

Wissensentdeckung in Datenbanken

Probabilistische Graphische Modelle

Nico Piatkowski und Uwe Ligges

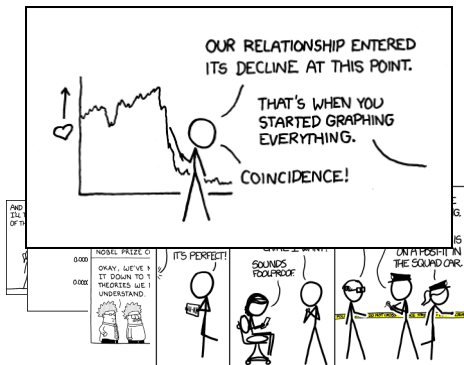
Informatik—Künstliche Intelligenz
Computergestützte Statistik
Technische Universität Dortmund

22.06.2017

Überblick

Was bisher geschah...

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



Heute

- Graphische Modelle

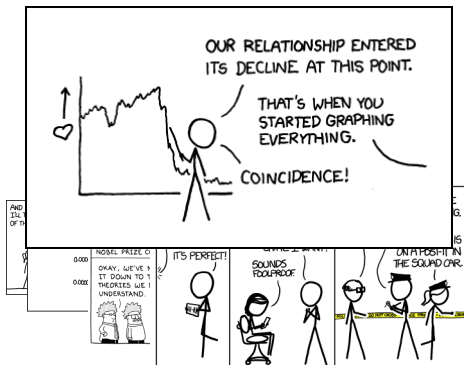
Überblick

Was bisher geschah...

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

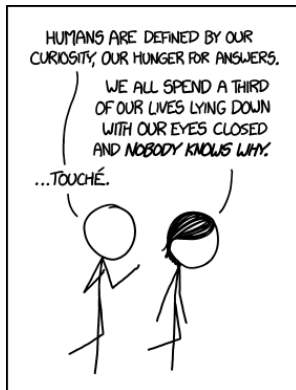
Heute

- Graphische Modelle



Überblick

- Wahrscheinlichkeiten
- Stochastische Abhängigkeiten
- Graphen
- Graphische Modelle—Intuition
- Graphische Modelle—Notation
- Anwendungsbeispiele



Erinnerung: Wahrscheinlichkeiten und Klassifikation

- Zufallsvariable X mit diskreter Domäne \mathcal{X}
- Wahrscheinlichkeit: $\mathbb{P}(X = x) \in [0; 1]$
- Normalisiert: $\sum_{x \in \mathcal{X}} \mathbb{P}(X = x) = 1$
- Bedingte Wahrscheinlichkeit:

$$\underbrace{\mathbb{P}(Y = y | X = x)} = \frac{\mathbb{P}(Y=y, X=x)}{\mathbb{P}(X=x)}$$

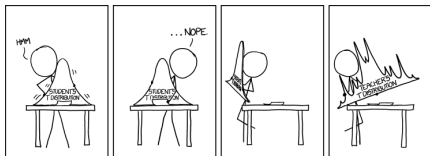
≈SVM, Log. Regr., Entscheidungsbäume, ...

- ↑↑ Klassifikation mittels ↑↑

diskriminativem Modell

- Alternative:

Generatives Modell: $\mathbb{P}(Y = y, X = x)$



Erinnerung: Wahrscheinlichkeiten und Klassifikation

- Zufallsvariable X mit diskreter Domäne \mathcal{X}
- Wahrscheinlichkeit: $\mathbb{P}(X = x) \in [0; 1]$
- Normalisiert: $\sum_{x \in \mathcal{X}} \mathbb{P}(X = x) = 1$
- Bedingte Wahrscheinlichkeit:

$$\underbrace{\mathbb{P}(Y = y | X = x)} = \frac{\mathbb{P}(Y=y, X=x)}{\mathbb{P}(X=x)}$$

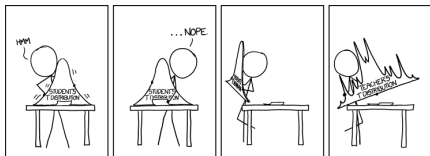
≈SVM, Log.Regr., Entscheidungsbäume, ...

- ↑↑ Klassifikation mittels ↑↑

diskriminativem Modell

- Alternative:

Generatives Modell: $\mathbb{P}(Y = y, X = x)$



Erinnerung: Wahrscheinlichkeiten und Klassifikation

- Zufallsvariable X mit diskreter Domäne \mathcal{X}
- Wahrscheinlichkeit: $\mathbb{P}(X = x) \in [0; 1]$
- Normalisiert: $\sum_{x \in \mathcal{X}} \mathbb{P}(X = x) = 1$
- Bedingte Wahrscheinlichkeit:

$$\underbrace{\mathbb{P}(Y = y | X = x)} = \frac{\mathbb{P}(Y=y, X=x)}{\mathbb{P}(X=x)}$$

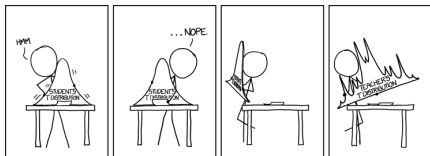
≈SVM, Log.Regr., Entscheidungsbäume, ...

- ↑↑ Klassifikation mittels ↑↑

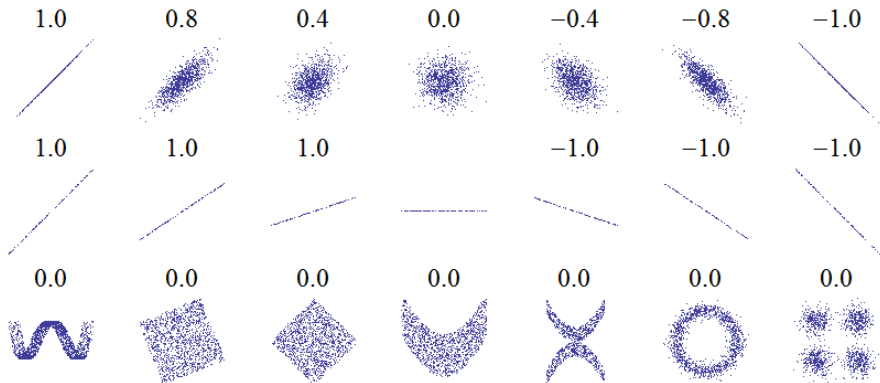
diskriminativem Modell

- Alternative:

Generatives Modell: $\mathbb{P}(Y = y, X = x)$



Exkurs: Korrelation vs. Abhängigkeit



Hier: Achsen sind Domänen reelwertiger Zufallsvariablen X, Y



Abhängigkeiten zwischen Zufallsvariablen

X, Y sind **unabhängig**, falls

$$\forall x \in \mathcal{X}, \forall y \in \mathcal{Y} : \mathbb{P}(Y = y \mid X = x) = \mathbb{P}(Y = y)$$

Unabhängigkeit kann auch nur für einige $x \in S$ mit $S \subset \mathcal{X}$ gelten!!

$$\forall x \in S, \forall y \in \mathcal{Y} : \mathbb{P}(Y = y \mid X = x) = \mathbb{P}(Y = y)$$

oder in Abhängigkeit von dritter Variable W

$$\forall x \in \mathcal{X}, \forall y \in \mathcal{Y} : \mathbb{P}(Y = y \mid X = x, W = 1) = \mathbb{P}(Y = y \mid W = 1)$$

Frage: Wie können alle Abhängigkeiten einer hochdimensionalen Zufallsvariable kompakt dargestellt (gespeichert) werden??



Abhängigkeiten zwischen Zufallsvariablen

X, Y sind **unabhängig**, falls

$$\forall x \in \mathcal{X}, \forall y \in \mathcal{Y} : \mathbb{P}(Y = y \mid X = x) = \mathbb{P}(Y = y)$$

Unabhängigkeit kann auch nur für einige $x \in S$ mit $S \subset \mathcal{X}$ gelten!!

$$\forall x \in S, \forall y \in \mathcal{Y} : \mathbb{P}(Y = y \mid X = x) = \mathbb{P}(Y = y)$$

oder in Abhängigkeit von dritter Variable W

$$\forall x \in \mathcal{X}, \forall y \in \mathcal{Y} : \mathbb{P}(Y = y \mid X = x, W = 1) = \mathbb{P}(Y = y \mid W = 1)$$

Frage: Wie können alle Abhängigkeiten einer hochdimensionalen Zufallsvariable kompakt dargestellt (gespeichert) werden??



Abhängigkeiten zwischen Zufallsvariablen

X, Y sind **unabhängig**, falls

$$\forall x \in \mathcal{X}, \forall y \in \mathcal{Y} : \mathbb{P}(Y = y \mid X = x) = \mathbb{P}(Y = y)$$

Unabhängigkeit kann auch nur für einige $x \in S$ mit $S \subset \mathcal{X}$ gelten!!

$$\forall x \in S, \forall y \in \mathcal{Y} : \mathbb{P}(Y = y \mid X = x) = \mathbb{P}(Y = y)$$

oder in Abhängigkeit von dritter Variable W

$$\forall x \in \mathcal{X}, \forall y \in \mathcal{Y} : \mathbb{P}(Y = y \mid X = x, W = 1) = \mathbb{P}(Y = y \mid W = 1)$$

Frage: Wie können alle Abhängigkeiten einer hochdimensionalen Zufallsvariable kompakt dargestellt (gespeichert) werden??



Abhängigkeiten zwischen Zufallsvariablen

X, Y sind **unabhängig**, falls

$$\forall x \in \mathcal{X}, \forall y \in \mathcal{Y} : \mathbb{P}(Y = y \mid X = x) = \mathbb{P}(Y = y)$$

Unabhängigkeit kann auch nur für einige $x \in S$ mit $S \subset \mathcal{X}$ gelten!!

$$\forall x \in S, \forall y \in \mathcal{Y} : \mathbb{P}(Y = y \mid X = x) = \mathbb{P}(Y = y)$$

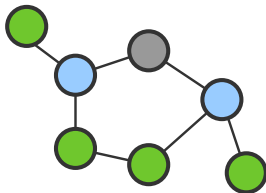
oder in Abhängigkeit von dritter Variable W

$$\forall x \in \mathcal{X}, \forall y \in \mathcal{Y} : \mathbb{P}(Y = y \mid X = x, W = 1) = \mathbb{P}(Y = y \mid W = 1)$$

Frage: Wie können alle Abhängigkeiten einer hochdimensionalen Zufallsvariable kompakt dargestellt (gespeichert) werden??

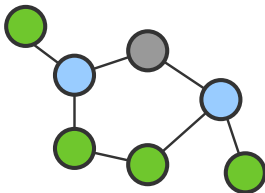
Graphen

- Möglicherweise wichtigstes abstraktes Werkzeug der Informatik
- Erlaubt Repräsentation beliebiger Beziehungen (Relationen) zwischen beliebigen Objekten
- Formal spezifiziert via $G = (V, E)$ mit Knotenmenge V und Kantenmenge E
- Auch Zusatzinformationen möglich wie Knoten- oder Kantengewichte, Farben, ...



Graphen

- Möglicherweise wichtigstes abstraktes Werkzeug der Informatik
- Erlaubt Repräsentation beliebiger Beziehungen (Relationen) zwischen beliebigen Objekten
- Formal spezifiziert via $G = (V, E)$ mit Knotenmenge V und Kantenmenge E
- Auch Zusatzinformationen möglich wie Knoten- oder Kantengewichte, Farben, ...



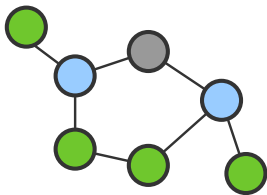
Probabilistische Graphische Modelle

Graph $G = (V, E)$, $|V|$ -dimensionale Zufallsvariable X

- Indexierung der Zufallsvariable mittels Knoten(mengen)
- Kodiere Liste aller bedingten Unabhängigkeiten als Nachbarschaften in einem Graph
- Kantenmenge E entspricht bedingten Unabhängigkeiten gemäß **Markov Eigenschaft**:

$$X_v \perp\!\!\!\perp X_{V \setminus \mathcal{N}(v) \cup \{v\}} \mid X_{\mathcal{N}(v)} = x_{\mathcal{N}(v)}$$

Für alle Knotenpaare $v, u \in V$ gilt:
 Gibt es einen Pfad durch G auf dem alle Knoten unbeobachtet sind, so sind X_v und X_u nicht unabhängig ($X_v \not\perp X_u$).



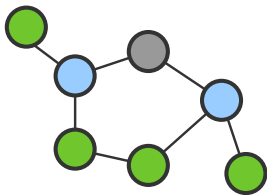
Probabilistische Graphische Modelle

Graph $G = (V, E)$, $|V|$ -dimensionale Zufallsvariable X

- Indexierung der Zufallsvariable mittels Knoten(mengen)
- Kodiere Liste aller bedingten Unabhängigkeiten als Nachbarschaften in einem Graph
- Kantenmenge E entspricht bedingten Unabhängigkeiten gemäß **Markov Eigenschaft**:

$$X_v \perp\!\!\!\perp X_{V \setminus \mathcal{N}(v) \cup \{v\}} \mid X_{\mathcal{N}(v)} = \mathbf{x}_{\mathcal{N}(v)}$$

Für alle Knotenpaare $v, u \in V$ gilt:
 Gibt es einen Pfad durch G auf dem alle Knoten unbeobachtet sind, so sind X_v und X_u nicht unabhängig ($X_v \not\perp X_u$).



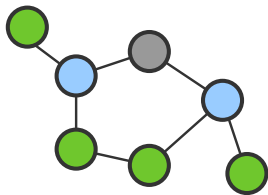
Probabilistische Graphische Modelle

Graph $G = (V, E)$, $|V|$ -dimensionale Zufallsvariable X

- Indexierung der Zufallsvariable mittels Knoten(mengen)
- Kodiere Liste aller bedingten Unabhängigkeiten als Nachbarschaften in einem Graph
- Kantenmenge E entspricht bedingten Unabhängigkeiten gemäß **Markov Eigenschaft**:

$$X_v \perp\!\!\!\perp X_{V \setminus \mathcal{N}(v) \cup \{v\}} \mid X_{\mathcal{N}(v)} = x_{\mathcal{N}(v)}$$

Für alle Knotenpaare $v, u \in V$ gilt:
 Gibt es einen Pfad durch G auf dem alle Knoten unbeobachtet sind, so sind X_v und X_u nicht unabhängig ($X_v \not\perp X_u$).



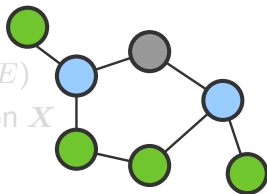
Probabilistische Graphische Modelle (II)

Graph $G = (V, E)$, $|V|$ -dimensionale Zufallsvariable \mathbf{X}

Fakt: Wahrscheinlichkeits(dichte) faktorisiert über den (maximalen) Cliques des Graph

$$\mathbb{P}(\mathbf{X} = \mathbf{x}) = \frac{1}{Z} \prod_{C \in \mathcal{C}(G)} \psi_C(\mathbf{x}_C)$$

- $A \subseteq V$ ist Clique $\Leftrightarrow (\forall \{v, u\} \subseteq A \Rightarrow \{v, u\} \in E)$
- Z ist Summe über **alle** möglichen Werte von \mathbf{X}
- Jedes $\psi_C : \mathcal{X}_C \rightarrow \mathbb{R}^+$ mit diskretem Zustandsraum \mathcal{X}_C ist darstellbar als $\exp(\langle \beta_C, \phi_C(\cdot) \rangle)$
- Dichten der Form $\mathbb{P}(\mathbf{X} = \mathbf{x}) = \frac{1}{Z} \prod_{C \in \mathcal{C}(G)} \exp(\langle \beta_C, \phi_C(\cdot) \rangle)$ gehören zu einer **Exponentialfamilie**



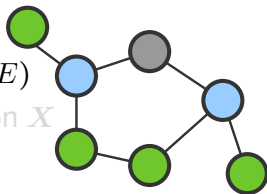
Probabilistische Graphische Modelle (II)

Graph $G = (V, E)$, $|V|$ -dimensionale Zufallsvariable \mathbf{X}

Fakt: Wahrscheinlichkeits(dichte) faktorisiert über den (maximalen) Cliques des Graph

$$\mathbb{P}(\mathbf{X} = \mathbf{x}) = \frac{1}{Z} \prod_{C \in \mathcal{C}(G)} \psi_C(\mathbf{x}_C)$$

- $A \subseteq V$ ist **Clique** $\Leftrightarrow (\forall \{v, u\} \subseteq A \Rightarrow \{v, u\} \in E)$
- Z ist Summe über **alle** möglichen Werte von \mathbf{X}
- Jedes $\psi_C : \mathcal{X}_C \rightarrow \mathbb{R}^+$ mit diskretem Zustandsraum \mathcal{X}_C ist darstellbar als $\exp(\langle \beta_C, \phi_C(\cdot) \rangle)$
- Dichten der Form $\mathbb{P}(\mathbf{X} = \mathbf{x}) = \frac{1}{Z} \prod_{C \in \mathcal{C}(G)} \exp(\langle \beta_C, \phi_C(\cdot) \rangle)$ gehören zu einer **Exponentialfamilie**



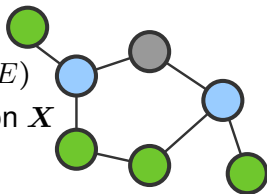
Probabilistische Graphische Modelle (II)

Graph $G = (V, E)$, $|V|$ -dimensionale Zufallsvariable \mathbf{X}

Fakt: Wahrscheinlichkeits(dichte) faktorisiert über den (maximalen) Cliques des Graph

$$\mathbb{P}(\mathbf{X} = \mathbf{x}) = \frac{1}{Z} \prod_{C \in \mathcal{C}(G)} \psi_C(\mathbf{x}_C)$$

- $A \subseteq V$ ist Clique $\Leftrightarrow (\forall \{v, u\} \subseteq A \Rightarrow \{v, u\} \in E)$
- Z ist Summe über **alle** möglichen Werte von \mathbf{X}
- Jedes $\psi_C : \mathcal{X}_C \rightarrow \mathbb{R}^+$ mit diskretem Zustandsraum \mathcal{X}_C ist darstellbar als $\exp(\langle \beta_C, \phi_C(\cdot) \rangle)$
- Dichten der Form $\mathbb{P}(\mathbf{X} = \mathbf{x}) = \frac{1}{Z} \prod_{C \in \mathcal{C}(G)} \exp(\langle \beta_C, \phi_C(\cdot) \rangle)$ gehören zu einer **Exponentialfamilie**





Parameterlernen in Graphischen Modellen

Gegeben Datensatz \mathcal{D} Verlustfunktion: Negative mittlere log-Likelihood

$$\begin{aligned} \ell(\boldsymbol{\beta}, \mathcal{D}) &= -\frac{1}{|\mathcal{D}|} \sum_{\mathbf{x} \in \mathcal{D}} \log \mathbb{P}_{\boldsymbol{\beta}}(\mathbf{x}) \\ &= -\frac{1}{|\mathcal{D}|} \sum_{\mathbf{x} \in \mathcal{D}} \log \left(\frac{1}{Z} \prod_{C \in \mathcal{C}(G)} \psi_C(\mathbf{x}_C) \right) \\ &= -\frac{1}{|\mathcal{D}|} \sum_{\mathbf{x} \in \mathcal{D}} \log \left(\frac{1}{Z} \prod_{C \in \mathcal{C}(G)} \exp(\langle \boldsymbol{\beta}_C, \boldsymbol{\phi}_C(\mathbf{x}_C) \rangle) \right) \\ &= \left(-\frac{1}{|\mathcal{D}|} \sum_{\mathbf{x} \in \mathcal{D}} \langle \boldsymbol{\beta}, \boldsymbol{\phi}(\mathbf{x}) \rangle \right) + \underbrace{\log Z}_{\text{Komplexität: \#P-vollständig}} \end{aligned}$$



Parameterlernen in Graphischen Modellen

Gegeben Datensatz \mathcal{D} Verlustfunktion: Negative mittlere log-Likelihood

$$\begin{aligned} \ell(\beta, \mathcal{D}) &= -\frac{1}{|\mathcal{D}|} \sum_{\mathbf{x} \in \mathcal{D}} \log \mathbb{P}_{\beta}(\mathbf{x}) \\ &= -\frac{1}{|\mathcal{D}|} \sum_{\mathbf{x} \in \mathcal{D}} \log \left(\frac{1}{Z} \prod_{C \in \mathcal{C}(G)} \psi_C(\mathbf{x}_C) \right) \\ &= -\frac{1}{|\mathcal{D}|} \sum_{\mathbf{x} \in \mathcal{D}} \log \left(\frac{1}{Z} \prod_{C \in \mathcal{C}(G)} \exp(\langle \beta_C, \phi_C(\mathbf{x}_C) \rangle) \right) \\ &= \left(-\frac{1}{|\mathcal{D}|} \sum_{\mathbf{x} \in \mathcal{D}} \langle \beta, \phi(\mathbf{x}) \rangle \right) + \underbrace{\log Z}_{\text{Komplexität: \#P-vollständig}} \end{aligned}$$



Parameterlernen in Graphischen Modellen

Gegeben Datensatz \mathcal{D} Verlustfunktion: Negative mittlere log-Likelihood

$$\begin{aligned} \ell(\boldsymbol{\beta}, \mathcal{D}) &= -\frac{1}{|\mathcal{D}|} \sum_{\mathbf{x} \in \mathcal{D}} \log \mathbb{P}_{\boldsymbol{\beta}}(\mathbf{x}) \\ &= -\frac{1}{|\mathcal{D}|} \sum_{\mathbf{x} \in \mathcal{D}} \log \left(\frac{1}{Z} \prod_{C \in \mathcal{C}(G)} \psi_C(\mathbf{x}_C) \right) \\ &= -\frac{1}{|\mathcal{D}|} \sum_{\mathbf{x} \in \mathcal{D}} \log \left(\frac{1}{Z} \prod_{C \in \mathcal{C}(G)} \exp(\langle \boldsymbol{\beta}_C, \phi_C(\mathbf{x}_C) \rangle) \right) \\ &= \left(-\frac{1}{|\mathcal{D}|} \sum_{\mathbf{x} \in \mathcal{D}} \langle \boldsymbol{\beta}, \phi(\mathbf{x}) \rangle \right) + \underbrace{\log Z}_{\text{Komplexität: \#P-vollständig}} \end{aligned}$$

Parameterlernen in Graphischen Modellen

Gegeben Datensatz \mathcal{D} Verlustfunktion: Negative mittlere log-Likelihood

$$\begin{aligned}
 \ell(\boldsymbol{\beta}, \mathcal{D}) &= -\frac{1}{|\mathcal{D}|} \sum_{\mathbf{x} \in \mathcal{D}} \log \mathbb{P}_{\boldsymbol{\beta}}(\mathbf{x}) \\
 &= -\frac{1}{|\mathcal{D}|} \sum_{\mathbf{x} \in \mathcal{D}} \log \left(\frac{1}{Z} \prod_{C \in \mathcal{C}(G)} \psi_C(\mathbf{x}_C) \right) \\
 &= -\frac{1}{|\mathcal{D}|} \sum_{\mathbf{x} \in \mathcal{D}} \log \left(\frac{1}{Z} \prod_{C \in \mathcal{C}(G)} \exp(\langle \boldsymbol{\beta}_C, \phi_C(\mathbf{x}_C) \rangle) \right) \\
 &= \left(-\frac{1}{|\mathcal{D}|} \sum_{\mathbf{x} \in \mathcal{D}} \langle \boldsymbol{\beta}, \phi(\mathbf{x}) \rangle \right) + \underbrace{\log Z}_{\text{Komplexität: \#P-vollständig}}
 \end{aligned}$$



Exkurs: Allgemeine Vorgehensweise bei Graphischen Modellen

Gegeben Datensatz \mathcal{D}

- 1 Modellselektion; [Hier: \equiv Wähle graphische Struktur (und ϕ)]
- 2 Aufteilen der Daten in Trainings und Testdaten
- 3 Lerne Modellparameter β auf den Trainingsdaten \mathcal{D} (z.B. mit Gradientenabstieg)
- 4 Vorhersage beliebiger Variablen $X_U \mid X_{V \setminus U}$ auf Testdaten und berechne die Vorhersagegenauigkeit
- 5 Wiederhole Schritte 2-4 so oft wie möglich und berechne mittleren Vorhersagefehler
- 6 Entscheide ob Schritt 1 gut genug war..



Exkurs: Allgemeine Vorgehensweise bei Graphischen Modellen

Gegeben Datensatz \mathcal{D}

- 1 Modellselektion; [Hier: \equiv Wähle graphische Struktur (und ϕ)]
- 2 Aufteilen der Daten in Trainings und Testdaten
- 3 Lerne Modellparameter β auf den Trainingsdaten \mathcal{D} (z.B. mit Gradientenabstieg)
- 4 Vorhersage beliebiger Variablen $X_U \mid X_{V \setminus U}$ auf Testdaten und berechne die Vorhersagegenauigkeit
- 5 Wiederhole Schritte 2-4 so oft wie möglich und berechne mittleren Vorhersagefehler
- 6 Entscheide ob Schritt 1 gut genug war..



Exkurs: Allgemeine Vorgehensweise bei Graphischen Modellen

Gegeben Datensatz \mathcal{D}

- 1 Modellselektion; [Hier: \equiv Wähle graphische Struktur (und ϕ)]
- 2 Aufteilen der Daten in Trainings und Testdaten
- 3 Lerne Modellparameter β auf den Trainingsdaten \mathcal{D} (z.B. mit Gradientenabstieg)
- 4 Vorhersage beliebiger Variablen $X_U \mid X_{V \setminus U}$ auf Testdaten und berechne die Vorhersagegenauigkeit
- 5 Wiederhole Schritte 2-4 so oft wie möglich und berechne mittleren Vorhersagefehler
- 6 Entscheide ob Schritt 1 gut genug war..



Exkurs: Allgemeine Vorgehensweise bei Graphischen Modellen

Gegeben Datensatz \mathcal{D}

- 1 Modellselektion; [Hier: \equiv Wähle graphische Struktur (und ϕ)]
- 2 Aufteilen der Daten in Trainings und Testdaten
- 3 Lerne Modellparameter β auf den Trainingsdaten \mathcal{D} (z.B. mit Gradientenabstieg)
- 4 Vorhersage beliebiger Variablen $X_U \mid X_{V \setminus U}$ auf Testdaten und berechne die Vorhersagegenauigkeit
- 5 Wiederhole Schritte 2-4 so oft wie möglich und berechne mittleren Vorhersagefehler
- 6 Entscheide ob Schritt 1 gut genug war..



Exkurs: Allgemeine Vorgehensweise bei Graphischen Modellen

Gegeben Datensatz \mathcal{D}

- 1 Modellselektion; [Hier: \equiv Wähle graphische Struktur (und ϕ)]
- 2 Aufteilen der Daten in Trainings und Testdaten
- 3 Lerne Modellparameter β auf den Trainingsdaten \mathcal{D} (z.B. mit Gradientenabstieg)
- 4 Vorhersage beliebiger Variablen $X_U \mid X_{V \setminus U}$ auf Testdaten und berechne die Vorhersagegenauigkeit
- 5 Wiederhole Schritte 2-4 so oft wie möglich und berechne mittleren Vorhersagefehler
- 6 Entscheide ob Schritt 1 gut genug war..



Exkurs: Allgemeine Vorgehensweise bei Graphischen Modellen

Gegeben Datensatz \mathcal{D}

- 1 Modellselektion; [Hier: \equiv Wähle graphische Struktur (und ϕ)]
- 2 Aufteilen der Daten in Trainings und Testdaten
- 3 Lerne Modellparameter β auf den Trainingsdaten \mathcal{D} (z.B. mit Gradientenabstieg)
- 4 Vorhersage beliebiger Variablen $X_U \mid X_{V \setminus U}$ auf Testdaten und berechne die Vorhersagegenauigkeit
- 5 Wiederhole Schritte 2-4 so oft wie möglich und berechne mittleren Vorhersagefehler
- 6 Entscheide ob Schritt 1 gut genug war..

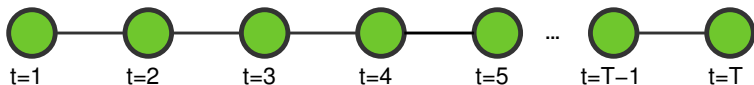
Nutzung des Mobilfunknetzes

- Angemeldete Mobilfunkzelle ist dem Mobiltelefon bekannt.
- **Idee:** Modell der Zellnutzung über den **Tag**.
- Jeder Knoten $t \in V$ entspricht der aktuellen Zelle zur Zeit t
- Zeitliche Auflösung ist fest, z.B.: 15min. Insgesamt:
 $|V| = T = 96$

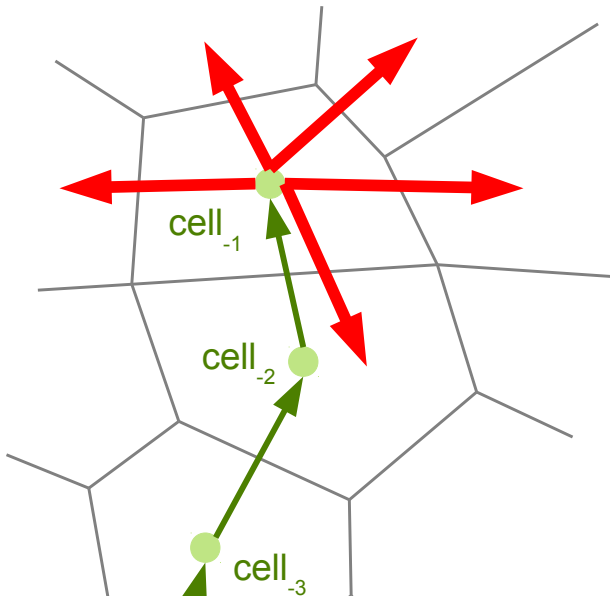


Nutzung des Mobilfunknetzes

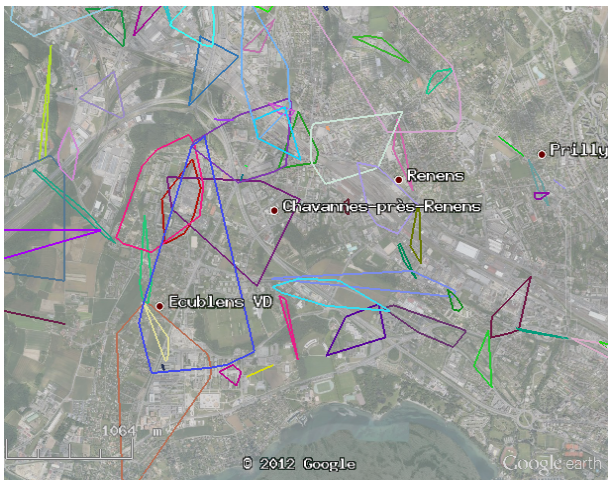
- Angemeldete Mobilfunkzelle ist dem Mobiltelefon bekannt.
- **Idee:** Modell der Zellnutzung über den **Tag**.
- Jeder Knoten $t \in V$ entspricht der aktuellen Zelle zur Zeit t
- Zeitliche Auflösung ist fest, z.B.: 15min. Insgesamt:
 $|V| = T = 96$



Netzwerkzellen (idealisiert)



Netzwerkzellen (geschätzt)





Suffiziente Statistik für Netzwerkzellen

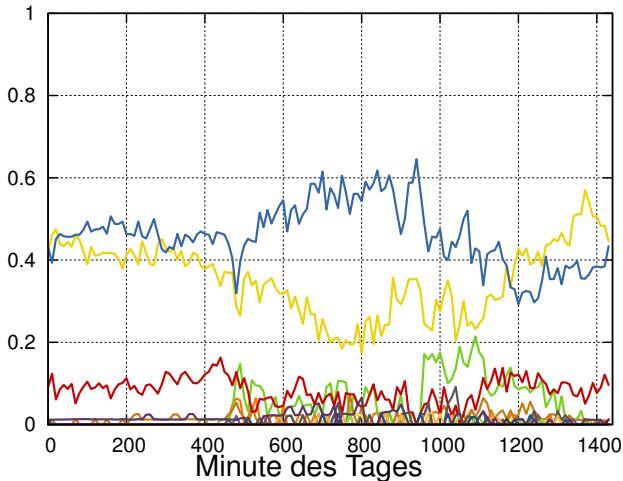
- Daten werden fortlaufend aufgezeichnet aber **nur die suffiziente Statistik** $\phi(\mathcal{D})$ muss gespeichert werden!
- Für Knoten $t \in V$ und Kanten $(i, i+1) \in E$:

$$\phi_t(\mathbf{x}) = \begin{pmatrix} \mathbb{1}\{\text{Zelle } 1 \text{ Zeit } t\} \\ \dots \\ \mathbb{1}\{\text{Zelle } M \text{ Zeit } t\} \end{pmatrix}$$

$$\phi_{(i,i+1)}(\mathbf{x}) = \begin{pmatrix} \mathbb{1}\{\text{Zelle } 1 \text{ Zeit } i\} & \mathbb{1}\{\text{Zelle } 1 \text{ Zeit } i+1\} \\ \mathbb{1}\{\text{Zelle } 1 \text{ Zeit } i\} & \mathbb{1}\{\text{Zelle } 2 \text{ Zeit } i+1\} \\ \dots & \dots \\ \mathbb{1}\{\text{Zelle } M \text{ Zeit } i\} & \mathbb{1}\{\text{Zelle } M \text{ Zeit } i+1\} \end{pmatrix}$$

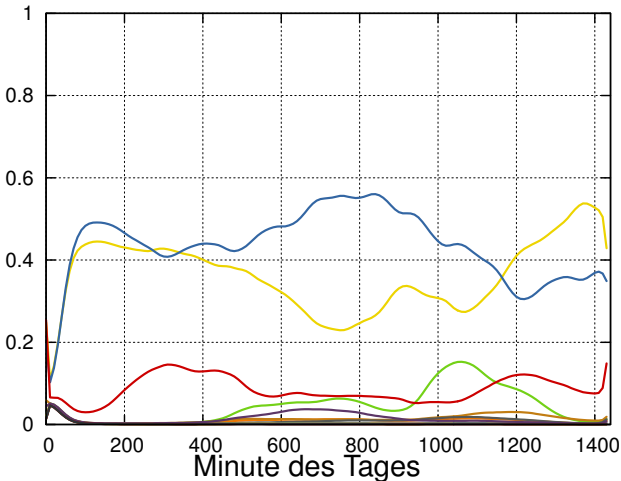
$\phi(\mathcal{D})$

- Eine Farbe pro Zelle



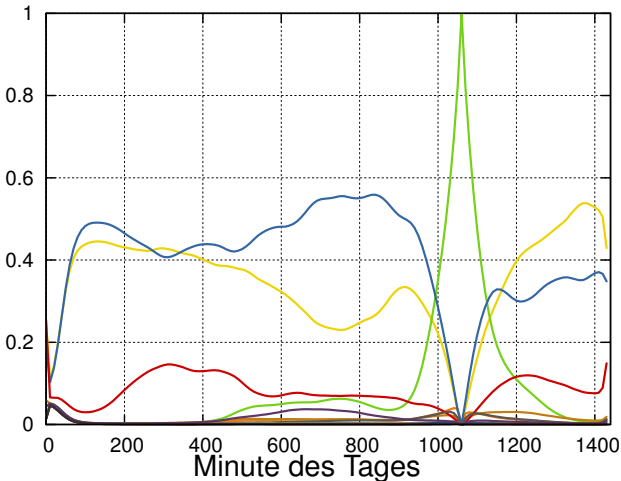
$$\mathbb{P}(X_t)$$

- Eine Farbe pro Zelle

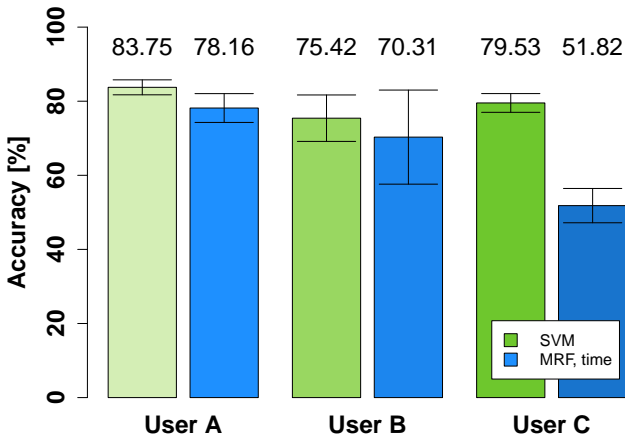


$$\mathbb{P}(X_t \mid X_{17:30-17:45} = \text{“grüne Zelle”})$$

- Eine Farbe pro Zelle



Vorhersage Ergebnisse



- SVM hat “mehr Daten”: $\underbrace{\mathbb{P}(\mathbf{X}, \mathbf{Y})}_{\text{MRF}} = \underbrace{\mathbb{P}(\mathbf{Y} | \mathbf{X})}_{\approx \text{Klassifikation}} \mathbb{P}(\mathbf{X})$



Ressourcennutzung I

Auf dem Gerät:

- Wiederholte Verbindungsversuche verbrauchen viel Energie
- ⇒ Vorhersage der Netzverfügbarkeit verwenden um Datentransfers zu verschieben

Beim Provider:

- Vorhersage der Gesamtauslastung von Mobilfunkzellen

$$\mathbb{E}[\text{\#Nutzer in Zelle } i \text{ um Zeit } t] = \sum_{u \in \text{Nutzer}} p_u(x_t = i)$$

- Vorhersage der Anzahl von Nutzern die im nächsten Zeitpunkt in eine bestimmte Zelle wechseln werden



Ressourcennutzung I

Auf dem Gerät:

- Wiederholte Verbindungsversuche verbrauchen viel Energie
- ⇒ Vorhersage der Netzverfügbarkeit verwenden um Datentransfers zu verschieben

Beim Provider:

- Vorhersage der Gesamtauslastung von Mobilfunkzellen

$$\mathbb{E}[\text{\#Nutzer in Zelle } i \text{ um Zeit } t] = \sum_{u \in \text{Nutzer}} p_u(x_t = i)$$

- Vorhersage der Anzahl von Nutzern die im nächsten Zeitpunkt in eine bestimmte Zelle wechseln werden

Apps



- Vorhersage der App Nutzung ermöglicht:

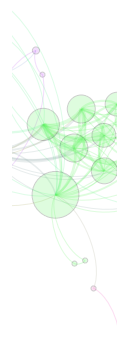
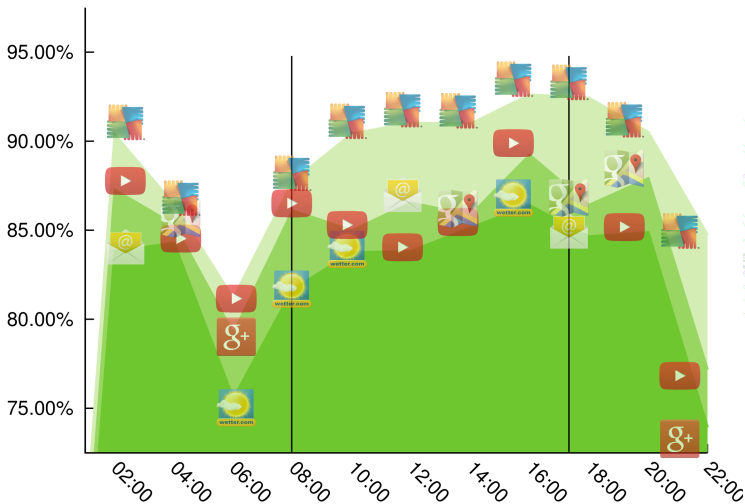
- Verbesserte Ressourcennutzung
- Verbesserte Vorhersage verbleibender Ressourcen

- Installierte Apps sind immer bekannt

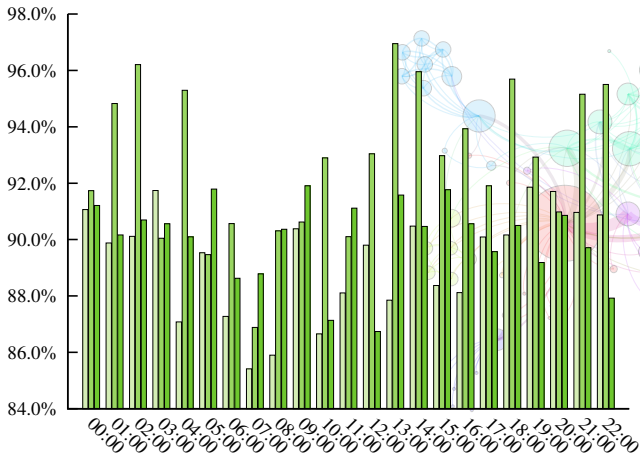
- Jede App läuft oder nicht (binärer Zustandsraum)

- Graph wird aus Daten bestimmt (nächste Woche)

Apps über die Zeit



App Modell



Smarte Ressourcennutzung

- Apps bestimmen den Ressourcenverbrauch (Batterie, Netzwerk)
 - Vorhersage: “offline-Apps” genutzt \Rightarrow Trenne Internetverbindung oder Wechsel zu Netz mit geringerer Bandbreite
- Batterieverbrauch einzelner Apps geschätzt per Regression auf Gesamtverbrauch
 - Vorhersage der App-Nutzung kann dann für eine verbesserte Batterielaufzeitprognose verwendet werden (auch: E-Auto, Roboter, Drohnen, usw.)

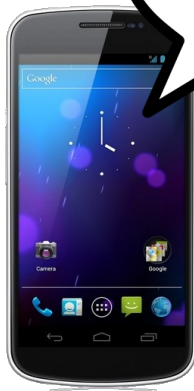
$$\mathbb{E}[\text{Batterie Nutzung zur Zeit } t] = \sum_{\text{App}} \mathbb{P}(\text{App läuft um } t) \cdot \text{Batterie}(\text{App})$$

Smarte Ressourcennutzung

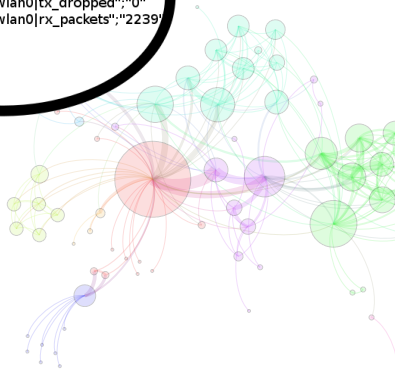
- Apps bestimmen den Ressourcenverbrauch (Batterie, Netzwerk)
 - Vorhersage: “offline-Apps” genutzt \Rightarrow Trenne Internetverbindung oder Wechsel zu Netz mit geringerer Bandbreite
- Batterieverbrauch einzelner Apps geschätzt per Regression auf Gesamtverbrauch
 - Vorhersage der App-Nutzung kann dann für eine verbesserte Batterielaufzeitprognose verwendet werden (auch: E-Auto, Roboter, Drohnen, usw.)

$$\mathbb{E}[\text{Batterie Nutzung zur Zeit } t] = \sum_{\text{App}} \mathbb{P}(\text{App läuft um } t) \cdot \text{Batterie}(\text{App})$$

Smarte Geräte



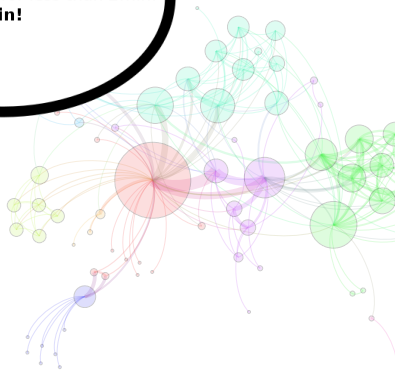
1346357007381;"battery|level";"66"
1346357007676;"memory|MemFree";"39024"
1346357007858;"net|wlan0|tx_packets";"2458"
1346357007858;"net|wlan0|tx_dropped";"0"
1346357007858;"net|wlan0|rx_packets";"2239"



Smarte Geräte



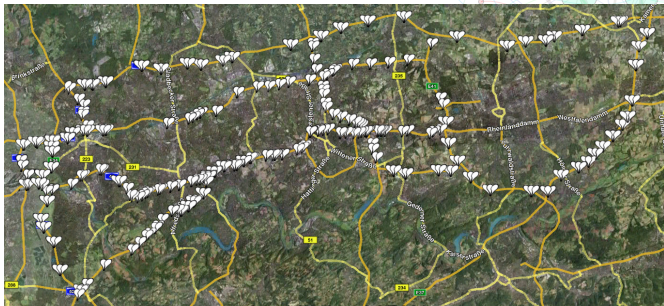
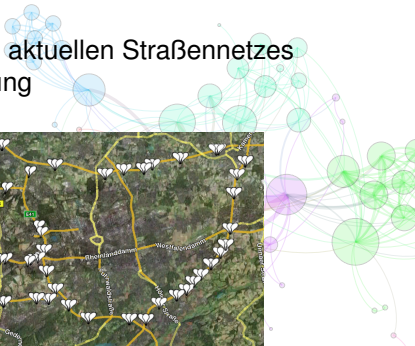
If you start the game now, your battery will retain for less than 2min.
Otherwise, 35min!



Straßennetze I

- Daten:** Sensoren messen Verkehrsfluss und Verkehrsdichte

- Verstehen und verbessern des aktuellen Straßennetzes
- Unterstützung für Routenplanung



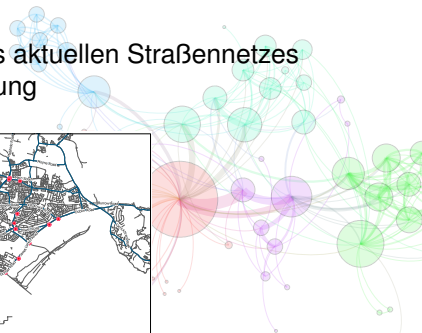
Autobahn.NRW

Straßennetze II

- Daten:** Sensoren messen Verkehrsfluss und Verkehrsdichte
 - Verstehen und verbessern des aktuellen Straßennetzes
 - Unterstützung für Routenplanung



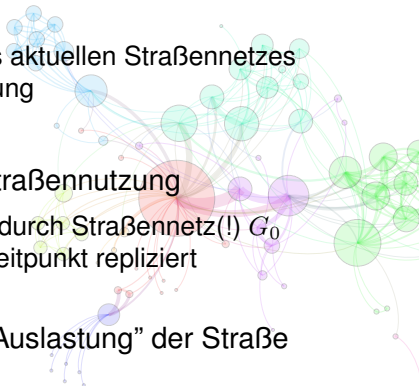
Stadt Dublin



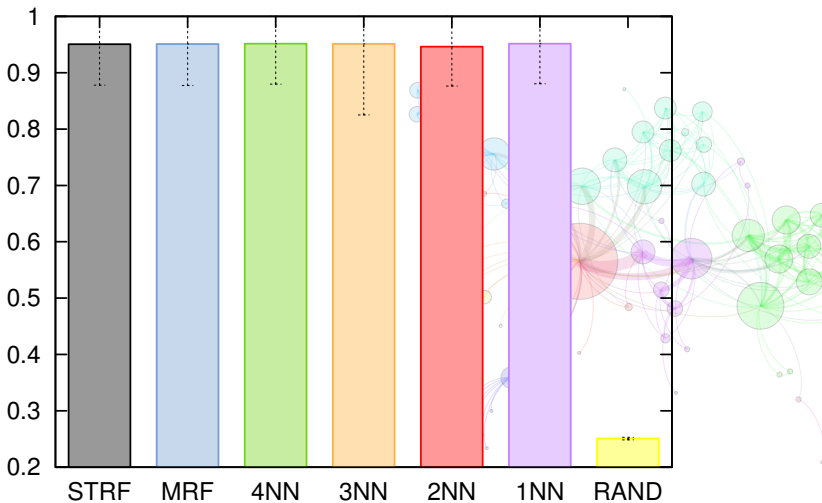


Straßennetze III

- **Daten:** Sensoren messen Verkehrsfluss und Verkehrsdichte
 - Verstehen und verbessern des aktuellen Straßennetzes
 - Unterstützung für Routenplanung
- **Idee:** Graphisches Modell der Straßennutzung
 - Graphische Struktur gegeben durch Straßennetz(!) G_0
 - Straßengraph wird für jeden Zeitpunkt repliziert
 $G = G_1 \circ G_2 \circ \dots \circ G_T$
- Zustandsraum gegeben durch “Auslastung” der Straße



Güte





Verbesserte Nutzung des Straßennetzes

- Durch Stau werden jedes Jahr **Millionen EURO** umgewandelt in **Tonnen** von **CO2**
 - Optimierung des Straßennetzes durch Minimierung der Stauwahrscheinlichkeit
 - Verhindern von Staus durch frühe Vorhersagen und automatische Umleitungen
- Verwendung von Stauwahrscheinlichkeiten in Navigationssystemen.

