Anomaly/Outlier Detection

Dr. Josif Grabocka

ISMLL, University of Hildesheim

Business Analytics

Anomaly/Outlier Detection

- Detect patterns that do not follow the "usual" trend
- Items that have a low probability of occurrence
- The problem is relevant in particular for predictive maintenance, i.e. fault detection

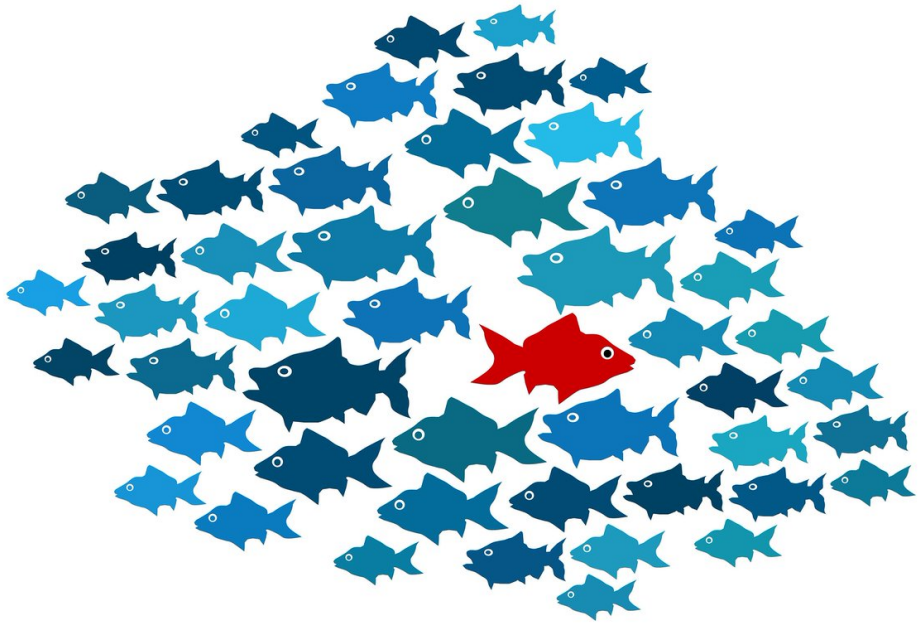


Illustration: Sergio Santoyo

Problem Definition

- **Given** a set of N instances $X \in \mathbb{R}^{N \times M}$
  - i.e. $X_1, \ldots, X_N$, where $\forall n = 1, \ldots, N$ each instance $X_n \in \mathbb{R}^M$
  - such that X includes only **normal** data,
- **Find** if a new test instance x is **unusual** w.r.t. X?
- Examples:
  - Find unusual user behavior?
  - Find unusually behaving production components?

Statistical Test Approach

- Model the probability of the occurrence of an instance $x \in \mathbb{R}^M$ as:
  - $p(x) : \mathbb{R}^M \rightarrow [0, 1]$
- Judge:
  - **If** $p(x) < \epsilon$ then **anomaly**
  - **If** $p(x) \geq \epsilon$ then **normal**
  - for a small probability threshold $\epsilon \in \mathbb{R}$
- Use collected normal instances to compute $p(x)$

Example: Network Intrusion Detection

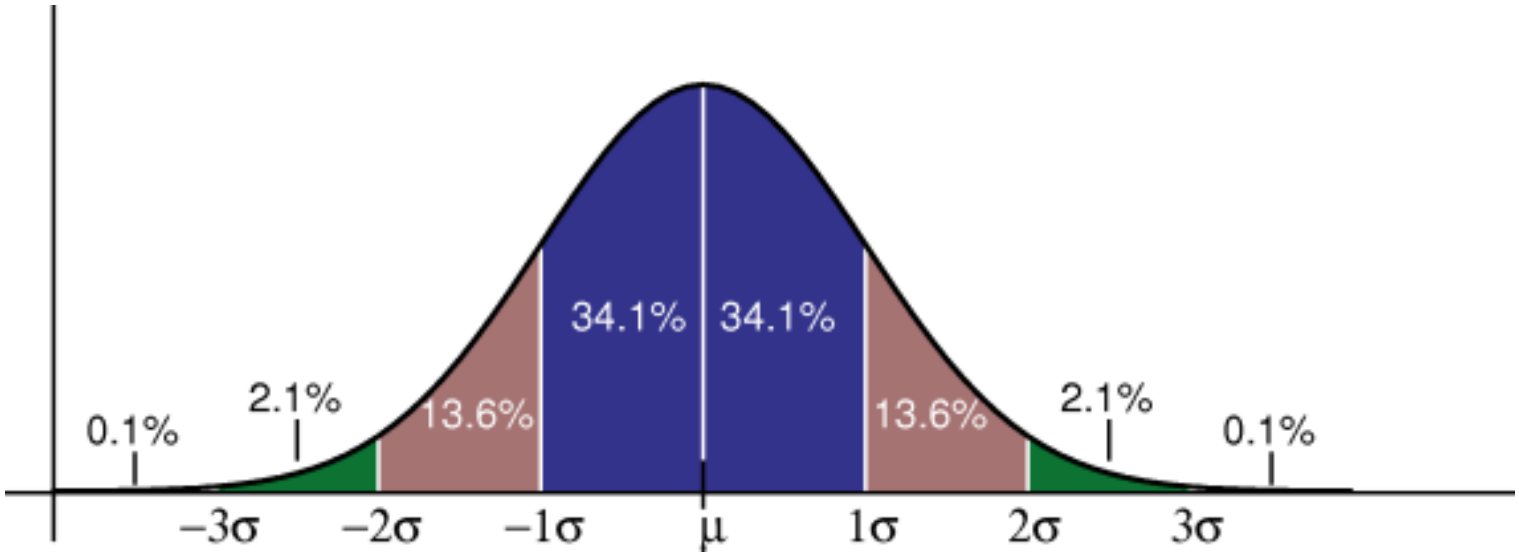
- $X \in \mathbb{R}^{N \times M}$, N intrusions, M features per intrusion

Attribute	Capture location tap1	Capture location tap2
Duration of capture	3,918 s	4,561 s
No. of packets	3,168,660	2,838,227
Avg. size of packets	135.10 bytes	134.23 bytes
Avg. data rate	109,257.90 bytes/s	83,525.07 bytes/s
Capture size	478,792,912 bytes	426,382,869 bytes
Avg. packet rate	808.70 packets/s	622.26 packets/s
No. of connections	613	930
No. of TCP connections	8	174
No. of UDP flows	598	713
No. of ICMP flow	0	36
Portion of multicast packets	92.38%	96.97%

Mantere et al. 2015

Remember: Gaussian Distribution

$$p(x;\mu,\sigma^2) = \mathcal{N}(x;\mu,\sigma^2) = \frac{1}{\sqrt{2\pi}\sigma} e^{\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)}$$



$p(x;\mu,\sigma^2)$

Density estimation

- Assume each variable $1, \ldots, M$ of an instance $x \in \mathbb{R}^M$ follows a normal distribution:
  - $x_1 \sim \mathcal{N}(x; \mu_1, \sigma_1^2)$, $\ldots$, $x_M \sim \mathcal{N}(x; \mu_M, \sigma_M^2)$

- Then the probability of an instance x across all features is:

$$p(x) \quad = \quad p(x_1; \mu_1, \sigma_1^2) p(x_2; \mu_2, \sigma_M^2) \ldots p(x_M; \mu_M, \sigma_M^2)$$

- Note the independence assumption across features

- Or more compactly:

$$p(x) \quad = \quad \prod_{m=1}^{M} p(x_m; \mu_m, \sigma_M^2)$$

Anomaly Detection Algorithm

1. Choose features $m \in \{1, \ldots, M\}$, which you think are anomalous

2. Fit parameters $\mu_1, \ldots, \mu_M$ and $\sigma_1, \ldots, \sigma_M$:

$$\mu_m = \frac{1}{N} \sum_{n=1}^{N} X_{n,m}$$

$$\sigma_m^2 = \frac{1}{N} \sum_{n=1}^{N} (X_{n,m} - \mu_m)^2$$

3. Given a **new** example x, compute $p(x)$:

$$p(x) = \prod_{m=1}^{M} p(x_m; \mu_m, \sigma_M^2) = \prod_{m=1}^{M} \frac{1}{\sqrt{2\pi}\sigma_m} e^{\left(-\frac{(x_m - \mu_m)^2}{2\sigma_m^2}\right)}$$

4. If $p(x) < \epsilon$ then **anomaly**

Evaluation and Parameter Tuning (1)

- Need a set of ground-truth annotated anomalies
  - N^{Train} **normal** $X^{\text{Train}} \in \mathbb{R}^{N^{\text{Train}} \times M}$ with anomaly values $Y^{\text{Train}} \in 0^{N^{\text{Train}}}$
  - N^{Test} **anomaly** $X^{\text{Test}} \in \mathbb{R}^{N^{\text{Test}} \times M}$ with anomaly values $Y^{\text{Test}} \in 1^{N^{\text{Test}}}$
- Further divide the anomalies into a cross-validation and **new** test:
  - N^{CV} **anomaly** $X^{\text{Test}} \in \mathbb{R}^{N^{\text{CV}} \times M}$ with anomaly values $Y^{\text{CV}} \in 1^{N^{\text{CV}}}$
  - N^{Test} **anomaly** $X^{\text{Test}} \in \mathbb{R}^{N^{\text{Test}} \times M}$ with anomaly values $Y^{\text{Test}} \in 1^{N^{\text{Test}}}$
- Example: 10000 normal and 20 anomaly instances
  - Training: 6000 normal
  - Validation: 2000 normal, 10 anomalies
  - Test: 2000 normal, 10 anomalies

Evaluation and Parameter Tuning (2)

- Fit $p(x)$ using $X^{\text{Train}} \in \mathbb{R}^{N^{\text{Train}} \times M}$
- Find features $m \in \{1, \ldots, M\}$ and threshold $\epsilon \in \mathbb{R}$ that
  - maximize **detection quality** on the cross-validation set
- Test the **detection quality** on the test set
- Detection quality expressed in terms of:
  - True positive, False positive, True Negative, False Negative
  - Precision/Recall
  - F1 measure

Ground truth and supervised learning

- Can we use the ground truth labels to learn a prediction model?
- Yes, but:
  - Very few positive examples
  - Large negative examples
  - Various types of anomalies which might not be present in the limited labeled set
  - I.e. high degree of intra-class variation
- Anomaly detection largely treated as unsupervised learning

Multivariate Gaussian Distribution

- Do not model $p(x_1), \ldots, p(x_m)$ separately
- Instead model $p(x)$ jointly

- Parameters: Mean $\mu \in \mathbb{R}^M$, Covariance matrix $\Sigma \in \mathbb{R}^{M \times M}$

$$p(x, \mu, \Sigma) = \frac{1}{(2\pi)^{\frac{M}{2}} |\Sigma|^{\frac{1}{2}}} e^{\left(-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)\right)}$$

- where $|\Sigma|$ is the determinant of Σ

Anomaly Detection Algorithm (2)

1. Fit parameters μ and σ^2:

$$\mu = \frac{1}{N}\sum_{n=1}^{N} X_{n,:}$$

$$\Sigma = \frac{1}{N}\sum_{n=1}^{N} (X_{n,:} - \mu)(X_{n,:} - \mu)^T$$

2. Given a **new** example x, compute $p(x)$:

$$p(x, \mu, \Sigma) = \frac{1}{(2\pi)^{\frac{M}{2}} |\Sigma|^{\frac{1}{2}}} e^{\left(-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)\right)}$$

3. If $p(x) < \epsilon$ then **anomaly**

Comparison

- Independent model

$$p(x) = \prod_{m=1}^{M} \frac{1}{\sqrt{2\pi}\sigma_m} e^{\left(-\frac{(x_m-\mu_m)^2}{2\sigma_m^2}\right)}$$

  - Computationally cheap

- Multivariate Gaussian

$$p(x,\mu,\Sigma) = \frac{1}{(2\pi)^{\frac{M}{2}}\,|\Sigma|^{\frac{1}{2}}} e^{\left(-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right)}$$

  - Captures correlations between features

- Identical if $\Sigma = \text{diag}(\sigma_1^2, \ldots, \sigma_M^2)$

Reconstruction-based Anomaly Detection

- Learn a parametric model $f(X, \theta) \in \mathbb{R}^{N \times M}$ that reconstructs the normal instances $X \in \mathbb{R}^{N \times M}$
- Optimize (considering the sum-of-squared error):

$$\underset{\theta}{\operatorname{argmin}} \sum_{n=1}^{N} \sum_{m=1}^{M} (X_{n,m} - f(X_{n,:}, \theta)_m)^2$$

- A new test instance $x \in \mathbb{R}^M$ is flagged as anomaly if:

$$\left(\sum_{m=1}^{M} (x_m - f(x, \theta)_m)^2 \right) > \epsilon$$

Reconstruction-based Anomaly Detection

- Learn a parametric model $f(X, \theta) \in \mathbb{R}^{N \times M}$ that reconstructs the normal instances $X \in \mathbb{R}^{N \times M}$
- Optimize (considering the sum-of-squared error):

$$\underset{\theta}{\operatorname{argmin}} \sum_{n=1}^{N} \sum_{m=1}^{M} (X_{n,m} - f(X_{n,:}, \theta)_m)^2$$

- A new test instance $x \in \mathbb{R}^M$ is flagged as anomaly if:

$$\left(\sum_{m=1}^{M} (x_m - f(x, \theta)_m)^2 \right) > \epsilon$$

Reconstruction Model $f(x)$

- Neural network of L layers

$$
\begin{aligned}
a_{1,i} &= W_{1,m,0} + \sum_{m=1}^{M} W_{1,m,i}\, x_m, & h_{1,i} &= \tanh(a_{1,i}), \quad i = 1, \ldots, M^{(1)} \\
a_{2,i} &= W_{2,m,0} + \sum_{m=1}^{M^{(1)}} W_{2,m,i}\, h_{1,m}, & h_{2,i} &= \tanh(a_{2,i}), \quad i = 1, \ldots, M^{(2)} \\
a_{\ell,i} &= W_{\ell,m,0} + \sum_{m=1}^{M^{(\ell-1)}} W_{\ell,m,i}\, h_{\ell-1,m}, & h_{\ell,i} &= \tanh(a_{\ell,i}), \quad i = 1, \ldots, M^{(\ell)} \\
a_{L,i} &= W_{L,m,0} + \sum_{m=1}^{M^{L-1}} W_{\ell,m,i}\, h_{L-1,m}, & h_{L,i} &= a_{\ell,i}, \quad i = 1, \ldots, M
\end{aligned}
$$

- Reconstructed output: $\hat{x} = f(x) = h_L$
- Typically $M \geq M^{(1)} \geq \cdots \geq M^{(\frac{L}{2})} \leq M^{(\frac{L}{2}+1)} \leq \ldots M$

Optimizing parameters W

- We can minimize the loss

$$\mathcal{L} = \sum_{m=1}^{M} (x_m - f(x, \theta)_m)^2$$

- Via first-order optimization techniques:

$$W_{\ell,m,i} \leftarrow W_{\ell,m,i} - \eta \frac{\partial \mathcal{L}}{\partial W_{\ell,m,i}}$$

- But we do not yet know the gradients:

$$\frac{\partial \mathcal{L}}{\partial W_{\ell,m,i}} = ?$$

Chain-rule of Calculus

- Suppose $y = g(x)$ and $z = f(g(x)) = f(y)$, then

$$\frac{dz}{dx} = \frac{dz}{dy}\frac{dy}{dx}$$

- In the vector case, suppose $x \in \mathbb{R}^m$, $y \in \mathbb{R}^n$, and $y = g(x), g : \mathbb{R}^m \to \mathbb{R}^n$ together with $z = f(y), f : \mathbb{R}^n \to \mathbb{R}$:

$$\frac{\partial z}{\partial x_i} = \sum_{j=1}^{n} \frac{\partial z}{\partial y_j}\frac{\partial y_j}{\partial x_i}$$

- Compactly written using the Jacobian matrix $\frac{\partial y}{\partial x} \in \mathbb{R}^{n \times m}$ as

$$\nabla_x z = \left(\frac{\partial y}{\partial x}\right)^T \nabla_y z$$

Back-propagation

1. Easily derive $\frac{\partial \mathcal{L}}{\partial W_{L,m,i}}$ for the last layer
2. Iterate backwards $l = L-1, \ldots, 1$

$$\frac{\partial \mathcal{L}}{\partial h_{l,i}} = \sum_{j=1}^{M^{l+1}} \frac{\partial \mathcal{L}}{\partial h_{l+1,j}} \frac{\partial h_{l+1,j}}{\partial h_{l,i}}$$

- Then:

$$\begin{aligned} \frac{\partial h_{l+1,j}}{\partial h_{l,i}} &= \frac{\partial h_{l+1,j}}{\partial a_{l+1,j}} \frac{\partial a_{l+1,j}}{\partial h_{l,i}} \\ \frac{\partial \mathcal{L}}{\partial W_{l,m,i}} &= \frac{\partial \mathcal{L}}{\partial h_{l,i}} \frac{\partial h_{l,i}}{\partial a_{l,i}} \frac{\partial a_{l,i}}{\partial W_{l,m,i}} \end{aligned}$$

- Store $\frac{\partial \mathcal{L}}{\partial h_{l,i}}$ for the next layer $l-1$