Knowledge Discovery in Databases II

Lecture 1 – Introduction

Prof. Dr. Peer Kröger, Yifeng Lu

Sommer Semester 2019

Credits:

Based on material of Eirini Ntoutsi, Matthias Schubert,

Arthur Zimek, Peer Kröger, Yifeng Lu

Organization of the Course

- Lecture: Wed, 09:00-11:30, room B 001 (Oettingenstr. 67)
- Tutorials:
  - Mon, 14:00-16:00, 16:00-18:00
  - Tue, 14:00-16:00, 16:00-18:00
- All information and news can be found at:
  `http://www.dbs.ifi.lmu.de/cms/studium_lehre/lehre_master/kdd2181/`
- Exam:
  - Written exam, 90 min
  - 6 ECTS points
  - Registration for the written exam through UniWorX

Table of Contents

Kapitel 1: BuzzWord Bingo i

Data Everywhere

Problem

We are drowning in data, yet starving for knowledge.

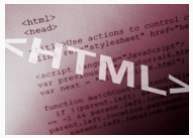
(`http://www.kdnuggets.com/news/2007/n06/3i.html`)

- Large amounts of data in multiple applications
- Manual analysis is infeasible

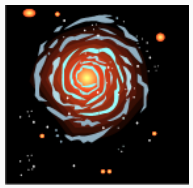
Solutions

- Descriptive modeling: explains the characteristics and behavior of observed data
- Predictive modeling: predicts the behavior of new data based on some model

connection data



web data



telescope data

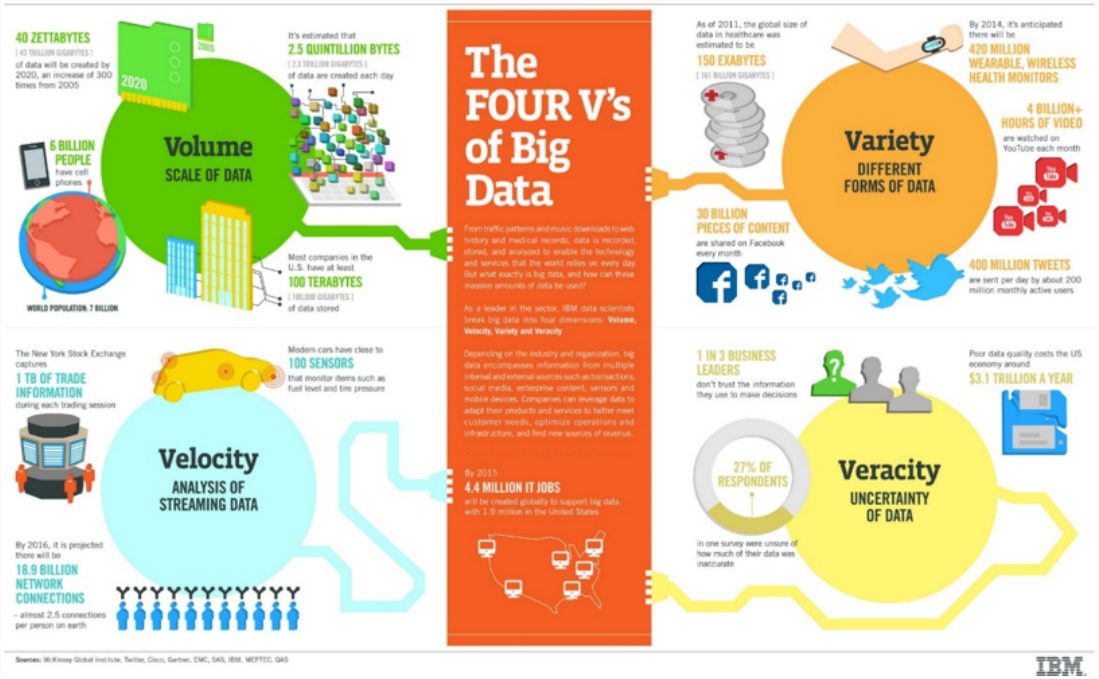
transaction data

...

Play the BuzzWord Bingo: Big Data

Big Data

- Term triggered by McKinsey 2011, characterized by some V's
- BIG vs. VERY LARGE $\Rightarrow$ More a Data Engineering task
- Related: Industry 4.0, Data Lake, ...

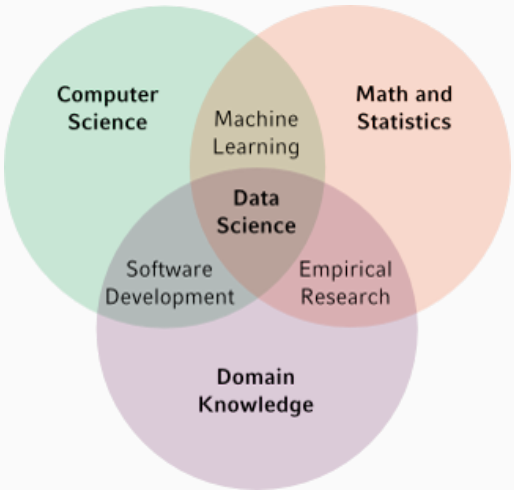


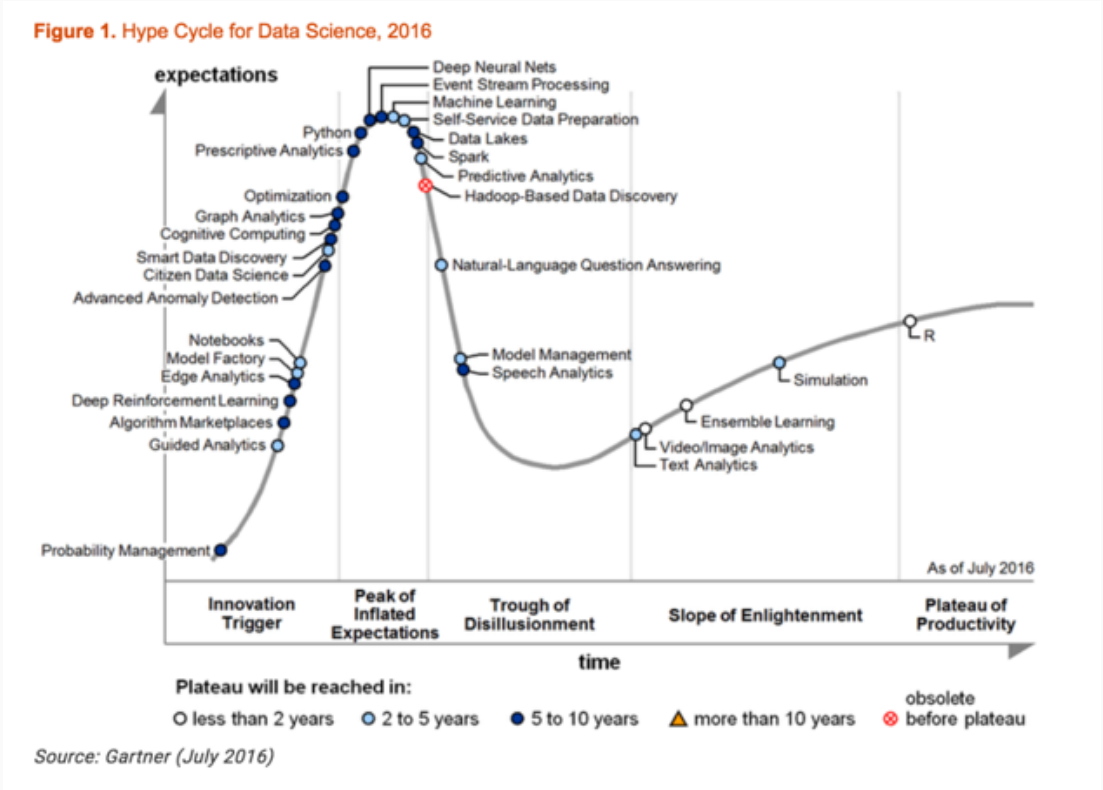
Picture from IBM: Big Data characterized by

- Volume
- Velocity
- Variety
- Varecity

Data Science

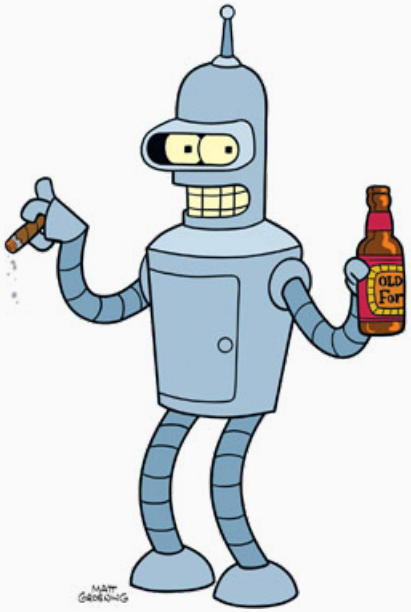
- Often considered as a more general process: not only collect data but also do gain value from data





Artificial Intelligence

- AI is an extremely broad subject within CS, including reasoning, problem solving, knowledge representation, planning, (machine) learning, natural language processing, perception, motion and manipulation, social intelligence, creativity, general intelligence ... (to name a few)
- There is a major overlap to Machine Learning and Data Analytics
- In Germany: AI is often (mis-)used for deductive learning approaches in the context of Robotics



Source: Wikipedia

Deductive Learning

- Use a set of facts and rules (knowledge base) to derive new facts with logic inference (deduction)
- The learning is from general to specific facts
- Example:
  - Facts: (1) Kröger is German; (2) all Germans have no sense of humor
  - Derived fact: Kröger has no sense of humor
- If the knowledge base is true, the derived facts are true, too
- No real predictive power

Machine Learning uses an inductive learning approach

- Use a set of (individual) observations to learn general facts
- Example:
  - Observations:
    (1) Kröger is German and has no sense of humor
    (2) Schmidt is German and has no sense of humor
    (3) Meier is German and has no sense of humor
    (4) Blatter is Swiss and has a sense of humor
  - Learned: Germans have no sense of humor
- Since we usually have not all possible observations, the derived rules are probably not 100% true
- Predictive power: Müller is German $\Rightarrow$ has no sense of humor
- ML vs Data Mining: modelling vs. algorithmic approach

So what is KDD?

Definition: Knowledge Discovery in Databases

Knowledge Discovery in Databases (KDD) is the nontrivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data. (Fayyad, Piatetsky-Shapiro, and Smyth 1996)

Remarks

- Nontrivial: it is not just the obvious things
- Valid: patterns should also hold for previously unseen instances
- Novel: at least to the system and preferable to the user
- Potentially useful: to create some benefit to the user or task
- Ultimately understandable: for the end user

KDD is an Iterative Process

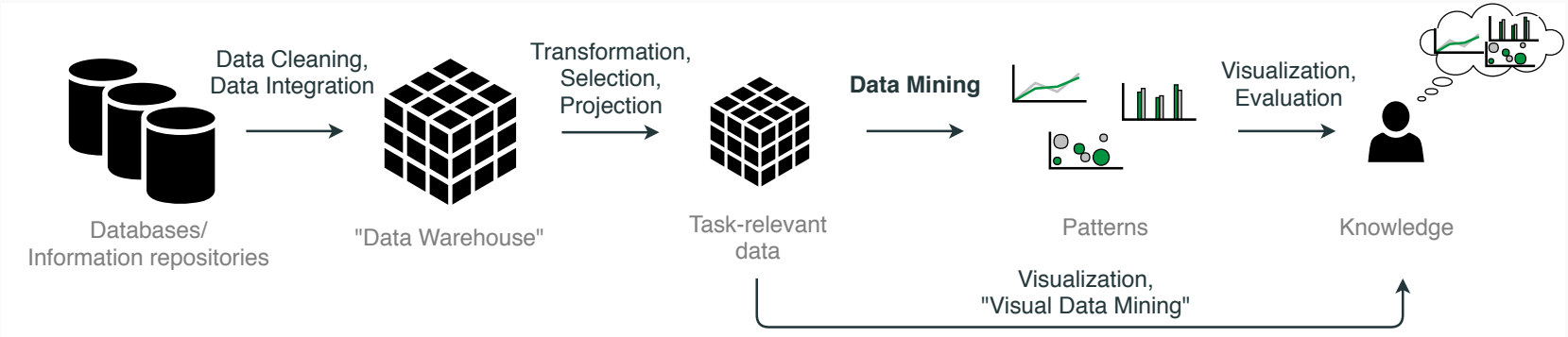
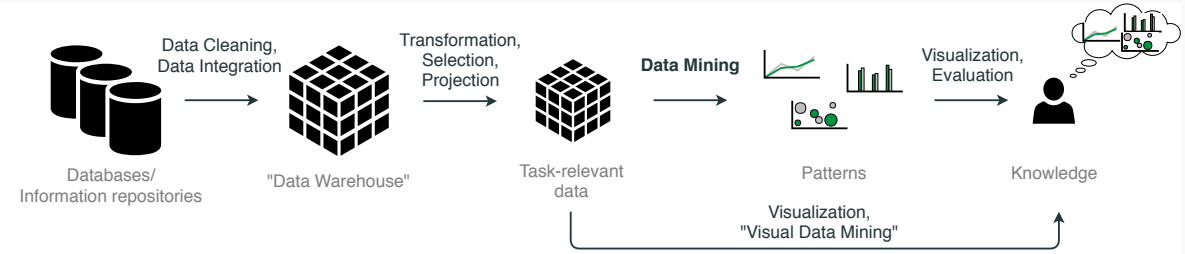


Figure: KDD-process following (Fayyad, Piatetsky-Shapiro, and Smyth 1996).

Remarks

- The process may have back-loops!!!
- Many very similar variants (e.g. CRISP-DM) exist
- First two steps (preprocessing) usually take $>$ 85% of the time/ressources

Tasks During Preprocessing



Cleaning and Integration

- Identify and select the relevant data sources
- Integrate data
- Increase data quality (noise, missing values, inconsistencies, ...)

Transformation

- Select useful features and objects
- Derive new features
- Feature transformation/ discretization
- Dimensionality reduction

Kapitel 2: Recap i

Kapitel 2: Recap ii

4. Recommended Literature

Kapitel 2: Recap

Overview of Major KDD I Topics

- Unsupervised Methods
  - Clustering
  - Outlier Detection
  - Assosiation rule mining and frequent pattern mining
- Supervised Methods
  - Classification
  - Regression

Observation

Most of the methods covered by KDD I assume the data to be a set of feature vectors.

What are Feature Vectors?

Feature Vector/Feature Space

- A domain (set of objects) *Dom*
- A similarity measure or (more often) a distance measure *dist*

Metric Space

- Distance measure *dist* is reflexive, positive definite, symmetric, and meets triangle inequality

The Canonical Metric Feature Space

- $Dom = \mathbb{R}^d$, i.e., objects are points in a d-dimensional space
- $dist(x, y) = \sqrt{\sum_{1=1}^{d}(x_i - y_i)^2}$ is the Euclidean distance

How to Derive Feature Vectors for Complex Objects?

Explicit Modeling (Similarity Modeling, Feature Transformation)

- Define features manually (see next slide)
- Requires a lot of engineering but always works (in principle)

Implicit Modeling

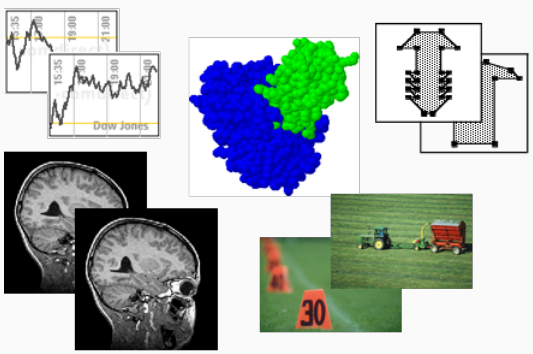
- Use a Kernel e.g. a shortest-path kernel for graphs, etc.
- Works only if a Kernel trick can be applied

Feature Learning (Representation Learning)

- Learn the representation through e.g. deep learning
- Usually works only for supervised settings and unstructured data
- Attention: NNs need a lot of training data!!!

The General Concept of FT

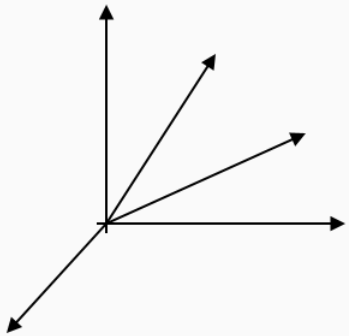
- Extract characteristic (usually *numeric*) features from each object
- Each object is represented as a high-dimensional (feature) vector (in case of numeric features: points)
- Characteristic features: similar vectors indicate similar objects



Data Space

Feature Transformation

Histogramms
Moment Invariants
Covering
Sectoring
Fourier Transformation
...

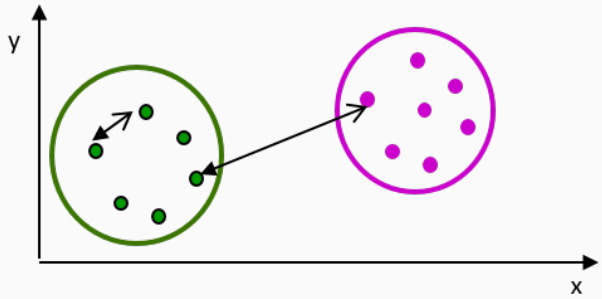


Feature Space

Kapitel 2: Recap

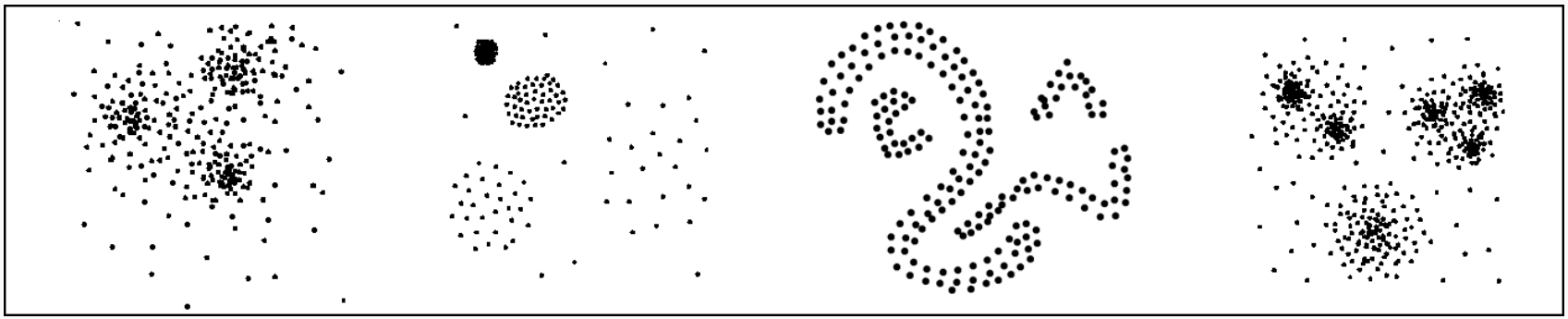
Patterns = Clusters

Cluster = group of similar objects
E.g. minimize intra-cluster distances
and maximize inter-cluster distances



Remarks

- Similarity/distance measure is important
- Unsupervised: how do the patterns look like???

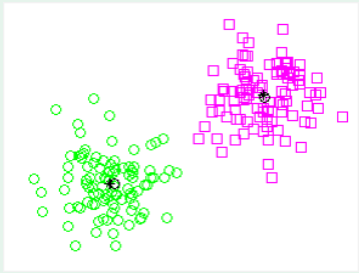


General Considerations

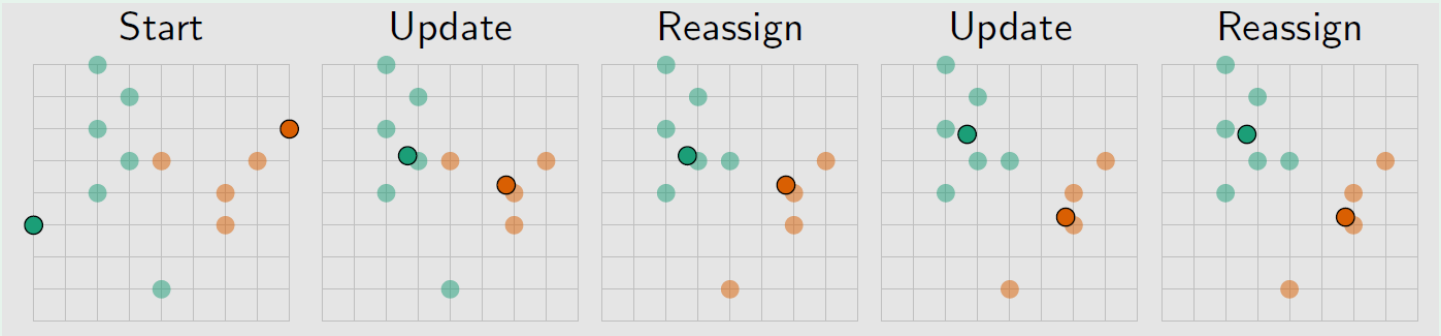
- Unsupervised: we have no clue about the characteristics of the clusters, i.e. the patterns
  - How many are there?
  - What data distribution do they have?
  - What shape do they have?
  - Is there noise?
- Clustering methods usually have to make a lot of assumptions regarding these questions
- Typically, a particular method assumes a given pattern (its cluster model) which is defined implicitly or by its input parameters, or both
- Then, the method searches for these patterns (and often nothing else)

Partitioning Methods

- Construct k (usually spherical) partitions that are optimal w.r.t. some cluster criterion, e.g. compactness
- Popular methods: k-Means, k-Medoid

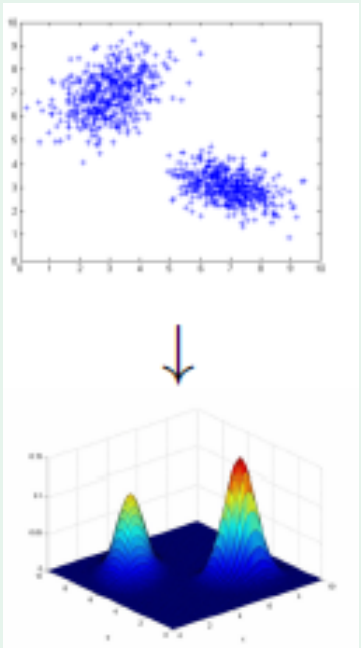


- Important concepts:
  - Cluster representative (mean, medoid, median, mode, ...)
  - Cluster criterion (sum of squared distances to representatives, ...)
- Basic algorithmic schema:



Probabilistic (Model-based) Methods

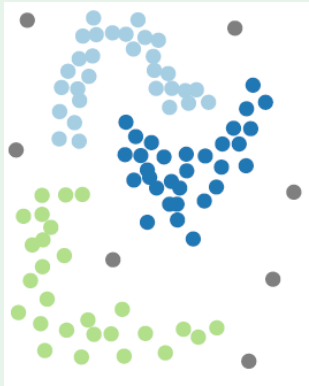
- Fit k probability distributions ("models" — usually Gaussians) to the data
- Optimize log-likelihood of points being generated by these k distributions
- Generates a fuzzy clustering (each point belongs to all clusters each with a certain probability)
- Popular methods: Expectation Maximization (EM)



- Basic algorithmic schema similar to partitioning methods (see above — but usually converge slower)

Density-based Methods

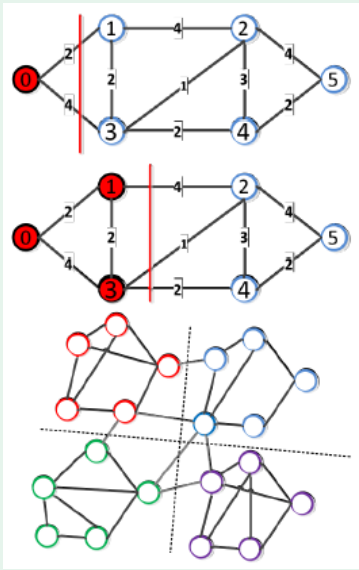
- Clusters are regions of high point density separated by regions of low point density
- Key concept: definition of density, e.g.
  - Minimum number of points in a volume
  - Probability density w.r.t. a certain type of distribution
  - Common mode within a volume



- Various algorithmic schemata based on different concepts of volumes, density test, and connectivity
- Popular methods: Grid-Clustering, DBSCAN, DenClu, Mean-Shift

Spectral Methods

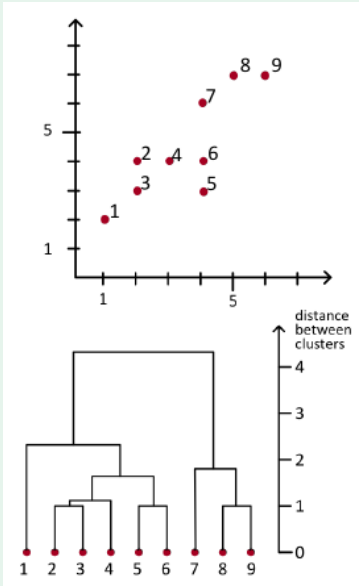
- Data is modeled by a similarity graph G
- Partitioning the adjacency matrix of G
- Problem is generally NP-hard, thus, use approximations/simplifications instead, e.g. partition G into k subsets, minimizing a function of the edge weights between/within the partitions
- Resulting clusters may have arbitrary shape



- Basic algorithmic schema: represent each vertex by a vector of its corresponding components in the eigenvectors and apply k-Means on this representation

Hierarchical Methods

- Hierarchical decomposition of data represented e.g. as a tree (dendrogram)
- Close relationship to minimum spanning tree problem
- Key concept: similarity measures between clusters (e.g. single link)
- Popular methods: Single-/Complete-/Average-Linkage, OPTICS

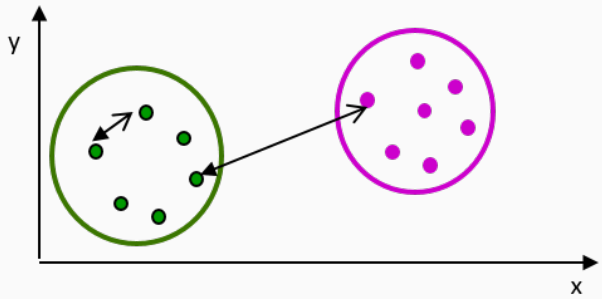


- Basic algorithmic schema: merge the most similar clusters starting from singleton clusters until all objects are in one cluster

Kapitel 2: Recap

Patterns = Classes

Given a set of labeled objects (training data) derive a model for each label (class) that can be used to predict the label of new data objects



Remarks

- Supervised: we already know how many patterns (classes) and some characteristics of these patterns (from the training data)
- But still, we need to make some assumptions about how to model these characteristics and how to differentiate between classes

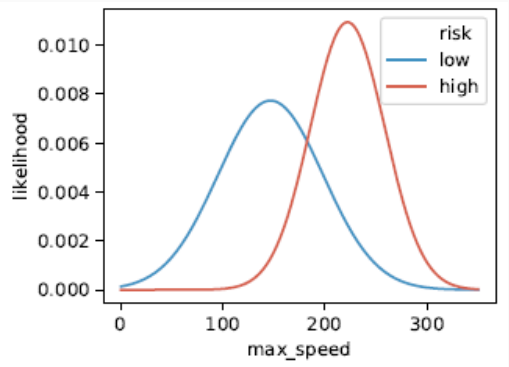
Basics

- Usually, 2-step approach (except lazy learners)
  - Training phase: building the model/classifier from the training data
  - Prediction phase: applying the classifier to unknown data
- Classifier should generalize rather than overfit to the training data (worst case: memorization)
- Apply train-and-test for smaller training sample sizes
- Ensembles: combine several (weak) classifiers in order to improve accuracy

Classification Paradigms

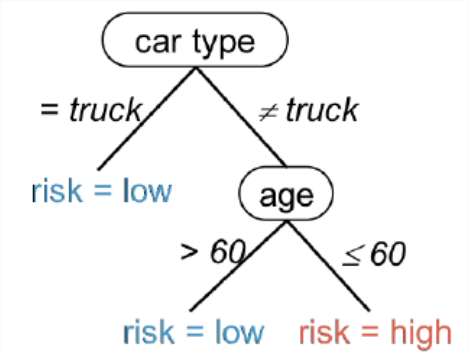
Bayes classifiers

- Model classes as probability distributions
- Prediction: maximum likelihood
- Attribute independence assumption (Naive Bayes)



Trees

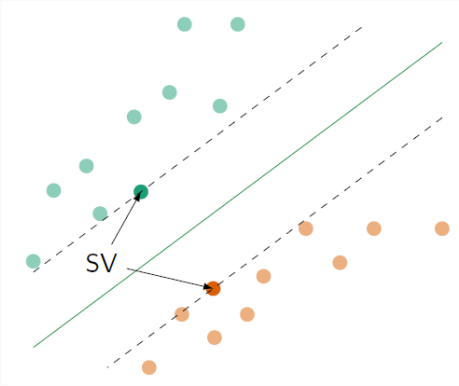
- Partitioning along attributes
- Purity measures (IG, Entropy)
- Attribute independence assumption
- Avoid overfitting by using Forests (ensemble of trees)



Classification Paradigms

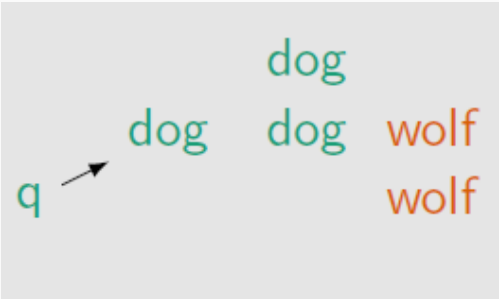
Linear Models

- Find a linear separation of classes (e.g. maximum margin hyperplane in SVMs)
- SVM: soft margin and Kernels for non-linear problems



Nearest Neighbor

- Instance-based learning: use similarities/distances distances to training objects
- Class labels of (k) nearest neighbors determine the prediction

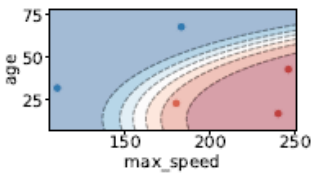


Discussion

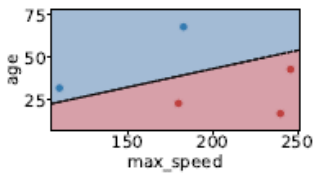
- There are many more classification models (e.g. Neural Nets, ...)
- Models basically differ in the shape of the decision boundaries

Example:

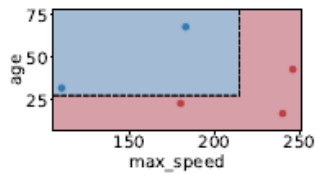
age	car_type	max_speed	risk
23	family	180	high
17	sportive	240	high
43	sportive	246	high
68	family	183	low
32	truck	110	low



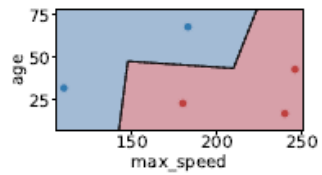
Bayes classifier



Linear classifier



Decision tree

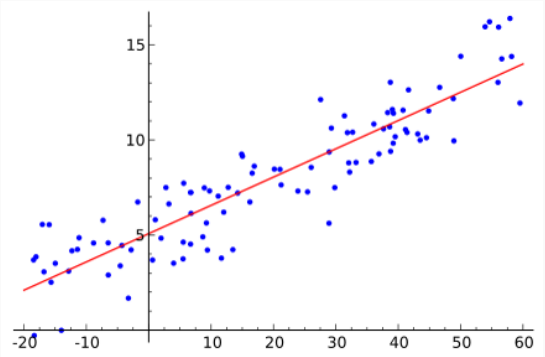


k-NN classifier

Kapitel 2: Recap

Patterns = Input/Output Relationship

- Similar to classification but the output is continuous (rather than categorical)
- Task: model the relationship between input (the features of the objects) an output (the prediction value)



Remarks

- Again, we need to make some assumptions about the type of relationship between input and output (e.g. linear, ...)
- Popular methods: linear regression, polynomial regression, piece-wise linear regression, logistic regression (categorical output)

Kapitel 2: Recap

Frequent Patterns

Patterns = Frequently Occurring Objects

- Determine the objects that appear frequently in the data (exceeds a threshold)
- Objects: values/value combinations (items/item sets), substructures, ...

Tid	Transaction items
1	Butter, Bread, Milk, Sugar
2	Butter, Flour, Milk, Sugar
3	Butter, Eggs, Milk, Salt
4	Eggs
5	Butter, Flour, Milk, Salt, Sugar

Remarks

- Simplest kind of patterns: counting frequency
- Search space grows exponential with increasing number of objects
- Different basic search strategies based on monotonicity of counts: depth-first (e.g. FP-Growth) and breadth-first (Apriori)

Association Rules

Patterns = Frequent IF-THEN-rules

- Find all association rules $X \rightarrow Y$ w.r.t. minimum support and minimum confidence thresholds,
  e.g. "*buys diapers* $\rightarrow$ *buys beer*"

Remarks

- Association rules can be derived from frequent items
- Attention to negatively correlated item sets

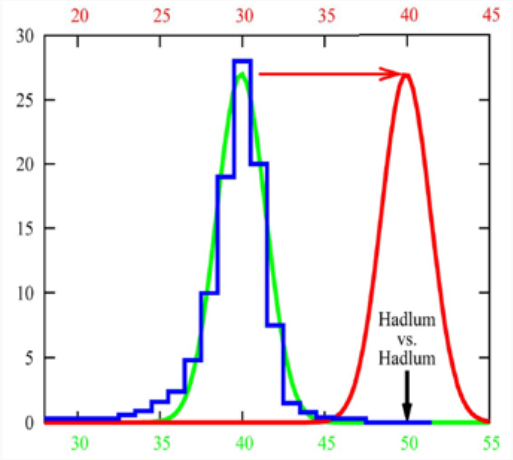
Kapitel 2: Recap

Outlier Detection

Patterns = Abnormal Objects

An outlier is an observation which deviates so much from the other observations as to arouse suspicions that it was generated by a different mechanism

(Hawkins 1980)



Remarks

- Outlier detection is usually unsupervised, i.e. no information on the characteristics of the outlier process(es) is available
- Many different approaches based on varying assumptions about these characteristics

Kapitel 3: Topics i

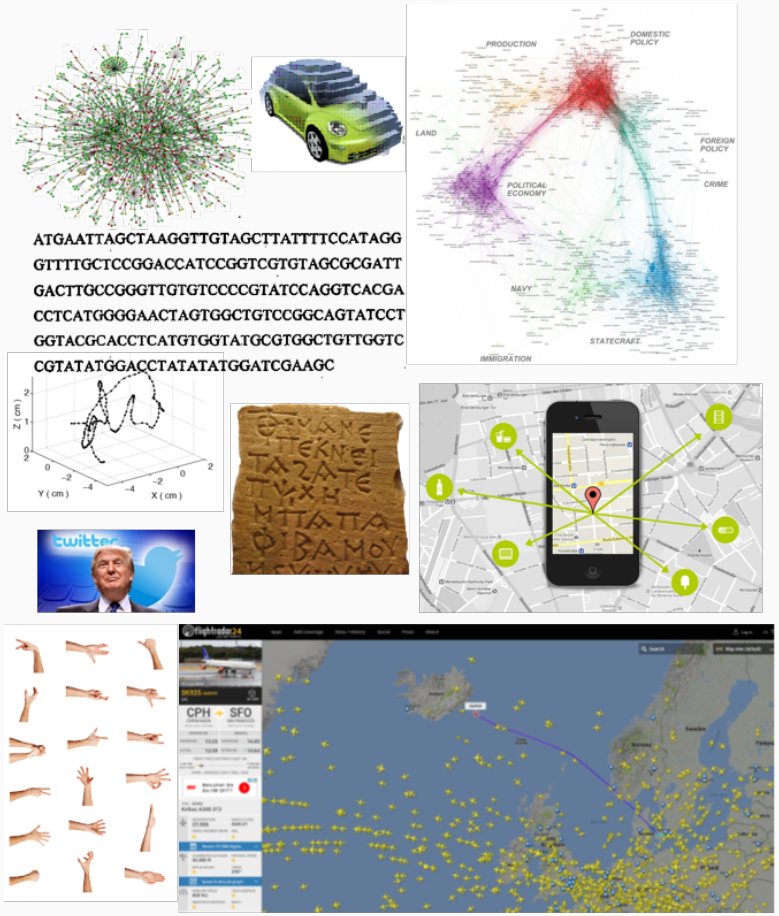
KDD I versus KDD II

- In KDD I, we focus on how to solve specific data mining tasks
- Observations:
  - Almost all methods work on feature vectors (only)
  - Similarity / distance measures play a key role in various data mining tasks
  - However, only simple distance functions were introduced
- In real world applications, useful information are hidden in data with different forms
  - Suitable Feature Transformation / Kernel not easy to find
  - Feature Transformation / Kernel is a simple model that might loose object semantics (compare: relational vs. object model, table vs. graphs, . . .)
  - Representation Learning not applicable
- KDD II is mostly on how to handle different types of data

KDD I versus KDD II

Data Complexity:

- High-dimensional
- Time series
- Sequences
- Spatial-temporal data
- Graphs
- Text
- Shapes
- ...



Data can often no longer be modeled by plain tables (feature vectors) without loosing important semantics ...

Kapitel 4: Literature i

Text Books for Further reading

- Han J., Kamber M., Pei J.: Data Mining: Concepts and Techniques. 3rd ed., Morgan Kaufmann, 2011
- Tan P.-N., Steinbach M., Kumar V.: Introduction to Data Mining. Addison-Wesley, 2006
- Mitchell T. M. (English): Machine Learning. McGraw-Hill, 1997
- Lescovec J, Rajaraman A., Ulman J.: Mining of Massive Data Sets. Cambridge University Press, 2014
- Ester M., Sander J. (German): Knowledge Discovery in Databases: Techniken und Anwendungen. Springer Verlag, September 2000
- C. M. Bishop: Pattern Recognition and Machine Learning. Springer 2007.
- R. O. Duda, P. E. Hart, and D. G. Stork: Pattern Classification. 2ed., Wiley-Inter-science, 2001.