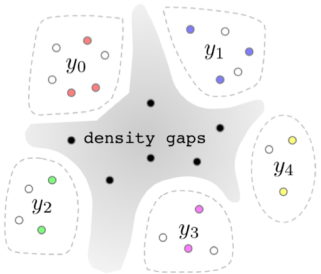
Learning From Networks

——*Algorithms, Theory, & Applications*

Xiao Huang, Peng Cui, Yuxiao Dong, Jundong Li, Huan Liu, Jian Pei, Le Song,
Jie Tang, Fei Wang, Hongxia Yang, Wenwu Zhu

xhuang@tamu.edu; cuip@tsinghua.edu.cn; yuxdong@microsoft.com; jundongl@asu.edu;
huan.liu@asu.edu; jpei@cs.sfu.ca; le.song@antfin.com; jietang@tsinghua.edu.cn;
few2001@med.cornell.edu; yang.yhx@alibaba-inc.com; wwzhu@tsinghua.edu.cn;

Graph Neural Networks



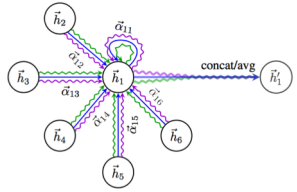
FastGCNs, Graph attention — 2018: Velickovic et al., ICLR'18, Chen et al., ICLR 2018

NetMF & NetSMF — 2018: Qiu et al., WSDM'18 & WWW'19

Neural message passing, GraphSage — 2017: Gilmer et al., ICML'17; Hamilton et al., NIPS'17

Gated graph neural network — 2016: Li et al., ICLR'16

structure2vec — 2016: Dai et al., ICML'16



Graph convolutional network — 2015: Duvenaud et al., NIPS'15; Kipf & Welling ICLR'17

PTE, metapath2vec — 2015: Tang et al., KDD'15; Dong et al., KDD'17

LINE, node2vec — 2015: Tang et al., WWW'15; Grover & Leskovec, KDD'16

DeepWalk — 2014: Perozzi et al., KDD'14

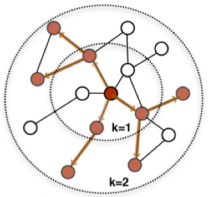
Spectral graph convolution — 2014: Bruna et al., ICLR'14

word2vec (skip-gram) — 2013: Mikolov et al., ICLR'13

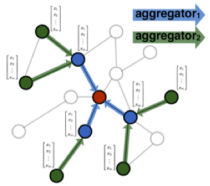
Graph neural network — 2005: Gori et al., IJCNN'05

Spectral clustering — 2000: Ng et al. & Shi, Malik

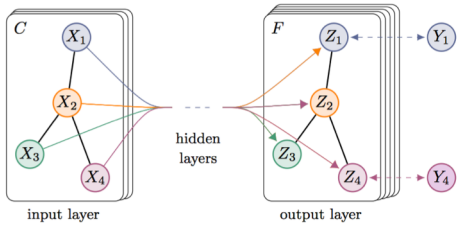
Spectral partitioning — 1973: Donath & Hoffman



1. Sample neighborhood

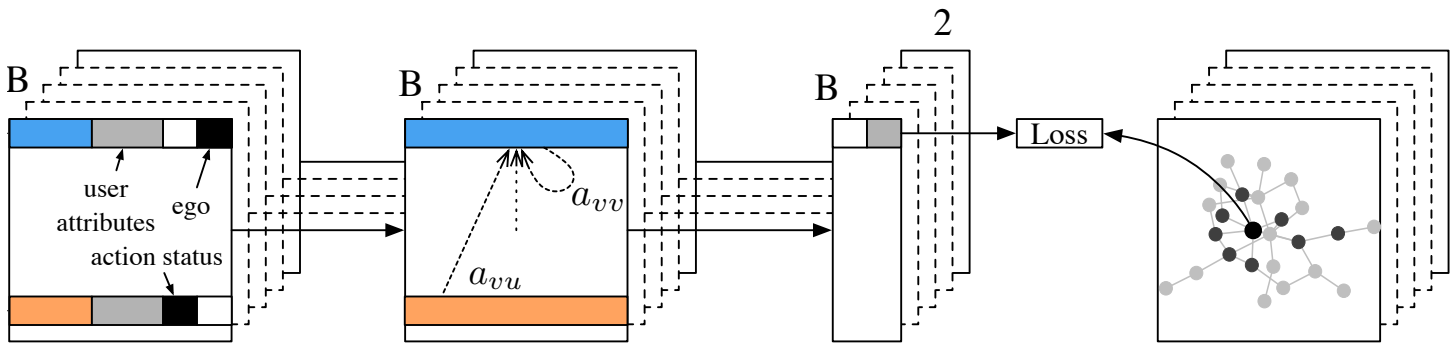


2. Aggregate feature information from neighbors



GCN

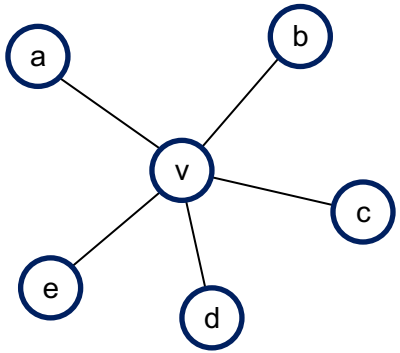
- GCNs can be considered as a simplification of the traditional graph spectral methods
- The common strategy is to model a node's neighborhood as the receptive field and then apply the *graph convolution* operation



Graph Neural Networks

- Input: an undirected weighted network $G = (V, E)$ with $|V| = n$ & $|E| = \mathrm{m}$
  - Adjacency matrix $\boldsymbol{A} \in \mathbb{R}_{+}^{n \times \mathrm{n}}$
  - $A_{i,j} = \begin{cases} a_{i,j} > 0 & (i,j) \in E \\ 0 & (i,j) \notin E \end{cases}$
  - Degree matrix $\boldsymbol{D} = diag(d_1, d_2, \cdots, d_n)$
  - Node feature matrix $\boldsymbol{X} \in \mathbb{R}^{n \times q}$
- Output: for each node, its k-dimension latent feature representation vector $\mathbf{Z}^{n \times k}$
  - –Latent feature embedding matrix $\mathbf{Z} \in \mathbb{R}^{n \times k}$

The Core of Graph Neural Networks

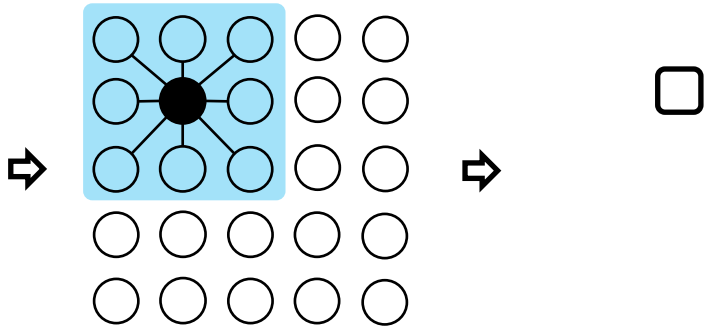


$$\boldsymbol{h}_v = f(\boldsymbol{h}_a, \boldsymbol{h}_b, \boldsymbol{h}_c, \boldsymbol{h}_d, \boldsymbol{h}_e)$$

Neighborhood Aggregation:

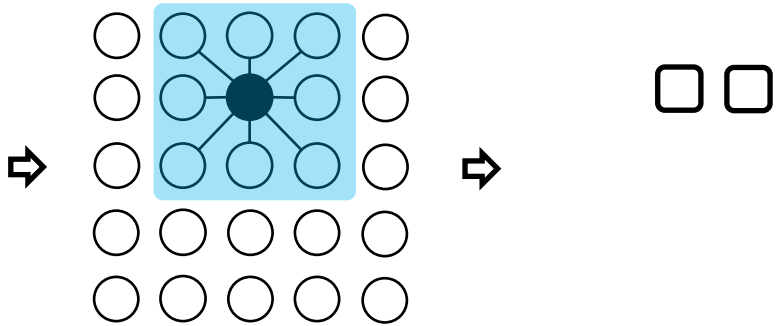
Aggregate neighbor information and pass into a neural network

Convolutional neural network



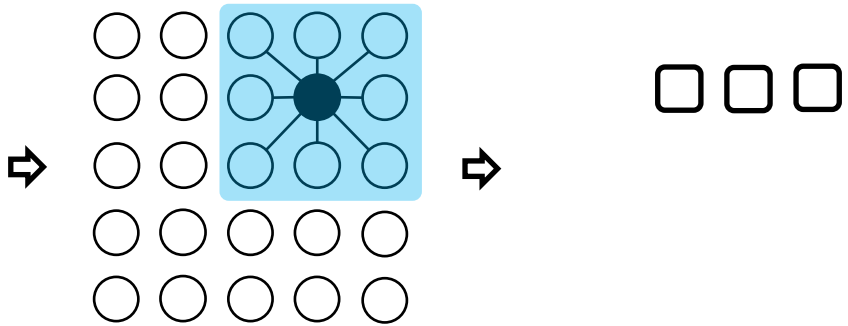
$$\sum WX$$

Convolutional neural network



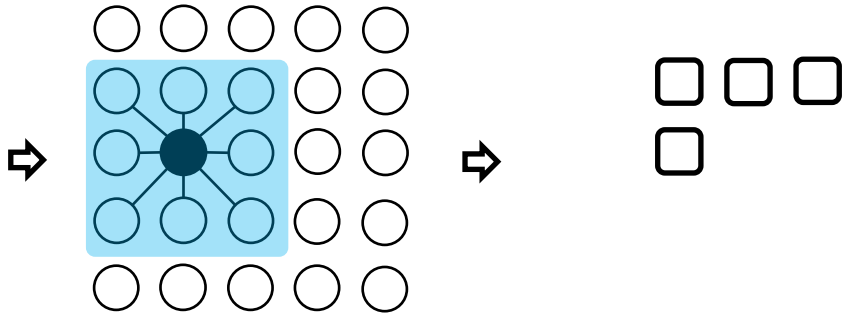
$$\sum WX$$

Convolutional neural network



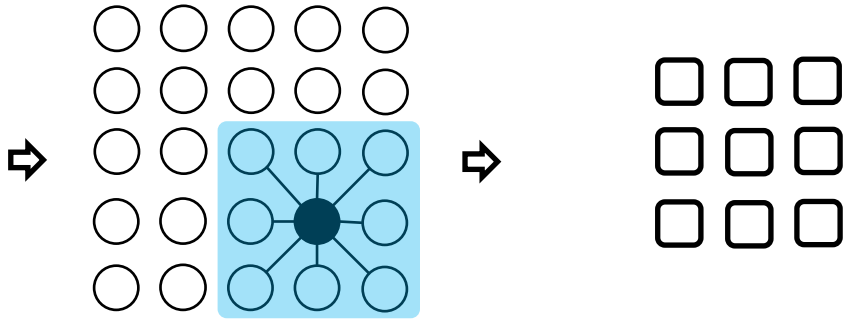
$$\sum WX$$

Convolutional neural network



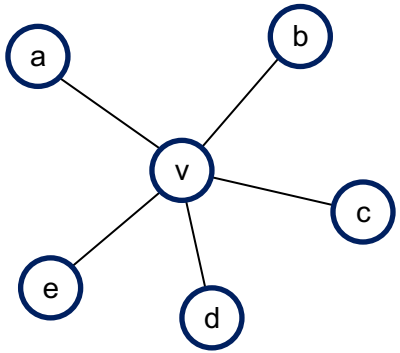
$$\sum WX$$

Convolutional neural network



$$\sum WX$$

Graph Neural Networks



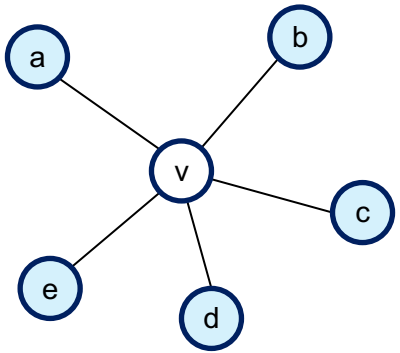
$$\boldsymbol{h}_a = f(\boldsymbol{h}_a, \boldsymbol{h}_b, \boldsymbol{h}_c, \boldsymbol{h}_d, \boldsymbol{h}_e)$$

Neighborhood Aggregation:

Aggregate neighbor information and pass into a neural network

It can be viewed as a center-surround filter in CNN---graph convolutions!

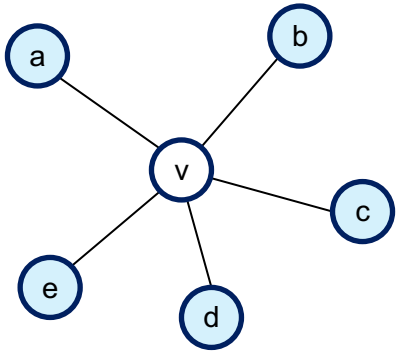
GNN: Graph Convolutional Networks



$$\boldsymbol{h}_v^k = \sigma(\boldsymbol{W}_k \sum_{u \in \mathrm{N}(v) \cup v} \frac{\boldsymbol{h}_u^{k-1}}{\sqrt{|N(u)||N(v)|}})$$

1. Kipf et al. Semisupervised Classification with Graph Convolutional Networks. ICLR 2017

GNN: Graph Convolutional Networks



parameters in layer k

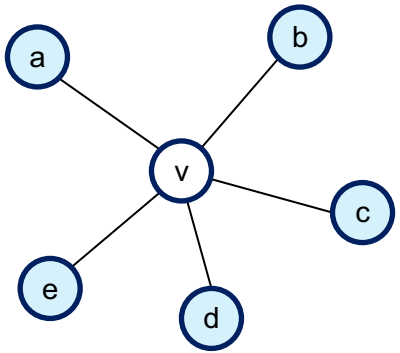
Non-linear activation function (e.g., ReLU)

$$\boldsymbol{h}_v^k = \sigma\left(\boldsymbol{W}_k \sum_{u \in \mathrm{N}(v) \cup v} \frac{\boldsymbol{h}_u^{k-1}}{\sqrt{|N(u)||N(v)|}}\right)$$

node v's embedding at layer k

the neighbors of node v

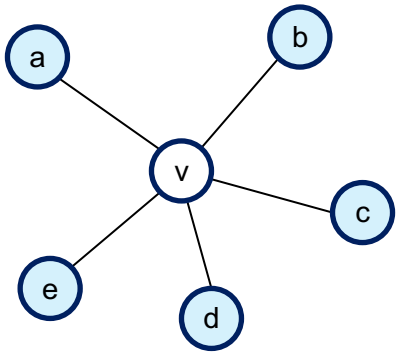
GNN: Graph Convolutional Networks



$$\boldsymbol{h}_v^k = \sigma(\boldsymbol{W}_k \sum_{u \in \mathrm{N}(v) \cup v} \frac{\boldsymbol{h}_u^{k-1}}{\sqrt{|N(u)||N(v)|}})$$

1. Kipf et al. Semisupervised Classification with Graph Convolutional Networks. ICLR 2017

GNN: Graph Convolutional Networks



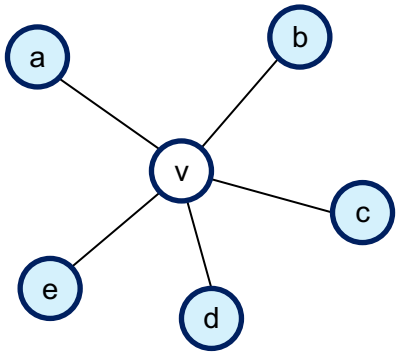
Aggregate from v's neighbors

$$\boldsymbol{h}_v^k = \sigma\left(\boldsymbol{W}_k \sum_{u \in N(v)} \frac{\boldsymbol{h}_u^{k-1}}{\sqrt{|N(u)||N(v)|}} + \boldsymbol{W}_k \sum_{v} \frac{\boldsymbol{h}_v^{k-1}}{\sqrt{|N(v)||N(v)|}}\right)$$

Aggregate from itself

1. Kipf et al. Semisupervised Classification with Graph Convolutional Networks. ICLR 2017

GNN: Graph Convolutional Networks

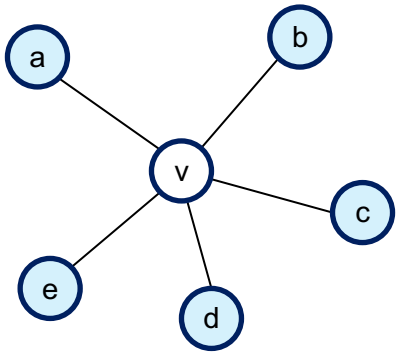


The same parameters for both its neighbors & itself

$$\boldsymbol{h}_v^k = \sigma\left(\boldsymbol{W}_k \sum_{u \in N(v)} \frac{\boldsymbol{h}_u^{k-1}}{\sqrt{|N(u)||N(v)|}} + \boldsymbol{W}_k \sum_{v} \frac{\boldsymbol{h}_v^{k-1}}{\sqrt{|N(v)||N(v)|}}\right)$$

Kipf et al. Semisupervised Classification with Graph Convolutional Networks. ICLR 2017

GNN: Graph Convolutional Networks



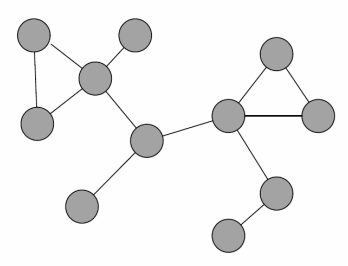
$$\boldsymbol{D}^{-\frac{1}{2}}\boldsymbol{A}\boldsymbol{D}^{-\frac{1}{2}}\boldsymbol{H}^{(k)}\boldsymbol{W}^{(k)}$$

$$\boldsymbol{h}_v^k = \sigma\left(\boldsymbol{W}_k \sum_{u \in \mathrm{N}(v)} \frac{\boldsymbol{h}_u^{k-1}}{\sqrt{|N(u)||N(v)|}} + \boldsymbol{W}_k \sum_{v} \frac{\boldsymbol{h}_v^{k-1}}{\sqrt{|N(v)||N(v)|}}\right)$$

$$\boldsymbol{D}^{-\frac{1}{2}}\boldsymbol{I}\boldsymbol{D}^{-\frac{1}{2}}\boldsymbol{H}^{(k)}\boldsymbol{W}^{(k)}$$

Kipf et al. Semisupervised Classification with Graph Convolutional Networks. ICLR 2017

GNN: Graph Convolutional Networks



$G = (V, E, \boldsymbol{A})$

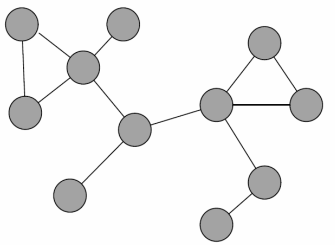
$\boldsymbol{H}^0 = \boldsymbol{X}$

$$\Rightarrow \boldsymbol{H}^k = \sigma\left(\boldsymbol{D}^{-\frac{1}{2}}(\boldsymbol{A}+\boldsymbol{I})\boldsymbol{D}^{-\frac{1}{2}}\boldsymbol{H}^{(k-1)}\boldsymbol{W}^{(k)}\right) \Rightarrow \quad \boldsymbol{Z} = \boldsymbol{H}^K$$

Input

Output

GNN: Graph Convolutional Networks



$G = (V, E, \boldsymbol{A})$

$\boldsymbol{H}^0 = \boldsymbol{X}$

$$\Rightarrow \boldsymbol{H}^k = \sigma\left(\boldsymbol{D}^{-\frac{1}{2}}(\boldsymbol{A}+\boldsymbol{I})\boldsymbol{D}^{-\frac{1}{2}}\boldsymbol{H}^{(k-1)}\boldsymbol{W}^{(k)}\right) \Rightarrow \quad \boldsymbol{Z} = \boldsymbol{H}^K$$

Input

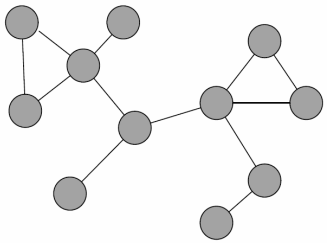
Model training

The common setting is to have an end to end training framework with a supervised task

That is, define a loss function over $\boldsymbol{Z}$

Output

GNN: Graph Convolutional Networks



$G = (V, E, \boldsymbol{A})$

$\boldsymbol{H}^0 = \boldsymbol{X}$

Input

$$\Rightarrow \boldsymbol{H}^k = \sigma\left(\boldsymbol{D}^{-\frac{1}{2}}(\boldsymbol{A}+\boldsymbol{I})\boldsymbol{D}^{-\frac{1}{2}}\boldsymbol{H}^{(k-1)}\boldsymbol{W}^{(k)}\right) \Rightarrow \quad \boldsymbol{Z} = \boldsymbol{H}^K$$

Output

Benefits: Parameter sharing for all nodes

- #parameters is subline in |V|
- Enable inductive learning for new nodes

GNN: Graph Convolutional Networks

$$\boldsymbol{H}^{k} = \sigma\left(\boldsymbol{D}^{-\frac{1}{2}}(\boldsymbol{A}+\boldsymbol{I})\boldsymbol{D}^{-\frac{1}{2}}\boldsymbol{H}^{(k-1)}\boldsymbol{W}^{(k)}\right)$$

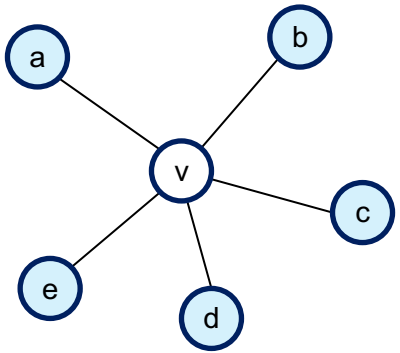
GCN is one way of neighbor aggregations

GraphSage

Graph Attention

… …

GraphSage



GCN

$$\boldsymbol{h}_v^k = \sigma(\boldsymbol{W}_k \sum_{u \in \mathrm{N}(v) \cup v} \frac{\boldsymbol{h}_u^{k-1}}{\sqrt{|N(u)||N(v)|}})$$

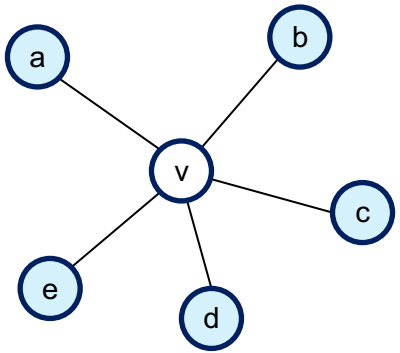
GraphSage

$$\boldsymbol{h}_v^k = \sigma([\boldsymbol{A}_k \cdot \mathrm{AGG}\left(\{\boldsymbol{h}_u^{k-1}, \forall u \in N(v)\}\right), \boldsymbol{B}_k \boldsymbol{h}_v^{k-1}])$$

Generalized aggregation: any differentiable function that maps set of vectors to a single vector

Hamilton et al. Inductive Representation Learning on Large Graphs. NIPS 2017
Hamiltion & Tang, AAAI 2019 Tutorial on Graph Representation Learning

GraphSage



GCN

$$\boldsymbol{h}_v^k = \sigma(\boldsymbol{W}_k \sum_{u \in \mathrm{N}(v) \cup v} \frac{\boldsymbol{h}_u^{k-1}}{\sqrt{|N(u)||N(v)|}})$$

GraphSage

Instead of summation, it concatenate neighbor & self embeddings

$$\boldsymbol{h}_v^k = \sigma([\boldsymbol{A}_k \cdot \mathrm{AGG}(\{\boldsymbol{h}_u^{k-1}, \forall u \in N(v)\}), \boldsymbol{B}_k \boldsymbol{h}_v^{k-1}])$$

Generalized aggregation: any differentiable function that maps set of vectors to a single vector

Hamilton et al. Inductive Representation Learning on Large Graphs. NIPS 2017
Hamiltion & Tang, AAAI 2019 Tutorial on Graph Representation Learning

GraphSage

$$\boldsymbol{h}_v^k = \sigma([\boldsymbol{A}_k \cdot \mathrm{AGG}(\{\boldsymbol{h}_u^{k-1}, \forall u \in N(v)\}), \boldsymbol{B}_k \boldsymbol{h}_v^{k-1}])$$

- Mean:

$$\mathrm{AGG} = \sum_{u \in N(v)} \frac{\mathbf{h}_u^{k-1}}{|N(v)|}$$

- Pool
  - Transform neighbor vectors and apply symmetric vector function.

element-wise mean/max

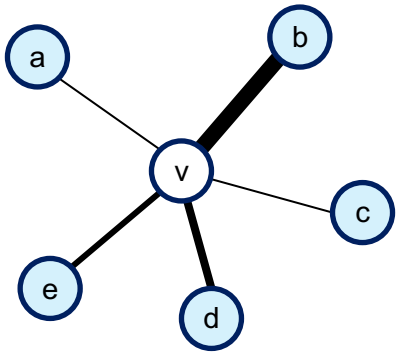
$$\mathrm{AGG} = \gamma(\{\mathbf{Q}\mathbf{h}_u^{k-1}, \forall u \in N(v)\})$$

- LSTM:
  - Apply LSTM to random permutation of neighbors.

$$\mathrm{AGG} = \mathrm{LSTM}\left([\mathbf{h}_u^{k-1}, \forall u \in \pi(N(v))]\right)$$

Hamilton et al. Inductive Representation Learning on Large Graphs. NIPS 2017
Slide snipping from "Hamiltion & Tang, AAAI 2019 Tutorial on Graph Representation Learning"

Graph Neural Networks



Realistically, neighbors are of different importance to each node

Velickovic et al. Graph Attention Networks. ICLR 2018

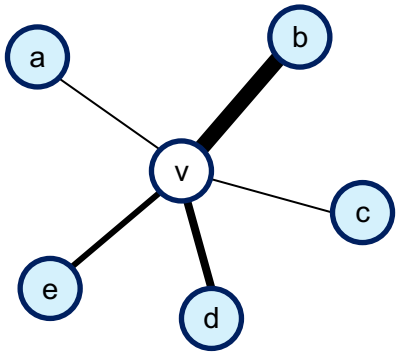
Graph Neural Networks

$$\boldsymbol{H}^{k} = \sigma\left(\boldsymbol{D}^{-\frac{1}{2}}(\boldsymbol{A}+\boldsymbol{I})\boldsymbol{D}^{-\frac{1}{2}}\boldsymbol{H}^{(k-1)}\boldsymbol{W}^{(k)}\right)$$

GCN is one way of neighbor aggregations

- GraphSage
- **Graph Attention**
- … …

GNN: Graph Attention



GCN

$$\boldsymbol{h}_v^k = \sigma(\boldsymbol{W}_k \sum_{u \in \mathrm{N}(v) \cup v} \frac{\boldsymbol{h}_u^{k-1}}{\sqrt{|N(u)||N(v)|}})$$

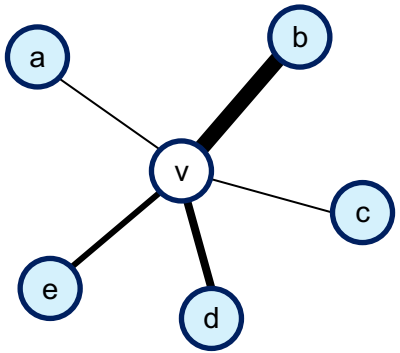
Graph Attention

$$\boldsymbol{h}_v^k = \sigma(\sum_{u \in \mathrm{N}(v) \cup v} \alpha_{v,u} \boldsymbol{W}^k \boldsymbol{h}_u^{k-1})$$

Learned attention weights

Velickovic et al. Graph Attention Networks. ICLR 2018

GNN: Graph Attention

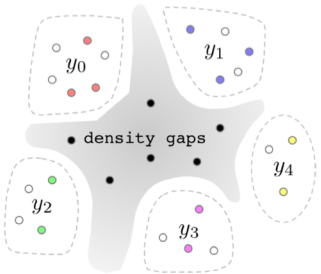


$$\alpha_{v,u} = \frac{\exp\left(\text{LeakyReLU}\left(\mathbf{a}^\top [\mathbf{Q}\mathbf{h}_v, \mathbf{Q}\mathbf{h}_u]\right)\right)}{\sum_{u' \in N(v) \cup \{v\}} \exp\left(\text{LeakyReLU}\left(\mathbf{a}^\top [\mathbf{Q}\mathbf{h}_v, \mathbf{Q}\mathbf{h}_{u'}]\right)\right)}$$

Various ways to define attention!

Velickovic et al. Graph Attention Networks. ICLR 2018

Graph Neural Networks



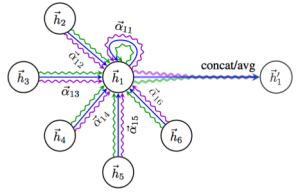
FastGCNs, Graph attention — 2018: Velickovic et al., ICLR'18, Chen et al., ICLR 2018

NetMF & NetSMF — 2018: Qiu et al., WSDM'18 & WWW'19

Neural message passing, GraphSage — 2017: Gilmer et al., ICML'17; Hamilton et al., NIPS'17

Gated graph neural network — 2016: Li et al., ICLR'16

structure2vec — 2016: Dai et al., ICML'16



Graph convolutional network — 2015: Duvenaud et al., NIPS'15; Kipf & Welling ICLR'17

PTE, metapath2vec — 2015: Tang et al., KDD'15; Dong et al., KDD'17

LINE, node2vec — 2015: Tang et al., WWW'15; Grover & Leskovec, KDD'16

DeepWalk — 2014: Perozzi et al., KDD'14

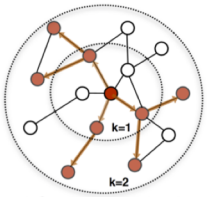
Spectral graph convolution — 2014: Bruna et al., ICLR'14

word2vec (skip-gram) — 2013: Mikolov et al., ICLR'13

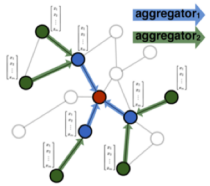
Graph neural network — 2005: Gori et al., IJCNN'05

Spectral clustering — 2000: Ng et al. & Shi, Malik

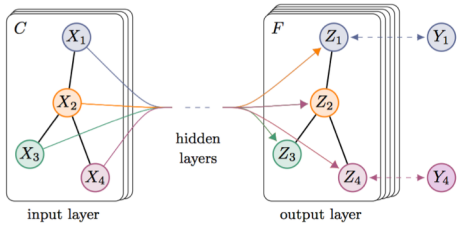
Spectral partitioning — 1973: Donath & Hoffman



1. Sample neighborhood



2. Aggregate feature information from neighbors

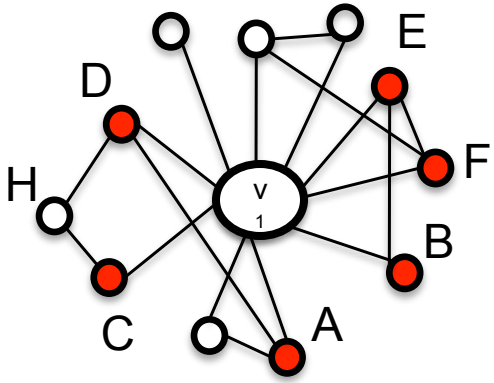


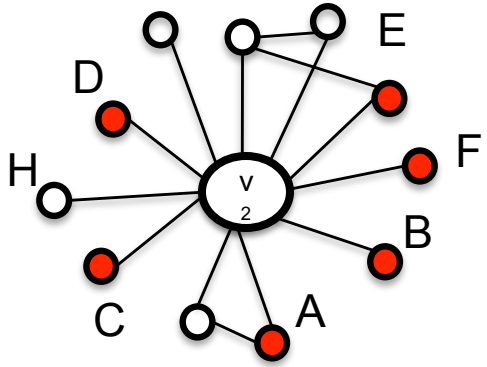
Example Application: Social Influence

- Social influence occurs when one's opinions, emotions, or behaviors are affected by others, intentionally or unintentionally.[1]
  - **Informational social influence**: to accept information from another;
  - **Normative social influence**: to conform to the positive expectations of others.

1. http://en.wikipedia.org/wiki/Social_influence
2. Qiu et al. DeepInf: Social Influence Prediction with Deep Learning. In KDD'18.

Structural Social Influence



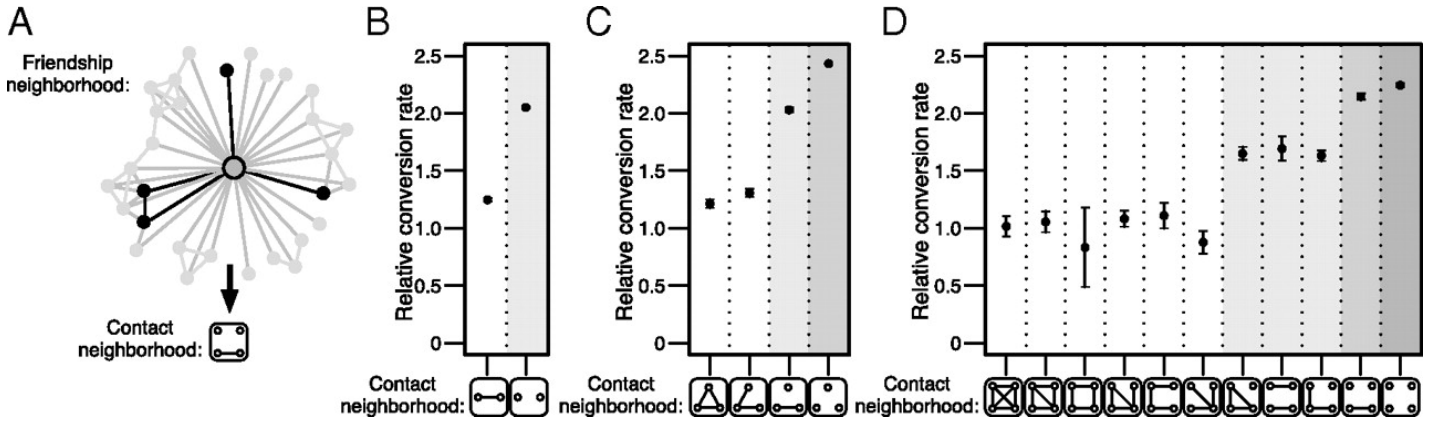


Who are more likely to be "active", v_1 or v_2?

e.g., active = "to attend CAAI AIDL"

Active neighbor Inactive neighbor v User to be influenced

Structural Social Influence



Invite user to join FB by existing users

(J. Ugander, L. Backstrom, C. Marlow, and J. Kleinberg, PNAS'12)

Previous Solution

Hand craft features + prediction model. E.g.,

Name	Description
Vertex	Coreness [3].
	Pagerank [30].
	Hub score and authority score [8].
	Eigenvector Centrality [5].
	Clustering Coefficient [46].
	Rarity (reciprocal of ego user's degree) [1].
	Network embedding (DeepWalk [31], 64-dim).
Ego	The number/ratio of active neighbors [2].
	Density of subgnetwork induced by active neighbors [40].
	#Connected components formed by active neighbors [40].

+ Logistic Regression

Network Representation Learning

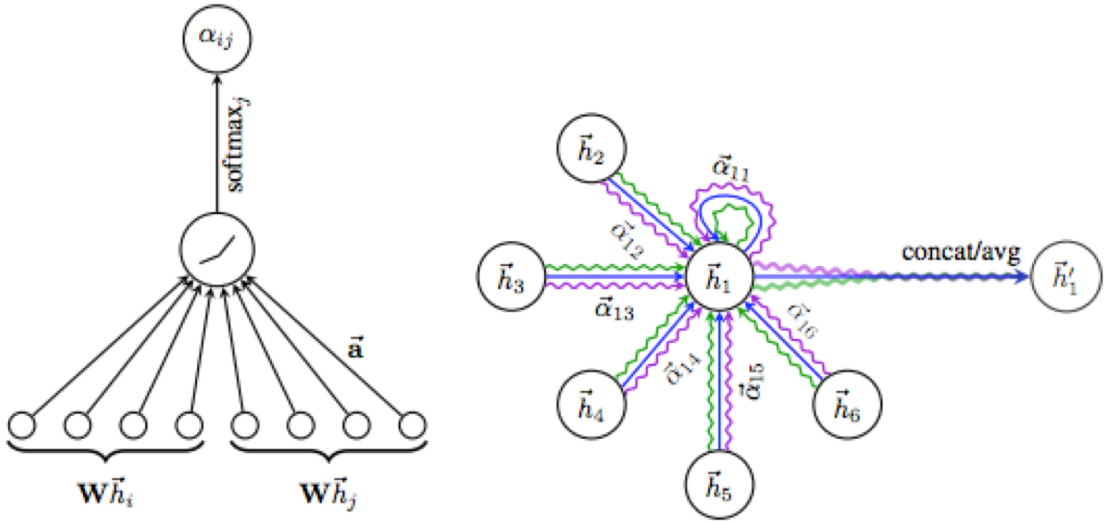
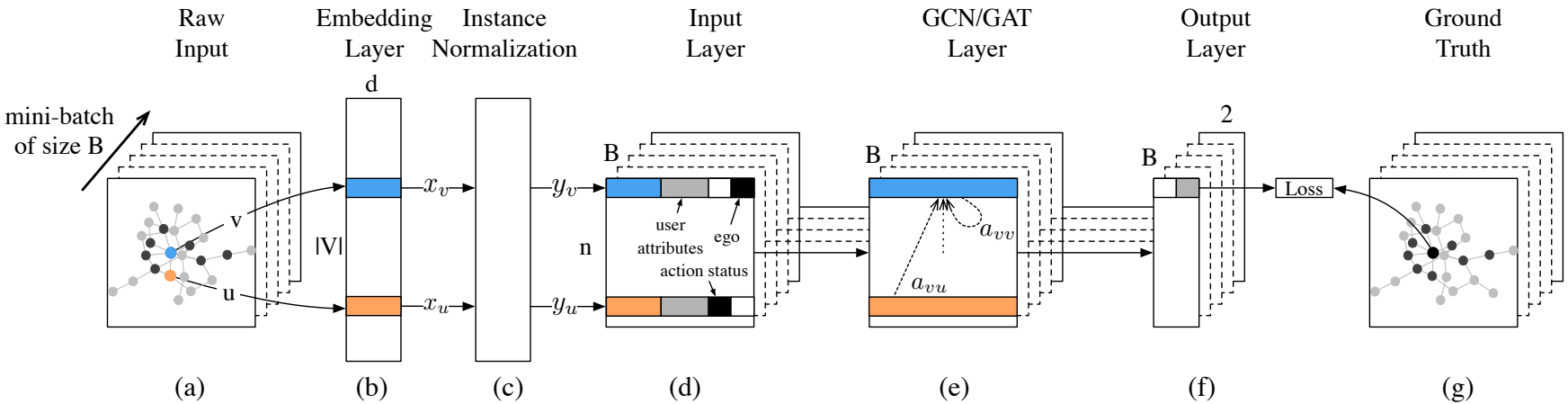


Figure 1: **Left:** The attention mechanism $a(\mathbf{W}\vec{h}_i, \mathbf{W}\vec{h}_j)$ employed by our model, parametrized by a weight vector $\vec{\mathbf{a}} \in \mathbb{R}^{2F'}$, applying a LeakyReLU activation. **Right:** An illustration of multi-head attention (with $K = 3$ heads) by node 1 on its neighborhood. Different arrow styles and colors denote independent attention computations. The aggregated features from each head are concatenated or averaged to obtain $\vec{h}'_1$.

Graph Attention Networks



1. Qiu et al. DeepInf: Social Influence Prediction with Deep Learning. In KDD'18.

Experiments --- Datasets

Table 1: Summary of datasets. $|V|$ and $|E|$ indicates the number of vertices and edges in graph $G = (V, E)$, while N is the number of social influence locality instances (observations) as described in Section 2.

	OAG	Digg	Twitter	Weibo
$\|V\|$	953,675	279,630	456,626	1,776,950
$\|E\|$	4,151,463	1,548,126	12,508,413	308,489,739
N	499,848	24,428	499,160	779,164

- OAG (Open Academic Graph): predict paper citations
- Digg (a Reddit-like platform): predict vote-up
- Twitter & Weibo: predict retweet

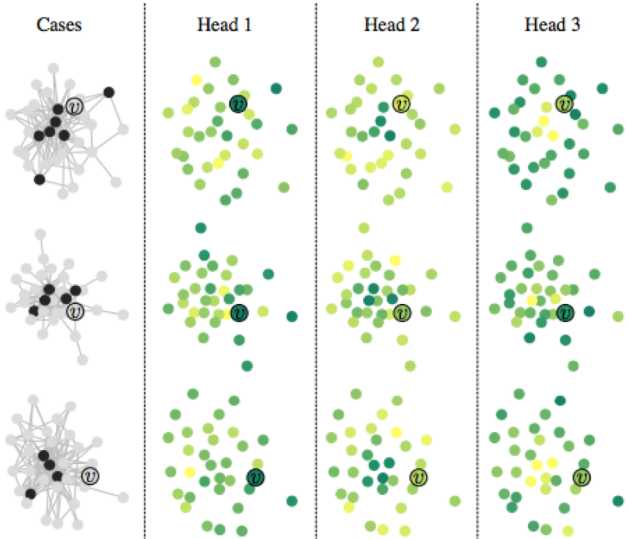
Experiments --- Results

- LR/SVM: Logistic regression/Support vector machine
- PSCN: SOTA deep graph kernel model (ICML'16)
- DeepInf-GAT: Our solution with GAT as bulding blocks

Data	Model	AUC	Prec.	Rec.	F1
OAG	LR	65.55	32.26	69.97	44.16
	SVM	65.48	32.17	**69.82**	44.04
	PSCN	67.70	36.24	60.46	45.32
	DeepInf-GAT	**70.59**	**38.93**	61.29	**47.61**
Digg	LR	84.72	56.78	73.12	63.92
	SVM	86.01	63.42	67.34	65.32
	PSCN	83.96	62.16	67.34	64.65
	DeepInf-GAT	**88.97**	**68.80**	**73.79**	**71.21**
Twitter	LR	78.07	45.86	**69.81**	55.36
	SVM	79.42	49.12	67.31	56.79
	PSCN	79.40	48.43	68.06	56.59
	DeepInf-GAT	**80.01**	**49.39**	67.47	**57.03**
Weibo	LR	77.10	42.34	72.88	53.56
	SVM	77.11	43.27	70.79	53.71
	PSCN	79.54	44.89	73.48	55.73
	DeepInf-GAT	**82.75**	**48.86**	**74.13**	**58.90**

Case Study

- How different graph attention heads highlight different areas of the network.
- Head 1: Focus on the ego-user
- Head 2: Highlight active users
- Head 3: Highlight inactive users



We applied the learned influence to help real applications

—in very big networks

Big Data Analytics in Game Data

- Online gaming is one of the largest industries on the Internet…
- Facebook
  - 250 million users play games monthly
  - 200 games with more than 1 million active users
  - 12% of the company's revenue is from games
- Tencent (Market Cap: ~150B $)
  - More than 400 million gaming users
  - 50% of Tencent's overall revenue is from games

[1] Zhanpeng Fang, Xinyu Zhou, Jie Tang, Wei Shao, A.C.M. Fong, Longjun Sun, Ying Ding, Ling Zhou, and Jarder Luo. Modeling Paying Behavior in Online Social Networks. **CIKM'14.**

Two games: DNF

- Dungeon & Fighter Online (DNF)
  - A game of melee combat between users and large number of underpowered enemies
  - 400+ million users, the 2nd largest online game in China
  - Users in the game can fight against enemies by individuals or by groups

Two games: QQ Speed

- QQ Speed
  - A racing game that users can partake in competitions to play against other users
  - 200+ million users
  - Users can race against other users by individuals or forma a group to race together
  - Some users may pay…

Task

- Given behavior log data and paying logs of online game users, build a model to help

item recommendation

- Will social influence play an important role in this task?

Online Test

- Test setting
  - Two groups: *test group* and *control group*
  - Send msgs to invite the user to attend a promotion activity.

	Online Test 1 2013.12.27 - 2014.1.3		Online Test 2 2014.1.24 - 2014.1.27		
Group name	test group	control group	test group	control group	random
Group size	600K	200K	400K	400K	200K
#Message read	345K	106K	229K	215K	106K
Message read rate	57.50%	53.00%	57.25%	53.75%	53.00%
#Message clicked	47584	7466	23325	20922	6299
Message clicked rate	7.93%	3.73%	5.83%	5.23%	3.15%
Lift_Ratio	196.87%	0%	123.63%	73.40%	0%

Dropout Prediction

- Dropout prediction with influence
  - Problem: Call back of users
  - Game data: *King of Glory (王者荣耀)*

	With Influence			w/o influence		
Top k	#message	#success	ratio	#message	#success	ratio
1	3996761	1953709	**48.88%**	6617662	2732167	41.29%
2	2567279	1272037	**49.55%**	9756330	4116895	42.20%
3	1449256	727728	**50.21%**	10537994	4474236	42.46%
4	767239	389588	**50.78%**	9891868	4255347	43.02%
5	3997251	2024859	**50.66%**	15695743	6589022	41.98%

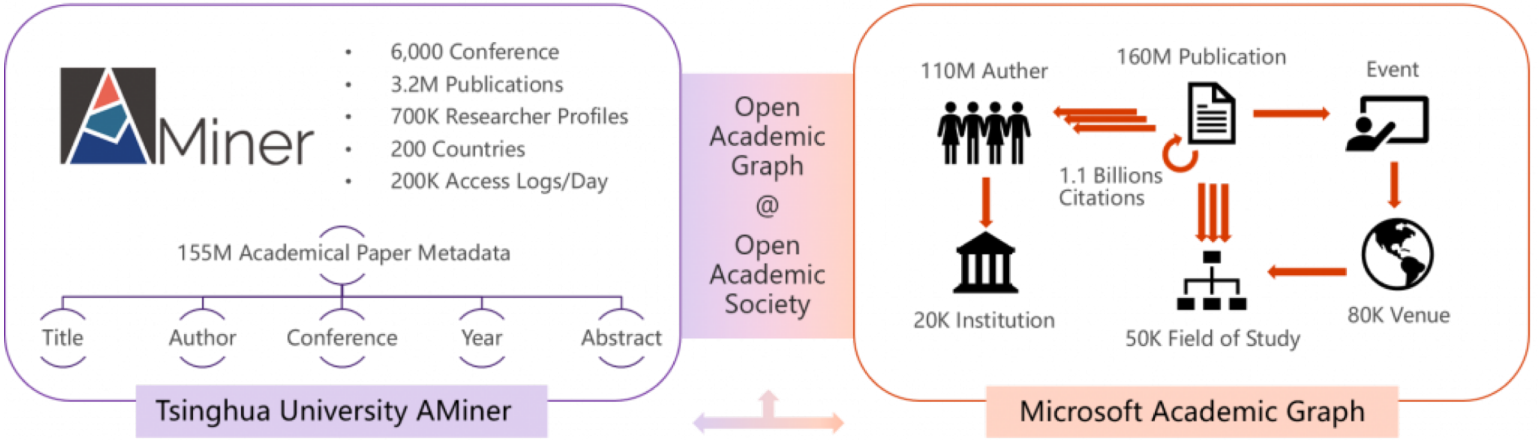
Dropout Prediction

- Deployed to 9 Games
  - 王者荣耀
  - 欢乐斗地主
  - QQ飞车手游
  - 穿越火线手游龙之谷
  - 英雄联盟
  - 逆战
  - NBA
  - CF
- In 2017, the total #callback: 129,000,000
- In 2018, up to now, the number is 52,890,000

OAG: Toward Linking Large-scale Heterogeneous Entity Graphs

Zhang et al. OAG: Toward Linking Large-scale Heterogeneous Entity Graphs. KDD'19

OAG overview

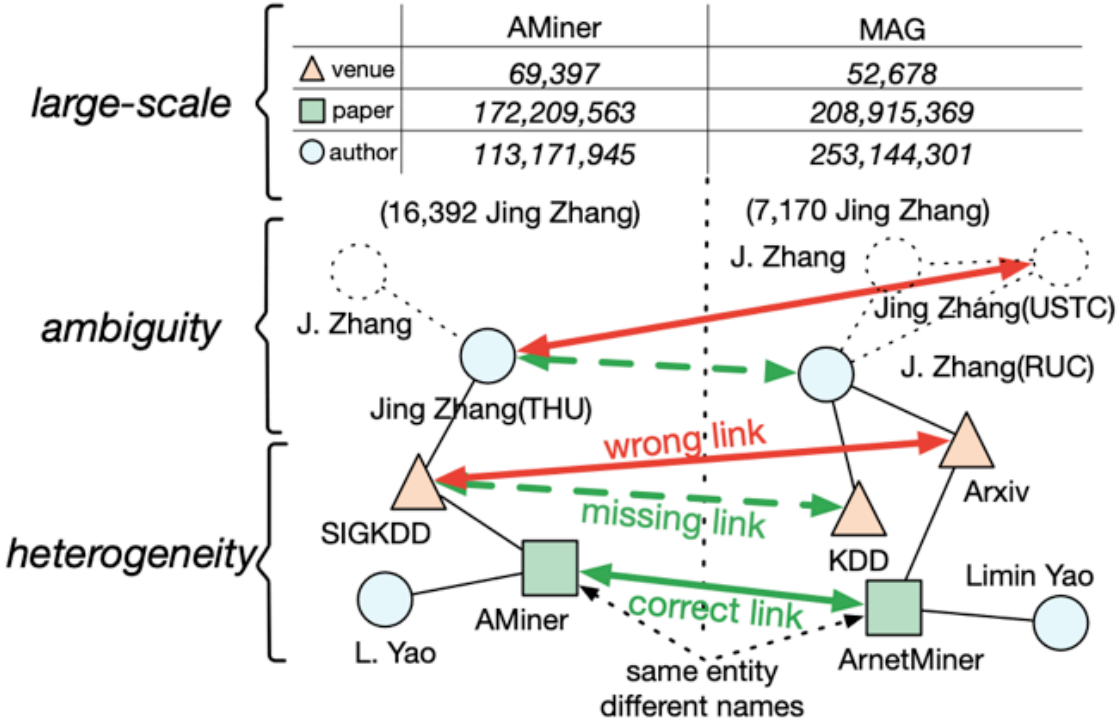


Linking *large-scale heterogeneous* academic graphs

Problem Definition

- **Heterogeneous Entity Graph (HEG):** $HG = \{E, R\}$ where each entity $e \in E$ and each relation $r \in R$ are associated with different types.
- **Input**: two heterogeneous entity graphs HG_1 and HG_2.
- **Output**: entity linkings $L = \{(e_1, e_2) | e_1 \in HG_1, e_2 \in HG_2\}$ such that e_1 and e_2 represent exactly the same entity.

Challenges

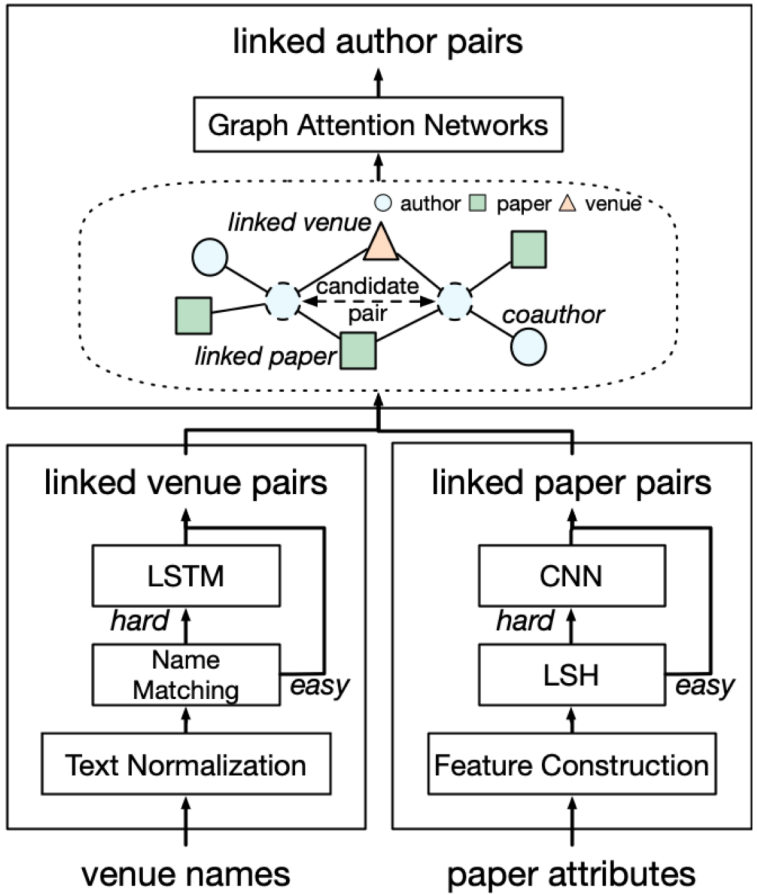

large-scale
ambiguity
heterogeneity
AMiner
MAG
venue
69,397
52,678
paper
172,209,563
208,915,369
author
113,171,945
253,144,301
(16,392 Jing Zhang)
(7,170 Jing Zhang)
J. Zhang
J. Zhang
Jing Zhang(USTC)
J. Zhang(RUC)
Jing Zhang(THU)
wrong link
missing link
Arxiv
SIGKDD
KDD
Limin Yao
correct link
AMiner
L. Yao
ArnetMiner
same entity
different names

Solution -- LinKG

Author linking module

Venue linking module

Paper linking module

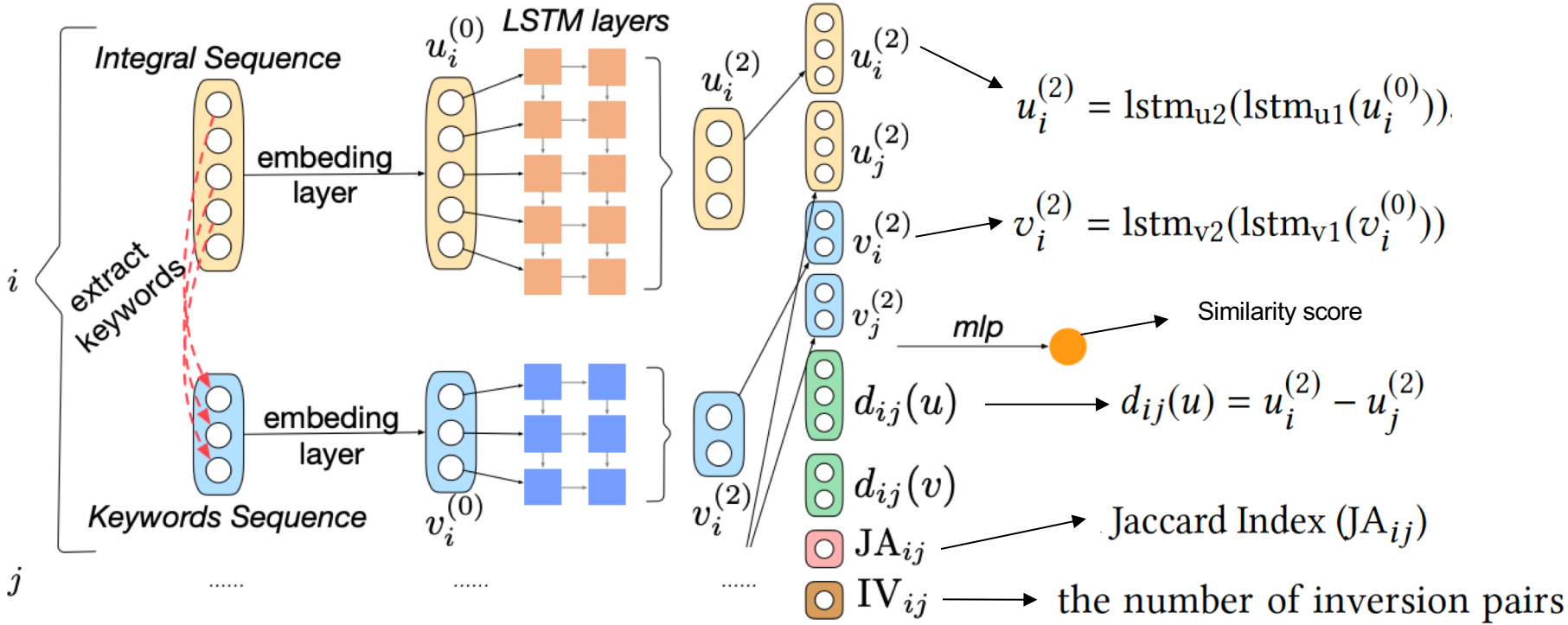


Linking Venues — Sequence-based Entities

- **Problem setting**: given the full names of venues in each graph, the goal is to link the same venues from both graphs.

- **Idea**: use direct name matching to link easy cases, and use LSTM to perform fuzzy-sequence linking.

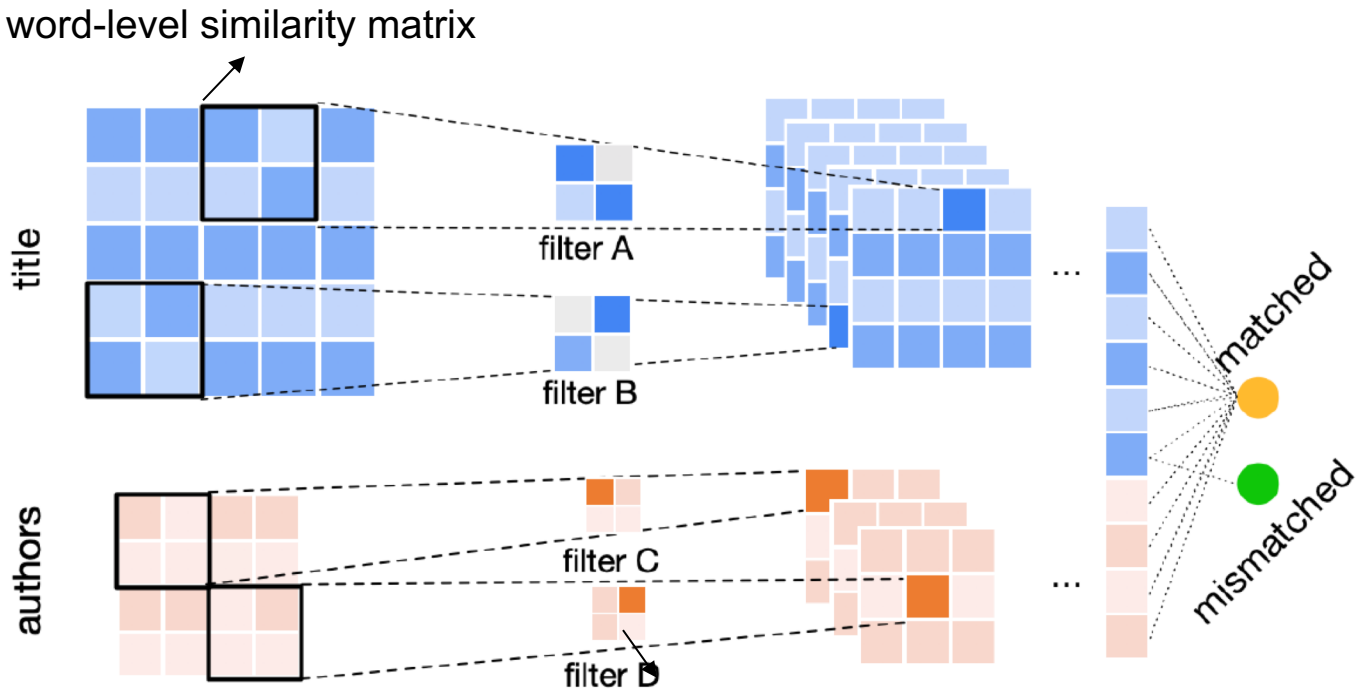
Venue Linking model



Linking Papers — Large-scale Entities

- **Problem setting**: To link paper entities, we fully leverage the heterogeneous information, including a paper's title and publication year, authors and venues.
- Leverage the **hashing technique** (LSH) for fast processing
  - Adopt Doc2Vec to transform titles to real-valued vectors
  - Use LSH to map real-valued paper features to binary codes.
- And the **convolutional neural network** for effective linking.

Paper linking model — CNN model

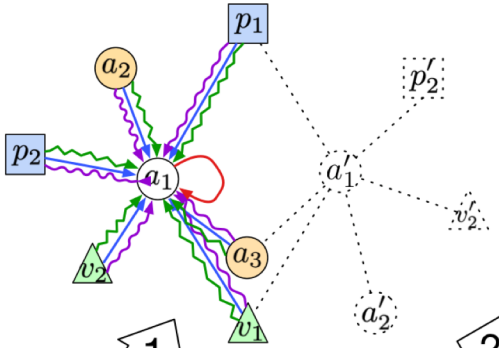


Linking Authors — Ambiguous Entities

- **Problem setting**: To link author entities, we generate a heterogeneous subgraph for each author. One author's subgraph is composed of his or her coauthors, papers, and publication venues.
- Also incorporate the **venue and paper linking results**.
- Present a heterogeneous graph attention network based technique for linking author entities.

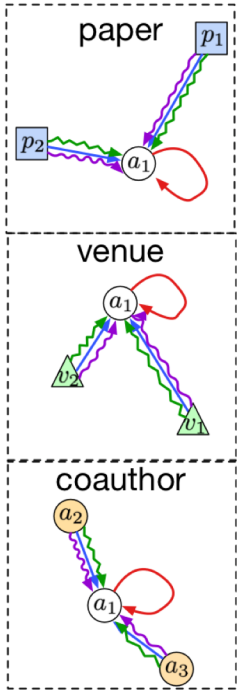
Author linking model — Heterogenous Graph Attention

Heterogeneous subgraph
for a candidate author pair



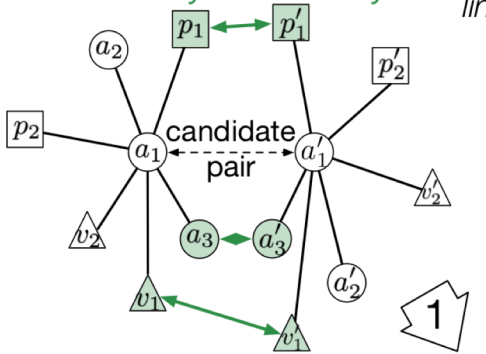
1 merge linked nodes

Split 2



Different attention parameters for different entity types

already linked entity



1

$$\alpha_{ij} = \frac{\exp(\text{LeakyReLU}(c^{\top}_{\tau(e_i)} W h_i + c^{\top}_{\tau(e_j)} W h_j))}{\sum_{k \in \mathcal{N}_i} \exp(\text{LeakyReLU}(c^{\top}_{\tau(e_i)} W h_i + c^{\top}_{\tau(e_k)} W h_k))}$$

Attention coefficient

Experimental Results

Table 1: Results of linking heterogeneous entity graphs. "–" indicates the method does not support the entity linking.

Methods		Keyword	SVM	Dedupe	COSNET	MEgo2Vec	LinKG$_C$	LinKG$_L$	LinKG
Venue	Prec.	80.15	81.69	84.25	–	–	84.67	**91.16**	**91.16**
	Rec.	83.76	83.45	80.92			85.81	**87.58**	**87.58**
	F1	81.91	82.56	82.55			85.23	**89.33**	**89.33**
Paper	Prec.	91.01	96.93	**99.30**	–	–	98.68	86.72	98.68
	Rec.	80.53	96.78	87.09			**98.10**	86.59	**98.10**
	F1	85.45	96.86	92.80			**98.39**	86.66	**98.39**
Author	Prec.	44.48	84.70	50.65	91.73	91.03	81.30	84.92	**95.37**
	Rec.	80.63	92.22	85.46	85.33	90.82	84.95	**94.75**	93.48
	F1	57.33	88.30	63.60	88.42	90.92	83.09	89.57	**94.42**
Overall	Prec.	74.80	92.36	82.26	91.73	91.03	92.38	86.21	**97.36**
	Rec.	80.64	94.89	86.38	85.33	90.82	93.29	89.41	**96.26**
	F1	77.61	93.61	84.27	88.42	90.92	92.83	87.78	**96.81**

OAG: Open Academic Graph

https://www.openacademic.ai/oag/

Data set	#Pairs/Venues	Date
Linking relations	29,841	2018.12
AMiner venues	69,397	2018.07
MAG venues	52,678	2018.11

Table 1: statistics of OAG venue data

Data set	#Pairs/Papers	Date
Linking relations	91,137,597	2018.12
AMiner papers	172,209,563	2019.01
MAG papers	208,915,369	2018.11

Table 2: statistics of OAG paper data

Data set	#Pairs/Authors	Date
Linking relations	1,717,680	2019.01
AMiner authors	113,171,945	2018.07
MAG authors	253,144,301	2018.11

Open Academic Graph

Open Academic Graph (OAG) is a large knowledge graph unifying two billion-scale academic graphs: Microsoft Academic Graph (MAG) and AMiner. In mid 2017, we published OAG v1, which contains 166,192,182 papers from MAG and 154,771,162 papers from AMiner (see below) and generated 64,639,608 linking (matching) relations between the two graphs. This time, in OAG v2, author, venue and newer publication data and the corresponding matchings are available.

Overview of OAG v2

The statistics of OAG v2 is listed as the three tables below. The two large graphs are both evolving and we take MAG November 2018 snapshot and AMiner July 2018 or January 2019 snapshot for this version.

1. Zhang, et al. OAG: Toward Linking Large-scale Heterogeneous Entity Graphs. In KDD'19.