LEHRSTUHL 14
SOFTWARE ENGINEERING

Vorlesung

Methodische Grundlagen des Software-Engineering

im Sommersemester 2013

Prof. Dr. Jan Jürjens

TU Dortmund, Fakultät Informatik, Lehrstuhl XIV

Teil 2.3: Data-Mining

v. 11.06.2013

2.3 Data-Mining

[mit freundlicher Genehmigung basierend auf einem englischen Foliensatz von Prof. Dr. Wil van der Aalst (TU Eindhoven)]

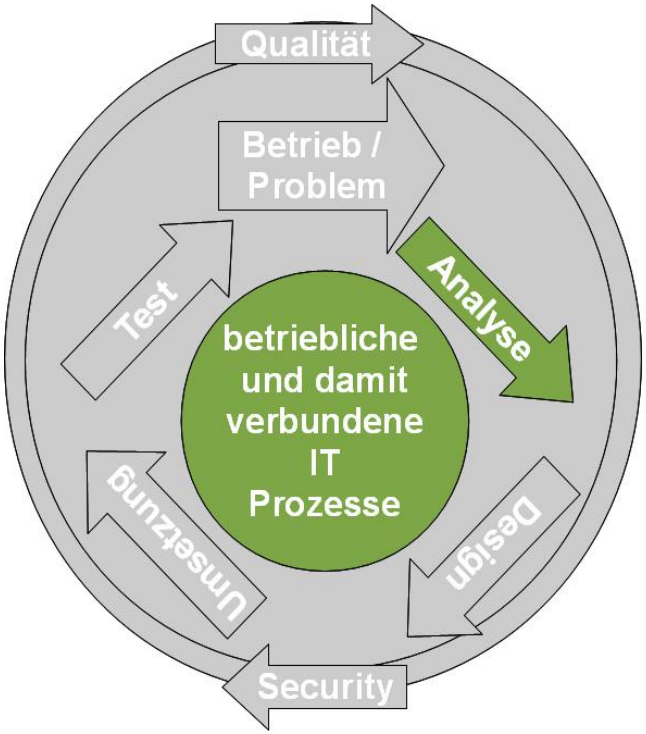
Literatur:

[vdA11] Wil van der Aalst: **Process Mining: Discovery, Conformance and Enhancement of Business Processes**, Springer-Verlag. 2011.
Unibibliothek (6 Exemplare): http://www.ub.tu-dortmund.de/katalog/titel/1332248
(Bei Engpässen kann eine **Kopiervorlage** der relevanten Ausschnitte zur Verfügung gestellt werden.)

- **Kapitel 3**

Einordnung
2.3: Data-Mining

- Geschäftsprozessmodellierung
- Process-Mining
  - Einführung: Process-Mining
  - Petrinetze
  - Prozessmodellierung und Analyse
  - Data-Mining
  - Datenbeschaffung
  - Prozessextraktion
  - Konformanzanalyse
  - Mining: Zusätzliche Perspektiven
  - Betriebsunterstützung
  - Werkzeugunterstützung
  - Analysiere „Lasagne Prozesse“
  - Analysiere „Spaghetti Prozesse“
  - Kartographie und Navigation
  - Epilog
- Modellbasierte Softwareentwicklung
- Modellbasierte Entwicklung sicherer Software



Data-Mining vs. Process-Mining

Process-Mining: Data-Mining auf **Prozess-Daten.**

Daher in diesem Abschnitt kurzer Hintergrund zu Data-Mining.

Vergleich Data-Mining vs. Process-Mining:

- **Process-Mining**: Ende-zu-Ende Prozesse.
- **Data-Mining**: **Datenbasiert** und nicht prozessbasiert.
- **Qualität** von Data-Mining und Process-Mining bewerten: viele Ähnlichkeiten, aber auch Unterschiede.
- Process-Mining-Techniken können Vorteile aus Erfahrungen im Bereich des Data-Mining ziehen.

Datensatz 1

drinker	smoker	weight	age
yes	yes	120	44
no	no	70	96
yes	no	72	88
yes	yes	55	52
no	yes	94	56
no	no	62	93
...	...		

Daten über 860 kürzlich verstorbene Personen.
Studie über Auswirkung von Alkoholkonsum, Rauchen und Körpergewicht auf die Lebenserwartung.

Fragen:
- Welchen Einfluss hat Rauchen und Trinken auf das Körpergewicht?
- Trinken Leute, die rauchen, auch?
- Welche Faktoren beeinflussen die Lebenserwartung am meisten?
- Kann man Gruppen mit ähnlichem Lebensstil identifizieren?

Datensatz 2

linear algebra	logic	program-ming	operations research	workflow systems	...	duration	result
9	8	8	9	9	...	36	cum laude
7	6	-	8	8	...	42	passed
-	-	5	4	6	...	54	failed
8	6	6	6	5	...	38	passed
6	7	6	-	8	...	39	passed
9	9	9	9	8		38	cum laude
5	5	-	6				
...	...	...	...				

Daten über 420 Studenten, um Verbindungen zwischen Kursnoten und Gesamtleistung im Bachelor zu untersuchen.

Fragen:
- Haben die Noten bestimmter Kurse eine hohe Korrelation?
- Welche Wahl treffen Studenten mit Auszeichnung (cum laude)?
- Welche Kurse verzögern den Abschluss signifikant?
- Warum brechen Studenten ab?
- Kann man Gruppen mit ähnlichem Lernverhalten identifizieren?

Datensatz 3

cappuccino	latte	espresso	americano	ristretto	tea	muffin	bagel
1	0	0	0	0	0	1	0
0	2	0	0	0	0	1	1
0	0	1	0	0	0	0	0
1	0	0	0	0	0	0	0
0	0	0	0	0	1	2	0
0	0	0	1	1	0	0	0
...	...	...	...	...	...	...	...

Daten über 240 Bestellungen in einem Café, aufgenommen von der Kasse.

Fragen:
- Welche Produkte werden häufig zusammen gekauft?
- Wann kaufen Kunden bestimmte Produkte?
- Ist es möglich, Gruppen typischer Kunden zu charakterisieren?
- Wie fördert man den Verkauf von Produkten mit hoher Gewinnmarge?

Variablen

- Datensatz besteht aus **Instanzen** (Individuen, Entitäten, Fälle, Objekte oder Aufzeichnungen).
- **Variablen**: oft als Attribute, Features oder Datenelemente bezeichnet.

Zwei Typen:

- Kategorielle Variablen:
  - **Ordinal** (hoch-mittel-niedrig, mit Auszeichnung-bestanden-durchgefallen) oder
  - **Nominal** (true-false, rot-pink-grün)
- Numerische Variablen (geordnet, können nicht einfach aufgezählt werden)

Überwachtes Lernen

- **Klassifizierte Daten** (Labeled Data): Jede Instanz durch Response-Variable gekennzeichnet.
- **Ziel**: Erkläre Response-Variable (abhängige Variable) in Form von Predictor-Variablen (unabhängige Variable).
- Klassifikationstechniken (z.B.: Lernen mit Entscheidungsbäumen): Setzen kategorielle Response-Variablen voraus.
  Ziel: Instanzen anhand Predictor-Variablen klassifizieren.
- Regressionstechniken: Benötigen numerische Response-Variablen.
  Ziel: Funktion finden, welche zu Daten passt, mit wenigsten Fehlern.

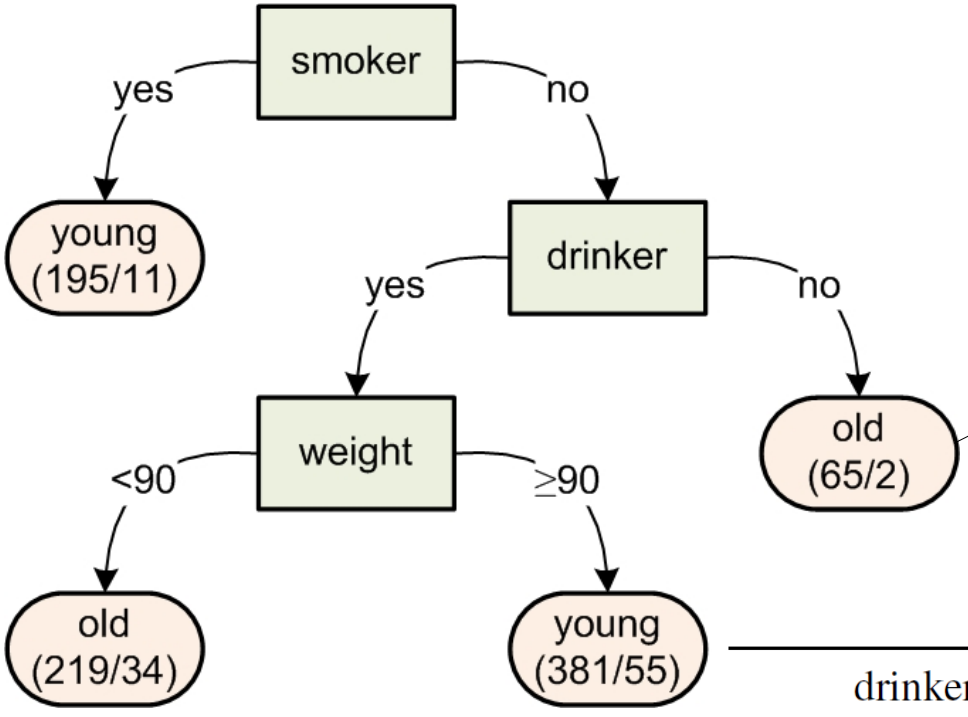
Nicht überwachtes Lernen

- **Nicht überwachtes Lernen** verwendet unlabeled data. Variablen also nicht in Response- und Predictor-Variablen unterteilt.

Beispiele:

- Clustering (z.B.: k-means clustering und agglomerative hierarchical clustering)
- Pattern discovery (association rules)

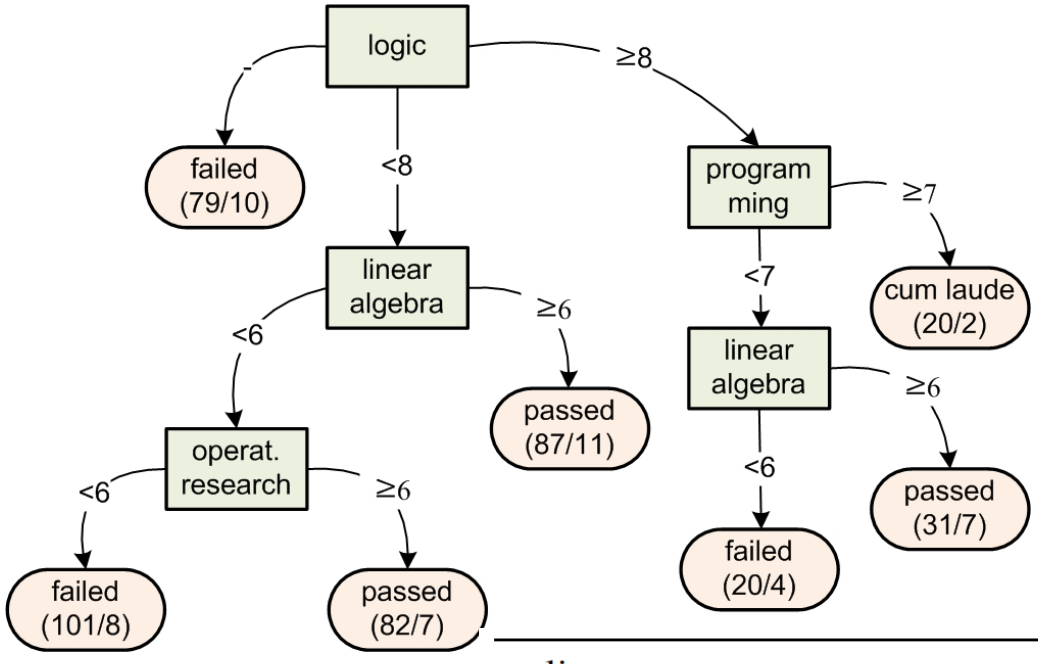
Entscheidungsbaum: Datensatz 1



65 als „old“ vorhergesagt, davon 2 inkorrekt.

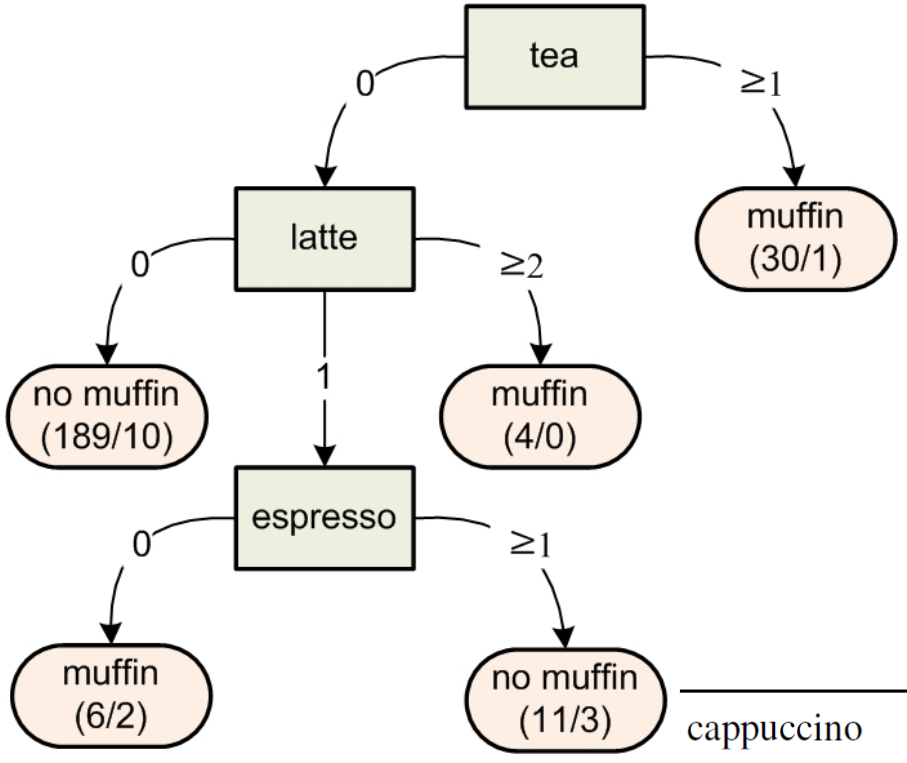
drinker	smoker	weight	age
yes	yes	120	44
no	no	70	96
yes	no	72	88
yes	yes	55	52
no	yes	94	56
no	no	62	93
...	...	...	...

Entscheidungsbaum: Datensatz 2



linear algebra	logic	program-ming	operations research	workflow systems	...	duration	result
9	8	8	9	9	...	36	cum laude
7	6	-	8	8	...	42	passed
-	-	5	4	6	...	54	failed
8	6	6	6	5	...	38	passed
6	7	6	-	8	...	39	passed
9	9	9	9	8	...	38	cum laude
5	5	-	6	6	...	52	failed
...	...	...	...	...	...	...	...

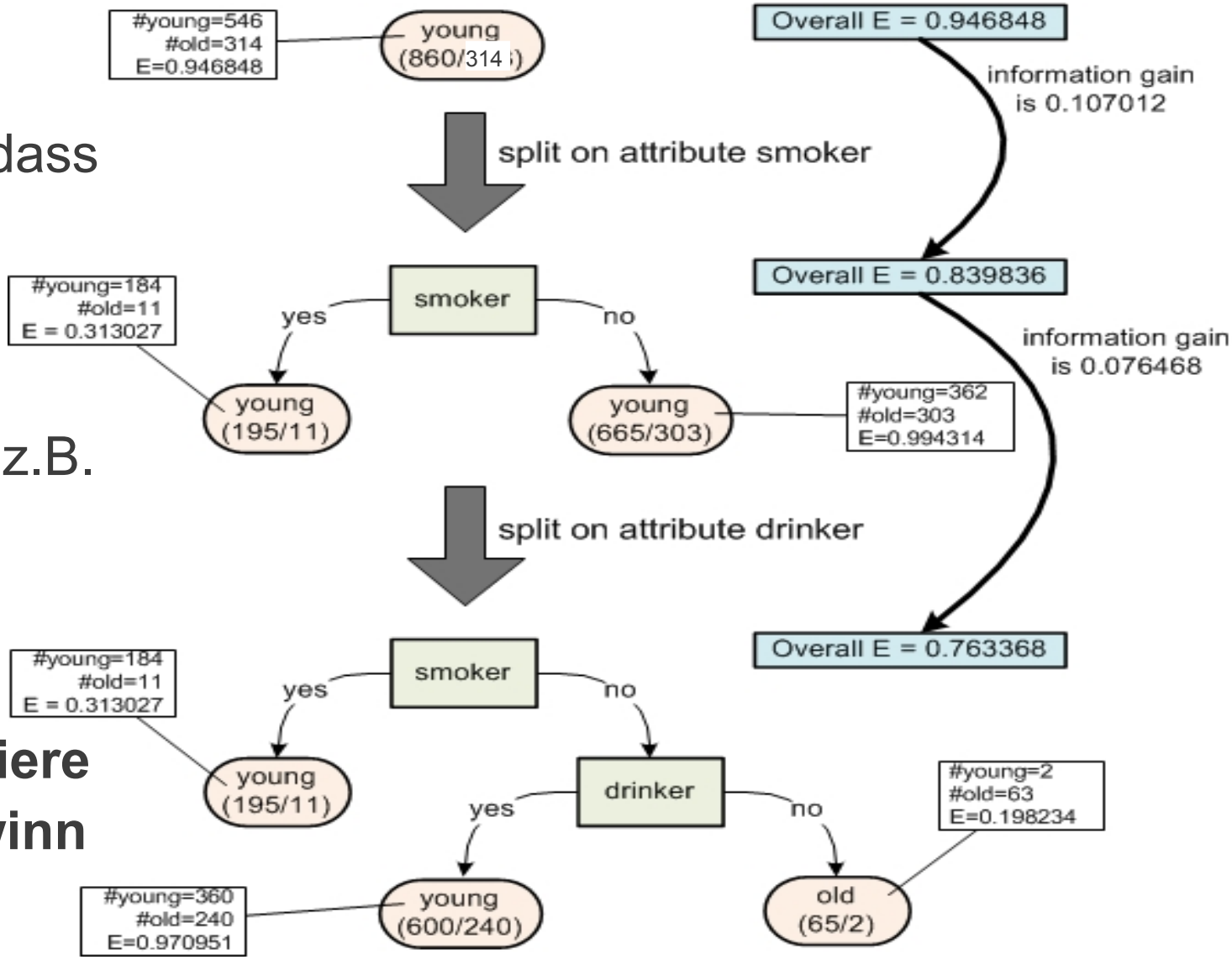
Entscheidungsbaum: Datensatz 3



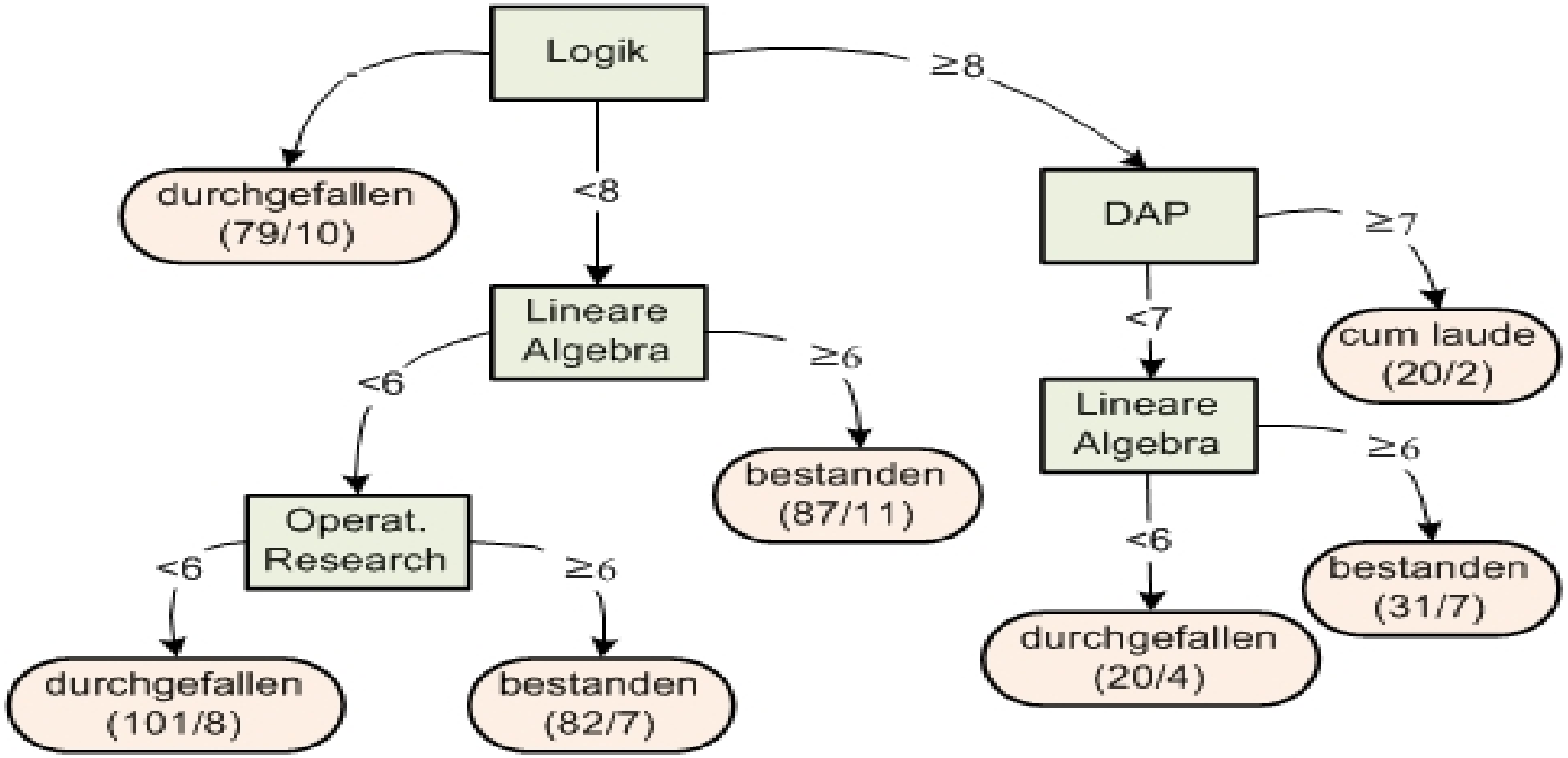
cappuccino	latte	espresso	americano	ristretto	tea	muffin	bagel
1	0	0	0	0	0	1	0
0	2	0	0	0	0	1	1
0	0	1	0	0	0	0	0
1	0	0	0	0	0	0	0
0	0	0	0	0	1	2	0
0	0	0	1	1	0	0	0
...	...	...	...	...	...	...	...

Grundidee

- Teile Menge der Instanzen auf, sodass **Variation** in jeder Teilmenge **kleiner** wird.
- Variation messen z.B. durch **Entropie**.
- **Minimiere** durchschnittliche **Entropie**; **maximiere Informationsgewinn** pro Schritt.



Konfusionsmatrix



		vorhergesagte Bewertung		
		durchge-fallen	bestanden	cum laude
reale Bewertung	durchge-fallen	178	22	0
	bestanden	21	175	2
	cum laude	1	3	18

Konfusionsmatrix: Metriken

		vorhergesagte Klasse		
		+	−	
reale Klasse	+	*tp*	*fn*	*p*
	−	*fp*	*tn*	*n*
		p'	*n'*	*N*

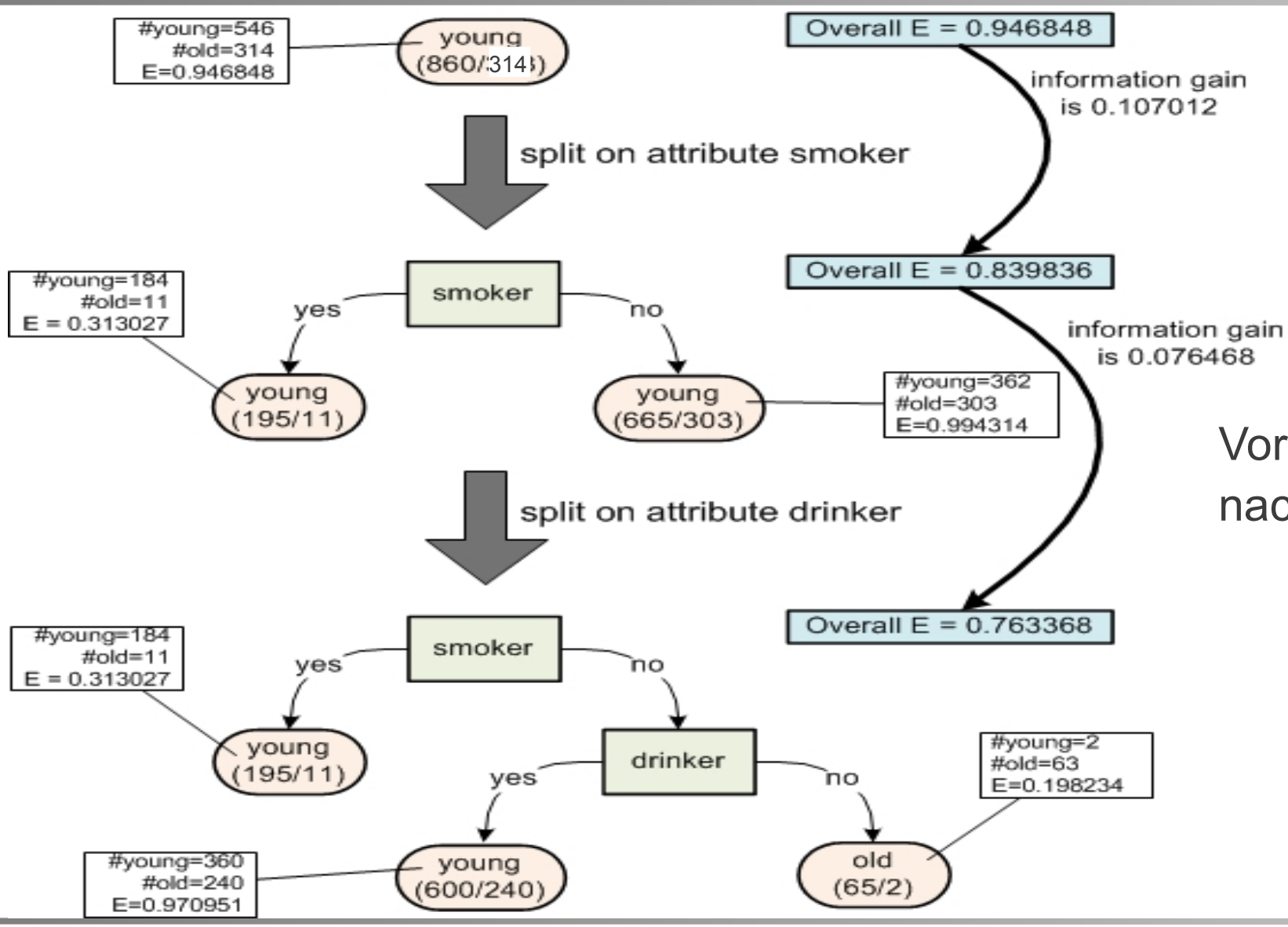
(a)

Name	Formel
Error	*(fp+fn)/N*
accuracy	*(tp+tn)/N*
tp-Rate	*tp/p*
fp-Rate	*fp/n*
precision	*tp/p'*
recall	*tp/p*

(b)

tp: Anzahl true positives; korrekterweise als positiv klassifiziert.
fn: Anzahl false negatives; als negativ klassifiziert, aber positiv.
fp: Anzahl false positives; als positiv klassifiziert, aber negativ.
tn: Anzahl true negatives; korrekterweise als negativ klassifiziert.

Konfusionsmatrix: Beispiel

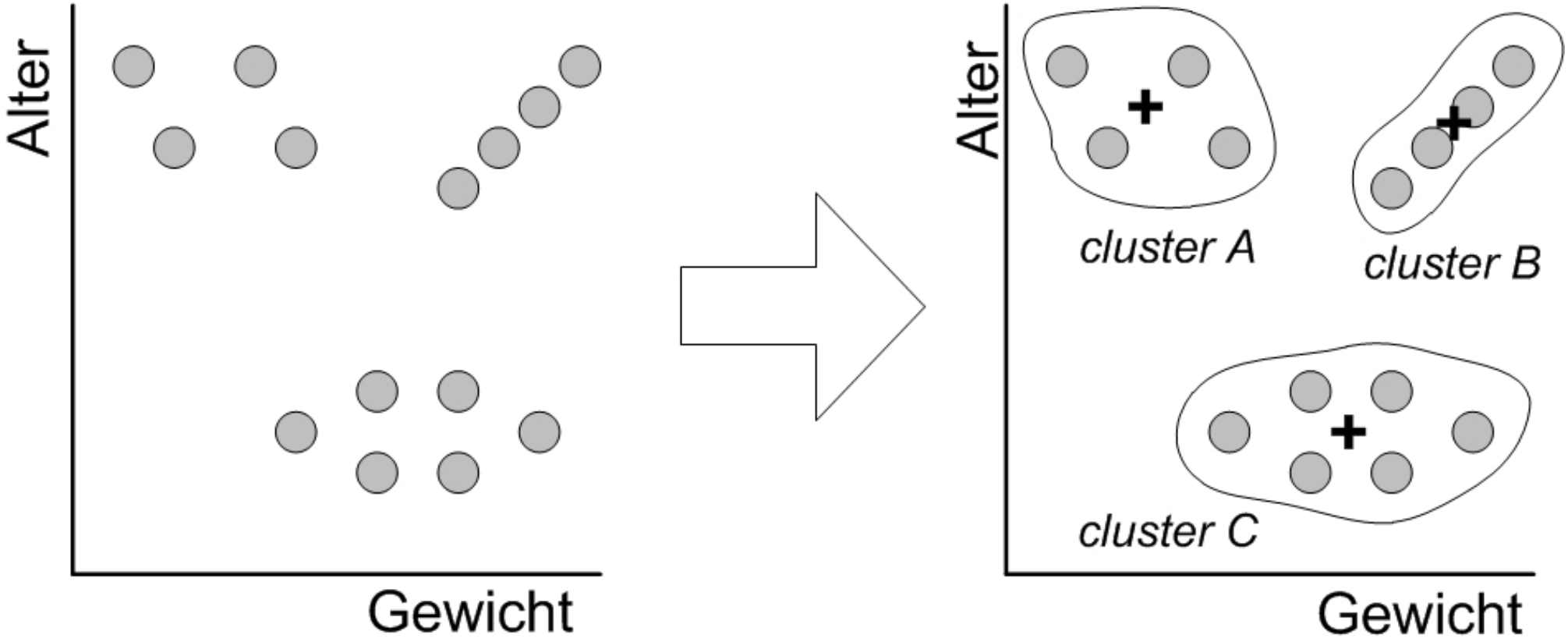


		predicted class	
		young	old
actual class	young	546	0
	old	314	0

Vorhersage vor bzw. nach Aufsplitten.

		predicted class	
		young	old
actual class	young	544	2
	old	251	63

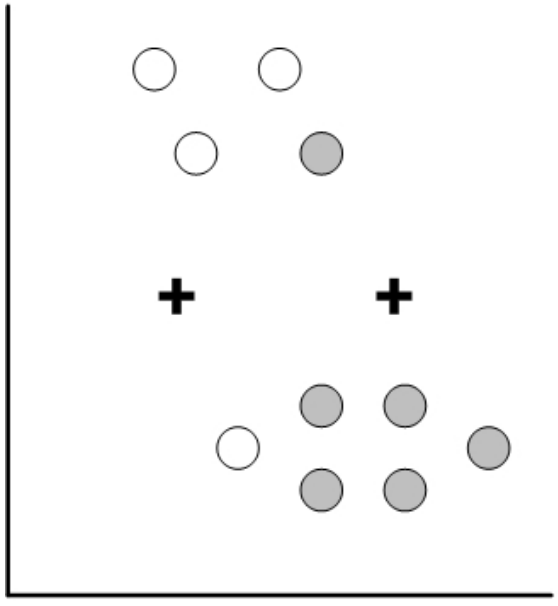
Cluster-Analyse



+: Schwerpunkt eines Clusters

k-Means-Cluster-Analyse

k-Means-Clustering: Anzahl *k* von Clustern a-priori festgelegt.



(a)

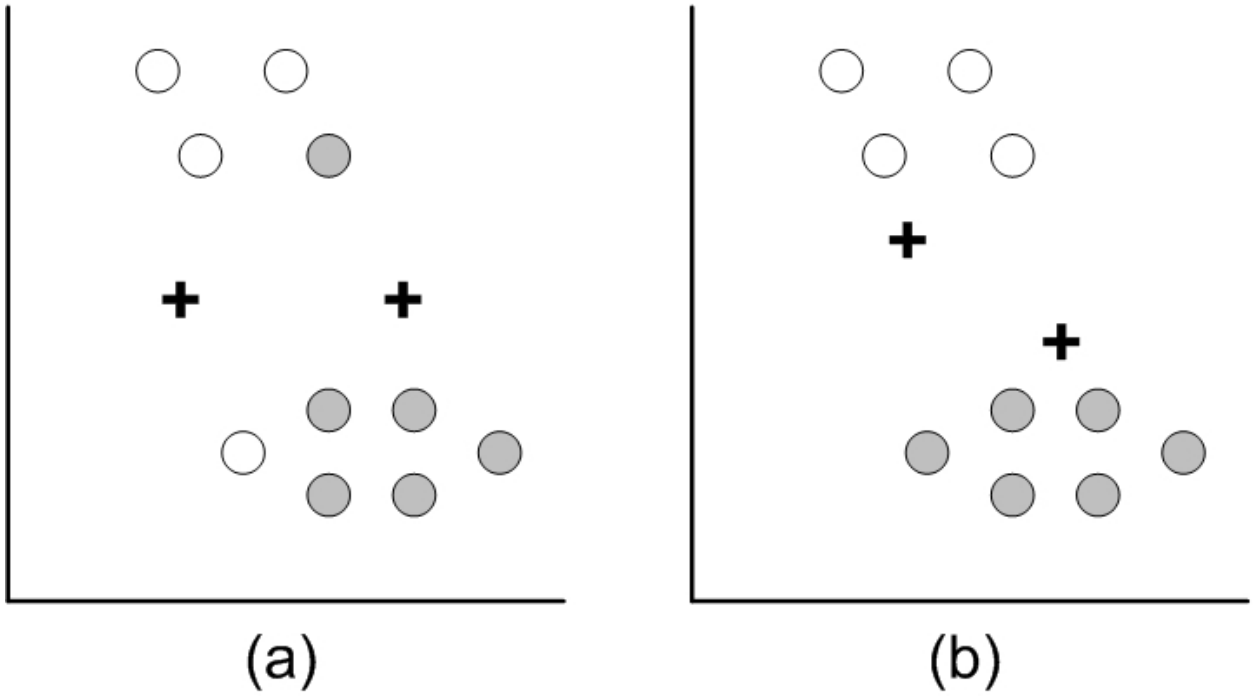
K-Means-Algorithmus veranschaulicht für *k*=2:

(a) „Schwerpunkte“ zufällig setzen; Punkte nächstgelegenen Schwerpunkten zuordnen.

(b) Schwerpunkte neu berechnen und Punkte neu zuordnen.

k-Means-Cluster-Analyse

k-Means-Clustering: Anzahl *k* von Clustern a-priori festgelegt.



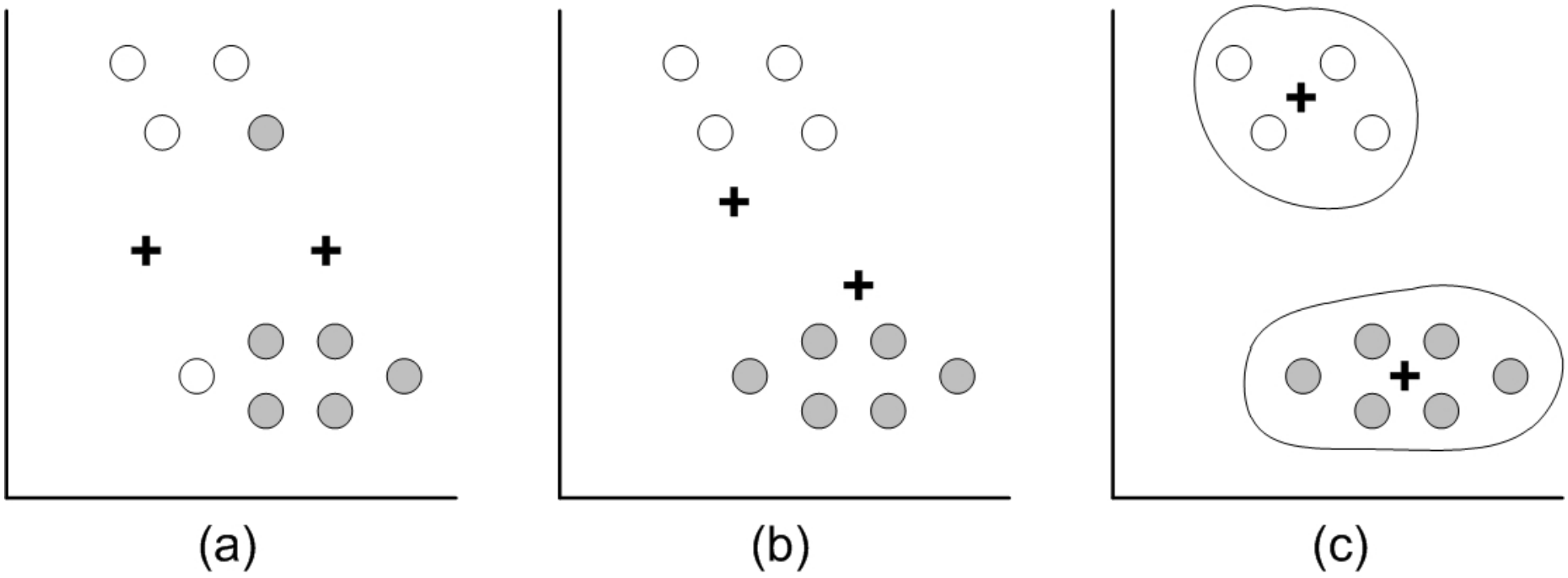
K-Means-Algorithmus veranschaulicht für *k*=2:

(a) „Schwerpunkte“ zufällig setzen; Punkte nächstgelegenen Schwerpunkten zuordnen.

(b) Schwerpunkte neu berechnen und Punkte neu zuordnen.

k-Means-Cluster-Analyse

k-Means-Clustering: Anzahl *k* von Clustern a-priori festgelegt.



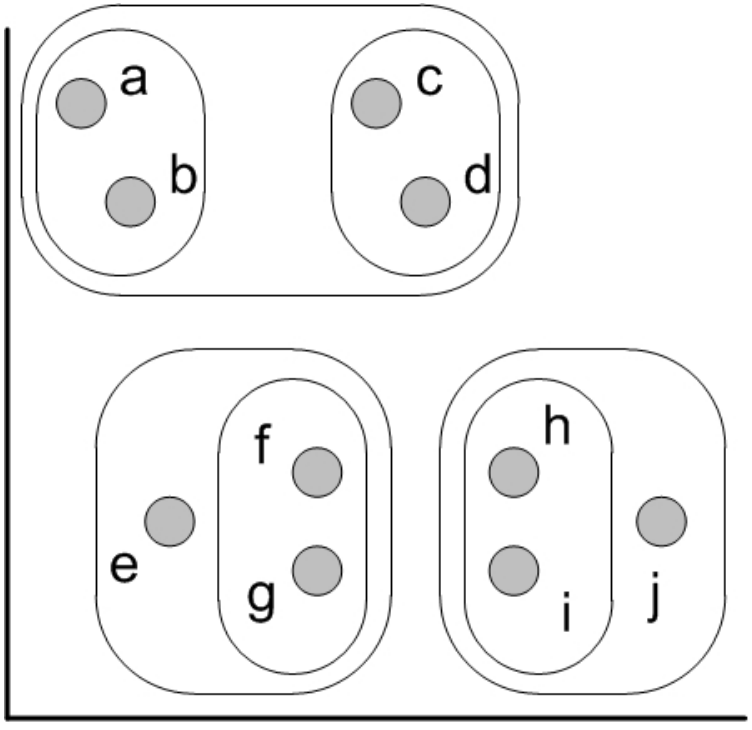
K-Means-Algorithmus veranschaulicht für *k*=2:

(a) „Schwerpunkte“ zufällig setzen; Punkte nächstgelegenen Schwerpunkten zuordnen.

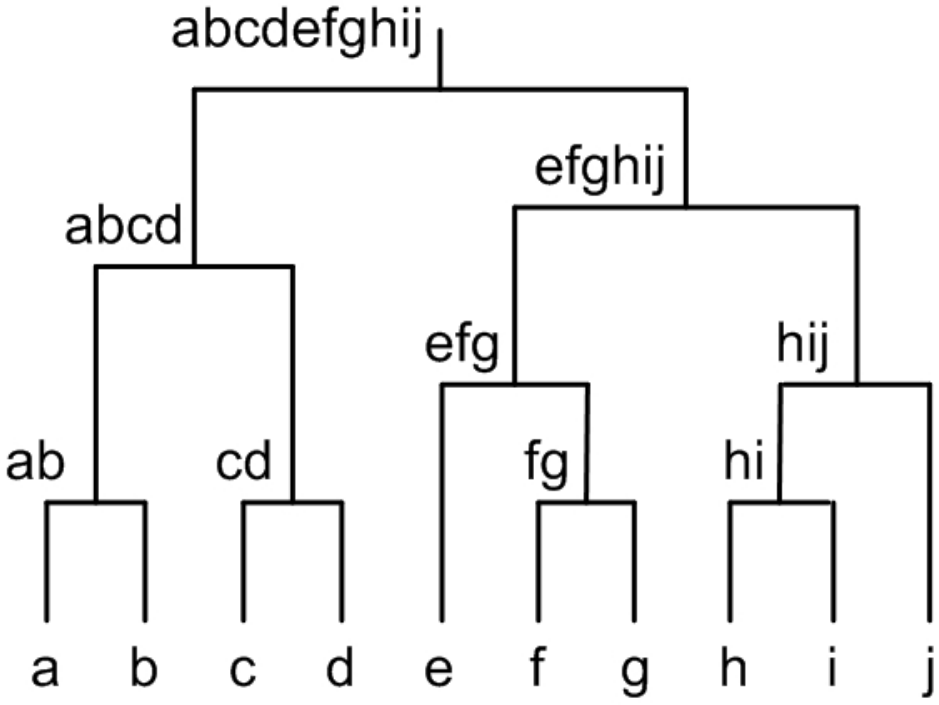
(b) Schwerpunkte neu berechnen und Punkte neu zuordnen.

(c) Fixpunkt erreicht.

Agglomerative Hierarchische Cluster-Analyse



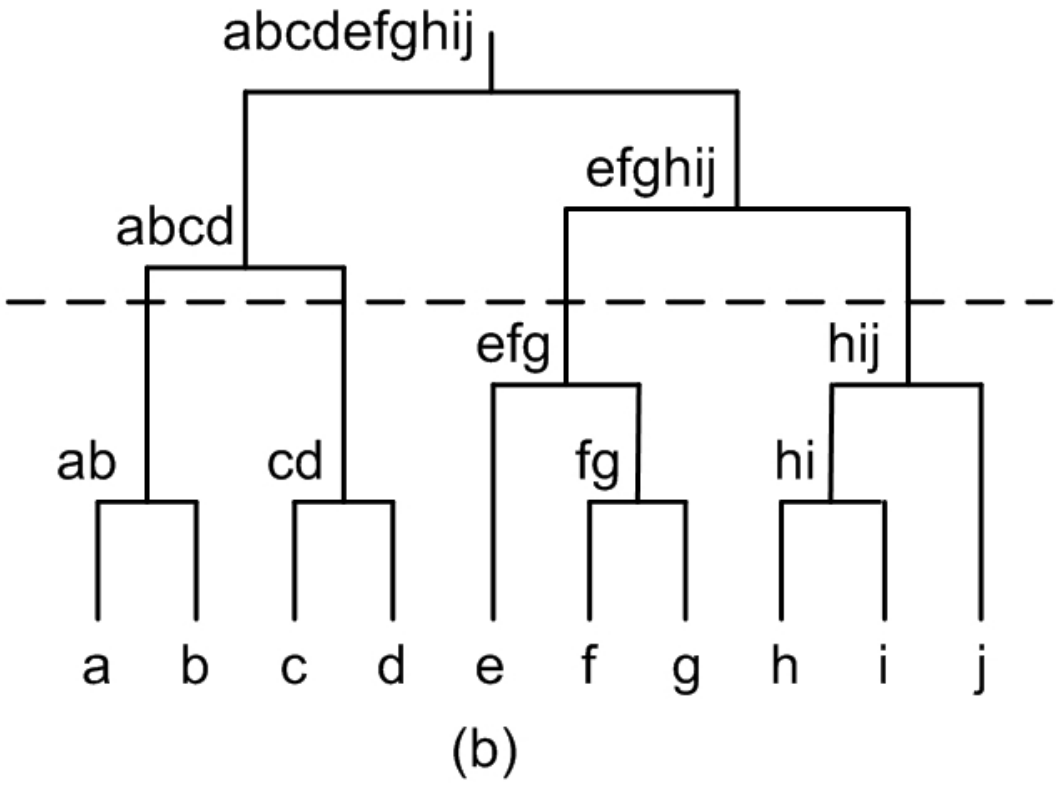
(a)



(b)

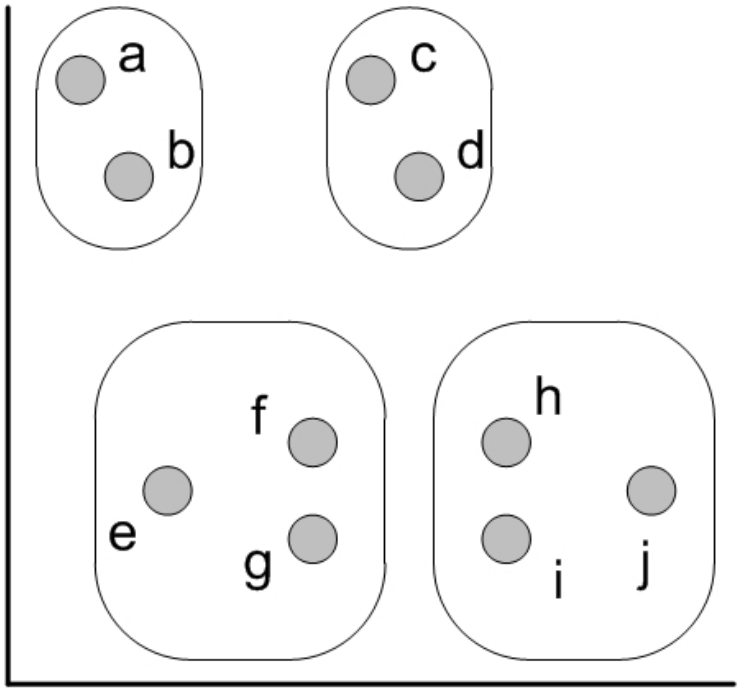
(a) Agglomerative hierarchische Cluster-Analyse: **Bottom-up**-Vorgehen.
(b) Visualisierung durch Dendrogramm.

Ebenen bei Agglomerativer Hierarchischer Cluster-Analyse

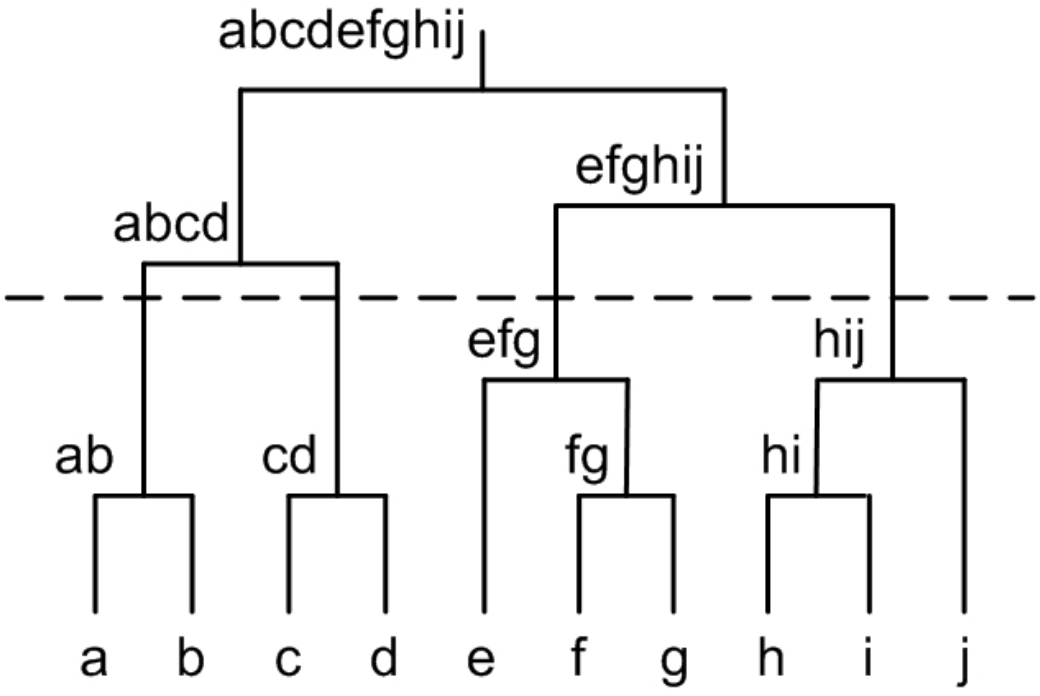


Horizontale Linie in Dendrogramm korrespondiert zu konkretem Clustering auf bestimmter **Abstraktionsebene**.

Ebenen bei Agglomerativer Hierarchischer Cluster-Analyse



(a)



(b)

Horizontale Linie in Dendrogramm korrespondiert zu konkretem Clustering auf bestimmter **Abstraktionsebene**.

Assoziationsregel-Lernen

Ziel: Regeln der Form “IF X THEN Y” lernen: $X \Rightarrow Y$.

Definere Maße für Relevanz / Gültigkeit / Aussagekraft der Regel:

$$support(X \Rightarrow Y) = N_{X \wedge Y} / N$$

$$confidence(X \Rightarrow Y) = N_{X \wedge Y} / N_X$$

$$lift(X \Rightarrow Y) = \frac{N_{X \wedge Y}/N}{(N_X/N)\ (N_Y/N)} = \frac{N_{X \wedge Y}\ N}{N_X\ N_Y}$$

(wobei N_X die Anzahl der Datensätze, die alle Eigenschaften in X erfüllen).

Welche Bedeutung könnten hohe / niedrige Werte haben ?

Assoziationsregel-Lernen: Maße und Bedeutung

$$support(X \Rightarrow Y) = N_{X \wedge Y} / N$$

$$confidence(X \Rightarrow Y) = N_{X \wedge Y} / N_X$$

$$lift(X \Rightarrow Y) = \frac{N_{X \wedge Y}/N}{(N_X/N)\ (N_Y/N)} = \frac{N_{X \wedge Y}\ N}{N_X\ N_Y}$$

- **Support**: So hoch wie möglich (ist aber oft niedrig). Niedriger Support: relativ geringe Fallzahl, evt. nur „zufällig“ geltende Regel.
- **Confidence**: Immer <=1 (da $X \wedge Y \Rightarrow X$). Möglichst nahe an 1.
- **Lift**:
  - **Hoch**: positive Korrelation
  - **Niedrig**: negative Korrelation
  - **Nahe 1**: Unabhängigkeit (bei probabilistisch unabhängigen Ereignissen X, Y gilt $p(X \wedge Y)=p(X)*p(Y)$).

Beispiel: Einkaufswagen-Analyse

cappuccino	latte	espresso	americano	ristretto	tea	muffin	bagel
1	0	0	0	0	0	1	0
0	2	0	0	0	0	1	1
0	0	1	0	0	0	0	0
1	0	0	0	0	0	0	0
0	0	0	0	0	1	2	0
0	0	0	1	1	0	0	0
...	...	...	...	...	...	...	...

Zwei Beispiel-Hypothesen:

$$tea \wedge latte \Rightarrow muffin$$

=> Leute, die tea oder latte bestellen, bestellen auch muffins

$$tea \Rightarrow muffin \wedge bagel$$

Beispiel: tea ∧ latte => muffin

$tea \wedge latte \Rightarrow muffin$, d.h.: $X = tea \wedge latte$ und $Y = muffin$

$$support(X \Rightarrow Y) = N_{X \wedge Y}/N = N_{tea \wedge latte \wedge muffin}/N = 15/240 = 0.0625$$

$$confidence(X \Rightarrow Y) = N_{X \wedge Y}/N_X = N_{tea \wedge latte \wedge muffin}/N_{tea \wedge latte} = 15/20 = 0.75$$

$$lift(X \Rightarrow Y) = \frac{N_{X \wedge Y}\, N}{N_X\, N_Y} = \frac{N_{tea \wedge latte \wedge muffin}\, N}{N_{tea \wedge latte}\, N_{muffin}} = \frac{15 \times 240}{20 \times 40} = 4.5$$

Brute-force-Algorithmus

Definition: Für gegebenen *support*-Schwellwert *minsup*: Menge Z heißt ***frequent item-set***, wenn $N_z / N >= minsup$.

Kann **Assoziationsregeln** wie folgt **generieren**:

1) Generiere ***frequent item-sets***: alle Mengen Z sodass N_z / N größer als gegebener Schwellwert für *support*.
2) Für jedes *frequent item-set* Z betrachte Partition in nicht-leere Teilmengen X, Y.
   Behalte die Regeln $X \Rightarrow Y$, für die *confidence* gegebenen Schwellwert überschreitet.

Beobachtung

Beobachtung:

Unter welcher Voraussetzung ist eine **Teilmenge** einer ***frequent item-set*** ebenfalls eine ***frequent item-set*** ?

Beobachtung

Beobachtung: Jede nicht-leere **Teilmenge** einer ***frequent item-set*** ist ebenfalls ***frequent***: Durch **Teilmengen**-Bildung kann sich Menge der zu erfüllenden **Eigenschaften** allenfalls **verringern** (nicht vergrößern), daher kann sich der ***support*** allenfalls **vergrößern** (nicht verringern), da es einfacher wird, alle Eigenschaften zu erfüllen.

1. If an item-set is *frequent* (i.e., an item-set with a support above the threshold), then all of its non-empty subsets are also frequent. Formally, for any pair of non-empty item-sets X, Y: if $Y \subseteq X$ and $N_X/N \geq minsup$, then $N_Y/N \geq minsup$.
2. If, for any k, I_k is the set of all frequent item-sets with cardinality k and $I_l = \emptyset$ for some l, then $I_k = \emptyset$ for all $k \geq l$.

=> Kann **maximale *frequent item-sets*** generieren ausgehend von **1-elementigen** *frequent item-sets*.

Optimierte Generierung von *frequent item-sets* mit o.g. Idee

1. Create I_1. This is the set of singleton frequent item-sets, i.e., item-sets with a support above the threshold *minsup* containing just one element.
2. $k := 1$
3. If $I_k = \emptyset$, then output $\bigcup_{i=1}^{k} I_i$ and end. If $I_k \neq \emptyset$, continue with the next step.
4. Create C_{k+1} from I_k. C_{k+1} is the candidate set containing item-sets of cardinality $k+1$. Note that one only needs to consider elements that are the union of two item-sets A and B in I_k such that $|A \cap B| =$ k-1 and $|A \cup B| = k+1$. (wg. o.g. Beobachtung)
5. For each candidate frequent item-set $c \in C_{k+1}$: examine all subsets of c with k elements; delete c from C_{k+1} if any of the subsets is not a member of I_k. (wg. o.g. Beobachtung)
6. For each item-set c in the pruned candidate frequent item-set C_{k+1}, check whether c is indeed frequent. If so, add c to I_{k+1}. Otherwise, discard c.
7. $k := k+1$ and return to Step 3.

Sequence-Mining

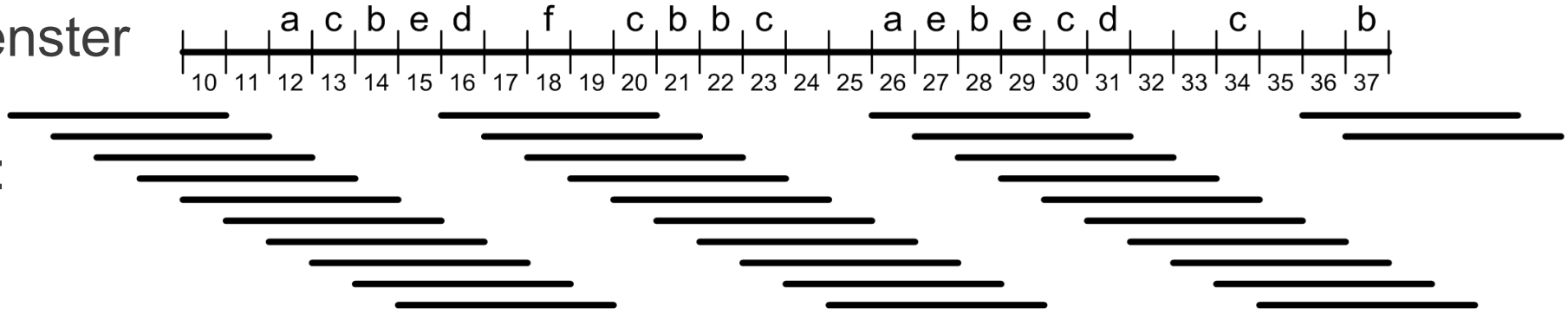
customer	seq. number	timestamp	items
Wil	1	02-01-2011:09.02	{*cappuccino*}
	2	03-01-2011:10.06	{*espresso,muffin*}
	3	05-01-2011:15.12	{*americano,cappuccino*}
	4	06-01-2011:11.18	{*espresso,muffin*}
	5	07-01-2011:14.24	{*cappuccino*}
	6	07-01-2011:14.24	{*americano,cappuccino*}
Mary	1	30-12-2010:11.32	{*tea*}
	2	30-12-2010:12.12	{*cappuccino*}
	3	30-12-2010:14.16	{*espresso,muffin*}
	4	05-01-2011:11.22	{*bagel,tea*}
Bill	1	30-12-2010:14.32	{*cappuccino*}
	2	30-12-2010:15.06	{*cappuccino*}
	3	30-12-2010:16.34	{*bagel,espresso,muffin*}
	4	06-01-2011:09.18	{*ristretto*}
	5	06-01-2011:12.18	{*cappuccino*}
...	...	...	...

$$X = \langle \{cappuccino\}, \{espresso\} \rangle$$

$$Y = \langle \{cappuccino\}, \{espresso\}, \{latte, muffin\} \rangle$$

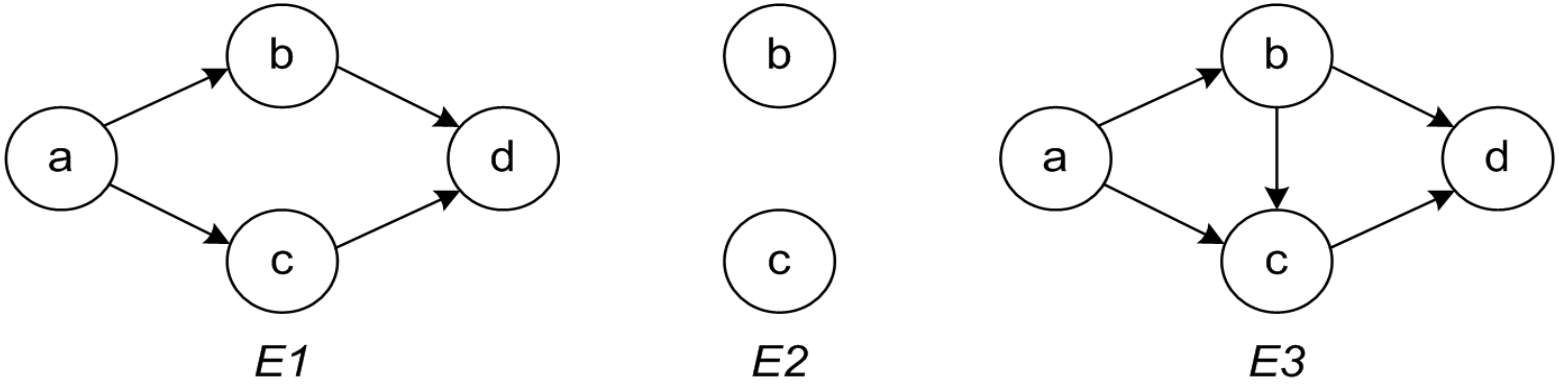
Episode-Mining

32 Zeitfenster der Länge 5:



„**Episode**“: Prozessmodell-Fragment.

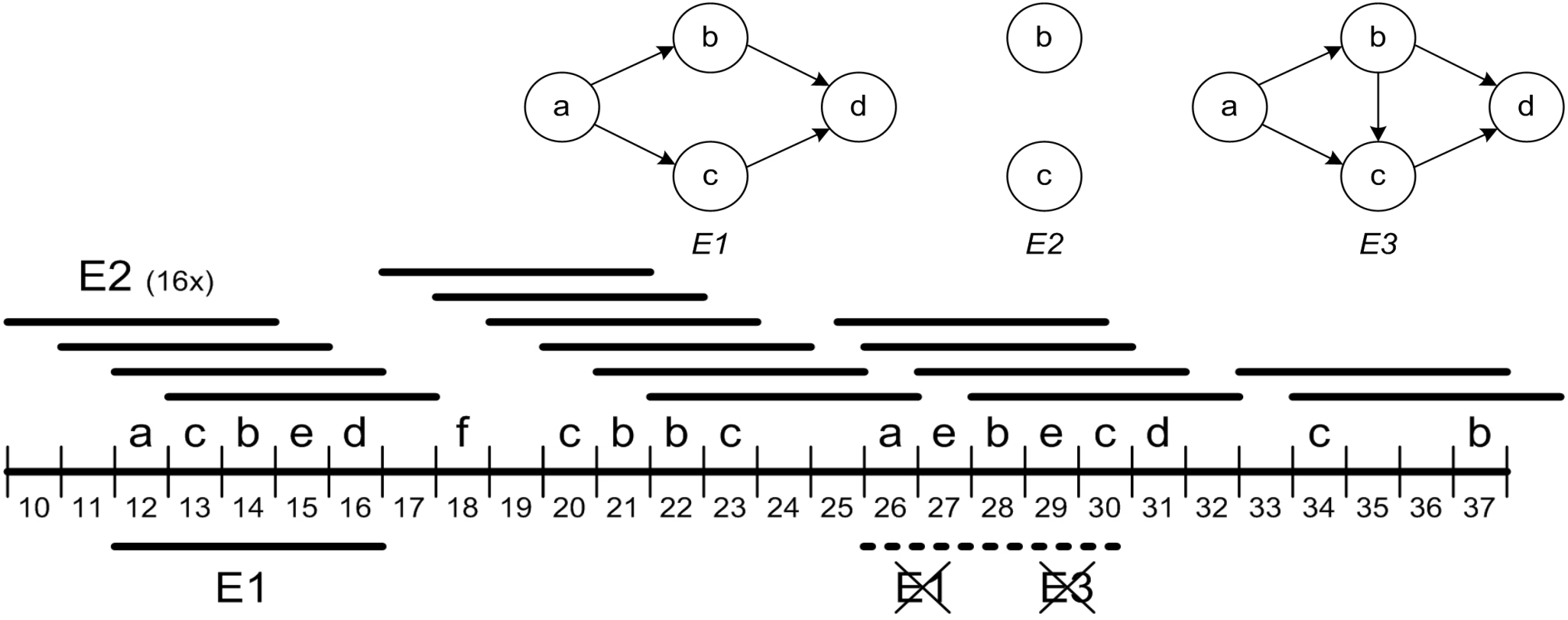
Drei Beispiele:



(weitere sind möglich, z.B. unter Verwendung von e,f).

Wie oft treten diese Episoden hier auf (d.h. wie oft gibt es vollständigen Ablauf eines Modells in Zeitfenster der Länge 5) ?

Auftritte von Episoden



16 Auftritte von E2 (b, c in beliebiger Reihenfolge).

Davon 1 Auftritt von E1: *E2 => E1* hat somit *confidence* 1/16.

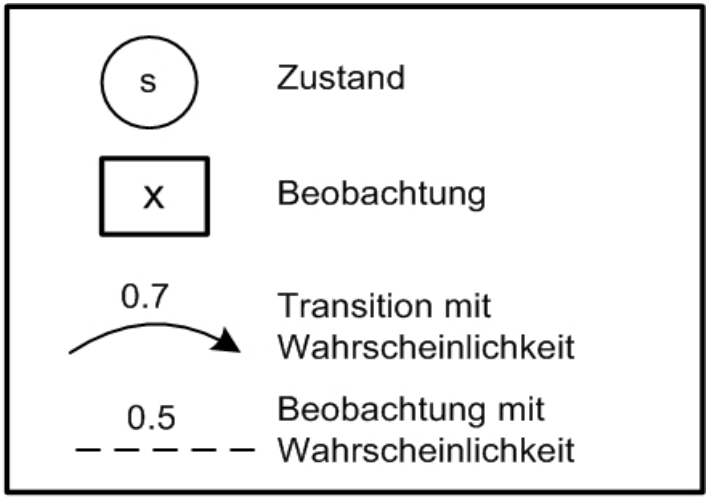
Weitere Auftritte von E1 und E3 nur bei Fenstergröße 6.

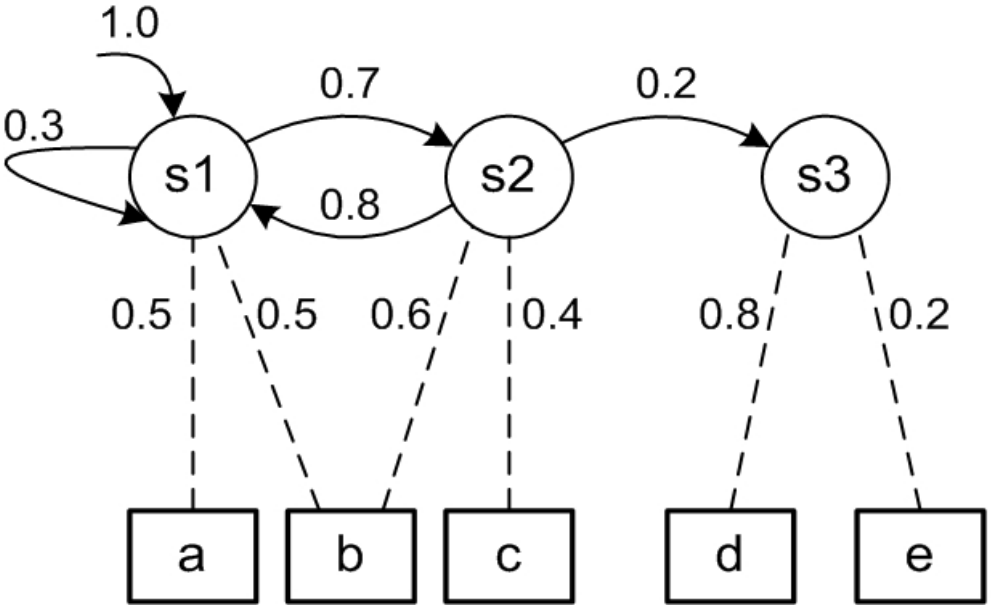
Hidden-Markov-Modell

1) **Wahrscheinlichkeit** der Sequenz berechnen (gegeben Beobachtungs-sequenz und Hidden-Markov-Modell).

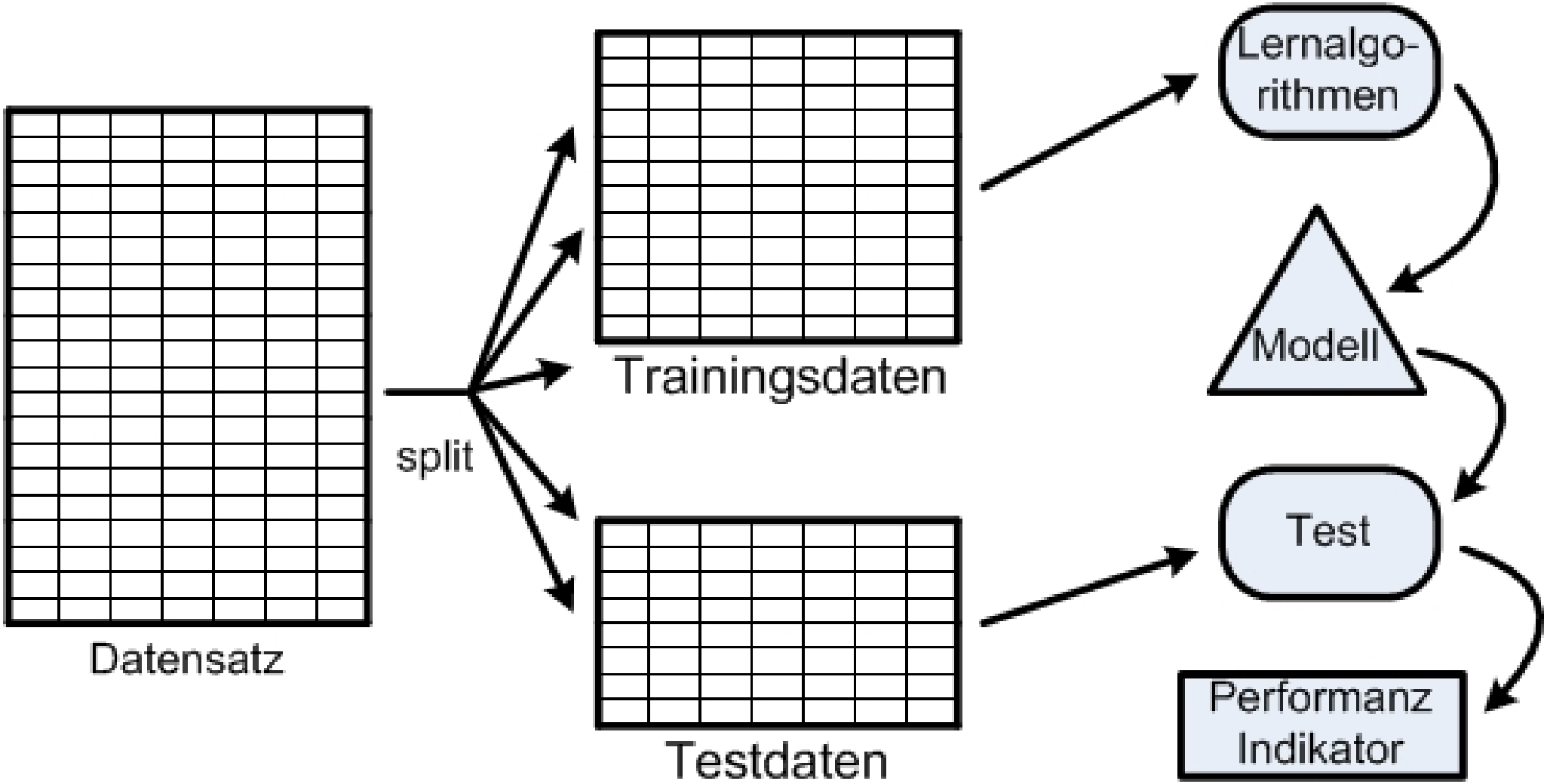
2) Gegeben Beobachtungssequenz und Hidden-Markov-Modell, wahrscheinlichsten „**hidden path**“ im Modell berechnen (= interne Zustandsfolge; kann zur z.T. beobachtet werden).

3) Bei gegebener Beobachtungs-sequenz **Hidden-Markov-Modell** ableiten, welches mit maximaler Wahrscheinlichkeit die Sequenzen erzeugt.

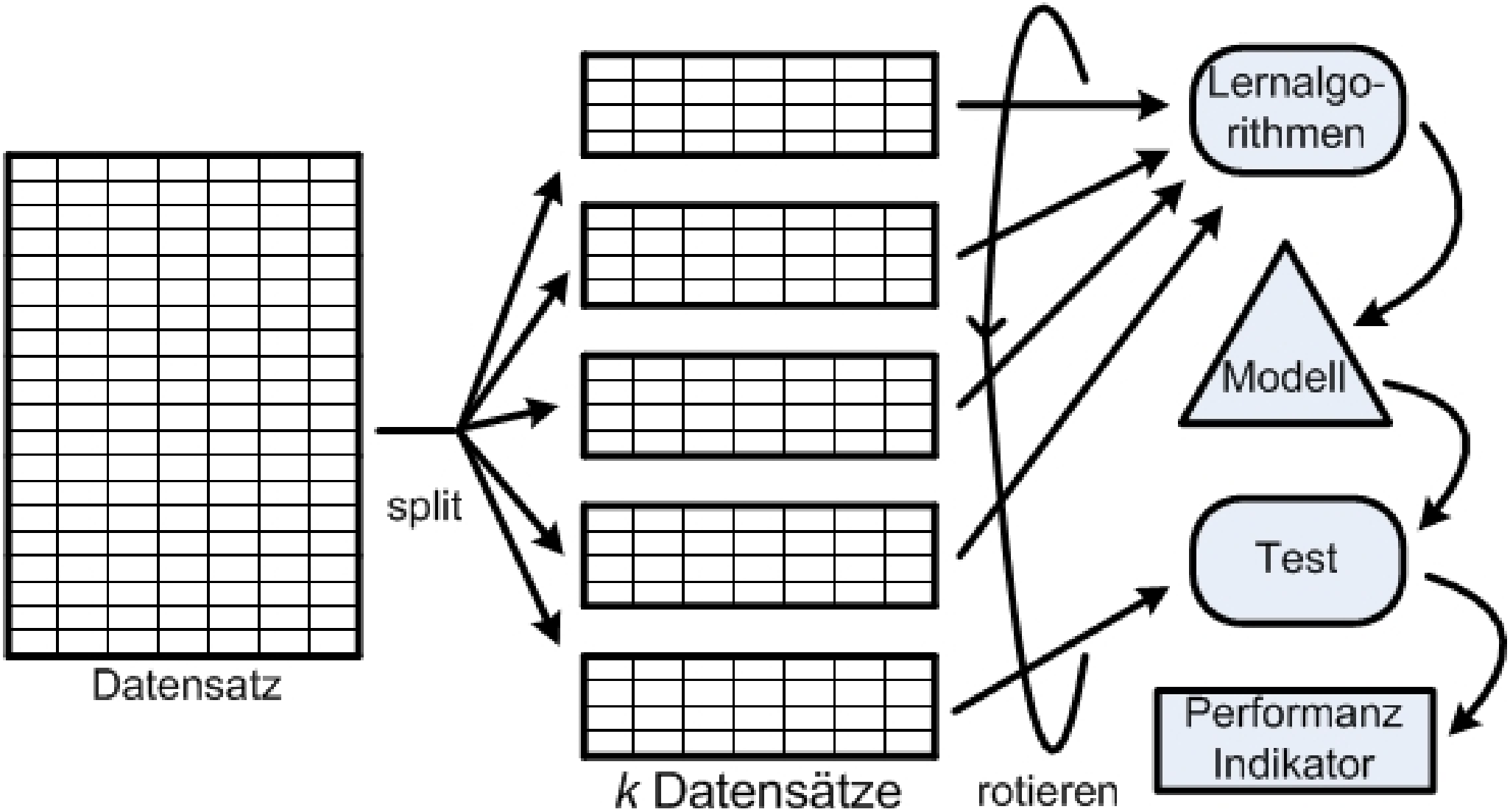




Qualitätsbewertung von Lernalgorithmen: Cross-Validierung



Verlässlichere Qualitätsbewertung: k-fache Cross-Validierung



Zusammenfassung 2.3 Data-Mining

- Von Datensätzen zu Entscheidungsbäumen durch überwachtes / nicht überwachtes Lernen.
- Konfusionsmatrix
- Cluster-Analyse
- Assoziationsregel-Lernen
- Sequence- und Episode-Mining
- Hidden-Markov-Modell und Validierung