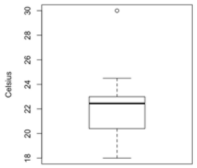
Anomaly Detection

Anomaly
Datapoint or collection of data points which
- not follow usual pattern
- have different structure as rest of the data

=> anomalies get „flagged“ by algorithm



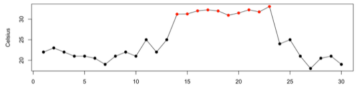
Anomaly as Outlier
- anomaly usually some kind of outlier
  => statistical methods for outlier detection can be used
- eg.
  - Univariate: Boxplot, Number of standard deviations from the mean
  - Multivariate: usually needs adaption
    - parametric: multivariate normal distribution
    - distance score has to be defined

```
summary(temperature)
```

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
18.00	20.45	22.45	22.30	22.98	30.00

Point Anomaly
- single datapoint that is unusual / anomalous compared to the rest of the data
- 2 Types:
  - one-dimensional anomaly: often extreme value in one dimension of the data point
  - multi-dimensional anomaly: unusual combination values across multiple dimensions

Collective anomaly
- collection of similar data points
  that is anomalous when considered together
- can but need not be point anomalies
- especially important in data over time
- Example:
  - 10 consecutive high daily temperatures
  - 1 hotter day maybe normal
    => but multiple consecutive days can cause event to be considered an anomaly



Measures
F1-Score: Harmonic mean between Precision and Recall
Recall: Anomalies correctly identified / Total anomalies
=> Perfect recall = every anomaly detected by algorithm
Precision: Anomalies correctly identified / Total scored as anomalous
=> Perfect precision = no normal instances incorrectly labeled

Problems of anomaly detection
- defining a normal region that covers all normal behavior patterns
- normal behavior evolves over time
- nature / pattern of anomaly can often not be anticipated
- anomalies due to malicious behavior will adapt over time
- lack of availability of data of anomalies for training
- if noise is similar to anomaly and leads to false detections

Summary: Point Anomaly techniques

Basic question: How likely is this value?
- how likely in general: use density, which describes the likelihood
- how likely based on previous values: and point in time - eg. considering season, current level

Can we anticipate the nature of the anomaly?
- Yes: treat as supervised
- No: treat as anomaly detection - by nature unsupervised

Univariate techniques
- density estimation using normal
  => transformation if necessary
- Graphical techniques:
- Statistical tests:
  - Grubbs: tests for one outlier
  - Tietjen-Moore: multiple outliers => exact number must bed specified
  - generalized ESD: multiple outliers => upper bound must be specified

Multivariate techniques
- Statistical / parametric approaches
  - by adapting univariate
    - eg. density estimation - multiply densities
    - extension: incorporate correlation for unusual feature combinations resulting in anomalies
      1. manually creating features
      2. using multivariate normal

- ML-Approaches
  - use distance score
    => calculate distance score in different ways
  - Distance-based: KNN => only global outliers
  - Density-based: LOF => favors local outliers
  - Isolation Trees: based on how fast can point can separated using random splits
  - Isolation Forests: improves Isolation trees by bagging

Supervised vs Anomaly / Unsupervised problem
- Anomaly detection if:
  Easier to understand / learn what is „normal“, and determine if point „deviates from normal“
  => eg. detecting defects and frauds
- nature / pattern of anomalies be learned from sample => supervised learning
- nature / pattern be anticipated => supervised earning
- no anomalies yet => anomaly detection
- too few anomalies seen too small to learn pattern => anomaly detection

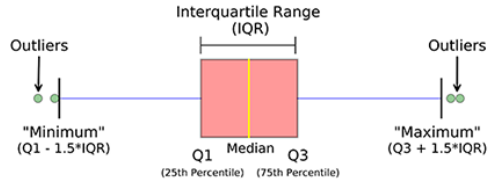
=> eg. no anomalies & knowledge about them until now. => anomaly detection

Univariate Methods

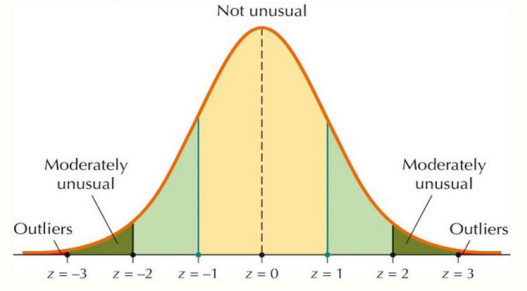
Graphical Methods
- Boxplots
- Normal Probability Plot
- Run Sequence Plot

Statistical
- Statistical Summary
- z-score:
  If normally distributed: > 3 standard deviation from mean
  => may need transform
- Grubbs's Test
- Tietjen-Moore Test
- Generalized ESD

Boxplots



z-score
Check if data point > 3 standard deviations from the mean



Statistical Tests

Grubbs' Test
- test if datapoint that is farthest from the mean (can be positive or negative) is outlier
- assumes normality of the data
- Alternative: most is extreme value is outlier
- Problem: Multiple Outliers:
  - Problem: tests only for one anomaly at a time
  - Solution:
    1. remove Outlier and repeat test
    2. Tietjen-Moore or Generalized ESD

Tietjen-Moore test
- used to detect multiple outliers in univariate data set, which is approximately normal
  => BUT requires exact specification of number of suspected outliers
- is generalization of Grubb's Test for case of multiple outliers
- Tietjen-Moore equivalent to Grubbs' test: if testing for a single outlier
- H0: no outliers; HA = exactly k outliers
  => k not known - use generalized extreme studentized deviate test - requires only upper bound

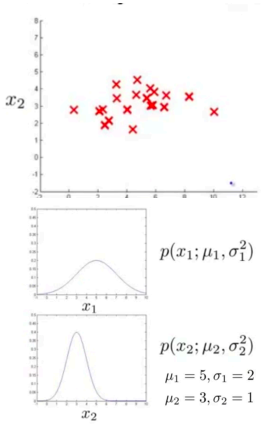
Limitation of Grubbs's & Tietjen-Moore Test
- suspected number of outliers, k, must be specified exactly
- If k is not specified correctly, this can distort the conclusions of these tests.

Generalized ESD:
- generalized extreme studentized deviate test
- also used to detect multiple outliers in univariate data set, which is approximately normal
  => BUT only requires upper bound for the suspected number of outliers be specified
- H0: no outliers; HA = up to k outliers

Multivariate methods
- univariate methods: as Grubb's Test measure distance from mean
- multivariate methods: first needs measure for distance => distance score

Statistical / parametric outlier detection

Anomaly Detection Problem
- Given unlabeled dataset of non-anomalous examples $=\{x^{(1)}, \ldots, x^{(m)}\}$
- Each example is a feature-vector $x^{(1)} \in \mathbb{R}^n = (x_1, \ldots, x_n)$
- Goal:
  Determine $P(x_{test})$ = probability of observing new observation x_{test}
  => $P(x)$ modeled from the data

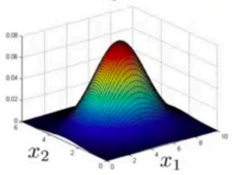


- Then: Detect if outlier:
  If
  $P(x_{test}) < \epsilon$ => flag as anomaly => else:
  $P(x_{test}) \geq \epsilon$ => ok

Examples:
- Fraud Detection: $x^{(i)}$ = feature of user i's activity
  => if $P(x) < \epsilon$, then user shows unusual behavior
- Monitoring servers: $x^{(i)}$ = features of machine i
  x_1 =memory user, x_2 = number of disk accesses, x_3 = cpu-load, x_4 = cpu-load/network traffic
  => => if $P(x) < \epsilon$, then flag with admins

Density Estimation
- for each feature: estimates univariate normal distributions
- determine μ, σ for each feature, so that $x_i = \mathcal{N}(\mu_i, \sigma_i^2)$
- using independence assumption of assumption:

$$P(x) = \prod_{i}^{n} P(x_i; \mu_i, \sigma_i^2)$$



Algorithm
1. Choose features x_i that might be indicative of anomalous examples
2. Fit parameters: for normal distribution of features $\mu_1, \ldots, \mu_n, \sigma_1^2, \ldots \sigma_n^2$
3. Given new example: compute $P(X)$ and then anomaly if $P(x) < \epsilon$

Choosing ϵ
- Using cross-validation
- chooses ϵ that maximizes a given evaluation metric

=> same approach for which features to include

Metric
- data unbalanced => accuracy not suited
- Better: F1-Score

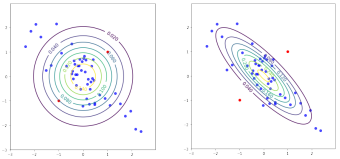
Evaluation of algorithm
- using labeled data
- using cross-validation and hold-out set
- eg. 4.000 good, 20 anomalous => train(cv): 2000 vs 10 and test: 2000 vs 10

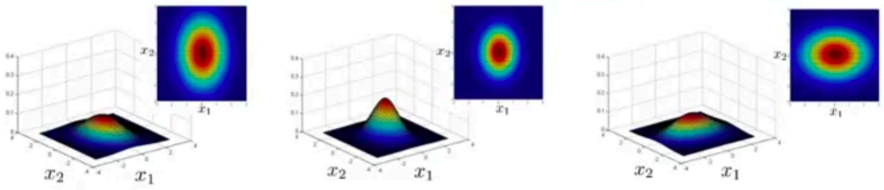
Non-Gaussian Features
- use transformations to make gaussian

Density estimation using multivariate gaussian distribution

Extension: Multivariate gaussian distribution
- using multiple univariate distributions classifies some points as norm
-

Problem of using univariate gaussians for each feature
- it detects some anomalies as normal
  because algorithm looks at density for each feature seperately
- each feature is treated as independent
- results in special case of multivariate gaussian
  - with **0 at off-diagonal elements**
  - **contours are aligned with the axis**
  - with no correlation between the features
  - does not allow for elongated patterns





Anomaly Detection using Multivariate gaussian distribution
- **automatically captures captures correlations** between features
- does not model $P(x_1), P(x_2), \ldots$ separately
- models $P(x)$ all at same time
- Disadvantage
  - computationally more expensive
  - needs more training data

Emulating in simple model (with univariate distributions)
- Manually capture correlations by:
  - manually create features to capture anomalies where x_1, x_2 take unusual combinations
  - to capture anomalies which would else have been seen as normal
- eg. $x_2 = \frac{x_1}{x_2}$
- advantage:
  - computationally cheaper than using multivariate approach
  - works better with small training set size

Error Analysis & finding new features
- Problem: $P(X)$ is similar for normal an anomalous examples
- Solution: look at example that is classified as normal
  => think of feature for which example is anomalous

Multivariate gaussian distribution

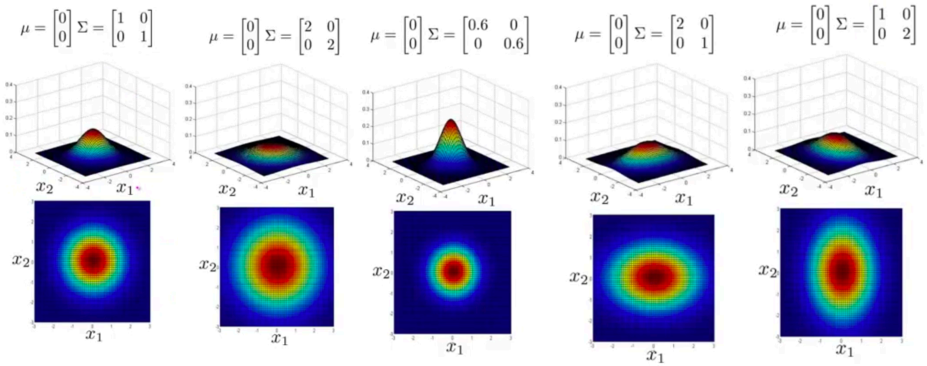
Parameters $\mu \in \mathbb{R}$ and $\Sigma \in \mathbb{R}^{n \times n}$ = covariance matrix

Changing

μ	= mean	= shifts the center
diagonal of	= variance of the univariate distribution	= stretches along the axis
off-diagonal of Σ	= correlation between the features	= stretches at angle / diagonally

Changing the diagonal

= changes width and heights = stretches along the axes - eg along x_1 or x_2

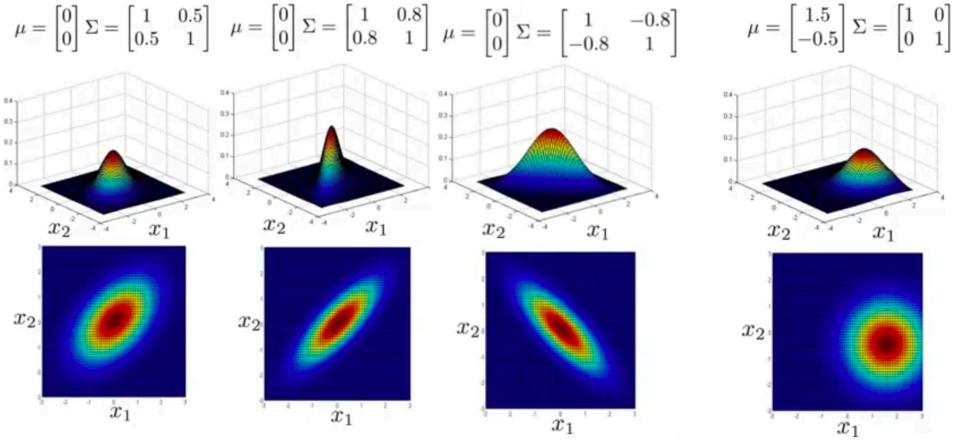


Changing off-diagonal elements

= adding correlation = change together => stretches at an angle = „diagonally“

Changing the mean

= shifting the center of the distribution



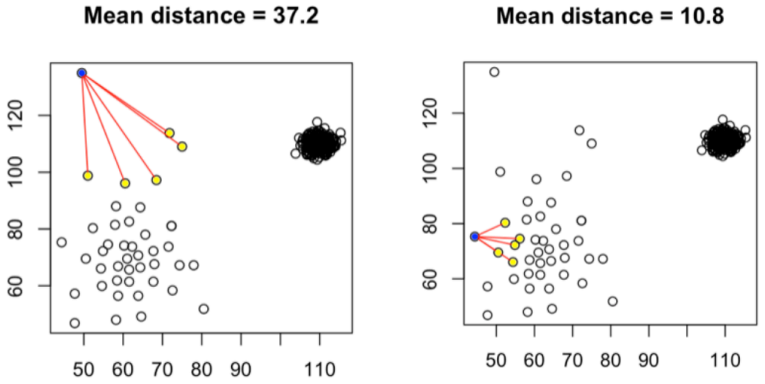
K Nearest Neighbors Outlier Detection

KNN outlier detection

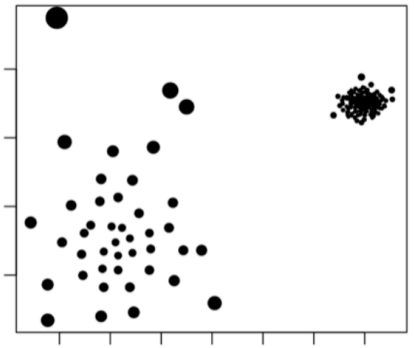
- produces continuous anomaly scores for each data point when data have multiple features
- K-NN distance score:
  1. measures the distance to the n nearest points = neighbors
  2. use mean of those distances
- Intuition: how isolated is point from neighboring points
  => larger values are more likely anomalies

Example

2 points with distances for the 5 nearest neighbors

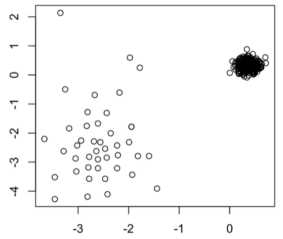


Distance-score visualized by size of points



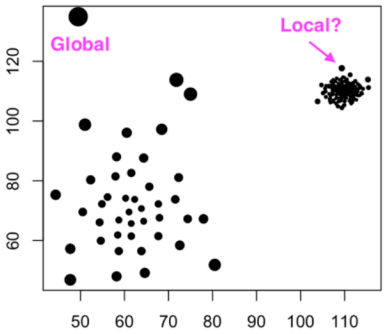
Standardizing

- features needs to be standardized
- otherwise some features will have undue influence
- ensures that features with large mean or large variance do not disproportionately influence the kNN distance score
  => **changes the scale but not the pattern**!!!!!



LOF: Local Outlier Factor

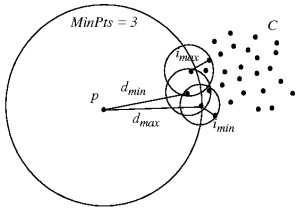
Problem KNN: local anomalies
- Global anomalies
  - KNN good at detecting global anomalies
    => can detect points which are far from their neighbors
- Local anomalies
  - if point is apart from densely clustered neighbors
  - Intuitive: if knn was applied to only this local region it would find outlier

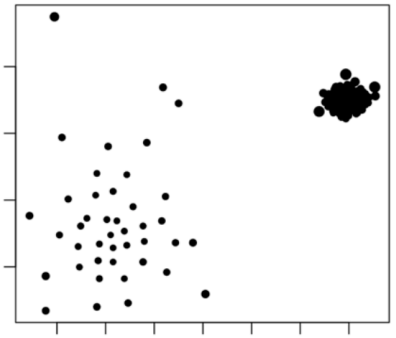


Local Outlier Factor
- Idea: compare density of point to the density of its nearest neighbors
- vs KNN: use relative density instead of distance to construct anomaly scores for each point
- can identify local outliers
  => puts emphasis on local outliers not on global
- more dense: point more dense than its neighbors, if it is in a cluster
- useful if there are clusters in data
- LOF: of point is defined as
  - average of the densities around the k nearest neighbors of the point
  - divided by the density of point itself
    => full formal definition is more involved
- Formal: Density
  Local reachability density of an object A
  is inverse of the average reachability distance of the object A from its neighbors

Interpretation of LOF
- LOF is a ratio of densities
  => centered around 1
  LOF ≤ 1: less likely to be anomalous = outlier
  LOF > 1: more likely to be anomalous = inlier
  => large LOF values indicate more isolated points

Example
- global outlier at top left **no longer the highest outlier**
- several local outliers around cluster have higher score!



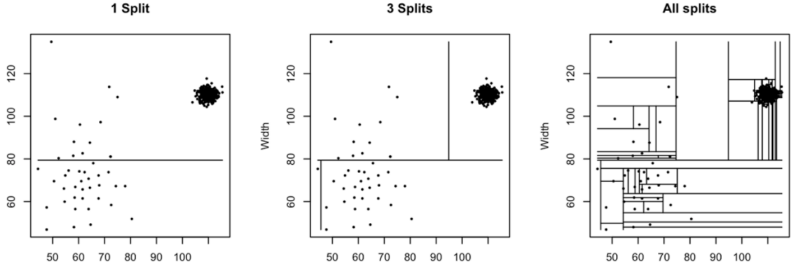


Isolation Trees & Forests

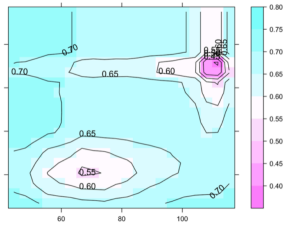
Isolation Trees
- tree-based approach to construct anomaly scores
- fast and robust method of detecting anomalies
- Assumption: points that can be easily separated from the other points are more anomalous
- Idea:
  - measure how easily points can be separated
  - by randomly splitting the data into smaller and smaller regions

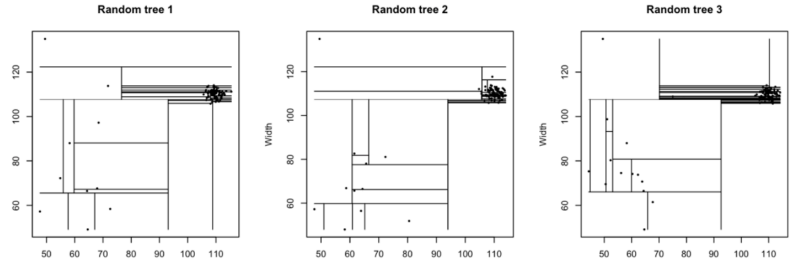
Algorithm
Separate all of the points by randomly splitting region into smaller and smaller pieces
1. randomly choose: choose feature and randomly choose value
2. continue to make splits
Until: each point lies in its own subregion, or subregion contains maximum number of points

Isolation Score
- Isolation of a point: measured by how fast it can be separated by sequence of random splits
  = number of splits needed for isolation
- Maximum path-Length = number of splits needed to separated each point = build the tree
- Path-length: number of random splits needed to isolate point
- uses standardized path length:
  - between 0 and 1
  - Score near 0 => long path length => not easily separated => inlier
  - Score near 1 => small path length => easily isolated by random splits => likely to be outlier



Isolation Forests
- grows many isolation trees
- used together for a better measure of how anomalous point is
- each tree uses a subsample of all the points
- number of observations per tree can set set - eg 100 trees
- score: is average of the score of all trees
- advantage: average score is more robust & faster tree growing





Anomaly detection using supervised learning

Motivation
- if labeled data exists for evaluation of unsupervised learning
- then why not used supervised learning directly - eg. logistic regression

Anomaly detection vs supervised learning problem
- wich to use depends on:
  1. number of positive examples
  2. will new anomalies be similar to previously seen anomalies
- anomaly detection: use if
  - if very small number of positive example eg. 0-10 positives and many negatives
    eg 10 vs 10.000
    => **not enough examples** for supervised to learning algorithm to detect pattern
  - if there are different types of anomalies
    => if **future anomalies might be very different from previously seen anomalies**
- supervised learning:
  - if large number of positive and negatives

  => if **anticipated anomalies will be similar to current anomalies**

Example: Fraud detection

only seen very few examples	=> treat as anomaly detection problem
seen many examples and various ways of fraud	=> treat as supervised learning problem
never fraud until now but want to be able to detect it	=> use anomaly algorithm

Anomaly Detection in Time Series

Outliers in Time Series
Are shifts in the level of a time series that cannot be explained

Methods
- Main Method:
  - Time Series plot
  - STL
  - Seasonal-Hybrid ESD algorithm

- Decomposition
  - standard STL decomposition
  - SH-ESD

- Prediction based
  - Idea: fit model on previous observations to predict current observation

  - Simple TS Models:
    - Simple Moving Average
    - Exponential Moving Average
  - Advanced TS Models:
    - ARIMA
    - ETS

  - Machine Learning:
    - Type
      - supervised = predict anomaly
      - unsupervised = predict next observation & residual analysis
    - Tree-based: CART & xgboost
    - Neural Networks

Decomposition based detection
Idea: remove trend and seasonality to extract remainder
=> allows to find global and local outliers
=> local outliers are not masked by seasonality

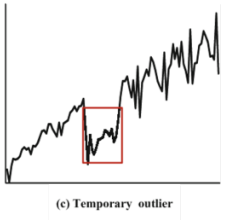
Prediction based detection
1. fit time-series model on previous observations
2. predict current observation
3. Create confidence bounds using in-build CI / PI or multiple of standard deviation
4. Check if estimate is within bounds
   => Yes = Inlier
   => No = Outlier

General Approach
1. try simplest approaches first => time series plot & STL decomposition
2. use advanced approaches

Types of time series anomalies

Additive outlier
- surprisingly large / small value occurring for single observation
- Subsequent observations are unaffected
- additive outlier patches: consecutive additive outliers

Innovational outlier
- has initial impact with effects lingering over subsequent observations
- influence may increase as time proceeds



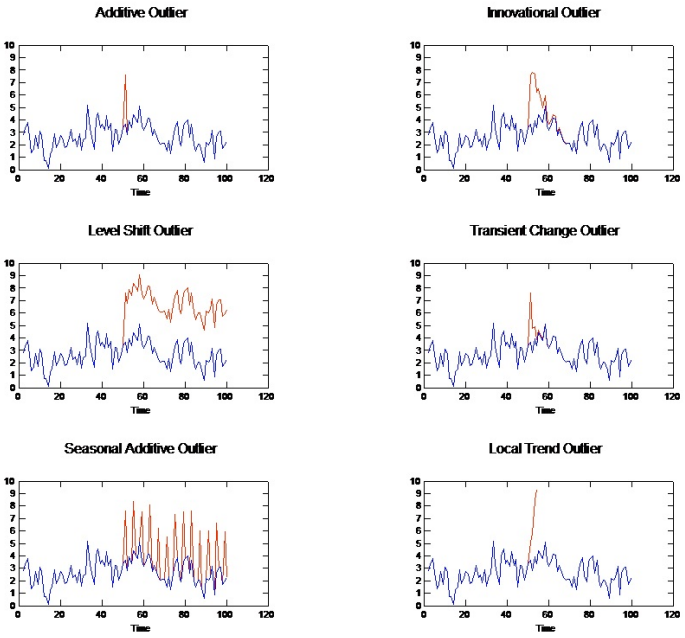
Temporal change outlier
- level shift for some a number of observations - eg server down

Level shift outlier
- change in mean of series
- all observations appearing after the outlier move to a new level
- vs additive: level shift outlier affects many observations and has a permanent effect

Transient change outlier
- similar to level shift outliers
- effect of outlier diminishes exponentially over the subsequent observations
- series returns to its normal level
-

Seasonal additive outlier
- surprisingly large / small value occurring repeatedly at regular intervals

Local trend outlier
- yields general drift in series caused by a pattern in outliers after the onset of initial outlier.



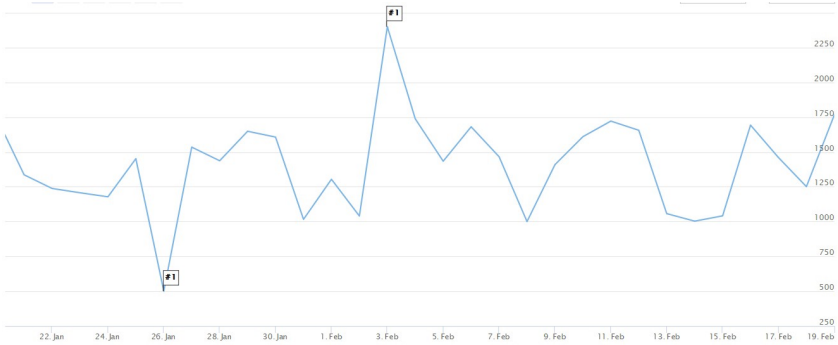
Moving average based detection

Moving Average Method
- moving average of past data used to estimate present day value is

Simple or Exponential MA
- can use simple or exponential smoothing average
- Simple: equal weightage
- Exponential: more weight to recent data

Algorithm
Basic and easy to execute

1. Prediction: use moving average of previous days as expected value of present day
2. Calculate Confidence Band: use multiple (3) of standard deviation of the MA of previous days
3. Check if estimated value is within confidence band



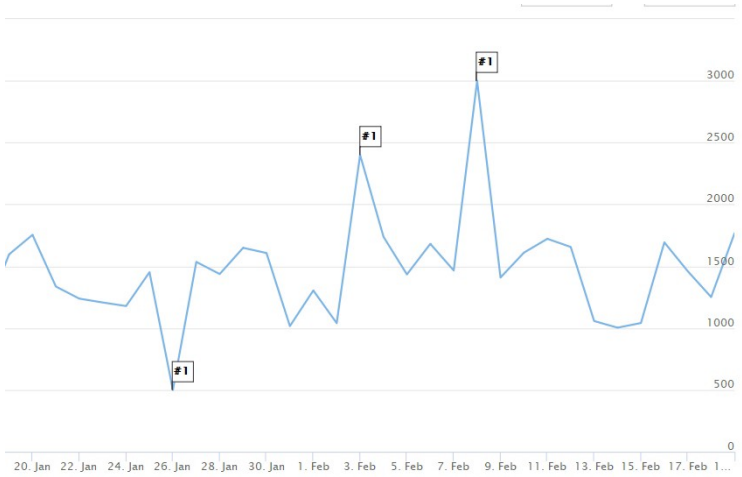
Date	Count	Upper Bound	Lower Bound	Expected Value
2018-01-26	500	2134.01	677.90	1405.95
2018-02-03	2400	2281.45	437.22	1359.33

Modification: Weekday Basis
- data from previous same weekdays is taken into account
- i.e use data from previous Mondays to estimate value on present Monday
- confidence band is multiple of standard deviation of previous weekdays data

Time-Series ARIMA based detection

ARIMA based
- better than moving average because it is more sophisticated forecasting method
- better because uses combination of auto-regression and moving average to estimate value
- Disadvantage: need to select p,d,q parameters for standard and seasonal components
- used by tsoutlier package in R

Example
Detected those points as an anomaly which seems to be an anomaly at first glance

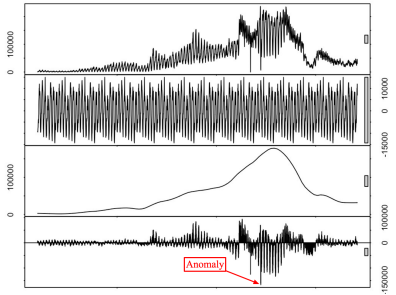


Date	Count	Upper Bound	Lower Bound	Expected Value
2018-01-26	500	2189	877	1385.39
2018-02-03	2400	2161	884	1381.93
2018-02-08	3000	2108	917	1390.29

Exponential Smoothing
- using Holt-Winters seasonal method
- if need to track multiple seasonal periods: use shortest one

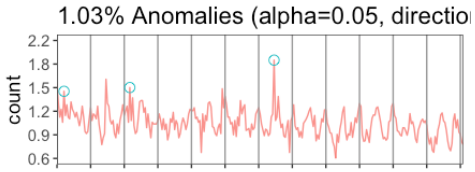
STL decomposition & SH-ESD

STL decomposition for anomaly detection
- uses simple statistical method
  => more transparent than ARIMA or tree ensembles
  => can still be interpreted
- can be used to detect both global and local anomalies
- seasonality removed to avoid fake anomalies due to seasonal behavior
- STL is based on LOESS splits time series into: trend, seasonal and remainder (residue)
- Anomaly detection algorithm:
  1. use STL to extract to remainder component
     => unexplained part after accounting for trend and seasonality
  2. analyze the residuals
  3. use threshold to determine if anomaly
- Advantage:
  - **trend** and **seasonality is already accounted** for
  - algorithm only needs to analyze the residuals
    => allows to **detect local anomalies** that would **otherwise be masked by the seasonality**
- vs Grubbs: finds unusual points with respect to seasonal contribution at time of occurrence
- detect level changes: (vs additive outlier) by analyzing rolling avg of series instead of original



SH-ESD: Seasonal Hybrid - Extreme Studentized Deviate
- used by Twitter's Anomaly Detection library - R library
- improvement on STL decomposition that builds upon Generalized ESD test
- Improvement:
  1. using more robust statistical metric - median
  2. using modified ESD
- Important - uses median absolute deviation
  - instead of using z-score => calculates modified z-score based on the median
  - analyze deviation of residues from the median(!)
    => allows more robust detection

Modified ESD
Modified ESD test is applied to residuals
Calculation of modified z-score
- how many deviations below or above the median a raw score
- higher z-score = higher mean deviation from median
  X = Original data, MAD = Median Absolute Deviation
  Modified Z-Score = (X - Median) / MAD
- „Median“ and „MAD„ used instead of “Mean” and “Standard Deviation” unlike z-score

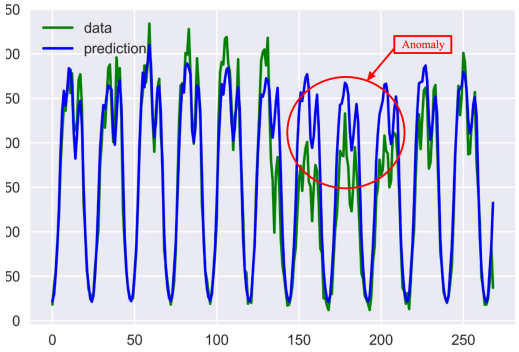


Machine Learning based detection

Tree-based detection
- Applied in 2 ways
  1. supervised: use supervised learning classify anomaly / non-anomaly data points
     => needs labeled data
  2. unsupervised:
     - tree learns to predict the next data point in series
     - use confidence interval to check to determine if data point is outlier
- Advantage:
  - process does not to be modeled
  - can easily incorporate independent variables
- can also use ESD test

=> use xgboost to predict next observation in series

Example: Prediction using CART model



Neural Networks
- using LSTM
- also supervised and unsupervised
- especially if multivariate series