Wissensentdeckung in Datenbanken

Merkmalsauswahl, Clustering

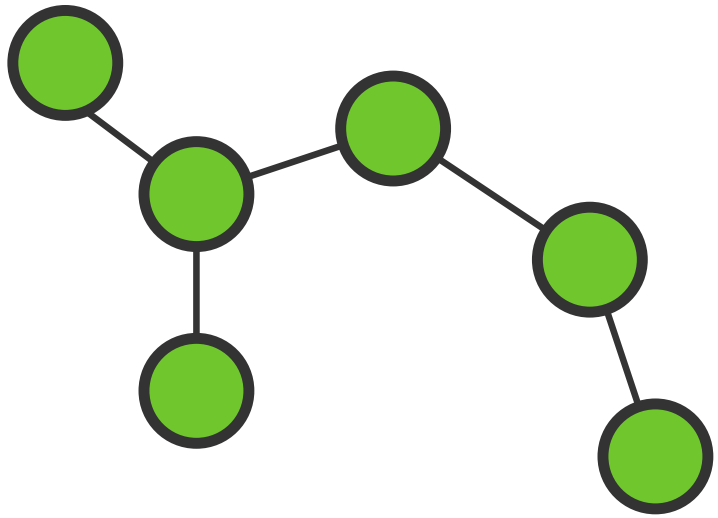
Nico Piatkowski und Uwe Ligges

Informatik—Künstliche Intelligenz
Computergestützte Statistik
Technische Universität Dortmund

06.07.2017

Überblick

- Strukturlernen—“Woher kommt der Graph?”
  - l_1-Regularisierung
- Merkmalsauswahl
  - Greedy-Selection
  - Regularisierung
- Clustering
  - Problemstellung
  - k-Means
  - DBSCAN (Dienstag)
  - LDA (Dienstag)



Bestimmung der bedingten Unabhängigkeitsstruktur = Graph—Warum?

- Der Graph G wird gebraucht, weil die suffiziente Statistik ϕ von G abhängt.
  - Bei diskreten Daten, bestimmen die **Kanten** von G und die **Zustandsräume $\mathcal{X}_v$ der Knoten**, welche Indikatorfunktionen in ϕ vorkommen. (27.06.)

- Data Mining / Wissens**entdeckung**:
  - **Gegeben**: Ein Datensatz D (Kunden, Produkte, Telefondaten, . . .)
  - **Gesucht**: “Zusammenhang” zwischen den Daten, welche Variablen werden durch welche anderen Variablen beeinflusst?
  - Falls Daten nicht diskret sind: diskretisieren, z.B. mittels Quantile oder Clustering (heute).

Bestimmung der bedingten Unabhängigkeitsstruktur = Graph—Warum?

- Der Graph G wird gebraucht, weil die suffiziente Statistik ϕ von G abhängt.
  - Bei diskreten Daten, bestimmen die **Kanten** von G und die **Zustandsräume $\mathcal{X}_v$ der Knoten**, welche Indikatorfunktionen in ϕ vorkommen. (27.06.)

- Data Mining / Wissens**entdeckung**:
  - **Gegeben**: Ein Datensatz D (Kunden, Produkte, Telefondaten, . . .)
  - **Gesucht**: “Zusammenhang” zwischen den Daten, welche Variablen werden durch welche anderen Variablen beeinflusst?
  - Falls Daten nicht diskret sind: diskretisieren, z.B. mittels Quantile oder Clustering (heute).

Bestimmung der bedingten Unabhängigkeitsstruktur = Graph—Warum?

- Der Graph G wird gebraucht, weil die suffiziente Statistik ϕ von G abhängt.
  - Bei diskreten Daten, bestimmen die **Kanten** von G und die **Zustandsräume $\mathcal{X}_v$ der Knoten**, welche Indikatorfunktionen in ϕ vorkommen. (27.06.)

- Data Mining / Wissens**entdeckung**:
  - **Gegeben**: Ein Datensatz D (Kunden, Produkte, Telefondaten, . . .)
  - **Gesucht**: “Zusammenhang” zwischen den Daten, welche Variablen werden durch welche anderen Variablen beeinflusst?
  - Falls Daten nicht diskret sind: diskretisieren, z.B. mittels Quantile oder Clustering (heute).

Bestimmung der bedingten Unabhängigkeitsstruktur = Graph—Warum?

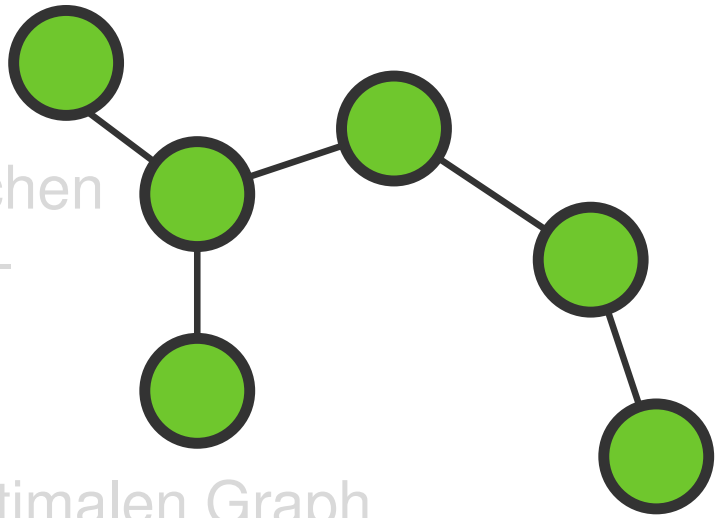
- Der Graph G wird gebraucht, weil die suffiziente Statistik ϕ von G abhängt.
  - Bei diskreten Daten, bestimmen die **Kanten** von G und die **Zustandsräume $\mathcal{X}_v$ der Knoten**, welche Indikatorfunktionen in ϕ vorkommen. (27.06.)

- Data Mining / Wissens**entdeckung**:
  - **Gegeben**: Ein Datensatz D (Kunden, Produkte, Telefondaten, . . .)
  - **Gesucht**: “Zusammenhang” zwischen den Daten, welche Variablen werden durch welche anderen Variablen beeinflusst?
  - Falls Daten nicht diskret sind: diskretisieren, z.B. mittels Quantile oder Clustering (heute).

Bestimmung des Graphen

Dienstag: Chow-Liu Algorithmus

- Berechnet optimale Baumstruktur
- Basiert auf Berechnung eines maximalen Spannbaums des vollständigen Graphen
- Kantengewichte für die Spann-baumberechnung sind die empirischen (aus den Daten bestimmte) Mutual-Informations zwischen den Knoten
- Gefundener Baum hat minimale Kullback-Leibler Divergenz zum optimalen Graph

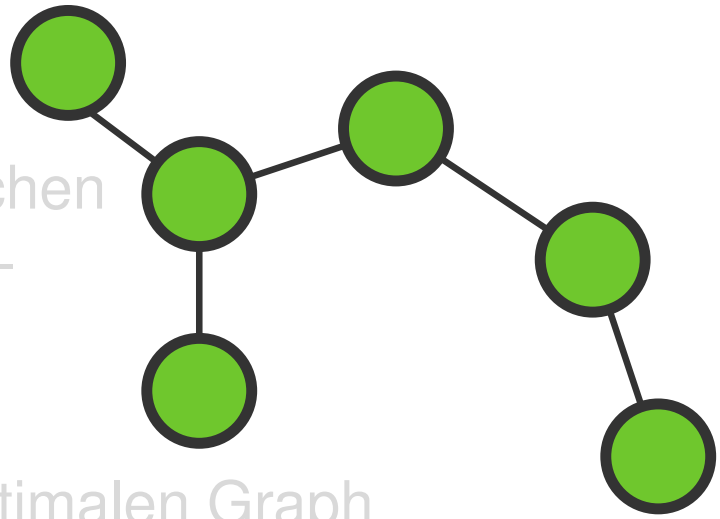


Jetzt: Bestimmung von G Regularisierung

Bestimmung des Graphen

Dienstag: Chow-Liu Algorithmus

- Berechnet optimale Baumstruktur
- Basiert auf Berechnung eines maximalen Spannbaums des vollständigen Graphen
- Kantengewichte für die Spannbaumberechnung sind die empirischen (aus den Daten bestimmte) Mutual-Informations zwischen den Knoten
- Gefundener Baum hat minimale Kullback-Leibler Divergenz zum optimalen Graph

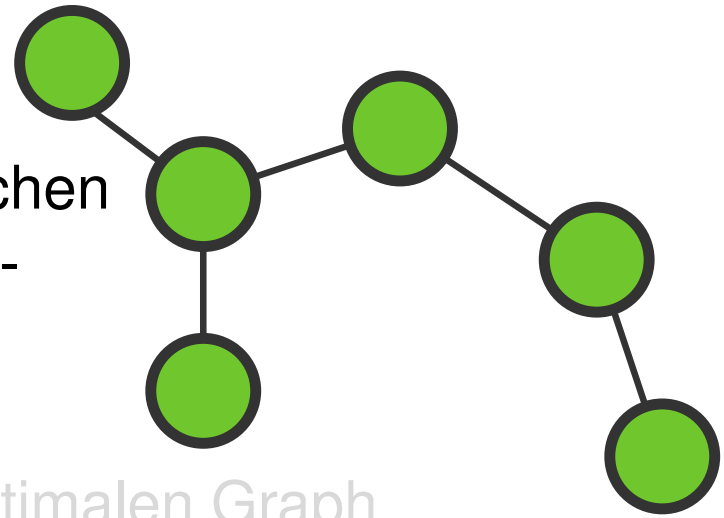


Jetzt: Bestimmung von G Regularisierung

Bestimmung des Graphen

Dienstag: Chow-Liu Algorithmus

- Berechnet optimale Baumstruktur
- Basiert auf Berechnung eines maximalen Spannbaums des vollständigen Graphen
- Kantengewichte für die Spannbaumberechnung sind die empirischen (aus den Daten bestimmte) Mutual-Informations zwischen den Knoten
- Gefundener Baum hat minimale Kullback-Leibler Divergenz zum optimalen Graph

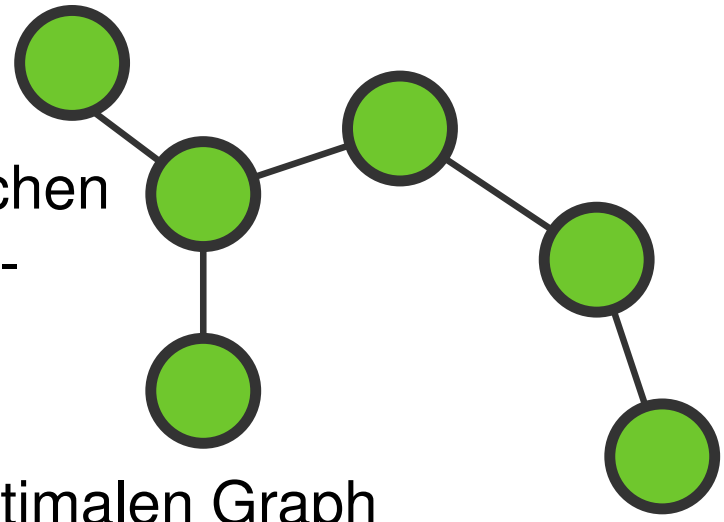


Jetzt: Bestimmung von G Regularisierung

Bestimmung des Graphen

Dienstag: Chow-Liu Algorithmus

- Berechnet optimale Baumstruktur
- Basiert auf Berechnung eines maximalen Spannbaums des vollständigen Graphen
- Kantengewichte für die Spann-baumberechnung sind die empirischen (aus den Daten bestimmte) Mutual-Informations zwischen den Knoten
- Gefundener Baum hat minimale Kullback-Leibler Divergenz zum optimalen Graph

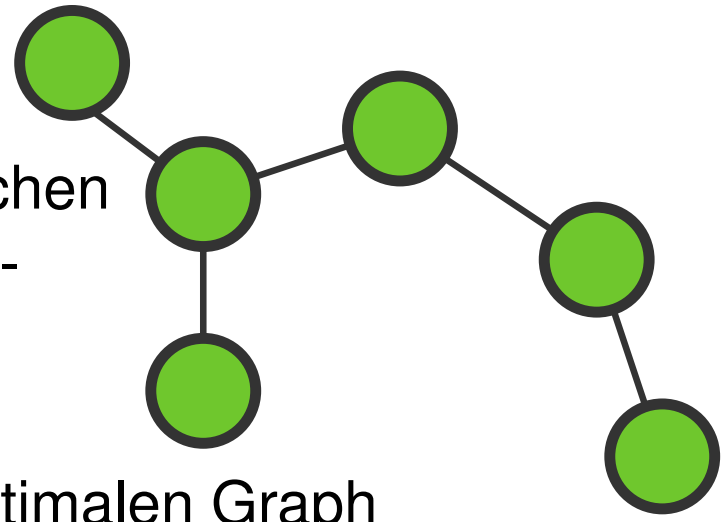


Jetzt: Bestimmung von G Regularisierung

Bestimmung des Graphen

Dienstag: Chow-Liu Algorithmus

- Berechnet optimale Baumstruktur
- Basiert auf Berechnung eines maximalen Spannbaums des vollständigen Graphen
- Kantengewichte für die Spann-baumberechnung sind die empirischen (aus den Daten bestimmte) Mutual-Informations zwischen den Knoten
- Gefundener Baum hat minimale Kullback-Leibler Divergenz zum optimalen Graph



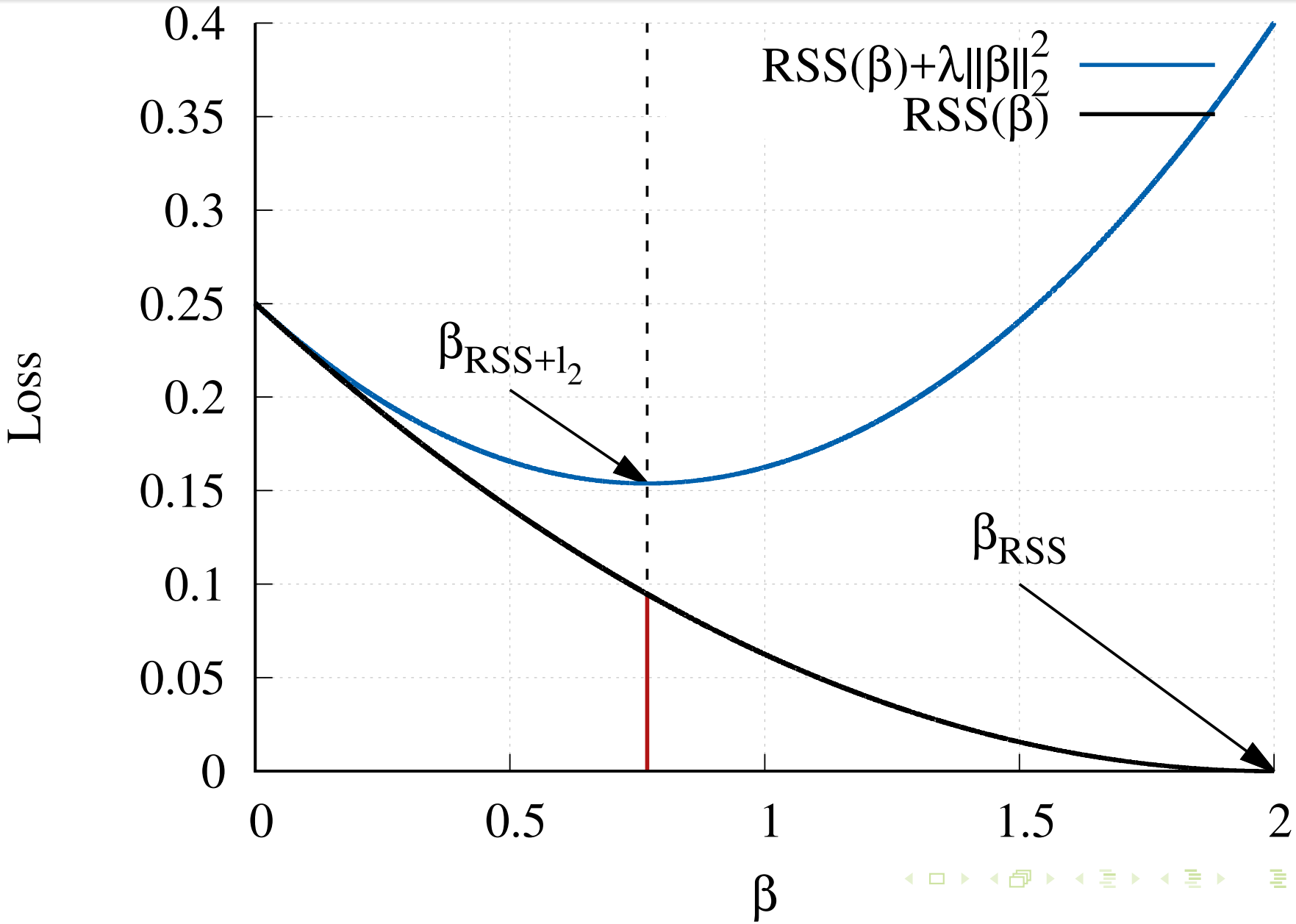
Jetzt: Bestimmung von G Regularisierung

Erinnerung: Regularisierung

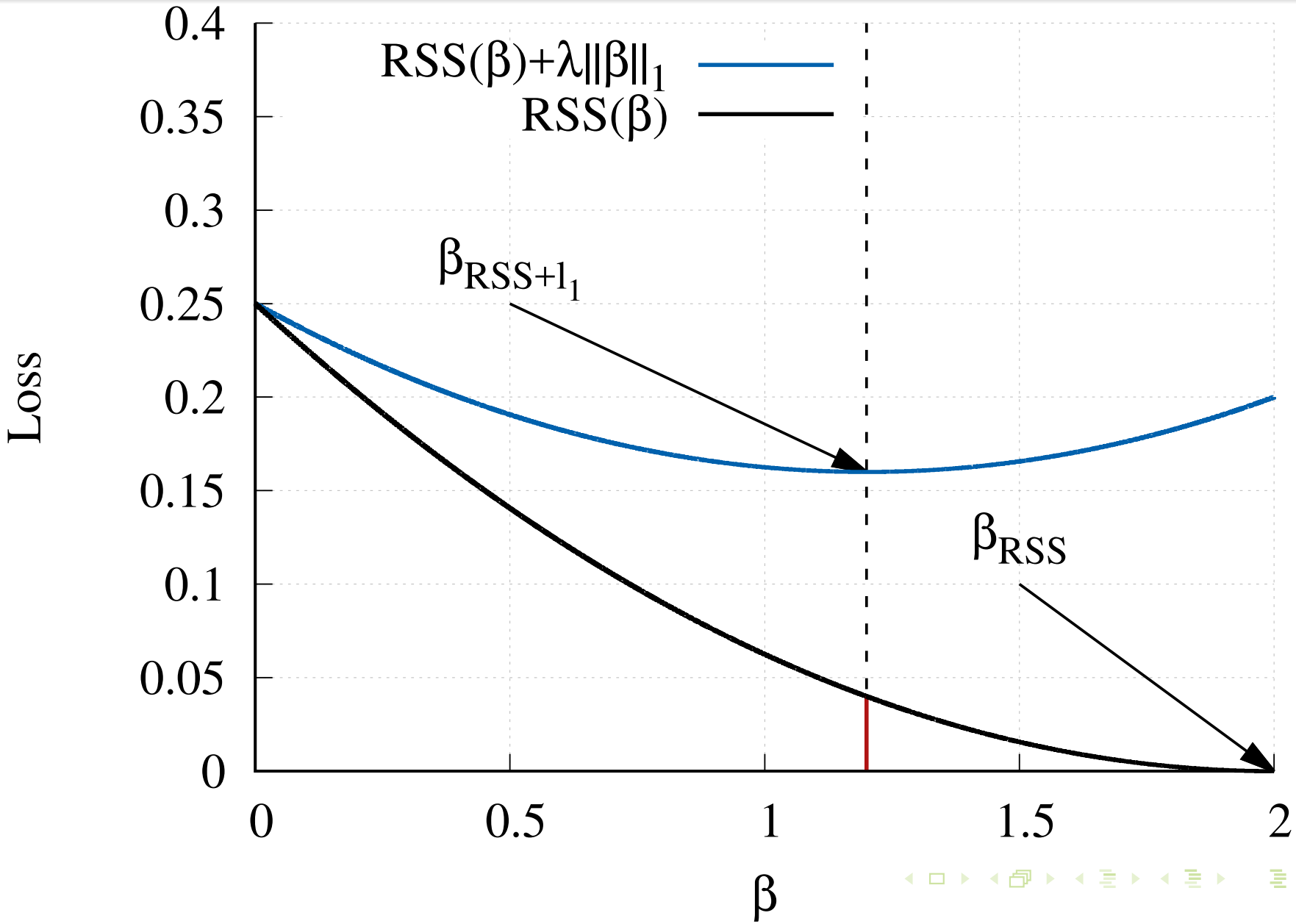
$$\min_{\boldsymbol{\beta}} \ell(\boldsymbol{\beta}; \mathcal{D}) + \lambda R(\boldsymbol{\beta})$$

- **Fakt**: Regularisierung verschiebt das Optimum (hier: Minimum) von ℓ zu einem Punkt mit "besseren" Eigenschaften
- Welche Eigenschaften das sind, hängt von der konkreten Wahl von $R : \mathbb{R}^d \to \mathbb{R}$ ab.

Erinnerung: l_2-Regularisierung



Erinnerung: l_1-Regularisierung



Graphen mittels Regularisierung

Baobachtung: Falls Parametervektor $\boldsymbol{\beta}_{\{v,u\}}$ einer Kante $= \mathbf{0}$, so hat diese Kante keinen Einfluss auf $\mathbb{P}(\boldsymbol{X} = \boldsymbol{x})$!

$$\mathbb{P}(\boldsymbol{X} = \boldsymbol{x}) = \frac{\prod_{\{v,u\} \in E} \exp(\left\langle \boldsymbol{\beta}_{\{v,u\}}, \phi_{\{v,u\}}(\boldsymbol{x}_{\{v,u\}}) \right\rangle)}{\sum_{\boldsymbol{x}' \in \mathcal{X}} \prod_{\{v,u\} \in E} \exp(\left\langle \boldsymbol{\beta}_{\{v,u\}}, \phi_{\{v,u\}}(\boldsymbol{x}'_{\{v,u\}}) \right\rangle)}$$

Falls $\boldsymbol{\beta}_{\{v,u\}} = \mathbf{0}$ für eine Kante $\{v, u\}$, so ist $\exp(\left\langle \boldsymbol{\beta}_{\{v,u\}}, \phi_{\{v,u\}}(\boldsymbol{x}_{\{v,u\}}) \right\rangle) = 1$.

Graphen mittels Regularisierung

Baobachtung: Falls Parametervektor $\boldsymbol{\beta}_{\{v,u\}}$ einer Kante $= \mathbf{0}$, so hat diese Kante keinen Einfluss auf $\mathbb{P}(\boldsymbol{X} = \boldsymbol{x})$!

$$\mathbb{P}(\boldsymbol{X} = \boldsymbol{x}) = \frac{\prod_{\{v,u\} \in E} \exp(\left\langle \boldsymbol{\beta}_{\{v,u\}}, \phi_{\{v,u\}}(\boldsymbol{x}_{\{v,u\}}) \right\rangle)}{\sum_{\boldsymbol{x}' \in \mathcal{X}} \prod_{\{v,u\} \in E} \exp(\left\langle \boldsymbol{\beta}_{\{v,u\}}, \phi_{\{v,u\}}(\boldsymbol{x}'_{\{v,u\}}) \right\rangle)}$$

Falls $\boldsymbol{\beta}_{\{v,u\}} = \mathbf{0}$ für eine Kante $\{v, u\}$, so ist $\exp(\left\langle \boldsymbol{\beta}_{\{v,u\}}, \phi_{\{v,u\}}(\boldsymbol{x}_{\{v,u\}}) \right\rangle) = 1$.

Graphen mittels Regularisierung (II)

$$\frac{\partial}{\partial \boldsymbol{\beta}_i} \|\boldsymbol{\beta}\|_2^2 = \frac{\partial}{\partial \boldsymbol{\beta}_i} \sum_{i=1}^{d} \boldsymbol{\beta}_i^2 = \frac{\partial}{\partial \boldsymbol{\beta}_i} \boldsymbol{\beta}_i^2 = 2\boldsymbol{\beta}_i$$

Falls $\beta_i > 0$:

$$\frac{\partial}{\partial \boldsymbol{\beta}_i} \|\boldsymbol{\beta}\|_1 = \frac{\partial}{\partial \boldsymbol{\beta}_i} \sum_{i=1}^{d} |\boldsymbol{\beta}_i| = \frac{\partial}{\partial \boldsymbol{\beta}_i} \sqrt{\boldsymbol{\beta}_i^2} = \frac{\boldsymbol{\beta}_i}{|\boldsymbol{\beta}_i|} = 1$$

Aber: $\|\cdot\|_1$ an 0 nicht differenzierbar(!) $\rightarrow$ Gradient kann nicht berechnet werden!!

Graphen mittels Regularisierung (II)

$$\frac{\partial}{\partial \boldsymbol{\beta}_i} \|\boldsymbol{\beta}\|_2^2 = \frac{\partial}{\partial \boldsymbol{\beta}_i} \sum_{i=1}^{d} \boldsymbol{\beta}_i^2 = \frac{\partial}{\partial \boldsymbol{\beta}_i} \boldsymbol{\beta}_i^2 = 2\boldsymbol{\beta}_i$$

Falls $\boldsymbol{\beta}_i > 0$:

$$\frac{\partial}{\partial \boldsymbol{\beta}_i} \|\boldsymbol{\beta}\|_1 = \frac{\partial}{\partial \boldsymbol{\beta}_i} \sum_{i=1}^{d} |\boldsymbol{\beta}_i| = \frac{\partial}{\partial \boldsymbol{\beta}_i} \sqrt{\boldsymbol{\beta}_i^2} = \frac{\boldsymbol{\beta}_i}{|\boldsymbol{\beta}_i|} = 1$$

Aber: $\|\cdot\|_1$ an 0 nicht differenzierbar(!) $\rightarrow$ Gradient kann nicht berechnet werden!!

Graphen mittels Regularisierung (II)

$$\frac{\partial}{\partial \boldsymbol{\beta}_i} \|\boldsymbol{\beta}\|_2^2 = \frac{\partial}{\partial \boldsymbol{\beta}_i} \sum_{i=1}^{d} \boldsymbol{\beta}_i^2 = \frac{\partial}{\partial \boldsymbol{\beta}_i} \boldsymbol{\beta}_i^2 = 2\boldsymbol{\beta}_i$$

Falls $\boldsymbol{\beta}_i > 0$:

$$\frac{\partial}{\partial \boldsymbol{\beta}_i} \|\boldsymbol{\beta}\|_1 = \frac{\partial}{\partial \boldsymbol{\beta}_i} \sum_{i=1}^{d} |\boldsymbol{\beta}_i| = \frac{\partial}{\partial \boldsymbol{\beta}_i} \sqrt{\boldsymbol{\beta}_i^2} = \frac{\boldsymbol{\beta}_i}{|\boldsymbol{\beta}_i|} = 1$$

Aber: $\|\cdot\|_1$ an 0 nicht differenzierbar(!) $\rightarrow$ Gradient kann nicht berechnet werden!!

Proximaler Gradientenabstieg

Falls Funktion f nicht differenzierbar: Für die Minimierung von

$$F(\boldsymbol{\beta}; \mathcal{D}) = \ell(\boldsymbol{\beta}^t; \mathcal{D}) + f(\boldsymbol{\beta})$$

nutzen wir anstatt

$$\boldsymbol{\beta}^{t+1} = \boldsymbol{\beta}^t - \eta_t \nabla F(\boldsymbol{\beta}^t; \mathcal{D})$$

jetzt

$$\boldsymbol{\beta}^{t+1} = \text{prox}_f(\boldsymbol{\beta}^t - \eta_t \nabla \ell(\boldsymbol{\beta}^t; \mathcal{D}))$$

Funktion F wird aufgeteilt in den differenzierbaren Teil ℓ und "Rest" f

Proximaler Gradientenabstieg

Falls Funktion f nicht differenzierbar: Für die Minimierung von

$$F(\boldsymbol{\beta}; \mathcal{D}) = \ell(\boldsymbol{\beta}^t; \mathcal{D}) + f(\boldsymbol{\beta})$$

nutzen wir anstatt

$$\boldsymbol{\beta}^{t+1} = \boldsymbol{\beta}^t - \eta_t \nabla F(\boldsymbol{\beta}^t; \mathcal{D})$$

jetzt

$$\boldsymbol{\beta}^{t+1} = \text{prox}_f(\boldsymbol{\beta}^t - \eta_t \nabla \ell(\boldsymbol{\beta}^t; \mathcal{D}))$$

Funktion F wird aufgeteilt in den differenzierbaren Teil ℓ und "Rest" f

Proximaler Gradientenabstieg für l_1-Regularisierung

Jetzt: $f(\cdot) = \lambda \| \cdot \|_1$

$$\text{prox}_{\lambda \| \cdot \|_1}(\gamma_i) = \min_a \left\{ \|a\|_1 + \frac{1}{2\lambda} \|a - \gamma_i\|_2^2 \right\}$$

Lösung:

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

Algorithmus zur Bestimmung von G:

(1) Wähle vollständigen Graphen G' (mit allen möglichen Kanten)

(2) Löse $\min_\beta \ell(\beta; \mathcal{D}) + \lambda \|\beta\|_1$ mittels proximalem Gradientenabstieg

(3) Entferne alle Kanten $\{v, u\}$ mit $\beta_{\{v,u\}} = 0$ aus G'

Proximaler Gradientenabstieg für l_1-Regularisierung

Jetzt: $f(\cdot) = \lambda \| \cdot \|_1$

$$\text{prox}_{\lambda \| \cdot \|_1}(\gamma_i) = \min_a \left\{ \|a\|_1 + \frac{1}{2\lambda} \|a - \gamma_i\|_2^2 \right\}$$

Lösung:

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

Algorithmus zur Bestimmung von G:

(1) Wähle vollständigen Graphen G' (mit allen möglichen Kanten)

(2) Löse $\min_\beta \ell(\beta; \mathcal{D}) + \lambda \|\beta\|_1$ mittels proximalem Gradientenabstieg

(3) Entferne alle Kanten $\{v, u\}$ mit $\beta_{\{v,u\}} = 0$ aus G'

Proximaler Gradientenabstieg für l_1-Regularisierung

Jetzt: $f(\cdot) = \lambda \| \cdot \|_1$

$$\text{prox}_{\lambda \| \cdot \|_1}(\gamma_i) = \min_a \left\{ \|a\|_1 + \frac{1}{2\lambda} \|a - \gamma_i\|_2^2 \right\}$$

Lösung:

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

Algorithmus zur Bestimmung von G:

(1) Wähle vollständigen Graphen G' (mit allen möglichen Kanten)

(2) Löse $\min_{\boldsymbol{\beta}} \ell(\boldsymbol{\beta}; \mathcal{D}) + \lambda \|\boldsymbol{\beta}\|_1$ mittels proximalem Gradientenabstieg

(3) Entferne alle Kanten $\{v, u\}$ mit $\boldsymbol{\beta}_{\{v,u\}} = \mathbf{0}$ aus G'

Merkmalsauswahl

Das war's mit graphischen Modellen, aber einige Ideen und Techniken werden wir wiedersehen..

Jetzt: Klassifikationsproblem auf Datensatz $\mathcal{D} = \{(\boldsymbol{x}, y)^1, (\boldsymbol{x}, y)^2, \ldots, (\boldsymbol{x}, y)^N\}$

- Angenommen $\boldsymbol{x}$ ist hochdimensional, z.B. $n = 1000000$
- Falls einige Dimensionen "unwichtig" sind (z.B. Rauschen, stark fehlerbehaftet, usw.), kann das die Klassifikation beeinträchtigen
- In diesem Fall kann es helfen die unwichtigen oder störenden Variablen zu entfernen(!)

Die Auswahl relevanter Variablen heißt Merkmalsselektion oder Merkmalsauswahl.

Merkmalsauswahl

Das war's mit graphischen Modellen, aber einige Ideen und Techniken werden wir wiedersehen..

Jetzt: Klassifikationsproblem auf Datensatz
$\mathcal{D} = \{(\boldsymbol{x}, y)^1, (\boldsymbol{x}, y)^2, \ldots, (\boldsymbol{x}, y)^N\}$

- Angenommen $\boldsymbol{x}$ ist hochdimensional, z.B. $n = 1000000$
- Falls einige Dimensionen "unwichtig" sind (z.B. Rauschen, stark fehlerbehaftet, usw.), kann das die Klassifikation beeinträchtigen
- In diesem Fall kann es helfen die unwichtigen oder störenden Variablen zu entfernen(!)

Die Auswahl relevanter Variablen heißt Merkmalsselektion oder Merkmalsauswahl.

Merkmalsauswahl

Das war's mit graphischen Modellen, aber einige Ideen und Techniken werden wir wiedersehen..

Jetzt: Klassifikationsproblem auf Datensatz
$\mathcal{D} = \{(\boldsymbol{x}, y)^1, (\boldsymbol{x}, y)^2, \ldots, (\boldsymbol{x}, y)^N\}$

- Angenommen $\boldsymbol{x}$ ist hochdimensional, z.B. $n = 1000000$
- Falls einige Dimensionen "unwichtig" sind (z.B. Rauschen, stark fehlerbehaftet, usw.), kann das die Klassifikation beeinträchtigen
- In diesem Fall kann es helfen die unwichtigen oder störenden Variablen zu entfernen(!)

Die Auswahl relevanter Variablen heißt Merkmalsselektion oder Merkmalsauswahl.

Merkmalsauswahl

Das war's mit graphischen Modellen, aber einige Ideen und Techniken werden wir wiedersehen..

Jetzt: Klassifikationsproblem auf Datensatz
$\mathcal{D} = \{(\boldsymbol{x}, y)^1, (\boldsymbol{x}, y)^2, \ldots, (\boldsymbol{x}, y)^N\}$

- Angenommen $\boldsymbol{x}$ ist hochdimensional, z.B. $n = 1000000$
- Falls einige Dimensionen "unwichtig" sind (z.B. Rauschen, stark fehlerbehaftet, usw.), kann das die Klassifikation beeinträchtigen
- In diesem Fall kann es helfen die unwichtigen oder störenden Variablen zu entfernen(!)

Die Auswahl relevanter Variablen heißt Merkmalsselektion oder Merkmalsauswahl.

Merkmalsauswahl (II)

Bei n Variablen gibt es insgesamt 2^n mögliche Merkmalsauswahlen $\rightarrow$ Für große n ist der Rechenbedarf zu hoch

Erneut haben wir (hier) zwei Optionen:

(A) Greedy-Algorithms

(B) Regularisierung ;-)

Merkmalsauswahl (II)

Bei n Variablen gibt es insgesamt 2^n mögliche Merkmalsauswahlen $\rightarrow$ Für große n ist der Rechenbedarf zu hoch

Erneut haben wir (hier) zwei Optionen:

(A) Greedy-Algorithms

(B) Regularisierung ;-)

Greedy-Algorithms: Forward-Selection

Eingabe: Klassifikations- oder Regressionsproblem auf Datensatz $\mathcal{D} = \{(\boldsymbol{x}, y)^1, (\boldsymbol{x}, y)^2, \ldots, (\boldsymbol{x}, y)^N\}$ mit n-dimensionalen Datenvektoren $\boldsymbol{x}^i$; beliebiges Modell

(1) Merkmalsmenge $M = \varnothing$, Güte $G = -\infty$

(2) Wähle neues Merkmal $1 \leq i \leq n$ und setze $M' = \{i\} \cup M$

(3) Lerne Modell mit Merkmalsmenge M' und berechne Güte G' (Kreuzvalidiert)

(4) Falls $G' > G$: $M = M'$ und $G = G'$

(5) Falls keines der Merkmale zu einer Verbesserung führt: Gib M aus und beende

(6) Ansonsten: Gehe zu (2)

Greedy-Algorithms: Forward-Selection

Eingabe: Klassifikations- oder Regressionsproblem auf Datensatz $\mathcal{D} = \{(\boldsymbol{x}, y)^1, (\boldsymbol{x}, y)^2, \ldots, (\boldsymbol{x}, y)^N\}$ mit n-dimensionalen Datenvektoren $\boldsymbol{x}^i$; beliebiges Modell

(1) Merkmalsmenge $M = \varnothing$, Güte $G = -\infty$

(2) Wähle neues Merkmal $1 \leq i \leq n$ und setze $M' = \{i\} \cup M$

(3) Lerne Modell mit Merkmalsmenge M' und berechne Güte G' (Kreuzvalidiert)

(4) Falls $G' > G$: $M = M'$ und $G = G'$

(5) Falls keines der Merkmale zu einer Verbesserung führt: Gib M aus und beende

(6) Ansonsten: Gehe zu (2)

Merkmalsauswahl mittels l_1-Regularisierung

Eingabe: Klassifikations- oder Regressionsproblem auf Datensatz $\mathcal{D} = \{(\boldsymbol{x}, y)^1, (\boldsymbol{x}, y)^2, \ldots, (\boldsymbol{x}, y)^N\}$ mit n-dimensionalen Datenvektoren $\boldsymbol{x}^i$; Lineares Modell mit Parametervektor $\boldsymbol{\beta} \in \mathbb{R}^n$; λ

(1) Lerne lineares Modell durch lösen von

$$\min_{\boldsymbol{\beta}} \sum_{i=1}^{d} (y - \langle \boldsymbol{\beta}, \boldsymbol{x} \rangle)^2 + \lambda \|\boldsymbol{\beta}\|_1$$

mit proximalem Gradientenabstieg

(2) Gib Merkmalsmenge $M = \{i \mid \beta_i \neq 0\}$ aus

Diese Prozedur heißt LASSO (Least Absolute Shrinkage And Selection Operator)

Merkmalsauswahl mittels l_1-Regularisierung

Eingabe: Klassifikations- oder Regressionsproblem auf Datensatz $\mathcal{D} = \{(\boldsymbol{x}, y)^1, (\boldsymbol{x}, y)^2, \ldots, (\boldsymbol{x}, y)^N\}$ mit n-dimensionalen Datenvektoren $\boldsymbol{x}^i$; Lineares Modell mit Parametervektor $\boldsymbol{\beta} \in \mathbb{R}^n$; λ

(1) Lerne lineares Modell durch lösen von

$$\min_{\boldsymbol{\beta}} \sum_{i=1}^{d} (y - \langle \boldsymbol{\beta}, \boldsymbol{x} \rangle)^2 + \lambda \|\boldsymbol{\beta}\|_1$$

mit proximalem Gradientenabstieg

(2) Gib Merkmalsmenge $M = \{i \mid \boldsymbol{\beta}_i \neq 0\}$ aus

Diese Prozedur heißt LASSO (Least Absolute Shrinkage And Selection Operator)

Merkmalsauswahl mittels l_1-Regularisierung

Eingabe: Klassifikations- oder Regressionsproblem auf Datensatz $\mathcal{D} = \{(\boldsymbol{x}, y)^1, (\boldsymbol{x}, y)^2, \ldots, (\boldsymbol{x}, y)^N\}$ mit n-dimensionalen Datenvektoren $\boldsymbol{x}^i$; Lineares Modell mit Parametervektor $\boldsymbol{\beta} \in \mathbb{R}^n$; λ

(1) Lerne lineares Modell durch lösen von

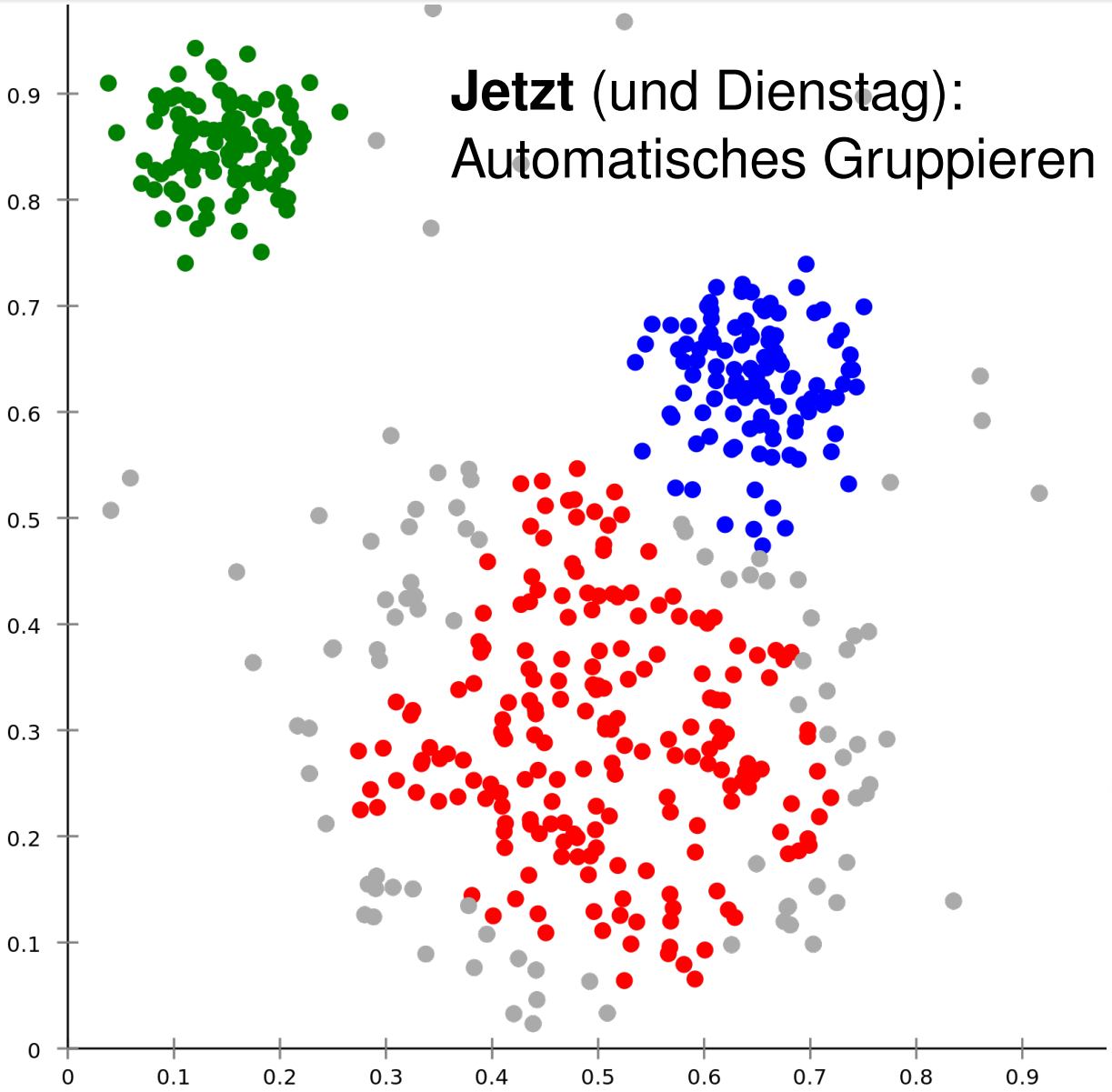
$$\min_{\boldsymbol{\beta}} \sum_{i=1}^{d} (y - \langle \boldsymbol{\beta}, \boldsymbol{x} \rangle)^2 + \lambda \|\boldsymbol{\beta}\|_1$$

mit proximalem Gradientenabstieg

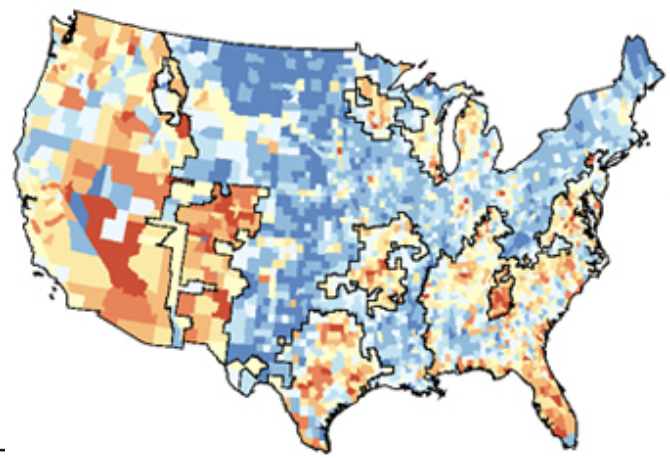
(2) Gib Merkmalsmenge $M = \{i \mid \boldsymbol{\beta}_i \neq 0\}$ aus

Diese Prozedur heißt LASSO (Least Absolute Shrinkage And Selection Operator)

Clusteranalyse



Jetzt (und Dienstag):
Automatisches Gruppieren von Daten



k-Means / Lloyd's Algorithmus

Eingabe: Daten $\mathcal{D}$, Anzahl Cluster k, Metrik/Distanzmaß $f : \mathcal{X} \times \mathcal{X} \to \mathbb{R}_+ \cup \{0\}$

(1) Weise jedem Punkt in D einen zufälligen Cluster zu

(2) Bestimme Clusterzentrum $\boldsymbol{c}$ ("Mittelpunkt") jedes Clusters

(3) Weise jedem Punkt $\boldsymbol{x}$ den Cluster zu, dessen Mittelpunkt $\boldsymbol{c}$ am nächsten zu $\boldsymbol{x}$ ist (mittels f)

(4) Wiederhole Schritte 2 und 3 so lange, bis sich die Clusterzuweisung nicht mehr ändert oder Zeit aufgebraucht

Man kann zeigen: Minimiert Distanzen innerhalb der Cluster und maximiert Distanz zwischen den Clustern

k-Means / Lloyd's Algorithmus

Eingabe: Daten $\mathcal{D}$, Anzahl Cluster k, Metrik/Distanzmaß $f : \mathcal{X} \times \mathcal{X} \to \mathbb{R}_+ \cup \{0\}$

(1) Weise jedem Punkt in D einen zufälligen Cluster zu

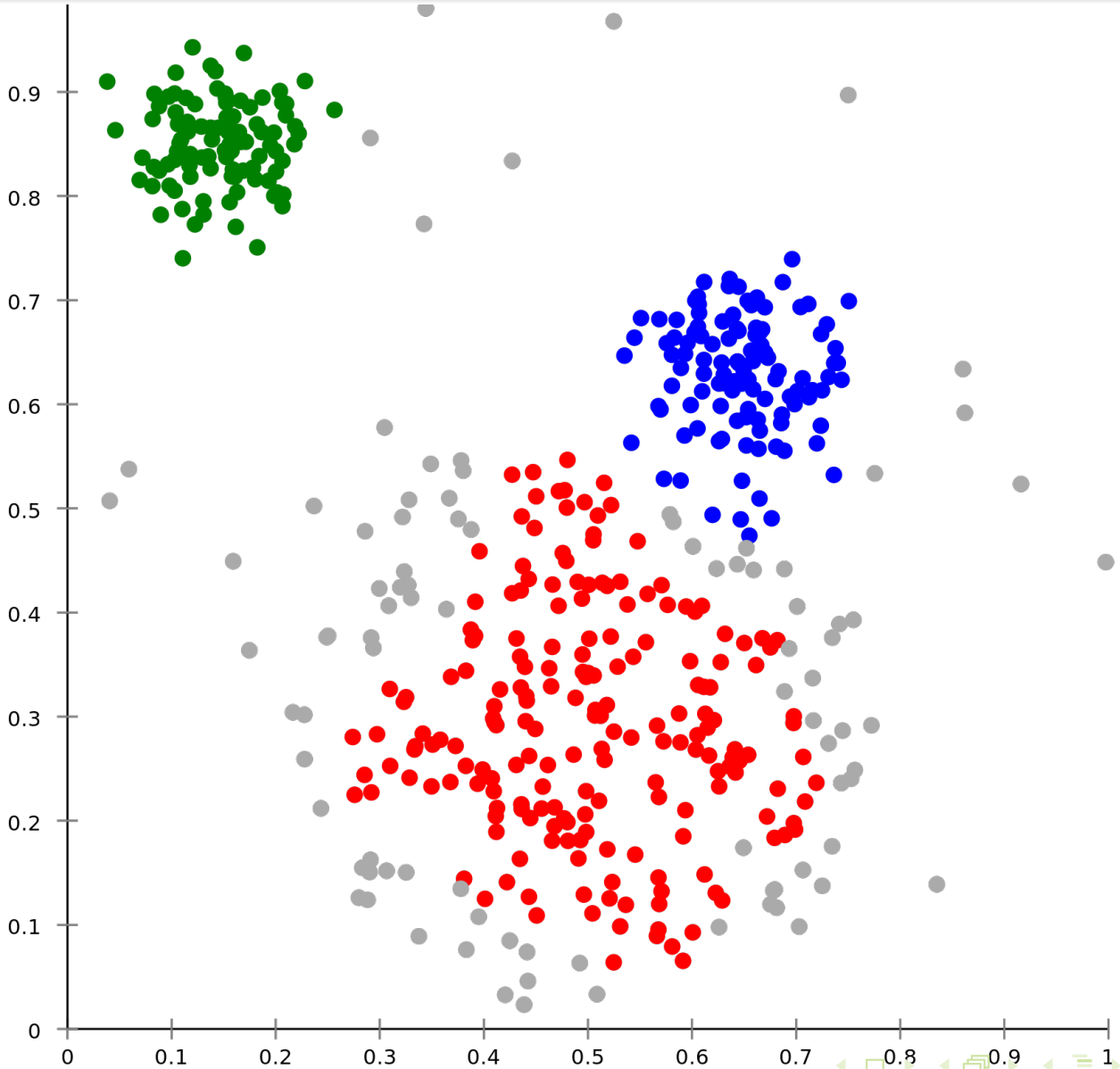
(2) Bestimme Clusterzentrum $\boldsymbol{c}$ ("Mittelpunkt") jedes Clusters

(3) Weise jedem Punkt $\boldsymbol{x}$ den Cluster zu, dessen Mittelpunkt $\boldsymbol{c}$ am nächsten zu $\boldsymbol{x}$ ist (mittels f)

(4) Wiederhole Schritte 2 und 3 so lange, bis sich die Clusterzuweisung nicht mehr ändert oder Zeit aufgebraucht

Man kann zeigen: Minimiert Distanzen innerhalb der Cluster und maximiert Distanz zwischen den Clustern

Intuition: Auswahl von k und Distanzmaß (I)



Intuition: Auswahl von k und Distanzmaß (II)