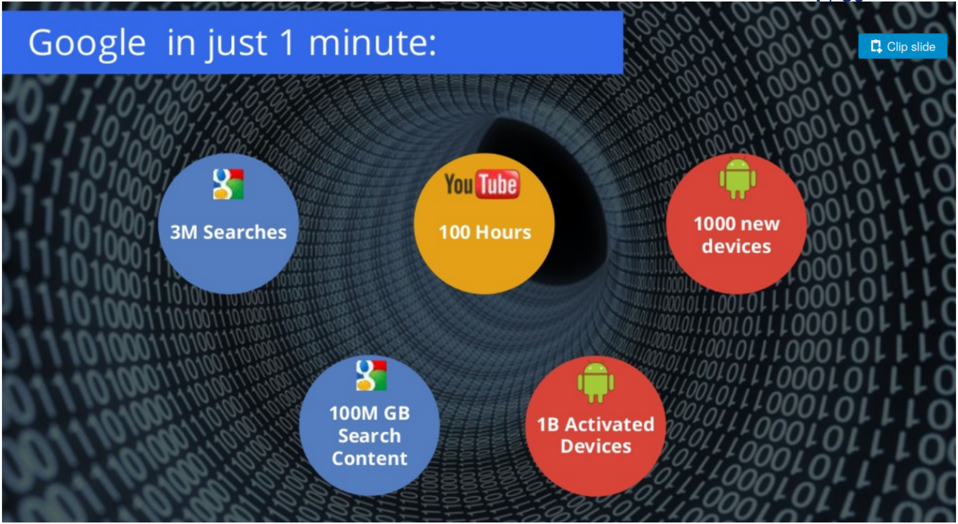
Machine Learning for Networks: Clustering and Anomaly Detection

Andrea Araldo

October 5, 2023

The challenge of Big Data



From D. Marcus (Google, Waze) presentation

Supervised Learning is impossible:

- Users do not label their actions
- We cannot label data manually

Section 1

Clustering

Unsupervised Learning: Clustering
- K-means clustering

Unsupervised Learning: Dimensionality reduction (next class)

Anomaly detection
- k-means anomaly detection
- Isolation Forests
- Neural Networks: Autoencoders

What is Clustering?

- Clustering means grouping M observations into clusters (partitions).

- There are no labels $\mathbf{y}$
  - We group observations, $\mathbf{x}^{(i)}$, by their similarity
  - *unsupervised learning*

- Exploratory technique to discover interesting relationships in data.

Clustering Application: Marketing

- Customer segmentation based on brand loyalty and price sensitivity scores.

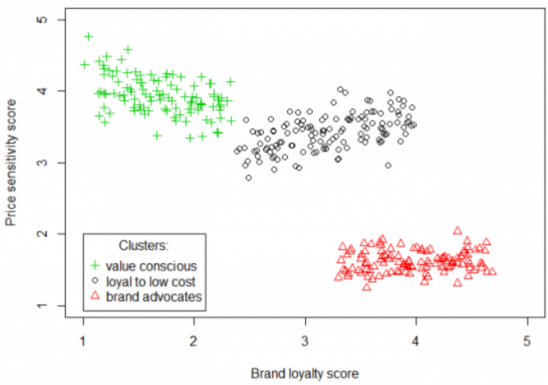
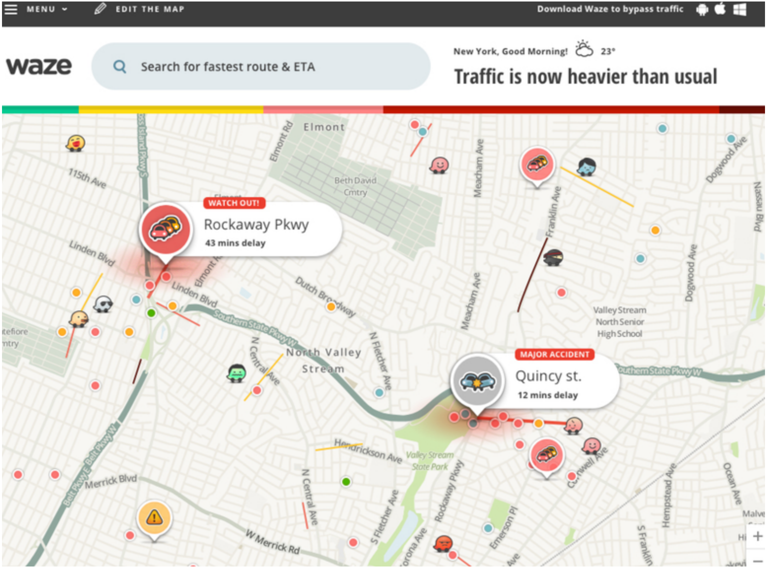
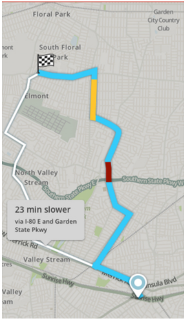


Fig. from http://www.select-statistics.co.uk/

Discover events in Waze





K-means Clustering

- K: Number of clusters. Hyper-parameter.
- Cluster centroid: mean of observations assigned to cluster C_k:

$$\bar{\mathbf{x}}_k \triangleq \frac{1}{|C_k|} \sum_{\mathbf{x} \in C_k} \mathbf{x}$$

($|C_k|$ is the number of samples of group k)

- **Within-cluster variation** of C_k (§10.3.1 of [JWHT13])

$$W(C_k) \triangleq \frac{1}{|C_k|} \sum_{\mathbf{x} \in C_k} d(\mathbf{x}, \bar{\mathbf{x}}_k)^2$$

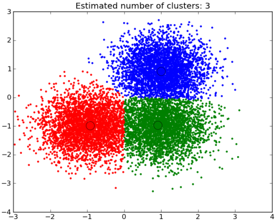
also called **inertia** [Cha].

- $d(.,.)$ is usually Euculidean distance
  - Scale!

- Goal: minimize total inertia

$$\min W = \sum_{k=1}^{K} W(C_k)$$

- $\Rightarrow$ Assign $\mathbf{x}$ to C_k with minimum $d(\mathbf{x}, \bar{\mathbf{x}}_k)$



Source: www.scikit-learn.org

K-means Clustering: Illustration

Minimize total inertia:

$$\min \sum_{k=1}^{K} \frac{1}{|C_k|} \sum_{\mathbf{x}\in C_k} d(\mathbf{x}, \bar{\mathbf{x}}_k)^2$$

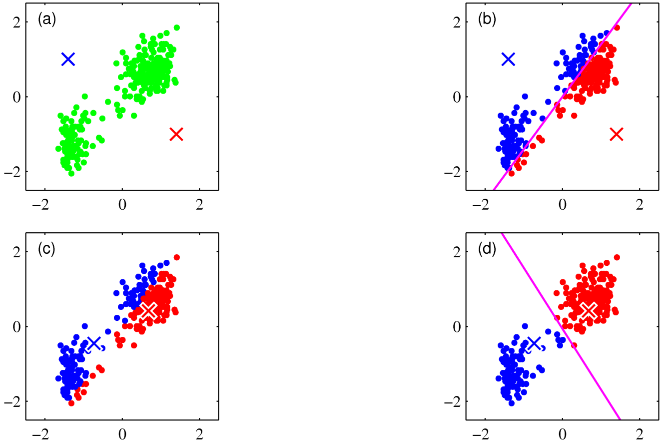


Fig.: Cristopher M. Bishop, *Pattern Recognition and Machine Learning*

K-means Clustering: Illustration

Minimize total inertia:

$$\min \sum_{k=1}^{K} \frac{1}{|C_k|} \sum_{\mathbf{x} \in C_k} d(\mathbf{x}, \bar{\mathbf{x}}_k)^2$$

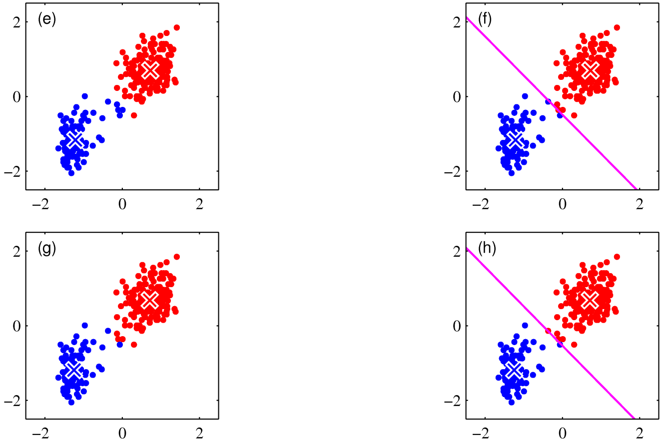
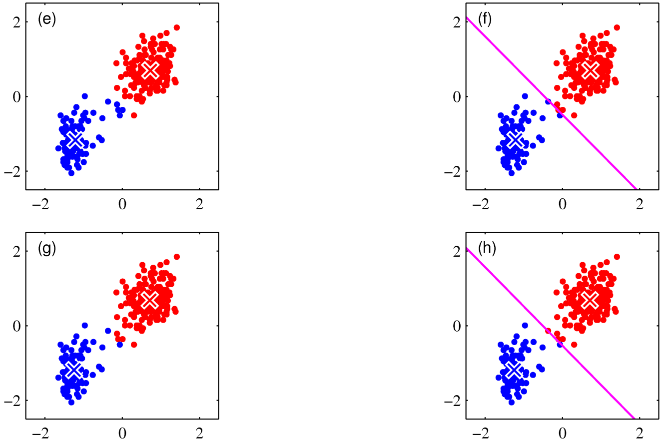


Fig.: Cristopher M. Bishop, *Pattern Recognition and Machine Learning*

K-means Clustering: Illustration

Minimize total inertia:

$$\min \sum_{k=1}^{K} \frac{1}{|C_k|} \sum_{\mathbf{x} \in C_k} d(\mathbf{x}, \bar{\mathbf{x}}_k)^2$$

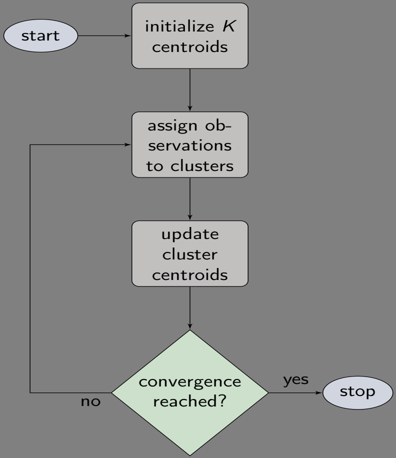


If you change initial centroids ⇒ final clusters change

- Repeat random initial centroids at least 20 times , and choose the clustering with the lowest W.

Fig.: Cristopher M. Bishop, *Pattern Recognition and Machine Learning*

K-means Clustering: Algorithm

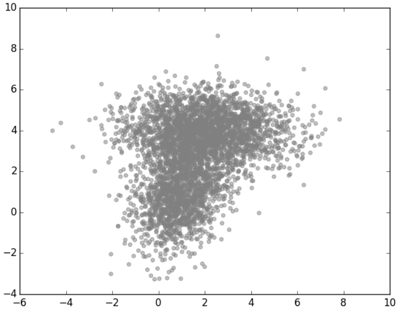


From [ZZPBA17]

- Thm: at each Assign and Update, the total W decreases until *convergence*.
- Smallest possible W at convergence?
- The W at convergence depends on the initial centroids chosen. So?
- Repeat the algorithm with different random initial centroids at least 20 times , and choose the clustering with the lowest W.

Choice of K

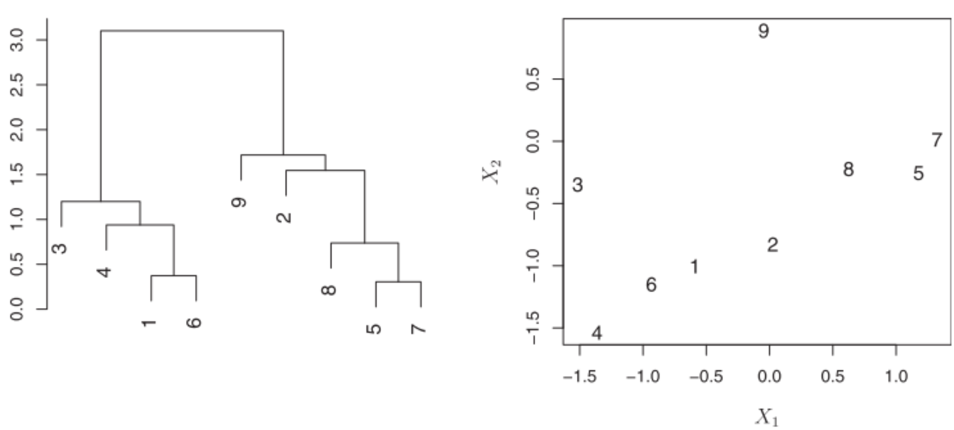
Which K would you choose?



Other clustering techniques choose K for you.

- Hierarchical Clustering
- DBScan
- ...

Hierarchical Clustering: Dendrogram

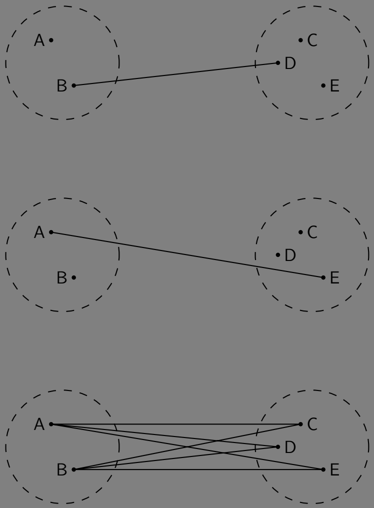


Source: James et Al, Introduction to Statistical Learning

- Distance in the y axis
- Distance between 5 and 7? And between {5,7,8,2} and { 9 }?
- Are 2 and 9 “close”? Distance between 2 and 9? Distance between {5,7,8} and {9}?

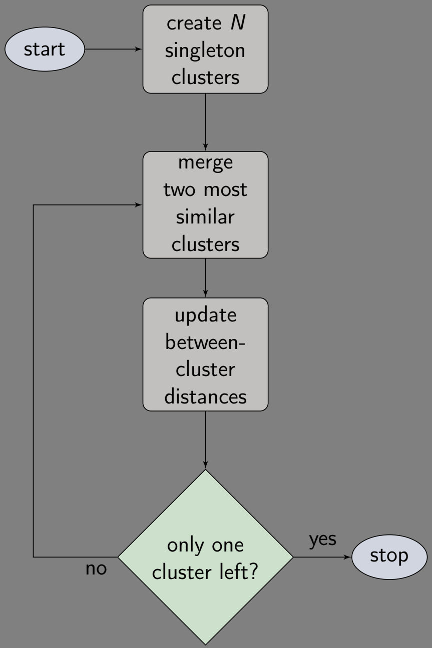
Hierarchical Clustering: Linkage

- Single linkage:
  minimum distance or nearest neighbor (2 closest border points)
- Complete linkage:
  maximum distance or farthest neighbor (2 farthest border points)
- Average linkage:
  average distance (all to all)



From [ZZPBA17]

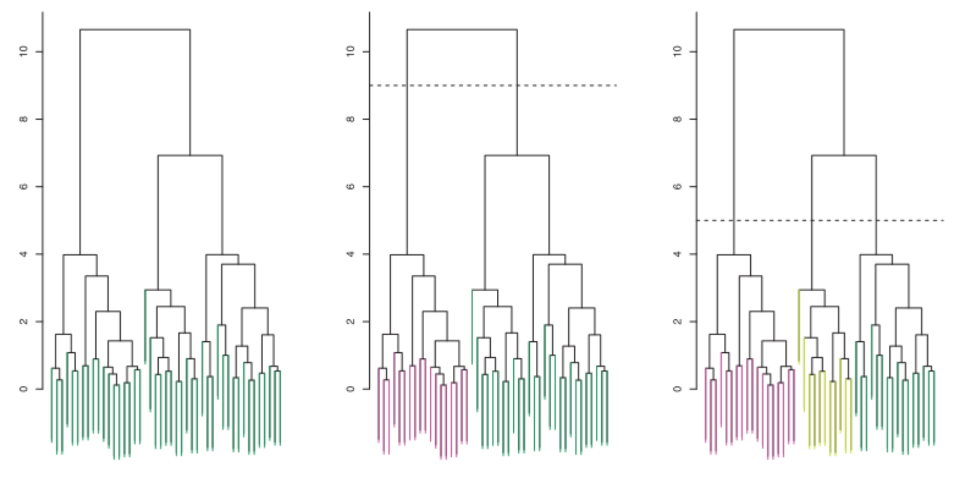
Building a dendrogram



From [ZZPBA17]

Choosing the clusters

We decide a Cut



Source: James et Al., Introduction to Statistical Learning

Same cluster with K-means, K=2?

Practical Considerations

Advantages

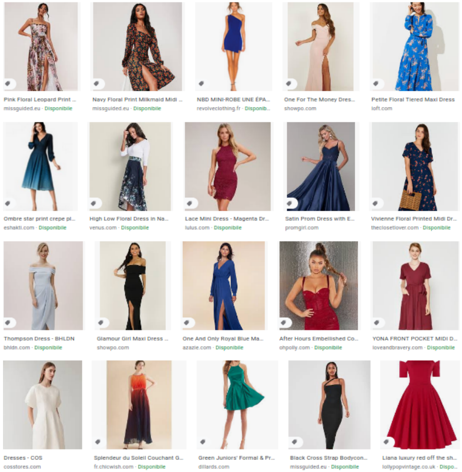
- No apriori number of clusters required
- Simple algorithms
- Self-organized structural view of data

Disadvantages

- Dendrogram often difficult to visualize
- Bad performance when inherent clusters are not hierarchical by nature
  - Ex.: incidents in the road

Similarity Measures

- How similar are two observations?
  - Color
  - Price
  - Size
  - Brand
  - Fabric
  - ...



Similarity Measures: Distance

Comparing two vectors, $\mathbf{z}_k$ and $\mathbf{z}_j$, with r variables

- With *Numerical data*:
  - **Euclidean distance**

$$d(\mathbf{z}_k, \mathbf{z}_j) = \sqrt{\sum_{i=1}^{r} (z_{ki} - z_{ji})^2}$$

  - Manhattan distance

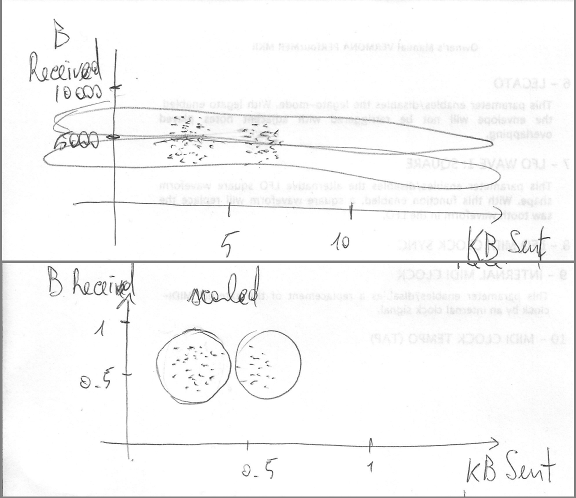
$$d(\mathbf{z}_k, \mathbf{z}_j) = \sum_{i=1}^{r} |z_{ki} - z_{ji}|$$

  - Minkowski distance

$$d(\mathbf{z}_k, \mathbf{z}_j) = \left[\sum_{i=1}^{r} |z_{ki} - z_{ji}|^m\right]^{1/m}$$

- Observe that:
  - Different groupings
  - Subjective and domain-dependent
  - Dependent on the variable type (discrete, continuous, binary).

Scaling



Without scaling, cluster would be just driven by features with large values.

Limits of within-cluster variation

Good clustering should:

- Minimize within-cluster variation (W)
- ... but also maximize the separation between clusters
- $\Rightarrow$ Inertia W is not enough

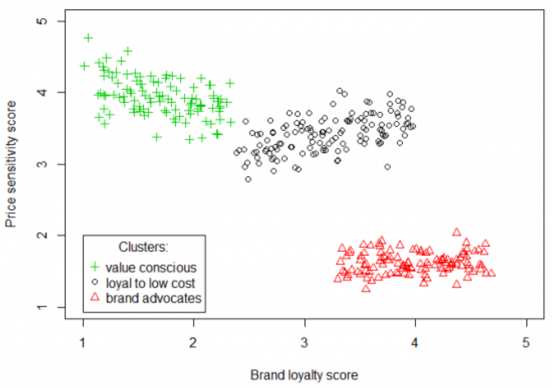


Fig. from http://www.select-statistics.co.uk/

Silhouette

Silhouette:

- Silhouette of sample $\mathbf{x}$:

$$s(\mathbf{x}) \triangleq \frac{b(\mathbf{x}) - a(\mathbf{x})}{\max(a(\mathbf{x}), b(\mathbf{x}))}$$

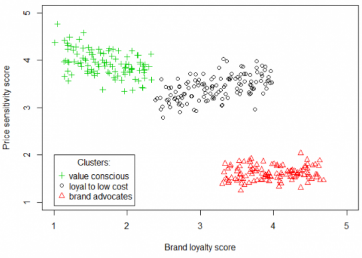
- $a(\mathbf{x})$= average distance between $\mathbf{x}$ and *all* other elements of its cluster (intra-cluster distance)
- $b(\mathbf{x})$= average distance between $\mathbf{x}$ and *all* elements of the second nearest cluster.

- Measures how well an observation fits a cluster

$$-1 < s(\mathbf{x}) < 1$$

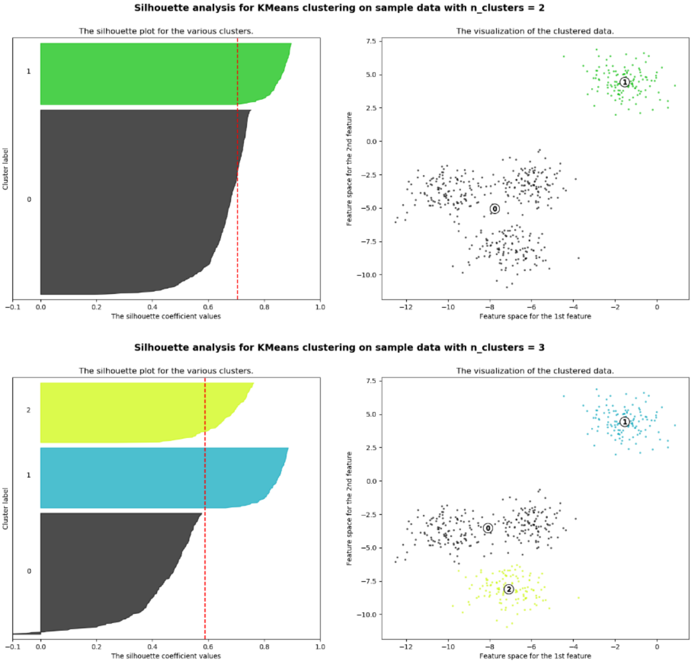
- We want $a(\mathbf{x})$ to be small and $b(\mathbf{x})$ to be large:

$$a(\mathbf{x}) \ll b(\mathbf{x}) \implies s(\mathbf{x}) \to 1$$



Source: http://www.select-statistics.co.uk/

Silhouette: visualization



See:

- **silhouette score**: Avg silhouette across all samples
- Cluster size

Fig. from Scikit-learn doc.

Choose the number K of clusters

Check the **silhouette score**

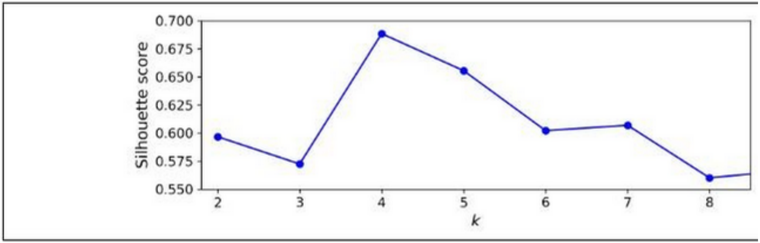


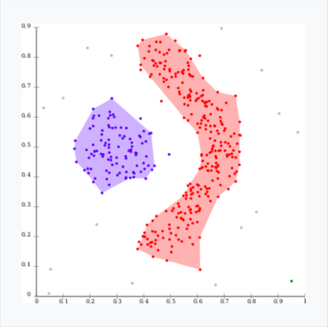
Figure 9-9. Selecting the number of clusters k using the silhouette score

Take $k = 4$ in the example.

Fig. From [Ger19]

Limits of K-means clustering

It can only find “spherical clusters”, all with the same size.
Otherwise you need to resort to other approaches (like DBSCAN)



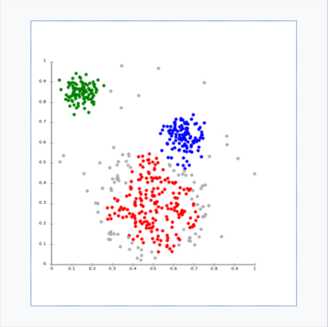


Fig. from Wikipedia, User:Chire.

Section 2

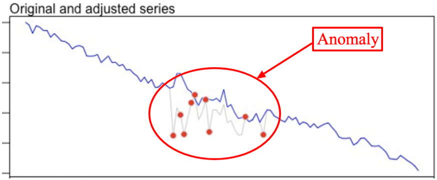
Anomaly detection

Definition

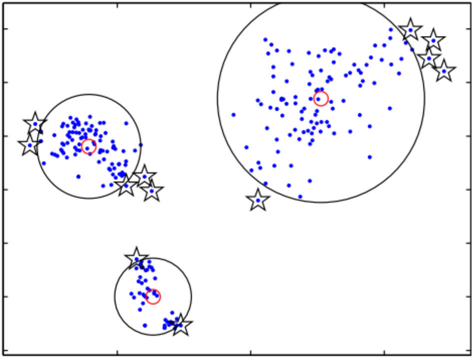
Anomaly: A sample that is not "normal" = outlier.

Causes:

- Intrusion or Fraud
- Sensor measurement errors
- Fault or Damage
- Unpredictable event (accident)



Credits to Tiunov



From [KH08]

Methods

Supervised methods:

- Training set with samples labeled as “normal”, “anomaly type 1”, , “anomaly type 2”

What if a new anomaly occurs, never seen before?

Methods

Supervised methods:

- Training set with samples labeled as “normal”, “anomaly type 1”, , “anomaly type 2”

What if a new anomaly occurs, never seen before?

Unsupervised methods:

- Clustering
- Isolation Forests
- Neural Networks
  - Auto-Encoders
  - ~~Self-organizing map~~

Application: Insider Threat

Wikipedia:
Edward J. Snowden *copied and leaked highly classified information from the National Security Agency (NSA) in 2013 when he was a Central Intelligence Agency (CIA) employee*

The Washington Post

MONDAY, JUNE 10, 2013

Man who leaked NSA secrets steps forward

A REPORTER'S ACCOUNT

To leaker, personal risks were clear

EDWARD SNOWDEN: 'I'M NOT GOING TO HIDE'

Booz Allen consultant could face prosecution

Risks of outsourcing

A historic leak

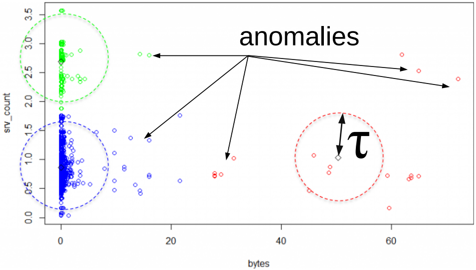
The activity of a malicious employee is different than normal $\Longrightarrow$ Anomaly.

We take inspiration from [GSG+15, TKH+17].
A sample is the activity of one employee in a day:

- Role (director, manager, intern)
- Department
- Project
- Num of files open
- Num of files written
- Num of copies to USB device
- Num of emails sent
- Size of attachments in emails
- …

Anomaly detection with K-means clustering

- Find K clusters
- **Anomaly score** $s(\mathbf{x}^{(i)})$: distance to the closest centroid
  - Don't confuse it with the silhouette score.
- If $s(\mathbf{x}^{(i)}) > \tau \implies \mathbf{x}^{(i)}$ is an anomaly



To know more: [HKF04]

Anomaly detection with K-means clustering

- Find K clusters
- **Anomaly score** $s(\mathbf{x}^{(i)})$: distance to the closest centroid
  - Don't confuse it with the silhouette score.
- If $s(\mathbf{x}^{(i)}) > \tau \implies \mathbf{x}^{(i)}$ is an anomaly

To know more: [HKF04]

Variation:
Clusters with few samples are also considered anomalies.

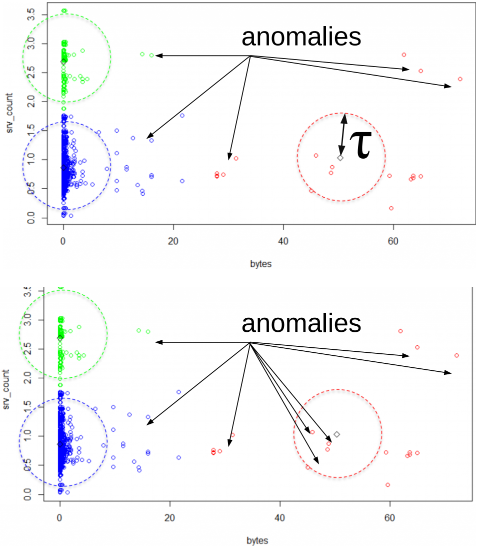
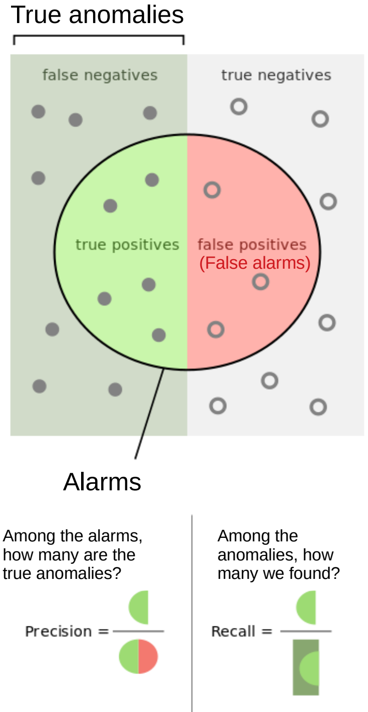


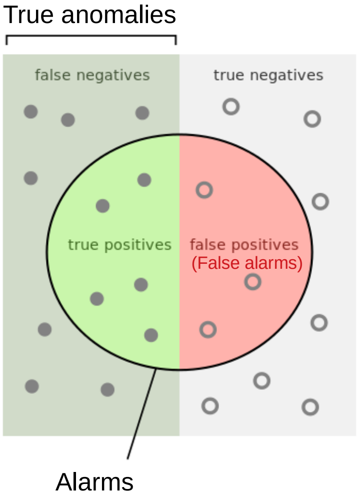
Fig. from Keppel and Schmalz.

Precision and Recall



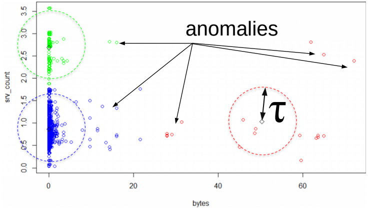
Figs. from Wikipedia and Walber (Wikipedia), modified.

Precision and Recall



Among the alarms, how many are the true anomalies?

Among the anomalies, how many we found?

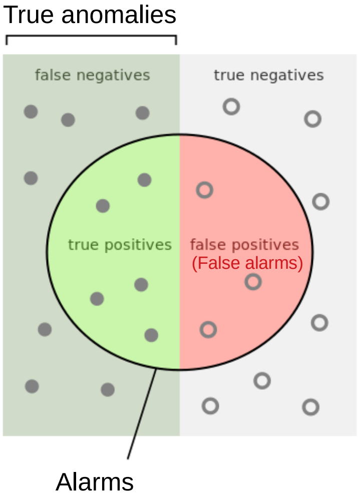


$$\tau \nearrow \Longrightarrow \text{precision} \nearrow, \text{recall} \searrow$$

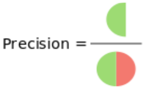
$$\tau \searrow \Longrightarrow \text{precision} \searrow, \text{recall} \nearrow$$

Figs. from Wikipedia and Walber (Wikipedia), modified.

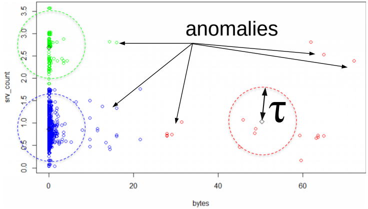
Precision and Recall



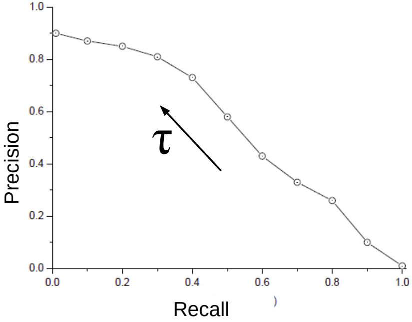
Among the alarms, how many are the true anomalies?



Among the anomalies, how many we found?



$$\tau \nearrow \Longrightarrow \text{precision} \nearrow, \text{ recall} \searrow$$
$$\tau \searrow \Longrightarrow \text{precision} \searrow, \text{ recall} \nearrow$$



Figs. from Wikipedia and Walber (Wikipedia), modified.

Precision-Recall Curve

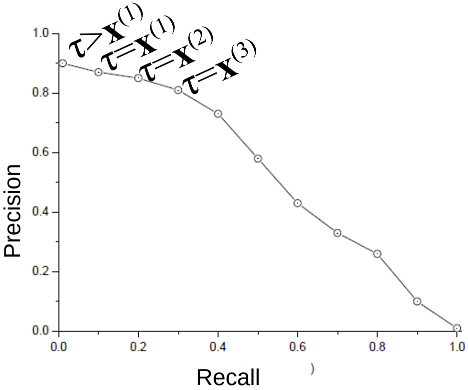


Fig. from Wikipedia.

To build the curve with K-means clust.:

- K-means clustering
- Compute anomaly score $s(\mathbf{x}^{(i)})$
  (nothing to do with silhouette!)
- Order $\mathbf{x}^{(i)}$ from the highest to the lowest $s(\mathbf{x}^{(i)})$
- Compute Pr. and Re. when anomaly is $\mathbf{x}^{(1)}$
- Compute Pr. and Re. when anomalies are $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots$
- Compute Pr. and Re. when anomalies are $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \mathbf{x}^{(3)} \ldots$
- $\ldots$

Area Under the Curve

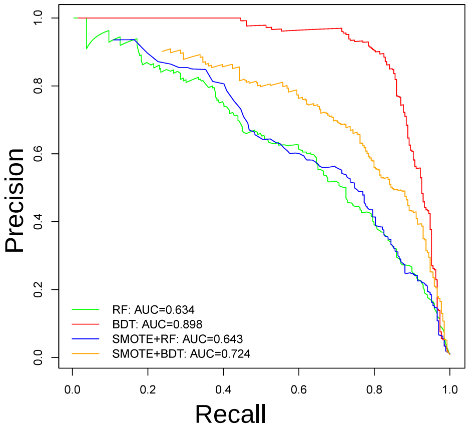


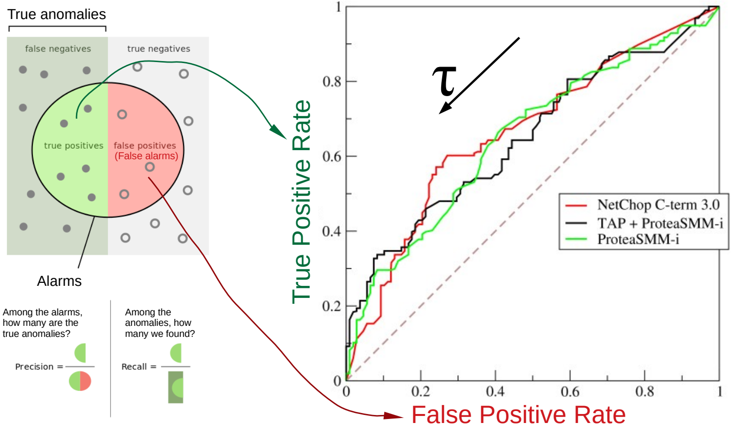
Fig. from [LC19].

In this figure:

- Each curve represents a different anomaly detection method
- Each point represents an anomaly detector
- **Area Under the Curve** (**AUC**): quality of a method

ROC Curve

Receiver-Operating Characteristic (ROC) Curve

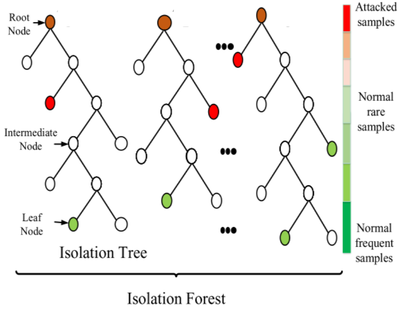


Right Fig. from Wikipedia

$$\text{True Positive Rate} = TP / \underbrace{(TP + FN)}_{\text{All positives}} = \text{Recall}$$

$$\text{False Positive Rate} = FP / \underbrace{(FP + TN)}_{\text{All normal samples}} \; : \text{Probability of False Alarms.}$$

Isolation Forest



From [ALHK19]

Assumption:

- The samples that isolate immediately are very different from the others
  $\implies$ anomalies

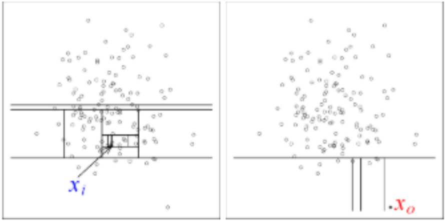
Train an extra-tree

- No need to compute impurity metrics
  $\implies$ No labels needed

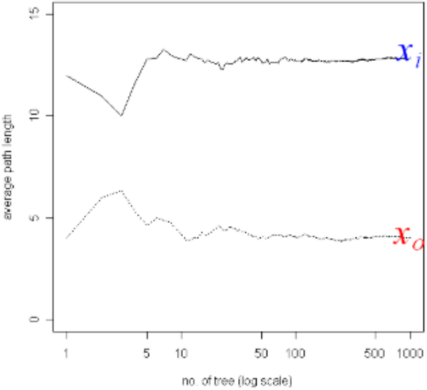
Compute sample scores (simplified):

- P: average tree depth
- $h(\mathbf{x}^{(i)})$: path length
  Depth of the leaf in which the sample falls, averaged across all trees
- $s(\mathbf{x}^{(i)}) = 2^{-\frac{h(\mathbf{x}^{(i)})}{P}}$

Isolation Forest

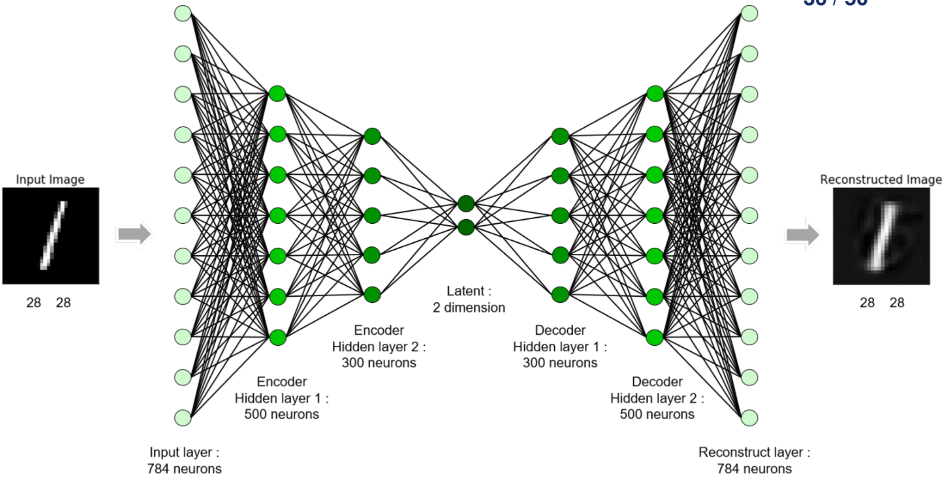


(a) Isolating x_i (b) Isolating x_o



From [TMZ08]

Autoencoder



From medium.com

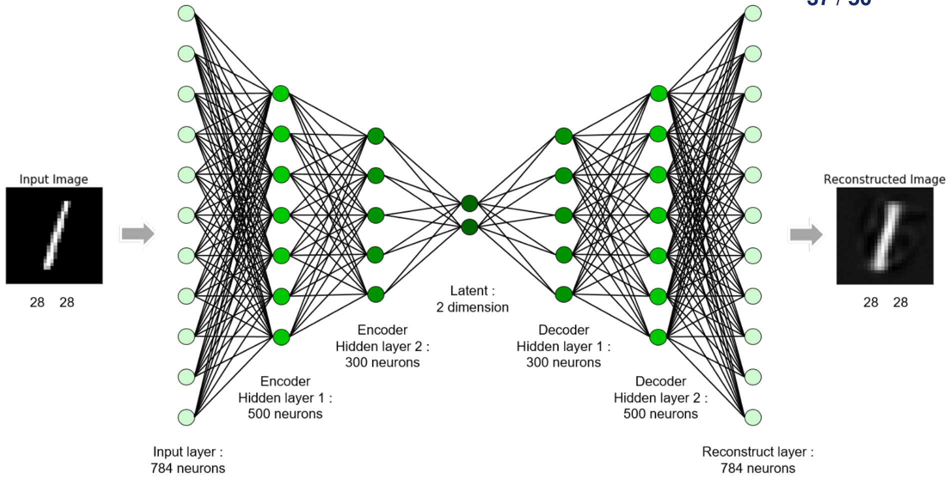
Autoencoder:

- Symmetric NN: num of outputs = num of features
- Train the NN to produce output$\simeq$ input:

$$J(\theta, \mathbf{x}^{(i)}) = ||\mathbf{x}^{(i)} - h_\theta(\mathbf{x}^{(i)})||^2$$

- Bottleneck: to “compress” the information in fewer neurons

Autoencoder for anomaly detection



From medium.com

Assumption:

- Normal samples are the majority
- $\Longrightarrow$ NN learns to reconstruct normal samples
- $\Longrightarrow$ and fails to reconstruct anomalous samples
- $\Longrightarrow$ anomalies are not compressible!

Score = reconstruction error:

$$s(\mathbf{x}^{(i)}) = ||\mathbf{x}^{(i)} - h_\theta(\mathbf{x}^{(i)})||^2$$

Self-Organizing Map (SOM)

Characteristics:

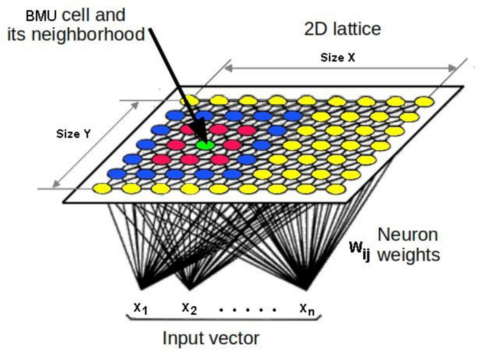
- NN with one layer only
- M neurons, disposed as a square.
- Each feature is connected to all neurons
- θ_q: weight of the q-th neuron.

Output:

- For each $\mathbf{x}^{(i)}$, the best matching unit (neuron) is activated

$$\mathrm{bmu}(\mathbf{x}^{(i)}) = \arg\min_q ||\mathbf{x}^{(i)} - \theta_q||^2$$

- Dimensionality reduction: we describe all the samples with fewer neurons



From [BD13]

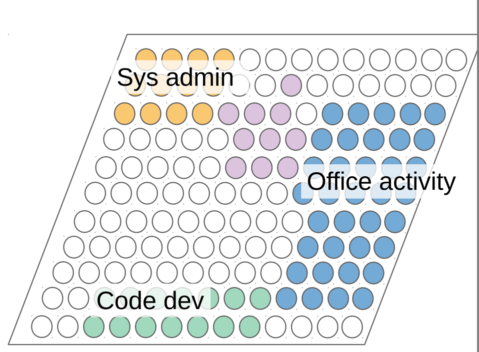
SOM: topological properties

Training:

- For each $\mathbf{x}^{(i)}$, find the best matching unit (bmu) q
- Modify θ_q in order to get closer to $\mathbf{x}^{(i)}$
- Modify also the weight of the neighbors of q

After training, similar $\mathbf{x}^{(i)}$ tend to activate close units

- Each cluster of samples corresponds to a region of the map



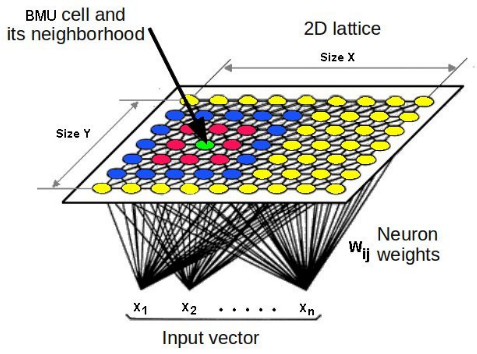
Anomaly detection with SOMs

- A SOM compresses the information of a dataset into few neuron weights (similar to auto-encoders).
- A SOM is trained to compress well the majority of samples (normal)
- The error is large with anomalies.

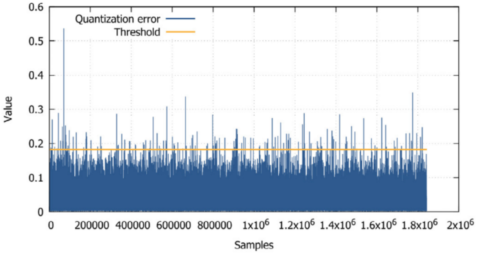
$$q^* = \mathrm{bmu}(\mathbf{x}^{(i)})$$

Quantization error:

$$s(\mathbf{x}^{(i)}) = ||\mathbf{x}^{(i)} - \theta_{q^*}||^2$$

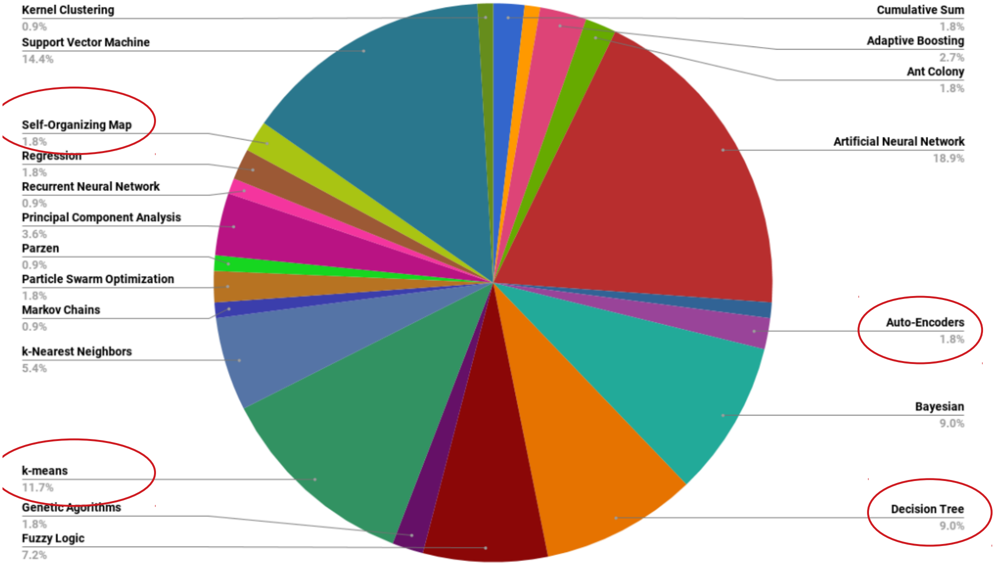


From [BD13]



From [VBMN18]

Different methods



From [HBB+18]

Applications to networks

Goal	Features	Method	Reference
Discover insider threat	Computer activity from logs	Deep NN, Recurrent NN	[TKH+17]
Discover Network Intrusion	IP and TCP connection info	Almost all	A lot
Find malicious sensors	Link delays	SOM	[WWW+13]
Find smart energy grid meters reporting wrong measurements	Electric measures	iForests	[ALHK19]
Predictive Maintenance: Predict which turbine is going to fail	Recordings of rotation speeds	SOM	[VBMN18]
Anomalous electric signals	Time series of signals	KMeans	Amid Fish blog

(Un - Semi)Supervised approaches

Unsupervised approach:

- Form clusters or forests or train NN on all the dataset $\mathscr{D}$ (normal + anomaly)
- Compute the score $s(\mathbf{x}), \forall \mathbf{x} \in \mathscr{D}$
- If $s(\mathbf{x}) > \tau \implies$ anomaly

Semi-supervised approach:

- Form clusters or forests or train NN on only normal samples
- When a new sample $\mathbf{x}$ arrives, compute the score $s(\mathbf{x})$
- If $s(\mathbf{x}) > \tau \implies$ anomaly

Note: You need to have **samples labeled as normal**.

Supervised approach:

- Classify in normal / anomaly
- You can also classify anomaly types

Note: You need to have a training set with **all** samples labeled.

(Un - Semi)Supervised approaches

Unsupervised approach:

- Form clusters or forests or train NN on all the dataset $\mathscr{D}$ (normal + anomaly)
- Compute the score $s(\mathbf{x})$, $\forall \mathbf{x} \in \mathscr{D}$
- If $s(\mathbf{x}) > \tau \implies$ anomaly

Semi-supervised approach:

- Form clusters or forests or train NN on only normal samples
- When a new sample $\mathbf{x}$ arrives, compute the score $s(\mathbf{x})$
- If $s(\mathbf{x}) > \tau \implies$ anomaly

Note: You need to have **samples labeled as normal**.

Supervised approach:

- Classify in normal / anomaly
- You can also classify anomaly types

Note: You need to have a training set with **all** samples labeled.

Note

The unsupervised and semi-supervised approaches assume **anomalies are a minority** (less than 1%).
Not always true (Denial of Service attack)

Training and Test sets

- Hyper-parameters to tune:
  In K-means clustering:
  - K
  - The distance function (Euclidean, Manhattan, Minkowski, etc.)

  In iForests
  - Number of trees

  In autoencoders
  - The NN architecture

  ...
  In general
  - The score threshold τ to recognize anomalies
- We compute Precision, Recall, ROC Curve, etc, based on **ground truth**

To avoid **data leakage**:

- Split the dataset in training/test data
- Choose the hyperparameters only based on training data
- Check the performance on test data

To know more

- Anomaly Detection in Telecommunications by Valentina Djordjevic, Video
- Jan van der Vegt: A walk through the isolation forest | PyData Amsterdam 2019, Video.
- Johnson, R.A. and Wichern, D.W. (2002). Applied Multivariate Statistical Analysis, 5th ed. Prentice Hall Sections 12.1, 12.2, 12.3 and 12.4
- Isolation Forest: original paper [TMZ08] (1062 citations)
- Anomaly Detection with Robust Deep Autoencoders [ZP17]

Unsupervised Learning: Clustering

- K-means clustering

Unsupervised Learning: Dimensionality reduction (next class)

Anomaly detection

- k-means anomaly detection
- Isolation Forests
- Neural Networks: Autoencoders

References I

[ALHK19] Saeed Ahmed, Youngdoo Lee, Seung Ho Hyun, and Insoo Koo, *Unsupervised Machine Learning-Based Detection of Covert Data Integrity Assault in Smart Grid Networks Utilizing Isolation Forest*, IEEE Transactions on Information Forensics and Security **14** (2019), no. 10, 2765–2777.

[BD13] Juan Carlos Burguillo and Bernabe Dorronsoro, *Using Complex Network Topologies and Self-Organizing Maps for Time Series Prediction*, Advances in Intelligent Systems and Computing (2013).

[Cha] Marie Chavent, *Clustering methods - Lecture Notes*, Université de Bordeaux.

[Ger19] Aurelien Geron, *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*, O'Reilly, 2019.

References II

[GSG+15] Gaurang Gavai, Kumar Sricharan, Dave Gunning, John Hanley, Mudita Singhal, and Rob Rolleston, *Supervised and unsupervised methods to detect insider threat from enterprise social and online activity data*, Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications **6** (2015), no. 4, 47–63.

[HBB+18] Hanan Hindy, David Brosset, Ethan Bayne, Amar Seeam, Christos Tachtatzis, Robert Atkinson, and Xavier Bellekens, *A Taxonomy and Survey of Intrusion Detection System Design Techniques, Network Threats and Datasets*, no. June.

[HKF04] Ville Hautamaki, Ismo Karkkainen, and Pasi Franti, *Outlier Detection Using k-Nearest Neighbour Graph*, IAPR International Conference on Pattern Recognition (ICPR), 2004.

[JWHT13] Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani, *An introduction to Statistical Learning*, vol. 7, 2013.

References III

[KH08] Pekka Kumpulainen and Kimmo Hätönen, *Local anomaly detection for mobile network monitoring*, Information Sciences **178** (2008), no. 20, 3840–3859.

[LC19] Sunbok Lee and Jae Young Chung, *The machine learning-based dropout early warning system for improving the performance of dropout prediction*, MDPI Applied Sciences **9** (2019), no. 15.

[TKH+17] Aaron Tuor, Samuel Kaplan, Brian Hutchinson, Nicole Nichols, and Sean Robinson, *Deep learning for unsupervised insider threat detection in structured cybersecurity data streams*, Artificial Intelligence for Cyber-Security, 2017, pp. 224–234.

[TMZ08] Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou, *Isolation Forest*, EEE International Conference on Data Mining, 2008.

References IV

[VBMN18] Alexander Von Birgelen, Davide Buratti, Jens Mager, and Oliver Niggemann, *Self-Organizing Maps for Anomaly Localization and Predictive Maintenance in Cyber-Physical Production Systems*, Procedia CIRP **72** (2018), 480–485.

[WWW+13] Wei Wang, Huiran Wang, Beizhan Wang, Yaping Wang, and Jiajun Wang, *Energy-aware and self-adaptive anomaly detection scheme based on network tomography in mobile ad hoc networks*, Information Sciences **220** (2013), 580–602.

[ZP17] Chong Zhou and Randy C. Paffenroth, *Anomaly detection with robust deep autoencoders*, ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2017.

[ZZPBA17] Fang Zhao, Haizheng Zhang, Francisco Pereira, and Moshe Ben-Akiva, *Clustering - Lecture Notes - Multivariate Data Analysis*, 2017.