Intelligent Maintenance and Prognostics for Industrial Systems

Data Pre-processing

Yixiang Huang

2020

Types of Signals

- Event type (e.g. alarm codes)
  - Occurrence at sparse, discrete time
- Value type:
  - 1-D signal (1 sample at a time)
  - Extract features, or used directly
- Waveform type (e.g. vibrations):
  - 1-D signal (n samples at a time)
  - Feature extraction required
- Image type:
  - 2-D signal (m x n samples at a time)
  - feature extraction required

Major Tasks in Data Preprocessing

- Data quality check
- Format and visualization
- Data Cleaning
- Data Integration
- denoise/smooth
- outlier
- missing data / imputation

Data Quality: Why Preprocess the Data?

- Measures for data quality: A multidimensional view
  - Accuracy: correct or wrong, accurate or not
  - Completeness: not recorded, unavailable, …
  - Consistency: some modified but some not, dangling, …
  - Timeliness: timely update?
  - Believability: how trustable the data are correct?
  - Interpretability: how easily the data can be understood?

Data Cleaning

- **Data in the Real World Is Dirty**: Lots of potentially incorrect data, e.g., instrument faulty, human or computer error, transmission error
  - incomplete: lacking attribute values, lacking certain attributes of interest, or containing only aggregate data
    - e.g., *Occupation*=“ ” (missing data)
  - noisy: containing noise, errors, or outliers
    - e.g., *Salary*=“−10” (an error)
  - inconsistent: containing discrepancies in codes or names, e.g.,
    - *Age*=“42”, *Birthday*=“03/07/2010”
    - Was rating “1, 2, 3”, now rating “A, B, C”
    - discrepancy between duplicate records
  - Intentional (e.g., *disguised missing* data)
    - Jan. 1 as everyone’s birthday?

Data Cleaning as a Process

- Data discrepancy detection
  - Caused by
    - poorly designed data entry forms, human error, deliberate errors, data decay (e.g. outdated addresses),
    - inconsistent data representation and inconsistent use of codes,
    - errors in instrumentation devices, system errors, inconsistencies due to data integration
  - Use metadata (e.g., domain, mean, median, mode, symmetric or skewed, range, standard deviation, dependency, distribution)
  - Check field overloading – when developers squeeze new attribute definitions into unused (bit) portions of already defined attributes
    - E.g. using an unused bit of an attribute whose value range uses only, say 31 out of 32 bits

Data Cleaning as a Process

- Data migration and integration
  - Data migration tools: allow transformations to be specified, e.g. replace the string "*gender*" by "*sex*"
  - ETL (Extraction/Transformation/Loading) tools: allow to specify transformations through a graphical user interface
- Integration of the two processes
  - Iterative and interactive, e.g., data cleaning tool that integrates discrepancy detection and transformation
  - Declarative languages for the specification of data transformation operators
    - SQL, algorithms that express data cleaning specification efficiently

Data Integration

- **Data integration:**
  - Combines data from multiple sources into a coherent store
  - Integrate metadata from different sources
- Entity identification problem:
  - *Schema integration* and *object matching*: e.g., A.cust-id ≡ B.cust-#
  - Identify real world entities from multiple data sources, e.g., Bill Clinton = William Clinton
  - Metadata – name, meaning, data type, range, null rules
  - Metadata can help avoid errors in schema integration
  - Metadata may help transform the data
  - When matching attributes from two databases, *structure* of data should be checked

Data Integration

- Detecting and resolving data value conflicts
  - For the same real world entity, attribute values from different sources are different
  - Possible reasons: different representations, different scales, e.g., metric vs. British units

Data Transformation

- Maps the entire set of values of a given attribute to a new set of replacement values
  s.t. each old value can be identified with one of the new values
- Methods
  - Smoothing: Remove noise from data
  - Attribute/feature construction
    - New attributes constructed from the given ones
  - Aggregation: Summarization, data cube construction
  - Normalization: Scaled to fall within a smaller, specified range
    - min-max normalization
    - z-score normalization
    - normalization by decimal scaling
  - Discretization

Normalization

- **Min-max normalization**: to [new_min$_A$, new_max$_A$]

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

  - Ex. Let income range \$12,000 to \$98,000 normalized to [0.0, 1.0]. Then \$73,000 is mapped to $\frac{73{,}600 - 12{,}000}{98{,}000 - 12{,}000}(1.0 - 0) + 0 = 0.716$

- **Z-score normalization** (μ: mean, σ: standard deviation):

$$v' = \frac{v - \mu_A}{\sigma_A}$$

  - Ex. Let μ = 54,000, σ = 16,000. Then $\frac{73{,}600 - 54{,}000}{16{,}000} = 1.225$

- **Normalization by decimal scaling**

$$v' = \frac{v}{10^j}$$ Where j is the smallest integer such that Max(|v'|) < 1

Noisy Data

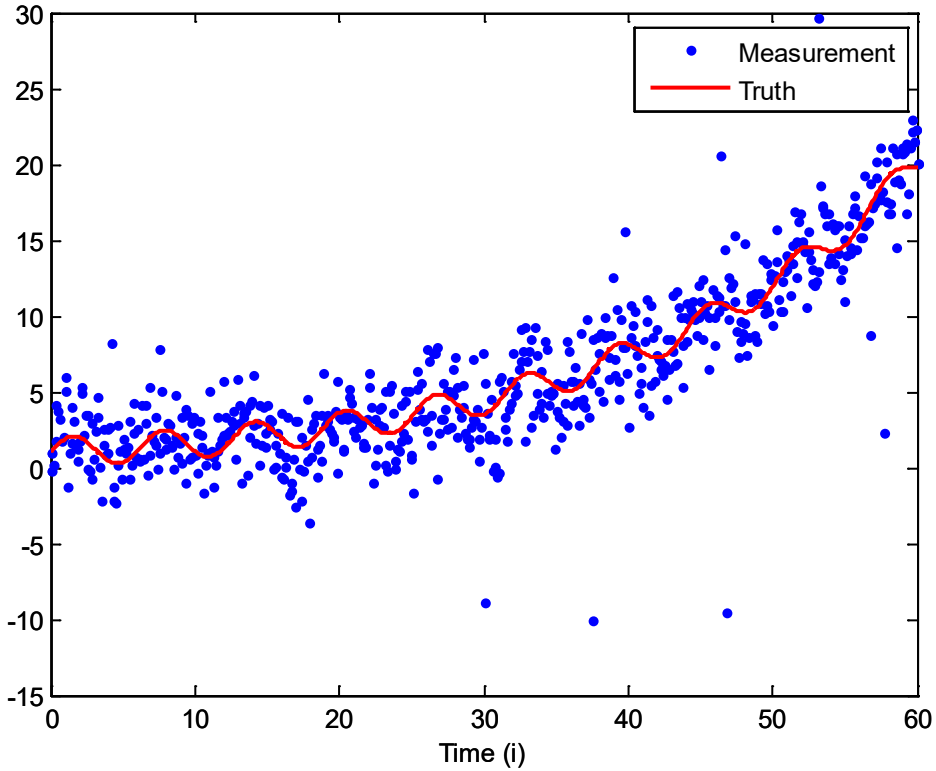
- Noise: random error or variance in a measured variable
- Incorrect attribute values may be due to
  - faulty data collection instruments
  - data entry problems
  - data transmission problems
  - technology limitation
  - inconsistency in naming convention
- Other data problems which require data cleaning
  - duplicate records
  - incomplete data
  - inconsistent data

How to Handle Noisy Data?

- Regression
  - smooth by fitting the data into regression functions
  - *Linear regression* involves finding "best" line to fit two attributes, so that one attribute can be used to predict the other.
  - *Multiple Linear regression* – more than two attributes involved and data fit to a multidimensional surface
- Clustering
  - detect and remove outliers
  - Outliers – values outside of the set of clusters
- Combined computer and human inspection
  - detect suspicious values and check by human (e.g., deal with possible outliers)

Data Smoothing and Outlier Removal

- Usually, the value-type sensor data, or the extracted features need to be smoothed, including outlier removal



Data Smoothing Methods

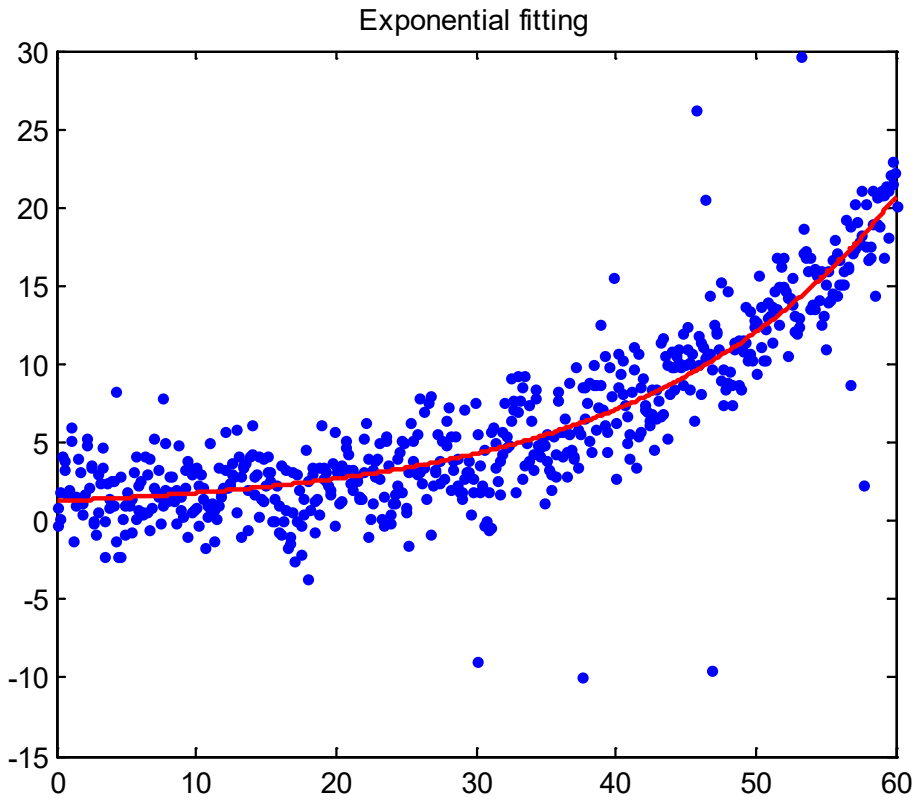
Problem: Time series $\{(t_i, x_i)\}$, i=1,…,n → smoothed time series $\{(t_i, y_i)\}$

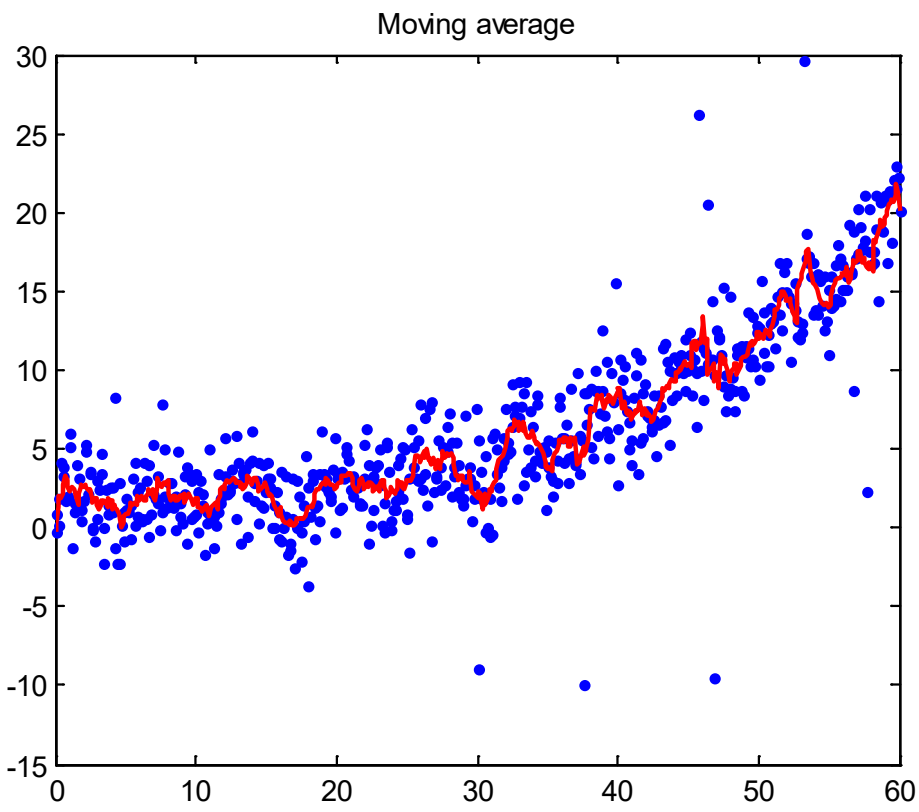
- Curve fitting (linear, non-linear): $y_i = f(t_i)$
  - *e.g. exponential regression*: $y=a*\exp(b*t + c)+d$
- Moving average: $y_i = \text{mean}(\{x_{i-k},\ldots,x_i,\ldots,x_{i+k}\})$
- Median filter: $y_i = \text{mean}(\{x_{i-k},\ldots,x_i,\ldots,x_{i+k}\})$
- Local regression smoothing
  - LOWESS: Local regression using weighted linear least squares and a 1st degree polynomial model
  - LOESS: Local regression using weighted linear least squares and a 2nd degree polynomial model
- Kernel regression

$$y_i = \sum_{j=1}^{n} x_j K(t_i, t_j) \Big/ \sum_{j=1}^{n} K(t_i, t_j) \qquad \sum_{j=1}^{n} K(t_i, t_j) = \exp\left(-\frac{(t_i - t_j)^2}{2\sigma^2}\right)$$

- Linear smoothing: $y_i = a* y_{i-1} + (1-a)*x_i$
  - For example, a = 0.9

Comparison of Smoothing Methods





How to Handle Noisy Data?

- Binning (Discretization)
  - first sort data and partition into (equal-frequency) bins
  - then one can smooth by bin means, smooth by bin median, smooth by bin boundaries, etc.

Discretization

- Three types of attributes
  - Nominal—values from an unordered set, e.g., color, profession
  - Ordinal—values from an ordered set, e.g., military or academic rank
  - Numeric—real numbers, e.g., integer or real numbers
- Discretization: Divide the range of a continuous attribute into intervals
  - Interval labels can then be used to replace actual data values
  - Reduce data size by discretization
  - Supervised vs. unsupervised
  - Split (top-down) vs. merge (bottom-up)
  - Discretization can be performed recursively on an attribute
  - Prepare for further analysis, e.g., classification

Data Discretization Methods

- Typical methods:

  All the methods can be applied recursively
  - Binning
    - Top-down split, unsupervised
  - Clustering analysis (unsupervised, top-down split or bottom-up merge)
  - Decision-tree analysis (supervised, top-down split)
  - Correlation (e.g., χ^2) analysis-based discretization (supervised, bottom-up merge)

Simple Discretization: Binning

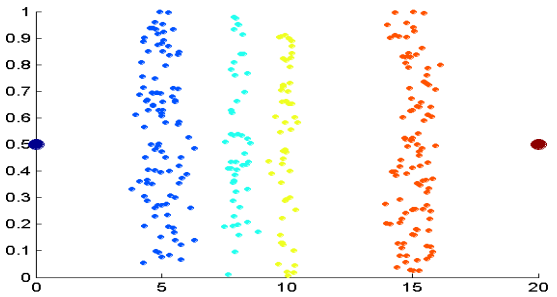
- Equal-width (distance) partitioning
  - Divides the range into *N* intervals of equal size: uniform grid
  - if *A* and *B* are the lowest and highest values of the attribute, the width of intervals will be: $W = (B - A)/N$.
  - The most straightforward, but outliers may dominate presentation
  - Skewed data is not handled well
- Equal-depth (frequency) partitioning
  - Divides the range into *N* intervals, each containing approximately same number of samples
  - Good data scaling
  - Managing categorical attributes can be tricky

Binning Methods for Data Smoothing

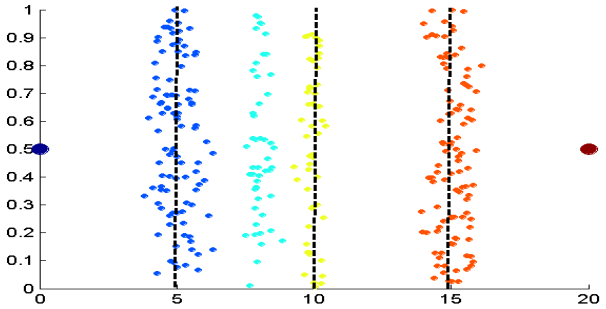
- Sorted data for price (in dollars): 4, 8, 9, 15, 21, 21, 24, 25, 26, 28, 29, 34

* Partition into equal-frequency (**equal-depth**) bins:
  - Bin 1: 4, 8, 9, 15
  - Bin 2: 21, 21, 24, 25
  - Bin 3: 26, 28, 29, 34
* Smoothing by **bin means**:
  - Bin 1: 9, 9, 9, 9
  - Bin 2: 23, 23, 23, 23
  - Bin 3: 29, 29, 29, 29
* Smoothing by **bin boundaries**:
  - Bin 1: 4, 4, 4, 15
  - Bin 2: 21, 21, 25, 25
  - Bin 3: 26, 26, 26, 34

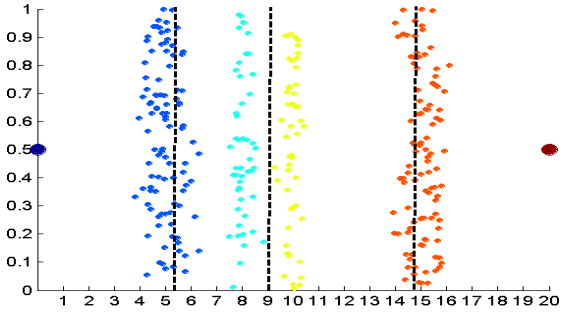
Discretization Without Using Class Labels (Binning vs. Clustering)



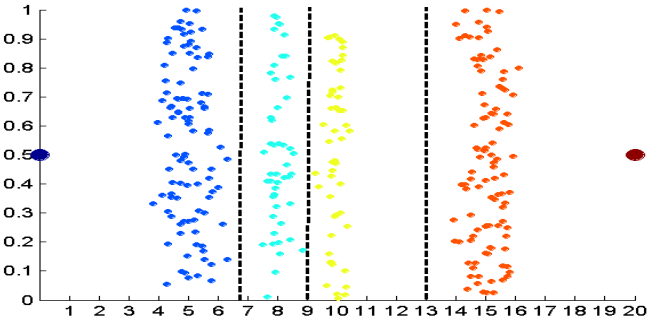
Data



Equal width (binning)



Equal frequency (binning)

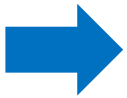


K-means clustering leads to better results

One-hot Encoding

- A popular technique for treating categorical variables.
- One-hot encoding is a sparse way of representing data in a binary string in which only a single bit can be 1, while all others are 0.

	Feature_1	Feature_2	Feature_3
Sample_1	1	4	3
Sample_2	2	3	2
Sample_3	1	2	2
Sample_4	2	1	1

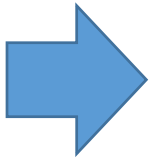


1 -> 0001
2 -> 0010
3 -> 0100
4 -> 1000

One-hot Encoding

Simply creates additional features based on the number of unique values in the categorical feature. Every unique value in the category will be added as a feature.

Country	Age	Salary
0	44	72000
2	34	65000
1	46	98000
2	35	45000
1	23	34000



0	1	2	Age	Salary
1	0	0	44	72000
0	0	1	34	65000
0	1	0	46	98000
0	0	1	35	45000
0	1	0	23	34000

3 new features are added as the country contains 3 unique values – India, Japan, and the US.

Correlation Analysis (Numeric Data)

- Correlation coefficient (also called Pearson's product moment coefficient)

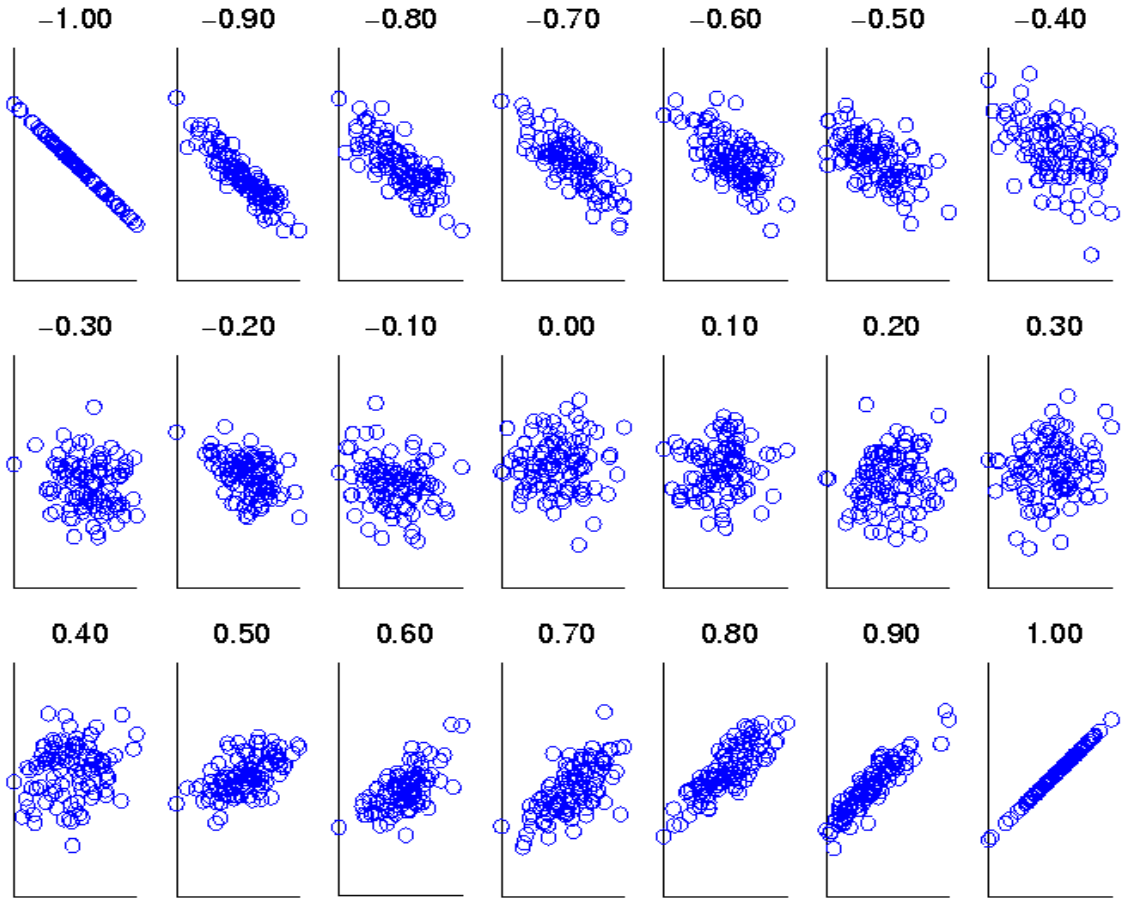
$$r_{A,B} = \frac{\sum_{i=1}^{n}(a_i - \overline{A})(b_i - \overline{B})}{(n-1)\sigma_A\sigma_B} = \frac{\sum_{i=1}^{n}(a_i b_i) - n\overline{A}\,\overline{B}}{(n-1)\sigma_A\sigma_B}$$

  where n is the number of tuples, $\overline{A}$ and $\overline{B}$ are the respective means of A and B, σ_A and σ_B are the respective standard deviation of A and B, and $\Sigma(a_i b_i)$ is the sum of the AB cross-product.

- $-1 \leq r_{A,B} \leq 1$
  - If $r_{A,B} > 0$, A and B are positively correlated (A's values increase as B's). The higher, the stronger correlation.
  - If $r_{A,B} = 0$: not correlated
  - If $r_{AB} < 0$: negatively correlated

Visually Evaluating Correlation

Scatter plots showing the similarity from -1 to 1.



Covariance

- Covariance: $Cov(A,B) = E((A-\bar{A})(B-\bar{B})) = \frac{\sum_{i=1}^{n}(a_i-\bar{A})(b_i-\bar{B})}{n}$
- **Correlation coefficient:** $r_{A,B} = \frac{Cov(A,B)}{\sigma_A \sigma_B}$

  where n is the number of tuples, $\overline{A}$ and $\overline{B}$ are the respective mean or **expected values** of A and B, σ_A and σ_B are the respective standard deviation of A and B.

- **Positive covariance**: If $Cov_{A,B} > 0$, then A and B both tend to be larger than their expected values.
- **Negative covariance**: If $Cov_{A,B} < 0$ then if A is larger than its expected value, B is likely to be smaller than its expected value.
- **Independence**: $Cov_{A,B} = 0$ but the converse is not true:
  - Some pairs of random variables may have a covariance of 0 but are not independent. Only under some additional assumptions (e.g., the data follow multivariate normal distributions) does a covariance of 0 imply independence

Handling Outlier

Example: Outlier

- Can skew regression analysis
- Could also simply indicate that data really do have large variations
- Comprehensive analysis of a set of data should
  - Look for outliers
  - Examine their possible causes
  - Examine their effect on analysis
  - Discuss whether they should be excluded from calculations
- Outliers have less effect on larger samples

Outlier example: Driving Test Scores

- A school is making assumption that the correlation between **instructional hours** and **test scores** is indication of its service quality
  - Difficult to prove definitively
  - Reasonable if consistently strong correlations have been found between time the students spent with classes and students' test scores

Outlier example: Driving Test Scores

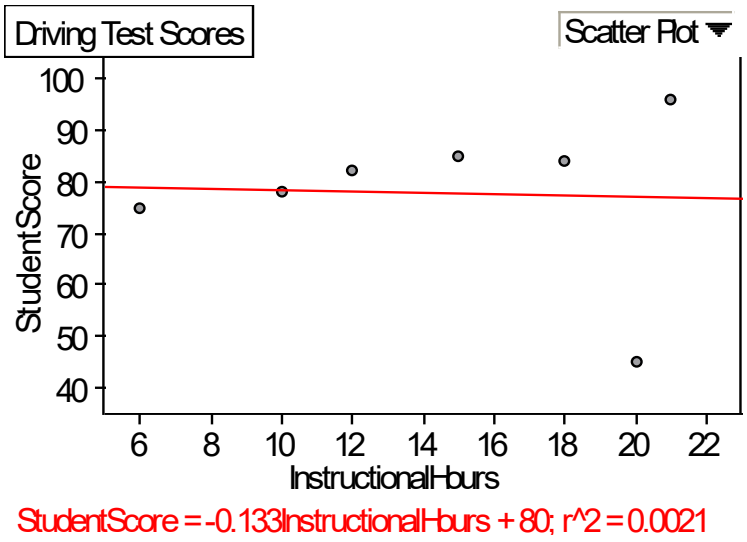
- Number of hours is independent variable
- Scatterplot indicates strong positive linear correlation

- Linear regression analysis yields line of best fit:

  $y = -0.13x + 80$ and $r = -0.05$!

  - virtually zero linear correlation
  - line of best fit has negative slope:
  - less time instructing

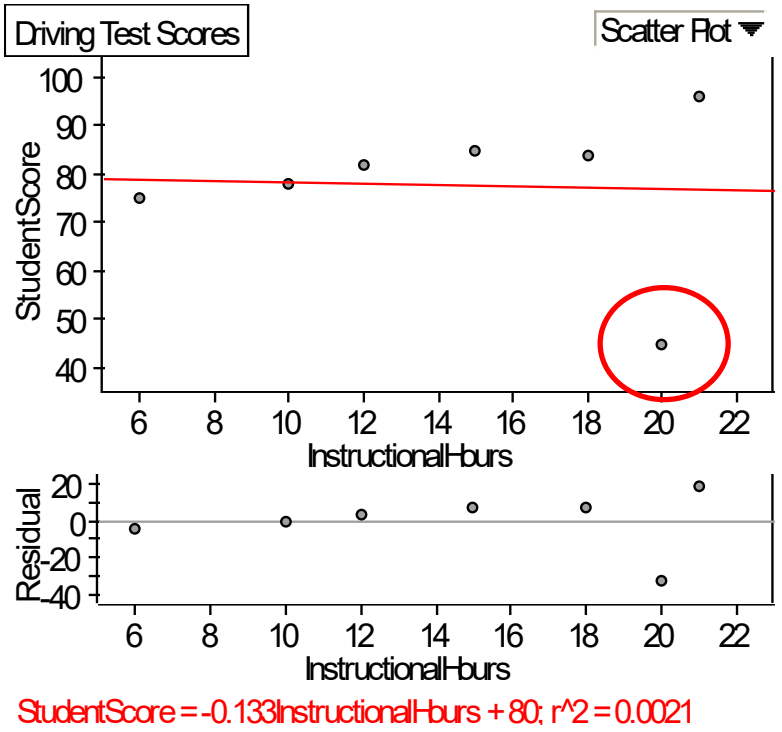
    -> better test scores?



- Scatterplot looked favorable, but regression analysis suggests that lessons had no positive effect on students' test results

Outlier example: Driving Test Scores

- Outlier at (20, 45)
- Residual plot: substantially below the others
- Suggests special circumstance may have caused abnormal result on test
  - Accidental event, emotional upset, really slow learner...
  - Bad regression model?

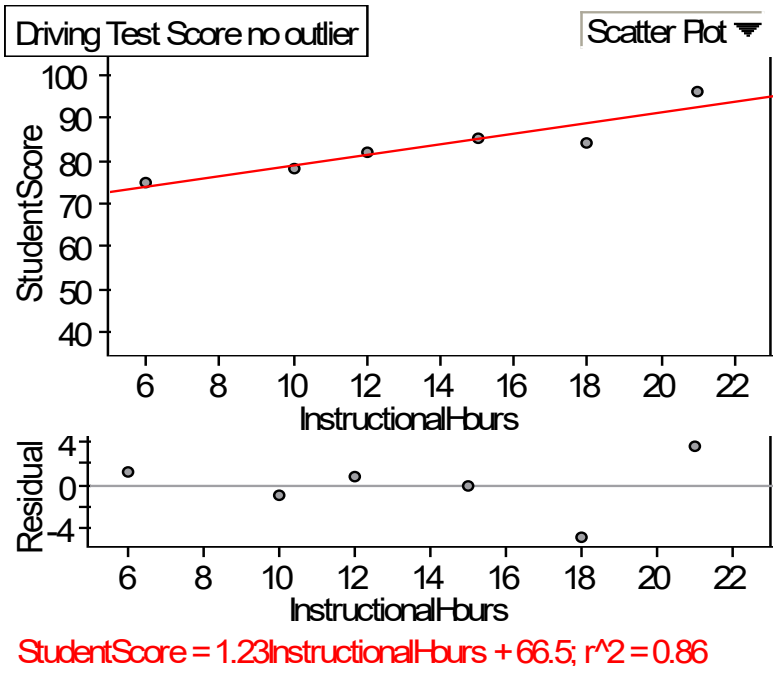


Reasonable to exclude this data point when evaluating the service quality

Outlier example: Driving Test Scores

Removing outlier and repeating analysis

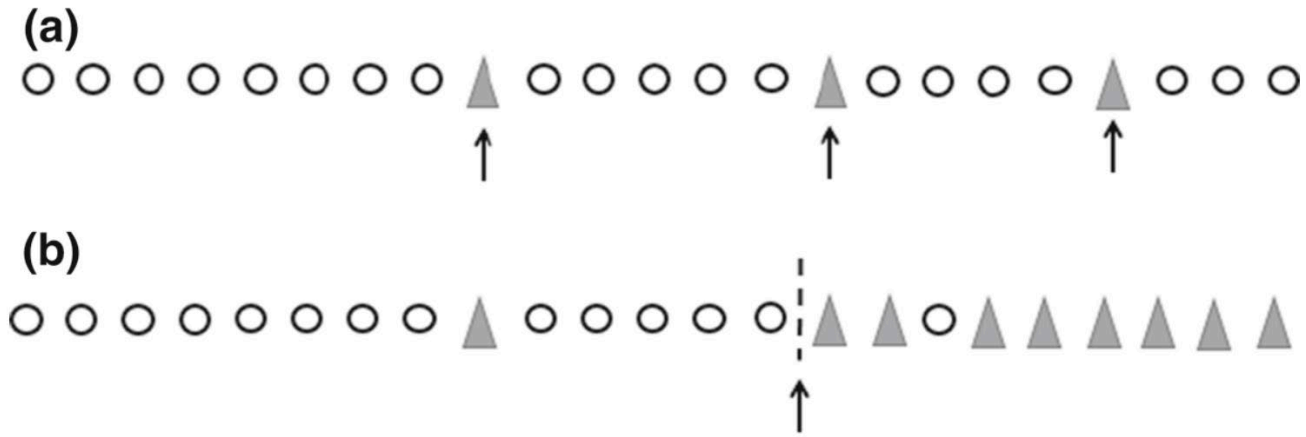
- $y = 1.2299x + 66.525$,
  $r = 0.93$
- Strong positive linear correlation between instructional hours and test result



- The school may be of great training quality after all!

Example: Outlier and anomaly pattern detection

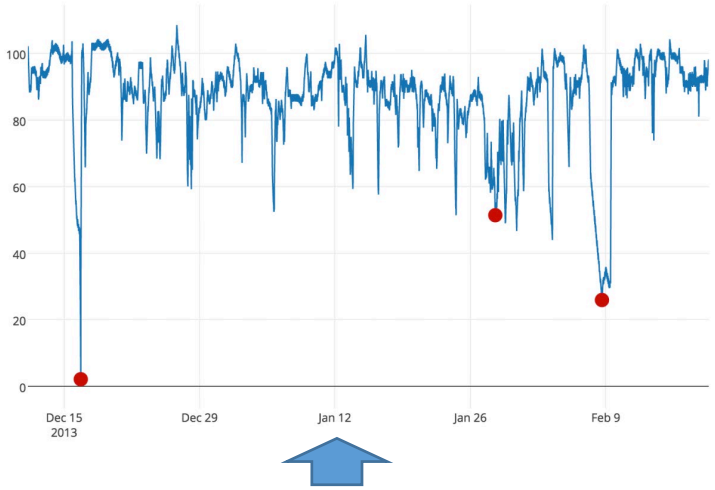
- To find data instances which significantly deviate from the underlying data distribution
- Outliers can be noise or interesting instances
- Anomaly detection / Novelty detection

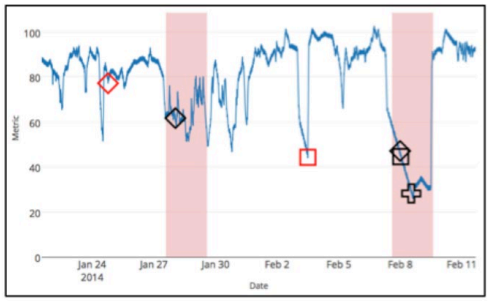


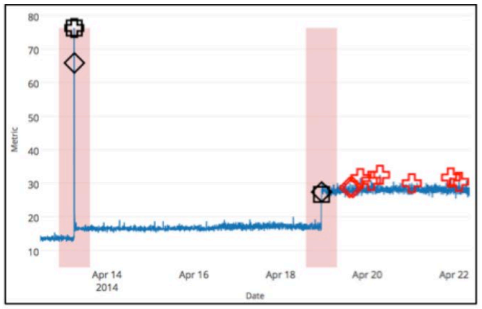
a. Outlier detection. b. Anomaly pattern detection

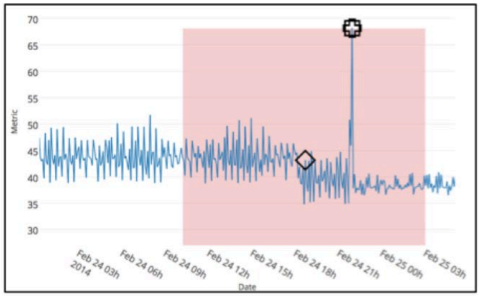
Example: Outlier and anomaly pattern detection

Real-world temperature sensor data from an internal component of a large industrial machine









- 1st anomaly: a planned shutdown.
- 3rd anomaly: a catastrophic system failure.
- 2nd anomaly: a subtle but observable change, indicated the actual onset of the problem that led to the eventual system failure.

Handling Outlier

- **Outlier detection** aims to find data instances which significantly deviate from the underlying data distribution.
- Outliers can be noise or interesting instances
- **Outlier detection** is performed at an individual instance level, and **anomalous pattern detection** involves detecting a point in time where the behavior of the data becomes unusual and differs from normal behavior.
- Advanced topic: **concept drift detection** looks for a concept-changing point in the streaming data and try to adapt the model to the new emerging pattern.

Outlier and anomaly pattern detection

Outlier detection can be applied in situations:

1. Under the assumption that the **majority** of the instances in a data set are normal, the instances that seem to fit least to the remainder of the data set are detected. (referred as unsupervised outlier detection)
2. A set of data samples that have been labeled as “normal” and “abnormal” is given, and a classifier trained from the data is used for detecting abnormal instances of test data. (referred as supervised outlier detection)
3. Given a normal data set, a model representing **normal behavior** is built and the likelihood of a test instance to be generated from the model is computed. (partially supervised outlier detection)

Outlier detection methods

Distance-based methods

- Knorr & Ng

Density-based methods

- KDIST: K^{th} nearest distance
- MeanDIST: Mean distance

Graph-based methods

- MkNN: Mutual K-nearest neighbor
- ODIN: Indegree of nodes in k-NN graph

Tree-based methods

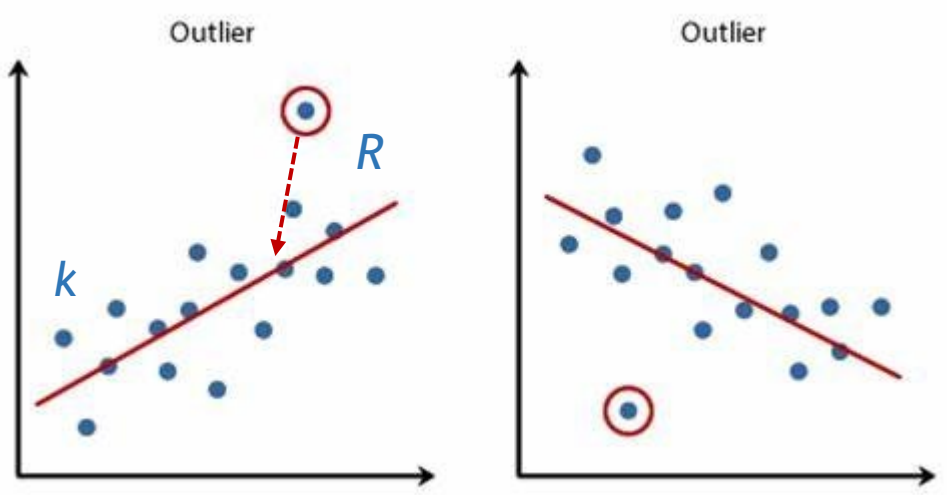
Clustering-based methods

Deep learning-based methods

…

Distance-based methods

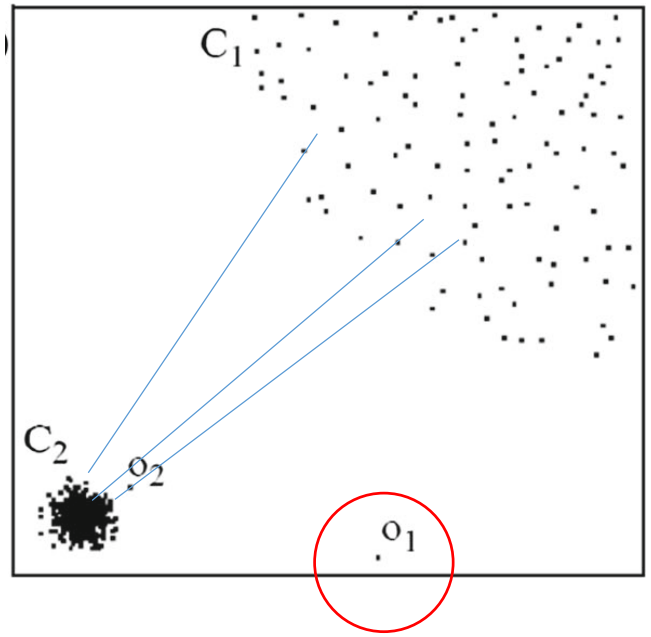
- Given parameters k and R, a data instance is an outlier if less than k data instances in the input data set lie within distance R from it.



Density-based methods

LOF-based outlier detection

- When the clusters have different densities, LOF (local outlier factor) gives an **outlier score** based on local density around a data instance.
- LOF provides an indication of whether p is in a denser or sparser region of the neighborhood than its neighbors.



$$LOF(p) = \frac{\frac{1}{k}\sum_{q \in \mathrm{kNN}(p)} lrd(q)}{lrd(p)}$$

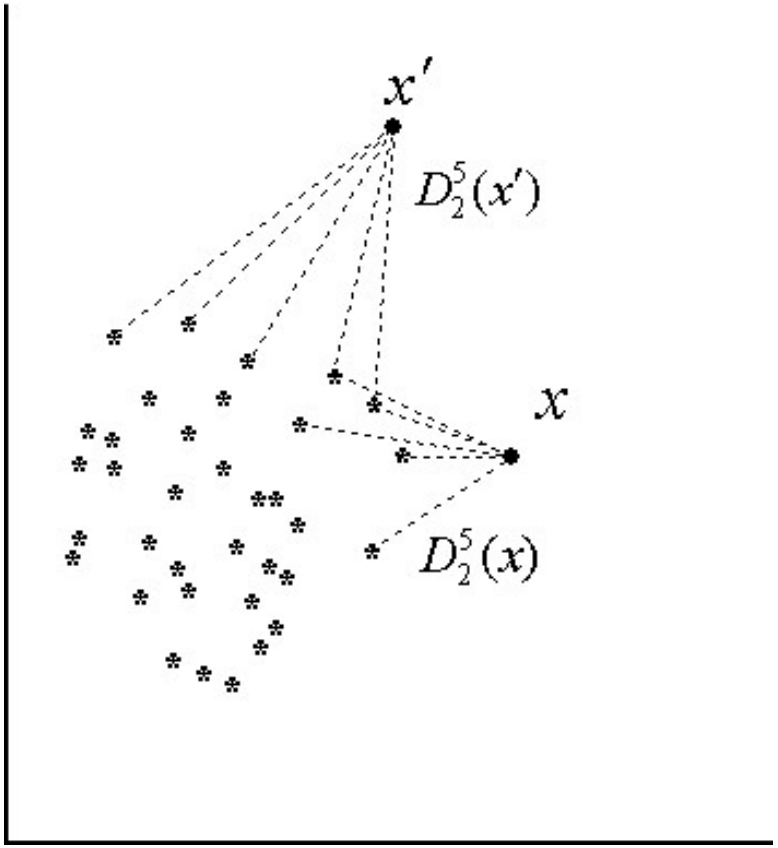
- kNN(p) denotes a set of k-nearest neighbors of a data instance p.
- Local reachability density (lrd) of p is computed from the inverse of the average reachability distance to the k-nearest neighbors of p.

Density-based methods

KDIST

- Define *k nearest neighbour distance* (**KDIST**) as the distance to the k^{th} nearest vector.

- Vectors are sorted by their *KDIST* distance. The last *n* vectors in the list are classified as outliers.



[Ramaswamy et al. , 2000: ACM SIGMOD]

Density-based methods

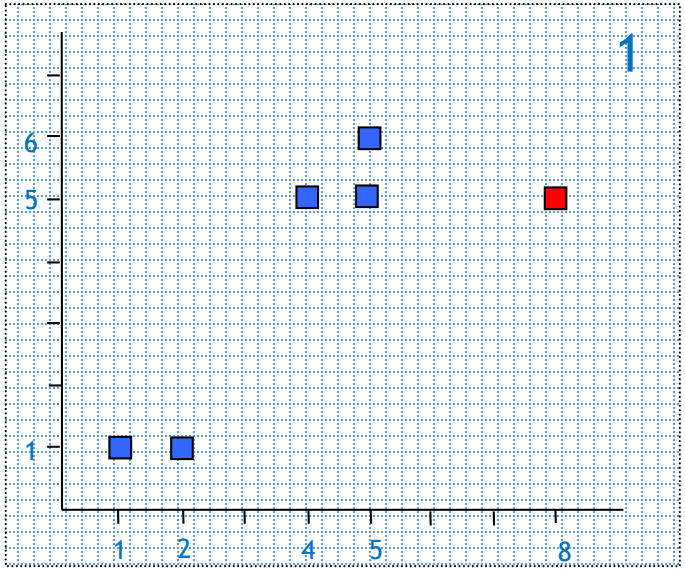
Mutual *k*-nearest neighbor

- Generate directed k-NN graph.
- Create undirected graph as follows:
  1. Vectors *a* and *b* are *"mutual neighbors"* if both links $a \rightarrow b$ and $b \rightarrow a$ exist.
  2. Change all mutual links $a \leftrightarrow b$ to undirected link $a—b$.
  3. Remove the rest.
- Connected components are clusters.
- Isolated vectors as outliers.

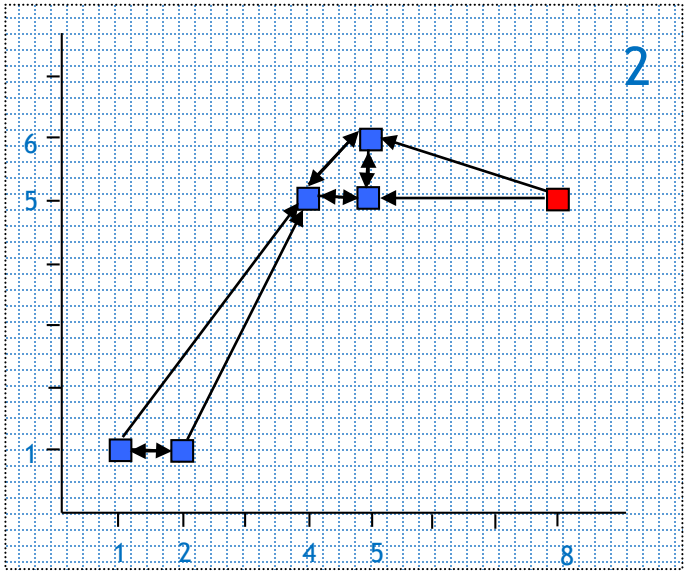
[Brito et al., 1997: Statistics & Probability Letters]

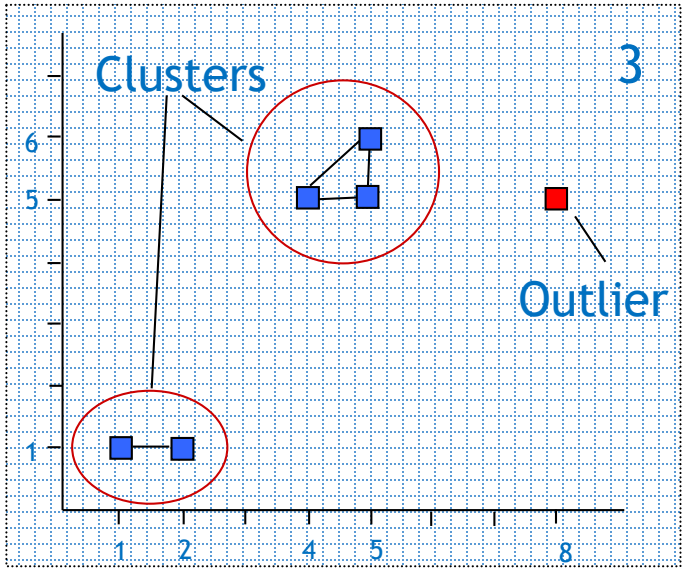
Mutual *k*-nearest neighbor

$k = 2$



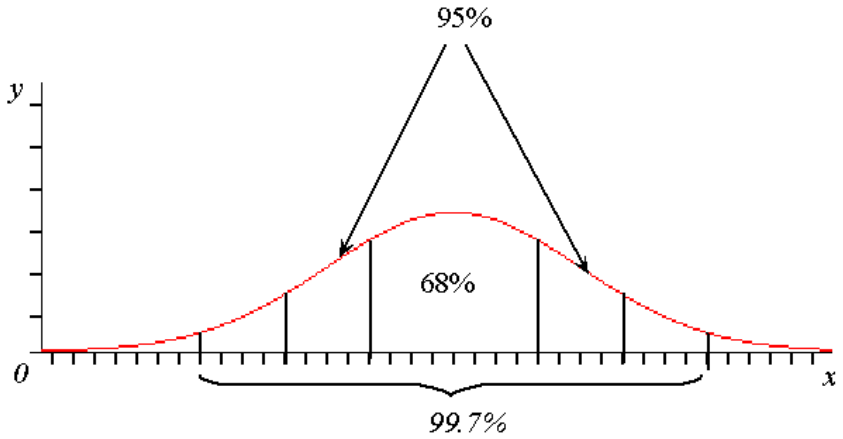
1. Given a data with one outlier.
2. For each vector find two nearest neighbours and create *directed 2-NN* graph.
3. For each pair of vectors, create edge in *mutual graph*, if there are edges a→b and b→a.





Distribution-based methods

- Define outliers by examining the projections of the data, which have abnormally low density.
- Assumption: There are a total of *N* points in the database. If the data were uniformly distributed, then the presence of any point in a k-dimensional cube is a bernoulli random variable with probability f^k.
- The number of points in a cube can be approximated by a normal distribution. The expected standard deviation of the points in a *k*-dimensional cube is given by *N* multiplies f^k and the square root of $N \times f^k \times (1 - f^k)$.
- Let n(D) be the number of points in a k-dimensional cube D. Then, the **sparsity coefficient** S(D) of the cube D can calculate.



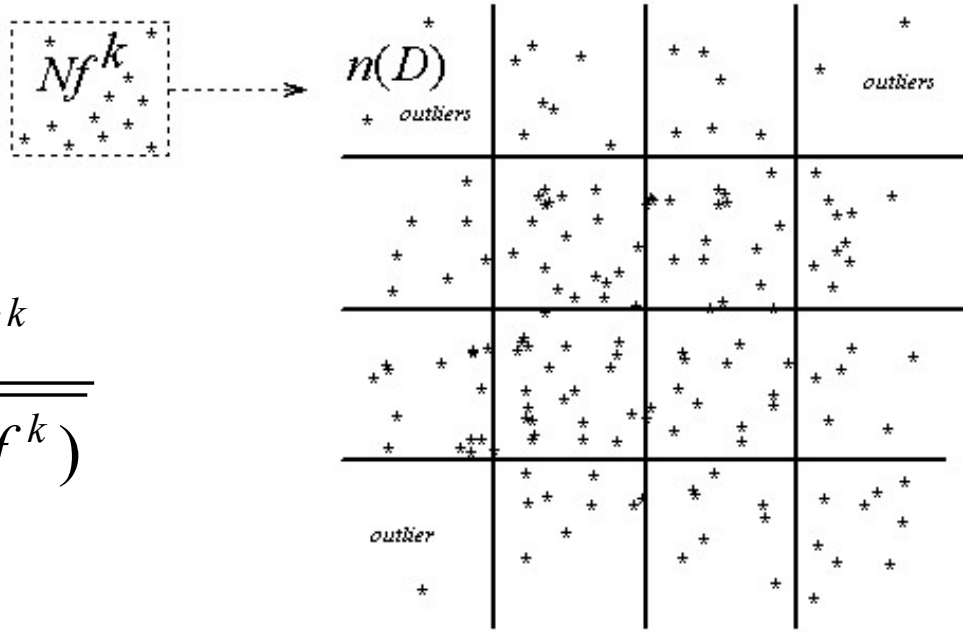
$$S(D) = \frac{n(D) - N \cdot f^k}{\sqrt{N \cdot f^k \cdot (1 - f^k)}}$$

[Aggarwal and Yu , 2001: ACM SIGMOD]

Distribution-based methods

Detection of sparse cells

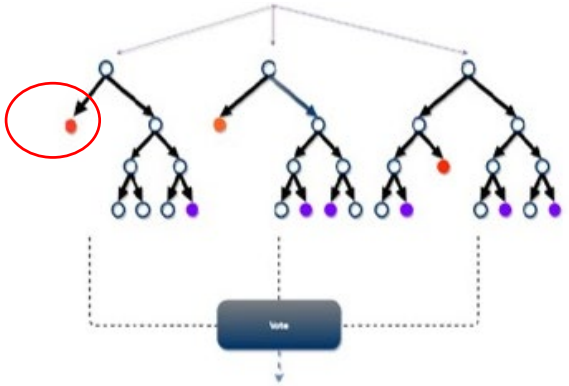
$$S(D) = \frac{n(D) - N \cdot f^k}{\sqrt{N \cdot f^k \cdot (1 - f^k)}}$$



- The sparsity coefficients which are negative indicate cubes in which the presence of the points is significantly lower than expected.
- In general, the uniformly distributed assumption is not always true. However, the sparsity coefficient provides an intuitive idea of the level of significance.

Tree-based methods

- Isolation forest
  - Outliers are susceptible to isolation than normal data
  - Isolation tree is built by repeating recursively the random selection of an attribute and a splitting value until all instances are isolated.
  - The length of a path where a point traverses from the root node to an external node is used to compute an outlier score.
  - On average, a normal point requires more partitions to be identified than an abnormal point.
  - Outliers are less frequent than regular observations and require less splits (closer to the root of the tree)



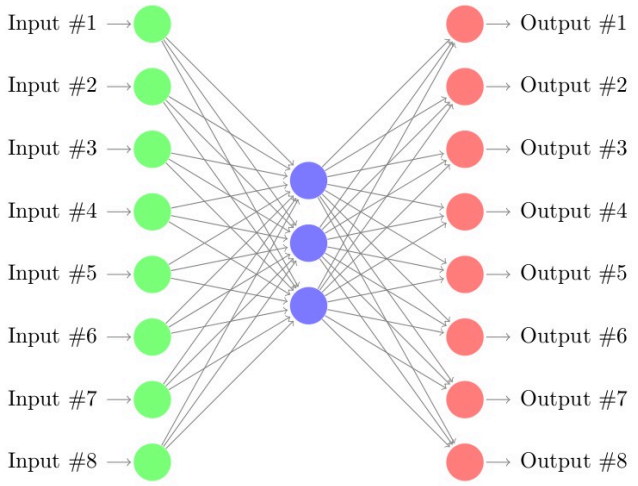
Isolation forest

Deep learning-based methods

- **Autoencoder** is a type of ANN used to learn efficient data encodings in an unsupervised manner.
  - Relies on input signal rebuilding after putting it through a compressive path.
  - After training, the autoencoder can optimally **reconstruct** data similar to the one it was trained with.
  - Anomalies will present a high **reconstruction error rate** after being forwarded through this architecture.

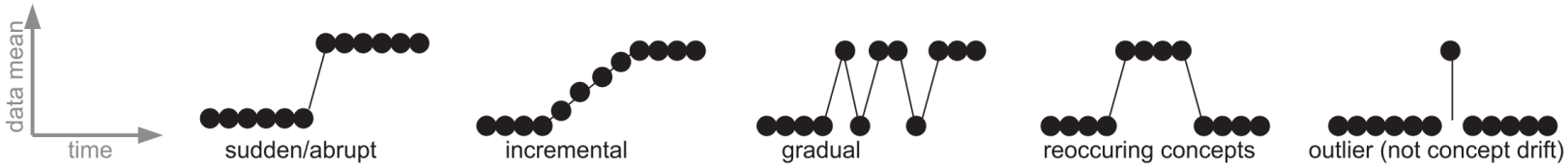
- Autoencoders,
- Replicator neural networks,
- Deep structured energy based models
- ...

Autoencoder

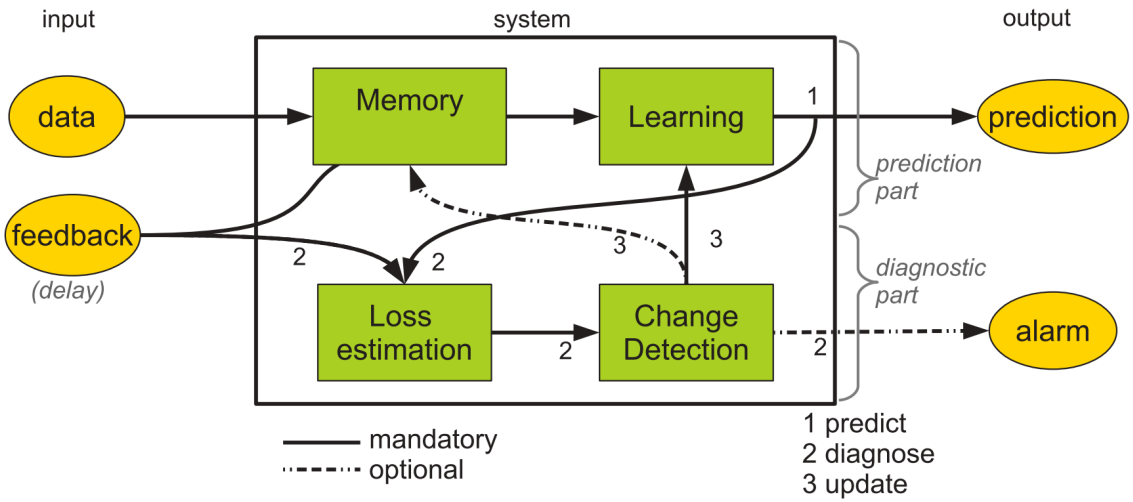


the encoder and the decoder

Concept drift

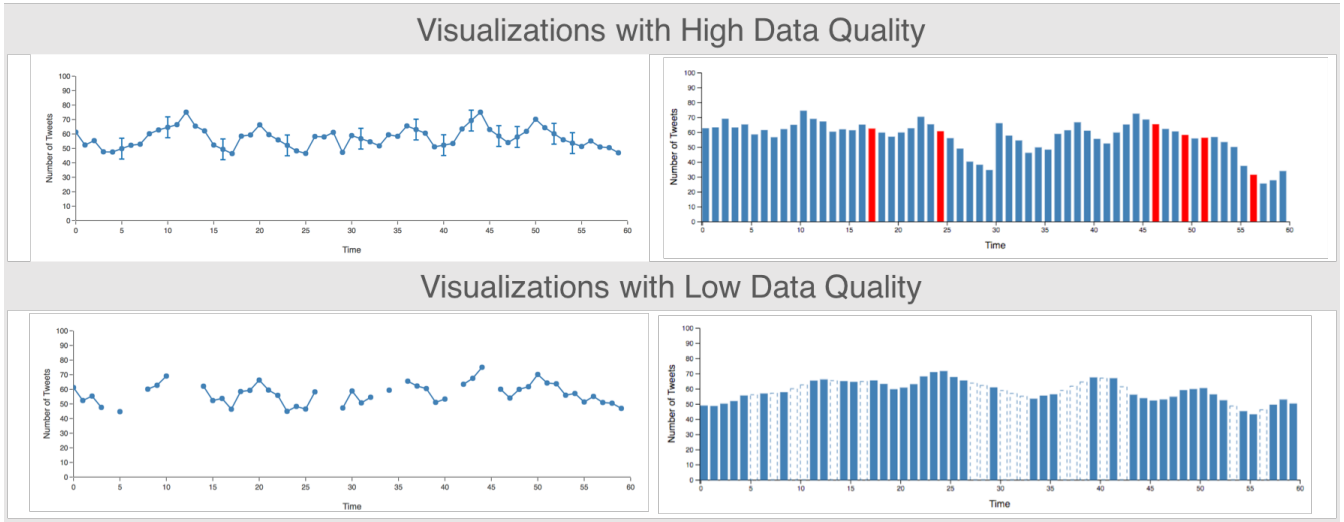


Patterns of changes over time



A generic schema for an online **adaptive learning algorithm**

Missing Data



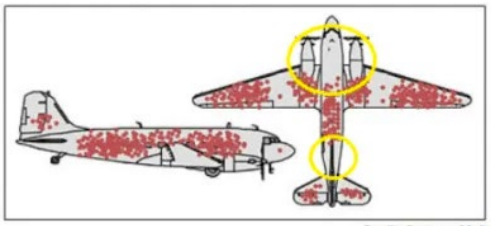
Data is not always available

Sources of Missing Data

- Monitoring system failed to store the data
- Data/texts are indistinct or ambiguous
- Numeric data are obviously wrong
- Broken objects cannot be measured
- Equipment failure or malfunction
- Detailed analysis of subsample
- Inconsistent with other recorded data and thus deleted
- Data not entered due to misunderstanding
- Certain data may not be considered important at the time of entry
- Not register history or changes of the data
- People refuse to give the data/info

Survivorship Bias

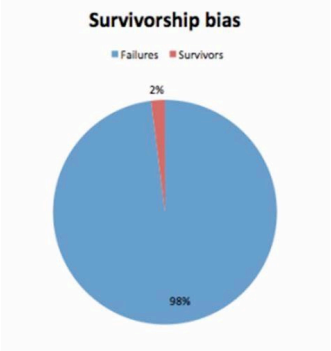
- Survivorship bias is a type of sample selection bias
- Sample selection bias is the bias that results from the failure to ensure the proper randomization of a population sample. The flaws of the sample selection that occurs when a data set only considers "surviving" or existing observations and fails to consider observations that already ceased to exist.



Credit: Cameron Moll

Gentlemen, you need to put more armour-plate where the holes aren't because that's where the holes were on the airplanes that didn't return - Abraham Wald 1942.

Planes that were hit that did not make it back!



Abraham Wald's research in World War II

A statistician studying World War II airplanes. His research group at Columbia University was asked to figure out how to better protect airplanes from damage.

The initial approach to the problem was to look at the planes coming back, seeing where they were hit the worst, then reinforcing that area.

Realigning: Interpolation

need to fill in the gaps after transformation, using interpolation

Realigning: Interpolation

- Interpolation involves constructing new data points based on known data
- Simple interpolation:
  - Nearest neighbour: Take value of closest voxel
  - Tri-linear: Take weighted average of neighbouring voxels
- B-Spline interpolation
  - Improves accuracy – SPM uses this as standard
- There may still be residual errors

How to Handle Missing Data?

- Fill in the missing value manually: tedious + infeasible?
- Fill in it automatically with
  - A global constant : e.g., “unknown”, a new class?!
  - The attribute mean?
  - The attribute mean for all samples belonging to the same class: smarter?
  - The most probable value: inference-based such as Bayesian formula or decision tree?

Data imputation for missing data

1. Collection of several complete datasets to perform the experiments.
   - Depending on the nature of the domain, the datasets may encompass several feature types and different dimensionality;
2. Mechanisms of missing data.
   - Missing values can occur in only one feature or several features, at several percentages (missing rates).
   - 3 different underlying mechanisms: Missing Completely At Random (**MCAR**), Missing At Random (**MAR**) and Missing Not At Random (**MNAR**) ;
3. Data imputation using several strategies: common choices rely on **statistical-based methods** (e.g. mean imputation) or **machine learning-based methods** (e.g. KNN imputation);
4. Evaluation of imputation algorithms, either in terms of classification performance or quality of imputation, by comparing the substitute values with the ground truth (known original values).

Missing Mechanism

Age	Number of cigarettes			
	Complete	MCAR	MAR	MNAR
15	2	2	⊗	2
15	9	⊗	⊗	⊗
15	4	⊗	⊗	4
16	2	2	⊗	2
16	2	2	⊗	2
16	7	4	⊗	⊗
16	3	3	⊗	3
17	9	⊗	9	⊗
17	6	6	6	⊗
17	4	⊗	4	4
17	5	5	5	5
17	5	5	5	5
18	7	⊗	7	⊗
18	6	6	6	⊗
18	7	⊗	7	⊗
19	3	3	3	3
19	8	⊗	8	⊗
19	3	⊗	3	3
20	9	9	9	⊗
20	2	2	2	2

- Missing mechanisms example: a simulated dataset of a study in adolescent tobacco use, where the daily average of smoked cigarettes is missing under different mechanisms (MCAR, MAR, and MNAR).

Missing Value Replacement

1. Listwise Deletion
2. Pairwise Deletion
3. Dummy Variable Adjustment
4. Imputation

Listwise Deletion

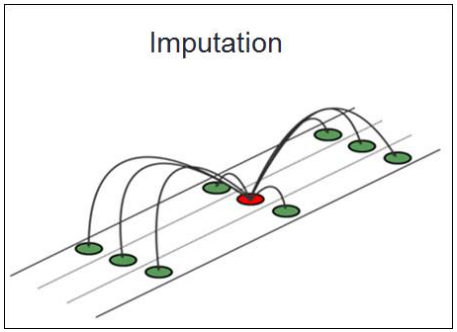
- Delete any samples with missing data
  - Often includes removal of variables (columns) as well as cases (rows)
  - Finding an optimal complete data set involves removing variables with many missing variables, and then rows still having missing variables
  - If MCAR: an unbiased estimate of the full data set
  - If MAR: may produce biased estimates if missing values in independent variables are dependent on dependent variable
  - Main issue
    - **The loss of observations and the increase in standard errors**

Pairwise Deletion

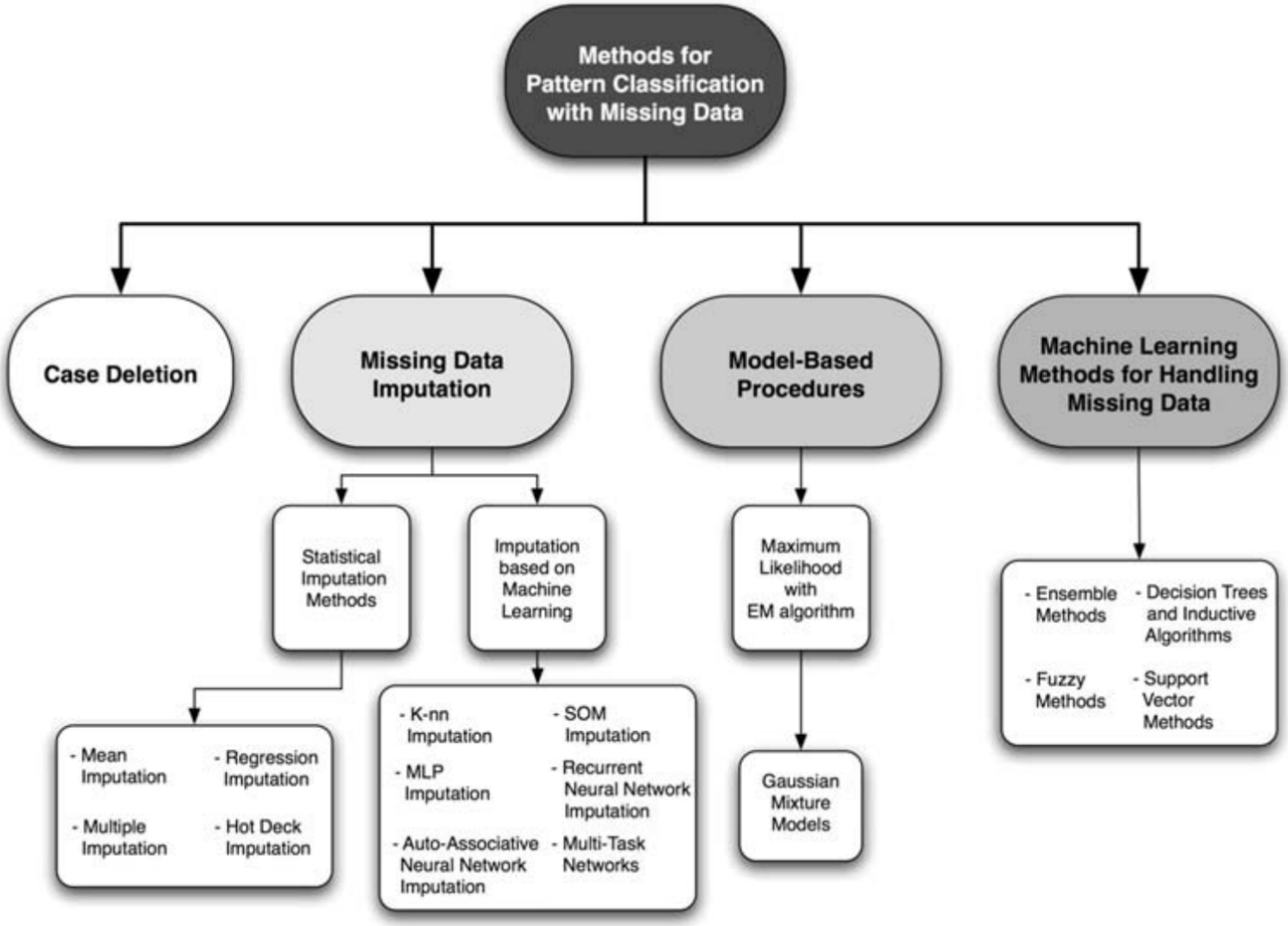
- Compute means using available data and covariances using cases with observations for the pair being computed
- Uses more of the data
- If MCAR: reasonably unbiased estimates
- If MAR: estimates may be seriously biased
- Covariance/Correlation matrix may be singular

Imputation

- Replace missing values with an estimate:
  1. **Mean** imputation – biased estimates of variances and covariances
  2. **Multiple regression** imputation to predict value – complicated with multiple variables containing missing values, but can still lead to underestimated standard errors
  3. Maximum Likelihood
  4. Default value imputation
  5. Inverse probability weighting
  6. KNN



Other topics about missing data



Thanks