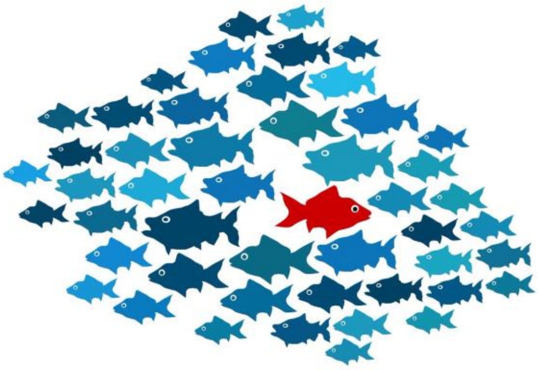
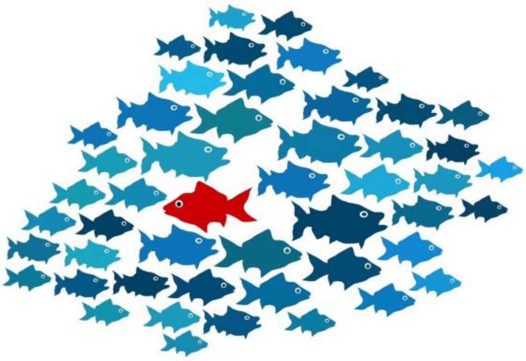
Scratch book unsupervised learning 12

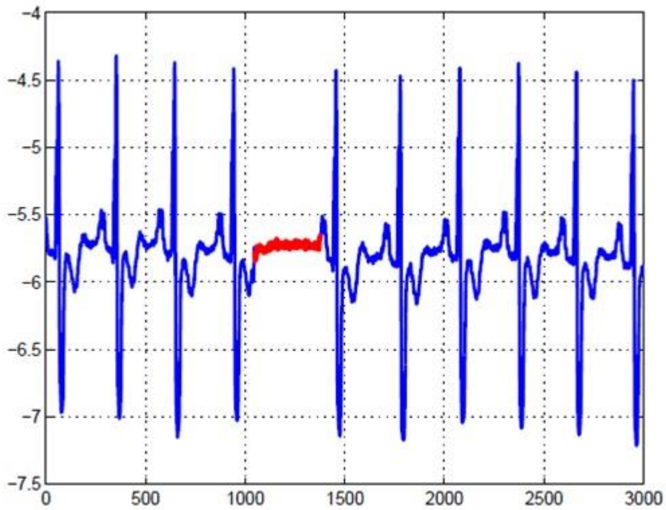
Anomaly detection





Anomaly – anything that does not look like everything else.

- Point anomaly – one point that differs from every one.
- **Contextual anomaly** - a point that differs from is environment.
- Collective anomaly



Unsupervised anomaly detection

- Statistical - We build a model for the data with the anomalies.
- Proximity-based – An anomalous point is far from other points.
- Density-based - An anomalous point has a low density.
- Clustering-based - An anomalous point does not sit well in any cluster.

Statistical anomaly detection

A) Define a statistical model on the data
B) Find improbable data points.

Example I – 1D gaussian data. - Anomalous points are points that are K SD from average.

Example II – Define GMM on the data and compute points with low probability in GMM.

A) Throw a coin if face then anomalous, if number then not. – probably lambda to be anomalous.
B) Each category has a different distribution.

Fit the data to the model.

More formally:

Assume a latent variable that defines for each point – with probability lambda anomalous, with probability (1-lambda) I am normal. Normal points have a distribution P1. We assume known \lambda and P.

$$P(\{x_i\}) = \Pi_i p(x_i) = \Pi_i\left[\left((\lambda * P(x_i|i\ An)\right)\ or\ \left((1-\lambda) * P1(x_i|model)\middle|i\ Normal\right)\right]$$

1) Assume that I know, which point is anomalous (A) and which point is not (N) and I have |A| number of anomalous point, |N| number of normal points.

$$L(\{x_i\}) = \Pi_i p(x_i) = \lambda^{|A|} * (1-\lambda)^{|N|} * \Pi_A P(x_i) * \Pi_N P(x_i|\theta)$$

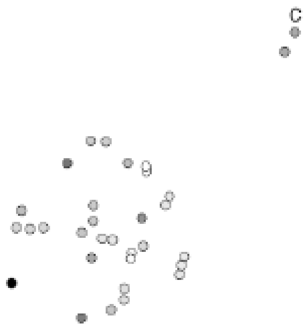
$$\log\left(L(\{x_i\})\right) = |A|\log(\lambda) \quad + |N| * log(1-\lambda) \quad + \sum_A \log(P) + \sum_N \log(P(x_i|\theta))$$

$$= |A|\,[\log(\lambda * P))] \quad + |N| * log(1-\lambda) \quad + \sum_N \log(P(x_i|\theta))$$

Derivate the last term by theta and obtain the ML parameters of the model.

2) Assume, I know the parameters theta, I can decide if a point is anomalous if its probability in the normal model is bellow some threshold.
3) Return to 1, until convergence.

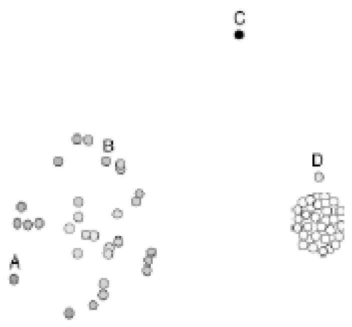
Proximity-based outlier detection



Outlier score is distance to kth nearest neighbor.

- Problem – sensitive to the selection of k (k=5)

  Also, cannot detect contextual anomalies.



Density-based outlier detection

An anomaly is not dense (compared to its neighbors)

1) DBSCAN
2) K Density
   - $K\ density(x) = [\frac{1}{K}\sum_{\{KNN\}} distance(x, Neighbor(x))]^{-1}$
   - $Anomalies\ have\ low\ K\ density$ $(compared\ with\ its\ KNN)$
   - $Anomaly\ Score = K\ density(x) * K/\sum K\_Density(KNN(x))$
   - Sensitive (slightly less than distance based) to k
   - Expensive to compute if not implemented (Can cost up to $O(N^2)$ if not using a KD-tree, else $O(N * log(N))$).

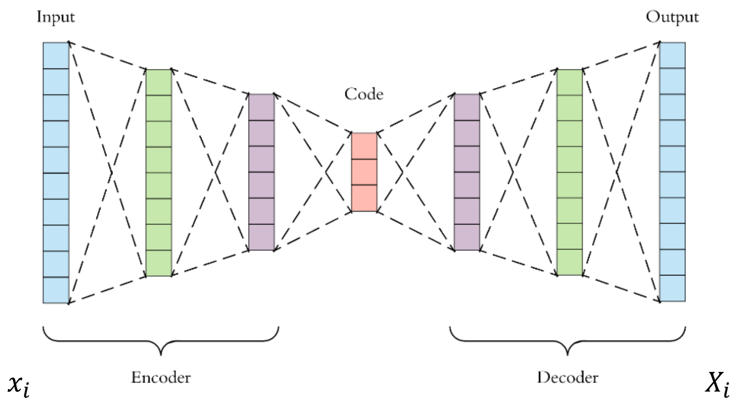
Cluster-based outlier detection

1) Cluster all the data (usually GMM/FCM/K means).
2) Remove all the points that have a low clustering score (negative silhouette score, or likelihood in GMM).
3) Go back to 1 until we find no points to remove.

Machine learning based anomaly detection

- One class classifier.
- Autoencoder.

Autoencoder

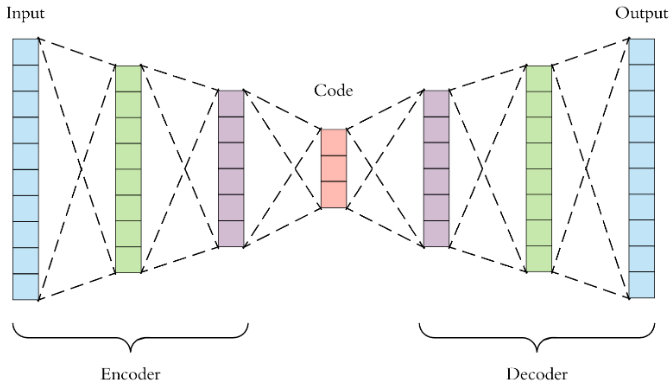


Train a machine to reproduce precisely the input by passing through a narrow layer.

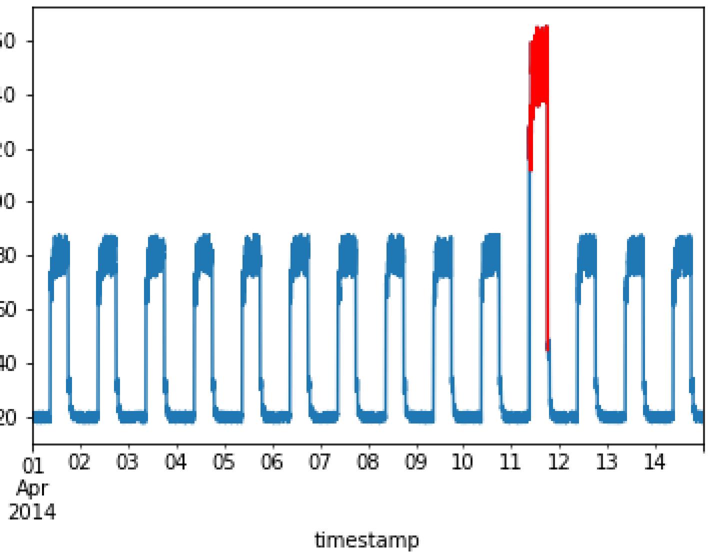
$$Score = \sum \left\| x_i - X_i \right\|^2 ; x_i = inpute, X_i = output$$

$$Error\ in\ reconstruction\ of\ x_i - AS(x_i) = \left\| x_i - X_i \right\|^2$$

x(t)



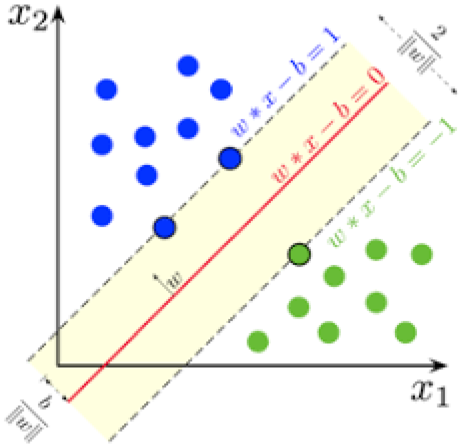
$x(t+1)$

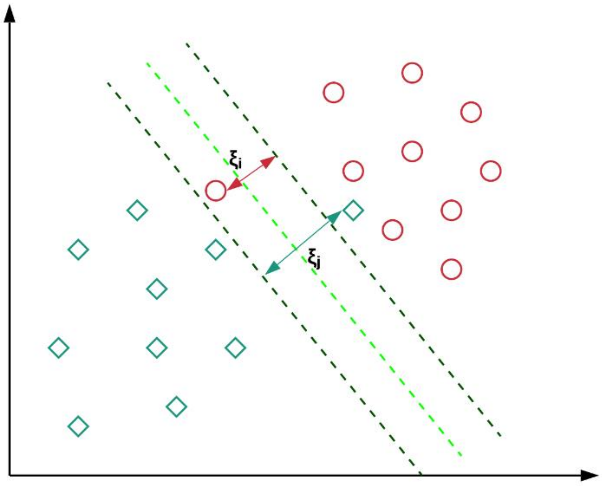


One class SVM

SVM – support vector machine

Given input x_i(a vector) and tag $y_i \in \{-1,1\}$. I want a linear function s.t $f(x_i) = w * x_i + b \geq 1$ if $y_i = 1$, and $w * x_i + b \leq -1$ if $y_i = -1$.





$$y_i[w * x_i + b] \geq 1 \ \forall i$$

$$y_i[w * x_i + b] \geq 1 - \chi_i \ \forall i, \chi_i >= 0$$

$$Loss = \frac{1}{2}\left||W|\right|^2 + C * \sum_i \chi_i$$

One class SVM.

$$Min_{X0}, r\left(r^2 + C * \sum_i \chi_i\right), s.t. \left||x_i - Xo|\right| < r + \chi_i \ \forall i, \chi_i \geq 0$$