# Academic Recommendation Based on Network Embedding

Yu Cong 517030910280
yu.cong@sjtu.edu.cn

Zhou Qinye 517030910281
zhouqinye@sjtu.edu.cn

## ABSTRACT

In this project, we take authors' recommendation and papers' recommendation as the goal, and implement them based on network embedding method.

In our method, we introduce the effect of influence on recommendation system. Firstly by analyzing influence, we get the algorithm of generating influence context. Then we take the classic method of network Eembedding: word2vec to generate eigenvectors. Finally, we recommend based on the cosine distance of the eigenvector.

Our experiment follows the above process and forms a complete recommendation system step by step. Through our experimental results, the feasibility and reliability of our proposed method are proved. In the examples of author recommendation and paper recommendation, we get the recommendation results that users want, and recommend from the best to the worst.

However, there is still room for improvement in our project, such as manual verification of experimental results. In addition, we propose two possible improvement methods, which can make the recommendation results more accurate.

In a word, we have introduced a new perspective, influence, and achieved good results in academic recommendation. This is just the beginning of this work, it is worth more in-depth exploration.

## Group work Division

Yu Cong: Design the algorithm and achieve the experimental steps in section 4.2
Zhou Qinye:Design the algorithm and achieve the data preprocessing in section 4.1

## Keywords

Network embedding, academic recommendation, influence context

## 1. INTRODUCTION

Recommendation system is a hot topic in recent years. In this project, we hope to implement the academic recommendation based on the network embedding algorithm. Our goal is to achieve two kinds of recommendation:

- Author recommendation: given authors, ten authors are recommended.
- Paper recommendation: given papers, three papers are recommended.

We use network embedding to achieve basic recommendation and introduce influence factor. The essence of network embedding is to use a low dimensional vector to represent the points in the network, which can reflect the network structure. Using this method to represent graph can avoid the uneconomical use of adjacency matrix, and we have more methods to use in vector space. For example, vector representation can input any machine learning model to solve specific problems.

The keypoint of our recommendation is that:

- The recommended papers or authors should have high influence;
- The recommended paper or scholar should have a certain correlation with the original input.

In section 2, we introduce some related work; in section 3, we illustrate our algorithm in details; in section 4, we introduce how we implement our experiment step in step; in chapter 5, we show the result of both author recommendation and paper recommendation; in chapter 6, we look into our future development; in chapter 7, we draw out conclusions of the whole project; finally in chapter 8, references are shown.

Abstract, introduction, experiment(4.2), result and future development (Section 1,4.2,5,6) are written by Yu Cong(517030910280) Related work, algorithm for academic recommendation, experiment(4.1) and conclusion (Section 2,3,4.1,7) are written by Zhou Qinye(517030910281).

## 2. RELATED WORK

In the paper "Inf2vec: Latent Representation Model for Social Influence Embedding"[2], Shanshan Feng develop a

new model Inf2vec, which combines both the local influence neighborhood and global user similarity to learn the representations about users in the social network. We got inspiration from the model building process of in2vec, extended the influence relationship to the academic network, and designed our own network embedding model.

In this paper "PathSim: Meta PathBased TopK Similarity Search in Heterogeneous Information Networks"[3], Yizhou Sun study the similarity search that is defined among the same type of objects in heterogeneous networks. They introduce the concept of meta path-based similarity, where a meta path is a path consisting of a sequence of relations defined between different object types (i.e., structural paths at the meta level).

In this paper "Improving Knowledge Graph Embedding Using Simple Constraints"[1], Boyang Ding investigates the potential of using very simple constraints to improve the KG embedding task. Specifically, they examine two types of constraints: non-negativity constraints on entity representations and approximate entailment constraints over relation representations.By using the former, they learn compact representations for entities, which would naturally induce sparsity and interpretability . By using the latter, they further encode regularities of logical entailment between relations into their distributed representations, which might be advantageous to downstream tasks like link prediction and relation extraction. In our work, We used the non-negativity constraints mentioned in this paper in order to induce sparsity and interpretability.

# 3. ALGORITHM FOR ACADEMIC RECOMMENDATION

In this section, we first introduce our Academic Network Model in section 3.1, and then show some observation and define the author influence pair and paper influence pair in section 3.2. Third, we define the influence subgraph in section 3.3 and define the context for a node in section 3.4. Finally, we show our algorithm for academic recommendation in section 3.5.

## 3.1 Academic Network Model

A academic network model can be modeled as a directed graph G=(V,E), where V is the set of authors and papers and E is the set of edges between two nodes. If there is edge between node u and node v (i.e. edge(u,v)), then the relationship of these two nodes have the following cases.

1. If the two nodes are both authors, the relationship is coorperationAt this time, the edge between the two nodes is bidiretional).
2. If the two nodes are both papers, the relationship is citation.
3. If u is a author node and v is a paper node, the relationship is citation.

The academic network modeled above has two kinds of nodes. When we want to recommend authors for a given author and recommend papers for a given paper, we need to obtain the corresponding author subgraph and paper subgraph from graph G.

## 3.2 Influence Pair

In the real world, we have the following observation.

1. If two authors have a cooperative relationship, and both author cite the same paper i, then we can think that one of the scholars' citation of paper i is influenced by another scholar.
2. If paper i cites paper j, and paper i and paper j both cite paper z, then we can think that paper j cites paper z is influenced by paper i.

Based on the above observation, we can define author influence pair and paper influence pair.

*Definition 1(author influence pair):* If there is a cooperative relationship between two authors u and v, and both authors cite the same paper i, then these two authors (u, v) are author influence pair about paper i. That is to say, if there exist edge (u,v), (u,i) and (v,i) in graph G, then (u,v) are author influence pair about paper i. For example, we can see that (u,v) are author influence pair about paper i in Figure 1.

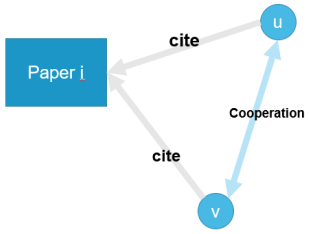



**Figure 1: A Example of Author Influence Pair**

*Definition 2(paper influence pair):* If paper i cites paper j, and paper i and paper j both cite paper z, then these two paper (i,j) is paper influence pair about paper z. That is to say, if there exist edge (i,j), (i,z) and (j,z) in graph G, then (i,j) are paper influence pair about paper z.

## 3.3 Influence Subgraph

Based on the definition of author influence pair and paper influence pair, we can obtain author influence subgraph and paper influence subgraph for each paper. We model author influence subgraph as a Undirected graph and model paper influence subgraph as directed graph.

Specifically, the generation process of author influence subgraph is as follows.

For a given paper(suppose it is paper z), we can find all author influence pairs about paper z in given academic graph G. Then we can use all these author influence pairs to generate the author influence subgraph $G_1 = (V_1, E_1)$ where $V_1$ is the set of all author nodes in the set of author influence

pairs and $E_1$ is the edge of author nodes (if (u,v) are author influence pairs, then there is edge (u,v) ).

Foe example, we can get the author influence subgraph about paper z and paper q for given academic graph G shown in Figure 2.

- For paper z, we can get the author influence subgraph with $V_1 = \{a, b, c\}$ and $E_1 = \{(a, b), (b, c)\}$.
- For paper q, we can get the author influence subgraph with $V_1 = \{b, c, d\}$ and $E_1 = \{(b, c), (b, d)\}$.

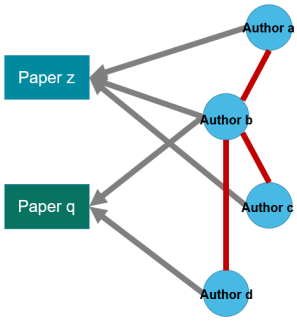



**Figure 2: Example 1 of academic graph G**

Similarly,the generation process of paper influence subgraph is as follows.

For a given paper(suppose it is paper z), we can find all paper influence pairs about paper z in given academic graph G. Then we can use all these paper influence pairs to generate the paper influence subgraph $G_2 = (V_2, E_2)$ where $V_2$ is the set of all paper nodes in the set of paper influence pairs and $E_2$ is the edge of paper nodes (if (u,v) are paper influence pairs, then there is edge (u,v) ).

Foe example, we can get the paper influence subgraph about paper z and paper q for given academic graph G shown in Figure 3.

- For paper z, we can get the paper influence subgraph with $V_2 = \{a, b, c\}$ and $E_1 = \{(a, b), (c, b)\}$.
- For paper q, we can get the paper influence subgraph with $V_2 = \{b, c, d\}$ and $E_1 = \{(c, b), (b, d)\}$.

## 3.4 Context

For each node in the influence subgraph, we can get its context sequence. We divide the context sequence into local context and global context. Among them, the local context is obtained by performing random walk on the influence subgraph (same as the random walk in the Deepwalk method). The local context sequence is more indicative of the positional similarity and cooperation relationship of the two nodes (for the author influence subgraph, it is the cooperative relationship, and for the paper influence subgraph, it is the citation relationship). In order to make the recommended author (or paper) have a better correlation with

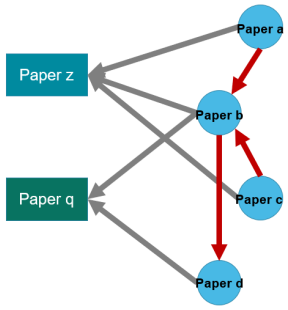



**Figure 3: Example 2 of academic graph G**

the input, the citation to the same paper by two authors (or two papers) can show that the two have the same focus and similarity. So we introduce the global context sequence. Given the influence subgraph and node u in this subgraph, we randomly select some nodes in this subgraph and add them to the context sequence of u.

We can use a parameter $\alpha$ to balance the local context and global context. We first fix the length of context to be L, then the length of local context and global context is $L\alpha$ and $L(1-\alpha)$. For each influence subgraph which include node u, we can get a context C for node u about this influence subgraph. Finally, the final context for node u is the sum of all the context about these influence subgraphs.

Besides, we add the non-negativity constraints to the context mentioned in paper "Improving Knowledge Graph Embedding Using Simple Constraints" in order to induce sparsity and interpretability.

## 3.5 Our Algorithm

The main steps of our algorithm are

1. For each node, find its context sequence.
2. Use word2vec, skip-gram architecture to get the vector representation corresponding to the node.
3. According to the vector representation, we can obtain the cosine similarity between nodes. Thus, the similarity can be used for recommendation

The key point of the algorithm is how we can get the context sequence of a node (probably a author node or a paper node).

# 4. EXPERIMENT

## 4.1 Data Preprocessing

We use the Academic Social Network dataset on AMiner to build our academic graph.

The data set includes paper information, paper citation relationships, scholar information, and scholar partnerships. There are a total of 2,092,356 papers, 8,024,869 paper citation relationships, and 1,712,433 scholars and 4,258,615 scholar cooperation information.

Based on the above data set and our definition of impact subgraphs, we have obtained 6,014 author influence subgraphs and 59,495 paper influence subgraphs.

## 4.2 Experimental Steps

The content of this part may overlap with the content of the previous algorithm part, but we still want to explain it from the perspective of experimental implementation.

We can summarize the experiment into four steps:

1. Generate influence context sequence;
2. Obtain eigenvector of certain node;
3. Calculate cosine distance between one source node and other nodes;
4. Sort the distance and give recommendation advice.

### 4.2.1 Generate Influence Context Sequence

With preprocessed data, we can generate a general graph with authors(papers) as nodes and author partnership(paper citation relationship) as undirected edges(directed edges). For each node in general graph, we can generate its influence subgraph. It is in the influence subgraph that we generate the influence context sequence corresponding to specific node. In our implementation, the influence context sequence is represented as a list containing a sequence of nodes. The sequence of nodes is got in the following way:

1. Random walk in influence subgraph. By this step, we get a sequence $C_1$ of length $L$(one can decide the value of $L$ to any number desired).
2. Random sample in influence subgraph. By this step, we get a sequence $C_2$ of length $L$.
3. Obtain context sequence. Here we define an influence factor $\alpha$ to indicate the importance of $C_1$ compared to $C_2$. The final context sequence is defined as $C = C_1[0 : L\alpha] + C_2[0 : L(1-\alpha)]$. So finally we get a context sequence of length $L$, which is a splicing of the front part of two sequences.

### 4.2.2 Obtain Eigenvector

Now we have the influence context sequence of each node in the graph. Then we obtain eigenvector of each node using word2vec method. For the convenience of the next step, we store all the feature vectors in a TXT file.

From the previous description, we know that nonnegativity can cause sparsity and interpretability, which is very beneficial to our experiment. So we set all the values in the eigenvector to be nonnegative in practice (if it is negative, let it be 0).

### 4.2.3 Calculate Cosine Distance

For a given node, we need to calculate its cosine distance to any other node in the general graph. Following is the equations to calculate cosine distance:

$$consine\ similarity = \frac{AB}{\sqrt{|A||B|}} \tag{1}$$

$$consine\ distance = \frac{1 - cosine\ similarity}{2} \tag{2}$$

where A is eigenvector of a given node and B is eigenvector of another node in the graph.

### 4.2.4 Sort and Recommend

We sort the cosine distance from small to big. For authors, we get ten other authors with smallest cosine distance. And for papers, we recommend top three papers.

# 5. RESULT

## 5.1 Outcome Measurement

Before we show the final result, we should determine: What result can be viewed as a good result? What is the property of a good result that users want?

To answer these questions, we define three indicators to show the quality of the recommendation results(take author recommendation as an example):

- Local influence power: if the paper quotes references, the author of the references will affect the author of the paper. We will check the local influence power both from given author to recommended author and from recommended author to given author.
- Overall influence power: investigate number of citations of recommended author's main paper. It is fine to investigate author's total citation times.
- Academic relevance: see if the recommended author and the given author are working in the same field.

Note that these indicators may overlap or differentiate with some concepts we mentioned before. Because we give these three indicators from the perspective of user preferences in practice. Although they do not conflict with the previous content in meaning, we are afraid that readers will confuse these concepts, so we specifically use different phrases.

## 5.2 Result Verification

Until now we have the indicator to show whether a recommendation is good or not. Then we randomly choose one author and one paper to examine our recommendation method.

### 5.2.1 Author Recommendation

We have the honor to choose a leader in the industry: Serge Abiteboul (we call him S.A. in following paragraphs for convenience). And we get ten authors, as shown in Figure 4.

First we take a view at basic information of S.A., as shown in Figure 5. From this brief introduction we find that S.A. is focusing on computer science and related areas, and have made great contribution in the areas of finite model theory, database theory, and database systems.

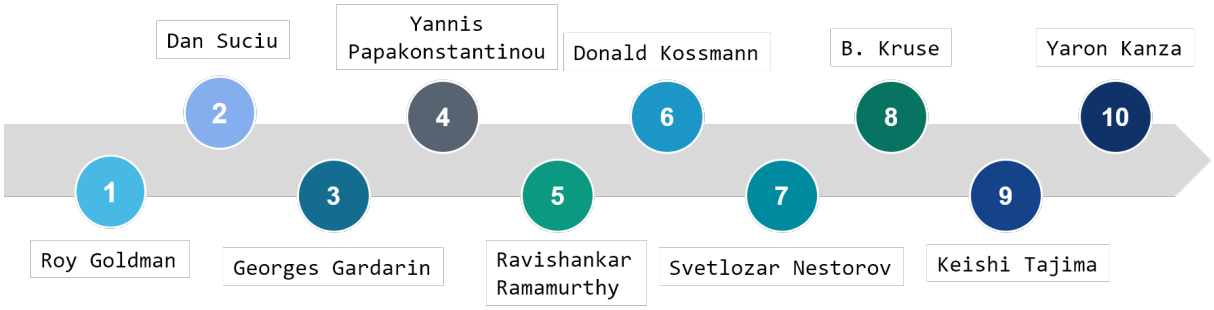
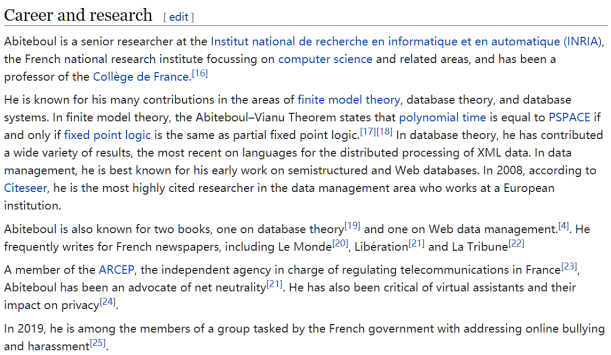



**Figure 4: Ten recommended author of S.A.**



Career and research [ edit ]

Abiteboul is a senior researcher at the Institut national de recherche en informatique et en automatique (INRIA), the French national research institute focussing on computer science and related areas, and has been a professor of the Collège de France.[16]

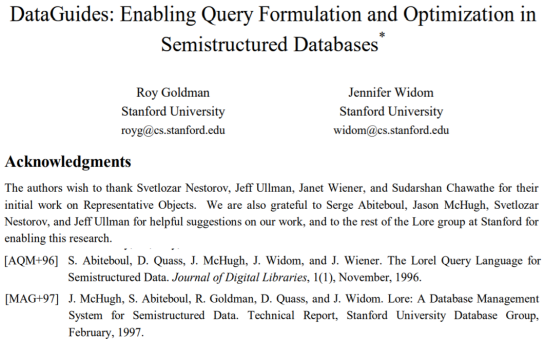
He is known for his many contributions in the areas of finite model theory, database theory, and database systems. In finite model theory, the Abiteboul–Vianu Theorem states that polynomial time is equal to PSPACE if and only if fixed point logic is the same as partial fixed point logic.[17][18] In database theory, he has contributed a wide variety of results, the most recent on languages for the distributed processing of XML data. In data management, he is best known for his early work on semistructured and Web databases. In 2008, according to Citeseer, he is the most highly cited researcher in the data management area who works at a European institution.

Abiteboul is also known for two books, one on database theory[19] and one on Web data management.[4]. He frequently writes for French newspapers, including Le Monde[20], Libération[21] and La Tribune[22]

A member of the ARCEP, the independent agency in charge of regulating telecommunications in France[23], Abiteboul has been an advocate of net neutrality[21]. He has also been critical of virtual assistants and their impact on privacy[24].

In 2019, he is among the members of a group tasked by the French government with addressing online bullying and harassment[25].

**Figure 5: Basic information of S.A.**



DataGuides: Enabling Query Formulation and Optimization in Semistructured Databases*

Roy Goldman
Stanford University
royg@cs.stanford.edu

Jennifer Widom
Stanford University
widom@cs.stanford.edu

**Acknowledgments**

The authors wish to thank Svetlozar Nestorov, Jeff Ullman, Janet Wiener, and Sudarshan Chawathe for their initial work on Representative Objects. We are also grateful to Serge Abiteboul, Jason McHugh, Svetlozar Nestorov, and Jeff Ullman for helpful suggestions on our work, and to the rest of the Lore group at Stanford for enabling this research.

[AQM+96] S. Abiteboul, D. Quass, J. McHugh, J. Widom, and J. Wiener. The Lorel Query Language for Semistructured Data. *Journal of Digital Libraries*, 1(1), November, 1996.

[MAG+97] J. McHugh, S. Abiteboul, R. Goldman, D. Quass, and J. Widom. Lore: A Database Management System for Semistructured Data. Technical Report, Stanford University Database Group, February, 1997.

**Figure 6: Information of Roy Goldman's first paper.**

Then let us focus on top three recommeded authors:

1. Roy Goldman. Figure 6 and Figure show his two main papers. We can find that Roy Goldman has close academic cooperation with S.A. Furthermore, the first paper was cited 1939 times, and the second paper was cited 1075 times. So Roy Goldman reaches our three indicators perfectly.

2. Dan Suciu. Figure 8 show the basic information of Dan Suciu and Figure 9 and 10 shows his first and second main paper. In first paper, he cooperated with S.A. and in second paper he cited a paper of S.A. Furthermore, the first paper was cited 2246 times, and the second paper was cited 777 times. So Dan Suciu can also reach our three indicators, although slightly worse than first one in local influence power.

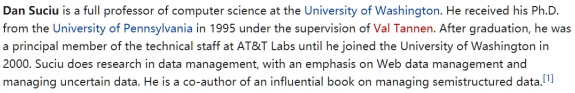
3. Georges Gardarin. Figure 11 and Figure 12 show his two main papers. We can find that Roy Goldman also has close academic cooperation with S.A. However, his global influence power is a lot worse than previous two. To be specific, the first paper was cited 335 times, and the second paper was cited only 42 times. We would evaluate Georges Gardarin as a good choice, but not as good as Roy Goldman and Dan Suciu.

Lore: a database management system for semistructured data

Authors: Jason McHugh, Serge Abiteboul, Roy Goldman, Dallas Quass, Jennifer Widom

Authors Info & Affiliations

**Figure 7: Information of Roy Goldman's second paper.**



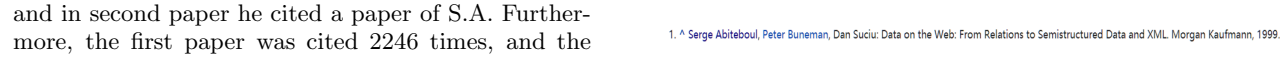
**Dan Suciu** is a full professor of computer science at the University of Washington. He received his Ph.D. from the University of Pennsylvania in 1995 under the supervision of Val Tannen. After graduation, he was a principal member of the technical staff at AT&T Labs until he joined the University of Washington in 2000. Suciu does research in data management, with an emphasis on Web data management and managing uncertain data. He is a co-author of an influential book on managing semistructured data.[1]

**Figure 8: Information of Dan Suciu.**

So you can see, the top three places we recommend are arranged from the best to the worst according to the indicators, and all of them meet the requirements of the indicators well.

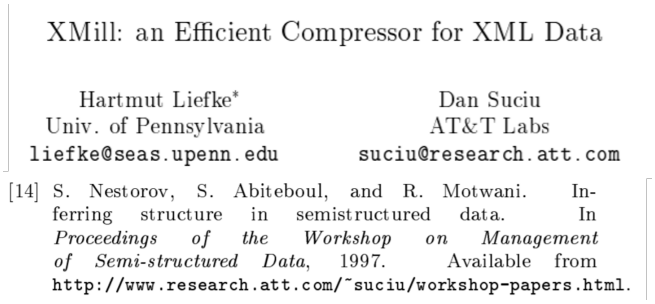
Then we came up with an interesting question: what would happen if we took the first three recommended scholars as input to check their recommended scholars?

Figure 13, 14 and 15 show the answer of above question. However, we frustratedly find no place for S.A. (whose ID is 555493). At the same time, we find another interesting phenomenon: author with ID 1114017 and author with ID 479495 are mutual recommded in the first place. In our consideration, there might be two reasons:



1. ^ Serge Abiteboul, Peter Buneman, Dan Suciu: Data on the Web: From Relations to Semistructured Data and XML. Morgan Kaufmann, 1999.

**Figure 9: Information of Dan Suciu's first paper.**


XMill: an Efficient Compressor for XML Data

Hartmut Liefke*
Univ. of Pennsylvania
liefke@seas.upenn.edu

Dan Suciu
AT&T Labs
suciu@research.att.com

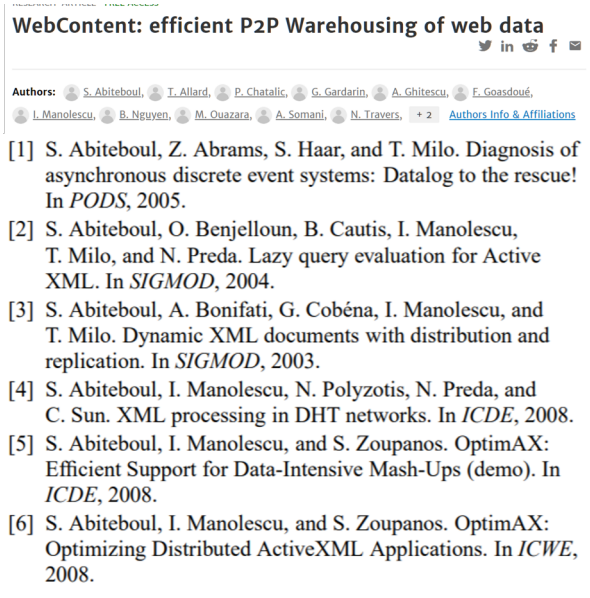
[14] S. Nestorov, S. Abiteboul, and R. Motwani. Inferring structure in semistructured data. In *Proceedings of the Workshop on Management of Semi-structured Data*, 1997. Available from http://www.research.att.com/~suciu/workshop-papers.html.

**Figure 10: Information of Dan Suciu's second paper.**


Join and Semijoin Algorithms for a Multiprocessor Database Machine

Authors: Patrick Valduriez, Georges Gardarin Authors Info & Affiliations

**Figure 11: Information of Georges Gardarin's first paper.**


WebContent: efficient P2P Warehousing of web data

Authors: S. Abiteboul, T. Allard, P. Chatalic, G. Gardarin, A. Ghitescu, F. Goasdoué, I. Manolescu, B. Nguyen, M. Ouazara, A. Somani, N. Travers, + 2 Authors Info & Affiliations

[1] S. Abiteboul, Z. Abrams, S. Haar, and T. Milo. Diagnosis of asynchronous discrete event systems: Datalog to the rescue! In *PODS*, 2005.

[2] S. Abiteboul, O. Benjelloun, B. Cautis, I. Manolescu, T. Milo, and N. Preda. Lazy query evaluation for Active XML. In *SIGMOD*, 2004.

[3] S. Abiteboul, A. Bonifati, G. Cobéna, I. Manolescu, and T. Milo. Dynamic XML documents with distribution and replication. In *SIGMOD*, 2003.

[4] S. Abiteboul, I. Manolescu, N. Polyzotis, N. Preda, and C. Sun. XML processing in DHT networks. In *ICDE*, 2008.

[5] S. Abiteboul, I. Manolescu, and S. Zoupanos. OptimAX: Efficient Support for Data-Intensive Mash-Ups (demo). In *ICDE*, 2008.

[6] S. Abiteboul, I. Manolescu, and S. Zoupanos. OptimAX: Optimizing Distributed ActiveXML Applications. In *ICWE*, 2008.

**Figure 12: Information of Georges Gardarin's second paper.**

```
给定学者ID 1114017
推荐学者1 479495 0.03631798997023694
推荐学者2 1240639 0.04677435053750273
推荐学者3 205043 0.0562339758885918
推荐学者4 992703 0.06487339401358261
推荐学者5 993669 0.06570798019397672
推荐学者6 653760 0.0679570194180677
推荐学者7 1535661 0.07149440071736973
推荐学者8 90939 0.07405869995399011
推荐学者9 1172277 0.07600679633212293
推荐学者10 500412 0.07646953305501253
```

**Figure 13: Recommended authors' ID of Roy Goldman.**

```
给定学者ID 479495
推荐学者1 1114017 0.03631798997023694
推荐学者2 1240639 0.04157142334948649
推荐学者3 653760 0.04881554522106474
推荐学者4 992703 0.0621941237273364
推荐学者5 459021 0.06731859871384005
推荐学者6 1172277 0.07261615849181569
推荐学者7 462814 0.07455766678671627
推荐学者8 1535661 0.07479709856811972
推荐学者9 867986 0.07682297470642957
推荐学者10 1699772 0.07767479458483639
```

**Figure 14: Recommended authors' ID of Dan Suciu.**

- In paper recommendation, we use undirected graph, which means we consider mutual influence rather than one-way influence;
- When generating influence context, we use a factor $\alpha$. Maybe we set the $\alpha$ too big so that the model pay too much attention to local influence.

### 5.2.2 *Paper Recommendation*

Also we test the result of paper recommendation. We randomly choose one paper: The temporal query language TQuel. The paper studies in Relational database, Relational database management system, Computer science, QUEL query languages and Database.

We get its top three recommended papers:

1. Evaluation of relational algebras incorporating the time dimension in databases.
   - Citation: 264;

```
给定学者ID 1535661
推荐学者1 650652 0.040395587497300545
推荐学者2 657313 0.0425559985755094
推荐学者3 1240639 0.057232327700837304
推荐学者4 500412 0.05888280307356608
推荐学者5 1541372 0.05890473388708811
推荐学者6 205043 0.0596015465766917
推荐学者7 459021 0.06601216765564266
推荐学者8 1636800 0.06811758033992815
推荐学者9 1626749 0.07038325928289496
推荐学者10 1114017 0.07149440071736973
```

**Figure 15: Recommended authors' ID of Georges Gardarin.**

- Citation relationship: cite the given paper;
- Study area: Relational database, Multiple time dimensions, Computer science, Transaction time and Database.

2. A Temporal Relational Algebra as Basis for Temporal Relational Completeness.
   - Citation: 112;
   - Citation relationship: cite the given paper;
   - Study area: Codd's theorem, Domain relational calculus, Relational model, Completeness (statistics), Computer science.
3. Historical Multi-Media Databases.
   - Citation: 29;
   - Citation relationship: not cite the given paper (but it quoted But it quoted (a paper very similar to the one under investigation) ,a paper very similar to the one under investigation);
   - Study area: Computer science, Database.

Through the above results, we can get that the paper recommendation also basically meets our requirements, and is recommended from the best to the worst.

# 6. FUTURE DEVELOPMENT

Based on our existing work, there are still many interesting topics to be explored. We pick out the three most significant topics:

- The author's cooperation times may be many times, and the edge can be weighted in the graph to make the recommendation more accurate.
- Use metapath2vec method, and add influence factors to compare the effect of the two methods.
- Use a text-based recommendation system to evaluate our recommendation results (measure the semantic similarity of the whole paper or abstract).

# 7. CONCLUSIONS

In this paper, we first propose a new network embedding model to get the vector representation of author node and paper node in academic graph. And then we design a algorithm based on the our network embedding model to do academic recommendation (given author, recommend authors and given paper, recommend papers). Finally, we evaluate our model and algorithm in the Academic Social Network dataset in Aminer and prove that our algorithm is effective.

# 8. REFERENCES


[1] B. Ding, Q. Wang, B. Wang, and L. Guo. Improving knowledge graph embedding using simple constraints. 1:110–121, 2018.

[2] S. Feng, G. Cong, A. Khan, X. Li, Y. Liu, and Y. M. Chee. Inf2vec: Latent representation model for social influence embedding. pages 941–952, 04 2018.

[3] Y. Sun, J. Han, X. Yan, S. P. Yu, and T. Wu. Pathsim: Meta path-based top-k similarity search in heterogeneous information networks. *PVLDB*, pages 992–1003, 2011.