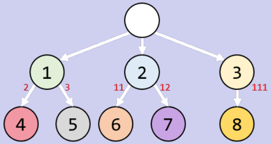


Master Thesis Seminar Talk

Progress Update

Fabrice Beaumont

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

10. August 2022

Recap progress

- Cleaning the datasets
- Preparing comparison
- Re-thinking the WLLT structure
- Tree-Wasserstein distances

  [2019, Tam Le, Tree-Sliced Variants of Wasserstein Distances]
- "Naive" feedback loop

Naive feedback loop

UNIVERSITÄT BONN

- Initialize all edge weights as 1.0.
- Compute the *Tree Wasserstein Distance*[1] between two graphs
- Pics the n highest differences in the weighted difference vector.[2]
- Push and pull graphs by changing the weights py percentage (0.1):

$$w' = \begin{cases} w * (1 + p_{\text{push}}) \\ w * (1 - p_{\text{pull}}) \end{cases}$$

[1] Normalized weighted distance between their wl-label histograms.
[2] Most expensive earth that had to be moved.

Naive feedback loop

- Initialize all edge weights as 1.0.
- Compute the *Tree Wasserstein Distance*[3] between two graphs
- Pics the n highest differences in the weighted difference vector.[4]
- Push and pull graphs by changing the weights py percentage (0.1):

$$w' = \begin{cases} w * (1 + p_{\text{push}}) \\ w * (1 - p_{\text{pull}}) \end{cases}$$

! Ensure that the sum of the edge weights is the same

! Ensure that the impact on the weights is proportional to the number of graphs in the sample

[3] Normalized weighted distance between their wl-label histograms.

[4] Most expensive earth that had to be moved.

Evaluation process

Implemented:

- Silhouette score
- Mean weight per WLLT layer

Outlook for august

1. Play with the settings, document different results
2. Fix and extend the evaluation process:
   - Mean **intra distance** in and **inter distance** between clusters
   - Percentage of changed weights
   - **Classification accuracy** compared to other methods

Further outlook

Further outlook:

- Implement different edge weight training:
  - Batch learning, Weight update after each distance computation
  - Treat weights in WLLT layers differently (e.g. update only leaves)
  - Update all weights in the WLLT path
  - ...
- Initialize edge weights via FRM method

Thank you all for listening.

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

Preparation of the performance comparison

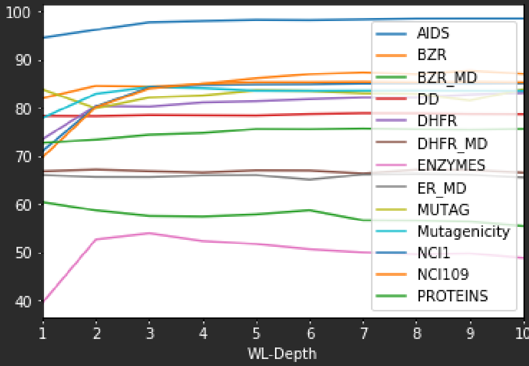
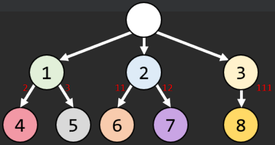


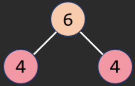
Figure: Classification accuracies on databases using Weisfeiler-Lehman.

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

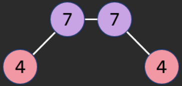
Example of the whole procedure



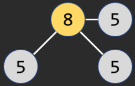
A



B



C



$^2/_3$		$^1/_3$		
$^1/_2$			$^1/_2$	
	$^3/_4$			$^1/_4$

Tree metric:

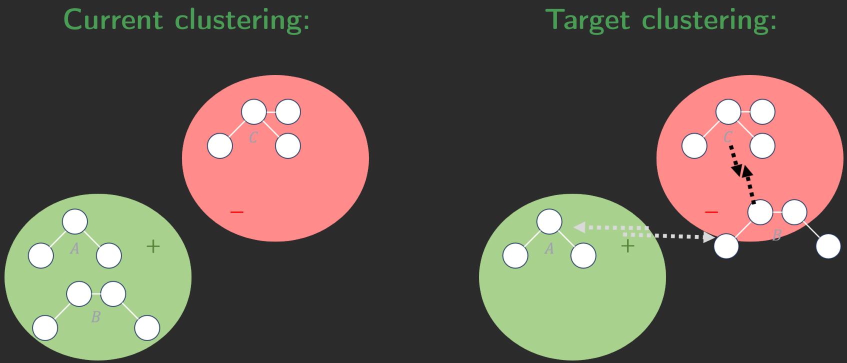
$$\begin{array}{c}\begin{array}{ccccc} 4 & 5 & 6 & 7 & 8 \end{array}\\ \begin{pmatrix} \cdot & 2 & 4 & 4 & 4 \\ & \cdot & 4 & 4 & 4 \\ & & \cdot & 2 & 4 \\ & \upuparrows & & \cdot & 4 \\ & & & & \cdot \end{pmatrix}\end{array}$$

Wasserstein Dist.:

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

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

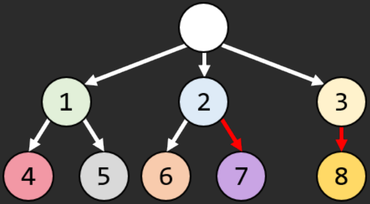
Example of the whole procedure



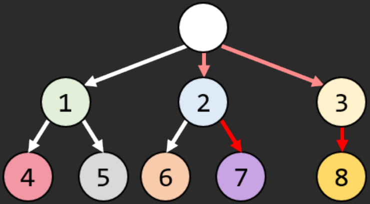
Idea: Reduce distance between B and C, by updating the edge weights.

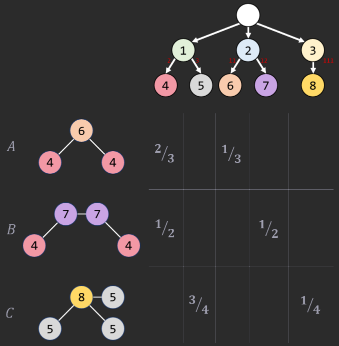
Example of the whole procedure

Local update $P_{7,8}$:



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





Implementation road-map 1/2

UNIVERSITÄT BONN ML AI lab

- **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.