Graph Node Embedding

a second round literature review

Chengbin Hou

Department of Computer Science and Engineering
Southern University of Science and Technology, ShenZhen, China

chengbin.hou10@foxmail.com

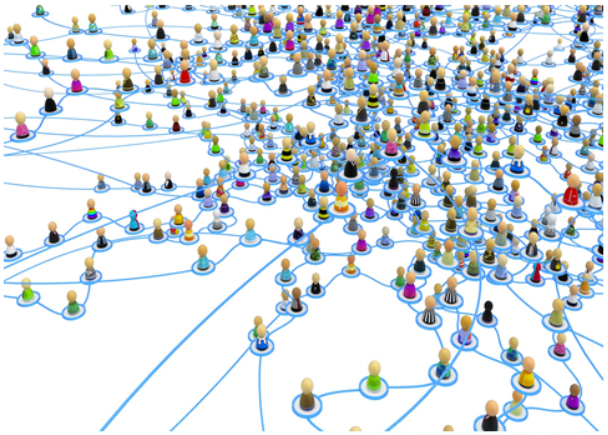
March 7, 2018

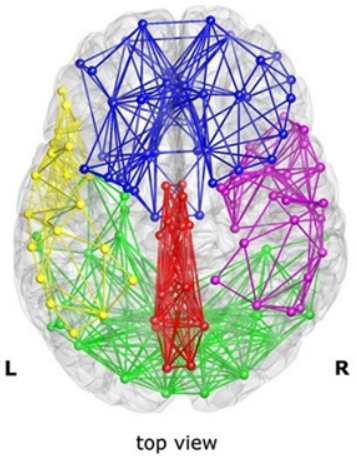
Overview

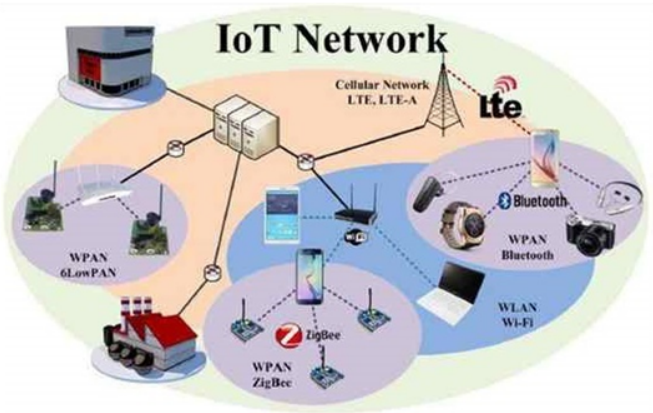
Graph-Structured Data

Ubiquitous Real-World Applications

Social Networks, Biological Networks, Recommendation Systems, Searching Engines, Word Co-Occurrence Networks, IoT Networks, Traffic Networks, Terrorist-Attack Networks, Co-Author Network, and etc.



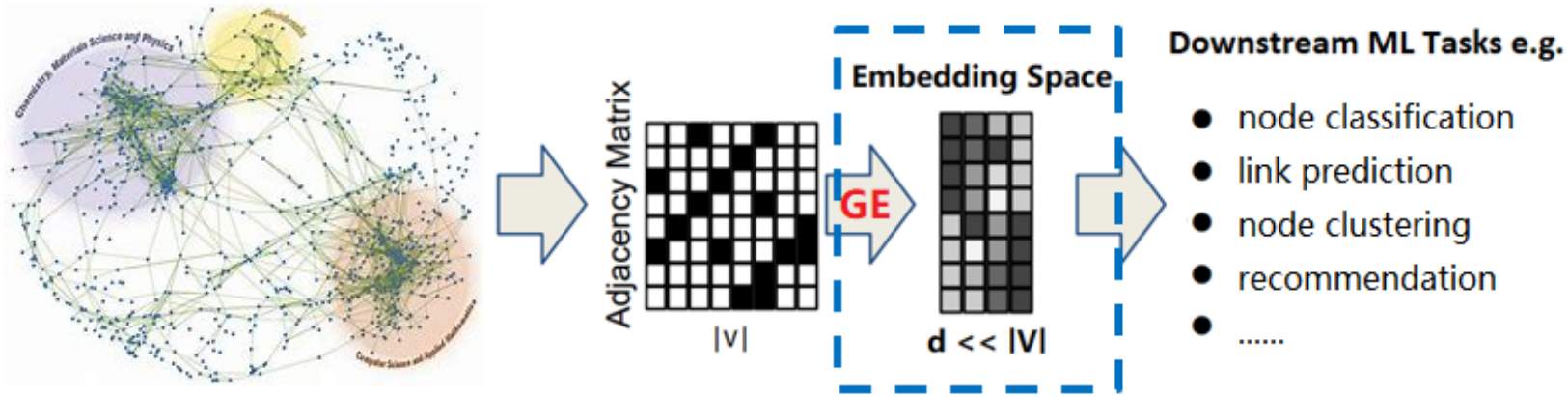




Challenges

- hard to represent data or lacking of syntactic/semantic meaning
- increasing needs of large-scale graph e.g. millions of nodes and billions of edges
- heterogeneous, multi-modal info., dynamic, incomplete, uncertain graph ...

GE Role in the Big Picture

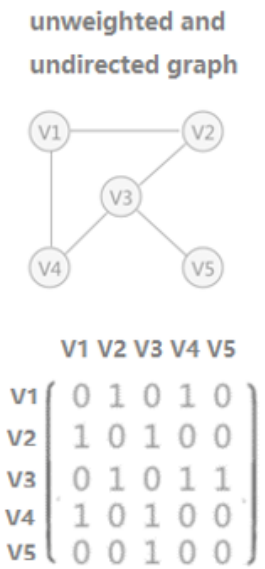


Advantages of GE

- as preprocessing (unsupervised) to extract meaningful representation
- much lower dimension: less space to store and faster downstream tasks
- transforming a complex network that contains both structure info and auxiliary info into a unified metric vector space

GE Concepts

- Graph Embedding (GE) is a.k.a. Network Representation Learning (NRL) Unless otherwise specified, we refer GE to **Graph Node Embedding** as nodes can naturally model the real-world instances in many applications.
- A **Graph** $G(V, E)$ where Vs are vertices/nodes and Es are edges/links is often described by **Adjacency Matrix** $A \in \mathbb{R}^{n \times n}$ where i^{th} row of A is its node representation w.r.t all nodes and $V_i \in \mathbb{R}^n$
- GE aims to **learn node representation** Z via $\mathcal{F}$: $A \in \mathbb{R}^{n \times n} \rightarrow Z \in \mathbb{R}^{n \times d}$ where $d \ll n$ such that some graph properties are maximally preserved.
- Graph **structure properties*** given by graph structure information A:
  – micro-scope: 1^{st}, 2^{nd}, and k^{th} order proximity;
  – macro-scope: community, structure role, asymmetry, and etc.



*Except graph structure properties, we would have other properties to be preserved if other graph auxiliary information is available e.g. graph node attributes/labels.

GE Output and Input

GE Output

- **Node Embedding**: learning representation for each node
- Edge Embedding: learning representation for each edge
- Subgraph Embedding: learning representation for each subgraph
- Others: whole-graph embedding and hybrid (e.g. node&edge) embedding

GE Input

- **Naive Graph**: (un)weighted and (un)directed graph described by A alone
- **Information Graph**: Naive Graph + nodes and/edges with auxiliary information (information can be attributes, labels, and others...)
- Others: Heterogeneous Graph, Dynamic Graph, Multi-modal Information Graph, Knowledge Graph, Signed Graph, Graph constructed from Discrete Instances, etc. (they can be converted to Information Graph or series of Naive Graph)

GE Problem Description

Naive GE Problem

Input: adjacency matrix $A \in \mathbb{R}^{n \times n}$
Output: node embedding $Z \in \mathbb{R}^{n \times d}$ where $d \ll n$
Methodology: find a mapping $\mathcal{F}$: $A \rightarrow Z$ with some properties to be preserved

Information GE Problem

Input: adjacency matrix $A \in \mathbb{R}^{n \times n}$ and **node auxiliary information** matrix $X \in \mathbb{R}^{n \times m}$
Output: node embedding $Z \in \mathbb{R}^{n \times d}$ where $d \ll n$
Methodology: find a mapping $\mathcal{F}$: $(A, X) \rightarrow Z$ with some properties to be preserved

Without otherwise specified, the Information Graph usually refers to graph with node auxiliary information

GE Problem Remarks

1. **Key differences**: the input, as well as the resulting properties to be reserved

2. Other GE problems can be seen as **Information Graph** or series of **Naive Graph**:
   - node heterogeneous info for each node can be coded into attribute matrix X
   - node degree of Naive Graph may be treated as auxiliary info and coded into X
   - graph constructed from discrete instances then gives A (Manifold Learning)
   - dynamic graph can be expressed as A^{t0}, A^{t1}, A^{t2}, ...
   - multi-modal information graph can be formulated as A with X^1, X^2, X^3...

3. In short, different GE problems depend on A + given different auxiliary info

4. Our taxonomy of GE methods would be clearly associated with GE problems in the following contents and again, we will focus on
   Input: Naive Graph/Information Graph
   Output: Node Embedding

Limitations of Previous Survey Papers

Recent four GE survey papers:

1. H. Cai and et al, "A Comprehensive Survey of Graph Embedding: Problems, Techniques and Applications", *IEEE TKDE*, 2017.
(under 2nd round review, last updated @Jan 2018, from ADSC Singapore & UIUC)
2. P. Goyal and et al, "Graph Embedding Techniques, Applications, and Performance: A Survey", *Knowledge Based Systems*, 2017.
(under review, last updated @Dec 2017, from Uni. of Southern California)
3. P. Cui and et al, "A Survey on Network Embedding", *arXiv:1711.08752*, 2017.
(from Tsinghua Uni, last updated @Nov 2017)
4. W. Hamilton and et al, "Representation Learning on Graphs: Methods and Applications", *IEEE Data Engineering Bulletin*, Sep 2017.
(accepted, from Stanford Uni. SNAP group)

However, due to the rapidly development of GE techniques, these surveys more or less have some limitations. For examples, we argue that:

- For Survey 1, it was too ambitious to divide about 100 papers into over 10 small categories. However, most papers were motivated by similar ideas and really hard to distinguish them.
- For Survey 2, it focused on several well-known methods and the categories are easy and clear. However, all methods therein were designed to cope with Naive GE problem.
- For Survey 3, it categorized methods based on the types of information/properties preserved. However, the similar/dissimilar ideas among different methods are not well-presented.
- For Survey 4, it reviewed recent popular GE methods wrt. different GE problem settings. However, it did not explicitly associate GE methods with GE problems.

Our Criteria for Methodology Taxonomy

Notice

We are handling **graph** embedding problem and as the name suggested, the **graph structure information i.e. A must be one of the inputs**. Otherwise, we may say this is not a typical graph related problem (although we may still try GE methods).

Our criteria to classify GE methods

- the input is either Naive Graph or Information Graph
- the general form of objective function i.e. how to play with A or A and X

The auxiliary information can be node attributes or labels or others, however, we should not be confused by unsupervised learning or (semi-)supervised learning regarding labels.
Without otherwise specified, all methods are unsupervised. Nonetheless, they may be switched to (semi-)supervised if the downstream task requires to do so e.g. given node label in node classification tasks.

GE methods wrt. GE inputs

	Naïve Graph i.e. given A	Information Graph i.e. given A and X
MF	$min\|\|S(A) - S(Z)\|\|$ or $min \sum_j \sum_i S_{ij}\|\|Z_i - Z_j\|\|$ $S(\cdot)$ for similarity measure between every two rows e.g. $< A_i, A_j >$; S_{ij} for penalty calculated by A e.g. simply A_{ij}; one way is to use SGD for training where $Z_i = A_i W$ and $W \in \mathbb{R}^{n \times d}$; another ways is to transform problem and then apply SVD;	also have two kind of objective functions similar as Naïve ones; just replace $A \triangleq ATX$ where T is the bilinear transformation; the idea is to unify A and X by T (related to Metric Learning); for multi-modal case, we will have more Xs and more Ts ; regularization terms may be required for Ts;
RW	$max\ P(\mathcal{N}_i\|i) = max \prod_{j \in \mathcal{N}_i} \left(\frac{e^{z_i z_j^T}}{\sum_{k \in V} e^{z_i z_k^T}} \right) = min - log(P) = \cdots$ $\mathcal{N}_i$ for neighborhoods of node i after walking/sampling e.g. DFS; Skip-gram: max node co-occurrence probability of its nearby nodes; using SGD for training where $Z_i = A_i W$;	in case of heterogeneous as auxiliary information, two ways: 1) similar as MF for Information Graph to obtain $A \triangleq ATX$ first; 2) define heterogenous walking strategy to obtain $\mathcal{N}_i$ first; then, follow ideas of RW for Naïve Graph
GC	same as RW if $Z^0 \triangleq I$ and $\hat{A} \triangleq A$ with direct mapping $Z_i = A_i I W$; to use GC in Naïve Graph, one may assign node degree to X;	$min - log(P) = min - \sum_{j \in \mathcal{N}_i} \left(\frac{e^{z_i^T \cdot z_j}}{\sum_{k \in V} e^{z_i^T \cdot z_k}} \right) = softmax(Z_i^T, Z_j)$ the same objective as RW; $z_i^{K+1} = f(\hat{A}_i \dots f(\hat{A}_i Z^0 W^0) \dots W^K)$ where $\hat{A} \triangleq \frac{1}{\sum_j (I+A)_{ij}} (I + A)$, $Z^0 = X$, and $f(\cdot)$ is nonlinear mapping e.g. ReLU; related to Graph Convolution or Weisfeiler-Lehman graph kernel;
AE	$min \sum_{i \in V} \|\|\hat{Z}_i - Z_i\|\|$ the idea is minimize reconstruction error (Auto-Encoder); using SGD for training where $\hat{Z}_i = Z_i^{K+1} = f(\dots f(A_i W^0) \dots W^K)$; the embedding is given by some middle layer Z_i^{hidden};	first, similar as MF for Information Graph to unify A and X ; then, similar as AE for Naïve Graph to apply Auto-Encoder; Or, apply Auto-Encoder first, then unify different sources

MF: Matrix Factorization based; RW: Random Walk based; GC: Graph Convolution based; AE: Auto-Encoder based;
graph structure info: $A \in \mathbb{R}^{n \times n}$; graph auxiliary info: $X \in \mathbb{R}^{n \times m}$;
trainable parameters: $W \in \mathbb{R}^{? \times ?}$; GE output: $Z \in \mathbb{R}^{n \times d}$;

Matrix Factorization (MF) based approach

MF basic idea, for a undirected and unweighted Naive GE problem

- represent a node w.r.t. all nodes as a row vector (1 if linked, 0 otherwise)
- form an adjacency matrix A as the representation for all nodes accordingly
- factorize matrix $A \in \mathbb{R}^{n \times n}$ using linear algebra tool e.g. SVD
- seek a much lower rank approximation $Z \in \mathbb{R}^{n \times d}$ by remaining top d eigenvalues
- Alternatively, establish loss between A and Z (e.g. taking inner product of every two rows as metric) and then, apply gradient based method for optimization

MF for GE variants:

- For Naive GE: we may further encode A to an advanced matrix $\hat{A}$ e.g. [65] according to our interest of graph property, and then measure node pairwise similarity on $\hat{A}$. Besides, the way to construct A might be different (think of Manifold Learning).
- For Information GE: both structure A and attribute X (or more Xs) should be considered and hence, additional projection/correlation matrices are required to unified them. Besides, we need to properly weight these terms as well as regularize the correlation matrices.

Matrix Factorization (MF) based approach

Representative MF based Methods

- Naive GE: 1) PCA and MDS came from classical linear dim reduction [13], since we can naturally think it as a dimension reduction problem; 2) LLE and ISOMAP came from manifold learning [13], since they constructed a graph based on discrete dataset by a kind of approximation and then preserve graph structure property using matrix factorization; 3) HOPE[4] and GraRep[5], considered micro-scope properties e.g. k^{th} proximity; 4) M-NMF[21] and PRUNE[71], considered macro-scope properties e.g. community property; 5) MMDW[44], considered downstream task and learned discriminative embedding using labels
- Information GE: 1) auxiliary info X fully observable e.g. HNE[18], TADW[60] and AANE[70]; 2) auxiliary info X partially observable e.g. CVLP[23];

Recent Development of MF:

- For Naive Graph, explore new structure properties,or combine multiple properties, or improve performance regarding well-known properties.
- For Information Graph, unify A and X into one metric space for a complete observable and static graph, or adapt it to partially observable and dynamic graph.

Random Walk (RW) based approach

RW basic idea, for a undirected and unweighted Naive GE problem

- choose a walking/sampling strategy e.g. DFS
- for every node, generate a "graph sentence" with walking length l
- for every sentence, generate a series of "training corpus" with size $(w+1+w)$
- for every training corpus, given central node i, maximize the nodes co-occurrence probability $P(i-w,...,i-1,i+1,...,i+w|i)$ (SkipGram Model)
- for every node pair (i,j), we have $P(j|i) \triangleq softmax(Z_i, Z_j) = \frac{e^{Z_i Z_j^T}}{\sum_{k\in|V|} e^{Z_i Z_k^T}}$ where $Z_i \triangleq A_i W$: $Z_i \in \mathbb{R}^{1\times d}$, $A_i \in \mathbb{R}^{1\times n}$ and $W \in \mathbb{R}^{n\times d}$ contains trainable weights.

RW for GE variants:

- For Naive GE: design different walking/sampling strategy so as to bias to our interested property;
- For Information GE: 1) for Heterogeneous Graph, establish heterogeneous sampling strategy according to heterogeneous auxiliary info; 2) for others e.g. with node attributes, see GC method...

Random Walk (RW) based approach

Representative RW based Methods

- Naive GE: different sampling strategies: DeepWalk[8] with BFS; Node2Vec[11] with BFS+DFS; NBNE[10] with random permutation; LINE[7] bias to 1^{st} and 2^{nd} order proximity; comE[21] bias to community; struc2vec[6] and SNS[74] bias to structure role, APP[26] bias to asymmetric property; DP-Walker[104] bias to scale-free property (i.e. "node degree VS node #" follows power law);
- Information GE: 1) for heterogeneous, define heterogeneous sampling strategy [30][61][82]; 2) for graph with node attributes, see GC method...

Recent Development of RW:

- Milestone: Word2Vec(2013 from NLP) $\rightarrow$ DeepWalk(2014) $\rightarrow$ Node2Vec & LINE(2015) $\rightarrow$ many others..
- To bias to our interested property:
  – directly alter walking strategy e.g. DeepWalk, Node2Vec and NBNE;
  – use empirical probability as "penalty" e.g. LINE and its following works [61][64][79][104] used slightly different objective function which employed KL-divergence to measure difference between $\hat{P}$ (user defined empirical probability) and P (by SkipGram as usual);
- top conference indicator: 1) novel graph property? and the way to preserve it; 2) proper heterogeneous sampling strategy?; 3) Except existing algorithm to simplify *softmax* denominator calculation e.g. Negative Sampling[40], is there any other algorithm to further speed up GE? i.e. truly scalable say millions of nodes and billions of links; Really? It might be very hard since Negative Sampling has reduced complexity to $O(|V|)$ where $|V|$ is the # of nodes. (but how about to get much faster embedding with little prediction accuracy lost? binary embedding?)

Graph Convolution (GC) based approach

GC basic idea, considering also auxiliary information, the forward propagation:

- set depth K and employ e.g. BFS to explore node 1-hop neighbors $\mathcal{N}_i$
- for every node i:
  at $k = 0$: assign auxiliary info $X_i^{k=0}$ to embedding $Z_i^{k=0}$ (initialization)
  at $k = 1$: collect its neighbors $\{Z_{\mathcal{N}_i}^{k=0}\}$; combine $(Z_i^{k=0}, \{Z_{\mathcal{N}_i}^{k=0}\}) \rightarrow Z_i^{k=1}$
  repeated until k=K
- the combining operation(s) involve trainable parameters

NB: The above idea gives the **forward propagation** (theoretically related to Weisfeiler-Lehman isomorphism test[38]). In order to train the parameters for combining operation(s), simply apply SkipGram Model as the **objective function for backward propagation** where the Z_i now comes from auxiliary information X instead of A (compared with RW).

GC for GE variants:

- For Naive GE: treat node degree as node attribute and form X
- For Information GE: methods mainly differ in 1) the sampling strategy based on A to generate $\mathcal{N}_i$; 2) the way to combine information collected from X according to $\mathcal{N}_i$;

Graph Convolution (GC) based approach

Representative GC based Methods

- GCN[36] was motivated by first-order approximation of localized spectral filters on graph, and end up with two-layer model $Z^2 = softmax(\hat{A}ReLU(\hat{A}Z^0W^0)W^1)$ where the normalized Laplacian $\hat{A} = I + D^{-\frac{1}{2}}AD^{-\frac{1}{2}} \in \mathbb{R}^{n\times n}$, $Z^0 = X \in \mathbb{R}^{n\times m}$; parameter $W^0 \in \mathbb{R}^{m\times h}$, element-wise operation $Z^1 = ReLU(\hat{A}Z^0W^0) \in \mathbb{R}^{n\times h}$; parameters $W^1 \in \mathbb{R}^{h\times d}$, row-wise operation $Z^2 = softmax(\hat{A}Z^1W^1) \in \mathbb{R}^{n\times d}$. Then employ SkipGram Model to train W^0 and W^1.
- GraphSAGE[24] mainly demonstrated from Weisfeiler-Lehman isomorphism test perspective. It chose limited neighboring info for computation efficiency (but in GCN, full neighboring info for node i were chosen and weighted summed by $\hat{A}_iX$), and then employ e.g. max pooling to combine them.
- MoNet[45] ChebNet[105], GAT[58] and etc. addressed similar ideas either from Graph Convolution or Weisfeiler-Lehman perspective.

Recent Development of GC:

- All these methods were just developed in later 2016 and can be seen as a more general case of RW for naturally dealing with graph auxiliary information.
- Intuitively, think of convolution on a $(n \times m)$ **real image i.e. fine grid graph** (gives $A \in \mathbb{R}^{nm\times nm}$ very regular and sparse) where each node is with RBG values i.e. auxiliary info (gives $X \in \mathbb{R}^{nm\times 3}$).
Graph Convolution is to generalize convolution on a irregular graph!

Auto-Encoder (AE) based approach

AE based methods summary

- The general idea is nothing special about well-known AE: minimize reconstruction error between input and output, then the middle layer gives embedding.
- Methods mainly differ in:
  1) the input to AE: [34] takes a row of A; [35] takes a row of $PPMI(A)$
  2) in order to merge multiple information, modify AE framework [66][67][78][86] so as to first concatenate multi-modal info and then learned by AE or v.v.

Recent Development of AE:

- In 2016, two works [34][35] designed for Naive Graph.
- In 2017, four works [66][67][78][86] designed for Information Graph and specially, [66] can deal with edge auxiliary information and hence is ideal for Knowledge Graph.

Paper Statistics

	MF based	RW based	GC based	AE based	Others
A	4(HOPE), 5(GraRep), 21(M-NMF), 32(distributedGF), 33(LE), 65(NEU,meta), 71(PRUNE, newProperty global node importance ranking), 84(NetMF), 92(MNE,unified MF and RW as MF), 104(DP-Spectral, power law), 44(MMDW, overcome RW, RW as MF, use MM-SVM) **11**	6(struc2vec, BFS),8(DeepWalk, DFS), 10(NBNE, random permutation), 11 (Node2Vec, bias RW), 20 (word2vec), 22(HARP, meta), 26 (APP, asymmetric), 74(SNS, with graphlets), 77 (HIN2Vec), 83 (comE, see 21), 100 (BE, binary embedding), 104(DP-Walker, power law),80 (URGE, uncertenty),87 (MINES, multiview of A), 79 (MVE, multiview of A, LINE+Attention), 64 (CANE, modified from LINE),7(LINE, edge sampling) **17**	**0**	34(SDNE, AE),35 (DNGR, AE with ppmi), **2**	15 (graph kernel), 16 (graph kernel), 53 (hyperbolic), 54 (hyperbolic), 89(GAN),95 (GAN), 102(GAN), 97(DHNE, hyper-graph), 48 (t-SNE), 38 (Weisfeiler-Lehman), 55 (modifed t-SNE best paper), 91(DynamicTriad, EM),96(DepthLGP, Gauss Process+DL) **13**
A + X	18(HNE), 23(CVLP), 60 (TADW, DW as MF), 69(SignedFS), 70 (AANE), 75(SNEA simiar to 69), 76 (DANE dynamic), 25(LANE), 81 (MCGE, multiview), **9**	30 (metapath2vec, heteogeneous), 61 (PTE, like advaced LINE, NLP heterogeneous), 82(IGE, dynamic, heterogeneous), **3**	24(GraphSAGE), 36(GCN), 45 (MoNet), 39(Planetoid), 105(ChebNet), 58 (GAT), 72 (EP), 37 (CLN similar to GCN), 56 (using X and E to guild RW), **9**	66 (TransNet, hinge loss, KL-divergence, AE, the way to use label), 67 (VAEbased, AE), 78 (MVC-DNE,AE) , 86 (SHINE, signed network, multipleview, AE, add-hot) **4**	88(N/A, RL, heterogeneous),90 (N/A) 93 (N/A), 85 (NEEC, expert cognition, new problem, allow any GE methods) **4**

- MF is compatible for both Naive and Information graphs.
- RW is mainly for Naive graph, whereas GC is designed for Information graph.
  In fact, RW can be seen as a special case or building block of GC.
- AE is less popular comparing to MF, RW and GC.
- Others: embedding to Hyperbolic space[53][54], relating to Graph Kernel[15][16], using GAN[89][95][102], ...

Methods Comparison for Naive Graph

1. Popularity: **RW**>**MF**>**AE** (see paper statistics)
2. Complexity: **RW**<**MF**<**AE** (RW was proven as an approximation of MF $\propto O(|V|^2)$, and with Negative Sampling $\rightarrow O(|V|)$; or with Hierarchical Softmax $\rightarrow O(|V|lg|V|)$;
3. Flexibility: **RW**>**MF**≈**AE** (RW allows us to design different intuitive sampling strategies to explore or exploit graph, and hence bias to our different interested property)
4. easy to cope with downstream tasks or end-to-end training: **RW**≈**AE**>**MF** (RW and AE were motivated by DL; easy to switch to (semi-)supervised; optimal embedding for particular task; However, for MF: obtain embedding first, then use e.g. SVM to realize downstream task)
5. easy to adapt to online learning: **RW**>**MF**≈**AE** (RW only requires sampling neighbors, but MF requires full knowledge of all nodes and AE inputs are fixed at # of nodes)
6. easy to parallelize: **RW**>**AE**>**MF** (both sampling and training of RW can be parallelized; refer to its formulation, no parameter sharing! but AE and MF need e.g. TensorFlow)
7. able to capture nonlinearity: **AE**>**RW**≈**MF** (only AE is equipped with nonlinear mapping)
8. able to generalize to unseen nodes: **AE**>**RW**≈**MF** (AE shares training parameters all the time, whereas RW and MF may need additional regularization on training parameters)
9. easy to perform theoretical analysis: **MF**>**RW**≈**AE** (MF is strictly based on Linear Algebra)
10. easy to capture global structure: **MF**≈**AE**>**RW** (RW only obtains limited neighbors)

Methods Comparison for Information Graph

Here we will particularly address GC since it can naturally involve auxiliary info and it is ideal for Information Graph, whereas in case of Information Graph, RW, MF and AE all need to additionally consider how to merge graph structure info and graph auxiliary info.

1. GC and AE are equipped with nonlinear mapping functions $\rightarrow$ capture nonlinearity
2. GC and AE share training parameters $\rightarrow$ better generalizability
3. GC, RW and AE were motivated by DL $\rightarrow$ end-to-end training with downstream task
4. GC, similar to RW, also needs sampling and SkipGram for training $\rightarrow$ parallelizable
5. GC, similar to RW, does not require full knowledge of all nodes $\rightarrow$ online learning
6. GC, similar to RW, employs stochastic way to approximate graph structure, and it only investigates limited neighbors $\rightarrow$ **less** global structure awareness
7. GC can naturally involve auxiliary information;
   but for MF, AE and RW dealing with Information Graph, one common "add-hot" way:
   firstly, apply bilinear transformation T to unify A and X i.e. $\hat{Z} = ATX$;
   secondly, perform further operations on $\hat{Z}$ e.g. factorize $\hat{Z}$ or feed $\hat{Z}_i$ into AE;
   if we have more Xs, use more Ts, and then weights these terms as well as regularizes Ts

why RW and GC

From above discussion and comparison, we conclude that:
– for Naive GE problem, our preference: RW$>$MF$>$AE;
– for Information GE problem, our preference: GC$>$RW$\approx$AE$\approx$MF;

Therefore, let us now focus on **RW for Naive GE** and **GC for Information GE**:

RW for Naive Graph

- refer to *"Discussion $\rightarrow$ Details of GE methods"* for RW recent developments
- baseline algorithm: MF based[4][5], DeepWalk[8], LINE[7], Node2Vec[11];
- benchmark tasks: (multi-label) node classification, link prediction and visualization

GC for Information Graph

- refer to *"Discussion $\rightarrow$ Details of GE methods"* for GC recent developments
- baseline algorithm: Planetoid[39], GCN[36], MoNet[45], graphSAGE[24]
- benchmark tasks: (multi-label) node classification and visualization (why no link prediction?)

RW for Naive Graph, state-of-the-art

HARP[22]: Hierarchical Representation Learning for Networks

- published in *AAAI*2018 by Stony Brook Uni, Google (previous DeepWalk author) and Yahoo
- HARP is a high level algorithm or say a general framework that involves two core ideas: 1) simplify original graph to a series of successively smaller graphs by collapsing nodes into super-nodes; 2) learn coarse embedding **using any RW method** e.g. DeepWalk for the smallest graph, and that embedding acts as **good initialization** to the second small graph. **Repeat** coarse embedding until reaching at the largest graph i.e. original graph embedding.
- the dataset summary and HARP performance of node classification are as follows:

Name	DBLP	Blogcatalog	CiteSeer
# Vertices	29,199	10,312	3,312
# Edges	133,664	333,983	4,732
# Classes	4	39	6
Task	Classification	Classification	Classification

Algorithm	**Dataset**		
	DBLP	BlogCatalog	CiteSeer
DeepWalk	57.29	24.88	42.72
HARP(DW)	**61.76***	**25.90***	**44.78***
Gain of HARP[%]	**7.8**	**4.0**	**4.8**
LINE	57.76	22.43	37.11
HARP(LINE)	**59.51***	**23.47***	**42.95***
Gain of HARP[%]	**3.0**	**4.6**	**13.6**
Node2vec	62.64	23.55	44.84
HARP(N2V)	**62.80**	**24.66***	**46.08***
Gain of HARP[%]	**0.3**	**4.7**	**2.8**

NB: the author ignored classical MF based methods since RW based methods developed recently are superior.

GC for Information Graph, state-of-the-art

GAT[58]: Graph Attention Networks

- published in *ICLR*2018, by Cambridge Uni and Montreal Uni
- GAT, a convolution-style neural networks, operates on graph-structured data by leveraging marked self-attentional layers, which allows for assigning different importance (learned by attention mechanism) to different nodes within a neighborhood while dealing with different sized neighborhoods.

	Cora	Citeseer	Pubmed	PPI
Task	Transductive	Transductive	Transductive	Inductive
# Nodes	2708 (1 graph)	3327 (1 graph)	19717 (1 graph)	56944 (24 graphs)
# Edges	5429	4732	44338	818716
# Features/Node	1433	3703	500	50
# Classes	7	6	3	121 (multilabel)
# Training Nodes	140	120	60	44906 (20 graphs)
# Validation Nodes	500	500	500	6514 (2 graphs)
# Test Nodes	1000	1000	1000	5524 (2 graphs)

Transductive

Method	**Cora**	**Citeseer**	**Pubmed**
Planetoid (Yang et al., 2016)	75.7%	64.7%	77.2%
Chebyshev (Defferrard et al., 2016)	81.2%	69.8%	74.4%
GCN (Kipf & Welling, 2017)	81.5%	70.3%	**79.0%**
MoNet (Monti et al., 2016)	81.7 ± 0.5%	—	78.8 ± 0.3%
GCN-64*	81.4 ± 0.5%	70.9 ± 0.5%	**79.0** ± 0.3%
GAT (ours)	**83.0** ± 0.7%	**72.5** ± 0.7%	**79.0** ± 0.3%

Inductive

Method	**PPI**
GraphSAGE-GCN (Hamilton et al., 2017)	0.500
GraphSAGE-mean (Hamilton et al., 2017)	0.598
GraphSAGE-LSTM (Hamilton et al., 2017)	0.612
GraphSAGE-pool (Hamilton et al., 2017)	0.600
GraphSAGE*	0.768
GAT (ours)	**0.973** ± 0.002

Transductive Learning is a kind of semi-supervised learning utilizes both labeled data and unlabeled testing data during training and then, infer label for unlabeled **already-seen** testing data; whereas **Inductive** Learning is a kind of supervised learning only utilizes labeled data during training and then, learn a **classifier** to infer label for unlabeled **unseen/incoming** testing data [39][36][24][58].

RW and GC: deeper insight

RW deeper insight

- Two key steps: 1) walk on graph to generate many short node sequences (approximating graph structure by local structure info); 2) employ Maximum Likelihood Estimate to maximize the node co-occurrence probabilities for each sequence.
- For 1) we may have different random walk strategies and note that it is a stochastic process and is related to Markov chain as sampling next nodes only depend on current node.
- For 2) think of NLP where sentences $\approx$ node sequences and then Skip-gram model can be used, which is the original motivation of the first RW work i.e. DeepWalk [8].
- Not surprisingly, RW has been recently proven to be as an approximation to MF [60][44].

GC deeper insight

- Two key steps: 1) define node neighborhoods spatially or spectrally [58]; 2) extend convolution idea to graph-structured data and borrow Deep Learning framework for training.
- Intuitively, think of convolution on a $(n \times m)$ **real image i.e. fine grid graph** (gives $A \in \mathbb{R}^{nm \times nm}$ very regular and sparse) where each node is with RBG values i.e. auxiliary info (gives $X \in \mathbb{R}^{nm \times 3}$). **Graph Convolution is to generalize convolution on a irregular graph!**
- The inputs for GC, differently from RW, come from the graph auxiliary info $X \in \mathbb{R}^{n \times m}$, while graph structure info $A \in \mathbb{R}^{n \times n}$ will be injected via sampling node neighborhoods which may be similar as RW step 1) e.g. BFS-like strategy.

RW and GC: literature gap

Scalability

- In the four review papers, they all pointed out that scalability (say easily scalable to millions of nodes and billions of edges) is one future promising direction. We also checked the state-of-the-art methods that were usually tested on the graph about 10^3 to 10^5 nodes and 10^4 to 10^6 edges. Notice that: **the real-world graph nowadays might be even larger than millions of nodes and billions of edges, and the situation would be far more complicated if we are dealing with online learning or dynamic graph problems.**
- The complexity of most state-of-the-art methods is $O(cN)$ or $O(cE)$ where c is usually given by the embedding output size, the node neighborhoods being sampled and taken into calculation, and the depth of stacked mapping functions; N is the number of nodes; and E is the number of edges. Notices that: **these methods are already good enough in terms of complexity since they are proportional to the number of nodes or edges**.
- To further reduce complexity, we may consider:

  1) During sampling: # of node neighborhoods, sampling times for a node, sampling strategy; (may reduce c)
  2) During training: using all data, using partial selected data (reduce N but how to select);
  3) Embedding outputs: the dimension, real number or binary number representation (binary may be faster);
  4) Downstream Tasks: independent of embedding, unified with embedding (need deeper insight);
  5) Sparse computation: graph data are usually sparsely linked and many methods have considered that;
  6) Real implementation time: C++ or Python, distributed/parallel computing, and etc.

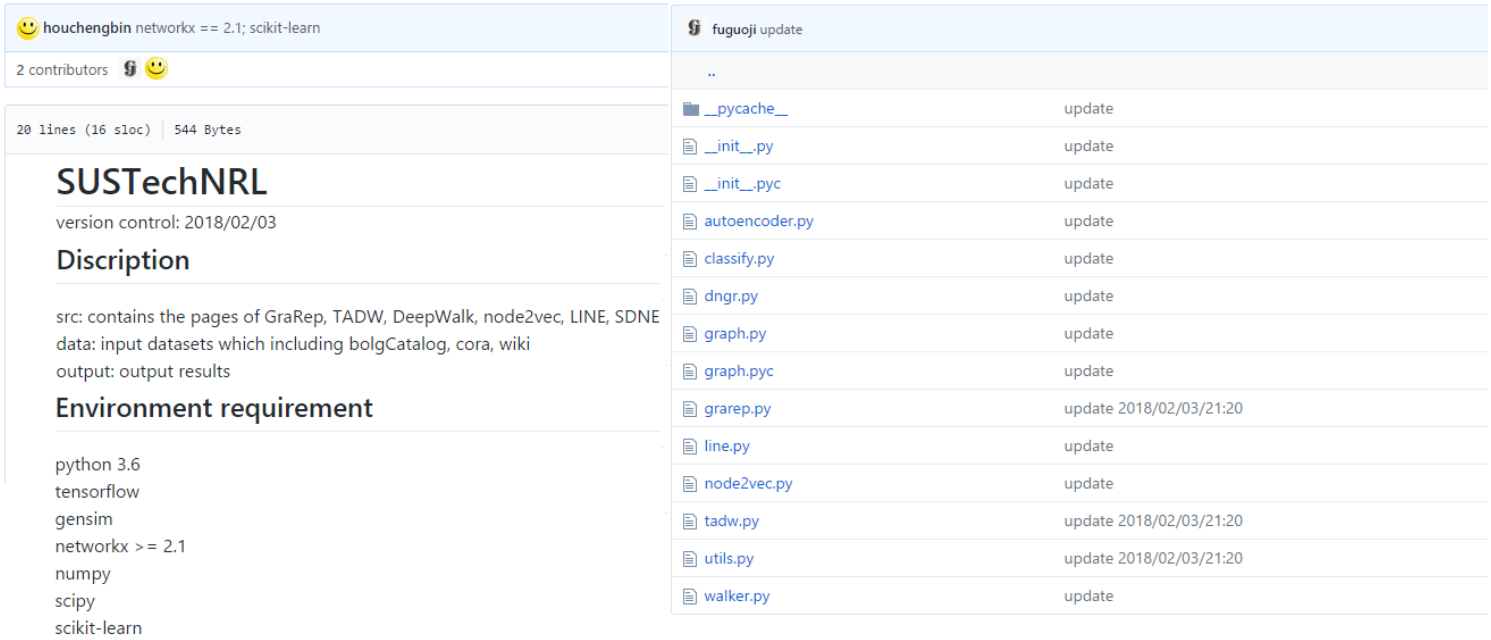
RW and GC: literature gap

Others

- Many ideas from CV and NLP have been extended to graph-structured data, but why recent methods still consider relatively shallow NN model? We know a deeper NN model should have better capability to learn a much more complicated functions and therefore may yield better classification result.
- As addressed previously, for GC method, the graph structure info will be injected via sampling strategy which basically only considers 1^{st} order proximity. Does this reason result in rare GC methods evaluated in link prediction tasks? If it is YES, how can we injected higher order proximity via sampling strategy? We need further investigation on this point.
- The challenge of RW and GC might be their nature of using limited neighborhood and hence less global structure awareness. However, HARP[22] overcame that by collapsing nodes into super-nodes and resulting in smaller graph where other state-of-the-art RW methods can be applied. Motivated by HARP, 1) is there a better way to collapse nodes? 2) can we adapt similar idea to GC method and the challenge then might be how to deal with the features of the nodes being collapsed into a super-nodes?

Current Progress

- Literature Review: we have review most related papers before 2018 and we will track recent conference papers such as AAAI2018, ICLR2018 and IJCAI2018 papers related to our research title.
- Coding: working with Guoji Fu, we have built a private Github Repository called **"SUSTechNRL"** to re-implement state-of-the-art graph embedding methods such as DeepWalk, Node2Vec, GraRep, SDNE, DNGR and TADW. And we will re-implement more methods under "SUSTechNRL" framework in the future.
- What we are planing to do next: verify and get deeper insight of literature gaps from coding and implementation perspectives.



Summary

END, to be CONT'D...

References

[1] Palash Goyal and Emilio Ferrara, "Graph Embedding Techniques, Applications, and Performance: A Survey", *IEEE TPAMI*, 2017. <14>
KEY: a review from Uni of Southern California, good categories on apps e.g. compression, visualization, clustering, classification and prediction

[2] H. Cai, V. Zheng and K. Chang, "A Comprehensive Survey of Graph Embedding: Problems, Techniques and Applications", *TKDE*, 2017. <1>
KEY: a review from UIUC and Advanced Digital Sciences Center Singapore, very detailed categories e.g. graph inputs, outputs, techniques

[3] W. Hamilton, R. Ying and J. Leskovec, "Representation Learning on Graphs: Methods and Applications", *IEEE Data Engineering Bulletin*,2017 <6>
KEY: a review from Sandford Uni SNAP group, a encoder-decoder framework, direct VS indirect embedding

[4] M. Ou, P. Cui, J. Pei, Z. Zhang, and W. Zhu, "Asymmetric transitivity preserving graph embedding," in *KDD*, 2016. <42>
KEY: HOPE, from Tsinghua-Tencent lab, kth order, asymmetric widely exists in directed graph, shot&more path gives high prob, given A using MF

[5] S. Cao, W. Lu, and Q. Xu, "Grarep: Learning graph representations with global structural information," in *CIKM*, 2015. <88>
KEY: GraRep, from IBM, kth order, DeepWalk's sampling is to translate A into linear sequences, RW as a kind of MF, given A using MF

[6] L.F.R. Ribeiro, P.H.P. Saverese, and D.R. Figueiredo, "struc2vec: Learning node representations from structural identity", in *KDD*, 2017. <7>
KEY: struc2vec, structure role may be similar but too far from each other i.e. "window can not cover", why focus on A only, given A using RW

[7] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei, "Line: Large-scale information network embedding," in *WWW*, 2015.<362>
KEY: LINE, KL divergence between empirical and stochastic prob., consider both 1st and 2nd, edge sampling according to weight, given A using RW

[8] B. Perozzi, R. Al-Rfou, and S. Skiena, "Deepwalk: Online learning of social representations," in *KDD*, 2014. <501>
KEY: DeepWalk, random walk to get neighbors, skip-gram, hierarchical SoftMax O(V^2)->O(VlogV), Zipf's Law, very first paper, given A using RW

[9] Y. LeCun, Y. Bengio, and G. Hinton, "Deep Learning", *Nature*, 2015. <5401>
KEY: a review of Deep Learning, automatic feature extraction is important, recent development in DL e.g. CNN and RNN

[10] T. Pimentel, A. Veloso and N. Ziviani, "Unsupervised and Scalable Algorithm for Learning Node Representation", in *ICLR workshop*, 2017. <2>
KEY: NBNE, idea from DeepWalk and skip-gram, the sampling is not RW but random permutation to get sequence, faster, given A using RW

References

[11] A. Grover and J. Leskovec, "Node2vec: Scalable feature learning for networks," in *KDD*, 2016. <296>
KEY: Node2Vec, biased random walk trade-off DFS (exploitation) and BFS (exploration) , Negative Sampling O(V^2)->O(KV), given A using RW

[12] N. Biggs, E. Lloyd and R. Wilson, "Graph Theory 1736-1936", *Oxford University Press*, Oxford, 1986. <1097>
KEY: Graph Theory firstly proposed by L. Euler in 1736, Seven Bridges Problem

[13] L. Cayton, "Algorithms for Manifold Learning", *Univ. of California at San Diego Tech. rep*, 2005. <206>
KEY: non-linear dim reduction e.g. manifold learning: IsoMap (geodesic + MDS), LLE (local linear), LE (Gauss heat kernel) and SDE (physical rigid rod)

[14] S. Nishana and S. Surendran, "Graph embedding and dimensionality reduction-a survey," *IJCSET*, vol. 4, no. 1, pp. 29–34, 2013. <2>
KEY: PCA, MDS, IsoMap, LLE, LE, SDE (Semidefinite E) i.e. MVU (max variance unfolding), MVE (max volume E), SPE (structure preserving E)

[15] D. Haussler, "Convolution kernels on discrete structures", *Tech Rep*, 1999. <1364>
KEY: R-convolutional, a general framework for handing discrete objects by recursively decomposing structured objects into "atomic" substructures

[16] K. M. Borgwardt and H. Kriegel, "Shortest-path kernels on graphs," in *ICDM*, 2005. <447>
KEY: trainddional based on random path cocurrence frequency...

[17] Y. Zhou, H. Cheng and J. X. Yu, "Clustering Large Attributed Graphs: An Efficient Incremental Approach", *in ICDM*, 2010. <113>
KEY: Inc-Cluster (see 41~43), attributed graphs, cluttering based on structure and attributes, for clustering particularly, given A and X using RW

[18] S. Chang, W. Han, and et al, "Heterogeneous network embedding via deep architectures", in *KDD*, 2015. <86>
KEY: HNE, structure and other multimodal info.→ in common space/unified vector representation, given A (heterogenous) and X using MF

[19] Y. Bengio, R. Ducharme, P. Vincent, and C. Janvin, "A Neural Probabilistic Language Model", *JMLR*, 2003. <2999>
KEY: this paper → Word2Vec → DeepWalk → LINE → Node2Vec, NB: graph embedding with random walks analogue to sentences in NLP

[20] T. Mikolov, I. Sutskever, K. Chen, et al. "Distributed Representations of Words and Phrases and their Compositionality", in *NIPS*, 2013. <6277>
KEY: Word2Vec, from Google, basic tool in NLP, make embedding become popular, Skip-gram language model, Hierarchical Softmax, using RW

References

[21] X. Wang, P. Cui, J. Wang, J. Pei, W. Zhu, and S. Yang, "Community preserving network embedding," in *AAAI*, 2017. <24>
KEY: M-NMF, community property, unified framework for micro and macro scope properties, given A using MF

[22] H. Chen, B. Perozzi, Y. Hu, and S. Skiena, "HARP: Hierarchical representation learning for networks", in *AAAI*, 2018. <1>
KEY: HARP, hierarchical, a meta-method: Graph Coarsening + other GE methods, like to aggregate global and local property, given A using RW

[23] X. Wei, L. Xu, B. Cao, and P. S. Yu, "Cross view link prediction by learning noise-resilient representation consensus," in *WWW*, 2017. <7>
KEY: CVLP, new problem, jointly optimize: node with (partially observable) links and/or attributes, theoretical bound, given A and X using MF

[24] W.L. Hamilton, R. Ying, and J. Leskovec. "Inductive representation learning on large graphs," in *NIPS*, 2017. <11>
KEY: GraphSAGE, sampling fixed neighborhood iteratively, then aggregating by mean/max pooling or LSTM, given A and X using RW

[25] X. Huang, J. Li and X. Hu, "Label Informed Attributed Network Embedding", in *WSDM*, Cambridge, UK, 2017. <21>
KEY: LANE, 3 objectives: labels + attributes + geometric/topological (weighted sum), affiliate labels due to heterogenous, given A, X and Y using MF

[26] C. Zhou, Y. Liu, X. Liu, Z. Liu, and J. Gao, "Scalable graph embedding for asymmetric proximity," in *AAAI*, 2017. <5>
KEY: APP, asymmetric also in undirected graph if by RW, from Alibaba, sampling by PangeRank & MC, why RW instead of MF, given A using RW

[27] Y. Lin, Z. Liu, M. Sun, Y. Liu, and X. Zhu, "Learning entity and relation embeddings for knowledge graph completion," in *AAAI*, 2015 <262>
KEY: Trans-R, here Trans-R projects entities and relations to two different planes, while Trans-H and Trans-E project to the same plane

[28] B. Shi and T. Weninger, "Proje: Embedding projection for knowledge graph completion," in *AAAI*, 2017, pp. 1236–1242. <12>
KEY: ProjE

[29] Z. Wang, J. Zhang, J. Feng, and Z. Chen, "Knowledge graph embedding by translating on hyperplanes," in *AAAI*, 2014. <256>
KEY: Trans-H, here Trans-H for 1-N, N-1 and N-N cases by projection to a hyperplane, while previous Trans-E for only 1-1 case

[30] Y. Dong, N.V. Chawla, and A. Swami, "metapath2vec: Scalable representation learning for heterogeneous networks", in *KDD*, 2017. <14>
KEY: metapath2vec, formalize the problem of heterogenous, given A (heterogenous) using RW

References

[31] S. Yan, D. Xu, B. Zhang, H. Zhang, Q. Yang, and S. Lin, "Graph embedding and extensions: a general framework for dimensionality reduction", *IEEE TPAMI*, 2007. <2218>
KEY: a review, non-relational data -> constructed as graph, graph embedding as a framework for developing dimensionality reduction techniques

[32] A. Ahmed, N. Shervashidze, and et al, "Distributed large-scale natural graph factorization," in *WWW*, 2013, pp. 37–48. <58>
KEY: small Graph Factorization by MF via SGD, graph partitioning, asynchronous and distributed mechanism, from Google, given A using MF

[33] M. Belkin and P. Niyogi, "Laplacian eigenmaps and spectral techniques for embedding and clustering," in *NIPS*, 2001. <3090>
KEY: LE, Laplacian Eigenmaps, LE: *Wij||Yi-Yj||*, LLE: *||Yi–Wij Yj||*, LE form Wij by k-nearest neighbors, given A using MF

[34] D. Wang, P. Cui and W. Zhu, "Structural Deep Network Embedding", in *KDD*, 2016. <113>
KEY: SNDE, first work using AE, 2nd for global (sparse->more weight to non-zero,) + 1st for local + reg, given A using AE

[35] S. Cao, W. Lu, and Q. Xu, "Deep neural networks for learning graph representations," in *AAAI*, 2016. <36>
KEY: DNGR, stacked denoising autoencoder, PMI → PPMI matrix, random surfing motivated by PageRank, RW drawbacks, given A using AE

[36] T.N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks", in *ICLR*, 2016. <81>
KEY: GCN, sampling based on A, aggregating based on X, semi-supervised by Y, given A, X and Y using RW

[37] T. Pham, T. Tran, D.Q. Phung, and S. Venkatesh, "Column networks for collective classification," In *AAAI*, 2017 <8>
KEY: CLN, Column Network, knowledge graph for relation data?????????????, given A, X, Y and E using RW

[38] N. Shervashidze, P. Schweitzer, E. J. van Leeuwen, K. Mehlhorn, and K. M. Borgwardt, "Weisfeiler-lehman graph kernels." *JMLR*, 2011. <320>
KEY: Weisfeiler-lehmann kernel, construct subgraphs by labels where label generated by Weisfeiler-lehmann subtree

[39] Z. Yang, W. Cohen, and R. Salakhutdinov, "Revisiting semi-supervised learning with graph embeddings," In *ICML*, 2016. <84>
KEY: Planetoid, inductive embedding, sampling based on A, aggregating based on X, semi-supervised by Y, given A, X and Y using RW

[40] C. Dyer, "Notes on Noise Contrastive Estimation and Negative Sampling", *Computer Science*, 2014. <15>
KEY: Negative Sampling is simplified version of NCE (has theoretical guarantee)

References

[41] M. E. Newman and M. Girvan, "Finding and evaluating community structure in networks," *Physical review E*, 2004. <9501>
KEY: clustering for community

[42] X. Xu, N. Yuruk, Z. Feng, and T. A. Schweiger, "SCAN: a structural clustering algorithm for networks," in *KDD*, 2007, pp. 824–833 <573>
KEY: SCAN, clustering for structure role

[43] Y. Zhou, H. Cheng, and J. X. Yu, "Graph clustering based on structural/attribute similarities," in *VLDB*, vol. 2, no. 1, pp. 718–729, 2009. <524>
KEY: SA-cluster, clustering for attributes

[44] C. Tu, W. Zhang, Z. Liu, and M. Sun, "Max-Margin DeepWalk: Discriminative learning of network representation," in *IJCAI*, 2016. <31>
KEY: MMDW, see [60] prove RW as MF, but here use simplified MF approximation + label, semi-supervised, given A and Y using MF

[45] F. Monti, D. Boscaini, et al ` Bronstein, "Geometric deep learning on graphs and manifolds using mixture model cnns," in *CVPR*, 2017. <31>
KEY: MoNet, adapt DL and CNN to non-Euclidean domain e.g. graph and manifold, given A and X using RW

[46] A. Clauset, C. Moore, and M. E. Newman, "Hierarchical structure and the prediction of missing links in networks," *Nature*, 2008. <1422>
KEY: link prediction, tradition way, the hierarchical structure

[47] L. Lu and T. Zhou, "Link prediction in complex networks: A ¨survey," *Physica A: Statistical Mechanics and its Applications*, 2011. <1185>
KEY: link prediction, tradition way, a suvery

[48] L. van der Maaten and G. Hinton., "Visualizing data using t-SNE", *JMLR*, 2008 <3776>
KEY: data visualization, t-SNE

[49] M.C.F. De Oliveira and H. Levkowitz, "From visual data exploration to visual data mining: a survey", *IEEE TVCG*, 9(3):378–394, 2003 <531>
KEY: data visualization

[50] H. Toivonen, F. Zhou, A. Hartikainen, and A. Hinkka, "Compression of weighted graphs," in *KDD*, 2011, pp. 965–973. <81>
KEY: data compression

References

[51] P. Yanardag and S. Vishwanathan, "Deep graph kernels," in *KDD*, 2015, pp. 1365–1374. <73>
KEY: DGK, Deep Graph Kernels, combine ideas from Graph kernel and Language model, subgraph embedding given A using RW

[52] A. Muscoloni, "Machine learning meets complex networks via coalescent embedding in the hyperbolic space," *Nature Communication*, 2017
KEY: see [54]

[53] B.P. Chamberlain, J. Clough, and M.P. Deisenroth, "Neural embeddings of graphs in hyperbolic space", *arXiv preprint*, 2017. <3>
KEY: hyperbolic space, modify the inner-product decoder of node2vec to in hyperbolic rather than Euclidean, Imperial College, given A

[54] A. Muscoloni, J.M. Thomas, S. Ciucci, et al. "Machine learning meets complex networks via coalescent embedding in the hyperbolic space", *Nature Communication*, 2017
KEY: hyperbolic, radius -> node centrality; angular displacement -> topological proximity, interesting visualization, given A

[55] J. Tang, J. Liu, M. Zhang, and Q. Mei, "Visualizing large-scale and high-dimensional data", in *WWW*, 2016. <41>
KEY: data visualization, modified t-SNE and compare to LINE, best paper nomination

[56] L. Backstrom and J. Backstrom, "Supervised random walks: predicting and recommending links in social networks", in *WSDM*, 2011. <735>
KEY: a way to guild random walks by training functions according to attributes which give a probability matrix (like PageRank), given A X E using RW

[57] H. Zhao, L. Du, and W. Buntine, "Leveraging Node Attributes for Incomplete Relational Data", in *ICML*, 2017. <1>
KEY: NARM, use node attribute to improve link prediction accuracy, not graph embedding, a generative model

[58] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph Attention Networks", in *ICLR*, 2018 <under review>
KEY: GAT, from Cambridge&Montreal, different from graphSAGE, sampling all neighborhood, use attention mechanism to 1) calculate a weight for each pair of nodes based on their attribute similarity; and 2) use these weights to do aggregation, given A X using RW

[59] Non-transitive Hashing with Latent Similarity Components. *Mingdong Ou, Peng Cui, Fei Wang, Jun Wang, Wenwu Zhu.* KDD 2015. paper
KEY: not for Graph Embedding, non-transitive similarity due to various latent similarity components, given X to obtain A?

References

[60] Network Representation Learning with Rich Text Information. *Cheng Yang, Zhiyuan Liu, Deli Zhao, Maosong Sun, Edward Y. Chang.* IJCAI 2015. paper code KEY: TADW, the DW history and prove how DW as MF, given A and X using MF

[61] PTE: Predictive Text Embedding through Large-scale Heterogeneous Text Networks. *Jian Tang, Meng Qu, Qiaozhu Mei.* KDD 2015. paper code KEY: PTE, fully unsupervised by using A only might not be good, fill the gap and semi-supervised, given A Y using RW

[62] Asymmetric Transitivity Preserving Graph Embedding. *Mingdong Ou, Peng Cui, Jian Pei, Ziwei Zhang* KDD 2016. paper KEY: see [4]

[63] Max-Margin DeepWalk: Discriminative Learning of Network Representation. *Cunchao Tu, and etc.* IJCAI 2016. paper code KEY: see [44]

[64] CANE: Context-Aware Network Embedding for Relation Modeling. *Cunchao Tu, Han Liu, Zhiyuan Liu, Maosong Sun.*ACL 2017. paper code KEY: CANE, context-aware: the embedding will change according to different contexts, see Figure 2 convolution pooling, given A X using RW

[65] Fast Network Embedding Enhancement via High Order Proximity Approximation. *Cheng Yang, Maosong Sun, Zhiyuan Liu, Cunchao Tu.* IJCAI 2017. paper code KEY: NEU, meta-method for GEs, K-th order proximity, better if higher order encoded into the proximity matrix, given A using MF

[66] TransNet: Translation-Based Network Representation Learning for Social Relation Extraction. *Cunchao Tu, Zhengyan Zhang, Zhiyuan Liu, Maosong Sun.* IJCAI 2017. paper code KEY: TransNet, use edge information as guild, similar to Trans-E in KG, given A E using AE (notMFRW)

[67] Variation Autoencoder Based Network Representation Learning for Classification. *Hang Li, Haozheng Wang, Zhenglu Yang, Masato Odagaki.* ACL 2017. paper KEY: VAEbased, train the parameter of distribution for A and X under unifed VAE, given A X using VAE (notMFRW)

[68] Learning from Labeled and Unlabeled Vertices in Networks. *Wei Ye,Linfei Zhou,Dominik Mautz,Claudia Plant,Christian Böhm.* KDD2017 paper KEY: wvGN: weighted-vote Geometric Neighbor Classifier, given A and Y using RW

[69] Unsupervised Feature Selection in Signed Social Networks. *Kewei Cheng, Jundong Li, Huan Liu.* KDD 2017. paper KEY: SignedFS, consider positive and negative link, new problem, given A_neg, A_pos, and X using MF

[70] Accelerated Attributed Network Embedding. *Xiao Huang, Jundong Li, Xia Hu.* SDM 2017. paper code KEY: AANE, distributed way to accelerate, see figure 1, given A X using MF

References

[71] Preserving Proximity and Global Ranking for Node Embedding. *Yi-An Lai, Chin-Chi Hsu, Wenhao Chen, Mi-Yen Yeh, Shou-De Lin.* NIPS 2017.
KEY: PRUNE, new property global node importance ranking, k-th order means k-hop distinct path via RW, see Table 1 summary, given A using MF

[72] Learning Graph Embeddings with Embedding Propagation. *Alberto Garcia Duran, Mathias Niepert.* NIPS 2017.paper
KEY: EP, GCN like is a special case of Message Passing Neural Network (MPNN), EP unsupervised, given A X using GC

[73] Name Disambiguation in Anonymized Graphs using Network Embedding. *Baichuan Zhang, Mohammad Al Hasan.*CIKM 2017. paper
KEY: specially study Name Disambiguation problem i.e. use A to solve problem of given Y but disambiguation, given A and Y using RW

[74] Enhancing the Network Embedding Quality with Structural Similarity. *Tianshu Lyu, Yuan Zhang, Yan Zhang.* CIKM 2017. paper
KEY: SNS, the weakness of RW, target node's neighbors and their similar nodes as input, given A using RW

[75] Attributed Signed Network Embedding. *Suhang Wang, Charu Aggarwal, Jiliang Tang, Huan Liu.* CIKM 2017. paper
KEY: SNEA, similar to [69] the same author, given A X using MF

[76] Attributed Network Embedding for Learning in a Dynamic Environment. *Jundong Li, Harsh Dani, Xia Hu, Jiliang Tang, Yi Chang, Huan Liu.* CIKM 2017. paper KEY: DANE, dynamic graph!!!, given A X and time using MF

[77] HIN2Vec: Explore Meta-paths in Heterogeneous Information Networks for Representation Learning. *Tao-yang Fu,* CIKM 2017.
KEY: HIN2Vec, different from defining metapath in [30], learning also the metapath (node relations) to help node embedding, given A using RW

[78] From Properties to Links: Deep Network Embedding on Incomplete Graphs. *Dejian Yang, Senzhang Wang, Chaozhuo Li, Xiaoming Zhang, Zhoujun Li.* CIKM 2017. KEY: MVC-DNE, given A X using AE

[79] An Attention-based Collaboration Framework for Multi-View Network Representation Learning. *Meng Qu, Jian Tang, Jingbo Shang, Xiang Ren, Ming Zhang, Jiawei Han.* CIKM 2017. paper KEY: MVE, attention gives weights to multi views, given A1 A2 A3 ... LINE+Attention, using RW

[80] On Embedding Uncertain Graphs. *Jiafeng Hu, Reynold Cheng, Zhipeng Huang, Yixang Fang, Siqiang Luo.* CIKM 2017.paper
KEY: URGE, for uncertain graph, Jaccard Similarity, given A and uncertainty using RW

References

[81] Multi-view Clustering with Graph Embedding for Connectome Analysis. *Guixiang Ma, Lifang He, Chun-Ta Lu, Weixiang Shao, Philip S Yu, Alex D Leow, Ann B Ragin.* CIKM 2017. paper KEY: MCGE, multi-view clustering of graph instances → GE → clustering, given A and.. using MF

[82] Learning Node Embeddings in Interaction Graphs. *Yao Zhang, Yun Xiong, Xiangnan Kong, Yangyong Zhu.* CIKM 2017. paper
KEY: IGE, Interaction Graph Embedding, dynamic and consider time changing, given A E using RW

[83] Learning Community Embedding with Community Detection and Node Embedding on Graphs. *Sandro Cavallari, Vincent W. Zheng, Hongyun Cai, Kevin ChenChuan Chang, Erik Cambria.* CIKM 2017. paper code
KEY: comE, community embedding <-> node embedding, , see [21] M-NMF based on MF but here, given A using RW

[84] Network Embedding as Matrix Factorization: Unifying DeepWalk, LINE, PTE, and node2vec. *Jiezhong Qiu, Yuxiao Dong, Hao Ma, Jian Li, Kuansan Wang, Jie Tang.* WSDM 2018. paper KEY: NetMF, RW as a special case of MF, given A using MF

[85] Exploring Expert Cognition for Attributed Network Embedding. *Xiao Huang, Qingquan Song, Jundong Li, Xia Ben Hu.* WSDM 2018.
KEY: NEEC, given A X + expert cognition (installed learned from database queries)

[86] SHINE: Signed Heterogeneous Information Network Embedding for Sentiment Link Prediction. *Hongwei Wang, Fuzheng Zhang, Min Hou, Xing Xie, Minyi Guo, Qi Liu.* WSDM 2018. KEY: SHINE, signed network, Multiview graph, see [69] [75], see fig 2&3&4, given A X + sign using AE

[87] Multidimensional Network Embedding with Hierarchical Structures. *Yao Ma, Zhaochun Ren, Ziheng Jiang, Jiliang Tang, Dawei Yin.* WSDM 2018. KEY: MINES, from JD, a new problem, given A and others (multidimension relation due to heterogenous and hierarchical info.) using RW

[88] Curriculum Learning for Heterogeneous Star Network Embedding via Deep Reinforcement Learning. *Meng Qu, Jian Tang, Jiawei Han.* WSDM 2018. KEY: currently unavailable, gauss: given A X using RL

[89] Adversarial Network Embedding. *Quanyu Dai, Qiang Li, Jian Tang, Dan Wang.* AAAI 2018. paper
KEY: ANE, see Figure2, two parts: structure persevering + adversarial learning (generator + discriminator): , given A using GAN

[90] COSINE: Community-Preserving Social Network Embedding from Information Diffusion Cascades. *Yuan Zhang, Tianshu Lyu, Yan Zhang.* AAAI 2018. KEY: currently unavailable, gauss: given A X

References

[91] Dynamic Network Embedding by Modeling Triadic Closure Process. *Lekui Zhou, Yang Yang, Xiang Ren, Fei Wu, Yueting Zhuang.* AAAI 2018. paper KEY: DynamicTriad, use closed and open triad to capture dynamic, given A and using RW (dynamic)

[92] Multi-facet Network Embedding: Beyond the General Solution of Detection and Representation. *Liang Yang, Xiaochun Cao, Yuanfang Guo.* AAAI 2018. KEY: MNE, good summary in Table 1, given A using MF

[93] RSDNE: Exploring Relaxed Similarity and Dissimilarity from Completely-imbalanced Labels for Network Embedding. *Zheng Wang, Xiaojun Ye, Chaokun Wang, YueXin Wu, Changping Wang, Kaiwen Liang.* AAAI 2018. KEY: currently unbailable, gauss: given A Y

[94] Link Prediction via Subgraph Embedding-Based Convex Matrix Completion. *Zhu Cao, Linlin Wang, Gerard De melo.* AAAI 2018. KEY: currently unbailable

[95] Generative Adversarial Network based Heterogeneous Bibliographic Network Representation for Personalized Citation Recommendation. *J. Han, Xiaoyan Cai, Libin Yang.* AAAI 2018. KEY: currently unbailable, gauss: using GAN

[96] DepthLGP: Learning Embeddings of Out-of-Sample Nodes in Dynamic Networks. *Jianxin Ma, Peng Cui, Wenwu Zhu.* AAAI 2018. paper KEY: DepthLGP, given A using RW (dynamic for unseen node)

[97] Structural Deep Embedding for Hyper-Networks. *Ke Tu, Peng Cui, Xiao Wang, fei Wang, Wenwu Zhu.* AAAI 2018.paper KEY: DHNE, graph node embedding for hyper graph, given A using AE&DL (hypergraph)

[98] TIMERS: Error-Bounded SVD Restart on Dynamic Networks. *Ziwei Zhang, Peng Cui, Jian Pei, Xiao Wang, Wenwu Zhu.* AAAI 2018. paper KEY: TIMERS, a good review of SVD in GE, follow SVD especially tackle incremental SVD limitation, given A using MF (dynamic)

[99] Community Detection in Attributed Graphs: An Embedding Approach. *Ye Li, Chaofeng Sha, Xin Huang, Yanchun Zhang.* AAAI 2018. KEY: currently unbailable, gauss: given A X

[100] Bernoulli Embeddings for Graphs. *Vinith Misra, Sumit Bhatia.* AAAI 2018. paper KEY: BE, different from all previous ones, find binary embedding, given A using RW (using NCE not NS)

References

[101] Distance-aware DAG Embedding for Proximity Search on Heterogeneous Graphs. *Zemin Liu, Vincent W. Zheng, Zhou Zhao, Fanwei Zhu, Kevin Chen-Chuan Chang, Minghui Wu, Jing Ying.* AAAI 2018.
KEY: currently unbailable, gauss:

[102] GraphGAN: Graph Representation Learning with Generative Adversarial Nets. *Hongwei Wang, jia Wang, jialin Wang, MIAO ZHAO, Weinan Zhang, Fuzheng Zhang, Xie Xing, Minyi Guo.* AAAI 2018. paper
KEY: generative VS discriminative, G generate "fake" pairs, P sample real pairs, D distinguish them till not distinguishable, given A using GAN

[103] HARP: Hierarchical Representation Learning for Networks. *Haochen Chen, Bryan Perozzi, Yifan Hu, Steven Skiena.*AAAI 2018. paper
KEY: see [22]

[104] Representation Learning for Scale-free Networks. *Rui Feng, Yang Yang, Wenjie Hu, Fei Wu, Yueting Zhuang.* AAAI 2018. paper
KEY: DP-Walker/Spectral, DP: degree penalty, macroscopic property: scale-free i.e. follow a heavy-tailed distribution, given A using MF or RW

[105] M. Defferrard, X. Bresson and P. Vandergheynst, “Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering”, NIPS2016
KEY: ChebNet, CNN on graph

[106] L. Lin, “Cross-Domain Visual Matching via Generalized Similarity Measure and Feature Learning”, IEEE TPAMI 2017