The Recommended Online Social Networks (OSN) and Big Data System Using Large Scale Graph Partitioning with Mahout and PowerGraph

Ford Lumban Gaol (✉ fgaol@binus.edu)

Columbia University <https://orcid.org/0000-0002-5116-5708>

Tokuro Matsuo

Advanced Institute of Industrial Technology

Research

Keywords: Social Big data, Social recommendation, Mahout, PowerGraph, Data analytics

Posted Date: September 4th, 2020

DOI: <https://doi.org/10.21203/rs.3.rs-45888/v1>

License: © ⓘ This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

The Recommended Online Social Networks (OSN) and Big data System using Large Scale Graph Partitioning with Mahout and PowerGraph

Ford Lumban Gaol ^{1*}, and Tokuro Matsuo ²

¹ Computer Science Department, BINUS Graduate Program – Doctor of Computer Science, Bina Nusantara University, Jakarta 11480, Indonesia; fgaol@binus.edu

² Advanced Institute of Industry Technology, Tokyo, Japan, matsuo@aiit.ac.jp

* Correspondence: fgaol@binus.edu

ABSTRACT

Introduction : Social Big data is generated by interactions of connected users on social network. Sharing of opinions and contents amongst users, reviews of users for products, result in Social Big data. If any user intends to select product such as movies, books etc. from e-commerce sites or view any topic or opinion on social networking sites, there are a lot of options and these options result in information overload.

Case Description : Social recommendation systems assist users to make better selection as per their likings. Recent research works have improved recommendation systems by using matrix factorization, social regularization or social trust inference. Furthermore, these improved systems are able to alleviate cold start and sparsity but not efficient for scalability.

Discussion and Evaluation: The main focus of this paper is to improve scalability and provide better recommendations to users with large-scale data in less response time. We have partitioned social big graph and distributed it on different nodes based on Mahout and PowerGraph like system.

Conclusion : In our approach, partitioning is based on direct as well as indirect trust between users and comparison with state-of-the-art approaches proves that statistically better partitioning quality is achieved using our approach. In our proposed approach ScaleRec, hyperedge and transitive closure are used to enhance social trust amongst users. Experiment analysis on standard datasets proves that better locality and recommendation accuracy is achieved by using our proposed approach.

Keywords: Social Big data, Social recommendation, Mahout, PowerGraph, Data analytics

INTRODUCTION

Big data is generated by social media on social networking sites [1]. Recommender systems reduce the large information space generated by Social Big data. It is information filtering tool which provides users suggestions based on their interest. The applications of recommender systems are in various domains such as books, movies or other products recommendations on e-commerce sites, friends recommendations on social networking sites, project recommendation on GitHub etc.

Collaborative filtering, content based and hybrid based are different techniques of recommender systems [2][3][4]. In these techniques, user provides ratings to products which result in user-item matrix. This matrix is important for analyzing user's interest. Sparsity, cold start and scalability are limitations of conventional recommender systems. Sparsity and cold start are addressed by several re-searchers [5][6][7][8]. The main concern for researchers is scalability which needs to be addressed for large-scale data. Traditional recommender systems works well for limited scale of social data. Moreover these algorithms are designed for centralized approach only. If these systems are deployed on large-scale data, throughput is degraded significantly, which results in reduced users' interest in these systems. In this paper, the key motivation is to improve recommendation accuracy even for large numbers of nodes in social graph.

Recommendation systems leverage Big data in the form of large-scale social graph and efficient graph algorithms are important for these systems. Large-scale social graph cannot be processed on centralized system. There is need for distributed approach where sub-graphs can be processed in parallel. Large-scale recommender systems have leveraged distributed algorithms for finding recommendation [9]. Graph partitioning is technique which can address scalability issue. Large-scale graph partitioning in traditional recommendation models uses random walk, Fork-Join [10] or hash partitioning to divide graph into sub-graphs. In our proposed approach ScaleRec, direct and indirect trust-based walk is used to partition graph with relevant nodes only which improves locality. Social graph is partitioned based on social trust amongst nodes to reduce communication between nodes in inter-subgraphs and maximize communication in intra-subgraphs. Improved locality minimizes communication overhead which results in improved scalability [11].

Conventional data analytics technologies based on centralized approach cannot store and process large-scale data. Big data frameworks such as Hadoop [12], MapReduce [13], Pregel[14], GraphLab[15], Mahout [16] and Giraph [17] , PowerGraph [18], GraphX [19], CUDA and GPU are used by many researchers to deal with large-scale data. We have used Giraph and Pregel in our approach, as these can effectively process large-scale social graph. Social graph is distributed on multiple machines with some vertices replication [20]. This is efficiently implemented by using Giraph API.

Users interact with other users on social network and share their likings. On the basis of their interaction, users' trust builds on other user. It is also concluded that users with same interest are connected to each other on social networks [21]. Trust is used as important factor for social recommender systems [22][23][24]. Traditional recommendation systems use random walk which does not consider any

weights assigned to edges [21]. Moreover, the drawback of existing social trust based recommender system is assumption of only direct trust i.e. single edge between users. In our proposed approach, indirect trust i.e. friends-of-friends is also utilized to improve recommendation accuracy.

If recommendations systems are able to predict accurate ratings for user, it will reveal the user's level of likings for product or topic. In our proposed approach, rating is predicted by considering ratings of trusted users only.

$$P(u, e) = \frac{\sum_{v=0}^n T(u, v) * R(v, e)}{T(u, v)} \quad (1)$$

where, Predicted rating P for product e provided by user u is calculated by user trusted v users of u , $T(u, v)$ and ratings of v users for product e , $R(v, e)$. It is clear from Equation 1 that if numbers of trusted users are increased then prediction accuracy is better. Our approach increase number of trusted users which improves ratings prediction.

RELATED WORK

Several approaches are proposed for social recommendation improvement such as matrix factorization [6][8][25], social regularization [26], trust-based[22][23][24], hierarchical clustering[27] etc. In this section, existing research works are categorized into social recommendation, large-scale graph partitioning and scalable social recommendation. These approaches are described as follows.

Social recommendation

In [28], content-based and collaborative filtering are mentioned as recommender systems techniques. Hybrid system can utilize advantages of these techniques and ignore the disadvantages. Fab is proposed in this paper which uses profiles of users provided by content-based and user experience provided by collaborative filtering. Experiment analysis proves that Fab system provide improved performance as compared to existing approaches. In [29], TrustSVD is proposed which exploits trust for improving recommendations. Implicit and explicit influence of ratings and trust is incorporated in recommendation model. Experiments are conducted on four datasets and this approach is compared with ten recommendation models to validate the effectiveness of this approach. In [30], trust inferred algorithms are demonstrated. Trust is calculated for users who are not directly connected. Significance of trust is analyzed by authors and non rounding algorithm is explained for inferring trust. TrustMail, an email prototype is presented in this paper to score email based on ratings of users. [31] proposed framework for developers so that recommendation algorithms can be easily implemented and integrated to use social datasets. Apache Solr is used to index social data after preprocessing. Four algorithms are provided to users – MostPopular(MP), Collaborative Filtering (CF), Content Based (C) and Hybrid Recommendations (CCF).

Large-scale graph partitioning

Pregel is described in [14]. Vertex centric approach is explained with the use of communication between nodes by messages. Scalable and efficient implementation of clusters for large-scale graph is described in

this paper. Abstract API is developed to use large-scale distribution of social graph. In [20], graph is partitioned based on bipartite graph. BiGraph is implemented using PowerGraph in this paper. Authors have explained how communication overhead is minimized by this approach. Empirical results have shown 17.75 times improvement and reduction in 65% vertices replication. Authors have proposed DPM (Distribute-Partition-Merge) in [32]. In this paper, limitations of random walk and Fork-Join based partitioning are highlighted. Merging in FJ based method is centralized. In DPM, sub-merge is also implemented on distributed nodes. Experiment analysis concluded that DPM improves recommendation time as compared to Pregel and FJ. In [33], authors have analyzed the difference between conventional graph and social graph. In this paper, existing graph partitioning algorithm is improved with objective functions. Several graph algorithms exhibit inefficient performance which is improved by using Pregel-like systems in [34]. Minimum spanning forest and graph coloring are improved by using parallel approach. In [35], guided walk is demonstrated which is based on relevance of link in user-item complex network. It is shown to be better than random walk as supervised proceed of link is used instead of any random link. Large-scale data is processed by using Pregel in this research work. In [36], modularity Q value is used to check the relevance of graph partitioning. It is defined as difference of edges which are in same community and average edges on random connections between vertices.

Scalable social recommendation

In [2], it is mentioned that recommendation process can be enhanced by adapting to dynamically changing graph and process large-scale graphs. Data variety, volatility and volume are the major issues which need to be addressed by scalable recommender systems. Hadoop, MapReduce, Mahout, CUDA, GPU and GraphLab are distributed solution for processing high volume data. In [37], scalable algorithms are presented for large-scale data to provide top-N recommendation. Asymmetric cosine and prediction is integrated in algorithm to improve MAP (Mean Average Precision). It is shown in this work that these techniques works well for CF. Social big data application in social networks analysis is presented in [1]. Various new implementations for social big data like Apache Spark and Mahout is demonstrated to reflect the practical implications of social media and social networks. Applications of social networks in context of social big data is elaborated like text analytics, social recommendation etc.

METHODS

Several research works have improved social recommendation by exploiting matrix factorization, social regularization [26] and social trust [22][23][24]. Sparsity and cold-start are addressed efficiently by these novel approaches. The concern for existing recommendation systems is scalability. There is need for improved approach which can overcome scalability issue. In this work, large-scale graph are partitioned into sub-graphs. These sub-graphs work in synchronization by using Pregel and Giraph. Furthermore, graph partitioning is improved by using trust-based instead of random or hash partitioning. Our proposed approach ScaleRec is described with following algorithm.

Input: Users- Items ratings and User- User trust

Output: Recommendation and predicted ratings in less response time

1. Initialize trust matrix and ratings matrix.

2. Calculate direct social trust

Let $MT = \{mt_{ij}\}$ is a social trust matrix of users u_i and u_j . Value of m_{ij} is 1 if there is trust between users otherwise m_{ij} is 0.

$$mt_{ij} = \begin{cases} 1 & \text{if } u_i \xrightarrow{\text{trust}} u_j \\ 0 & \text{otherwise} \end{cases}$$

3. $G = \{V, E\}$ is directed graph which is derived from matrix mt_{ij} . Edge e_{ij} connects Vertex u_i to Vertex u_j .4. Calculate indirect trust values by using proposed approach **if** $u_i \xrightarrow{\text{trust}} u_j$ and $u_j \xrightarrow{\text{trust}} u_k$ it is concluded that $u_i \xrightarrow{\text{trust}} u_k$, Vertex u_i is connected to u_k by edge e_{ik} .

5. Partition modified social trust graph by most influence nodes.

6. Distribute user-user trusts and user-item ratings matrix values on different nodes.

7. Predict ratings for users on distributed nodes

$$R_{i,j} = \frac{\sum_{k=0}^n R_{k,j}}{n}$$

8. Calculate global predicted ratings after combining local predicted ratings.

9. Calculate MAE

$$M_{ae} = \sum_{i=0}^n (P(u, i) - p(u, i)) / n$$

10. Calculate locality

$$\theta = \frac{n_{ep}}{n_e}$$

11. Calculate modified MAE, locality and throughput of proposed technique

12. Compare these metrics with existing approaches

13. Exit

In this section, we have focused on dealing with three major issues in context of large-scale social graph and social recommendation improvement, which are as follows:

- Large-scale social graph analysis with the use of trust and influence
- Partitioning of large-scale social graph on different nodes using our proposed approach
- Improving recommendation accuracy and throughput on sub-graphs and also for complete graph.

We have solved these issues in sequential manner, which is elaborated as follows.

Large scale social graph analysis

Social graph is represented by users u_i which are connected by the set of edges e_i . Trust between users is represented by directed edges. The core motive of social graph mining and analysis is to extract important information like neighbor nodes characteristics and finding out the relevant nodes on which many users have strong trust. We have assumed binary trust values to better incorporate our approach on available social trust datasets. In our proposed approach, trust is improved between users. The motivation for improving trust is to provide better recommendation to users. Figure 1 shows social trust graph of 7 users. User 1 trusts user 2 and it is represented by directed edge. In social trust graph, if user 4 trusts user 5, it is high probability that their likings for product and topic will be same. If user 5 likes any product or topic on social networking sites, recommender system which are based on trust, should conclude that this product

or topic can be suggested to user 4 also.

Figure 1. User-User trust graph

Our approach follows both asymmetric and transitive property of social trust graph. Edges are directed in social graph because if User B is trusted by user A, this is not certain that A is also trusted by B i.e. it follows asymmetric property. If A - B i.e. A trusts user B and B - C i.e. B trusts user C, then by using indirect trust, A - C i.e. A trusts user C i.e. it follows transitive property. Hyperedge i.e. direct edge between A and C will be drawn in improved social trust graph as depicted in Figure 2. Indirect trust along with direct trust increases the number of trusted users which results in improved social recommendation. In improved social graph, threshold is set to 3 i.e. only strongly connected friends-of-friends will be included in improved social trust graph.

Figure 2. Improved social graph by using hyperedge

In Figure 2, it is clearly depicted that user 3 and user 5 are connected through hyperedge i.e. indirect trust which enhances trust values in social graph. User provides ratings to products in the scale of 1-5. In Figure 3, it is clearly demonstrated that ratings is provided by very few users. User 1 and 2 provide ratings to Product 1 only and user 4 provides ratings to product 2 only. This results in sparse user-item matrix where many users do not provide ratings. User 3 and 5 do not provide ratings to any product as they are new in recommender systems hence results in cold start.

Figure 3. User-Item ratings graph

Equation 2 describes matrix M which is filled with trust values between user 'm' with user 'n'. Small improvement in trust values results in better recommendation.

$$M = U^{m \times n} \quad (2)$$

In Equation 3, it is shown that user- user trust matrix is filled by indirect trusted users with $1, 2, \dots, n$ edges. In our proposed approach, threshold 3 is set so that trust values are maintained and very far indirectly connected users are not considered in improving trust values.

$$M = M^1 \cup M^2 \dots \cup M^n \quad (3)$$

User-user trust matrix signifies trusts between users as shown in Table 1. It is clear from this table that trust value is 1 if there is trust between users, otherwise no trust value. This results in sparse matrix which is the limitation of traditional recommender systems. Our proposed approach improves and increases trusts between users by indirect trust as shown in Table 2.

Table 1. User- User matrix (traditional approach)

Users	u1	u2	u3	u4	u5	u6	u7
u1		1					1
u2			1				
u3				1			
u4					1		
u5						1	
u6			1		1		
u7							

Cold start is due to the fact that when a user is new in recommender system, less numbers of entries are

filled for that user. We have exploited the social connection feature based on trust of social graph. Entries in user-user matrix is filled with users' trusted users hence overcomes cold start issue as clear from Table 2. Our proposed approach improves and increases trusts between users by indirect trust as shown in Table 2. Recommendations from more trusted users improve the choice making of a user.

Table 2. User-User matrix (proposed approach)

Users	u1	u2	u3	u4	u5	u6	u7
u1		1	1				1
u2			1	1			
u3				1	1		
u4				1	1		
u5							
u6			1	1	1		
u7							

Scalability issue is solved by using partitioning large scale social graph based on our proposed approach which is explained in next subsection.

PROPOSED WORK

Partitioning of large-scale graph

Graph partitioning is NP complete and many heuristic functions are proposed to solve it [38][39]. The core motive of these approaches is to minimize communication between distributed sub-graphs. The quality of graph partitioning is highly correlated with communication cost and load balance [40]. We have partitioned social graph in a way so that maximum communication should be within sub-graph. In this manner, communication between nodes is minimum hence avoiding communication overhead.

Figure 4. Ratings prediction using distributed approach

In Figure 4, it is clearly depicted that user-user trust matrix is distributed after graph partitioning, trust is predicted for target user and then it is combined to predict global trust. Predicted trust is used as input to

user-item matrix which is also distributed on clusters. Predicted item ratings are calculated which is the motive of any recommender system. In Figure 4, it is also explained that trust matrix and item rating matrix are input to distributed recommendation model. Our model distributed these matrices based on hyperedge on different nodes. These prediction ratings are combined by our approach to provide global recommendation to target user.

Directed graph can be used for better representation of trust as represented by Figure 5(a). Pregel is best suited for our approach as the input in pregel should be directed graph Using pregel, rank of nodes is calculated and most relevant node in a particular cluster is identified [14]. Partitioning of large scale data nodes is difficult as it contains complex structure embedded in it. Our approach uses transitive closure to cluster only nodes which are having strong trust and influence amongst them. Recommendation accuracy is improved by giving more relevance to nodes which are in same cluster and less relevance on nodes which are in different clusters. Partitions can have different number of nodes depending upon the trust strength between users, i.e. some clusters are large and some are small. Balanced graph partitioning is used for distribution of these clusters on nodes so that some partitions are not assigned large number of vertices which cannot be stored and processed by these partitions. Uniform distribution of nodes on clusters is implemented by setting threshold to 3 i.e. users connected in trust network by directed edges with maximum path of length 3.

Figure 5(a). Original Social trust graph

Figure 5(b). Subgraph1

Figure 5(a). Subgraph2

Graph partitioning based on social connections should be distributed on nodes such that no social connections between connected users are removed. Influence of node in complete social graph should not be reduced in sub-graph. Our approach has achieved these requirements of social graph partitioning. Following are some important points while partitioning this social graph.

- Vertex 3 is repeated in both sub-graphs, but social trust is lossless due to replication.
- Hyperedge in sub-graphs is same as was in original graphs, so recommendation accuracy is not degraded due to partitioning.
- All vertices are covered in sub-graphs and also no vertices are replicated except highest influence node i.e. 3.

On every sub-graph, trust metric $T(u,v)$ is defined for vertex u and v such that

- $T(u,v)$ is not equal to $T(v,u)$ as trust is asymmetric, if a user u trust v , it is not necessary that v also trusts u .
- $T(u,u)=0$.
- if $T(u,v)$ and $T(v,w)$ exists, then it is concluded that $T(u,w)$ exist to infer indirect trust, but upto certain threshold value.

Improving recommendation accuracy

In this step, another motive of proposing our approach i.e. improvement of accuracy and throughput of recommendation is discussed. As clear from Figure 5(b) and Figure 5(c), most reliable node i.e. trusted node by other nodes retains its trust also in sub- graph. Transitive trust between friends-of-friends also remains same. Improvement is due to the fact that correlated nodes within same sub-graph are provided with better recommendation as compared to nodes in different sub-graphs.

$$R_{i,j} = \frac{\sum_{k=0}^n R_{k,j}}{n} \quad (4)$$

where, ratings of any user i for product j , $R_{i,j}$ is calculated by average of ratings of all k trusted users on product j , $R_{k,j}$. It is clear from Equation 4 that ratings prediction is not same as other similarity measures calculations. Trust based prediction of ratings on every sub-graphs provides local ratings predication which can be enhanced to global ratings predications for sub-graphs. Throughput is also better as sub-graphs are on nodes which are distributed and can provide recommendation in less duration. In Experiment section, it is proved empirically that both characteristics of recommendation – accuracy and throughput are better as compared to centralized approach as well as partitioning based on similarity measures.

IMPLEMENTATION ON PREGEL AND GIRAPH

Pregel

Pregel is based on Bulk Synchronous parallel model [41]. Pregel can process large-scale graph algorithms on different clusters [14]. It provides transparent scalability and fault-tolerance [34]. It works as vertex centric approach. Directed graph should be the input to Pregel and output is modified graph with changed neighbor nodes or modified topology. Every node in graph can send message to other nodes and updates its status. Node is identified by unique id and in superstep or iteration i , node sends message to other nodes, and in next superstep $i+1$, other nodes read this message.

MapReduce is also used for large scale graph processing but the disadvantage is that state of complete graph is to be shared on every cluster. Moreover, it is not suitable for large-scale data processing because it do not support many important data mining and machine learning algorithms [15]. Large scale graph processing needs model which can work on message passing technique [14]. In Pregel, communication through messages between nodes to update the status has solved this disadvantage. Pregel is also used for partitioning graph so that sub-graphs can be processed on different nodes. This embedded partitioning technique in Pregel is advantageous for our approach also. It is not mandatory to partition as described in API of Pregel. These API are abstract and can be modified as per our approach. Graph is partitioned based on trust and influence in our approach. We have modified compute() to achieve this.

Figure 6 Graph with node weights

In Figure 6, example is used to describe that if global popularity of a node is to be calculated using Pregel, it can be easily implemented by using message passing between nodes. In this graph, global popularity of node is to be calculated i.e. node with highest value of weight is to be identified. In Pregel like system, it can be identified by using message between nodes. Node 1 sends message to all connected nodes with its weight value, max value of weight is updated to 5. Then node 2 has less weight value than max value, it will not update max value. Node 4 works same as it does not have weight more than max value. But when message is received by node 3, it updates max value to 7. Node with highest weight is identified by using message.

Giraph

Apache Giraph is open source framework which is implemented in Java. Vertex class is implemented and compute method is overridden to manipulate social graph. Master node assigns load and process to worker nodes. These nodes can communicate through messages. Zookeeper is used for synchronization and fault tolerance. Giraph runs as map in Hadoop and Pregel provides the API which is used by Giraph. Vertex is identified by ID and value is assigned to edges. Method voteToHalt() is used by vertices to terminate and send the confirmation message of completion of job. The advantage of using Giraph is that methods are not written by programmers, only it is modified as per requirements. There is no need to implement methods to send and receive messages between nodes. It is embedded within Giraph API. Social graph analysis requires many graph algorithms like Shortest Path, Transitive closure, asymmetric, local and

global popularity of nodes. These algorithms are iterative in nature, so cannot be deployed on Hadoop clusters. Giraph, which is open source implementation of Pregel, can be used for implementing these algorithms. We have implemented our approach using Giraph 1.2.0.

EXPERIMENTAL RESULT

In this section, we present experiment setup, datasets, evaluation metrics and analysis of comparison results of our approach with existing approaches. In our proposed approach, partitioning quality, recommendation accuracy and scalability are considered as the most important factors. We have proposed that large-scale social graph partitioning based on trust using transitive closure with some vertices replication is better as compared to similarity based partitioning strategies. Improved social graph using our proposed approach on datasets is deployed on Giraph 1.2.0 configured with Hadoop 1.0.3 on Linux nodes. LiveJournal [42] dataset is used to validate that social trust is improved and better partitioning quality is achieved with increased scale of data. In this dataset, there are 4847571 nodes describing users and 68993373 edges describing the friendship between users as described in Table 3.

Table 3. LiveJournal dataset statistics

Dataset Statistics	
Nodes	4847571
Edges	68993373

Epinions [43] dataset is used for validation of recommendation accuracy and throughput. Epinions is collection of feedback of products by users and dataset statistics is described in Table 4. This dataset also contains trust information of users i.e. who-trusts-who. It is standard dataset used for analyzing social recommendation accuracy as it contains users' mutual trust and user- item ratings information. MAE calculation and recommendations per second (throughput) requires ratings of users and this is easily available in this dataset, so it is best suited for our proposed approach.

Table 4. Epinions dataset statistics

Dataset Statistics	
Users	40163
Items	139738
Ratings	664824
Density	0.051%

Improved partitioning quality and social trust

LiveJournal dataset is used for analyzing quality of partitions and social trust with our approach. In previous research works, several evaluation metrics have been used for analyzing quality of partitioning

such as Q value, replication factor [15], load balance factor [18], modularity [36], standard deviation and locality. In this paper, locality is used as evaluation metrics due to its significance in partitioning quality. Locality is the measure of ratio of edges which connect vertices in same partition with total number of edges [44]. Communication cost is dependent on the number of vertices and edges in partition [40], so if locality is improved then communication cost is reduced significantly.

$$\partial = \frac{n_{ep}}{n_e} \quad (5)$$

In Equation 5, n_{ep} is the number of edges with in partition with same vertices as in original graph and n_e is total number of edges.

Figure 7. Comparison of existing partitioning and proposed approach partitioning quality based on locality

Our proposed approach is compared with existing graph partitioning approaches as depicted in Figure 7. In [44], scalable graph partitioning is implemented by using label propagation technique. [45] explained multi-level label propagation for efficiently partitioning graph. Graph partitioning streaming technique is implemented during loading of graph on cluster in [46]. It is clear from Figure 7 that in less numbers of partitions, locality of our approach is better. With the increase in number of partitions, it is degraded as connectivity between vertices is reduced for all graph partitioning approaches. Improved locality by our approach is due to the fact that trusts between users are maintained in partitions which result is strong connectivity. In addition to improving partitioning quality, trust improvements on sub-graphs are also applied to improve throughput and recommendation accuracy. Very few research works have applied manipulations on partitions to improve recommendation.

Figure 8. Social trust improvement

In [27] recommendation is improved by using social information on collaborative filtering. Graph partitioning is implemented using Normalized cut in [47]. In Figure 8, trust values i.e. numbers of trusters are increased using our proposed approach. The reason for improvement is that we are implementing our approach on sub-graphs and not on original graph. It is described in proposed approach section that transitive closure and hyperedge are implemented on sub-graphs after partitioning. The reason for implementing our approach after partitioning is that if we have used approach on original sub-graphs, processing and analyzing social trust will take enormous time and resources.

DISCUSSION

Recommendation accuracy and Throughput

Epinions dataset is used for analyzing recommendation accuracy and throughput in our approach. Evaluation metrics for analyzing accuracy of recommendation are Mean Absolute Error - MAE and throughput. MAE is the average of difference between predicated ratings by proposed approach and actual ratings.

$$MAE = \sum_{i=0}^n (p_r(u, i) - p(u, i)) / n \quad (6)$$

Where, p_r is predicted ratings by proposed approach for user u on product i and p is actual ratings for user u on product i . Several research works have proved that even small improvements in the value of MAE is significant achievement for the approach. Lower value of MAE indicates better prediction accuracy.

Figure 9. MAE comparison of collaborative filtering and proposed approach

In Figure 9, when numbers of partitions are less, MAE for collaborative filtering is better as compared to large number of partitions. It is due to fact that more numbers of partitions means less number of neighbors who have rated same product with same ratings. This also applies on our approach as well, because trusted users will be less when more numbers of partitions are used for experiment.

Figure 10. Throughput comparison of collaborative filtering and proposed approach

Throughput is measured as recommendations per second. Throughput is very important factor for analyzing response time for any recommender system for large scale of data. Accurate partition with retaining social trust between nodes improves throughput which is verified from Figure 10. MAE and throughput prove that our approach provides better recommendation accuracy.

CONCLUSION

Recommendation for products on e-commerce sites or topics, friends on social networking sites is of great interest for data analyst and researchers. In the era of Online Social Networks (OSN) and Big data, there are a lot of choices available for users. Recommendation systems provide support to users to make better choices. Social Network is represented in the form of social graph, so combining social graph and big data result in social big graph. Social big graph analysis and mining specifically for recommendation is the core motive of our research work. Sparsity and cold start are alleviated by using indirect and direct trust in social graph using transitive closure and hyperedge. Scalability issue is addressed by using partitioning social graph based on trust and providing local recommendations on sub-graph and combining recommendation for complete social graph. These sub-graphs can be processed effectively on every node. Mahout and PowerGraphh are used to partition large-scale graph as these are most suitable for large-scale graph analysis. In experiment analysis, LiveJournal and Epinions datasets are used as these contain trust values between users. Mean Absolute Error (MAE) and locality are used as evaluation metrics to validate accuracy of proposed approach. Experiment analysis proves that our proposed approach improves partitioning quality as well as improves recommendation accuracy for large-scale data.

DECLARATIONS

Availability of data and materials

Not applicable

Competing interests

The authors declare that they have no competing interests

Funding

The funding of this research was supported by the universities

Authors' contributions

"FLG provided with the problem analyzing and data collecting as well as processing that initiate the drafting the manuscript. TM performed with testing and some calculations of the experiment result and continuing in writing the manuscript. The FLG & TM provide some survey literatures and some administration matters as well as process the payment of the research. All authors read and approved the final manuscript."

Acknowledgements

Not applicable

REFERENCES

- [1] Bello-Orgaz, G., Jung, JJ., & Camacho, D. (2016). Social Big Data: Recent achievements and new challenges. *Information Fusion. Science Direct*, 28, 45-59.
- [2] Eirinaki, M., Gao, J., Varlamis, I., & Tserpes, K. (2018). Recommender Systems for Large-Scale Social Networks: A review of challenges and solutions. *Future Generation Computer Systems*.78, 413-418.
- [3] Resnick, P., & Varian, HR. (1997). Recommender Systems. *communications of the ACM*, 40(3), 56-58.

- [4] Su, X., & Khoshgoftaar, T. (2009). A survey of collaborative filtering techniques. *Advances in Artificial Intelligence*, 2009(421425).
- [5] Guo, G., Zhang, J., & Yorke-Smith, N. (2015). TrustSVD: collaborative filtering with both the explicit and implicit influence of user trust and of item ratings. *In Twenty-Ninth AAAI Conference on Artificial Intelligence*.
- [6] Jamali, M., & Ester, M. (2010). A matrix factorization technique with trust propagation for recommendation in social networks, *In Proceedings of the fourth ACM conference on Recommender systems*, 135-142.
- [7] Yang, B., Lei, Y., Liu, D., & Liu, J. (2013). Social collaborative filtering by trust. *in Proceedings of the 23rd International Joint Conference on Artificial Intelligence*, 2747-2753.
- [8] Fang, H., Bao, Y., & Zhang, J. (2014). Leveraging decomposed trust in probabilistic matrix factorization for effective recommendation. *in Proceedings of the 28th AAAI Conference on Artificial Intelligence*, 30-36.
- [9] Sardianos, C., Tsirakis, N., & Varlamis, I. (2018). A Survey on the Scalability of Recommender Systems for Social Networks. *In Social Networks Science: Design, Implementation, Security, and Challenges*, 89-110.
- [10] Mateos, C., Zunino, A., & Hirsch, M. (2013). EasyFJP: Providing hybrid parallelism as a concern for divide and conquer Java applications, *Comput. Sci. Inf. Syst.*, 10 (3), 1129-1163.
- [11] Lumsdaine, A., et al. (2008). Challenges in parallel graph processing. *Parallel Processing Letters*.
- [12] Apache hadoop, <http://hadoop.apache.org>.
- [13] Dean, J., & Ghemawat, S. (2008). MapReduce: simplified data processing on large clusters, *Communications of the ACM*, 51(1), 107-113.
- [14] Malewicz, G., Austern, M.H., Bik, A.J.C., Dehnert, J.C., Horn, J., Leiser, N., & Czajkowski, G. (2010). Pregel: A System for Large-Scale Graph Processing. *Proceedings: ACM SIGMOD International Conference on Management of data*, ISBN No. 9781450300322, 135-146.
- [15] Low, Y., Bickson, D., Gonzalez, J., Guestrin, C., Kyrola, A., & Hellerstein, J.M. (2012). Distributed GraphLab: A Framework for Machine Learning and Data Mining in the Cloud. *Proceedings: VDLB*, 5(8), 716-727.
- [16] Owen, S., Anil, R., Dunning, T., & Friedman, E. (2011) Mahout in Action. *Manning Publications Co. Greenwich, CT, USA*.
- [17] Apache Giraph Project. <http://giraph.apache.org>.
- [18] Gonzalez, J.E., Low, Y., Gu, H., Bickson, D., & Guestrin, C. (2012). PowerGraph: distributed graph-parallel computation on natural graphs. *In: Proceedings of OSDI*.
- [19] Xin, R.S., Gonzalez, J.E., Franklin, M.J., & Stoica, I. (2013). GraphX: a resilient distributed graph system on Spark. *In: Proceeding of GRADES*.
- [20] Chen, R., Shi, J., Zang, B., & Guan, H. Bipartite-oriented distributed graph partitioning for big learning. *Proceedings: APSys*. ISBN No. 9781450332044, 14.
- [21] Yang, S., Long, B., Smola, A., Sadagopan, N., Zheng, Z., & Zha, H. (2011). Like like alike: joint friendship and interest propagation in social networks. *Proceedings: International Conference on World Wide Web*, 537-546.
- [22] Bedi, P., Kaur, H., & Marwaha, S. (2007). Trust based recommender systems for semantic web. *Proceedings: International Joint Conference on Artificial Intelligence*, 2677-2682.
- [23] Massa, P., & Avesani, P. (2007). Trust-aware recommender systems. *In Proc. of RecSys '07*, 17-24, Minneapolis, MN, USA.
- [24] O'Donovan, J., & Smyth, B. (2005). Trust in recommender systems. *In Proc. of IUI '05*, 167-174, San Diego, California, USA.
- [25] Koren, Y., Bell, R., & Volinsky C. Matrix factorization techniques for recommender systems, *Computer* 42 (8).
- [26] Ma, H., Zhou, D., Liu, C., Lyu, M.R. and King, I. (2011). Recommender systems with social regularization. *In Proceedings of the fourth ACM international conference on Web search and data mining*, pp. 287-296.
- [27] Pham, M.C., Cao, Y., Klammar, R., & Jarke M. (2011). A clustering approach for collaborative filtering recommendation using social network analysis. *JUCS*, 17(4), 583-604.
- [28] Balabanović, M., & Shoham, Y. (1997). Fab: content-based, collaborative recommendation. *Communications of the ACM*, 40 (3), 66-72.
- [29] Guo, G., Zhang, J., & Yorke-Smith, N. (2016). A Novel Recommendation Model Regularized with User Trust and Item Ratings. *IEEE Transactions on knowledge and Data Engineering*. 28(7), 1607-1620.
- [30] Golbeck, J., & Hendler, J. (2006). Inferring binary trust relationships in Web-based social networks. *ACM Transactions on Internet Technology*. 6(4), 497-529.
- [31] Lacić, E., Kowald, D., & Trattner, C. (2014). SocRecM: A scalable social recommender engine for online marketplaces. *Proceedings: ACM Conference on Hypertext and social media*, 308-310.
- [32] Corbellini, A., Godoy, D., Mateos, C., Schiaffino, S., Zunino, A. (2018) DPM: A novel distributed large-scale social graph processing framework for link prediction algorithms. *Future Generation Computer Systems*, 78, 474-480.
- [33] Qian, T., Yang, Y., & Wang, S. (2010). Refining Graph Partitioning for Social Network Clustering. *Proceedings: International Conference on Web information Systems Engineering*, 6488, 77-90.
- [34] Salihoglu, S., & Widom, J. (2014). Optimizing Graph Algorithms on Pregel-Like Systems. *Proceedings: International Conference on Very Large Data Bases*. 7(7), 577-588.
- [35] Levin, R., Abassi, H., & Cohen, U. (2016). Guided Walk: A Scalable Recommendation Algorithm for Complex Heterogeneous Social Networks. *Proceedings: ACM Conference on RecSys*, ISBN No. 978145034035, 293-300.
- [36] Newman, M.E.J., & Girvan, M. (2004). Finding and evaluating community structure in networks. *Physical review. E, Statistical, nonlinear and soft matter physics*, 69.
- [37] Aioli, F. (2013). Efficient top-n recommendation for very large scale binary rated datasets. *Proceedings: ACM Conference on RecSys*, ISBN No. 9781450324090, 273-280.
- [38] Karypis, G., & Kumar V. (1998). A fast and highly quality multilevel scheme for partitioning irregular graphs. *SIAM Journal on Scientific Computing*. 20(1), 359-392.
- [39] Ng, A., Jordan, M., Weiss, Y. (2001). On spectral clustering: analysis and an Algorithm. *Advances in Neural Information*

- [40] Onizuka, M., Fujimori, T., Shiokawa, H.(2017). Graph Partitioning for Distributed Graph Processing. *Data Science and Engineering*, 2(1), 94-105.
- [41] Valiant, LG. (1990) A bridging model for parallel computation. *Communications of ACM*, 33(8), 103-111.
- [42] <https://snap.stanford.edu/data/soc-LiveJournal1.html>
- [43] <http://www.trustlet.org/epinions.html>
- [44] Martella, C., Logothetis, D., Loukas, A., Siganos, G. (2017). Spinner: Scalable Graph Partitioning in the Cloud. *Proceedings: IEEE Conference on Data Engineering*, 1083-1094.
- [45] Wang, L., Xiao, Y., Shao, B. &, Wang, H. How to partition a billion-node graph. *Proceedings: IEEE International Conference on Data Engineering*, 568-579.
- [46] Stanton, I., & Kliot, G. Streaming graph partitioning for large distributed graphs. *Proceedings: ACM SIGKDD international conference on Knowledge discovery and data mining*, 1222-1230.
- [47] Bellogin A, & Parapar J. (2012). Using graph partitioning techniques for neighbour selection in user-based collaborative filtering. *Proceedings: ACM conference on Recommender systems*, 213-216.

Figures

Figure 1

User-User trust graph

Figure 2

Improved social graph by using hyperedge

Figure 3

User-Item ratings graph

Figure 4

Ratings prediction using distributed approach

Figure 5(a).

Figure 5(b).

Figure 5(a).

Figure 5

(a). Original Social trust graph (b). Subgraph1 (a). Subgraph2

Figure 6

Graph with node weights

Figure 7

Comparison of existing partitioning and proposed approach partitioning quality based on locality

Figure 8

Social trust improvement

Figure 9

MAE comparison of collaborative filtering and proposed approach

Figure 10

Throughput comparison of collaborative filtering and proposed approach