EP2420 *Intrusion Detection - Task 2*

Federico Giarrè

December 8, 2023

1 Task 2

This task consists of creating a CategoricalHMM, and use such model to decode an observation sequence and return the most probable sequence of hidden states that can generate such observations.

1.1 Observation Mapping

Since the number of unique observation symbols in the dataset used is **6672**, we need to map the symbols to a smaller space. We are going to perform this operation by employing the K-means method over different K (the number of resulting symbols) and keep the one with the highest silhouette score [1]. Given K being the number of clusters tested, Table 1 reports the silhouette score related to different Ks.

K	Silhouette-Score
6	6.22e-01
7	6.04e-01
8	5.95e-01
9	5.81e-01
10	5.63e-01
11	5.48e-01
12	5.43e-01

Table 1: Silhouette-Score of different K values

As a result, grouping the observations into six clusters is the best choice. The centers of these clusters are **(7.54e+03, 1.27e+04, 4.43e+03, 1.52e+04, 9.89e+03, 2.26e+03)**

1.2 Supervised Learning

Performing Supervised Learning in HMMs is trivial (as described in the previous section) once the values of the hidden states and observations are available. We can compute these probabilities matrixes and obtain the following tables, respectively: Table 2 for the learned transition table ($\hat{A}$), Table 3 for the learned emission table ($\hat{B}$), and Table 4 for the learned starting probability table ($\hat{\pi}$).

[1] Metric to assess the goodness of the resulting clustering. Ranges from [-1,1] where a higher value means better clustering

States	0	1	2	3	4	5
0	6.25e-01	3.75e-01	0	0	0	0
1	0	0	0	4.59e-01	0	5.41e-01
2	0	1	0	0	0	0
3	0	0	0	0	1	0
4	0	0	1	0	0	0
5	0	0	0	0	1	0

Table 2: Transition Matrix ($\hat{A}$)

State/Observation	0	1	2	3	4	5
0	0	0	2.84e-02	0	0	9.72e-01
1	2.85e-02	2.66e-02	4.68e-03	2.92e-01	1.50e-01	6.24e-04
2	1.12e-01	0	1.56e-01	9.32e-04	0	8.31e-01
3	6.00e-03	0	1.12e-01	0	0	8.82e-01
4	0	0	2.31e-02	0	0	9.76e-01
5	1.24e-02	0	2.29e-01	0	0	7.59e-01

Table 3: Emission Matrix ($\hat{B}$)

Hidden State	0	1	2	3	4	5
Probability	1	0	0	0	0	0

Table 4: Starting Probability for hidden states ($\hat{\pi}$)

1.3 Decoding

After splitting the dataset using 70% of the attacks and relative observations for training (1400 samples) and the remaining for testing (600 samples), the computed probability matrixes are given to the CategoricalHMM model. Once given the matrixes, the model can decode the sequence of hidden states that most likely produced the sequence of observations given to it. The Viterbi algorithm was run on the test set and, to evaluate the goodness of the prediction, two accuracy metrics were designed: ACC_{start} and ACC_{action}. ACC_{start} is defined as the correct prediction of the start of the attack hence, in the case of this specific dataset, when the first Ping Scan is issued by the attacker. It is calculated as in Equation 1:

$$ACC_{start} = \frac{1}{l}\sum_{i=1}^{l} \mathbb{1}\{t^i_{start} == \hat{t}^i_{start}\} \tag{1}$$

Where i is the index of the tested samples and t_{start} is the timestamp of the first action different from "Continue". In the same fashion, ACC_{action} is the accuracy of the model when predicting which action occurred at every timestep. ACC_{action} can be formalized as in Equation 2:

$$ACC_{action} = \frac{1}{lT}\sum_{i=1}^{l}\sum_{t=1}^{T} \mathbb{1}\{s^i_t == \hat{s}^i_t\} \tag{2}$$

Where s is the hidden state, t is the timestamp, and i the sample index. After computing the two metrics for the trained model, the ACC_{start} scored **2.3e-01** and the ACC_{action} scored **3.75e-01**.