Predicting sector configuration transitions with autoencoder-based anomaly detection

ICRAT 2018, Barcelona, 28/06/18

Thomas Dubot
thomas.dubot@onera.fr
Presentation outline

- **The airspace configuration problem**
  - Transition cost between sector configurations
  - Limits of the current methods

- **The autoencoder-based anomaly detection**
  - Using the reconstruction error of autoencoders
  - Training the model with ACC historical data

- **First results**
  - Test set and transitions never realized
  - Experimental evaluation

- **Using the anomaly detection model**
  - Filtering outliers
  - Data mining
  - Sector configuration path & quasimetric
Sectorization reminder

1 ACC

38 ES

RL

10 ES
Sector configurations

- Set of sectors for a given time period
- Dynamic process
- Sector configuration plan / opening scheme
- Importance of smooth transitions
Transition cost between sector configurations

- Smooth transitions between sector configurations

Two approaches:
- Keep sectors from the last configuration
- Distance function
Limits of the current methods

- Fluidity versus exploration possibilities

- Distance functions insufficient to classify transitions
Limits of the current methods

- Multiple feedbacks with operational experts to fine-tune logical rules
  - Costly
  - Some rules are never expressed

- The Machine Learning approach
  - Directly learn from historical data, i.e. transitions realized in the past
  - Classification and regression methods
  - Unsupervised anomaly detection method
  - Autoencoders to predict and quantify anomalies

Artificial Neural Networks

- Network of neurons
- Weights $w$ as synaptic strengths
- Activation function to model the firing rate
- Directed weighted graph
- Feedforward neural network
Autoencoders

- Feedforward neural networks
- Input and output layers with same number of nodes
- Encoding and decoding parts through hidden layers
- Reconstruction error
# Analysis of historical data

<table>
<thead>
<tr>
<th></th>
<th>00:00:00</th>
<th>04:19:00</th>
<th>259'</th>
<th>UBDX</th>
<th>FBDX</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>04:19:00</td>
<td>04:22:00</td>
<td>3'</td>
<td>UBDX</td>
<td>FBDX</td>
</tr>
<tr>
<td>3</td>
<td>04:22:00</td>
<td>04:38:00</td>
<td>16'</td>
<td>UNOR</td>
<td>FBDX</td>
</tr>
<tr>
<td>4</td>
<td>04:38:00</td>
<td>04:41:00</td>
<td>3'</td>
<td>PT</td>
<td>RL</td>
</tr>
<tr>
<td>5</td>
<td>04:41:00</td>
<td>04:46:00</td>
<td>5'</td>
<td>PT</td>
<td>RL</td>
</tr>
<tr>
<td>6</td>
<td>04:46:00</td>
<td>04:47:00</td>
<td>1'</td>
<td>PT</td>
<td>RL12</td>
</tr>
<tr>
<td>7</td>
<td>04:47:00</td>
<td>05:00:00</td>
<td>13'</td>
<td>P14</td>
<td>T14</td>
</tr>
<tr>
<td>8</td>
<td>05:00:00</td>
<td>05:38:00</td>
<td>38'</td>
<td>P14</td>
<td>T14</td>
</tr>
<tr>
<td>9</td>
<td>05:38:00</td>
<td>09:24:00</td>
<td>226'</td>
<td>P14</td>
<td>T14</td>
</tr>
<tr>
<td>10</td>
<td>09:24:00</td>
<td>10:20:00</td>
<td>56'</td>
<td>P14</td>
<td>T14</td>
</tr>
<tr>
<td>9</td>
<td>10:20:00</td>
<td>11:01:00</td>
<td>41'</td>
<td>P14</td>
<td>T14</td>
</tr>
<tr>
<td>10</td>
<td>11:01:00</td>
<td>11:27:00</td>
<td>26'</td>
<td>P34</td>
<td>T14</td>
</tr>
<tr>
<td>9</td>
<td>11:27:00</td>
<td>12:57:00</td>
<td>90'</td>
<td>P14</td>
<td>T14</td>
</tr>
<tr>
<td>8</td>
<td>12:57:00</td>
<td>13:41:00</td>
<td>44'</td>
<td>P14</td>
<td>T14</td>
</tr>
<tr>
<td>9</td>
<td>13:41:00</td>
<td>14:04:00</td>
<td>23'</td>
<td>P14</td>
<td>T14</td>
</tr>
<tr>
<td>10</td>
<td>14:04:00</td>
<td>14:43:00</td>
<td>39'</td>
<td>P14</td>
<td>T14</td>
</tr>
<tr>
<td>11</td>
<td>14:43:00</td>
<td>14:59:00</td>
<td>16'</td>
<td>P14</td>
<td>T14</td>
</tr>
<tr>
<td>10</td>
<td>14:59:00</td>
<td>15:27:00</td>
<td>28'</td>
<td>P14</td>
<td>T14</td>
</tr>
<tr>
<td>11</td>
<td>15:27:00</td>
<td>16:32:00</td>
<td>65'</td>
<td>P12</td>
<td>T14</td>
</tr>
</tbody>
</table>
Building the input data

\[ x^{(i)} = [9, 10, 0, 0, \ldots, 2, 0] \]
Data samples

Transitions realized in the past three years in a French ACC (Bordeaux)

- Transitions realized in the past:
  - The number of samples $n$ is equal to 2617
  - 2093 training labels $\{x^{(1)}, x^{(2)}, x^{(3)}, \ldots, x^{(m)}\}$ where $x^{(i)} \in \mathbb{R}^d$, $\forall i \in [1, m]$
  - 524 test labels $\{x^{(m+1)}, x^{(m+2)}, \ldots, x^{(n)}\}$ where $x^{(i)} \in [0,1]^d$, $\forall i \in [m + 1, n]$
  - The number of nodes $d$ is equal to 55
  - The number of occurrences of each transition is not taken into account

- Transitions never realized in the past
  - Between realized configurations
Hyperparameters of the autoencoder

- One hidden layer
- Same number of neurons
- Sigmoid activation function for all layers

\[ y = \sigma(W_1 x + b_1) \quad \hat{x} = \sigma(W_2 y + b_2) \]
Model training

\[(W_1, b_1, W_2, b_2) = \arg\min_{W_1, b_1, W_2, b_2} \frac{1}{m} \sum_{i=1}^{m} \|x^{(i)} - \hat{x}^{(i)}\|^2\]

\[\hat{x}^{(i)} = \sigma(W_2 (\sigma(W_1 x^{(i)} + b_1)) + b_2)\]

\[\sigma(z) = \frac{1}{1 + e^{-z}}\]

- Performance measure: MSE estimator
- Adam optimization algorithm

## Results

<table>
<thead>
<tr>
<th>Set</th>
<th>Evaluation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Min error</td>
</tr>
<tr>
<td>Realized transitions used in training (training set)</td>
<td>7.21e-6</td>
</tr>
<tr>
<td>Realized transitions not used in training (test set)</td>
<td>7.17e-6</td>
</tr>
<tr>
<td>Transitions never realized</td>
<td>7.13e-6</td>
</tr>
</tbody>
</table>
Results

Reconstruction error of realized transitions

Reconstruction error vs Occurrences of transition
Experimental evaluation of “perfect” transitions

Model prediction

- Popular realized transitions
- Probable transitions never realized
- Improbable accepted transitions
- Improbable filtered transitions

Expert perception

- Transition already realized
- Transition realistic
- Nothing wrong but probably never realized
- Never realized
Using the anomaly detection model

- Filtering outliers
- Example of threshold: train max (0.0009)
Using the anomaly detection model

- Data mining

![Graph showing reconstruction error vs. subdimension index]
Using the anomaly detection model

- Probabilistic graph using the reconstruction error
- Probabilistic path with a shortest path algorithm
- Number of intermediate steps = quasimetric
Work summary

- Set of features extracted from sector configuration data
- A simple autoencoder is defined with one hidden layer with the same number of nodes
- The autoencoder is trained with 2093 instances
- The reconstruction error is used to quantify the degree of abnormality of a new transition
- This score can be used to filter some transitions/configurations, for data mining, or to build a probabilistic path between two configurations
Computational consideration

- **Libraries Python**
  - TensorFlow: Google open source machine learning framework
  - NetworkX: creation, manipulation, and study of the structure, dynamics, and functions of complex networks

- **Autoencoder based anomaly detection with TensorFlow:**
  [https://github.com/thomasdubdub/autoencoder-anomaly-detection](https://github.com/thomasdubdub/autoencoder-anomaly-detection)

<table>
<thead>
<tr>
<th></th>
<th>Code complexity</th>
<th>Computational cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reconstruction error</td>
<td>++</td>
<td>+</td>
</tr>
<tr>
<td>Probabilistic path</td>
<td>+</td>
<td>+++</td>
</tr>
</tbody>
</table>
Limits and perspectives

- Feedback from only one operational expert
- Good results on one specific ACC
- Generalization?
- New model
  - Hyperparameters fine-tuning
  - New autoencoder type
- Filtering data
- Adding new features