Mining Structural Hole Spanners in Online Social Networks via Supervised Learning

Zhiwei Xu
Dept. of Computer Engineering and Science
Shanghai University
Shanghai, China
zhiweixu@shu.edu.cn

Pinxue Guo
Dept. of Computer Engineering and Science
Shanghai University
Shanghai, China
sunchaser0@shu.edu.cn@shu.edu.cn

Jingyi Pan
Dept. of Communication and Information Engineering
Shanghai University
Shanghai, China
isabelle98@shu.edu.cn

Guobing Zou
Dept. of Computer Engineering and Science
Shanghai University
Shanghai, China
gbzou@shu.edu.cn

***Abstract*—The theory of structural hole suggests that the people who act as intermediaries or bridges between otherwise disconnected communities will get benefits. Mining structural hole spanners has become a hot research issue in recent years. Existing solutions for identifying structural hole spanners normally require exploring the entire graph, which leads to a severe cold start problem. In this paper, we propose a novel machine learning-based approach to identify the structural hole spanners with the consideration of multiple characteristics, including user's basic, node and UGC (User Generated Contents) features. Extensive experiments conducted on a real-world dataset have demonstrated the effectiveness of our approach. The results show that our model can achieve better performance in terms of multiple evaluation metrics.**

Keywords-Online Social Networks; Structural Hole Spanners; Supervised Learning

I. Introduction

Complex system, such as information, biological and social systems, can often be described in terms of complex networks that have a topology of interconnected nodes. Online social networks (OSNs) are one of the most typical networks among them, in which vertices represent users and edges represent the relationships between them. Thus, there are high demands for developing efficient approaches to detect unique properties from online social networks.

Most social networks consist of so-called community structure property, namely, the vertices can be partitioned into various close-connected groups, called as communities [1], where vertices in the same community often share similar features and densely connect with each other. Communities play a significant role in information diffusion, that is, information travels very fast within the same community and spread widely to other communities through the bridges or intermediaries. In sociology, there are a few well-established theories on how position in social networks benefits when people occupy them. One theory is that people who situated between otherwise disconnected communities possess advantageous position. This position is called structural hole (SH) in terminology [2].

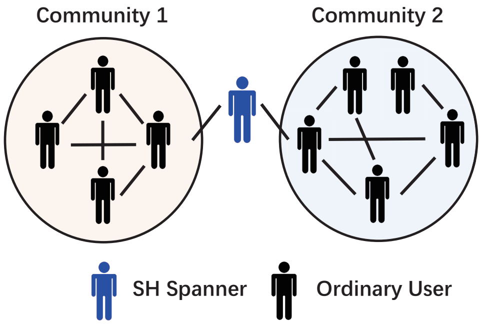


Figure 1. Structural hole spanner in online social networks

As shown in Figure 1, the people locating in the middle of two communities is known as SH spanner, who span the structural hole and will benefit from its position. On one hand, SH spanners serve as bridges or intermediaries and thus can access information more easily than ordinary people [3]; on the other hand, they have the ability to control the information flow and acquire potential resources from it. A number of researches [4, 5, 6, 7] have proven the strong correlation between SH spanners and social success and the important role played in information dissemination. As a result, how to effectively and efficiently detect structural hole spanners has become a hot research issue in many real-world application scenarios.

Recently, a number of researches [8, 9, 10, 11, 12] have been done to identify SH spanners in OSNs. However, all of their approaches are graph-based algorithms, and it potentially leads to many serious problems. First, all of the mentioned algorithms exist cold start problem, which results in huge time consumption for every detection. Second, in some OSNs like Facebook, users are allowed to hide their social connections, making third-party application providers hard to obtain the entire social graph. Last but not least, for newly registered OSN users, their social ties are still under development so they cannot provide complete information. Therefore, a more general and practical algorithm is needed to solve the above problems.

In this paper, we propose an integrated model, by combining the user's profile and UGC, to identify the SH spanners. It is machine learning-based and can be more convenient and efficient while fulfilling the task. Though much work has been performed, to the best of our knowledge, the issues we focus on in this work have not been sufficiently investigated. Our model addresses all these problems holistically. The contributions of the paper can be summarized as below.

- We formulate and model the machine learning-based problem, and propose an integrated model combining multiple heterogeneous features to improve performance on mining SH spanners in online social networks;
- Several classifiers and strategies are compared to furnish third-party service providers with useful information;
- Extensive experiments based on a real-world dataset are carried out to validate the effectiveness and efficiency of our approach.

The rest of the paper is organized as follows. Section 2 provides a motivating example for this research. Section 3 introduces our proposed approach. Section 4 describes the observations of our dataset and the experimental results. Section 5 presents related work. Finally, section 6 concludes and discusses the future work.

II. Motivating Example

Taking the situation shown in Figure 1 as an example, let us consider a simple scenario. A small online social network with two communities has been built as time goes on. Each user in the community is densely connected and almost know any other person in the same community.

Assuming at this moment, a new user, who is blue in figure, signs up, logs in and joins this small social network that already existed. It will result in an unpredictable update of the social ties of graph and thus a new detection for special users is needed to conduct again to obtain a new result.

However, the classical graph-based algorithms always require the knowledge of the entire graph. As a consequence, they have no other choices but to rerun all the processes, which brings out a severe cold start problem and puts a heavy burden on the computation resources.

The scale of the OSNs in the real-world scenarios can of course be significantly larger than this example, let alone the rapid growth of users. Hence, there are high demands for researchers to find a new solution to solve the above problems effectively.

III. Approach

A. Framework

In this study, we aim at proposing a machine learning-based model to mining SH spanners in OSNs. Note that the proposed hybrid approach with little modification can be applicable to any similar scenarios.

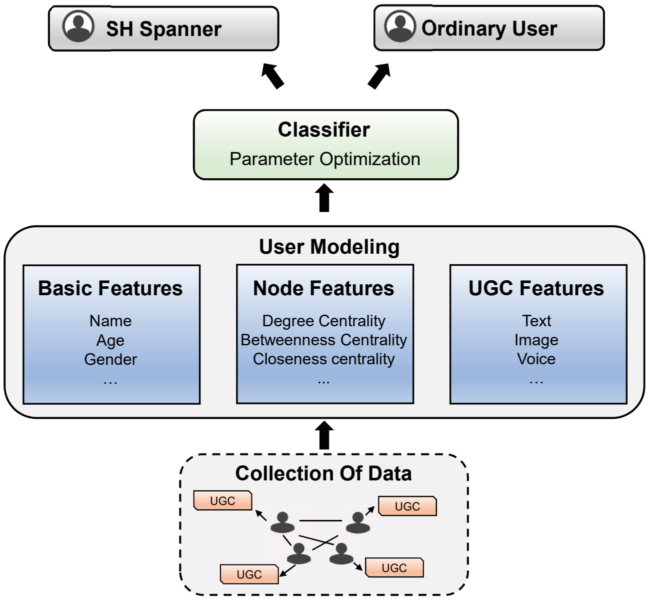


Figure 2. The overall framework of mining structural hole spanners in online social networks.

Figure 2 shows the overall framework of our integrated model. To quantify the user's characteristics, the process of user modeling is conducted first to obtain user's individual features. It consists of three independent parts, namely, basic features, node features and UGC features. The classifier takes the user's three types of features as input, and outputs whether this user is a SH spanner or not. Finally, parameters tuning is then applied to achieve better performance of our model. It is thus defined as a supervised learning problem to identify top-k SH spanners supposing that ground truth has already given.

B. User Modeling

User modeling plays a significant role in our integrated model. User modeling is a kind of portray of users, which shows that how user will behavior and act in our model. In other words, the quality of user modeling is the key to distinguish the special users and ordinary users and has a direct influence on the result.

In our framework, user's features are quantified and represented as a vector. For example, supposing that we have a 30-year-old male user who is socially active and have made the acquaintance of many other people. Moreover, he often writes some positive words to develop a sunny self-image. As a result, if we feed this user's characteristics into our user modeling process, and it will output a multidimensional vector to represent this user. In this way, a further discussions and operations can be carried out to draw useful conclusions.

Therefore, it is very easy to see that user modeling is a combination of diverse user's characteristics, such as name, age, gender, behaviors, sentiment, and so on. In order to cover the most prevalent and related modalities, we incorporate three forms of data and can be summarized as:

- Basic Features
- Node Features
- UGC (User Generated Contents) Features

In the following sections, we will introduce how do our approach model users in detail. Note that for the interested reader who intend to apply our approaches for real-world applications, a little modification may be required according to the actual occasion.

Basic Features. Basic features are user's fundamental descriptive characteristics such as name, age, gender, nationality, personality, behaviors, etc. Not only it consists of the user's personal information, but also it contains the user's behaviors. In our study, daily routine, eating habits, mobility patterns, political behaviors and so on are all considered as one of the parts of user's basic features.

User's basic features helps a lot in the task of identifying SH spanners. For example, children under 3 years old are definitely not SH spanners due to their age. The social ties of them are still under development and they even don't have the ability to interact with strangers and unknown groups.

Furthermore, in order to obtain a more general features of users and avoid bias in our model, the features are needed to normalized accordingly. Namely, the sum of vector is equal to 1 for any users in online social networks.

Node Features. Node features are one of the most crucial features because to some extent it is highly related to the graph-based algorithm. It focuses on the social networks and mainly reflects the special properties of nodes in graph. For example, it illustrates how many users a user connected to, how much attention a user gets in his or her groups, how much information a user control in the process of information diffusion, and so on.

Therefore, to make deeper insights into user's node features and achieve the goal of covering most of the special properties of users in OSNs, we have selected 6 typical metrics in the field of graph computing. Figure 3 demonstrates the framework of node features, which is comprised of degree centrality, betweenness centrality, closeness centrality, eigenvector centrality, farness centrality and PageRank. The first 5 metrics are so-called centrality, which is the quantification of node's importance, and the last is PageRank [3], which is a kind of random walk algorithm and has been widely used on a lot of issues, such as page ranking, influential user identification, etc. In our model, they are all normalized to eliminate bias accordingly. The following contents explains how we calculate the 6 metrics specifically.

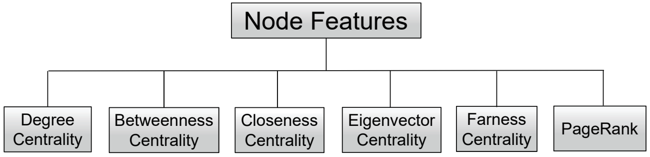


Figure 3. Illustration of node features and its compositions.

Degree centrality focus on the individual value of nodes, which convey how many people a user interacts with in OSNs. The degree centrality of node v is described as

$$\mathrm{DC}(v)=(\mathrm{Degree}(v))/(V\text{-}1) \tag{1}$$

Where V represents the number of nodes.

Betweenness centrality illustrates to what extent a user tends to be an intermediary or broker. It somewhat reflects a user's ability to control the information dissemination. The betweenness centrality of node v is given by

$$\mathrm{BC}(v)=|\mathrm{SP}(v)|/(V\text{-}1)(V\text{-}2) \tag{2}$$

Where SP(v, u) denotes the shortest path between the other nodes where v lies in.

Closeness centrality demonstrates the group value of nodes. For example, if a user has a larger closeness centrality, he or she is more likely to be at the center of the OSNs and can reach the other users as soon as possible. The closeness centrality of node v is calculated as

$$\mathrm{CC}(v)=(V\text{-}1)/(\sum \mathrm{SP}(v,u)) \tag{3}$$

Eigenvector centrality holds that if a user's friends are influential users, and then he or she is also an influential user. It concentrates on the impact brought by the neighbors. The eigenvector centrality of node v is computed as

$$\mathrm{DC}(v)=(\mathrm{Degree}(v))/(V\text{-}1)$$

$$\mathrm{DC}(v)=(\mathrm{Degree}(v))/(V\text{-}1) \tag{4}$$

Where c is a proportional coefficient and a_uv is a weight coefficient. EC(v) needs multiple iterations until the steady state is reached. Hence, eigenvector centrality is a variant of PageRank to some extent.

Farness centrality is reciprocal of closeness centrality. This indicator is selected in our work for the purpose of visualization and analysis. It is defined as

$$\mathrm{FC}(v)=1/\mathrm{BC}(v) \tag{5}$$

PageRank, however, is different from the above indicators. It characterizes the importance of nodes and the relevance of nodes to other nodes. The idea of random walk was applied to this algorithm and this method thus can capture network features compared to others. It has two basic assumptions [3]: first, the more influential a node is, the more nodes are linked with this node; second, the nodes with higher influence will convey a larger weight coefficient.

Hence, through the six metrics mentioned above, we could quantify and evaluate the user's node features effectively.

UGC Features. In our study, user's features not only include the user's basic attributes, but also contain the user generated contents (UGC). UGC are inevitable products of user's behaviors in their daily life, and can express the user's information in an indirect way. Basically, the form of UGC is texts, images and voices, etc. Therefore, regarding the UGC as one of the user's features is crucial for the identification task.

However, on account of the sake of our dataset and the situation that UGC are always in the form of texts – voices can be converted into texts with speech recognition, we only describe the way to deal with text in detail. For the interested readers who wants to deal with images, we suggest that bag-of-visual-words may be one of the good choices.

In order to extract and quantify the textual features properly, we apply LDA [13] on the texts. LDA has been proven to be a good tool to extract implicit text topic features [14, 15]. Furthermore, [16] shows its high performance in the identification of influential users. In LDA terminology, each user is modelled as LDA document, while each word in documents is considered as LDA word. In addition, the numbers of topics of LDA are supposed to be given at first. Hence, the procedure of textual features extraction is as follows:

- For every user, combine all of his or her generated texts into one document;
- Eliminate the stop words, pronunciations and special words and then apply the word segmentation to each document;
- Utilize bag of words model to represent each document;
- Preset the number of topics and then feed every document into LDA to train the model.

After LDA is well-trained, we can obtain both the topic distribution of each document and the word distribution of each topic accordingly. Figure 4 shows an example of 5 LDA topics, the textual features of both "User1" and "User2" can be represent by 5-dimensional vector. Moreover, the sum of each vector is equal to 1. It can be seen that "User1" prefer to use words of topic 1, 3 and 5 while "User2" likes to use words of topic 3. As a result, disparate sentiment in language habits has been portrayed obviously and easily based on LDA topics. In summary, the UGC features of each user can be quantified and represented properly by our approaches.

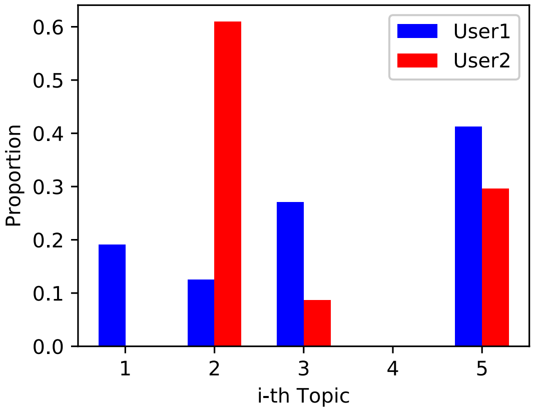


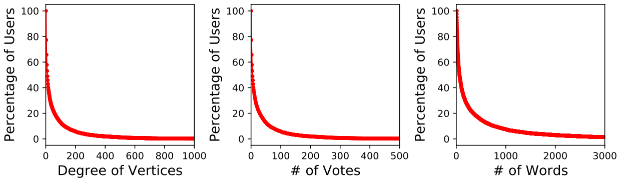
Figure 4. Example of LDA topic distribution.

IV. Experiments

We conducted extensive experiments to validate the effectiveness and efficiency of the proposed integrated model for mining structural hole spanners in online social networks. The experiments were carried out on a platform with 2-core 3.1 GHz Intel Core i5 processors and macOS Mojave 10.14.6 operating system.

A. Dataset and Observations

The Stanford Large Network Dataset Collection [17] website hosts a wide range of datasets, including Tweets, social networks, product reviews, road networks and so on, resulting in a plenty of meaningful work and valuable literature done nowadays. Our dataset was also obtained from this website named wiki-RfA (Wikipedia Requests for Adminship), i.e. for a Wikipedia editor to become an administrator, a request for adminship (RfA) must be submitted, either by the candidate himself or by another community member. Subsequently, members of Wikipedia can vote for suppose, neutrality and negative with a few sentences to explain the reason. These behaviors were transformed into relationships between nodes.



(a) (b) (c)

Figure 5. The statistics of wiki-RfA dataset. (a) illustrates the CDF of the degree of vertices; (b) illustrates the CDF of # of words; (c) illustrates the CDF of # of votes.

The dataset contains 10, 835 users and a total of 158, 388 edges connected with them, and was crawled and collected in 2003 through May 2013. In our experiment, 26.95% of users are eliminated to get rid of incomplete user features, for example, some users only wrote 10 or less words and thus couldn't express the emotional semantic information. Figure 5 demonstrates the CDF (Cumulative Distribution Function) of degree of vertices, number of words and number of votes. It shows that all of them are scale-free and have a power-law or exponential form – most of the users interact with less than 300 the other users, write less than 1000 words and vote less than 200 times respectively [18].

Ground Truth. Ground truth is a vital component for our work, which severely determines what kinds of people we are interested in and what will we do later. To obtain a more general and more representative ground truth of SH spanners, HIS [8] is chosen to perform this job, since it is one of the typical algorithms to identify the SH spanners and has been widely recognized. We further apply Louvain [19], an algorithm to maximize the Q-Modularity of a partition quickly, to unfold communities of OSNs initially due to it is the preliminary of HIS. Then, the most significant 4 communities (96.59% of users) were feed into HIS as the

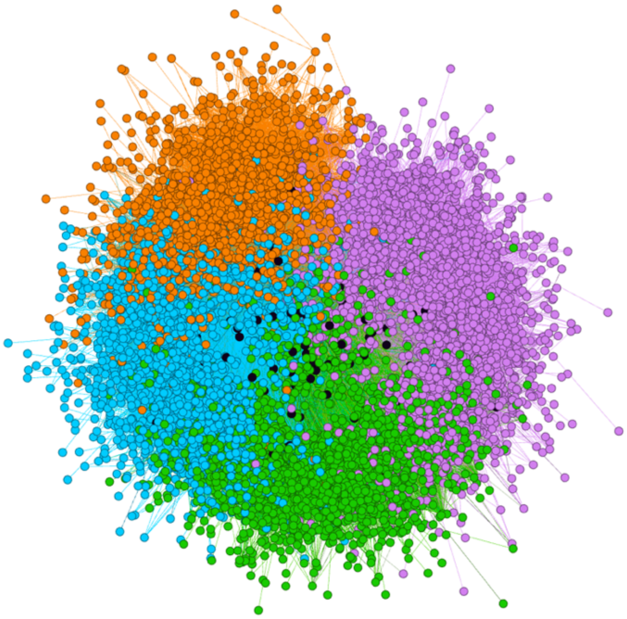
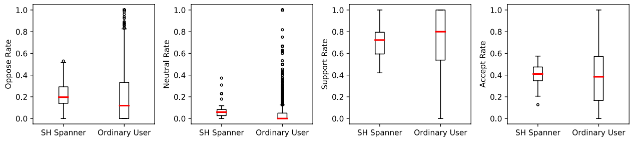
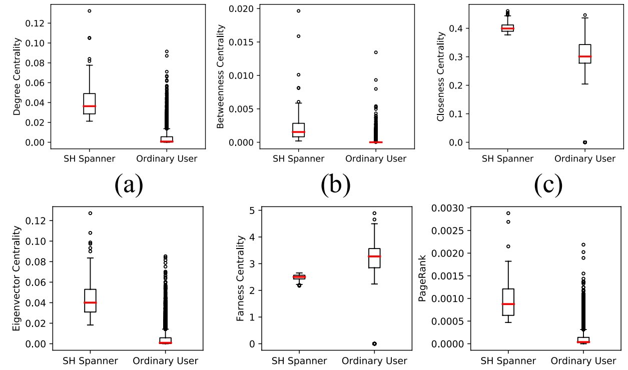


Figure 6. Visualization of the social network: Orange, purple, blue, green nodes represent the users in 4 communities respectively; black nodes represent the SH spanners found by HIS.



(a) (b) (c) (d)

Figure 7. The statistics of users' voting behaviors: (a), (b) and (c) shows the difference of voting habit between SH spanner and ordinary users; (d) shows the results after vote.



(a) (b) (c)

(d) (e) (f)

Figure 8. The statistics of users' node features: (a), (b), (c), (d), (e) and (f) illustrate the degree centrality, betweenness centrality, closeness centrality, eigenvector centrality, farness centrality and PageRank of SH spanners and ordinary users respectively.

key communities and top-79 (1% of users) users are identified as SH spanners.

Figure 6 explicitly reflects that four main communities of users and the SH spanners found by HIS. It is interesting but not surprising to notice that the SH spanners (black users) are almost located at the junction of the different communities, which is highly compatible with the functionality of structural hole – SH spanners bridge otherwise disconnected communities.

Basic Features. In our work, user's voting behavior is selected to reflect user's basic features, which including what percentage of a user will vote for suppose, neutrality or oppose and what percentage of a user will be accepted after vote. For example, for a Wikipedia user who tends to become an administrator, he or she or the other community member will submit a request. Subsequently, any Wikipedia member may cast a supporting, neutral, or opposing vote, and then this user will obtain adminship or not up to vote. Hence, after this process is carried out many times, each user's voting behavior can be calculated.

Figure 7 illustrates the difference of voting habit between SH spanner and ordinary users. We see that because the SH spanners as a kind of influential users generally have a more complete knowledge system and more rigorous discriminant thinking, they slightly do not tent to vote in favor and have a higher accept rate than ordinary users. In other words, compared to the ordinary users, SH spanners are more likely to be the center of power and thus affect the results.

Node Features. The metrics of each users are calculated and represent their node features. As shown in Figure 8, we also analyzed the difference between the SH spanners and ordinary users in these six indicators.

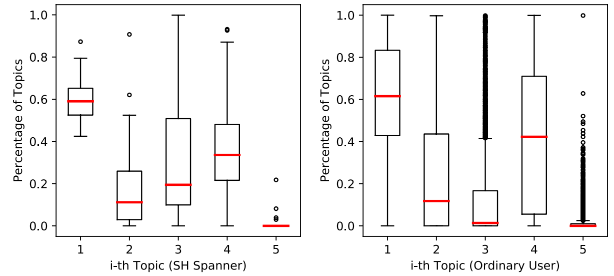
Figure 8(a) presents the degree centrality of users. It demonstrates that the SH spanners prefer to interact and communicate with more people than the ordinary users, namely, they are more socially active and willing to get acquainted with more people. Figure 8(b) shows the betweenness centrality of users. It can be seen that SH spanners are more likely lies in the shortest path of two other users, so they will have more opportunities to serve as intermediaries or brokers of them and benefits from this process. Figure 8(c) and 8(e) respectively shows the closeness centrality and farness centrality of users. It explicitly reflects SH spanners are inclined to locate at the center of the OSNs. Therefore, they possess the high attention and exposure and can spread their own ideas and thinking quickly. Figure 8(d) visualizes the eigenvector centrality of users, it can be explained that not only do the SH spanners have strong social influence, but also the users who they connected to also have great social ability. Figure 8(f), in contrast, shows the PageRank value of each users. It can be seen that structural hole users are either linked to a large number of others, or linked to some high-influential ones. In summary, user's node features are tightly knit to users' identity, such as SH spanners or not.

UGC Features. In this paper, we regard user textual contents as a form of UGC features. Since our dataset is generated in the process of adminship in Wikipedia, the comments wrote by the users are almost highly related to evaluation of the others.

We empirically found the number of the LDA topics equals to 5 will gives more human-interpretable category-topic distribution results. The model has been trained 50 passes to acquire each users' topic distribution. We then

TABLE I. WORD DISTRIBUTION AMONG LDA TOPICS

ID	Words	Topics
T1	Deserve, Helpful, Definitely, Dedicated, Nice	Positive
T2	Weak, Vandalism, Low	Negative
T3	There, Someone, Others, That, Something	Pronoun
T4	Mdashvery, Deo, Nbspnbsp, Kosovo, Slg	Abnormal Words
T5	Understanding, Qualified, Knowledge, Experienced, Trustworthy	Expert



(a) (b)

Figure 9. The statistics of users' textual features: (a) and (b) illustrate the LDA topic distribution of SH spanners and ordinary users respectively.

label each topic, as shown in Table 1, by considering the corresponding popular words for the purpose of analysis. As a result, we can interestingly define 5 groups, i.e. "Positive", "Negative", "Pronoun", "Abnormal Words" and "Expert".

Figure 9 demonstrates the statistics of the LDA distribution of the SH spanners and the ordinary users. As it can be seen, there are only slightly different on topic 1, 2, 4 and 5. Nevertheless, the SH spanners prefer to use pronoun compared to the ordinary users, that is, they are more likely to write the words like "There", "That", "This", "Someone", "Something", "Others" and so on. To conclude, there are still difference between SH spanners and ordinary users in terms of the language habits, and it help us distinguish them powerfully.

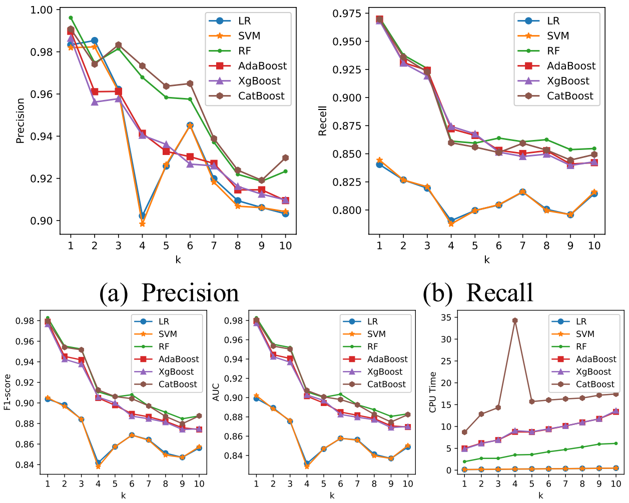
B. Experimental Settings

Basic features, in our work, includes the user's voting behavior. It has been normalized to obtain more balanced representation of user's features. Node features, however, is a kind of features that can reflects the information in social network, which contains degree centrality, betweenness centrality, closeness centrality, eigenvector centrality, farness centrality and PageRank. We regard user's text contents as UGC features and further utilize LDA to quantify into a feature vector. Note that the data used to describe user's feature can be changed based on the actual situation and our model can work without some of features but including them will improve the performance.

In order to get a more convincing and general result, we define k as a sequence of 1% to 10% to regard top-k% of users as SH spanners, and then we randomly select k% of users from the rest of users to ensure the balance of the dataset. Hence, the proportion between the numbers of SH spanners and ordinary users in the training subset and test subset is 1:1. 5-fold cross-validation are conducted 30 times and we take the average as the final results. Five classic metrics are selected to evaluate the results, which is precision, recall, F1-score, AUC [20] and elapsed CPU time. Precision denotes the fraction of identified SH spanners who are really SH spanners. Recall is the fraction of SH spanners who are correctly identified. F1-score is the harmonic mean of precision and recall. AUC (area under the ROC curve) means the probability that this model would rank a randomly selected SH spanner higher than a randomly chosen ordinary user. CPU time reflects how much time spent training the model. Six classical classifiers, Logistic Regression, Support Vector Machine, Random Forest, AdaBoost, XgBoost [21] and CatBoost [22], are selected to train the model. We employed both the early late fusion approach to obtain the results.

C. Experimental Evaluations

Table 2 shows the performance of classifiers under different user's features and ensemble approaches. According to the table, we can intuitively draw five conclusions. First, as one of the ensemble learning classifiers, Random Forest and CatBoost always achieve the best performance whatever the first 4 metrics are. However, considering the elapsed CPU time when training the model, Random Forest seems to be a better choice for the actual situation. Second, multi-source of features will often result in better performance in classification task compared to single source of features, and it achieves the best when using all of them. Third, the late fusion approach leads to great improvement of some classifiers like Logistic Regression and Support Vector Machine while doesn't improve the performance of the ensemble learning algorithms. Fourth, node features play an important role in the identification task because the ground truth is obtained by the graph-based algorithms. Nevertheless, although our dataset contains quiet limited basic features and UGC features, excluding the node features can still achieve a high F1-score of 0.756 and AUC of 0.758, which proves it practicability on real-world application systems. Fifth, for third-party service providers, late fusion SVM is a good choice when pursuing high precision while early fusion Random Forest seems to be better when focusing on recall. It can be decided according to the situation.



(a) Precision (b) Recall

(c) F1-score (d) AUC (e) CPU Time

Figure 10. Illustration of performance of classifiers with respect to top-k% of SH spanners using all the features: (a), (b), (c), (d) and (e) respectively illustrate the precision, recall, F1-score, AUC and CPU time for different classifiers and parameter k.

Figure 10 shows the performance of different classifiers. It can be seen that all aspects of metrics drop down when parameter k increases. The reason is obviously that the classifiers need to identify more users and it leads to higher requirements. (a), (b), (c) and (d) demonstrates that ensemble learning algorithms outperform traditional methods in precision and recall in general. Moreover, when taking all of the aspects of metrics into consideration, Random Forest stands out due to its excellent performance and short training time. Therefore, the trained model could be qualified to predict whether a newly registered user will become a future SH spanner. The above results validate that our model can achieve a good performance to identify the SH spanner without analyzing the entire social networks every time. It proves our model's practicability on the task of predicting whether a newly registered user will become a future SH spanner.

V. RELATED WORK

The concept of structural hole is first introduced in [2] and further elaborated in [4, 23, 24]. After that, it has been extensive researches on mining structural holes from social networks. The identification of SH spanners was initially formulated and studied in [8], as a few people who fill the structural holes can bridge the different communities. It proposed two algorithms HIS and MaxD leveraging the information diffusion and minimal cut respectively, by assuming that communities in social networks are already given. Hence, the quality of SH spanners relies on the chosen communities heavily. [10] focused on this problem and came up with several fast heuristics. [9] defined a SH spanner as one whose removal would result in the greatest increase in the pairwise distance among the remaining users. [11] considered SH spanners as the vertices who lie on a plenty of shortest path, which is quite similar to the betweenness centrality. Since the calculation needs great time consumption. [12] recently developed innovative filtering techniques to filter out unlikely solution as early as possible and achieve a high computation performance, but it encountered the scalability problem caused by large social networks. However, all of these literatures concentrated on graph-based algorithms and ignored the potential issues caused by them. To the best of knowledge, this is the first work to systematically study the problem of mining structural hole spanners in online social networks using machine learning.

VI. CONCLUSIONS AND FUTURE WORK

In this paper, we develop an integrated model to identify the SH spanners in OSNs without checking the entire social graph. Three types of features are combined to feed into classifiers and then the model can predict whether a newly registered user will be the SH spanner in the future. The real-world dataset is exploited to conduct extensive experiments and it validates the effectiveness and efficiency in multiple evaluation metrics. It is potentially applied for third-party service providers to find SH spanners. In the future work, a large-scale multi-source dataset will be considered to verify the quality of SH spanner detection in our scheme. Furthermore, we plan to do more in-depth researches on classifiers to enhance its performance.

ACKNOWLEDGMENT

We would like to appreciate all of the anonymous reviewers for their constructive suggestions and insightful comments that helped us significantly improve the quality of the paper.

REFERENCES

[1] N. Girvan, and M. E. Newman. Community Structure in Social and Biological Networks. In Proceedings of National Academy Sciences of the United States, 99:7821–7826, 2002.

[2] R.S. Burt. Structural Holes: The Social Structure of Competition. Harvard University Press, 1992.

[3] L. Page, S. Brin, R. Motwani, and T. Winograd. The PageRank Citation Ranking: Bringing Order to the Web. Stanford InfoLab, 1999.

[4] R. S. Burt. Structural Holes and Good Ideas. American Journal of Sociology, 110:349–399, 2004

[5] R. S. Burt. Secondhand brokerage: Evidence on the importance of local structure for managers, bankers, and analysts. Academy of Management Journal, 50:119–148, 2007.

[6] J. M. Podolny and J. N. Baron. Resources and relationships: Social networks and mobility in the workplace. American Sociological Review, 62(5):673, 1997.

[7] K. Stovel and L. Shaw. Brokerage. Annual Review of Sociology, 38(1):139–158, 2012.

[8] T. Lou and J. Tang. Mining structural hole spanners through information diffusion in social networks. In Proceedings of the International Conference on World Wide Web, 2013.

[9] C. Song, W. Hsu, and M. Lee. Mining Brokers in Dynamic Social Networks. In Proceedings of the 24th ACM International on Conference on Information and Knowledge Management, 2015.

[10] M. Rezvani, W. Liang, W. Xu, and C. Liu. Identifying top-k structural hole spanners in large-scale social networks". In

Proceedings of the 24th ACM International on Conference on Information and Knowledge Management, 2015.

[11] S. Goyal and F. Vega-Redondo, Structural holes in social networks. Journal of Economic Theory, 137(1):460–492, 2007.

[12] W. Xu, M. Rezvani, W. Liang, J. X. Yu and C. Liu. Efficient Algorithms for the Identification of Top-k Structural Hole Spanners in Large Social Networks. In IEEE Transactions on Knowledge and Data Engineering. 29(5):1017-1030, 2017.

[13] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent Dirichlet allocation. Journal of Machine Learning Research, 2003.

[14] Y. Qu and J. Zhang. Trade area analysis using user generated mobile location data. In Proceedings of the International Conference on World Wide Web, 2013.

[15] A. Farseev, L. Nie, M. Akbari, T. Chua. Harvesting Multiple Sources for User Profile Learning: a Big Data Study. In Proceedings of the 5th ACM on International Conference on Multimedia Retrieval. 23-26, 2015.

[16] J. Leskovec and A. Krevl. SNAP Datasets: Stanford large network dataset collection. http://snap.stanford.edu/data. 2014.

[17] N. Girvan, and M. E. Newman. Community Structure in Social and Biological Networks. In Proceedings of National Academy Sciences of the United States, 99:7821–7826, 2002.

[18] A. Mislove, M. Marcon, K.P. Gummadi, P. Druschel, and B. Bhattacharjee. Measurement and analysis of online social networks. In Proceedings of the 7th ACMSIGCOMM Conference on Internet Measurement. 29–42, 2007.

[19] V. D. Blondel, J. Guillaume, R. Lambiotte, and E. Lefebvre. 2008. Fast unfolding of communities in large networks. Journal of Statistical Mechanics 10008, 2008.

[20] T. Fawcett. An introduction to ROC analysis. Pattern Recognition Letters, 27(8):861–874, 2006.

[21] T. Chen, C. Guestrin. XGBoost: A Scalable Tree Boosting System. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 2016.

[22] L. O. Prokhorenkova, G. Gusev, A. Vorobev, A. V. Dorogush, and A. Gulin. CatBoost: unbiased boosting with categorical features. In Proceedings of NeurIPS. 6639–6649, 2018.

[23] G. Ahuja. Collaboration networks, structural holes, and innovation: A longitudinal study. Administrative Science Quarterly, 45(3):425–455, 2000.

[24] R. S. Burt. Secondhand brokerage: Evidence on the importance of local structure for managers, bankers, and analysts. Academy of Management Journal, 50:119–148, 2007.

TABLE II. PERFORMANCE OF CLASSIFIERS IN IDENTIFICATION OF SH SPANNERS. LR – LOGISTIC REGRESSION; SVM – SUPPORT VECTOR MACHINE; RF – RANDOM FOREST; AB – ADABOOST; XB – XGBOOST; CB - CATBOOST

Classifier	Precision	Recall	F1-score	AUC	CPU Time
LR (Basic)	0.614	0.523	0.561	0.528	0.499
SVM (Basic)	0.613	0.520	0.558	0.524	0.345
RF (Basic)	0.696	**0.711**	0.703	0.706	5.794
AB (Basic)	0.724	0.708	0.716	0.713	8.141
XB (Basic)	0.723	0.710	0.716	0.714	8.146
CB (Basic)	**0.734**	0.708	**0.720**	**0.716**	16.900
LR (Node)	0.914	0.792	0.848	0.837	0.396
SVM (Node)	0.914	0.792	0.848	0.837	0.377
RF (Node)	0.903	**0.831**	0.865	0.860	5.367
AB (Node)	0.925	0.812	0.864	0.856	9.913
XB (Node)	0.926	0.812	0.865	0.855	8.766
CB (Node)	**0.928**	0.823	**0.872**	**0.864**	11.346
LR (UGC)	0.604	0.613	0.607	0.611	0.333
SVM (UGC)	0.604	0.612	0.607	0.610	0.324
RF (UGC)	0.702	0.664	0.682	0.674	4.992
AB (UGC)	0.732	0.694	0.712	0.704	7.848
XB (UGC)	**0.732**	0.693	0.711	0.703	7.871
CB (UGC)	0.719	**0.707**	**0.712**	**0.710**	10.809
RF (Basic + Node)	0.911	**0.826**	0.866	0.859	5.601
AB (Basic + Node)	0.917	0.811	0.861	0.852	10.605
XB (Basic + Node)	0.917	0.810	0.859	0.850	10.606
CB (Basic + Node)	**0.931**	0.817	**0.870**	**0.861**	14.767
RF (Basic + UGC)	0.747	0.734	0.740	0.738	6.061
AB (Basic + UGC)	0.761	0.720	0.739	0.732	9.609
XB (Basic + UGC)	0.760	0.722	0.740	0.733	9.612

CB (Basic + UGC)	**0.771**	**0.743**	**0.756**	**0.758**	14.439
RF (Node + UGC)	0.916	**0.830**	0.870	0.864	5.622
AB (Node + UGC)	0.915	0.808	0.864	0.856	10.968
XB (Node + UGC)	0.917	0.819	0.865	0.858	10.969
CB (Node + UGC)	**0.930**	0.823	**0.873**	**0.865**	14.646
RF (All + Early Fusion)	0.923	**0.855**	**0.887**	**0.883**	6.124
AB (All + Early Fusion)	0.909	0.842	0.874	0.869	13.341
XB (All + Early Fusion)	0.910	0.843	0.875	0.870	13.625
CB (All + Early Fusion)	**0.930**	0.849	0.887	0.882	17.436
LR (All + Late Fusion)	0.898	0.793	0.842	0.832	1.933
SVM (All + Late Fusion)	**0.938**	0.805	**0.866**	**0.856**	48.158
RF (All + Late Fusion)	0.857	**0.811**	0.832	0.828	9.168