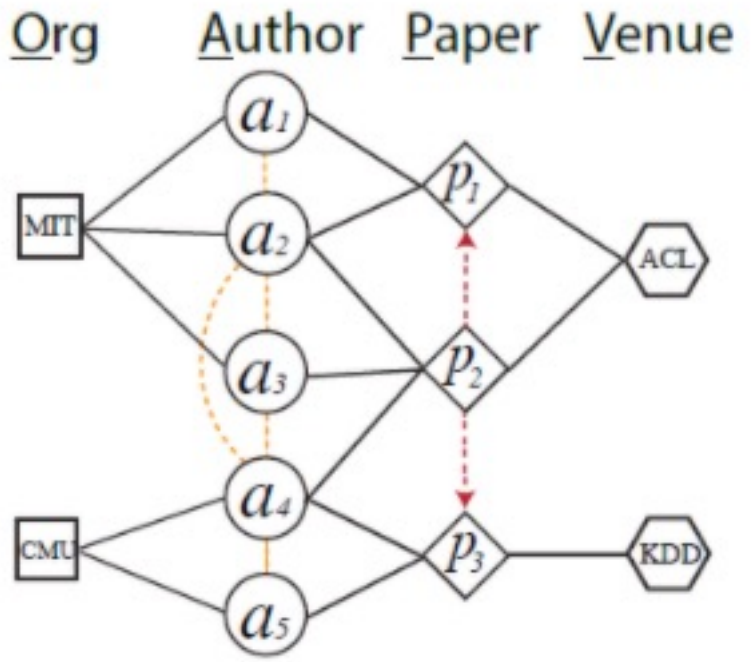
Metapath2vec: Scalable Representation Learning for Heterogeneous Networks

2022. 01. 25. Seung Yoon Choi

CONTENTS

Heterogeneous Information Network

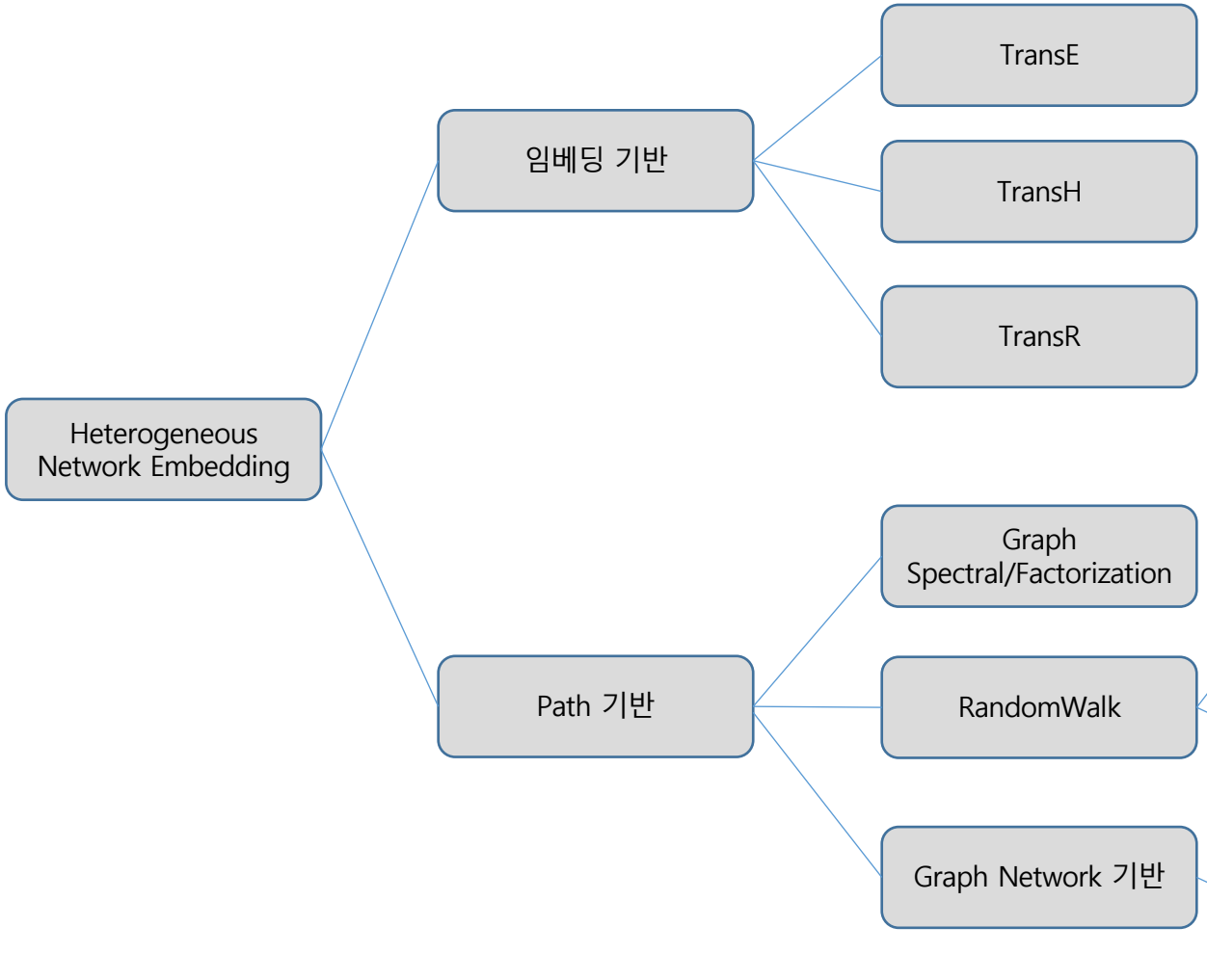


- 같은 노드지만 다양한 관계를 가질 수 있음
- 노드들 간의 관계를 통해 다른 정보를 추출
- 다양한 type 노드를 통해 이웃을 다양하게 정의

Node : authors(A), papers(P), venues(V), organizations(O)

Edges : coauthor(A-A), publish(A-P, P-V), affiliation(O-A) relationships.

Heterogeneous Network Embedding



Relation의 정보가 임베딩 공간에서 보존 되도록 학습

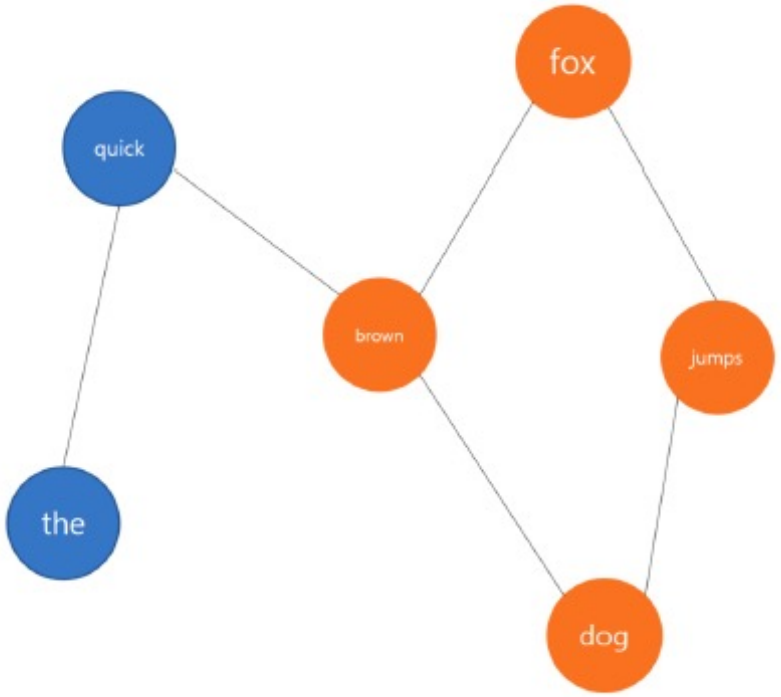
서로 다른 Relation 정보를 다르게 학습

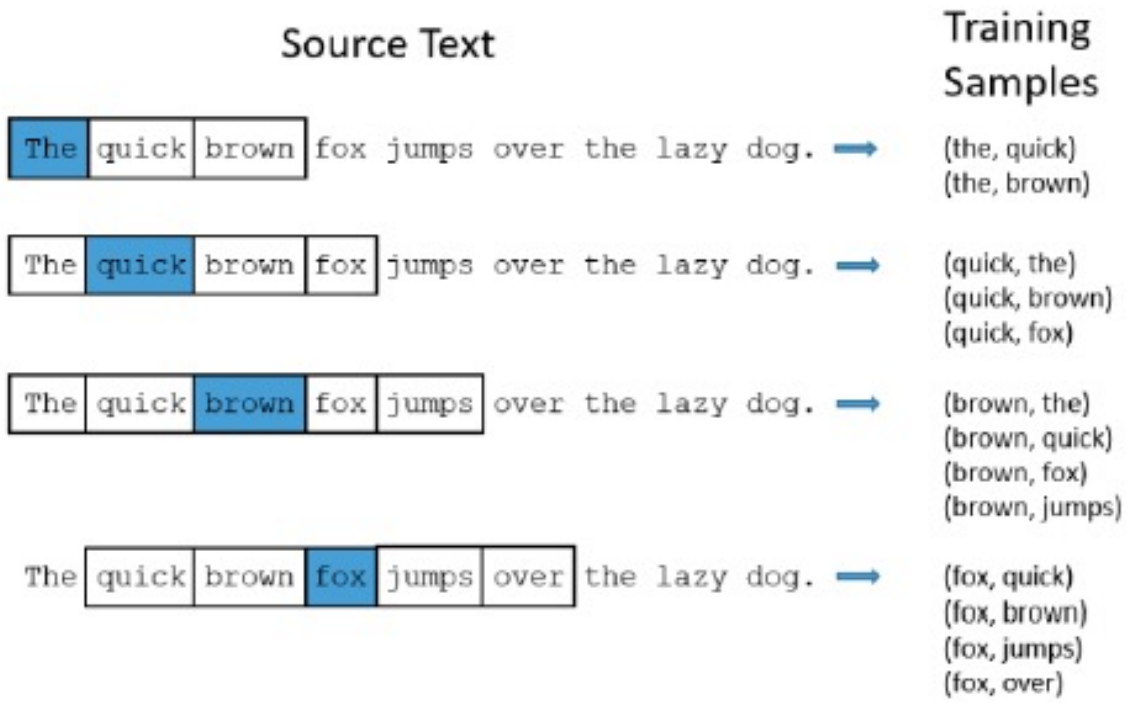
Entity와 Relation을 서로 다른 공간에서 학습

모든 노드를 같은 type으로 간주
연결된 노드의 빈도 기반 유사성 학습

선행지식 기반으로, path를 정의
이후 Deepwalk와 같은 방식

DeepWalk & Skip-gram

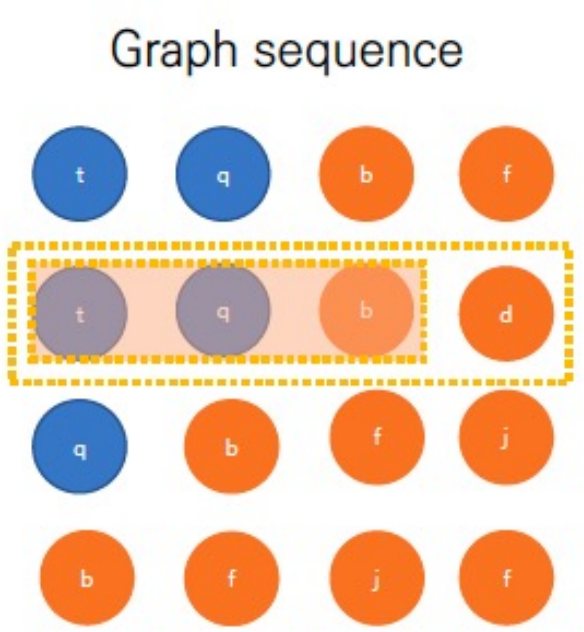


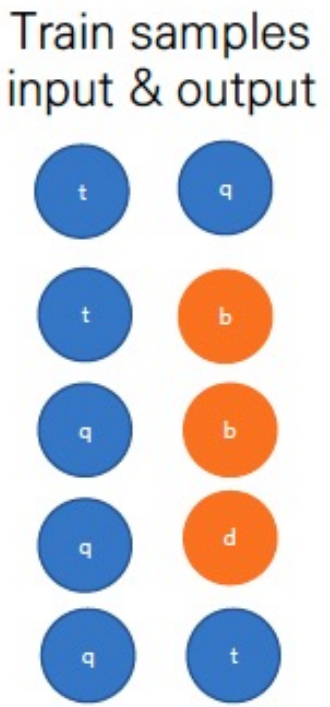


그래프 source --> Sequence 생성 --> Skip-gram 등의 방식 학습

01 Background

DeepWalk & Skip-gram





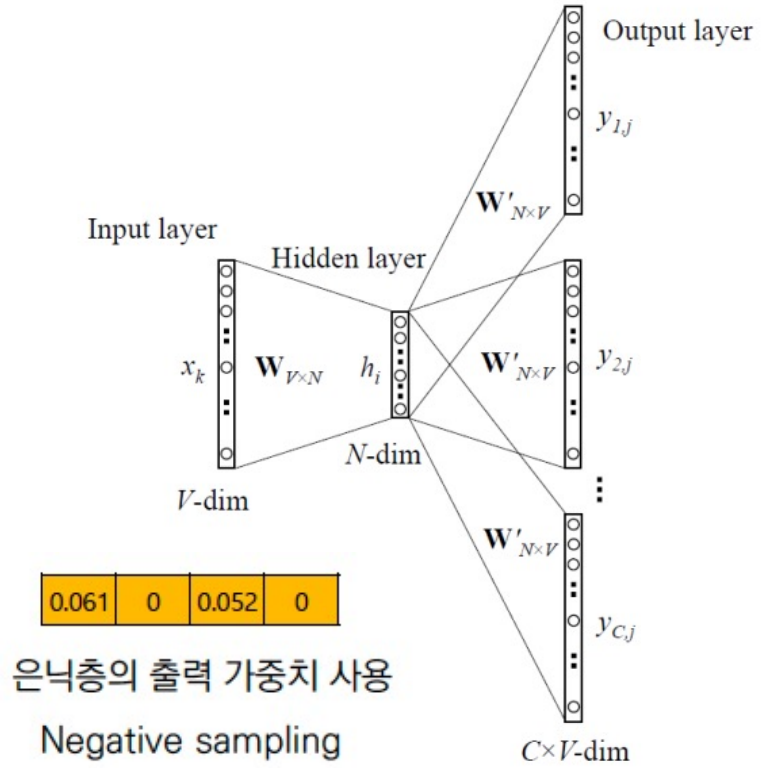
Training Samples

(the, quick)
(the, brown)

(quick, the)
(quick, brown)
(quick, fox)

(brown, the)
(brown, quick)
(brown, fox)
(brown, jumps)

(fox, quick)
(fox, brown)
(fox, jumps)
(fox, over)

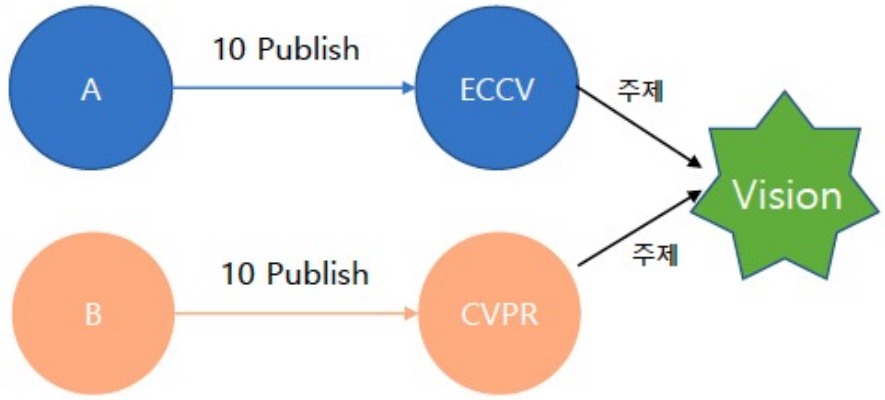


일반적으로, 한 노드로부터 32~64개 sequence 생성

생성된 sequence 길이 : 약 40

Window : 10

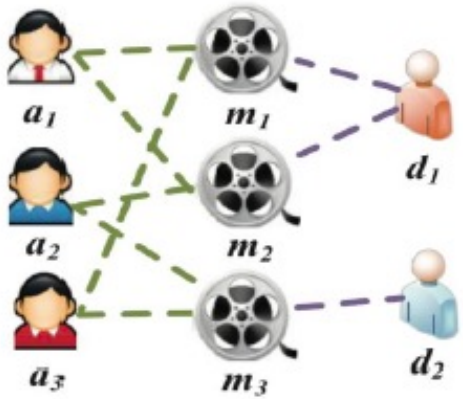
DeepWalk (Homogeneous)의 한계



A와 B는 비슷한 연구를 하는 사람으로 추정 가능
-> **ECCV와 CVPR은 유사한 Vision의 학회**

물론, 다른 사람들이 ECCV, CVPR에 동시에 기재를 많이 해서 두 학회가 비슷하게 학습됨 -> 결국 A와B 또한 유사함 (이전의 방식)

학회의 다른 정보를 반영하여 A와B가 유사함을 학습 (Metapath 방식)



Path1 : 같은 배우가 출연한 작품 (M-A-M)
Path2 : 같은 감독의 작품 (M-D-M)

Node & relation type이 다를 때, 이전의 Deepwalk 기반 방법론 적용은 문제 야기

Actor와 Movie라는 node의 특성이 비슷하게 임베딩 됨 (비슷한 공간에 위치)

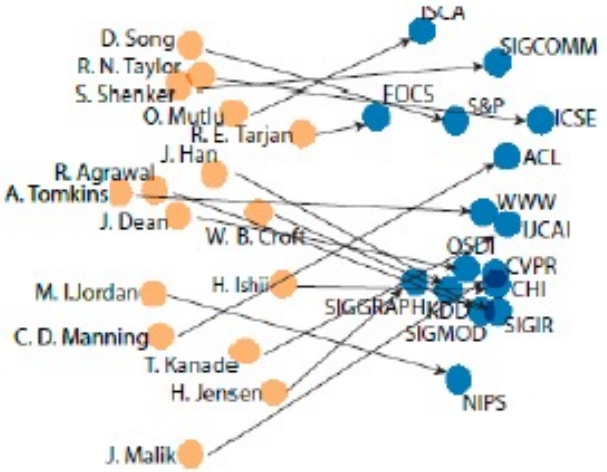
Node 별 특성에는 구별되는 고유한 정보가 있음

DeepWalk (Homogeneous)의 한계

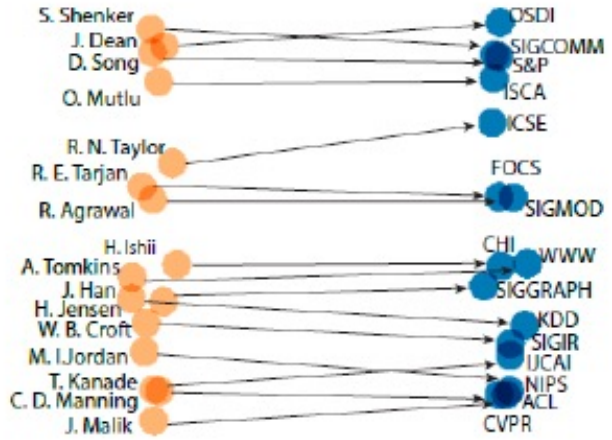
문제의 이유 : 다른 type (다른 정보) Node가 같은 공간에 임베딩 되기 때문

DeepWalk의 이웃 (유사 node) : 해당 node 기준 근처에 있는 노드
이웃 node와 타겟 node는 비슷하게 임베딩 되고, 이는 확률 기반

다른 Type의 Node에서 이웃을 새롭게 정의함



(a) DeepWalk / node2vec



(d) *metapath2vec++*

- Node의 Type 구별이 더 잘됨
- 화살표의 방향을 통해, 동일한 관계는 동일하게 표현 (화살표의 방향과 크기)

Network Representation Learning Framework

- Word2vec 기반의 network representation learning framework
  - DeepWalk, LINE, node2vec 등
  - Raw network로 부터 유용하고 의미있는 latent feature를 발견함
  - Homogeneous network 기반의 algorithm

- Metapath-guided RandomWalk Strategy
  - 여러 다른 type의 node/relation에서 구조적이고 의미론적인 상관관계를 포착
  - 복수의 node type 맥락 속에서 skip-gram 기반으로 network probability를 최대화
  - 효과적이고 효율적인 heterogenous negative sampling 기법 적용
  - Similarity search, node classification, clustering 등에 적용 가능

- Two methods
  - Metapath2vec : 모든 node type에 대해서 같은 space에서 embedding을 생성
  - Metapath2vec++ : 각 node type 마다 별도의 space에서 embedding 생성

Problem Definition

Definition 2.1. **A Heterogeneous Network** is defined as a graph $G = (V, E, T)$ in which each node v and each link e are associated with their mapping functions $\phi(v) : V \rightarrow T_V$ and $\varphi(e) : E \rightarrow T_E$, respectively. T_V and T_E denote the sets of object and relation types, where $|T_V| + |T_E| > 2$.

PROBLEM 1. ***Heterogeneous Network Representation Learning:*** *Given a heterogeneous network G, the task is to learn the d-dimensional latent representations $\mathbf{X} \in \mathbb{R}^{|V| \times d}$, $d \ll |V|$ that are able to capture the structural and semantic relations among them.*

03 Metapath2vec Framework

Homogeneous Network Embedding

- Leverage random walks and utilize the skip-gram model
- Representation of a node that facilitates the prediction of its structural context in a homogeneous network.

- Network $G = (V, E)$
  - ➢ Objective is to maximize the network probability

$$\arg\max_{\theta} \prod_{v \in V} \prod_{c \in N(v)} p(c|v;\theta)$$

$N(v)$: neighborhood of node v in the network G
$p(c|v;\theta)$: conditional probability of having a context node c given a node v

Heterogeneous Network Embedding: metapath2vec

- Skip-gram 모델의 아이디어와 마찬가지로 중심 node가 있을 때 주변 node의 특성을 학습.
- 이 때, 적절한 node type에 따라 학습.

$$\arg\max_{\theta} \sum_{v \in V} \sum_{t \in T_V} \sum_{c_t \in N_t(v)} \log p(c_t | v; \theta)$$

$N_t(v)$: v's neighborhood with the t^{th} type of nodes

$p(c_t|v;\theta) = \frac{e^{X_{c_t} X_v}}{\sum_{u \in V} e^{X_u X_v}}$: softmax function

X_v : v^{th} row of X

Heterogeneous Network Embedding: metapath2vec

- Negative Sampling 사용
  - ➢ Node type과 상관없이 negative sample을 추출

$$\log \sigma(X_{c_t} \cdot X_v) + \sum_{m=1}^{M} \mathbb{E}_{u^m \sim P(u)} [\log \sigma(-X_{u^m} \cdot X_v)]$$

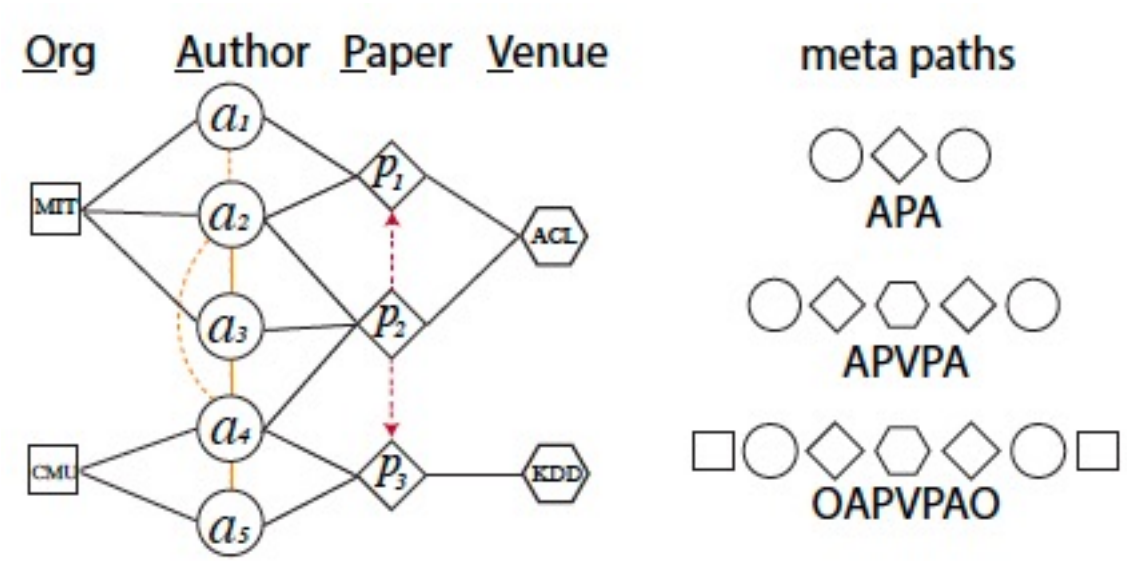
$\sigma(x) = \frac{1}{1+e^{-x}}$: sigmoid function

M개의 negative sample을 추출

Heterogeneous Network Embedding: metapath2vec

Meta-path scheme $\mathcal{P}$

$$V_1 \xrightarrow{R_1} V_2 \xrightarrow{R_2} \cdots V_t \xrightarrow{R_t} V_{t+1} \cdots \xrightarrow{R_{l-1}} V_l$$



- APVPA : 2명의 작가가 어떤 paper를 냈고, 이들이 같은 venue에서 accept 되었다는 것
- metapath가 domain 지식을 통해 사전에 설정되어야 하는 단점
- Graph Transformer Networks : meta-path를 사전에 설정하지 않아도 자동으로 찾는 기법

Heterogeneous Network Embedding: metapath2vec

Transition Probability

$$p(v^{i+1}|v_t^i, \mathcal{P}) = \begin{cases} \frac{1}{|N_{t+1}(v_t^i)|} & (v^{i+1}, v_t^i) \in E, \phi(v^{i+1}) = t+1 \\ 0 & (v^{i+1}, v_t^i) \in E, \phi(v^{i+1}) \neq t+1 \\ 0 & (v^{i+1}, v_t^i) \notin E \end{cases}$$

- v^i와v^{i+1}가 이웃 관계가 아니거나, 의도한 type이 아닐 경우, transition probability가 0
- Type이 같은 이웃 node끼리 probability를 균등하게 분배

03 Metapath2vec Framework

Metapath2vec++

- Metapath2vec에서는 softmax 함수를 사용할 때 node type을 무시
- Metapath2vec++에서 softmax 함수는 context c_t의 node type에 따라 normalized 됨
- 따라서 skip-gram 모델의 output layer에서 각 type에 맞는 multinomial distribution을 정의

Softmax

$$p(c_t|v;\theta) = \frac{e^{X_{c_t} \cdot X_v}}{\sum_{u_t \in V_t} e^{X_{u_t} \cdot X_v}}$$

Objective Function

$$O(\mathrm{X}) = \log \sigma(X_{c_t} \cdot X_v) + \sum_{m=1}^{M} \mathbb{E}_{u_t^m \sim P_t(u_t)} [\log \sigma(-X_{u_t^m} \cdot X_v)]$$

03 Metapath2vec Framework

Metapath2vec++

- MetaPathRandomWalk
  - 주어진 walk length와 meta-path 등을 고려하여 random walk 진행

- HeterogeneousSkipGram
  - MetaPathRandomWalk 함수에서 생성된 path를 활용
  - Neighborhood size 등을 고려하여 gradient descent 를 통해 embedding을 update

Input: The heterogeneous information network $G = (V, E, T)$, a meta-path scheme $\mathcal{P}$, #walks per node w, walk length l, embedding dimension d, neighborhood size k

Output: The latent node embeddings $\mathrm{X} \in \mathbb{R}^{|V| \times d}$

initialize X ;
for $i = 1 \rightarrow w$ **do**
  for $v \in V$ **do**
    MP = MetaPathRandomWalk($G, \mathcal{P}, v, l$) ;
    X = HeterogeneousSkipGram(X, k, MP) ;
  end
end
return X ;

MetaPathRandomWalk($G, \mathcal{P}, v, l$)
$MP[1] = v$;
for $i = 1 \rightarrow l-1$ **do**
  draw u according to Eq. 3 ;
  MP[i+1] = u ;
end
return MP ;

HeterogeneousSkipGram(X, k, MP)
for $i = 1 \rightarrow l$ **do**
  $v = MP$[i] ;
  for $j = max(0, i-k) \rightarrow min(i+k, l)$ & $j \neq i$ **do**
    $c_t = MP$[j] ;
    $X^{new} = X^{old} - \eta \cdot \frac{\partial O(\mathrm{X})}{\partial X}$ (Eq. 7) ;
  end
end

ALGORITHM 1: The *metapath2vec++* Algorithm.

04 Experiment

Data & Setup

Data

- Aminer Computer Science (CS) dataset
- Database and Information Systems (DBIS) dataset
- 위의 두 데이터 셋 모두 author, paper, venue 등의 관계를 탐색

Setup

(1) The number of walks per node w : 1000

(2) The walk length l : 100

(3) The vector dimension d : 128

(4) The neighborhood size k : 7

(5) The size of negative samples : 5

(6) Metapath : "APA", "APVPA"

- APA : coauthor semantic
- APVPA : heterogeneous semantic of authors publishing papers at the same venues

⟷ Compare with :

(1) DeepWalk / node2vec
(2) LINE
(3) PTE
(4) Spectral Clustering / Graph Factorization

04 Experiment

Multi-Class Classification

- 8개의 카테고리로 분류
- Author 별로 자신의 논문이 가장 많이 속한 카테고리로 labeling
- 각 method를 사용하여 얻은 embedding을 logistic classification의 input으로 활용

Result

- 대체적으로 metapath2vec 계열의 성능이 우수
- 특히 적은 데이터를 사용한 구간(20% 이하)에서의 성능이 돋보임

Table 2: Multi-class venue node classification results in AMiner data.

Metric	Method	5%	10%	20%	30%	40%	50%	60%	70%	80%	90%
Macro-F1	DeepWalk/node2vec	0.0723	0.1396	0.1905	0.2795	0.3427	0.3911	0.4424	0.4774	0.4955	0.4457
	LINE (1st+2nd)	0.2245	0.4629	0.7011	0.8473	0.8953	0.9203	0.9308	0.9466	0.9410	0.9466
	PTE	0.1702	0.3388	0.6535	0.8304	0.8936	0.9210	0.9352	0.9505	0.9525	0.9489
	metapath2vec	0.3033	0.5247	0.8033	0.8971	0.9406	0.9532	0.9529	0.9701	0.9683	0.9670
	metapath2vec++	0.3090	0.5444	0.8049	0.8995	0.9468	0.9580	0.9561	0.9675	0.9533	0.9503
Micro-F1	DeepWalk/node2vec	0.1701	0.2142	0.2486	0.3266	0.3788	0.4090	0.4630	0.4975	0.5259	0.5286
	LINE (1st+2nd)	0.3000	0.5167	0.7159	0.8457	0.8950	0.9209	0.9333	0.9500	0.9556	0.9571
	PTE	0.2512	0.4267	0.6879	0.8372	0.8950	0.9239	0.9352	0.9550	0.9667	0.9571
	metapath2vec	0.4173	0.5975	0.8327	0.9011	0.9400	0.9522	0.9537	0.9725	0.9815	0.9857
	metapath2vec++	0.4331	0.6192	0.8336	0.9032	0.9463	0.9582	0.9574	0.9700	0.9741	0.9786

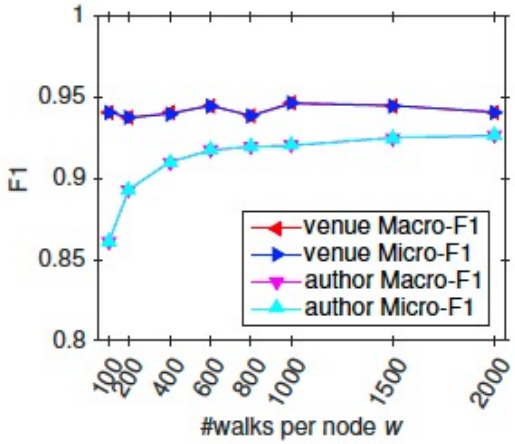
Table 3: Multi-class author node classification results in AMiner data.

Metric	Method	5%	10%	20%	30%	40%	50%	60%	70%	80%	90%
Macro-F1	DeepWalk/node2vec	0.7153	0.7222	0.7256	0.7270	0.7273	0.7274	0.7273	0.7271	0.7275	0.7275
	LINE (1st+2nd)	0.8849	0.8886	0.8911	0.8921	0.8926	0.8929	0.8934	0.8936	0.8938	0.8934
	PTE	0.8898	0.8940	0.897	0.8982	0.8987	0.8990	0.8997	0.8999	0.9002	0.9005
	metapath2vec	0.9216	0.9262	0.9292	0.9303	0.9309	0.9314	0.9315	0.9316	0.9319	0.9320
	metapath2vec++	0.9107	0.9156	0.9186	0.9199	0.9204	0.9207	0.9207	0.9208	0.9211	0.9212
Micro-F1	DeepWalk/node2vec	0.7312	0.7372	0.7402	0.7414	0.7418	0.7420	0.7419	0.7420	0.7425	0.7425
	LINE (1st+2nd)	0.8936	0.8969	0.8993	0.9002	0.9007	0.9010	0.9015	0.9016	0.9018	0.9017
	PTE	0.8986	0.9023	0.9051	0.9061	0.9066	0.9068	0.9075	0.9077	0.9079	0.9082
	metapath2vec	0.9279	0.9319	0.9346	0.9356	0.9361	0.9365	0.9365	0.9365	0.9367	0.9369
	metapath2vec++	0.9173	0.9217	0.9243	0.9254	0.9259	0.9261	0.9261	0.9262	0.9264	0.9266

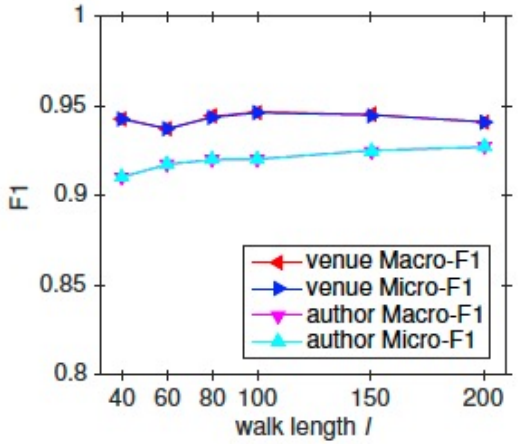
04 Experiment

Multi-Class Classification

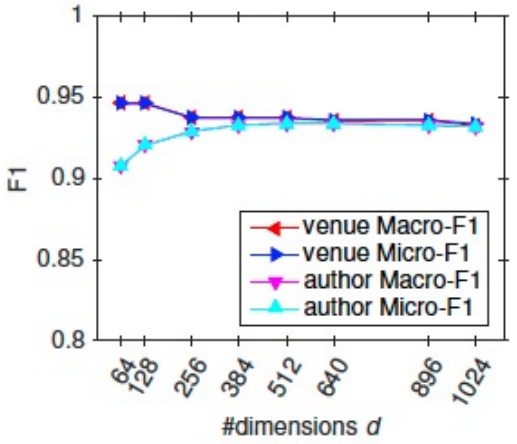
Parameter Sensitivity



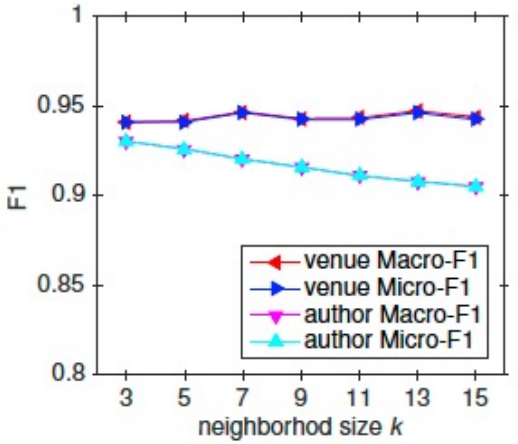
(a) #walks per node w



(b) walk length l



(c) #dimensions d



(d) neighborhood size k

- Venue의 경우 4가지 실험에서 모두 크게 영향을 받지 않음
- Author의 경우 앞선 세가지 case에서는 값이 늘어날수록 좋은 performance를 보였고,
- Neighborhood size에서는 값이 커질수록 오히려 negative 영향을 받음 (homogeneous에서와 차이)

04 Experiment

Node Clustering

- 8개의 카테고리로 분류
- Author 별로 자신의 논문이 가장 많이 속한 카테고리로 labeling
- 각 method를 사용하여 얻은 embedding을 k-means algorithm의 input으로 활용
- NMI(normalized mutual information)을 활용하여 결과 평가

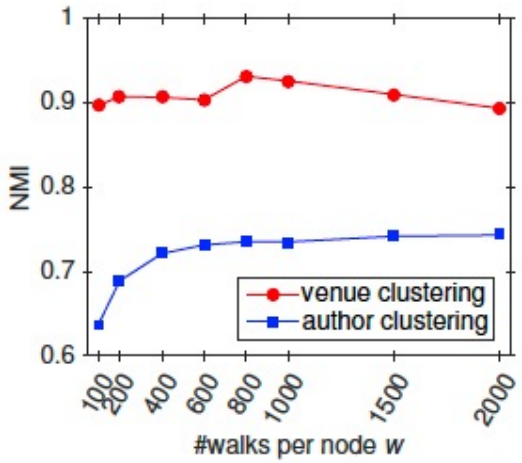
Table 4: Node clustering results (NMI) in AMiner data.

methods	venue	author
DeepWalk/node2vec	0.1952	0.2941
LINE (1st+2nd)	0.8967	0.6423
PTE	0.9060	0.6483
metapath2vec	0.9274	0.7470
metapath2vec++	0.9261	0.7354

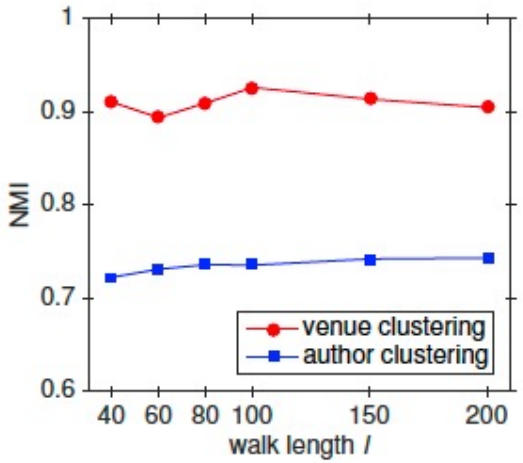
04 Experiment

Node Clustering

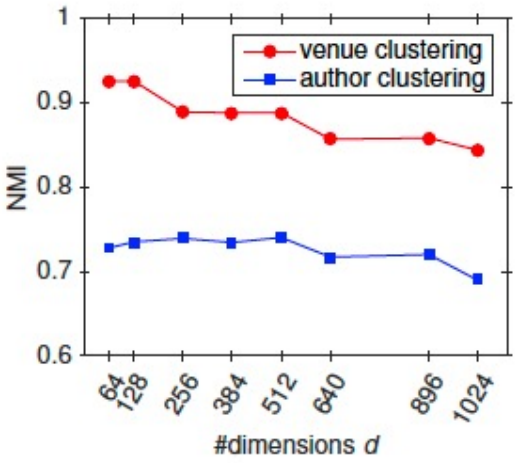
Parameter Sensitivity



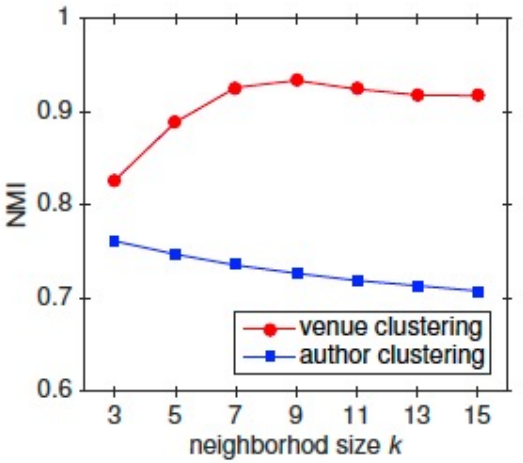
(a) #walks per node w



(b) walk length l



(c) #dimensions d



(d) neighborhood size k

d = 128, k = 7 정도면 performance와 computational cost 간의 적절한 balancing

04 Experiment

Similarity Search

- 16 top CS conferences & 5 from DBIS data
- 노드 간의 distance를 계산하기 위해 cosine similarity 사용

Table 6: Case study of similarity search in DBIS Data

Rank	KDD	SIGMOD	SIGIR	WWW	WSDM
0	KDD	SIGMOD	SIGIR	WWW	WSDM
1	SDM	PVLDB	TREC	CIKM	WWW
2	ICDM	ICDE	CIKM	SIGIR	SIGIR
3	DMKD	TODS	IPM	KDD	KDD
4	KDD E	VLDBJ	IRJ	ICDE	AIRWeb
5	PKDD	PODS	ECIR	TKDE	CIKM
6	PAKDD	EDBT	TOIS	VLDB	WebDB
7	TKDE	CIDR	WWW	TOTT	ICDM
8	CIKM	TKDE	JASIST	SIGMOD	VLDB
9	ICDE	ICDT	JASIS	WebDB	VLDBJ
10	TKDD	DE Bull	SIGIR F	WISE	SDM

Table 5: Case study of similarity search in AMiner Data

Rank	ACL	NIPS	IJCAI	CVPR	FOCS	SOSP	ISCA	S&P	ICSE	SIGGRAPH	SIGCOMM	CHI	KDD	SIGMOD	SIGIR	WWW
0	ACL	NIPS	IJCAI	CVPR	FOCS	SOSP	ISCA	S&P	ICSE	SIGGRAPH	SIGCOMM	CHI	KDD	SIGMOD	SIGIR	WWW
1	EMNLP	ICML	AAAI	ECCV	STOC	TOCS	HPCA	CCS	TOSEM	TOG	CCR	CSCW	SDM	PVLDB	ECIR	WSDM
2	NAACL	AISTATS	AI	ICCV	SICOMP	OSDI	MICRO	NDSS	FSE	SI3D	HotNets	TOCHI	TKDD	ICDE	CIKM	CIKM
3	CL	JMLR	JAIR	IJCV	SODA	HotOS	ASPLOS	USENIX S	ASE	RT	NSDI	UIST	ICDM	DE Bull	IR J	TWEB
4	CoNLL	NC	ECAI	ACCV	A-R	SIGOPS E	PACT	ACSAC	ISSTA	CGF	CoNEXT	DIS	DMKD	VLDBJ	TREC	ICWSM
5	COLING	MLJ	KR	CVIU	TALG	ATC	ICS	JCS	E SE	NPAR	IMC	HCI	KDD E	EDBT	SIGIR F	HT
6	IJCNLP	COLT	AI Mag	BMVC	ICALP	NSDI	HiPEAC	ESORICS	MSR	Vis	TON	MobileHCI	WSDM	TODS	ICTIR	SIGIR
7	NLE	UAI	ICAPS	ICPR	ECCC	OSR	PPOPP	TISS	ESEM	JGT	INFOCOM	INTERACT	CIKM	CIDR	WSDM	KDD
8	ANLP	KDD	CI	EMMCVPR	TOC	ASPLOS	ICCD	ASIACCS	A SE	VisComp	PAM	GROUP	PKDD	SIGMOD R	TOIS	TIT
9	LREC	CVPR	AIPS	T on IP	JAIG	EuroSys	CGO	RAID	ICPC	GI	MobiCom	NordiCHI	ICML	WebDB	IPM	WISE
10	EACL	ECML	UAI	WACV	ITCS	SIGCOMM	ISLPED	CSFW	WICSA	CG	IPTPS	UbiComp	PAKDD	PODS	AIRS	WebSci

04 Experiment

Visualization

TensorFlow embedding projector을 사용하여 visualize

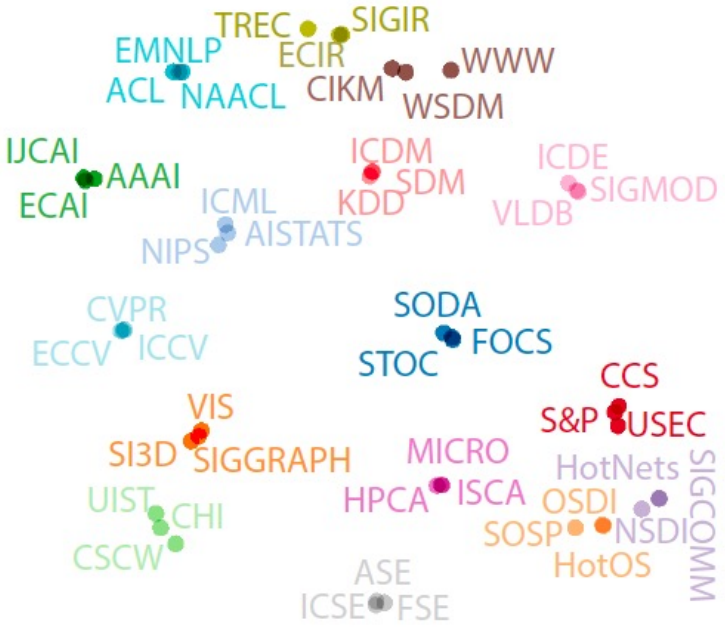
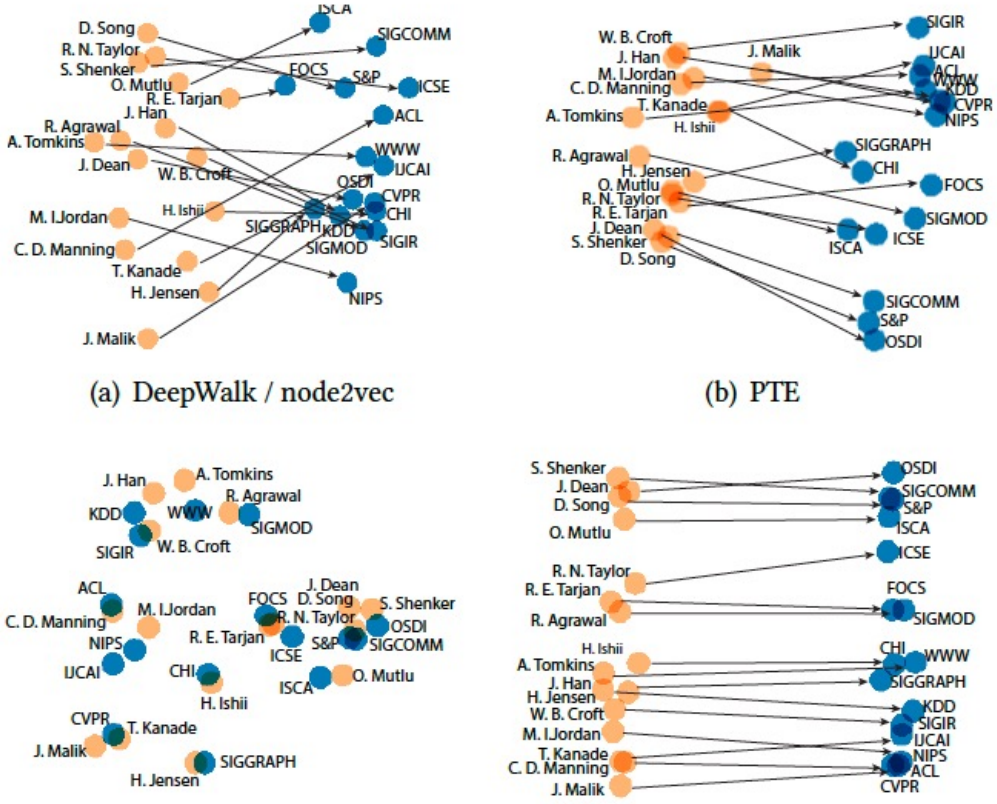


Figure 5: 2D t-SNE projections of the 128D embeddings of 48 CS venues, three each from 16 sub-fields.



04 Experiment

Scalability

- Aminer CS data로 스레드의 개수를 늘려가면서 실험을 진행
- Metapath2vec, metapath2vec++ 두 가지 경우에서 모두 optimal line과 크게 멀지 않은 reasonable한 그래프 관측 가능
- 두 모델 모두 large-scale heterogeneous networks에 대해서도 efficient하고 scalable함

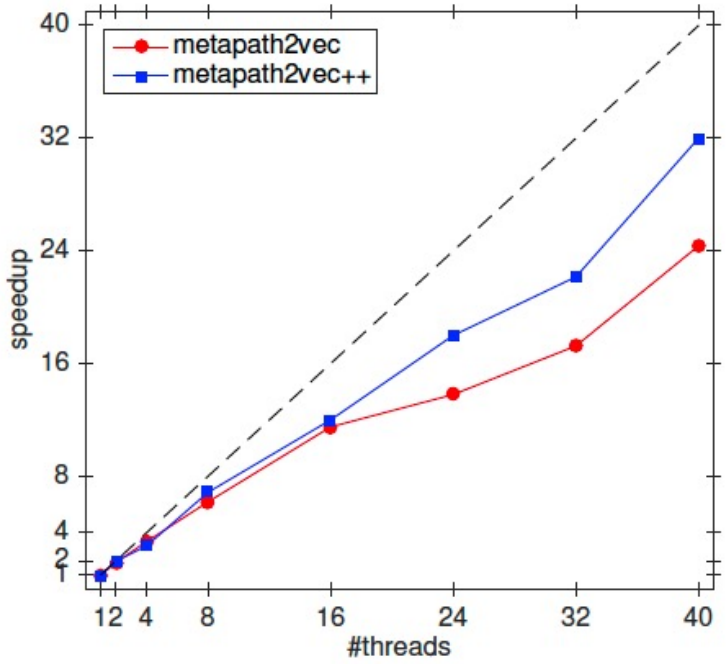


Figure 6: Scalability of *metapath2vec* and *metapath2vec++*.

Conclusion

- Meta-path-guided random walk strategy
  - ➢ Capable of capturing both the structural and semantic correlations of differently typed nodes and relations
- Metapath2vec and metapath2vec++ are able to improve various heterogeneous network mining tasks
  - ➢ Similarity search, node classification, and clustering
- Can be naturally applied to real-world applications
  - ➢ Academic networks, such as author, venue, and paper

Limitations

- Large intermediate output data가 존재할 때 학습하는 것이 어려움
- 도메인에 대한 이해를 바탕으로 meta-path를 사전에 설정해야함

Thank You