Deep Learning on Graphs: Advanced Topics, Part 1

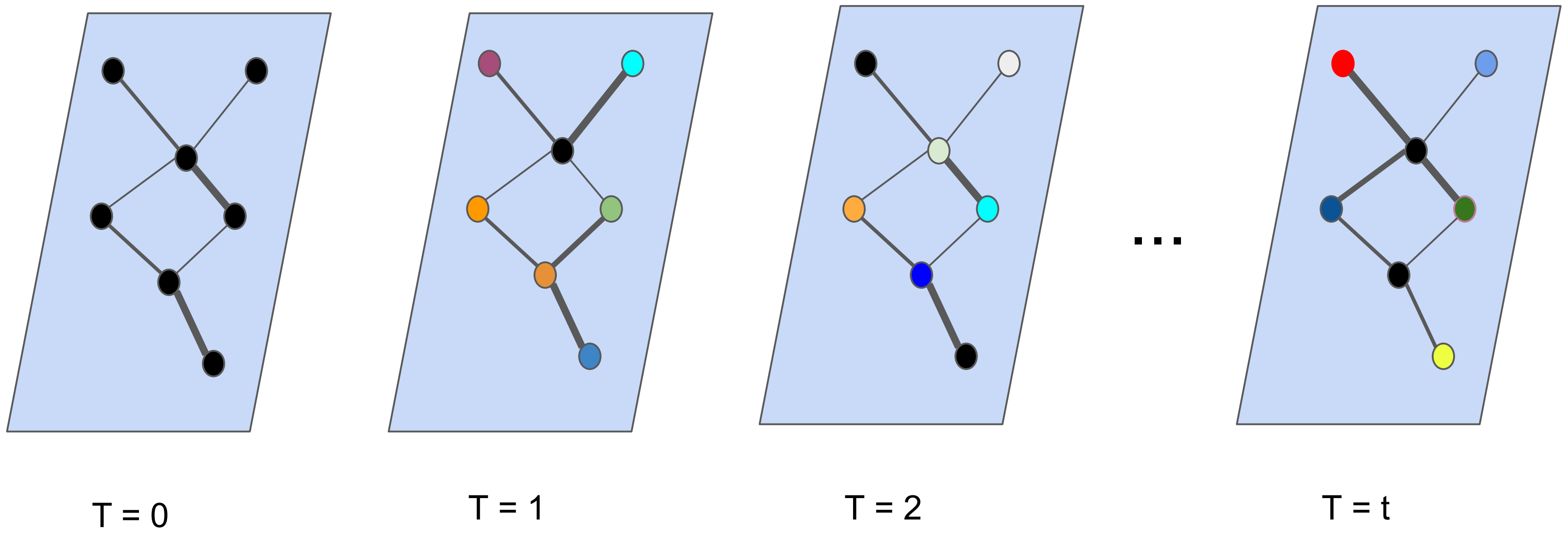
Michael Kenning, Stavros Georgousis

Temporal Graphs

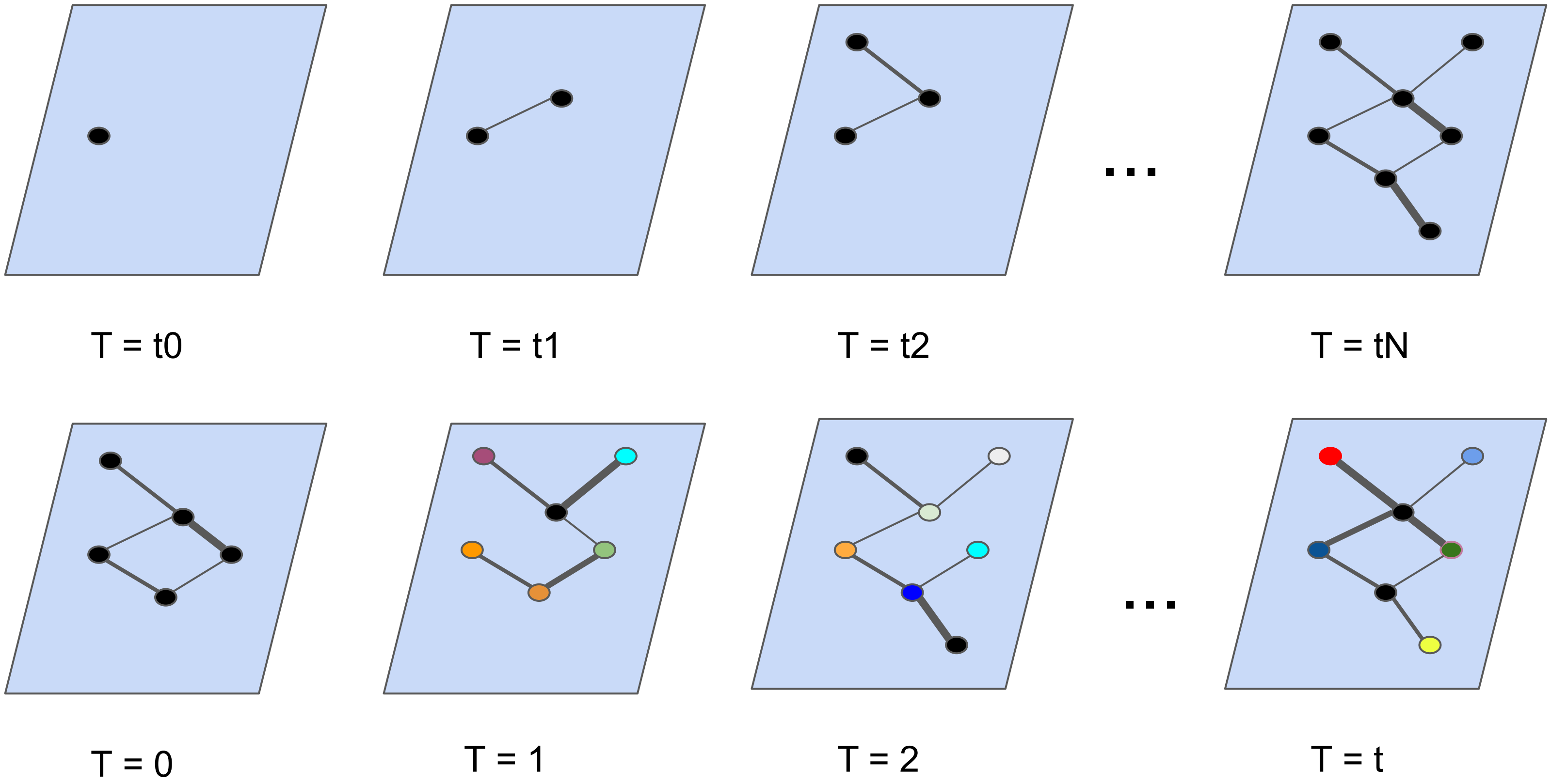
Graphs that have one or more attributes changing over time

- Dynamic Graphs:
  - Graphs with set vertices. Graph signal (vertex features) and connectivity (edges) change over time.
  - Special case: Spatio-temporal graphs (only graph signal change over time).
  - Applications: S-T attributes forecasting (traffic flow), action recognition
- Evolving Graphs:
  - Graphs where everything change over time.
  - Addition + removal of vertices and/or edges.
  - Applications: Temporal relational graphs, communication networks

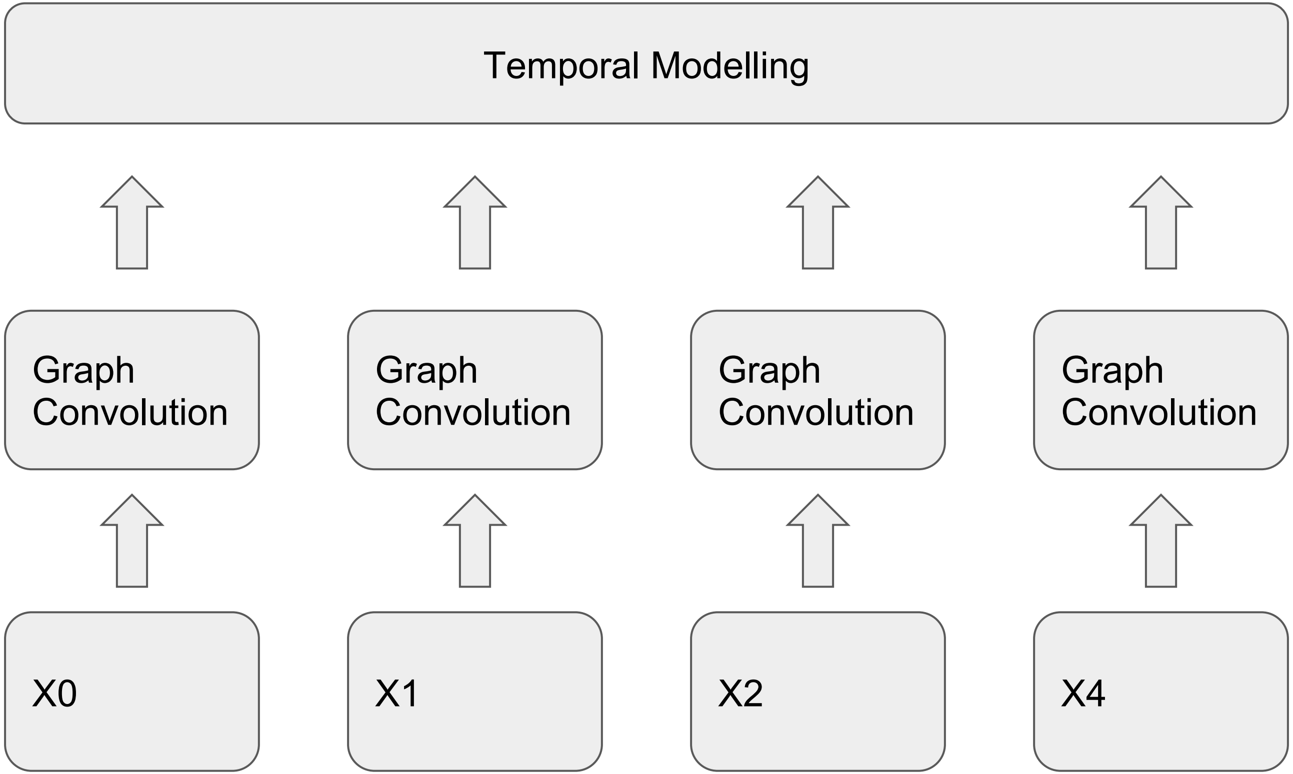
Temporal Graph Representation: Dynamic Graphs



Temporal Graph Representation: Evolving Graphs

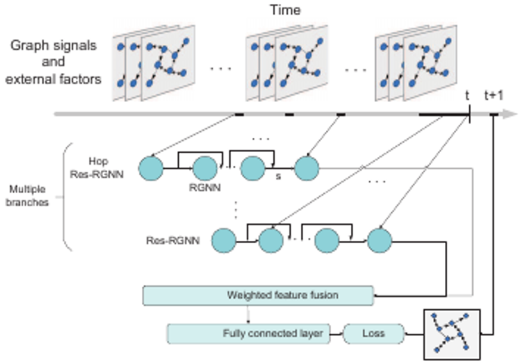


Dynamic Graphs



Dynamic Graphs

- Temporal modelling
  - Temporal convolution
  - Recurrent network
  - Attention networks
- Temporal convolution can be prior to graph convolution
- Interesting techniques:
  - Multiple temporal scales
  - Adversarial training



Source: Chen et al. 2019

Discrete-Time Evolving Graphs

Focus is mostly link prediction, event-time prediction

- EvolveGCN (Pareja et al., 2020)
  - Hypothesis that graph convolution layer limits the generalisability of the graph model. The recurrent units model the temporal variation of the GC layer’s weights rather than the graph representation.

Continuous Time Evolving Graphs

- Limited research

Basic idea

- Incremental growth of the graph.
- Information propagated from newly added relations (edges).

Examples of techniques:

- Know-Evolve (Trivedi et al., 2017). Relations (edges) are modelled as temporal point processes.
- Streaming Graph Neural Networks (Ma et al. 2020), event-based, spatial convolution approach updating neighbouring vertex features.

Temporal Graphs Limitations

- Deletion operations have largely been ignored (Not interesting enough? / Difficult?)
- In streaming scenarios vertex features are treated as secondary elements and relations (edges) are the primary focus which dictate the evolution of vertex features.

Network Depth and Over-smoothing

Over-smoothing problem: Degradation of performance with increasing architecture depth.

Graph convolution applies a smoothing operation; with consecutive convolutional layers vertex features converge to similar values.

Inference on vertex properties is greatly reduced.

Over-smoothing problem: What we know

1) Over-smoothing rate depends on spectral properties (smallest positive eigenvalue of the Laplacian) + the maximum singular value of weights. The problem is exaggerated with larger and denser graphs. Oono and Suzuki, 2019.
2) Non-linear activations ReLU and Leaky ReLU reduce expressive power, affecting over-smoothing, Cai and Wang, 2020, Luan et al, 2019.. Tanh is better at retaining linear independence of features.
3) Transformation and propagation coupling increases the rate of over-smoothing. Liu et al, 2020.

Over-smoothing problem: Mitigation

1) Weight and feature normalisation
    a) Feature normalisation can be a post-hoc mitigation strategy.
    b) E.g; PairNorm - Zhao and Akoglu, 2020.
2) Residual connections
    a) Residual connections alone only delay the problem, Chen et al.
3) Identity mapping
    a) Acts as normaliser to the weight values. E.g. Chen et al.
4) Edge sampling
    a) Artificially sparsify the graph, with random edge sampling with replacement for retaining information
5) Non-linear activation functions

Other considerations

- Model inductivity
- Orders of structure
  - Similar to network depth necessity (learn more complex features).
- Structure and data
- Incomplete information

References

1) Chen, Cen, et al. "Gated residual recurrent graph neural networks for traffic prediction." *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 33. No. 01. 2019.
2) Pareja, Aldo, et al. "Evolvegcn: Evolving graph convolutional networks for dynamic graphs." *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 34. No. 04. 2020.
3) Trivedi, Rakshit, et al. "Know-evolve: Deep temporal reasoning for dynamic knowledge graphs." *International Conference on Machine Learning*. PMLR, 2017.
4) Ma, Yao, et al. "Streaming graph neural networks." *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2020.
5) Oono, Kenta, and Taiji Suzuki. "Graph neural networks exponentially lose expressive power for node classification." *arXiv preprint arXiv:1905.10947* (2019).
6) Cai, Chen, and Yusu Wang. "A note on over-smoothing for graph neural networks." *arXiv preprint arXiv:2006.13318* (2020).
7) Luan, Sitao, et al. "Break the ceiling: Stronger multi-scale deep graph convolutional networks." *arXiv preprint arXiv:1906.02174* (2019).
8) Liu, Meng, Hongyang Gao, and Shuiwang Ji. "Towards deeper graph neural networks." *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2020.
9) Zhao, Lingxiao, and Leman Akoglu. "Pairnorm: Tackling oversmoothing in gnns." *International Conference on Learning Representations, 2020.*