Artificial Intelligence

2–20 September

:

4 contributing lectures, 1,2 / 4

:

im Rahmen der

:

Sommer der KI, FU Berlin

Christoph Schommer
University Luxembourg

10 September 2019

Screenplay, 9 und 10 September 2019

Generelles

1. Die (traditionelle) Künstliche Intelligenz (**strong AI**) basiert auf der Idee, kognitive, menschlicher Fähigkeiten durch eine Maschine zu simulieren. Beispiele: Sprachverstehen, Sprechen, Objekterkennung, Wissensrepräsentation, Planen, Logik, und vieles mehr.

2. Das Thema *Data Science* hat sich etabliert und eine Schnittstelle zur traditionellen KI geschaffen (**weak AI**). Insbesondere werden maschinelle Lernverfahren als Träger genannt.

3. Im Allgemeinen soll die Künstliche Intelligenz dem Menschen dienen (Künstliche Intelligenz für den Menschen, Artificial Intelligence for social good). Anwendungen von autonomen Systems müssen deswegen sehr kritisch hinterfragt werden (weniger: Deep Learning and Arts, mehr: Deep Learning and financial transactions, Healthcare).

4. Knowledge Discovery/Data Mining ist ein Fachgebiet, das in der Schnittmenge liegt: einerseits Massendaten und andererseits die Anwendung von machinellen Lernverfahren.

5. Unter Knowledge Discovery/Data Mining versteht man eine explorative (daten-gesteuerte, bottom-up) Suche nach versteckten, aber wertvollen/nützlichen Informationen (oder Wissen) in Massen von Daten (*Big Data*).

6. Der Term *Data Mining* wird eher im industriellen Sprachgebrauch und der Term *Knowlwedge Discovery* eher im akademischen Sprachgebrauch benutzt. Eine weitere Beobachtung ist, dass die Diskussion um die Datenaufbereitung im Sinne eines *Knowledge Discovery* oft schon abgeschlossen ist, während sie beim Data Mining Teil des Prozesses ist.

7. Eine weitere Unterscheidung: verifikative Ansätze sind solche, bei denen es darum geht, aufgestellte Anfragen *lediglich* zu verarbeiten/verifizieren. Beispiele: SQL-Anfragen an die Datenbank, Eingabe von Schlüsselwörter in einer Suchmaschine. Explorative Ansätze sind solche, die daten-getrieben agieren und verschiedene Arten von Informationen liefern: einige davon sind nützlich, vielleicht sogar neu ('gold nuggets'), andere redundant bzw wertlos. Die Entscheidung (durch Interpretation) trifft meistens der Mensch (aber auch: Maschine; siehe Finanzmarkt).

8. Knowledge Discovery/Data Mining ist explorativ und bedingt einen **Kreislauf**: Daten ⇒ Informationen ⇒ Wissen⇒ Daten. Dabei gilt:

   (a) Daten müssen unter Umständen aufbereitet (*Data preprocessing*) und/oder anonymisiert werden (siehe *GDPR*) und vor allem <u>verstanden</u> werden. Man betrachtet hier eine Reihe von Themen, wie etwa *Data Cleaning* (Eliminieren von Rechtschreibfehler, gleiche

Representation für gleiche Ausdrücke,...), Datenreduktion (*Sampling*, Attribute Reduction/Variablendiskussion, Generalisierungen von Datenwerte, ...), Anonymisierung von Daten (GDPR), Visualisierung von Daten, et cetera.

(b) Machinelle Lernverfahren können dann auf bereinigte Datenmengen angewendet werden, etwa supervised/unsupervised learning, symbolische/subsymbolische Verfahren zum Zwecke der Regelfindung, des Pattern Matching, et cetera. Typische Answendungsgebiete: Association Discovery, Classification, Clustering, Zeitreihenanalyse, und andere.

(c) Die gefundenen Informationen werden diskutiert und interpretiert. einige Informationen kommen in den Müll, andere zur Anwendung. Beispiele: *Wer Schuhe kauft, kauft auch eine Plastiktüte* ist statistisch sicher interessant (bedingte Wahrscheinlichkeit jenseits von 0.9), aber praktisch nicht relevant (da Plastiktüte als Teil eines Kaufesun interessant).

(d) Mögliche Aktionen auf gefundene (valuable) Informationen (= Erkenntnisse): Optimierungen (Kataloge, Verkaufsraum im Supermarkt), Verbessern der Kundenzufriedenheit, Aufdecken saisonaler Effekte (Ostern, Karneval, Urlaussaison, Weihnachten, et cetera).

(e) Da diese Aktionen ein verändertes Verhalten bedingt, müssen Daten wieder behandelt werden (siehe oben).

9. Vorgestellte Beispiele: TARGET (kanadische Drogeriekette → der Manager des Kaufhauses weiß mehr als die Eltern; England-Beispiel / Debenhams, Newcastle).

Disziplin 1: Assoziationsanalyse (Association Discovery)

1. Ziel: entdecke Regeln der Form $X \to Y$ mit $X = (x_1, \ldots, x_k)$ und $Y = (y_1, \ldots, y_l)$. x_i, y_j heißen *items*, X und Y *itemsets*.

2. Der Begriff *assoziativ* bezieht sich auf ein gemeinsames + ausreichendes Vorkommen von items in einer zugrunde liegenden Transaktionsdatenbank T.

3. Wie findet man *Assoziationsregeln*? Beispiel *apriori*-Algorithm von Agrawal, Srikant (1994):

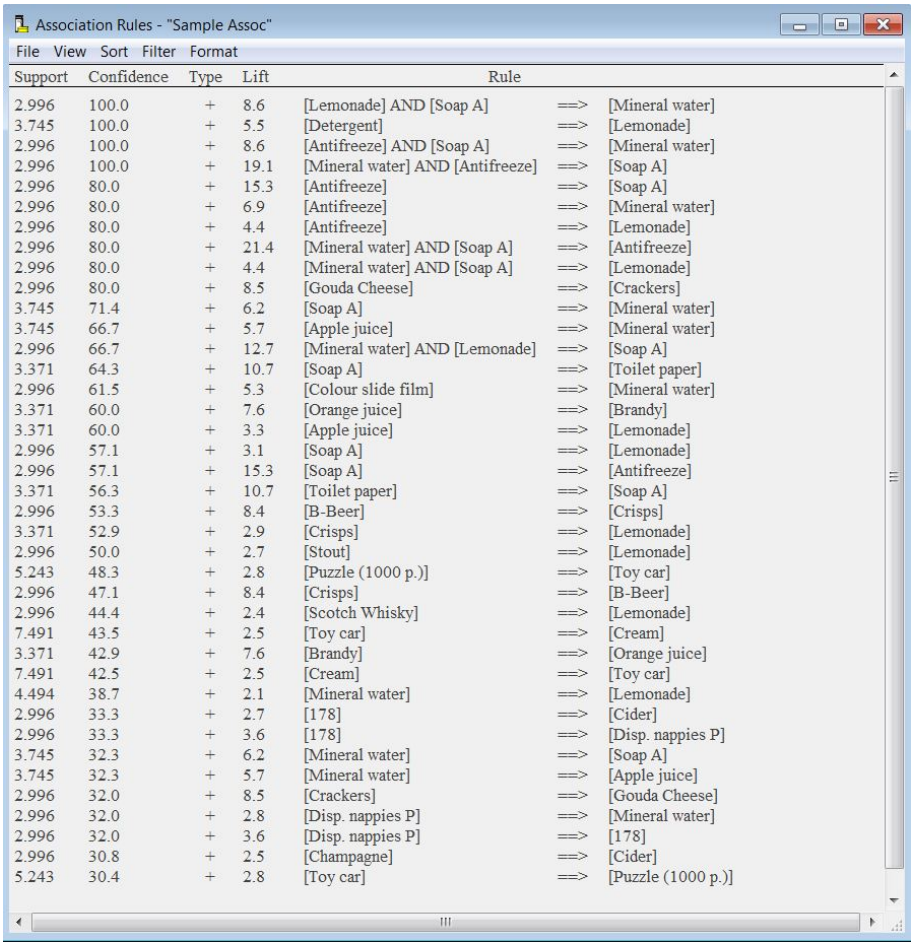
   (a) Gegeben seien Transaktionsdaten, wie etwa:

   100: A, B, C

   200: B, C

   ...

   (b) Support(X) = $\frac{Anzahl-der-Transaktionen,-die-X-enthalten}{Alle-Transaktionen}$ (= relative Häufigkeit). *min_support* bezeichnet einen Schwellenwert, der das Mindestmaß an gemeinsam vorkommenden Items/Itemsets misst (Komplexität der Joins).



Support	Confidence	Type	Lift	Rule		
2.996	100.0	+	8.6	[Lemonade] AND [Soap A]	==>	[Mineral water]
3.745	100.0	+	5.5	[Detergent]	==>	[Lemonade]
2.996	100.0	+	8.6	[Antifreeze] AND [Soap A]	==>	[Mineral water]
2.996	100.0	+	19.1	[Mineral water] AND [Antifreeze]	==>	[Soap A]
2.996	80.0	+	15.3	[Antifreeze]	==>	[Soap A]
2.996	80.0	+	6.9	[Antifreeze]	==>	[Mineral water]
2.996	80.0	+	4.4	[Antifreeze]	==>	[Lemonade]
2.996	80.0	+	21.4	[Mineral water] AND [Soap A]	==>	[Antifreeze]
2.996	80.0	+	4.4	[Mineral water] AND [Soap A]	==>	[Lemonade]
2.996	80.0	+	8.5	[Gouda Cheese]	==>	[Crackers]
3.745	71.4	+	6.2	[Soap A]	==>	[Mineral water]
3.745	66.7	+	5.7	[Apple juice]	==>	[Mineral water]
2.996	66.7	+	12.7	[Mineral water] AND [Lemonade]	==>	[Soap A]
3.371	64.3	+	10.7	[Soap A]	==>	[Toilet paper]
2.996	61.5	+	5.3	[Colour slide film]	==>	[Mineral water]
3.371	60.0	+	7.6	[Orange juice]	==>	[Brandy]
3.371	60.0	+	3.3	[Apple juice]	==>	[Lemonade]
2.996	57.1	+	3.1	[Soap A]	==>	[Lemonade]
2.996	57.1	+	15.3	[Soap A]	==>	[Antifreeze]
3.371	56.3	+	10.7	[Toilet paper]	==>	[Soap A]
2.996	53.3	+	8.4	[B-Beer]	==>	[Crisps]
3.371	52.9	+	2.9	[Crisps]	==>	[Lemonade]
2.996	50.0	+	2.7	[Stout]	==>	[Lemonade]
5.243	48.3	+	2.8	[Puzzle (1000 p.)]	==>	[Toy car]
2.996	47.1	+	8.4	[Crisps]	==>	[B-Beer]
2.996	44.4	+	2.4	[Scotch Whisky]	==>	[Lemonade]
7.491	43.5	+	2.5	[Toy car]	==>	[Cream]
3.371	42.9	+	7.6	[Brandy]	==>	[Orange juice]
7.491	42.5	+	2.5	[Cream]	==>	[Toy car]
4.494	38.7	+	2.1	[Mineral water]	==>	[Lemonade]
2.996	33.3	+	2.7	[178]	==>	[Cider]
2.996	33.3	+	3.6	[178]	==>	[Disp. nappies P]
3.745	32.3	+	6.2	[Mineral water]	==>	[Soap A]
3.745	32.3	+	5.7	[Mineral water]	==>	[Apple juice]
2.996	32.0	+	8.5	[Crackers]	==>	[Gouda Cheese]
2.996	32.0	+	2.8	[Disp. nappies P]	==>	[Mineral water]
2.996	32.0	+	3.6	[Disp. nappies P]	==>	[178]
2.996	30.8	+	2.5	[Champagne]	==>	[Cider]
5.243	30.4	+	2.8	[Toy car]	==>	[Puzzle (1000 p.)]

Figure 1: Assoziativregeln, sortiert nach *Confidence*.

(c) Confidence(X→Y) = $\frac{P(X,Y)}{P(X)}$. *min_confidence* bezeichnet das Mindestmaß an Regelstärke (minimale bedingte Wahrscheinlichkeit).

(d) Lift(X→Y) = $\frac{Confidence(X \to Y)}{Confidence^*(X \to Y)}$, wobei Confidence* sich auf die Idee, dass X und Y statistisch unabhängig sind, bezieht. Deswegen ist hier $P(X, Y)$ gleich dem Produkt der Einzelwahrscheinlichkeiten.

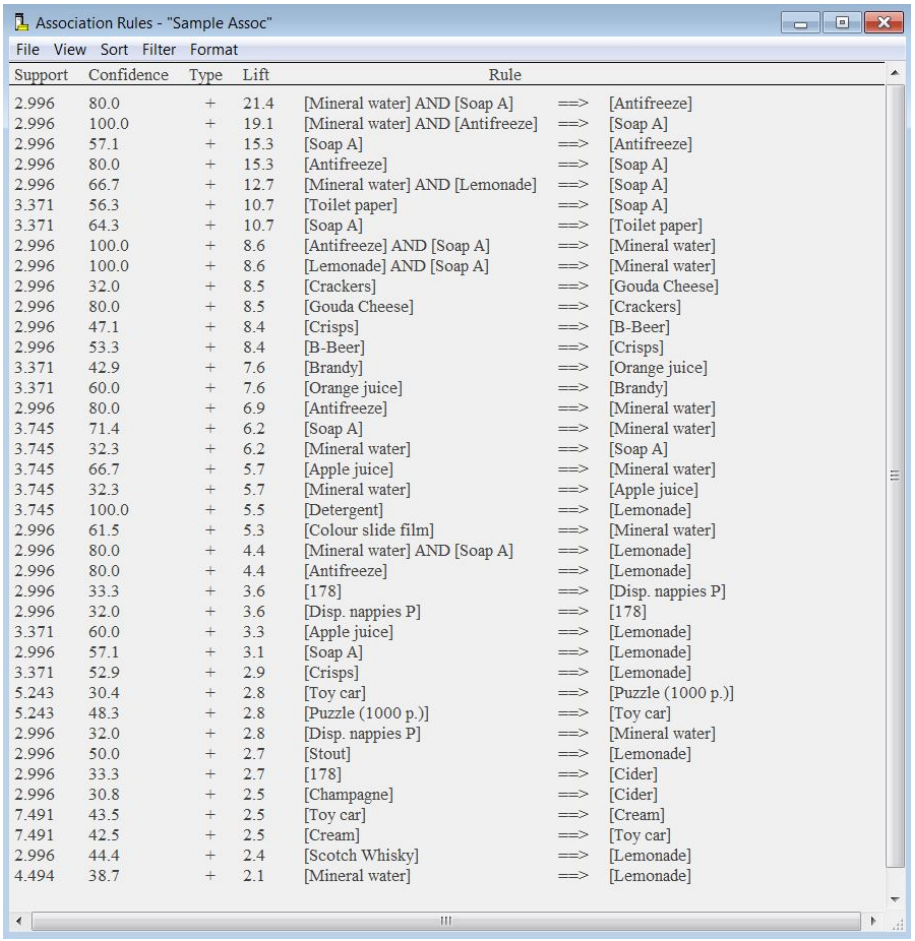
(e) Es existiert eine Ordnung (nicht quantitativ zu sehen!): $x_1 < x_2, \ldots < x_k$.

(f) Gegeben die Itemsets $X = (x_1, \ldots, x_k)$ und $Y = (y_1, \ldots, y_l)$. Dann ist der Join(X,Y) = $X = (x_1, \ldots, x_k, y_l)$, falls $x_1 = y_1, x_2 = y_2, \ldots, x_{k-1} = y_{l-1}, x_k \neq y_l$.

(g) Candidate Itemsets (C) sind solche itemsets, die möglicherweise den support-Schwellenwert erfüllen. Falls sie es tun, werden sie large itemset genannt (L).

(h) Der Algorithmus besteht aus **2 Phasen**.

- Phase 1(Kandidatenbildung). Diese Phase ist iterativ und besteht aus zwei wiederkehrenden Schritte: im ersten Schritt werden *Large Itemsets* ge-joined und – gemäß der



Support	Confidence	Type	Lift	Rule		
2.996	80.0	+	21.4	[Mineral water] AND [Soap A]	==>	[Antifreeze]
2.996	100.0	+	19.1	[Mineral water] AND [Antifreeze]	==>	[Soap A]
2.996	57.1	+	15.3	[Soap A]	==>	[Antifreeze]
2.996	80.0	+	15.3	[Antifreeze]	==>	[Soap A]
2.996	66.7	+	12.7	[Mineral water] AND [Lemonade]	==>	[Soap A]
3.371	56.3	+	10.7	[Toilet paper]	==>	[Soap A]
3.371	64.3	+	10.7	[Soap A]	==>	[Toilet paper]
2.996	100.0	+	8.6	[Antifreeze] AND [Soap A]	==>	[Mineral water]
2.996	100.0	+	8.6	[Lemonade] AND [Soap A]	==>	[Mineral water]
2.996	32.0	+	8.5	[Crackers]	==>	[Gouda Cheese]
2.996	80.0	+	8.5	[Gouda Cheese]	==>	[Crackers]
2.996	47.1	+	8.4	[Crisps]	==>	[B-Beer]
2.996	53.3	+	8.4	[B-Beer]	==>	[Crisps]
3.371	42.9	+	7.6	[Brandy]	==>	[Orange juice]
3.371	60.0	+	7.6	[Orange juice]	==>	[Brandy]
2.996	80.0	+	6.9	[Antifreeze]	==>	[Mineral water]
3.745	71.4	+	6.2	[Soap A]	==>	[Mineral water]
3.745	32.3	+	6.2	[Mineral water]	==>	[Soap A]
3.745	66.7	+	5.7	[Apple juice]	==>	[Mineral water]
3.745	32.3	+	5.7	[Mineral water]	==>	[Apple juice]
3.745	100.0	+	5.5	[Detergent]	==>	[Lemonade]
2.996	61.5	+	5.3	[Colour slide film]	==>	[Mineral water]
2.996	80.0	+	4.4	[Mineral water] AND [Soap A]	==>	[Lemonade]
2.996	80.0	+	4.4	[Antifreeze]	==>	[Lemonade]
2.996	33.3	+	3.6	[178]	==>	[Disp. nappies P]
2.996	32.0	+	3.6	[Disp. nappies P]	==>	[178]
3.371	60.0	+	3.3	[Apple juice]	==>	[Lemonade]
2.996	57.1	+	3.1	[Soap A]	==>	[Lemonade]
3.371	52.9	+	2.9	[Crisps]	==>	[Lemonade]
5.243	30.4	+	2.8	[Toy car]	==>	[Puzzle (1000 p.)]
5.243	48.3	+	2.8	[Puzzle (1000 p.)]	==>	[Toy car]
2.996	32.0	+	2.8	[Disp. nappies P]	==>	[Mineral water]
2.996	50.0	+	2.7	[Stout]	==>	[Lemonade]
2.996	33.3	+	2.7	[178]	==>	[Cider]
2.996	30.8	+	2.5	[Champagne]	==>	[Cider]
7.491	43.5	+	2.5	[Toy car]	==>	[Cream]
7.491	42.5	+	2.5	[Cream]	==>	[Toy car]
2.996	44.4	+	2.4	[Scotch Whisky]	==>	[Lemonade]
4.494	38.7	+	2.1	[Mineral water]	==>	[Lemonade]

Figure 2: Assoziativregeln, sortiert nach *Support*.

Definition des Join – um 1 Element erweitert. Diese neue Itemset nennt sich dann *Candidate Itemset*. In einem weiteren Schritt wird dann geprüft, ob diese *Candidate Itemset* den *support-Schwellenwert* erfüllt (oder nicht). Diesen Vorgang nennt man *Pruning*. Im ersten Fall geht es iterativ weiter (die *Candidate Itemset* ist jetzt *large*), im zweiten Schritt wird die Itemset verworfen.

- Phase 2 (Regelbildung). Die *Large Itemset* der ersten Phase werden verwendet und zu Regeln geformt. Beispiel: ein Large Itemset {A B C} generiert folgende Regeln: A B $\rightarrow$ C; C $\rightarrow$ A B; A C $\rightarrow$ B; B $\rightarrow$ A C; A $\rightarrow$ B C; B C $\rightarrow$ A. Für alle gilt: ist der *Confidence* (= bedingte Wahrscheinlichkeit) ausreichend, dann ist die Regel akzeptiert, ansonsten nicht.

(i) Die Wahl der Schwellenwerte müssen vorsichtig gewählt, ansonsten kann die Laufzeit durchaus 12h oder mehr betragen...

(j) Man Assoziationsregeln als Regeln zeigen oder auch als Graph (Knoten entspricht dem

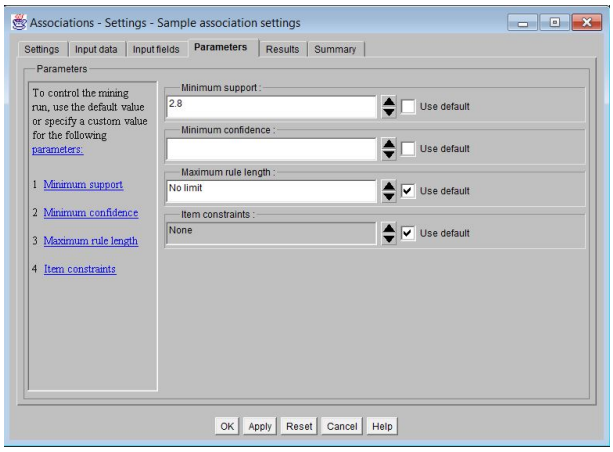


Figure 3: Parameter Setting: $min_{support}$, $min_{confidence}$, Regellänge, Item-spezifische Operationen, etwa: zeige nur Regeln, in denen "Coca Cola" vorkommt.

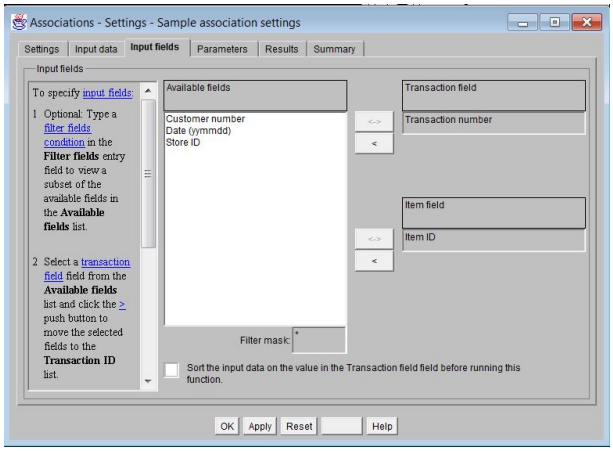


Figure 4: Verwendete Beispielattribute aus einer Transaktionsdatenbank

Large Itemset; Knotendicke entspricht dem Support; Farbe der Kanten zwischen den Knoten entsprechen den Confidence-Werten, Dicke der Kante dem Lift; und so weiter).

(k) Anwendung in der Warenkorbanalyse (Market Basket Analysis): Offerierung optimierte Portfolios und Kataloge; neues Arrangement der Artikel im Supermarkt/Kaufhaus; und andere).

(l) Praktisches: man kann auf Basis zeitlich (unabhängiger) Datenmengen natürlich Assoziationen über die Zeit hinweg vergleichen und auf Basis der statistischen Kennzahlentwicklung eventuelle Trends motivieren (etwa: der Confidenec für A→B hat sich in den letzten 52 Wochen stark erhöht, von 0,1 auf 0,8).

4. Der Algorithmus beinhalt viel Raum für Weiterentwicklungen. Etwa: falls ein item doppelt oder häufiger in derselben Transaktion vorkommt, sollte das auch in der Berechnung des

Supportwertes eine Rolle spielen. Die Wichtigkeit eines Items wäre unter Umständen auch zu berücksichtigen: das Gewicht eines items 'Pelzmantel' ist sicher höher als das Gewicht für ein item 'Strumpf'.

5. Wir haben auch gesehen, dass wir *Taxonomien* (= Hierarchien) verwenden können. Hierbei werden innere Knoten des Taxonomiebaumes wie die Blätter des Baumes behandelt: falls ein Nachkomme (eines inneren Knotens) in der Transaktion auftritt, erhöht sich auch der Support dieses inneren Knotens. Beispiel: der Support von *FOOD* erhöht sich, wenn etwa *Bananen* in der Transaktion auftritt. Regeln der Art $FOOD \rightarrow Schuhe$ sind also möglich.

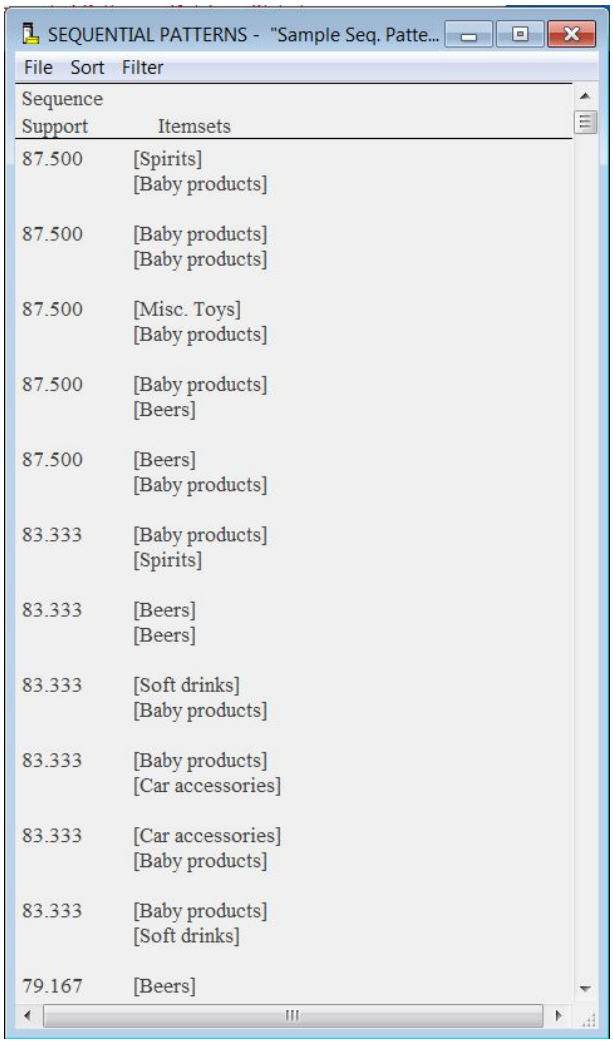


Figure 5: Ergebnis (Ausschnitt) von entdeckten Sequences, etwa: Wer *Spirits* kauft, kauft auch später *Baby Products*. Andmerkung: die Zeiteinheiten zwischen den Items sind in der Implementation nicht berücksichtigt, sodass Vorkommen der beiden Items innerhalb eines Tages oder eines Monats als gleich angesehen werden. Das ist natürlich ein gravierender Nachteil.

Disziplin 2: Sequential Patterns

1. Dieses Problem betrifft den Einbezug der Zeitkomponente in der Betrachtung von Items. Im Gegensatz zum Problem mit den Assoziationen geht es hier aber nicht um Regeln, sondern um Sequenzen. Beispiel: Kaufverhalten über die Zeit hinweg bei Amazon.

2. Man beschreibt eine Gleichzeitigkeit unter Verwendung einer Klammer. Beispiel: eine Sequenz `ab(cf)` bedeutet, dass zum Zeitpunkt t_1 `a` vorkommt, zum Zeitpunkt t_2 `b` und zum Zeitpunkt t_3 `c` und `f` (siehe etwa Kaufvorgänge bei Otto, Neckermann, Amazon, Ebay, und

anderen Online-Shops).

3. Eine sub-sequence S_k ist in einer sequence S enthalten, falls alle items von S_k auch entsprechend in S vorkommen. So ist S_1 = `ac` in einer sequence S = `ab(cf)` enthalten, eine Subsequence S_2 = `(ac)` aber nicht.

4. Es gibt mehrere Ideen, *large subsequences* in eine Ansammlung von sequences zu finden. Eine davon ist der apriori-algorithmus (wie oben Beschrieben). Hier gibt es keine Confidence- und Lift-werte mehr, nur noch einen Support-wert. Gegeben S = `ab(cf)` und T = `acf`, dann ist der Support(`c`) = 2, der Support(`a`) = 2 und der Support(`b`) = 1. Man beachte, dass es sich auch hier lediglich um eine binäre Entscheidung handelt: der Support erhöht sich lediglich um +1, auch wenn ein item mehrmals in der Sequence enthalten ist.

5. Der Algorithmus zum Finden von Large Subsequences in einer Sequence Datenbank folgt den Vorgaben wie oben beschrieben und ist eine iterative Folge von *Joins* und *Prunes* (Abgleich mit dem Schwellenwert). Ein Join zweier sequences, etwa von S = `a` und T = `b` ergibt: `ab`, `ba`, und `(ab)`. Eine sequence `(ba)` geht wegen der Ordnung nicht (siehe oben).

Disziplin 3: Clustering

1. Clustering bezeichnet die Idee, aufgrund eines Ähnlichkeitsmaßes eine Generalisierung von Daten (Punkte, Datenbankeinträge) durchzuführen. Die Generalisierung entspricht hierbei dem Lernen.

2. Wird zuviel oder gar nicht generalisiert (= bei k Daten erhalten wir k Cluster bzw. es wird nur 1 Cluster erstellt), geht der Sinn des Lernens verloren. Eventuelle Zuordnungen neuer Daten wären nur oder gar nicht möglich.

3. Wir haben 2 Verfahren gesehen: k-means und Demographic Clustering.

   (a) k-means ist ein einfaches iteratives Verfahrenm das durch folgende Aspekte gekennzeichnet ist:

   - Gegeben seien n ($\in N$) Datenpunkte im Raum. Wähle ein k ($\in N$). Generiere zufällig k Datenpunkte (sogenannte *Centroide*; aber beachte, dass keine der kreierten Punkte ein Ausreißer ist...). Entscheide, welches Ähnlichkeitsmaß verwendet wird (`k-means`: *Euclidean distance*, `k-median`: *Manhattan distance*).
   - Die Zahl der Cluster ist hier bereits auf k begrenzt. Ein Clustering mit einer höheren oder niedrigeren Anzahl von Clustern erhält man dann, wenn das k entsprechend gewählt wird.

   Idee

Repeat

(1) Weise jeden Datenpunkt dem am nächsten stehenden Centroid zu (durch Anwendung eines Ähnlichkeitsmaßes, etwa *Euklidische Distanz*).

(2) Wenn alle Datenpunkte zugeordnet sind, `adaptiere die Positionen aller Centroide in Abhängigkeit der ihnen zugeordneten Datenpunkte`. Jeder `Centroid` ist dann der Mittelpunkt (means).

Until a) es gibt keine Veränderung bzgl der Zuordnung mehr oder b) die letzte Veränderung war geringer als ein vordefiniertes ϵ (Schwellenwert).

- Die Qualität eines Clusters erhält man, wenn man alle Abstände aller Datenpunkt zum jeweiligen Centroid addiert.
- Die Qualität dieses Clustering-verfahrens hängt aber nicht allein von den Qualitäten der Cluster ab (Grund: sonst würde man ja k = n wählen und hätte ein optimales Ergebnis).

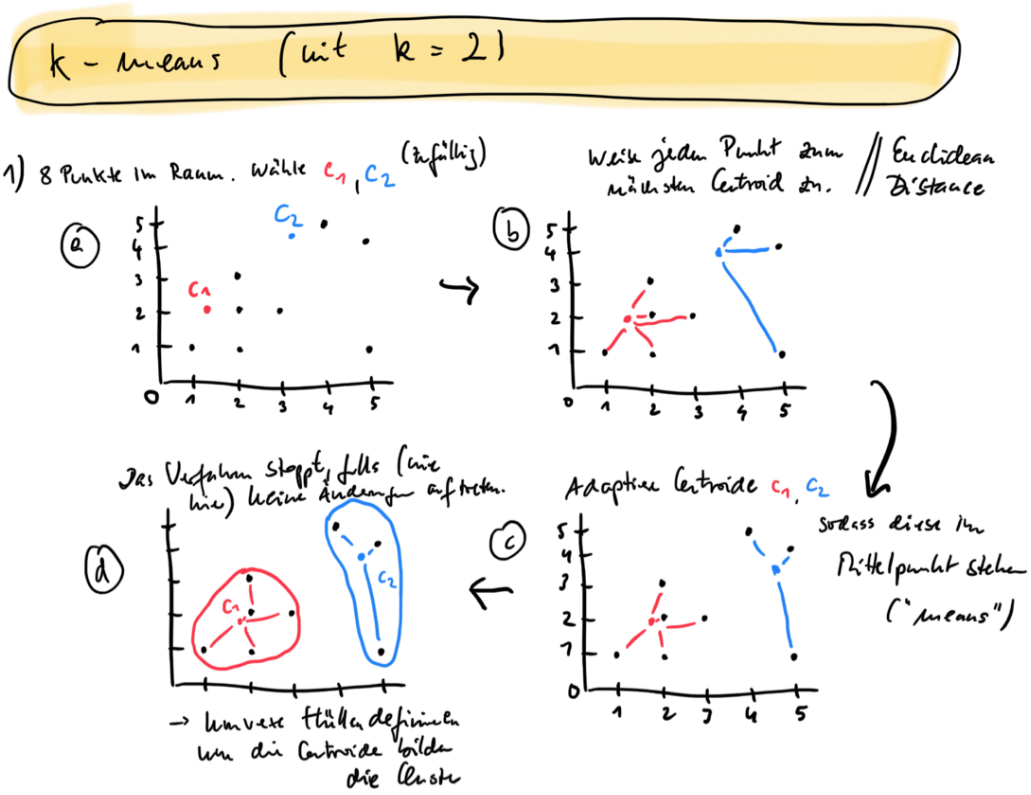


Figure 6: Funktionsweise des kmeans-Algorithmus.

(b) Demographic Clustering

- Bei dem angewendeten Distanzmaß zur Bestimmung einer Ähnlichkeit handelt es

sich hierbei um die `Hamming distance` *H(x,y) = 1 (x und y sind unterschiedlich) oder = 0 (x und y sind gleich).*

- Grundidee des Algorithmus ist das paarweise Vergleichen von Datenpunkten (in der Datenbank) auf Basis der *Hamming Distance* und das Aufstellen einer *Distance* bzw *Similarity Matrix*. Falls sich zwei Datenpunkte in allen, ausgewählten Attributen übereinstimmen, ist die Distanz gleich 0 und die Ähnlichkeit maximal.
- Die Verteilung der Punkte auf die Cluster geschieht mit dem Condorcet-Kriterium (siehe Literatur).
- Die Gesamtqualität des Clustering errechnet sich durch Aufsummieren der Qualitätswerte aller Cluster. Dieser berechnet sich wie folgt: *Summe der berechneten (paarweisen) Ähnlichkeit zwischen allen Clustermitgliedern GETEILT durch die maximale Ähnlichkeit im Cluster.*

  Beispiel: es werden 8 Attribute in das Clustering einbezogen. Im Cluster1 seien nun 3 Datenpunkte enthalten, die je eine Ähnlichkeit zueinander von 5 besitzen. Die maximale Ähnlichkeit zwischen diesen 3 Datenpunkten wäre jeweils 8. Dann ist die Qualität dieses Clusters: $\frac{5+5+5}{8+8+8} = \frac{15}{24} = \frac{3}{8}$.
- Falls ein zweiter Cluster existiert, etwa mit einer Qualit
  "at von $\frac{6}{7}$, dann ist die Gesamtqulität die Summe der beiden Clusterqualitäten geteilt durch 2 (= Anzahl der Cluster): $\frac{9}{14}$.

.

4. Beachten Sie: Ein Clustering ist typischerweise eine 'look-and-see' Angelegenheit. Das bedeutet:

   - Man sollte immer mehrere Clusteringverfahren verwenden und miteinander vergleichen.
   - Der Vergleich sollte nicht nur mit Hilfe der berechneten Qualität erfolgen. Auch bieten einfachere Modelle Anreize (siehe *Occam's razor*), etwa, wenn die Verwendung einer kleinen Attributmenge ähnliche Ergenisse liefert wie die Verwendung einer größeren Attributmenge.

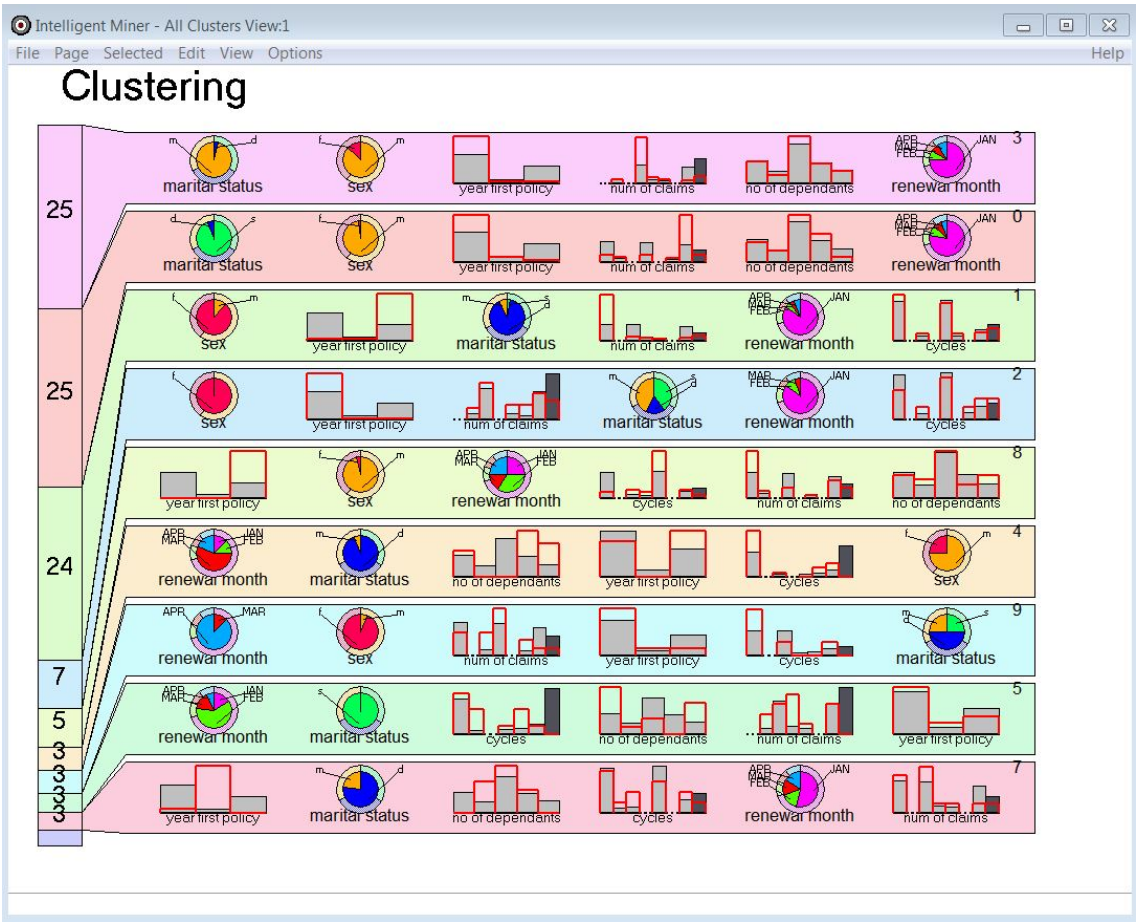


Figure 7: Beispiel: Die Abbildung zeigt die Gesamtansicht von 10 Cluster. Der stärkster Cluster ist Cluster #0 mit 32%, gefolgt von Cluster #0 mit 25% und Cluster #3 mit 24%. Bitte beachten: jeder Cluster beinhaltet 2 Typen von Attributen: numerische und kategorische. Numerische Attribute werden als Histogramme dargestellt, kategorische Attribute als Pie Charts. Weiterhin gilt: rote Umrandungen bei den Histogrammen deutet auf die Verteilung der Clustermitglieder hin, graue Bereiche auf die Gesamtpopulation. Für die Pie Charts gilt: innerer Bereich = Verteilung der Clustermitglieder, äußerer Bereich = Verteilung der Gesamtpopulation. Und: die Reihenfolge der Attribute ist bestimmt durch die relative Ungleichheit zwischen Attributwertverteilung der Datenpunkte innerhalb des Clusters in Bezug zur Gesamtpopulation. Für Cluster 3 (25%) hat das Attribut *marital status* die höchste Ungleichheit und ist damit am Wichtigsten für den Cluster.

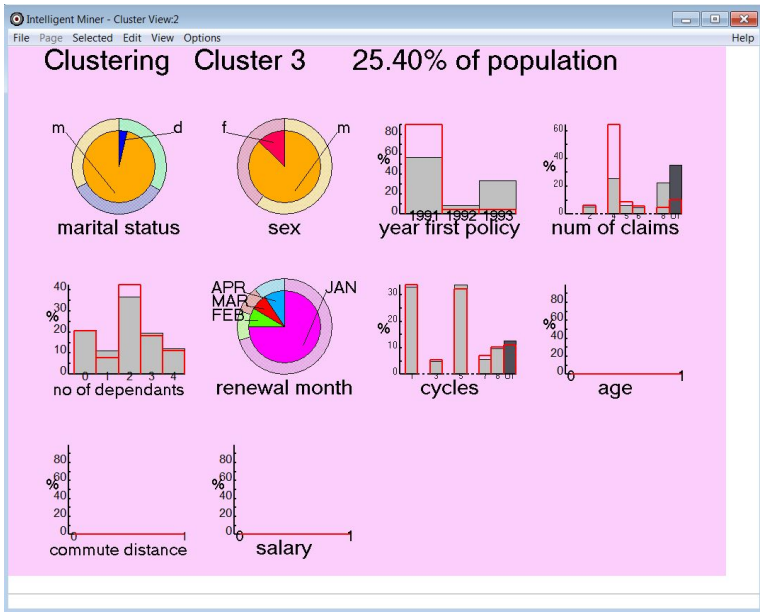


Figure 8: Cluster #3; 25.4% Anteil an Versicherungsteilnehmer. Vor allem verheiratet männliche Personen, die 1991 die erste Versicherungspolice abgeschlossen haben, sind hier vertreten (Anmerkung: Alter, die Distanz von Arbeitsplatz zum Büro, und das Gehalt werden hier nicht verwendet).

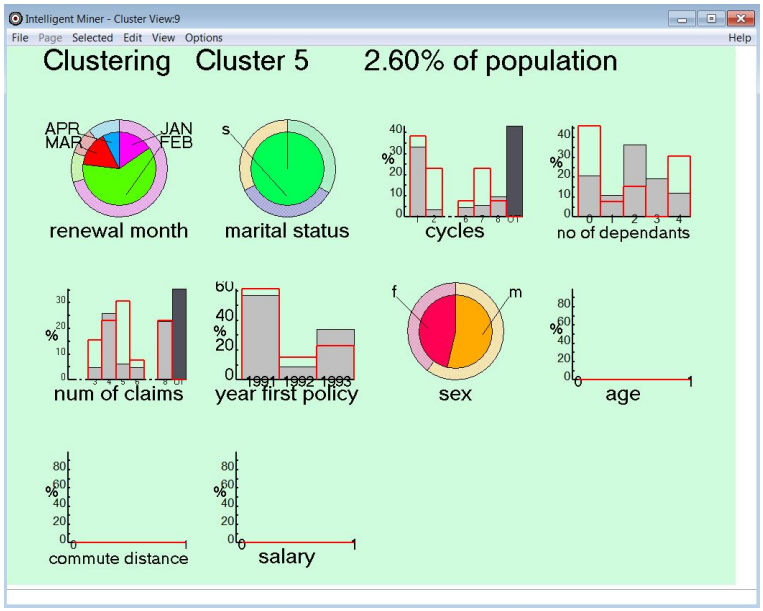


Figure 9: Cluster #5; 2.6% Anteil an Versicherungsteilnehmer. Vor allem Singles, die ihre Versicherungspolice vor allem im Februar erneuern, sind hier vertreten (Anmerkung: Alter, die Distanz von Arbeitsplatz zum Büro, und das Gehalt werden hier nicht verwendet).