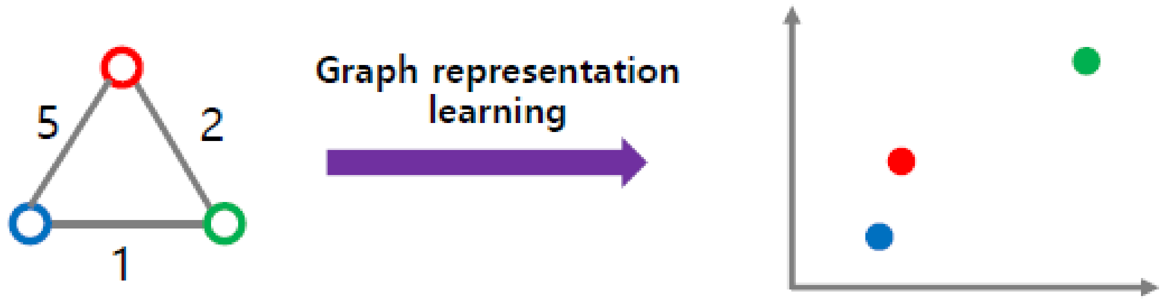
Graph Representation Learning

The University of Texas at Dallas
Kaiyuan Zhang
Oct 13, 2019

Definition

- **Graph representation learning** tries to embed each node of a graph into a low-dimensional vector space, which preserves the structural similarities or distances among the nodes in the original graph
- Also known as **network embedding** / **graph embedding** / **network representation learning**



Why graphs?

Graphs are a general language for describing and modeling complex systems.

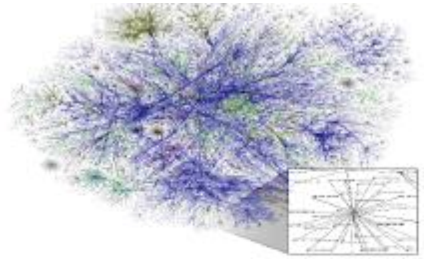
Why graphs?

- Universal language for describing complex data
  - Networks from science, nature, and technology are more similar than one would expect
- Shared vocabulary between fields
  - Computer Science, Social science, Physics, Economics, Statistics, Biology
- Data availability (+computational challenges)
  - Web/mobile, bio, health, and medical
- Impact!
  - Social networking, Social media, Drug design

Adapt from Representation Learning on Networks, snap.stanford.edu/proj/embeddings-www, WWW 2018

Graphs: general and flexible data structures

- Ubiquitous in real-world, arises in multiple disciplines
  - computer science, social science, healthcare, bioinformatics, ...



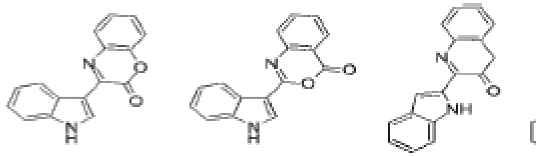
World Wide Web

Social Network

Road Graph

Knowledge Graph

Protein-protein Interaction Graph

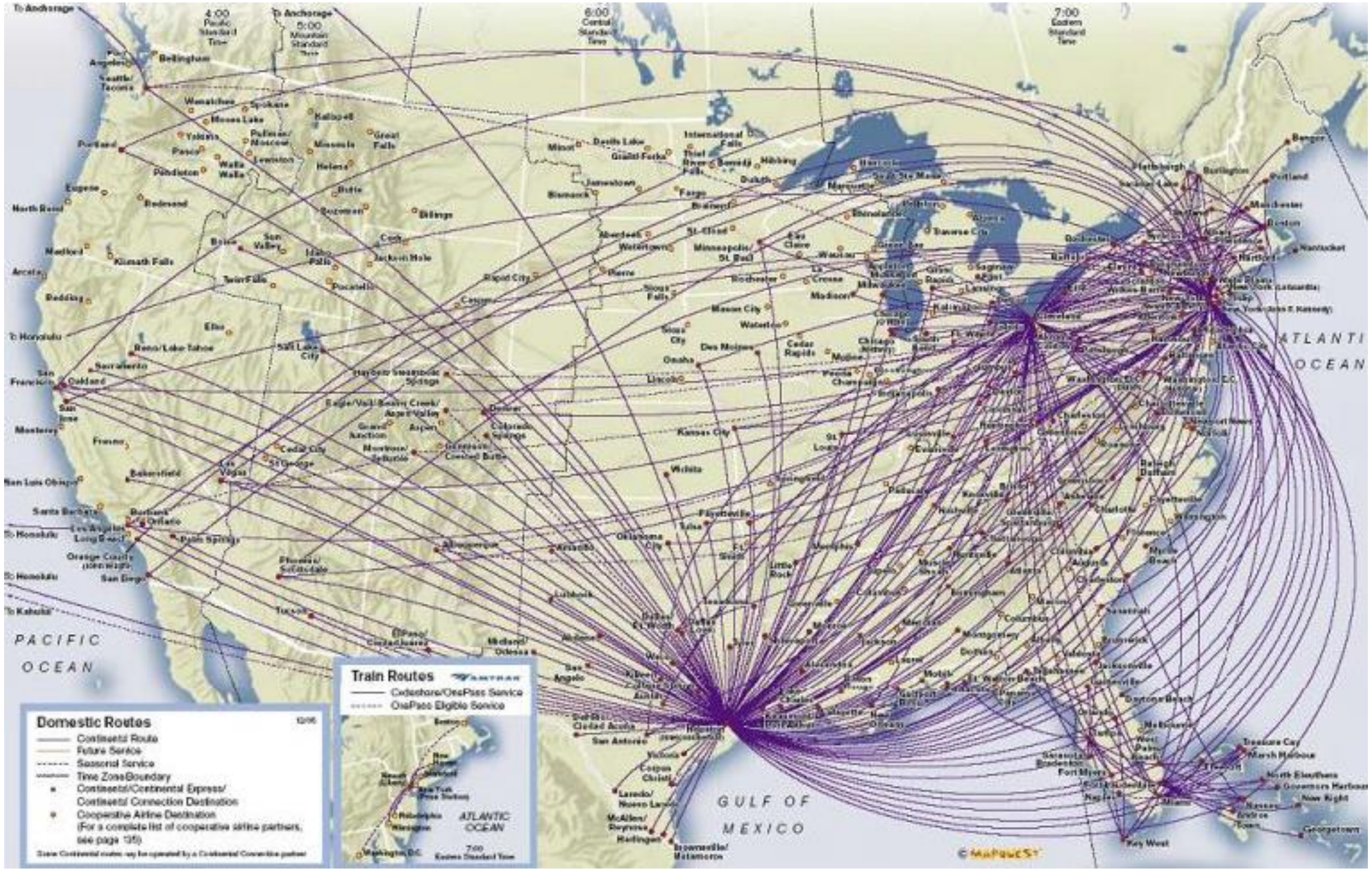


1 2 3

Molecular structures

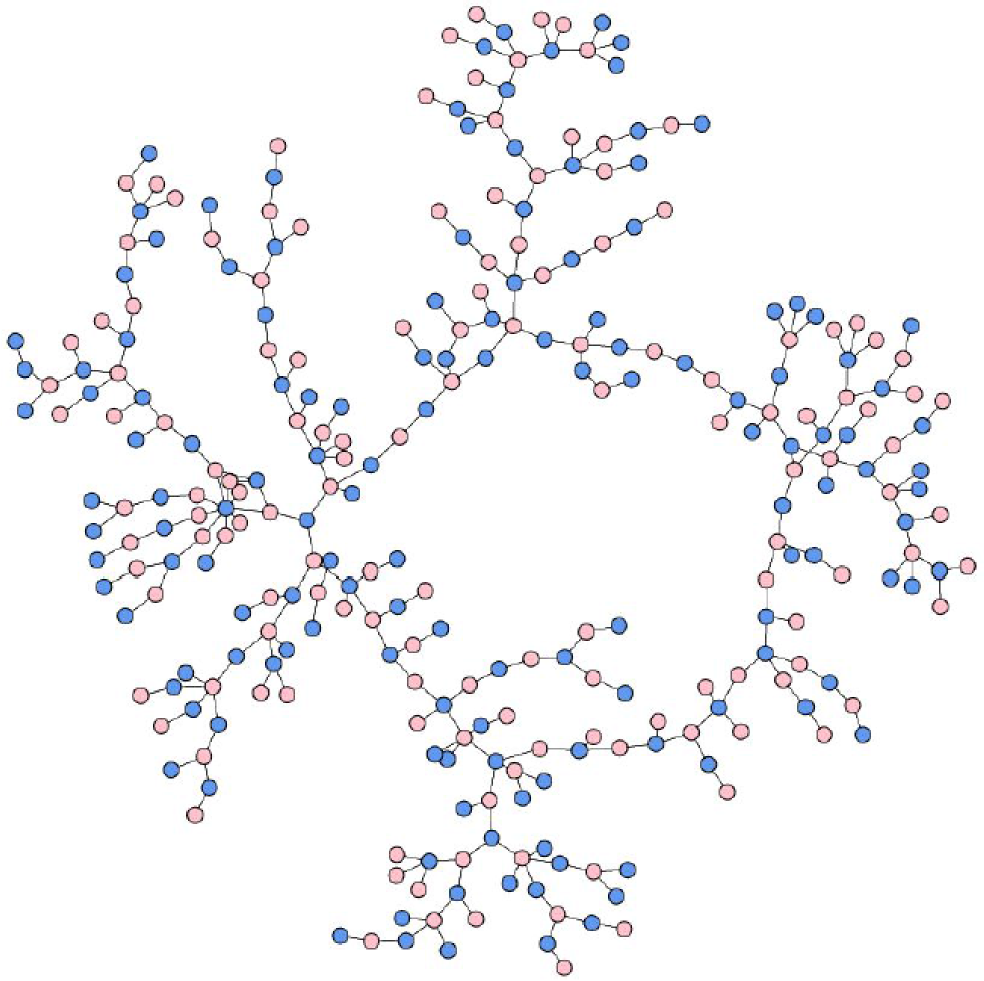
- Many data can be formulated as graphs
  - Images as graphs with two-dimensional grid structures

Applications



Continental Airline: Source: http://www.airlineroutemaps.com/

Applications



Peter S. Bearman, James Moody and Katherine Stovel Chains of affection: The structure of adolescent romantic and sexual Graphs , American Journal of Sociology 110 44-91 (2004)

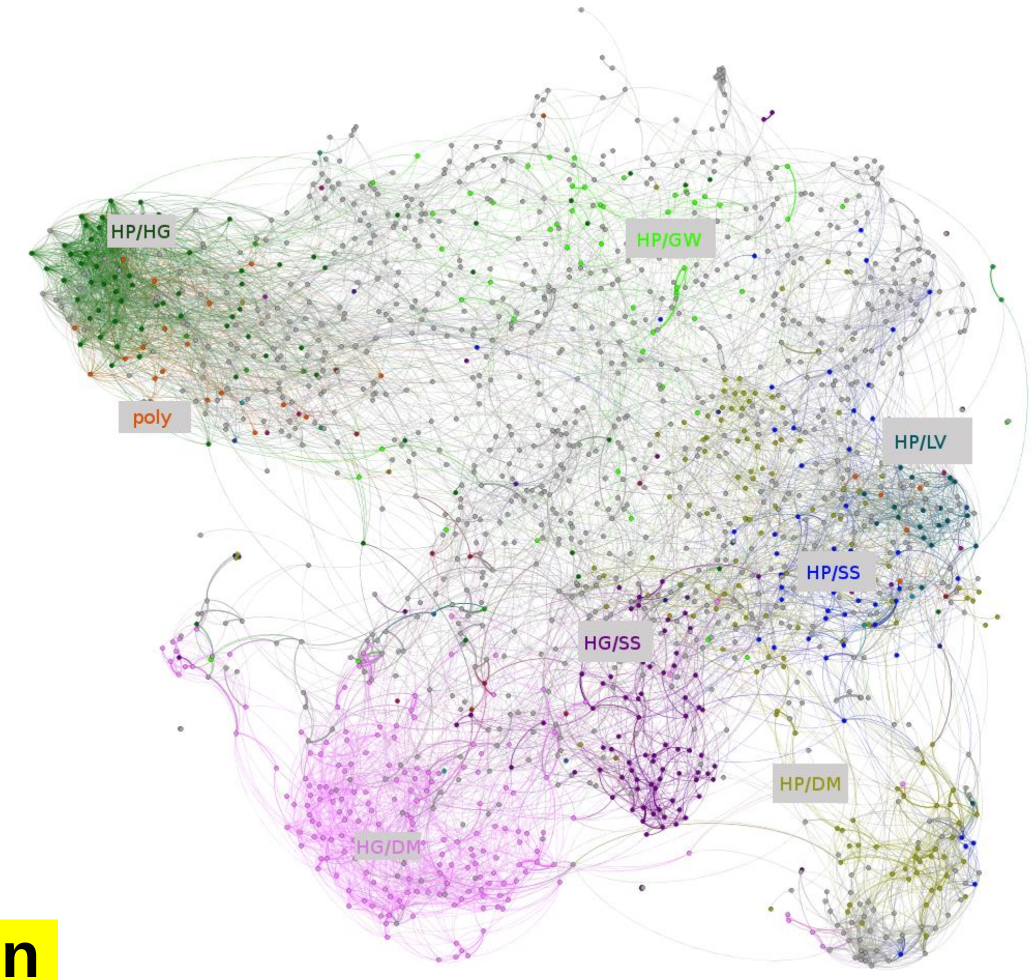
Applications

Graph representation learning can benefit a wide range of real-world applications:

- Link prediction (Gao, Denoyer, and Gallinari, CIKM 2011)
- Node classification (Tang, Aggarwal, and Liu, SDM 2016)
- Recommendation (Yuet al., WSDM 2014)
- Visualization (Maatenand Hinton, JMLR 2008)
- Knowledge graph representation (Lin et al., AAAI 2015)
- Clustering (Tian et al., AAAI 2014)
- Text embedding (Tang, Qu, and Mei, KDD 2015)
- Social network analysis (Liu et al., IJCAI 2016)

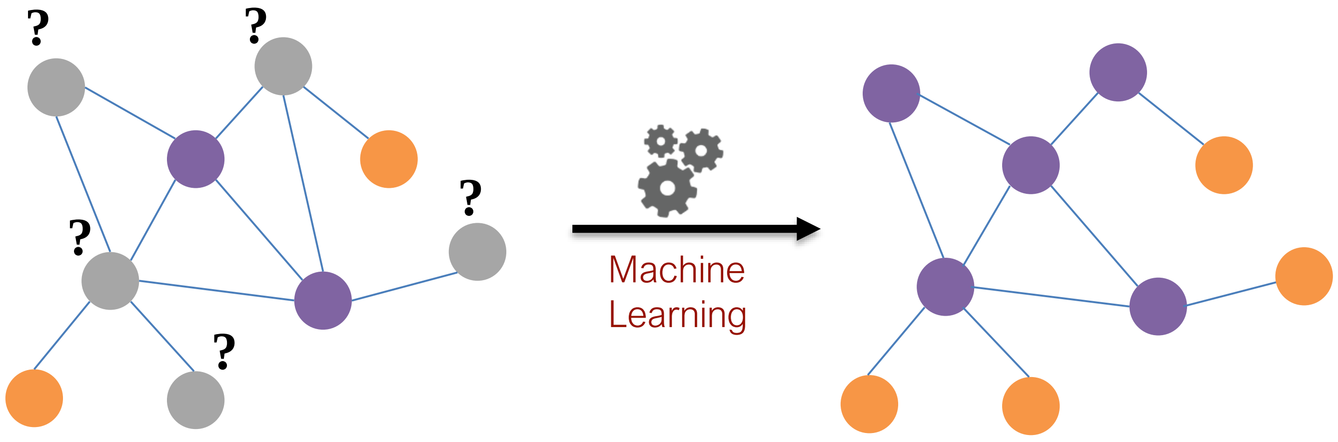
Classical ML tasks in graphs

- Node classification
  - Predict a type of a given node
- Link prediction
  - Predict whether two nodes are linked
- Community detection
  - Identify densely linked clusters of nodes
- Network similarity
  - How similar are two (sub)networks…



Graph embedding have vital impact on various real applications.

Example: Node Classification



Example: Node Classification

Classifying the function of proteins in the interactome!

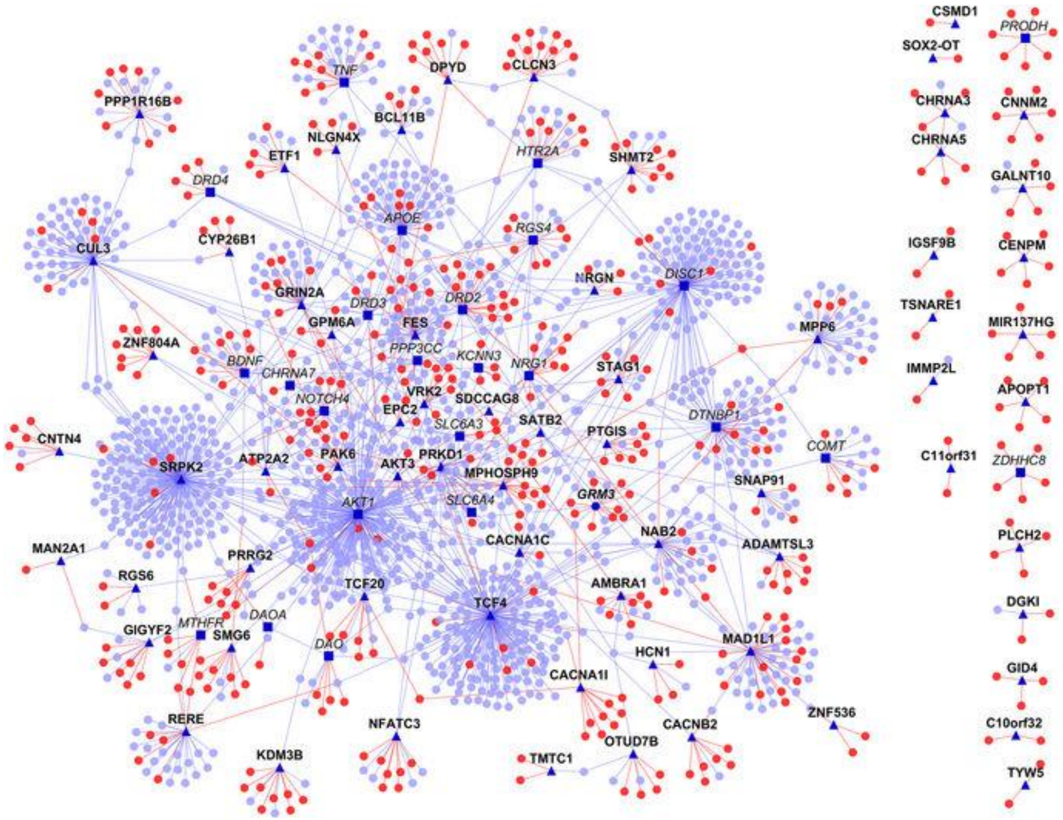
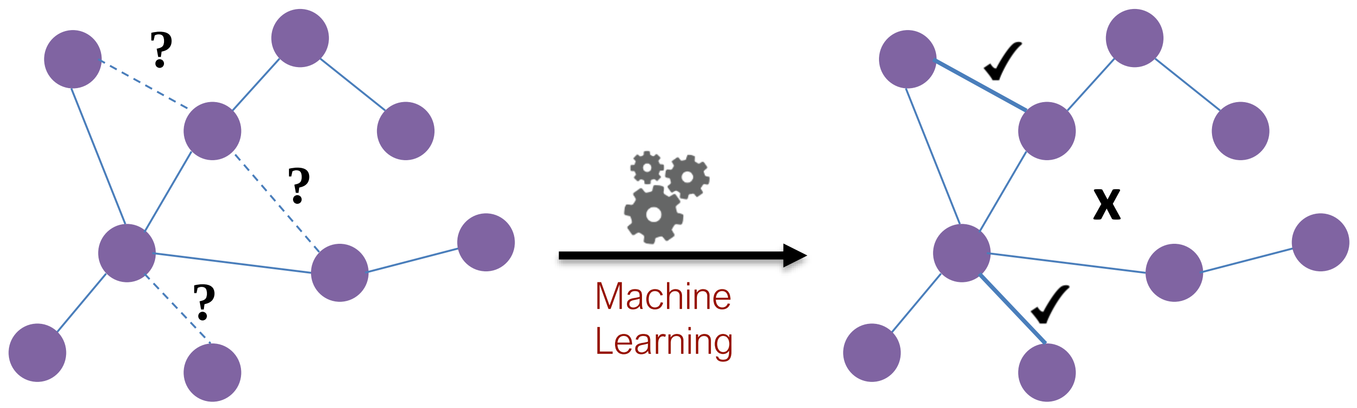


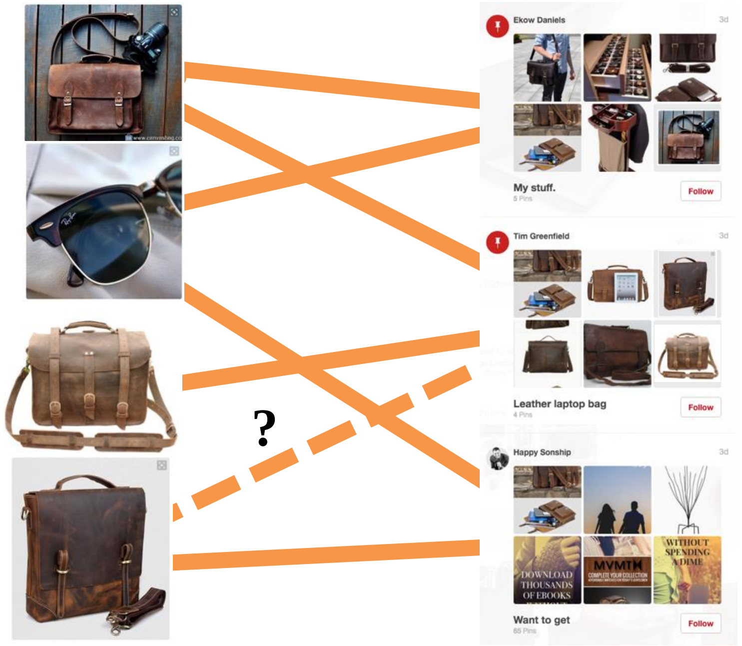
Image from: Ganapathiraju et al. 2016. [Schizophrenia interactome with 504 novel protein–protein interactions](). *Nature*.

Example: Link Prediction



Example: Link Prediction

Content recommendation is link prediction!



The voices from industry…

Social Graphs

Facebook: ~2 billion active users

Wechat: ~1 billion active users

E-commerce Graphs

Amazon: 400M active customers, 400M products

Taobao: 500M customers, 800M products

The Graph grows to such a scale that no sophisticated Graph analytics is doable.

Adapt From KDD 2018 graph representation tutorial

Taxonomy

Taxonomy graph representation learning
in three ways:

1. By Input
2. By Output
3. By Method

Taxonomy (1/3)

Taxonomy graph representation learning in three ways:

1. By Input
2. By Output
3. By Method

Taxonomy (1/3)

Input

1. Homogeneous graph (e.g., citation network)
   - Weighted / Unweighted
   - Directed / Undirected
   - Signed / Unsigned
2. Heterogeneous graph
   - Multimedia network
   - Knowledge graph
3. Graph with side information
   - Node/edge label (categorical)
   - Node/edge attribute (discrete or continuous)
   - Node feature (e.g., texts)
4. Graph transformed from non-relational data
   - Manifold learning

Taxonomy (2/3)

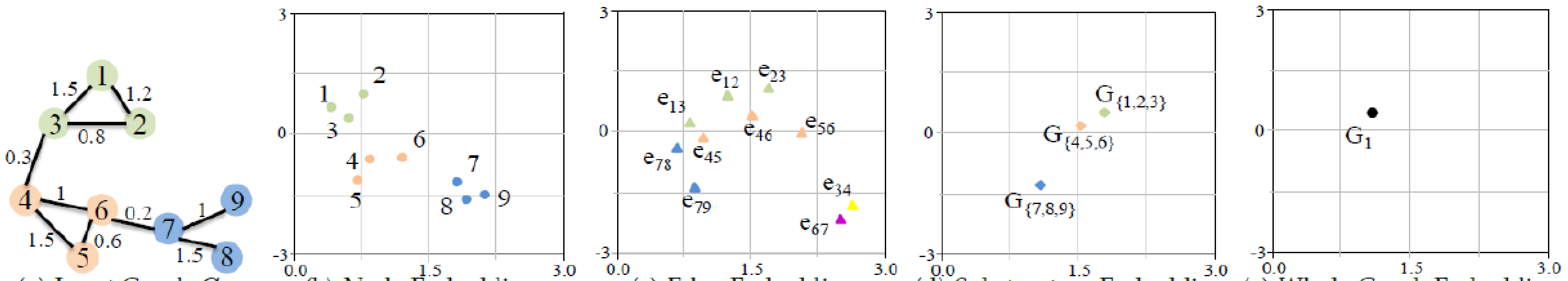
Taxonomy graph representation learning in three ways:

1. By Input
2. By Output
3. By Method

Taxonomy (2/3)

Output

1. Node embedding (the most common case)
2. Edge embedding
   - ➢ Relations in knowledge graph
   - ➢ Link prediction
3. Sub-graph embedding
   - ➢ Substructure embedding
   - ➢ Community embedding
4. Whole-graph embedding
   - ➢ Multiple small graphs, e.g., molecule, protein



(a) Input Graph G_1 (b) Node Embedding (c) Edge Embedding (d) Substructure Embedding (e) Whole-Graph Embedding

Adapt from Graph Representation Learning and GraphGAN by Hongwei Wang

Taxonomy (3/3)

Taxonomy graph representation learning in three ways:

1. By Input
2. By Output
3. By Method

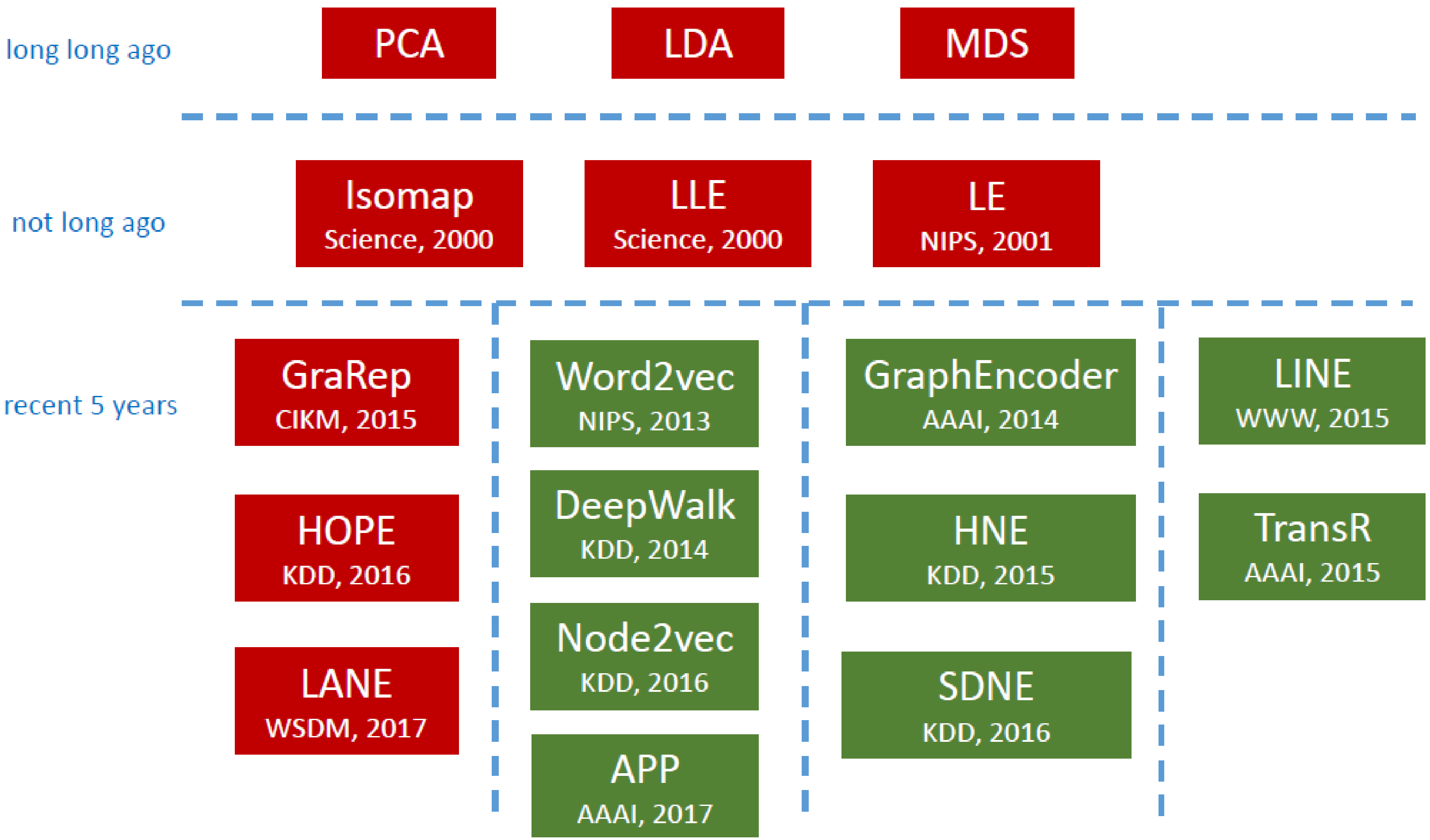
Taxonomy (3/3)

Method

- Traditional dimension reduction methods
  - **PCA** (principle component analysis)
  - **LDA** (linear discriminant analysis)
  - **MDS** (multiple dimensional scaling)
- Manifold Learning methods
  - Isomap (isometric mapping) [Science 2000]
  - LLE (locally linear embedding) [Science 2000]
  - LE (Laplacian eigenmaps) [NIPS 2001]
- Random-walk-based methods
  - DeepWalk [KDD 2014]
  - Node2vec [KDD 2016]
  - LINE (large-scale information network embedding) [WWW 2015]
- Deep-learning-based methods
  - SDNE (structural deep network embedding) [KDD 2016]
  - HNE (heterogeneous network embedding) [KDD 2015]
  - Gated Graph Neural Networks [ICLR 2016]
  - GCN (Graph Convolutional Networks) [ICLR 2017]
  - GraphSAGE (sample and aggregate) [NIPS 2017]
  - Graph Attention Networks [ICLR 2018]
  - ….

Adapt from Graph Representation Learning and GraphGAN by Hongwei Wang

Representative Work



Adapt from Graph Representation Learning and GraphGAN by Hongwei Wang

Many Extensions …

- Leverage global structural information (Cao et al. 2015)
- Non-linear methods based on autoencoders (Wang et al. 2016)
- Matrix-factorization based approaches (Qiu et al. 2018)
- Directed network embedding (Ou et al. 2016)
- Signed network embedding (Wang et al. 2017)
- Multi-view networks (Qu and Tang et al. 2017)
- Networks with node attributes (Yang et al. 2015)
- Heterogeneous networks (Chang et al. 2015)
- Task-specific network embedding (Chen et al. 2017)
- …

SEMI-SUPERVISED CLASSIFICATION WITH GRAPH CONVOLUTIONAL NETWORKS

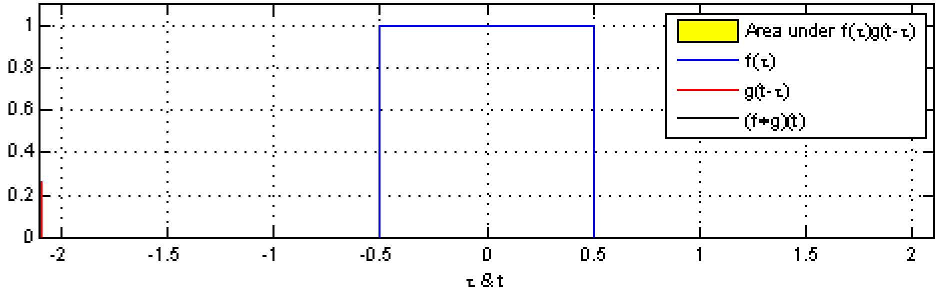
Thomas N. Kipf, Max Welling
University of Amsterdam

CNN

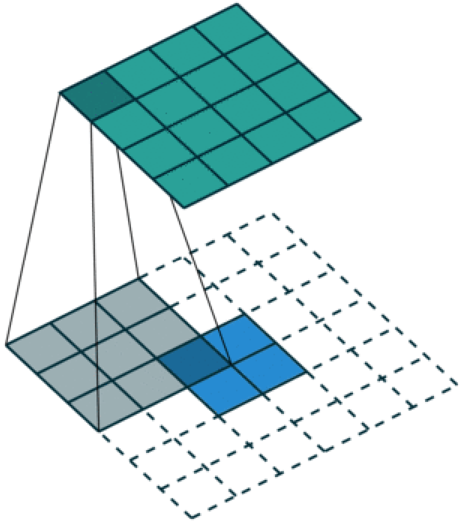
Convolution:

$$(f * g)(t) = \int_{\mathbb{R}} f(x)g(t-x)dx$$

1D convolution:

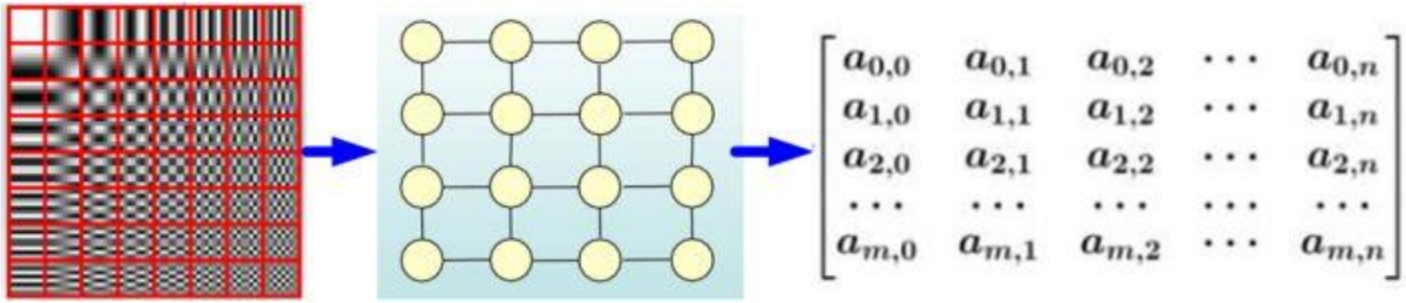


2D convolution:



CNN

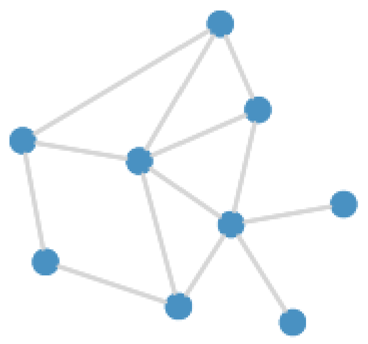
Euclidean Structure:



How to deal with Non Euclidean Structure?

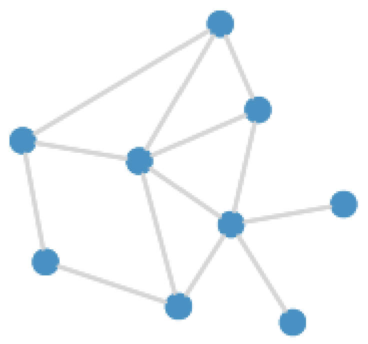
Graph-structured data

What if our data looks like this?

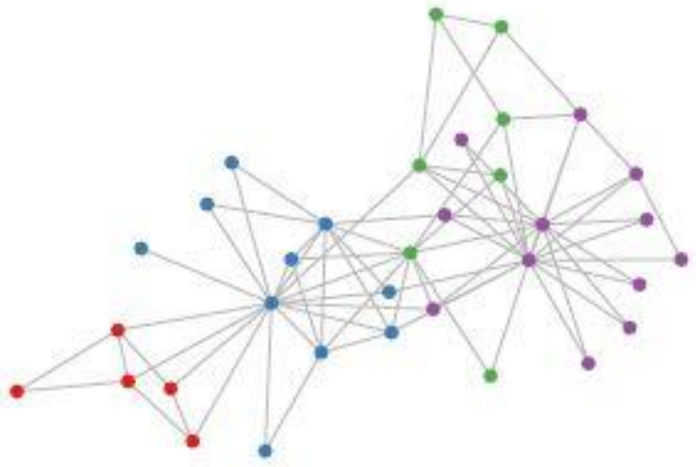


Graph-structured data

What if our data looks like this?

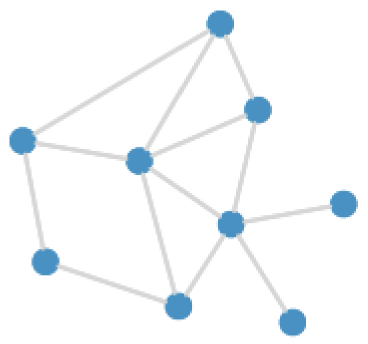


Or this:

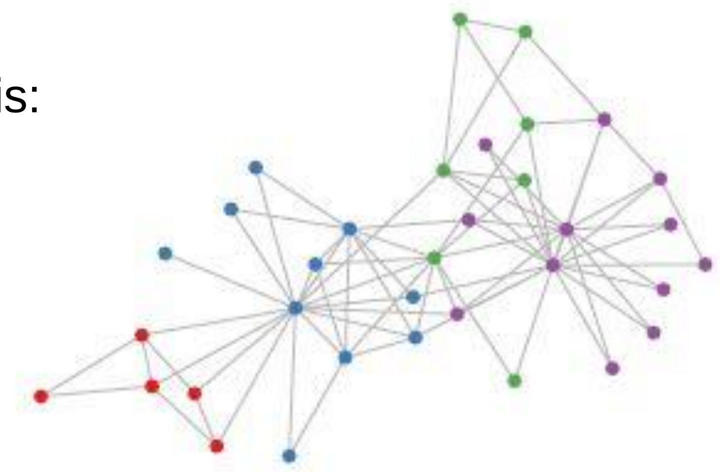


Graph-structured data

What if our data looks like this?



Or this:



Real-world examples:

- Social networks
- World-wide-web
- Protein-interaction networks
- Telecommunication networks
- Knowledge graphs
- …

Adapt from Thomas Kipf’s talk

Motivation

GRAPH CONVOLUTIONAL NETWORKS

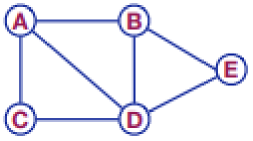
Non Euclidean Structure

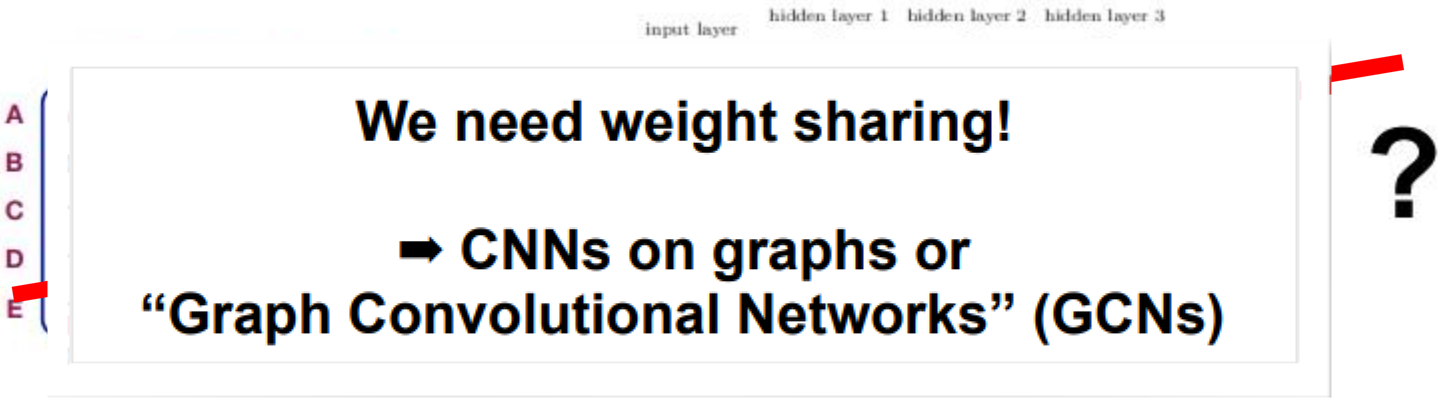
Then, how to do GCN?

Before we started, take a look at a naïve approach…

A naïve approach

- Take adjacency matrix $\mathbf{A}$ and feature matrix $\mathbf{X}$
- Concatenate them $\mathbf{X}_{\text{in}} = [\mathbf{A}, \mathbf{X}]$
- Feed them into deep (fully connected) neural net
- Done?





We need weight sharing!

➡ CNNs on graphs or "Graph Convolutional Networks" (GCNs)

Problems:

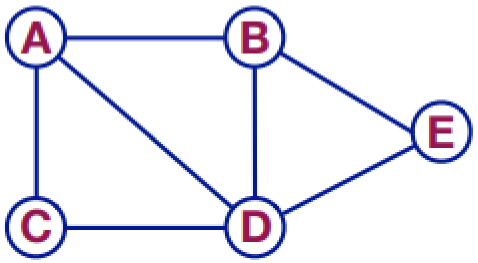
- Huge number of parameters $\mathcal{O}(N)$
- Re-train if graph changes

Adapt from Thomas Kipf's talk

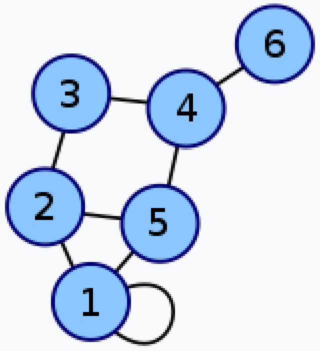
Graph-structured data

$\mathcal{G} = (\mathcal{V}, \mathcal{E}) \quad v_i \in \mathcal{V}$, edges $(v_i, v_j) \in \mathcal{E}$

A: adjacency matrix
D: degree matrix
L: Laplacian matrix



$$\begin{array}{c} \\ A \\ B \\ C \\ D \\ E \end{array} \begin{array}{c} \begin{array}{ccccc} A & B & C & D & E \end{array} \\ \begin{bmatrix} 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 \end{bmatrix} \end{array}$$

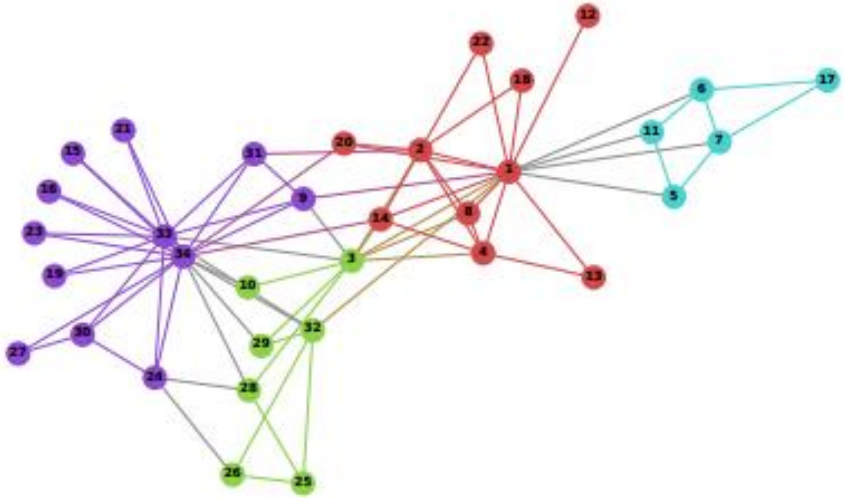
Vertex labeled graph	Degree matrix
	$\begin{pmatrix} 4 & 0 & 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$

Adapt from Thomas Kipf's talk

GCN

Some Preparations:

- Graph Fourier Transformation
- Laplace operator



Graph Fourier Transformation

- Fourier transform:

$$\mathcal{F}\{f\}(v) = \int_{\mathbb{R}} f(x) e^{-2\pi i x \cdot v} dx$$

- Inverse Fourier transform:

$$\mathcal{F}^{-1}\{f\}(x) = \int_{\mathbb{R}} f(v) e^{2\pi i x \cdot v} dv$$

- Assume:

$$h(z) = \int_{\mathbb{R}} f(x) g(z - x) dx$$

- Then,

$$f * g = \mathcal{F}^{-1}\{\mathcal{F}\{f\} \cdot \mathcal{F}\{g\}\}$$

Laplace operator

$$f'(x) = \lim_{h \to 0} \frac{f(x+h) - f(x)}{h}$$

$$f'_{*g}(x) = f(x) - f(y)$$

$$\Delta f(x) = \lim_{h \to 0} \frac{f(x+h) - 2f(x) + f(x-h)}{h^2}$$

$$\Delta_{*g} f'(x) = \Sigma_{y \sim x} f(x) - f(y)$$

Graph convolution

- Laplacian matrix:

$$L = D - A$$

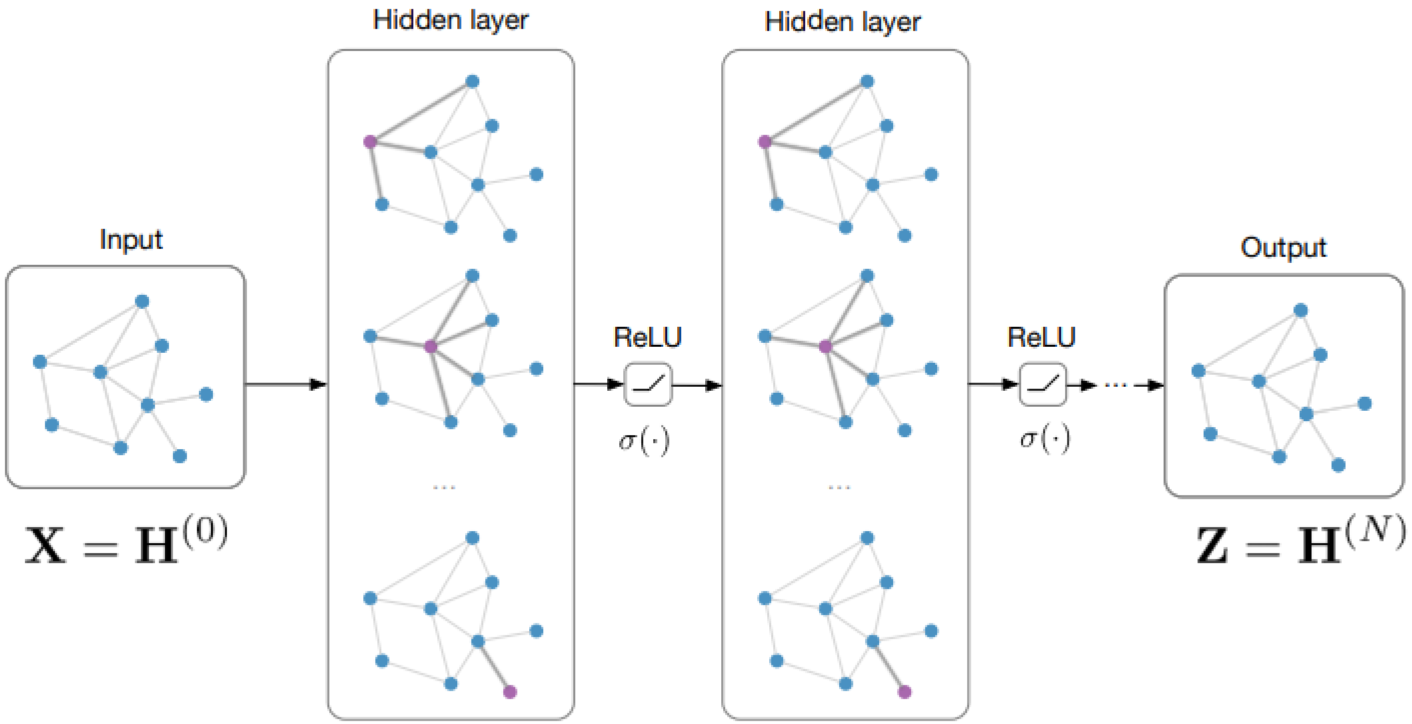
- Graph convolution:

$$g_\theta * x = U g_\theta U^T x = U g_{\theta'}(\Lambda) U^T x$$

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

GCN model architecture

Input: Feature matrix $\mathbf{X} \in \mathbb{R}^{N \times E}$, preprocessed adjacency matrix $\hat{\mathbf{A}}$



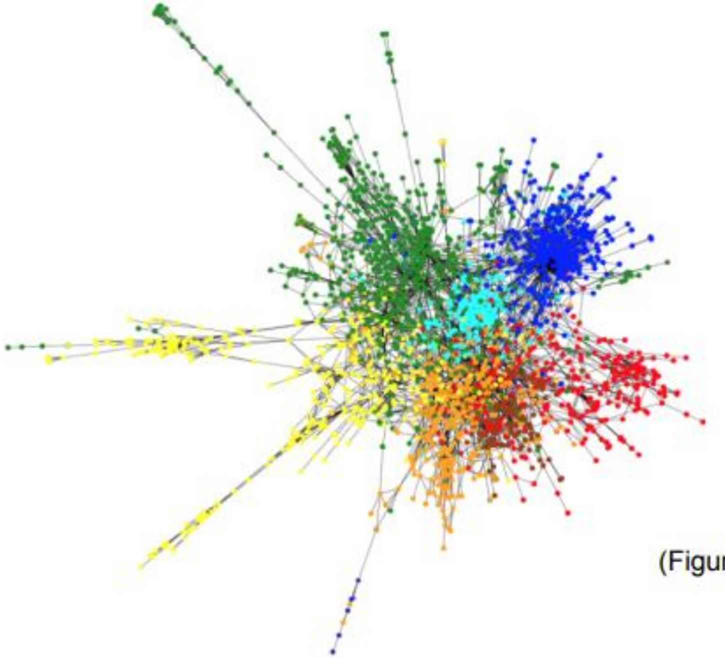
$$\mathbf{H}^{(l+1)} = \sigma\left(\hat{\mathbf{A}}\mathbf{H}^{(l)}\mathbf{W}^{(l)}\right)$$

Adapt from Thomas Kipf's talk

Application: Classification on citation networks

Input:
Citation networks

Target:
Paper category (e.g. stat.ML, cs.LG, …)



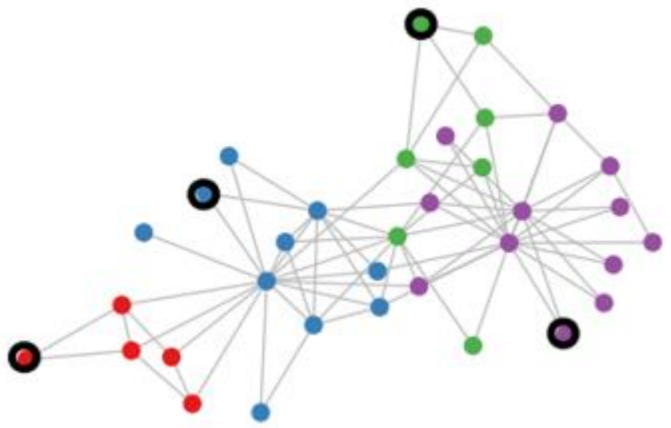
(Figure from: Bronstein, Bruna, LeCun, Szlam, Vandergheynst, 2016)

Adapt from Thomas Kipf's talk

Semi-supervised classification on graphs

Setting:
Some nodes are labeled (black circle)
All other nodes are unlabeled

Task:
Predict node label of unlabeled nodes



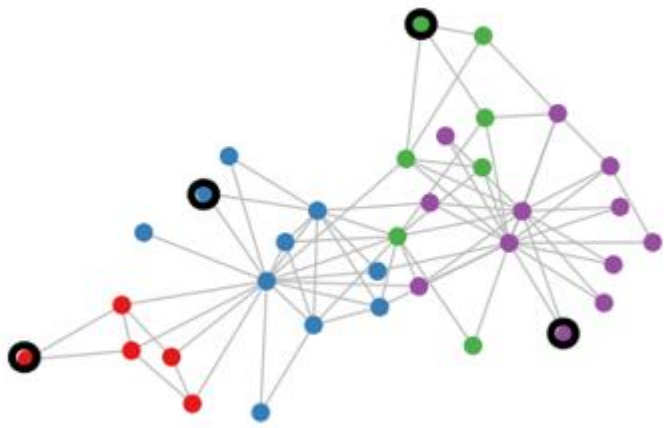
Adapt from Thomas Kipf's talk

Semi-supervised classification on graphs

Setting:
Some nodes are labeled (black circle)
All other nodes are unlabeled

Task:
Predict node label of unlabeled nodes

Idea:
Train graph-based classifier end-to-end using GCN



Evaluate loss on labeled nodes only:

$$\mathcal{L} = -\sum_{l \in \mathcal{Y}_L} \sum_{f=1}^{F} Y_{lf} \ln Z_{lf}$$

$\mathcal{Y}_L$ set of labeled node indices

$\mathbf{Y}$ label matrix

$\mathbf{Z}$ GCN output (after softmax)

Adapt from Thomas Kipf's talk

Experiment data

Dataset statistics

Dataset	Type	Nodes	Edges	Classes	Features	Label rate
Citeseer	Citation network	3,327	4,732	6	3,703	0.036
Cora	Citation network	2,708	5,429	7	1,433	0.052
Pubmed	Citation network	19,717	44,338	3	500	0.003
NELL	Knowledge graph	65,755	266,144	210	5,414	0.001

Experiment results

Summary of results in terms of classification accuracy (in percent)

Method	Citeseer	Cora	Pubmed	NELL
ManiReg [3]	60.1	59.5	70.7	21.8
SemiEmb [28]	59.6	59.0	71.1	26.7
LP [32]	45.3	68.0	63.0	26.5
DeepWalk [22]	43.2	67.2	65.3	58.1
ICA [18]	69.1	75.1	73.9	23.1
Planetoid* [29]	64.7 (26s)	75.7 (13s)	77.2 (25s)	61.9 (185s)
GCN (this paper)	**70.3** (7s)	**81.5** (4s)	**79.0** (38s)	**66.0** (48s)
GCN (rand. splits)	67.9 ± 0.5	80.1 ± 0.5	78.9 ± 0.7	58.4 ± 1.7

References

Kipf & Welling, Semi-Supervised Classification with Graph Convolutional Networks, ICLR' 17:
https://arxiv.org/abs/1609.02907
Blog post Graph Convolutional Networks:
http://tkipf.github.io/graph-convolutional-networks
Code on Github:
http://github.com/tkipf/gcn
Hongwei Wang et.al GraphGAN: Graph Representation Learning with Generative Adversarial Nets, AAAI' 18
https://arxiv.org/abs/1711.08267
KDD 2018 Graph Representation Tutorial:
https://ivanbrugere.github.io/kdd2018/
WWW 2018 Representation Learning on Networks Tutorial:
http://snap.stanford.edu/proj/embeddings-www/
AAAI 2019 Graph Representation Learning Tutorial
https://jian-tang.com/files/AAAI19/aaai-grltutorial-part0-intro.pdf

Thank you!

You can get in touch with me via:

E-Mail: zhangkaiyuan20@gmail.com

Web: https://kaiyuanzhang.com