Academic Journal Draft*

Matin Pirouz
California State University, Fresno
Fresno, California
mpirouz@csufresno.edu

Alexander Madrigal
California State University, Fresno
Clovis, California
lxndrmdrgl@mail.fresnostate.edu

Nathaniel Madrigal
California State University, Fresno
Clovis, California
nthnlmdrgl@mail.fresnostate.edu

***Abstract*—Maintaining academic connections and experience is an important problem for academic fields. Connecting similar scholars may assist in facilitating the creation of new connections and the sharing of experiences. As such, a method to generate recommendations for collaborations between scholars utilizes academic social networks that may be used to represent the relationships between relevant information in an academic setting. In this paper, we consider how to represent our academic social network for scholar generated academic information. We study the organization of academic databases as well as how to share information between scholars while maintaining privacy. To find optimal recommendations, we study algorithms on how to find significant parts of the network and how to represent those recommendations to scholars. We present a software that allows for efficient writing and sharing of academic information between scholars in an academic social network.**

I. Introduction

In one's academic career, maintaining academic connections and relevant academic experience is critical to working on research projects. As research continues to grow, it is of increasing importance to develop a method that allows scholars to make academic collaborations, exchange experiences, and find new research opportunities. This requires finding and offering recommendations to scholars of possible collaborators, as well as sharing relevant experiences. As such, a suitable model can be made that allows scholars to maintain a personal academic social network of their connections and a record of academic works.

An academic social network can be modeled using entities and the relationships between them. Suppose that we represent our network with scholars, the connection between them can be represented by an edge between two scholars who have worked together before. The complexity of the academic social network depends on the number of entities added and the types of relationships between those entities.

Recommendations are modeled by offering potential relationships between entities. Recommendations should consider the current academic social network and provide the most probable relationship between entities. This serves as the most likely and best suited connection that is recommended to the scholars. These recommendations can serve as a way to connect projects between scholars and exchange information about that entity. They can also serve to make new connections between scholars who share entities and relationships.

The relationships between entities can be made of various attribute information. Suppose that we are describing the relationship between two projects, the relationship can include the number of common collaborators. This attribute information provides a more detailed view of the relationship between entities and can be used to generate more representative recommendations for the network.

When sharing entities and recommending connections to scholars through an academic social network for personal use, we can also consider the privacy of a scholar's network. An open social network would allow sharing and recommendations between any entities in the network. If the goal is to provide a safe personal network, scholar's entries can be concealed from the networks of other scholars. Restrictions can be placed on this academic social network to limit the exposure of scholars' entries. We can say that sharing entities can only be done between connected scholars and that connections can only be made between scholars who share a common connection. This provides an academic social network of scholars and entities with common connections and proximity within the network.

In this paper, we provide a software to make an academic social network that provides possible recommendations for scholars to connect and share with. Entities and relationships within the academic social network will be generated and updated using scholar generated records for individual scholars. Recommendations are made using the best possible predictions of relationships in the network and ensure the privacy of scholars through restrictions on connecting and sharing. The use of records in the academic social network is used to share academic information between connected scholars.

A. Related Work

1) Link Prediction Problem: In [1], they sought to predict the edges of a social network that will be added to the network in the future using existing data in the network. They utilized co-author networks, networks where authors are the nodes and edges are between pairs of authors that have written a paper together. Given a pair of nodes, they produced multiple methods that generated connection weight scores that represent the proximity of nodes for the entire network. The score for the set of neighbors could be the number of common neighbors between the nodes. The Jaccard coefficient method measured the probability of the pair of nodes sharing a neighbor from any neighbor of the two nodes to be the score. The method by Adamic and Adar [6] returns a score which weights nodes with less common neighbors more heavily. The preferential attachment method [7], [9] returns a score which considers the number of neighbors of the pair of nodes. The method by Katz [8] returns a score set to the sum of all paths from one of the nodes in the pair to the other node which weighs shorter paths more heavily. The hitting time method uses a random walk between the pair of nodes and returns the expected steps needed to walk as the score. They found that all of the methods predicted the edges of the network better than random.

In [12], they modeled scientific collaboration networks by making connections using co-author relationships in different scientific fields. They found that these networks made a small world network, a network represented with high clustering and low distances between nodes such that the number of steps to traverse between two random nodes is proportional to the logarithm of the number of nodes in the network. They found that distances between nodes in co-author collaboration networks were strongly correlated to the logarithm of authors in the network. Furthermore, clustering in the network was high suggesting that authors are more likely to be connected if they share a common author. In [1], they illustrate that small world networks are not useful for link prediction. Since nodes have short average distances in a network, they say that short distances between unrelated authors, an example being authors of different scientific fields are not a useful property when they model link prediction methods assuming connections between nodes of different fields are rare.

One field related to link prediction is community detection in social networks. In [15], a community is represented by a subset of vertices in a network such that those nodes have more edges inside the community than edges connecting the community to the network. In [16], they model community detection using modularity which gives the fraction of edges within the community minus the edges within a community for a randomized network. They found that community detection algorithms exist for disjoint and overlapping communities. Different community detection algorithms have advantages and disadvantages in regard to their ability to detect overlapping communities, needing information beforehand such as clustering, uncertainty, and accuracy of results. Community detection algorithms also exist for heterogeneous networks, networks with many types of nodes and edges. In [17], they propose a heterogeneous network for community detection by utilizing user models, where a user model is a node in the network and represents a combination of user interests and social relationships. Utilizing this heterogeneous network, they perform community detection by finding multiple seeds that compose communities. Using the initial set of communities, they find the overlapping communities and generate a hierarchical network with communities as nodes for community detection. Using artificial and real datasets, their methods outperformed other methods and are applicable to large scale social network sites.

2) Relational Social Recommendation : In [2], they describe an algorithm which creates an entity-relation social network using a social graph for scholars which recommended users peers at scientific conferences. Recommendations returned a ranked list of peers for users which was ordered by similarity to the users. Utilizing weighted and labeled connecting paths they also provided guidance to users when recommending peers. The rank for nodes in the graph was determined using the Personalized PageRank random walk algorithm. They found that the relational social recommendation algorithm assigned recommendations of higher rank for recommendations relevant to the user. The recommendations that were highly ranked included recommendations to peers the user already knew and relevant peers the user did not know.

3) Recommendation Framework Using Word Embedding and Network Embedding Models : In [3], they created a framework to make ranked recommendation lists to recommend possible academic collaborations for scholars. Using a co-author network which represents authors as nodes and edges as the co-author relationship between authors, they found recommendations utilizing both the scholar's research interests and the topology of the network. They generated a model to weigh scholar's research interests using web resources. The method of taking keywords and generating weights to represent research interests was not sufficient because the words did not consider the context or semantics of source documents. The Word2Vec model was used to generate a word embedding that understood the context and semantics of words to make up the research interests. Then the similarity of research interests between scholars was found between pairs of scholars using cosine similarity methods. To find recommendations considering the network topology of the co-author network, they found that link prediction methods were not sufficient to accurately make predictions since they required manual design and the selection of methods to obtain effective results. The Node2Vec method was used to transform the co-author network into vectors and measure network topology similarities using cosine distances. The Word2Vec and Node2Vec models were combined using the CombMNZ method to make ranked recommendations that considered the results of both models.

4) Multidimensional Academic Networks : In [4], they propose a multidimensional academic network to recommend collaboration using scholarly big data, multiple types of academic data, and activities. The network was represented with researchers and articles with the edges of the network thus consisting of multiple types of co-author, citation, and academic relationships. The relationships between the researchers and articles allow for edges in the graph to have weights 0 to 1 to represent the dynamic relationships between edges of the network. Using a Random Walk with Restart algorithm and measuring the academic influence of nodes within a time period, they found a ranked list of nodes that represent influential researchers and articles that can be used as recommendations for the network. Their model was effective in making recommendations that considered both publications and social activity in academic environments.

In [14], they made an academic social network to make scholar-friend recommendations were scholar-friends where scholar-friends represent friends in academic social networks. They propose that scholar-friend recommendations require multiple entities and relationships between those entities including researchers, affiliations, research interests, papers, updates, groups, and academic resources requiring a heterogeneous network. These entity relationships were then used to generate weights between researchers and thus scholar-friend recommendations. They found that scholar-friend recommendations were different from friend and expertise recommendations. Friend recommendations consider entertainment and relationships that may be long term meanwhile expert recommendations consider short term problem oriented relationships. They demonstrate that scholar-friend relationships have aspects of friend and expertise relationships that are research oriented shown through the entities of the heterogeneous network.

II. Methods

1) Relational Database: It is useful for a scholar to keep a journal of academic information for themselves and those they are connected to in the network. Our journal consists of scholars, educations, projects, organizations, connections between scholars, and schedule records. These records are provided by the scholar using the software. Each record has attribute information. It is important to consider the many possible relationships between journal entries as well as the multiple types of data that can be stored in entries as attributes. Thus, the journal records were represented using a relational database.

A relational database has data stored in tables where rows represent entities and columns as attributes [5]. A table for each record is sufficient to record all data within the journal where each journal entry to a record is an inserted row to the associated table, and the columns store attribute information for each inserted record. New journal entries can be inserted by adding a row to the correct table depending on the type of entry and filling in the columns with the correct attribute information. Relational databases are useful since they allow for easy modification to the database while maintaining structure and tabular for simplicity. Each entity has a unique identification number in its table as an attribute called a primary key. Entities may have the primary keys of other entities as attribute information called foreign keys. Foreign keys can be used to model relationships between tables in three ways, one-to-one, one-to-many, and many-to-many. These three relationships allow for complex relationships between entities of the relational database for any combination of tables. The relational database also enables queries of data through the relationships in the relational database that allow for easy searching for connected data.

One-to-one relationships represent how one entity of a table can be connected to at most one other entity of another table. For instance, each entity of the scholar table has one connection to an entity in the account table representing login information for the software. One-to-one relationships are modeled by storing the primary key of one entity as a foreign key to the other entity [5]. For the relationships between the scholar table and the account table, every account entity has a foreign key to a unique scholar entity.

One-to-many relationships represent how one entity of a table can be connected to many entities of another table. In one-to-many relationships, the entities of one table with the foreign key are connected to the entities of another table where multiple entities can have the foreign key to the same entity. The table with the foreign key represents the many entities that a single entity of another table can be connected to [5]. For instance, multiple project entities can have the same foreign key as a scholar entity since each scholar can be associated with more than one project. In our software, the scholar table has one-to-many relationships with projects, organizations, educations, schedules, scholar connections, and requests where requests are requests sent by a scholar to connect with another scholar.

Many-to-many relationships represent how entities of a table can have many connections to entities of another table. In many-to-many relationships, foreign keys in one table alone are not sufficient enough to represent this relationship. An additional table is utilized which contains foreign keys to both entities of both tables, where each entity of the additional table is a relationship between two entities [5]. For instance, scholar connections may be associated with many projects, and projects may be associated with many scholar connections. In our software, many-to-many relationships are used to represent the relationships between projects and scholar connections, organizations and scholar connections, projects and schedules, and organizations and schedules.

Scholar connections can be modeled using the one-to-many relationship where each scholar is in a relationship with multiple other scholars or the many-to-many relationship where each connection is the pair of two scholars. Modeling using the one-to-many relationship has redundancy in the relational database since two records in the database exists for the same relationship between two scholars. The one-to-many relationship scholar connection approach was selected instead of the many-to-many relationship to enable simple queries on the database. Each entry in the connection table has an attribute for the scholar primary key and the foreign key to the primary key of the connected scholar. This enables simple queries when searching for connections of a particular scholar by simply finding all foreign keys for the respective entries in the table with the same primary key as the scholar. The many-to-many relationship for the scholar connections has both scholar primary keys as foreign keys in its attributes. When querying for scholar connections of a particular scholar in this method, it is not clear which attribute is the scholar and which attribute is the scholar connection for any given entry.

2) Using the Relational Database: Since journal entries are entered by individual scholars, the relational database considers the locality of the scholar. Each table except for the tables used to model many-to-many relationships have foreign keys to the scholar creating the entity. Entities with relationships with other entities are within the context of the scholar. For instance, the entities of the organization and schedule many-to-many relationship table represent one of the scholar's organizations being associated with one of the scholar's schedules.

The relational database management system used was MySQL. The web application was generated using Flask, a web framework for Python. SQLAlchemy was used as the object-relational mapping to allow access to the database in Python. The Flask-SQLAlchemy extension was used to utilize SQLAlchemy for Flask. In Flask-SQLAlchemy, relational database tables are created by defining models and tables where tables are used for many-to-many relationship tables [10]. Flask-SQLAlchemy allows for modification and querying of the database.

Our software allows for the insertion and deletion of entities for scholars. HTML inputs were used for insertion of categorical and numerical attribute information for entities. Naming entities used text inputs for attributes. Categorical attributes were imputed using single of multiple select keywords in a list or search format. Time and date inputs were utilized for numerical attributes. Inputting entity relationships for a given scholar utilizes entities in the relational database connected with the scholar. Entities generated by the scholar can be connected together with one entity being in a relationship with one or multiple other entities from the same scholar to represent one-to-one, one-to-many, and many-to-many relationships.

Querying of the database was utilized to showcase related entity information for each scholar's scholar connections. Desired entities and their respective attributes were returned using SQL queries. Select statements are used to select data from the database. Join clauses are used to combine tables given a related attribute. Where clauses are used to filter the query using conditions on the attribute information. The combination of select statements, join clauses, and where clauses, are sufficient to find entities related to each scholar in some way. For each scholar, our software is capable of finding unconnected scholars through a common scholar connection, as well as projects and organizations associated with scholar connections. Privacy is easily maintained by only sharing entries with attributes set to public, allowing for view-ability to other scholars. Each scholar can only see their own entries and attribute information of other entries queried from their scholar connections. Shared entries for each scholar do no display which scholars they come from nor does it provide information for why the entry was shared.

3) Recommendations: With a relational database for scholarly journal activities, an academic social network can be created to recommend scholars to other scholars. The academic social network was created in NetworkX, a Python library for graphs and networks. An academic social network graph was created using entities as nodes and relationships between entities as edges. A scholar connection graph was created using scholars as nodes and the connections between scholars as edges.

With a graph, NetworkX link prediction algorithms were used to generate a ranked list of connected pairs of scholars. Our software tested NetworkX's implementation of [1] the Jaccard coefficient algorithm [11]. With the ranked list of connected pairs of scholars, these identity relationships which are significant to the network. A ranked recommendation can be generated for each scholar by only including pairs that include the selected scholar and that do not include an existing scholar connection for the selected scholar.

The ranked recommendation list generated for each scholar allows scholars to request a scholar connection to each of the recommendations. A request table is created in the relational database to represent pending requests for scholars requesting scholar connections with recommended scholars. Each request entity has the foreign key to the primary key of the scholar requesting the connection as well as the foreign key to the primary key of the recipient scholar of the request. For each scholar, a query is made on all requests sent to the given scholar. If the scholar approves the request, a connection is made between both scholars. If the scholar ignores the request, the request entity is deleted from the relational database. The ranked recommendations and requests allow for growth in the edges of the academic social network and give scholars more access to shared data.

III. Experiments

We utilized small scale synthetic datasets to run our experiments. We generated 10 scholars to represent the scholar connection graph used for the recommendation algorithm. Each scholar had between 1-3 scholar connections, and the scholar connections were chosen at random.

In our experiment, we sought to measure the effectiveness of the recommendation ranking generated for each scholar in comparison to the importance of all scholars in the network. One measure of a scholar's importance in a network is represented by its centrality with the graph. The centrality of a node in a graph is a measure of that node's importance to the graph. One quantification of its importance to the graph is degree centrality. The degree centrality of a scholar is represented by the fraction of scholars they are connected to with respect to the entire graph [11]. Degree centrality supposes that graphs that are more connected are more important. Another measure of centrality is closeness centrality which supposes that a node is more important when it is closer to all other nodes in the graph. For our graph, the measure of centrality is the reciprocal of the average shortest path from one scholar to all other scholars [11].

IV. Results

Fig. 1 shows the scholar connection graph for the synthetic dataset. The graph showcases a connected network where the edges of a node illustrate the connection between scholars.

Fig. 2 shows the degree centrality and closeness centrality for each of the scholars for the network shown in Fig. 1. The degree centrality for each scholar in the network is lower than the closeness centrality for each respective scholar. This is influenced by the short distances between the nodes of the small network in comparison to the density of connected scholars in the graph. The degree centrality and closeness centrality both

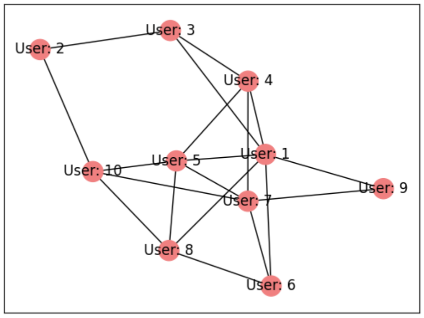

Fig. 1. Scholar Connection Graph

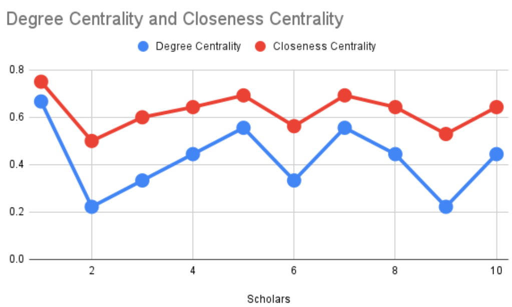

Fig. 2. Centrality Comparison

follow a similar pattern for the relative centralities between the scholars.

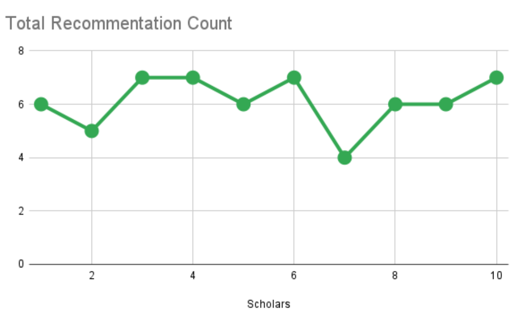

Fig. 3. Total Recommendation Count

Fig. 3 shows the total number of times each scholar got recommended to another scholar for all scholars in the network. Each scholar was recommended to another scholar with near equal likely probability with little to no influence from the scholar's centrality. This is likely due to the small size of the network returning a larger proportion of recommended scholars to each scholar with respect to the total number of scholars in the network.

Fig. 4 shows the sum of recommendation scores for each scholar in the network. The score of a recommendation was a measure of the Jaccord coefficient algorithm between the scholars. The sum of all Jaccord coefficients was taken for each scholar among all the times it was recommended to another scholar. The variance of the sum of recommended scores is much higher than the total recommendation count in Fig. 3. The distribution of the sum of recommendation scores most closely resembles the closeness centrality showcased in Fig. 1.

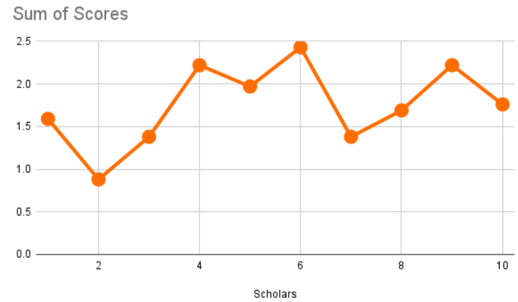

Fig. 4. Sum of Recommendation Scores

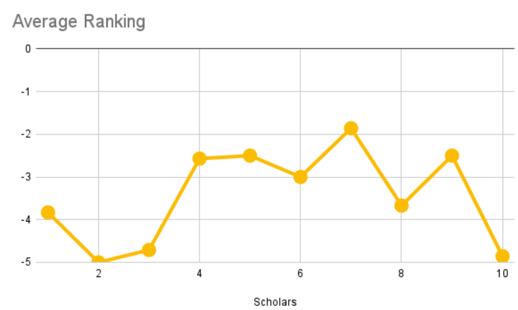

Fig. 5. Average Recommendation Ranking

Fig. 5 shows the average recommendation ranking for each scholar in the network. It only considered the relative ranking of recommendation scores between scholars that got recommended to other scholars. The average ranking for each scholar resembles the sum of recommendation scores between scholars shown in Fig. 4. This is likely a factor of how the sum of recommendation scores is closely related to the average ranking for recommended scholars among a ranked list.

V. Conclusion

We propose a software to support dynamic updates and sharing of academic information through the scholar created private records. Using the relational database, a generated academic social network is capable of recommending important scholars to individual scholars. Expanding the model includes using a multidimensional social network to model more relationships within the data using attribute information to generate weighted edges. More research can be done on the types of academic data included in academic social networks, and representative multi-factorial recommendation methods. Future work involves using the software with real academic information and human feedback to share information and recommendations.

Appendix

TABLE I
NOTATIONS

Entity	A unique object stored and collected as data
Attribute	Characteristics of an entity
Relationship	A connection between entities through foreign keys
Recommendation	A offered potential relationship between unconnected entities
Primary Key	One or more attributes to uniquely identity a row in a table
Foreign Key	The primary key of one table as an attribute in another table

References

[1] D. Liben-Nowell and J. Kleinberg, “The link-prediction problem for social networks,” Journal of the American Society for Information Science and Technology, vol. 58, no. 7, pp. 1019–1031, 2007, doi: https://doi.org/10.1002/asi.20591.

[2] S. Amal, C.-H. Tsai, P. Brusilovsky, T. Kuflik, and E. Minkov, “Relational social recommendation: Application to the academic domain,” Expert Systems with Applications, vol. 124, pp. 182–195, Jun. 2019, doi: https://doi.org/10.1016/j.eswa.2019.01.061.

[3] X. Xi, J. Wei, Y. Guo, and W. Duan, “Academic collaborations: a recommender framework spanning research interests and network topology,” Scientometrics, vol. 127, no. 11, pp. 6787–6808, Oct. 2022, doi: https://doi.org/10.1007/s11192-022-04555-8.

[4] X. Zhou, W. Liang, K. I-Kai. Wang, R. Huang, and Q. Jin, “Academic Influence Aware and Multidimensional Network Analysis for Research Collaboration Navigation Based on Scholarly Big Data,” IEEE Transactions on Emerging Topics in Computing, vol. 9, no. 1, pp. 246–257, Jan. 2021, doi: https://doi.org/10.1109/tetc.2018.2860051.

[5] C. Coronel and S. Morris, Database systems : design, implementation, and management, 13th ed. Australia ; United States: Cengage Learning, 2019.

[6] L. A. Adamic and E. Adar, “Friends and neighbors on the Web,” Social Networks, vol. 25, no. 3, pp. 211–230, Jul. 2003, doi: https://doi.org/10.1016/s0378-8733(03)00009-1.

[7] A. L. Barabási, H. Jeong, Z. Néda, E. Ravasz, A. Schubert, and T. Vicsek, “Evolution of the social network of scientific collaborations,” Physica A: Statistical Mechanics and its Applications, vol. 311, no. 3–4, pp. 590–614, Aug. 2002, doi: https://doi.org/10.1016/s0378-4371(02)00736-7.

[8] L. Katz, “A new status index derived from sociometric analysis,” Psychometrika, vol. 18, no. 1, pp. 39–43, Mar. 1953, doi: https://doi.org/10.1007/bf02289026.

[9] M. E. J. Newman, “Clustering and preferential attachment in growing networks,” Physical Review E, vol. 64, no. 2, Jul. 2001, doi: https://doi.org/10.1103/physreve.64.025102.

[10] “Flask-SQLAlchemy — Flask-SQLAlchemy Documentation (3.1.x),” flask-sqlalchemy.palletsprojects.com. https://flask-sqlalchemy.palletsprojects.com/en/3.1.x/

[11] NetworkX, “NetworkX — NetworkX documentation,” networkx.org. https://networkx.org/

[12] L. C. Freeman, “Centrality in social networks conceptual clarification,” Social Networks, vol. 1, no. 3, pp. 215–239, Jan. 1978, doi: https://doi.org/10.1016/0378-8733(78)90021-7.

[13] M. E. J. Newman, “The structure of scientific collaboration networks,” Proceedings of the National Academy of Sciences, vol. 98, no. 2, pp. 404–409, Jan. 2001, doi: https://doi.org/10.1073/pnas.98.2.404.

[14] Y. Xu, D. Zhou, and J. Ma, “Scholar-friend recommendation in online academic communities: An approach based on heterogeneous network,” Decision Support Systems, vol. 119, pp. 1–13, Apr. 2019, doi: https://doi.org/10.1016/j.dss.2019.01.004.

[15] C. Wang, W. Tang, B. Sun, J. Fang, and Y. Wang, “Review on community detection algorithms in social networks,” IEEE Xplore, Dec. 01, 2015, doi: https://doi.org/10.1109/PIC.2015.7489908.

[16] M. E. J. Newman and M. Girvan, “Finding and evaluating community structure in networks,” Physical Review E, vol. 69, no. 2, Feb. 2004, doi: https://doi.org/10.1103/physreve.69.026113.

[17] M. Huang, G. Zou, B. Zhang, Y. Liu, Y. Gu, and K. Jiang, “Overlapping community detection in heterogeneous social networks via the user model,” Information Sciences, vol. 432, pp. 164–184, Mar. 2018, doi: https://doi.org/10.1016/j.ins.2017.11.055.