Adaptive Law-Based Transformation (ALT)

Wigner Summer Camp
Data and Compute Intensive Sciences Research Group

Balázs, Paszkál, Vince, Levente,
Antal
Éva, Hajni

7-11 July 2025

Table of Contents

Dataset

Dataset

Consider the following dataset:

- Train instance from class **a**: $a = [1, 1, 2, 3, 5]$.
- Train instance from class **b**: $b = [2, 1, 5, 7, 17]$.
- Test instance from an unknown class: $x = [2, 3, 5, 8, 13]$.

We now apply ALT to classify the test instance, i.e., determine to which class x belongs.

The detailed calculation

Step 1: Extracting the Laws

We define parameters $r = 3,\ l = 2,\ k = 1$. (Their meaning will be discussed later.)

From $a = [1, 1, 2, 3, 5]$ we extract the following segments of length $r = 3$:

$$[1, 1, 2],\ [1, 2, 3],\ [2, 3, 5]. \tag{1}$$

Then we embed them to $l \times l = 2 \times 2$ matrices:

$$\begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix}, \quad \begin{bmatrix} 1 & 2 \\ 2 & 3 \end{bmatrix}, \quad \begin{bmatrix} 2 & 3 \\ 3 & 5 \end{bmatrix}. \tag{2}$$

From those, the laws are the eigenvectors corresponding to the smallest absolute eigenvalues:

$$\begin{bmatrix} -0.8507 \\ 0.5257 \end{bmatrix}, \quad \begin{bmatrix} -0.8507 \\ 0.5257 \end{bmatrix}, \quad \begin{bmatrix} -0.8507 \\ 0.5257 \end{bmatrix}. \tag{3}$$

Step 1: Extracting the Laws (continued)

We combine the results into one matrix and repeat the process for all training instances (separately for each class).
From class **a**, we obtain laws:

$$L_a = \begin{bmatrix} -0.8507 & -0.8507 & -0.8507 \\ 0.5257 & 0.5257 & 0.5257 \end{bmatrix}. \tag{4}$$

From class **b**, we obtain laws:

$$L_b = \begin{bmatrix} -0.9571 & -0.8702 & -0.9085 \\ 0.2898 & 0.4927 & 0.4179 \end{bmatrix}. \tag{5}$$

Step 2: Embedding the Test Instance

The test instance $x = [2, 3, 5, 8, 13]$ is embedded as:

$$E = \begin{bmatrix} 2 & 3 \\ 3 & 5 \\ 5 & 8 \\ 8 & 13 \end{bmatrix}. \tag{6}$$

Step 3: Projection by Laws

Multiplying with class **a** laws:

$$F_a = EL_a = \begin{bmatrix} -0.1241 & -0.1241 & -0.1241 \\ 0.0767 & 0.0767 & 0.0767 \\ -0.0474 & -0.0474 & -0.0474 \\ 0.0293 & 0.0293 & 0.0293 \end{bmatrix}. \tag{7}$$

Multiplying with class **b** laws:

$$F_b = EL_b = \begin{bmatrix} -1.0448 & -0.2623 & -0.5635 \\ -1.4224 & -0.1471 & -0.6363 \\ -2.4672 & -0.4094 & -1.1997 \\ -3.8895 & -0.5565 & -1.8360 \end{bmatrix}. \tag{8}$$

Step 4: Feature Calculation

Using the mean of all squared values:

- Class **a**: mean $= 0.0061$.
- Class **b**: mean $= 2.5359$.

Conclusion: The test instance is closer to class **a**.

The Parameters r, l, k

Meaning of Parameters r, l, k

- r: Length of the original sequence window used for feature extraction.
- l: Embedding size, defines a symmetric $l \times l$ matrix.
- $2l - 1$: Number of equally spaced elements sampled from the window of length r.
- k: Step size (stride) used for shifting the window over the time series.

Example r, l, k values

Consider the following instance:
$x' = [2, 1, 5, 7, 17, 31, 65, 127, 257, 511]$.
Let $r = 5, l = 2$, and $k = 1$. Then $2l - 1 = 3$ and

$$x'_1 = [\boxed{\boxed{2}, 1, \boxed{5}, 7, \boxed{17}}, 31, 65, 127, 257, 511].$$

$$x'_2 = [2, \boxed{\boxed{1}, 5, \boxed{7}, 17, \boxed{31}}, 65, 127, 257, 511].$$

$$x'_2 = [2, 1, \boxed{\boxed{5}, 7, \boxed{17}, 31, \boxed{65}}, 127, 257, 511].$$

$$x'_3 = [2, 1, 5, \boxed{\boxed{7}, 17, \boxed{31}, 65, \boxed{127}}, 257, 511].$$

$$x'_4 = [2, 1, 5, 7, \boxed{\boxed{17}, 31, \boxed{65}, 127, \boxed{257}}, 511].$$

$$x'_5 = [2, 1, 5, 7, 17, \boxed{\boxed{31}, 65, \boxed{127}, 257, \boxed{511}}].$$

Example r, l, k values

Consider the following instance:
$x'' = [2, 1, 5, 7, 17, 31, 65, 127, 257, 511]$.
Let $r = 7, l = 2$, and $k = 2$. Then $2l - 1 = 3$ and

$$x''_1 = [\boxed{\boxed{2}, 1, 5, \boxed{7}, 17, 31, \boxed{65}}, 127, 257, 511].$$

$$x''_2 = [2, 1, \boxed{\boxed{5}, 7, 17, \boxed{31}, 65, 127, \boxed{257}}, 511].$$

Summary of ALT Method

- Extract subsequences and embed into symmetric matrices.
- Compute eigenvectors of smallest absolute eigenvalues — these are the laws.
- Multiply embedded test instances with class-specific laws.
- Compute summary statistics (e.g., mean of squared values).
- Classify by choosing the class that minimizes the statistic.