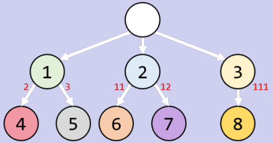


Master Thesis Seminar Talk

Progress Update

Fabrice Beaumont

Department of Information Systems and Artificial Intelligence - **Dr. Pascal Welke**

13. July 2022

Recap

RECAP PROGRESS:

May:

1. Task formulation
2. Dataset Loader module

June:

1. Wasserstein Distance

1.' *DataLoader:*
Software -> script.

2. "Naive" feedback loop

NEXT STEPS:

May:

1. Wasserstein Distance
2. "Naive" feedback loop

June:

1. "Naive" feedback loop.
2. Investigate its performance (and measures for it)

Overview & Outlook

RECAP PROGRESS:

June:

1. Wasserstein Distance

1.' *DataLoader:*
Software -> script.

2. “Naive” feedback loop

July (Update for today):

1. Cleaning the datasets
2. *Preparing comparison*
3. *Rethinking the WLLT structure*

NEXT STEPS:

June:

1. “Naive” feedback loop.
2. Investigate its performance (and measures for it)

July (Outlook):

1. Finish the feedback loop
2. Different edge weight trainings
3. Edge weights via FRM

New WLLT structure

For the computation of the WLLT I now use for files:

- Meta data (file names, wl-depth, nr. of tree vertices)
- Tree data in form of **path lists**
- Vertex labels of the **whole dataset** (**Vector**)
- Edge weights (**Vector**)

New WLLT structure

For the edge weight update:

- WLLT
- *Chose between different update mechanism*
- **Tree-Wasserstein distances**
  [2019, Tam Le, Tree-Sliced Variants of Wasserstein Distances]

For the evaluation:

- Cluster measures (max intra, min inter) (**Learning feedback**)
- Classification accuracy compared to other methods

Preparation of the performance comparison

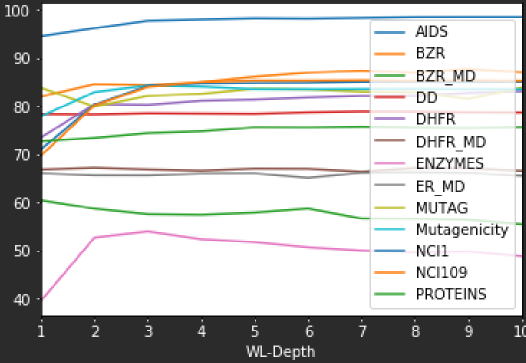


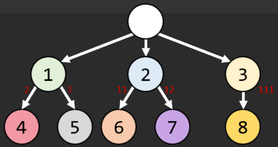
Figure: Classification accuracies on databases using Weisfeiler-Lehman.

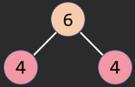
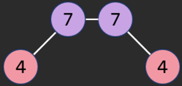
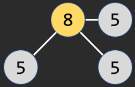
```
grakel.kernels.WeisfeilerLehman(n_iter=[1-10], base=grakel.kernels.VertexHistogram, normalize=True)
grakel.utils.cross_validate_Kfold_SVM(K, y, n_iter=10)
```

Thank you all for listening.

I will be happy to answer any **questions** and hear your **comments**.

Example of the whole procedure



A		$^2/_3$		$^1/_3$		
B		$^1/_2$			$^1/_2$	
C			$^3/_4$			$^1/_4$

Tree metric:

$$\begin{array}{ccccc} 4 & 5 & 6 & 7 & 8 \\ \end{array}$$
$$\begin{pmatrix} \cdot & 2 & 4 & 4 & 4 \\ & \cdot & 4 & 4 & 4 \\ & & \cdot & 2 & 4 \\ & \upuparrows & & \cdot & 4 \\ & & & & \cdot \end{pmatrix}$$

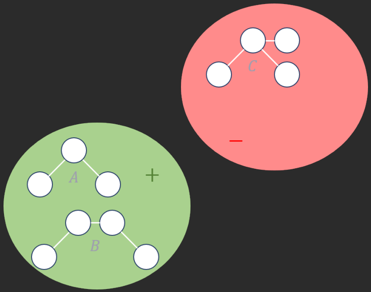
Wasserstein Dist.:

$$\mathcal{W}_t(A, B) = \frac{4}{3}$$
$$\mathcal{W}_t(A, C) = 3$$
$$\mathcal{W}_t(B, C) = 3$$

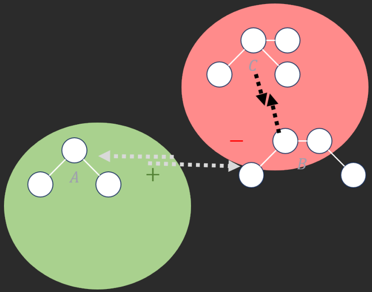
$$d_{\mathrm{WLLT}}(B, C) = 2 * \frac{2}{4} + 4 * \frac{1}{4} + 4 * \frac{1}{4} = \frac{12}{4} = 3$$

Example of the whole procedure

Current clustering:



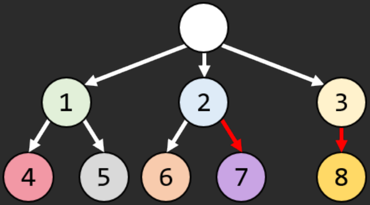
Target clustering:



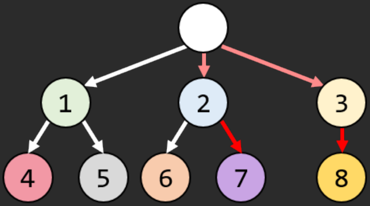
Idea: Reduce distance between B and C, by updating the edge weights.

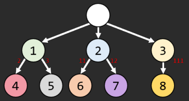
Example of the whole procedure

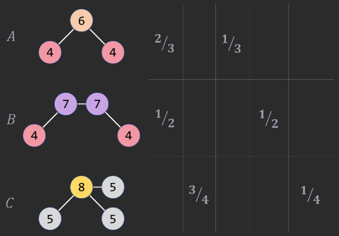
Local update $P_{7,8}$:



Weighted path update $P_{7,8}$:







Implementation road-map 1/2

- **WLLT Construction**:
  - Write to file and read from file. Construct WL-iteration based.
  - All weights *equal*.
  - (*Random* initial weights.)
  - (Use *a priori* knowledge.)

- **Wasserstein-Distance feedback**:
  - "Biggest pile of dirt". ("Smallest", to increase the distance.)
  - Distribution proportional to the pile size.
  - Distribution proportional to the cost of moving the pile size.

Implementation road-map 2/2

- **Update rule**:
  - Value:
    - Constant λ.
    - *Gradient descent*.
  - Location:
    - *Local*: Only update the first and last edge weights of the connecting path.
    - *Weighted path*: Update all edge weights on the path, with less magnitude for edges closer to the root.
    - *Path*: Update all edges on the path.
    - *Global*: Update all edges, related to all occurring labels.