UNIVERSITÉ DE FRIBOURG
UNIVERSITÄT FREIBURG

# Pattern Recognition

## Exercise Session 9

## Last Group Projects + Evaluation Infos

Paul Märgner (paul.maergner@unifr.ch)

# Exam Registration!

- You have to register for the exam!
  - Deadline: May 17th, 2019
  - Withdrawal without reason until June 3rd, 2019

- Exam is on June 17th, 2019  10:00-11:00AM

# Task 3 – Keyword Spotting

Don’t forget the current group task

- Keyword Spotting
- Deadline: **April 29, 2019 (end of day)**

Data and Info on Github:
https://github.com/lunactic/PatRec17_KWS_Data

# Last Group Projects

Deadline: **Monday, May 20th, 23:59**

## Signature Verification

Can be solved with DTW

## Molecules

Use approximate Graph Edit Distance

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

# Final Report (tentative plan)

Please, create a final group report until
**Monday, May 20th**

It should be between 1-3 pages (A4)
The report should describe the “lessons learned”

You should describe:

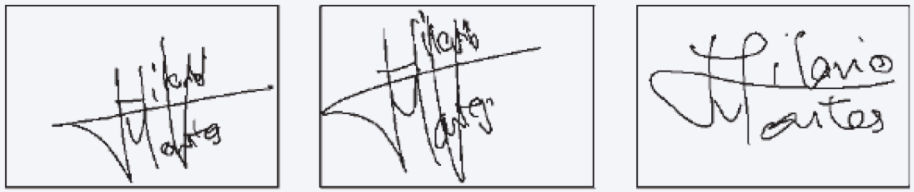
- How you organized your group
- What worked, what did not work
- For each task:
  - What is special about your solution
  - What was your approach
  - What worked, what did not work
- General thoughts about the group exercise

# Signature Verification

30 writers

Enrollment: 5 genuine signatures each

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



Task

Compute dissimilarity for each verification signature wrt the 5 genuine ones

# Signature Verification

## Input

`t x y pressure penup azimuth inclination`

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

`Azimuth` / `inclination` angles of the pen

## Recommendation: DTW

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

`vx, vy` velocity in x and y with respect to $\Delta t$

Normalize for each signature individually

Sakoe-Chiba band can be helpful

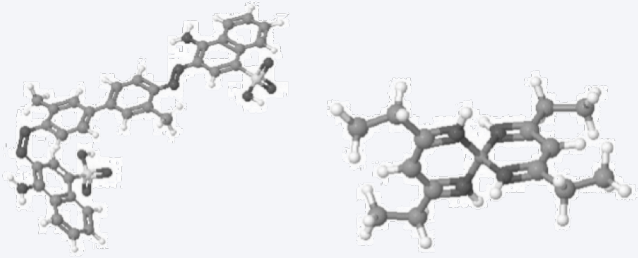
## Evaluation: mean average-precision

# Molecules

AIDS Antiviral Screen Database of Active Compounds

250 training, 250 validation molecules

- Two classes active ‘a’ and inactive ‘i’
- Annotated in `train.txt` and `valid.txt`



(a) Active (b) Inactive

Task

- Classify the molecules of the validation set using KNN
- Distance: approximate GED

Input: Graph xml

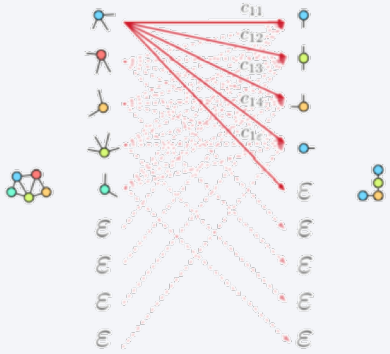
- `Nodes` labeled with their chemical `symbol`
- Unlabeled undirected `edges`

# Molecules

Compute approximate GED between pairs of molecules with bipartite graph matching
*(lecture 10, slide 21)*

Build cost matrix (*Dirac*)

$$\mathbf{C} = \left[\begin{array}{cccc|cccc} c_{11} & c_{12} & \cdots & c_{1m} & c_{1\varepsilon} & \infty & \cdots & \infty \\ c_{21} & c_{22} & \cdots & c_{2m} & \infty & c_{2\varepsilon} & \ddots & \vdots \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \ddots & \infty \\ c_{n1} & c_{n2} & \cdots & c_{nm} & \infty & \cdots & \infty & c_{n\varepsilon} \\ \hline c_{\varepsilon 1} & \infty & \cdots & \infty & 0 & 0 & \cdots & 0 \\ \infty & c_{\varepsilon 2} & \ddots & \vdots & 0 & 0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \infty & \vdots & \ddots & \ddots & 0 \\ \infty & \cdots & \infty & c_{\varepsilon m} & 0 & \cdots & 0 & 0 \end{array}\right]$$




Hungarian Algorithm
To find optimal assignment

Derive Edit Path costs from the result
(distance for classification)

KNN for classification (optimize for `K`)

# Molecules

## Recommendation

Use *Dirac* cost function for GED (optimize `Cn` and `Ce`)
*(lecture 9, slide 36)*

- Node substitution: `2*Cn` if symbols ≠, 0 otherwise
- Node deletion/insertion: `Cn`
- Edge deletion/insertion: `Ce`

Use an existing framework for the Hungarian algorithm

# Evaluation

# Evaluation General Information

You have to run evaluations on all group tasks except MNIST

Dates and times are still tentative!

Test Data available:

**Tuesday, May 21st, 10:00**

Results/Software/Readme on github:

**Thursday, May 23rd, 23:59**

Your results will be crosschecked

# Evaluation for KWS

## Input:

`keywords.txt` (keywords to search, take word images from training)

`test.txt` (containing the page numbers to search)

## Expected Output:

ASCII plaintext file

One line per keyword

```
Keyword1, testword_ID1, dissimilarity1, testword_ID2, dissimilarity2, ...
Keyword2, testword_ID1, dissimilarity1, testword_ID2, dissimilarity2, ...
```

Example:

```
h-u-n-d-r-e-d, 307-16-02, 732.2, 301-23-07, 792.666666667, 305-30-09, […]
G-e-o-r-g-e, 304-07-13, 905.6, 308-33-10, 935.533333333, 303-24-07, […]
[…]
```

# Evaluation for Signatures

## Input

- Same folder and input structures (enrollment/verification)
- Without `gt.txt`

## Expected Output:

- ASCII plaintext file
- One line per user (can be sorted by dissimilarity)

```
user1, signature_ID11, dissimilarity11, signature_ID12, dissimilarity12, ...
user2, signature_ID21, dissimilarity21, signature_ID22, dissimilarity22, ...
```

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,  […]
[…]
```

# Evaluation for Molecules

## Input:

Folder with gxl files

(gxl/`test_ID1`.gxl, gxl/`test_ID2`.gxl, …)

## Expected Output:

ASCII plaintext file

One line per test sample (gxl file)

```
test_ID1,predicted_classX
test_ID2,predicted_classY
```

Example:

```
556,i
5204,a
33,i
[…]
```

UNI
FR
UNIVERSITÉ DE FRIBOURG
UNIVERSITÄT FREIBURG

# Questions?