Pattern Recognition

Exercise Session 10

Michael Jungo (michael.jungo@unifr.ch)

File mapping is incorrect, because of the sorting.

```
self.images = sorted(os.listdir(root_dir))
[…]
image_paths = [os.path.join('test', str(i) + '.png') for i in range(len(predictions))]
```
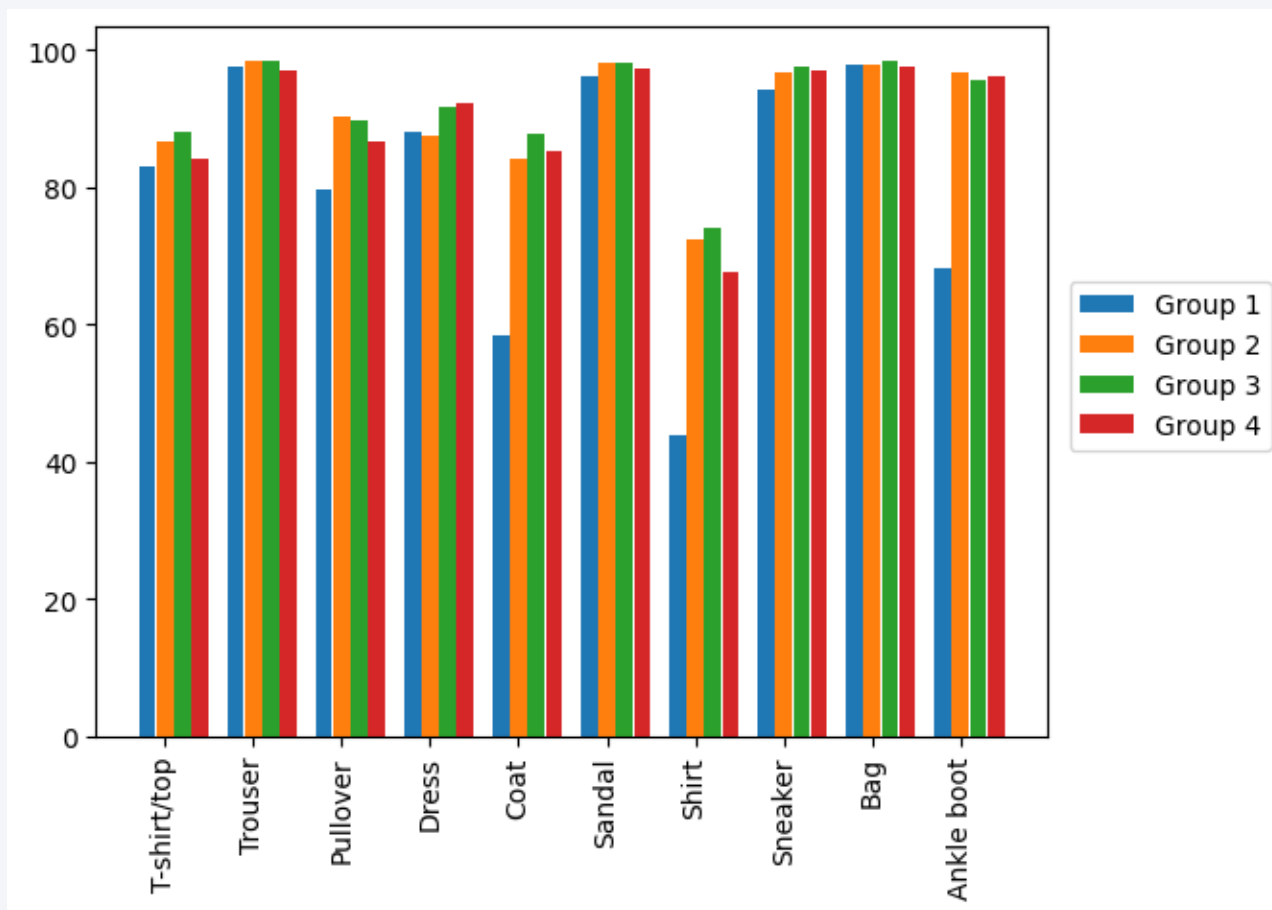
String sorting is lexicographical (not numerical):

```
    [
      …
      '1888.png',
      '1889.png',
      '189.png',
      '1890.png',
      '1891.png',
      …
    ]
```

Competition 1 - FashionMNIST

Accuracy

Group 1	Group 2	Group 3	Group 4
80.70	90.90 (09.81*)	**91.97**	90.09

*incorrect file mapping

Exercise 3 – Remark

Deadline: **Monday, 6th of May 2024 (end of the day)**

You still have one week left.

But the remaining exercises are introduced today.

Last Exercises and Competitions

Deadline: **Monday, 20th of May 2024 (end of the day)**

Exercise 4: Signature Verification
> Can be solved with DTW

Exercise 5: Molecules
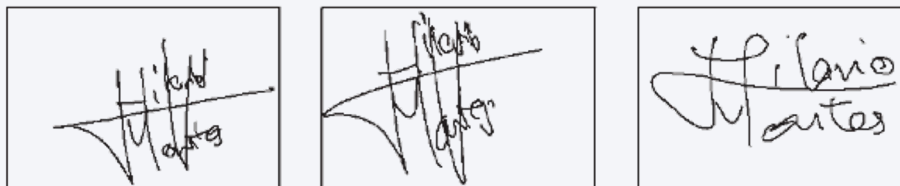> Use approximate Graph Edit Distance

You only need to solve one, but you can solve both!

30 writers (listed in `writers.tsv`)

Enrollment: 5 genuine signatures each

Verification: 45 signatures each (20 genuine, 25 forgeries)
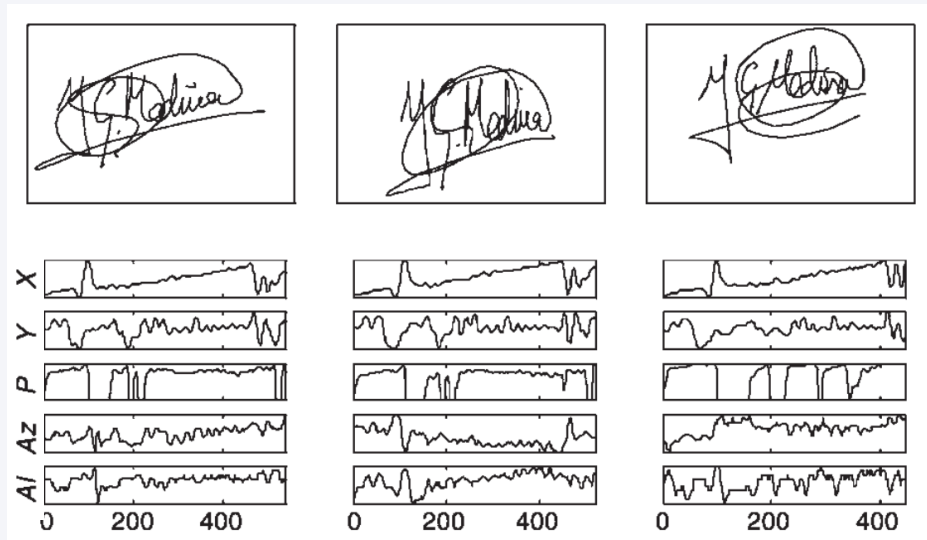Ground truth in `gt.tsv`

Task

Compute dissimilarity for each verification signature with the 5 genuine ones

MCYT Signature – introduced in the paper:

"MCYT baseline corpus: a bimodal biometric database"

See `exercises/4-Signature/mcyt.pdf` on ILIAS → Section 4

Input – TSV files with the following columns:

`t x y pressure penup azimuth inclination`

`penup` 1 if change between pen-up and pen-down

`azimuth / inclination` horizontal / vertical angle of the pen

Recommendation: **DTW**

Features: `x, y, vx, vy, pressure`

`vx, vy` velocity in `x` and `y` with respect to `Δt`

Normalise for each signature individually

Sakoe-Chiba band can be helpful

Evaluation: mean average precision (mAP)

Competition – Signature Verification

On ILIAS: `competitions/3-Signature`

Input:

`enrollment/`	Genuine signatures for each writer
`verification/`	Signatures to verify
`writers.tsv`	List of all writers
`test.tsv`	List of test files to predict (like GT but without the label)

Expected Output:

A TSV file: `test.tsv` (Columns separated by a tab: `\t`)

One line per writer, listing pairs of ID and their dissimilarity.

Only include the validation samples for the same writer!

Order from least dissimilar to most dissimilar (i.e. most similar first)

```
Writer_ID1 signature_ID1 dissimilarity1 signature_ID2 dissimilarity2 ...
Writer_ID2 signature_ID1 dissimilarity1 signature_ID2 dissimilarity2 ...
```
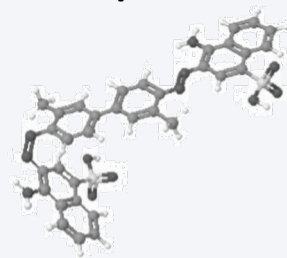
Example:

```
051 46 6.40341144 21 7.62949846 17 9.18516724 03 10.47132116 […]
043 02 0.99152807 22 4.82357323 14 2.14435743 42 5.05044537  […]
[…]
```

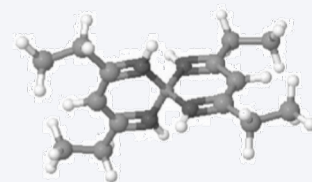AIDS Antiviral Screen Database of Active Compounds

250 training, 250 validation molecules

Two classes: `active` and `inactive`

Annotated in `train.tsv` and `validation.tsv`

(a) Active   (b) Inactive

Task

Classify the molecules of the validation set using KNN with the approximate Graph Edit Distance (GED)

XMLs with a lot of information

Only use the chemical `symbol` node label and the unlabeled, undirected edges:

```
<node id="_1">
        <attr name="symbol">
                <string>C</string>
        </attr>
        […]
</node>
<edge from="_1" to="_2">
        […]
</edge>
```

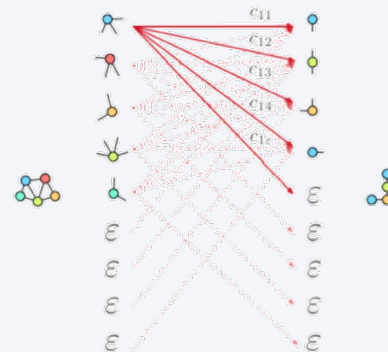Compute approximate GED between pairs of molecules with **bipartite graph matching**

Build cost matrix (*Dirac*)

$$\mathbf{C} = \begin{bmatrix} c_{11} & c_{12} & \cdots & c_{1m} & c_{1\varepsilon} & \infty & \cdots & \infty \\ c_{21} & c_{22} & \cdots & c_{2m} & \infty & c_{2\varepsilon} & \ddots & \vdots \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \ddots & \infty \\ c_{n1} & c_{n2} & \cdots & c_{nm} & \infty & \cdots & \infty & c_{n\varepsilon} \\ c_{\varepsilon 1} & \infty & \cdots & \infty & 0 & 0 & \cdots & 0 \\ \infty & c_{\varepsilon 2} & \ddots & \vdots & 0 & 0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \infty & \vdots & \ddots & \ddots & 0 \\ \infty & \cdots & \infty & c_{\varepsilon m} & 0 & \cdots & 0 & 0 \end{bmatrix}$$

Hungarian Algorithm
To find optimal assignment

Derive edit path costs from the result
(distance for classification)

KNN for classification (optimise for k)

Use *Dirac* cost function for GED (optimise c_n and c_e)

> Node substitution: $2*c_n$ if symbols not equal, otherwise 0
>
> Node deletion/insertion: c_n
>
> Edge deletion/insertion: c_e

Use an existing framework for the Hungarian algorithm

On ILIAS: `competitions/4-Molecules`

Input:

`gxl/`	XML files with graph definitions for each molecule
`test.tsv`	List of test files to predict (like GT but without the label)

Expected Output:

A TSV file: `test.tsv` (Columns separated by a tab: `\t`)

One line per molecule and whether it's `active` or `inactive`.

```
ID1 Label
ID2 Label
[…]
```

Example:

```
5 inactive
7 active
15 active
[…]
```

Final Presentation

Every group will give a short presentation in the last week (27th of May).

The presentation should be around 10 – 15 minutes and describe:

- How you organised your group
- Problems you encountered
- For each task:
  - Your approach to solve the task
  - What you think was crucial to get a good result
  - Improvements you would have liked to make
- General thoughts about the group exercises

Hand in the presentation on ILIAS before the start of the last lecture.

Deadline: Monday, 27th of May at 14:15

Submit via ILIAS
-> FS2024: 63082 Pattern Recognition
-> Group Presentation: hand-in

Questions?

29/04/2024