INTELLIGENTE DATENANALYSE IN MATLAB

Sequenzanalyse

Michael Brückner/Tobias Scheffer

Überblick

- Schritte der Datenanalyse:

Daten-vorverarbeitung → Problemanalyse → Problemlösung → Anwendung der Lösung

Daten-vorverarbeitung	Problemanalyse	Problemlösung	Anwendung der Lösung
Aggregation und Selektion von Daten. Integration und Säuberung der Daten. Feature-Extraktion.	Bestimmen von gegeb./gesuchten Größen. Wahl des Performanzmaß/ Zielkriteriums. Modellraum und Modellannahmen.	Algorithmen für das Optimieren des Zielkriteriums finden. Implementieren der Algorithmen.	Modell-Selektion & -Anpassung. Training & Evaluation des Modells auf gegebenen Daten. Vorhersage für neue Daten.

Überblick

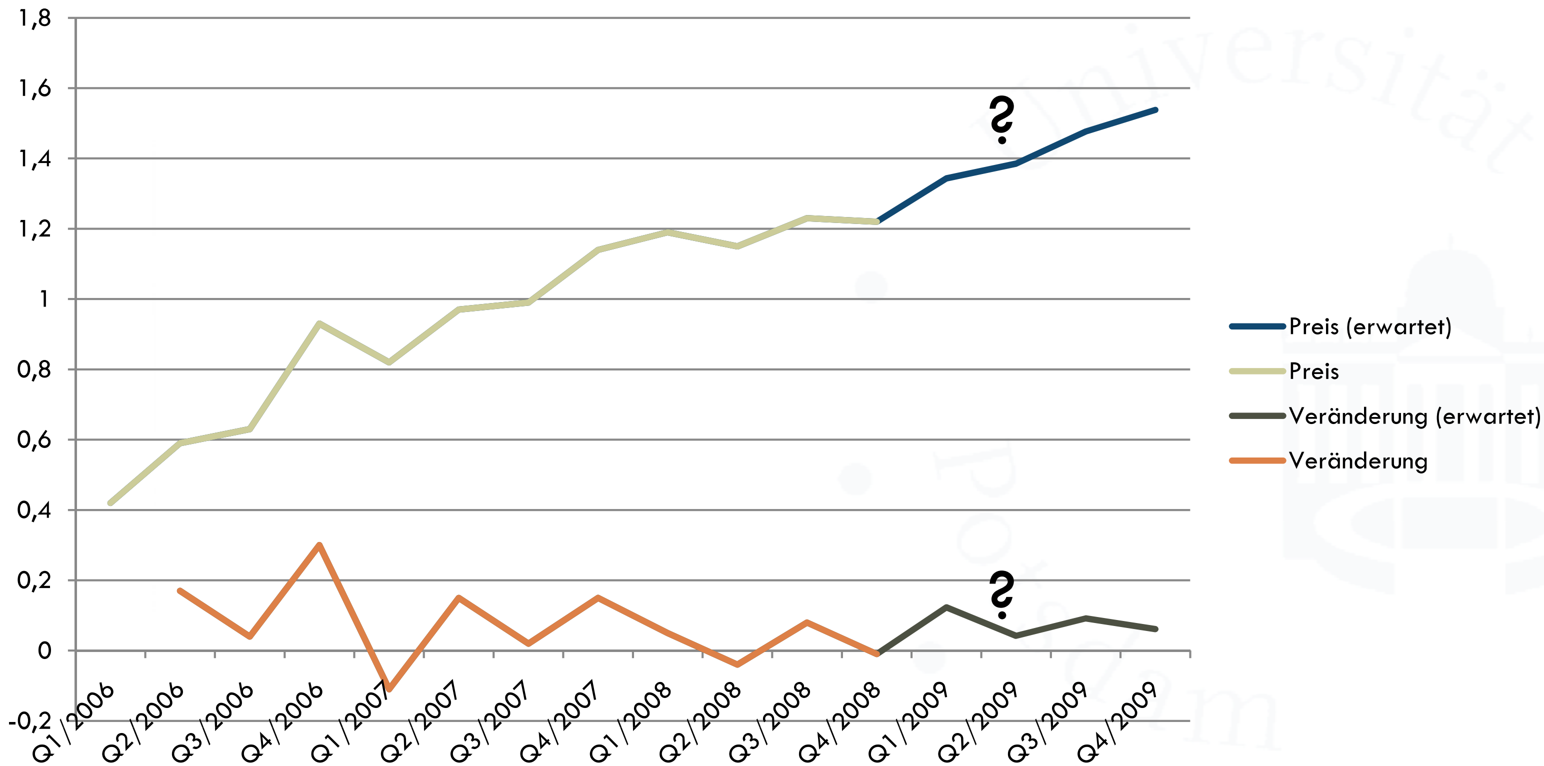
- Problemstellungen.
- Eigenschaften von Sequenzen.
- Modellierung von Sequenzen.
- Sequenzvorhersage.
- Lernen aus sequenziellen Daten.

Problemstellungen

- Eingabe: Sequenz von Datenpunkten (Beobachtungen, Messpunkte) wie bspw. Zeitreihen.
  - Tägliche Messung der Temperatur an einem Ort.
  - EKG eines Patienten.
  - Entwicklung der Einwohnerzahl eines Landes.
  - Aktienkurse an aufeinanderfolgenden Börsentagen.
- Gesucht: Modell welches Sequenz erklärt wie z.B. $f : x_t \mapsto x_{t+1}$.

Problemstellungen:
Beispiel

- Vorhersage der Preisentwicklung:



Problemstellungen:
Aufgaben der Sequenzanalyse

- Beschreibung/Eigenschaften:
  - Untersuchen der Charakteristika der Sequenz über Kennzahlen und Diagramme.
- Modellierung und Prognose:
  - Formulierung eines stochastischen Modells und Schätzung der Modellparameter.
  - Vorhersage zukünftiger Werte aufgrund des angepassten Modells.
- Kontrolle und Regelung:
  - Optimale Steuerung des datenerzeugenden Prozesses.

Problemstellungen:
Aufgaben der Sequenzanalyse

- Klassifikation sequenzieller Daten:
  - z.B. Klassifikation von EKG-Zeitreihen zur Vorhersage von Herzproblemen.
  - Kernel-Modelle benötigen Kernel (Ähnlichkeitsfunktion zwischen Zeitreihen).
- Anomalie-Erkennung:
  - z.B. Erdbeben-/Tsunami-Vorhersage.
- Clustering (Gruppieren von ähnlichen Zeitreihen).
- Retrieval/Suche:
  - z.B. Query-by-Humming.

Eigenschaften von Sequenzen:

Darstellung und Kenngrößen

- Sequenz wird durch Paare $S = \{(x_i, t_i)\}$ definiert.
  - t_i gibt Position innerhalb der Sequenz an.
  - Äquidistante Folge (gleiche Abstände $t_{i+1} - t_i$): $S = \{x_t : t = 1 \ldots n\}$
  - Nicht-äquidistante Folge: $S = \{x(t_i) : i = 1 \ldots n\}$

- Kenngrößen:
  - Mittelwert bis Position *t*: $\mu_x(t) = \frac{1}{t} \sum_{i=1}^{t} x_i$ $\qquad \mu_x = \mu_x(n)$
  - Varianz bis Position *t*: $\sigma_x^2(t) = \frac{1}{t} \sum_{i=1}^{t} (x_i - \mu_x(t))^2$ $\qquad \sigma_x^2 = \sigma_x^2(n)$

Eigenschaften von Sequenzen:

Korrelation

- Korrelation = lineare Abhängigkeit zwischen verschiedenen Zeitpunkten/Zeitreihen.
- Empirische Kovarianz: $$c_{x,y} = \frac{1}{n} \sum_{t=1}^{n} (x_t - \mu_x)(y_t - \mu_y)$$
- Empirischer Korrelations-koeffizient: $$r_{x,y} = \frac{c_{x,y}}{\sigma_x \sigma_y}$$
- Autokovarianzfunktion: $$c_x(k) = \frac{1}{n-k} \sum_{t=1}^{n-k} (x_t - \mu_x)(x_{t+k} - \mu_x)$$
- Autokorrelationsfunktion: $$r_x(k) = \frac{c_x(k)}{\sigma_x^2}$$

Eigenschaften von Sequenzen:

Korrelation – Beispiel

Abb. 1.1.1 *KONKURSE in den USA, (Quelle: B.Greenstein [1935])*

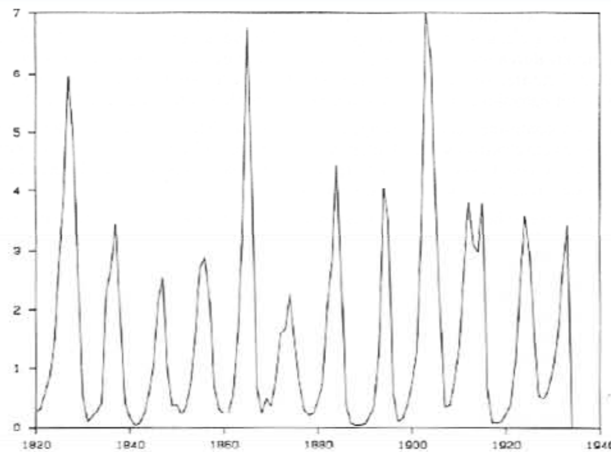


Abb. 1.1.2 *Anzahl gefangener Luchse (Quelle: M.J. Campbell and A.M. Walker [1977])*

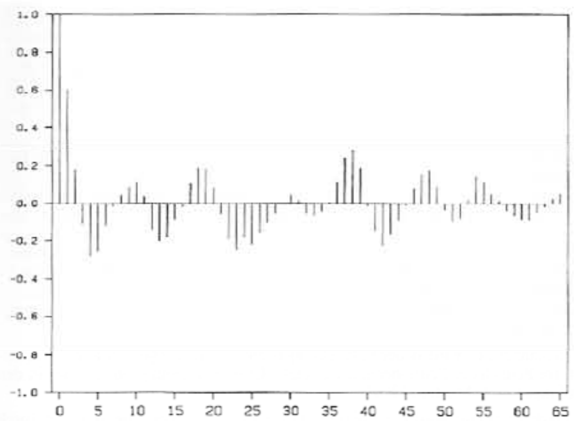


Abb. 1.2.3 *Autokorrelationsfunktion der Reihe KONKURSE*

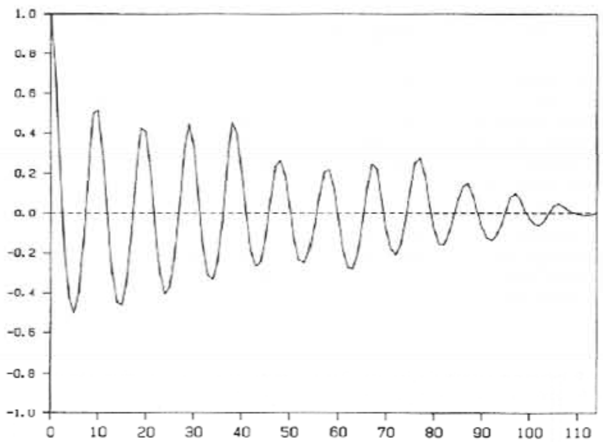
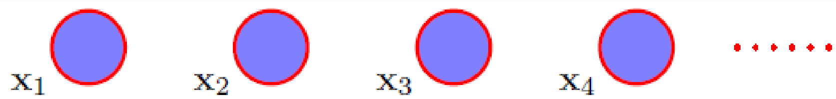


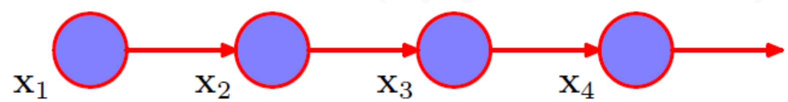
Abb. 1.2.4 *Autokorrelationsfunktion der Reihe LUCHS*

Modellierung von Sequenzen

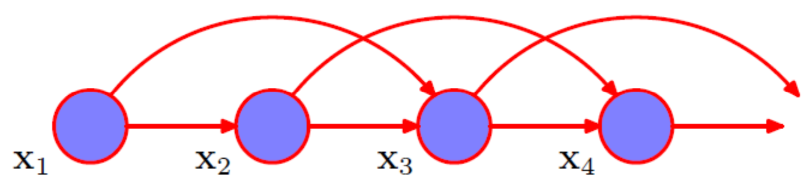
- Keine Abhängigkeiten:



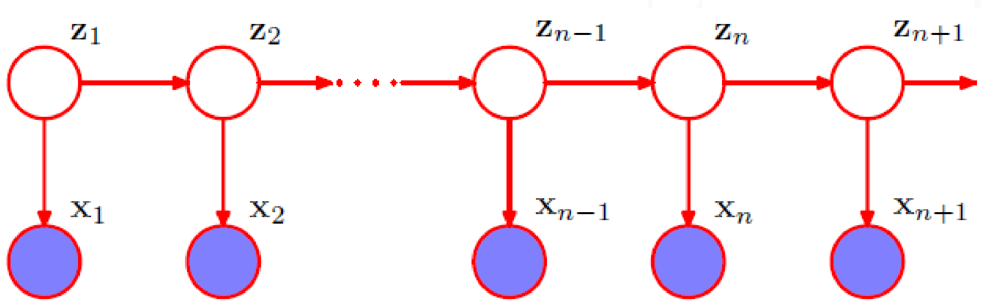
- Direkte Abhängigkeiten (Markov-Kette):
  - Erster Ordnung:



  - Zweiter Ordnung:

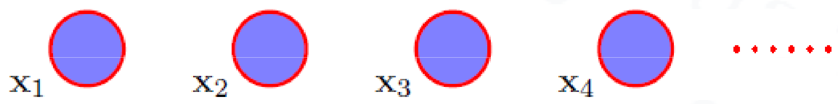


- Indirekte Abhängigkeiten (Hidden-Markov-Modell):



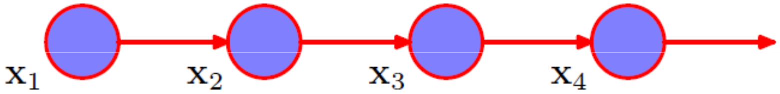
Modellierung von Sequenzen:
Keine Abhängigkeiten

- Annahmen:
  - Aufenthaltsort x zum Zeitpunkt t ist unabhängig vom Aufenthaltsort zu einem vorherigen Zeitpunkt.
- Beispiel: Rauschen (White Noise)
  - Mittelwert: $\mu_x(t) = 0$
  - Varianz: $\sigma_x^2(t) = \sigma_x^2$
  - Autokovarianzfunktion: $c_x(k) = 0$



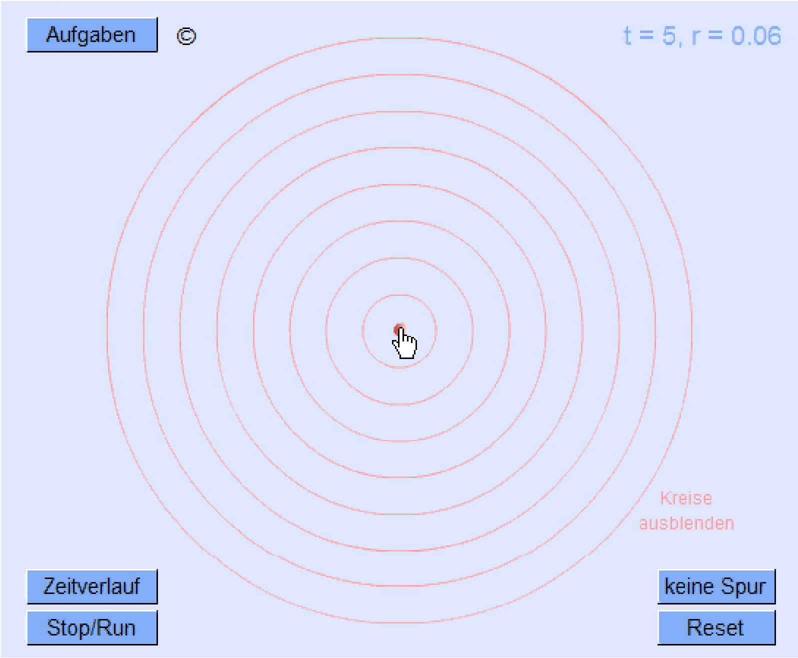
Modellierung von Sequenzen:
Direkte Abhängigkeiten

- Annahmen:
  - Aufenthaltsort x zum Zeitpunkt t hängt direkt von den k Aufenthaltsorten zu den Zeitpunkten $t-k,\ldots,t-1$ ab.
- Beispiel: Random Walk
  - Mittelwert: $\mu_x(t) = x_0$
  - Varianz: $\sigma_x^2(t) = t\sigma_x^2$
  - Beste Prognose: $E(x_{t+k} \mid x_t) = x_t$ $Var(x_{t+k} \mid x_t) = k\sigma_x^2$



Modellierung von Sequenzen:
Direkte Abhängigkeiten

- Beispiel für Markov-Kette (1. Ordnung): Random Walk
  - Aufenthaltsort zum Zeitpunkt t = Aufenthaltsort zum Zeitpunkt $t-1$ zuzgl. eine zufällige Strecke.

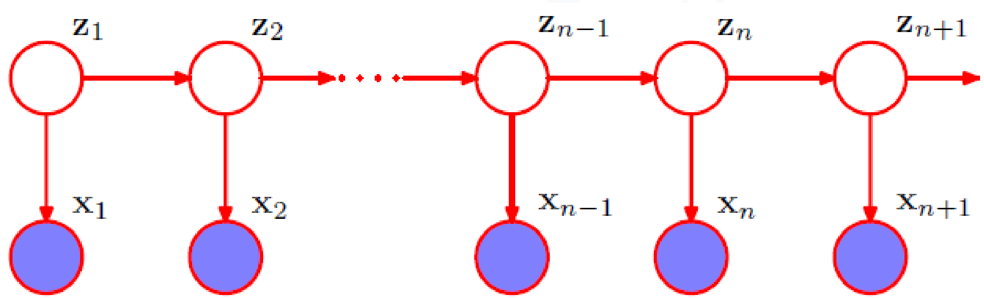


White Noise

Modellierung von Sequenzen:

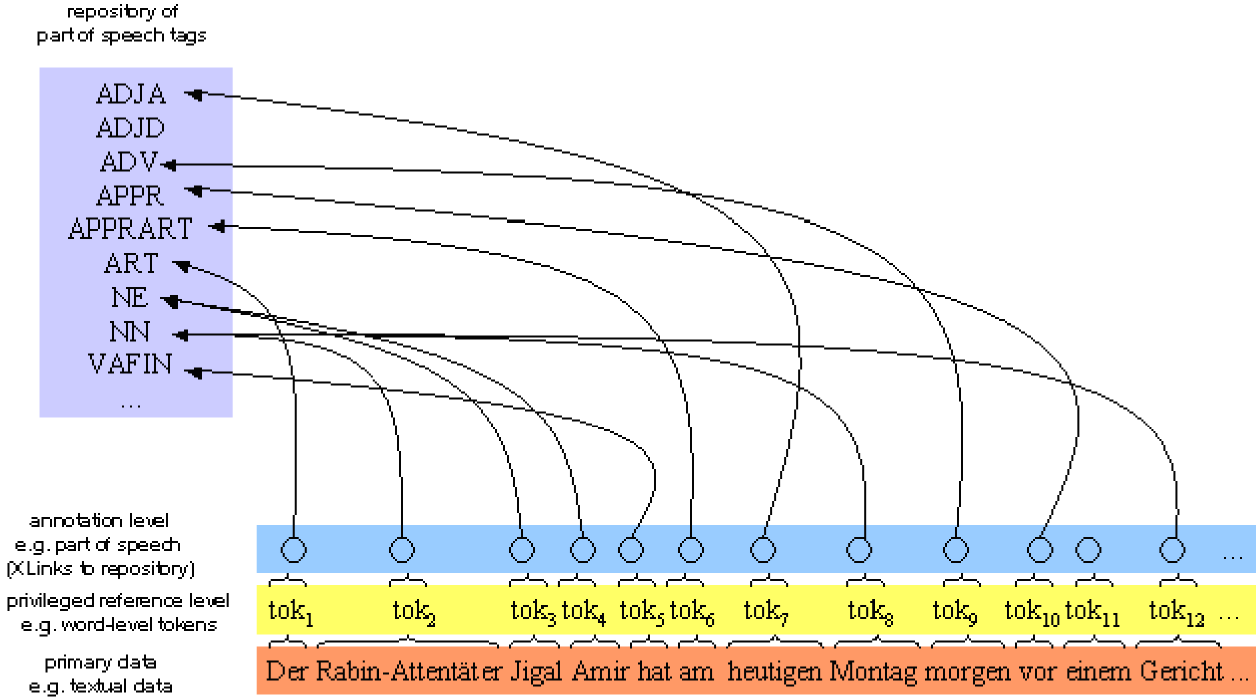
Indirekte Abhängigkeiten

- Annahmen:
  - Aufenthaltsort x zum Zeitpunkt t hängt direkt vom Zustand z zum Zeitpunkt t ab.
  - Zustand z zum Zeitpunkt t hängt direkt von den k Zuständen zu den Zeitpunkten $t-k,\ldots,t-1$ ab.
  - Zustände sind unbekannt/unbeobachtet.



Modellierung von Sequenzen:
Indirekte Abhängigkeiten

- Beispiel für HMM (1. Ordnung): Wortartenerkennung
  - Zuordnung eines Worts x zu Wortart (Zustand z).



Modellierung von Sequenzen:
Indirekte Abhängigkeiten

- Einsatzmöglichkeiten von HMMs:
  - Vorhersagen zukünftiger Beobachtungen.
    - z.B. welches Wort wird als nächstes eigegeben, Autovervollständigung.
  - Vorhersagen über zukünftige/vergangene Zustände bzw. wahrscheinlichste Zustandsfolge einer Sequenz.
    - z.B. Spracherkennung (Beobachtung = Audiosignale, Zustände = Wörter).
  - Parameterschätzung/Lernen des HMMs.

Sequenzvorhersage

- Sequenzvorhersage an den Beispielen…
  - Moving-Average-Prozess (MA).
  - Autoregressiver Prozess (AR).
  - Autoregressiver Moving-Average-Prozess (ARMA).
- Grundidee: Sequenz ist Linearkombination von Werten aus einer Sequenz von Rauschen (White Noise).
  - Sequenz von n unabhängigen Rauschwerten $\{r_1, \ldots, r_n\}$.
- Ziel: Faktoren dieser Linearkombination aus Daten schätzen.

Sequenzvorhersage:
Moving-Average-Prozess

- Annahmen:
  - Aufenthaltsort x zum Zeitpunkt t ist das gewichtete Mittel aus den $q+1$ Rauschwerten zu den Zeitpunkten $t-q,\ldots,t$:

$$x_t = r_t + \sum_{j=1}^{q} \alpha_j r_{t-j}$$

    mit Rauschwerten r_i wobei $E(r_i) = \mu_r = 0$ und $Var(r_i) = \sigma_r^2$.
- Ziel: Berechnung der Gewichte α_j.

Sequenzvorhersage:
Moving-Average-Prozess

- Beispiel: Ein Eisverkäufer möchte Verkaufszahlen von Eiscreme vorhersagen.
  - Nur bisherige Verkaufszahlen x_t gegeben.
- Annahme: Kunden essen nur Eis wenn in den letzten Tagen die Sonne oft zu sehen war.
- Modell:
  - (Unbeobachtete) Rauschwerte r_t sind die Sonnenstunden pro Tag normiert auf Mittelwert 0.
  - Beispiel für Modell mit Modellparametern:

$$x_t = r_t + 0,8 \cdot r_{t-1} + 0,5 \cdot r_{t-2} + 0,2 \cdot r_{t-3}$$

Sequenzvorhersage:
Moving-Average-Prozess

- Berechnung der Gewichte α_j am Beispiel eines MA(1)-Prozesses: $x_t = r_t + \alpha_1 r_{t-1}$
- Umformung des Prozesses ergibt:

$$\begin{aligned} r_j &= x_j - \alpha_1 r_{j-1} \\ r_{j-1} &= x_{j-1} - \alpha_1 r_{j-2} \\ &\vdots \\ r_1 &= x_1 \end{aligned} \quad \Rightarrow \quad \begin{aligned} r_j &= x_j - \alpha_1 r_{j-1} \\ r_j &= x_j - \alpha_1 \left(x_{j-1} - \alpha_1 r_{j-2} \right) \\ &= x_j + \left(-\alpha_1\right)^1 x_{j-1} + \left(-\alpha_1\right)^2 r_{j-2} \\ &\vdots \end{aligned} \quad \Rightarrow \quad r_j(\boldsymbol{\alpha}) = \sum_{i=0}^{j-1} \left(-\alpha_1\right)^i x_{j-i}$$

- Idee: Minimieren der Varianz der Rauschfolge $Var(r_t) = \sigma_r^2$ mit Mittelwert $E(r_t) = \mu_t = 0$:

$$\boldsymbol{\alpha}^* = \arg\min_{\boldsymbol{\alpha}} \frac{1}{t} \sum_{j=1}^{t} \left(r_j - \mu_r \right)^2 = \arg\min_{\boldsymbol{\alpha}} \sum_{j=1}^{t} r_j(\boldsymbol{\alpha})^2$$

Optimierungs-kriterium

Sequenzvorhersage:

Moving-Average-Prozess

- Iteratives Lösen des Minimierungsproblems mit Startlösung $r_j^0 = 0 \;\; \forall j = 1 \ldots t$ und $\alpha_j^0 = 0 \;\; \forall j = 1 \ldots q$.
- Taylor-Approximation der Rauschterme r_j in der l-ten Iteration: $r_j^{l+1}(\boldsymbol{\alpha}) = r_j^l + r_j'\left(\boldsymbol{\alpha}^l\right)^{\mathbf{T}}\left(\boldsymbol{\alpha} - \boldsymbol{\alpha}^l\right)$

Gradient von r_j an der Stelle α^l

- Minimieren der Varianz durch Null-Setzen der Ableitung $\min \sum_{j=1}^{t} r_j\left(\boldsymbol{\alpha}\right)^2 \Rightarrow 0 = \sum_{j=1}^{t} \frac{\partial r_j^{l+1}(\boldsymbol{\alpha}^l)}{\partial \boldsymbol{\alpha}^l}(\boldsymbol{\alpha}^{l+1})$ und Bestimmen von α^{l+1}.
- Wiederholung bis α konvergiert.

Sequenzvorhersage:

Autoregressiver Prozess

- Annahmen:
  - Aufenthaltsort x zum Zeitpunkt t ist die Summe aus dem Rauschwert zum Zeitpunkt t und dem gewichteten Mittel der p Aufenthaltsorte zu den Zeitpunkten $t-p,\ldots,t-1$:

$$x_t = r_t + \sum_{j=1}^{p} \beta_j x_{t-j}$$

    mit Rauschwerten r_t wobei $E(r_t)=0$ und $Var(r_t)=\sigma_r^2$.
- Ziel: Berechnung der Gewichte β_j.

Sequenzvorhersage:

Autoregressiver Prozess

- Beispiel: Ein Eisverkäufer möchte Verkaufszahlen von Eiscreme vorhersagen.
  - Nur bisherige Verkaufszahlen x_t gegeben.
- Annahme: Kunden essen nur Eis wenn sie in den letzten Tagen wenig Eis gegessen haben.
- Modell:
  - Verkaufszahl x_t ist nur abhängig von den vorherigen Verkaufszahlen und einem Rauschwert r_t (Laune der Kunden).
  - Beispiel für Modell mit Modellparametern:

$$x_t = r_t - 0,7 \cdot x_{t-1} - 0,6 \cdot x_{t-2} + 0,1 \cdot x_{t-3}$$

Sequenzvorhersage:
Autoregressiver Prozess

- Berechnung der Gewichte β_j eines AR-Prozesses:

$$x_t = r_t + \sum_{j=1}^{p} \beta_j x_{t-j}$$

- Idee: Minimieren des mittleren quadratischen Fehlers.

$$\boldsymbol{\beta}^* = \arg\min_{\boldsymbol{\beta}} \frac{1}{t-p-1} \sum_{i=p+1}^{t} \underbrace{\left(x_i - r_i - \sum_{j=1}^{p} \beta_j x_{i-j} \right)^2}_{\text{Erwartungswert Null \& konstante Varianz}} = \arg\min_{\boldsymbol{\beta}} \sum_{i=p+1}^{t} \left(x_i - \sum_{j=1}^{p} \beta_j x_{i-j} \right)^2$$

$$\boldsymbol{\beta}^* = \arg\min_{\boldsymbol{\beta}} \left(\underbrace{\begin{bmatrix} x_{p+1} \\ x_{p+2} \\ \vdots \\ x_n \end{bmatrix}}_{y} - \underbrace{\begin{bmatrix} x_p & x_{p-1} & \cdots & x_1 \\ x_{p+1} & x_p & \cdots & x_2 \\ \vdots & \vdots & \ddots & \vdots \\ x_{n-1} & x_{n-2} & \cdots & x_{n-p} \end{bmatrix}}_{A} \begin{bmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_p \end{bmatrix} \right)^2 \Rightarrow \underbrace{\boldsymbol{\beta}^* = \mathbf{A}^+ \mathbf{y}}_{\text{Pseudo-Inverse von } A}$$

Sequenzvorhersage:
Autoregressiver Moving-Average-Prozess

- Motivation:
  - Fast jeder endliche Datensatz gut an MA- oder AR-Modell mit hoher Ordnung anpassbar.
  - Je größer die Ordnung, desto mehr Parameter.
  - Ziel: Modell mit möglichst wenigen Parametern welches zudem reale Bedeutung (Interpretation) hat.
- Idee: Kombination von MA- und AR-Modellen zu ARMA(p,q)-Prozess.

$$x_t = r_t + \sum_{j=1}^{q} \alpha_j r_{t-j} + \sum_{j=1}^{p} \beta_j x_{t-j}$$

Sequenzvorhersage:

Autoregressiver Moving-Average-Prozess

- Überlagerungssatz von ARMA-Prozessen:
  - x_t und y_t seien zwei unabhängige ARMA-Prozesse der Ordnung (p_1, q_1) und (p_2, q_2).
  - Summe $z_t = x_t + y_t$ ist wieder ein ARMA-Prozess der Ordnung (p, q).
  - Für AR-Ordung gilt: $p \leq p_1 + p_2$
  - Für MA-Ordung gilt: $q \leq \max(p_1 + q_2, p_2 + q_1)$
- Folgerung:
  - Summe zweier MA-Prozesse ergibt wieder MA-Prozess.
  - Summe zweier AR-Prozesse ergibt ARMA-Prozess.

Sequenzvorhersage:
Autoregressiver Moving-Average-Prozess

- Beispiel: Vorhersage der Werbeausgaben zweier konkurrierender Unternehmen.
  - Nur bisherige Werbeausgaben x_t und y_t gegeben.
- Modell:
  - Werbeausgabe x_t ist abhängig von der vorherigen Werbeausgabe des Konkurrenten $y_{t\text{-}1}$ und einem Rauschwert.
  - Werbeausgabe y_t ist abhängig von der vorherigen Werbeausgabe des Konkurrenten $x_{t\text{-}1}$ und einem Rauschwert.

$$\begin{aligned} x_t &= r_t + \beta_1 y_{t-1} \\ y_t &= r'_t + \beta'_1 x_{t-1} \end{aligned} \quad \Rightarrow \quad x_t = r_t + \beta_1 \left(r'_{t-1} + \beta'_1 x_{t-2} \right) = r_t + \alpha_1 r_{t-1} + \beta_2 x_{t-2}$$

ARMA(2,1)-Prozess

Sequenzvorhersage:
Autoregressiver Moving-Average-Prozess

- Berechnung der Gewichte α_j und β_j eines ARMA-Prozesses analog zu MA-Prozessen.
- Erweiterungen von ARMA-Prozessen:
  - ARIMA: Ist x_t nach d-maligem Anwenden des Differenzen-filters ein stationärer ARMA(p,q)-Prozess, so bezeichnet man den Prozess x_t als ARIMA(p,d,q)-Prozess.
  - ARMAX: Abhängigkeiten zwischen den Rauschwerten.
  - …

Sequenzvorhersage:

Autoregressiver Moving-Average-Prozess

- Vorhersage mittels angepasstem ARMA-Prozess (Prognose für x_{t+h}):
  - x_t, x_{t-1}, … x_1 entsprechen den tatsächlichen Beobachtungen.
  - x_{t+h}, … x_{t+1} werden durch ihre Prognosen ersetzt.
  - Störterme r_t, r_{t-1}, … r_1 entsprechen den Prognosefehlern der 1-Schrittprognosen in der Vergangenheit.
  - Störungen r_{t+h}, … r_{t+1} werden durch ihren Erwartungswert Null ersetzt.

Lernen aus sequenziellen Daten:
Problemstellung

- Gegeben: sequenzielle Trainingsdaten mit bekannten Zielattributen (gelabelte Daten).
- Eingabe: Sequenz von Attribut-Belegungen $\mathbf{x}$.
- Ausgabe: Belegung des/der Zielattribut(e) y.
- Gesucht: Modell $f : \mathbf{x} \mapsto y$.
- Ansatz: Verwendung von Kernel-Modellen basierend auf Sequenz-Kernel.
  - Quadratische Euklidische Distanz (RBF-Kernel).
  - Dynamic Time Warping (DTW-Kernel).
  - Editierdistanz.

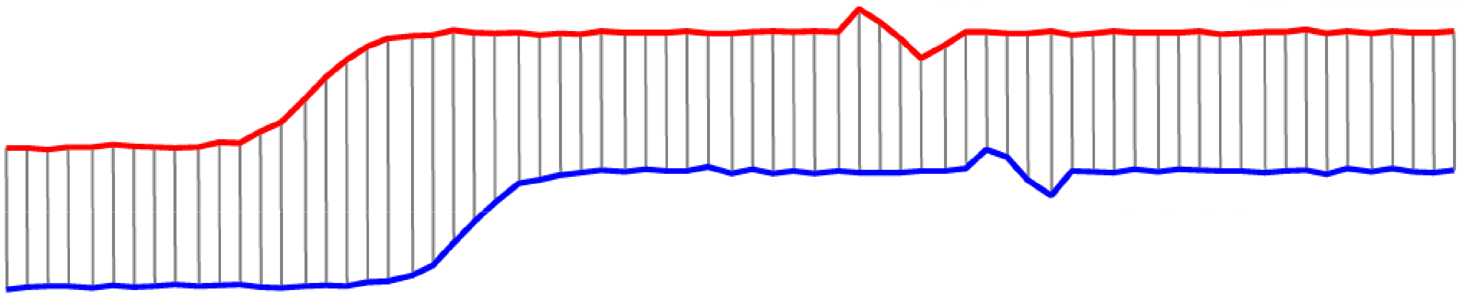
Lernen aus sequenziellen Daten:

Quadratische Euklidische Distanz

- Für numerische Attribute.
- Sequenzen als Vektoren auffassen und quadratischen euklidischen Abstand bestimmen:

$$D_{RBF}(\mathbf{x},\mathbf{y}) = \sum_{i=1}^{t} (x_i - y_i)^2 \quad \Rightarrow \quad k_{RBF}(\mathbf{x},\mathbf{y}) = \exp\left(-\lambda D_{RBF}(\mathbf{x},\mathbf{y})\right)$$

- Verschobene/gedehnte Motive werden nicht berücksichtigt.

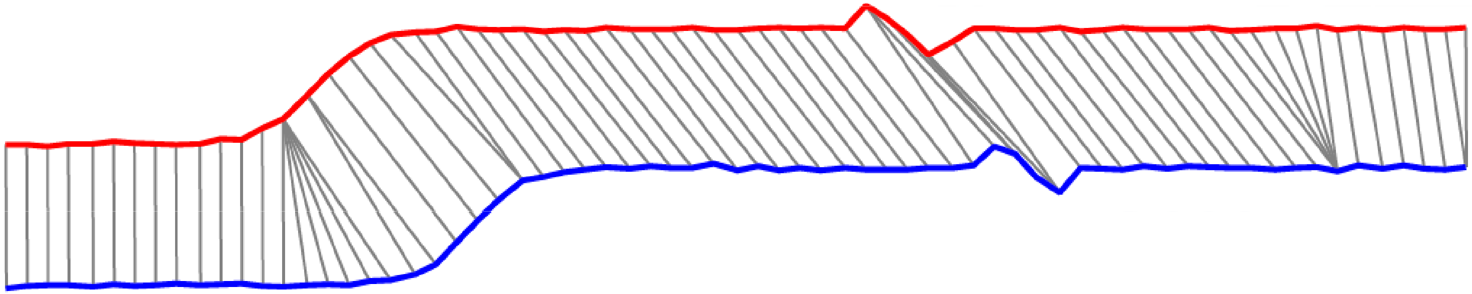


Lernen aus sequenziellen Daten:

Dynamic Time Warping

- Für numerische Attribute.
- Zuordungsfunktionen $\pi_x(i) \in [1, t_x]$ und $\pi_y(i) \in [1, t_y]$.
- DTW-Distanz ist Minimum des verschobenen (quadratischen euklidischen) Abstands:

$$D_{DTW}(\mathbf{x}, \mathbf{y}) = \min_{\pi_x, \pi_y} \sum_{i=1}^{t} (x_{\pi_x(i)} - y_{\pi_y(i)})^2 \Rightarrow k_{DTW}(\mathbf{x}, \mathbf{y}) = \exp\left(-\lambda D_{DTW}(\mathbf{x}, \mathbf{y})\right)$$



Lernen aus sequenziellen Daten:

Dynamic Time Warping

- Berechnung mit dynamischer Programmierung, rekursive Definition:
  - Sei $\gamma(i, j)$ minimale (verschobene) quadrierte euklidische Distanz bis zu den Zeitpunkten i und j:

$$\gamma(i, j) = (x_i - y_j)^2 + \min\left(\gamma(i-1, j-1), \gamma(i-1, j), \gamma(i, j-1)\right)$$

- Algorithmus:

DTW(*Sequenzen* **x** *und* **y**)

  Setze $\gamma(0,0) = 0, \; \forall i, j \; \gamma(i,0) = \infty, \; \gamma(0, j) = \infty$

  FOR $i = 1 \ldots t_x$

    FOR $j = 1 \ldots t_y$

$$\gamma(i, j) = (x_i - y_j)^2 + \min\left(\gamma(i-1, j-1), \gamma(i-1, j), \gamma(i, j-1)\right)$$

  RETURN $\gamma(t_x, t_y)$

Lernen aus sequenziellen Daten:
Editierdistanz

- Für nominale und ordinale Attribute (Sonderform von Dynamic Time Warping).
- Editierdistanz ist minimale Anzahl an Operationen (Einfügen, Löschen, Ersetzen) um zwei Sequenzen ineinander zu überführen.
- Berechnung mit dynamischer Programmierung, rekursive Definition:
  - Sei $\gamma(i, j)$ minimale Editierdistanz zu den Zeitpunkten i und j:

$$\gamma(i, j) = \left[x_i \neq y_j \right] + \min\left(\gamma(i-1, j-1), \gamma(i-1, j), \gamma(i, j-1) \right)$$

Zusammenfassung

- Zahlreiche Problemstellungen/Anwendungsgebiete
  - Charakterisierung, Visualisierung, Beschreibung.
  - Lernen von/aus sequenziellen Daten.
- Modellierung direkter Abhängigkeiten (Markov-Kette).
- Modellierung indirekter Abhängigkeiten (HMM).
- Vorhersage univariater Sequenzen durch stochastische Prozesse (z.B. MA, AR, ARMA usw.)
- Sequenz-Kernel (z.B. DTW).