Graph Neural Networks in Action

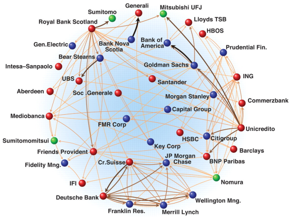
Gonzalo Mateos
Dept. of ECE and Goergen Institute for Data Science
University of Rochester
gmateosb@ece.rochester.edu
http://www.hajim.rochester.edu/ece/sites/gmateos/

April 17, 2023

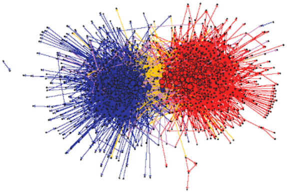
Learning from relational data

- ▶ Graphs are natural models for relational data that can help to learn in various timely applications

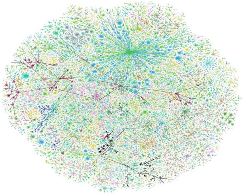
Economic Networks



Social and Information Networks

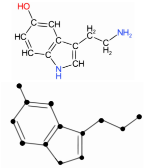


Internet



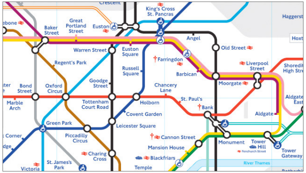
3D Meshes

Molecules

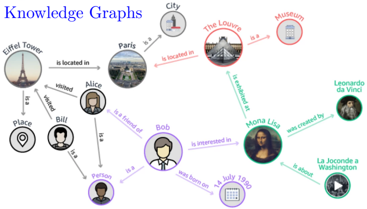


Brain Connectomes

Transportation Networks



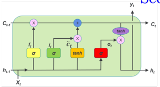
Knowledge Graphs

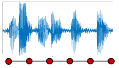


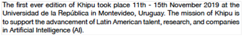
Machine learning on graphs: Challenges

- We're quite good at learning from data in Euclidean domains (sequences, multi-dim. grids)...

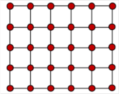
Sequences (text, speech)

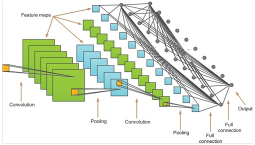





The first ever edition of Khipu took place 11th - 15th November 2019 at the Universidad de la República in Montevideo, Uruguay. The mission of Khipu is to support the advancement of Latin American talent, research, and companies in Artificial Intelligence (AI).

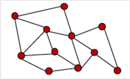
Grids (images, video)





...but we want to learn from data defined on graphs $G(V, E)$

Graphs (networks, relational data)



- Challenges: no geometry (in general), irregular neighborhoods, arbitrary size, often dynamic

  ⇒ Ordering? Translation? Convolution? Structural priors (stationarity, invariances)?

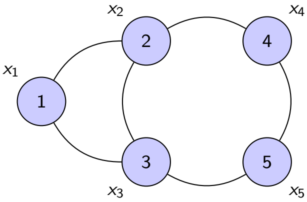
M. Bronstein *et al*, "Geometric deep learning: Going beyond Euclidean data," *IEEE Signal Process Mag*, 2017

Graph notation

- Graph $G(V, E)$ with adjacency matrix $\mathbf{A} \in \mathbb{R}_{+}^{N_v \times N_v}$
  - $\Rightarrow$ A_{ij} = proximity between i and j
  - $\Rightarrow$ Neighborhood of i is $\mathcal{N}_i = \{j \in V : (i, j) \in E\}$
- Define a signal $\mathbf{x} \in \mathbb{R}^{N_v}$ supported on V
  - $\Rightarrow$ x_i = signal (feature, attribute) value at node i
  - $\Rightarrow$ Could be a feature *vector* $\mathbf{x}_i \in \mathbb{R}^F$

$$\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_{N_v}]^\top \in \mathbb{R}^{N_v \times F}$$

- Target labels $\mathbf{y}$ for (semi)-supervised learning



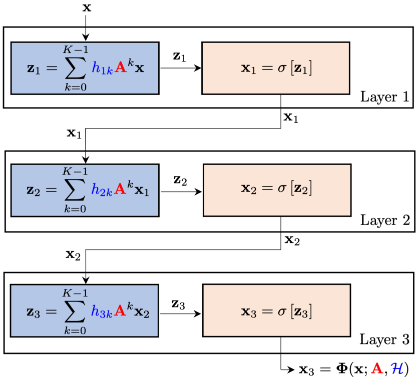
Recap: Graph neural networks (GNNs)

- A GNN is a minor variation of a graph filter
  $\Rightarrow$ Pointwise nonlinearities and compositions

- Equivariance $\boldsymbol{\Phi}(\mathbf{Px}; \mathbf{PAP}^\top, \mathcal{H}) = \mathbf{P}\boldsymbol{\Phi}(\mathbf{x}; \mathbf{A}, \mathcal{H})$
  $\Rightarrow$ Independent of node labeling

- GNNs can be transferred across different graphs
  $\Rightarrow$ Graph $\mathbf{A}$ reinterpreted as input in $\boldsymbol{\Phi}(\mathbf{x}, \mathbf{A}; \mathcal{H})$

- Convolutional (C)NNs are a particular case
  $\Rightarrow$ When $\mathbf{A}$ is adjacency of a directed cycle



$\mathbf{x}$
$\mathbf{z}_1 = \sum_{k=0}^{K-1} h_{1k}\mathbf{A}^k\mathbf{x}$
$\mathbf{z}_1$
$\mathbf{x}_1 = \sigma[\mathbf{z}_1]$
Layer 1
$\mathbf{x}_1$
$\mathbf{x}_1$
$\mathbf{z}_2 = \sum_{k=0}^{K-1} h_{2k}\mathbf{A}^k\mathbf{x}_1$
$\mathbf{z}_2$
$\mathbf{x}_2 = \sigma[\mathbf{z}_2]$
Layer 2
$\mathbf{x}_2$
$\mathbf{x}_2$
$\mathbf{z}_3 = \sum_{k=0}^{K-1} h_{3k}\mathbf{A}^k\mathbf{x}_2$
$\mathbf{z}_3$
$\mathbf{x}_3 = \sigma[\mathbf{z}_3]$
Layer 3
$\mathbf{x}_3 = \boldsymbol{\Phi}(\mathbf{x}; \mathbf{A}, \mathcal{H})$

L. Ruiz *et al*, "Graph neural networks: Architectures, stability, and transferability," *Proc. IEEE*, 2021

What is this lecture about?

Canonical tasks and examples

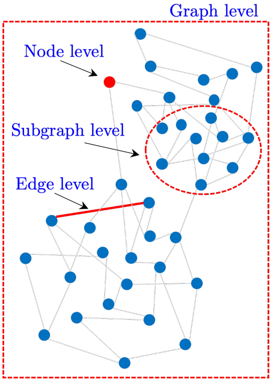
Link prediction: from handcrafted features to learned representations

GNNs in the wild

Parting words

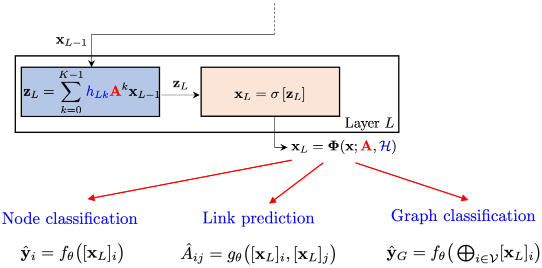
Machine learning on graphs: Tasks

- Graph visualization and pattern discovery
  - Ex: How is the science and technology enterprise developing?
- Graph modeling and generation
  - Ex: Generate new molecules with antibacterial properties?
- Clustering and community detection
  - Ex: Which groups of individuals have similar political beliefs?
- Node regression/classification and semi-supervised learning
  - Ex: Can we identify protein function from their physical binding?
- Link/relation prediction
  - Ex: Predict user-item interactions in recommendation systems?
- Graph regression/classification
  - Ex: Predict cognitive decline from brain connectomes?



Task-dependent readout

- The learning task at hand dictates how we process GNN outputs



$\mathbf{x}_{L-1}$

$\mathbf{z}_L = \sum_{k=0}^{K-1} h_{Lk} \mathbf{A}^k \mathbf{x}_{L-1}$

$\mathbf{z}_L$

$\mathbf{x}_L = \sigma\left[\mathbf{z}_L\right]$

Layer L

$\mathbf{x}_L = \mathbf{\Phi}(\mathbf{x}; \mathbf{A}, \mathcal{H})$

Node classification

$\hat{\mathbf{y}}_i = f_\theta\big([\mathbf{x}_L]_i\big)$

Link prediction

$\hat{A}_{ij} = g_\theta\big([\mathbf{x}_L]_i, [\mathbf{x}_L]_j\big)$

Graph classification

$\hat{\mathbf{y}}_G = f_\theta\big(\bigoplus_{i \in \mathcal{V}} [\mathbf{x}_L]_i\big)$

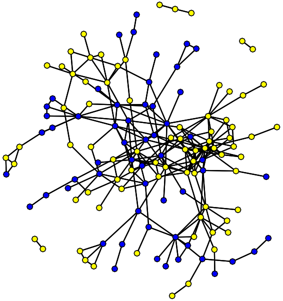
- Ex: Reduce final-layer node representations $\{[\mathbf{x}_L]_i\}_{i \in \mathcal{V}}$ for graph classification

  $\Rightarrow$ $\bigoplus$ is a permutation-invariant aggregator; e.g., sum, mean, max

  $\Rightarrow$ Train over a set of graphs $\{\mathbf{y}_{G_t}, (\mathbf{A}_t, \mathbf{x}_t)\}_{t \in \mathcal{T}}$ using e.g., a cross-entropy loss

Example: Predicting protein function

- Baker's yeast data, formally known as *Saccharomyces cerevisiae*
  - Graph: 134 vertices (proteins) and 241 edges (protein interactions)

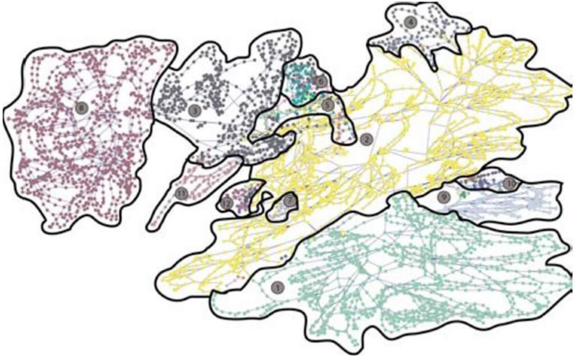


- Target signal: functional annotation intracellular signaling cascade (ICSC)
  - Signal transduction, how cells react to the environment
  - $y_i = 1$ if protein i annotated ICSC (yellow), $y_i = 0$ otherwise (blue)

E. Kolaczyk, *Statistical Analysis of Network Data: Methods and Models*, Springer, 2009

Example: Unveiling network communities

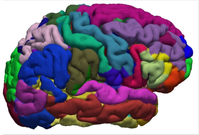
- Partition the electrical power grid into areas with minimum inter-area interactions



- **Applications** to grid operation and monitoring
  - Decide control areas for distributed power system state estimation
  - Parallel computation of power flow
  - Controlled islanding to prevent spreading of blackouts

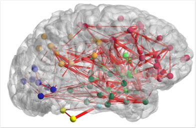
Example: Network neuroscience

- Challenge: understanding human brain function and structure



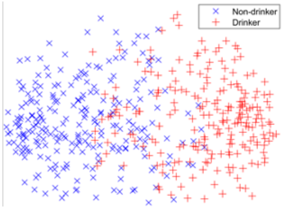
Brain atlas

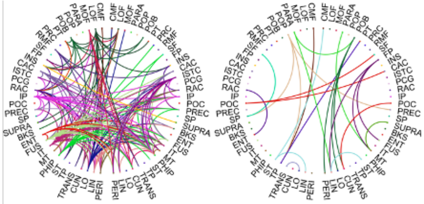
Tractography



Brain graph

- Subject-level classification. Q: Is brain connectivity affected by heavy drinking?

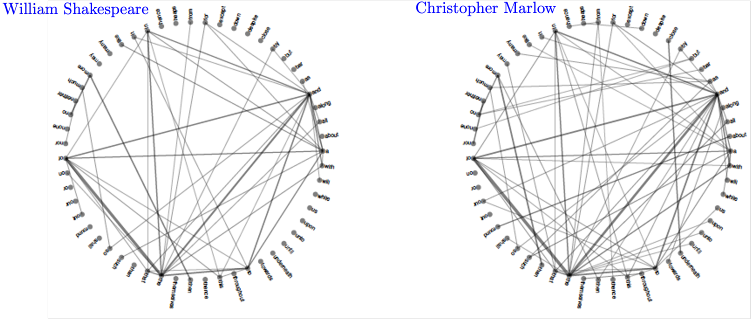




Y. Li *et al*, "Learning to model the relationship between brain structural and functional connectomes," *IEEE TSIPN*, 2022

Example: Authorship attribution with Word Adjacency Networks (WANs)

- Nodes are different function words and edges how often words appear close to each other
  - ⇒ Proxy for the different ways in which authors use the English language grammar



- Shakespeare's and Marlowe's WANs different enough to ascertain their collaboration on Henry VI

S. Segarra *et al*, "Attributing the authorship of the Henry VI plays by word adjacency," *Shakespeare Quarterly*, 2016

Roadmap

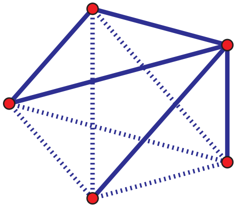
Canonical tasks and examples

Link prediction: from handcrafted features to learned representations

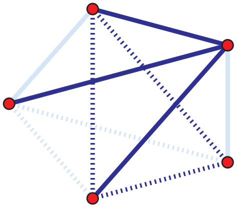
GNNs in the wild

Parting words

Link prediction



Original graph



Link prediction

- Given signal $\mathbf{x} = [x_1, \ldots, x_{N_v}]^\top$ and edge status only for subset of pairs $V^{(2)}_{obs} \subset V^{(2)} = V \times V$
- **Goal:** predict edge status for other pairs in $V^{(2)}_{miss} = V^{(2)} \setminus V^{(2)}_{obs}$

Problem statement

- Graph $G(V, E)$ with adjacency matrix $\mathbf{A}$ $\Rightarrow$ $\mathbf{A}^{obs}$ and $\mathbf{A}^{miss}$ denote entries in $V_{obs}^{(2)}$ and $V_{miss}^{(2)}$

Link prediction

Predict entries in $\mathbf{A}^{miss}$, given observations $\mathbf{A}^{obs}$ and possibly various vertex attributes $\mathbf{X} \in \mathbb{R}^{N_v \times F}$

- Link information may be missing due to:
  - $\Rightarrow$ Difficulty in observation, issues of sampling
  - $\Rightarrow$ Edge is not yet present, wish to predict future status
- Given a model for $\mathbf{X}$ and $\{\mathbf{A}^{obs}, \mathbf{A}^{miss}\}$, jointly predict $\mathbf{A}^{miss}$ based on

$$\mathrm{P}\left(\mathbf{A}^{miss} \,\middle|\, \mathbf{A}^{obs}, \mathbf{X}\right)$$

$\Rightarrow$ More manageable to predict the variables A_{ij}^{miss} individually

Informal scoring methods

- Idea: compute handcrafted score $s(i,j)$ for missing 'potential edges' $\{i,j\} \in V_{miss}^{(2)}$

  $\Rightarrow$ Predicted edges returned by retaining the top n^* scores

- Scores designed to assess certain local structural properties of G^{obs}

  $\Rightarrow$ Distance-based, inspired by the small-world principle

$$s(i,j) = -\text{dist}_{G^{obs}}(i,j)$$

  $\Rightarrow$ Neighborhood-based, e.g., the number of common neighbors

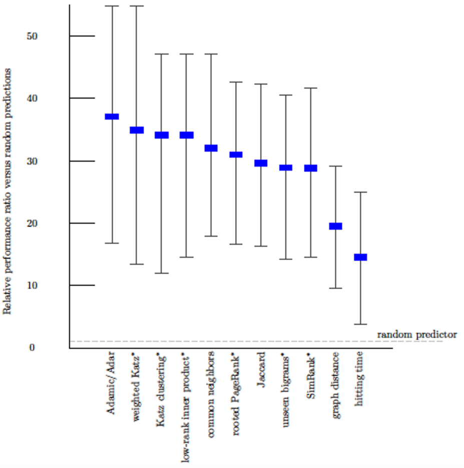
$$s(i,j) = |\mathcal{N}_i^{obs} \cap \mathcal{N}_j^{obs}| \;\text{ or }\; s(i,j) = \frac{|\mathcal{N}_i^{obs} \cap \mathcal{N}_j^{obs}|}{|\mathcal{N}_i^{obs} \cup \mathcal{N}_j^{obs}|}$$

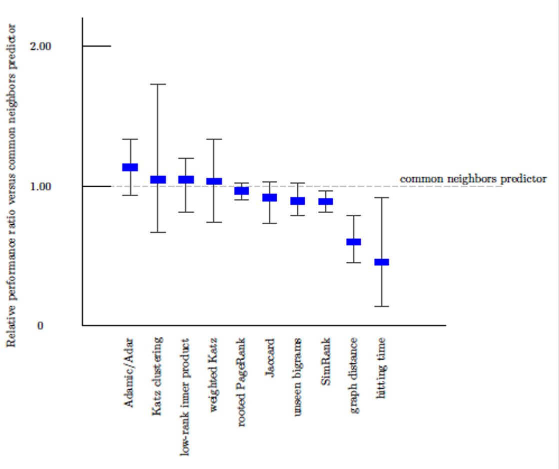
  $\Rightarrow$ Favor loosely-connected common neighbors

$$s(i,j) = \sum_{k \in \mathcal{N}_i^{obs} \cap \mathcal{N}_j^{obs}} \frac{1}{\log |\mathcal{N}_k^{obs}|}$$

L. Adamic and E. Adar, "Friends and neighbors on the Web," *Social Networks*, 2003

Tests on co-authorship networks

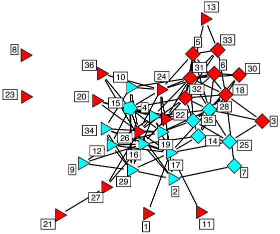




D. Liben-Nowell and J. Kleinberg, “The link-prediction problem for social networks,” *J. Assoc. Inf. Sci. Technol.*, 2007

Example: predicting lawyer collaborations

- Network G^{obs} of working relationships among lawyers
  - Nodes are $N_v = 36$ partners, edges indicate partners worked together



- Data includes various node-level attributes:
  - Seniority (node labels indicate rank ordering)
  - Office location (triangle, square or pentagon)
  - Type of practice, i.e., litigation (red) and corporate (cyan)
  - Gender (three partners are female labeled 27, 29 and 34)
- Goal: predict cooperation among social actors in an organization

E Lazega, "The Social Mechanisms of Cooperation Among Peers in a Corporate Law Partnership," 2001

Shallow classification methods tested

- Define the following set of explanatory variables:

$$Z_{ij}^{(1)} = \text{seniority}_i + \text{seniority}_j, \quad Z_{ij}^{(2)} = \text{practice}_i + \text{practice}_j$$

$$Z_{ij}^{(3)} = \mathbb{I}\left\{\text{practice}_i = \text{practice}_j\right\}, \quad Z_{ij}^{(4)} = \mathbb{I}\left\{\text{gender}_i = \text{gender}_j\right\}$$

$$Z_{ij}^{(5)} = \mathbb{I}\left\{\text{office}_i = \text{office}_j\right\}, \quad Z_{ij}^{(6)} = |\mathcal{N}_i^{obs} \cap \mathcal{N}_j^{obs}|$$

Method 1: standard logistic regression with $Z_{ij}^{(1)}, \ldots, Z_{ij}^{(5)}$

Method 2: standard logistic regression with $Z_{ij}^{(1)}, \ldots, Z_{ij}^{(6)}$

Method 3 informal scoring method with $s(i,j) = Z_{ij}^{(6)}$

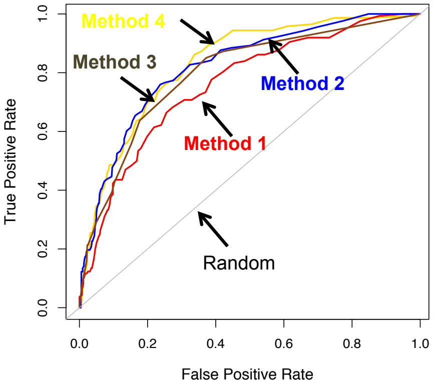
Method 4: logistic regression with $Z_{ij}^{(1)}, \ldots, Z_{ij}^{(5)}$ and latent eigenmodel

- Five-fold cross-validation over the set of $36(36-1)/2 = 630$ vertex pairs

  $\Rightarrow$ For each fold, $630/5 = 126$ pairs in $\mathbf{A}^{miss}$ and the rest in $\mathbf{A}^{obs}$

P. Hoff, "Modeling homophily and stochastic equivalence in symmetric relational data," *NeurIPS*, 2008

Receiver operating characteristic

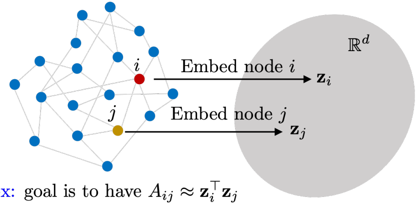


- Method 1 performs worst ⇒ Agnostic to network structure
- Informal Method 3 yields slightly worst performance than 2 and 4
- Q: Can we efficiently learn (task-independent) features for machine learning on graphs?

Graph representation learning

- Network embedding: Learn a mapping from a discrete graph to a continuous domain
- **Goal**: Given $G(V, E)$, learn (low) d-dimensional vector representation $\{\mathbf{z}_i\}_{i \in V}$

  $\Rightarrow$ Criterion is to preserve local and global graph properties

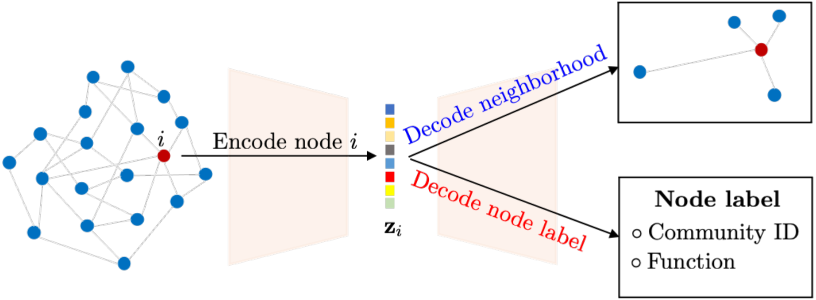


Ex: goal is to have $A_{ij} \approx \mathbf{z}_i^\top \mathbf{z}_j$

- Output is node embedding matrix $\mathbf{Z} = [\mathbf{z}_1, \ldots, \mathbf{z}_{N_v}]^\top \in \mathbb{R}^{N_v \times d}$

  $\Rightarrow$ Pick $d \ll N_v$ for scalability. Effectively a dimensionality reduction technique

An encoder-decoder perspective



W. L. Hamilton *et al*, "Representation learning on graphs: Methods and applications," *IEEE Data Engineering Bulletin*, 2018

I. Chami *et al*, "Machine learning on graphs: A model and comprehensive taxonomy," *J Mach Learn Res*, 2022

Matrix factorization methods

- **Idea:** learn low-rank representation of similarity matrix $s(\mathbf{A})$ by minimizing the loss

$$\mathcal{L}_G(\mathbf{Z}) = \|s(\mathbf{A}) - \mathbf{Z}\mathbf{Z}^\top\|_F^2$$

$\Rightarrow$ Outer product decoder: $\hat{\mathbf{A}} = \mathbf{Z}\mathbf{Z}^\top$ an inner-product approximation $[s(\mathbf{A})]_{ij} \approx \mathbf{z}_i^\top \mathbf{z}_j$

- Graph factorization (GF) preserves first-order similarity in G
  - Set $[s(\mathbf{A})]_{ij} = A_{ij}$ and evaluate $\mathcal{L}_G(\mathbf{Z})$ on $(i,j) \in V_{obs}^{(2)}$

A. Ahmed *et al*, "Distributed large-scale natural graph factorization," *WWW*, 2013

- GraRep preserves higher-order similarity in G
  - Set e.g., $[s(\mathbf{A})]_{ij} = [\mathbf{A}^k]_{ij}$, $k \geq 2$, for length-k path counts

S. Cao *et al*, "GraRep: Learning graph representations with global structural information," *CIKM*, 2015

- HOPE preserves general similarity measures in (directed) G
  - Jaccard, Adamic-Adar and related neighborhood scores

M. Ou *et al*, "Asymmetric transitivity preserving graph embedding," *KDD*, 2016

From texts to graphs: Random walk approaches

- **Idea:** asign similar $\mathbf{z}_i$ to nodes that tend to co-occur in random walks over G
- View sentences in NLP as random walks over the vocabulary

  $\Rightarrow$ Generate short random walks on G to sample node sequences

  $\Rightarrow$ Learn node positional distributions just like words in skip-gram models
- $\mathrm{P}\left(j \,\middle|\, i\right)$ of visiting j in random walk from i as similarity measure $[s(\mathbf{A})]_{ij}$ to decode from $\mathbf{Z}$

  $\Rightarrow$ Minimize cross-entropy loss

$$\mathcal{L}_G(\mathbf{Z}) = -\sum_{i,j \in V_{obs}^{(2)}} \log \hat{A}_{ij}, \quad \text{where} \quad \hat{A}_{ij} = \frac{e^{\mathbf{z}_i^\top \mathbf{z}_j}}{\sum_{k \in V} e^{\mathbf{z}_i^\top \mathbf{z}_k}}$$

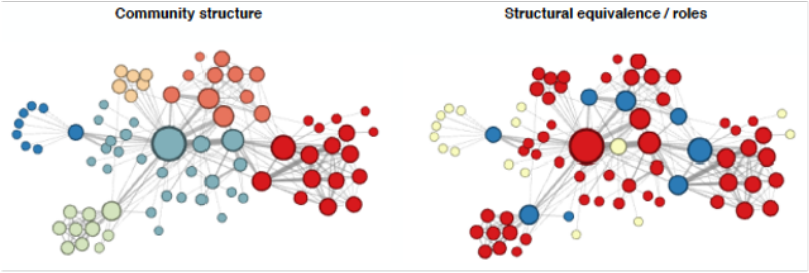
$\Rightarrow$ Implies an approximation $\hat{A}_{ij} \approx [s(\mathbf{A})]_{ij} = \mathrm{P}\left(j \,\middle|\, i\right)$

$\Rightarrow$ Evaluating the softmax denominator is challenging ($\mathcal{O}(N_v)$ complexity)

B. Perozzi *et al*, "DeepWalk: Online learning of social representations," *KDD*, 2014

Scalable representation learning via node2vec

- node2vec offers a flexible definition of (biased) random walks
  - ⇒ Smoothly interpolates between walks akin to BFS or DFS
  - ⇒ Flexibility: Effective for capturing structural roles or community structures
  - ⇒ Scalability: Approximates $\sum_{k \in V*} e^{\mathbf{z}_i^\top \mathbf{z}_k}$ via samples $V*$
- Ex: character interaction graph from the novel 'Les Miserables'



- ⇒ Left coloring indicates membership to communities (global positions)
- ⇒ Right coloring indicates roles played within (local) neighborhoods

A. Grover and J. Leskovec, "node2vec: Scalable feature learning for networks," *KDD*, 2016

Limitations GNNs overcome

- Shallow embeddings: encoder a simple embedding lookup

  $\Rightarrow$ Directly optimizes an embedding $\mathbf{z}_i$ for each node $i \in V$

- No parameter sharing between nodes in the encoder
  - Statistically inefficient: parameter sharing can act as a regularizer
  - Computationally inefficient: number of parameters is $\mathcal{O}(N_v)$
- Fails to leverage graph signals during encoding
  - Attributes highly informative w.r.t. the node's position and role in G
- Limited expressive power to capture complex non-linear structures
  - Enter (deep) NN architectures in the encoder-decoder modules
- Inherently transductive
  - Challenge for dynamic networks or large graphs not stored in memory
  - Does not generalize to other graphs beyond G (used for training)

GNNs can tackle real-world problems

Canonical tasks and examples

Link prediction: from handcrafted features to learned representations

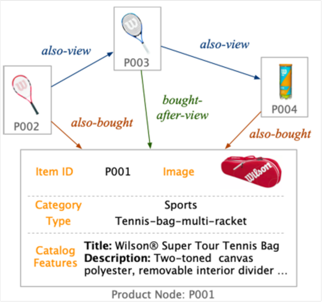
GNNs in the wild

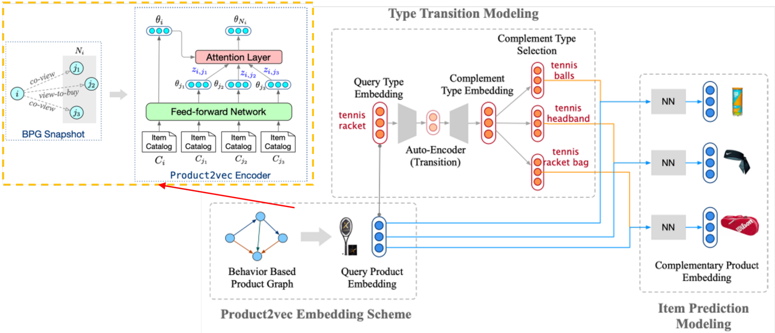
Parting words

Complementary product recommendation at Amazon

- Q: given an item purchase query, which other related and diverse set of items to recommend?
  - ⇒ Graph representation learning module on Behavior-based Product Graph
  - ⇒ Nodes are catalog items with features, edge types reflect browse and purchase behavior

Behavior-based Product Graph



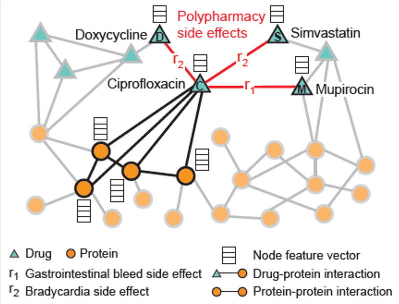


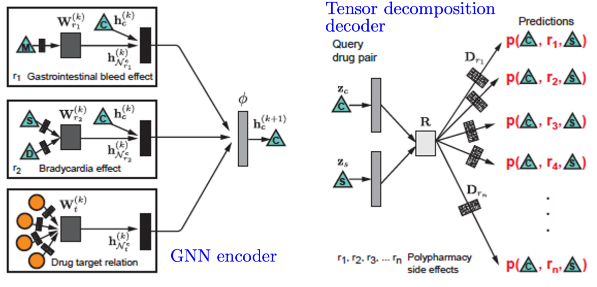
- Demonstrated business gains after production deployment at Amazon

J. Hao *et al*, "P-Companion: A principled framework for diversified complementary product recommendation," *CIKM*, 2020

Discovering polypharmacy side effects

- Multimodal graph of protein-protein, drug-protein, and drug-drug (i.e., side effect) interactions
  - ⇒ Problem: unveiling polypharmacy side effects as prediction of drug-drug edge type
  - ⇒ Multirelational link prediction in multimodal networks using GNNs


Polypharmacy side effects
Doxycycline
D
Simvastatin
S
Ciprofloxacin
C
Mupirocin
M
r_1
r_2
r_2
Drug
Protein
Node feature vector
r_1 Gastrointestinal bleed side effect
Drug-protein interaction
r_2 Bradycardia side effect
Protein-protein interaction

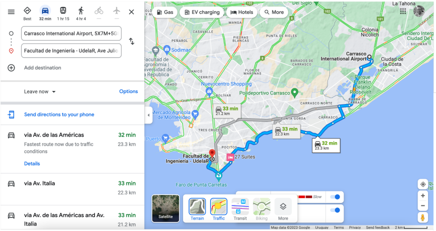


M. Zitnik *et al*, "Modeling polypharmacy side effects with graph convolutional networks," *Bioinformatics*, 2018

Vehicular traffic prediction in Google Maps

- Goal: accurate time-of-arrival (ETA) predictions from terabytes of traffic data

  ⇒ Road networks partitioned into "Supersegments" of adjacent roads sharing traffic volume



Best · 32 min · 1 hr 15 · 4 hr 4

Carrasco International Airport, 5X7M+5G

Facultad de Ingeniería - UdelaR, Ave Julio

Add destination

Leave now ▾ Options

Send directions to your phone

via Av. de las Américas 32 min
Fastest route now due to traffic conditions 23.3 km
Details

via Av. Italia 33 min
22.3 km

via Av. de las Américas and Av. Italia 33 min
21.2 km

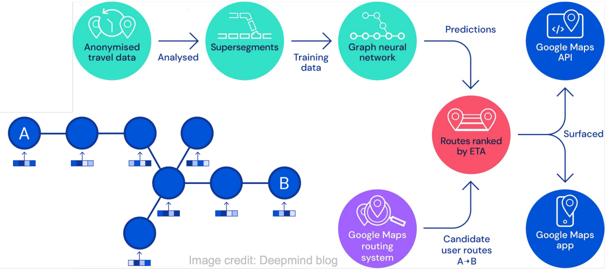


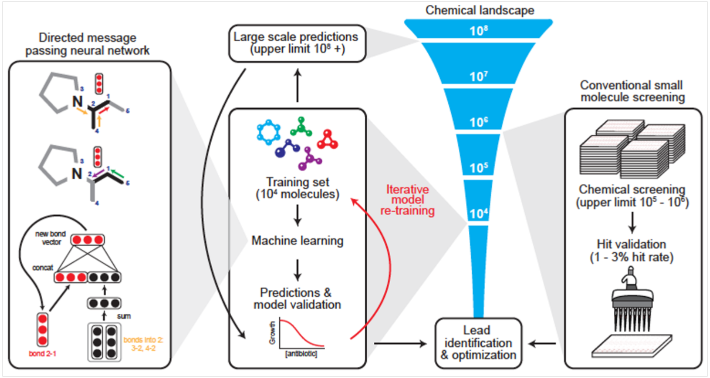
Image credit: Deepmind blog

- GNN-based ETA prediction model is an integral part of Google Maps today

A. Derrow-Pinion *et al*, "ETA prediction with graph neural networks in Google Maps," *CIKM*, 2021

Antibiotic discovery

- GNN model to predict antibiotic activity (e.g., *E. coli* growth inhibition) in molecules
  - ⇒ Train on 2335 diverse molecules, test on chemical libraries (> 107 million candidates)
  - ⇒ Conventional approaches: expensive, time-consuming, bounded exploration space

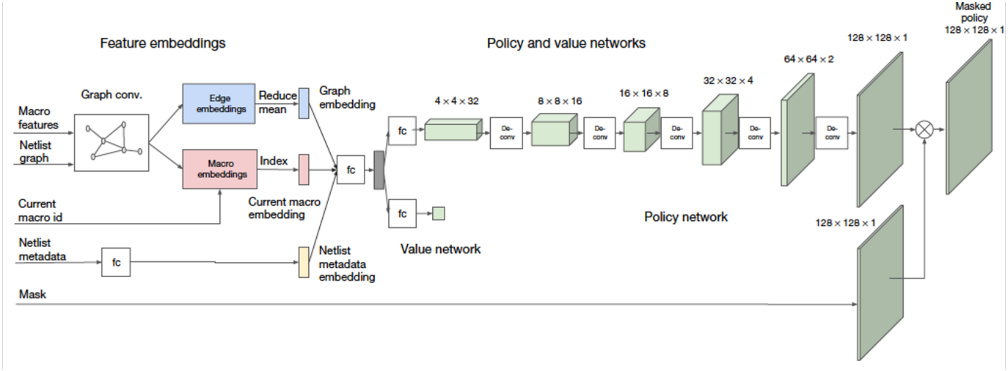

Directed message passing neural network
N
new bond vector
concat
sum
bond 2-1
bonds into 2: 3-2, 4-2
Large scale predictions (upper limit 10^8 +)
Training set (10^4 molecules)
Machine learning
Predictions & model validation
Growth
[antibiotic]
Iterative model re-training
Chemical landscape
10^8
10^7
10^6
10^5
10^4
Lead identification & optimization
Conventional small molecule screening
Chemical screening (upper limit 10^5 - 10^6)
Hit validation (1 - 3% hit rate)

- Model predicts antibacterial activity in Halicin ⇒ Structurally-different from known antibiotics

J. Stokes *et al*, "A deep learning approach to antibiotic discovery," *Cell*, 2020

Chip floorplanning for next-generation TPU design

- Chip floorplanning: engineering task of designing the physical layout of a computer chip
  - ⇒ Optimize for performance, power consumption, chip area, wirelength
- Computer chip as a *netlist*: hypergraph of modules (memory, gates) connected by wires



- Impact: used to design TPUv5, the next generation of Google's AI accelerators

A. Mirhoseini *et al*, "A graph placement methodology for fast chip design," *Nature*, 2021

Concluding remarks

Canonical tasks and examples

Link prediction: from handcrafted features to learned representations

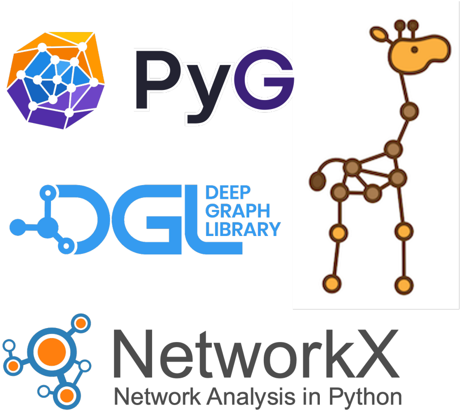
GNNs in the wild

Parting words

Useful resources

GNN and network analysis libraries

- PyTorch Geometric (PyG) – `http://pyg.org`
  - ⇒ Comprehensive interface to build GNNs
- Jraph – `http://github.com/deepmind/jraph`
  - ⇒ Lightweight library for GNNs in jax
- Deep Graph Library (DGL) – `http://www.dgl.ai`
  - ⇒ Framework agnostic DL on graphs
- NetworkX – `http://networkx.org`
  - ⇒ Create, manipulate, visualize, analyze graphs



GNN courses

- University of Pennsylvania – `http://gnn.seas.upenn.edu`
- Stanford University – `http://cs224w.stanford.edu`

The road ahead

- Graphs are natural models for relational data that can help to learn in various timely applications

  ⇒ Leveraging structure is necessary for scalable learning

- Host of fundamental and task/application-specific opportunities

  ⇒ Generative models for graphs (auto-regressive, normalizing flows, difussion)

  ⇒ Representational power, generalization, adversarial robustness

  ⇒ Tackling combinatorial optimization problems

  ⇒ Distributed learning and control in multi-agent systems