

Gauß'sche Prozesse

Dozenten:

Prof. Dr.-Ing. J. Marius Zöllner

Dr.-Ing. Stefan Ulbrich



Forschungszentrum Karlsruhe
in der Helmholtz-Gemeinschaft



Universität Karlsruhe (TH)
Forschungsuniversität • gegründet 1825

■ Motivation

- Regression
- Probabilistische Regression

■ Gauß'sche Prozesse

- Definition
- Verwendung zur Regression
- Lernen

■ Anwendungsbeispiele

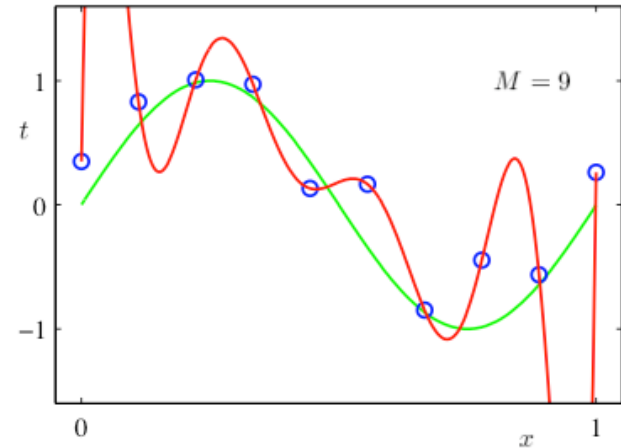
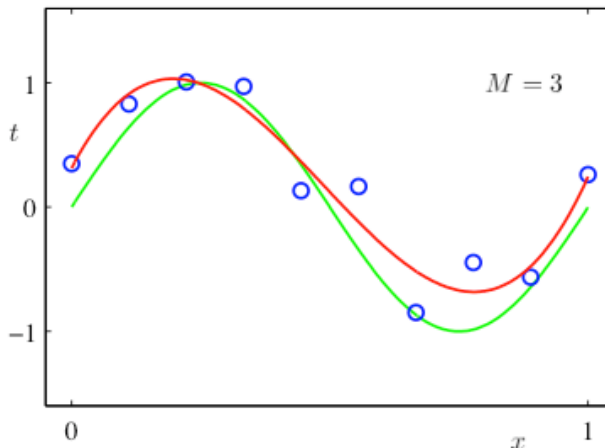
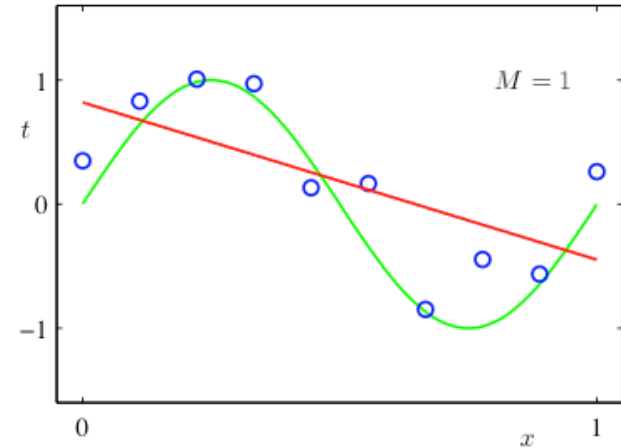
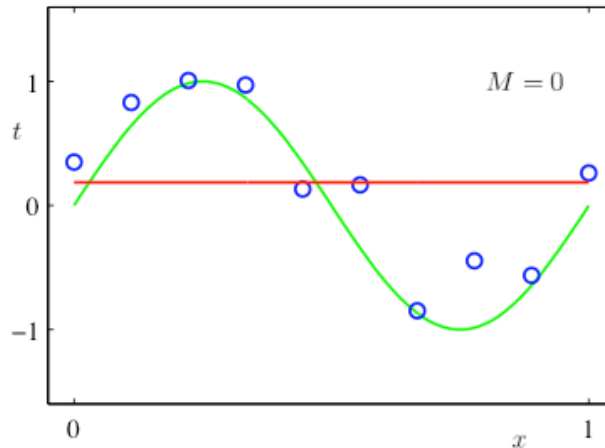
MOTIVATION

Regression: Problemstellung

- “Hier sind ein paar Datenpunkte. Von welcher Funktion wurden sie erzeugt?”
 - “Keine Ahnung.”
- “Oh. Okay. Hm, ob dieser Punkt wohl auch zu der Funktion gehört?”
 - “Keine Ahnung.”

Regression: Beispiele

- Ziel: Vorhersage einer kontinuierlichen Funktion

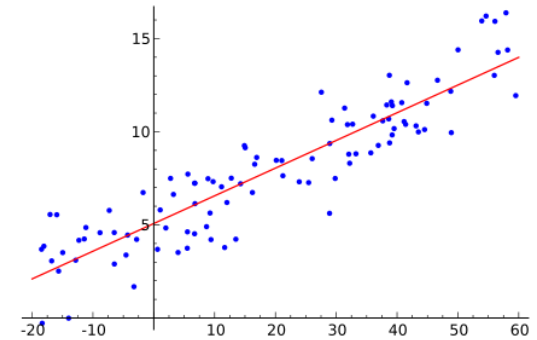


[1]

- **Klassifikation** sucht i. A. Entscheidungsfunktion $y(\mathbf{x})$, z.B. für 2 Klassen
 - $y(\mathbf{x}) > 0$ für alle Daten in Klasse A
 - $y(\mathbf{x}) < 0$ für alle Daten in Klasse B
- **Regression** sucht nach Funktion, die Daten möglichst gut beschreibt
 - Geg.: Trainingsdaten $\{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_n, y_n)\}$ mit Funktionswerten y_i
 - Ges.: Funktion $y : \mathbb{R}^d \rightarrow \mathbb{R}$
$$\mathbf{x} \mapsto y(\mathbf{x})$$
 - wobei die Hypothese möglichst gut zu Trainingsdaten passt, aber auch generalisiert!

Lineare Regression

- Klassische Fragestellung: Lineare Regression
 - Suchen einer „passenden“ Geraden in den Daten

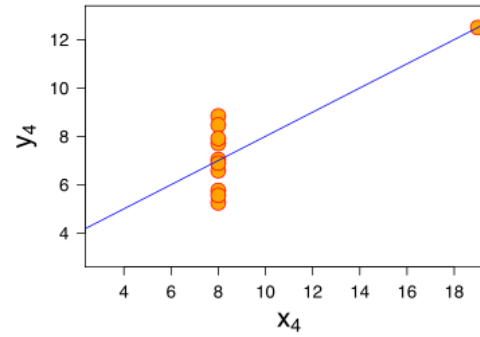
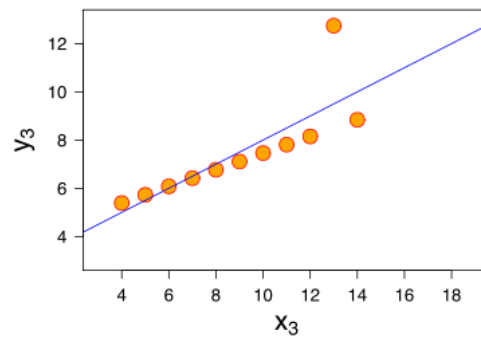
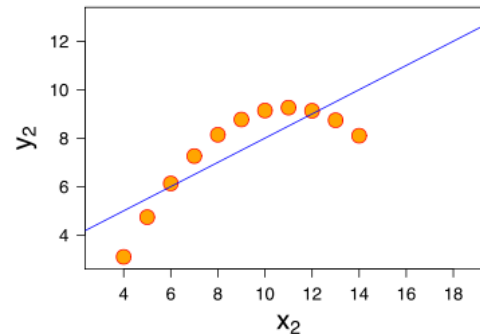
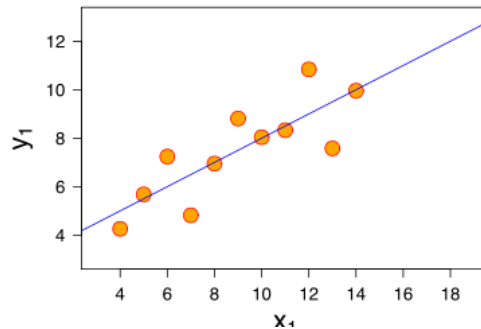


- Typische Lösung: $f(\mathbf{x}) = \mathbf{x}^T \mathbf{w}$
 - \mathbf{x} : Eingaben
 - \mathbf{w} : Gewichtung der einzelnen Eingaben
 - Beobachtungen: $y = f(\mathbf{x}) + \varepsilon$ (evtl. mit Gauß'schem Rauschen)
- Lösung z.B. mittels kleinste Quadrate-Methode (Matrixnotation)
 - $X \cdot \mathbf{w} = \mathbf{y} \Rightarrow \mathbf{w} = (X^T X)^{-1} X^T \mathbf{y}$, $X = (\mathbf{x}_1, \dots, \mathbf{x}_n)$, $\mathbf{y} = (y_1, \dots, y_n)$
 - Wobei $(X^T X)^{-1} X^T$ (Moore-Penrose) Pseudoinverse

Bewertung lineare Regression

■ Lineare Regression hat Vor- und Nachteile

- analytisch lösbar
- unterschiedlichste Daten können zu gleichem Ergebnis führen



[Anscombe's quartet
Abb. aus Wikipedia]

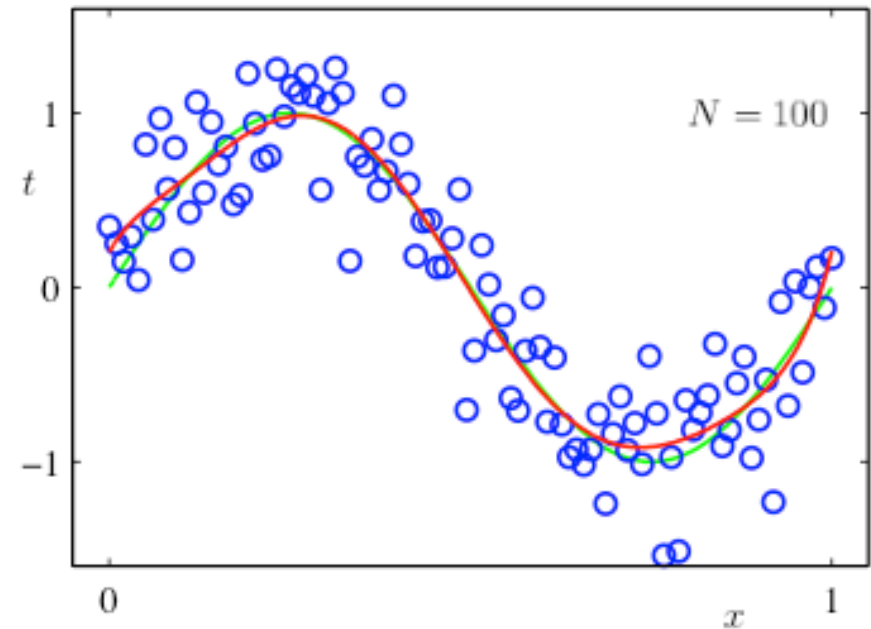
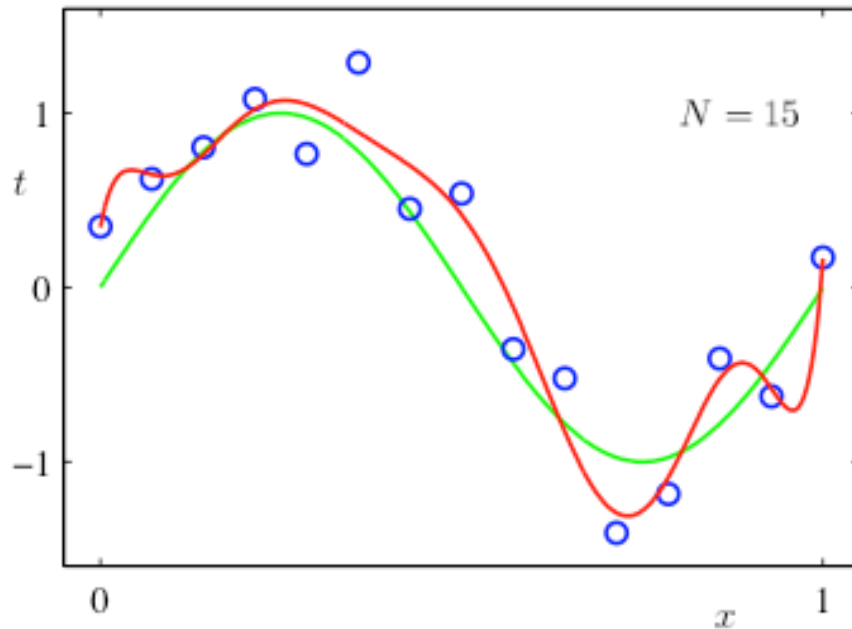
- Einfache lineare Regression: $f(\mathbf{x}) = \mathbf{x}^T \mathbf{w}$
- Erweiterung durch Projektion der Daten unter Verwendung von M festen Basisfunktionen $\phi(\mathbf{x})$ sodass gilt:

$$\begin{aligned} y(\mathbf{x}) &= \phi(\mathbf{x})^T \mathbf{w} \\ &= \sum_{i=0}^M \phi_i(\mathbf{x}) w_i \end{aligned}$$

- z.B. für skalare Eingaben: $\phi(x) = (1, x, x^2, x^3, \dots)^T$

Overfitting bei Regression

■ Polynom mit Grad 9 für Eingangsbeispiel



[1]

■ N: Anzahl der Trainingsbeispiele


Probleme mit üblicher Regression

- Overfitting
- Numerische Instabilität
 - Tabelle zeigt Gewichtskoeffizienten bei Polynomen unterschiedlichen Grades für das Beispiel


	$M = 0$	$M = 1$	$M = 6$	$M = 9$
w_0^*	0.19	0.82	0.31	0.35
w_1^*		-1.27	7.99	232.37
w_2^*			-25.43	-5321.83
w_3^*			17.37	48568.31
w_4^*				-231639.30
w_5^*				640042.26
w_6^*				-1061800.52
w_7^*				1042400.18
w_8^*				-557682.99
w_9^*				125201.43

[1]

- Interpretation der kleinsten Quadrate als Maximum Likelihood-Schätzung
- Annahmen:
 - Zielwerte y generiert durch zusätzliches Rauschen auf Funktionsschätzung: $y = f(\mathbf{x}, \mathbf{w}) + \varepsilon$
 - Rauschen ist Normalverteilt: $p(y|\mathbf{x}, \mathbf{w}, \beta) = \mathcal{N}(y|f(\mathbf{x}, \mathbf{w}), \beta^{-1})$



Erwartungswert



Varianz
- Ansatz: Maximiere bedingte Wahrscheinlichkeit $p(\mathbf{y}|\mathbf{X}, \mathbf{w}, \beta)$ bzgl. \mathbf{w}, β
- Lösung: $\mathbf{w}_{ML} = (\Phi^T \Phi)^{-1} \Phi^T \mathbf{y}$ mit $\Phi = (\phi(\mathbf{x}_1), \dots, \phi(\mathbf{x}_n))^T$
 - Analog zu kleinste Quadrate bei linearer Regression

Regression – ein anderer Blickwinkel

- zwei unterschiedliche Ansätze für Regression
- Klasse in Betracht gezogener Funktionen einschränken
 - z.B. lineare Funktionen der Eingabewerte
 - Nutzen kleinste Quadrate oder ML-Schätzungen
- Bayes'sche Modellierung
 - Jeder Funktion a priori Wahrscheinlichkeit zuweisen: $p(f)$
 - Berechne MAP-Schätzung:

Wahrscheinlichkeit
der Beobachtung

a priori der Funktion

$$p(f|\mathbf{y}, X) = \frac{p(\mathbf{y}|X, f)p(f)}{p(\mathbf{y}|X)}$$

Wahrscheinlichkeit
der Beobachtung

a priori der Funktion

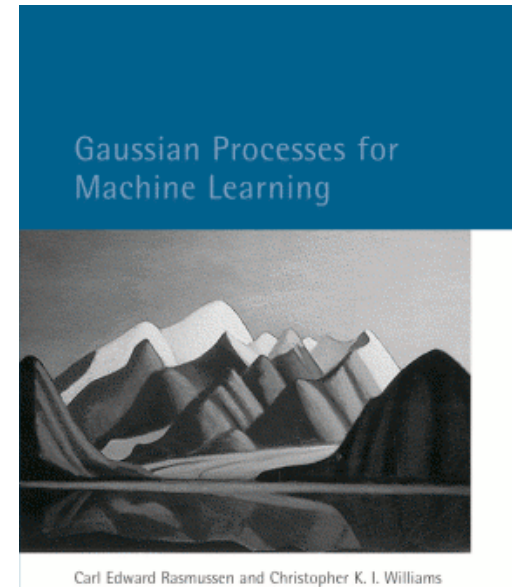
Normalisierung

- f : mögliche Funktionen, X : Eingabedaten, \mathbf{y} : Ausgaben

GAUSS'SCHE PROZESSE

Vorbemerkungen

- bekannt durch C.E. Rasmussen und C.K.I. Williams



<http://www.gaussianprocess.org/gpml/>

- Buch frei verfügbar (im Folgenden referenziert als [2])
- zentrale GP-Homepage: <http://www.gaussianprocess.org/>
- Vorlesung angelehnt an [2], Kapitel 1 & 2

■ Multivariate Gauß-Verteilung (Normalverteilung) $\mathbf{x} \sim \mathcal{N}(\boldsymbol{\mu}, \Sigma)$

■ Mittelwert

$$\boldsymbol{\mu} = [E[X_1], E[X_2], \dots, E[X_k]]$$

■ Kovarianzmatrix

$$\Sigma = [\text{Cov}[X_i, X_j]], \quad i = 1, \dots, k \wedge j = 1, \dots, k$$

$$\Sigma_{ij} = \text{cov}(X_i, X_j) = E[(X_i - \mu_i)(X_j - \mu_j)]$$

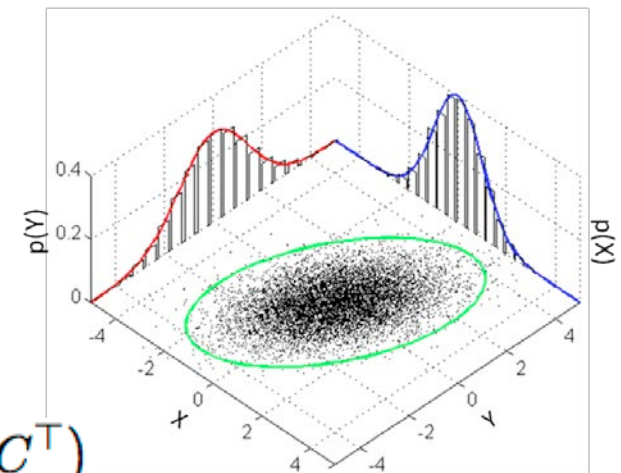
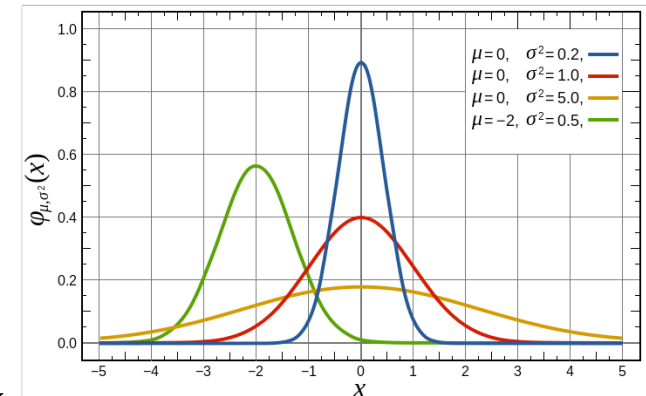
■ Gemeinsam Gauß-verteilte X, Y

$$\begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix} \sim \mathcal{N} \left(\begin{bmatrix} \boldsymbol{\mu}_x \\ \boldsymbol{\mu}_y \end{bmatrix}, \begin{bmatrix} A & C \\ C^\top & B \end{bmatrix} \right)$$

■ Marginal $\mathbf{x} \sim \mathcal{N}(\boldsymbol{\mu}_x, A)$

■ Bedingte Verteilung

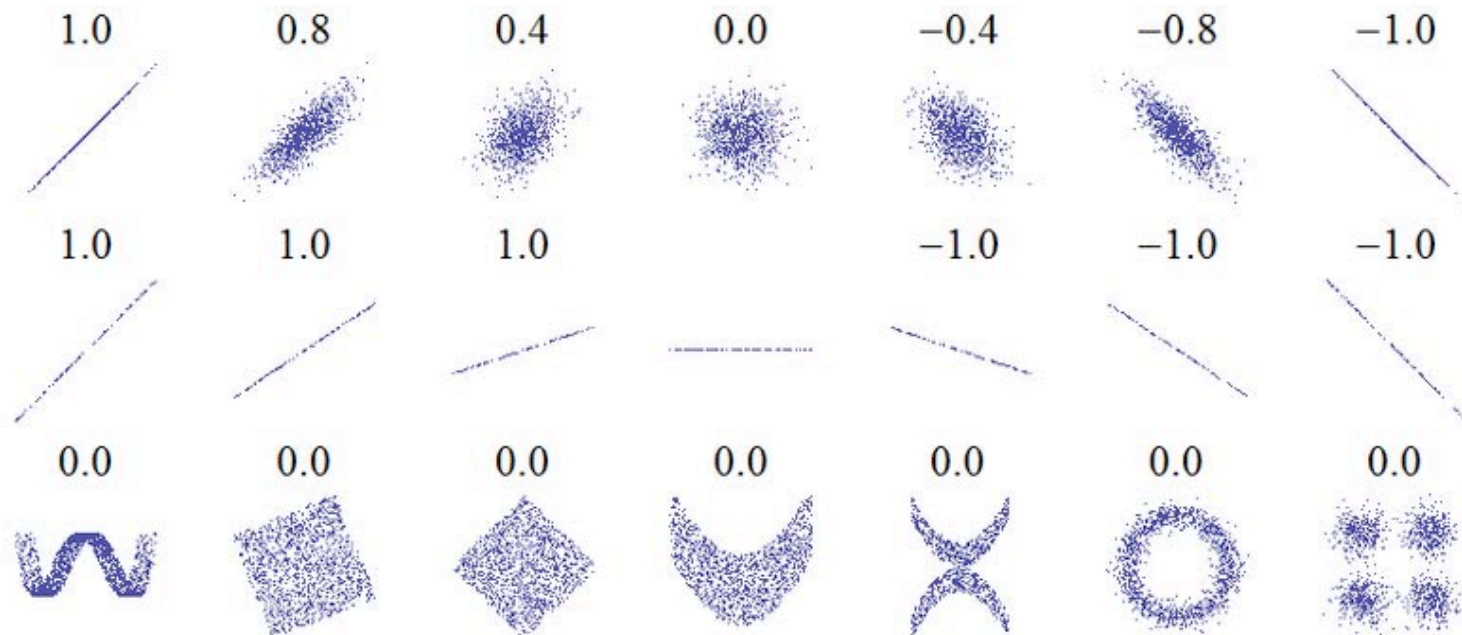
$$\mathbf{x}|\mathbf{y} \sim \mathcal{N}(\boldsymbol{\mu}_x + CB^{-1}(\mathbf{y} - \boldsymbol{\mu}_y), A - CB^{-1}C^\top)$$



- Maß für linearen Zusammenhang zwischen X und Y $[-1, 1]$

$$Kor(X, Y) = \frac{\text{Cov}(X, Y)}{\sigma(X)\sigma(Y)} \in [-1, 1]$$

- Beispiele:



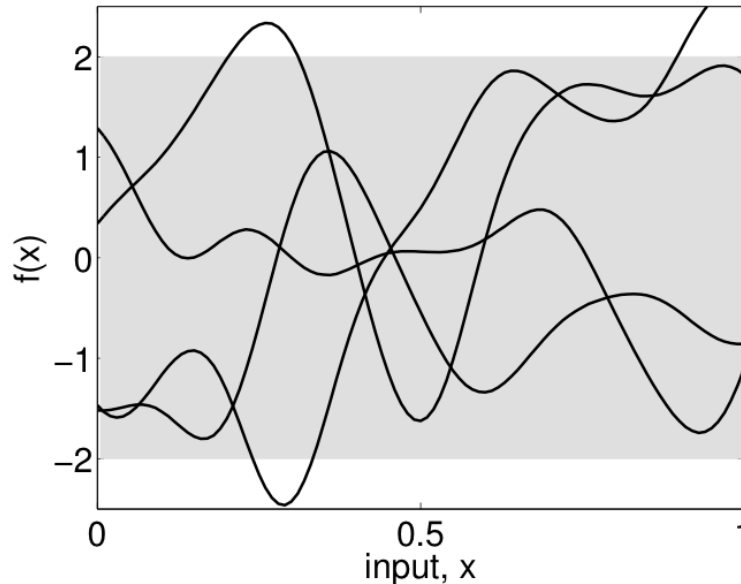
Warum Gauß'sche Prozesse?

- aus ML 1 bekannt: Induktiver Bias (Annahmen über das zu lernende) ist notwendig für Generalisierung
- GPs bieten eine angenehme Art, diese Art von a-priori-Wissen für Funktionen auszudrücken
 - Präferenz-Bias statt Restriktions-Bias
 - Explizite Modellierung des Induktiven Bias
- GPs haben große Bandbreite von Anwendungen
 - Regression
 - Klassifikation
 - Optimierung

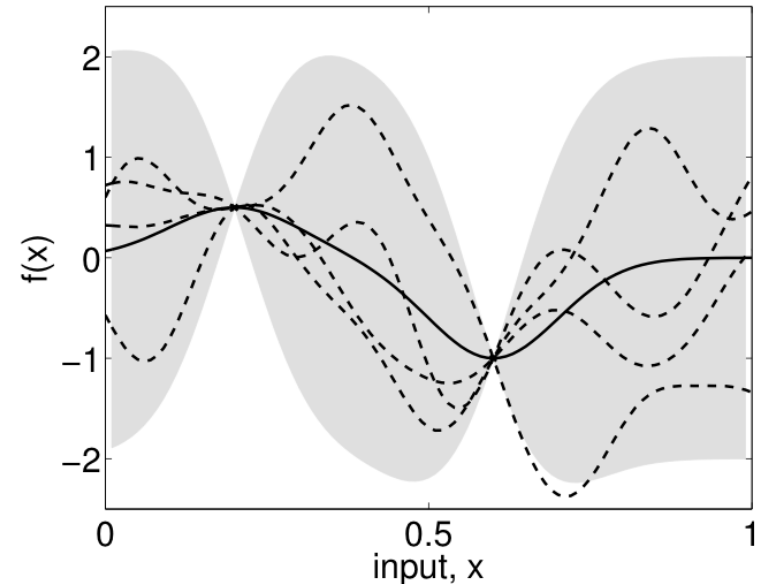
- Gauß'scher Prozess (Verallg. der Gauß-Verteilung)
 - beschreibt Eigenschaften von Funktionen
 - Vorstellung von Funktionen als (unendlich) lange Vektoren, deren Einträge x_i den Funktionswert $f(x_i)$ enthalten
- GPs werden benutzt um a priori-Wissen über Funktionen auszudrücken
 - GP kann z.B. beschreiben, dass Funktionswert sich nur „sanft“ ändert (glatte Funktion)
- Problem mit dieser Vorstellung: Wie geht man mit unendlich vielen Punkten um?
 - nur Eigenschaften an endlicher Menge von Punkten benötigt => Inferenz in GPs liefert gleiche Antwort, auch wenn unendlich viele Punkte ignoriert werden

Beispiel zur Illustration

Eindimensionales Regressionsproblem



(a), prior



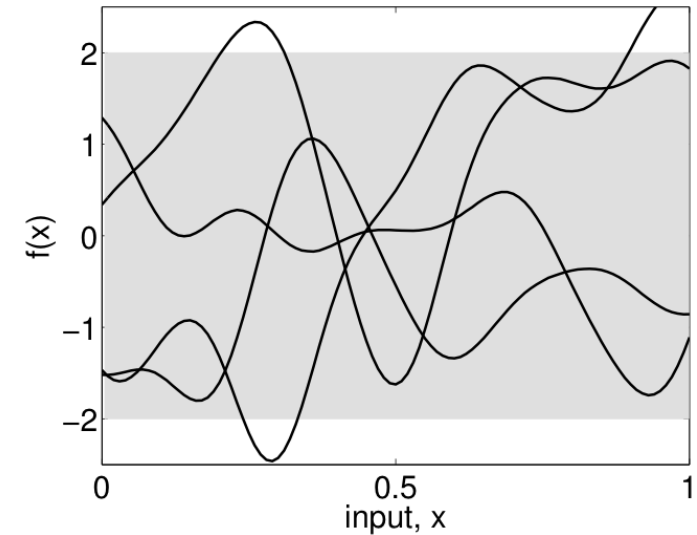
[2]

(b), posterior

- Links: 4 zufällig gewählte mögliche Funktionen aus a-priori-Verteilung eines GP
- Rechts: MAP mit zwei Messpunkten

Beispiele für a priori-Eigenschaften

- Repräsentation von Vorwissen über Funktionen
- Mittelwert einzelner Funktionen ungleich 0, aber Mittelwert von $f(x)$ für festes x ist 0 (im Bsp.)
- Bevorzugung glatter Funktionen, d.h. Funktionen können nicht zu schnell variieren (im Bsp.)
- *Glattheit* induziert von Kovarianzfunktion des GP
- Lernen in GPs: Hauptaufgabe ist das Finden geeigneter Eigenschaften der Kovarianzfunktionen



Formaler(er) Blick auf GPs

- Zwei Sichtweisen möglich:
- 1. „weight space view“
 - GP definiert Verteilung über den Parametern (Gewichten) der Funktionen
- 2. „function space view“
 - GP definiert Verteilung direkt über den Funktionen
- Äquivalente Ergebnisse mit beiden Sichtweisen

Definition Gauß'scher Prozess I

- Definition Gauß'scher Prozess (GP):
Ein GP ist eine Zusammenfassung von Zufallsvariablen, von denen eine beliebige endliche Menge zusammen einer gemeinsamen Gauß-Verteilung genügen
- Andere Formulierung:
Wahrscheinlichkeitsverteilung über Funktionen $y(x)$ sodass die Menge der Werte von $y(x)$ ausgewertet an einer beliebigen Menge von Punkten zusammen einer Gauß-Verteilung genügen.

Definition Gaußscher Prozess II

- Gauß-Verteilung für jedes x im Eingaberaum X , GP vollständig definiert durch
 - Erwartungsfunktion: $m(x) = \mathbb{E}[f(x)]$
 - Kovarianzfunktion: $k(x, x') = \mathbb{E}[(f(x) - m(x))(f(x') - m(x'))]$
 - Beschreibt Kovarianz von $f(x)$ an zwei beliebigen Stellen x, x'
 - Meistens: Erwartungswert wird als 0 angenommen ($m(\mathbf{x}) = 0$)
 \Rightarrow Vorwissen (prior, bias) in Kernel (Kovarianzfunktion) repräsentiert
- Notation: $f(\mathbf{x}) \sim \mathcal{GP}(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}'))$

- GP definiert als Zusammenfassung von Zufallsvariablen
→ impliziert Konsistenz
- Konsistenz hier:
 - wenn GP $(y_1, y_2) \sim \mathcal{N}(\mu, \Sigma)$ definiert, mit $\Sigma = \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix}$
 - dann muss auch gelten $y_1 \sim \mathcal{N}(\mu_1, \Sigma_{11})$
- Automatisch erfüllt durch direkte Wertebelegung der Kovarianzmatrix mit der Kovarianzfunktion
- Anders ausgedrückt:
Betrachtung einer größeren Variablenmenge ändert nicht die Verteilung der kleineren Menge

Beispiel für GP

■ Bayes'sches lineares Regressionsmodell: $f(\mathbf{x}) = \phi(\mathbf{x})^T \mathbf{w}$

■ Gauß'sche a priori: $\mathbf{w} \sim \mathcal{N}(\mathbf{0}, \Sigma_p)$

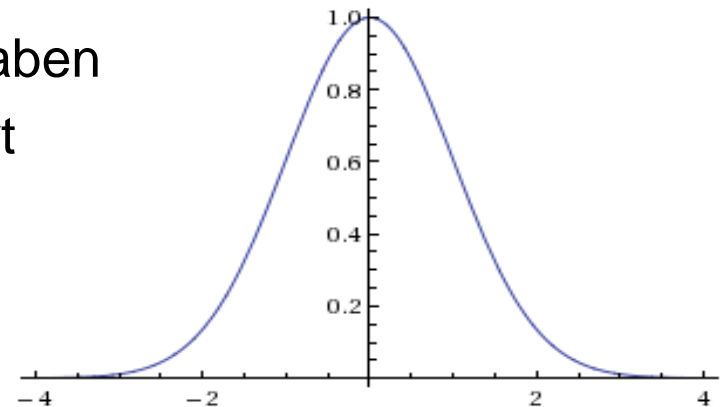
■ Mittelwert: $E[f(\mathbf{x})] = \phi(\mathbf{x})^T E[\mathbf{w}] = 0$

■ Kovarianz: $E[f(\mathbf{x})f(\mathbf{x}')] = \phi(\mathbf{x})^T E[\mathbf{w}\mathbf{w}^T] \phi(\mathbf{x}')$
 $= \phi(\mathbf{x})^T \Sigma_p \phi(\mathbf{x}')$

- Kovarianzfunktion definiert die Kovarianz zwischen Paaren von Zufallsvariablen
- Typische Kovarianzfunktion: „squared exponential“ (SE)

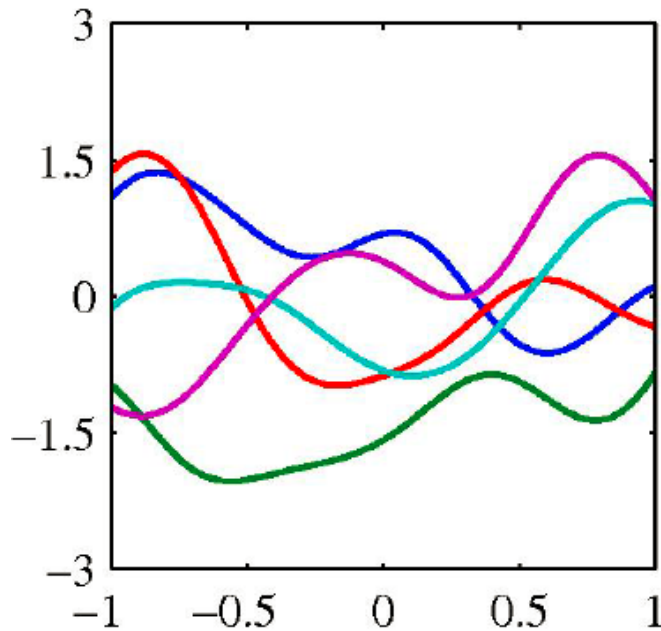
$$\text{cov}(f(\mathbf{x}_p), f(\mathbf{x}_q)) = k(\mathbf{x}_p, \mathbf{x}_q) = \exp \left\{ -\frac{1}{2} |\mathbf{x}_p - \mathbf{x}_q|^2 \right\}$$

- Kovarianz zwischen Ausgaben geschrieben als Funktion über Eingaben
- SE-Kovarianzfunktion korrespondiert zu Bayes'schem linearem Regressionsmodell mit unendlich vielen Basisfunktionen

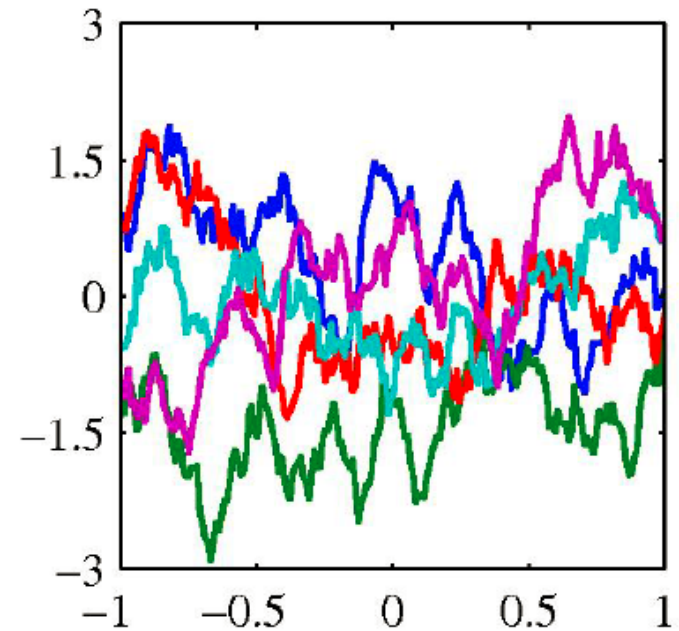


Beispiel: Kovarianzfunktionen

■ Zwei verschiedene Kovarianzfunktionen



$$k(x, y) = \exp \left(-\frac{(|x| - |y|)^2}{2} \right)$$



$$k(x, y) = \exp (-|x| - |y|)$$

- Wenn die Kovarianz nur von der Differenz von \mathbf{x}_p und \mathbf{x}_q abhängt, spricht man von einem **stationärem Kernel**
 - Beispiel SE

$$k(\mathbf{x}_p, \mathbf{x}_q) =: k(d) = \exp\left(-\frac{d^2}{2}\right), \quad \text{mit } d = \|\mathbf{x}_p - \mathbf{x}_q\|$$

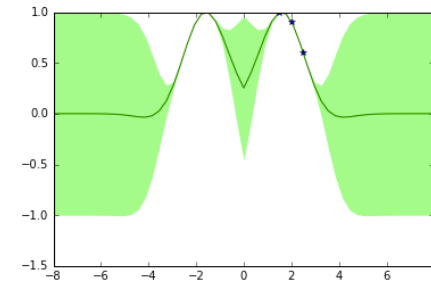
- Ansonsten von einem **nicht stationärem Kernel**
 - Beispiel: Linearer Kernel

$$k(\mathbf{x}_p, \mathbf{x}_q) = \mathbf{x}_p^t \cdot \mathbf{x}_q$$

■ Weitere Beispiele

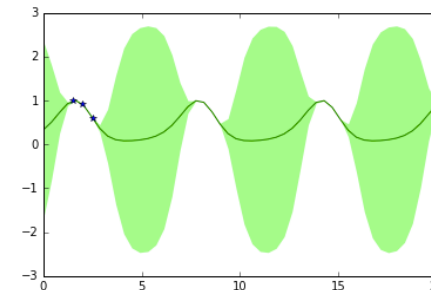
- Achsensymmetrische Kernel (nicht stationär)

$$k(x, y) = \exp \left(-\frac{(|x| - |y|)^2}{2} \right)$$



- Periodische Kernel (stationär)

$$k(d) = \exp(\cos(d))$$



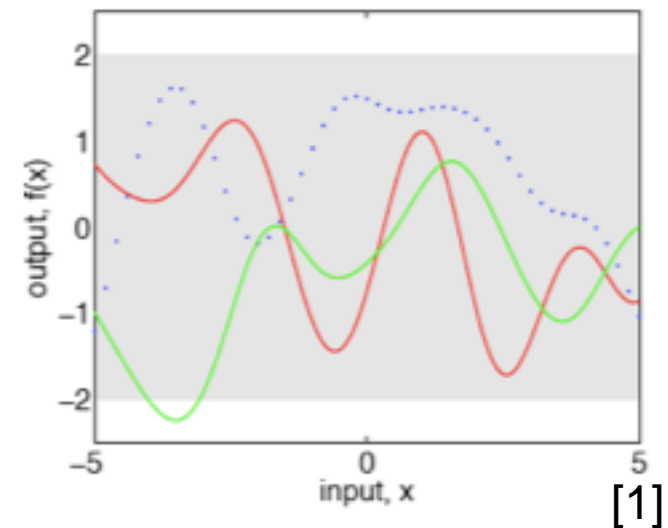
- Polynomielle Kernel (nicht stationär)

$$k(x, y) = \phi(x)^t \cdot \phi(y), \quad \phi(x) = (1, x, x^2, \dots)$$

GP - a priori-Verteilung über Funktionen (Sampling)

- Spezifikation/Wahl einer Kovarianzfunktion impliziert eine Verteilung über Funktionen
- => Beispiele aus der Verteilung über den Funktionen können gewählt und an (endlicher) Zahl von Punkten evaluiert werden
- Vorgehen:
 - Endliche Menge Punkte X_* wählen
 - Zugehörige Kovarianzmatrix (z.B. SE) elementweise aufschreiben: $K(X_*, X_*)$
 - Generiere Vektor aus Gauß-Funktion mit dieser Matrix als Kovarianz:

$$f_* \sim \mathcal{N}(\mathbf{0}, K(X_*, X_*))$$



- Annahme: Beobachtungen rauschfrei: $\{(\mathbf{x}_i, f_i) | i = 1, \dots, n\}$
- Multivariate Verteilung von Trainingsausgaben \mathbf{f} und Testausgaben \mathbf{f}_* gemäß a priori-Verteilung:

$$\begin{bmatrix} \mathbf{f} \\ \mathbf{f}_* \end{bmatrix} \sim \mathcal{N} \left(\mathbf{0}, \begin{pmatrix} K(X, X) & K(X, X_*) \\ K(X_*, X) & K(X_*, X_*) \end{pmatrix} \right)$$

- $K(X, X_*)$ enthält Kovarianzen für alle Paare von Trainings- und Testpunkten
- Zur Berechnung der a posteriori-Verteilung (nach Beob.)
 - a priori (oben) beschränken auf Funktionen, die mit Beobachtungen konsistent
 - Vorstellung: Funktionen mittels a priori generieren, inkonsistente verwerfen

■ Berechnung der a posteriori-Verteilung

- entspricht der multivariaten a priori-Gauß-Verteilung bedingt unter den Beobachtungen

$$p(\mathbf{f}_* | X_*, X, \mathbf{f}) \sim \mathcal{N}(\bar{\mathbf{f}}_*, \text{cov}(\mathbf{f}_*))$$

- Wobei

$$\bar{\mathbf{f}}_* = K(X_*, X) \cdot K(X, X)^{-1} \mathbf{f}$$

$$\text{cov}(\mathbf{f}_*) = K(X_*, X_*) - K(X_*, X) \cdot K(X, X)^{-1} \cdot K(X, X_*)$$

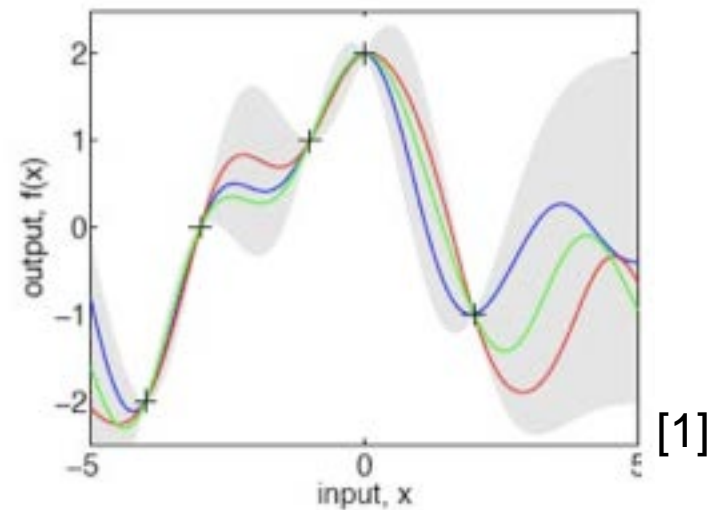
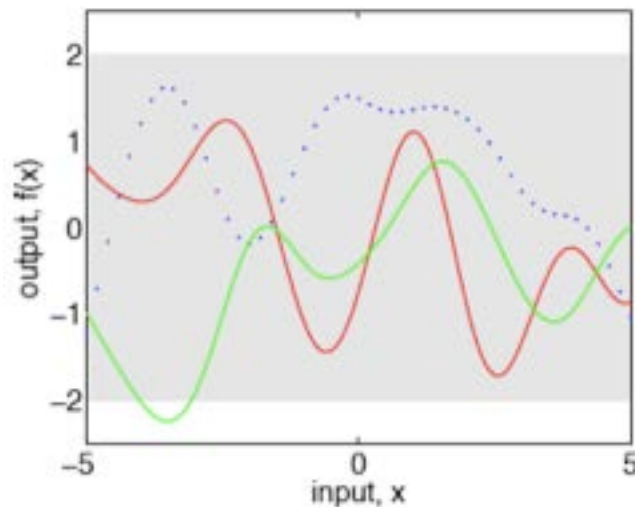
- Funktionswerte von \mathbf{f}_* (für die Testeingaben x_*) können aus der resultierenden Normalverteilung gesampelt werden

■ Anmerkung

- Für jedes neue Trainingsbeispiel wächst X (und y)!
 - **GPR ist instanzbasiertes Lernen**
- $K(X, X)^{-1}$ muss dann neu berechnet werden
- Matrizen können sehr groß werden und aufwändig zu berechnen
- Daher nur bedingt geeignet für inkrementelles Lernen
- Abhilfe
 - *Sparsification*: Ausdünnen der Matrix („Vergessen“ wenig relevanter Beispiele)

rauschfreie Prädiktion: Beispiel

- links: a priori-Verteilung
- rechts: a posteriori-Verteilung nach 5 rauschfreien Beobachtungen



Prädiktion bei verrauschten Beobachtungen

- Übliche Annahme: Rauschen in Beobachtungen

$$y = f(\mathbf{x}) + \epsilon \quad \epsilon \sim \mathcal{N}(0, \sigma_n^2)$$

- Veränderung der Kovarianz durch das Rauschen

- ausführlich: $\text{cov}(y_p, y_q) = k(\mathbf{x}_p, \mathbf{x}_q) + \sigma_n^2 \delta_{pq}$

- Kompakte Matrix-Notation: $\text{cov}(\mathbf{y}) = K(X, X) + \sigma_n^2 I$

- resultierende Verteilung der Beobachtungen und der Teststellen mit dieser a priori-Verteilung:

$$\begin{bmatrix} \mathbf{y} \\ \mathbf{f}_\star \end{bmatrix} \sim \mathcal{N} \left(\mathbf{0}, \begin{bmatrix} K(X, X) + \sigma_n^2 I & K(X, X_\star) \\ K(X_\star, X) & K(X_\star, X_\star) \end{bmatrix} \right)$$

- Berechnung der a posteriori-Verteilung entspricht im Prinzip dem unverrauschten Fall
 - entspricht der multivariaten a priori-Gauß-Verteilung bedingt unter den Beobachtungen

$$p(\mathbf{f}_* | X_*, X, \mathbf{f}) = \mathcal{N}(\bar{\mathbf{f}}_*, \text{cov}(\mathbf{f}_*))$$

- wobei

$$\begin{aligned}\bar{\mathbf{f}}_* &= E[\mathbf{f}_* | X, X_*, \mathbf{y}] = K(X_*, X) \cdot (K(X, X) + \sigma^2 I)^{-1} \mathbf{f} \\ \text{cov}(\mathbf{f}_*) &= K(X_*, X_*) - K(X_*, X) \cdot (K(X, X) + \sigma^2 I)^{-1} \cdot K(X, X_*)\end{aligned}$$

- Funktionswerte von \mathbf{f}_* (für die Testeingaben X_*) können wieder aus der resultierenden Normalverteilung gesampelt werden

■ Einige freie Parameter in den meisten Kovarianzfunktionen

- Beispiel: eindim. SE-Kovarianzftk. für verrauschte Daten

$$k_y(x_p, x_q) = \underline{\sigma_f^2} \exp \left\{ \frac{-(x_p - x_q)^2}{2 \cdot \underline{l^2}} \right\} + \underline{\sigma_n^2} \delta_{pq}$$

- Parameter: (l, σ_f, σ_n)

- Signal-Varianz: σ_f^2

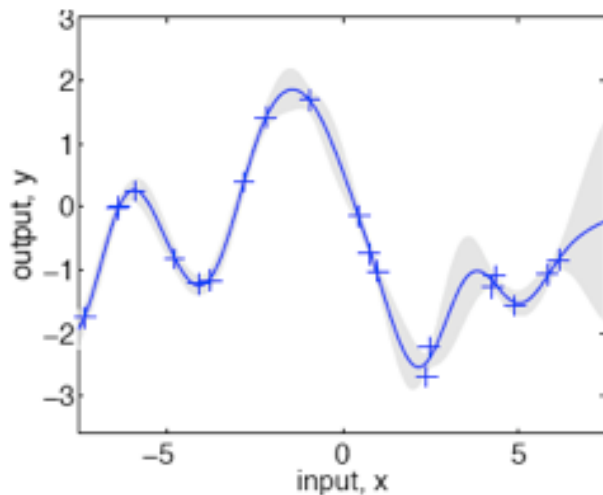
- „length scale“: l

- Rauschen in den Beobachtungen: σ_n^2

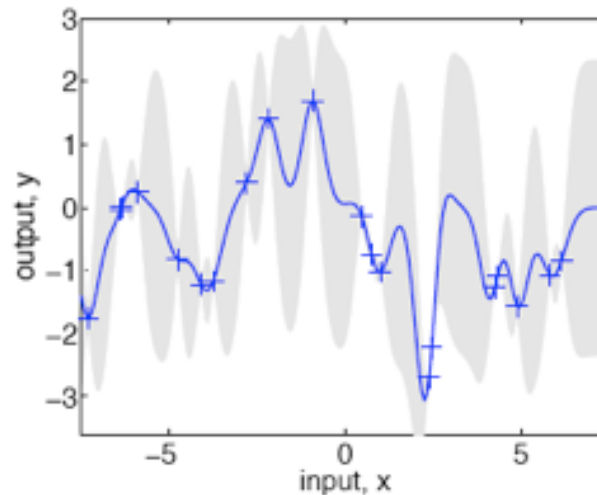
- Beispiele für verschiedene Werte der Hyperparameter
 - Length scale l variiert, andere Parameter optimiert

$$(l, \sigma_f, \sigma_n) =$$

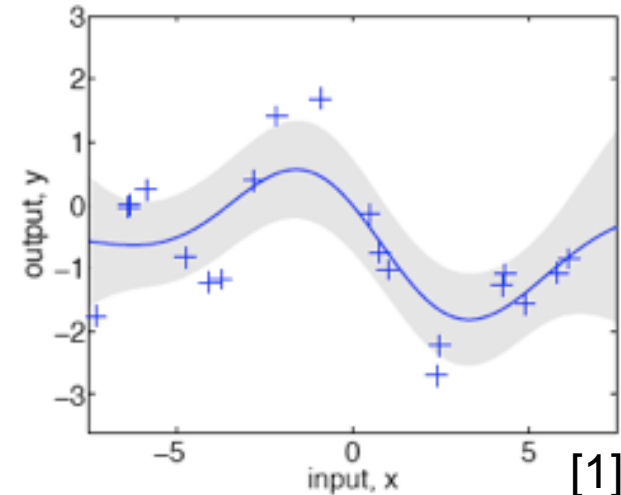
$$= (1, 1, 0.1)$$



$$= (0.3, 1.08, 0.00005)$$



$$= (3.0, 1.16, 0.89)$$



[1]

DEMONSTRATION

(Am Ende des PDF)

- Bayes'sche Modell-Auswahl
- Ziel: Verschiedene Parameter von GPs bestimmen/lernen
- Hierarchie von Parametern muss in Betracht gezogen werden
 - untere Ebene: \mathbf{w} , z.B. die Parameter eines linearen Modells
 - mittlere Ebene: Hyperparameter θ , z.B. die a priori-Verteilung von \mathbf{w} kontrollieren
 - obere Ebene: typischerweise eine diskrete Menge von Modellstrukturen \mathcal{H}_i
- Ansatz: Inferenz nacheinander auf den einzelnen Ebenen durchführen

Beispiel: Modellauswahl auf unterer Ebene

- a posteriori des Parameters \mathbf{w} mittels Bayes-Regel:

$$\begin{aligned} p(\mathbf{w}|\mathbf{y}, X, \theta, \mathcal{H}_i) &= \frac{p(\mathbf{y}|X, \mathbf{w}, \theta, \mathcal{H}_i)p(\mathbf{w}|X, \theta, \mathcal{H}_i)}{p(\mathbf{y}|X, \theta, \mathcal{H}_i)} \\ &= \frac{p(\mathbf{y}|X, \mathbf{w}, \mathcal{H}_i)p(\mathbf{w}|\theta, \mathcal{H}_i)}{p(\mathbf{y}|X, \theta, \mathcal{H}_i)} \end{aligned}$$

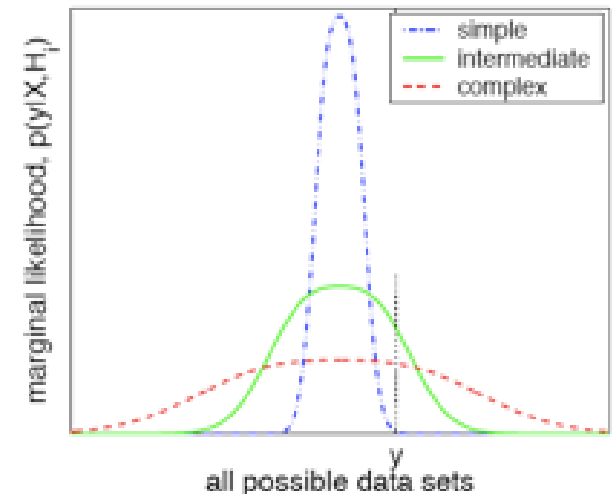
- mit

- Likelihood: $p(\mathbf{y}|X, \mathbf{w}, \mathcal{H}_i)$
- a priori des Parameters \mathbf{w} : $p(\mathbf{w}|\theta, \mathcal{H}_i)$
- Normalisierende Konstante im Nenner unabhängig von \mathbf{w} (marginalisierte Likelihood)

$$p(\mathbf{y}|X, \theta, \mathcal{H}_i) = \int p(\mathbf{y}|X, \mathbf{w}, \mathcal{H}_i)p(\mathbf{w}|\theta, \mathcal{H}_i)d\mathbf{w}$$

Bewertung der Bayesschen Modell-Auswahl

- Hauptunterschied zu nicht-Bayes'schen Verfahren: Marginalisierte Likelihood
- Automatischer Trade-off zwischen passendem Modell und Modellkomplexität
 - einfaches Modell: kann nur für geringe Menge der Zielwerte passen, daher hohe a posteriori, wenn das Modell gut passt
 - komplexes Modell: kann einen großen Bereich von möglichen Mengen von Zielwerten erklären, daher immer eine Beschränkung der a posteriori-Wahrscheinlichkeit



- Gauß'scher Prozess: Wahrscheinlichkeitsverteilung über Funktionen
- An jedem Punkt eine Gauß-Verteilung
- Vorwissen/Induktiver Bias steckt in der Kovarianzfunktion (Kernel)
- Hauptsächlichste Verwendung für Regression
 - Klassifikation auch möglich, aber erfordert zusätzlichen Aufwand

ANWENDUNGSBEISPIELE

■ Latentes Variablenmodell

- Projiziere hochdimensionale Daten (\mathbf{Y} , d-dimensional) auf niedrigdimensionalen latenten Raum (\mathbf{X} , q-dimensional, $q \ll d$)

■ Probabilistic PCA

- Wahrsch. eines Datenpunktes: $p(\mathbf{y}_n | \mathbf{W}, \beta) = \int p(\mathbf{y}_n | \mathbf{x}_n, \mathbf{W}, \beta) p(\mathbf{x}_n) d\mathbf{x}_n$

$$p(\mathbf{y}_n | \mathbf{x}_n, \mathbf{W}, \beta) = \mathcal{N}(\mathbf{y}_n | \mathbf{W}\mathbf{x}_n, \beta^{-1}\mathbf{I})$$

- Wahrsch. einer Datenmenge: $p(\mathbf{Y} | \mathbf{W}, \beta) = \prod_{n=1}^N p(\mathbf{y}_n | \mathbf{W}, \beta)$

- Marginalisiere nach \mathbf{W} :

- a-priori von \mathbf{W} : $p(\mathbf{W}) = \prod_{i=1}^D \mathcal{N}(\mathbf{w}_i | 0, \alpha^{-1}\mathbf{I})$

- Marginalisierte Wahrsch. von \mathbf{Y} :

$$p(\mathbf{Y} | \mathbf{X}, \beta) = \frac{1}{(2\pi)^{\frac{DN}{2}} |\mathbf{K}|^{\frac{D}{2}}} \exp\left(-\frac{1}{2} \text{Sp}(\mathbf{K}^{-1} \mathbf{Y} \mathbf{Y}^T)\right)$$

- Wobei $\mathbf{K} = \alpha \mathbf{X} \mathbf{X}^T + \beta^{-1} \mathbf{I}$ und $\mathbf{X} = [\mathbf{x}_1 \dots \mathbf{x}_N]^T$

■ Optimierte X:

■ Log-likelihood:
$$L = -\frac{DN}{2} \ln(2\pi) - \frac{D}{2} \ln |\mathbf{K}| - \frac{1}{2} \text{Sp}(\mathbf{K}^{-1} \mathbf{Y} \mathbf{Y}^T)$$

■ Optimierte X:
$$\frac{\partial L}{\partial \mathbf{X}} = \alpha \mathbf{K}^{-1} \mathbf{Y} \mathbf{Y}^T \mathbf{K}^{-1} \mathbf{X} - \alpha D \mathbf{K}^{-1} \mathbf{X}$$

■ Lösung: $\mathbf{X} = \mathbf{U}_q \mathbf{L} \mathbf{V}^T$

\mathbf{U}_q : N x q-Matrix mit q Eigenvektoren von $\mathbf{Y} \mathbf{Y}^T$
 \mathbf{L} : Diagonalmatrix mit Eigenwerten von $\mathbf{Y} \mathbf{Y}^T$
 \mathbf{V} : Orthogonale q x q-Matrix

■ Diese Lösung äquivalent zu PCA-Lösung

■ Kernel PCA: Ersetze $\mathbf{Y} \mathbf{Y}^T$ mit Kernel

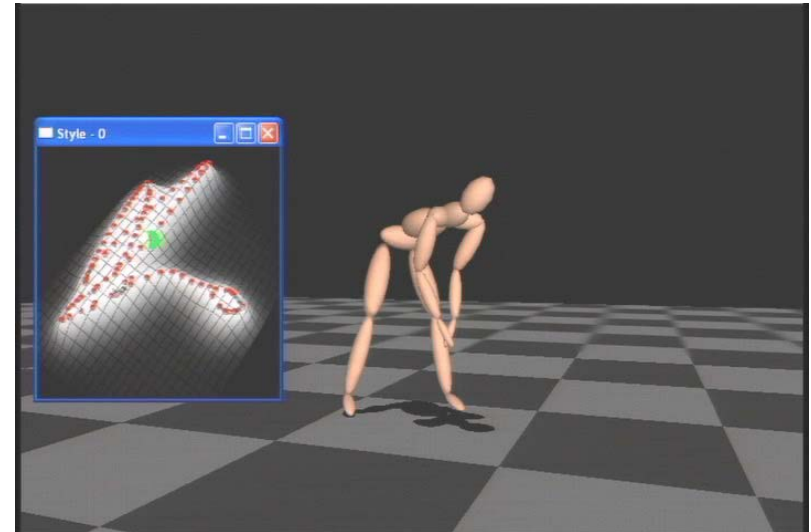
- PCA kann als GP interpretiert werden, der von X auf Y abbildet mit einer Kovarianzmatrix, die die Abbildungen auf lineare Abb. beschränkt
- Erweiterung: Nicht-lineare Abbildungen von latentem Raum auf Datenraum betrachten
 - Nicht-lineare Kovarianz-Funktion
 - Verwendung von Standard RBF Kernel statt $K = \alpha XX^T + \beta^{-1}I$
 - Berechne Gradient der Log-Likelihood mittels Kettenregel
 - Initialisiere X mittels PCA
 - Optimierte X und Hyperparameter des Kernels (z.B. mittels Conjugate Gradients)
 - Jede Gradienten-Berechnung benötigt Invertieren der Kernel-Matrix $\Rightarrow O(N^3)$

- GPLVM genutzt um menschliche Bewegungsdaten zu repräsentieren
 - Pose X : 42D Vector q (Gelenke, Position, Orientierung)
 - Immer ein spezifischer Bewegungsstil (z.B. Gehen)
 - Merkmalsvektor Y :
 - Gelenkwinkel
 - Geschwindigkeit und Beschleunigung für jedes Merkmal
 - > 100 Dimensionen
 - Latent Space X' : üblicherweise 2D oder 3D
- Skalierte Version von GPLVM

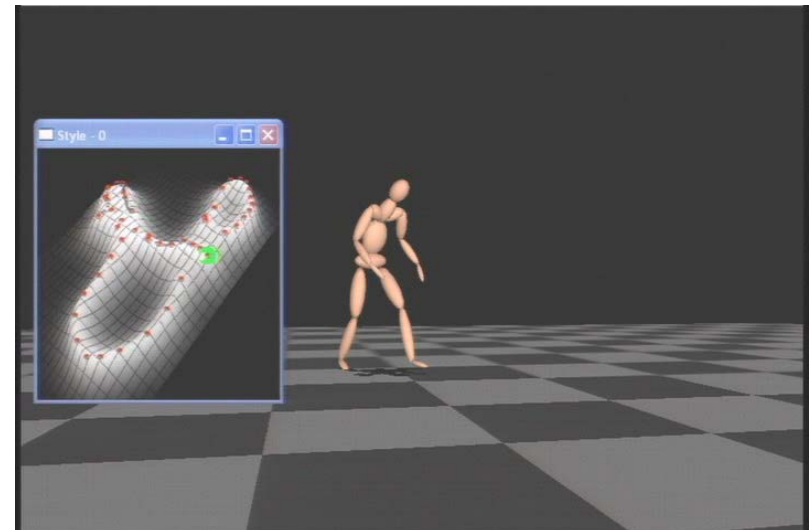
SBIK: Ergebnisse I

■ Unterschiedliche Stile:

■ Base-Ball Pitch

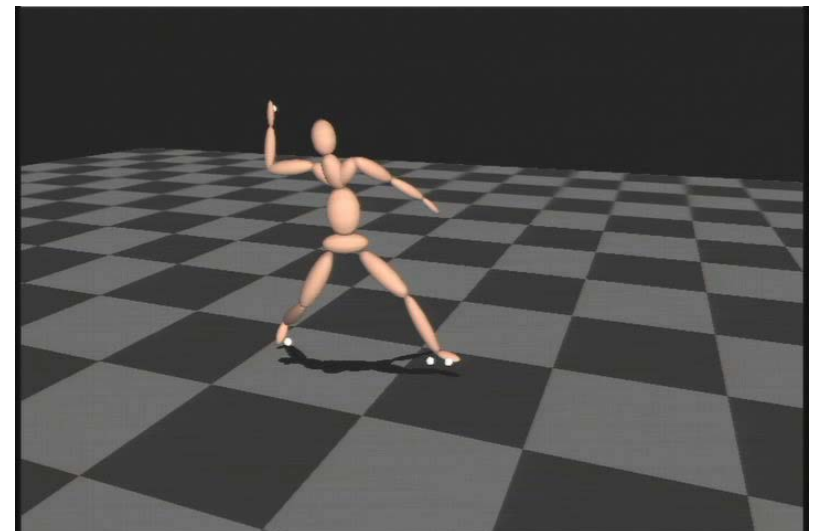
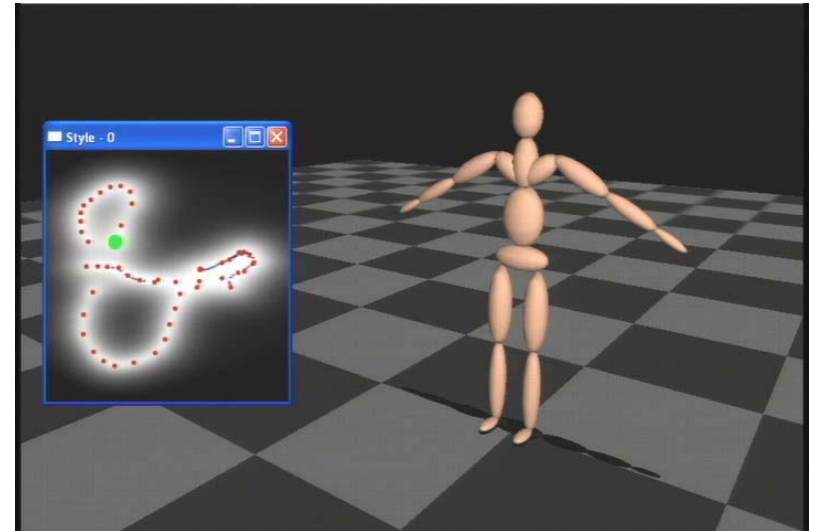


■ Tiefer Start f. Lauf



- Modell aufstellen
 - Position in 2-dim. latentem Raum spezifizieren

- Trajektorien festlegen/ändern



- [1] *Bernt Schiele / Stefan Roth: **Maschinelles Lernen: Statistische Verfahren II.** Vorlesungsfolien, Technische Universität Darmstadt, 2009.*
<http://www.gris.tu-darmstadt.de/teaching/courses/ws1213/ml2>
- [2] *C.E. Rasmussen, C.K.I. William: **Gaussian Processes for Machine Learning.** The MIT Press, 2006.*
<http://www.gaussianprocess.org/gpml/>
- [3] *N. Lawrence: **Gaussian Process Latent Variable Models for Visualisation of High Dimensional Data.** Advances in neural information processing systems 16: Proceedings of the 2003 conference, 2004.*
- [4] *C.M. Bishop, G.D. James: **Analysis of multiphase flows using dual-energy gamma densitometry and neural networks.** Nuclear Instruments and Methods in Physics Research, A327:580-593, 1993.*
- [5] *K. Grochow, S.L. Martin, A. Hertzmann, Z. Popovic: **Style-based Inverse Kinematics.** ACM Transactions on Graphics (Proceedings of SIGGRAPH 2004), 2004.*

Ergänzende Referenzen

- zentrale Homepage für Gauß'sche Prozesse
<http://www.gaussianprocess.org/>
- Video-Lecture von Carl E. Rasmussen
http://videlectures.net/epsrws08_rasmussen_lgp/