Mining relationships across academic networks

Xiao Zeng
Shanghai Jiao Tong University
Email: zengxiao1997@gmail.com

Yinan He
Shanghai Jiao Tong University
Email: qawb@foxmail.com

***Abstract*—Mining relationships across academic networks is a significant problem, which can improve many higher-level tasks with supervised or unsupervised machine learning approaches such as recommendation system and topic prediction. However, present relationships discovering approaches are not expressive enough to capture the topology as well as diversity of similarity patterns observed in academic networks.**

Here we propose two novel models to solve the relationships mining problem, each taking advantage of different information in the academic network. One aims at examining graph topology, the other focus on inner text information of the nodes.

We evaluate our model by testing the accuracy in the dblp network and visualizing the clustering. We aim to combine the two methods into an integrated and more expressive model.

I. Introduction

With the rapid growing of academic networks, various strategies are used to mine relationships across academic network, among with graph structure and node information are two main parts.

With regard to graph structure, path-based strategies are used by several methods. With the development of natural language processing, social network relationships can be treated as a kind of social language to be applied in the notably group of NLP models like word2vec. Therefore, a number of models have been proposed based on the word2vec method. However, these work has merely focused on the homogeneous network embedding without concentrating of non-singular type of nodes in the network, which is common in social as well as academic network. By handling these challenges, we propose the path2vec framework that incorporates certain kinds of rules in the sequence generation strategy and set up threshold for adaptive clustering.

Citation networks are a traditional social medium for the exchange of ideas and knowledge among researchers. It is important in studying relationship between papers. Methods have been proposed to mining paper relationship in citation networks mainly focusing on direct citation link, however this may cause the loss of information. In this paper, we propose a citation-based method, combining structure of citation networks and paper content to evaluate relevance between two papers.

In our work, we provide two methods based on graph structure and node information to solve the problem. The key novelty of our work is in generating a more flexible and rational clustering of nodes that share close relationships. By choosing rational and appropriate clustering, the path2vec framework can learn good relationships based on the network structure. We achieve this by adding walking rules as well as threshold, which leads to a efficient exploration and better performance in specific tasks. For citation graph, we judge the relationship between papers not only from paths between them, but also using paper context, so the evaluation can be more accurate. The contributions of this paper are as follows:

- We propose a novel method *path2vec* to generate node sequence constrained within specific rational rule, which can lead to more rational embedding.
- We use adaptive clustering we can get the most reasonable clustering that capitulates the specific degree of relevance of node relationship by using threshold of the embedding distance.
- We classify the citation type and assign weight to each edge in citation network, in order to better mimic the real situation.
- We develop a method by using graph distance and weight of links to measure the relevance between two papers in a citation graph.

The rest of the papers is structured as follows. In Section 2, we give a brief survey about related works in feature learning for networks. We present the technical details of our approach in Section 3. In Section 4, we evaluate our model on constructed datasets. Finally we conclude our work and highlight promising directions for future work in Section 5.

II. Related Work

A. Node embedding

Recent years have seen great improvement in dealing with sparsity network analysis problem with regard to node embedding. One recent model DeepWalk has brought language model into representation of social network relationship, by discovering similarity between the power-law distribution of vertices appearing in short random walks and the distribution of words in natural language. Based on the concept of transferring word to vector, node2vec[3] model has explored the relationship between nodes in network, presenting a more rational way combining classical search strategies DFS and BFS to obtain word sequence. Inspired by the DeepWalk and node2vec model, we establish a ranking model by simulating the network nodes as web pages. The existing works and drawbacks can be summarized as following parts:

- **Unsupervised feature learning based on DeepWalk model.** The DeepWalk model uses local information obtained from truncated random walks to learn latent

representations by treating walks as the equivalent of sentences. The node sequence generated by DeepWalk model is by simple random walk. However, random walk is too naive to contain the complicated relationship in a graph, which leads to the bottleneck of the algorithm. Based on random work, the breakthrough of word2vec is considering a more rational walking strategy, giving random walk a bias to balance BFS and DFS. By adding distance factor of node influence, the BiasWalk model presents the idea that the influence of node dissipates through walking along edges. But The obvious limitation with these two models has something to do with neighborhood information exploration, since they both neglect the network structure. As for DeepWalk model, it gives us no control over the explored neighborhoods, while BiasWalk tries to overcome the weakness by considering influence of nearby nodes. The proposed algorithm in BiasWalk indeed improves the random walk model, but it is not so comprehensive in whole network topology. At the same time, the two previous models are not content-relevant, which means that the node is a simple meaningless point when constructing sequence. All these simplifications add to the drawbacks of the model, which reduces its precision in node classification.

- **Semi-Supervised Classification of Network Data.** The semi-supervised learning framework[6][18] based on graph embeddings[12] builds its model on prediction of class label and neighborhood context. It develops both transductive and inductive variants of presented method. In the transductive variant of the method, the class labels are determined by both the learned embeddings and input feature vectors, while in the inductive variant, the embeddings are defined as a parametric function of the feature vectors, so predictions can be made on instances not seen during training. But the drawback is apparently related with complexity of training. Without transferring the graph into low dimensional word sequence, the semi-supervised classification training method can not deal with large and complex network.
- **Supervised Q-walk Network Representation.** The Q-walk model[1][9] dedicates to incorporate neighborhood information of network. It provides k-hops neighborhood based confidence values learner to learn confidence values of labels for nodes regardless of node embedding. These confidence values then aid in learning an apt reward function for Q-learning. However, the supervised Q-walk model meets with difficulty in non-homophily graph, which means it only considers the situation that instances belonging to the same class tend to link to each other or have higher edge weight between them. Due to this assumption, the drawback is clearly related to graph application. Network in reality is comprised of unpredictable components, where Q-walk may simply regard it as no structural equivalence graph that results in node classification error.

Path-based approaches[4][13] to mine relationships in academic network are also improving during the years, such as the pathSim method[15] and metapath2vec[2]. The method of metapath2vec is based on the deepwalk and node2vec. It formalizes the heterogeneous network representation learning problem, where the objective is to simultaneously learn the low-dimensional and latent embeddings for multiple types of nodes. It is to maximize the likelihood of preserving both the structures and semantics of a given heterogeneous network by proposing meta-path based random walks in heterogeneous networks to generate heterogeneous neighborhoods with network semantics for various types of nodes.

The main difference between our proposed model and existing work is that our model extends special rules of path pattern to be added in to walking strategy. Moreover, we design an adaptive method to get the right number of clustering by defining distance threshold between various kinds of nodes to get more rational clustering as well as node relationship.

B. Citation networks

- **Co-citation analysis** Co-citation analysis[14] suggested that the more two papers are related to each other, the more often they are co-cited. It is one of the first applications of co-occurrence, simple and effective. However, it only consider the direct citation between papers, making it hard to properly analyze relationship between two newly published paper, which have received few reference.
- **Katz graph distance** Katz graph distance is used to measure the relevance between nodes in a social networks. It takes the whole structure between two nodes into consideration. However, it regards all edges as equally important, which is unsuitable for the case of citation graph, as not all citation are euqally important.

The main difference between our method and existing ones is that ours combines graph structure and paper context. By assigning weight to citation links, our method can better reflect the real relationship between two papers.

III. Mining relationship strategies

A. The path2vec Framework

One strategy based on graph structure to mine relationship between two nodes is the path2vec, which is a novel framework that is able to take advantage of effective node representations to represent various kinds of relationships in heterogeneous networks. Especially, the type of nodes aimed to examine the relationship in the heterogeneous networks can refer to papers, authors and references. The final objective is to use the embedding clustering method to classify relationship into several reasonable kinds of relationships.

1) Homogeneous path2vec: The homogeneous path2vec is based on the node2vec model which uses word2vec to do the feature representation in graph. Feature learning methods based on Skip-gram architecture have been originally developed in the context of natural language[8], where DeepWalk[10] incorporates the word2vec embedding method

with graph structure pecularity by generating random walk sequence which takes nodes as words and node sequences as sentences. The core method is to represent the structure of graph by a series of node sequences and use word2vec to do dimension reduction. The objective function is to maximize the log-b a network $N_S(u)$ for a node u conditioned on its feature representation, given by mapping function from nodes to feature representations f.

$$\max_f \sum_{u \in V} logPr(N_s(u)|f(u)) \tag{1}$$

where $N_s(u) \subset V$ as a *network neighborhood* of node u generated through a neighborhood sampling strategy S, f is a matrix of size $|V| \times \lambda$ parameters.

2) Heterogeneous path2vec: Since the relationships across academic networks is different from original social network because of the specific features of the papers. Since in the academic field, the reference of papers are more likely to be a paper in the same previous venues, and the work of papers are share more relatedness when they come from same authors or collaborated authors, making the relationships capitulated by authors and venues can not be neglected. As the relationships between papers are not simply explained by the reference edges but rather interconnected by other important elements like authors and venues, modelling this kind of relationship in heterogeneous network seems more rational.

The heterogeneous path2vec methods incorporates the network structure into skip-gram by generating random walks in the network graph, then the output embedding represents the corresponding node in the graph, more specifically, an author or an venue. With the embedding we can use unsupervised clustering method to divide the nodes into several rational clusters with a threshold distance that is adaptive to the various datasets.

The model structure is shown here, it shows the whole process of clustering can be divided into three parts: generating heterogeneous node sequence, embedding and adaptive clustering.

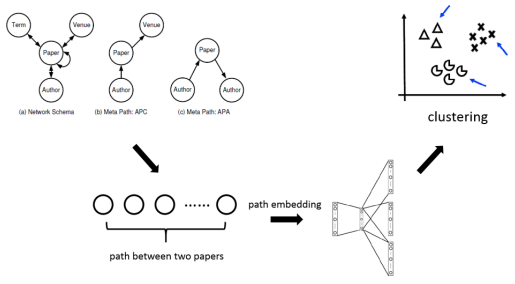

Figure 1. Architecture of our model

- **Heterogeneous node sequence generation.** To generate heterogeneous node sequence in academic graph requires other rules besides the balance of dfs and bfs strategy in node2vec. The previous way defines the transition probability to $\pi_{vx} = \alpha_{pq}(t,x) \cdot w_{vx}$, where

$$\alpha_{pq}(t,x) = \begin{cases} \frac{1}{p} & \text{if } d_{tx} = 0 \\ 1 & \text{if } d_{tx} = 1 \\ \frac{1}{q} & \text{if } d_{tx} = 2 \end{cases} \tag{2}$$

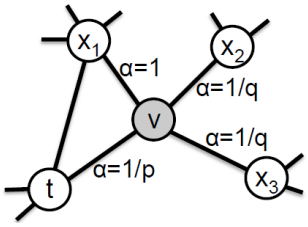

Figure 2. Illustration of bias walk balancing BFS and DFS

However, in the generation of heterogeneous academic network, we take the type of node into consideration, which means the rules become more strict as a result of the constraint that the sequence can only be the combination of the following pattern.

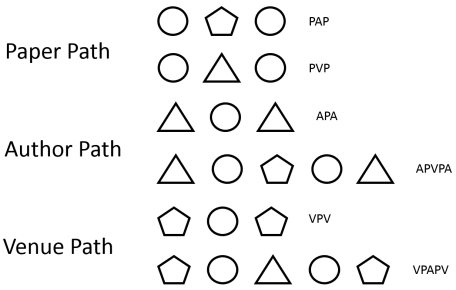

Figure 3. Path pattern

As the figure shown here now, suppose given a graph G(V,E,T), then the the paper path pattern can only be the the form denoted by

$$P_1 \xrightarrow{C_1} P_2 \xrightarrow{C_2} \quad ... \quad P_t \xrightarrow{C_t} P_{t+1} \quad ... \quad P_n$$

where C_i defines the set that contains the corresponding nodes that have the different node types from corresponding set P_i within the neighborhood of P_i. For example, for paper path, set C contains nodes that represents author or venue.

Similarly, the author path and the venue path have their own special pattern as denoted by

$$A_1 \xrightarrow{P_1} S_2 \xrightarrow{P_2} \quad ... \quad S_t \xrightarrow{P_t} S_{t+1} \quad ... \quad A_n$$

$$V_1 \xrightarrow{P_1} S_2 \xrightarrow{P_2} \quad ... \quad S_t \xrightarrow{P_t} S_{t+1} \quad ... \quad V_n$$

where S_i defines the set that contains the corresponding nodes that have author type or venue type within the neighborhood of the previous paper node.

The transition probability is denoted as

$$P(v^{i+1}|v_t^i, \mathcal{P}) = \begin{cases} \frac{1}{|N_{t+1}(v_t^i)|} & (v^{t+1}, v_t^i) \in E, \phi(v^{i+1} = t+1) \\ 0 & (v^{t+1}, v_t^i) \in E, \phi(v^{i+1} \neq t+1) \\ 0 & (v^{t+1}, v_t^i) \notin E \end{cases} \tag{3}$$

where $v_t^i \in V_t$ and $N_{t+1}(v_t^i)$ denote the V_{t+1} type neighborhood of node v_t^i.

The heterogeneous node sequence takes the semantic relationships between different kinds of nodes into account which makes the sequence more rational and reliably, which can also be used in the skip-gram model of word2vec to reduce dimension. By combining the original walk strategy with a certain kind of rule, the sequence becomes more meaningful, thus can lead to better embedding.

- **Network Embedding.** Network embedding uses the language model skipgram. SkipGram is a model that maximizes the cooccurrence probability among the words that appear within a window, w, in a sentence. For each sequence, we map each vertex v_j to its current representation vector $\phi(v_j) \in R^d$. Given the representation of v_j , we would like to maximize the probability of its neighbors in the walk. We can learn such posterior distribution using several choices of classifiers. For example, modeling the previous problem using logistic regression would result in a huge number of labels that is equal to $|V|$, which could be in millions or billions. Such models require large amount of computational resources that could span a whole cluster of computers. To speed the training time, Hierarchical Softmax can be used to approximate the probability distribution.
- **Adaptive clustering.** The adaptive clustering strategy is used for classifying the nodes into a rational number of categories. Since for various datasets, we don not know the number of clustering that can efficiently and effectively cluster the papers or other type of nodes, the main point is to adjust the clustering number by setting a threshold δ according to the maximum distance of the nodes in the embedding, then after iteration of k we can get the least k that satisfy the threshold, which is the optimal k. Normally, δ is set by

$$\delta = \beta \max_{v_i, v_j \in V} d(v_i, v_j)$$

where β refers to the degree of relevance of nodes that we want to get in the clustering. For example, as for finding importance relevance, β is normally set to 0.1.

The pseudocode for Heterogeneous path2vec is given in Algorithm 1, it shows the core framework of the code.

The outer loop specifies the number of times, which we set for the number of walks generated from each vertex. We think of each iteration as making a 'pass' over the data and sample one walk per node during this pass. In the inner loop, we iterate over all the vertices of the graph. For each vertex v we generate a type-based walk and then use it to update our representations.

Algorithm 1 Heterogeneous path2vec Algorithm

Input: Graph $G(V, E, W)$,Dimensions d, Walks per node r, Walk length l, Context size k, Return p, In-out q
Output: f

$G' = (V, E, \pi)$
Initialize $walks$ to Empty
for $iter = 1; iter \leq r; iter = iter + 1$ **do**
  forall *nodes* $u \in V$ **do**
    $walk$ = HeterogeneousWalk(G', u, l)
    Append $walk$ to $walks$
  end
end
f = StochasticGradientDescent$(k, d, walks)$
return f

function HETEROGENEOUSWALK($G'(V, E, W)$,u,l)
Initialize $walk$ to [u]
for $walk_{iter} = 1$ *to* l **do**
  $curr = walk[-1]$
  V_{curr} =GetNeighbors$(curr, G')$
  s = TypeSample$(V_{curr,\pi})$
  Append s to $walk$
end
return $walk$

B. The Citation-based Framework

As graph structure helps a lot to reveal the relationship between nodes, we proposed a link-based method to measure the relevance between academic papers in weighted citation networks.

A paper has dozens of citation, however, not all of the citations are equally important. A survey has done among authors, asking them to list the essential references in their paper[19]. The result shows that only 10.3% references (322 among 3143) are considered important by the author himself. To fit such reality better, we assign weight to edge in the citation networks, inferring the citation type between cite and cited paper.

A citation network can be represented by a direct graph $G = (V, E)$, where V is a set of vertices, and E is a set of weighted edges. For each $v_i \in V$, v_i denotes a paper p_i. For each $e_{ij} = (v_i, v_j, \omega_{ij}) \in E$, e denotes a direct citation link between two papers p_i and p_j, i.e. p_i cites p_j directly. And the weight ω_{ij} reflect the citation type. Our method consists of two main parts: generate weight and measure relevance.

1) Generating Weight: The weight assigned to edge is generated according to the type of citation link.

$$\omega_{ij} = \begin{cases} \omega_0, \text{ paper j is an important citation of paper i} \\ \omega_1, \text{ paper j is an unimportant citation of paper i} \end{cases}$$

The process to classify citation link is shown in figure 5:

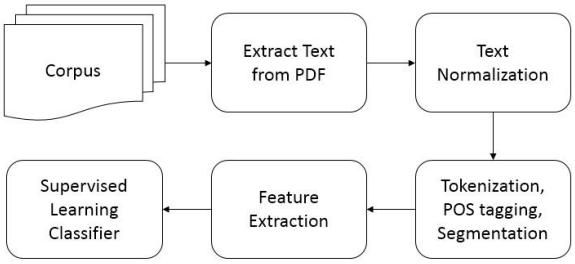

Figure 5. Classify Model Flowchart

Given a pair of citing and cited paper, we extract 12 features from them, and uses machine learning methods to classify their relationship into two types: important and unimportant. The 12 features are listed as follows:

1) **Number of direct citation** This feature computes how many times the citation is directly cited in the citing paper.
2) **Number of direct citation/Number of all direct citation** This feature divide the number of direct citation by the number of total citation, inferring the value of each citation.
3) **Number of direct citation per section** This feature computes how many times the citation is directly cited in each section of the citing paper. We use ParsCit to divide the whole paper into 6 section: introduction, literature review, method, experiment, discussion and conclusion, and count the direct citation respectively.
4) **Number of indirect citation** This feature computes how many times the citation is indirectly cited in the citing paper, i.e. the number of the alias of the citation appears. The alias is generate from 20 papers that cite the citation by extrating the noun phrase directly before the citation, or the noun phrase following the citation and a verb, collecting the unigrams and bigrams in these noun phrases, computing their tf-idf scores[7], and selecting those with tf-idf scores higher than 200.
5) **Number of indirect citation/Number of all indirect citation** Similar to the second feature but with indirect citation.
6) **Number of indirect citation per section** Similar to the third feature but with indirect citation.
7) **Cue words similarity for important class** This feature computes the cosine similarity between the cue words of important class and the text around the reference. Cue words are specific phrase usually appearing in the paper when referring to previous work, such as 'extend', 'was based on', 'although yet', 'except in' and so on. They hints the importance of a citation. We using the cue words extracting from 80 articles by Bornmann and Daniel[16], which are classified to indicating important citation and unimportant citation.
8) **Cue words similarity for unimportant class** This feature computes the cosine similarity between the cue words of unimportant class and the text around the reference.
9) **Abstract similarity** This feature computes the similarity between the cited and citing papers abstracts using the cosine similarity of the tf-idf scores. The closer the abstracts is, the more likely the paper extends the citation.
10) **Number of paper that cited the citation per year** This feature computes how many times the citation is cited by other paper. It is divided by the number of year that it has been published. This feature indicates the influence of the citation.
11) **Citation appears in table or caption** This is a boolean variable. It is set to true if at least a citation appears in a table or a caption of a figure or table. This is an indicator that the author of the citing work is comparing her results to the cited paper
12) **Author overlap** This is a boolean variable. It is set to true if the citing and the cited works share at least one common author. The intuition behind this feature is that shared authors indicate that the new work is likely to be an extension of the cited paper.

We use two datasets to train our model, one as mentioned above labels 3143 citation as important and unimportant, and the other one is a public dataset [17] that randomly labels 465 citations among 20527 papers. Using the conbined dataset, we train a model with the accuracy of 0.92. We use this model to classify citation type in the citation network and assign weight to each edge.

2) Measure Relevance: On the weighted citation network, we use Katz graph distance measure[5] to reveal the relationship between two papers. It measures the relevance between nodes considering not only the number of paths between x and y, but also the number of hops in each path. According to Katz, the relevance between to nodes x and y in a graph can be defined as follows:

$$R(x,y) = \Sigma_{l=1}^{\infty} \eta^l |\theta_{x,y}^{<l>}| \tag{4}$$

where $\eta \in [0,1]$ is a decay parameter, $|\theta_{x,y}^{<l>}|$ is the set of all l-hops paths from x to y. Let P denote the set of all path between x and y, the equation can be written as:

$$R(x,y) = \Sigma_{p_i \in P} \eta^{|p_i|} \tag{5}$$

The Katz method regard every edge as equally important, while we assign weight to each edge. After introducing the weight, the equation becomes:

$$R(x,y) = \Sigma_{p_i \in P} \eta^{|p_i|} \frac{\Sigma_{e_{jk} \in p_i} \omega_{jk}}{|p_i|} \tag{6}$$

Given a graph with n nodes, there may be n^2 paths between two nodes, which makes the complexity unaffordable. However, based on our experiment, short paths

contribute most weights while long paths can bring in noise. So we only use paths with hops less than 6 to measure the relevance.

IV. Experiment

A. Experiment Setup

1) Heterogeneous Networks Datasets: The heterogeneous networks' nodes which are constructed by papers, authors and venues come from DBLP. To be more specific, the Aminer CS dataset consists of 9323739 computer scientists and 3194405 papers from 3883 computer science venues(both conferences and journals). The DBIS dataset contains 72902 papers with top-5000 authors covered in 464 fields. Then the heterogeneous network with node type paper, author, venue is constructed from the dataset. The link in the network represents the direct relationships between these types of nodes. For example, the edge between paper and author means that the author is exactly one of the authors of the paper while if an edge between a paper and a venue exists, it means that the paper is accepted by this venue.

Table I
Data size of Academic Network datasets

Dataset	DBLP	DBIS	Aminer
Number of Papers	1.2M	72902	9323739
Number of Authors	710K	5000	3194405
Number of Venues	5K	464	3883

2) Citation Network Dataset: To train the weight assigning model, we combined two labeled dataset from Valenzuela et al. [17] and Zhu et al [19]:

Table II
Statistics of Annotated Data

Label	Number of Citation
Important	389
Unimportant	673

To evaluate our citation-based method, we used a dataset[11] collected from ACL anthology.

Table III
AAN Dataset

Paperl	Citation	Author	Venue
21290	12382	14981	341

B. Path2vec Framework

1) Embedding Visualization: For the path2vec model, we specify the meta-path scheme with the stated pattern in the model section. In addition, due to the large number of authors and papers, the number of walks from these nodes and relatively small. The embedding vector size is 128 dimension with walk length set to 100.

We use t-SNE method to perform embedding visualization, reducing dimension from 128D to 2D/3D. The embedding visualization is shown here, with the order of venue, author and paper.

2) Clustering Evaluation: The we perform adaptive clustering to the 128D dimensional embedding to adjust to a appropriate k for rational clustering. We evaluate the ultimate optimal clustering using CP and DBI.

- CP: Compactness

$$\overline{CP_i} = \frac{1}{|\Omega_i|} \sum_{x_i \in \Omega_i} ||x_i - w_i||$$

$$\overline{CP} = \frac{1}{K} \sum_{k=1}^{K} \overline{CP_k}$$

- DB/DBI: Davies-Bouldin Index

$$DB = \frac{1}{k} \sum_{i=1}^{k} \max_{i \neq i} \left(\frac{\overline{C_i} + \overline{C_j}}{||w_i - w_j||_2} \right)$$

The compactness is to value the inner distance of clustering, smaller CP means smaller inner distance and therefore better clustering. As DB calculates the ratio of inner average distance to distance between clustering centers, smaller DB also means smaller ratio of inner distance to between-class distance which leads to better clustering.

The result of CP and DBI with regard to different thresholds are shown in the picture. Compared to the original metapath2vec model, our model's clustering result are more rational.

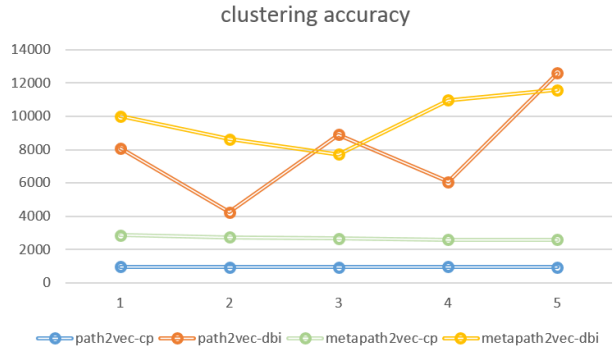

Figure 9. Embedding Clustering Evaluation

where the horizontal line refers to the different percent of threshold δ of the distance.

We also evaluate clustering by visualization. Since the embedding of thousands of hundreds nodes are not very clear to examine the relationships directly by visualization, we select a small portion of venue to show the close relationships in clustering. The embeddings represent around 40 venues in CS fields.

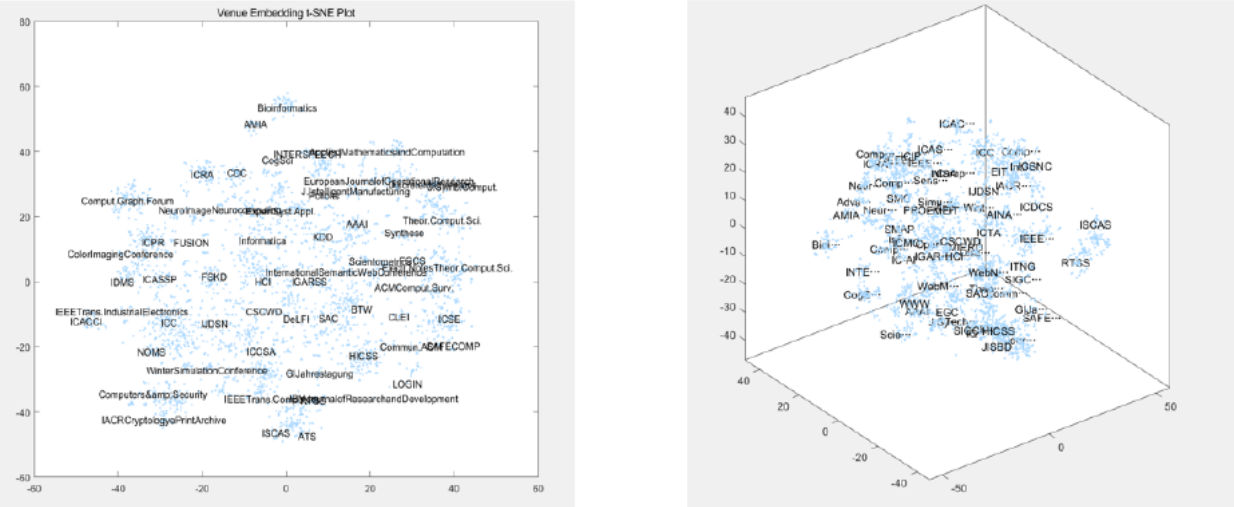

Figure 6. 2-D/3-D Venue Embedding Visualization

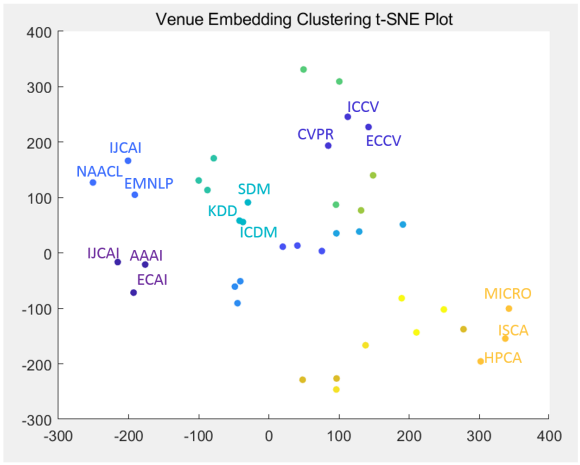

Figure 10. Venue Embedding Visualization

As we can see in the picture, the venues NAACL, IJCAI, EMNLP concerning NLP topis are clustered together, and the CVPR, ICCV, ECCV venues related to the topic of computer vision are also clustered together correctly in to a class.

3) Similarity Search: Then we also do some similarity search of the embeddings as case study, the distance here are measure by the cosine distance. The following tables are search cases of some venues and authors, we can find the close relevance of top rankers in the table, such as co-authors or authors in the same organizations and venues that share similar topics are clustered together, which shows the effectiveness of the algorithm.

For example, in the venues clustering, the related nodes of "ACL" is venues like EMNLP, NAACL, Computational Linguistics and so on. Similar performance can also be achieved in author clustering and paper clustering.

After getting the embedding, we also do some experiments to generate the relation sequence of the related nodes in the same clustering, which can show that they share similar characteristics.

C. Citation-based Method

1) Classify Citation: We have compared the performance of decision tree and SVM with RBF Kernel.

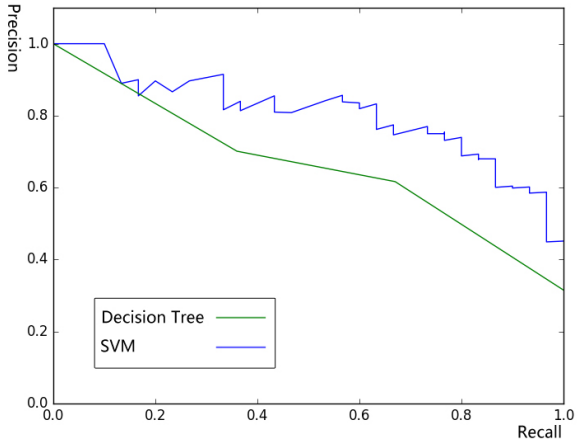

Figure 11. Percision Recall Curve

AUCPR of SVM is 0.87 while AUCPR of Decision Tree is 0.69, so we use SVM.

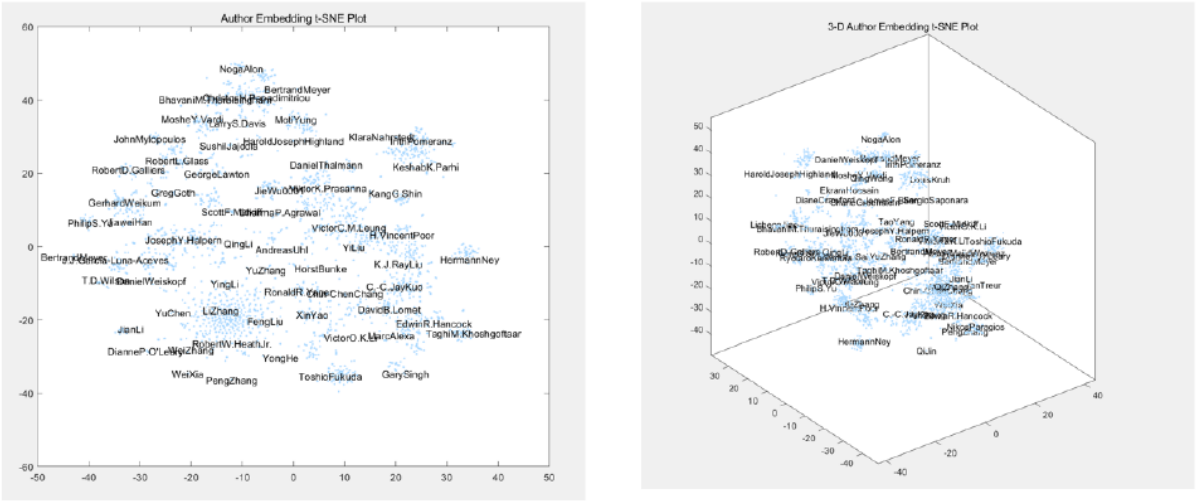

Figure 7. 2-D/3-D Author Embedding Visualization

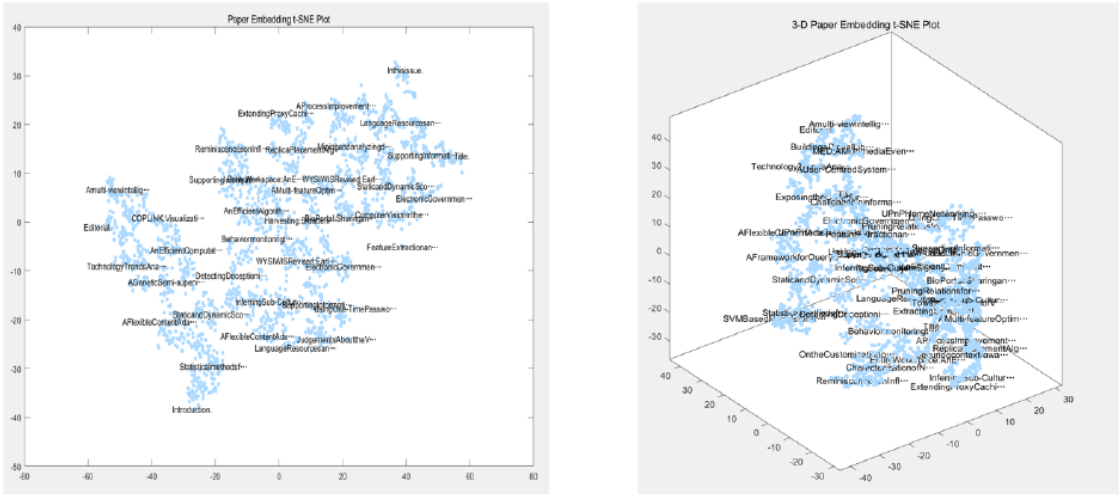

Figure 8. 2-D/3-D Paper Embedding Visualization

Table IV
CASE STUDY OF VENUE SIMILARITY SEARCH IN AMINER DATA

ACL		NIPS		INFOCOM	
ACL	1	NIPS	1	INFOCOM	1
EMNLP	0.966946	ICML	0.955095	IEEE/ACM TN	0.980632
CL	0.959138	AISTATS	0.945436	MobiHoc	0.939416
CoNLL	0.933703	NC	0.908183	MobiCom	0.91177
IJCNLP	0.922411	COLT	0.89621	SECON	0.905895
COLING-ACL	0.914321	UAI	0.873626	IWQoS	0.904628
NLE	0.913332	CVPR	0.842136	GLOBECOM	0.896472
LREC	0.902107	KDD	0.84182	WiOpt	0.896011
EACL	0.900098	ACML	0.832118	vCoNEXT	0.890572
ANLP	0.899777	ECCV	0.830614	SIGCOMM	0.888044
LREC	0.888303	AAAI	0.824888	vICC	0.885082

Table V
CASE STUDY OF AUTHOR SIMILARITY SEARCH IN AMINER DATA

LuoyiFu		JohnE.Hopcroft	
LuoyiFu	1	JohnE.Hopcroft	1
XinbingWang	0.887362	PrabhakarRaghavan	0.868889
WenyeWang	0.873246	AllanBorodin	0.842907
MichalisTitsias	0.864281	C.Seshadhri	0.829507
BenLiang	0.857967	AndrewChi-ChihYao	0.828785
KejieLu	0.857905	RudolphLanger	0.825836
aShiwenMao	0.856818	RobertEndreTarjan	0.825729
aShangqianHu	0.855743	RasmusPagh	0.82545
aKiTaekLee	0.852274	JurisHartmanis	0.821832
aMarcoFeletig	0.850694	JakubOcwieja	0.819203
aUlasC.Kozat	0.84784	VikrantSinghal	0.818596

We use SVM with RBF Kernel to train our model. We using 3-fold cross validation.

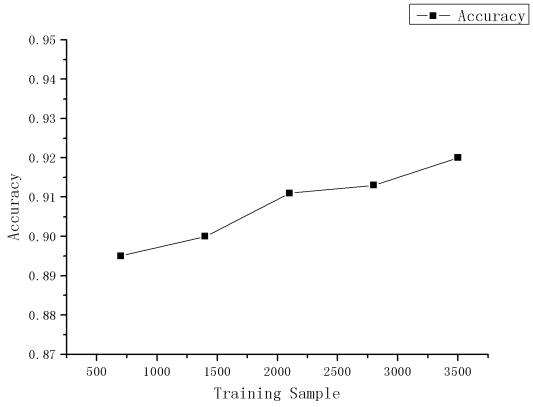

Figure 12. Learning curve of SVM

We also compare our model with the Valenzuela model [17].

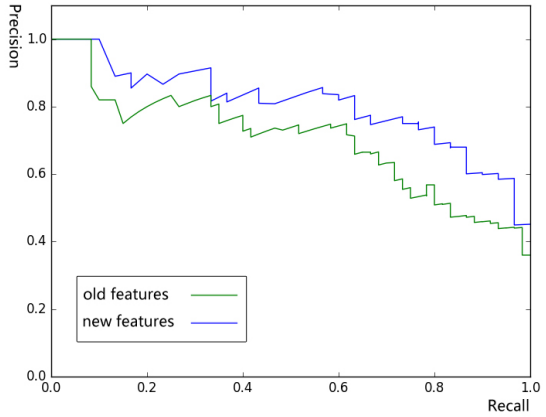

Figure 13. Percision Recall Curve

We can see that our model with new features outperforms the existing model.

2) Measure Relevance: We have done experiment to investigate the hops between related papers, e.g. paper with same author, paper of the same conference. We figured out that the hops between two papers with certain relationship are less than 6, and the hops between two papers with strong relationship are less than 3.

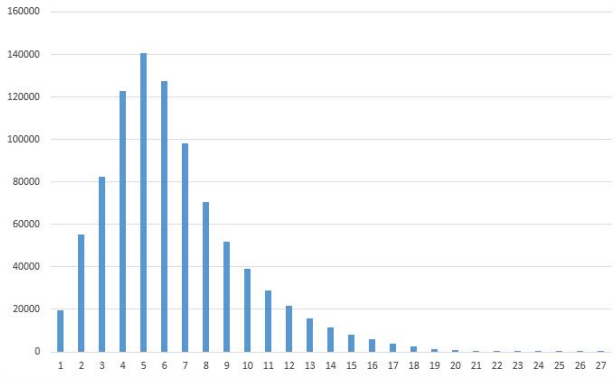

Figure 14. Hops Between Paper of Same Author

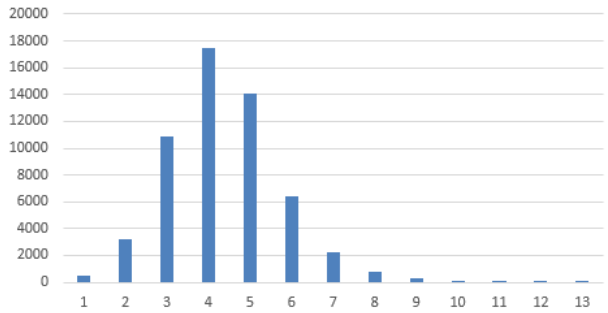

Figure 15. Hops Between Paper of Same Conference

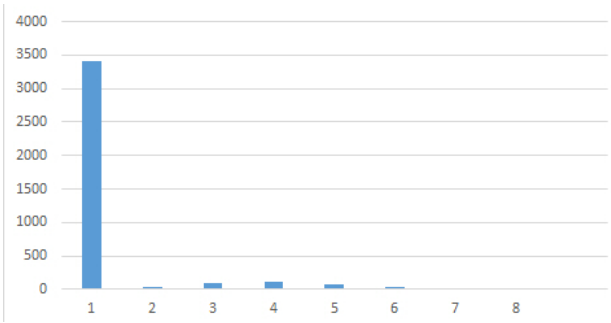

Figure 16. Hops Between Paper of Same Author and Conference

According to the Katz measure, we set the decay parameter $\eta = 0.005$. After classifying the citation, weights are assigned to edges as follows:

$$\omega_{ij} = \begin{cases} 0.75 & \text{if } p_j \text{ is an important citation of } p_i \\ 0.25 & \text{if } p_j \text{ is an unimportant citation of } p_i \end{cases} \quad (7)$$

We randomly choose 2500 pairs of paper from the AAN dataset for five times, calculating their relevance and taking the average result.

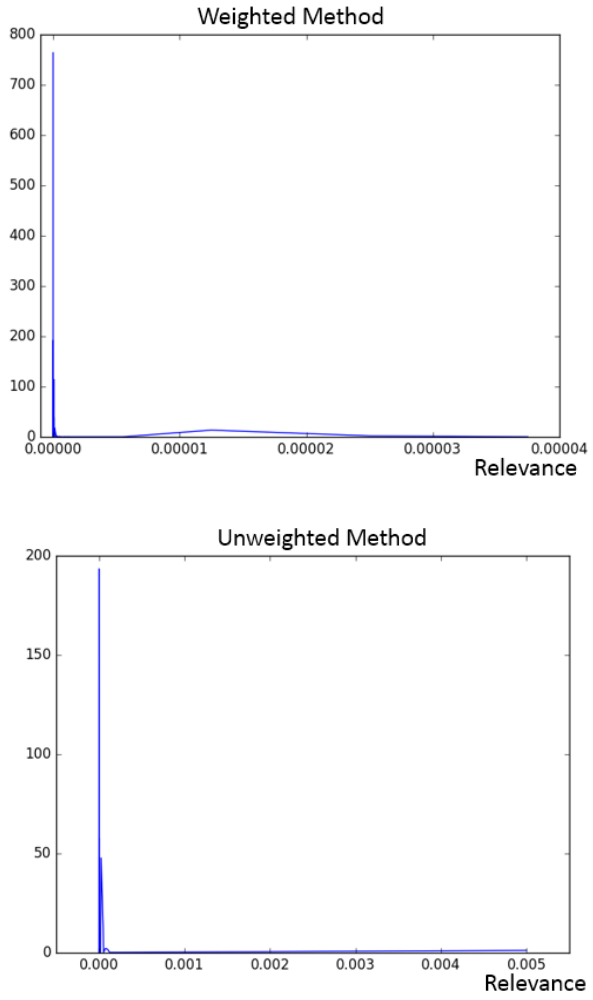

Figure 17.Relevance Distribution

The weighted model performs better when evaluating the strength of relationships between two papers that are closely connected(with shortest path length less than 3) or are remotely connected(with shortest path length greater than 6).

V. Conclusion

We introduce two methods to deal with relationship mining problem across academic network which incorporates the network topology and node information. To evaluate the performance of clustering, we also design corresponding experiments to visualize the clustering to make it more clear.

As novel relationship mining strategy uses both local information and global information of the network out model learns relationships that encodes structural regularities. By introducing more information in the network itself and setting the adaptive threshold we can get different degree of relevance.

Out future work is to combine the two methods are takes advantages of the characteristics of different methods to suit different kinds of relationships.

References

[1] Naimish Agarwal and Gora Chand Nandi. Supervised q-walk for learning vector representation of nodes in networks. *CoRR*, abs/1710.00978, 2017.

[2] Yuxiao Dong, Nitesh V. Chawla, and Ananthram Swami. metapath2vec: Scalable representation learning for heterogeneous networks. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Halifax, NS, Canada, August 13 - 17, 2017*, pages 135–144, 2017.

[3] Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*, pages 855–864, 2016.

[4] Jiangning He, Hongyan Liu, Raymond Y. K. Lau, and Jun He. Relationship identification across heterogeneous online social networks. *Computational Intelligence*, 33(3):448–477, 2017.

[5] David Liben-Nowell and Jon M. Kleinberg. The link-prediction problem for social networks. *JASIST*, 58(7):1019–1031, 2007.

[6] Frank Lin and William W. Cohen. Semi-supervised classification of network data using very few labels. In *International Conference on Advances in Social Networks Analysis and Mining, ASONAM 2010, Odense, Denmark, August 9-11, 2010*, pages 192–199, 2010.

[7] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to information retrieval*. Cambridge University Press, 2008.

[8] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781, 2013.

[9] Supriya Pandhre, Himangi Mittal, Manish Gupta, and Vineeth N. Balasubramanian. Stwalk: learning trajectory representations in temporal graphs. In *Proceedings of the ACM India Joint International Conference on Data Science and Management of Data, COMAD/CODS 2018, Goa, India, January 11-13, 2018*, pages 210–219, 2018.

[10] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: online learning of social representations. In *The 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '14, New York, NY, USA - August 24 - 27, 2014*, pages 701–710, 2014.

[11] Dragomir R. Radev, Pradeep Muthukrishnan, and Vahed Qazvinian. The ACL anthology network corpus. In *Proceedings, ACL Workshop on Natural Language Processing and Information Retrieval for Digital Libraries*, Singapore, 2009.

[12] Leonardo Filipe Rodrigues Ribeiro, Pedro H. P. Saverese, and Daniel R. Figueiredo. *struc2vec*: Learning node representations from structural identity. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Halifax, NS, Canada, August 13 - 17, 2017*, pages 385–394, 2017.

[13] Yu Shi, Po-Wei Chan, Honglei Zhuang, Huan Gui, and Jiawei Han. Prep: Path-based relevance from a probabilistic perspective in heterogeneous information networks. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Halifax, NS, Canada, August 13 - 17, 2017*, pages 425–434, 2017.

[14] Henry Small. Co-citation in the scientific literature: A new measure of the relationship between two documents. *JASIS*, 24(4):265–269, 1973.

[15] Yizhou Sun, Jiawei Han, Xifeng Yan, Philip S. Yu, and Tianyi Wu. Pathsim: Meta path-based top-k similarity search in heterogeneous information networks. *PVLDB*, 4(11):992–1003, 2011.

[16] Niket Tandon and Ashish Jain. *35th German Conference on Artificial Intelligence, Germany: Saarbrucken*, pages 24–27, 2012.

[17] Marco Valenzuela, Vu Ha, and Oren Etzioni. Identifying meaningful citations. In *Scholarly Big Data: AI Perspectives, Challenges, and Ideas, Papers from the 2015 AAAI Workshop, Austin, Texas, USA, January, 2015.*, 2015.

[18] Zhilin Yang, William W. Cohen, and Ruslan Salakhutdinov. Revisiting semi-supervised learning with graph embeddings. In *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, pages 40–48, 2016.

[19] Xiaodan Zhu, Peter D. Turney, Daniel Lemire, and André Vellino. Measuring academic influence: Not all citations are equal. *JASIST*, 66(2):408–427, 2015.