
Einführung in Web- und Data-Science

Prof. Dr. Ralf Möller

Universität zu Lübeck

Institut für Informationssysteme

Tanya Braun (Übungen)

Übersicht

- Einführung, Klassifikation vs. Regression, parametrisches und nicht-parametrisches überwachtes Lernen
- Neuronale Netze und Support-Vektor-Maschinen
- Häufungsanalysen, Warenkorbanalyse, Empfehlungen
- Statistische Grundlagen: Stichproben, Schätzer, Verteilung, Dichte, kumulative Verteilung, Skalen: Nominal-, Ordinal-, Intervall- und Verhältnisskala, Hypothesentests, Konfidenzintervalle, Reliabilität, Interne Konsistenz, Cronbach Alpha, Trennschärfe
- Bayessche Statistik, Bayessche Netze zur Spezifikation von diskreten Verteilungen, Anfragen, Anfragebeantwortung, Lernverfahren für Bayessche Netze bei vollständigen Daten
- Induktives Lernen: Versionsraum, Informationstheorie, Entscheidungsbäume, Lernen von Regeln
- Ensemble-Methoden, Bagging, Boosting, Random Forests
- Clusterbildung, K-Means, Analyse der Variation (Analysis of Variation, ANOVA), Inter-Cluster-Variation, Intra-Cluster-Variation, F-Statistik, Bonferroni-Korrektur, MANOVA, Discriminant Function Analysis
- Analyse Sozialer Strukturen

Web und Data Science

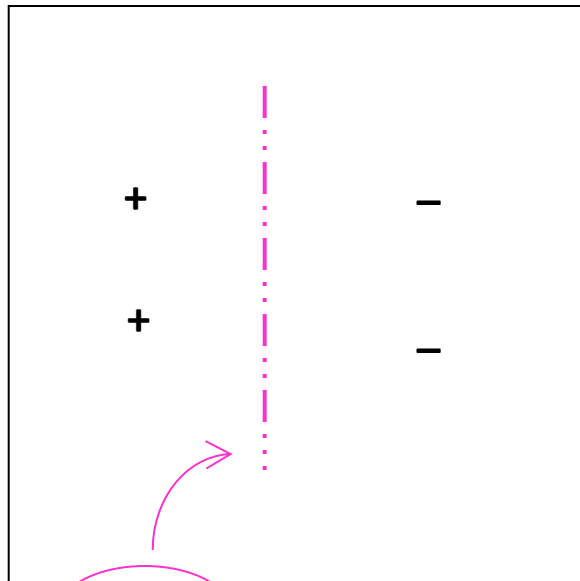


Bisher: Überwachtes Lernen

Wiederholung: Überwachtes Lernen

- Gegeben:
 - Tabellarische Daten,
 - Klassifikationsattribut vorhanden
(Überwachung durch klassifizierte Daten)
- Gesucht: Klassifikator für neue Daten
- Klassifikator erstellbar z.B. durch
 - Netze mit Parametrierungs- und Kompositionsmöglichkeit
 - Support-Vektor-Maschinen mit Kernel-Operator
- Heute: Unüberwachtes Lernen
(kein Klassifikationsattribut vorhanden)

Ausnutzung von Daten bei SVMs: Clusterbildung



SVM

überwacht

Nur klassifizierte Daten

Clusterbildung z.B. realisierbar durch k-nächste-Nachbarn-Klassifikation
(also instanzbasiert, nicht modellbasiert)

Web und Data Science



Semi-überwachtes Lernen

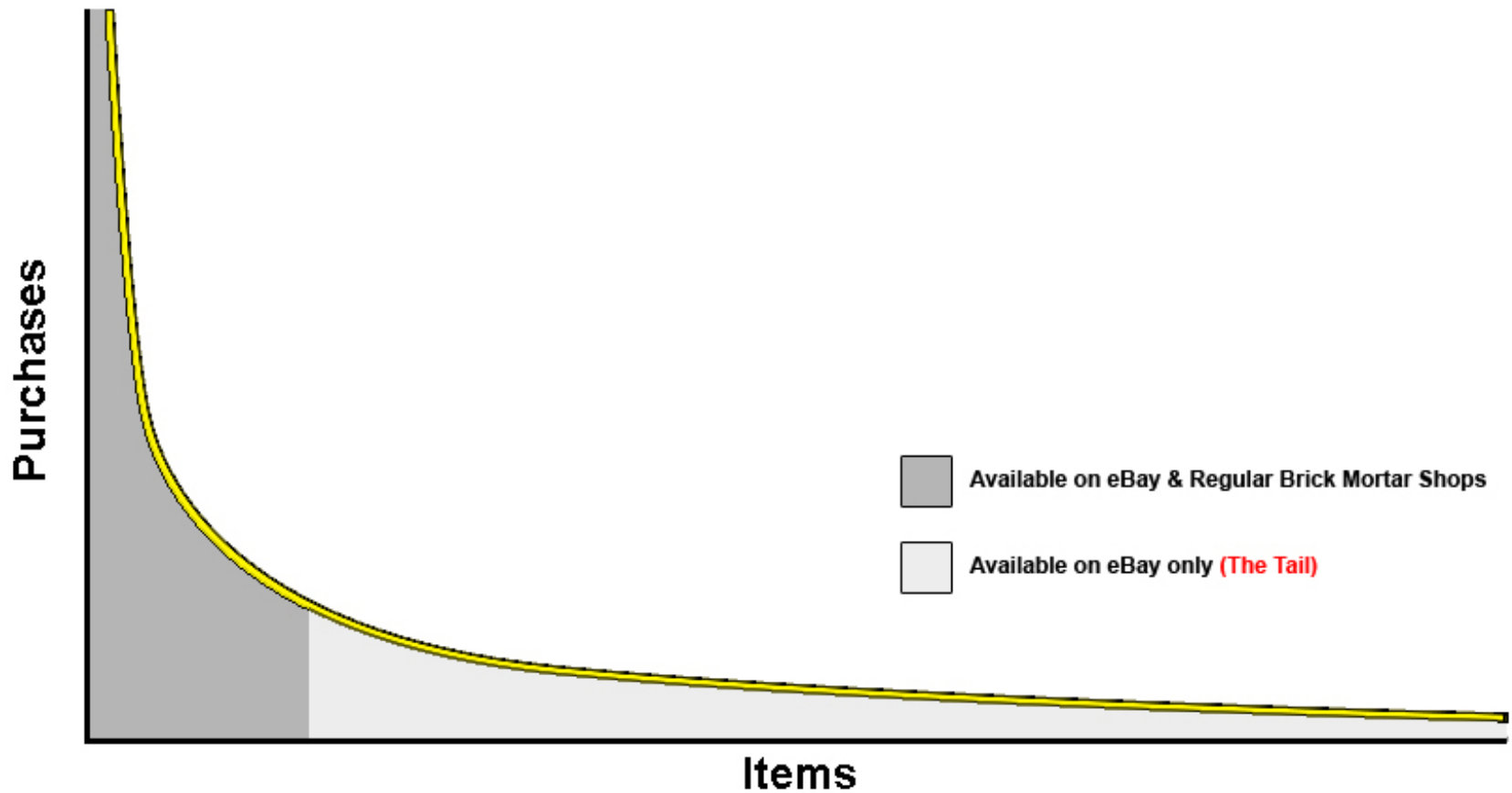
Über Windeln, Bier und Fahrräder ...



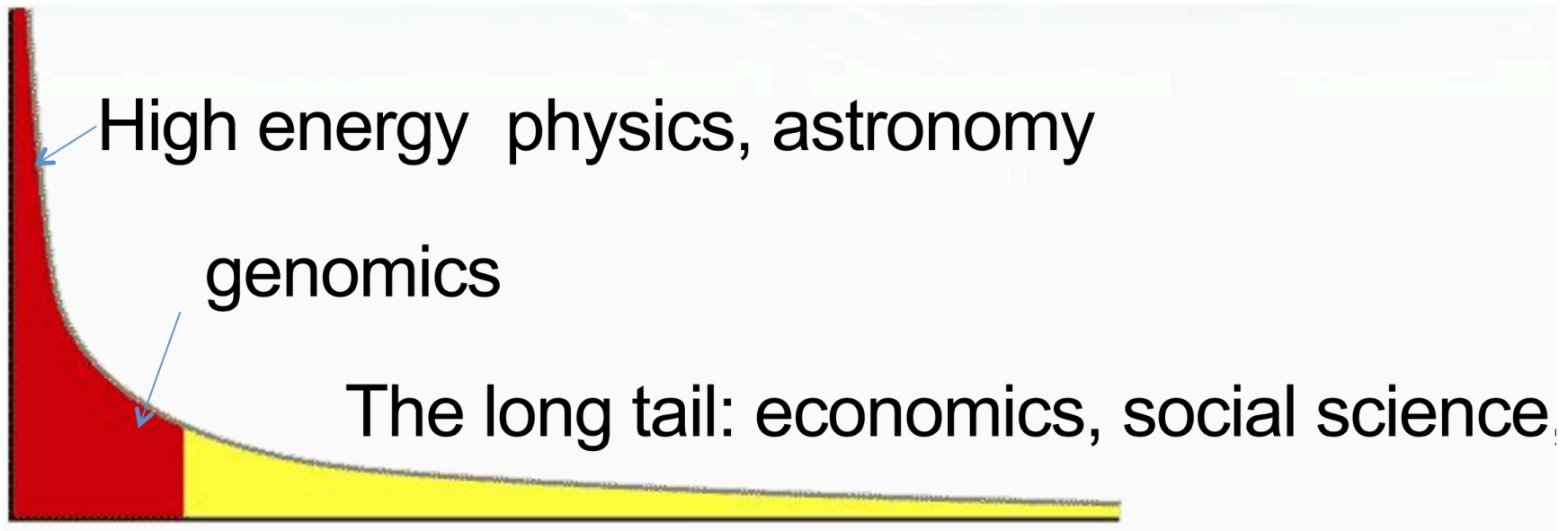
Eine Legende... Zeigt aber die Idee!

Long Tail

The Long Tail of **ebay**.



Nützliche Einsichten nur mit großen Datenmengen?



Wal-Mart-Beispiel: Unpersonalisierte Erfassung von "Warenkörben" über Kassenscanner.

Fragen

- Wie können Nutzerdaten personalisiert erfasst werden?
 - Haben Sie eine Payback-Karte?



Im stationären Handel: Die Karte wird während des Einkaufens beim Bezahlvorgang an der Kasse vorgelegt. Payback-Kundennummer, Datum, Filiale, Umsatz und von manchen Payback-Partnern auch Warengruppencodes werden an Payback übermittelt. Auf die Kaufsumme erhält der Kunde einen nach Unternehmen unterschiedlichen Bonus als Gegenleistung für seine Daten in Form von Punkten auf seinem Punktekonto gutgeschrieben. „Punktesammeln“ bezeichnet im Folgenden diesen Tauschhandel. [\[Wikipedia\]](#)

Als Vorteil erhofft sich der Händler, dass Kunden die eine Payback-Karte haben, bevorzugt bei ihm einkaufen, um Punkte zu sammeln, statt bei seinen Konkurrenten. Auch kann er Payback-Coupons erstellen, um damit Werbeaktionen zu unterstützen und die Aufmerksamkeit der Kunden auf bestimmte Warengruppen oder Termine zu richten. Zudem erhalten die Partnerunternehmen Zugriff auf die durch das Data Mining gewonnenen Informationen zum Kundenverhalten, nicht nur von seinen Kunden, sondern von allen Kunden aller Payback-Partner.

- Was sind die zentralen Techniken und Probleme?

Assoziationsregeln

- Gegeben eine Menge von **Warenkörben**, finde **Regeln**, die das Auftreten eines Artikels (oder mehrerer Artikel) **vorhersagt**
- Warenkorbeintragung in DB im Jargon **Transaktion** genannt (Daten aus Online-Transaction-Processing, OLTP)

Warenkorbtransaktionen

<i>TID</i>	<i>Artikel</i>
1	Bread, Milk
2	Bread, Diaper, Beer, Eggs
3	Milk, Diaper, Beer, Coke
4	Bread, Milk, Diaper, Beer
5	Bread, Milk, Diaper, Coke

Beispiele für Assoziationsregeln

$\{\text{Diaper}\} \rightarrow \{\text{Beer}\},$
 $\{\text{Milk, Bread}\} \rightarrow \{\text{Diaper, Coke}\},$
 $\{\text{Beer, Bread}\} \rightarrow \{\text{Milk}\},$

Rakesh Agrawal, Tomasz Imieliński, Arun Swami: Mining Association Rules between Sets of Items in Large Databases. In: Proc. 1993 ACM SIGMOD International Conference on Management of data, SIGMOD Record. Bd. 22, Nr. 2, Juni 1993

Häufige Artikelmenngen

- Gegeben eine Datenmenge **D** in Form von Warenkörben, finde Kombination von Artikeln, die häufig zusammen vorkommen

Warenkorbtransaktionen

<i>TID</i>	<i>Artikel</i>
1	Bread, Milk
2	Bread, Diaper, Beer, Eggs
3	Milk, Diaper, Beer, Coke
4	Bread, Milk, Diaper, Beer
5	Bread, Milk, Diaper, Coke

Beispiele für häufige Artikelmenngen

{Diaper, Beer},
{Milk, Bread}
{Beer, Bread, Milk},

Definition: Häufige Artikelmenge

- **Artikelmenge** \subseteq Gesamt-Artikelmenge I
 - Z.B.: {Milk, Bread, Diaper}
- **Unterstützungszähler (scount)**
 - Anzahl **scount(w)** des Auftretens der Artikelmenge w in den Daten
(Anzahl der Warenkörbe, in denen Artikelmenge w vorkommt)
 - Z.B.: **scount({Milk, Bread, Diaper}) = 2**
- **Unterstützung (support)**
 - Anteil der Warenkörbe, in denen Artikelmenge w vorkommt: **sup(w) = scount(w) / |D|**
 - Z.B.: **sup({Milk, Bread, Diaper}) = 2/5**
- **Häufige Artikelmenge (frequent itemset)**
 - Artikelmenge mit **Support $\geq \text{minsup}$** (Schwellwert)

<i>TID</i>	<i>Artikel</i>
1	Bread, Milk
2	Bread, Diaper, Beer, Eggs
3	Milk, Diaper, Beer, Coke
4	Bread, Milk, Diaper, Beer
5	Bread, Milk, Diaper, Coke

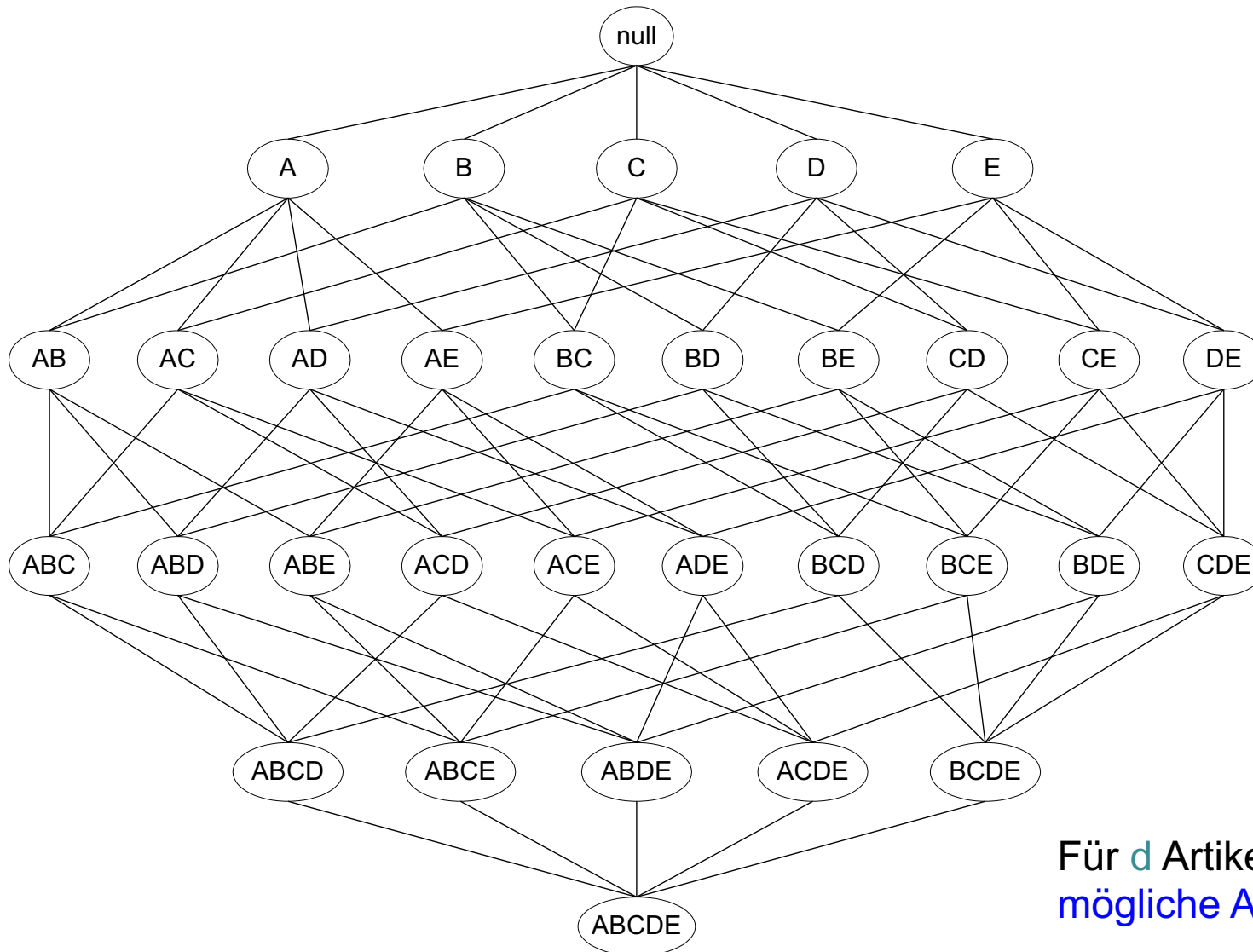
Warum häufige Artikelmengen finden?

- Interessant für **Platzierung** von Artikeln
(im Supermarkt, auf Webseiten, ...)
- Häufige Artikelmengen kennzeichnen positive Kombinationen
(seltene Artikelmengen kaum relevant),
bieten also **Zusammenfassung** einer Datenmenge
- ...

Suche nach häufigen Artikelmengen

- **Aufgabe:**
 - Gegeben eine Transaktionsdatenbasis D (Warenkörbe) und ein Schwellwert minsup
 - Finde alle häufigen Artikelmengen (und deren jeweilige Anzahl in den Daten)
- **Anders gesagt:** Zähle die jeweiligen Vorkommen von Kombinationen von Artikeln in den Daten über einem Schwellwert minsup
- **Annahme:** Gesamt-Artikelmenge I bekannt

Wie viele Artikelmengen gibt es?



Für d Artikel gibt es 2^d
mögliche Artikelmengen

Monotonie vom Support

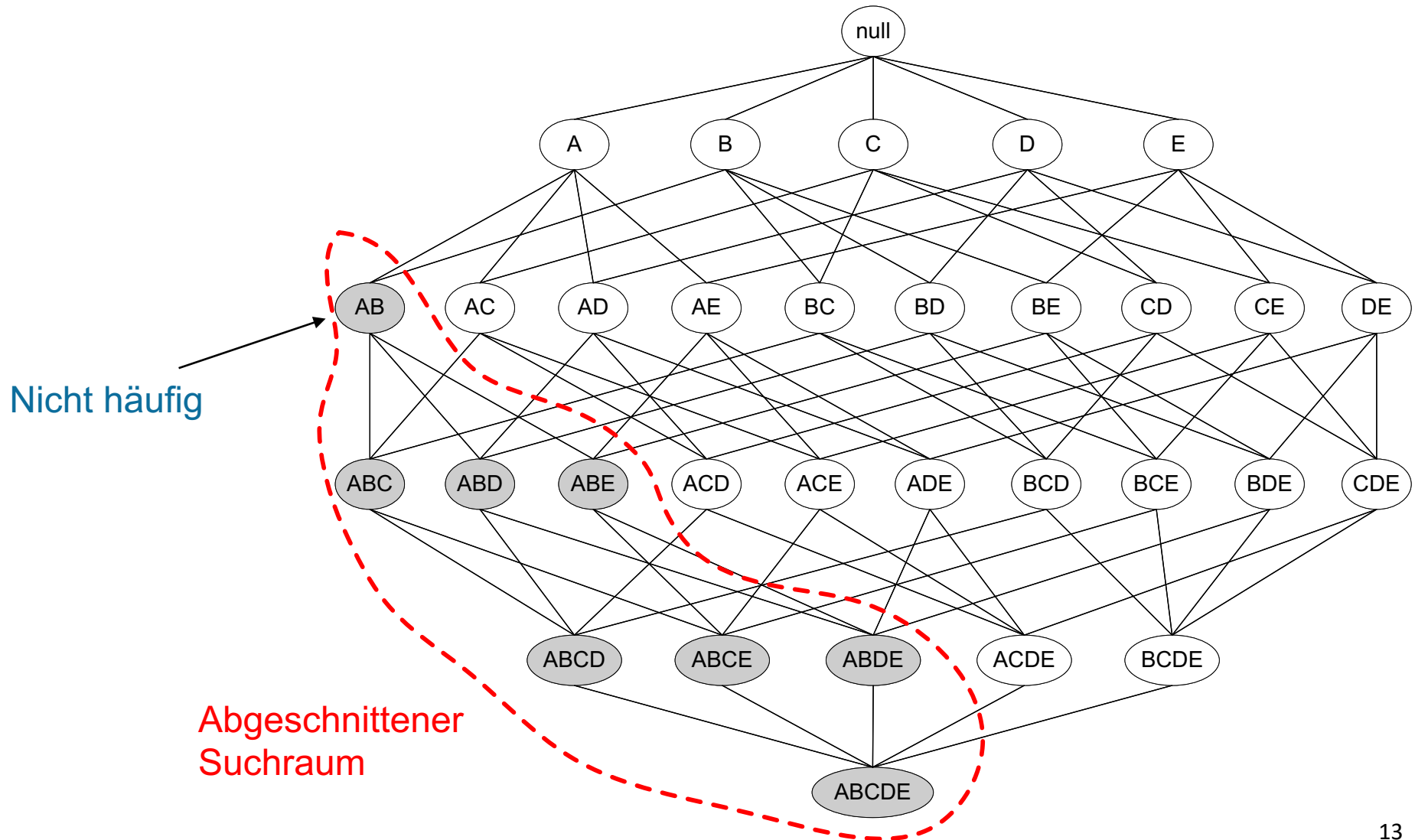
<i>TID</i>	<i>Artikel</i>
1	Bread, Milk
2	Bread, Diaper, Beer, Eggs
3	Milk, Diaper, Beer, Coke
4	Bread, Milk, Diaper, Beer
5	Bread, Milk, Diaper, Coke

$\text{sup}(\text{Bread}) > \text{sup}(\text{Bread, Beer})$

$\text{sup}(\text{Milk}) > \text{sup}(\text{Bread, Milk})$

$\text{sup}(\text{Diaper, Beer}) > \text{sup}(\text{Diaper, Beer, Coke})$

Apriori-Verfahren (Idee)



Apriori-Verfahren (Prinzip)

1er-Artikelmengen

Artikel	Anzahl
Bread	4
Coke	2
Milk	4
Beer	3
Diaper	4
Eggs	1

minsup = 3/5



Artikelmenge	Anzahl
{Bread,Milk}	3
{Bread,Beer}	2
{Bread,Diaper}	3
{Milk,Beer}	2
{Milk,Diaper}	3
{Beer,Diaper}	3

2er-Artikelmengen

(Cola und Eier nicht mehr berücksichtigt)



3er-Artikelmengen

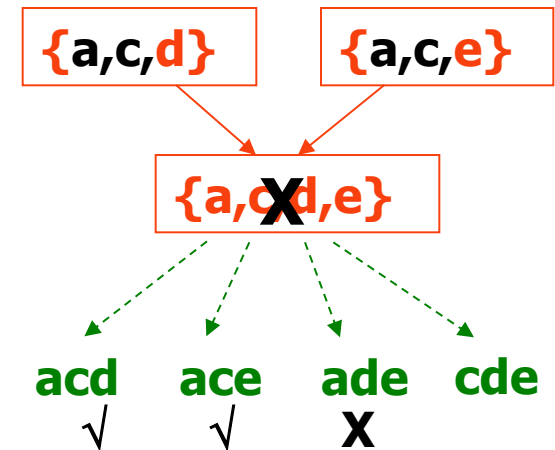
Artikelmenge	Anzahl
{Bread,Milk,Diaper}	3



Rakesh Agrawal, Ramakrishnan Srikant: *Fast Algorithms for Mining Association Rules*. In: Proc. 20th International Conference on Very Large Data Bases. Morgan Kaufmann Publishers Inc., 1994

Apriori-Verfahren (Prinzip)

- $L_3 = \{abc, abd, acd, ace, bcd\}$
- **Self-Join:** $L_3 \bowtie L_3$
 - $abcd$ aus abc und abd
 - $acde$ aus acd und ace
- **Beschneidung:**
 - $acde$ entfernt, weil ade nicht in L_3
- $C_4 = \{abcd\}$



Web und Data Science



Bisher: Analyse von Artikelmenngen



Heute im Angebot:
Personalisierte Empfehlungen

Definition: Assoziationsregel

Sei D eine Datenbasis von
Transaktionen

Transaction ID	Artikel
2000	A, B, C
1000	A, C
4000	A, D
5000	B, E, F

- Sei I die Artikelmenge in der DB, z.B.: $I = \{A, B, C, D, E, F\}$
- Eine Regel ist definiert durch $X \rightarrow Y$,
wobei $X \subset I$, $Y \subset I$, $X \neq \emptyset$, $Y \neq \emptyset$, and $X \cap Y = \emptyset$
 - Beispiel: $\{B, C\} \rightarrow \{A\}$ ist eine Regel

Bewertungsmaße für Regel $X \rightarrow Y$

■ Unterstützung/Support $\text{sup}(\cdot)$

- ▢ Anteil der Transaktionen, die X und Y enthalten

■ Konfidenz $\text{conf}(\cdot)$

- ▢ Maß, wie oft Artikel Y in Transaktionen vorkommen, die auch X enthalten

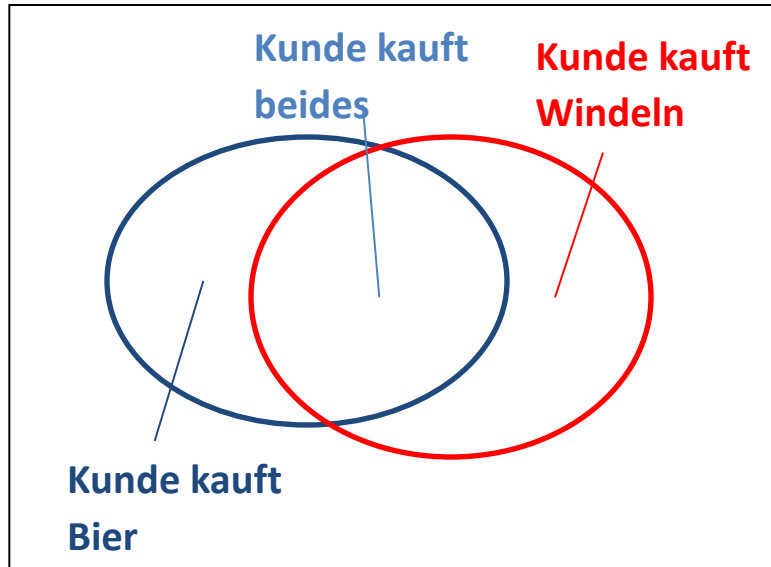
<i>TID</i>	<i>Items</i>
1	Bread, Milk
2	Bread, Diaper, Beer, Eggs
3	Milk, Diaper, Beer, Coke
4	Bread, Milk, Diaper, Beer
5	Bread, Milk, Diaper, Coke

Beispiel: $\{\text{Milk, Diaper}\} \rightarrow \text{Beer}$

$$\text{sup} = \frac{\text{scout}(\text{Milk, Diaper, Beer})}{|T|} = \frac{2}{5} = 0.4$$

$$c = \frac{\text{scout}(\text{Milk, Diaper, Beer})}{\text{scout}(\text{Milk, Diaper})} = \frac{2}{3} = 0.67$$

Mining von Assoziationsregeln



Finde alle Regeln $r = X \rightarrow Y$ mit

- $\text{sup}(r) \geq \text{minsup}$ und
- $\text{conf}(r) \geq \text{minconf}$
- **Support:** relative Häufigkeit (in %), von Transaktionen, die $X \cup Y$ enthalten
- **Konfidenz:** Bedingte relative Häufigkeit (in %) von Transaktionen, die Y enthalten, wenn sie auch X enthalten

TID	Items
100	A,B,C
200	A,C
300	A,D
400	B,E,F

Sei der minimale Support 50% und die minimale Konfidenz 50%:

- $A \rightarrow C$ (50%, 66.6%)
- $C \rightarrow A$ (50%, 100%)

Brute-Force-Verfahren

- Betrachte alle möglichen Assoziationsregeln
- Berechne Support und Konfidenz für jede Regel
- Eliminiere Regeln, deren Support oder Konfidenz kleiner als **minsup** und **minconf** Schwellwerte
- \Rightarrow **Zu aufwendig!** Kombinatorische Explosion

Einführung in Web und Data Science

Problem: Kombinatorische Explosion

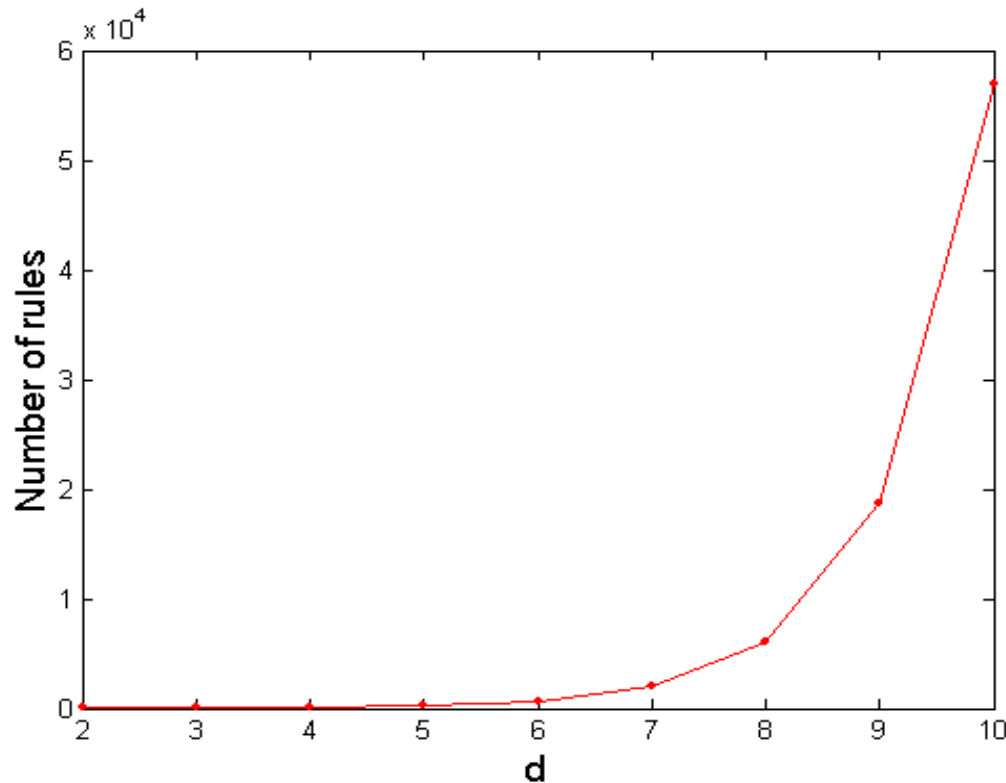


Killerkriterium auf komplexen Daten

Berechnungsaufwand

- Gegeben d Artikel in I :
 - Anzahl der Artikelmenngen: 2^d
 - Anzahl der Assoziationsregeln:

$$\sum_{k=1}^{d-1} \left[\binom{d}{k} \times \sum_{j=1}^{d-k} \binom{d-k}{j} \right]$$
$$= 3^d - 2^{d+1} + 1$$



Wenn $d=6$, $R = 602$ Regeln

Mining von Assoziationsregeln

<i>TID</i>	<i>Items</i>
1	Bread, Milk
2	Bread, Diaper, Beer, Eggs
3	Milk, Diaper, Beer, Coke
4	Bread, Milk, Diaper, Beer
5	Bread, Milk, Diaper, Coke

Beispiele für Regeln:

$\{\text{Milk, Diaper}\} \rightarrow \{\text{Beer}\}$ (sup=0.4, conf=0.67)
 $\{\text{Milk, Beer}\} \rightarrow \{\text{Diaper}\}$ (sup=0.4, conf=1.0)
 $\{\text{Diaper, Beer}\} \rightarrow \{\text{Milk}\}$ (sup=0.4, conf=0.67)
 $\{\text{Beer}\} \rightarrow \{\text{Milk, Diaper}\}$ (sup=0.4, conf=0.67)
 $\{\text{Diaper}\} \rightarrow \{\text{Milk, Beer}\}$ (sup=0.4, conf=0.5)
 $\{\text{Milk}\} \rightarrow \{\text{Diaper, Beer}\}$ (sup=0.4, conf=0.5)

Beobachtungen:

Regeln sind binäre Partitionen der gleichen Artikelmenge:

$\{\text{Milk, Diaper, Beer}\}$

Regeln von der gleichen Artikelmenge haben gleichen Support aber verschiedene Konfidenz

Entkopplung von Support und Konfidenz

Zweischrittiger Ansatz

- Generiere häufige Artikelmengen mit

$\text{support} \geq \text{minsup}$

- Generiere Assoziationsregeln

durch **binäre Partitionierung**
von häufigen Artikelmengen, so dass

$\text{confidence} \geq \text{minconf}$

Regelgenerierung – Einfacher Ansatz

- Gegeben die häufige Artikelmenge X , finde alle nichtleeren Teilmengen $y \subset X$, so dass $y \rightarrow X - y$ die Konfidenzanforderung erfüllt

Beispiel: $\{A,B,C,D\}$ sei eine häufige Artikelmenge:

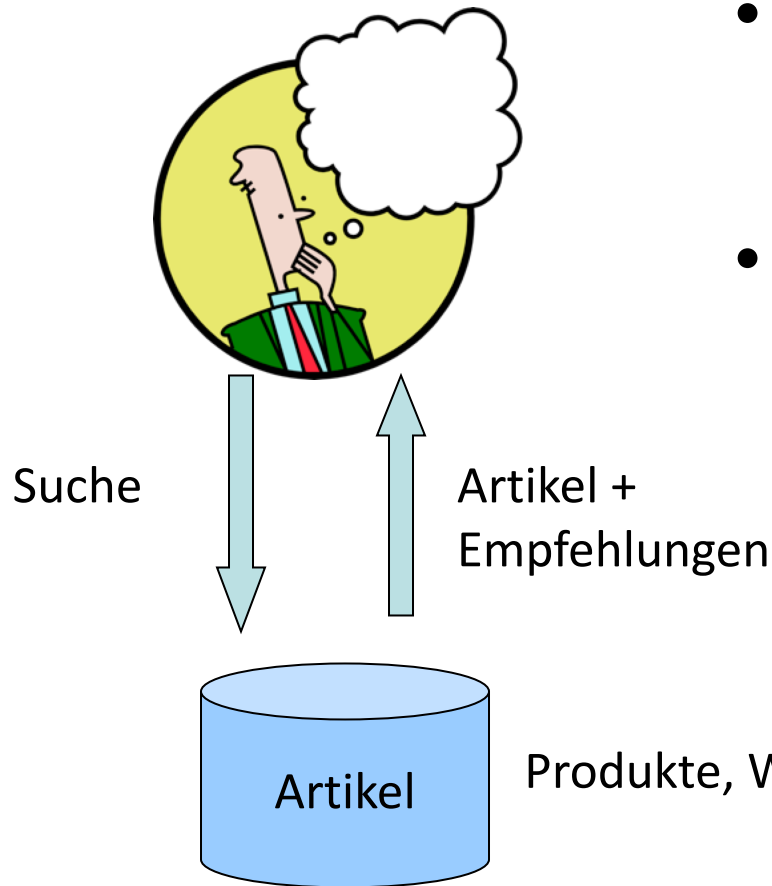
$ABC \rightarrow D,$	$ABD \rightarrow C,$	$ACD \rightarrow B,$	$BCD \rightarrow A,$
$A \rightarrow BCD,$	$B \rightarrow ACD,$	$C \rightarrow ABD,$	$D \rightarrow ABC$
$AB \rightarrow CD,$	$AC \rightarrow BD,$	$AD \rightarrow BC,$	$BC \rightarrow AD,$
$BD \rightarrow AC,$	$CD \rightarrow AB$		

- Falls $|X| = k$, dann gibt es $2^k - 2$ Kandidatenregeln (ohne $L \rightarrow \emptyset$ und $\emptyset \rightarrow L$)

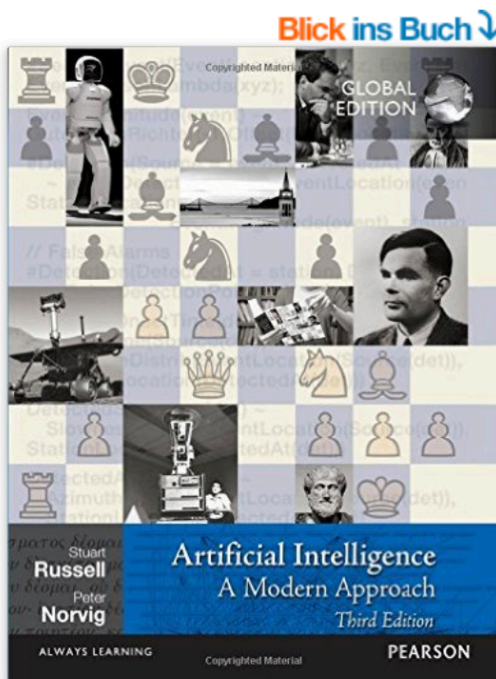
Anwendungen der Warenkorbanalyse

- "Verstehen" von Transaktionsdaten
- Data Mining (offline)
 - Systematische Anwendung statistischer Methoden auf große Datenbestände, mit dem Ziel, ...
 - ... neue Querverbindungen und Trends zu erkennen
- Empfehlungsgenerierung (online)
 - Vorhersage treffen, die quantifiziert wie stark das Interesse eines Benutzers an einem Objekt ist, ...
 - ... um dem Benutzer genau die Objekte aus der Menge aller vorhandenen Objekte zu empfehlen, für die er sich vermutlich am meisten interessiert

Anwendung: Empfehlungsgenerierung



- Beschränktheit der Ressource "Platz"
- Was sind "gute" Empfehlungen
 - Steigerung der Kundenzufriedenheit
 - Steigerung des Umsatzes des Anbieters
- Techniken
 - Verwendung von häufigen Artikelmenngen
 - Verwendung von Assoziationsregeln



Artificial Intelligence (Englisch) Taschenbuch – 2016

von [Stuart J. Russell](#) (Autor), [Peter Norvig](#) (Autor)



[2 Kundenrezensionen](#)

► [Alle Formate und Ausgaben anzeigen](#)

Kindle Edition
EUR 37,44

Taschenbuch
EUR 69,95

[Lesen Sie mit unserer **kostenfreien App**](#)

Lieferung Dienstag, 15. Nov.: Bestellen Sie innerhalb **47 Stunden und 26 Minuten** und wählen Sie dieses Datum an der Kasse. [Siehe Details.](#)

51 neu ab **EUR 40,48**

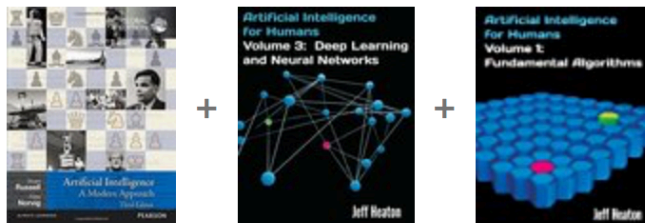
For one or two-semester, undergraduate or graduate-level courses in Artificial Intelligence. The long-anticipated revision of this best-selling text offers the most comprehensive, up-to-date introduction to the theory and practice of artificial intelligence.



[Dieses Bild anzeigen](#)

Verwendung von häufigen Artikelmengen

Wird oft zusammen gekauft



Gesamtpreis: **EUR 115,04**

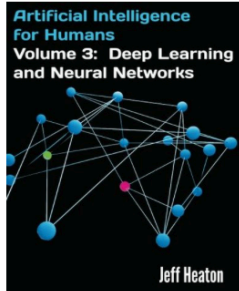
[Alle drei in den Einkaufswagen](#)

- ✓ **Dieser Artikel:** Artificial Intelligence von Stuart J. Russell Taschenbuch **EUR 69,95**
- ✓ Artificial Intelligence for Humans, Volume 3: Deep Learning and Neural Networks von Jeff Heaton Taschenbuch **EUR 24,61**
- ✓ Artificial Intelligence for Humans, Volume 1: Fundamental Algorithms von Jeff Heaton Taschenbuch **EUR 20,48**

Verwendung von Assoziationsregeln

Kunden, die diesen Artikel gekauft haben, kauften auch

Seite 1 von 24



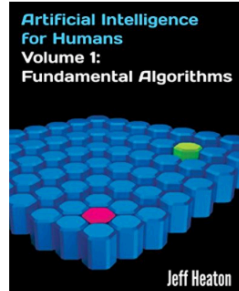
Artificial Intelligence for Humans, Volume 3: Deep Learning and Neural...

› Jeff Heaton

★★★★★ 2

Taschenbuch

EUR 24,61 ✓Prime



Artificial Intelligence for Humans, Volume 1: Fundamental Algorithms

› Jeff Heaton

★★★★★ 2

Taschenbuch

EUR 20,48 ✓Prime



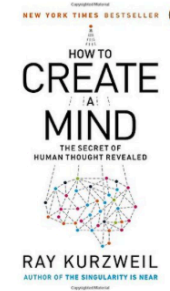
Artificial Intelligence for Humans, Volume 2: Nature-Inspired Algorithms

› Jeff Heaton

★★★★★ 1

Taschenbuch

EUR 19,83 ✓Prime



How to Create a Mind: The Secret of Human Thought Revealed

› Ray Kurzweil

★★★★★ 10

Taschenbuch

EUR 13,94 ✓Prime



Harry Potter und das verwunschene Kind - Teil eins und zwei und über 4,5 Millionen weitere Bücher verfügbar für Amazon Kindle. Erfahren Sie mehr

◀ Zurück zu den Suchergebnissen für "harry potter"



Harry Potter: Harry Potter und das verwunschene Kind. Teil eins und zwei (Special Rehearsal Edition Script) Gebundene

Ausgabe – 24. September 2016

von J.K. Rowling ▼ (Autor), John Tiffany (Autor), Jack Thorne (Autor), & 2 mehr

★★★★☆ 743 Kundenrezensionen

Bestseller Nr. 1 in Fantasy für Jugendliche

▶ Alle Formate und Ausgaben anzeigen

Kindle Edition
EUR 14,99Gebundene Ausgabe
EUR 19,99Lesen Sie mit unserer **kostenfreien App**

Teilen ✉ f t p

EUR 19,99

Alle Preisangaben inkl. MwSt.

Kostenlose Lieferung.

Auf Lager.

Verkauf und Versand durch Amazon.
Geschenkverpackung verfügbar.

Menge: 1 ▼



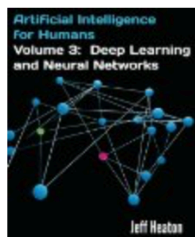
In den Einkaufswagen

Loggen Sie sich ein, um 1-Click®
einzuschalten.

Ihre zuletzt angesehenen Artikel und besonderen Empfehlungen

Inspiriert von Ihrem Browserverlauf

Seite 1 von 10

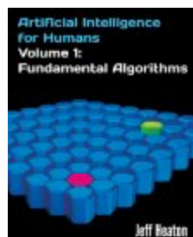
Artificial Intelligence for
Humans, Volume 3:...

▶ Jeff Heaton

★★★★☆ 2

Taschenbuch

EUR 24,61 ✓Prime

Artificial Intelligence for
Humans, Volume 1:...

▶ Jeff Heaton

★★★★☆ 2

Taschenbuch

EUR 20,48 ✓Prime

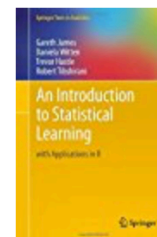
Artificial Intelligence for
Humans, Volume 2:...

▶ Jeff Heaton

★★★★☆ 1

Taschenbuch

EUR 19,83 ✓Prime

An Introduction to
Statistical Learning:...

▶ Gareth James

★★★★☆ 3

Gebundene Ausgabe

EUR 58,49 ✓Prime



Data Science from Scratch

▶ Joel Grus

★★★★☆ 2

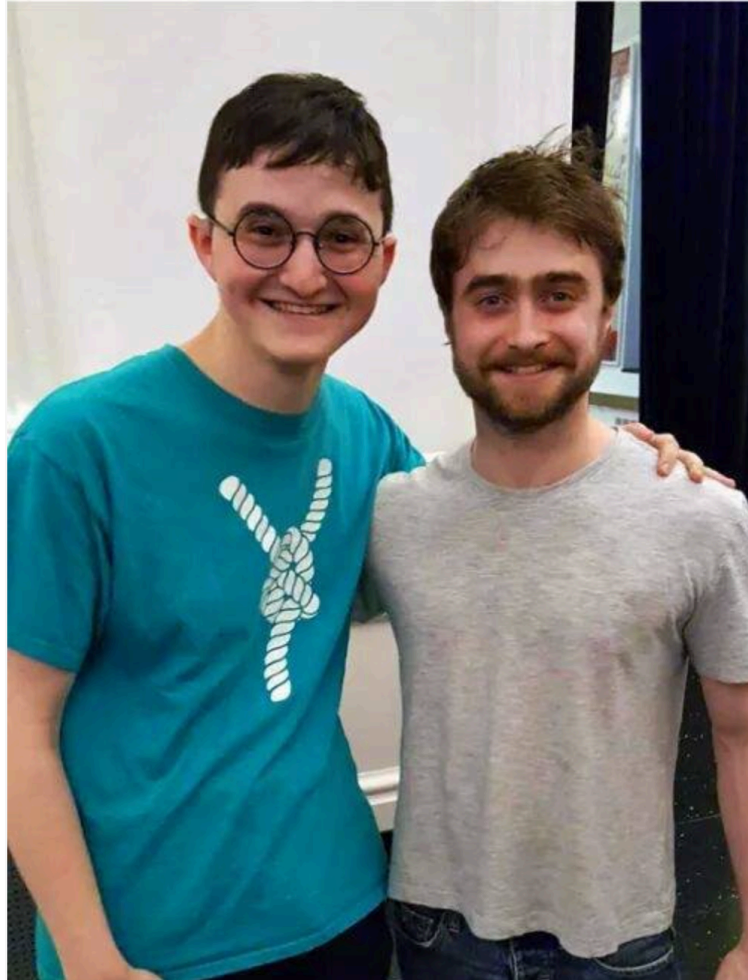
Taschenbuch

EUR 26,88 ✓Prime



Personalisierung

"My son looks more like Harry Potter than Harry Potter."



Verfeinerung der Empfehlungsgenerierung

Personalisierung: Kundenspezifische Empfehlung

Schätzung der Kundenzufriedenheit über "Nützlichkeit"

- **C:** Menge von Kunden
- **S:** Menge von Artikeln
- **Nützlichkeitsmaß:** $u : C \times S \rightarrow R$
 - **R:** Menge von Bewertungen (total geordnete Menge)
 - Beispiele: 0-5 Sterne, reelle Zahlen aus $[0,1]$
- Nützlichkeit = engl. Utility

Maximierung der Nützlichkeitsschätzung

- Für jeden Nutzer $c \in C$
bestimme diejenigen Artikel s' aus dem Sortiment S ,
die die Nützlichkeiten für den Nutzer c maximieren


$$\forall c \in C : s'_c = \arg \max_{s \in S} u(c, s)$$

- Kundenspezifische Nützlichkeit $u(c, s)$ eines Artikels s
definiert Rang von s für Kunden c

Zentrales Problem

- Nützlichkeit nur partiell definiert, also nicht für alle Elemente aus dem **CxS Raum** bekannt
- Nützlichkeit **u** muss extrapoliert werden

	King Kong	LOTR	Matrix	Nacho Libre
Alice	1		0.2	
Bob		0.5		0.3
Carol	0.2		1	
David				0.4

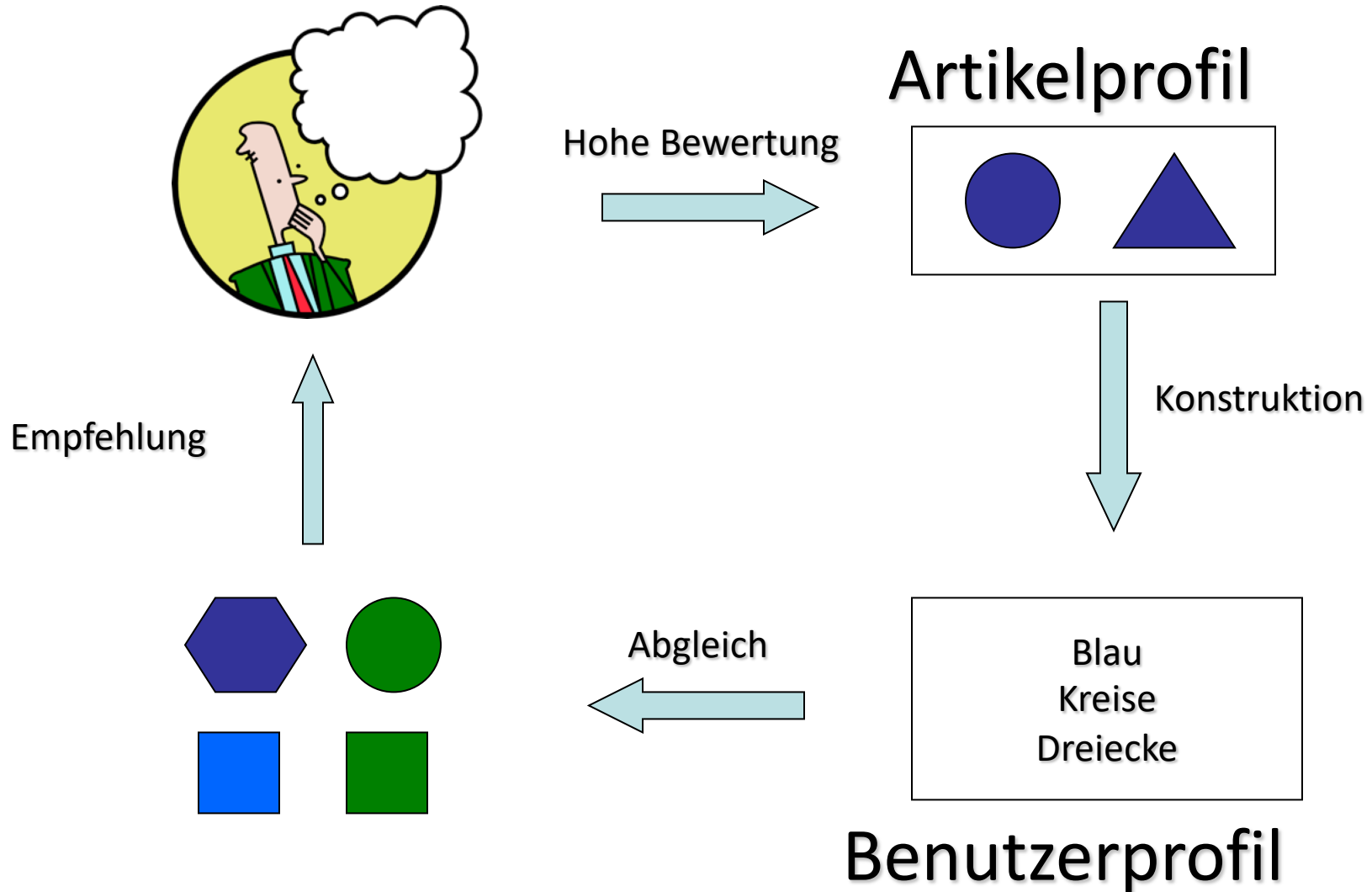


Woher
kommen die
Einträge?

Erfassung von Nützlichkeitsmaßen

- Explizit
 - Nutzer bewerten Artikel
 - Funktioniert nicht in der Praxis, Nutzer werden gestört
- Implizit
 - Erfasse Maße aus Nutzeraktionen
 - Kauf eines Artikels ergibt gute Bewertung
 - Was ist mit schlechten Bewertungen?

Übersicht



Einführung in Web- und Data-Science

Prof. Dr. Ralf Möller

Universität zu Lübeck

Institut für Informationssysteme

Tanya Braun (Übungen)

Empfehlungsgenerierung

- Nützlichkeitsmaß: $u : C \times S \rightarrow R$
- Für jeden Nutzer $c \in C$
bestimme diejenigen Artikel s' aus dem Sortiment S ,
die die Nützlichkeiten für den Nutzer c maximieren:
- Was ist das Empfehlungsgenerierungsproblem?

$$\forall c \in C : s'_c = \arg \max_{s \in S} u(c, s)$$

	King Kong	LOTR	Matrix	Nacho Libre
Alice	1		0.2	
Bob		0.5		0.3
Carol	0.2		1	
David				0.4

Extrapolierung der Nützlichkeiten

- Schlüsselproblem: Matrix U ist dünn besetzt
 - Die meisten Leute haben die meisten Artikel nicht bewertet
 - Extrapolation nötig (Filterung)

Ansätze: Wie löse ich das Empfehlungsgenerierungsproblem?

- Inhaltsbasierte Filterung

- Empfehlung von Artikeln, die "ähnlich" zu den schon hoch bewerteten sind: Wähle $u(c, s)$ wie $u(c, s')$ mit $\text{sim}(s, s')$

- Kollaborative Filterung

- Empfehlung von Artikeln, die von "ähnlichen" Benutzern hoch bewertet werden: $u(c, s)$ wie $u(c', s)$ mit $\text{sim}(c, c')$

Inhaltsbasierte Filterung: Artikelmerkmale

- Für jeden Artikel s generiere Artikelprofil $\text{content}(s)$
- Profil ist Menge von Merkmalswerten
 - Text: Menge von (Wort, Gewicht)-Paaren
 - Kann als Vektor gedeutet werden
 - Auch für Filme:
 - Text extrahieren aus Angaben zu Titel, Schauspieler, Regisseur, usw.
- Wie findet man Gewichtsangaben?
 - Standardansatz: TF.IDF
(Term Frequency by Inverse Doc Frequency)

TF.IDF

f_{ij} = relative Anzahl der Terme t_i im Document d_j

$$TF_{ij} = \frac{f_{ij}}{\max_k f_{kj}}$$

n_i = Anzahl der Dokumente in denen Term i vorkommt

N = Gesamtanzahl der Dokumente

$$IDF_i = \log \frac{N}{n_i}$$

TF.IDF-Maß $w_{ij} = TF_{ij} \cdot IDF_i$

Inhaltsbasierte Nützlichkeitsschätzung (Filterung)

- Für Nutzer c nehme zugeordnete Artikel $\text{items}(c)$...
- ... und bestimme $\text{content}(s)$ für alle $s \in \text{items}(c)$
- Definiere $\text{profile}(c)$ (z.B.) als
 - ♦ Mittel der $\text{content}(s)$ für alle $s \in \text{items}(c)$
 - ♦ (weitere Definitionen sind möglich)
- Wir erhalten: Menge von (Term, Gewicht)-Paaren
 - ♦ Kann als Vektor w gedeutet werden

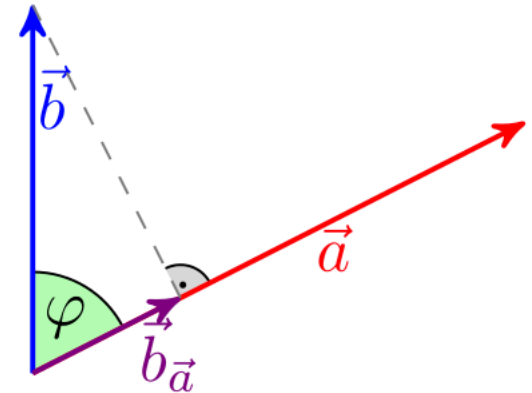
- Nützlichkeitsfunktion $u(c, s)$:
$$u(c, s) = \cos(\vec{w}_c, \vec{w}_s) = \frac{\vec{w}_c \cdot \vec{w}_s}{\|\vec{w}_c\|_2 \times \|\vec{w}_s\|_2}$$
- Profil auch Bewertung genannt
$$= \frac{\sum_{i=1}^K w_{i,c} w_{i,s}}{\sqrt{\sum_{i=1}^K w_{i,c}^2} \sqrt{\sum_{i=1}^K w_{i,s}^2}},$$

Skalarprodukt oder Punktprodukt

$$\mathbf{a} \cdot \mathbf{b} = \|\mathbf{a}\| \|\mathbf{b}_a\|$$

$$\mathbf{a} \cdot \mathbf{b} = \|\mathbf{a}\| \|\mathbf{b}\| \cos \varphi$$

$$\cos(\varphi) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\| \|\mathbf{B}\|} = \frac{\sum_{i=1}^n A_i B_i}{\sqrt{\sum_{i=1}^n A_i^2} \sqrt{\sum_{i=1}^n B_i^2}}$$



Orthogonale Projektion \vec{b}_a des Vektors \vec{b} auf die durch \vec{a} bestimmte Richtung

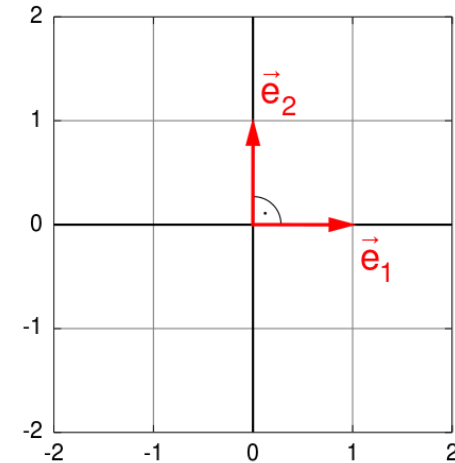
$$\vec{a} \cdot \vec{b} = a_1 b_1 + a_2 b_2$$

Für die **kanonischen Einheitsvektoren** $\vec{e}_1 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$ und $\vec{e}_2 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$ gilt nämlich:

$$\vec{e}_1 \cdot \vec{e}_1 = 1, \vec{e}_1 \cdot \vec{e}_2 = \vec{e}_2 \cdot \vec{e}_1 = 0 \text{ und } \vec{e}_2 \cdot \vec{e}_2 = 1$$

Daraus folgt

$$\begin{aligned} \vec{a} \cdot \vec{b} &= (a_1 \vec{e}_1 + a_2 \vec{e}_2) \cdot (b_1 \vec{e}_1 + b_2 \vec{e}_2) \\ &= a_1 b_1 \vec{e}_1 \cdot \vec{e}_1 + a_1 b_2 \vec{e}_1 \cdot \vec{e}_2 + a_2 b_1 \vec{e}_2 \cdot \vec{e}_1 + a_2 b_2 \vec{e}_2 \cdot \vec{e}_2 \\ &= a_1 b_1 + a_2 b_2 \end{aligned}$$



Kanonische Einheitsvektoren in der euklidischen Ebene

Begrenzungen der inhaltsbasierten Filterung

- Wie gut ist die gewählte Problemlösung?
- Merkmale nicht immer einfach zu definieren
 - Bilder?, Musik?
 - Meist umgebender oder zugeordneter Text verwendet
- Überspezialisierung
 - Artikel außerhalb des Profils werden nicht empfohlen
 - Menschen haben verschiedene Interessen
 - Clusterbildung?
- Empfehlungen für neue Benutzer
 - Wie ist das Profil definiert?
 - Rückgriff auf:
 - Häufige Artikelmenen (nutzerunspezifisch)
 - Assoziationsregeln (nutzerunspezifisch)

Nutzer-Nutzer kollaborative Filterung

- Betrachte Nutzer c
- Bestimme Menge D von Nutzern, deren Bewertungen "ähnlich" zu denen von c sind
- Schätze $\text{profile}(c)$ aus den Angaben $\text{profile}(d)$ für $d \in D$
- Was sind "ähnliche" Nutzer?

Ähnliche Nutzer: Distanzmaße

Sei eine Nutzerbewertung $r_c = \text{profile}(c)$ gegeben, dann definiere Ähnlichkeit sim der Nutzer c_1 und c_2 als

1. Kosinus-Ähnlichkeit

– $\text{sim}(c_1, c_2) = \cos(r_{c_1}, r_{c_2})$ oder als

2. Funktion über Bewertungen (Profile)

$x = r_{c_1}$ und $y = r_{c_2}$, so dass

- falls c_1 und c_2 gleiche Bewertungen vergeben $\rightarrow \max$
- Normalisierung von x und y nötig
- Allgemein bekannt als:
 - Pearson Korrelationskoeffizient oder
 - empirischer Korrelationskoeffizient

Korrelationskoeffizient

Normalisierte Werte (z-Transformation)

$$r = \frac{1}{n} \sum_{i=1}^n \frac{(x_i - \bar{x})}{s_x} \frac{(y_i - \bar{y})}{s_y}$$

n = Anzahl der Artikel		
x_i = Bewertungskomp. x		y_i = Bewertungskomp. y
\bar{x} = Mittel von x		\bar{y} = Mittel von y
s_x = Standardabweichung x		s_y = Standardabweichung y

Einschub

- Eigentlich

$$r = \frac{1}{n-1} \sum_{i=1}^n \frac{(x_i - \bar{x})}{s_x} \frac{(y_i - \bar{y})}{s_y}$$

- Warum man das so definiert, ist nicht gleich offensichtlich, wir kommen später darauf zurück
- Ähnliches gilt für die Varianz (korrigierte Varianz):

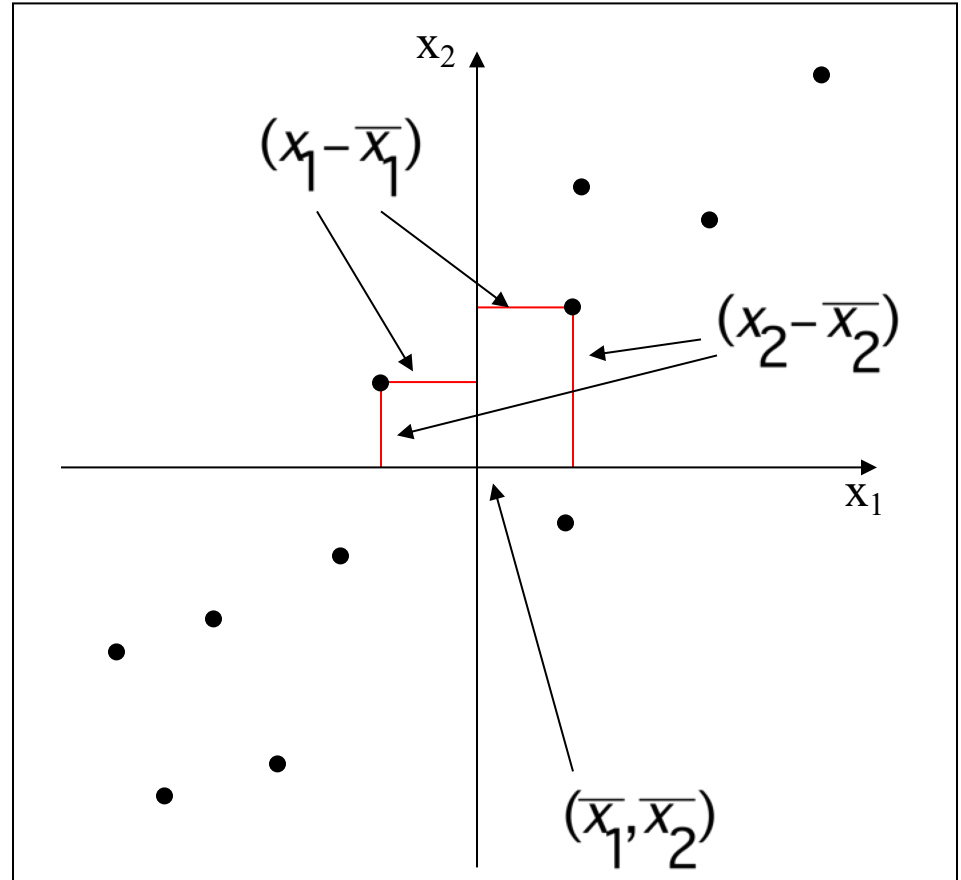
$$s^2 = \frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2$$

Korrelationskoeffizient

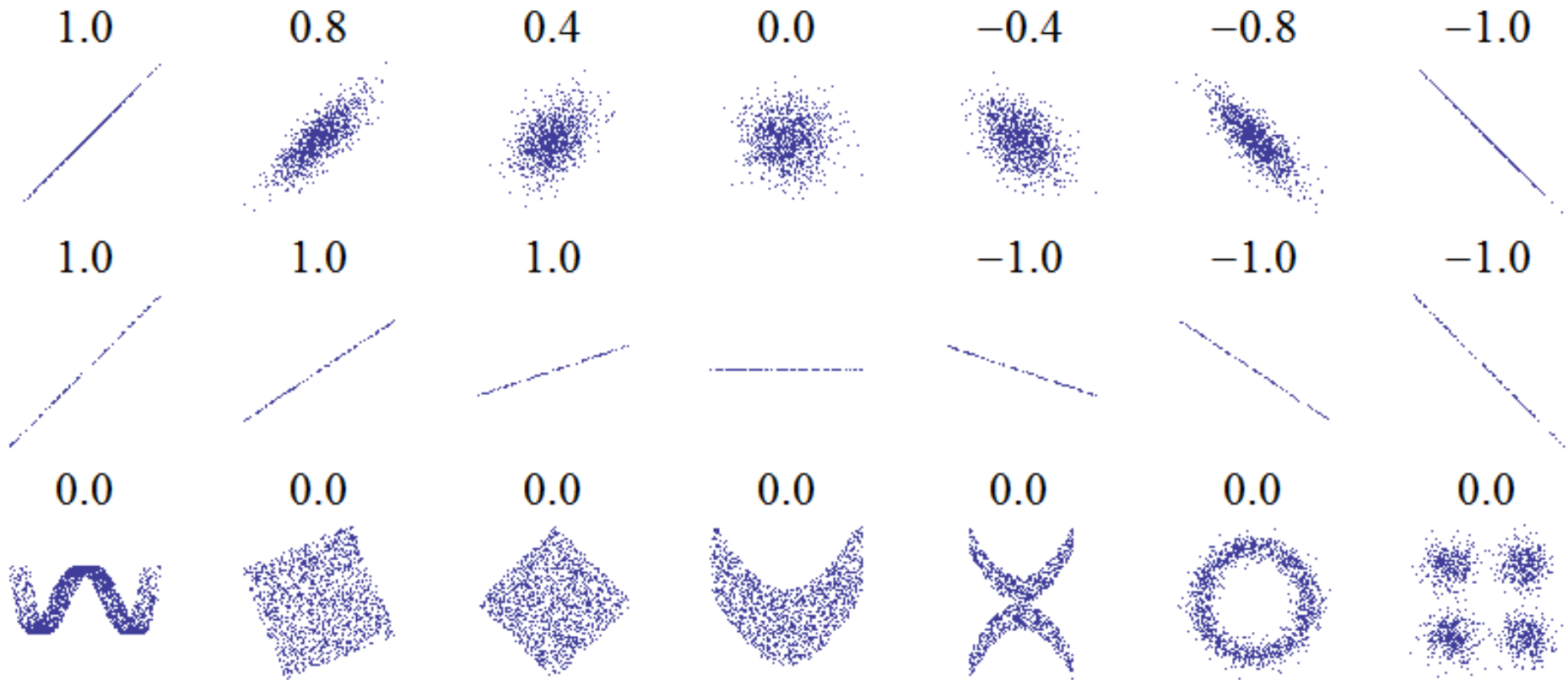
$$s_x = \sqrt{\frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2}$$

$$r = \frac{\sum_{i=1}^n (x_{i1} - \bar{x}_1)(x_{i2} - \bar{x}_2)}{\sqrt{\sum_{i=1}^n (x_{i1} - \bar{x}_1)^2} \sqrt{\sum_{i=1}^n (x_{i2} - \bar{x}_2)^2}}$$

$$-1 \leq r \leq +1$$



Korrelationskoeffizient: Veranschaulichung



Bewertungsschätzungen

- Sei D die Menge der k ähnlichsten Nutzer zu c , die Artikel s bewertet haben
- Definiere Schätzfunktion für Bewertung von s :
 - $r_{cs} = 1/k \sum_{d \in D} r_{ds}$ oder
 - $r_{cs} = (\sum_{d \in D} \text{sim}(c,d) \cdot r_{ds}) / (\sum_{d \in D} \text{sim}(c,d))$
 - ...

Aufwand?

- Aufwendige Suche nach k ähnlichsten Nutzern
 - Betrachtung aller Nutzer?
- Kann kaum zur "Laufzeit" erfolgen
 - Vorausberechnung nötig
- Alternative zur Berechnung von r_{cs} ?
 - Suche nach ähnlichen Artikeln
 - Artikel-Artikel-kollaborative-Filterung
 - Sonst gleiches Vorgehen
 - Suche Assoziationsregeln mit s als Vorbedingung
 - Suche häufige Artikelmenngen mit s als Element

Not much has changed



Really?

From Brick-and-Mortar Shops to Webshops