DATABASE SYSTEMS GROUP

Ludwig-Maximilians-Universität München
Institut für Informatik
Lehr- und Forschungseinheit für Datenbanksysteme

LMU

Knowledge Discovery in Databases

WS 2017/18

Kapitel 2: Daten Repräsentation

Vorlesung: Prof. Dr. Peer Kröger

Übungen: Anna Beer, Florian Richter

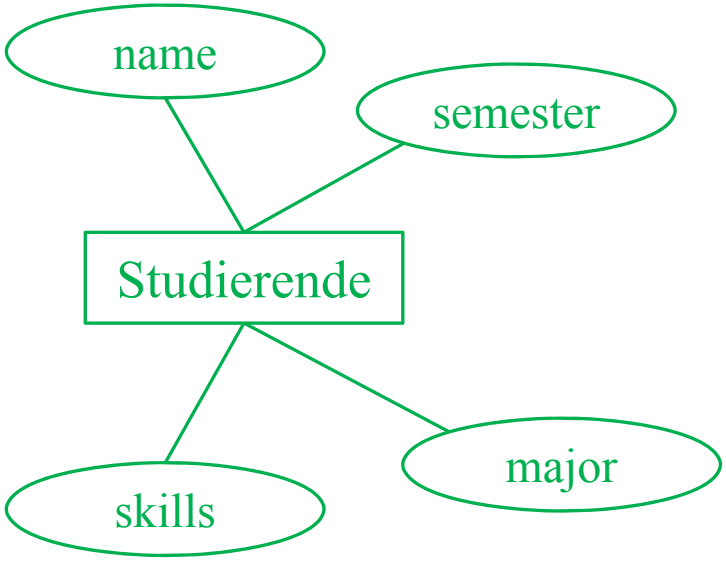
Überblick

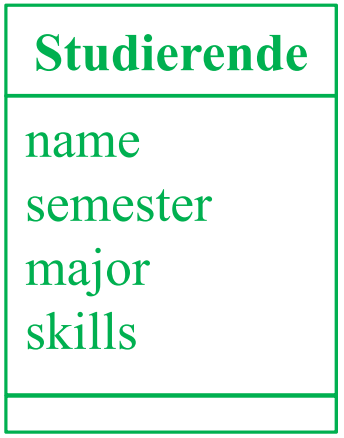
- Daten Repräsentation
  - Datentypen
  - Vergleich von Datenobjekten, Ähnlichkeit
  - Daten Visualisierung

- Data Warehousing
  - Data Cubes
  - Aggregation/Generalisierung

Objekte tragen Informationen

- Daten bestehen aus Objekten und Attributen (Merkmale, features)
  - Entity-Relationship Diagram (ER)
  - UML Klassendiagram
  - Datentabellen (relational model)





name	sem.	major	skills
Ann	3	CS	Java, C, R
Bob	1	CS	Java, PHP
Charly	4	History	Piano, …
Debra	2	Arts	Painting, …

Datentypen von Attributen: Überblick

- Einfache (atomare) Datentypen
  - Nummerisch (numerical) bzw. metrisch (metric), kategorisch (categorical), ordinal
- Zusammengesetzte Datentypen
  - Mengen, Sequenzen, Listen, …
- Komplexe Datentypen
  - Multimedia-Objekte: Bilder, Videos, Audio, Text, Dokumente, Webseiten, …
  - Räumliche/geometrische Objekte: Formen (shapes), Moleküle, Geo-Daten, …
  - Struktur-Objekte: Graphen, Netzwerke, Bäume, …
- Beispiel für komplexe Objekte:
  - Moleküle: Form + Struktur + nummerische Daten (pysikal.-chem. Eigenschaften)
  - City maps: Form + Verkehrsnetz + Points of Interest + …
  - Mechanische Teile: Form + physikal. Eigenschaften + Produktionsprozess-Beschreibung

Einfache (atomare) Datentypen

- Skalen Niveaus von Merkmalen

Nominal (kategorisch)

Charakteristik:

Nur feststellbar, ob der Wert gleich oder verschieden ist. Keine Richtung (besser, schlechter) und kein Abstand. Merkmale mit nur zwei Werten nennt man *dichotom*

Beispiele:

Geschlecht (dichotom)
Augenfarbe
Gesund/krank (dichotom)

Ordinal

Charakteristik:

Es existiert eine Ordnungsrelation (besser/schlechter) zwischen den Kategorien, aber kein einheitlicher Abstand

Beispiele:

Schulnote (metrisch?)
Güteklasse
Altersklasse

Metrisch

Charakteristik:

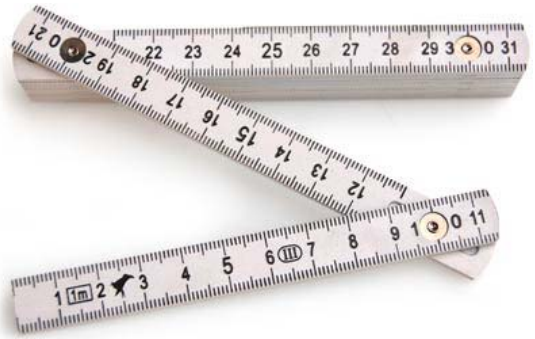
Sowohl Differenzen als auch Verhältnisse zwischen den Werten sind aussagekräftig. Die Werte können diskret oder stetig sein.

Beispiele:

Gewicht (stetig)
Verkaufszahl (diskret)
Alter (stetig oder diskret)

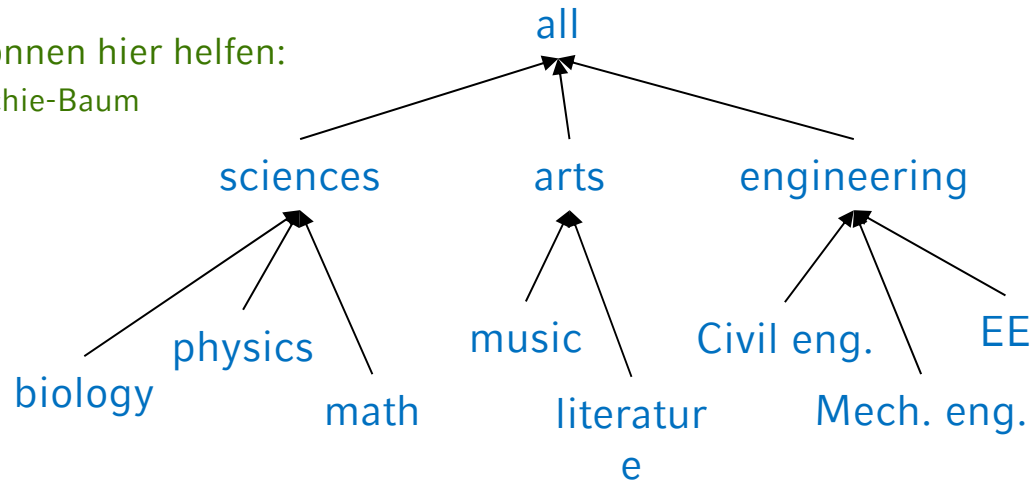
Vergleiche von atomaren Datentypen

- Nummerische Daten
  - Zahlen: natürliche, ganze (Integer), rationale, reelle Zahlen
  - Beispiele: Alter, Einkommen, Schuhgröße, Körpergröße, Gewicht
  - Vergleich von nummerischen Daten: Differenz
  - Beispiel: 30-3 = 27 verglichen zu 3000-3 = 2997
- Generell: metrische Daten
  - Metrischer Raum (O, d)
    - Menge der Daten-Objekte O
    - Metrische Distanzfunktion d
  - Vergleich durch metrische Distanzfunktion $d: O \times O \rightarrow \mathbb{R}_0^+$
    - Symmetrie: $\forall p, q \in O: d(p,q) = d(q,p)$
    - Identität: $\forall p, q \in O: d(p,q) = 0 \Leftrightarrow p = q$
    - Dreiecksungleichheit: $\forall p, q, o \in O: d(p,q) \leq d(p,o) + d(o,q)$
  - Beispiel: 2D Punkte – Euklidische Distanz



Vergleiche von atomaren Datentypen

- Kategorische Data
  - „Just identifiers“
    - Beispiel „occupation“ = {butcher, hairdresser, physicist, physician, … }
    - Beispiel „subjects“ = {physics, biology, math, music, literature, history, EE, … }
  - Vergleich: ???
    - Triviale Metrik: $d(p,q) = \begin{cases} 0 & if\ p = q \\ 1 & else \end{cases}$
      - Funktioniert immer, ist aber ziemlich grob weil nur identische Werte gezählt werden
    - Generalisierungs-Hierarchien können hier helfen:
      - Distanz = Pfadlänge im Hierarchie-Baum
      - $d(music, literature) = 2$
      - $d(music, biology) = 4$
      - $d(music, music) = 0$



Vergleiche von atomaren Datentypen

- Ordinale Daten
  - Diese Daten folgen einer (total) Ordnung $\leq$
    - Transitivität $\forall p, q, o \in O: \; p \leq q \wedge q \leq o \Rightarrow p \leq o$
    - Antisymmetrie $\forall p, q \in O: \; p \leq q \wedge q \leq p \Rightarrow p = q$
    - Abgeschlossenheit $\forall p, q \in O: \; p \leq q \vee q \leq p$
  - Beispiele
    - Zahlen $3 \leq 30 \leq 3{,}000$
    - Wörter high $\leq$ highschool $\leq$ highscore (i.e., lexicographical order)
    - Häufigkeiten „How often did you sleep bad last year?“
      never $\leq$ seldom $\leq$ rarely $\leq$ occasionally $\leq$ sometimes $\leq$ often $\leq$ frequently $\leq$ regularly $\leq$ usually $\leq$ always
    - (Vage) Größenangaben „How big was that problem?“
      tiny $\leq$ small $\leq$ medium $\leq$ big $\leq$ huge

Beispiele von ordinalen Daten

- Icecold < cold < cool < lukewarm < warm < hot
- Never < seldom < rarely < occasionally < sometimes < often < normally < regularly < usually < always (http://www.englisch-hilfen.de/en/grammar/adverbien1.htm)
- Never < rarely < seldom < occasionally < sometimes < often < usually < frequently < always (http://www.eslgold.com/grammar/frequency_adverbs.html)
- By no means < unlikely < maybe < likely < probably < for sure
- Impossible, Very Unlikely, Unlikely, Maybe, Likely, Very Likely , Obvious (http://forum.paradoxplaza.com/forum/showthread.php?485349-Impossible-Very-Unlikely-Unlikely-Maybe-Likely-Very-Likely)
- Very bad < bad < good < very good < excellent < outstanding
- Rare < medium < well-done
- Tiny < small < big < huge
- XS < S < M < L < XL < XXL
- Yesterday < today < tomorrow
- Early < on-time < late
- much, many, some, any, a little, a few, each, every, a lot of, lots of (http://www.englisch-hilfen.de/grammar_list/mengen.htm)

Distanzen und Ähnlichkeiten

Statt mit Distanzmaßen, die die Unähnlichkeit zweier Objekte messen, arbeitet man manchmal auch mit Ähnlichkeitsmaßen (bzw. *vice versa*):

$$sim(x,y) = 0 \approx \text{unendliche Distanz}$$

häufig maximale Ähnlichkeit 1:

$$sim(x,y) = 1 \Leftrightarrow dist(x,y) = 0$$

Abbildungen von Ähnlichkeiten auf Distanzen (Vorsicht!!!!!!!!!!!!!!!!!!):

$$dist(x,y) = 1 - sim(x,y)$$

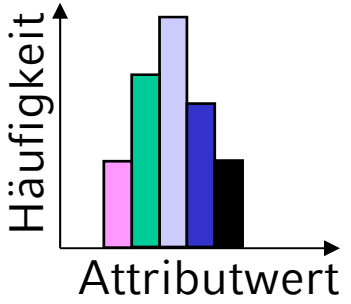
$$dist(x,y) = -\ln(sim(x,y))$$

(Univariate) Deskription von Merkmalen

Sei $x_1,\ldots,x_n$ eine Stichprobe eines Merkmals X.

- Absolute Häufigkeit: Für jeden Wert a ist $h(a)$ die Anzahl des Auftretens in der Stichprobe
- Relative Häufigkeit: $p(a) = h(a)/n$

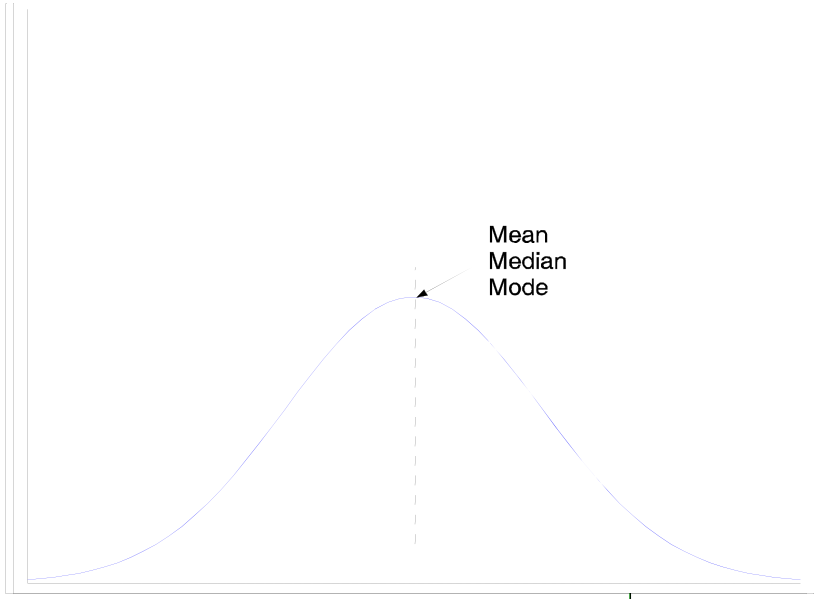
Histogramm:



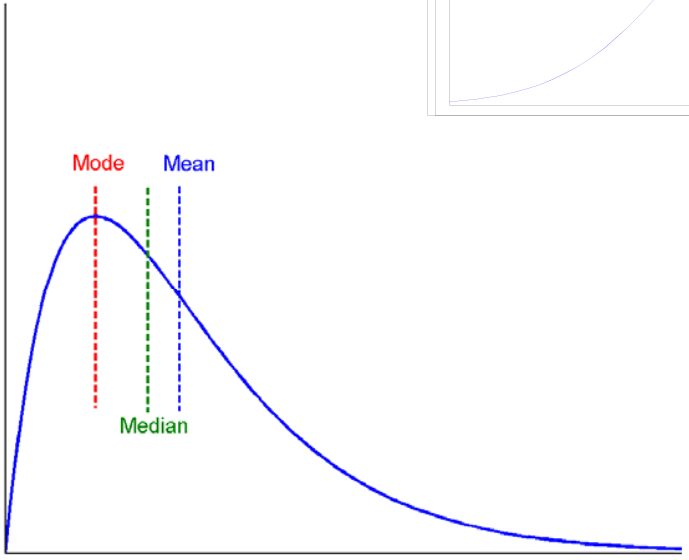
Die folgenden Maße sind nur für metrische Merkmale sinnvoll:

- Arithmetisches Mittel: $\mu = \overline{x} = \frac{1}{n} \cdot \sum_{i=1}^{n} x_i$
- Median: *Das mittlere Element bei aufst. Sortierung*
- Modus (mode): *Ausprägung mit größter Häufigkeit*
- Varianz: $VAR(X) = \frac{1}{n-1} \cdot \sum_{i=1}^{n} (x_i - \mu)^2$
- Standardabweichung: $\sigma = \sqrt{\frac{1}{n-1} \cdot \sum_{i=1}^{n} (x_i - \mu)^2}$

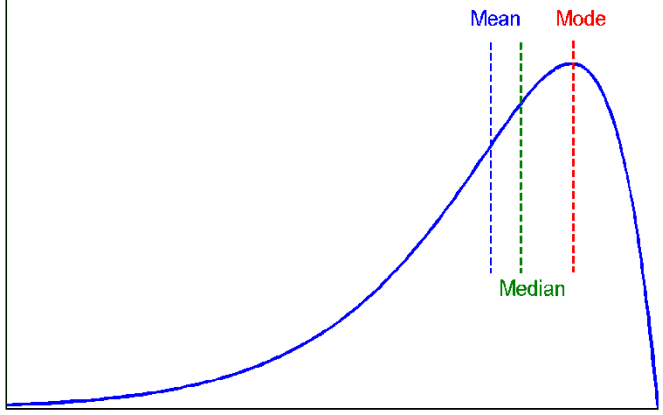
Symmetric vs. Skewed



Symmetric



Positively skewed



Negatively skewed

Zusammenhang von Merkmalen (multivariate Deskription)

Kontingenztabelle

- für kategorische Merkmale X und Y
- repräsentiert für zwei Merkmale X und Y die absolute Häufigkeit h_{ik} jeder Kombination (x_i, y_k) und alle Randhäufigkeiten $h_{.k}$ und $h_{i.}$ von X und Y

	Mittelfristige Arbeitslosigkeit	Langfristige Arbeitslosigkeit	
Keine Ausbildung	19	18	37
Lehre	43	20	63
	62	38	100

- Wie sollten die relativen Häufigkeiten verteilt sein, wenn die beiden Merkmale keinerlei Abhängigkeit besitzen?

$$p_i = \frac{h_{i.}}{n}, p_{ij} = p_i p_j$$

- *χ^2-Koeffizient*
  Differenz zwischen dem bei Unabhängigkeit erwarteten und dem tatsächlich beobachteten Wert von h_{ij} (Maß für die Stärke der Abhängigkeit)

$$\chi^2 = \sum_{i=1}^{c}\sum_{j=1}^{r}\frac{(o_{ij} - e_{ij})^2}{e_{ij}}$$

o_{ij}: beobachtete Häufigkeit
e_{ij}: erwartete Häufigkeit

$$e_{ij} = n \cdot p_i \cdot p_j = \frac{h_i h_j}{n}$$

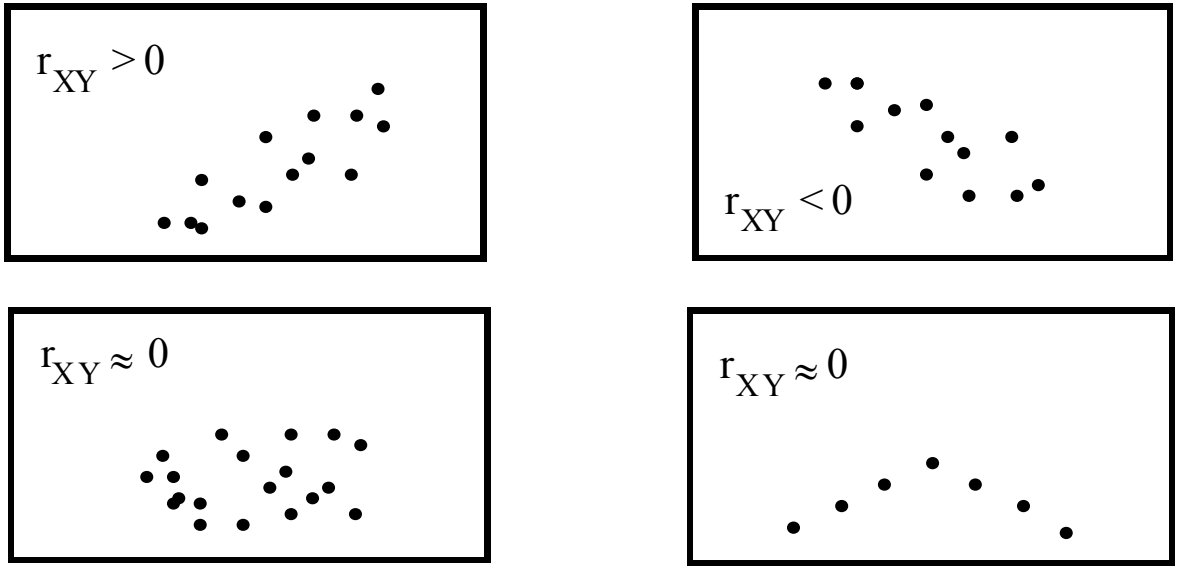
Zusammenhang von Merkmalen (multivariate Deskription)

Korrelationskoeffizient

- für numerische Merkmale X und Y
- wie stark sind die Abweichungen vom jeweiligen Mittelwert korreliert?

$$r_{XY} = \sum_{i=1}^{n} \frac{(x_i - \bar{x}) \cdot (y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n} (x_i - \bar{x})^2 \cdot \sum_{i=1}^{n} (y_i - \bar{y})^2}}$$

- Beispiele



Multivariate Deskription von Merkmalen

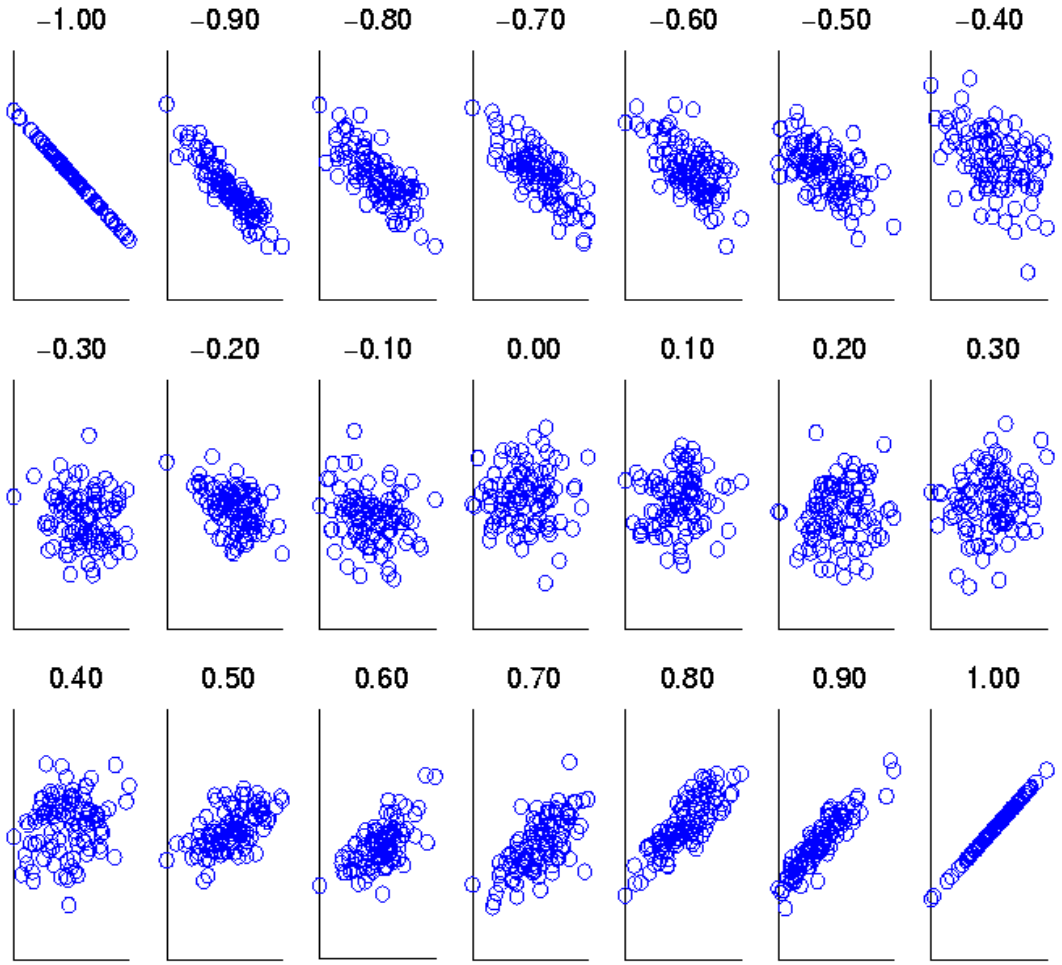


Figure 5.11. Scatter plots illustrating correlations from -1 to 1.

Normalisierung

- Attribute mit großem Wertebereich gehen stärker in Distanzen ein als solche mit kleinem Wertebereich
  - z.B. Einkommen [10K-100K]; Alter [10-100]
- Skalierung von Attributen in einen einheitlichen Wertebereich, um die Beiträge aller Attribute zur Distanz gleich zu gewichten
- min-max Normalisierung zu [new_min_A , new_max_A]

$$v' = \frac{v - min_A}{max_A - min_A}(new\_max_A - new\_min_A) + new\_min_A$$

  - z.B. normalisiere Alter=30 in [0,1], mit min=10,max=100. new_age=(30-10)/(100-10)=2/9
- z-score Normalisierung

$$v' = \frac{v - mean_A}{stand\_dev_A}$$

z.B. normalisiere 70000 mit μ=50000, σ=15000.
new_value = (70000-50000)/15000=1.33

Zusammengesetzte Datentypen

- Mengen
  - Zusammenfassung individueller Werte
  - Beispiel: $skills = \wp(\{Java, C, Python, R, \ldots\})$
  - Vergleich
    - Symmetrische Mengendifferenz: $d(R,S) = (R - S) \cup (S - R) = (R \cup S) - (R \cap S)$
    - Jaccard Distanz:

$$d(R,S) = \frac{(R \cup S) - (R \cap S)}{R \cup S}$$

  - Bitvector Repräsentation einer Menge bzgl. einer Basismenge:
    - Basismenge: $B = \langle math, physics, chemistry, biology, music, arts, english \rangle$
    - Beispielmengen: $S = \{math, music, english\} = \langle 1,0,0,0,1,0,1 \rangle$
      $R = \{math, physics, arts, english\} = \langle 1,1,0,0,0,1,1 \rangle$
    - Hamming Distanz = Summe der unterschiedlichen Einträge: $d(R,S) = 3$
      - Effiziente Berechnung der symmetrischen Mengendifferenz (s.o.)

Zusammengesetzte Datentypen

- Sequenzen, Listen, Vektoren
  - Zusammenfassen von n Werten der selben Domäne D
  - Reihenfolge Spielt nun eine Rolle: $\mathrm{I}_n \rightarrow D$ für eine Indexmenge $I_n = \{1, \ldots, n\}$

- Vergleich von zusammengesetzten Objekten: 2-stufig
  - Bestimme die individuellen Unterschiede $|o_i - q_i|$, oder Distanzen $d(o_i, q_i)$
  - Kombiniere diese individuellen Distanzen zu einer Gesamtdistanz $d(o, q)$

- Beispiele
  - (Simple) sum: $d_1(o,q) = \sum_{i=1}^{n} |o_i - q_i|$ (Manhattan)
  - Root of sum of squares: $d_2(o,q) = \sqrt{\sum_{i=1}^{n} (o_i - q_i)^2}$ (Euclidean)
  - Maximum: $d_\infty(o,q) = \max_{i=1,\ldots,n} \{|o_i - q_i|\}$ (Maximum)
  - Generelle Formel: $d_p(o,q) = \sqrt[p]{\sum_{i=1}^{n} |o_i - q_i|^p}$ (Minkowski)
  - Weighted Minkowski dist.: $d_{p,w}(o,q) = \sqrt[p]{\sum_{i=1}^{n} w_i \cdot |o_i - q_i|^p}$

Zusammengesetzte Datentypen

- Komplex zusammengesetzte Datentypen aus Komponenten
  - Strukturierte Komponenten: graphs, networks, trees
  - Geometrie: shapes / contoures, routes / trajectories
  - Multimedia: images, audio, text, etc.

- Vergleich: Ähnlichkeitsmodelle

  Generelle Ansätze zur Modellierung von Ähnlichkeit komplexer Objekte
  - Direkte Maße – extrem abhängig vom Daten Typ
  - Feature Extraction – explizite Einbettung in einen Vektorraum („embedding“)
  - Kernel Trick – implizite Einbettung in einen Vektorraum

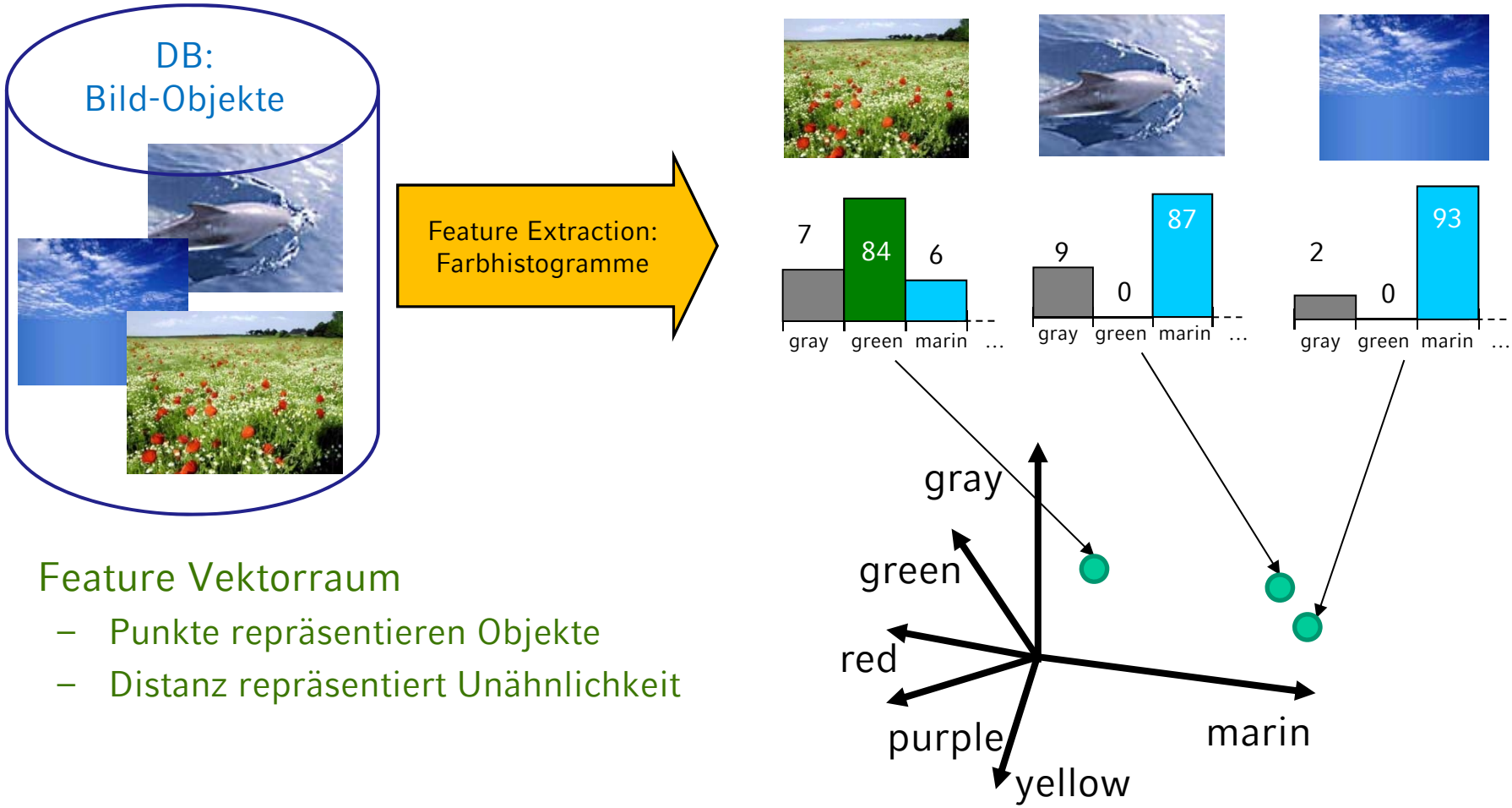
  Achtung: Ähnlichkeitsmodelle sind teilw. eigenständige Forschungsfragen

- Beispiele für Ähnlichkeitsmodelle

Examples	Direct distance	Feature-based	Kernel-based
Graphs	Structural alignment	Degree histograms	Label sequence kernel
Geometry	Hausdorff distance	Shape histograms	Spatial pyramid kernel
Sequences	Edit distance	Symbol histograms	Cosine distance

Feature Extraktion

- Datenobjekte werden abgebildet auf Merkmals-Vektoren („feature vectors“)



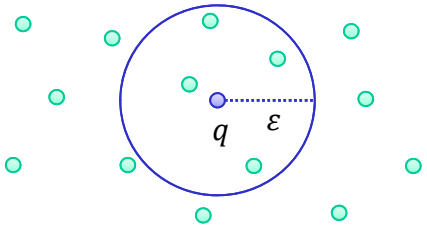
- Feature Vektorraum
  - Punkte repräsentieren Objekte
  - Distanz repräsentiert Unähnlichkeit

Ähnlichkeitsanfragen

- Statt Identisches zu zählen kann man in Feature-Räumen nur Ähnliches bestimmen => Ähnlichkeitsanfragen
- DB: (Feature-Vektor) Datenbank, Distanzfunktion d, Anfrageobjekt q:

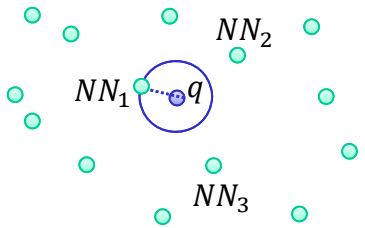
- **Range query** for range parameter $\varepsilon \in \mathbb{R}_0^+$:

$$range(DB, q, d, \varepsilon) = \{o \in DB | d(o, q) \leq \varepsilon\}$$



- **Nearest neighbor query**:

$$NN(DB, q, d) = \{o \in DB | \forall o' \in DB: \ d(o, q) \leq d(o', q)\}$$



- ***k*-nearest neighbor query** for parameter $k \in \mathbb{N}$:

$$NN(DB, q, d, k) \subset DB \ with \ |NN(DB, q, d, k)| = k \ and$$
$$\forall o \in NN(DB, q, d, k), o' \in DB - NN(DB, q, d, k): d(o, q) \leq d(o', q)$$

- **Ranking query** (partial sorting query): „get next“ functionality for picking database objects in an increasing order wrt. to their distance to q:

$$\forall i \leq j: \ d\left(q, rank_{DB,q,d}(i)\right) \leq d\left(q, rank_{DB,q,d}(j)\right)$$

Beispiel: Bilder

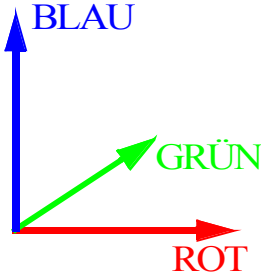
- Hauptkategorien von Features für Bilder
  - Farbverteilung (Farbhistogramme)
  - Textur (Oberflächen-Beschaffenheit von Bildsegmenten, z.B. Holzmaserung, Kieselsteine, Karomuster)
  - Formen (Konturen)

- Farbhistogramme:
  - Repräsentation der Farbverteilung in einem Bild (auf Pixelbasis)
  - Definition der Farbhistogramme
    - Farbraum festlegen (z.B. RGB, HSV, HLS, ...)
    - Menge von Repräsentanten im Farbraum auswählen (sample points), z.B. Gitter im Farbraum mit 4×4×4 = 64 Farben oder 8×8×8 = 512 Farben

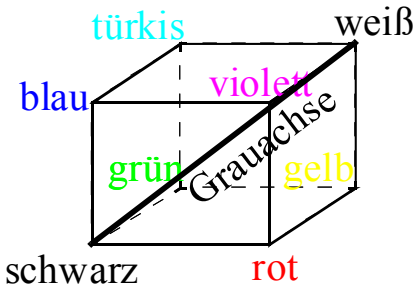
Beispiel: Bilder (Farbhistogramme)

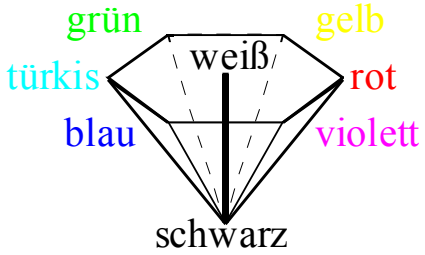
- Farbräume: Technische Modelle (RGB, CMY) und anschauliche Modelle (HSV, HLS)

RGB-Modell
(Bildschirm, additiv)

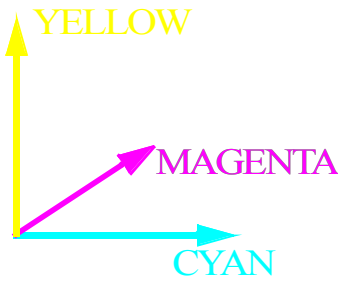


HSV-Modell: Hue, Saturation, Value
(Farbton, Sättigung, Helligkeit)

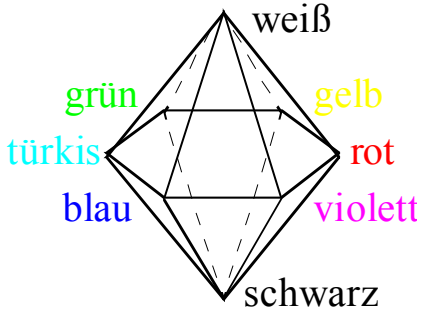




CMY-Modell
(Drucker, subtraktiv)

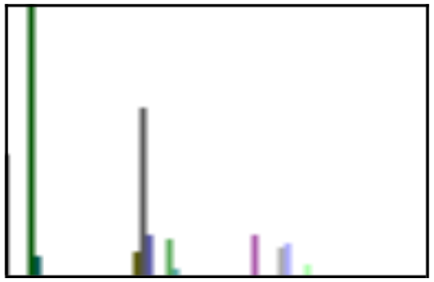


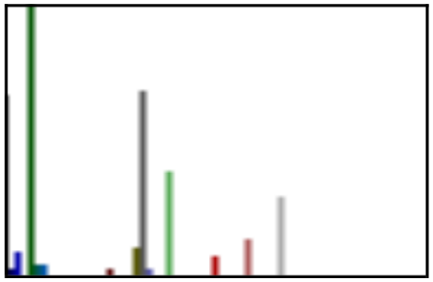
HLS-Modell: Hue, Luminance, Saturation
(Farbton, Leuchtkraft, Sättigung



Beispiel: Bilder (Farbhistogramme)

- Berechnung der Farbhistogramme
  - Für jedes Pixel, erhöhe den Zähler des nächstgelegenen Repräsentanten um eins
  - Evtl. Normierung, um Histogramm von der Bildgröße unabhängig zu machen
  - Beispiel (64 Repräsentanten):

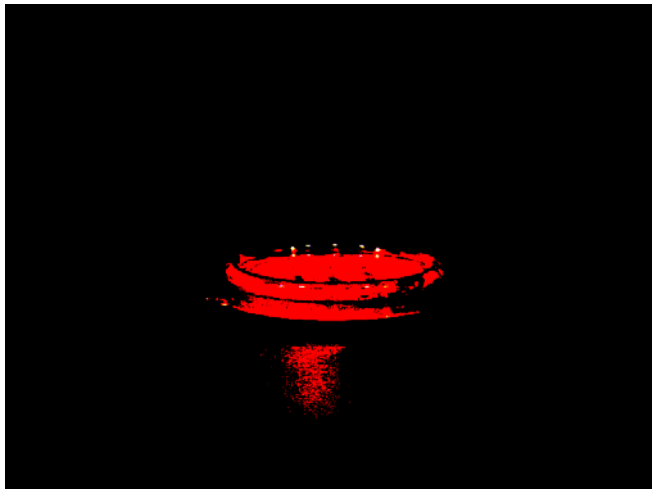


Beispiel: Bilder (Farbhistogramme)

Bins pro Achse: 256 2

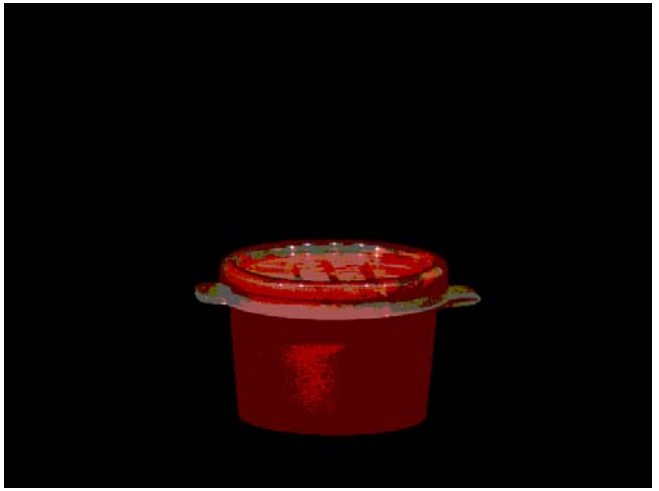
3 4

Beispiel: Bilder (Farbhistogramme)

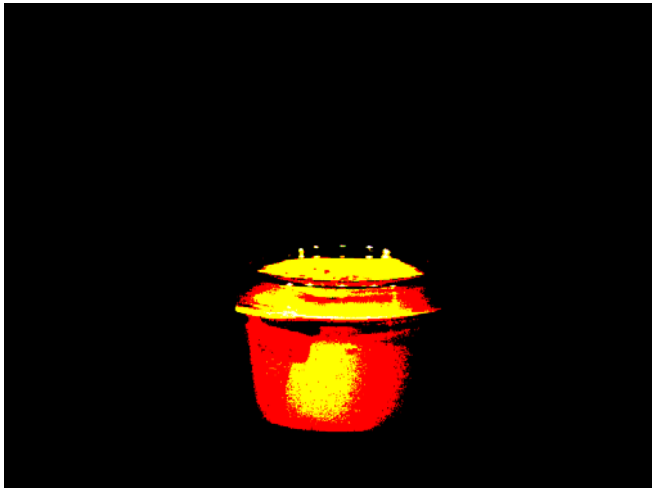


Bins pro Achse: 256 2

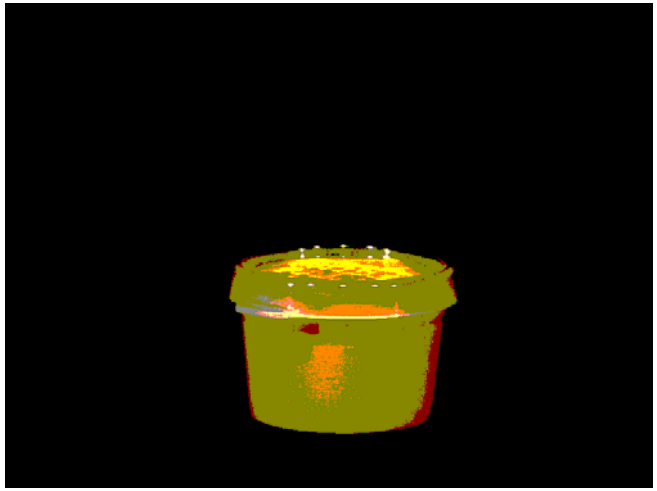
3 4



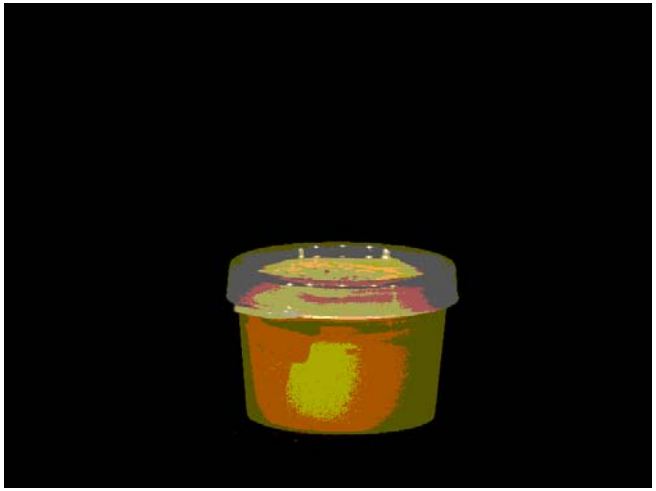
Beispiel: Bilder (Farbhistogramme)



Bins pro Achse: 256 2



3



4

Beispiel: Bilder (Farbhistogramme)

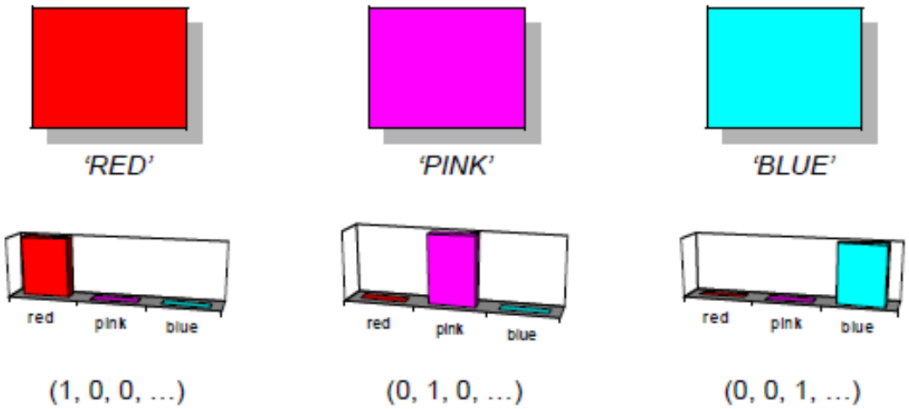
Bins pro Achse: 256 2

3 4

Distanzen für Farbhistogramme

- 1. Idee: euklidische Distanz für Farbhistogramme h_P und h_Q der Bilder P und Q:

$$dist(P,Q) = \sqrt{(h_P - h_Q)\cdot(h_P - h_Q)^T}$$



$$dist('RED','PINK') = \sqrt{2}$$

$$dist('RED','BLUE') = \sqrt{2}$$

$$dist('PINK','BLUE') = \sqrt{2}$$

- Alle Paare von Bildern haben denselben Abstandswert $\sqrt{2}$
- Distanz berücksichtigt nicht, dass rot (subjektiv) ähnlicher zu lila ist als zu blau.

Distanzen für Farbhistogramme

- Quadratische Form mit Ähnlichkeitsmatrix:

$$dist_A(P,Q) = \sqrt{(h_P - h_Q) \cdot A \cdot (h_P - h_Q)^T}$$
$$= \sqrt{\sum_i \sum_j a_{ij} \cdot (h_{P_i} - h_{Q_i}) \cdot (h_{P_j} - h_{Q_j})}$$

$$A = \begin{bmatrix} 1 & a_{21} & \cdots & \\ a_{12} & 1 & a_{ij} & \vdots \\ \vdots & & 1 & \\ & \cdots & & 1 \end{bmatrix}$$

- Einträge a_{ij} ($= a_{ji}$?) beschreiben die Ähnlichkeit der Dimensionen i und j in den Vektoren (Bins i und j in den Histogrammen)

$$A' = \begin{bmatrix} 1 & 0{,}9 & 0 \\ 0{,}9 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

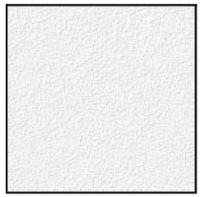
$$dist_{A'}('RED','PINK') = \sqrt{0{,}2}$$
$$dist_{A'}('RED','BLUE') = \sqrt{2}$$
$$dist_{A'}('PINK','BLUE') = \sqrt{2}$$

- Ähnlichkeitsmatrizen werden aus Ergebnissen der Perzeptionsforschung abgeleitet

Weitere Features: Texturen für Bilder

- Gerichtetheit, Orientiertheit (Directionality)
  - Vorhandensein von Vorzugsrichtungen
    (Verteilung der Gradientenrichtungen)
- Kontrast
  - Lebendigkeit (Unruhe) eines Musters
  - Berechnung aus Varianz im Grauwerthistogramm
- Granularität (Coarseness)
  - Größenordnung der Textur
  - Berechnung durch über das Bild verschobene Fenster unterschiedlicher Größe



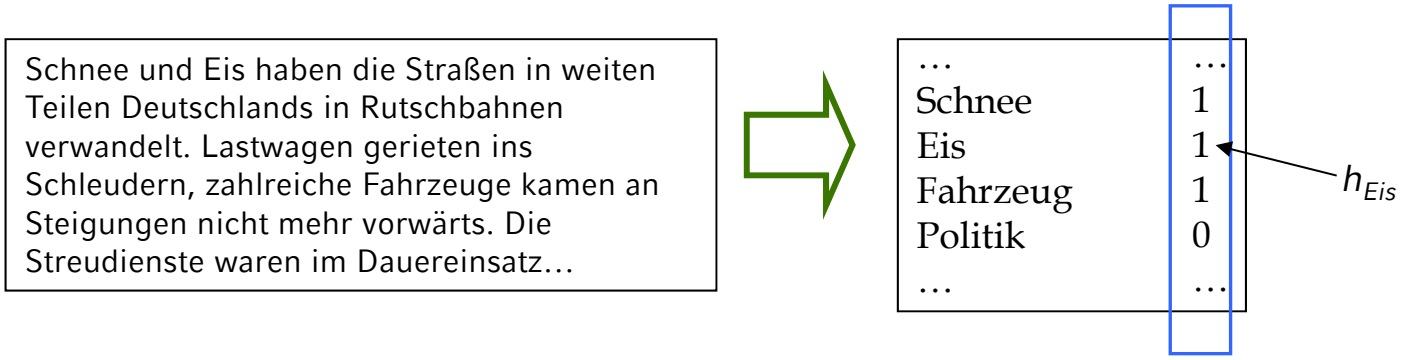



Toolbox für Feature-Extraktion von Bildern:
http://code.google.com/p/jfeaturelib/


Beispiel: Texte

- *Text als Mengen/Vektoren von Termen: („Bag-Of-Words“)*
  - Term:
    - einzelnes Wort („Schnee“, „Eis“..)
      oder
    - zusammenhängendes Satzfragment („nicht mehr vorwärts“..)
  - Transformation eines Dokuments D in Vektor $r(D) = (h_1, ..., h_d)$

$h_i \geq 0$: die Häufigkeit des Terms t_i in D



Beispiel: Texte

- Probleme im Textmining
  1. Viele Wörter nutzlos (z.B. er, sie, es, und, als, der, dies, das...)
  2. Wörter haben gleichen Wortstamm („gehen“ „ging“)
  3. Sehr hochdimensionale Featureräume (häufig $d > 10.000$)
  4. Nicht alle Terme sind gleich wertvoll
  5. Die meisten Termhäufigkeiten $h_i = 0$ („sparse feature space“)
- weitere Probleme aus der Linguistik:
  - unterschiedliche Wörter haben gleiche Bedeutung
    „laufen“ ⇔ „rennen“
  - Wörter haben mehrere Bedeutungen
    „Maus“: Computermaus, Nagetier…

Feature-Transformationen für Texte

- Problem 1: Viele Wörter nutzlos (z.B. er, sie, es, und, als, der, dies, das...)
  - Lösung: Streichen solcher Terme (Stopwords)

    Für alle Sprachen werden Stopwordlisten im WWW publiziert.

- Problem 2: Wörter haben gleichen Wortstamm („gehen“ „ging“)
  - Lösung: Stemming

    Worte auf Wortstamm rückführen (z.B. lief, läuft, lauft => laufen)

    Im Englischen algorithmisches Stemming möglich.
    (Porters Stemming Algorithms: http://tartarus.org/~martin/PorterStemmer/index.html)

    In anderen Sprachen werden Dictionaries benötigt, die die Wortstämme zu den Vokabeln enthalten.

Feature-Transformationen für Texte

- Problem 3: Sehr viele Terme müssen betrachtet werden.
  - Lösung: Auswahl der wichtigsten Features („Feature Selection“)
  - Beispiel: Mittlere Dokumentenhäufigkeit
    - Sehr häufige Terme kommen scheinbar in allen Dokumenten vor
      => Vorkommen unterscheidet kaum Dokumente
    - Sehr seltene Terme kommen nur in Bruchteil der Dokumente vor
      => Nichtvorkommen unterscheidet kaum Dokumente

  Vorgehen:

  1. Berechne Dokumenthäufigkeit für alle Terme t_i : $DF(t_i) = \frac{|Dok\_t_i|}{|ALL\_Doks|}$
  2. Sortiere Terme nach DF(t_i) und vergebe Rang rank(t_i)
  3. Sortiere Terme nach score(t_i)= DF(t_i) · rank(t_i)
     z.B. score(t_{23}) = 0.82 · 1 = 0.82
     score(t_{17}) = 0.65 · 2 = 1.5
  4. Wähle die k Terme mit dem größten Wert für score(t_i)

Rank	Term	DF
1.	t_{23}	0.82
2.	t_{17}	0.65
3.	t_{14}	0.52
4.	...	...

Feature-Transformationen für Texte

- Problem 4: Nicht alle Terme sind gleich wertvoll.
  - Idee:
    1. Gewichte seltene Terme höher als häufige.
    2. Gewichte häufig in einem Dokument auftretende Terme höher als solche die nur einmal vorkommen.
  - Lösung: TF-IDF (Term Frequency · Inverse Document Frequency)

    Berücksichtige sowohl die relative Anzahl der Vorkommen im Dokument als auch die Seltenheit des Terms.

$$TF(t,d) = \frac{n(t,d)}{\sum_{w \in d} n(w,d)}$$

relative Häufigkeit von t in d

$$IDF(t) = \frac{|DB|}{|\{d \mid d \in DB \wedge t \in d\}|}$$

inverse Häufigkeit von t bzgl. aller Dokumente

Featurevektor mit TF IDF : $r(d) = (TF(t_1,d) \cdot IDF(t_1), \ldots, TF(t_n,d) \cdot IDF(t_n))$

Feature-Transformationen für Texte

- Problem 5: die meisten Termhäufigkeiten $h_i = 0$

  => *Euklidische Abstände sehr ähnlich*

  – Lösung: Verwendung anderer Abstandsmaße

  Idee: Verwende Terme, die beide Dokumente (D_1, D_2) gemeinsam haben.

  Jaccard Coefficient: Dokumente als Termmengen

$$d_{Jaccard}(D_1, D_2) = 1 - \frac{|D_1 \cap D_2|}{|D_1 \cup D_2|}$$

  Cosinus Coefficient: Abstand für Wortvektoren (evtl. TF IDF)

$$d_{\text{cosinus}}(D_1, D_2) = 1 - \frac{\langle D_1, D_2 \rangle}{\|D_1\| \cdot \|D_2\|} = 1 - \frac{\sum_{i=0}^{n} (d_{1,i} \cdot d_{2,i})}{\sqrt{\sum_{i=0}^{n} d_{1,i}^{\ 2}} \cdot \sqrt{\sum_{i=0}^{n} d_{2,i}^{\ 2}}}$$