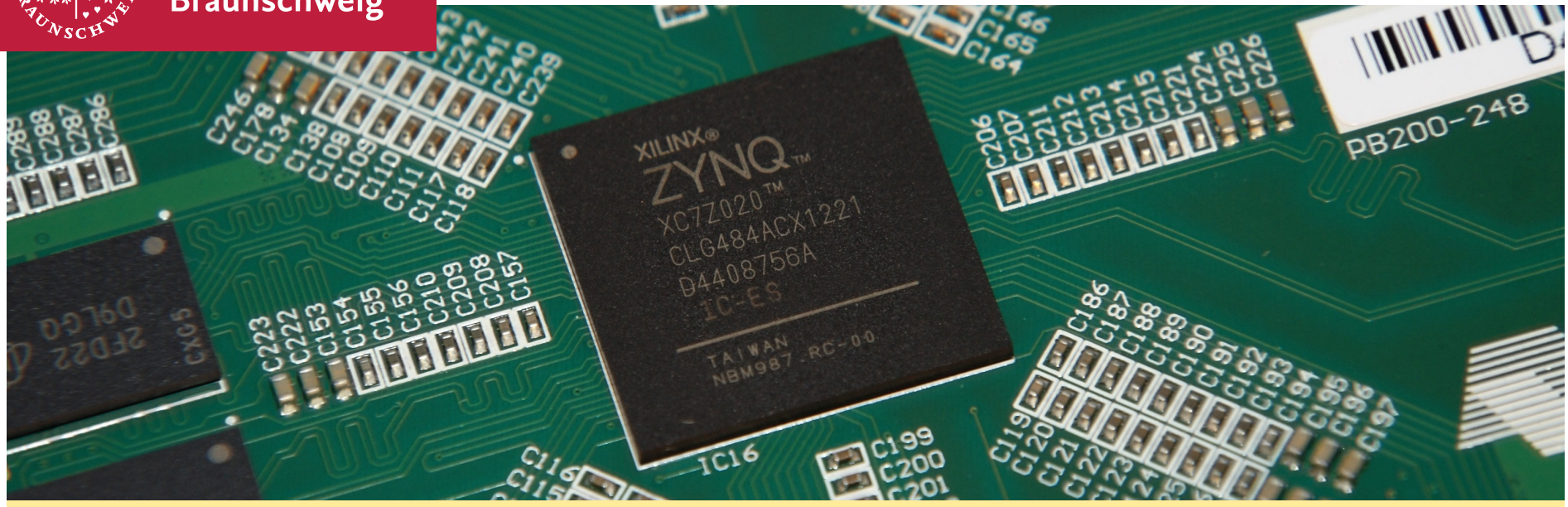
Technische Universität Braunschweig

Chair for Chip Design for Embedded Computing



Seminar Technische Informatik

Top 10 algorithms in data mining

Stephan Mielke, 22.01.2015

Motivation - Der Weltraum unendliche Weiten . . .

Abbildung 1: Hubble Ultra Deep Field [1]

Motivation - Einsatz von DM in der Astronomie

- Klassifizierung von Sternen mit *k*-nearest neighbor (*k*-nn)
- Manuelle Klassifizierung unmöglich [2]
- Pro Bild mehre 10000 Objekte
- Kepler z.B. hat 13.2m Objekte erkannt
- Benutzung von Klassifizierungsalgorithmen aus DM
- Je Objekt 9 Attribute (8 Isophotenformen, Leuchtkraft)
- Ausgabewert „stellary“
  - 0.0 – 0.1 Galaxie
  - 0.9 – 1.0 Stern

Name	Erkennung
Random Forest	82, 89%
Decision Tree	80, 68%
Artificial Neural Network	75.82%
Support Vector Machines	37, 82%

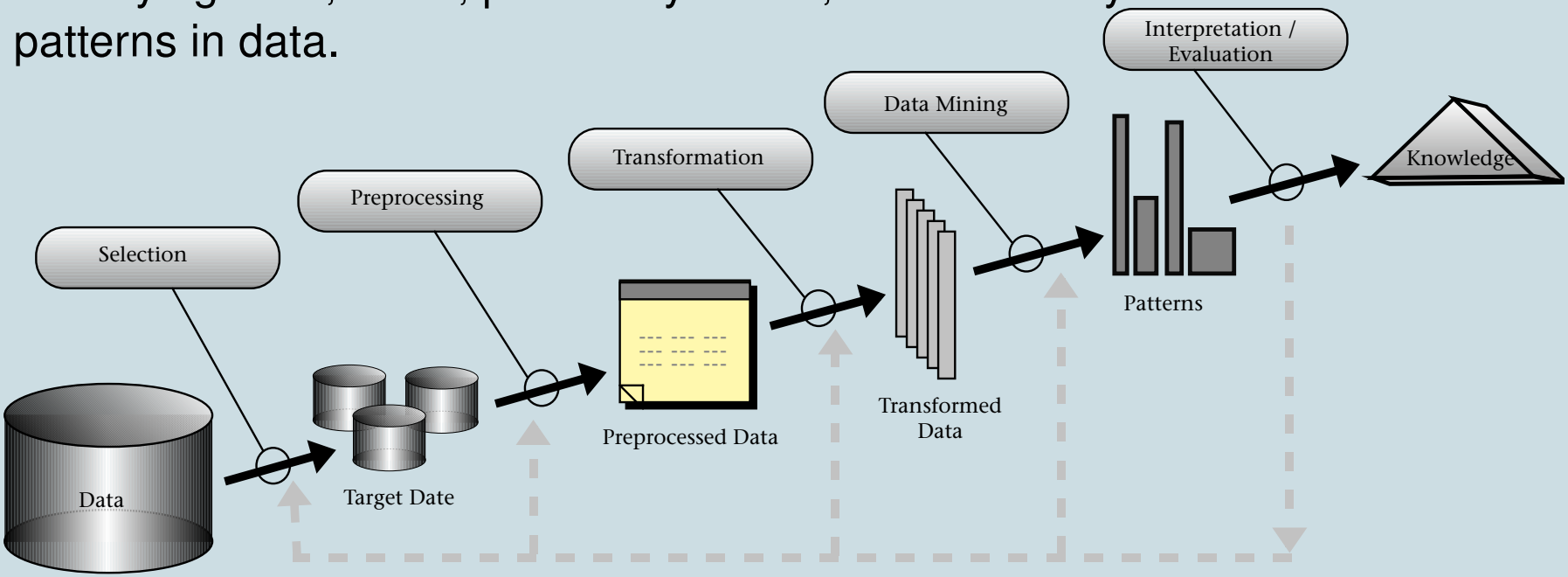
Tabelle 1: Erkennungsraten der Algorithmen Stern / Galaxie [3]

Data Mining - Einleitung [2]

- Idee: **Wissen** durch **Daten**
- Einsatz in der Forschung, Vermarktung, Medizin, (Wetter)-Vorhersagen, Betrugsaufklärung usw.

Definition nach Fayyad [4]

Knowledge Discovery in Databases describes the non-trivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data.



Data Mining - Top 10 algorithms in data mining [5]

- Anlass: IEEE International Conference on Data Mining
- Datum: Dezember 2006
- Erstellung: Jeder ACM KDD Innovation Award oder IEEE ICDM Research Contributions Award Preisträger nominierte 10 Algorithmen
- Nur Nominierte mit $\geqslant$ 50 Referenzierungen in *Google Scholar*
- `http://www.cs.uvm.edu/~icdm/algorithms/CandidateList.shtml`
- Per Abstimmung finden der Top 10
- Das Paper: Top 10 algorithms in data mining [5]

Data Mining - Top 10 algorithms in data mining [5]

Clustering

1. **C4.5 und ähnliche**
2. **k-means**
3. Suport Vector Machines
4. Apriori
5. **EM Algorithm**
6. PageRank
7. AdaBoost
8. k-nearest neighbor
9. Naive Bayes
10. **CART**

Data Mining - Top 10 algorithms in data mining [5]

Clustering

1. **C4.5 und ähnliche**
2. **k-means**
3. Suport Vector Machines
4. Apriori
5. **EM Algorithm**
6. PageRank
7. AdaBoost
8. k-nearest neighbor
9. Naive Bayes
10. **CART**

Classification

1. C4.5 und ähnliche
2. k-means
3. **Suport Vector Machines**
4. Apriori
5. EM Algorithm
6. PageRank
7. **AdaBoost**
8. **k-nearest neighbor**
9. **Naive Bayes**
10. CART

Data Mining - Top 10 algorithms in data mining [5]

Clustering

1. **C4.5 und ähnliche**
2. **k-means**
3. Suport Vector Machines
4. Apriori
5. **EM Algorithm**
6. PageRank
7. AdaBoost
8. k-nearest neighbor
9. Naive Bayes
10. **CART**

Classification

1. C4.5 und ähnliche
2. k-means
3. **Suport Vector Machines**
4. Apriori
5. EM Algorithm
6. PageRank
7. **AdaBoost**
8. **k-nearest neighbor**
9. **Naive Bayes**
10. CART

Assoziation

1. C4.5 und ähnliche
2. k-means
3. Suport Vector Machines
4. **Apriori**
5. EM Algorithm
6. PageRank
7. AdaBoost
8. k-nearest neighbor
9. Naive Bayes
10. CART

Data Mining - Clustering - Einleitung [2]

- Einordnung von Objekten in unbekannten Klassen
- Finden der Funktion die Objekte gruppiert
- Ähnlichkeit von Objekten durch eine Distanzfunktion ermitteln

Data Mining - Clustering - Cluster [6]

- Formen: sehr unterschiedlich
- Flach oder Hierarchisch
- Anzahl von Clustern:
  - Festgelegte Anzahl von *k*-Clustern
  - Anzahl hängt von der Qualitätsgüte der Cluster ab
- Qualitätsgüte: nicht zu klein oder groß
- Hard oder Soft - Clustering
- Keine großen „Lücken“ zwischen den Daten
- Cluster durch Heuristiken sonst zu großer Aufwand

Data Mining - Clustering - Distanzfunktion [2]

- Menge von Objekten $O = \{o_1, o_2, \ldots, o_n\}$
- Jedes Objekt x hat x_i Attribute
- Es muss gelten 1.-3., für Metrik 4.:

$$\text{dist}(o_1, o_2) = d \in R^{n \geqslant 0} \tag{1}$$

$$\text{dist}(o_1, o_2) = 0 \text{ genau dann wenn } o_1 = o_2 \tag{2}$$

$$\text{dist}(o_1, o_2) = \text{dist}(o_2, o_1) \text{ (Symmetrie)} \tag{3}$$

$$\text{dist}(o_1, o_3) \leqslant \text{dist}(o_1, o_2) + \text{dist}(o_2, o_3) \tag{4}$$

- Attributarten und jeweilige beispielhafte Distanzfunktionen:
  - Nummerisch $\text{dist}(x, y) = \sqrt{(x_1 - y_1)^2 + \ldots + (x_n - y_n)^2}$
  - Kategorisch $\text{dist}(x, y) = \sum_{i=1}^{n} \delta(x_i, y_i), \quad \delta(x_i, y_i) = \begin{cases} 0, x_i = y_i \\ 1, x_i \neq y_i \end{cases}$

Data Mining - Clustering - Beispiel [2]

- Clustering von Web-Sessions zur Bestimmung von Benutzergruppen
- Datenquelle: Logfile eines Webservers
- Eintrag: IP, User-ID, Timestamp, URL, . . .
- Einträge werden nach Session gruppiert, nach einem Zeitfenster
- Session: IP, User-ID, Liste von URLs
- URLs werden geclustert, z.B.: Distanzfunktion für endliche Mengen
- Wissen:
  - Benutzergruppen / Benutzerprofilen, für Marketingstrategien
  - URLs sind durch Interessen verbunden, Optimierung für Zugriffsgewohnheiten
- Ein Sozialmediabutton kann auch die nötigen Informationen liefern.

Data Mining - Clustering - k-means [6]

- Hartes Flaches Clustering
- Bekannte Anzahl von k Clustern
- Daten als Vektoren
- Idee: Minimiert den Abstand vom Clusterschwerpunkt zu den Daten
- Cluster ist Definiert als:
  - $A = \{d_1, \ldots, d_m\}$, A ist ein Cluster und d_i Element
  - $\mu(A) = \frac{1}{m} \sum_{i=1}^{m} d_i$ ist Schwerpunkt
- Qualität: gut wenn RSS$(\ldots)$ minimal ist

Cluster: $$\mathrm{RSS}(A) = \sum_{i=1}^{m} \left\| d_i - \mu(A) \right\|^2$$

Gesamt: $$\mathrm{RSS}(A_1, \ldots, A_k) = \sum_{j=1}^{k} \mathrm{RSS}(A_j)$$

Data Mining - Clustering - *k*-means [6]

Der *k*-means Algorithmus (Lloyd's Algorithmus)

1. Selektiere zufällig *k* Schwerpunkte als Startwert
2. Erstelle *k* leere Cluster
3. Weise jedem Cluser einen Schwerpunkt zu
4. Weise jedem Datenvektor den den Cluster mit dem nächstem Schwerpunkt zu
5. Berechne den Schwerpunkt jedes Clusters neu
6. Teste ob die Qualität des Clusterings ausreicht, sonst gehe zu 2.

Data Mining - Clustering - k-means

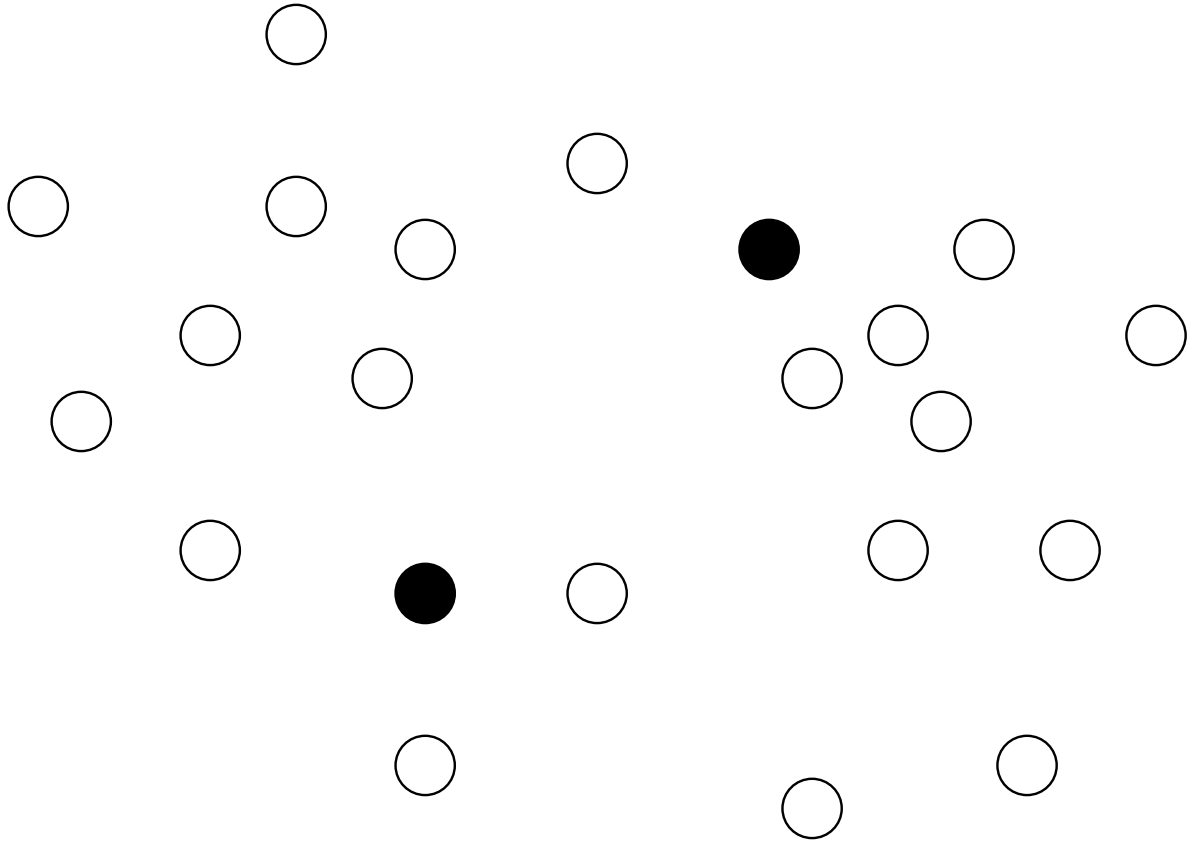


Abbildung 2: Ersten 3 Phasen, $k = 2$

Data Mining - Clustering - *k*-means

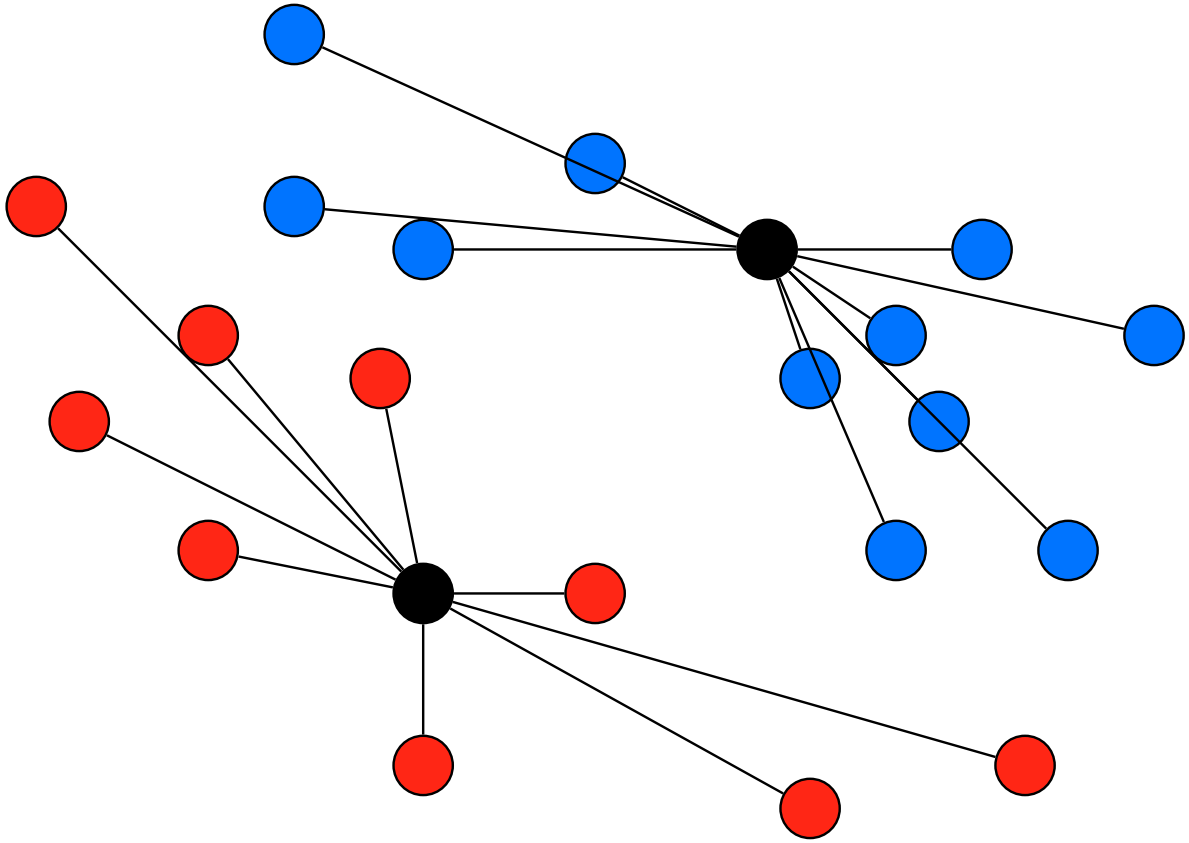


Abbildung 3: Phase 4, Zuordnung nur beispielhaft

Data Mining - Clustering - *k*-means

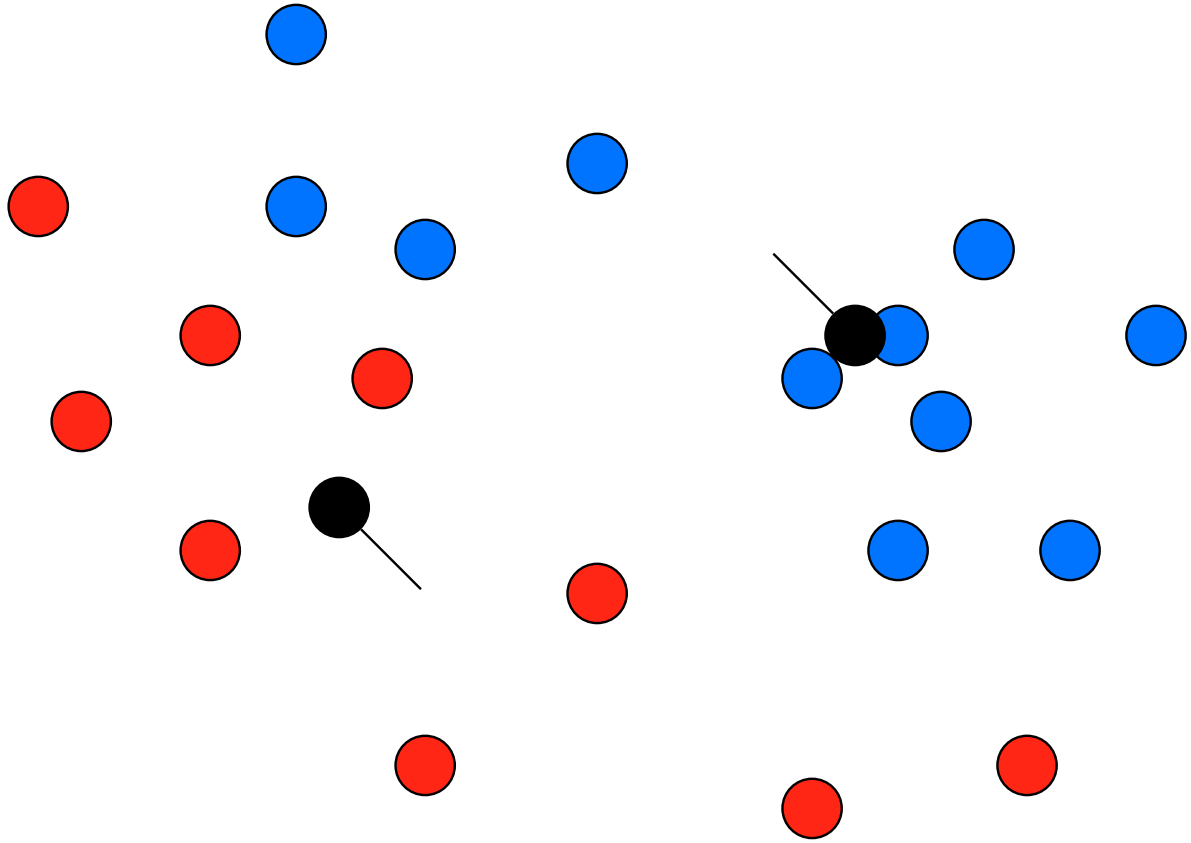


Abbildung 4: Phase 5, Schwerpunkte sind nur beispielhaft

Data Mining - Clustering - *k*-means

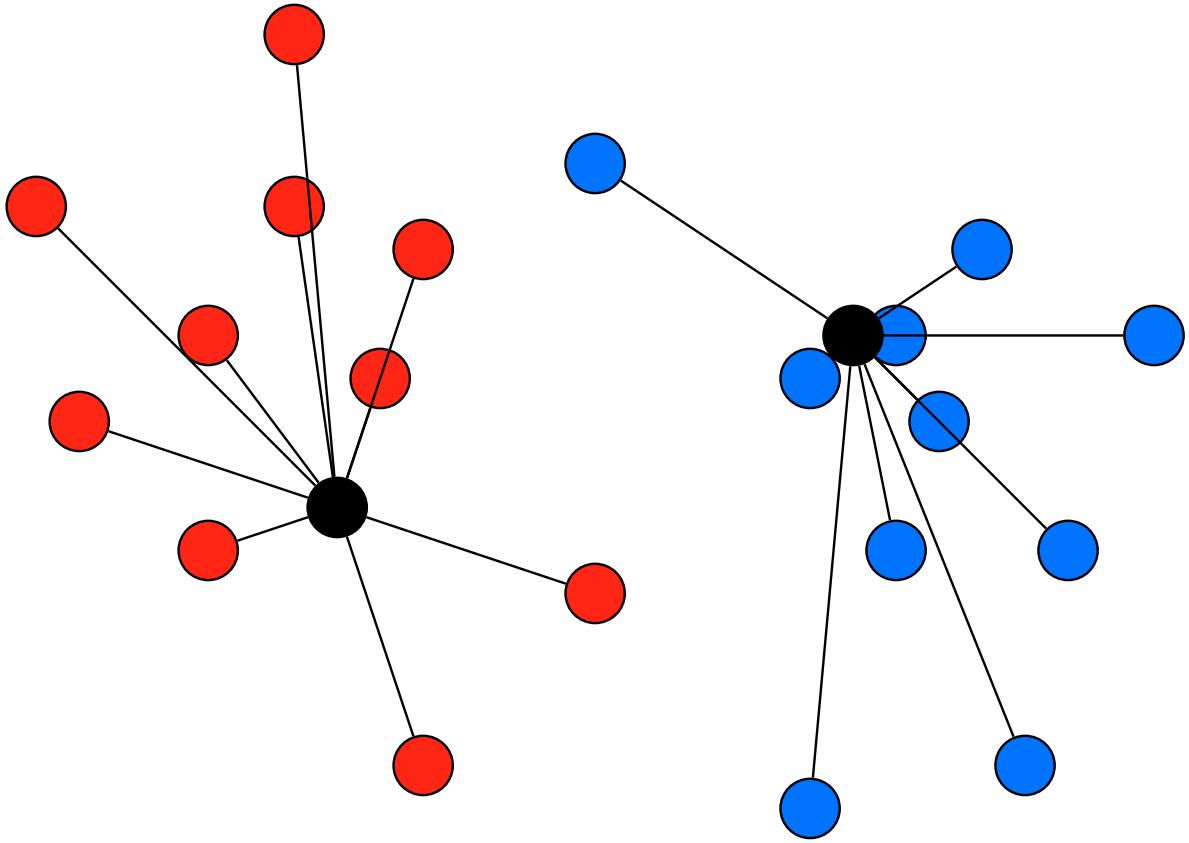


Abbildung 5: Phase 6 und noch mal von Phase 2 an

Data Mining - Classification - Einleitung [2]

- Einordnung von Objekten in bekannten Klassen
- Trainingsdaten für Klassen $\Rightarrow$ Klassen bekannt
- Finden der Funktion die Objekte möglichst genau zuordnet
- Teilaufgaben:
  - Zuordnung zu einer Klasse
  - Generierung von Wissen

Data Mining - Classification - Training [2]

- Menge von Objekten $O = \{o_1, o_2, \ldots, o_n\}$
- Klasse $c_i \in C = \{c_1, c_2, \ldots, c_n\}$ für jedes Objekt ist bekannt
- Jedes Objekt hat A_i Klassifizierung-Attribute
- Attributarten:
  - Kategorische Attribute
  - Nummerische Attribute

Data Mining - Classification - Beispiel [2]

Trainingsdaten:

ID	Alter	Autotyp	Risikoklasse
1	23	Familie	Hoch
2	17	Sport	Hoch
3	43	Sport	Hoch
4	68	Familie	Niedrig
5	32	LKW	Niedrig

Tabelle 2: Beispiele aus Knowledge discovery in databases: Techniken und Anwendungen [2]

Data Mining - Classification - Beispiel [2]

Trainingsdaten:

ID	Alter	Autotyp	Risikoklasse
1	23	Familie	Hoch
2	17	Sport	Hoch
3	43	Sport	Hoch
4	68	Familie	Niedrig
5	32	LKW	Niedrig

Tabelle 2: Beispiele aus Knowledge discovery in databases: Techniken und Anwendungen [2]

Das gesuchte Wissen

```
if Alter > 50                          then Risikoklasse = Niedrig
if Alter <= 50 and Autotyp = LKW then Risikoklasse = Niedrig
                                       else Risikoklasse = Hoch
```

Data Mining - Classification - Gesuchte Wissen [2]

Formen:
- Entscheidungsbaum
- Funktion
- Vektor im Koordinatensystem

Anwendung: Immer dann, wenn die Klassen bekannt sind
- Unterscheidung von Stern / Galaxie
- Sterne Einordnen
- Zuordnung von Risikogruppen
- Medizinforschung
- . . .

Data Mining - Classification - SVM [6]

Annahmen:
- Nur zwei Klassen
- Jedes Objekt ist ein Vektor im Koordinatensystem

Ziel: Hyperplane[1] die den Raum teilt

Training:
- Hyperplane mit maximalem Abstand zu allen Trainingsvektoren
- Hyperplane Begrenzungsobjekte sind Supportvektoren

Differenzfunktion: $\delta(o_1, o_2)$ ist ähnlich zum Clustering

[1] Hyperebene

Data Mining - Classification - SVM

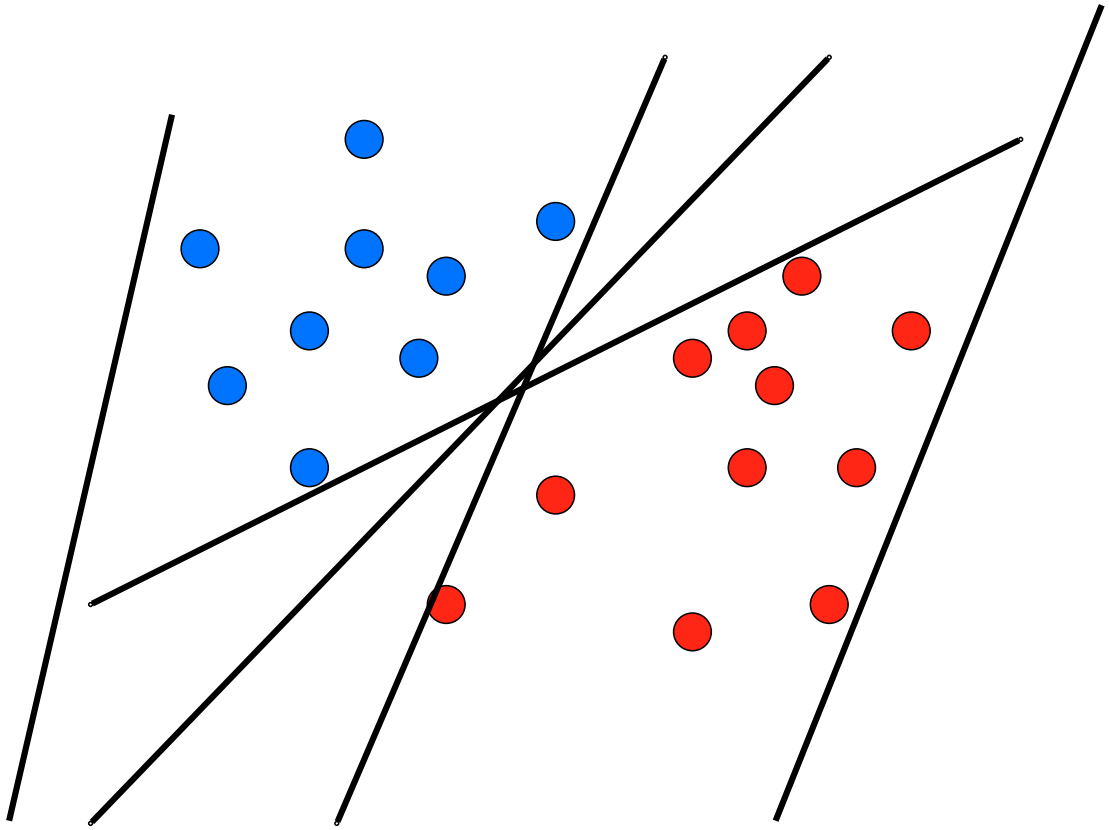


Abbildung 6: Gesucht: die richtige Hyperplane

Data Mining - Classification - SVM

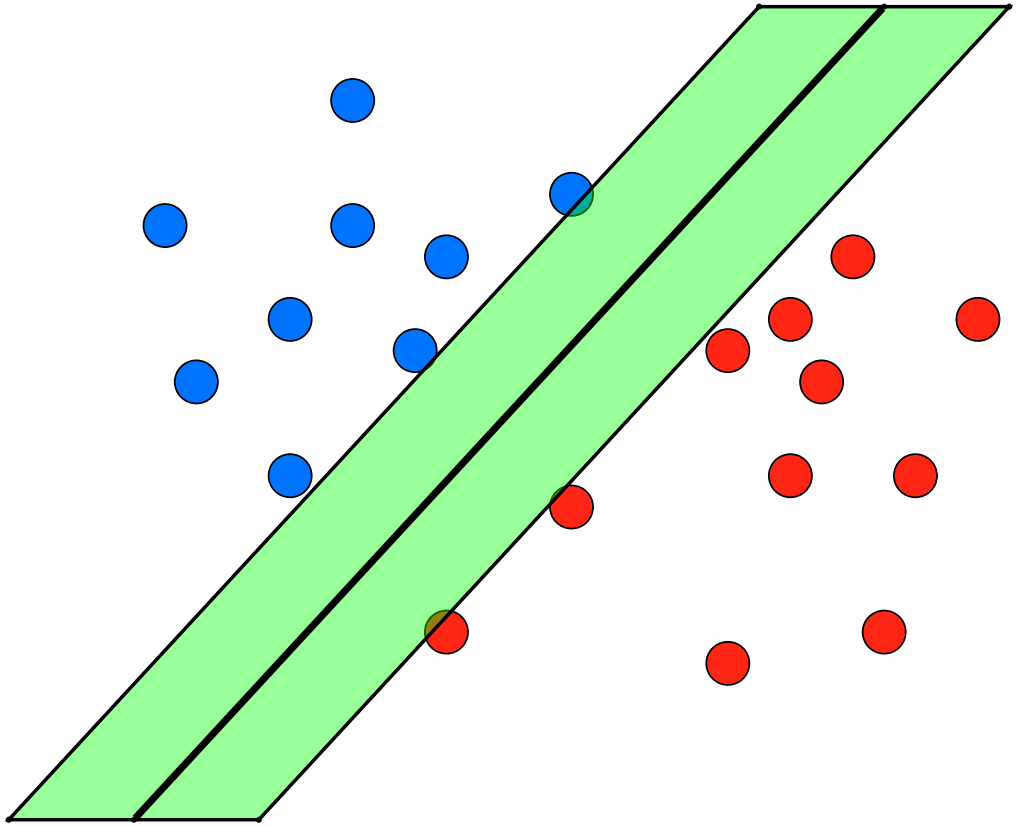


Abbildung 7: Gefunden: die richtige Hyperplane

Data Mining - Classification - SVM

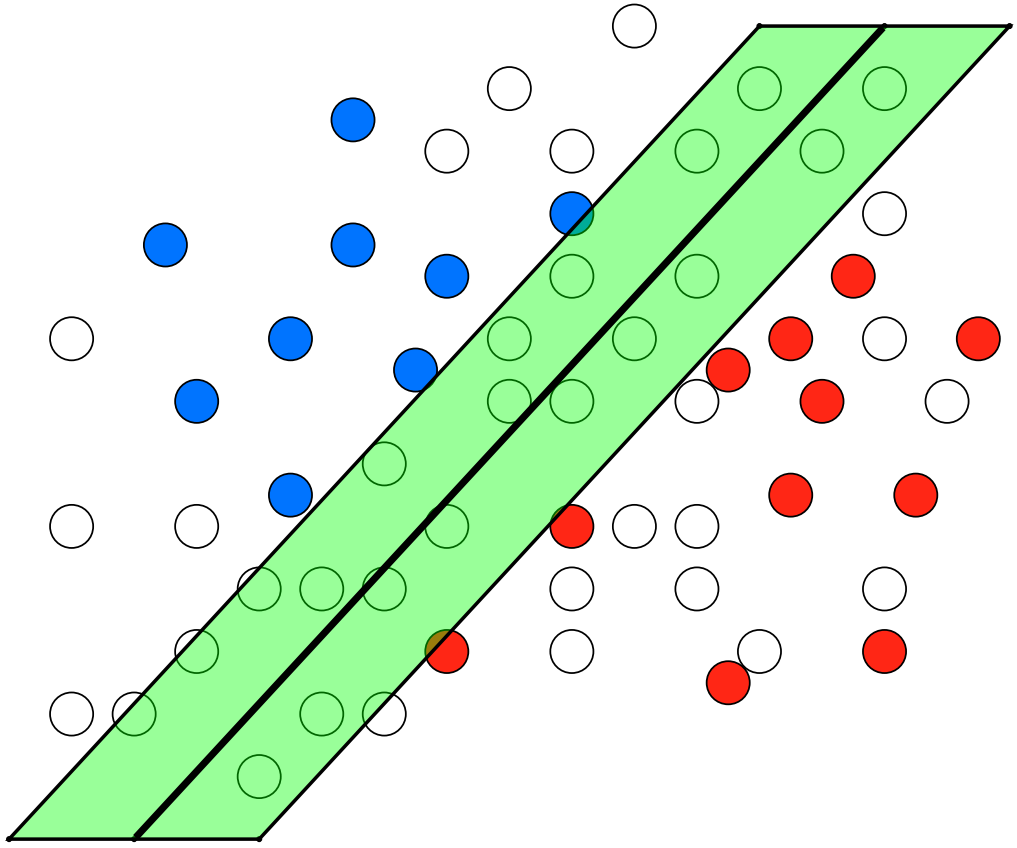


Abbildung 8: Einordnung: mit der richtige Hyperplane

Data Mining - Classification - SVM

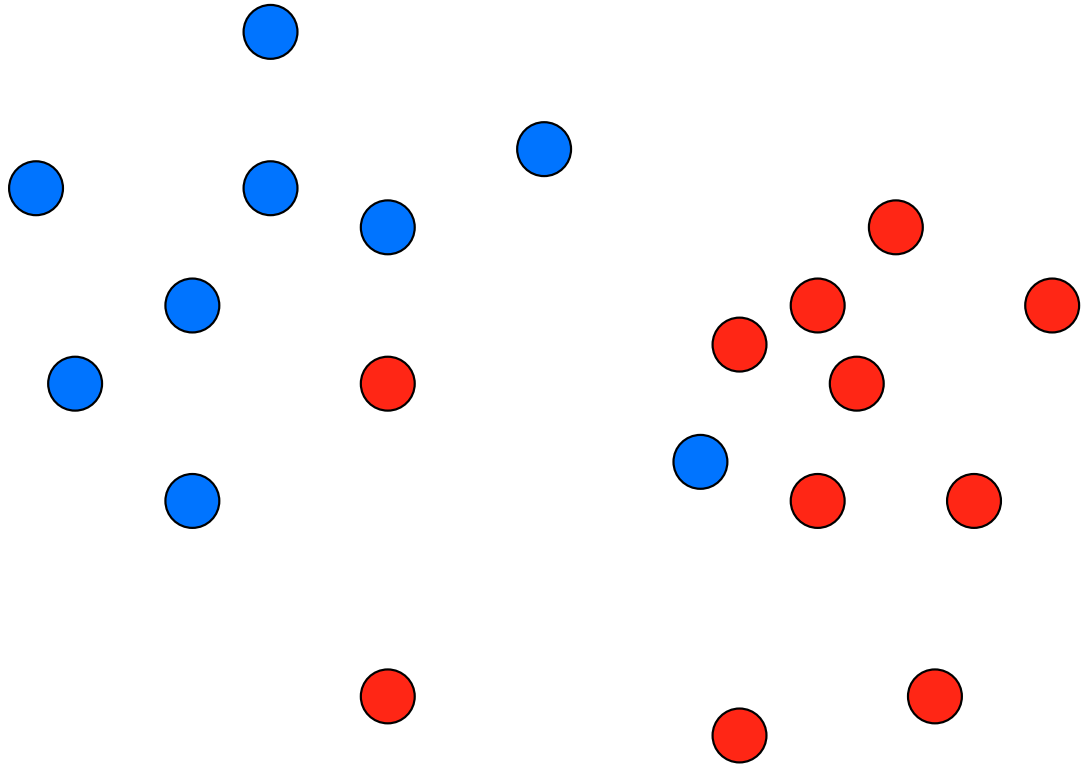


Abbildung 9: Training: ungünstige Daten

Data Mining - Assoziation - Einleitung [2]

Jedem bekannt?

Kunden, die diesen Artikel gekauft haben, kauften auch

Data Mining - Assoziation - Einleitung [2]

Jedem bekannt?

Kunden, die diesen Artikel gekauft haben, kauften auch

Gesucht: Beziehungen (Regeln: $A \Rightarrow B$) zwischen Objekten
Benötigt: Tranaktionsdatenbank (Einkaufshistory)

Data Mining - Assoziation - Einleitung [2]

Jedem bekannt?

Kunden, die diesen Artikel gekauft haben, kauften auch

Gesucht: Beziehungen (Regeln: $A \Rightarrow B$) zwischen Objekten
Benötigt: Tranaktionsdatenbank (Einkaufshistory)

T_i	Itemset (X_i)
1	Brot, Kaffee, Milch, Kuchen
2	Kaffee, Milch, Kuchen
3	Brot, Butter, Kaffee, Milch
4	Milch, Kuchen
5	Brot, Kuchen
6	Brot

Tabelle 3: Transaktionsdatenbank [2]

Support: {Kaffee, Milch}
$= 3/6 = 50\%$

Support: {Kaffee, Kuchen, Milch}
$= 2/3 \approx 33\%$

Support: {Kaffee, Milch} $\Rightarrow$ {Kuchen}
$= 2/3 \approx 33\%$

Konfidenz: {Kaffee, Milch} $\Rightarrow$ {Kuchen}
$= \frac{33\%}{50\%} \approx 66\%$

Data Mining - Assoziation - Grundbegriffe [2]

Items:	$I = \{i_1, \ldots, i_m\}$, ein Itemset $X \subseteq I$
Transaktionsset:	$D = \{T_1, \ldots, T_n\}$, für T_i gilt: $T_i \subseteq I$
Support der Menge:	$\delta(X, D)$: Anteil (%) aller T_i für die gilt $X \subseteq T_i$
Assoziationsregel:	$R_i = X \Rightarrow Y$ es gilt: $X, Y \subseteq I$ und $X \cap Y = \emptyset$
Support der Regel:	$\delta(R_i, D) = \delta(X \cup Y, D)$: Anteil (%)
Konfidenz der Regel:	$\phi(R_i, D) = \delta(Y, \{T_i \mid \forall T_i \in D \wedge X \subseteq T_i\})$
Idee:	Finden von Regeln die einen Support und Konfidenz von einer gewissen Schwelle besitzen

Big Data - Einleitung [8] [9]

- Himmelskartografie-Projekt Sloan Digital Sky Survey startete 2000[2]
- Sammelte in der ersten Wochen mehr Daten als die gesamte Astronomie davor

- Bis 2010 ca. 140 TB Daten gesammelt (ca. 35% Abdeckung) Sterne 260 562 744 und Galaxien 208 478 448 [7]
- 2019 geplanter Nachfolger Large Synoptic Survey Telescope[3]

$\Rightarrow$ **Erzeugt alle 5 Tage 140 TB an Daten!**

---

[2] Teleskop mit 2,5m Spiegel am Apache Point Observatory – New Mexico
[3] Teleskop mit 8,4m Spiegel am El-Peñón-Gipfel des Cerro Pachón – Chile

Big Data - Einleitung [8] [9]

- „Datenberge“ wachsen immer weiter an
  - Geschätzt 2007 an die 300 Exabyte[4] Daten
  - Geschätzt 2013 an die 1200 Exabyte Daten[5]
- Verarbeitung riesiger Datenmengen zur Gewinnung von Wahrscheinlichkeiten zu genaueren Vorhersagen
  - Das eine E-Mail Spam ist
  - Das „dei“ bei der Autokorrektur „die“ heißt
  - Bewegungen von Menschen, ob dies eine Gefahr für selbstlenkende Fahrzeuge sind
- Die Erkenntnis ist nicht das **WARUM** sondern das **WAS**
- „Was wir an Genauigkeit auf der Mikroebene verlieren, gewinnen wir an Erkenntnis auf der Makroebene.“ [8]

[4] 1 Exabyte = 1 000 000 TB

[5] in CDs: 5 Stapel zum Mond

Big Data - HACE Theorem [9]

*Big Data starts with large-volume, **H**eterogeneous, **A**utonomous sources with distributed and decentralized control, and seeks to explore **C**omplex and **E**volving relationships among data.*

Huge Heterogeneous Data:	Viele unterschiedliche Repräsentationen der „Datenhaufen“
Autonomous Sources:	Wahllose Generierung von Daten ohne zentrale Steuerung
Complex and Evolving Relationships:	Verflechtung der Daten untereinander wird immer komplexer und nimmt zu

Big Data - Herausforderungen für DM [9]

- Skalierung und Verarbeitung der Daten nach dem HACE Theorem
- Komplexität und Verarbeitungsdauer der Algorithmen

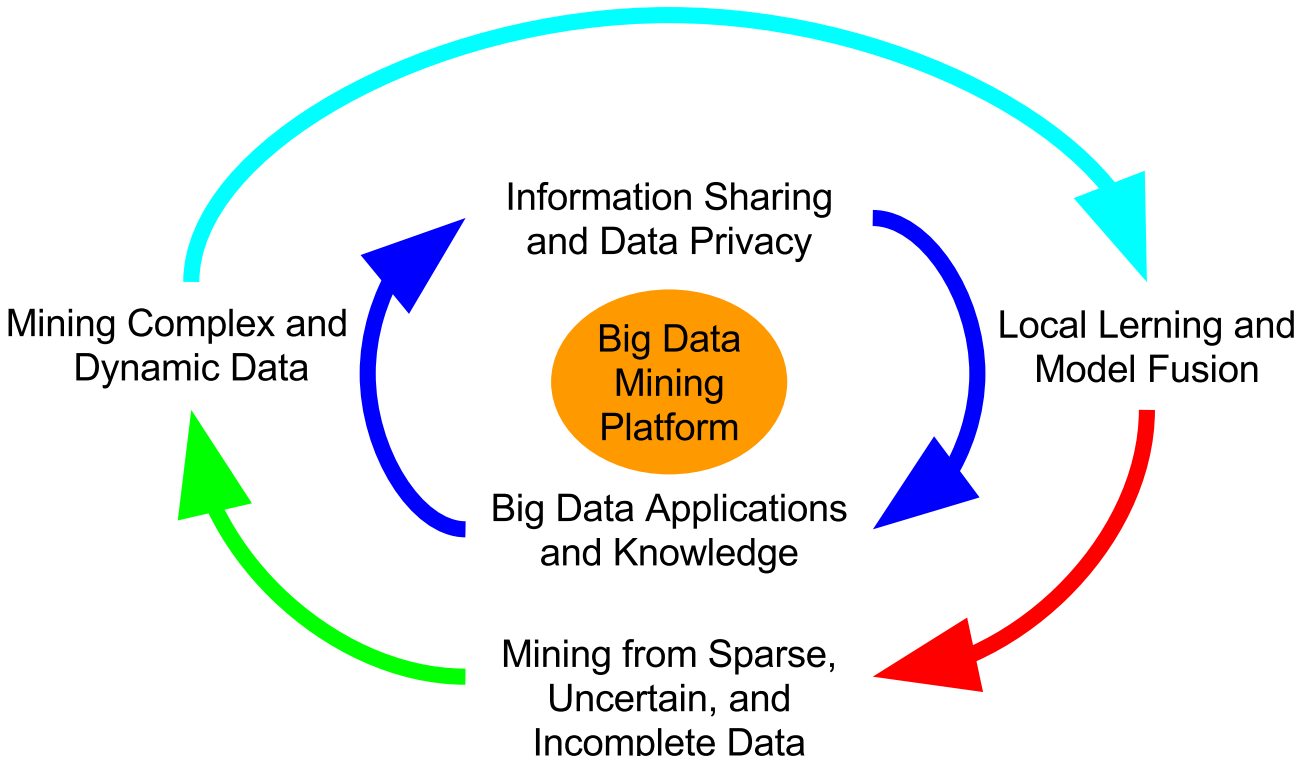


Abbildung 10: Datamining mit Big Data [9]

Fazit

Diskussion

Gibt es Fragen?

Danke

Vielen Dank für Ihre Aufmerksamkeit und Ihr Interesse.

Literatur I

[1] S. B. S. NASA, ESA and the HUDF Team. (2004) Hubble ultra deep field. [Online]. Available: http://imgsrc.hubblesite.org/hu/db/images/hs-2004-07-a-pdf.pdf

[2] M. Ester and J. Sander, *Knowledge discovery in databases: Techniken und Anwendungen*. Springer Heidelberg, 2000, vol. 2, no. 4.

[3] P. J. O'Keefe, M. G. Gowanlock, S. M. McConnell, and D. R. Patton, "Star-galaxy classification using data mining techniques with considerations for unbalanced datasets," in *Astronomical Data Analysis Software and Systems XVIII*, vol. 411, 2009, p. 318.

Literatur II

[4] U. Fayyad, G. Piatetsky-Shapiro, and P. Smyth, “From data mining to knowledge discovery in databases,” *AI magazine*, vol. 17, no. 3, p. 37, 1996.

[5] X. Wu, V. Kumar, J. R. Quinlan, J. Ghosh, Q. Yang, H. Motoda, G. J. McLachlan, A. Ng, B. Liu, S. Y. Philip *et al.*, “Top 10 algorithms in data mining,” *Knowledge and Information Systems*, vol. 14, no. 1, pp. 1–37, 2008.

[6] W.-T. Balke, “Data warehousing and data mining techniques,” University Lecture, 2014.

[7] SDSS-III. (2014, Nov.) The Scope of DR8. [Online]. Available: http://www.sdss3.org/dr8/scope.php

Literatur III

[8] V. Mayer-Schönberger and K. Cukier, *Big Data*. Computer Press, 2014.

[9] X. Wu, X. Zhu, G.-Q. Wu, and W. Ding, “Data mining with big data,” *Knowledge and Data Engineering, IEEE Transactions on*, vol. 26, no. 1, pp. 97–107, 2014.