# Wissensentdeckung in Datenbanken

## Strukturlernen, Merkmalsauswahl

Nico Piatkowski und Uwe Ligges

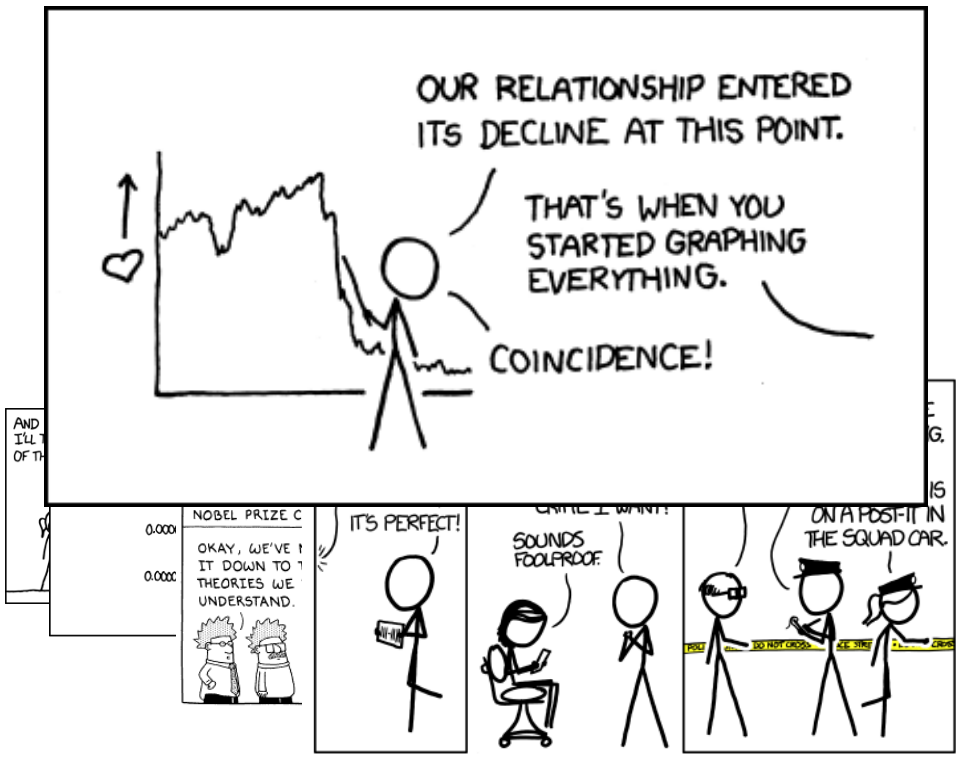
Informatik—Künstliche Intelligenz
Computergestützte Statistik
Technische Universität Dortmund

27.06.2017

# Überblick

Was bisher geschah...

- Modellklassen
- Verlustfunktionen
- Numerische Optimierung
- Regularisierung
- Überanpassung
- SQL, Häufige Mengen
- SVM, xDA, Bäume, . . .
- Graphische Modelle



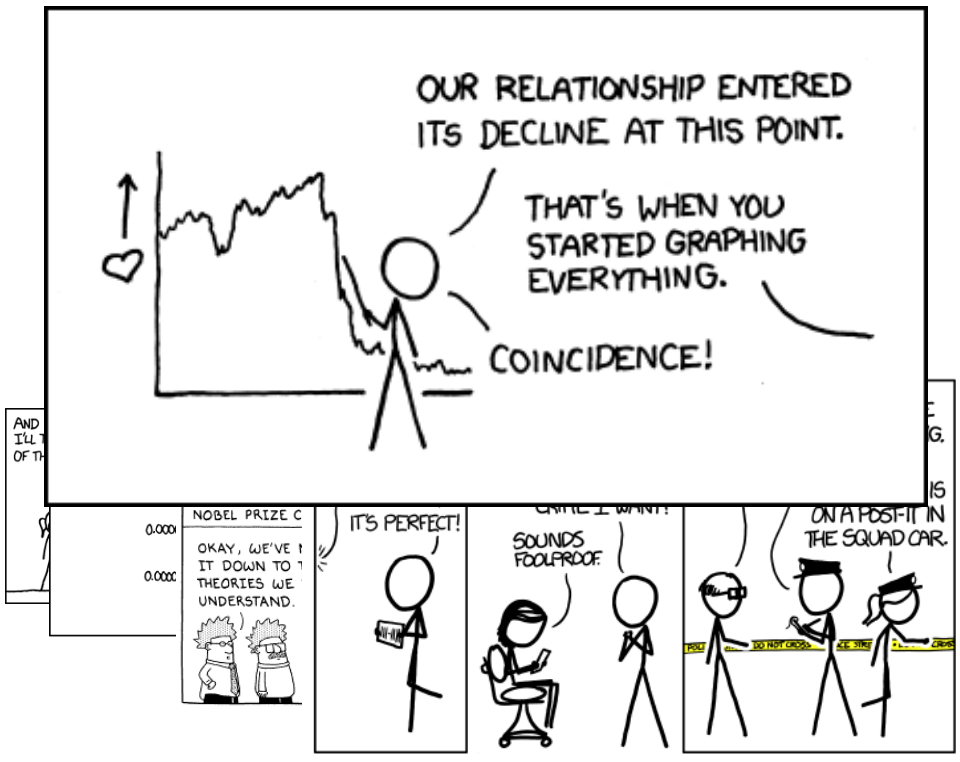

Heute

- Strukturlernen, Merkmalsauswahl

# Überblick

Was bisher geschah...

- Modellklassen
- Verlustfunktionen
- Numerische Optimierung
- Regularisierung
- Überanpassung
- SQL, Häufige Mengen
- SVM, xDA, Bäume, . . .
- Graphische Modelle

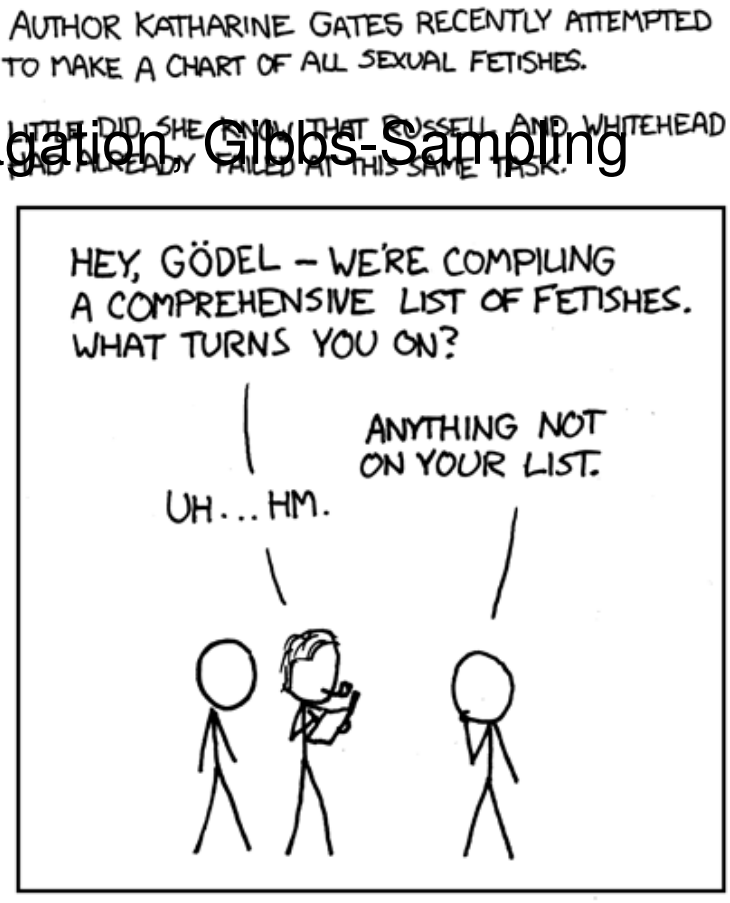



Heute

- Strukturlernen, Merkmalsauswahl

# Überblick

- Wiederholung: Belief Propagation, Gibbs-Sampling
- Strukturlernen
  - Chow-Liu Bäumen
  - Regularisierung
- Merkmalsauswahl
  - Forward-Selection
  - Backward-Selection
  - Regularisierung



# Belief Propagation

**Ziel**: Berechnung von $\mathbb{E}_{\boldsymbol{\beta}}[\phi(\boldsymbol{X})_{C=\boldsymbol{y}}] = \mathbb{P}(\boldsymbol{X}_C = \boldsymbol{y})$

$$Z(\boldsymbol{\beta}) = \sum_{\boldsymbol{x}_1 \in \mathcal{X}_1} \sum_{\boldsymbol{x}_2 \in \mathcal{X}_2} \cdots \sum_{\boldsymbol{x}_n \in \mathcal{X}_n} \prod_{uv \in E} \underbrace{\exp(\langle \boldsymbol{\beta}_{uv}, \phi_{uv}(\boldsymbol{x}_{uv}) \rangle)}_{\psi_{uv}(\boldsymbol{x}_{uv})}$$

Distributivität ausnutzen...

$$m_{v \to u}(x) = \sum_{y \in \mathcal{X}_v} \psi_{uv}(yx) \prod_{w \in \mathcal{N}(v) \setminus \{u\}} m_{w \to v}(y)$$

$$Z(\boldsymbol{\beta}) = \sum_{y \in \mathcal{X}_v} \prod_{w \in \mathcal{N}(v)} m_{w \to v}(y)$$

$$\mathbb{P}(\boldsymbol{X}_{uv} = xy) = \frac{\psi_{uv}(yx)}{Z(\boldsymbol{\beta})} \prod_{w \in \mathcal{N}(v) \setminus \{u\}} m_{w \to v}(y) \prod_{w \in \mathcal{N}(u) \setminus \{v\}} m_{w \to u}(y)$$

# Belief Propagation

**Ziel**: Berechnung von $\mathbb{E}_{\boldsymbol{\beta}}[\phi(\boldsymbol{X})_{C=\boldsymbol{y}}] = \mathbb{P}(\boldsymbol{X}_C = \boldsymbol{y})$

$$Z(\boldsymbol{\beta}) = \sum_{\boldsymbol{x}_1 \in \mathcal{X}_1} \sum_{\boldsymbol{x}_2 \in \mathcal{X}_2} \cdots \sum_{\boldsymbol{x}_n \in \mathcal{X}_n} \prod_{uv \in E} \underbrace{\exp(\langle \boldsymbol{\beta}_{uv}, \phi_{uv}(\boldsymbol{x}_{uv}) \rangle)}_{\psi_{uv}(\boldsymbol{x}_{uv})}$$

Distributivität ausnutzen...

$$m_{v \to u}(x) = \sum_{y \in \mathcal{X}_v} \psi_{uv}(yx) \prod_{w \in \mathcal{N}(v) \setminus \{u\}} m_{w \to v}(y)$$

$$Z(\boldsymbol{\beta}) = \sum_{y \in \mathcal{X}_v} \prod_{w \in \mathcal{N}(v)} m_{w \to v}(y)$$

$$\mathbb{P}(\boldsymbol{X}_{uv} = xy) = \frac{\psi_{uv}(yx)}{Z(\boldsymbol{\beta})} \prod_{w \in \mathcal{N}(v) \setminus \{u\}} m_{w \to v}(y) \prod_{w \in \mathcal{N}(u) \setminus \{v\}} m_{w \to u}(y)$$

# Belief Propagation

**Ziel**: Berechnung von $\mathbb{E}_{\boldsymbol{\beta}}[\phi(\boldsymbol{X})_{C=\boldsymbol{y}}] = \mathbb{P}(\boldsymbol{X}_C = \boldsymbol{y})$

$$Z(\boldsymbol{\beta}) = \sum_{\boldsymbol{x}_1 \in \mathcal{X}_1} \sum_{\boldsymbol{x}_2 \in \mathcal{X}_2} \cdots \sum_{\boldsymbol{x}_n \in \mathcal{X}_n} \prod_{uv \in E} \underbrace{\exp(\langle \boldsymbol{\beta}_{uv}, \phi_{uv}(\boldsymbol{x}_{uv}) \rangle)}_{\psi_{uv}(\boldsymbol{x}_{uv})}$$

Distributivität ausnutzen...

$$m_{v \to u}(x) = \sum_{y \in \mathcal{X}_v} \psi_{uv}(yx) \prod_{w \in \mathcal{N}(v) \setminus \{u\}} m_{w \to v}(y)$$

$$Z(\boldsymbol{\beta}) = \sum_{y \in \mathcal{X}_v} \prod_{w \in \mathcal{N}(v)} m_{w \to v}(y)$$

$$\mathbb{P}(\boldsymbol{X}_{uv} = xy) = \frac{\psi_{uv}(yx)}{Z(\boldsymbol{\beta})} \prod_{w \in \mathcal{N}(v) \setminus \{u\}} m_{w \to v}(y) \prod_{w \in \mathcal{N}(u) \setminus \{v\}} m_{w \to u}(y)$$

# Gibbs Sampling: Algorithmus

1. Erzeuge $\boldsymbol{x}$ zufällig Gleichverteilt (das entspricht **NICHT** $\mathbb{P}_{\boldsymbol{\beta}}$!)
2. Besuche jeden Knoten $v \in V$ und weise gemäß $\mathbb{P}_v(x \mid \boldsymbol{x}_{\mathcal{N}(v)})$ neuen Wert zu
3. Wiederhole Schritt 2 so oft wie möglich

Man kann zeigen: Nach endlicher Anzahl von Schritten ist $\boldsymbol{x}$ ein echtes Sample aus $\mathbb{P}_{\boldsymbol{\beta}}$!

**Dann**: Nutze den Algorithmus um "viele" (so viele wie möglich) Samples zu erzeugen und berechne $\mathbb{P}(\boldsymbol{X}_{uv} = xy)$ (für alle Kanten $\{v, u\} \in E$) durch "abzählen"

# Gibbs Sampling: Algorithmus

1. Erzeuge $\boldsymbol{x}$ zufällig Gleichverteilt (das entspricht **NICHT** $\mathbb{P}_{\boldsymbol{\beta}}$!)
2. Besuche jeden Knoten $v \in V$ und weise gemäß $\mathbb{P}_v(x \mid \boldsymbol{x}_{\mathcal{N}(v)})$ neuen Wert zu
3. Wiederhole Schritt 2 so oft wie möglich

Man kann zeigen: Nach endlicher Anzahl von Schritten ist $\boldsymbol{x}$ ein echtes Sample aus $\mathbb{P}_{\boldsymbol{\beta}}$!

**Dann**: Nutze den Algorithmus um "viele" (so viele wie möglich) Samples zu erzeugen und berechne $\mathbb{P}(\boldsymbol{X}_{uv} = xy)$ (für alle Kanten $\{v, u\} \in E$) durch "abzählen"

# Chow-Liu Bäume

Minimierung der Distanz zwischen optimalem Graph und "bestem" Baum $T$
Hier: Distanz gemessen durch Kullback-Leiber Divergenz

$$\mathrm{KL}(\mathbb{P}^*, \mathbb{P}_T) = \sum_{\boldsymbol{x} \in \mathcal{X}} \mathbb{P}^*(\boldsymbol{x}) \log \frac{\mathbb{P}^*(\boldsymbol{x})}{\mathbb{P}_T(\boldsymbol{x})}$$

Kann umgeform werden zu

$$\mathrm{KL}(\mathbb{P}^*, \mathbb{P}_T) = -\mathcal{H}(\mathbb{P}^*) + \sum_{v \in V} \mathcal{H}(\mathbb{P}^*_v) - \underbrace{\sum_{vu \in E(T)} I(\boldsymbol{X}_v, \boldsymbol{X}_u)}_{\text{Maximaler Spannbaum!}}$$

mit $I(\boldsymbol{X}_v, \boldsymbol{X}_u) = \mathrm{KL}(\mathbb{P}_{vu}, \mathbb{P}_v \mathbb{P}_u)$ (Allgemeines Maß für Unabhängigkeit!)

# Chow-Liu Bäume

Minimierung der Distanz zwischen optimalem Graph und "bestem" Baum $T$
Hier: Distanz gemessen durch Kullback-Leiber Divergenz

$$\mathrm{KL}(\mathbb{P}^*, \mathbb{P}_T) = \sum_{\boldsymbol{x} \in \mathcal{X}} \mathbb{P}^*(\boldsymbol{x}) \log \frac{\mathbb{P}^*(\boldsymbol{x})}{\mathbb{P}_T(\boldsymbol{x})}$$

Kann umgeform werden zu

$$\mathrm{KL}(\mathbb{P}^*, \mathbb{P}_T) = -\mathcal{H}(\mathbb{P}^*) + \sum_{v \in V} \mathcal{H}(\mathbb{P}^*_v) - \underbrace{\sum_{vu \in E(T)} I(\boldsymbol{X}_v, \boldsymbol{X}_u)}_{\text{Maximaler Spannbaum!}}$$

mit $I(\boldsymbol{X}_v, \boldsymbol{X}_u) = \mathrm{KL}(\mathbb{P}_{vu}, \mathbb{P}_v \mathbb{P}_u)$ (Allgemeines Maß für Unabhängigkeit!)

# Graphen mittels Regularisierung

**Baobachtung**: Sind ist kompletter Parametervektor $\boldsymbol{\beta}_{vu}$ einer Kante $= 0$, so hat diese Kante keinen Einfluss auf $\mathbb{P}(\boldsymbol{X} = \boldsymbol{x})$!

**Idee**: Minimiere $\ell(\boldsymbol{\beta}; \mathcal{D}) + \lambda \|\boldsymbol{\beta}\|_1$

$\|\cdot\|_1$ nicht differenzierbar!! $\rightarrow$ Proximaler Gradientenabstieg (nächste Woche)

**Spoiler**:

$$\text{prox}_{\lambda\|\cdot\|_1}(\boldsymbol{\beta}_i) = \begin{cases} \boldsymbol{\beta}_i - \lambda & , \boldsymbol{\beta}_i > \lambda \\ \boldsymbol{\beta}_i + \lambda & , \boldsymbol{\beta}_i < -\lambda \\ 0 & , \text{sonst} \end{cases}$$