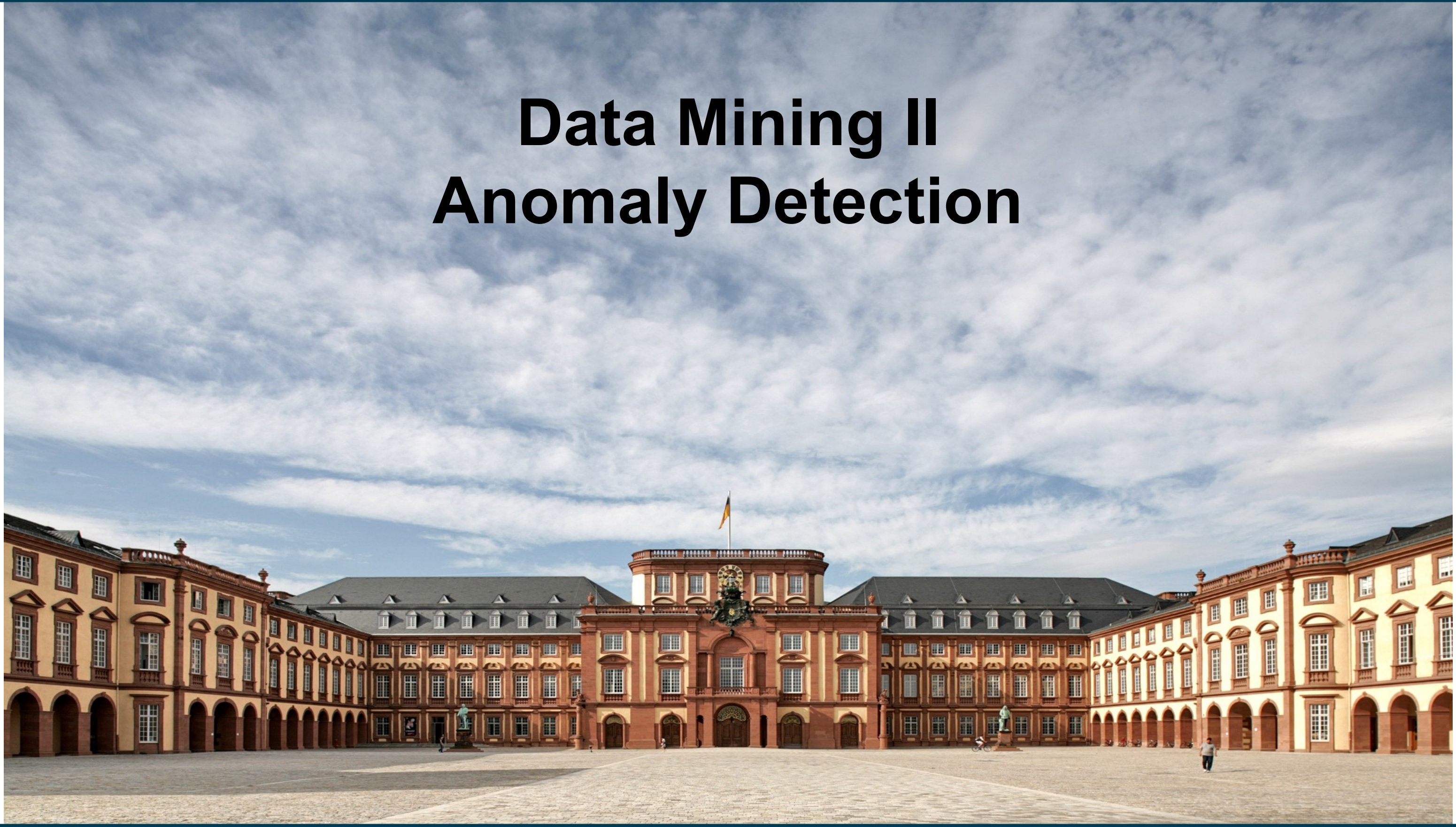
UNIVERSITÄT MANNHEIM

Data Mining II
Anomaly Detection



Heiko Paulheim

Anomaly Detection

- Also known as “Outlier Detection”
- Automatically identify data points
  that are somehow different from the rest

- Working assumption:
  – There are considerably more “normal” observations than “abnormal” observations (outliers/anomalies) in the data

- Challenges
  – How many outliers are there in the data?
  – What do they look like?
  – Method is unsupervised
    - Validation can be quite challenging (just like for clustering)

Recap: Errors in Data

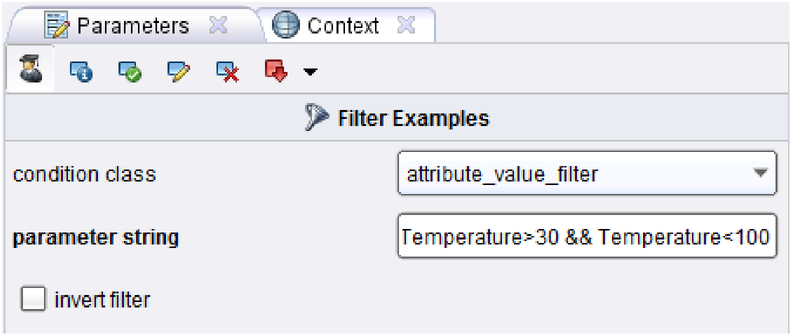
- Sources
  - malfunctioning sensors
  - errors in manual data processing (e.g., twisted digits)
  - storage/transmission errors
  - encoding problems, misinterpreted file formats
  - bugs in processing code
  - ...

Image: http://www.flickr.com/photos/16854395@N05/3032208925/

Recap: Errors in Data

- Simple remedy
  - remove data points outside a given interval
    - this requires some domain knowledge

- Advanced remedies
  - automatically find suspicious data points



Applications: Data Preprocessing

- Data preprocessing
  - removing erroneous data
  - removing true, but useless deviations

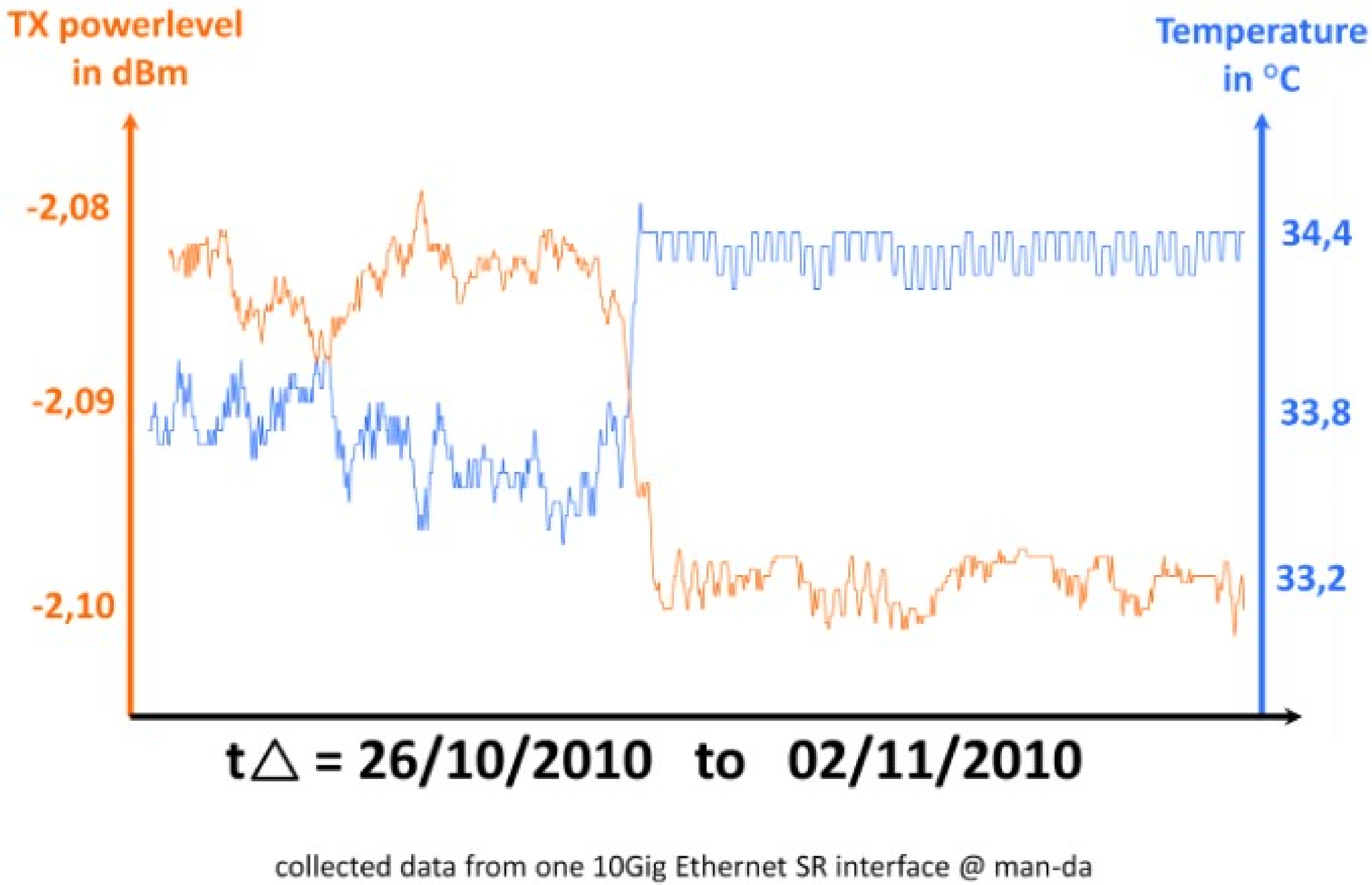
- Example: tracking people down using their GPS data
  - GPS values might be wrong
  - person may be on holidays in Hawaii
    - what would be the result of a kNN classifier?

Applications: Credit Card Fraud Detection

- Data: transactions for one customer
  - €15.10 Amazon
  - €12.30 Deutsche Bahn tickets, Mannheim central station
  - €18.28 Edeka Mannheim
  - $500.00 Cash withdrawal. Dubai Intl. Airport
  - €48.51 Gas station Heidelberg
  - €21.50 Book store Mannheim

- Goal: identify unusual transactions
  - possible attributes: location, amount, currency, ...

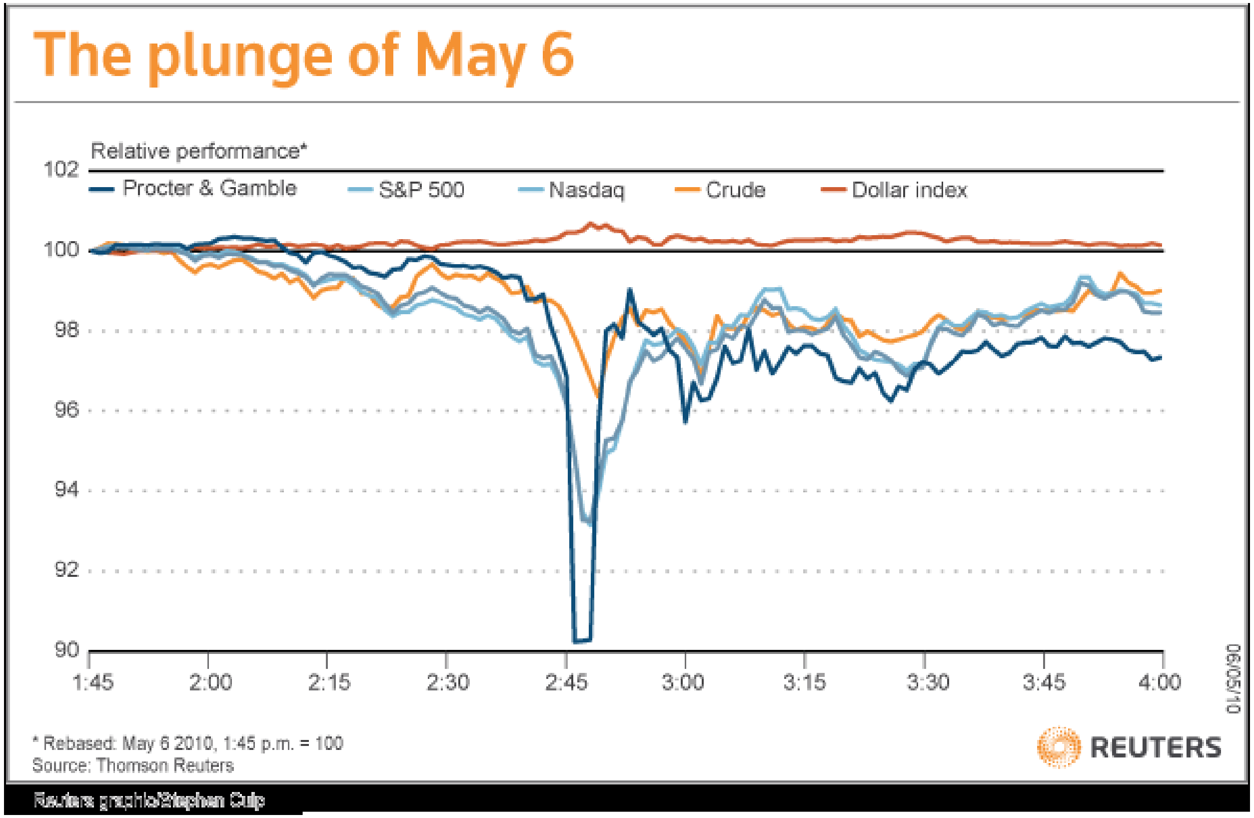
Applications: Hardware Failure Detection

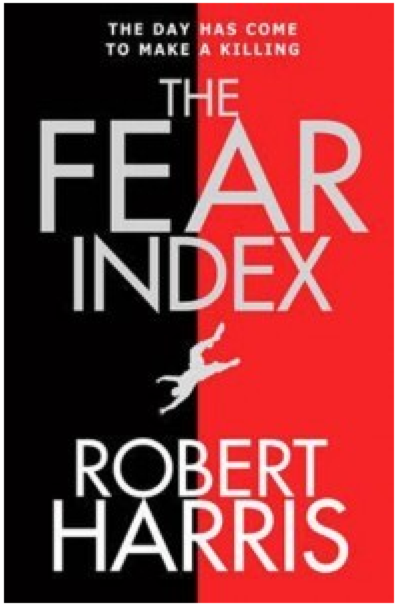


Thomas Weible: An Optic's Life (2010).

Applications: Stock Monitoring

- Stock market prediction
- Computer trading



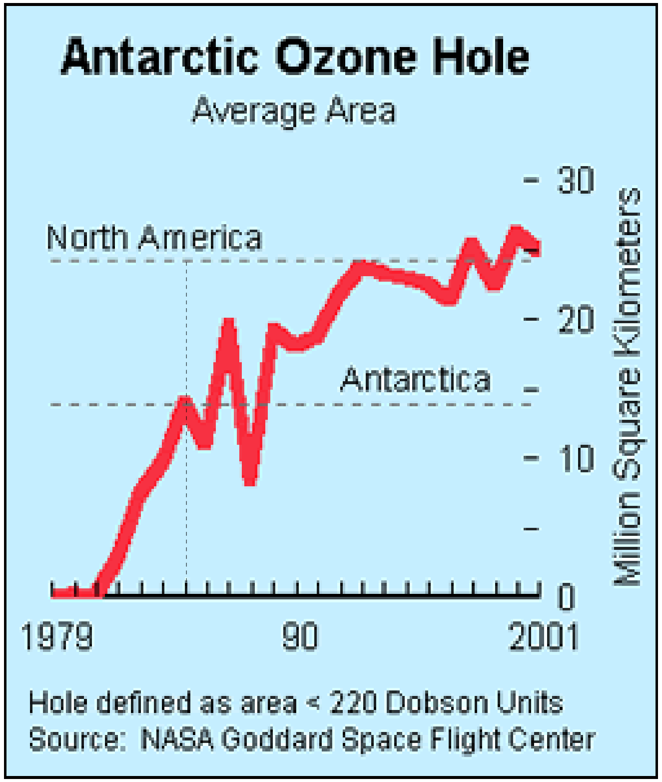


http://blogs.reuters.com/reuters-investigates/2010/10/15/flash-crash-fallout/

Errors vs. Natural Outliers

Ozone Depletion History

- In 1985 three researchers (Farman, Gardinar and Shanklin) were puzzled by data gathered by the British Antarctic Survey showing that ozone levels for Antarctica had dropped 10% below normal levels
- Why did the Nimbus 7 satellite, which had instruments aboard for recording ozone levels, not record similarly low ozone concentrations?
- The ozone concentrations recorded by the satellite were so low they were being treated as outliers by a computer program and discarded!



Sources:
http://exploringdata.cqu.edu.au/ozone.html
http://www.epa.gov/ozone/science/hole/size.html

Errors, Outliers, Anomalies, Novelties...

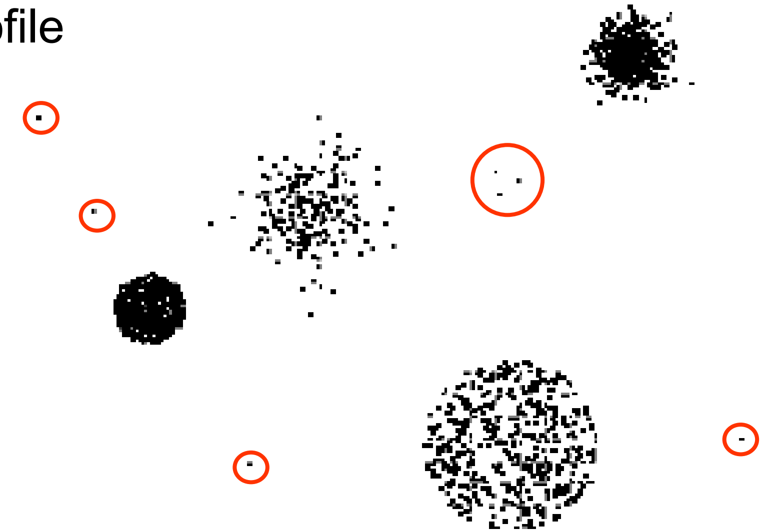
- What are we looking for?
  - Wrong data values (errors)
  - Unusual observations (outliers or anomalies)
  - Observations not in line with previous observations (novelties)
- Unsupervised Setting:
  - Data contains both normal and outlier points
  - Task: compute outlier score for each data point
- Supervised setting:
  - Training data is considered normal
  - Train a model to identify outliers in test dataset

Methods for Anomaly Detection

- Graphical
  - Look at data, identify suspicious observations
- Statistic
  - Identify statistical characteristics of the data
    - e.g., mean, standard deviation
  - Find data points which do not follow those characteristics
- Density-based
  - Consider distributions of data
  - Dense regions are considered the “normal” behavior
- Model-based
  - Fit an explicit model to the data
  - Identify points which do not behave according to that model

Anomaly Detection Schemes

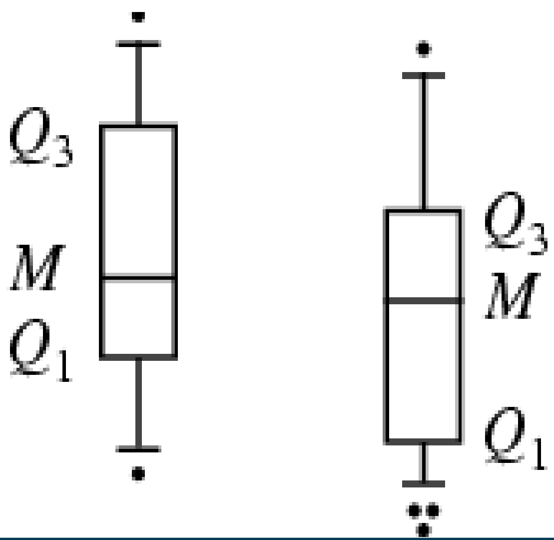
- General Steps
  - Build a profile of the “normal” behavior
    - Profile can be patterns or summary statistics for the overall population
  - Use the “normal” profile to detect anomalies
    - Anomalies are observations whose characteristics differ significantly from the normal profile

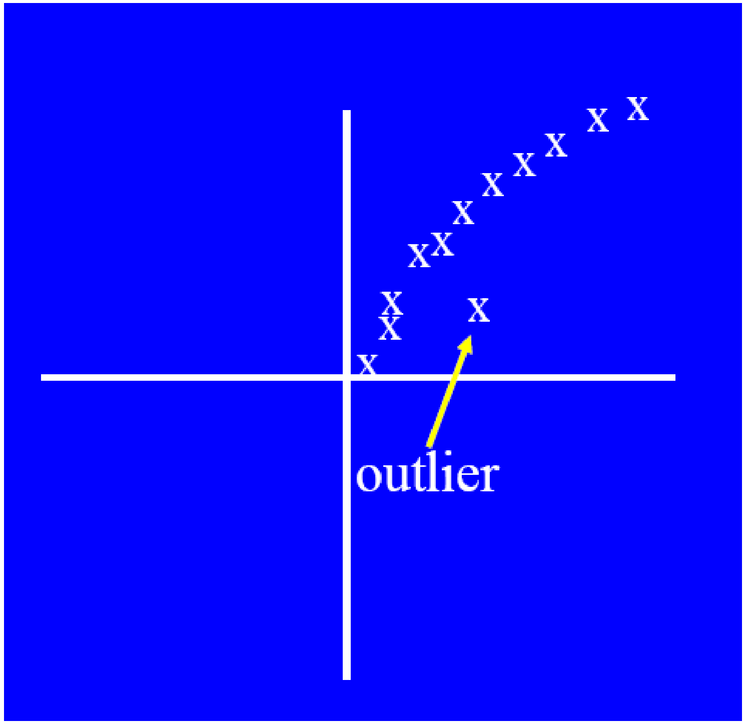


- Types of anomaly detection schemes
  - Graphical & Statistical-based
  - Distance-based
  - Model-based

Graphical Approaches

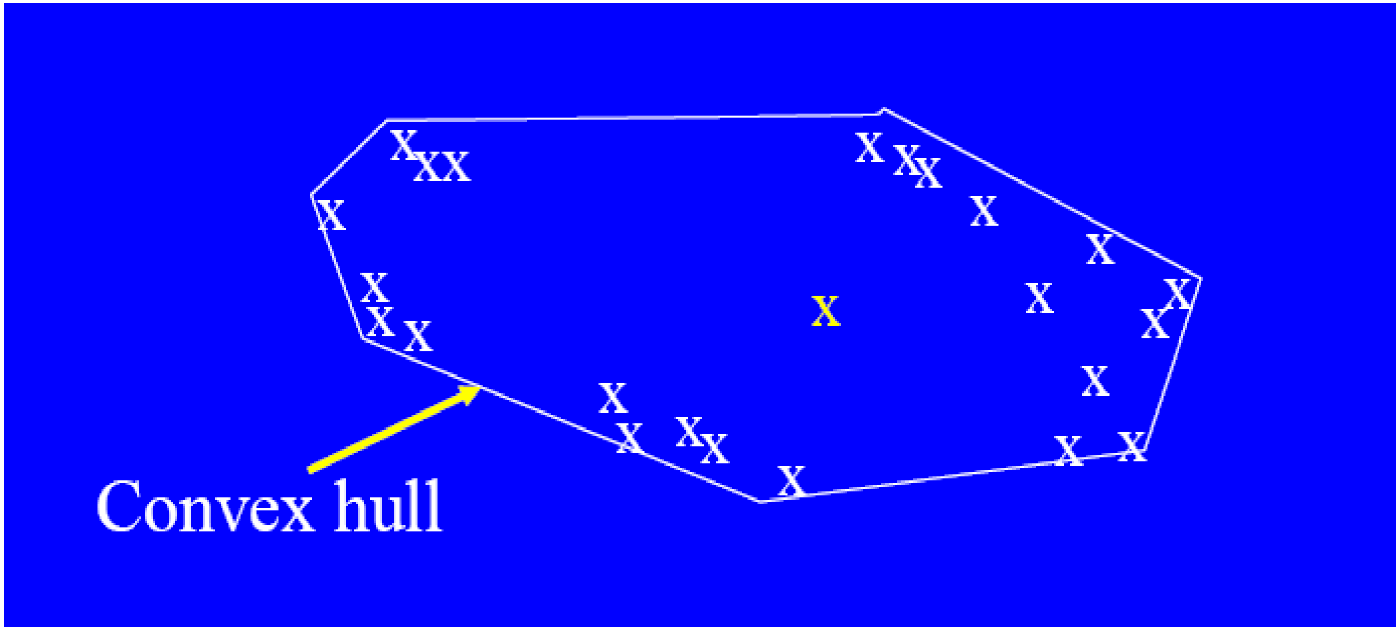
- Boxplot (1-D), Scatter plot (2-D), Spin plot (3-D)
- Limitations
  - Time consuming
  - Subjective





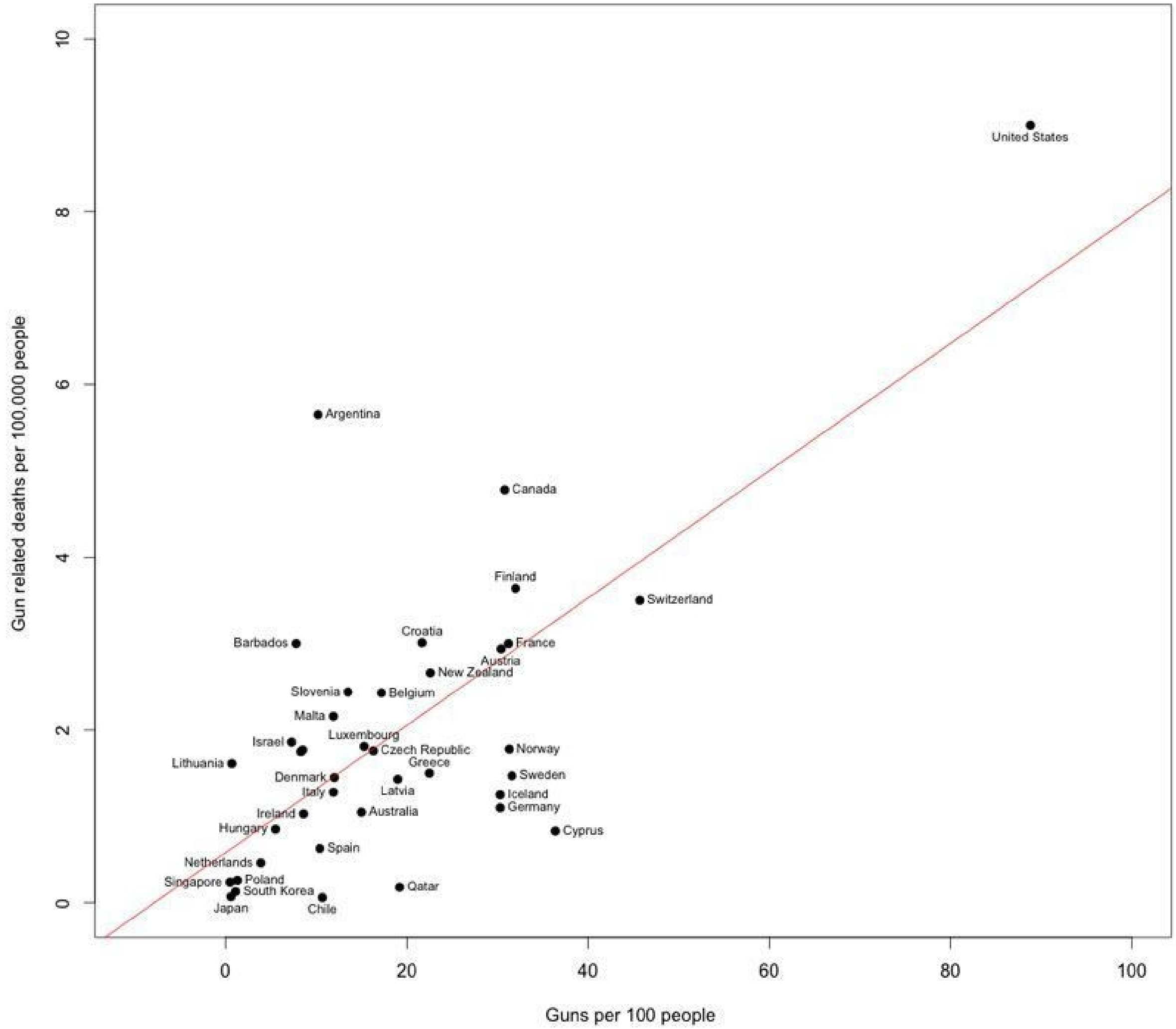
Convex Hull Method

- Extreme points are assumed to be outliers
- Use convex hull method to detect extreme values



- What if the outlier occurs in the middle of the data?

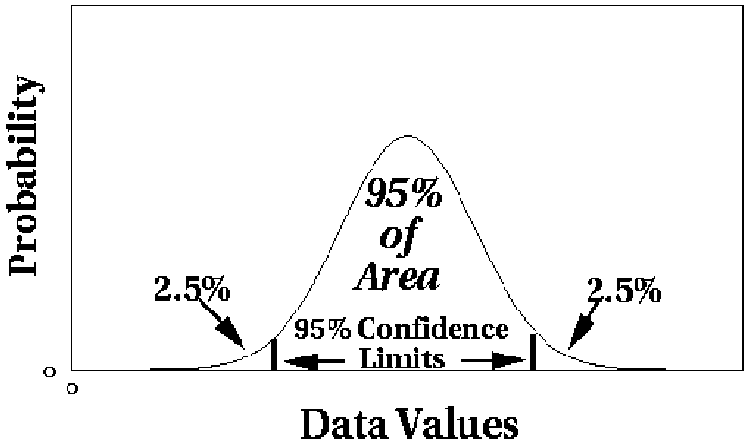
Interpretation: What is an Outlier?



Statistical Approaches

- Assume a parametric model describing the distribution of the data (e.g., normal distribution)

- Apply a statistical test that depends on
  - Data distribution
  - Parameter of distribution (e.g., mean, variance)
  - Number of expected outliers (confidence limit)

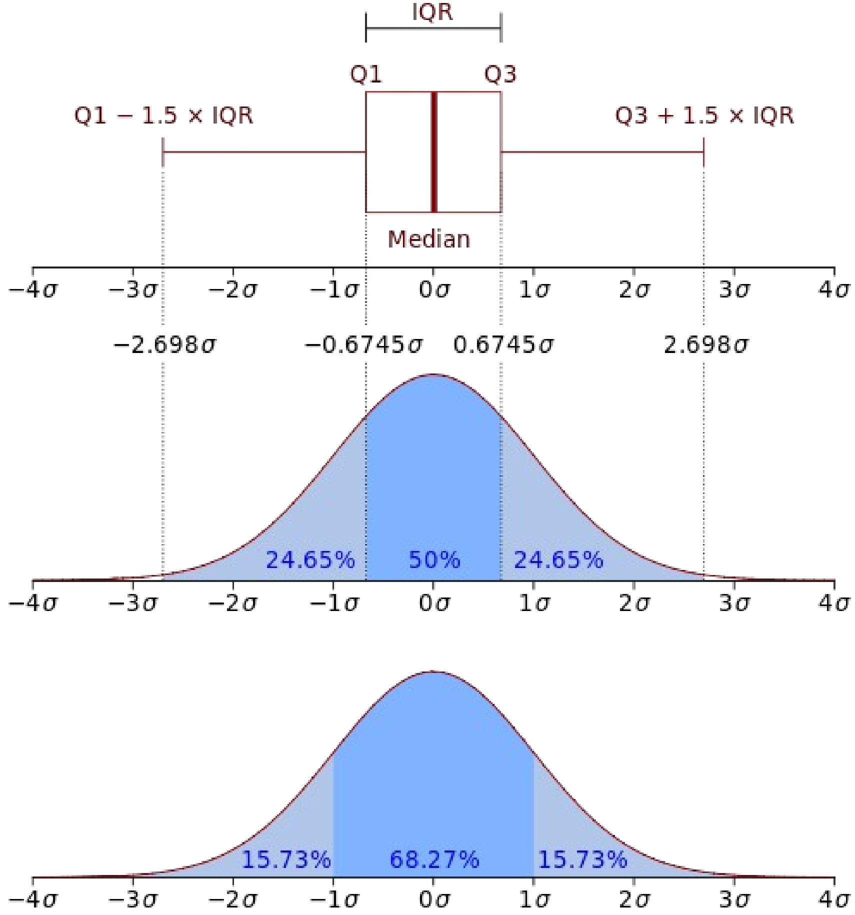


Interquartile Range

- Divides data in quartiles
- Definitions:
  - Q1: x ≥ Q1 holds for 75% of all x
  - Q3: x ≥ Q3 holds for 25% of all x
  - IQR = Q3-Q1
- Outlier detection:
  - All values outside [median-1.5*IQR ; median+1.5*IQR]
- Example:
  - 0,1,1,3,3,5,7,42 → median=3, Q1=1, Q3=7 → IQR = 6
  - Allowed interval: [3-1.5*6 ; 3+1.5*6] = [-6 ; 12]
  - Thus, 42 is an outlier

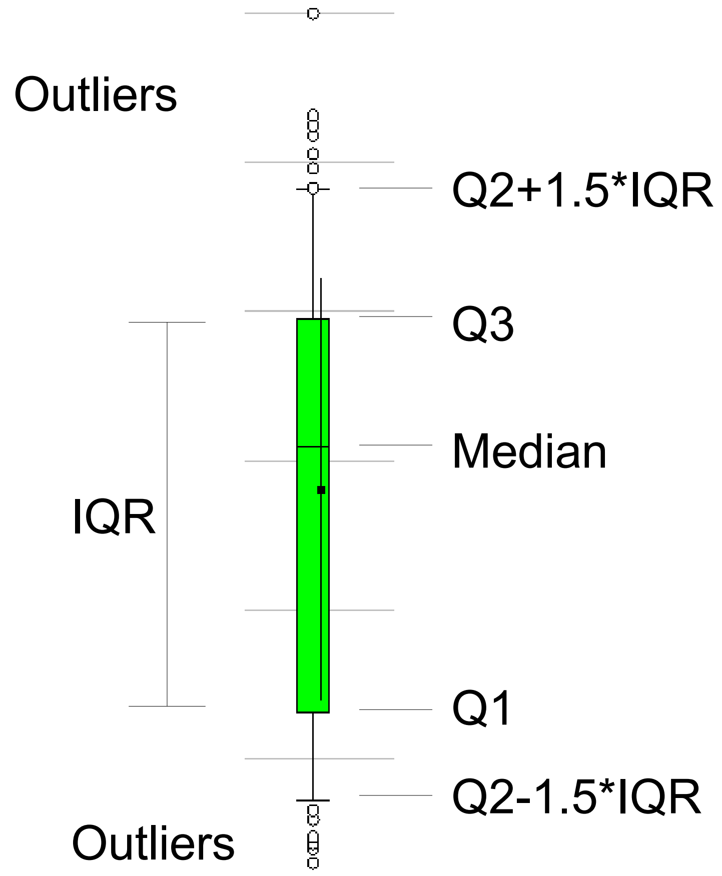
Interquartile Range

- Assumes a normal distribution



Interquartile Range

- Visualization in box plot



Median Absolute Deviation (MAD)

- MAD is the median deviation from the median of a sample, i.e.

$$MAD := median_i(X_i - median_j(X_j))$$

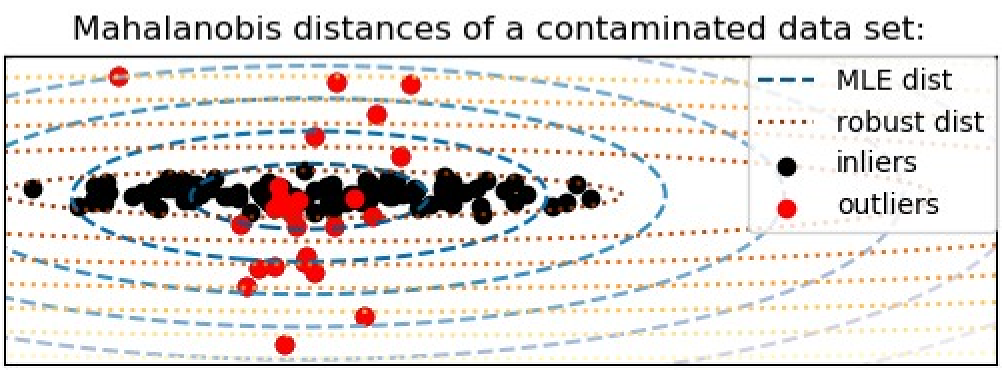
- MAD can be used for outlier detection
  - all values that are k*MAD away from the median are considered to be outliers
  - e.g., k=3

- Example:
  - 0,1,1,3,5,7,42 → median = 3
  - deviations: 3,2,2,0,2,4,39 → MAD = 2
  - allowed interval: [3-3*2 ; 3+3*2] = [-3;9]
  - therefore, 42 is an outlier

Carl Friedrich Gauss, 1777-1855

Fitting Elliptic Curves

- Multi-dimensional datasets
  - can be seen as following a normal distribution on each dimension
  - the intervals in one-dimensional cases become elliptic curves
- In Python: `covariance.EllipticEnvelope`



Limitations of Statistical Approaches

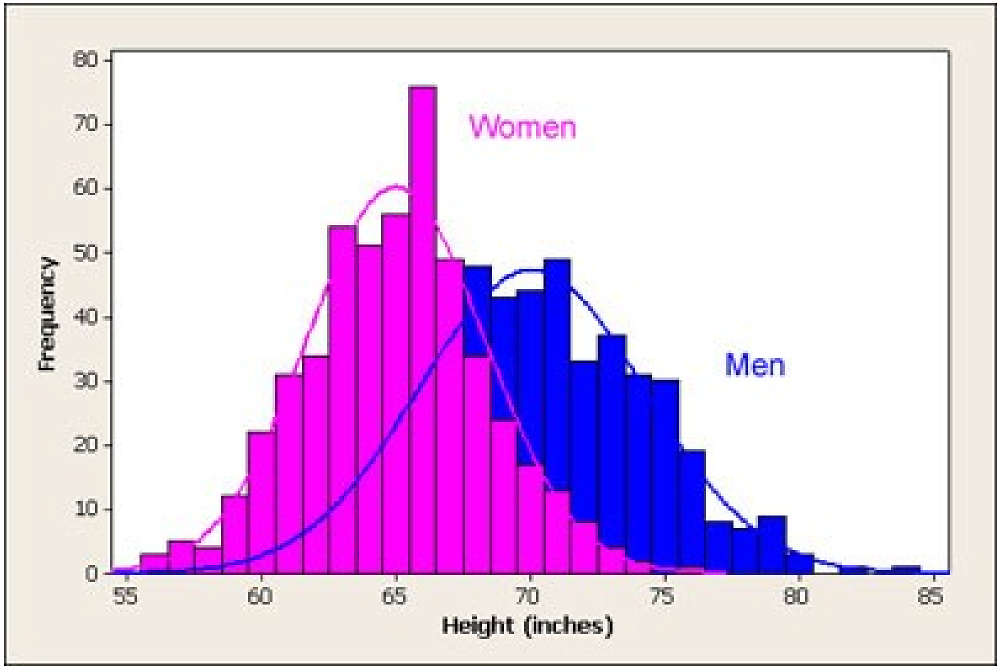
- Most of the tests are for a single attribute (called: *univariate*)

- For high dimensional data, it may be difficult to estimate the true distribution

- In many cases, the data distribution may not be known
  - e.g., IQR Test: assumes Gaussian distribution

Examples for Distributions

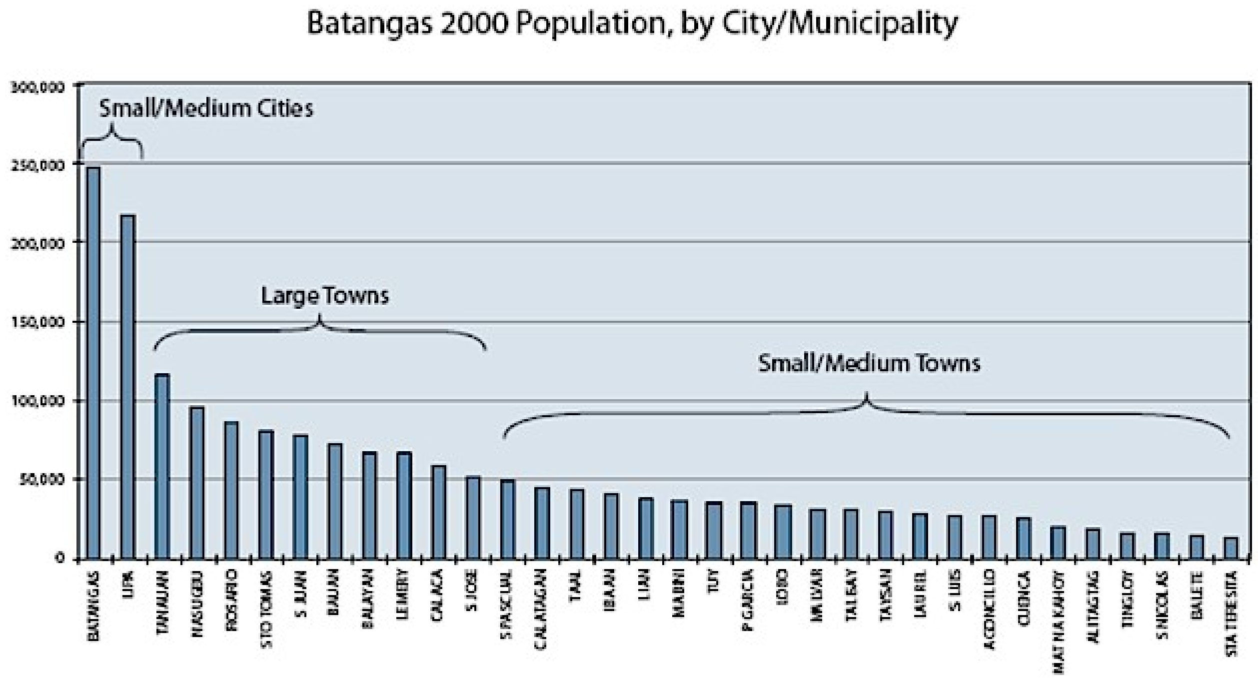
- Normal (gaussian) distribution
  - e.g., people's height



http://www.usablestats.com/images/men_women_height_histogram.jpg

Examples for Distributions

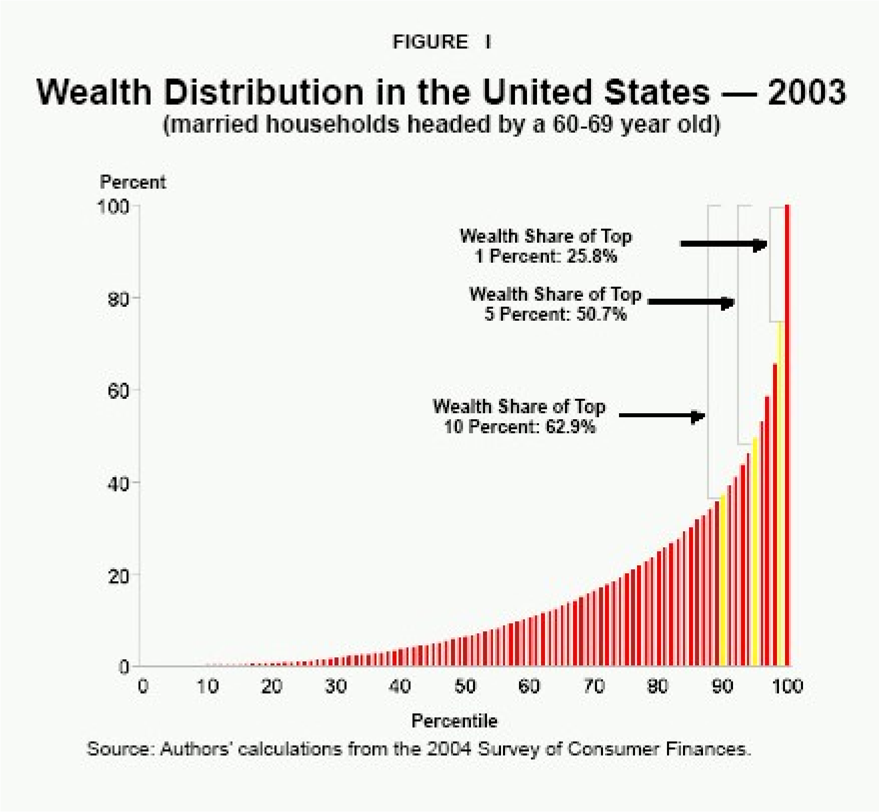
- Power law distribution
  - e.g., city population



http://www.jmc2007compendium.com/V2-ATAPE-P-12.php

Examples for Distributions

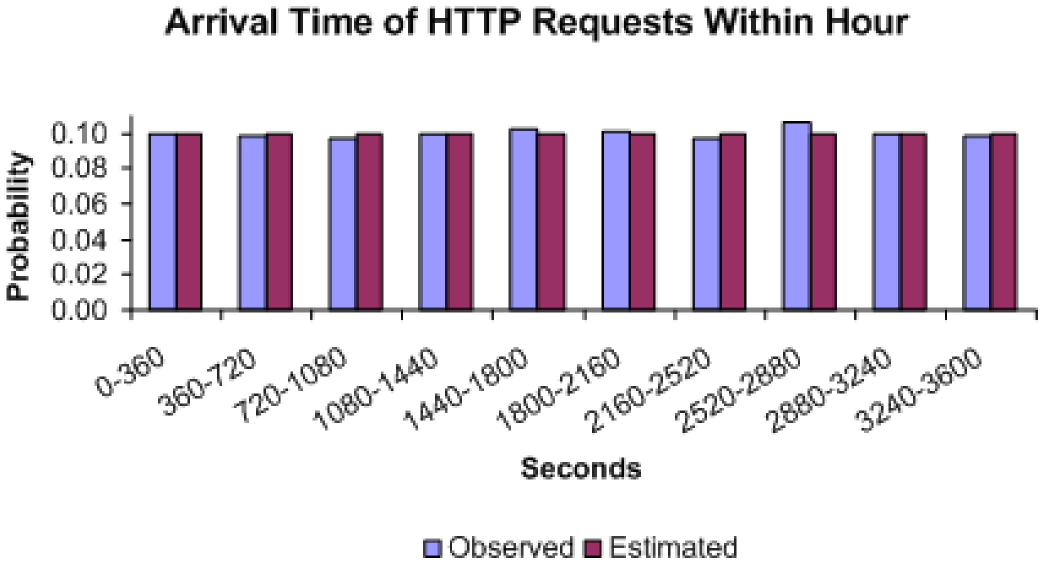
- Pareto distribution
  - e.g., wealth



http://www.ncpa.org/pub/st289?pg=3

Examples for Distributions

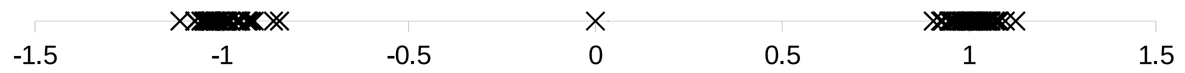
- Uniform distribution
  - e.g., distribution of web server requests across an hour



http://www.brighton-webs.co.uk/distributions/uniformc.aspx

Outliers vs. Extreme Values

- So far, we have looked at extreme values only
  - But outliers can occur as non-extremes
  - In that case, methods like IQR fail

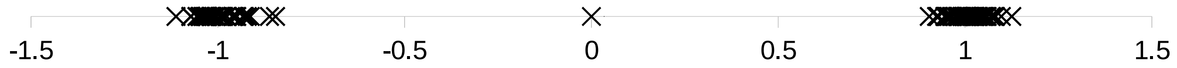


Outliers vs. Extreme Values

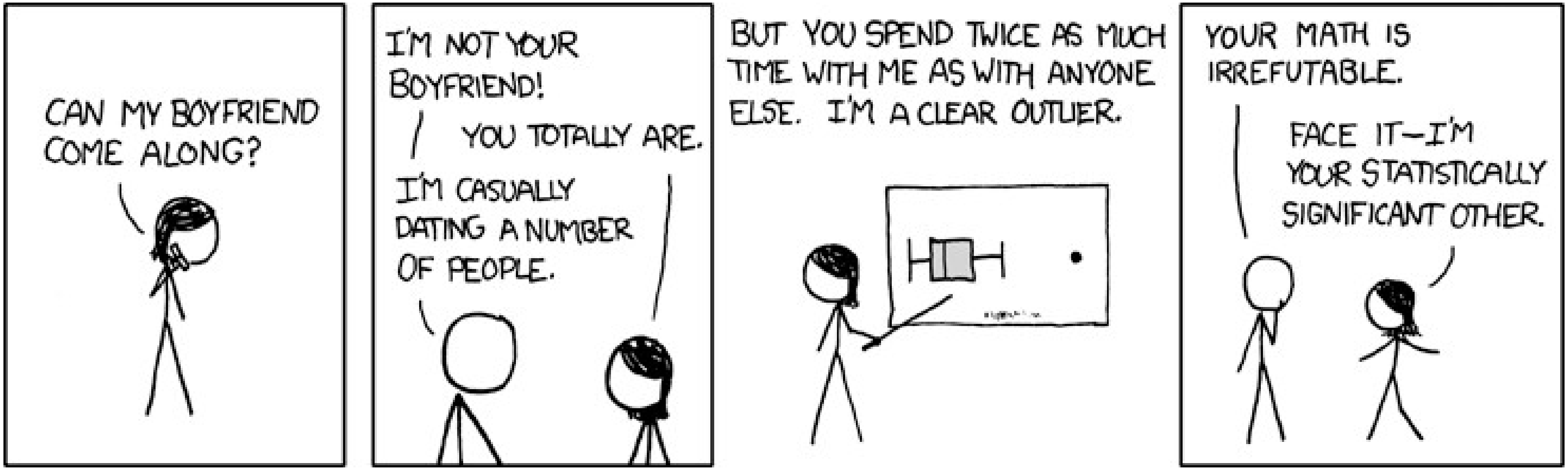
- IQR on the example below:
  - Q2 (Median) is 0
  - Q1 is -1, Q3 is 1

    → everything outside [-1.5,+1.5] is an outlier

    → there are no outliers in this example



Time for a Short Break



http://xkcd.com/539/

Distance-based Approaches

- Data is represented as a vector of features

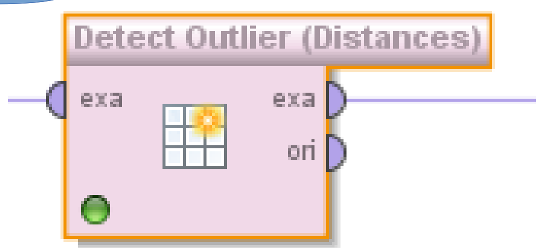
- Various approaches
  - Nearest-neighbor based
  - Density based
  - Clustering based
  - Model based

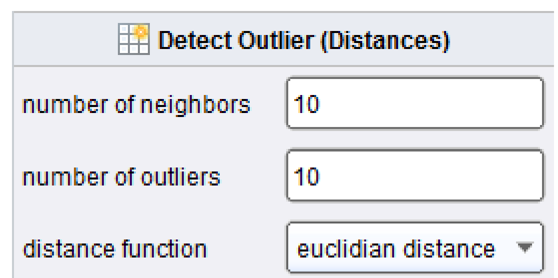
Nearest-Neighbor Based Approach

- Approach:
  - Compute the distance between every pair of data points
  - There are various ways to define outliers:
    - Data points for which there are fewer than *p* neighboring points within a distance *D*
    - The top n data points whose distance to the k^{th} nearest neighbor is greatest
    - The top n data points whose average distance to the k nearest neighbors is greatest

RapidMiner

Package PyOD



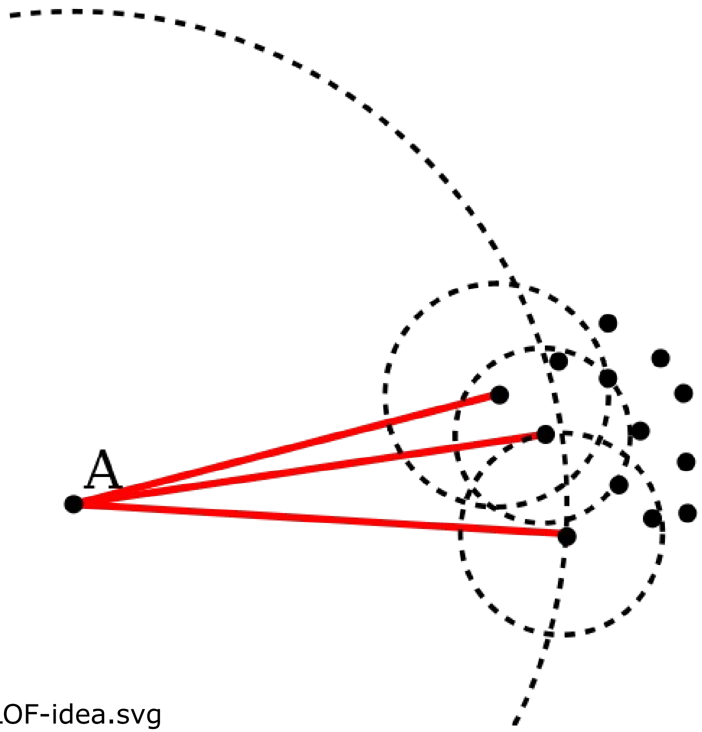


Density-based: LOF approach

- For each point, compute the density of its local neighborhood
  - if that density is higher than the average density, the point is in a cluster
  - if that density is lower than the average density, the point is an outlier
- Compute local outlier factor (LOF) of a point A
  - ratio of average density of A's neighbors to density of point A
- Outliers are points with large LOF value
  - typical: larger than 1

LOF: Illustration

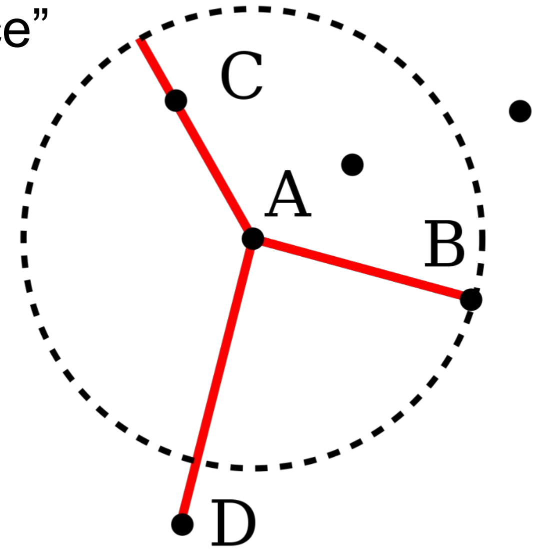
- Using 3 nearest neighbors
- We compute
  - the average density of A
  - the average density of A's neighbors
- If the density of A is lower than the neighbors' density
  - A might be an outlier



http://commons.wikimedia.org/wiki/File:LOF-idea.svg

LOF: Defining Density

- LOF uses a concept called “reachability distance”
- All points within the k-neighborhood have the same k-distance
  - in the example: $d_3(A,B) = d_3(A,C)$
- Reachability distance $rd_k(A,B)$:
  - distance of A,B, lower bound by $d_k(B)$
  - $rd_k(A,B) = \max(d_k(B), \text{distance}(A,B))$
- In the example:
  - $rd_k(D,A) = d(D,A)$, but
  - $rd_k(C,A) = \text{k-distance}(A)$
- Rationale: all sufficiently close points are regarded as equally close
  - lessens the impact of small variations

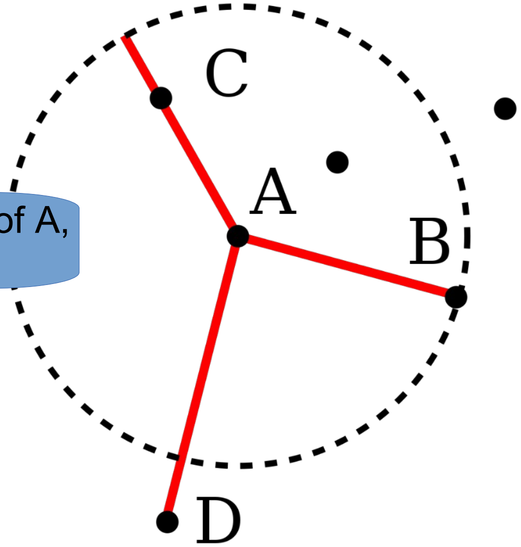


https://commons.wikimedia.org/wiki/File:Reachability-distance.svg

LOF: Defining Density

- Average reachability distance
  - $avgrd_k(A) = \frac{\sum_{P: k \text{ nearest neighbors of } A} rd_k(A,P)}{|N_k(A)|}$

no. of k nearest neighbors of A, usually =k



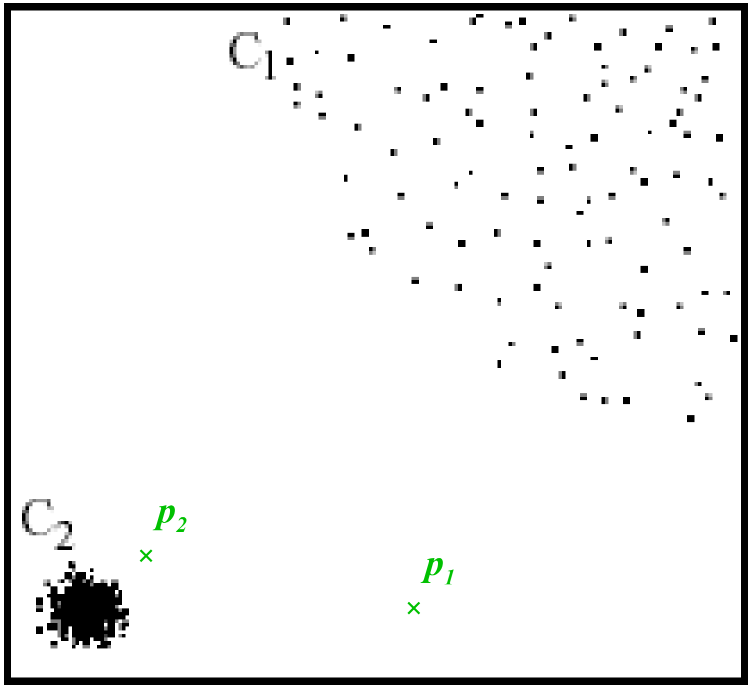
- Density is defined as the inverse
  - idea: the larger the avg. reachability distance, the sparser the region in which the data point lies
  - *local reachability density* $lrb_k(A) = 1/avgrd_k(A)$

- Local outlier factor: relation of density of A's neighbors to A's density:

$$LOF_k(A) = \frac{\sum_{P: k \text{ nearest neighbors of } A} \frac{lrb_k(P)}{lrb_k(A)}}{|N_k(A)|} = \frac{\sum_{P: k \text{ nearest neighbors of } A} lrb_k(P)}{|N_k(A)| \cdot lrb_k(A)}$$

Nearest-Neighbor vs. LOF

- With kNN, only p_1 is found as an outlier
  - there are enough near neighbors for p_2 in cluster C_2
- With LOF, both p_1 and p_2 are found as outliers

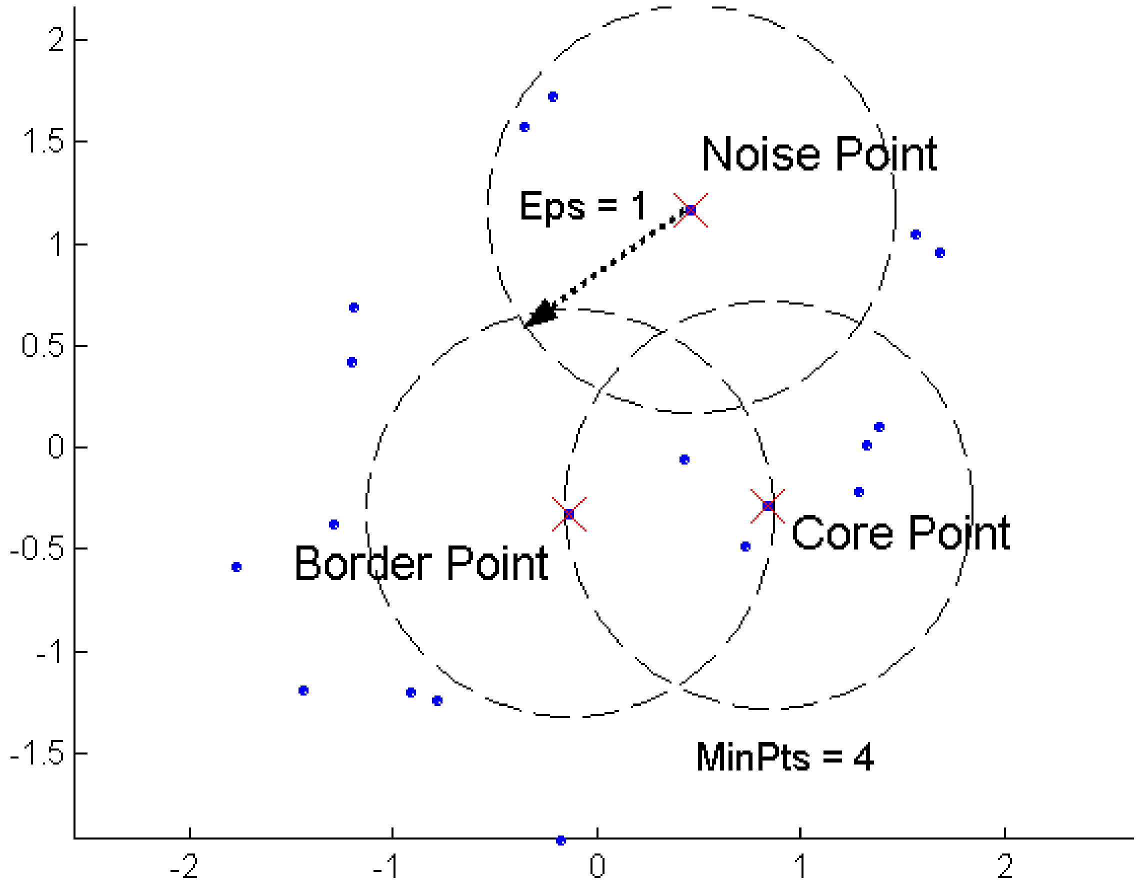


`sklearn.neighbors.LocalOutlierFactor`

Recap: DBSCAN

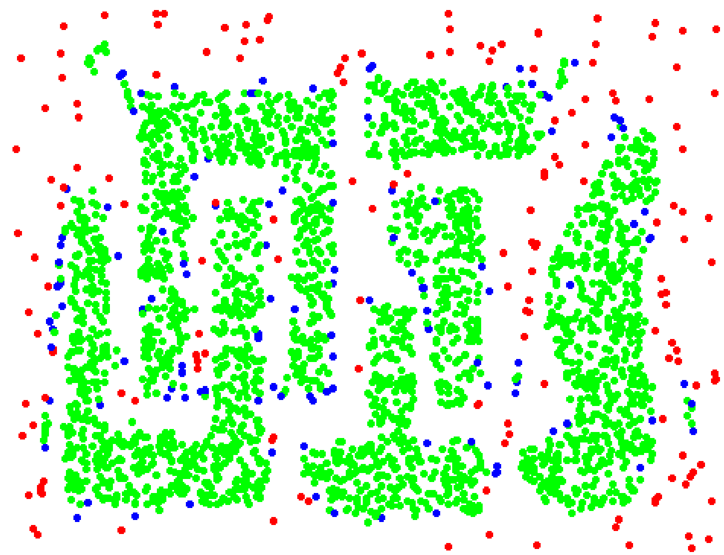
- DBSCAN is a density-based algorithm
  - Density = number of points within a specified radius (Eps)
- Divides data points in three classes:
  - A point is a core point if it has more than a specified number of points (MinPts) within Eps
    - These are points that are at the interior of a cluster
  - A border point has fewer than MinPts within Eps, but is in the neighborhood of a core point
  - A noise point is any point that is not a core point or a border point

Recap: DBSCAN



Recap: DBSCAN

Original Points

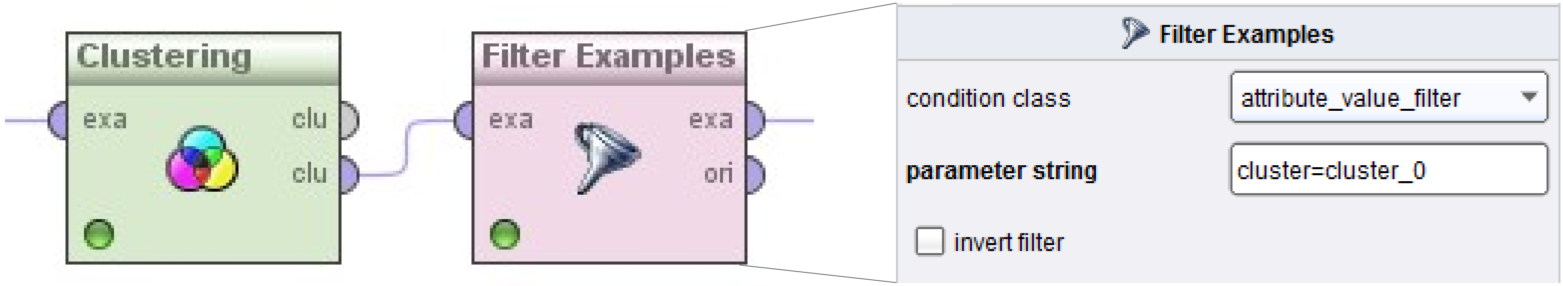


Point types: core, border and noise

Eps = 10, MinPts = 4

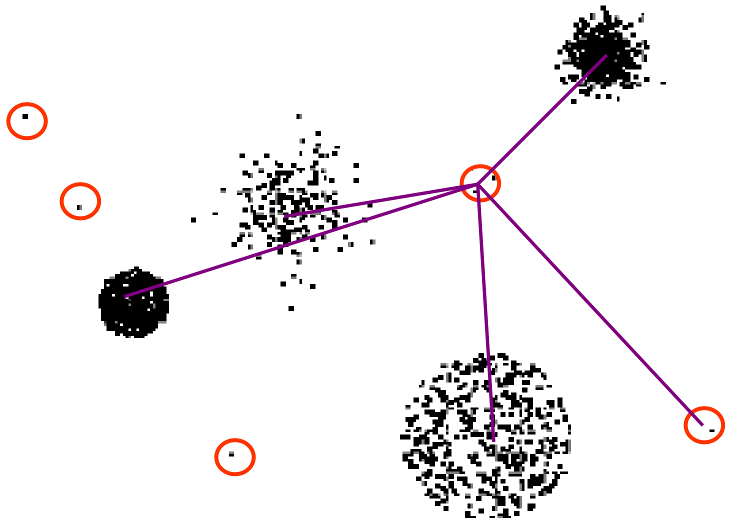
DBSCAN for Outlier Detection

- DBSCAN directly identifies noise points
  - these are outliers not belonging to any cluster
    - in RapidMiner: assigned to cluster 0
    - in scikit-learn: label -1
  - allows for performing outlier detection directly



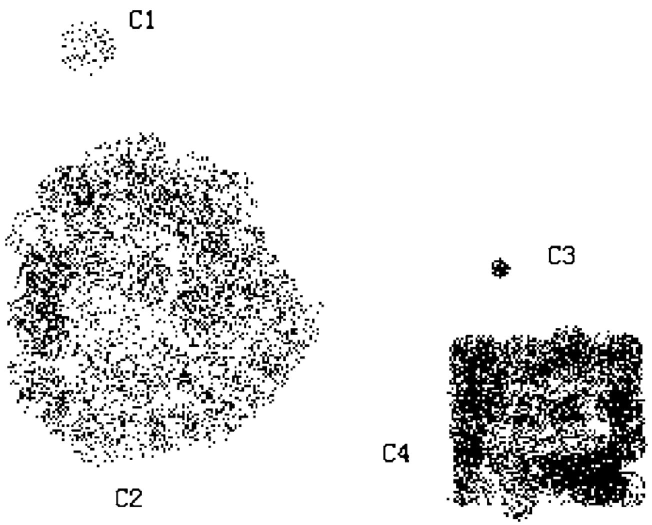
Clustering-based Outlier Detection

- Basic idea:
  - Cluster the data into groups of different density
  - Choose points in small cluster as candidate outliers
  - Compute the distance between candidate points and non-candidate clusters
  - If candidate points are far from all other non-candidate points, they are outliers



Clustering-based Local Outlier Factor

- Idea: anomalies are data points that are
  - in a very small cluster or
  - far away from other clusters



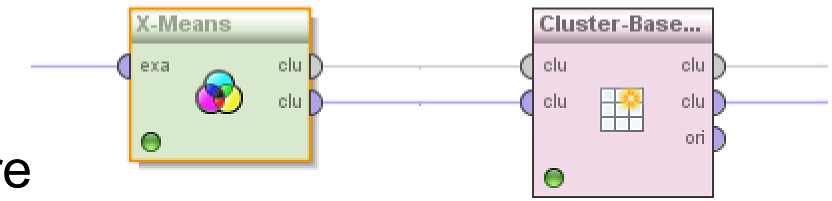
- CBLOF is run on clustered data
- Assigns a score based on
  - the size of the cluster a data point is in
  - the distance of the data point to the next large cluster

Clustering-based Local Outlier Factor

- General process:
  - first, run a clustering algorithm (of your choice)
  - then, apply CBLOF

Package PyOD

- Result: data points with outlier score



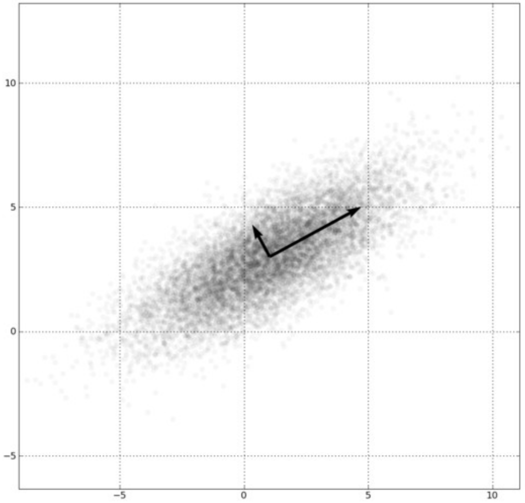
Data View | Meta Data View | Plot View | Advanced Charts | Annotations

ExampleSet (208 examples, 4 special attributes, 60 regular attributes)

Row No.	cluster	class	id	outlier	attribute_1	attribute_2	attribute_3	attribute_4	attribute_5	attribute_6	attribute_7
1	cluster_0	Rock	1	38.516	0.020	0.037	0.043	0.021	0.095	0.099	0.154
2	cluster_2	Rock	2	65.452	0.045	0.052	0.084	0.069	0.118	0.258	0.216
3	cluster_2	Rock	3	75.490	0.026	0.058	0.110	0.108	0.097	0.228	0.243
4	cluster_0	Rock	4	45.112	0.010	0.017	0.062	0.020	0.020	0.037	0.110
5	cluster_2	Rock	5	68.759	0.076	0.067	0.048	0.039	0.059	0.065	0.121
6	cluster_2	Rock	6	69.133	0.029	0.045	0.028	0.017	0.038	0.099	0.120
7	cluster_2	Rock	7	65.306	0.032	0.096	0.132	0.141	0.167	0.171	0.073
8	cluster_3	Rock	8	48.851	0.052	0.055	0.084	0.032	0.116	0.092	0.103
9	cluster_3	Rock	9	52.493	0.022	0.038	0.048	0.048	0.065	0.059	0.075
10	cluster_3	Rock	10	48.437	0.016	0.017	0.035	0.007	0.019	0.067	0.106
11	cluster_1	Rock	11	68.168	0.004	0.006	0.015	0.034	0.031	0.028	0.040
12	cluster_3	Rock	12	46.765	0.012	0.031	0.017	0.031	0.036	0.010	0.018
13	cluster_2	Rock	13	56.138	0.008	0.009	0.006	0.025	0.034	0.055	0.053
14	cluster_1	Rock	14	69.857	0.009	0.006	0.025	0.049	0.120	0.159	0.139
15	cluster_1	Rock	15	84.639	0.012	0.043	0.060	0.045	0.060	0.036	0.053
16	cluster_0	Rock	16	44.131	0.030	0.062	0.065	0.092	0.162	0.229	0.218
17	cluster_0	Rock	17	33.057	0.035	0.012	0.019	0.047	0.074	0.118	0.168

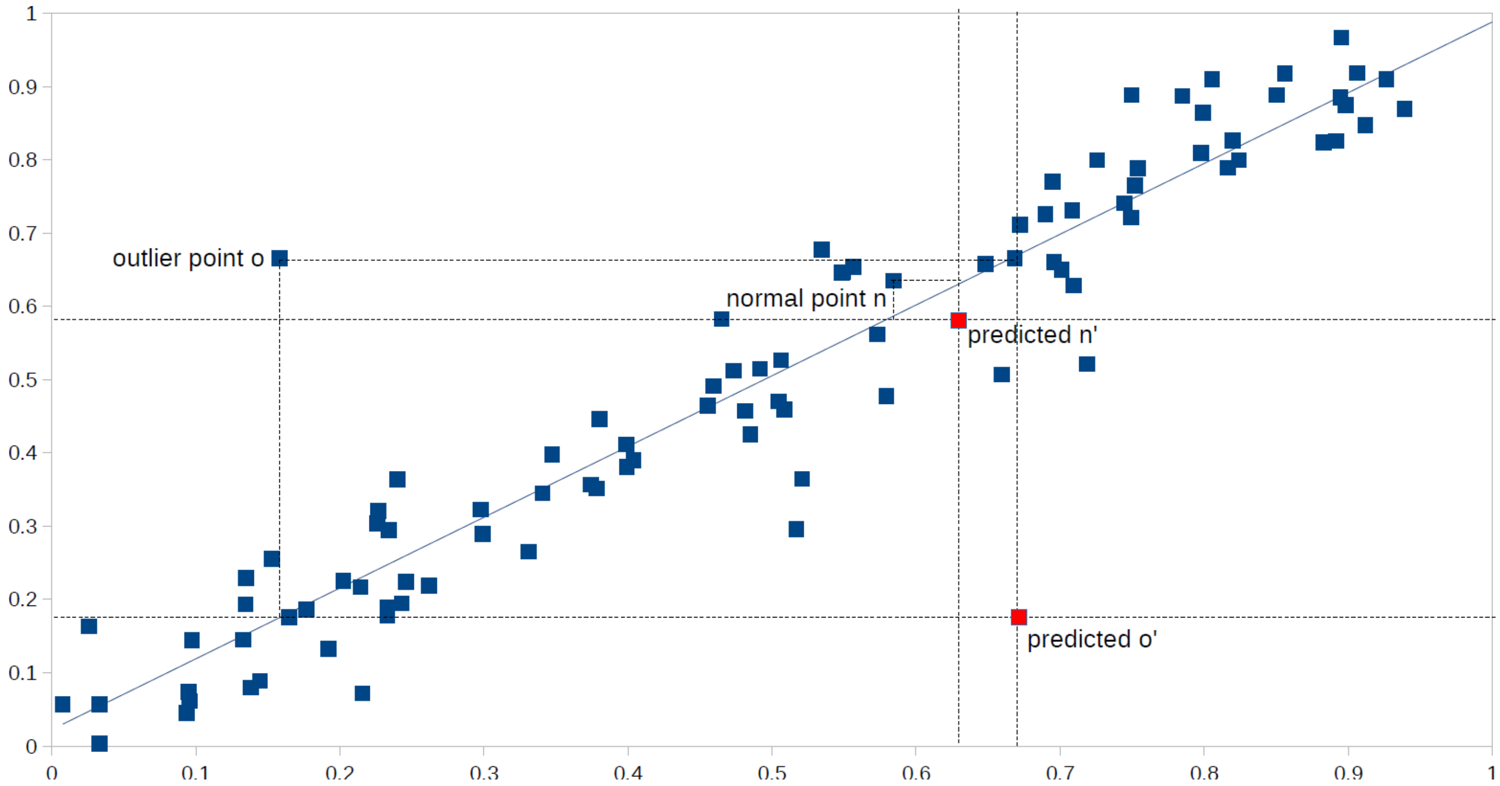
PCA and Reconstruction Error

- Recap: PCA tries to capture most dominant variations in the data
  - those can be seen as the “normal” behavior
- Reconstruct original data point by inversing PCA
  - close to original: normally behaving data point
  - far from original: unnormally behaving data point



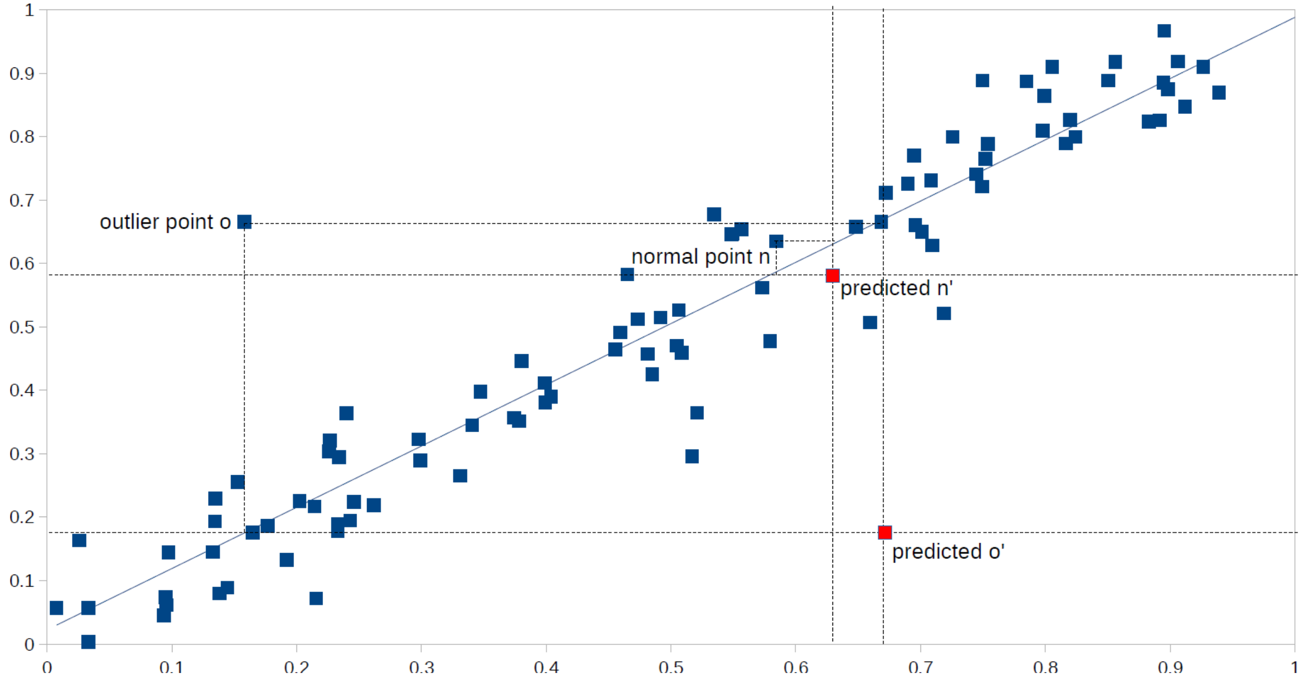
Model-based Outlier Detection (ALSO)

- Idea: there is a model underlying the data
  - Data points deviating from the model are outliers

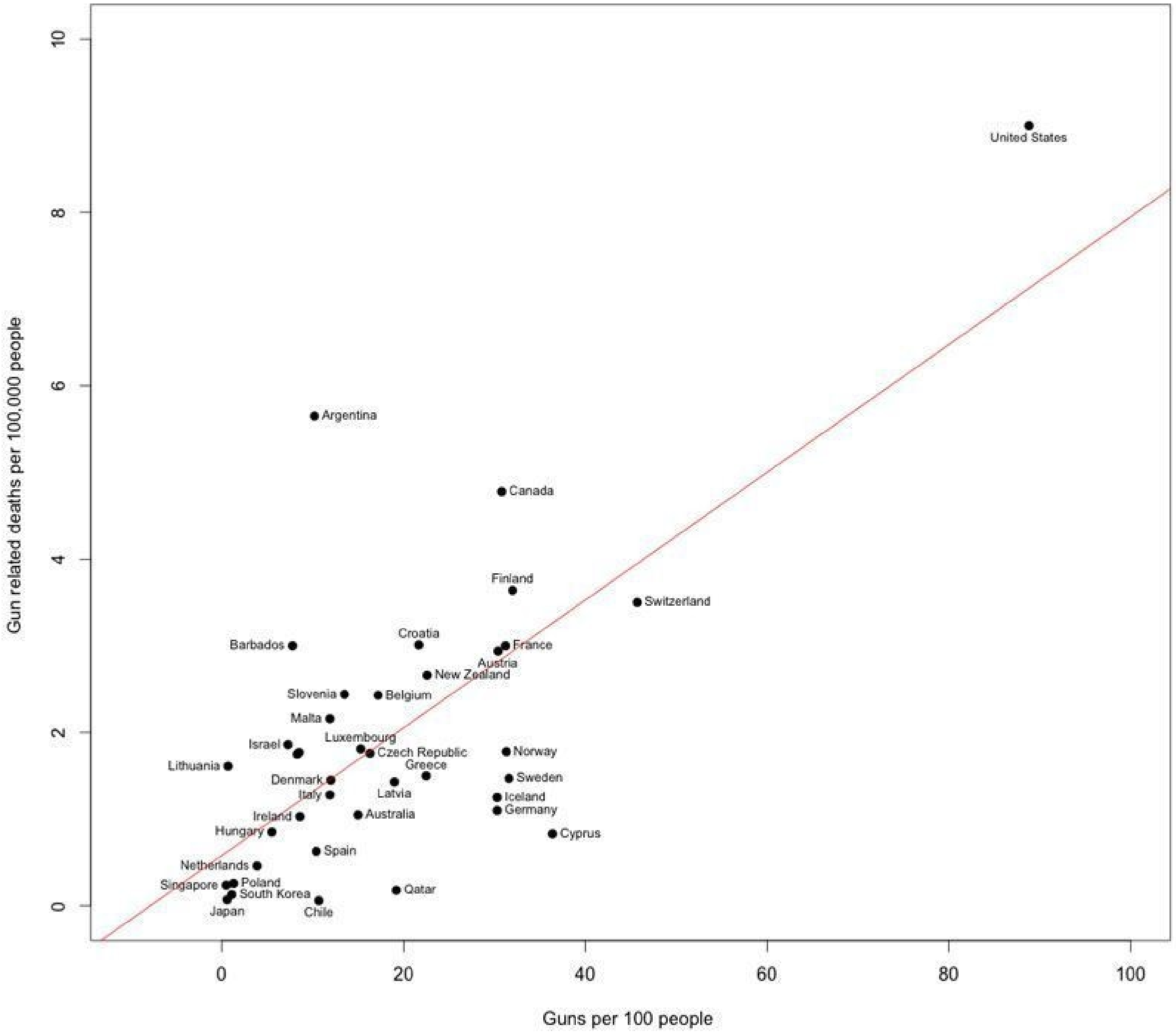


Model-based Outlier Detection (ALSO)

- ALSO (Attribute-wise Learning for Scoring Outliers)
  - Learn a model for each attribute given all other attributes
  - Use model to predict expected value
  - Deviation between actual and predicted value → outlier



Interpretation: What is an Outlier? (recap)



Model-based Outlier Detection (ALSO)

- For each data point *i*, compute vector of predictions *i'*
- Outlier score: Euclidean distance of *i* and *i'*
  - in z-transformed space

$$o_{unweighted}(i) := \sqrt{\sum_{k=1}^{n} (i_k - i'_k)^2}$$

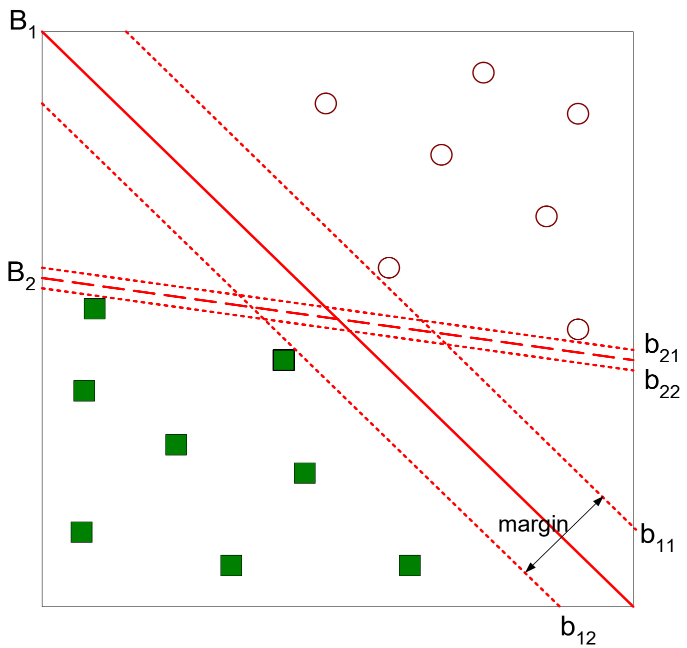
- Refinement: assign weights to attributes
  - given the strength of the pattern learned
  - measure: RRSE

$$o(i) := \sqrt{\frac{1}{\sum_{k=1}^{n} w_k} \sum_{k=1}^{n} w_k \cdot (i_k - i'_k)^2},$$

- Rationale:
  - ignores deviations on unpredictable attributes (e.g., database IDs)
  - for an outlier, require *both* a strong pattern *and* a strong deviation

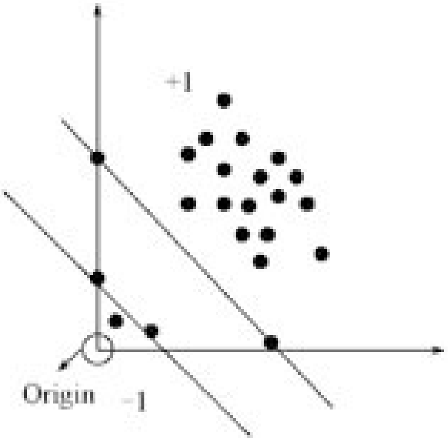
One-Class Support Vector Machines

- Recap: Support Vector Machines
  - Find a maximum margin hyperplane to separate two classes
  - Use a transformation of the vector space
    - Thus, non-linear boundaries can be found



One-Class Support Vector Machines

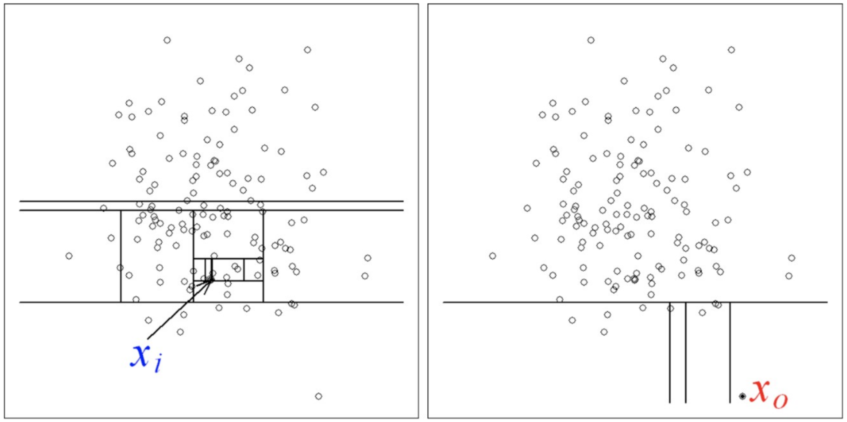
- One-Class Support Vector Machines
  - Find best hyperplane that separates the origin from the rest of the data
    - Maximize margin
    - Minimize errors
  - Points on the same side as the origin are outliers

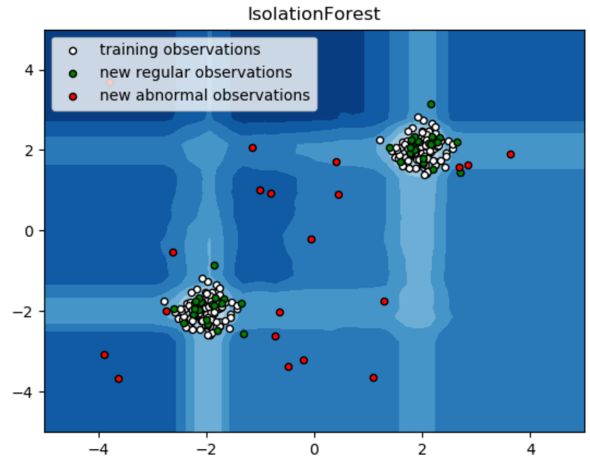


- Recap: SVMs require extensive parameter tuning
  - Difficult to automatize for anomaly detection,
    since we have no training data

Isolation Forests

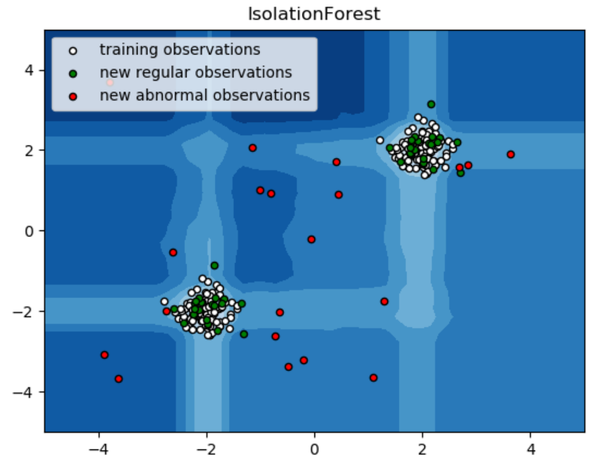
- Isolation tree:
  - a decision tree that has only leaves with one example each
- Isolation forests:
  - train a set of random isolation trees
- Idea:
  - path to outliers in a tree is shorter than path to normal points
  - across a set of random trees, average path length is an outlier score





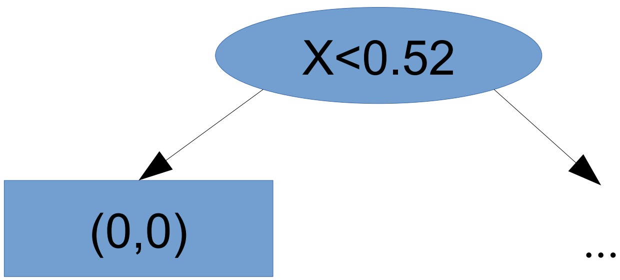
Isolation Forest

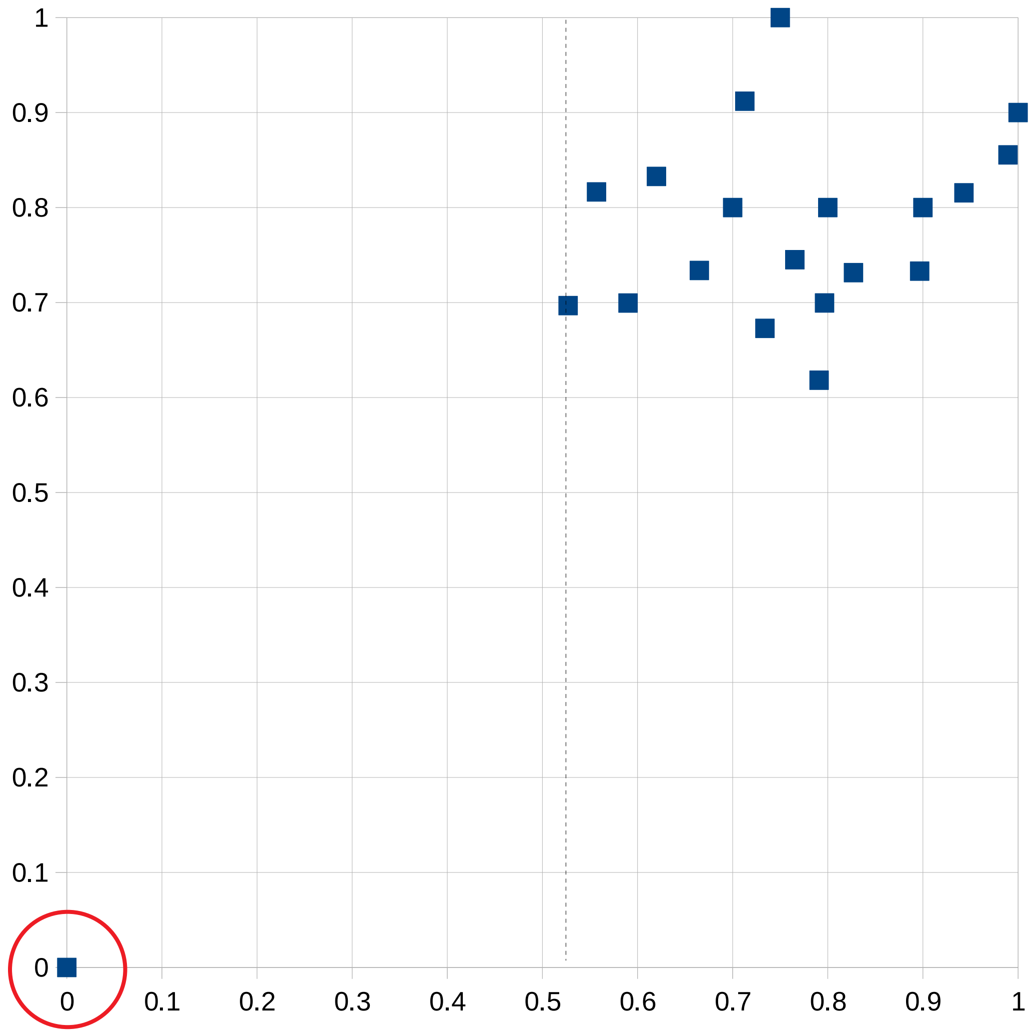
- Training a single isolation tree
  - for each leaf node w/ more than one data point
    - pick an attribute Att and a value V at random
    - create inner node with test Att<V
      - train isolation tree for each subtree
- Output
  - A tree with just one instance per node
  - Usually, an upper limit on height is used



Isolation Forest

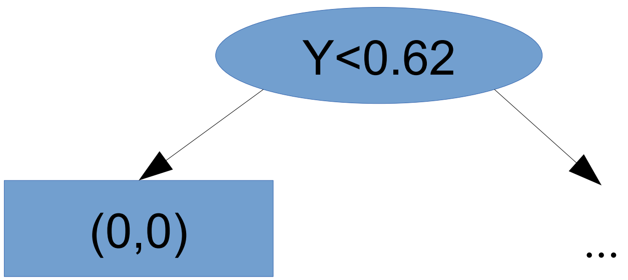
- Probability of (0,0) ending in a leaf at height 1
  - pick Att X, pick V<0.52

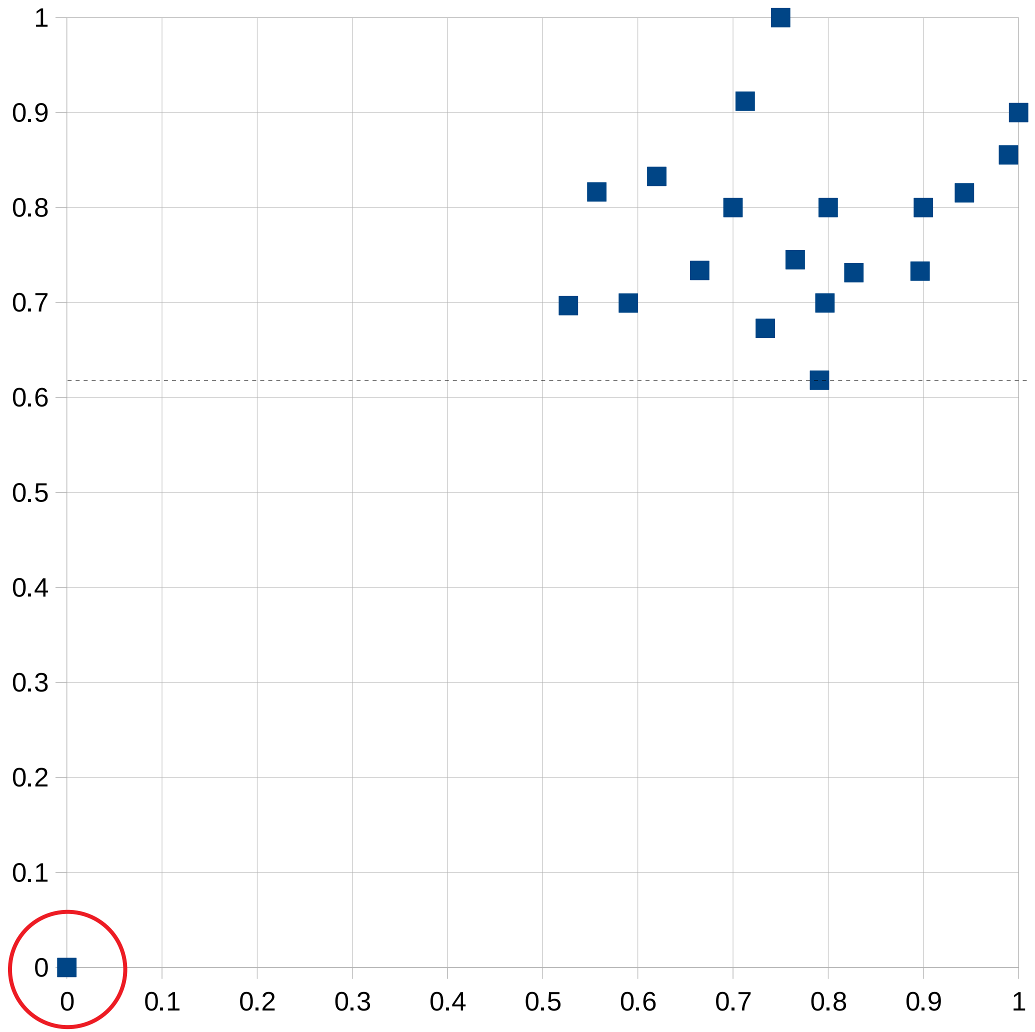




Isolation Forest

- Probability of (0,0) ending in a leaf at height 1
  - pick Att Y, pick V<0.62

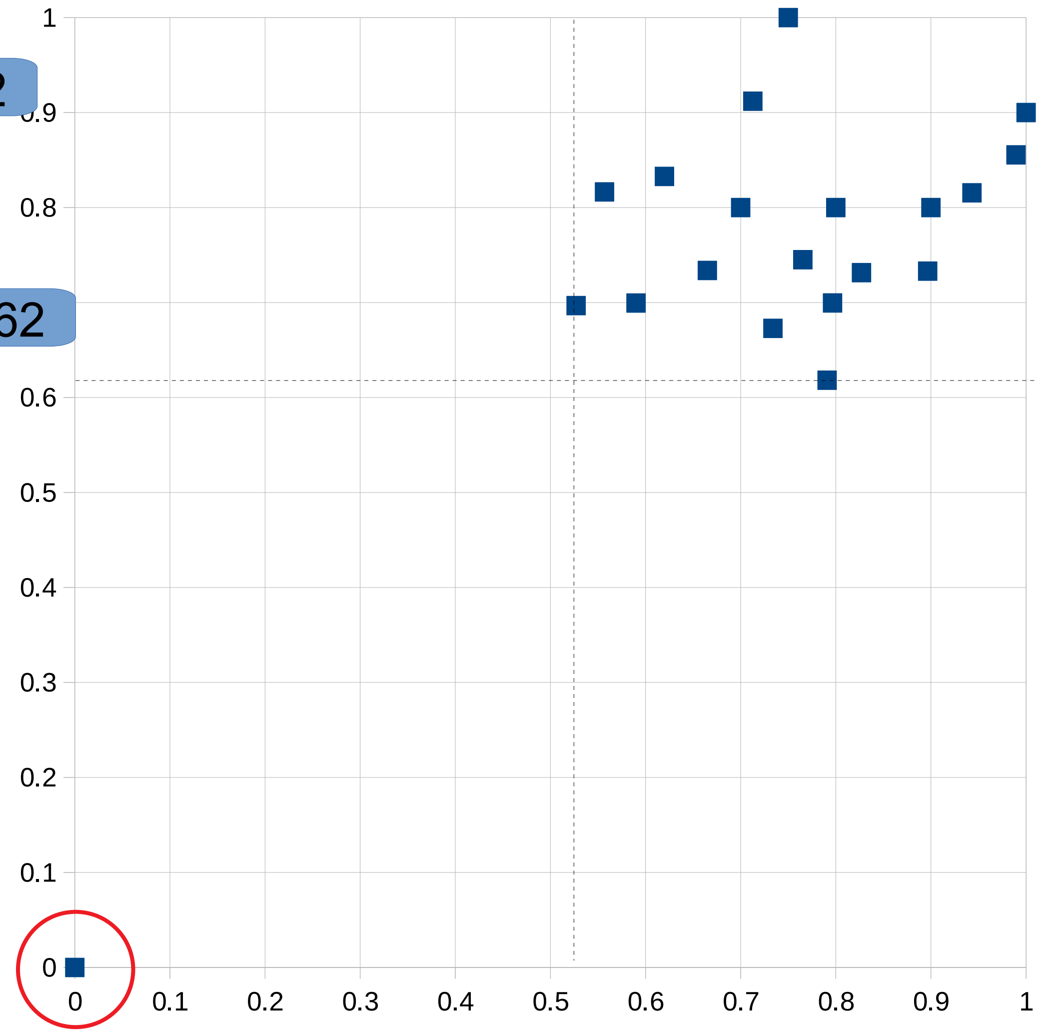




Isolation Forest

- Probability of (0,0) ending in a leaf at height 1
  - 0.5 – pick Att X, pick V<0.52, or
  - 0.5 – pick Att Y, pick V<0.62
- 0.5*0.52 + 0.5*0.62

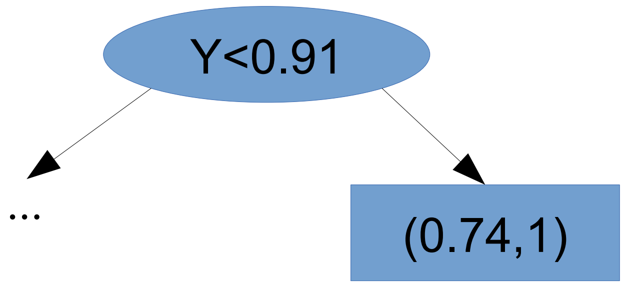
  → 0.57

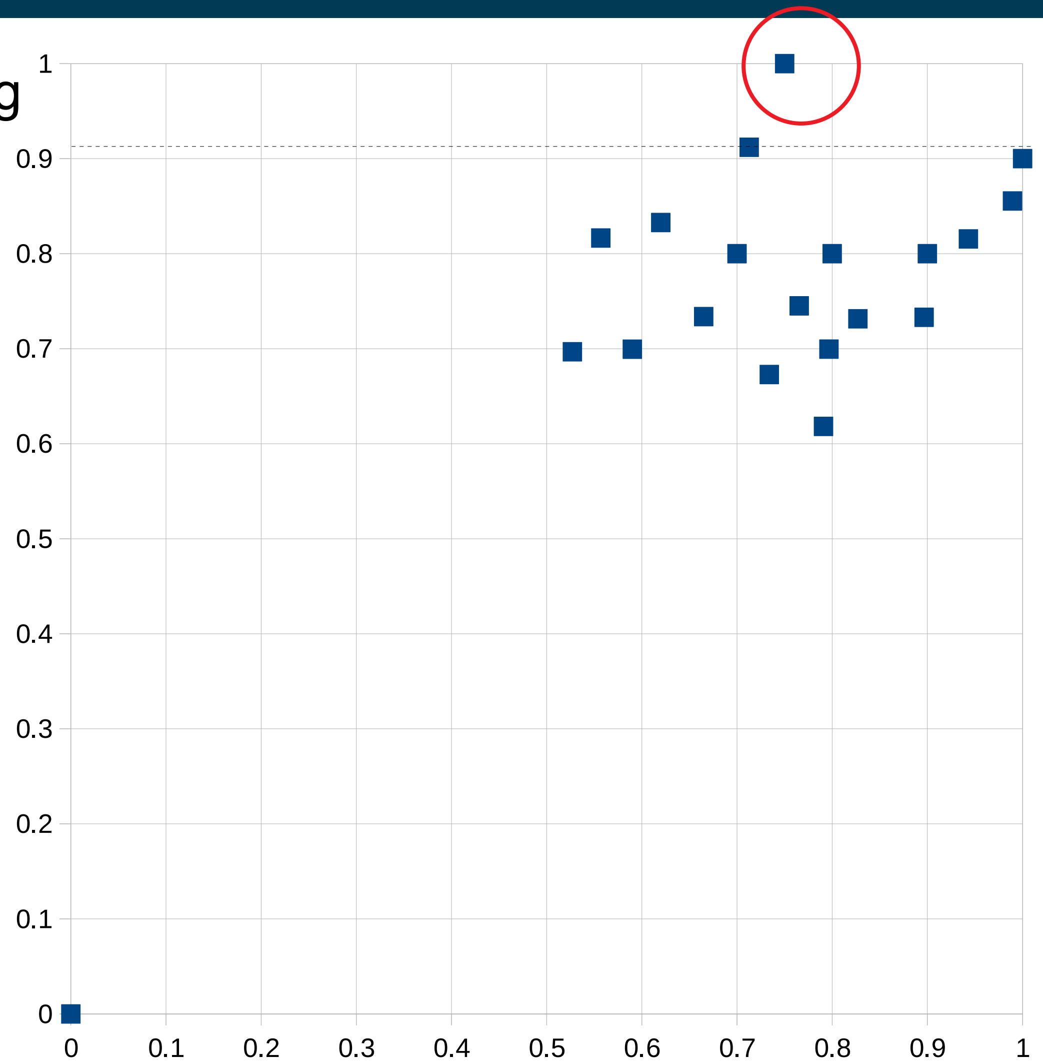


Isolation Forest

- Probability of (0.74,1) ending in a leaf at height 1
  - pick Att Y, pick V>0.91
- 0.5 * 0.09

  → 0.045



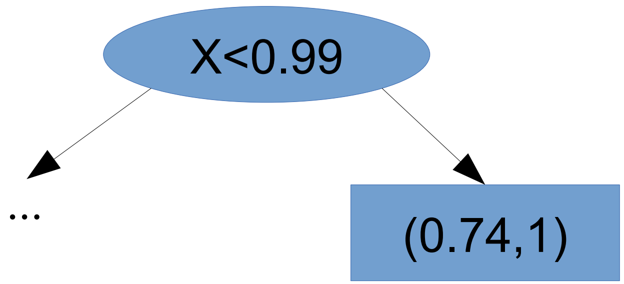


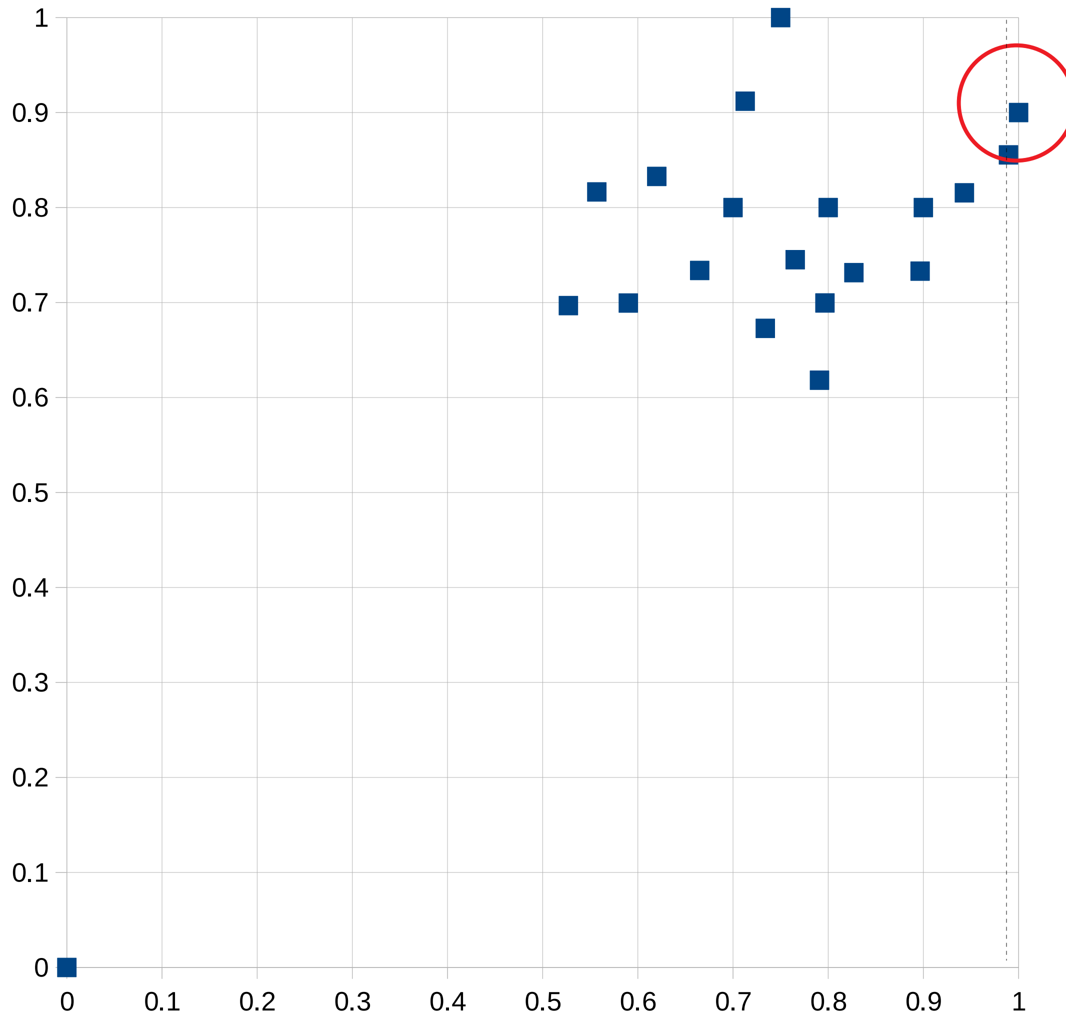
Isolation Forest

- Probability of (1,0.9) ending in a leaf at height 1
  - pick Att X, pick V>0.98

- 0.5 * 0.02

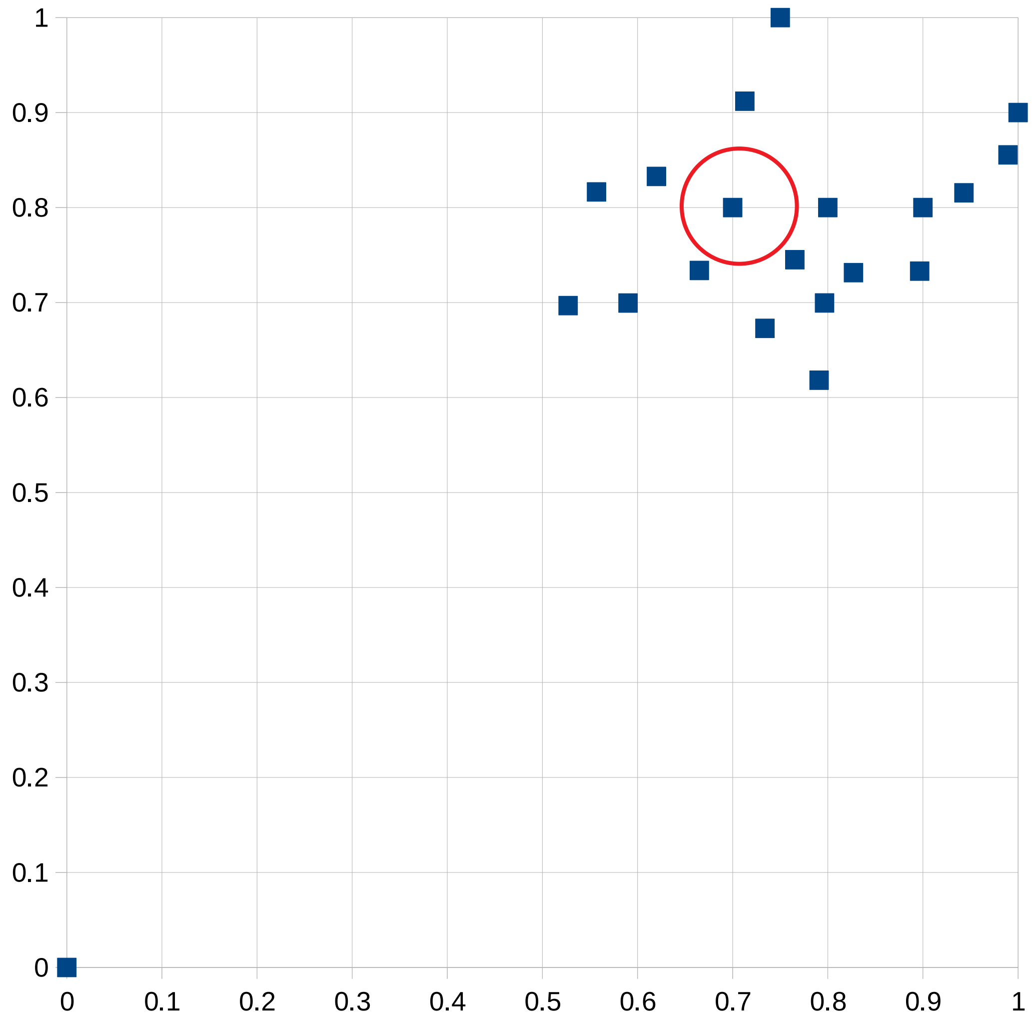
  → 0.01





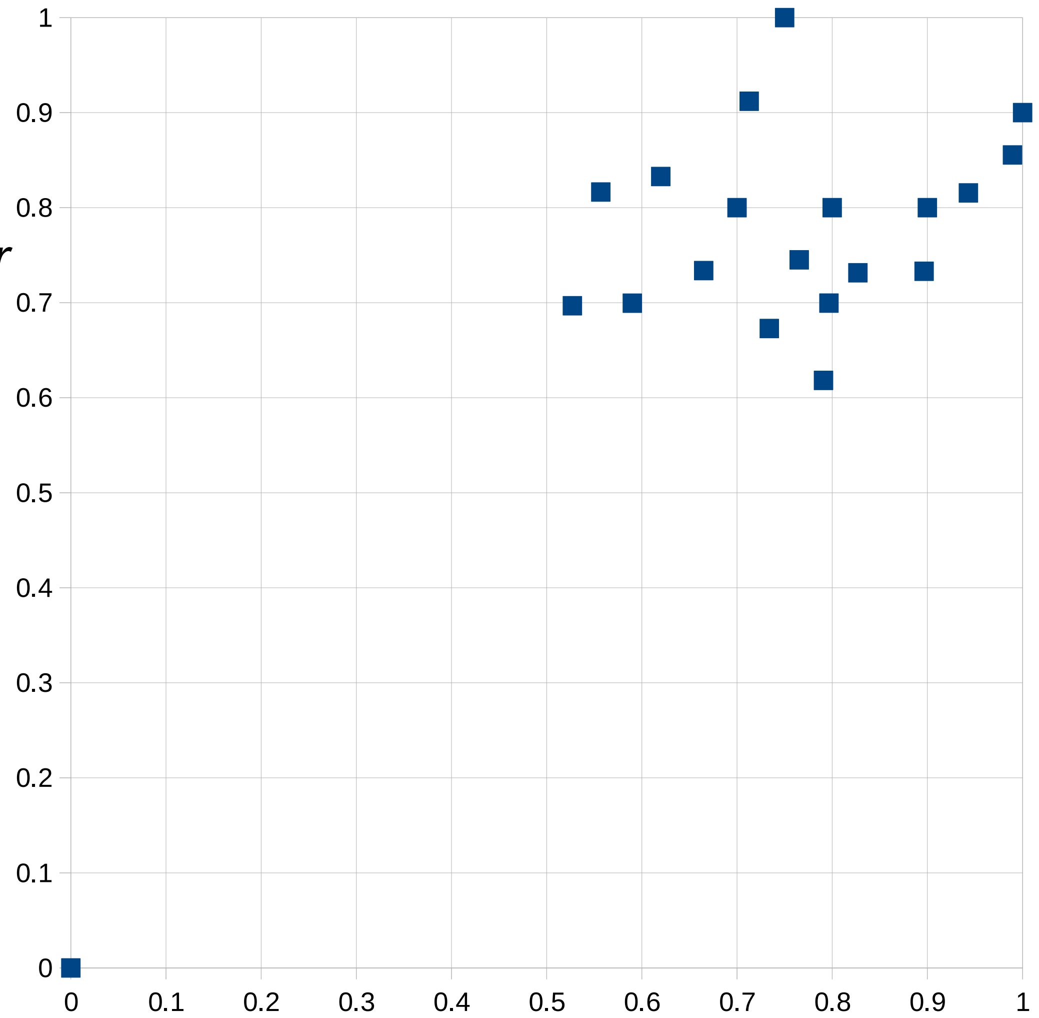
Isolation Forest

- Probability of any other data point ending in a leaf at height 1
  - this is not possible!
  - at least two tests are necessary



Isolation Forest

- Observations
  - data points in dense areas need more tests
    - i.e., they end up *deeper* in the trees
  - data points far away from the rest have a higher probability to be isolated earlier
    - i.e., they end up *higher* in the trees



High-Dimensional Spaces

- A large number of attributes may cause problems
  - many anomaly detection approaches use distance measures
  - those get problematic for very high-dimensional spaces
  - meaningless attributes obscure the distances
- Practical hint:
  - perform dimensionality reduction first
  - i.e., feature subset selection, PCA
  - note: anomaly detection is unsupervised
    - thus, supervised selection (like forward/backward selection) does not work

High-Dimensional Spaces

- Recap: attributes may have different scales
  - Hence, different attributes may have different contributions to outlier scores
- Compare the following two datasets:

- Baden-Württemberg
  - population = 10,569,111
  - area = 35,751.65 km²
- Bavaria
  - population = 12,519,571
  - area = 70,549.44 km²
- ...

- Baden-Württemberg
  - population = 10,569,111
  - area = 35,751,650,000 m²
- Bavaria
  - population = 12,519,571
  - area = 70,549,440,000 m²
- ...

High-Dimensional Spaces

- Baden-Württemberg
  - population = 10,569,111
  - area = 35,751.65 km²
- Bavaria
  - population = 12,519,571
  - area = 70,549.44 km²
- ...

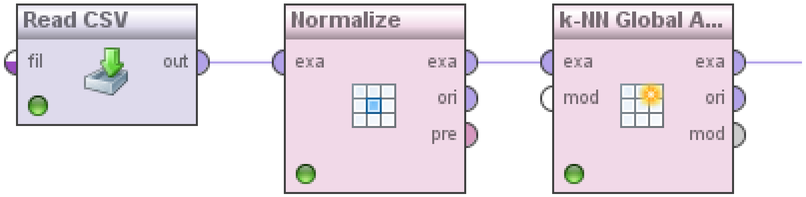
- Baden-Württemberg
  - population = 10,569,111
  - area = 35,751,650,000 m²
- Bavaria
  - population = 12,519,571
  - area = 70,549,440,000 m²
- ...

- In the second set, outliers in the population are unlikely to be discovered
  - Even if we change the population of Bavaria by a factor of 100, the Euclidean distance does not change much
- Thus, outliers in the population are *masked* by the area attribute

High-Dimensional Spaces

- Solution:
  - Normalization!
- Advised:
  - z-Transformation
  - More robust w.r.t. outliers than simple projection to [0;1]

$$x' = \frac{|x - \mu|}{\sigma}$$

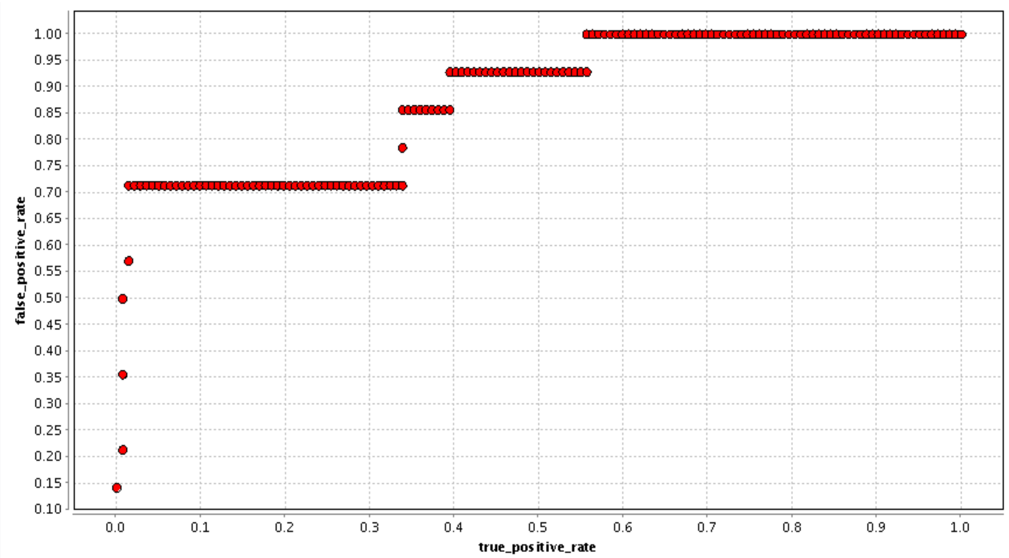


Evaluation Measures

- Anomaly Detection is an *unsupervised* task
- Evaluation: usually on a labeled subsample
- Evaluation Measures:
  - F-measure on outliers
  - Area under ROC curve

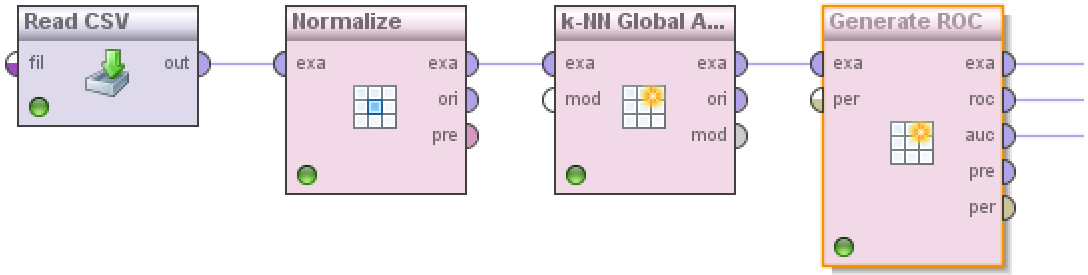
Evaluation Measures

- Anomaly Detection is an *unsupervised* task
- Evaluation: usually on a labeled subsample
  - Note: no splitting into training and test data!
- Evaluation Measures:
  - F-measure on outliers
  - Area under *ROC curve*
  - Plots false positives against true positives



Evaluation Measures

- Anomaly Detection is an *unsupervised* task
- Evaluation: usually on a labeled subsample
  - Note: no splitting into training and test data!
- Evaluation Measures:
  - F-measure on outliers
  - Area under *ROC curve*
  - Plots false positives against true positives

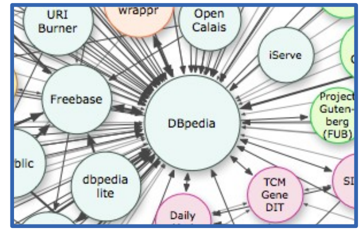


Semi-Supervised Anomaly Detection

- All approaches discussed so far are unsupervised
  - they run fully automatic
  - without human intelligence

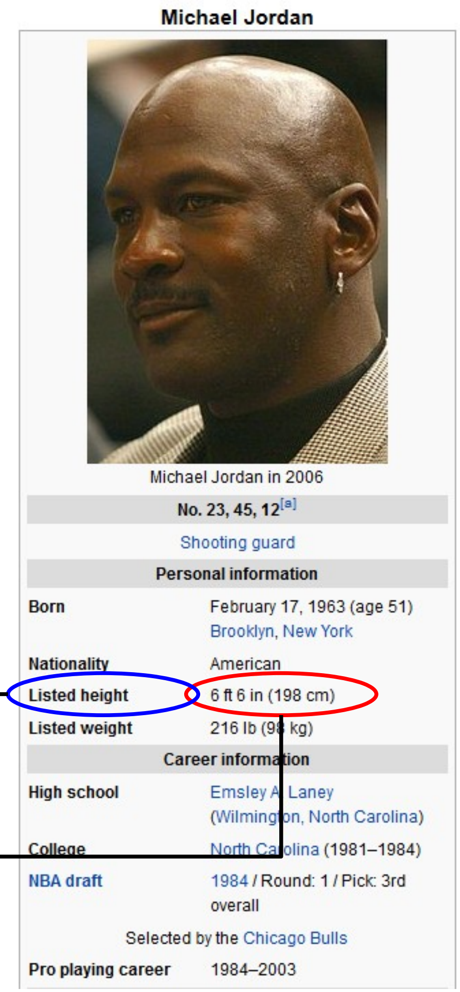
- Semi-supervised anomaly detection
  - experts manually label some data points as being outliers or not
    → anomaly detection becomes similar to a classification task
    - the class label being *outlier/non-outlier*
  - Challenges:
    - Outliers are scarce → unbalanced dataset
    - Outliers are not a class

Example: Outlier Detection in DBpedia



- DBpedia
  - extracts data from infoboxes in Wikipedia
  - based on crowd-sourced mappings to an ontology
- Example
  - Wikipedia page on Michael Jordan

```
dbpedia:Michael_Jordan
  dbpedia-owl:height
  "1.981200"^^xsd:double .
```



Dominik Wienand, Heiko Paulheim:
Detecting Incorrect Numerical Data in DBpedia. In: ESWC 2014

Example: Outlier Detection in DBpedia

- DBpedia is based on heuristic extraction
- Several things can go wrong
  - wrong data in Wikipedia
  - unexpected number/date formats
  - errors in the extraction code
  - …

- Can we use anomaly detection to remedy the problem?

Dominik Wienand, Heiko Paulheim:
Detecting Incorrect Numerical Data in DBpedia. In: ESWC 2014

Example: Outlier Detection in DBpedia

- Challenge
  - Wikipedia is made for humans, not machines
  - Input format in Wikipedia is not constrained

- The following are all valid representations of the same height value (and perfectly understandable by humans)
  - `6 ft 6 in`, `6ft 6in`, `6'6''`, `6'6”`, `6´6´´`, ...
  - `1.98m`, `1,98m`, `1m 98`, `1m 98cm`, `198cm`, `198 cm`, ...
  - `6 ft 6 in (198 cm)`, `6ft 6in (1.98m)`, `6'6'' (1.98 m)`, ...
  - `6 ft 6 in`[1], `6 ft 6 in` [citation needed], ...
  - ...

Dominik Wienand, Heiko Paulheim:
Detecting Incorrect Numerical Data in DBpedia. In: ESWC 2014

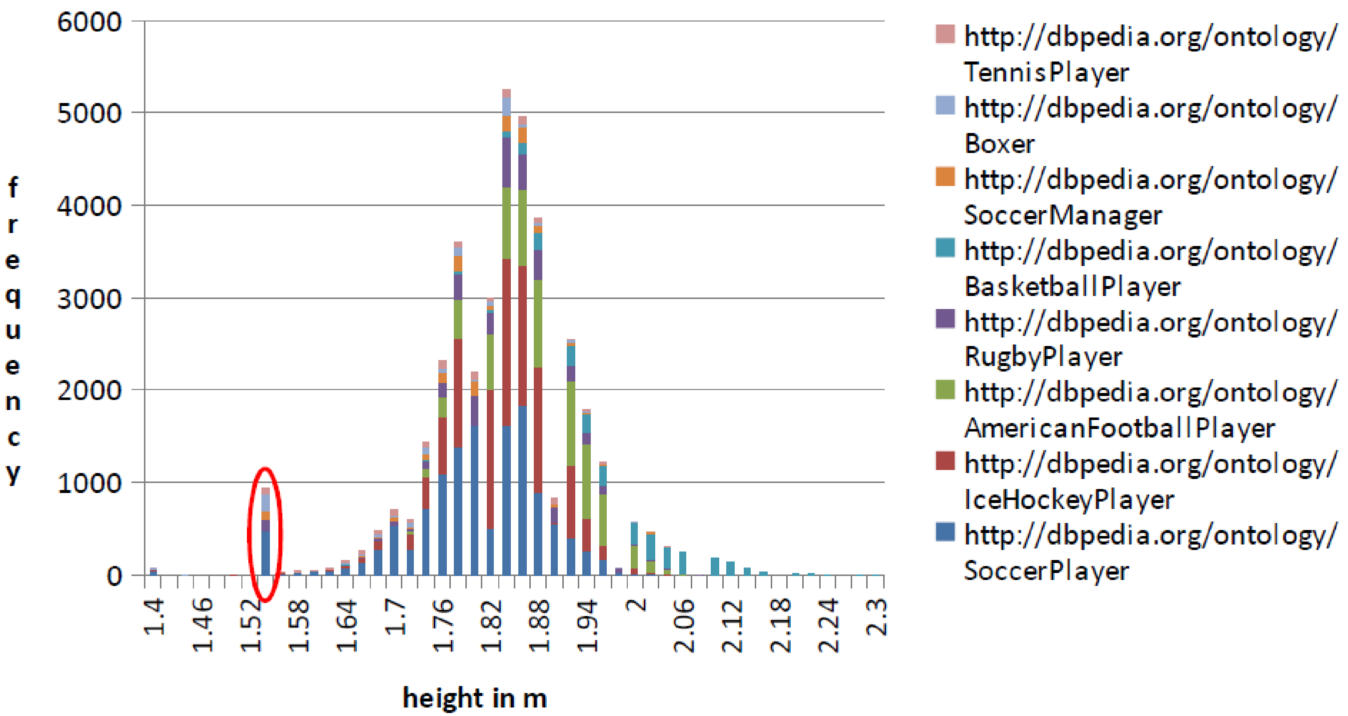
Example: Outlier Detection in DBpedia

- Preprocessing: split data for different types
  - height is used for persons or buildings
  - population is used for villages, cities, countries, and continents
  - …
- Separate into single distributions
  - makes anomaly detection better
- Result
  - errors are identified at ~90% precision
  - systematic errors in the extraction code can be found

Dominik Wienand, Heiko Paulheim:
Detecting Incorrect Numerical Data in DBpedia. In: ESWC 2014

Example: Outlier Detection in DBpedia

- Footprint of a systematic error

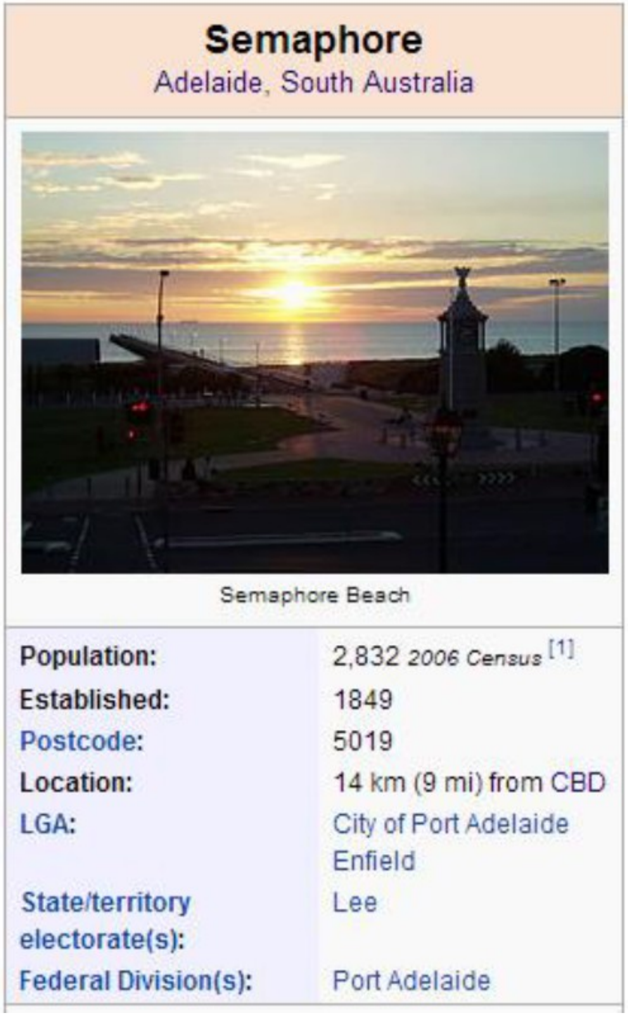


Dominik Wienand, Heiko Paulheim:
Detecting Incorrect Numerical Data in DBpedia. In: ESWC 2014

Example: Outlier Detection in DBpedia

- Typical error sources
  - unit conversions gone wrong (e.g., imperial/metric)
  - misinterpretation of numbers
- e.g., village *Semaphore* in Australia
  - population: 28,322,006

    (all of Australia: 23,379,555!)
  - a clear outlier among villages

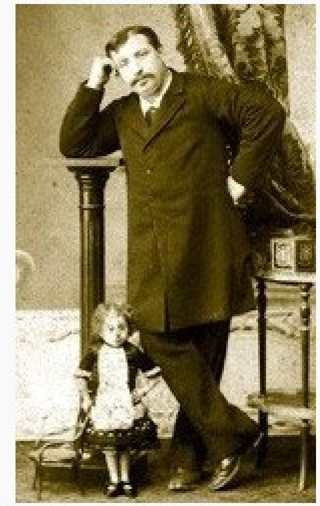


Dominik Wienand, Heiko Paulheim:
Detecting Incorrect Numerical Data in DBpedia. In: ESWC 2014

Errors vs. Natural Outliers

- Hard task for a machine
- e.g., an adult person 58cm high
- e.g., a 7.4m high vehicle

Pauline Musters



Musters next to an average man

Born	February 26, 1876 Ossendrecht, Netherlands
Died	March 1, 1895 (aged 19) New York City
Cause of death	Combination of pneumonia and meningitis
Known for	Shortest verified woman ever
Height	23 inches (58 cm)

Dominik Wienand, Heiko Paulheim:
Detecting Incorrect Numerical Data in DBpedia. In: ESWC 2014

Wrap-up

- Anomaly Detection is useful for
  - data preprocessing and cleansing
  - finding suspect data (e.g., network intrusion, credit card fraud)

- Methods
  - visual/manual
  - statistics based
  - density based
  - model based

Questions?