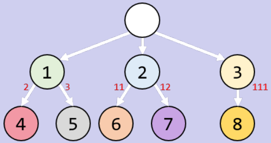


Master Thesis Seminar Talk

Progress Upade

Fabrice Beaumont

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

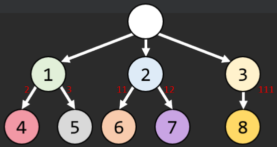
15. June 2022

Recap last months progress

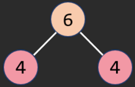
1. Task formulation, registration of the thesis:
   "**Learning graph similarity measures using the Weisfeiler-Lehman label hierarchy**"
   Definition of several sub-goals a programming road-map.

2. Implementation of a dynamic **Dataset Loader** (*GarKel*, *OGB*, from file).
   Easily expandable for other frameworks.

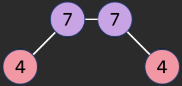
Example of the whole procedure



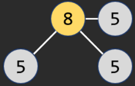
A



B



C



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

Tree metric:

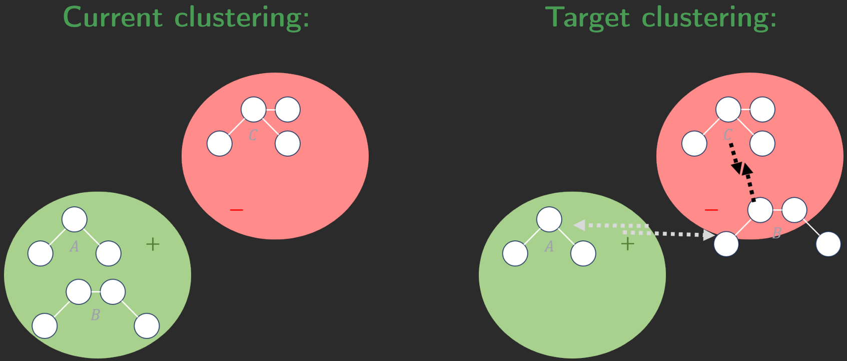
$$\begin{array}{c} \begin{matrix} 4 & 5 & 6 & 7 & 8 \end{matrix} \\ \begin{pmatrix} \cdot & 2 & 4 & 4 & 4 \\ & \cdot & 4 & 4 & 4 \\ & & \cdot & 2 & 4 \\ & \upuparrows & & \cdot & 4 \\ & & & & \cdot \end{pmatrix} \end{array}$$

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.

"Next steps" from last month

.

- Implement the usage of the **Wasserstein Distance**.
- Implement a "naive" **feedback loop** to update the WLLT edge weights.

  .

  (And the more and more complex variations.)

Current overview

.

- Implement the usage of the **Wasserstein Distance**.
  DONE
- Implement a “naive” **feedback loop** to update the WLLT edge weights.
  IN PROGRESS
  (And the more and more complex variations.)

Current overview

- Rethinking the data-loader procedure. “From software to script”.
  DONE
- Implement the usage of the Wasserstein Distance.
  DONE
- Implement a “naive” feedback loop to update the WLLT edge weights.
  IN PROGRESS
  (And the more and more complex variations.)

Next steps

- Complete one “naive” implementation.
- Investigate its performance (and measures for it).
- (Implement more complex variations.)

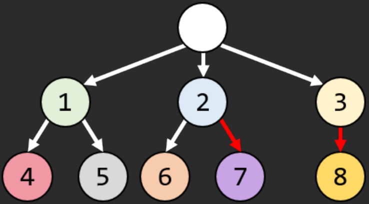
Goal for the next meeting: Present evaluations of at least one implementation compared to the state of the art.

Thank you all for listening.

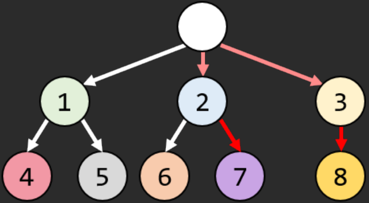
I will be happy to answer any **questions** and hear your **comments**.

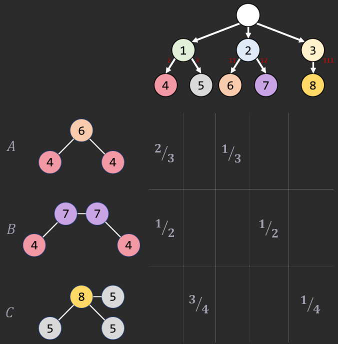
Example of the whole procedure

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.