Seminar Technische Informatik
Top 10 Algorithms in Data Mining

Stephan Mielke

Zusammenfassung—In diesem Paper werden die wichtigsten 10 Datamining Algorithmen aus dem Paper [1] von Xindong Wu vorgestellt sowie eingeordnet. Zu diesem Zweck wird auf die Kategorie der Cluster-, Classification- und Assoziation-Algorithmen eingegangen. Hierbei werden der k-means und der SVM näher vorgestellt. Anschließend wird auf Big Data sowie die Verbindung zwischen Datamining und Big Data eingegangen.

Index Terms—Data Mining, Big Data, Clustering, Classification, Assoziation, Support Vector Machines, k-means

Abbildung 1. Hubble Ultra Deep Field [2]

Tabelle 1
Erkennung [3]

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

1 Einleitung

Das Hubble Teleskop nahm vom 3. September bis zum 16. Januar 2004 das so genannte *Hubble Ultra Deep Field* Bild auf. Dieses Bild ist in Abbildung 1 zu sehen. Auf diesem Bild 10.000 kosmische Objekte zu erkennen. Problematisch ist jedoch, *WAS* ist ein solches Objekt? Ist es ein Stern, eine Galaxie, ein Quasar, eine Störung des Sensorchips usw.

Die Unterscheidung ob ein Objekt ein Stern, eine Galaxie oder etwas Unbekanntes ist, lässt sich mittels Data Mining Techniken Automatisieren. In dem Paper [3] von Peter J. O'Keefe und weiteren wird dieser Vorgehen beschrieben. Jedem sichtbarem Objekt werden 9 Attribute zugeordnet, die aus den i-Band Versionen der Aufnahmen extrahiert werden. Die Attribute bestehen aus der Leuchtkraft und 8 weiteren Lichteigenschaften und werden zu einer Zahl, genannt *stellary*, zwischen 0 und 1 ausgewertet. Das Intervall 0.0–0.1 repräsentiert eine Galaxie, 0.9–1.0 einen Stern und sonst ist das Objekt unbekannt aber könnte trotzdem ein Stern oder Galaxie sein. In Tabelle 1 sind die Erkennungswahrscheinlichkeiten einiger Data Mining Algorithmen für die Klassifizierung von stellaren Objekten aufgelistet.

Das Paper [1] von Xindong Wu und weiteren mit dem Titel *Top 10 Algorithms in Data Mining* wurde im Dezember 2006 für die *IEEE International Conference on Data Mining* erstellt, behandelt die zum damaligen Zeitpunkt wichtigsten Algorithmen fürs Data Mining und dient als Quelle für das gesamte Paper. Die Auswahl der einzelnen Algorithmen erfolge, indem jeder Preisträger eines *ACM KDD Innovation Awards* oder eines *IEEE ICDM Research Contributions Awards* jeweils 10 Algorithmen nominierte. Aus diesen Nominierten wurden nur die zur Abstimmung zugelassen, die mindestens 50 Referenzierungen in *Google Scholar* erreichen. Eine vollständige Liste der Kandidaten ist unter http://www.cs.uvm.edu/~icdm/

Tabelle 2
Top 10 Algorithmen [1]

Platz	Name	Art
1.	C4.5	Classification
2.	k-means	Clustering
3.	Suport Vector Machines	Classification
4.	Apriori	Association
5.	EM Algorithm	Classification
6.	PageRank	Link Mining
7.	AdaBoost	Classification
8.	k-nearest neighbor	Classification
9.	Naive Bayes	Classification
10.	CART	Classification

algorithms/CandidateList.shtml zu finden. In der Tabelle 2 werden die 10 besten Algorithmen genannt.

2 Data Mining

Der Begriff Data Mining wird im Deutschen für den gesamten Prozess des *Knowledge Discovery in Databases* (KDD) verwendet. Dies ist jedoch falsch, da Data Mining nur einen Teil des KDD einnimmt. Dies ist in Abbildung 2 zu sehen. Für KDD werden zuerst die Daten homogenisiert und dann mittels eines Data Mining Algorithmus verarbeitet, sodass „Wissen“ entsteht. Data Mining wird in der Forschung, Vermarktung, Medizin, (Wetter-) Vorhersagen, Betrugsaufklärung usw. eingesetzt. Die Idee von KKD ist *Wissen* durch *Daten* und wird nach Fayyad [4] wie folgt definiert:

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

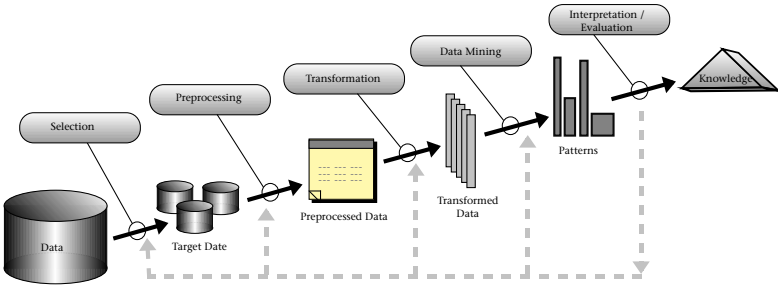

Abbildung 2. KDD nach Fayyad [4]

2.1 Clustering-Algorithmen

Die Kategorie der Clustering-Algorithmen beschreibt Algorithmen, die Daten unbekannten Klassen, den so genannten *Clustern* (Abschnitt 2.1.1), zu ordnen. Damit ist die Grundidee das Finden eines Algorithmus, der Objekte gruppiert. Diese Gruppierung wird mit Hilfe einer *Distanzfunktion* (Abschnitt 2.1.2) erreicht, die die Ähnlichkeit zwischen Objekten numerisch ermittelt. Alle Clustering-Algorithmen arbeiten mit Heuristiken, da das Clustering NP-Vollständig ist.

2.1.1 Cluster

Die Gestalt und Form der Cluster ist in den einzelnen Algorithmen sehr unterschiedlich. So gibt es Algorithmen bei denen die Cluster hierarchisch verschachtelt sind, das so genannte Flache- oder Hierarchische-Clustering. Es existiert ebenfalls das Hard- und Soft-Clustering, bei denen Objekte zu einem oder mehreren von einander unabhängigen Clustern zu geordnet sind. Bei allen Algorithmen ist die Anzahl der Cluster begrenzt durch direkte Festlegung der Anzahl oder durch eine Angegebene ausreichende Qualität der einzelnen Cluster. Die Cluster-Qualität ist ebenfalls nie genau definiert, jedoch kann man diese recht ungenau beschreiben durch, Anzahl der einzelnen Objekte die einem Cluster angehören, Größe des maximalen Unähnlichkeit eines Objektes zu seinem Cluster oder das vermeiden von *Lücken* im Cluster. In Abbildung 3(b) sind beispielhaft zwei Cluster (rot / blau) und die Clusterzentren (schwarz) des k-means Algorithmus (Abschnitt 2.1.3) gezeigt. Vergleiche das Buch [5] und die Vorlesung [6].

2.1.2 Distanzfunktion

Die Distanzfunktion bestimmt den Abstandsvektor zwischen zwei Objekten. Statt einer Distanzfunktion wird manchmal auch eine Ähnlichkeits- bzw. Simulierungsfunktion benutzt, jedoch sind in diesem Fall die Werte invertiert zu betrachten. Der Gebrauch von Distanzfunktionen hat sich jedoch durch gesetzt, da alle Rechnungen nummerisch stabiler sind, weil bei der Bedingung 2 mit dem $\vec{0}$ statt dem $\vec{\infty}$ gerechnet wird. Jedes Objekt

$o_i = (a_1, \ldots, a_n)$ besteht aus n Attributen, jedes Attribut ist ein nummerisches, kategorisches oder anders artiges Attribut und besitzt für sich selbst spezielle Distanzfunktionen. Die Bedingungen 1 – 3 müssen für jede Distanzfunktion gelten. Wenn die Bedingung 4 gilt, handelt es sich um eine Metrik.

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

$$\text{dist}(o_1, o_2) = \vec{0} \Leftrightarrow o_1 = o_2 \quad (2)$$

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

$$\text{dist}(o_1, o_3) \leq \text{dist}(o_1, o_2) + \text{dist}(o_2, o_3) \quad (4)$$

Für nummerische Attribute existieren die beispielhaften Distanzfunktionen (siehe Formeln 5 – 8) und für kategorische Attribute existiert die Summe der Unterschiede (siehe Formel 9). Allerdings existieren nicht nur kategorische und nummerische Attribute sondern ebenfalls textuelle Attribute usw. die dazu gehörigen Distanzfunktionen erfüllen mindestens genauso die Bedingungen 1 – 3.

Euklidische-Distanz:

$$\text{dist}(x, y) = \sqrt{(x_1 - y_1)^2 + \ldots + (x_n - y_n)^2} \quad (5)$$

Manhattan-Distanz:

$$\text{dist}(x, y) = |x_1 - y_1| + \ldots + |x_n - y_n| \quad (6)$$

Maximum-Metrik:

$$\text{dist}(x, y) = \max\left(|x_1 - y_1| + \ldots + |x_n - y_n|\right) \quad (7)$$

Alg. L_p-Metrik:

$$\text{dist}(x, y) = \sqrt[p]{\sum_{i=1}^{d} (x_i - y_i)^p} \quad (8)$$

Summe der Unterschiede:

$$\text{dist}(x, y) = \sum_{i=1}^{a} \delta(x_i, y_i) \quad (9)$$

$$\delta(x_i, y_i) = \begin{cases} 0 \text{ wenn } (x_i = y_i) \\ 1 \text{ wenn } (x_i \neq y_i) \end{cases} \quad (10)$$

2.1.3 k-means

Der k-means, oder auch Lloyd's Algorithmus genannt, gehört zu den Clustering-Algorithmen und basiert auf einem harten flachen Clustering. Die Objekte o_i werden als ein Vektor aus dem Vektorraum R^n interpretiert. Ein Cluster $A = \{o_1, \ldots o_i\}$ ist eine Menge von Objekten o_i und dessen Zentrum ist wie folgt definiert: $\mu(A) = \frac{l}{m} \sum_{i=l}^{m} o_i$. Ein Cluster besitzt eine hohe Güte, wenn $\text{RSS}(A) = \sum_{i=l}^{m} \left\| d_i - \mu(A) \right\|^2$ minimal ist. Das gesamte Clustering ist optimal, wenn $\text{RSS}(A_l, \ldots, A_k) = \sum_{j=l}^{k} \text{RSS}(A_j)$ minimal ist.

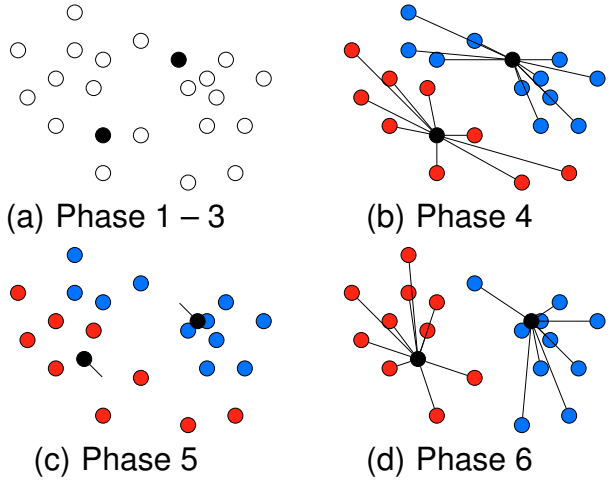

Abbildung 3. k-means

Der Algorithmus ist wie folgt:

1) Selektiere zufällig k Zentren als Startwert
2) Erstelle k leere Cluster
3) Weise jedem Cluster einen Zentren zu (siehe Abbildung 3(a))
4) Weise jedem Datenvektor den Cluster mit dem nächstem Zentrum zu (siehe Abbildung 3(b))
5) Berechne den Zentrum jedes Clusters neu (siehe Abbildung 3(c))
6) Teste, ob die Qualität des Clusterings ausreicht, sonst gehe zu 2. (siehe Abbildung 3(d))

2.2 Classification-Algorithmen

Bei der Kategorie der Classification-Algorithmen sind anders als bei den Clustering-Algorithmen die genauen Klassen in die eingruppiert wird bereits bekannt. Der einzige weitere große Unterschied zwischen beiden Kategorien ist, das bei den Classification-Algorithmen Trainingsdaten (siehe Abschnitt 2.2.1) verwendet werden. Die verwendeten Distanzfunktionen sind mit denen fürs Clustering (Abschnitt 2.1.2) vergleichbar.

2.2.1 Training

Tabelle 3
Beispiel Daten Versicherung

Alter	Autotyp	Risikoklasse
23	Familie	Hoch
17	Sport	Hoch
43	Sport	Hoch
68	Familie	Niedrig
32	LKW	Niedrig

TRAININGSDATEN sind eine Menge von Objekten $O = \{o_1, \ldots, o_n\}$ bei denen die Klassen $C = \{c_1, \ldots, c_m\}$ bereits bekannt sind oder manuell ermittelt wurden. Für die Objekte gelten die gleichen Eigenschaften wie beim Clustering (siehe Abschnitt 2.1.2). Um das Training zu verdeutlichen wird ein Beispiel aus dem Buch [5] verwendet. Dabei sind in Tabelle 3 Objekte mit ihren Attributen und der jeweils zugeordneten Klasse gezeigt. Aus diesen Trainingsdaten wird der, in Listing 1 als Bedingung formulierte, Entscheidungsbaum erzeugt. Mit Hilfe dieses Entscheidungsbaums können unklassifizierte Objekte klassifiziert werden.

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

Listing 1. Entscheidungsbaum

2.2.2 *Support Vector Machines*

BEI den Support Vector Machines (SVM) handelt es sich eigentlich um einen Algorithmus des *Statistical Learning* aber diese Algorithmen sind eine Unterkategorie der Classification. Der SVM kann eine Menge von Objekten nur in zwei disjunkte Teilmengen spalten. Zwischen den beiden Teilmengen der Trainingsdaten wird eine Hyperplane im n-dimensionalen Vektorraum erstellt. Der Abstand zwischen der Hyperebene und den Begrenzungsobjekten der Teilmengen, die so genannten Supportvektoren, ist maximal. Dieser Abstand wird mittels Differenzfunktionen wie beim Clustering (siehe Abschnitt 2.1.2) ermittelt.

IN der Abbildung 4(a) sind die Trainingsdaten den beiden Klassen (rot / blau) zugeordnet. Die beiden äußeren Hyperplanekadidaten sind ungültig, da diese nicht die Menge der

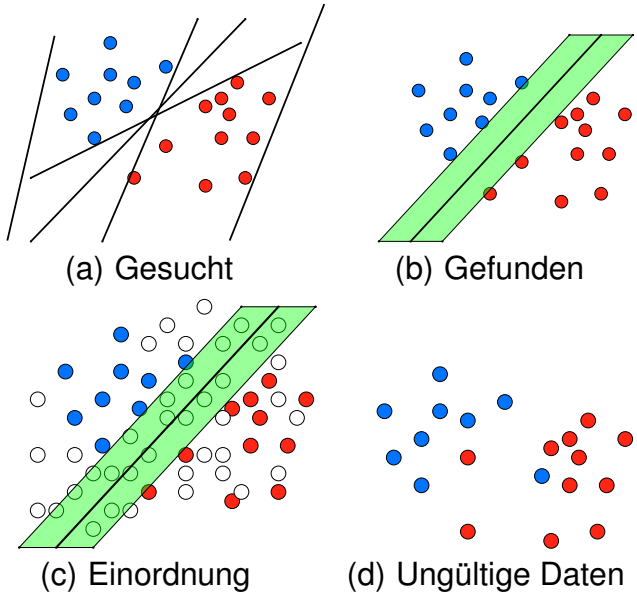

Abbildung 4. SVM

Objekten in zwei disjunkte Teilmengen schneidet. Von den inneren Hyperplanekadidaten fallen die beiden äußeren weg, da diese einen geringeren Abstand zu den Supportvektoren als die mittlere haben. Der mittlere Hyperplanekadidat ist die richtige Hyperplane und ist in Abbildung 4(b) zu sehen. Der grün markierte Bereich kennzeichnet den Bereich des maximalen Abstands zu den Supportvektoren. Nach dem Training werden die unklassifizierten Objekte eingefügt und werden je nach dem auf welcher Seite der Hyperplane sie sich befinden der passenden Klasse zugeordnet. Dieses ist in Abbildung 4(c) gezeigt. In Abbildung 4(d) sind ungültige Trainingsdaten dargestellt. Bei diesen Daten kann keine gültige Hyperplane gefunden werden. Somit müssen zwangsweise Objekte falsch klassifiziert werden.

2.3 Assoziation-Algorithmen

DIE meisten online Einkäufer ist der Satz „ Kunden, die diesen Artikel gekauft haben, kauften auch…“ wohl bekannt. Die angezeigten Ergebnis werden mit Hilfe von Data Mining Algorithmen der Assoziation-Kategorie ermittelt. Die Grundidee dieser Algorithmen ist es, Regeln der Form $A \Rightarrow B$ (siehe Definition 11) zu finden, die einen *Support* (siehe Definition 12) und eine *Konfidenz* (siehe Definition 14) besitzen, welche einen festgelegten Schwellenwert übersteigt. Um die Regeln zu

finden, benötigt man eine Transaktionsdatenbank, wie zum Beispiel eine Einkaufshistory. Die Transaktionsdatenbank $D = \{T_1, \ldots, T_n\}$ ist eine Menge von Transaktionen, wobei jede Transaktion $T_i \subseteq I$ eine Teilmenge aller Items (Waren) ist. Die Itemmenge $I = \{i_1, \ldots, i_m\}$ stellt die verkaufbaren Waren dar und ein Itemset $X \subseteq I$ ist eine Teilmenge aller Waren. Der Unterschied zwischen T_i und X ist, das T_i eine reale Transaktion und X nur ein Ausschnitt einer Transaktion ist.

Assoziationsregel:

$$R_i = X \Rightarrow Y \text{ es gilt: } X, Y \subseteq I \wedge X \cap Y = \emptyset \quad (11)$$

Support der Regel:

$$\delta(R_i, D) = \delta(X \cup Y, D) \quad (12)$$

Support der Menge:

$$\delta(X, D) = \text{Anteil (\%) aller } T_i \text{ für die gilt } X \subseteq T_i \quad (13)$$

Konfidenz der Regel:

$$\phi(R_i, D) = \delta(Y, \{T_i \mid \forall T_i \in D \wedge X \subseteq T_i\}) \quad (14)$$

Um den Sachverhalt zu verdeutlichen wird folgendes Beispiel aus dem Buch [5] genutzt. In Tabelle 4 ist eine Transaktionsdatenbank mit sechs Einträgen beschrieben. Von dieses Einträgen wird das Itemset $X = \{$Kaffee, Milch$\}$ betrachtet. Dieses ist in drei der sechs Einträge vorhanden und besitzt somit einen Support (siehe Formel 13) von 50% ($\delta(X, D) = 50\%$). Als nächstes wird ein Itemset gesucht, bei dem ein weiteres Item existiert und von dem X eine Teilmenge ist. Hierfür wird das Itemset $Z = \{$Kaffee, Milch, Kuchen$\}$ gewählt, mit dem die Regel $R = \{$Kaffee, Milch$\} \Rightarrow \{$Kuchen$\}$ gebildet wird. Die Regel R (siehe Formel 11) besitzt eine Konfidenz (siehe Formel 14) von 66% ($\phi(R, D) = 66\%$) und einen Support (siehe Formel 12) von 33% ($\delta(R, D) = 33\%$). Somit ist es zu 33% Wahrscheinlich, das ein Kunde zugleich alle drei Produkte kauft und zu 66% Wahrscheinlich, das ein Kunde, der Kaffee und Milch kauft, auch Kuchen kauft.

Tabelle 4
Transaktionsdatenbank

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

3 Big Data

Nach Schätzungen speicherte die gesamte Menschheit im Jahre 2007 ganze 300 Exabyte an digitalen Daten. Bis zum Jahr 2013 vervierfachte sich diese Datenmenge auf 1200 Exabyte, dies würde auf CDs gespeichert fünf Stapel von der Erde bis zum Mond bilden (siehe [7]). Diese Anzahl an Daten wird immer mehr, mit jedem Facebook Eintrag, jedem Tweet, jedem neuen Foto auf Instagram oder jeden neuen Nachricht per Whatsapp. Noch nie in der Geschichte der Menschheit, hat der Mensch so viele Daten produziert sowie aufgezeichnet wie heutzutage und damit sind nicht einmal die Sammelwut unserer Geheimdienste gemeint.

Big Data verschiebt die Schwierigkeit von der Datenbeschaffung zur Datenauswertung. Bis ins letzte Jahrzehnt war die Gewinnung der Daten, sei es in Forschung, Medizin, Marketing oder auch „Spionage“ usw., im Gegensatz zur Analyse und Auswertung das eigentliche Problem der Datenverarbeitung. Damit verschiebt sich die Erkenntnis durch Big Data vom *WARUM* ist etwas so wie es ist, zum *WAS* ist wie es ist. Der Grundgedanke von Big Data ist die Verarbeitung riesiger Datenmengen zur Gewinnung von Wahrscheinlichkeiten zu genaueren Vorhersagen von zum Beispiel: gehört eine E-Mail zum Spam, ist bei der Autokorrektur das eingegebene „dei“ doch besser ein „die“, welcher Spieler der Bundesliga passt am besten in das Team, wie ist die zukünftige Entwicklung an der Börse und so weiter. Für Big Data muss man sich jedoch immer wieder in den Kopf rufen, das alle Aussagen nur Wahrscheinlichkeiten sind und somit auch völlig falsch sein können. In dem Buch [7] ist es wie folgt beschrieben. „Was wir an Genauigkeit auf der Mikroebene verlieren, gewinnen wir an Erkenntnis auf der Makroebene.“

3.1 HACE Theorem

Das HACE Theorem wird im behandelt die dauerhafte Generierung unterschiedlichster und im Extremfall ungenauer bis falscher

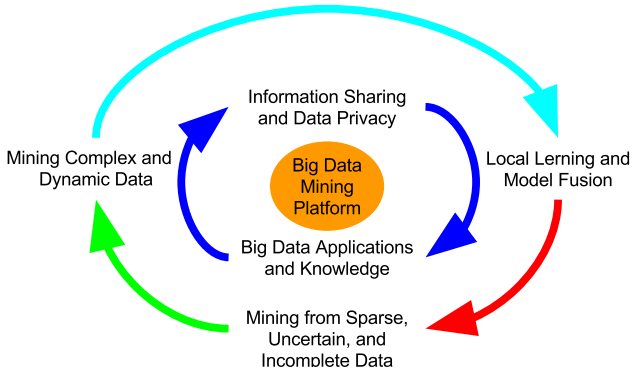

Abbildung 5. Big Data Herausforderungen [8]

Daten und deren immer komplexer sowie verzweigter werdenden Beziehungen. Im Paper von Wu Xindon [8] wird das Theorem wie folgt Definiert:

> 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.

3.2 Herausforderungen für Data Mining

DURCH das HACE Theorem entstehen für das Data Mining viele Schwierigkeiten. In der Abbildung 5 sind die drei Ebenen des Data Mining für Big Data und deren Aufgaben, von innen nach außen, beschrieben. In der ersten Ebene, existieren die Schwierigkeiten des Datenzugriffs und der Datenanalyse, Da sich die Datenquellen räumlich getrennt von einander befinden, müssen diese über das Internet an den Verarbeitungsknoten transportiert werden. Des Weiteren sind die meisten Data Mining Algorithem für kleine Datenmengen ausgelegt, weil alle Daten lokal im RAM gespeichert sein müssen. Auf der zweiten Ebene sind ethische und moralische Bedenken ein Problem. Das verknüpfen von Informationen über Patienten, wie Ernährung, Aktivitäten, Krankheitsverläufe usw., in der medizinischen Forschung ist für die Gesellschaft weniger ein Problem als das gleiche für die Risikobewertung in einer Krankenversicherung. Aus diesen Gründen beleuchtet die zweite Ebene die Gefahr des gläsernen Menschen. Die dritte Ebene befasst sich mit dem Algorithmen Design für die Anforderungen an Big Data mit dem HACE Theorem. Diese Algorithmen müssen die heterogenen und teilweise ungenauen bis falschen oder auch doppelte Daten filtern und homogenisieren. Das gefundene Wissen bzw. die Muster oder Modelle müssen nach der Analyse zusammen gefasst und in ihrer Gesamtheit betrachtet werden.

4 Zusammenfassung

MITTELS Data Mining werden Zusammenhänge und „Wissen“ aus relativ kleinen Datenbeständen erzeugt. Im Alltag kommt jeder mit dessen Ergebnissen in Kontakt, sei es bei der Google Suche oder einem Einkauf bei Amazon. Big Data verändert unser Verständnis von der Wissensgewinnung und verdrängt das „Bauchgefühl“ durch auf Fakten und Statistiken fundierte Entscheidungen. Die Gefahr des gläsernen Mensch ist bei Big Data omnipräsent.

Danksagungen

EIN großer Dank gilt meinem Seminarbetreuer Patrick Siegl (Kontakt unter www.tu-braunschweig.de/c3e/staff) sowie meinen Freunden für deren Unterstützung und Korrekturlesen.

Literatur

[1] 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.

[2] 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

[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.

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

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

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

[8] 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.