Anomaly/Outlier Detection

- What are anomalies/outliers?
  - The set of data points that are considerably different than the remainder of the data
- Variants of Anomaly/Outlier Detection Problems
  - Given a database D, find all the data points $\mathbf{x} \in D$ with anomaly scores greater than some threshold t
  - Given a database D, find all the data points $\mathbf{x} \in D$ having the top-n largest anomaly scores f($\mathbf{x}$)
  - Given a database D, containing mostly normal (but unlabeled) data points, and a test point $\mathbf{x}$, compute the anomaly score of $\mathbf{x}$ with respect to D
- Applications:
  - Credit card fraud detection, telecommunication fraud detection, network intrusion detection, fault detection

Anomaly Detection

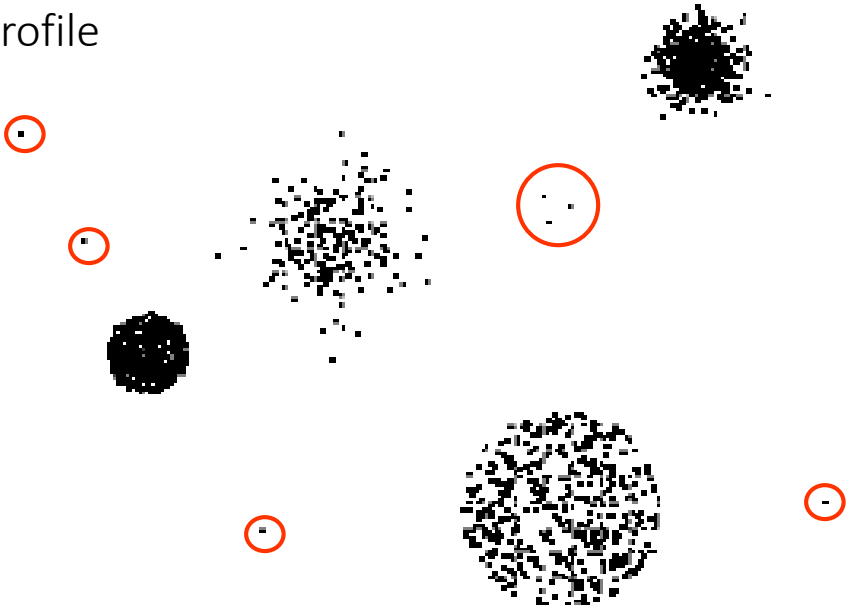
- Challenges
  - How many outliers are there in the data?
  - Method is unsupervised

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

Anomaly Detection Schemes

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

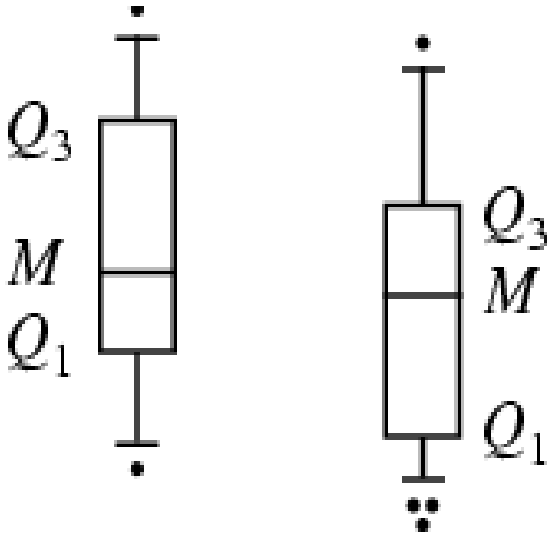
- Types of anomaly detection schemes
  - Graphical & Statistical-based
  - Distance-based
  - Model-based



Graphical Approaches

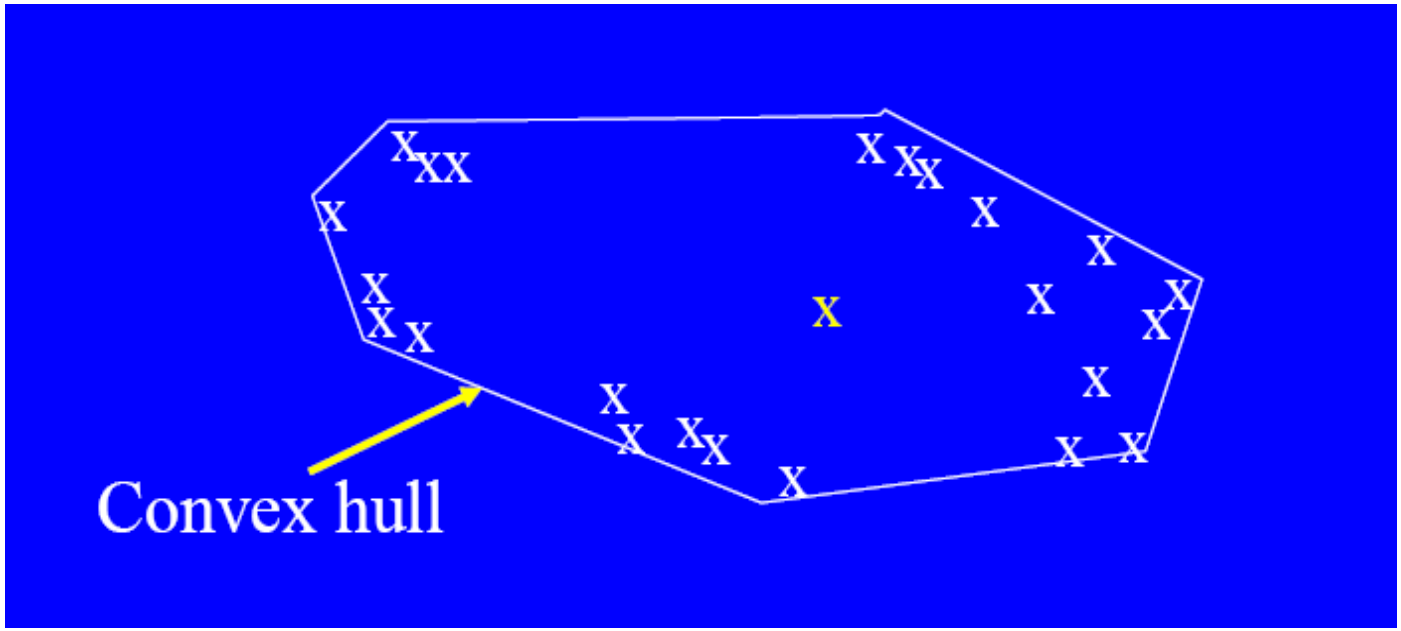
- Boxplot (1-D), Scatter plot (2-D), Spin plot (3-D)

- Limitations
  - Time consuming
  - Subjective



Convex Hull Method

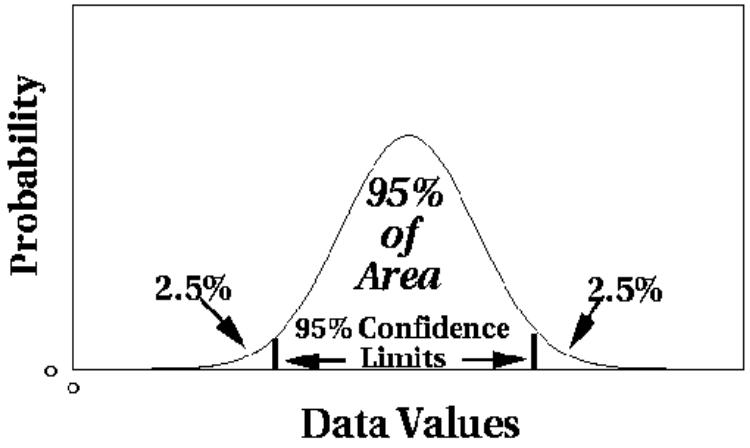
- Extreme points are assumed to be outliers
- Use convex hull method to detect extreme values



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

Statistical Approaches

- Assume a parametric model describing the distribution of the data (e.g., normal distribution)
- Apply a statistical test that depends on
  - Data distribution
  - Parameter of distribution (e.g., mean, variance)
  - Number of expected outliers (confidence limit)



Grubbs' Test

- Detect outliers in univariate data
- Assume data comes from normal distribution
- Detects one outlier at a time, remove the outlier, and repeat
  - H_0: There is no outlier in data
  - H_A: There is at least one outlier
- Grubbs' test statistic:

$$G = \frac{\max\left|X - \overline{X}\right|}{s}$$

- Reject H_0 if:

$$G > \frac{(N-1)}{\sqrt{N}} \sqrt{\frac{t^2_{(\alpha/N,N-2)}}{N-2+t^2_{(\alpha/N,N-2)}}}$$

Statistical-based – Likelihood Approach

- Assume the data set D contains samples from a mixture of two probability distributions:
  - M (majority distribution)
  - A (anomalous distribution)
- General Approach:
  - Initially, assume all the data points belong to M
  - Let $L_t(D)$ be the log likelihood of D at time t
  - For each point x_t that belongs to M, move it to A
    - Let L_{t+1} (D) be the new log likelihood.
    - Compute the difference, $\Delta = L_t(D) - L_{t+1}$ (D)
    - If $\Delta > c$ (some threshold), then x_t is declared as an anomaly and moved permanently from M to A

Statistical-based – Likelihood Approach

- Data distribution, D = (1 – λ) M + λ A
- M is a probability distribution estimated from data
  - Can be based on any modeling method (naïve Bayes, maximum entropy, etc)
- A is initially assumed to be uniform distribution
- Likelihood at time t:

$$L_t(D) = \prod_{i=1}^{N} P_D(x_i) = \left((1-\lambda)^{|M_t|} \prod_{x_i \in M_t} P_{M_t}(x_i) \right) \left(\lambda^{|A_t|} \prod_{x_i \in A_t} P_{A_t}(x_i) \right)$$

$$LL_t(D) = |M_t| \log(1-\lambda) + \sum_{x_i \in M_t} \log P_{M_t}(x_i) + |A_t| \log \lambda + \sum_{x_i \in A_t} \log P_{A_t}(x_i)$$

Limitations of Statistical Approaches

- Most of the tests are for a single attribute
- In many cases, data distribution may not be known
- For high dimensional data, it may be difficult to estimate the true distribution

Distance-based Approaches

- Data is represented as a vector of features
- Three major approaches
  - Nearest-neighbor based
  - Density based
  - Clustering based

Nearest-Neighbor Based Approach

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

Outliers in Lower Dimensional Projection

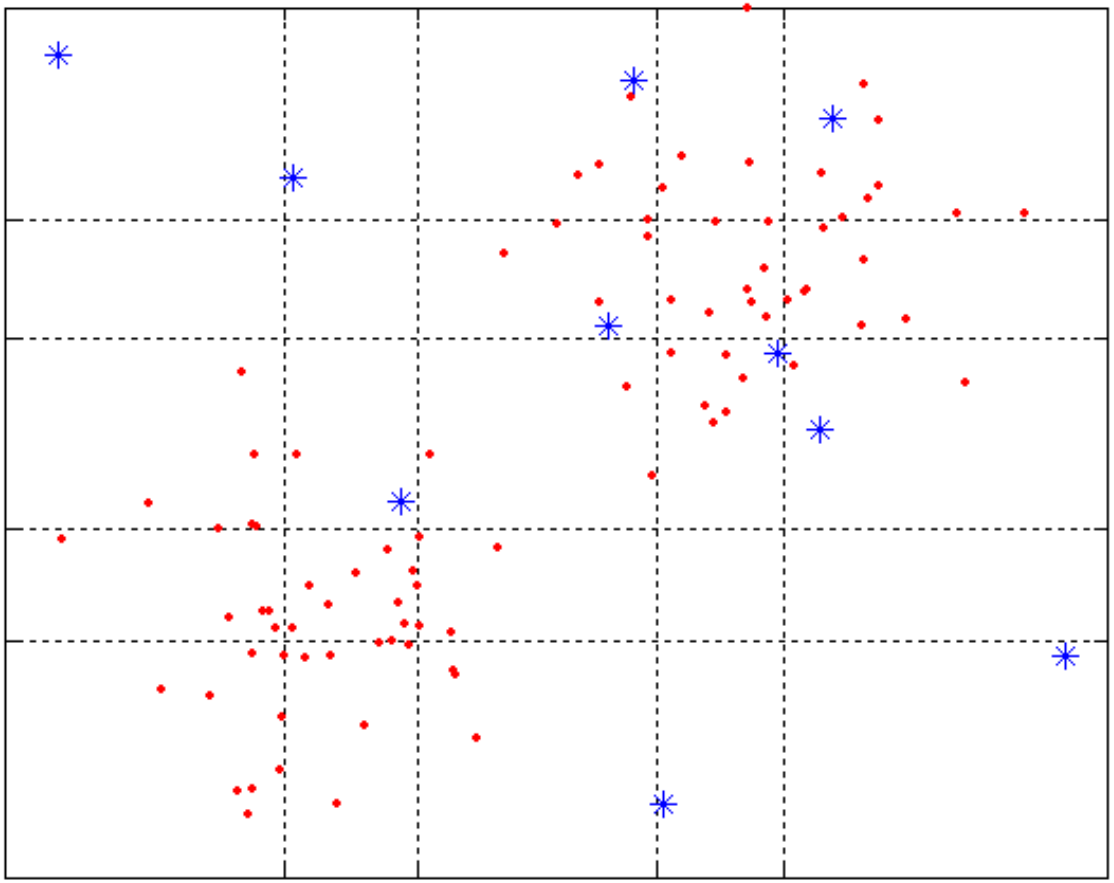
- Divide each attribute into ϕ equal-depth intervals
  - Each interval contains a fraction $f = 1/\phi$ of the records
- Consider a k-dimensional cube created by picking grid ranges from k different dimensions
  - If attributes are independent, we expect region to contain a fraction f^k of the records
  - If there are N points, we can measure sparsity of a cube D as:

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

  - Negative sparsity indicates cube contains smaller number of points than expected

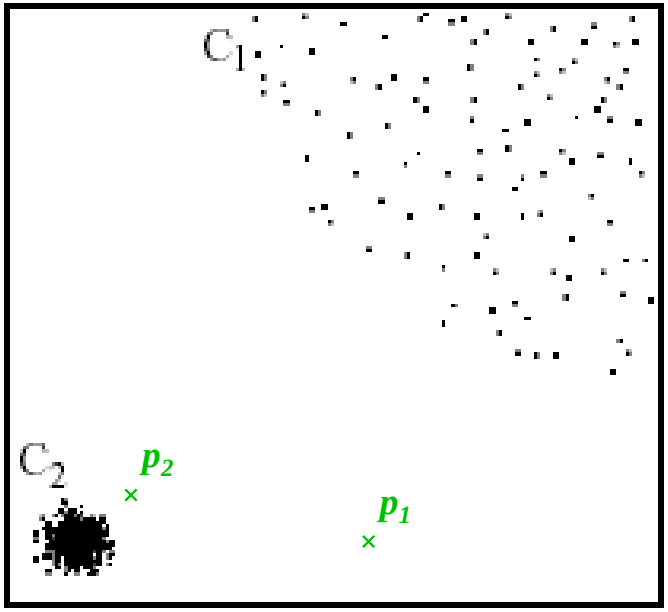
Example

- N=100, $\phi = 5$, $f = 1/5 = 0.2$, $N \times f^2 = 4$



Density-based: LOF approach

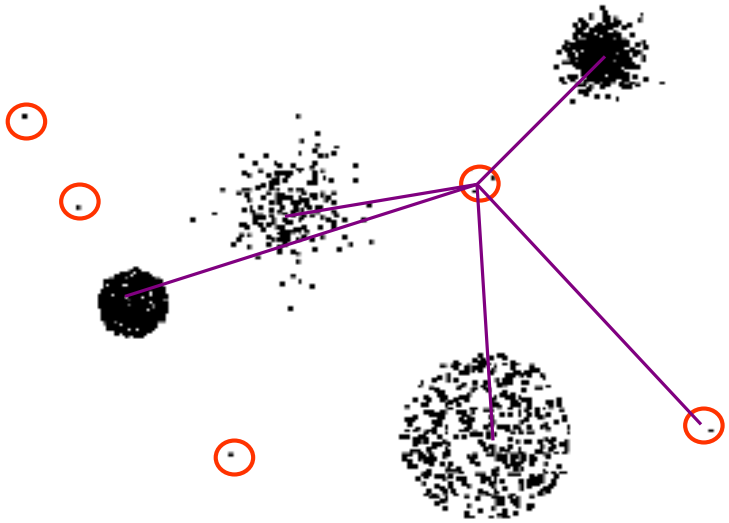
- For each point, compute the density of its local neighborhood
- Compute local outlier factor (LOF) of a sample *p* as the average of the ratios of the density of sample *p* and the density of its nearest neighbors
- Outliers are points with largest LOF value



In the NN approach, p_2 is not considered as outlier, while LOF approach find both p_1 and p_2 as outliers

Clustering-Based

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



Other approaches

- SVC methods: one-class SVM
- Data is unlabelled, unlike usual SVM setting.
- Goal: find hyperplane (in higher-dimensional kernel space) which encloses as much data as possible with minimum volume.
- Tradeoff between amount of data enclosed and tightness of enclosure; controlled by regularization of slack variables.

- Lets do it with python