

Data Mining Tasks

- Classification
- Prediction
- Clustering
- Dependency Modelling
- Summarization
- Change and Deviation Detection
- Visualization

Prediction

- prediction of a (future) category based on observed data → classification
- prediction of a (future) numerical value y based on observed data \vec{x}
 - y : response output, dependent variable
 - \vec{x} : input, regressors, explanatory variables, independent variables
- applications
 - the output is expensive to measure, the input not
 - the value of the inputs is known before the value of the output and a prediction is required
 - simulation of system behaviour by controlling the inputs
 - detecting causal links between the inputs and the output

Regression

- most common form: linear regression
 - assuming a linear function

$$y = f(\vec{x}) = a_0 + \sum_{i=1}^n a_i \cdot x_i$$

- inserting all m training samples $\rightarrow m$ new equations

$$y_j = \epsilon_j + a_{0j} + \sum_{i=1}^n a_i \cdot x_{ij}$$

$\epsilon_j (j = 1 \dots m)$: regression error for each given sample

- modify the linear coefficients a_i to minimize the sum of error squares $e = \sum_{i=1}^n \epsilon_i^2$

Regression

- special case: single predictor variable

$$y = f(x) = a_0 + a_1 \cdot x$$

$$e = \sum_{i=1}^n \epsilon_i^2 = \sum_{i=1}^n (y_i - y'_i)^2 = \sum_{i=1}^n (y_i - a_0 - a_1 x_i)^2$$

- minimizing for a_0 and a_1

$$\frac{\delta e}{\delta a_0} = -2 \sum_{i=1}^n (y_i - a_0 - a_1 x_i) = 0$$

$$\frac{\delta e}{\delta a_1} = -2 \sum_{i=1}^n (y_i - a_0 - a_1 x_i) \cdot x_i = 0$$

Regression

- minimizing (cont.)

$$a_0 + a_1 \sum_{i=1}^n x_i = \sum_{i=1}^n y_i$$

$$a_0 \sum_{i=1}^n x_i + a_1 \sum_{i=1}^n x_i^2 = \sum_{i=1}^n x_i y_i$$

$$a_0 = \mu_y - a_1 \mu_x$$

$$a_1 = \frac{\sum_{i=1}^n (x_i - \mu_x) \cdot (y_i - \mu_y)}{\sum_{i=1}^n (x_i - \mu_x)^2}$$

Regression

- multiple regression (multiple predictor variables)

$$y = a_0 + \vec{a} \cdot \vec{x}$$

$$e = (\vec{y} - a_0 \vec{a} \cdot X)^T \cdot (\vec{y} - a_0 \vec{a} \cdot X)$$

X : Matrix of all data vectors \vec{x}_j from the training set

$$\vec{a} = (X^T \cdot X)^{-1} (X^T \cdot \vec{y})$$

- solution of equation set requires exponential effort
- not feasible for realistic training sets

Regression

- identifying the relevant variables
 - selectively add to or delete variables from an initial set
 - testing for a linear relationship: correlation

$$r = \frac{\sum_{i=1}^n (x_i - \mu_x) \cdot (y_i - \mu_y)}{\sqrt{\sum_{i=1}^n (x_i - \mu_x)^2 \cdot \sum_{i=1}^n (y_i - \mu_y)^2}}$$

Regression

- non-linear relationships
 - transform to a linear equation

polynomial	$y = ax^2 + bx + c$	$x^* = x^2$
exponential	$y = ae^{bx}$	$y^* = \ln y$
power	$y = ax^b$	$y^* = \log y, x^* = \log x$
reciprocal	$y = a + b\frac{1}{x}$	$x^* = \frac{1}{x}$
hyperbolic	$y = \frac{x}{a+bx}$	$y^* = \frac{1}{y}, x^* = \frac{1}{x}$

- use neural networks to approximate a nonlinear function
→ low perspicuity

Data Mining Tasks

- Classification
- Prediction
- **Clustering**
- Dependency Modelling
- Summarization
- Change and Deviation Detection
- Visualization

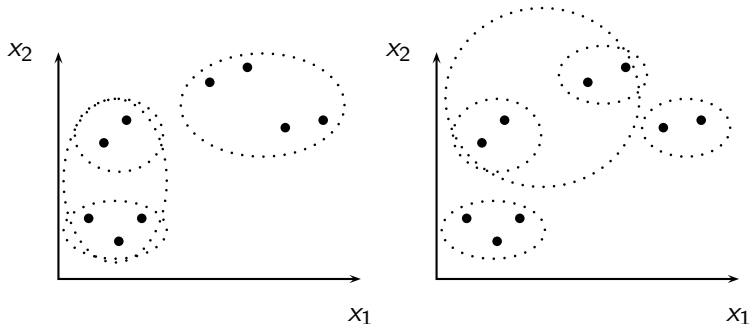
Clustering

- grouping of data points according to their inherent structure
 - based on a similarity measure
 - learning without teacher
- many clustering approaches
 - agglomerative clustering
 - partitioning clustering

 - incremental clustering
 - clustering with neural networks

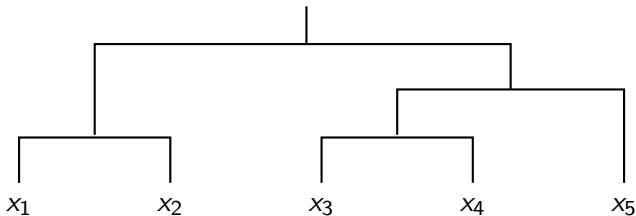
Clustering

- computing the optimal clustering is computationally infeasible
→ greedy, sub-optimal approaches
- different clustering algorithms might lead to different clustering results



Agglomerative Clustering

- agglomerative/hierarchical clustering
- successively merging data sets
- result can be displayed as a dendrogram



Agglomerative Clustering

- algorithm
 - initially each cluster consists of a single data point
 - determine all inter-cluster distances
 - merge the least distant clusters into a new one
 - continue until all clusters have been merged

Distance Measures

- distance measure for clusters
 - single link: minimum of distances between all pairs of data points
 - complete link: e.g. mean of distances between all pairs of data points
- local clustering criterion for data points: minimal mutual neighbor distance (MND)
 - distance depends also on the local context of a data point

$$d_{MND}(\vec{x}_i, \vec{x}_j) = r(\vec{x}_i, \vec{x}_j) + r(\vec{x}_j, \vec{x}_i)$$

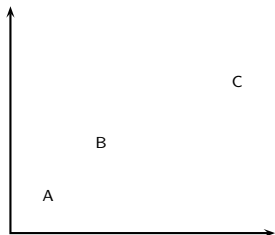
$r(\vec{x}_i, \vec{x}_j)$: rank of x_j according to distance from x_i

Partitioning Clustering

- mutual neighbor distance (MND)

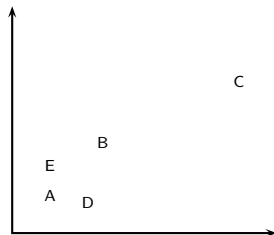
$$\begin{aligned}d_{MND}(A, B) &= r(A, B) + r(B, A) \\ &= 1 + 1 = 2\end{aligned}$$

$$\begin{aligned}d_{MND}(B, C) &= r(B, C) + r(C, B) \\ &= 2 + 1 = 3\end{aligned}$$



$$\begin{aligned}d_{MND}(A, B) &= r(A, B) + r(B, A) \\ &= 3 + 3 = 6\end{aligned}$$

$$\begin{aligned}d_{MND}(B, C) &= r(B, C) + r(C, B) \\ &= 4 + 1 = 5\end{aligned}$$



Partitioning Clustering

- algorithm for k -means partitioning clustering
 - select a randomly chosen initial partitioning with k clusters
 - compute the centroids
 - assign each sample to the nearest centroid
 - compute new centroids
 - continue until the clustering stabilizes (or another termination criterion based on the global error is met)
- often used as a data preprocessing step: Number of Values Reduction

Incremental Clustering

- huge data sets cannot be clustered in a single step
 - divide-and-conquer: cluster subsets and merge the results
 - incremental clustering: data points are loaded successively and the cluster representation is updated accordingly

Incremental Clustering

- algorithm
 - assign the first data point to the first cluster
 - consider the next data point
 - assign it to an already existing cluster, or
 - create a new cluster
 - recompute the cluster description for that cluster
 - continue until all data points are clustered

Incremental Clustering

- cluster description
 - centroid
 - number of data points in the cluster
 - "radius" of the cluster (based on the mean-squared distance to the centroid)
- problems
 - result depends on the order in which data points are processed
 - iterative incremental clustering
 - use the centroids of the previous iteration for partitioning in the next one

Data Mining Tasks

- Classification
- Prediction
- Clustering
- Dependency Modelling
- Summarization
- Change and Deviation Detection
- Visualization

Dependency Modelling

- prediction of events commonly occurring together
- market basket analysis: which items are often purchased together
 - placement of items in a store
 - layout of mail-order catalogues
 - targeted marketing campaigns
- association rules: rules of the form

$$a \wedge b \wedge \dots \wedge c \rightarrow d \wedge e$$

- finding good combinations of premises is a combinatorial problem

Association Rules

- example data base:

trans- action	item
001	cola
001	chips
001	peanuts
002	beer
002	chips
002	cigarettes
...	...

trans- action	items
001	{chips, cola, peanuts}
002	{beer, chips, cigarettes}
003	{beer, chips, cigarettes, cola}
004	{beer, cigarettes}

Association Rules

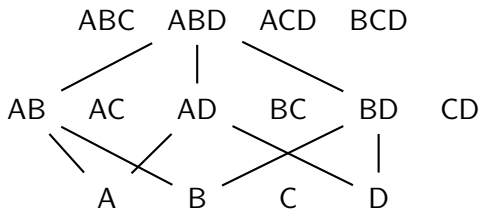
- set of n different items $I = \{x_j | j = 1, \dots, n\}$
- itemset: $I_k \subseteq I$
- i-itemset: $I_k^i \subseteq I, |I_k^i| = i$
- transaction $T_k \subseteq I$
- data base: $D = \{(k, T_k) | k = 1, \dots, m\}$
- support of an itemset: share of transactions which contain the itemset

$$s(I_i) = \frac{|\{T_k | I_i \subseteq T_k\}|}{|D|}$$

- frequent (strong, large) itemset: $s(I_i) \geq s_{cut}$

Association Rules

- downward closure: every subset of a frequent itemset is also a frequent itemset



- every superset of a not frequent itemset is also a not frequent itemset

Association Rules

- association rule: $X \rightarrow Y$, $X, Y \subseteq I, Y \cap X = \emptyset$
- support of a rule: share of transactions which contain both, premise and conclusion of the rule

$$s(X \rightarrow Y) = s(X \cup Y) = \frac{|\{T_k | X \cup Y \subseteq T_k\}|}{|D|} = p(XY)$$

- confidence of a rule: share of transactions supporting the rule from those supporting the premise

$$c(X \rightarrow Y) = \frac{s(X \cup Y)}{s(X)} = \frac{|\{T_k | X \cup Y \subseteq T_k\}|}{|\{T_k | X \subseteq T_k\}|} = p(Y|X)$$

Association Rules

- strong rule: high support + high confidence
 - detection of strong rules: two pass algorithm
1. find frequent (strong, large) itemsets (Apriori)
 - necessary to generate rules with strong support
 - uses the downward closure
 - itemsets are ordered
 2. use the frequent itemsets to generate association rules
 - find strong correlations in a frequent itemset

Association Rules

- Apriori: finding frequent itemsets of increasing size
itemsets are ordered!
 - start with all itemsets of size one: I^1
 - select all itemsets with sufficient support
 - from the selected itemsets I^i generate larger itemsets I^{i+1}

$$is(\{i_1, \dots, i_{n-2}, i_{n-1}\}) \wedge is(\{i_1, \dots, i_{n-2}, i_n\})$$

$$\rightarrow is(\{i_1, \dots, i_{n-2}, i_{n-1}, i_n\})$$

- already blocks some of the non-frequent itemsets, but not all of them
- remove those itemsets which still contain a non-frequent immediate subset
 - they cannot have enough support (downward closure)
- continue until no further frequent itemsets can be generated

Association Rules

- example data base again
- assumption: threshold for the required support $s_{cut} = 0.5$

k	T_k	I_k^1	#	$s(I_k^1)$
001	{chips, cola, peanuts}	{chips}	3	0.75
002	{beer, chips, cigarettes}	{cola}	2	0.5
003	{beer, chips, cigarettes, cola}	{peanuts}	1	0.25
004	{beer, cigarettes}	{beer}	3	0.75
		{cigarettes}	3	0.75

- no non-empty subsets

Association Rules

- 2-itemsets I_k^2

I_k^1	#	$s(I_k^1)$	I_k^2	#	$s(I_k^2)$
{chips}	3	0.75	{chips, cola}	2	0.5
{cola}	2	0.5	{beer, chips}	2	0.5
{beer}	3	0.75	{chips, cigarettes}	2	0.5
{cigarettes}	3	0.75	{beer, cola}	1	0.25
			{cigarettes, cola}	1	0.25
			{beer, cigarettes}	3	0.75

- no itemsets to prune

Association Rules

- 3-itemsets I_k^3

I_k^2	#	$s(I_k^2)$	I_k^3	#	$s(I_k^3)$
{chips, cola}	2	0.5	{beer, chips, cigar.}	2	0.5
{beer, chips}	2	0.5	{chips, cigar., cola}	1	0.25
{chips, cigar.}	2	0.5			
{beer, cigar.}	3	0.75			

Association Rules

- resulting frequent itemsets:

{beer, chips, cigarettes}

{chips, cola}

{chips, beer}

{chips, cigar.}

{beer, cigar.}

{beer}

{chips}

{cigarettes}

{cola}

Association Rules

- generation of strong association rules:
 - for all frequent itemsets I_j determine all nonempty subsets I_k for which

$$c = \frac{s(I_j)}{s(I_k)} \geq c_{min}$$

- add a rule $I_k \rightarrow Y$, $Y = I_j - I_k$ to the rule set
- e.g. $s(\{chips\}) = 0.75$, $s(\{cola\}) = 0.5$,
 $s(\{chips, cola\}) = 0.5$

rule	confidence
$\{cola\} \rightarrow \{chips\}$	1.00
$\{chips\} \rightarrow \{cola\}$	0.67

Association Rules

- interesting association rules: only those for which the confidence is greater than the support of the conclusion

$$c(X \rightarrow Y) > s(Y)$$

- negative border:

$$\{I_k \mid s(I_k) < s_{cut} \wedge \forall I_j \subset I_k . s(I_j) \geq s_{cut}\}$$

used

- to compute the set of frequent itemsets more efficiently
- to derive negative association rules

Association Rules

- Apriori: number of potential itemsets is exponential in the number of items
- but:
 - data is sparse: $|T_i| \ll |I|$
 - itemsets are generated in separate scans of the data base
 - size of generated itemsets grows monotonically
 - large itemsets are useless
 - only k scans required ($k \ll |I|$)

Association Rules

- modifications / extensions
 - rule mining on relational data
 - Apriori for hierarchically organised items
 - 2-scan Apriori
 - sampled transactions
 - incremental rule mining
 - non uniform support thresholds
 - class association rules
 - (mining sequential data)

Relational Data

- relational data has to be transformed into transaction data
- Apriori requires categorical data → binning has to be performed
- the same category can appear as value of different attributes

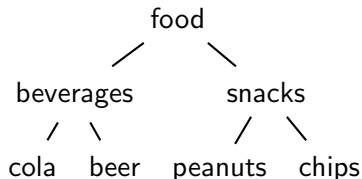
age	income	debt
low	low	low
middle	low	high
high	high	low

- values have to be combined with their attribute
 - attribute-value pairs are taken as items

1	(age, low)	(income, low)	(debt, low)
2	(age, middle)	(income, low)	(debt, high)
3	(age, high)	(income, high)	(debt, low)

Hierarchical Apriori

- in addition to the base level of items, determine also frequent itemsets on a higher level in an is-a hierarchy



- sometimes regularities can only be found at higher levels of abstraction

Partitioned Apriori

- Apriori requires several scans of the database
- can their number be reduced?
- partitioned Apriori: two scans
 - 1st scan: partition the database and compute locally frequent itemsets on the partitions
 - 2nd scan: determine the support of all locally frequent itemsets
 - heuristics: if an itemset is globally frequent it will be so locally in at least one partition
 - second scan deals with a superset of possible itemsets

Sampling the Data Base

- sampling requires multiple scans
 - 1st scan: take a sample and compute frequent itemsets
 - 2nd scan: count their support and the support for their immediate supersets
- if the itemset is at the negative border
 - all frequent itemsets have been found
 - else check supersets of the itemsets for being at the negative border in subsequent scans

Incremental Rule Mining

- incremental update: scan only the added transactions, whether they
 - invalidate a former frequent itemset, or
 - introduce new frequent itemsets

Non-uniform Support Thresholds

- items differ in their frequency of occurrence in the data base
- solution (1): using a single minimum support threshold s_{cut}
- problem: using a single threshold disfavors rare items
 - minimum support too high: itemsets containing rare items cannot be found
 - minimum support too low: too many itemsets are found (combinatorial explosion)
- associations with rare items might be particularly interesting

Non-uniform Support Thresholds

- solution (2):
 - assigning individual thresholds s_{cut}^i to the items
 - but preventing the generation of itemsets with extremely different thresholds

$$\max_{i \in I_k} s_{cut}^i - \min_{i \in I_k} s_{cut}^i < \delta$$

δ : global maximum support difference

- assigning a threshold $s_{cut}^i > 1$ excludes an item from consideration
 - can be used to guide the mining process towards the interesting items

Non-uniform Support Thresholds

- required support of an itemset has to be replaced by its minimal support:

$$s_{cut}(I_k) = \min_{i \in I_k} s_{cut}^i$$

- problem: downward closure property does not hold any longer
 - minimum support of an itemset is no longer monotonic

$$\neg(I_k \subset I_l \rightarrow s_{min}(I_k) \leq s_{min}(I_l))$$

- adding an item to an itemset might decrease the threshold for the minimum support
- an itemset might be frequent, while one of its subsets is not

Non-uniform Support Thresholds

- itemset generation
 - sort the items in the itemsets according to their minimum support values s_{cut}^i
 - a frequent itemset I_k needs not be extended by an item with a lower actual support $s^i < s_{min}(I_k)$
- rule generation
 - if $ab \rightarrow c$ is a rule with sufficient confidence $a \rightarrow bc$, or $b \rightarrow ac$ need not be
 - support values for all subsets of frequent itemsets have to be recorded

Class Association Rules

- so far: any item or combination of them can appear in the consequence part of a rule
- now: associations to a fixed target item required
- rules of the form

$$\{i_1, \dots, i_n\} \rightarrow c_j, i_i \in I, c_j \in C, C \cap I = \emptyset$$

- can be used used for classification
 - texts
 - search queries,
 - ...

Class Association Rules

- ruleitem: (condset,c)
- support and confidence defined as usual
 - support for the condition: $s_c = s(\text{condset})$
 - support for a rule: $s_r = s(\text{condset} \cup c)$
 - confidence of a rule: $c_r = s_r / s_c$
- similar algorithm: multiple scans with a growing number of items in the condition of the rule
- redundant rules: if a rule has a confidence of 1, each rule generated from it will also have a confidence of 1
 - should be avoided
- extension to relational data and non-uniform support thresholds possible

Data Mining Tasks

- Classification
- Prediction
- Clustering
- Dependency Modelling
- **Summarization**
- Change and Deviation Detection
- Visualization

Data Mining Tasks

- Classification
- Prediction
- Clustering
- Dependency Modelling
- Summarization
- Change and Deviation Detection
- Visualization

Temporal Data Bases

- snapshot databases: no support for temporal data
- transaction time databases: tuples or attribute values are timestamped when inserted
- valid time databases: tuples or attribute values can be annotated for the time range in which they are valid
- bitemporal databases: both types of temporal information are supported

Sequential Structures

- time is inherently sequential
- models for capturing sequential structures
 - Finite State Automata
 - Markov Models
 - Hidden Markov Models
- all require supervised training

Time Series Analysis

- event detection: classification based on a window of successive data points
- trend detection: smoothing by a moving average
- event prediction: classification based on preceding data points
- value prediction: fitting the coefficients of a (linear) equation
- (seasonal) cycle detection: autocorrelation
- outlier detection: not only global outliers but also outliers in a local context

Mining Sequential Patterns

- longest common subsequence
 - fraud detection
 - genomic analysis
 - failure prediction
 - disaster prediction (volcano eruptions, earthquakes, floodings)
- for categorical data: extension of Apriori to sequences
- flexible match required
 - extension of the similarity measures to sequences, e.g. LEVENSHTEIN-metric (elastic match, dynamic time warping)
 - general case: match with transpositions
- for numerical data: (Hidden) Markov Models

Mining Sequential Patterns

- in some applications the order of items is relevant: click streams, natural language text, repeated shopping
- extension of Apriori to sequences of itemsets

$$\langle e_1, \dots, e_n \rangle, \text{ with } e_i \subseteq I$$

Wei Wang and Jiong Yang: Mining sequential patterns from large data sets, Springer, New York 2005.

- items in the elements are lexicographically ordered (as in Apriori)

Mining Sequential Patterns

- size of a sequence: number of itemsets it contains

$$\text{size}(s) = |s|$$

$$\text{size}(\langle\{2\}, \{3, 5\}, \{1, 4\}\rangle) = 3$$

- length of a sequence: number of elements in the sequence

$$\text{length}(s) = \sum_{i=1}^n e_i, \text{ with } n = \text{size}(s)$$

$$\text{length}(\langle\{2\}, \{3, 5\}, \{1, 4\}\rangle) = 5$$

Mining Sequential Patterns

- subsequence/supersequence:

$s_1 = \langle a_1, a_2, \dots, a_n \rangle$ is a subsequence of $s_2 = \langle b_1, b_2, \dots, b_m \rangle$ (s_2 contains s_1),

if there exists integers $1 \leq j_1 \leq j_2 \leq \dots \leq j_o \leq m$

so that $a_1 \subseteq b_{j_1}, a_2 \subseteq b_{j_2}, \dots, a_n \subseteq b_{j_o}$

- support of a sequence: fraction of data sequences which contain the sequence

Mining Sequential Patterns

customer	date	transaction
1	2012/06/13	30
1	2012/06/19	90
2	2012/06/03	10, 20
2	2012/06/09	30
2	2012/06/16	10, 40, 60, 70
3	2012/06/16	30, 50, 70, 80
4	2012/06/05	30
4	2012/06/05	30, 40, 70, 80
4	2012/06/05	90
5	2012/06/19	90

customer	transaction sequence
1	$\langle \{30\}, \{90\} \rangle$
2	$\langle \{10, 20\}, \{30\}, \{10, 40, 60, 70\} \rangle$
3	$\langle \{30, 50, 70, 80\} \rangle$
4	$\langle \{30\}, \{30, 40, 70, 80\}, \{90\} \rangle$
5	$\langle \{90\} \rangle$

Mining Sequential Patterns

customer	transaction sequence
1	$\langle\{30\}, \{90\}\rangle$
2	$\langle\{10, 20\}, \{30\}, \{10, 40, 60, 70\}\rangle$
3	$\langle\{30, 50, 70, 80\}\rangle$
4	$\langle\{30\}, \{30, 40, 70, 80\}, \{90\}\rangle$
5	$\langle\{90\}\rangle$

length	sequential patterns with $s \geq 0.25$
1	$\langle\{30\}\rangle, \langle\{40\}\rangle, \langle\{70\}\rangle, \langle\{80\}\rangle, \langle\{90\}\rangle$
2	$\langle\{30\}, \{40\}\rangle, \langle\{30\}, \{70\}\rangle, \langle\{30\}, \{90\}\rangle, \langle\{30, 70\}\rangle, \langle\{30, 80\}\rangle, \langle\{40, 70\}\rangle, \langle\{70, 80\}\rangle$
3	$\langle\{30\}, \{40, 70\}\rangle, \langle\{30, 70, 80\}\rangle$

Mining Sequential Patterns

- can be extended to non-uniform minimum support
- sequential rules: $X \rightarrow Y$ with Y being a sequence and X a proper subsequence of Y
- useful rules
 - rules with wildcard symbols (label sequence rules)
 - wildcard symbols usually increase the confidence for a sequence
 - predict unseen/missing elements in a data sequence

$$\langle \{1\}, \{*\}, \{7, *\} \rangle \rightarrow \langle \{1\}, \{3\}, \{7, 8\} \rangle$$

- class sequential rules

Data Mining Tasks

- Classification
- Prediction
- Clustering
- Dependency Modelling
- Summarization
- Change and Deviation Detection
- **Visualization**

Visualization

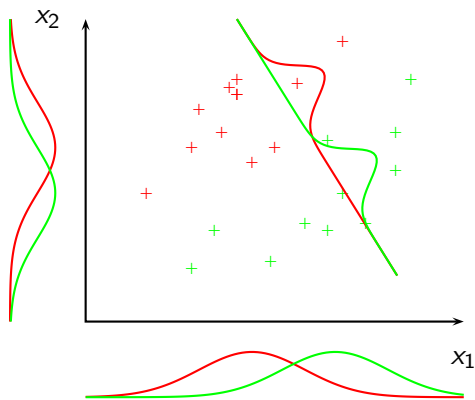
- seeing is the construction of a mental image
 - abstraction: identification of objects, assigning properties
 - generalization: summarized information about many data points
- basic graph types
 - bar charts
 - histograms (distributions)
 - line charts
 - pie charts
 - scatter plots

Visualization

- problem: limited dimensionality
 - two (three) basic dimensions
 - overlay of multiple graphs
 - color
 - texture
 - shape
 - animation
- combination of visualisation techniques with data cube operations
- interactive exploration of data: browsing

Visualization

- rolling the dice is not always sufficient

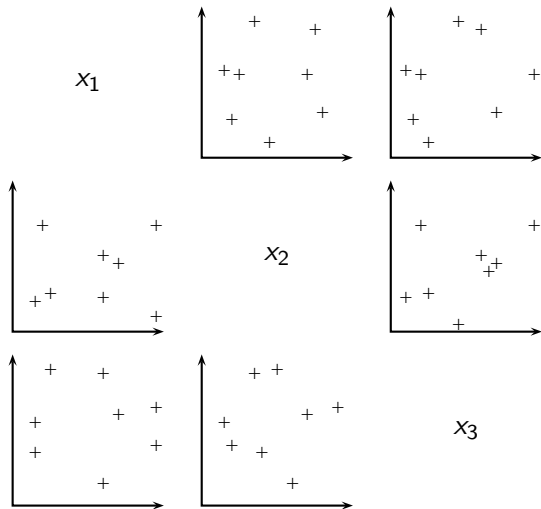


Multi-Dimensional Visualization

- scatter-plot matrix
- parameter stacks
- parallel coordinates
- star display
- radial visualization

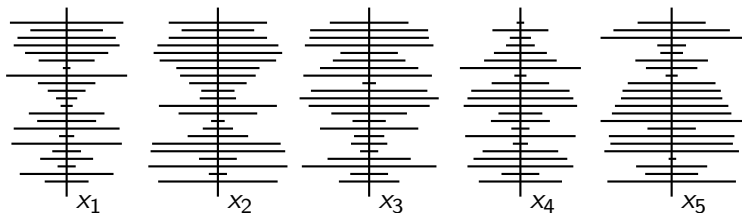
Scatter-Plot Matrix

- $n \times n$ -matrix of all combinations of two dimensions

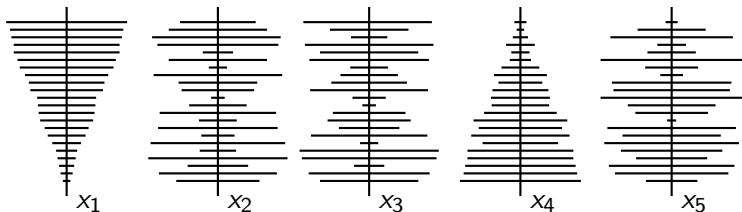


Parameter Stacks

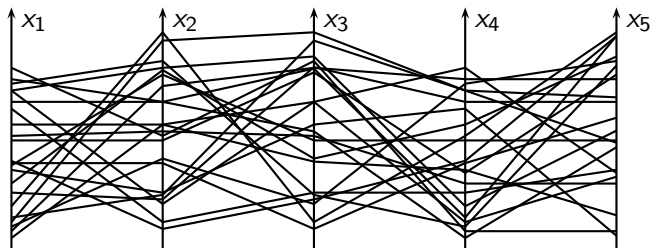
- data plots on vertical lines as centered horizontal lines



- exploring data by sorting along a dimension



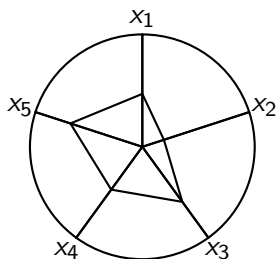
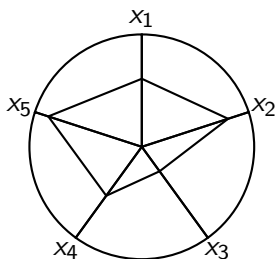
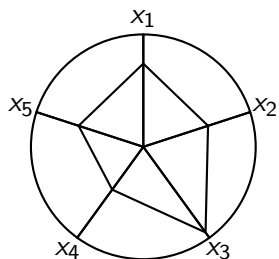
Parallel Coordinates



- exploring data by investigating neighborhood relationships between dimensions
 - rearranging

Star Display

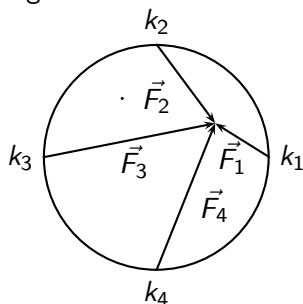
- radial version of parallel coordinates



- only for the display of few data points

Radial Visualization

- attraction-based: forces proportional to the n dimensions pull the point towards the dimension anchors
- equilibrium: forces must sum up to 0
- mapping the n -dimensional space into a two dimensional one
 $(k_1, k_2, k_3, k_4, \dots, k_n) \mapsto (x, y)$
- e.g. $n = 4$



$$\vec{F}_1 + \vec{F}_2 + \vec{F}_3 + \vec{F}_4 = 0$$

Radial Visualisation

$$k_1 \begin{pmatrix} 1-x \\ 0-y \end{pmatrix} + k_2 \begin{pmatrix} 0-x \\ 1-y \end{pmatrix} + k_3 \begin{pmatrix} -1-x \\ 0-y \end{pmatrix} + k_4 \begin{pmatrix} 0-x \\ -1-y \end{pmatrix} = 0$$

$$k_1 - k_1 \cdot x - k_2 \cdot x - k_3 - k_3 \cdot x - k_4 \cdot x = 0$$

$$-k_1 \cdot y + k_2 - k_2 \cdot y - k_3 \cdot y - k_4 - k_4 \cdot x = 0$$

$$k_1 - k_3 - x(k_1 + k_2 + k_3 + k_4) = 0$$

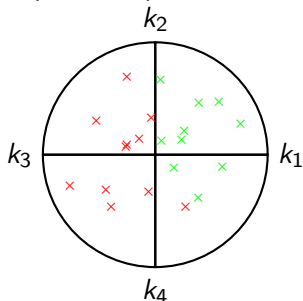
$$k_2 - k_4 - y(k_1 + k_2 + k_3 + k_4) = 0$$

$$x = \frac{k_1 - k_3}{k_1 + k_2 + k_3 + k_4}$$

$$y = \frac{k_2 - k_4}{k_1 + k_2 + k_3 + k_4}$$

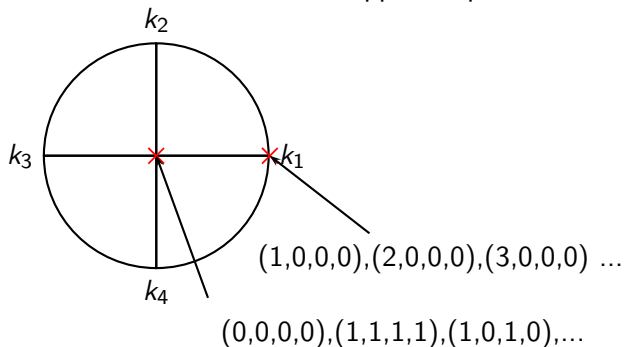
Radial Visualisation

- important spacial relationships are preserved: e.g. class separation



Radial Visualisation

- information loss: lines are mapped to points



Sonification

- hearing data
- auditory channel is inherently multidimensional
 - volume, rhythm, pitch, harmony, polyphony, sound color, ...
- approaches
 - audification
 - sound mapping
 - model-based sonification

Sonification

- audification: direct mapping of time-series data to sound patterns
 - detection of rhythmic patterns
 - traffic density
- sound mapping: controlling sound synthesis parameter by data items
 - high-dimensional data can be presented
- model-based sonification: excitation of an oscillating model by data items
 - energetically coupled particles, growing neural gas
 - interactive exploration of the (auditory) system response
 - linear structures in a high-dimensional space can be identified