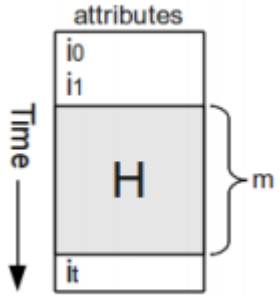
Anomaly Detection

Seminář z umělé intelligence II

Šimon Rozsíval

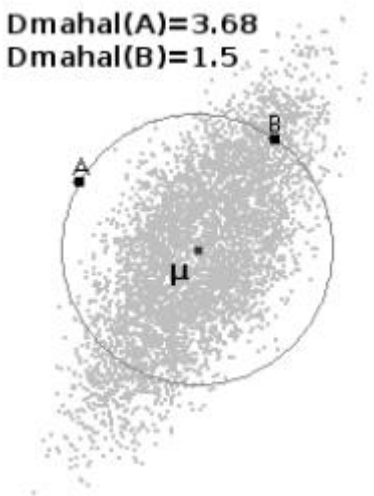
Online Anomaly Detection

- Attributes from sensors sampled every t milliseconds
  - $\vec{i}_t = \{i_{t,1}, i_{t,2}, \ldots, i_{t,n}\}$ … input vector
  - $i_{t,j} \in \mathbb{R}$ … value of attribute a_j at time t
- **Task:** Instantly decide if new sample $\vec{\mathrm{i}}_\mathrm{t}$ is an anomaly.
- Past data stored in matrix H … *a sliding window*
  - m rows – the last m sensor readings
  - n columns – the attributes (individual sensor readings)
  - Every time a new input $\vec{i}_t$ is received, the oldest reading is forgotten (the first row is removed) and the new one is added as the last row of the matrix.
  - The data in H is always assumed to be *nominal*.



Mahalanobis Distance

- Measure distance between point $\overrightarrow{i_t}$ and distribution H
- Distance in the units of standard deviations.
  - "How many standard deviations away $\overrightarrow{i_t}$ is from the mean of H"



$$D_M(\overrightarrow{i_t}, H) = \sqrt{(\overrightarrow{i_t} - \vec{\mu})\Sigma^{-1}(\overrightarrow{i_t}^T - \vec{\mu}^T)}$$

$\overrightarrow{i_t}$ … vector of current input

Σ … covariance matrix of H

$\vec{\mu}$ … means of all the dimensions of H

Three Types of Anomalies

1. Point anomalies
   - *illegal data instances*, corresponding to illegal values in $\overrightarrow{i_t}$
   - i.e., a malfunctioning sensor
2. Contextual anomalies
   - that is, data instances that are only *illegal with respect to specific context* but not otherwise
   - i.e., sudden change of acceleration – drone hits an obstacle
3. Collective anomalies
   - which are related data instances that are *legal apart, but illegal when they occur together*
   - i.e., a malfunctioning sensor

- Anomaly of any type can cause the representative point to be apart from the nominal points ⇒ outside of the dense area ⇒ *large* D_M.

Online Training

- Using the Mahalanobis Distance as an anomaly detector is prone to errors without guidance
  - the success depends on whether the dimensions are *correlated or not*
  - dimensions are not correlated ⇒ more probable a nominal point will differ from the observed nominal points in those dimensions, exactly as in contextual anomaly ⇒ large Mahalanobis Distance ⇒ false alarms.
- Find and group correlated attributes
  - Mahalanobis Distance can be applied per each correlated set of attributes afterwards
  - Difficult task

Finding Correlated Attributes

- *Online_Trainer(H)*
  - returns n sets of dynamically correlated attributes (**Alg. 1**) … $CS = \{CS_1, CS_2, \dots, CS_n\}$
  - … and a threshold per each set $TS = \{ts_1, ts_2, \dots, ts_n\}$
  - ts_a … the highest D_M of points with dimensions relating the attributes in CS_a extracted from H
  - Since every point in H is considered nominal, then any higher D_M indicates an anomaly.

- *CS, TS* … sets of *correlated attributes* and their *thresholds*.

Algorithm 1 Correlation_Detector(H)

```
for each a_i ∈ A do
    CS_i ← φ
    for each a_j ∈ A do
        if |ρ_{i,j}(H_i^T, H_j^T)| > ct then
            add a_j to CS_i
    add CS_i to CS
return CS
```

$ct \in \{0..1\}$.

- ρ … *Pearson* correlation coefficient

$$\rho = \frac{\sum_i (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_i (x_i - \bar{x})^2 \sum_i (y_i - \bar{y})^2}}$$

Specializing Anomaly Detection for Robots

- Data *obtained from sensors* that are used in the control loop *to affect the environment*
  - C*hanges in the environment* – a function of the *actions of the agent*.
  - Therefore, it makes sense to monitor the ***change in the values*** measured by the sensors (which originates from the robot's actions), rather than *the absolute values*.

$$\Delta\vec{i}_t = \vec{i}_t - \vec{i}_{t-1}$$

  - Smoothing filter:

$$Z(x, \vec{x}) = \frac{x - \bar{x}}{\sigma_x}$$

$$Z_{raw}(\vec{i}_t) = \left\{Z\left(i_{t,1}, H_1^T\right), \ldots, Z\left(i_{t,n}, H_n^T\right)\right\}$$

$$Z_\Delta(\vec{i}_t) = Z_{raw}(\Delta\vec{i}_t)$$

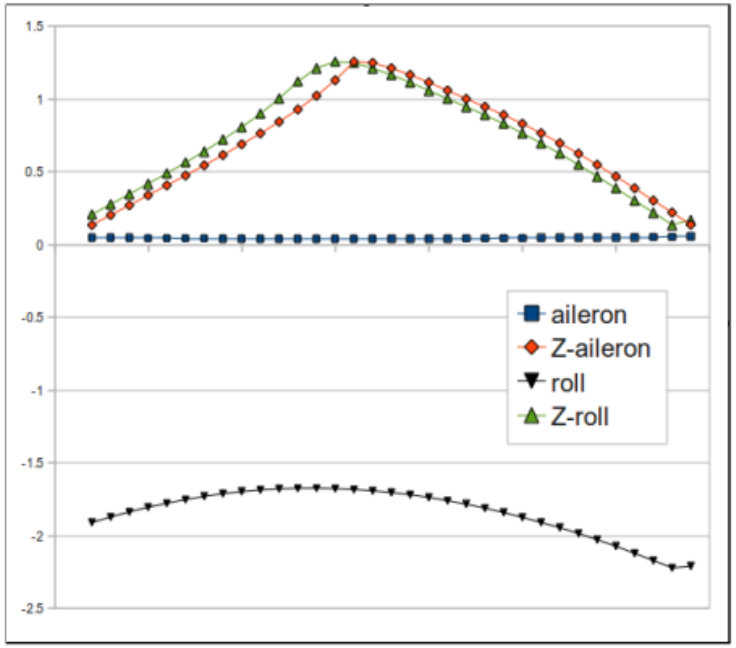


Figure 4: Illustration of the Z-transformation.

- Aileron data – nearly constant.
- We say that the *aileron* and *roll* attributes are correlated if they share the same effect of change.

The Anomaly Detector

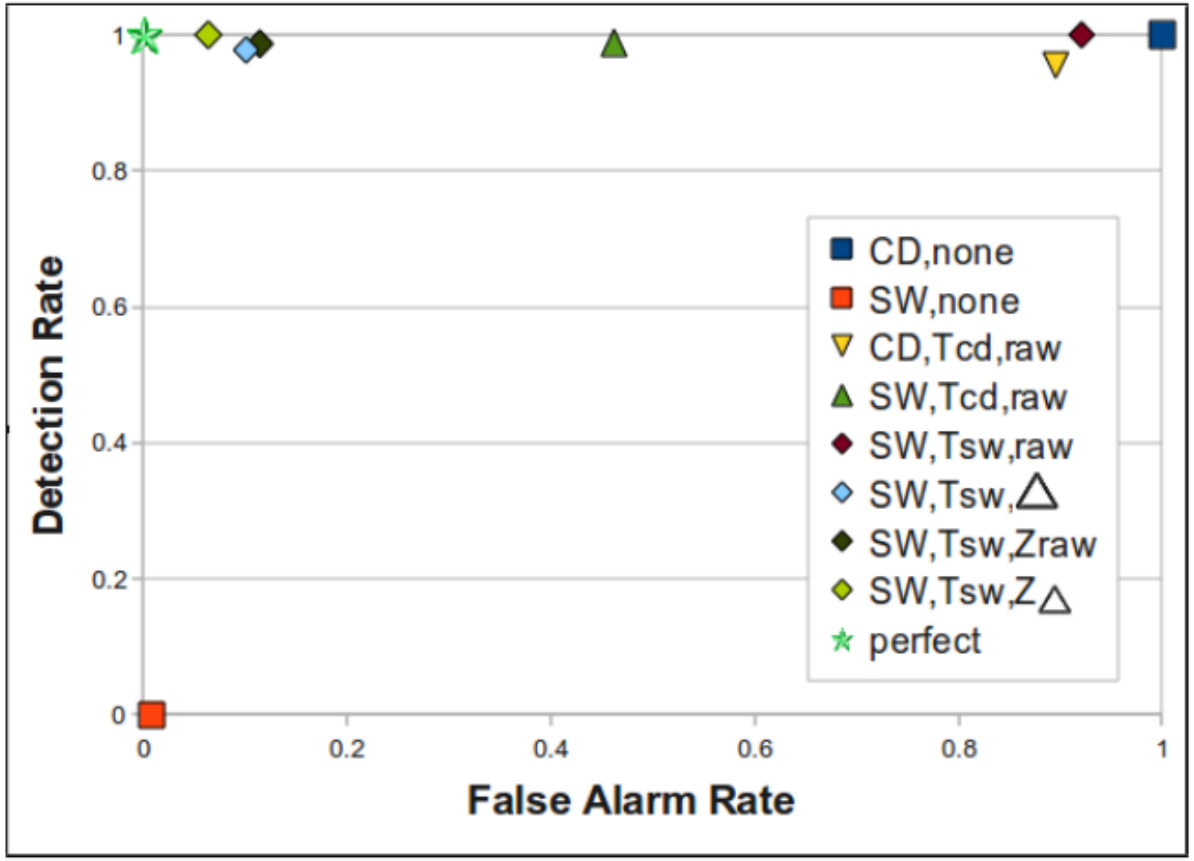
Algorithm 2 Anomaly_Detector($\vec{i_t}$)

$\vec{i_t} \leftarrow Z_{\triangle}(\vec{i_t})$
$H \leftarrow \{\vec{i}_{t-m-1}, \ldots, \vec{i}_{t-1}\}$
$CS, TS \leftarrow$ Online_Trainer(H)
for each a $(0 \leq a \leq |CS|)$ **do**
  Let CS_a be the a'th set of correlated attributes in CS
  Let $threshold_a$ be the a'th threshold, associated with CS_a
  $P_H \leftarrow$ points with dimensions relating to CS_a's attributes extracted from H
  $p_{new} \leftarrow$ point with dimensions relating to CS_a's attributes extracted from $\vec{i_t}$
  if $threshold_a < D_{mahal}(p_{new}, P_H)$ **then**
    declare "Anomaly".

The Goal

- Maximize *detections*
- Minimize *false alarms*

- *Method proposed by the authors of the paper:*
  - *(SW, Tsw, Z_{Δ})*
    - **online training** on the **sliding window** *with* ***Z-filter***
    - Results of the authors:
      - *Detection rate … 1*
      - *False alarms rate … 0.064*



The Project

- *(SW, Tsw, Z_Δ)*
  - Unsupervised method
  - The *ct* parameter must be chosen carefully
  - Worked on applications on UAVs
- I will try to implement this method and test it on our data.
  - It will be interesting to see the comparison with the *classifier* used as anomaly detector.

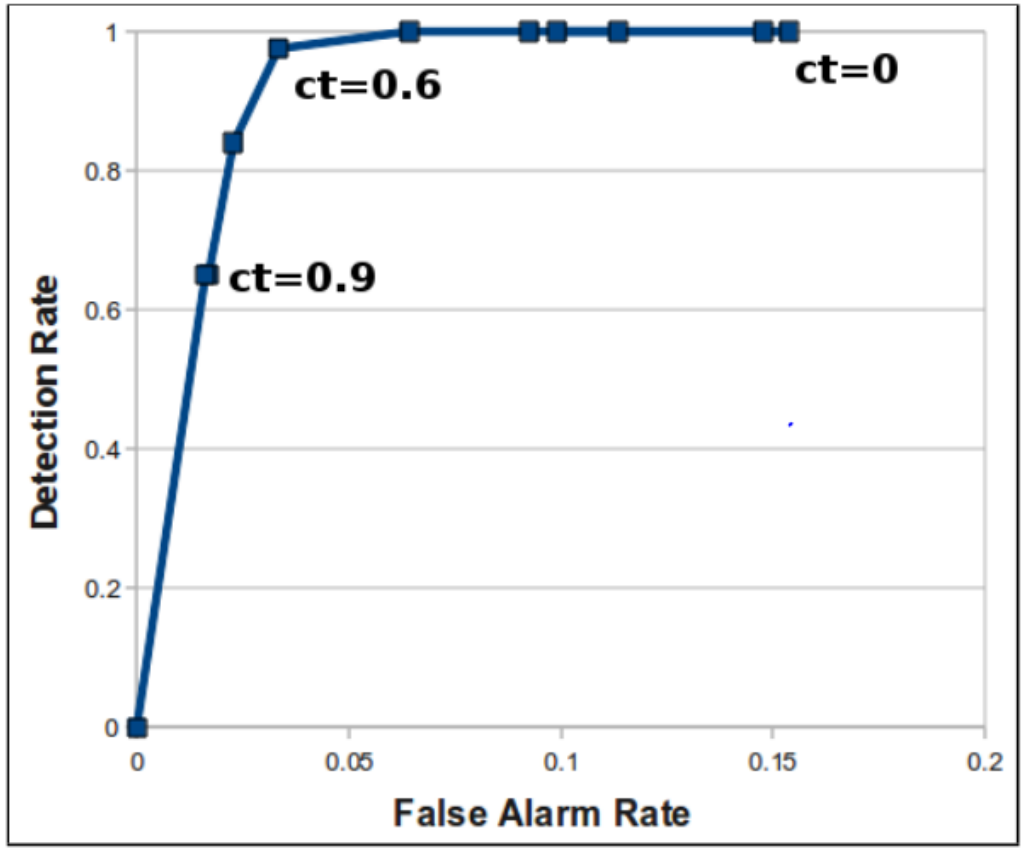


Figure 10: The influence of the correlation threshold.

Data Format

- Input:
  - raw stream of the data from the sensors of the drone
- Output:
  - [0, 1] … likelihood of an anomaly
  - 0/1 … “an anomaly is detected”
    - This detector cannot identify the *type* of anomaly/what happened to the drone

Sources

- Machine learning course by Andrew Ng, Coursera
- Online Anomaly Detection in Unmanned Vehicles, Eliahu Khalastchi, Gal A. Kaminka, Meir Kalech and Raz Lin, Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011), Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 115-122