International Journal of Cognitive Informatics and Natural Intelligence
Volume 14 • Issue 4 • October-December 2020

Scalable Recommendation Using Large Scale Graph Partitioning With Pregel and Giraph

Gourav Bathla, Punjabi Univeristy, India
https://orcid.org/0000-0003-4198-9647

Himanshu Aggarwal, Punjabi University, Patiala, India
https://orcid.org/0000-0003-1782-8376

Rinkle Rani, Thapar Institute of Engineering and Technology, India

ABSTRACT

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 products 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. 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. The main focus of this article is to improve scalability in terms of locality and throughput and provides better recommendations to users with large-scale data in less response time. In this article, the social big graph is partitioned and distributed on different nodes based on Pregel and Giraph. In the proposed approach ScaleRec, 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 proposed approach. In ScaleRec, hyperedge and transitive closure are used to enhance social trust amongst users. Experiment analysis on standard datasets such as Epinions and LiveJournal proves that better locality and recommendation accuracy is achieved by using ScaleRec.

KEYWORDS

Data Analytics, Giraph, Pregel, Social Big Data, Social Recommendation

1. INTRODUCTION

Big data is generated by social media on social networking sites (Bello-Orgaz, Jung & Camacho, 2016). 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.

DOI: 10.4018/IJCINI.2020100103
This article, originally published under IGI Global's copyright on October 1, 2020 will proceed with publication as an Open Access article starting on February 1, 2021 in the gold Open Access journal, International Journal of Cognitive Informatics and Natural Intelligence (converted to gold Open Access January 1, 2021), and will be distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/4.0/) which permits unrestricted use, distribution, and production in any medium, provided the author of the original work and original publication source are properly credited.

Collaborative filtering, content-based, and hybrid-based are different techniques of recommender systems (Eirinaki et al., 2018; Resnick & Varian, 1997; Su & Khoshgoftaar, 2009). 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 (Guo, Zhang & Yorke-Smith, 2015; Jamali & Ester, 2010; Yang et al., 2013; Fang, Bao & Zhang, 2014). The main concern for researchers is scalability which needs to be addressed for large-scale data. Traditional recommender systems work well for limited scale of social data. Moreover, their algorithms are designed for centralized approach only. If these systems are deployed on large-scale data, throughput is degraded significantly, which results in reducing the users' interest in these systems. In this paper, the key motivation is to improve recommendation accuracy even for a large number of nodes in the social graph.

Recommendation systems leverage Big data in the form of the large-scale social graph and efficient graph algorithms are important for these systems. The large-scale social graph cannot be processed on centralized system. There is need for a distributed approach where sub-graphs can be processed in parallel. Large-scale recommender systems have leveraged distributed algorithms for finding recommendation (Sardianos, Tsirakis, & Varlamis, 2018). Graph partitioning is a technique which can address the scalability issue. Large-scale graph partitioning in traditional recommendation models uses random walk, Fork-Join (Mateos, Zunino, & Hirsch, 2013) or hash partitioning to divide the graph into sub-graphs. In our proposed approach ScaleRec, the direct and indirect trust-based walk is used to partition graph with relevant nodes only which improves locality. The 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 (Lumsdaine, 2008).

Conventional data analytics technologies based on centralized approach cannot store and process large-scale data. Big data frameworks such as Hadoop, MapReduce (Dean & Ghemawat, 2008), Pregel (Malewicz et al., 2010), GraphLab (Low et al., 2012), Mahout (Owen et al., 2011) and Giraph, PowerGraph (Gonzalez et al., 2012), GraphX (Xin et al., 2013), CUDA (W3) and GPU (W3) 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 (Chen et al., 2014). 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 another user. It is also concluded that users with the same interest are connected to each other on social networks (Yang et al., 2011). Trust is used as an important factor for social recommender systems (Bedi, Kaur & Marwaha, 2007; Massa & Avesani, 2007; O'Donovan & Smyth, 2005). Traditional recommendation systems use random walk which does not consider any weights assigned to edges (Yang et al., 2011). Moreover, the drawback of existing social trust-based recommender system is the 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 a user, it will reveal the user's level of likings for product or topic. In our proposed approach, the rating is predicted by considering ratings of trusted users only.

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

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 increases number of trusted users which improves ratings prediction. The remainder of the paper is as follows. Section 2 discusses literature work in social recommendation, large-scale graph partitioning and scalable social recommendation. In Section 3, ScaleRec approach is proposed which covers large-scale graph analysis, large-scale graph partitioning and improving recommendation accuracy. Section 4 explains implementation on Pregel and Giraph in detail. Experiment analysis is demonstrated in Section 5 and finally Section 6 concludes the paper with future direction.

2. RELATED WORK

Several approaches are proposed for social recommendation improvement such as matrix factorization (Jamali & Ester, 2010) (Fang, Bao & Zhang, 2014) (Koren, Bell & Volinsky, 2009), social regularization (Ma et al., 2011), trust-based (Bedi, Kaur & Marwaha, 2007) (Massa & Avesani, 2007)(O'Donovan & Smyth, 2005), hierarchical clustering (Pham et al., 2011) 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.

2.1. Social Recommendation

In (Balabanovic & Shoham, 1997), content-based and collaborative filtering techniques are mentioned as recommender systems techniques. The hybrid system can utilize advantages of these techniques and ignore the disadvantages. The proposed approach uses profiles of users provided by content-based and user experience provided by collaborative filtering. Experiment analysis proved that proposed approach provides improved performance. In (Guo, Zhang & Yorke-Smith, 2015), TrustSVD was proposed which exploited the trust for improving recommendations. Implicit and explicit influence of ratings and trust is incorporated into the recommendation model. Experiments were conducted on four datasets and this approach was compared with ten recommendation models to validate the effectiveness of this approach. In (Golbeck & Hendler, 2006), 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 to score email based on ratings of users. The framework is proposed in (Lacic, Kowald & Trattner, 2014) for developers so that recommendation algorithms can be easily implemented and integrated to use social datasets. Four algorithms were provided to users – MostPopular(MP), Collaborative Filtering (CF), Content Based (C) and Hybrid Recommendations (CCF).

2.2. Large-Scale Graph Partitioning

Pregel is described in (Malewicz et al., 2010). 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 (Chen et al., 2014), the graph is partitioned based on bipartite graph. BiGraph is implemented using PowerGraph. 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 (Corbellini, 2018). 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 (Qian, Yang & Wang, 2010), 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 (Salihoglu & Widom, 2014). Minimum spanning forest

and graph coloring are improved by using a parallel approach. In (Levin, Abassi & Cohen, 2016), a guided walk is demonstrated which is based on the relevance of link in the 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 (Newman & Girvan, 2004), modularity Q value is used to check the relevance of graph partitioning. It is defined as the difference of edges which are in the same community and average edges on random connections between vertices.

2.3. Scalable Social Recommendation

In (Eirinaki, 2018), it is mentioned that the 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 (Aiolli, 2013), scalable algorithms are presented for large-scale data to provide top-N recommendation. Asymmetric cosine and prediction is integrated with an algorithm to improve MAP (Mean Average Precision). It is shown in this work that these techniques work well for CF. Social big data application in social networks analysis is presented in (Bello-Orgaz, Jung & Camacho, 2016). 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 the context of social big data is elaborated like text analytics, social recommendation etc.

3. PROPOSED WORK

Several research works have improved social recommendation by exploiting matrix factorization, social regularization (Ma et al., 2011) and social trust (Bedi, Kaur & Marwaha, 2007; Massa & Avesani, 2007; O'Donovan & Smyth, 2005). 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 is 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 by the following algorithm.

Algorithm ScaleRec (U_i, I_j, MR_{ij}, MT_{ij})
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.

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

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 $if \, u_i \xrightarrow{trust} u_j$ and $u_j \xrightarrow{trust} u_k$ it is concluded that $u_i \xrightarrow{trust} u_k$, 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

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

Where, i is user number and j is for product number. Ratings $R_{i,j}$ is calculated based on trusted j users who have provided rating to product j.

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

9. Calculate MAE

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

10. Calculate locality

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

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

12. Compare these metrics with existing approaches

13. Exit

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

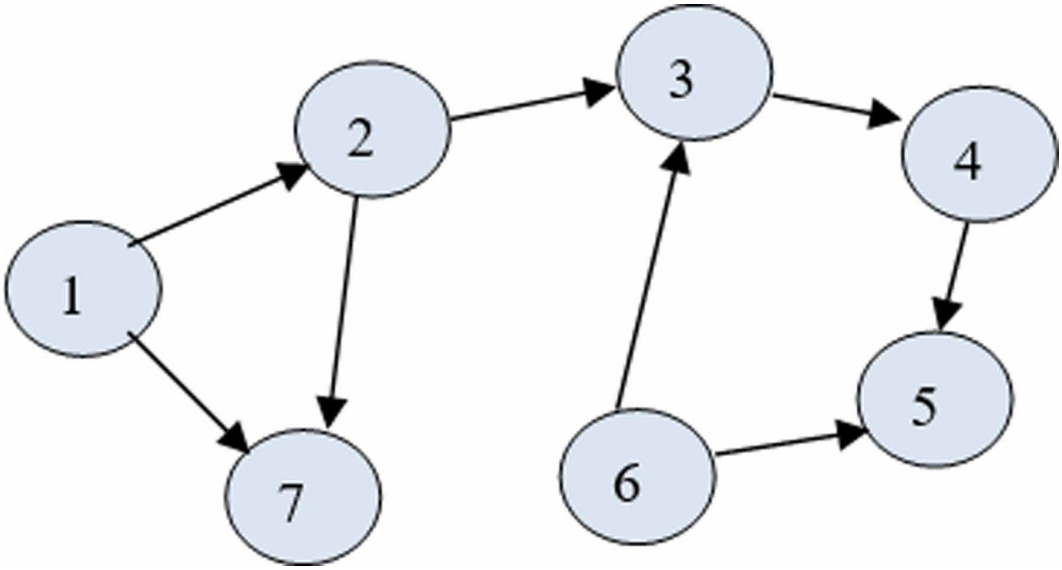
- Large-scale social graph analysis with the use of trust and influence
- Partitioning of large-scale social graph on different nodes using our proposed approach
- Improving recommendation accuracy and throughput on sub-graphs and also for complete graph. We have solved these issues in sequential manner, which is elaborated as follows.

3.1. Large Scale Social Graph Analysis

Graph is most suitable for analyzing social big data. Social graph, hypergraph and fuzzy-graph (Jan et al., 2019) are various types of graph. The 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 the social trust graph of 7 users. User 1 trusts user 2 and it is represented by a 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 the 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.

Our approach follows both asymmetric and transitive property of social trust graph. Edges are directed in the 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 the asymmetric property. If A - B i.e. A trusts user B and B-C i.e. B trusts user C, then by using an indirect trust, A-C i.e. A trusts user C i.e. it follows the transitive property. Hyperedge i.e. direct edge between A and C will be drawn in an 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 the improved social graph, the threshold is set to 3 i.e. only strongly connected friends-of-friends will be included in improved social trust graph.

Figure 1. User-User trust graph



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 the social graph. The user provides ratings to products on the scale of 1-5. In Figure 3, it is clearly demonstrated that ratings are 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 a 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 2. Improved social graph by using hyperedge

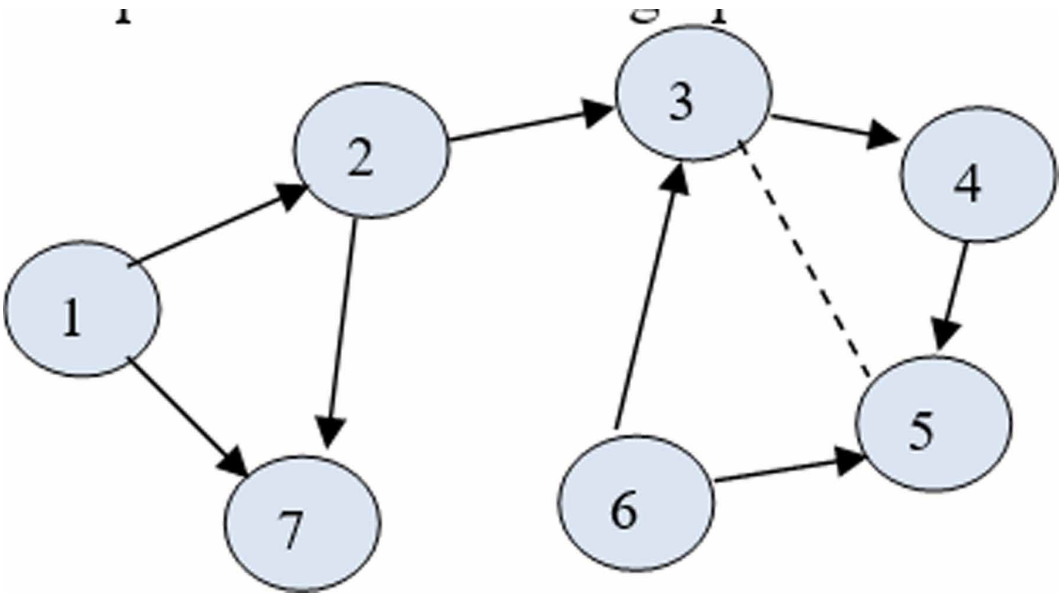
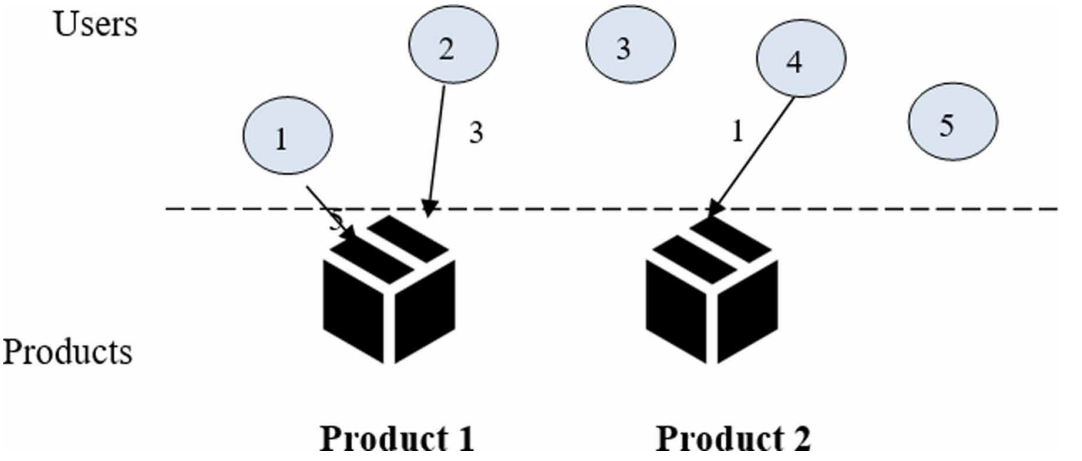


Figure 3. User-Item ratings graph



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.

$$M = U^{m*n} \tag{2}$$

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

$$M = M^1 \cup M^2 \tag{3}$$

Where, M^1 is trust matrix with nodes connected directly through 1 edge and M^2 is trust matrix with nodes connected through 2 edges. 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				1
u3				1			
u4					1		
u5							
u6			1		1		
u7							

Table 2. User-User matrix (proposed approach)

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

Cold start is due to the fact that when a user is new in recommender system, fewer numbers of entries are filled for that user. We have exploited the social connection feature based on trust of social graph. Entries in the user-user matrix are 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 an indirect trust as shown in Table 2. Recommendations from more trusted users improve the choice making of a user.

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

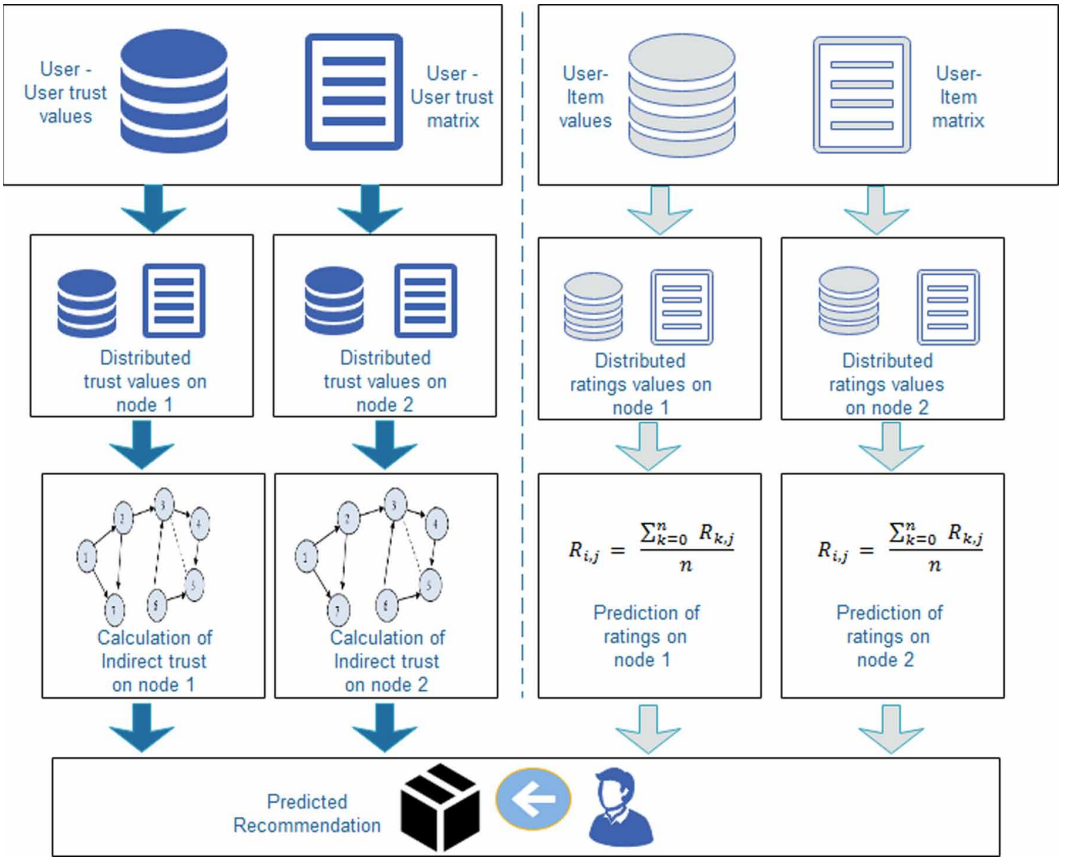
3.2. Partitioning of Large-Scale Graph

Graph partitioning is NP complete and many heuristic functions are proposed to solve it (Karypis & Kumar, 1998) (Ng, Jordon & Weiss, 2001). 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 (Onizuka, Fujimori & Shiokawa, 2017). 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.

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 a 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 (Malewicz et al., 2010). 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 4. Ratings prediction using distributed approach



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 (Figure 6).

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 t infer indirect trust, but up to certain threshold value.

3.3. 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.

Figure 5. (a) original social trust graph (b) subgraph 1 (c) subgraph 2

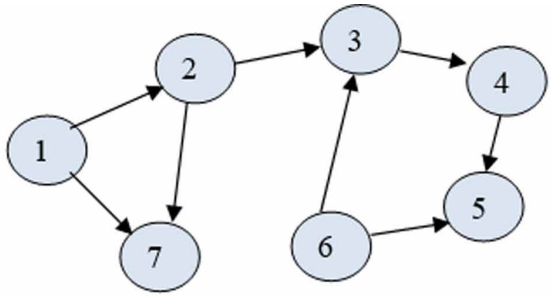


Figure 5(a). Original Social trust graph

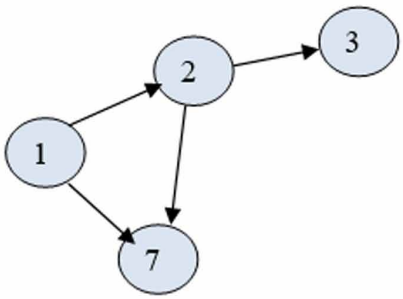


Figure 5(b). Subgraph1

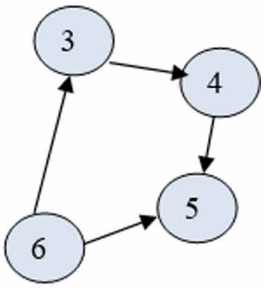


Figure 5(c). Subgraph2

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

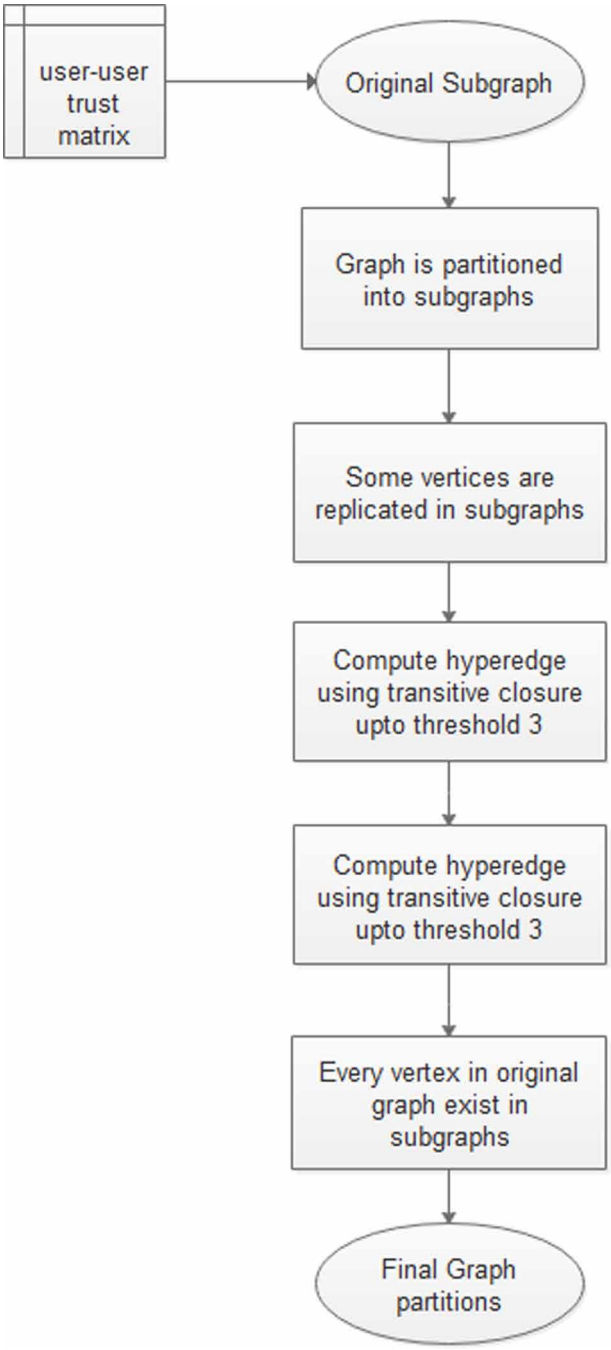
where, ratings of any user i for product j, $R_{i,j}$ is calculated by average of ratings of all k trusted users on product j, $R_{k,j}$. It is clear from Equation 4 that ratings prediction is not same as other similarity measures calculations. Trust based prediction of ratings on every sub-graphs provides 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 the experiment section, it is proved that empirically that both characteristics of recommendation – accuracy and throughput are better as compared to centralized approach as well as partitioning based on similarity measures.

4. IMPLEMENTATION ON PREGEL AND GIRAPH

4.1. Pregel

Pregel is based on Bulk Synchronous parallel model (Valiant, 1990). Pregel can process large-scale graph algorithms on different clusters (Malewicz et al., 2010). It provides transparent scalability and fault-tolerance (Salihoglu & Widom, 2014). It works as vertex centric approach. The directed graph

Figure 6. Flow chart of graph partitioning

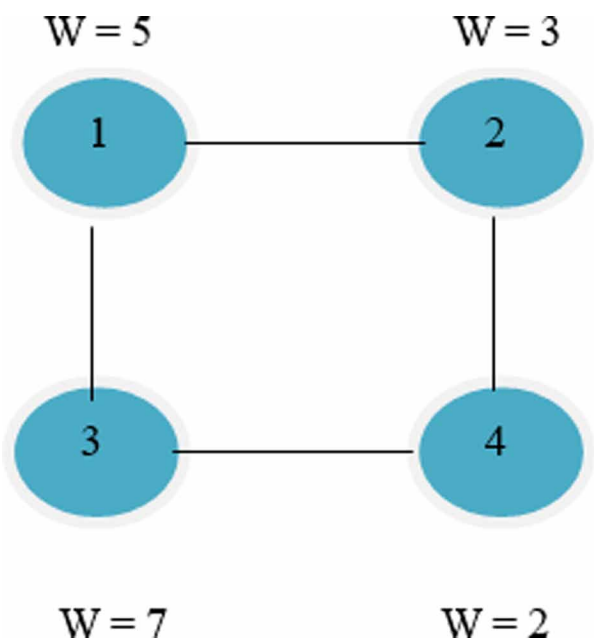


should be the input to Pregel, and output is modified graph with changed neighbor nodes or modified topology. Every node in the graph can send a message to other nodes and updates its status. Node is identified by a unique id and in superstep or iteration i, node sends message to other nodes, and in 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 does not support many important data mining and machine learning algorithms (Low et al., 2012). Large scale graph processing needs model which can work on message passing technique (Malewicz et al., 2010). 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.

In Figure 7, 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.

Figure 7. Graph with node weights



4.2. 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.

5. EXPERIMENT ANALYSIS

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 ScaleRec, 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 and it 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 (W1) dataset is used to validate that social trust is improved and better partitioning quality is achieved with the 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 (Year 2009)

Dataset Statistics	
Nodes	4847571
Edges	68993373

Epinions (W2) dataset is used for validation of recommendation accuracy and throughput. Epinions is a 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.

5.1. 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 the quality of partitioning such as replication factor (Low et al., 2012), load balance factor (Gonzalez et al.,2012), modularity (Newman & Girvan, 2004), standard deviation and the locality. In this paper, locality is used as evaluation metrics due to its significance in partitioning quality. The locality is the measure of ratio of the edges which connect vertices in same partition with total number of edges (Martella et al., 2017). Communication cost is dependent on the number of vertices and edges in partition (Onizuka, Fujimori & Shiokawa, 2017), so if the locality is improved then communication cost is reduced significantly.

Table 4. Epinions dataset statistics (Year 2003)

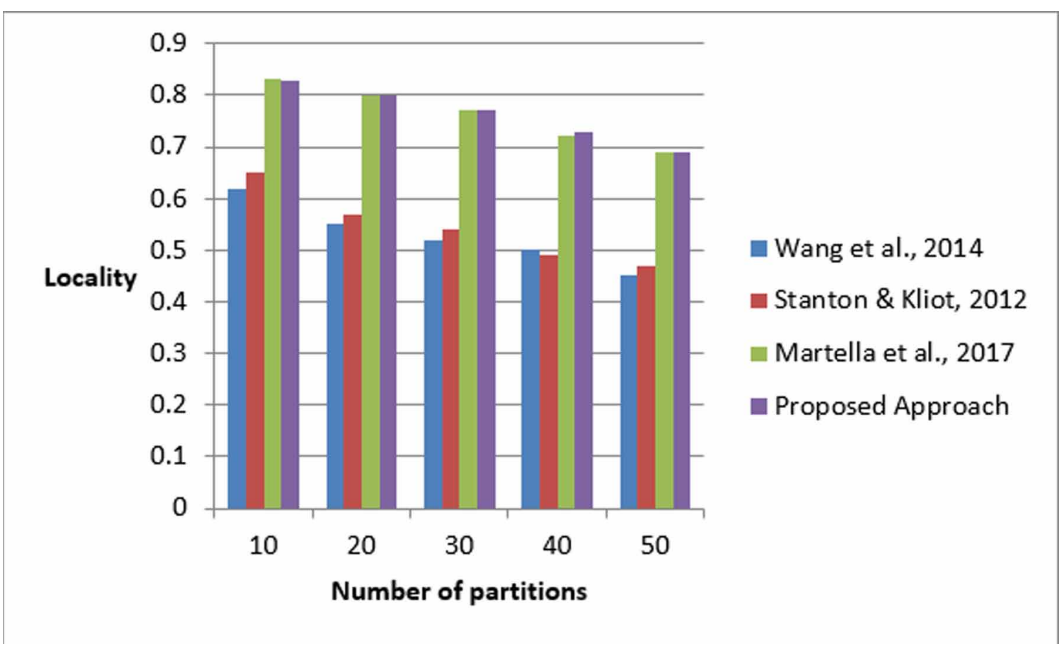
Dataset Statistics	
Users	40163
Items	139738
Ratings	664824
Density	0.051%

$$\partial = \frac{n_{ep}}{n_e} \tag{5}$$

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

Our proposed approach is compared with existing graph partitioning approaches as depicted in Figure 8. In (Martella et al., 2017), scalable graph partitioning is implemented by using label propagation technique. Multi-level label propagation for efficiently partitioning graph is explained in (Wang et al., 2014). Graph partitioning streaming technique is implemented during loading of graph on cluster in (Stanton & Kliot, 2012). It is clear from Figure 8 that in fewer 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 proposed approach ScaleRec is due to the fact that trusts between users are maintained in partitions which result is strong connectivity. The reason for sustaining trust in subgraph is that partitioning is based on the most influence node in original graph as explained in Section on Partitioning of large-scale

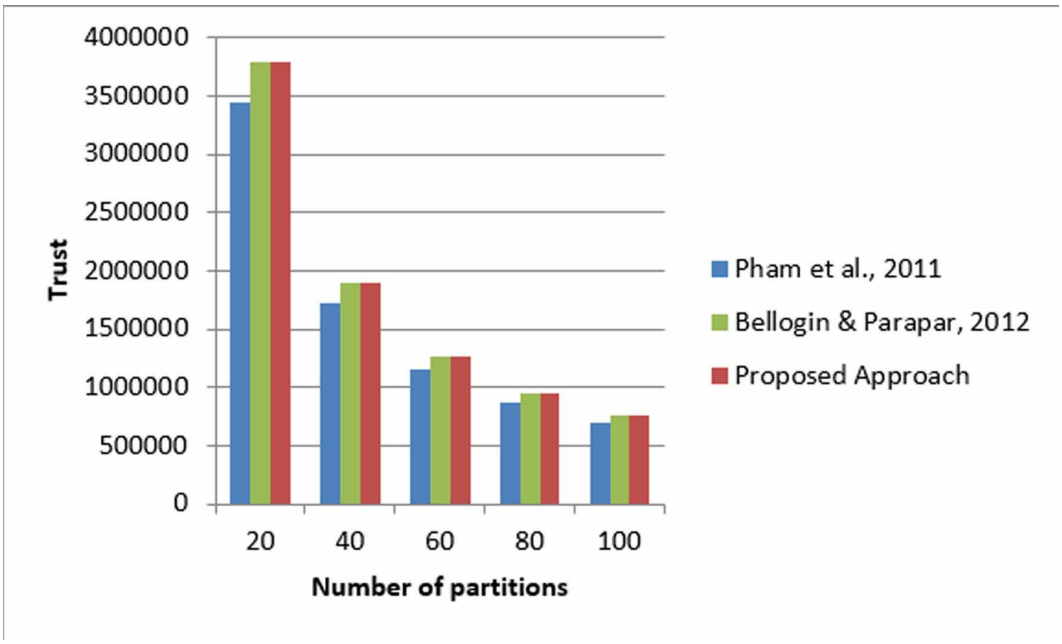
Figure 8. Comparison of locality for existing partitioning and proposed approach partitioning quality



graph. In subgraphs, other nodes will be connected as in original subgraph. 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.

In (Pham et al., 2011) recommendation is improved by using social information on collaborative filtering. Graph partitioning is implemented using Normalized cut in (Bellogin & Parapar, 2012). In Figure 9, trust values i.e. numbers of trust values 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.

Figure 9. Social trust improvement



5.2. Recommendation Accuracy and Throughput

Epinions dataset is used for analyzing recommendation accuracy and throughput in our approach. This dataset is already used by state-of-the-art approaches (Jamali & Ester, 2010) (Guo, Zhang, & Yorke-Smith, 2016) to analyze recommendation accuracy. Evaluation metrics for analyzing the 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 ((Guo, Zhang & Yorke-Smith, 2016; Ma et al., 2011).

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

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

In (Pham et al., 2011; Bellogin & Parapar, 2012), similarity between users is calculated by using Pearson coefficient.

$$\text{pcc}(U_{a,}\ U_{b)} = \frac{\sum_{i=1}^{n}\left(R(a)_i - \overline{R(a)}\right)\left(R(b)_i - \overline{R(b)}\right)}{\sqrt{\sum_{i=1}^{n}\left(R(a)_i - \overline{R(a)}\right)^2 \sum_{i=1}^{n}\left(R(b)_i - \overline{R(b)}\right)^2}} \tag{7}$$

where, R(a)i is rating provided by User a for product i and $\overline{R(a)}$ is average of ratings provided by user a for products, R(b)i is rating provided by User b for product i and $\overline{R(b)}$ is average of ratings provided by user b for products. Similarity is calculated for products 1……n between users Ua and Ub. The value of correlation is -1 to 1.

In Figure 10, when numbers of partitions are less, MAE for collaborative filtering is better as compared to large number of partitions. It is due to the fact that more numbers of partitions mean fewer numbers of neighbors who have rated the same product with the same ratings. It is also depicted in Figure 10 that our proposed approach outperforms existing approaches in terms of recommendation accuracy.

Figure 10. MAE comparison of collaborative filtering and proposed approach

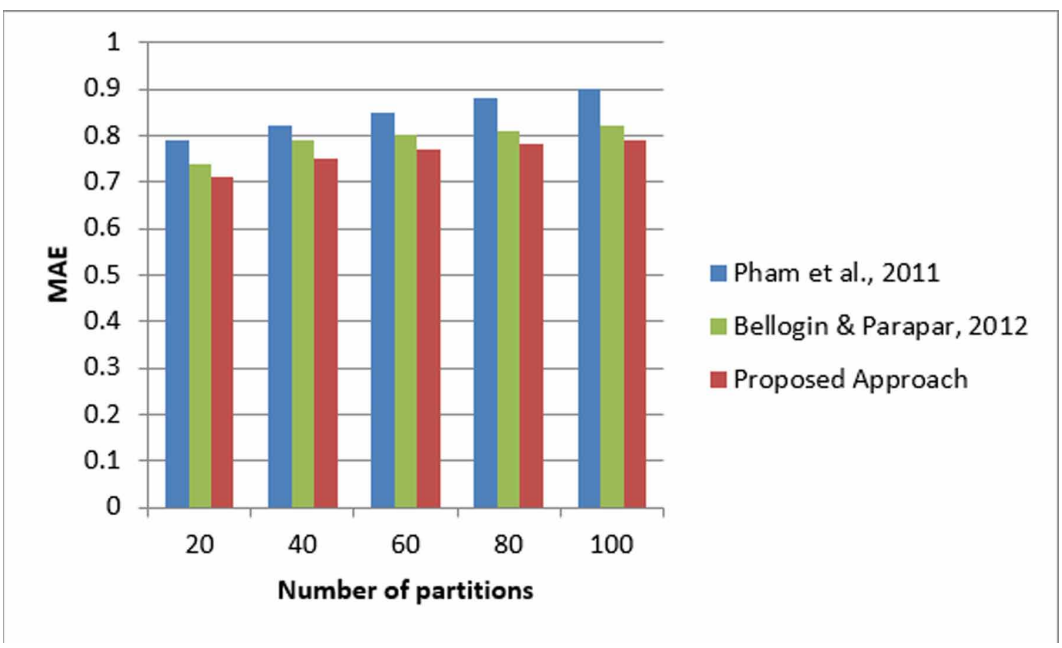
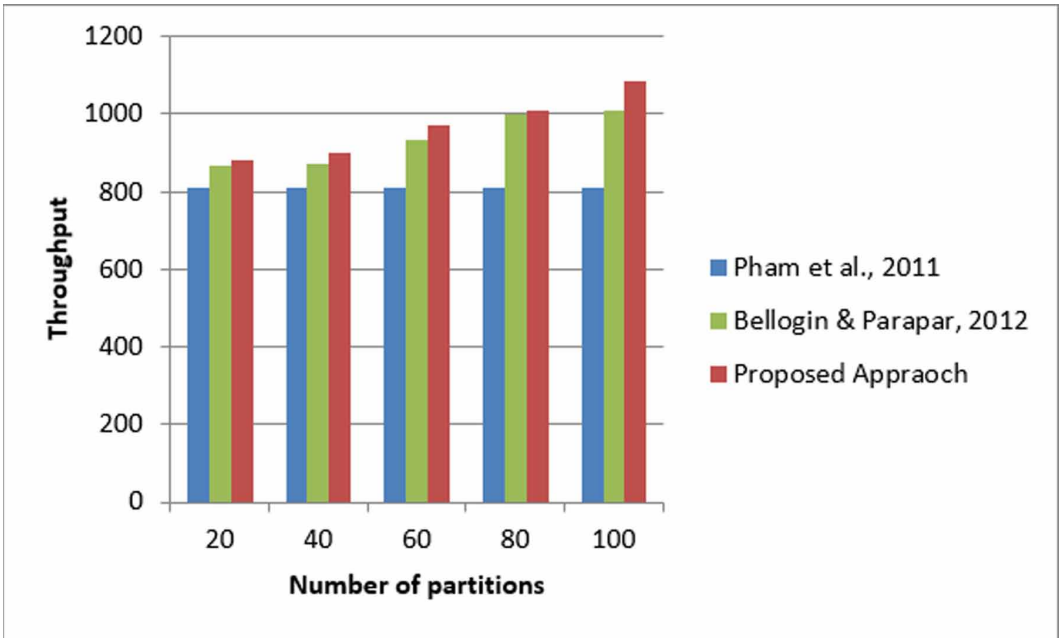


Figure 11. Throughput comparison of collaborative filtering and proposed approach



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 11. MAE and throughput prove that our approach provides better recommendation accuracy.

6. 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 the social graph, so combining social graph and big data result in the social big graph. Social big graph analysis and mining specifically for recommendation is the core motive of our research work. Scalability issue is addressed by using partitioning social graph based on trust and providing local recommendations on sub-graph and combining recommendation for the complete social graph. These sub-graphs can be processed effectively on every node. Pregel and Giraph 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 the accuracy of proposed approach. Experiment analysis proves that proposed approach improves partitioning quality as well as improves recommendation accuracy for large-scale data. In future, scalability will be improved for larger scale of data as compared to Epinions and LiveJournal dataset. Further improvements are required to reduce vertex replication in every subgraph.

REFERENCES

Aiolli, F. (2013). Efficient top-n recommendation for very large scale binary rated datasets. In *Proceedings: ACM Conference on RecSys* (pp. 273-280). ACM Press. doi:10.1145/2507157.2507189

Apache Giraph Project. (n.d.). Retrieved from http://giraph.apache.org

Apache hadoop. (n.d.). Retrieved from http://hadoop.apache.org

Balabanović, M., & Shoham, Y. (1997). Fab: Content-based, collaborative recommendation. *Communications of the ACM*, *40*(3), 66–72. doi:10.1145/245108.245124

Bedi, P., Kaur, H., & Marwaha, S. (2007). Trust based recommender systems for semantic web. *Proceedings of the International Joint Conference on Artificial Intelligence* (pp. 2677-2682). Academic Press.

Bello-Orgaz, G., Jung, J. J., & Camacho, D. (2016). Social Big Data: Recent achievements and new challenges. Information Fusion. *Science Direct*, *28*, 45–59.

Bellogin A, & Parapar J. (2012). Using graph partitioning techniques for neighbour selection in user-based collaborative filtering. In *Proceedings of the ACM conference on Recommender systems* (pp. 213-216). ACM Press.

Chen, R., Shi, J., Zang, B., & Guan, H. (2014, June). Bipartite-oriented distributed graph partitioning for big learning. In *Proceedings of 5th Asia-Pacific Workshop on Systems* (pp. 1-7). Academic Press.

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. doi:10.1016/j.future.2017.02.025

Dean, J., & Ghemawat, S. (2008). MapReduce: Simplified data processing on large clusters. *Communications of the ACM*, *51*(1), 107–113. doi:10.1145/1327452.1327492

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. doi:10.1016/j.future.2017.09.015

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* (pp. 30–36). AAAI Press.

Golbeck, J., & Hendler, J. (2006). Inferring binary trust relationships in Web-based social networks. *ACM Transactions on Internet Technology*, *6*(4), 497–529. doi:10.1145/1183463.1183470

Gonzalez, J. E., Low, Y., Gu, H., Bickson, D., & Guestrin, C. (2012). PowerGraph: distributed graph-parallel computation on natural graphs. In *Proceedings of OSDI*. Academic Press.

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 *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*. AAAI Press.

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* (pp. 135-142). ACM Press. doi:10.1145/1864708.1864736

Jan, N., Ullah, K., Mahmood, T., Garg, H., Davvaz, B., Saeid, A. B., & Broumi, S. (2019). Some root level modifications in interval valued fuzzy graphs and their generalizations including neutrosophic graphs. *Mathematics*, *7*(1), 72. doi:10.3390/math7010072

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. doi:10.1137/S1064827595287997

Koren, Y., Bell, R., & Volinsky, C. (2009). Matrix factorization techniques for recommender systems. *Computer*, *42*(8), 30–37. doi:10.1109/MC.2009.263

Lacic, E., Kowald, D., & Trattner, C. (2014). SocRecM: A scalable social recommender engine for online marketplaces. In *Proceedings of the ACM Conference on Hypertext and social media* (pp. 308-310). ACM Press. doi:10.1145/2631775.2631783

Levin, R., Abassi, H., & Cohen, U. (2016). Guided Walk: A Scalable Recommendation Algorithm for Complex Heterogeneous Social Networks. In *Proceedings of the ACM Conference on RecSys*. ACM Press. doi:10.1145/2959100.2959143

Low, Y., Gonzalez, J., Kyrola, A., Bickson, D., Guestrin, C., & Hellerstein, J. M. (2012). Distributed graphlab: A framework for machine learning in the cloud.

Lumsdaine, A., Gregor, D., Hendrickson, B., & Berry, J. (2007). Challenges in parallel graph processing. *Parallel Processing Letters*, *17*(1), 5–20.

Ma, H., Zhou, D., Liu, C., Lyu, M. R., & 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). ACM Press. doi:10.1145/1935826.1935877

Malewicz, G., Austern, M. H., Bik, A. J. C., Dehnert, J. C., Horn, I., Leiser, N., & Czajkowski, G. (2010). Pregel: A System for Large-Scale Graph Processing. In *Proceedings of the ACM SIGMOD International Conference on Management of data* (pp. 135-146). ACM Press. doi:10.1145/1807167.1807184

Martella, C., Logothetis, D., Loukas, A., & Siganos, G. (2017). Spinner: Scalable Graph Partitioning in the Cloud. In *Proceedings of the IEEE Conference on Data Engineering* (pp. 1083-1094). IEEE Press.

Massa, P., & Avesani, P. (2007). Trust-aware recommender systems. In *Proc. of RecSys '07* (pp. 17–24). Academic Press. doi:10.1145/1297231.1297235

Mateos, C., Zunino, A., & Hirsch, M. (2013). EasyFJP: Providing hybrid parallelism as a concern for divide and conquer Java applications. *Computer Science and Information Systems*, *10*(3), 1129–1163. doi:10.2298/CSIS120712021M

Newman, M. E. J., & Girvan, M. (2004). Finding and evaluating community structure in networks. *Physical Review E: Statistical, Nonlinear, and Soft Matter Physics*, 69(2 Part 2). PubMed PMID:14995526

Ng, A., Jordan, M., & Weiss, Y. (2001). On spectral clustering: Analysis and an Algorithm. *Advances in Neural Information Processing Systems*, *14*, 849–856.

Nvidia. (n.d.). CUDA. Retrieved from https://docs.nvidia.com/cuda/

O'Donovan, J., & Smyth, B. (2005). Trust in recommender systems. In *Proc. of IUI '05* (pp. 167–174). Academic Press. doi:10.1145/1040830.1040870

Onizuka, M., Fujimori, T., & Shiokawa, H. (2017). Graph Partitioning for Distributed Graph Processing. *Data Science and Engineering*, *2*(1), 94–105. doi:10.1007/s41019-017-0034-4

Owen, S., Anil, R., Dunning, T., & Friedman, E. (2011). *Mahout in Action*. Greenwich, CT: Manning Publications Co.

Pham, M. C., Cao, Y., Klamma, R., & Jarke, M. (2011). A clustering approach for collaborative filtering recommendation using social network analysis. *JUCS*, *17*(4), 583–604.

Qian, T., Yang, Y., & Wang, S. (2010, December). Refining graph partitioning for social network clustering. In *Proceedings of the International Conference on Web Information Systems Engineering* (pp. 77-90). Springer.

Resnick, P., & Varian, HR. (1997). Recommender Systems. *Communications of the ACM*, *40*(3), 56-58.

Salihoglu, S., & Widom, J. (2014). Optimizing Graph Algorithms on Pregel-Like Systems. *Proceedings: International Conference on Very Large Data Bases.*, *7*(7), 577–588.

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 (pp. 89–110). Springer. doi:10.1007/978-3-319-90059-9_5

Stanton, I., & Kliot, G (2012). Streaming graph partitioning for large distributed graphs. In *Proceedings of the ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 1222-1230). ACM Press.

Su, X., & Khoshgoftaar, T. M. (2009). A survey of collaborative filtering techniques. In *Advances in artificial intelligence*. Academic Press.

Trustlet. (n.d.). Epinions dataset. Retrieved from http://www.trustlet.org/epinions.html

Valiant, L. G. (1990). A bridging model for parallel computation. *Communications of the ACM*, *33*(8), 103–111. doi:10.1145/79173.79181

Wang, L., Xiao, Y., Shao, B., & Wang, H. (2014). How to partition a billion-node graph. In *Proceedings of the IEEE International Conference on Data Engineering* (pp. 568-579). IEEE Press.

Xin, R. S., Gonzalez, J. E., Franklin, M. J., & Stoica, I. (2013). GraphX: a resilient distributed graph system on Spark. In *Proceeding of GRADES*. Academic Press. doi:10.1145/2484425.2484427

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* (pp. 2747–2753). Academic Press.

Yang, S., Long, B., Smola, A., Sadagopan, N., Zheng, Z., & Zha, H. (2011). Like like alike: joint friendship and interest propagation in social networks. In *Proceedings of the International Conference on World Wide Web* (pp. 537-546). Academic Press. Retrieved from https://snap.stanford.edu/data/soc-LiveJournal1.html

Gourav Bathla is pursuing a PhD from Punjabi University, Punjab, India. He has 12 years of teaching experience. He has completed M.E from Delhi College of Engineering. He is GATE qualified with All India Rank 59. He is an active researcher and published 10 research papers in reputed journals and 5 research papers in International conferences. His areas of interest are Big Data, machine learning, deep learning, data analytics, and programming languages.

Himanshu Aggarwal, Ph.D., is currently serving as Professor in Department of Computer Science & Engineering at Punjabi University, Patiala. He has more than 26 years of teaching experience and served academic institutions such as Thapar Institute of Engineering & Technology, Patiala, Guru Nanak Dev Engineering College, Ludhiana and Technical Teacher's Training Institute, Chandigarh. He is an active researcher who has supervised more than 35 M.Tech. Dissertations and contributed 100 articles in various Research Journals. He is guiding PhD to 8 scholars and nine have completed their PhD. He is on the editorial board of 9 journals and review boards of 5 journals of repute. His areas of interest are software engineering, computer networks, information systems, ERP, and parallel computing.

Rinkle Rani (PhD) is working as an Associate Professor in Computer Science and Engineering Department, Thapar University, Patiala since 2000. She has done her Post graduation from BITS, Pilani and Ph.D. from Punjabi University, Patiala in the area of Computer Networks. She has more than 20 years of teaching experience. She has supervised 44 M.Tech. Dissertations and contributed 58 articles in Conferences and 49 papers in Research Journals. Her areas of interest are Big Data analytics, machine learning, and healthcare informatics. She is a member of professional bodies: ACM, IEEE, ISTE, and CSI.