Preprocessing

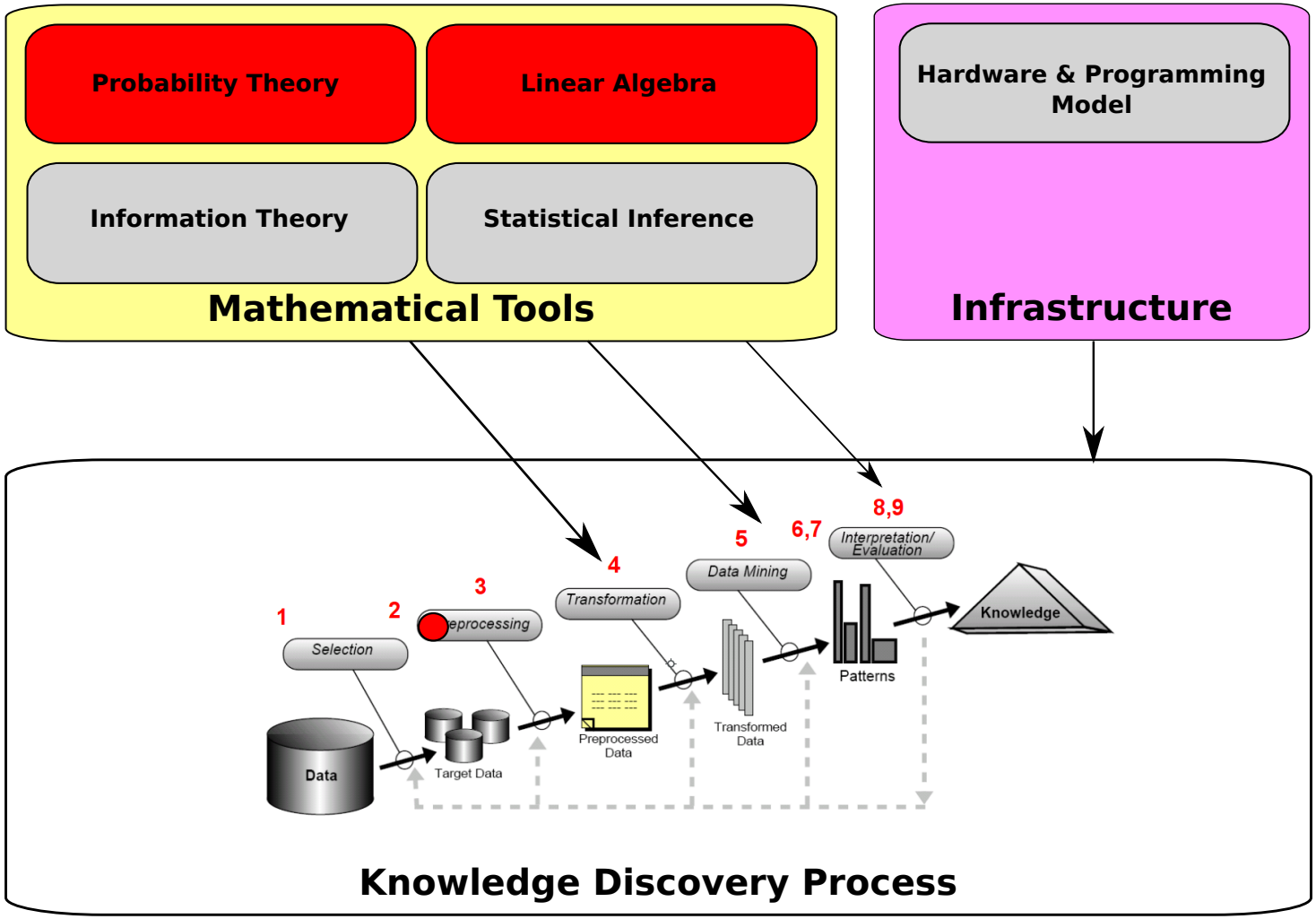
Knowledge Discovery and Data Mining 1

Roman Kern

KTI, TU Graz

2015-10-08

Big picture: KDDM



Outline

Introduction

Data acquisition & pre-processing

Introduction

- Initial phase of the Knowledge Discovery process
- ... acquire the data to be analysed
- e.g. by **crawling** the data from the Web
- ... prepare the data
- e.g. by **cleaning** and **removing outliers**

Web Crawling

Acquire data from the Web

Motivation for Web crawling

- Example:
- **Question**: How does a search engine know that all these pages contain the query terms?
- **Answer**: Because all of those pages have been crawled!

Motivation for Web crawling

Use Cases

- General web search engines (e.g. Google, Yandex, ...)
- Vertical search engines (e.g. Yelp)
- Business Intelligence
- Online Reputation Management
- Data set generation

Names for Web crawling

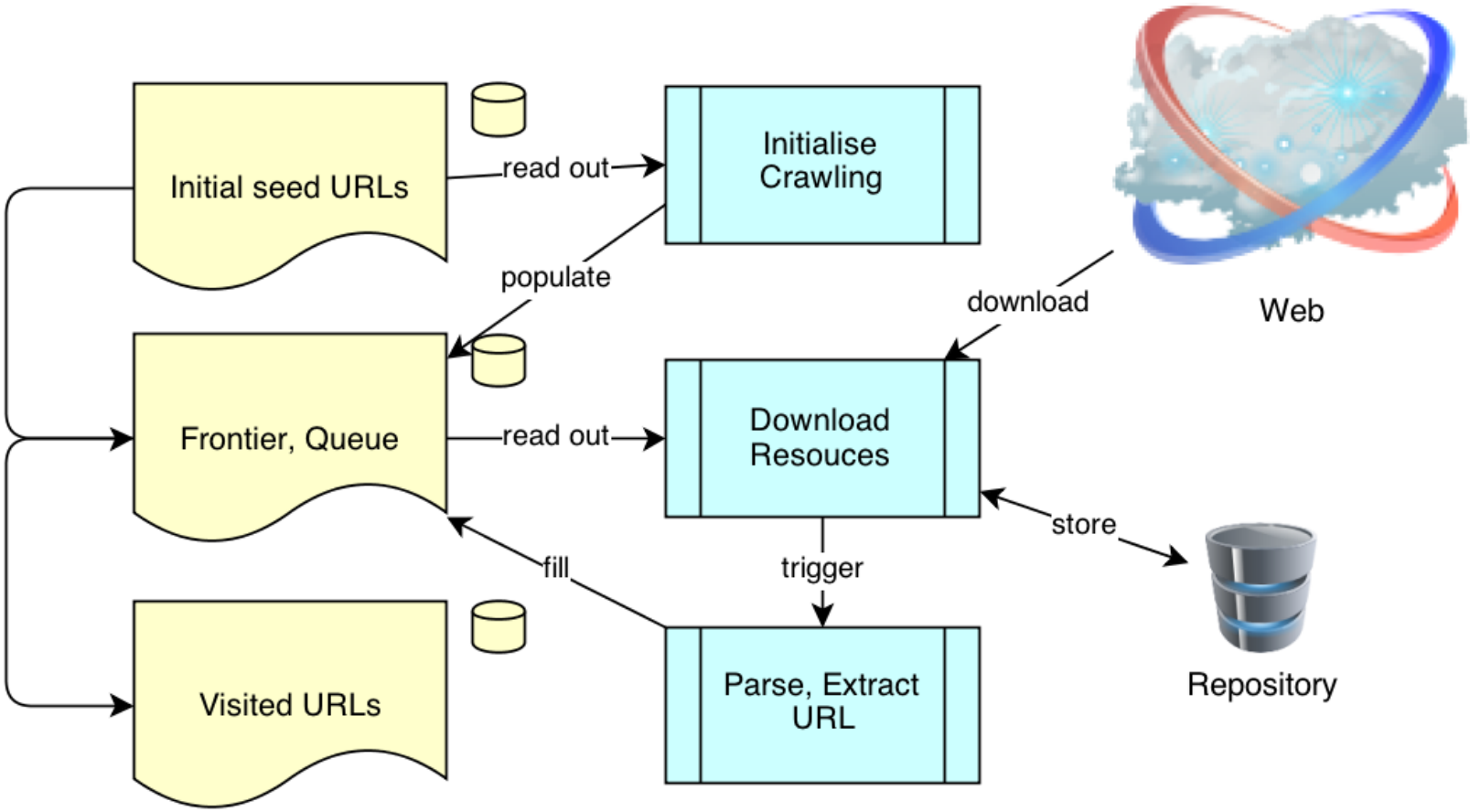
- A web crawler is a specialised Web client
- ... that uses the HTTP protocol
- There are many different names for Web crawlers:
- Crawler, Spider, Robot (or bot), Web agent, Web scutter, ...
- Wanderer, worm, ant, automatic indexer, scraper, ...
- Well known instances: googlebot, scooter, slurp, msnbot,
- Many libraries, e.g. Heretrix

Simple Web crawling schema

Basic Idea

- The crawler starts at an initial web page
- The web page is downloaded and its content gets analysed
- ... typically the web page will be HTML
- The links within the web page are being extracted
- All the links are candidates for the next web pages to be crawled

Simple Web crawling schema



Types of crawlers

- Batch crawler - snapshot of the current state, typically until a certain threshold is reached
- Incremental crawler - revisiting URLs to keep up to date
- Specialised crawlers, e.g focused crawler, topical crawler

Challenges of web crawling

- The large volume of the Web
- The volatility of the Web, e.g. many Web pages change frequently
- Dynamic Web pages, which are "rendered" in the client
- ... including dynamically generated URLs

Challenges of web crawling (cont.)

- Avoid crawling the same resources multiple times, e.g. normalise/canonicalise URLs
- Cope with errors in downloading, e.g. slow, unreliable connections
- Detect redirect loops
- Memory consumption, e.g. large frontier

Challenges of web crawling (cont.)

- Deal with many content types, e.g. HTML, PDF, Flash...
- Gracefully parse invalid content, e.g. missing closing tags in HTML
- Identify the structure of Web pages, e.g. main text, navigation, ...

Web crawling

Extract structured information

- Usually the information is embedded in HTML tailored towards being displayed
- ... but crawlers would prefer to have the data already in a structured way
- $\rightarrow$ **Semantic Web**, highly structured, little uptake
- $\rightarrow$ **Microformats**, less structured, but more uptake

Web crawling & semantic web

- The "Semantic Web" should aid the process of Web crawling
- As it is targeted at making the Web machine readable
- Web pages expose their content typically as RDF (Resource Description Language)
- ... instead of the human readable HTML, e.g. depending on the User Agent
- $\rightarrow$ specialised crawlers for the Semantic Web

Microformats

- Microformats as a lightweight alternative to the "Semantic Web"
- Embedded as HTML markup
- Supported by the major search engines
- http://microformats.org
- e.g. All 1.6+ billion OpenStreetMap nodes have a geo microformat

Example: Taken from openstreetmap.org

```
<div class="geo">
    <a href="/?lat=47.0591997&amp;lon=15.4632963&amp;zoom=18">
        <span class="latitude">47.0591997</span>,
        <span class="longitude">15.4632963</span>
    </a>
</div>
```

Crawling strategies

- Two main approaches:
- **Breath first search**
  - Data structure: Queue (FIFO)
  - Keeps shortest path to start
- **Depth first search**
  - Data structure: Stack (LIFO)
  - Quickly moves away from initial start node

Concurrent crawlers

- Run crawlers on multiple machines
- Even geographically dispersed
- $\rightarrow$ shared data structures need to be synchronised

Deep crawling

Deep Web

- Also called hidden Web (in contrast to the surface Web)
- Consider a Web site that contains a form for the user to fill out
- e.g. a search input box
- The task of the deep crawler is to fill out this box automatically and crawl the result

Topical crawler

Topical Crawler

- Application: On-the-fly crawling of the Web
- Starting point: small set of seed pages
- Crawler tries to find similar pages
- Seed pages are used as reference

Focused crawler

Focused Crawler

- Application: collect pages with specific properties, e.g. thematic, type
- For example: find all Blogs that talk about football
- ... where Blog is the type and football is the topic
- Predict how well the pages in the frontier match the criteria
- Typically uses a manually assembled training data set $\rightarrow$ classification

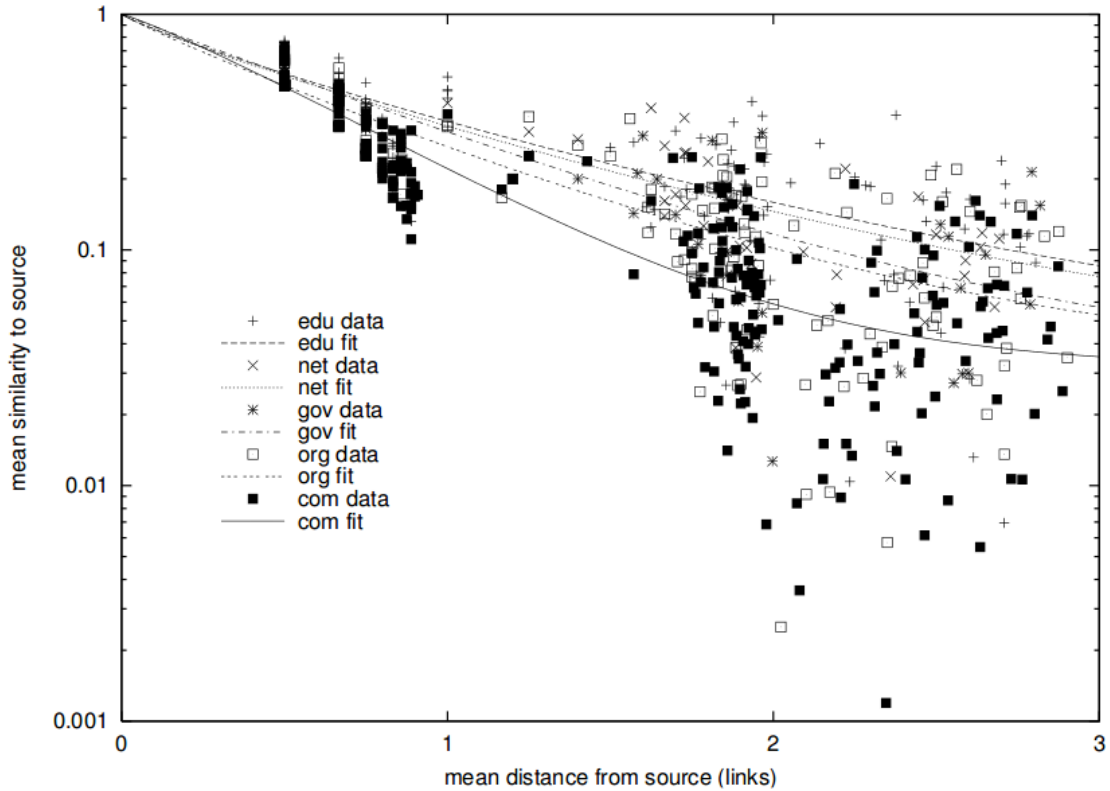
The distinction between topical crawler and focused crawler is not conclusive in the literature

Focused crawler

- Cues to predict relevant pages
  1. **Lexical**, e.g. the textual content of a page
  2. **Link topology**, e.g. the structure of the hyperlinks
- Cluster hypothesis: pages lexically (or topologically) close to a relevant page is also relevant with high probability.
- Need to address two issues:
  1. Link-content conjecture
  2. Link-cluster conjecture

Focused crawler

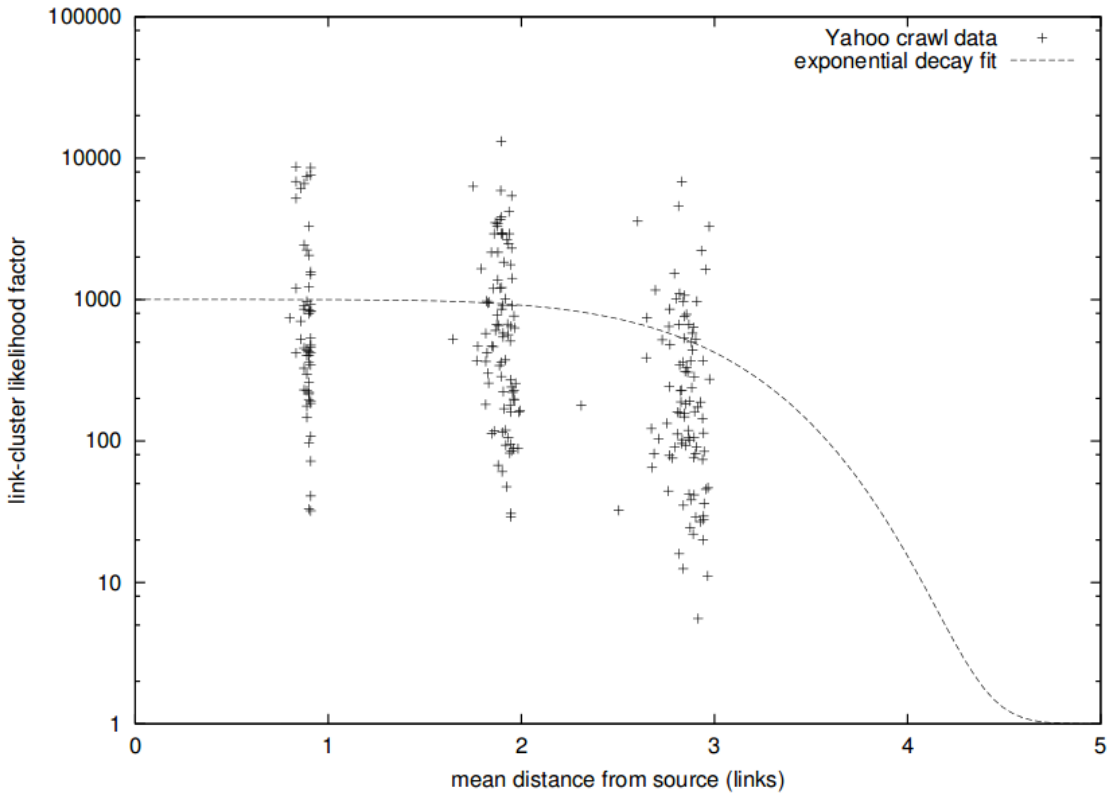
- Link-content conjecture
- Are two pages that link to each other more likely to be lexically similar?



Decay of the cosine similarity as a function of their mean directed link distance

Focused crawler

- Link-cluster conjecture
- Are two pages that link to each other more likely to be semantically related?



Decay in mean likelihood ration as a function of mean directed link distance, starting from Yahoo! directory topics

Evaluation

- Examples to measure and compare the performance of crawlers:
- **Harvest rate**
- $\rightarrow$ Percentage of good pages
- **Search length**
- $\rightarrow$ Number of pages to be crawled before a certain percentage of relevant pages are found

Web information extraction

- Web information extraction is the problem of extracting target information item from Web pages
- $\rightarrow$ Two problems
  1. Extract information from natural language text
  2. Extract structured data from Web pages

The first problem will be presented in the upcoming week, the second in the next minutes

Web information extraction

- Web information extraction via structure
- Motivation: Often pages on the Web are generated out of databases
- Data records are thereby transformed via templates into web pages
- For example: Amazon product lists & product pages
- Task: Extract the original data record out of the Web page

This task is often called wrapper generation.

Wrapper generation

- Three basic approaches for wrapper generation:
  1. Manual - simple approach, but does not scale for many sites
  2. Wrapper induction - supervised approach
  3. Automatic extraction - unsupervised approach

We will have a look at the wrapper induction.

Wrapper generation

- Wrapper induction
- Needs manually labelled training examples
- Learn a classification algorithm
- A simple approach:
  - Web page is represented by a sequence of tokens
  - Idea of landmarks: locate the beginning and end of a target item

Wrapper generation

- Manually labelling is tedious work
- Idea: reduce the amount of work by intelligently selecting the training examples
- $\rightarrow$ Active Learning approach
- In the case of simple wrapper induction use co-training:
  - Search landmarks from the beginning and from the back at the same time
  - Use disagreement as indicator for a training example to annotate

Web crawler and ethics

- Web crawlers may cause trouble
- If too many requests are sent to a single Web site
- ... it might look like a denial of service (DoS) attack
- $\rightarrow$ the source IP will be blacklisted
- Respect the robots.txt file (but it's not a legal requirement)
- Some bot disguise themselves and try to replicate a user's behaviour
- Some server disguise themselves, e.g. cloaking (various versions of the same Web page for different clients)

Web crawler and ethics

Example: orf.at/robots.txt

```
# do not index the light version
User-agent: *
Disallow: /1/stories/
Disallow: /full

# these robots have been bad once:

user-agent: stress-agent
Disallow: /

User-agent: fast
Disallow: /

User-agent: Scooter
Disallow: /
```

Data Cleaning

Filter out unwanted data

Motivation

- Often data sets will contain:
  - Unnecessary data
  - Missing values
  - Noise
  - Incorrect data
  - Inconsistent data
  - Formatting issues
  - Duplicate information
  - Disguised data
- These factors will have an impact on the results of the data mining process

Garbage in $\rightarrow$ garbage out

Unnecessary data

- Remove excess information
- Identify which parts contain relevant information
- Depends on the final purpose of the KD process
- In case of Web pages:
  - Get rid of navigation, ads, ...
  - Identify the main content of a page
  - Identify the main images

Unnecessary data

Web page cleaning

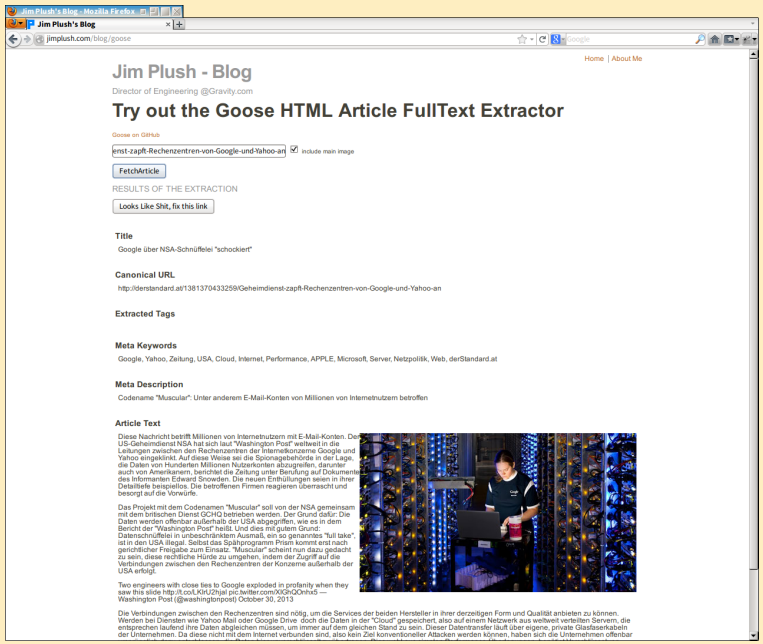


Figure: left: original Web page, right: applied "goose" to extract the article text, article image, meta-data, ...

Try out Sensium: https://www.sensium.io/

Missing values

- Sources of missing values: faulty equipment, human errors, anonymous data, ...
- e.g. Consider a data set consisting of multiple rows, and in some of the rows some values are missing
- Implications of missing data [Barnard&Meng1999]
  - Loss of efficiency, due to less data
  - Some methods may not handle missing data
  - Bias in the (data mining) results

Missing values may indicate errors in the data, but not necessarily so.

Missing values

- Categorisation of missing data [Little&Rubin1987]
  - MCAR - Missing completely at random, does not depend on the observed or missing data
  - MAR - Missing at random, depends on the observed data, but not the missing data
  - NMAR - Not missing at random, depends on the missing data

Need appropriate methods for each of the different types.

Missing values

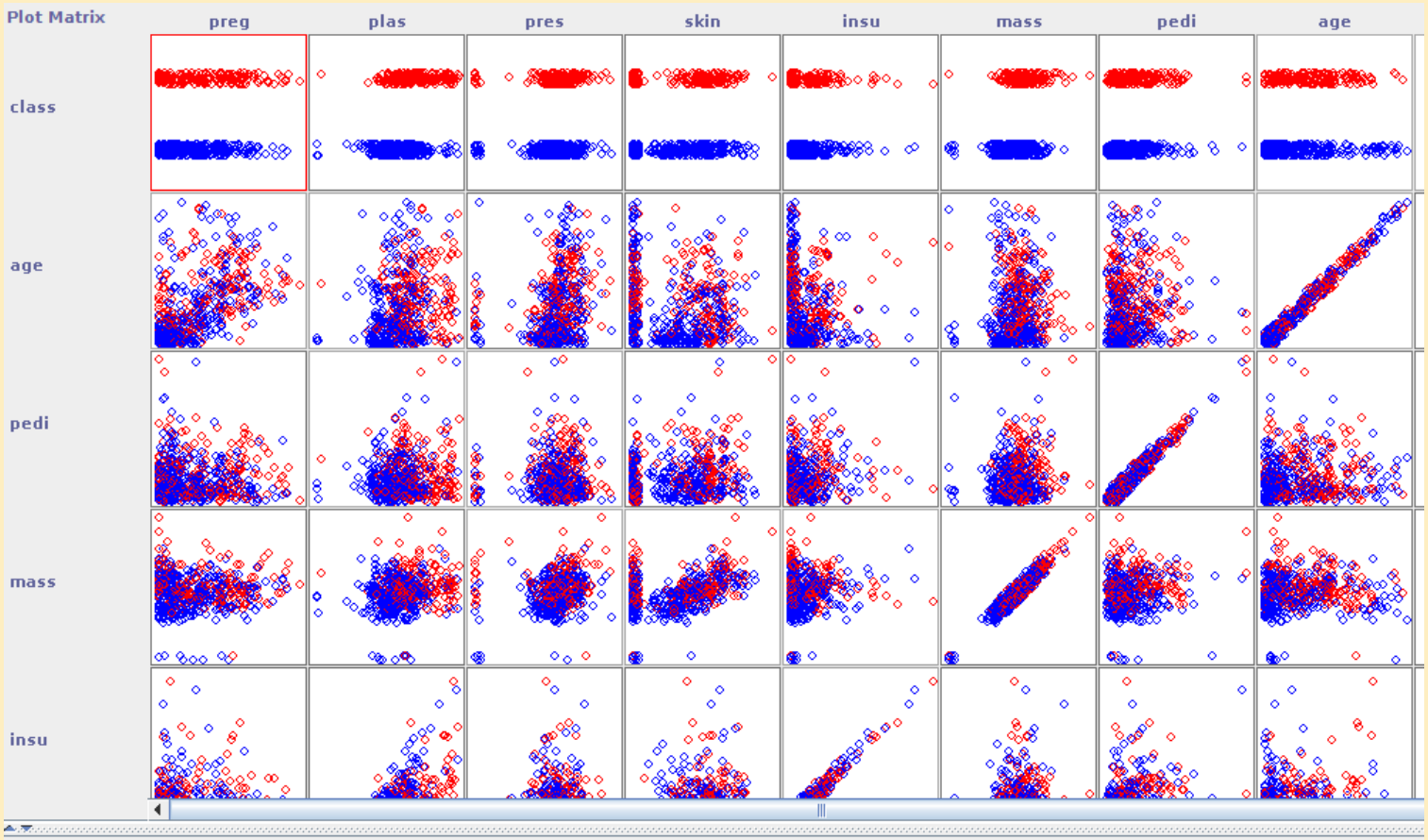
- How to deal with missing values?
  - Ignore the entire row
  - Global constant
  - Use the most common value
  - Use average (mean, median, ...) of all other values
  - Apply machine learning techniques, e.g. regression, k-NN, clustering

Redundant data

- Same data, but multiple times
- May lead to a bias in the (data mining) results
- How to deal with redundant data?
- Correlation and covariance analysis,
  - Manually inspecting scatter plots
  - Compute correlation, e.g. Pearson's correlation coefficient
- Near-duplicate detection
  - e.g. for search engines detect identical versions of the same Web page

Redundant data

Example for Weka's visualisation



Normalisation

- Normalisation of values, e.g. meta-data?
- Example: Normalisation of dates.
  - Many different formats: 09/11/01, 11.09.2001, 11/09/01, ...
  - → e.g. ISO-8601: `yyyy-mm-ddThh:mm:ss.nnnnnn+|-hh:mm`
- Example: Normalisation of person names.
  - → {lastName}, {firstName}, {title}

Noisy data

- Sources of noise: faulty equipment, data entry problems, inconsistencies, data transmission problems, ...
- How to deal with noisy data?
- Manual - sort out noisy data manually
- Binning - sort data and partition into bins
  - Equal width (distance) partitioning
  - Equal depth (frequency) partitioning
- Regression - fitting the data into regression functions
- Outlier detection

Outlier Detection

Filter out unwanted data

What is an outlier?

- No universally accepted definition of an **outlier** (**anomaly**, novelty, change, deviation, surprise, ...)
- **Outlier detection** = detecting patterns that do not conform to an established normal behavior (e.g., rare, unexpected, ...)

What is an outlier?

Definition #1

An outlying observation, or outlier, is one that appears to deviate markedly from other members of the sample in which it occurs [Grubbs1969]

Definition #2

An observation (or subset of observations) which appears to be inconsistent with the remainder of that set of data [Barnett1994]

Definition #3

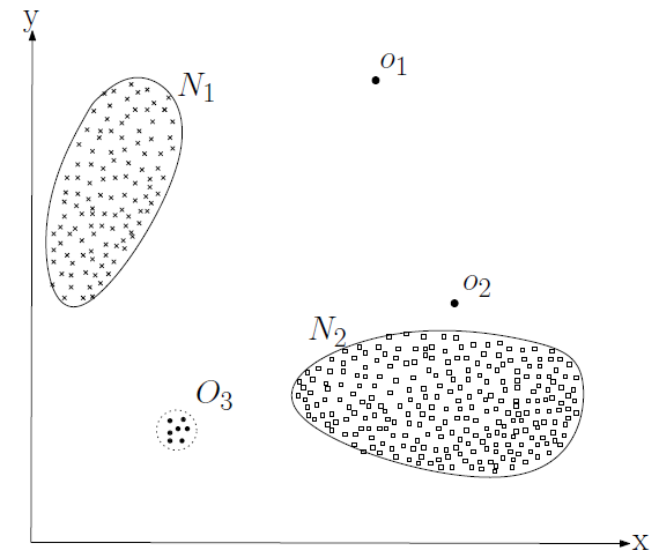
An outlier is an observation which deviates so much from the other observations as to arouse suspicions that it was generated by a different mechanism [Hawkins1980]

Sample applications

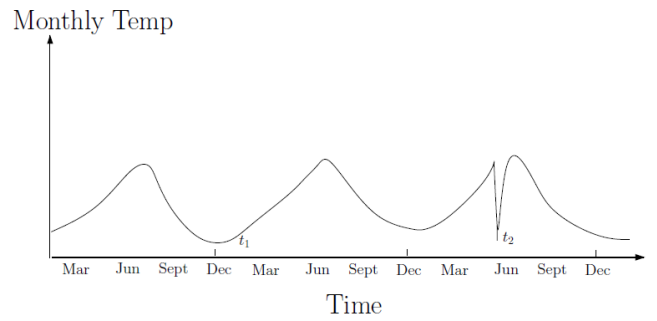
- Intrusion detection (host-based, network-based)
- Fraud detection (credit cards, mobile phones, insurance claims)
- Medical and public health (patient records, EEG, ECG, disease outbreaks)
- Industrial damage detection (fault detection, structural defects)
- Image processing (Satellite images, robot applications, sensor systems)
- Outlier detection in text data
- etc.

Types of outliers

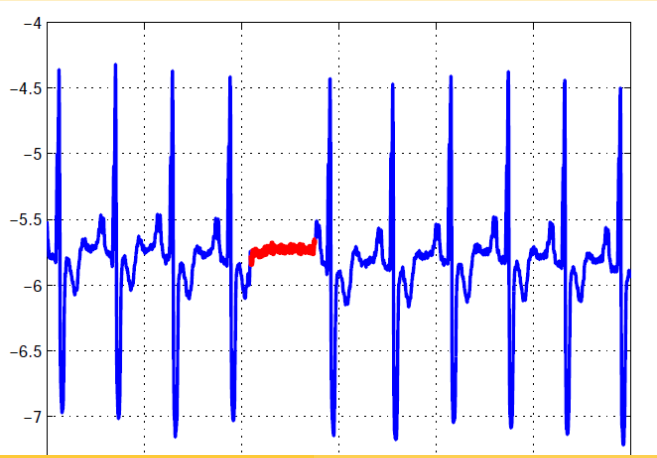
Point outliers



Contextual outliers



Collective outliers



Basic characterization of techniques I

Supervised outlier detection

- Training data: labeled instances for both normal and outlier class
- ... size of classes is inherently unbalanced
- ... obtaining representative samples of the outlier class is challenging

Semi-supervised outlier detection

- Training data: labeled instances for only one class (typically the normal class)
- ... construct a model corresponding to normal behavior, and test likelihood that new instances are generated by this model

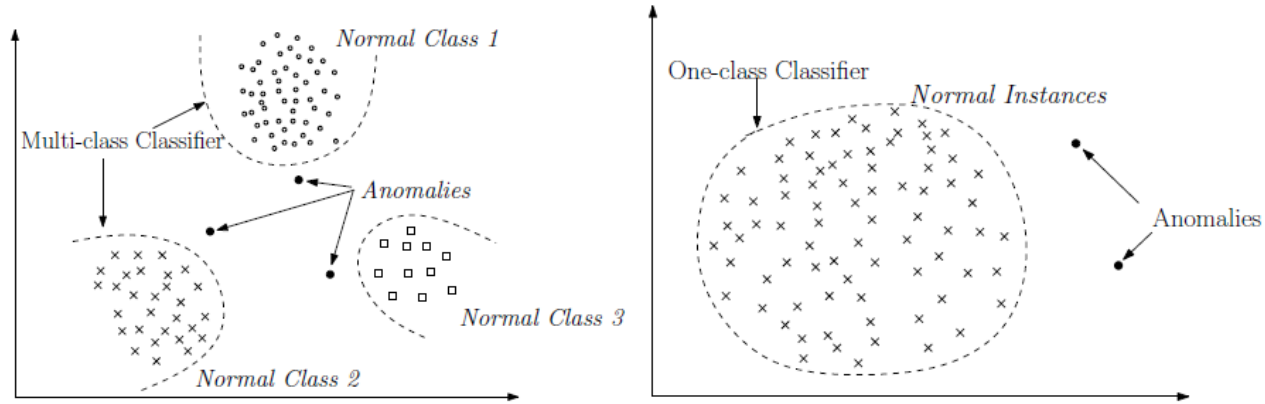
Basic characterization of techniques II

Unsupervised outlier detection

- Unlabeled training data
- ... assume that the majority of the instances in the data set are normal
- In most applications no labels are available

Classification based techniques I

train a classifier that distinguishes between normal and outlier classes (**multi-class** vs **one-class** detection)



(a) Multi-class Anomaly Detection (b) One-class Anomaly Detection

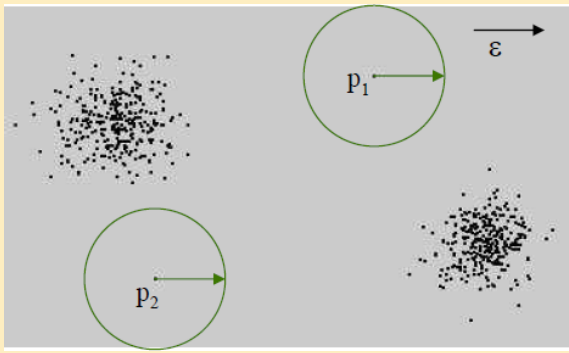
- Neural networks (e.g., Replicator neural networks)
- Bayesian networks
- Support vector machines (e.g., One-class-SVM)
- Rule based (e.g., decision trees)

Nearest-neighbor based techniques I

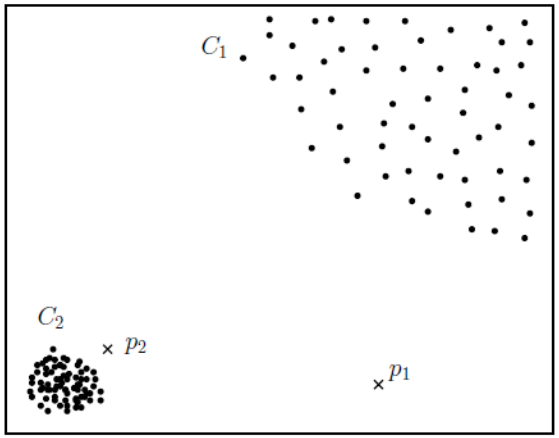
Assume that normal data instances occur in dense neighbourhoods, while anomalies occur far from their closest neighbours (requires **distance** or **similarity measure**)

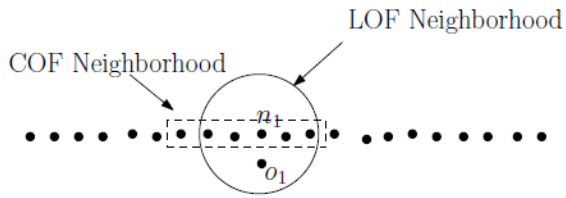
Using distance to k-th nearest neighbor

- Outlier score = distance to k-th nearest neighbour [Ramaswamy2000]
- Several variants:
  - count number of nearest neighbours within a certain distance
  - $DB(\epsilon, \pi)$ [Knorr1997]: A point p is considered an outlier if less than π percent of all other points have a distance to p greater than ϵ



Nearest-neighbor based techniques II





Examples for cases, where naive k-NN will not work - due to differences in density or types of regularities not captured by the distance function.

Clustering based techniques

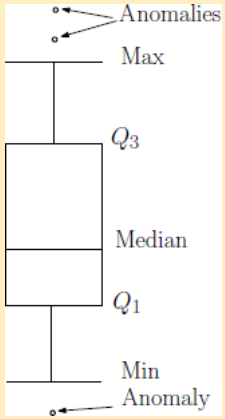
Three categories:

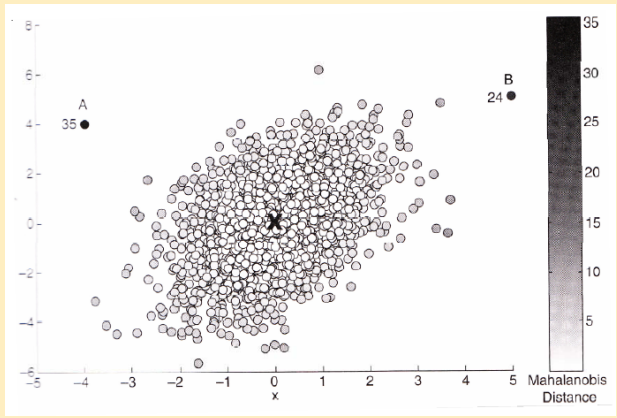
1. Normal instances belong to a cluster in the data, while anomalies do not belong to any cluster
   - cluster algorithms that do not assign every instance to a cluster (e.g., DBSCAN)
2. Normal data instances lie close to their closest cluster centroid, while anomalies are far away from their closest cluster centroid
   - Self-Organizing Maps (SOM), K-Means, Expectation Maximization, ...
3. Normal data instances belong to large and dense clusters, while anomalies belong to small or sparse clusters
   - Cluster-Based Local Outlier Factor (CBLOF): compares distance to centroid with size of the cluster

Statistical techniques

Parametric techniques

- assume the knowledge of an underlying distribution (e.g., Gaussian) and estimate parameters





Non-parametric techniques

- e.g., based on histograms

Other techniques I

Information theoretic techniques

- measure the complexity $\mathcal{C}(D)$ of a dataset D
- Outliers: $I = \arg\max_{I \subset D} [\mathcal{C}(D) - \mathcal{C}(D - I)]$

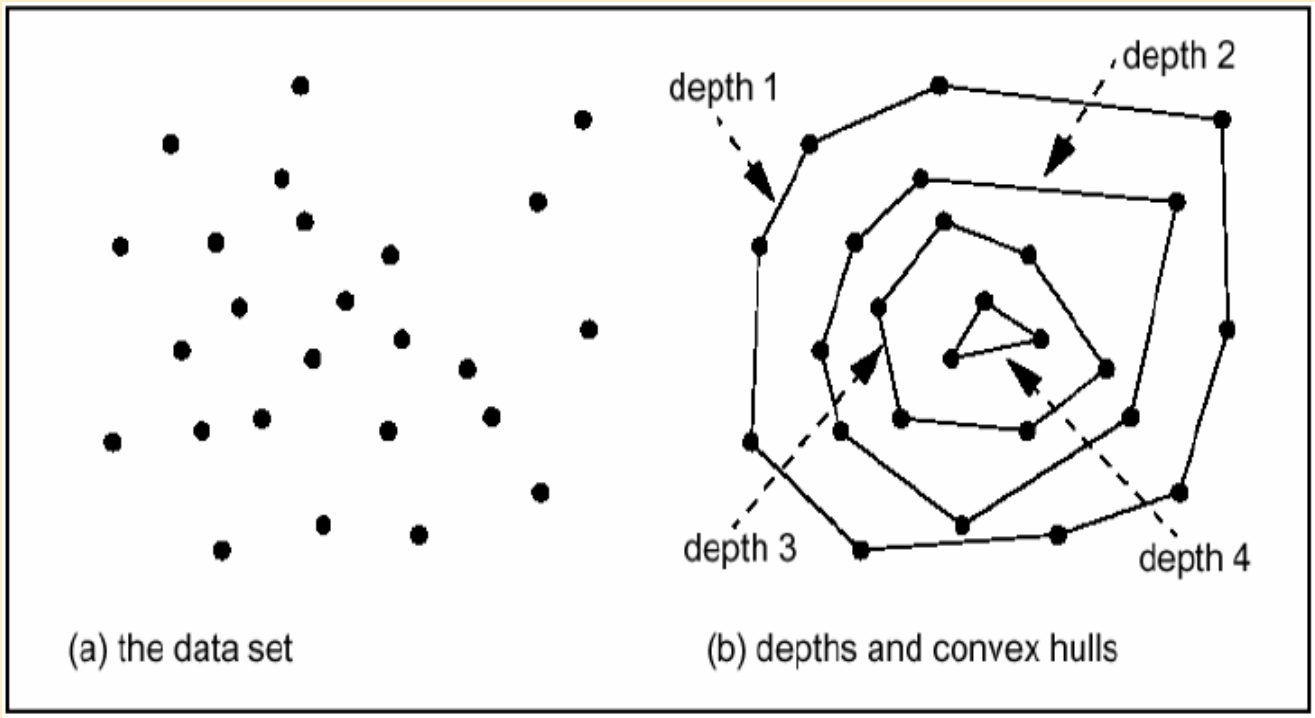
Spectral techniques

- find a lower dimensional subspace in which normal instances and outliers appear significantly different (e.g., PCA)

Other techniques II

Depth-based techniques

- Organize data into convex hull layers



- Points with depth $\leq k$ are reported as outliers

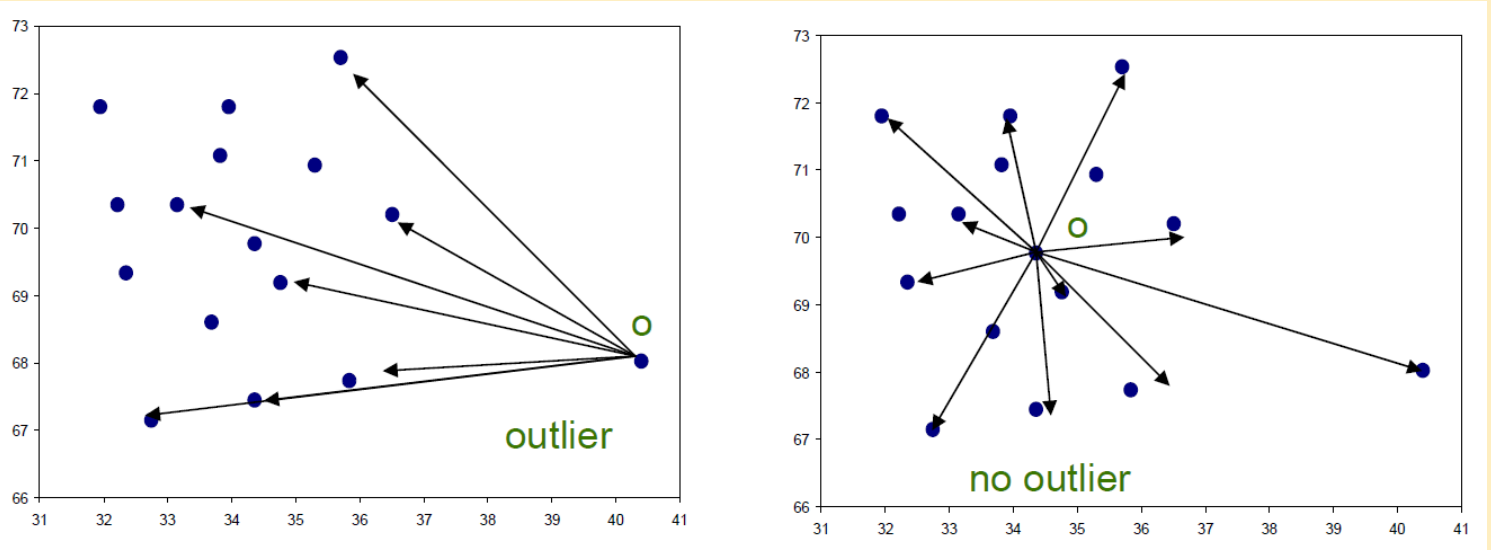
Other techniques III

Angle-based techniques

- Angle-based outlier degree (ABOD) [Kriegel2008]:

$$ABOD(p) = VAR\left(\frac{\langle \vec{xp}, \vec{yp} \rangle}{||\vec{xp}||^2 ||\vec{yp}||^2}\right)$$

- outliers have a smaller variance
- more stable than distances in high dimensions



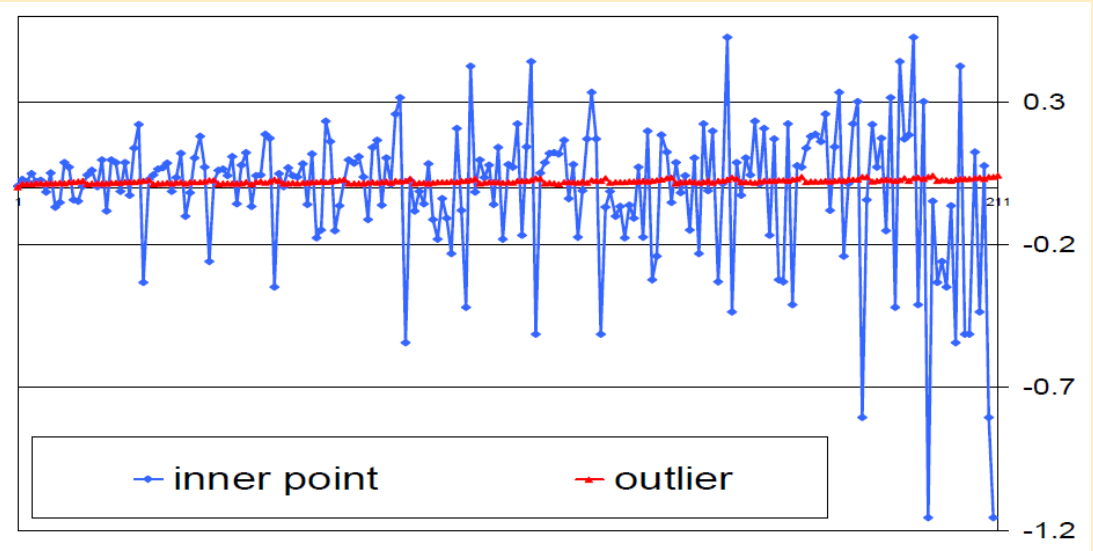
Other techniques III

Angle-based techniques

- Angle-based outlier degree (ABOD) [Kriegel2008]:

$$ABOD(p) = VAR\left(\frac{\langle \vec{xp}, \vec{yp} \rangle}{||\vec{xp}||^2 ||\vec{yp}||^2}\right)$$

- outliers have a smaller variance
- more stable than distances in high dimensions

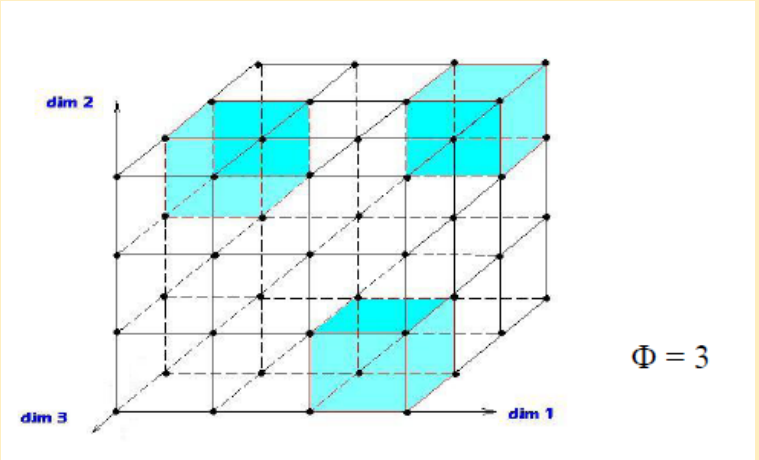


Other techniques IV

Grid-based subspace outlier detection [Aggarwal2001]

- Partition data space in equi-depth grid (Φ = number of cells in each dimension)
- Sparsity coefficient $S(C)$ of a grid cell C

$$S(C) = \frac{n(C) - n \cdot \left(\frac{1}{\Phi}\right)^k}{\sqrt{n \cdot \left(\frac{1}{\Phi}\right)^k \cdot \left(1 - \left(\frac{1}{\Phi}\right)^k\right)}}$$



$n(C)$ ... number of data objects in cell C

- Outliers are located in cells with $S(C) < 0$ ($n(C)$ is lower than expected)

Thank You!

Next up: Feature Extraction

Further information

Special thanks to Stefan Klampfl for his slides on outlier detection