Machine Learning for Graphs and Sequential Data

Graphs – Limitations of GNNs

Lecturer: Prof. Dr. Stephan Günnemann

www.daml.in.tum.de

Summer Term 2020

Data Analytics and
Machine Learning

Roadmap

- **Chapter: Graphs**
  1. Graphs & Networks
  2. Generative Models
  3. Clustering
  4. Node Embeddings
  5. Ranking
  6. Semi-Supervised Learning
  7. **Limitations of GNNs**
     - **Overview**
     - Robustness

Types of Limitations

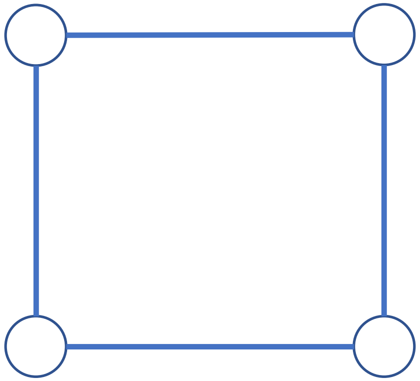
- GNNs have become extremely popular and are getting used in many domains
  - Though, still a very young research field – many open questions
  - In particular, various limitations regarding current models

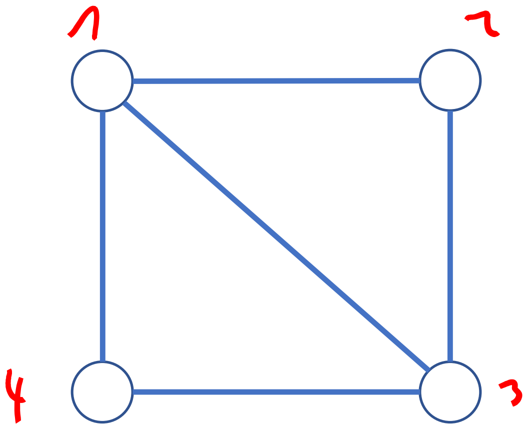
- Expressiveness
  - Worse than the simple WL test on the graph isomorphism problem
  - GCN-like models tend to overly smooth predictions with increasing depth
- Reliability / (Non-)Robustness
  - GNNs are susceptible to adversarial graph manipulations, see next section
- Scalability to large graphs
  - Message passing requires communication and synchronization at every step
  - At the same time you have to process the whole dataset at once because the nodes are not i.i.d.; batching/stochastic training not trivial
  - Adding a new node or edge affects large parts of the graph because of the small world phenomenon and requires a complete recomputation of the predictions/embeddings

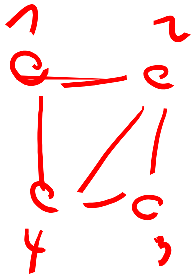
Graph Isomorphism Problem

- Determine whether two graphs are structurally identical
  - Can we map all nodes from one graph onto the nodes from the second graph such that the edge structure is preserved?
- Best known algorithm has exponential worst-case runtime

- Are these two graphs isomorphic?

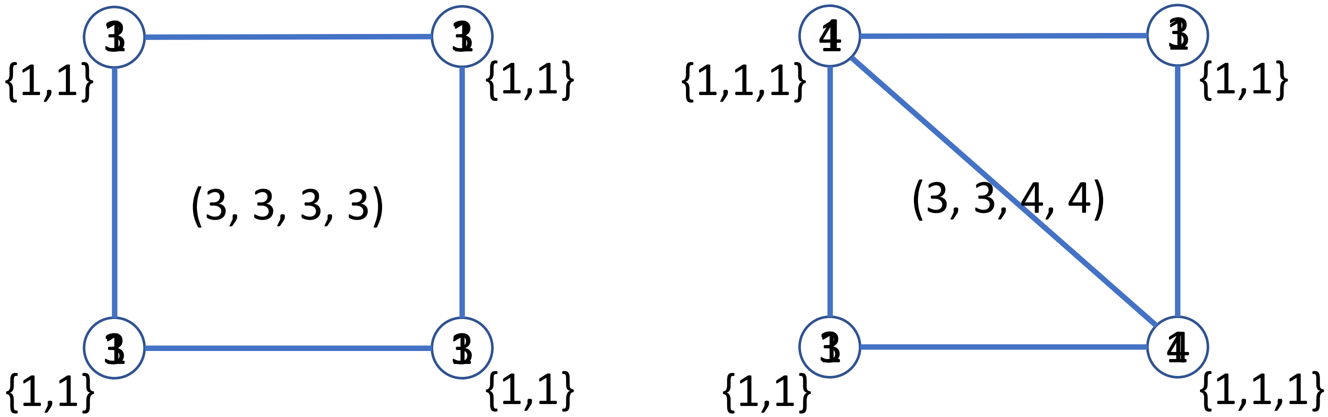


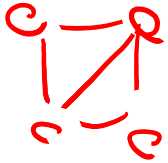




Weisfeiler-Lehman Test

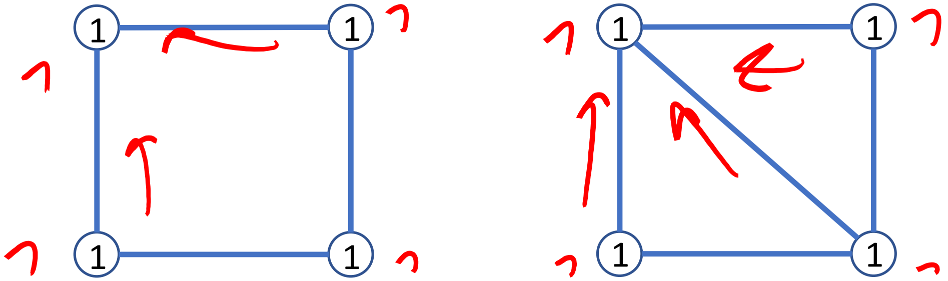
1. Label all nodes as 1
2. Collect all neighboring labels in ordered multisets
3. Relabel each node by hashing the current label concatenated with the neighbor labels with some hash function
4. Compare the sorted node labels
   - If they are different, the two graphs are definitely not isomorphic
   - Otherwise, the graphs *might* be isomorphic and we continue with step 2 until the sorted labels have converged or our computation budget is used up





Weisfeiler-Lehman vs. Graph Neural Networks

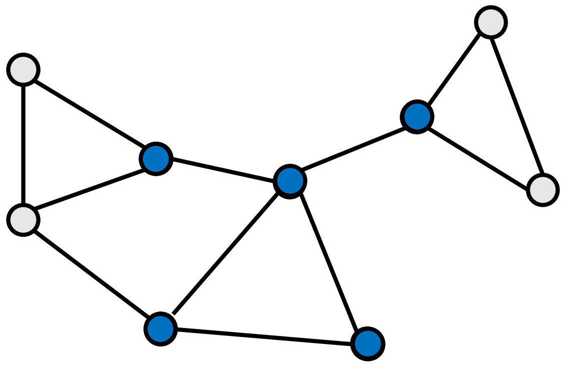
- The WL test can distinguish many non-isomorphic graphs but not all of them
- However, that looked a lot like message passing with hash aggregation
- How do GNNs compare to WL?
  - They often can distinguish even fewer graphs (consider mean or max aggregation in our running example)



- Increasing expressiveness of GNNs:
  - carefully designed aggregation functions lead to the same power as WL [Xu2019]
  - use of higher order structure [Morris2019] or directional/edge information [Klicpera2020]

Oversmoothing (I)

- In a 3-layer GCN information propagates from the blue to the red node along all possible walks of length 3 or less with about equal weight



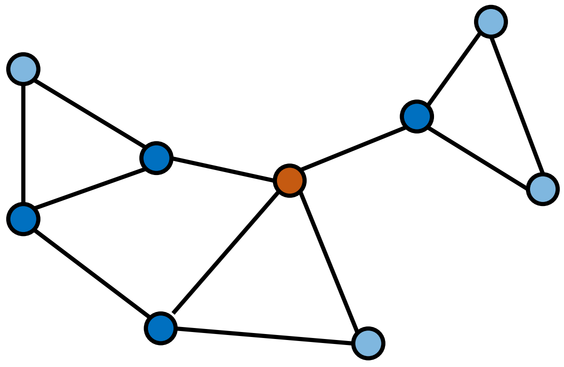
All paths along which information travels from blue to red

- Influence between the nodes is proportional to the probability of visiting blue from red via a random walk of length k

[Xu2018]

$X^{(t+1)} = D^{-1} A \cdot X^{(t)}$

Oversmoothing (II)



- In the limit of infinite layers, the influence becomes proportional to the stationary distribution of the graph as a markov chain (see PageRank)
- The stationary distribution is a global property of the graph

  → The blue node influences every other node in the same way regardless of their distance

- Deep GCN-like networks are unable to represent local neighborhoods

[Xu2018]

PageRank in the Message Passing Framework

$$H^{(l+1)} = \sigma\left(\tilde{D}^{-1/2}\tilde{A}\tilde{D}^{-1/2}H^{(l)}W^{(l)}\right)$$

$$r^{(t+1)} = \boxed{AD^{-1}}\boxed{r^{(t)}}$$

PageRank iteration

$$H^{(l+1)} = \boxed{\tilde{D}^{-1/2}\tilde{A}\tilde{D}^{-1/2}}\boxed{\sigma\left(H^{(l)}\right)W^{(l)}}$$

GCN iteration

- PageRank and a GCN layer exhibit some structural similarity
- If we "push" the nonlinearity one layer up, the similarity becomes even more apparent

● Features ● Message Passing

- We know how to adapt PageRank to focus on a node's local neighborhood

  → Personalized PageRank with teleport vector $\pi = e_i$, the i-th unit vector for the i-th node

  → Extend GCN in the same way!

Personalized Propagation of Neural Predictions (PPNP)

$\in \mathbb{R}^{n\times n}$

$$r^{(t+1)} = (1-\alpha)\underline{AD^{-1}}r^{(t)} + \alpha\pi \qquad H^{(l+1)} = \underline{(1-\alpha)\tilde{D}^{-½}\tilde{A}\tilde{D}^{-½}}H^{(l)} + \alpha H^{(0)}$$

Personalized PageRank iteration　　　　Personalized PNP iteration

$$R = \alpha \cdot \left(I - (1-\alpha)\underline{A \cdot D^{-1}}\right)^{-1} \begin{bmatrix} 0 & \vdots & 0 & \vdots \\ \vdots & \vdots & \vdots & \vdots \end{bmatrix}$$

α teleport probability

π teleport vector

1. Separate transformation and propagation
   - Instead of $H^{(0)} = X$, let $H_{i,:}^{(0)} = f_\theta(X_{i,:})$ for some f_θ, e.g. a neural network

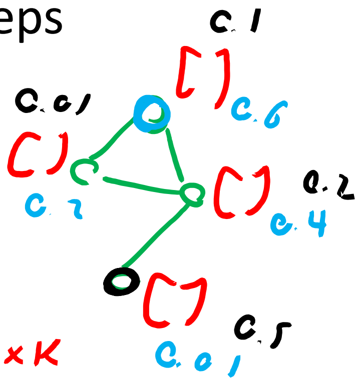
   $\in \mathbb{R}^{k}$　$\in \mathbb{R}^{n\times d}$　$\in \mathbb{R}^{d}$

2. Introduce personalized teleportation

- As with PageRank, we can take the limit of infinite propagation steps
- We arrive at the PPNP model for node classification

$$\underset{\in \mathbb{R}^{n\times k}}{Z} = \mathrm{softmax}\left(\underbrace{\alpha\left(I_n - (1-\alpha)\underline{\tilde{D}^{-½}\tilde{A}\tilde{D}^{-½}}\right)^{-1}}_{\in \mathbb{R}^{n\times n}} \underset{\in \mathbb{R}^{n\times k}}{H^{(0)}}\right)$$

iD



[Klicpera2019]

PPNP Alleviates Some of the Limitations

- Personalized PageRank prevents oversmoothing
  - Immediate neighborhood gets higher weight
- Clear separation between depth of transformation and message passing
  - Transformation depth is the depth of the deep model f_θ
  - PPNP uses the limit distribution of personalized PageRank which corresponds to infinite transformation depth
- Scalability
  - The predictions $f_\theta(v_i)$ can be computed efficiently for arbitrarily large graphs since they are made individually per node (no interaction)
  - Propagation only works with the lower dimensional predictions instead of high dimensional node attributes or intermediate representations
  - However: As with PageRank the exact solution is too costly
    - Computation involves a matrix inversion with runtime at least $O(n^{2.373})$
    - Inverse is dense and requires $O(n^2)$ memory

Approximate PPNP

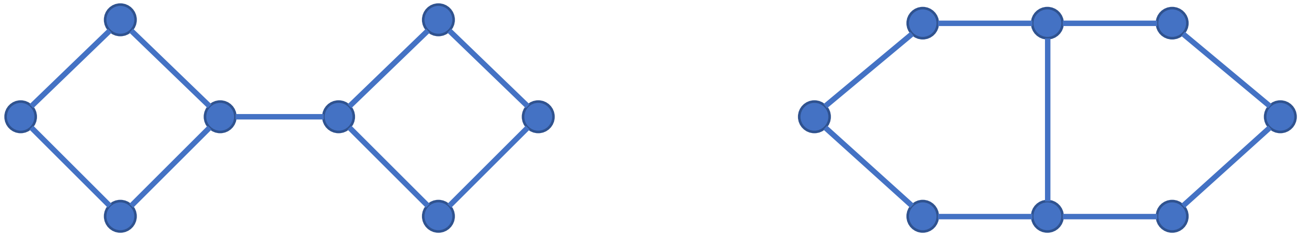
- Observation: The exact solution is the limit of an iterative update equation
- Approximate the exact solution as $Z \approx H^{(K)}$, i.e. with K iterations of

$$H^{(l+1)} = (1-\alpha)\widetilde{D}^{-1/2}\tilde{A}\widetilde{D}^{-1/2}H^{(l)} + \alpha H^{(0)}$$

- Never realizes a dense $n \times n$ matrix and exploits the sparsity of $\tilde{A}$
  - Only needs $O(Knd)$ runtime and $O(nd)$ memory
- Setting $K = 10$ already approximates the exact solution effectively

Questions

- The Weisfeiler-Lehman test works similar to message passing with hashing as an aggregation function. Why don't we just use hash aggregation in GNNs to lift them to the same level of expressiveness?



- Above you see two non-isomorphic graphs that the WL test cannot distinguish
  - Can you make this counter example smaller?

- Why is oversmoothing a problem for node-level prediction models?

Reading Material

- [Xu2018] Xu, K., Li, C., Tian, Y., Sonobe, T., Kawarabayashi, K., & Jegelka, S. (2018). Representation Learning on Graphs with Jumping Knowledge Networks. In ICML 2018: Thirty-fifth International Conference on Machine Learning (pp. 5449–5458).
- [Xu2019] Xu, K., Hu, W., Leskovec, J., & Jegelka, S. (2019). How Powerful are Graph Neural Networks. In ICLR 2019 : 7th International Conference on Learning Representations.
- [Klicpera2019] Klicpera, J., Bojchevski, A., & Günnemann, S. (2019). Predict then Propagate: Graph Neural Networks meet Personalized PageRank. In ICLR 2019 : 7th International Conference on Learning Representations.
- [Klicpera2020] J. Klicpera, J. Groß, S. Günnemann: Directional Message Passing for Molecular Graphs. ICLR 2020
- [Morris2019] C. Morris, M. Ritzert, M. Fey, W. L. Hamilton, J. E. Lenssen, G. Rattan, M. Grohe: Weisfeiler and Leman Go Neural: Higher-Order Graph Neural Networks. AAAI 2019: 4602-4609
- Blogpost on WL and GNNs: https://towardsdatascience.com/expressive-power-of-graph-neural-networks-and-the-weisefeiler-lehman-test-b883db3c7c49