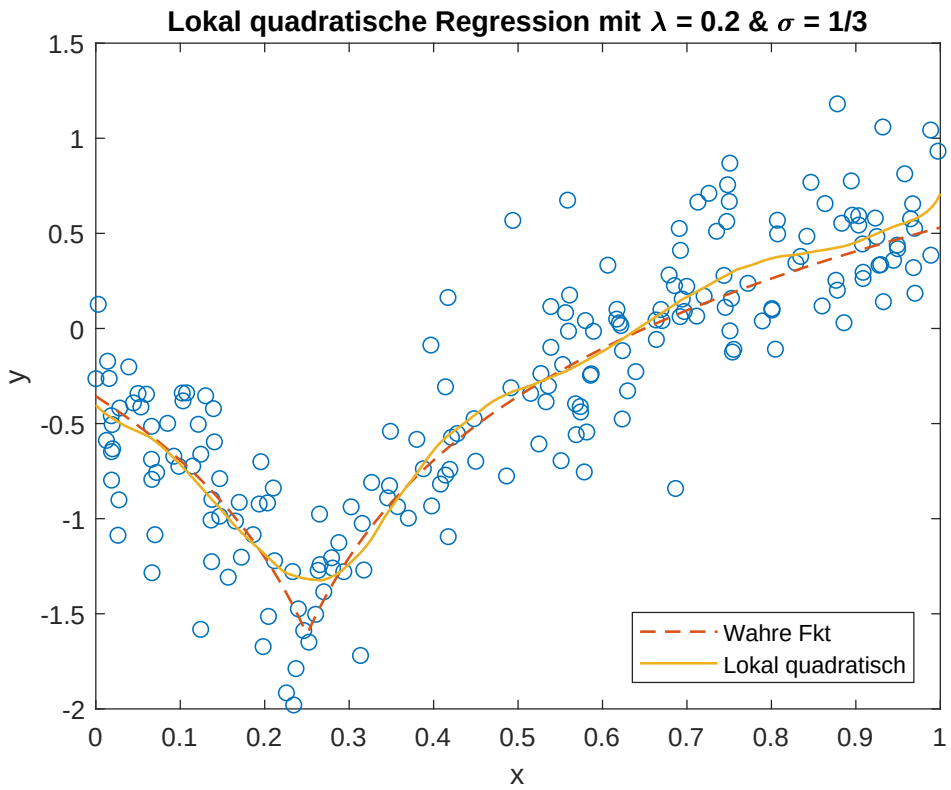
Lokal lineare vs. Lokal quadratische Regression





Lokal lineare vs. Lokal quadratische Regression



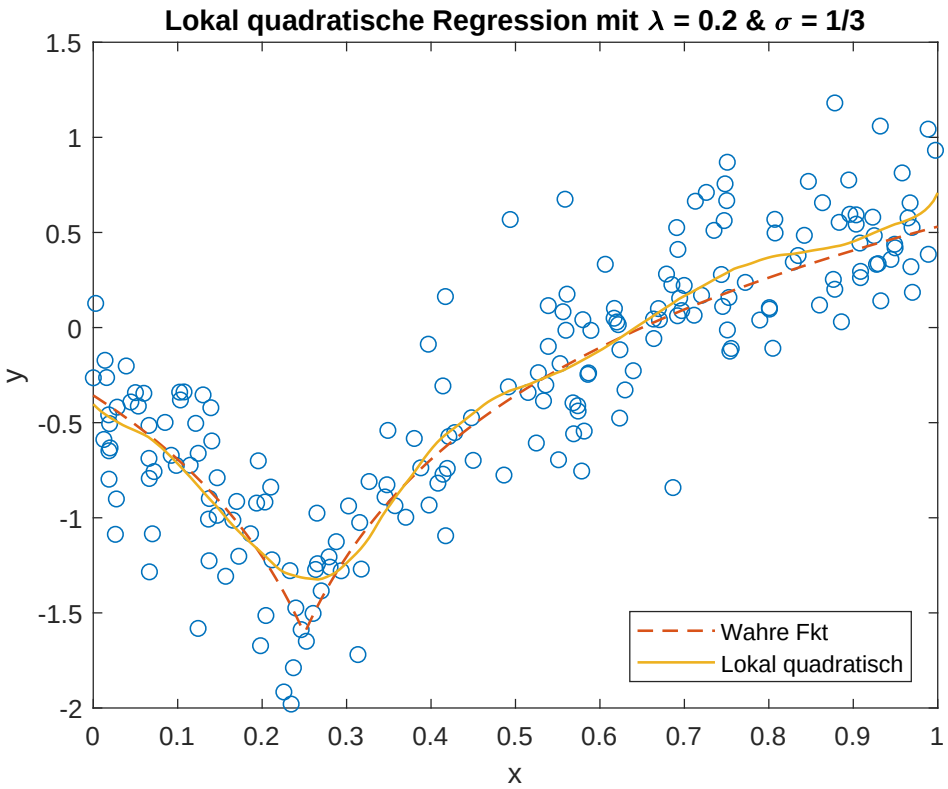


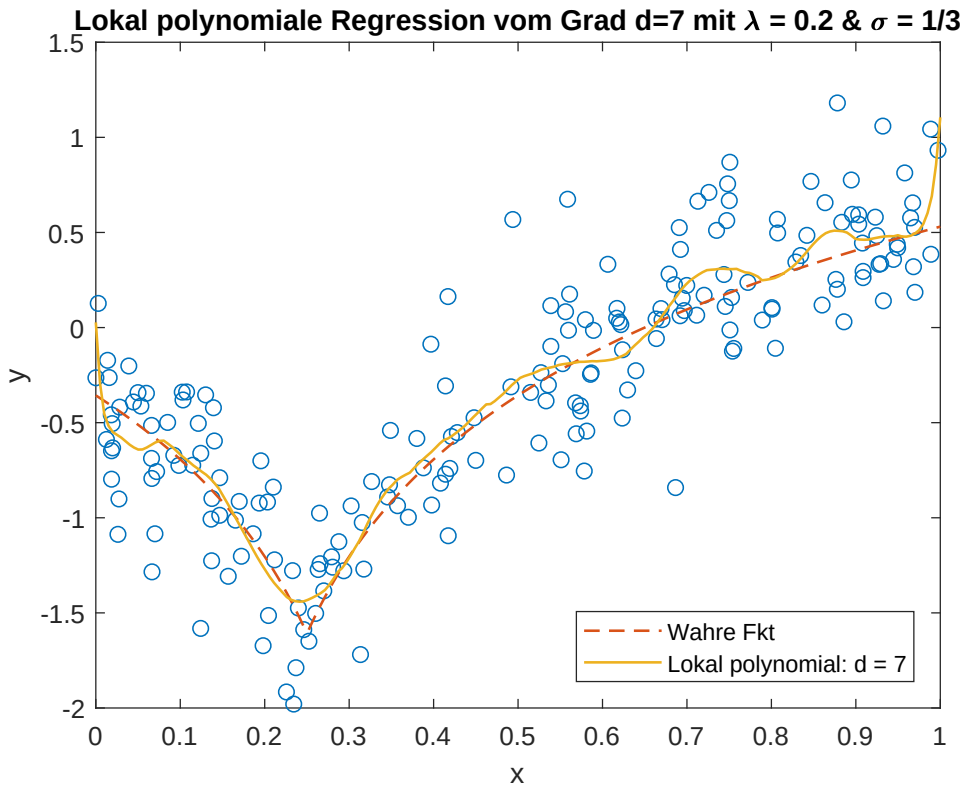
Preis für Bias-Korrektur: erhöhte Varianz

$$Var(\hat{f}(x_0)) = \sigma^2 ||l(x_0)||^2$$

$\rightarrow ||l(x_0)||^2$ steigt mit d: *Bias-Varianz Tradeoff*

Lokal quadratische Regression vs. Polynomgrad 7





Übersicht

Parameter λ

Breite des Kerns kontrolliert Größe der lokalen Nachbarschaft

- k-Nächste-Nachbarn: $\lambda =$ Anzahl k der nächsten Nachbarn
- Gauss-Kern: $\lambda =$ Standardabweichung σ
- Epanechnikov-/Tricube-Kern: $\lambda =$ Radius der Support-Region

Parameter λ

Breite des Kerns kontrolliert Größe der lokalen Nachbarschaft

- k-Nächste-Nachbarn: $\lambda =$ Anzahl k der nächsten Nachbarn
- Gauss-Kern: $\lambda =$ Standardabweichung σ
- Epanechnikov-/Tricube-Kern: $\lambda =$ Radius der Support-Region

Bias-Varianz Tradeoff, wenn wir λ ändern:

$$MSE(x_0) = Bias^2 + Var + \sigma^2$$

$$Bias(\hat{f}(x_0)) \sim O(\lambda^2)$$
$$Var(\hat{f}(x_0)) \sim O\left(\frac{1}{n\lambda^d}\right)$$

Bias-Varianz Tradeoff

Umgebung schmal, d.h. λ klein:

- $\hat{f}(x_0)$ Durchschnitt über kleine Anzahl von y_i nah an x_0
  $\rightarrow$ Varianz relativ groß
- Bias eher klein, da jedes $E(y_i) = f(x_i)$ nah an $f(x_0)$ sein sollte
  $\rightarrow$ Kleiner systematischer Fehler

Bias-Varianz Tradeoff

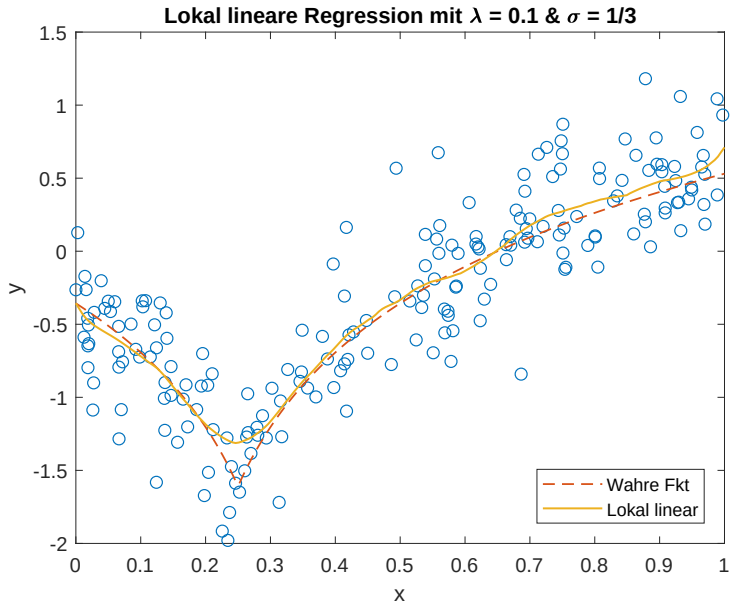
Umgebung schmal, d.h. λ klein:

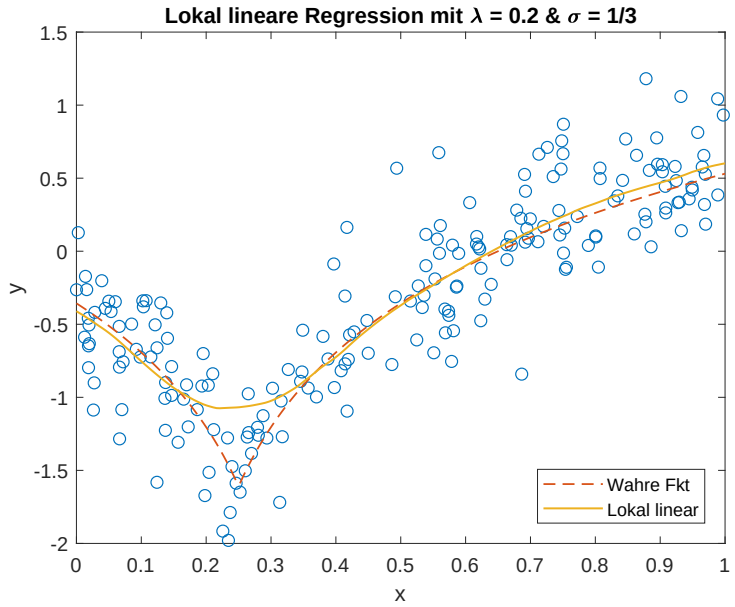
- $\hat{f}(x_0)$ Durchschnitt über kleine Anzahl von y_i nah an x_0
  $\rightarrow$ Varianz relativ groß
- Bias eher klein, da jedes $E(y_i) = f(x_i)$ nah an $f(x_0)$ sein sollte
  $\rightarrow$ Kleiner systematischer Fehler

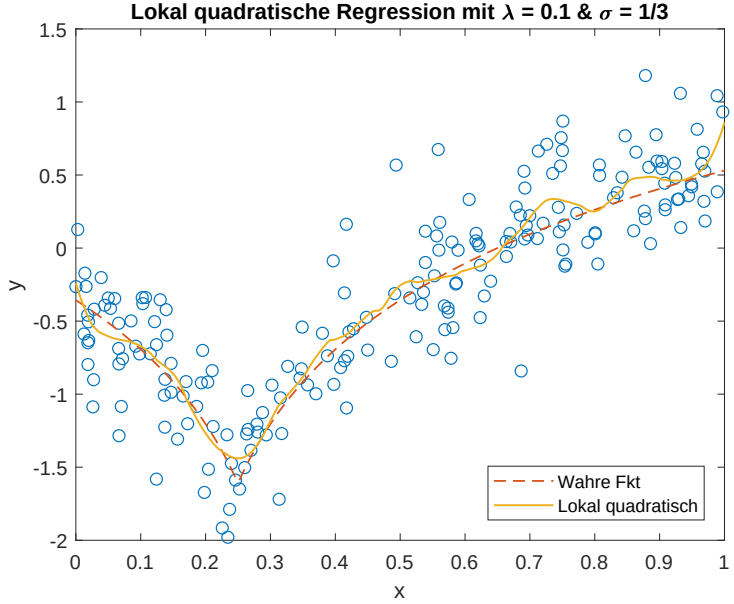
Umgebung breit, d.h. λ groß:

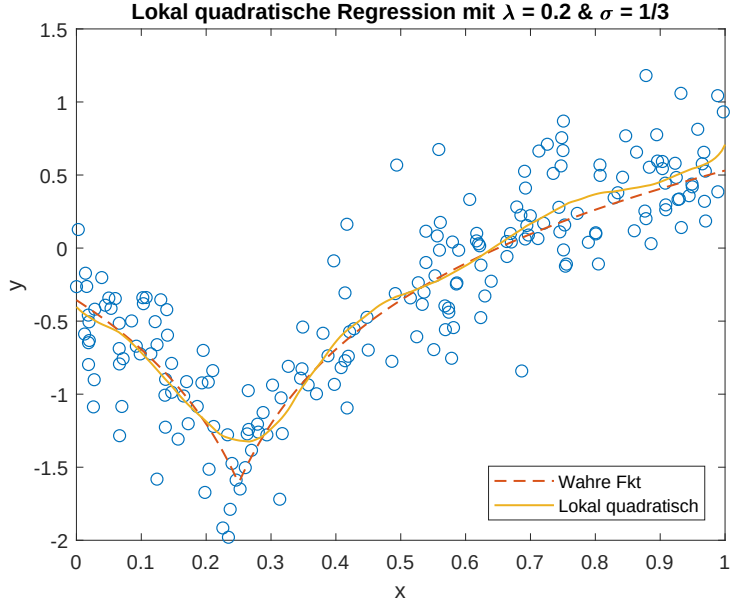
- Mittel über viele Beobachtungen,
  $\rightarrow Var(\hat{f}(x_0))$ klein relativ zu Varianz eines beliebigen y_i
- Bias höher: verwenden Beobachtungen x_i weiter entfernt von x_0
  $\rightarrow$ keine Garantie, dass $f(x_i)$ nah an $f(x_0)$ liegen wird. d.h. möglicherweise großer systematischer Fehler

Vergleich verschiedene λ









Fazit

- k-Nächste-Nachbarn problematisch: unstetige/unebene Schätzung
- Nadaraya-Watson: unterschiedliche Gewichte basierend auf Distanz
  $\rightarrow$ Glatte Schätzung, aber Probleme an Rändern: Bias

Fazit

- k-Nächste-Nachbarn problematisch: unstetige/unebene Schätzung
- Nadaraya-Watson: unterschiedliche Gewichte basierend auf Distanz
  $\rightarrow$ Glatte Schätzung, aber Probleme an Rändern: Bias
- Lokal linear: Bias am Rand kann zu erster Ordnung korrigiert werden
  $\rightarrow$ Aber: Probleme in gekrümmten Bereichen

Fazit

- k-Nächste-Nachbarn problematisch: unstetige/unebene Schätzung
- Nadaraya-Watson: unterschiedliche Gewichte basierend auf Distanz
  $\rightarrow$ Glatte Schätzung, aber Probleme an Rändern: Bias
- Lokal linear: Bias am Rand kann zu erster Ordnung korrigiert werden
  $\rightarrow$ Aber: Probleme in gekrümmten Bereichen
- Lokal quadratisch: Bias im Inneren aufgrund von Krümmungen kann korrigiert werden
  $\rightarrow$ Aber: erhöhte Varianz

Fazit

- k-Nächste-Nachbarn problematisch: unstetige/unebene Schätzung
- Nadaraya-Watson: unterschiedliche Gewichte basierend auf Distanz
  $\rightarrow$ Glatte Schätzung, aber Probleme an Rändern: Bias
- Lokal linear: Bias am Rand kann zu erster Ordnung korrigiert werden
  $\rightarrow$ Aber: Probleme in gekrümmten Bereichen
- Lokal quadratisch: Bias im Inneren aufgrund von Krümmungen kann korrigiert werden
  $\rightarrow$ Aber: erhöhte Varianz
- Wahl von λ: Bias-Varianz Tradeoff
  $\rightarrow$ Finde geeigneten Kompromiss, z.B. mit Cross-Validation

Literatur


Trevor Hastie, Robert Tibshirani, Jerome Friedman, *The Elements of Statistical Learning: Data Mining, Interference, and Prediction*, Second Edition, Springer Series in Statistics, 2008


Ludwig Fahrmeir, Thomas Kneib, Stefan Lang, *Regression: Modelle, Methoden und Anwendungen*, 2. Auflage, Springer, 2007, 2009


Gints Jekabsons, *Locally Weighted Polynomials toolbox for Matlab/Octave*, Version 2.2