Wissensentdeckung in Datenbanken

Überanpassung, Häufige Mengen

Nico Piatkowski und Uwe Ligges

Informatik—Künstliche Intelligenz
Computergestützte Statistik
Technische Universität Dortmund

09.05.2017

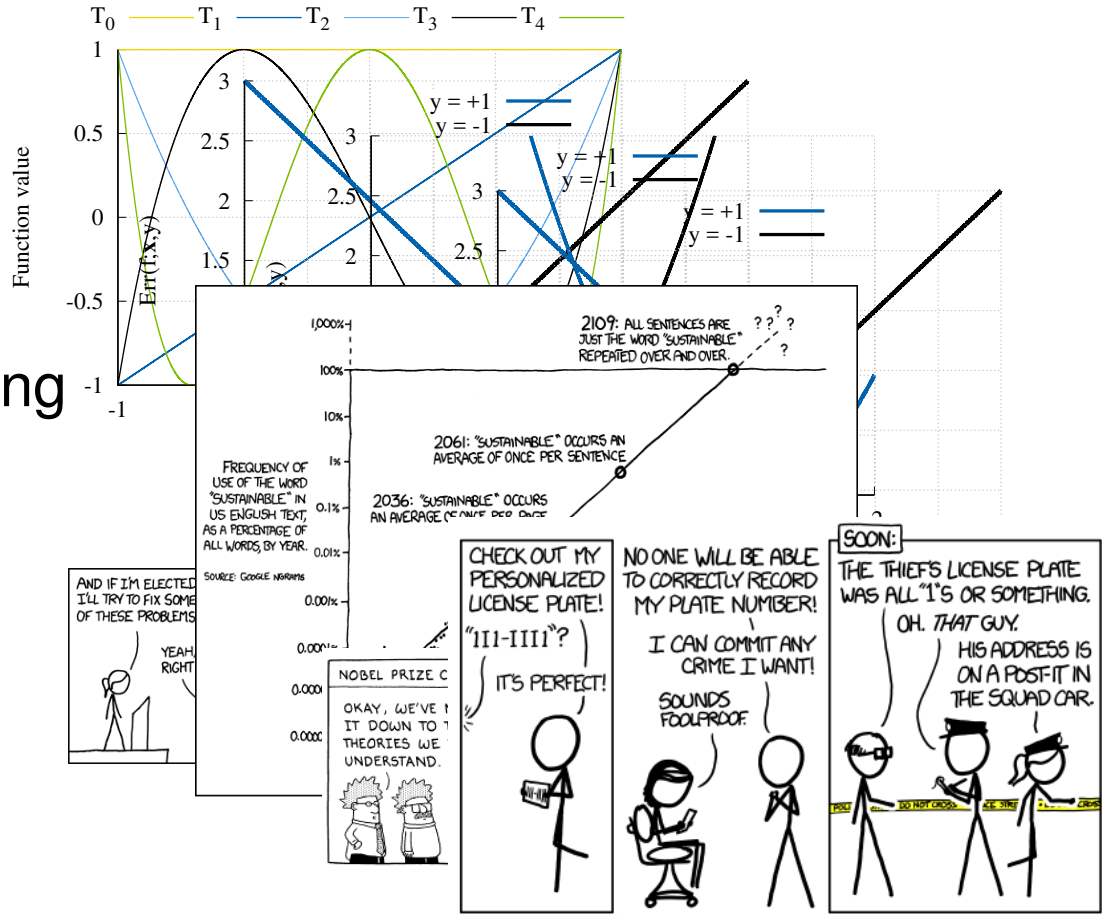
Überblick

Was bisher geschah...

- Modellklassen
- Verlustfunktionen
- Numerische Optimierung
- Regularisierung

Heute

- Überanpassung
- Häufige Mengen



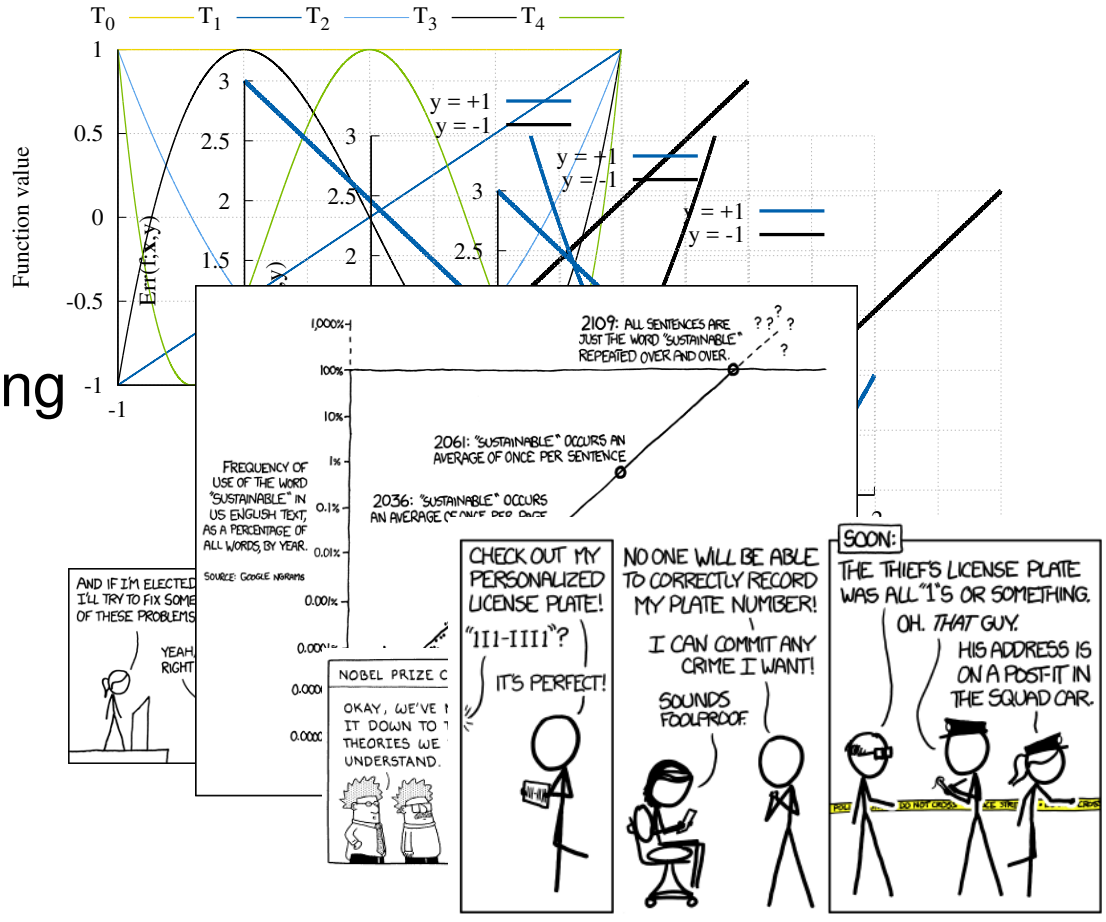
Überblick

Was bisher geschah...

- Modellklassen
- Verlustfunktionen
- Numerische Optimierung
- Regularisierung

Heute

- Überanpassung
- Häufige Mengen



Fehlertypen

- Lernen des Modells $f_{\mathcal{D}}$ auf Basis von *Trainingsdaten* $\mathcal{D}$
- Anwendung des Modells auf *Testdaten* $\mathcal{T}$
- $\mathcal{D}$ ist Zufallsvariable $\Rightarrow$ $f_{\mathcal{D}}$ ist Zufallsvariable
- Das **gelernte** Modell soll auf **zufälligen** Testpunkten $(\boldsymbol{X}, Y)$ möglichst gut funktionieren, d.h.

$$\mathbb{E}[\ell(f_{\mathcal{D}}; (\boldsymbol{X}, Y)) \mid \mathcal{D}\ ]$$

soll minimal sein.

- Aber numerische Optimierung wählt f so, dass

$$\ell(f; \mathcal{D}) \propto \frac{1}{|\mathcal{D}|} \sum_{(\boldsymbol{x}, y) \in \mathcal{D}} \ell(f; (\boldsymbol{x}, y)) = \hat{\mathbb{E}}[\ell(f; (\boldsymbol{X}, Y))]$$

minimiert wird.

Fehlertypen

- Lernen des Modells $f_{\mathcal{D}}$ auf Basis von *Trainingsdaten* $\mathcal{D}$
- Anwendung des Modells auf *Testdaten* $\mathcal{T}$
- $\mathcal{D}$ ist Zufallsvariable $\Rightarrow$ $f_{\mathcal{D}}$ ist Zufallsvariable
- Das **gelernte** Modell soll auf **zufälligen** Testpunkten $(\boldsymbol{X}, Y)$ möglichst gut funktionieren, d.h.

$$\mathbb{E}[\ell(f_{\mathcal{D}}; (\boldsymbol{X}, Y)) \mid \mathcal{D}\ ]$$

soll minimal sein.

- Aber numerische Optimierung wählt f so, dass

$$\ell(f; \mathcal{D}) \propto \frac{1}{|\mathcal{D}|} \sum_{(\boldsymbol{x}, y) \in \mathcal{D}} \ell(f; (\boldsymbol{x}, y)) = \hat{\mathbb{E}}[\ell(f; (\boldsymbol{X}, Y))]$$

minimiert wird.

Fehlertypen

- Lernen des Modells $f_{\mathcal{D}}$ auf Basis von *Trainingsdaten* $\mathcal{D}$
- Anwendung des Modells auf *Testdaten* $\mathcal{T}$
- $\mathcal{D}$ ist Zufallsvariable $\Rightarrow$ $f_{\mathcal{D}}$ ist Zufallsvariable
- Das **gelernte** Modell soll auf **zufälligen** Testpunkten $(\boldsymbol{X}, Y)$ möglichst gut funktionieren, d.h.

$$\mathbb{E}[\ell(f_{\mathcal{D}}; (\boldsymbol{X}, Y)) \mid \mathcal{D}\ ]$$

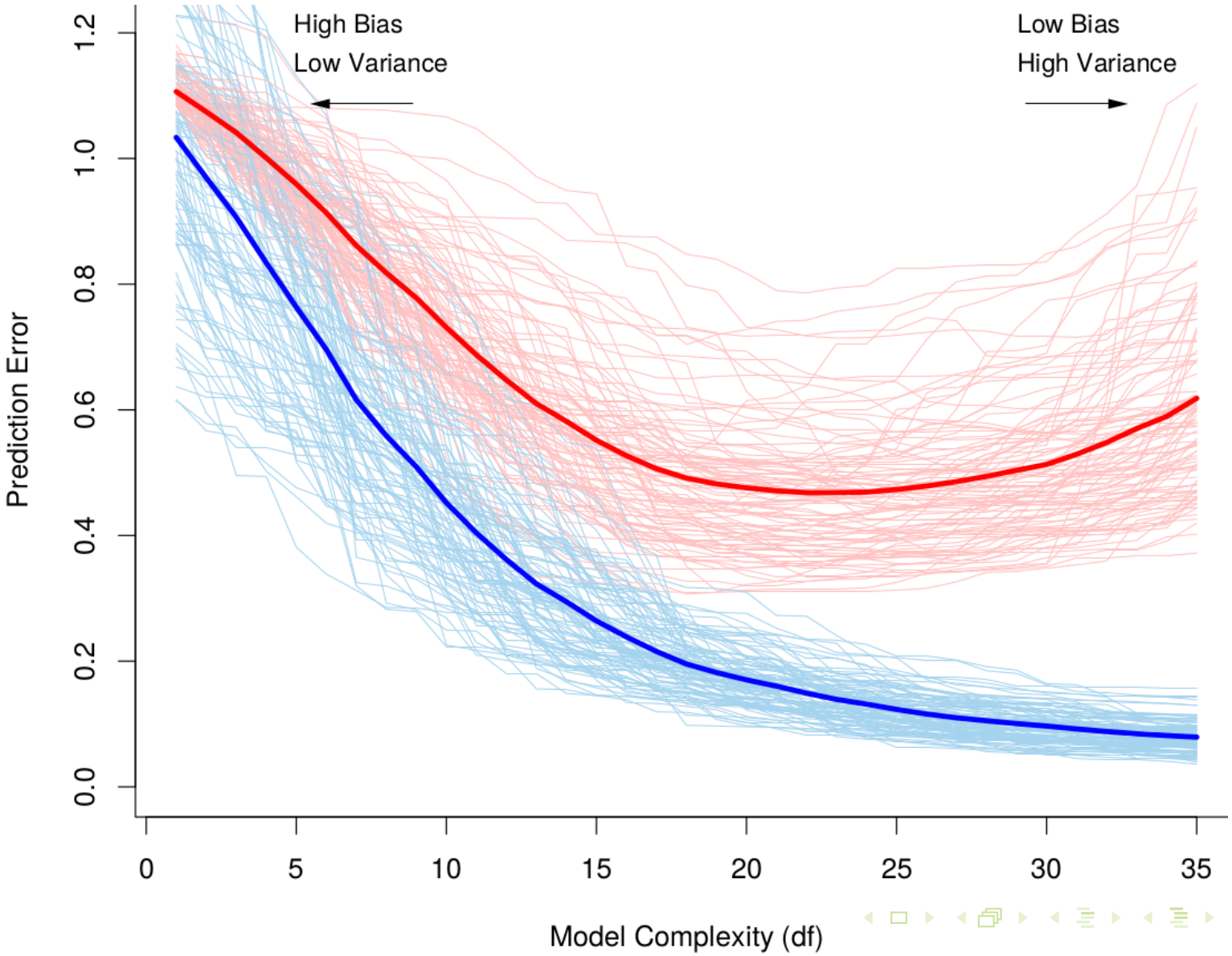
soll minimal sein.

- Aber numerische Optimierung wählt f so, dass

$$\ell(f; \mathcal{D}) \propto \frac{1}{|\mathcal{D}|} \sum_{(\boldsymbol{x}, y) \in \mathcal{D}} \ell(f; (\boldsymbol{x}, y)) = \hat{\mathbb{E}}[\ell(f; (\boldsymbol{X}, Y))]$$

minimiert wird.

Beispiel: Hellrot = $\mathbb{E}[\ell(f_{\mathcal{D}}; (\boldsymbol{X}, Y)) \mid \mathcal{D}]$, Hellblau = $\hat{\mathbb{E}}[\ell(f; (\boldsymbol{X}, Y))]$



Überanpassung

- Freiheitsgrade im linearen Modell: $\mathrm{df}_{\mathrm{RSS}}(\boldsymbol{\beta}) = d$
- Falls $d \gg n$ kann sich das Modell "perfekt" an die Daten anpassen
- Modell lernt die Daten "auswendig" ≡ *Überanpassung*
- Optimum der Verlustfunktion $\ell(f; \mathcal{D}) = \hat{\mathbb{E}}[\ell(f; (\boldsymbol{X}, Y))]$ liefert suboptimale Vorhersagen
- Hinzufügen von "Anpassungskosten" soll Überanpassung verhindern

Überanpassung

- Freiheitsgrade im linearen Modell: $\mathrm{df}_{\mathrm{RSS}}(\boldsymbol{\beta}) = d$
- Falls $d \gg n$ kann sich das Modell "perfekt" an die Daten anpassen
- Modell lernt die Daten "auswendig" $\equiv$ *Überanpassung*
- Optimum der Verlustfunktion $\ell(f; \mathcal{D}) = \hat{\mathbb{E}}[\ell(f; (\boldsymbol{X}, Y))]$ liefert suboptimale Vorhersagen
- Hinzufügen von "Anpassungskosten" soll Überanpassung verhindern

Modellkomplexität

Allgemein für lineare Modelle mit $\boldsymbol{\beta} \in \mathbb{R}^d$ und $y = f(\boldsymbol{x}) + \varepsilon$

$$\mathrm{df}(f) = \frac{1}{\sigma_\varepsilon^2} \sum_{(\boldsymbol{x},y)\in\mathcal{D}} \mathbb{C}[f(\boldsymbol{x}), y]$$

Idee: bestrafe Überanpassung mittels *Regularisierung*

$R : \mathcal{M} \to \mathbb{R}$

$$f_\mathcal{D} = \arg\min_{f\in\mathcal{M}} \ell(f;\mathcal{D}) + \lambda R(f)$$

Der Parameter $\lambda > 0$ bestimmt den Einfluss der Regularisierung.

Oft: $R(f) = \|f\|_q^q$

Modellkomplexität

Allgemein für lineare Modelle mit $\boldsymbol{\beta} \in \mathbb{R}^d$ und $y = f(\boldsymbol{x}) + \varepsilon$

$$\mathrm{df}(f) = \frac{1}{\sigma_\varepsilon^2} \sum_{(\boldsymbol{x},y)\in\mathcal{D}} \mathbb{C}[f(\boldsymbol{x}), y]$$

Idee: bestrafe Überanpassung mittels *Regularisierung*
$R : \mathcal{M} \to \mathbb{R}$

$$f_\mathcal{D} = \arg\min_{f\in\mathcal{M}} \ell(f; \mathcal{D}) + \lambda R(f)$$

Der Parameter $\lambda > 0$ bestimmt den Einfluss der Regularisierung.
Oft: $R(f) = \|f\|_q^q$

Modellkomplexität

Allgemein für lineare Modelle mit $\boldsymbol{\beta} \in \mathbb{R}^d$ und $y = f(\boldsymbol{x}) + \varepsilon$

$$\mathrm{df}(f) = \frac{1}{\sigma_\varepsilon^2} \sum_{(\boldsymbol{x},y)\in\mathcal{D}} \mathbb{C}[f(\boldsymbol{x}), y]$$

Idee: bestrafe Überanpassung mittels *Regularisierung*
$R : \mathcal{M} \to \mathbb{R}$

$$f_\mathcal{D} = \arg\min_{f\in\mathcal{M}} \ell(f;\mathcal{D}) + \lambda R(f)$$

Der Parameter $\lambda > 0$ bestimmt den Einfluss der Regularisierung.
Oft: $R(f) = \|f\|_q^q$

Regularisierung kontrolliert Modellkomplexität

Freiheitsgrade für $R_{l_2}(f) = \|\boldsymbol{\beta}\|_2^2$ und $R_{l_1}(f) = \|\boldsymbol{\beta}\|_1$:

$$\mathrm{df}_{\mathrm{RSS}+l_2}(\boldsymbol{\beta}) = \mathrm{trace}[\boldsymbol{D}(\boldsymbol{D}^\top \boldsymbol{D} + \lambda \boldsymbol{I})^{-1} \boldsymbol{D}^\top]$$

$$\mathrm{df}_{\mathrm{RSS}+l_1}(\boldsymbol{\beta}) \approx \sum_{i=1}^{d} \mathbb{1}_{\boldsymbol{\beta}_i \neq 0}$$

mit Datenmatrix $\boldsymbol{D} = (\boldsymbol{x}^1, \boldsymbol{x}^2, \ldots, \boldsymbol{x}^N)^\top$ und Einheitsmatrix $\boldsymbol{I}$.

Beispiel:

- $f(\boldsymbol{x}) = \langle \beta, \boldsymbol{x} \rangle$
- $\ell(f; \mathcal{D}) = \mathrm{RSS}(f; \mathcal{D}) = \sum_{(\boldsymbol{x},y)\in\mathcal{D}} (y - f(\boldsymbol{x}))^2$
- $\lambda = 0.1$, $\mathcal{D} = \{(0.25, 0.5)\}$

Regularisierung kontrolliert Modellkomplexität

Freiheitsgrade für $R_{l_2}(f) = \|\boldsymbol{\beta}\|_2^2$ und $R_{l_1}(f) = \|\boldsymbol{\beta}\|_1$:

$$\mathrm{df}_{\mathrm{RSS}+l_2}(\boldsymbol{\beta}) = \mathrm{trace}[\boldsymbol{D}(\boldsymbol{D}^\top \boldsymbol{D} + \lambda \boldsymbol{I})^{-1} \boldsymbol{D}^\top]$$

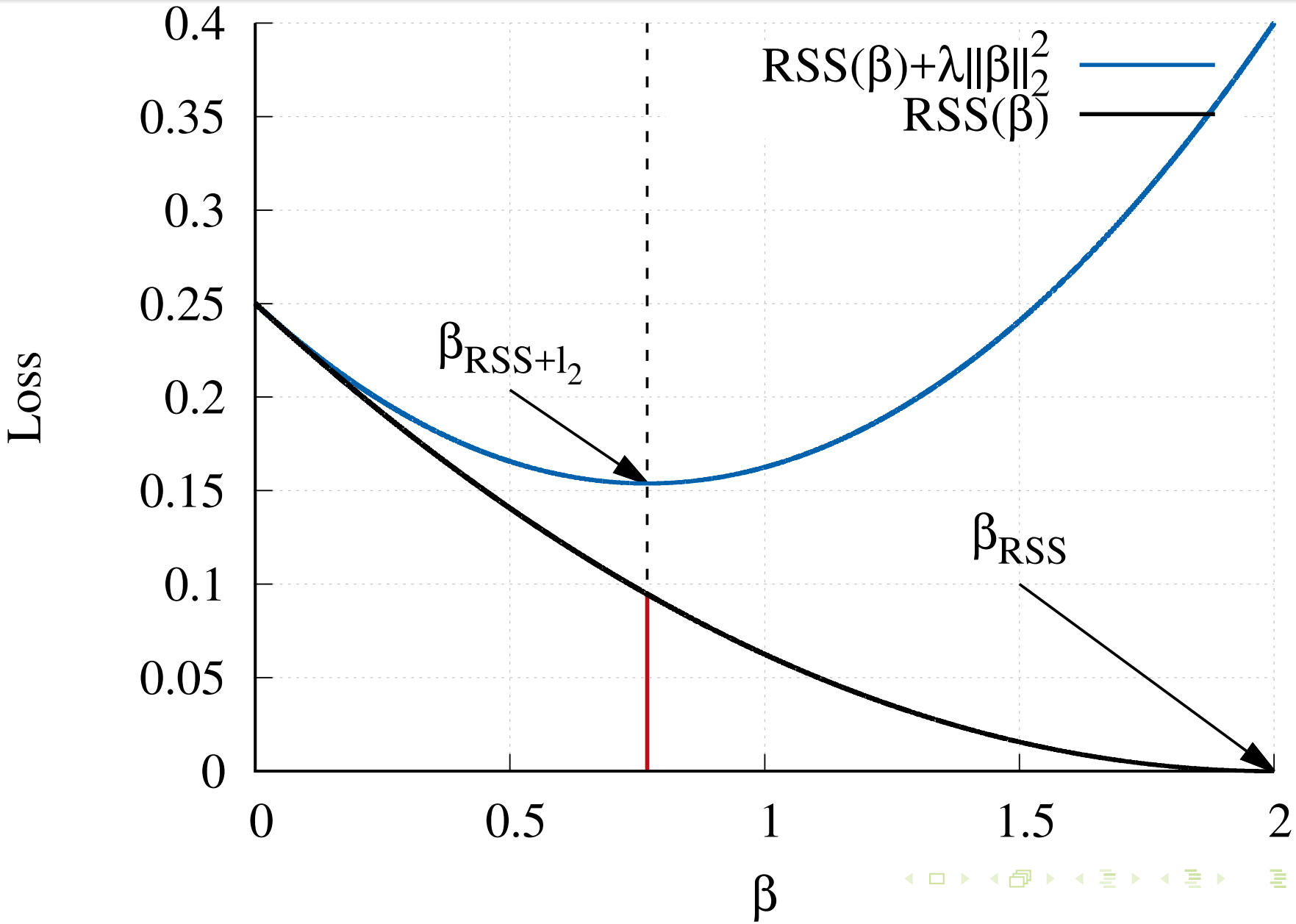
$$\mathrm{df}_{\mathrm{RSS}+l_1}(\boldsymbol{\beta}) \approx \sum_{i=1}^{d} \mathbb{1}_{\boldsymbol{\beta}_i \neq 0}$$

mit Datenmatrix $\boldsymbol{D} = (\boldsymbol{x}^1, \boldsymbol{x}^2, \ldots, \boldsymbol{x}^N)^\top$ und Einheitsmatrix $\boldsymbol{I}$.

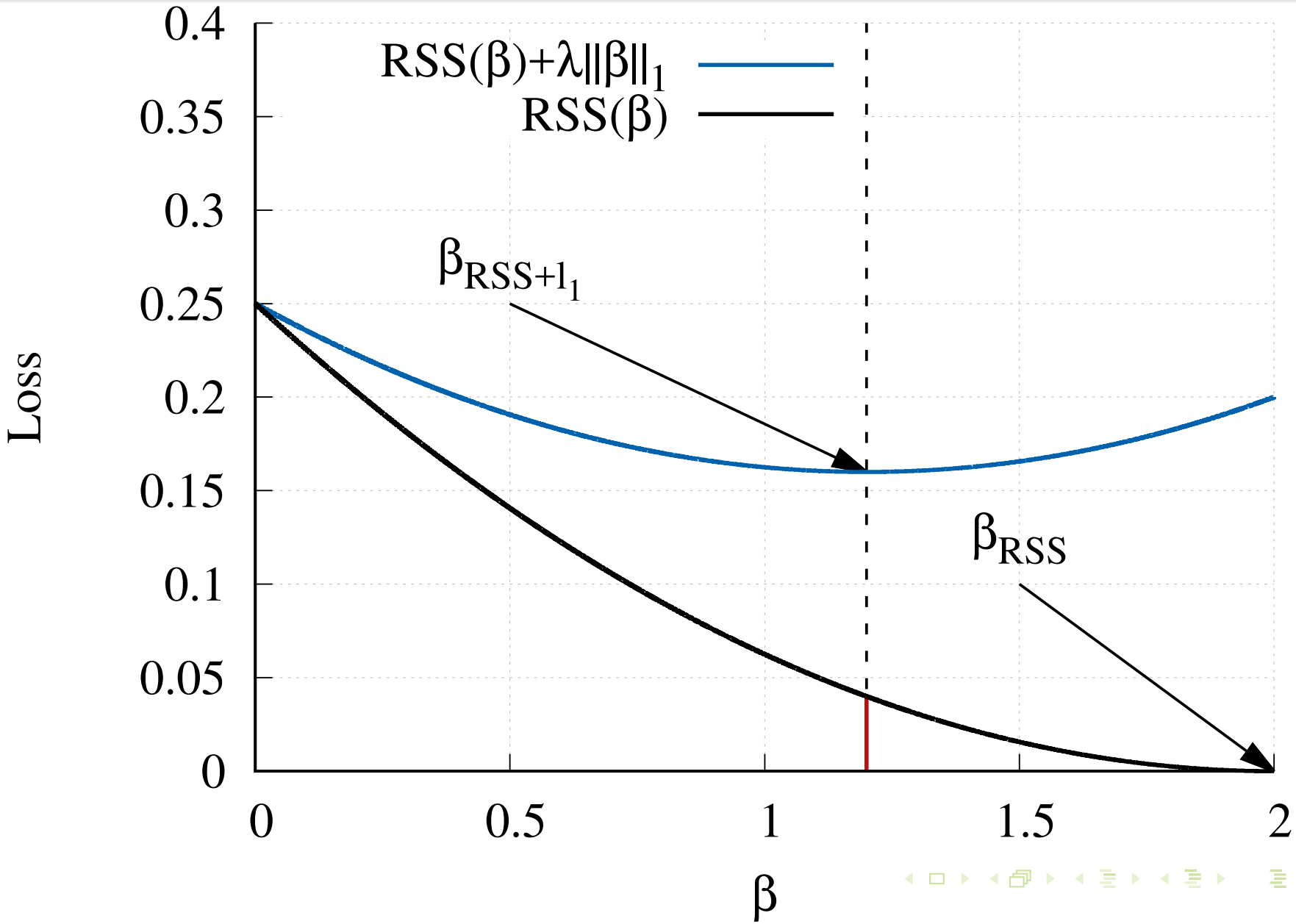
Beispiel:

- $f(\boldsymbol{x}) = \langle \boldsymbol{\beta}, \boldsymbol{x} \rangle$
- $\ell(f; \mathcal{D}) = \mathrm{RSS}(f; \mathcal{D}) = \sum_{(\boldsymbol{x},y)\in\mathcal{D}} (y - f(\boldsymbol{x}))^2$
- $\lambda = 0.1$, $\mathcal{D} = \{(0.25, 0.5)\}$

Beispiel: l_2-Regularisierung



Beispiel: l_1-Regularisierung



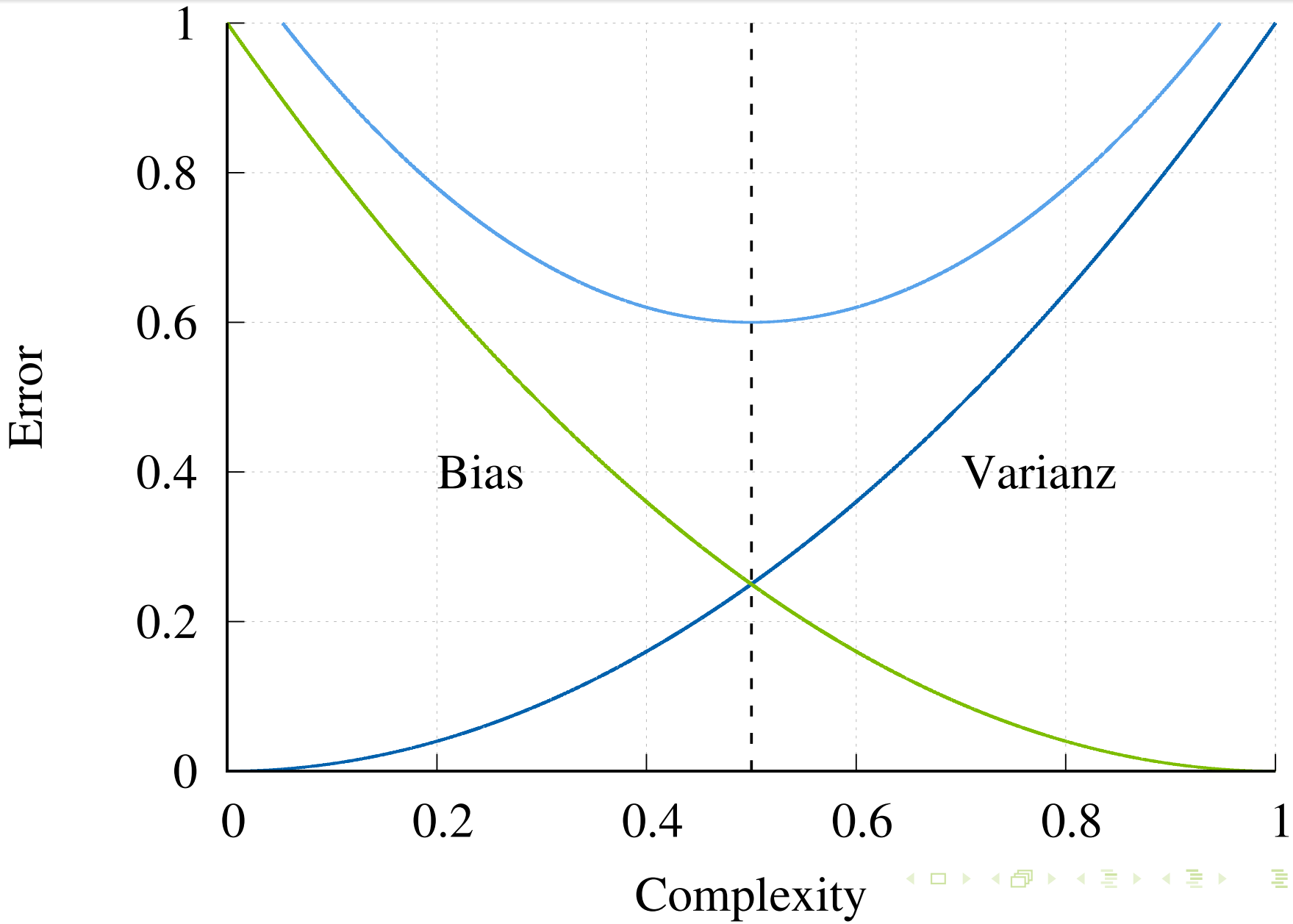
Verzerrung und Varianz

Falls $Y = f(\boldsymbol{x}) + \varepsilon$ und $\mathbb{E}[\varepsilon] = 0$, dann

$$
\begin{aligned}
\mathbb{E}[\mathrm{RSS}(f_{\mathcal{D}}; (\boldsymbol{x}, Y))] &= \mathbb{E}[(Y - f_{\mathcal{D}}(\boldsymbol{x}))^2] \\
&= \mathbb{E}[Y^2 - 2Y f_{\mathcal{D}}(\boldsymbol{x}) + f_{\mathcal{D}}(\boldsymbol{x})^2] \\
&= \mathbb{E}[(f(\boldsymbol{x}) + \varepsilon)^2] - \mathbb{E}[2Y f_{\mathcal{D}}(\boldsymbol{x})] + \mathbb{E}[f_{\mathcal{D}}(\boldsymbol{x})^2] \\
&= \mathbb{E}[f(\boldsymbol{x})^2 + 2f(\boldsymbol{x})\varepsilon + \varepsilon^2] - \mathbb{E}[2Y f_{\mathcal{D}}(\boldsymbol{x})] + \mathbb{E}[f_{\mathcal{D}}(\boldsymbol{x})^2] \\
&= f(\boldsymbol{x})^2 + \sigma_\varepsilon^2 - \mathbb{E}[2Y f_{\mathcal{D}}(\boldsymbol{x})] + \mathbb{V}[f_{\mathcal{D}}(\boldsymbol{x})] + \mathbb{E}[f_{\mathcal{D}}(\boldsymbol{x})]^2 \\
&= \sigma_\varepsilon^2 + \mathbb{V}[f_{\mathcal{D}}(\boldsymbol{x})] + \mathbb{B}[f_{\mathcal{D}}(\boldsymbol{x})]^2
\end{aligned}
$$

mit $\mathbb{B}[f_{\mathcal{D}}(\boldsymbol{x})] = f(\boldsymbol{x}) - \mathbb{E}[f_{\mathcal{D}}(\boldsymbol{x})]$ und alle Erwartungswerte sind bedingt auf $\boldsymbol{x}$.

Verzerrung und Varianz (II)



Datenbanken und Häufige Mengen

Datenbanken und SQL

Jetzt: $\mathcal{M} \subset \mathcal{D}$ bzw. $\mathcal{M} \subset \mathcal{X}$

- Relationale Datenbanken ≡ Menge von Tabellen
- Relationales Datenbankmanagementsystem erlaubt Anfrage und Manipulation von Daten mittels Structured Query Language (SQL)
- Beispielhafte Anfragen `Q`:
  - `SELECT DISTINCT x1,x2 FROM data WHERE ...`
  - `SELECT * FROM ... ORDER BY x`
  - `SELECT AVG(x) FROM ... GROUP BY y`
  - `SELECT MIN(x) FROM ... GROUP BY y`
  - `SELECT MAX(x) FROM ... GROUP BY y`
  - `SELECT COUNT(x) AS c FROM ... HAVING c>10`

Datenbanken und SQL

Jetzt: $\mathcal{M} \subset \mathcal{D}$ bzw. $\mathcal{M} \subset \mathcal{X}$

- Relationale Datenbanken ≡ Menge von Tabellen
- Relationales Datenbankmanagementsystem erlaubt Anfrage und Manipulation von Daten mittels Structured Query Language (SQL)
- Beispielhafte Anfragen `Q`:
  - `SELECT DISTINCT x1,x2 FROM data WHERE ...`
  - `SELECT * FROM ... ORDER BY x`
  - `SELECT AVG(x) FROM ... GROUP BY y`
  - `SELECT MIN(x) FROM ... GROUP BY y`
  - `SELECT MAX(x) FROM ... GROUP BY y`
  - `SELECT COUNT(x) AS c FROM ... HAVING c>10`

Anfragen und Mengen

Resultat einer Datenbankanfrage $\mathtt{Q}$ ist eine neue Tabelle $\mathtt{D}$

- Annahme: Datenbank besteht aus Binärdaten, oder wird mittels SQL Anfragen konvertiert
- Die Einträge von $\mathtt{D}$ heißen dann *Transaktionen*
- Transaktion $t \in \mathtt{D}$ entspricht Indikatorvektor einer Menge

$$00101001110100011001 0 \equiv \{x_3, x_5, x_8, x_9, x_{10}, x_{12}, x_{16}, x_{17}, x_{20}\}$$

- Die Elemente x_i der Menge nennt man auch *Items*

Anfragen und Mengen

Resultat einer Datenbankanfrage $\mathtt{Q}$ ist eine neue Tabelle $\mathtt{D}$

- Annahme: Datenbank besteht aus Binärdaten, oder wird mittels SQL Anfragen konvertiert
- Die Einträge von $\mathtt{D}$ heißen dann *Transaktionen*
- Transaktion $t \in \mathtt{D}$ entspricht Indikatorvektor einer Menge

$$00101001110100011001 0 \equiv \{x_3, x_5, x_8, x_9, x_{10}, x_{12}, x_{16}, x_{17}, x_{20}\}$$

- Die Elemente x_i der Menge nennt man auch *Items*

Ausblick: Häufige Mengen und Apriori

Frequent Itemset Mining:

- Bestimme alle Mengen von Items die in mindestens s-prozent aller Transaktionen vorkommen
  - Solche Menge heißen *häufig*
- 2^n mögliche häufige Mengen!
- Aber: Eine Menge kann nur dann häufig sein, wenn alle ihre Teilmengen häufig sind
- *Apriori*: Bottom-Up Algorithmus über Teilmengenverband zur Berechnung aller häufigen Mengen