Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 96 (2016) 682 – 691

Procedia Computer Science

20th International Conference on Knowledge Based and Intelligent Information and Engineering Systems, KES 2016, 5–7 September 2016, York, United Kingdom

Knowledge Discovery from Social Graph Data

Peter Braun[a], Alfredo Cuzzocrea[b], Carson K. Leung[a,*], Adam G.M. Pazdor[a], Kimberly Tran[a]

[a]*Department of Computer Science, University of Manitoba, Winnipeg, MB, R3T 2N2, Canada*
[b]*Department of Engineering and Architecture (DIA), University of Trieste & ICAR-CNR, 34127 Trieste (TS), Italy*

Abstract

High volumes of a wide variety of valuable data can be easily collected and generated from a broad range of data sources of different veracities at a high velocity. In the current era of big data, many traditional data management and analytic approaches may not be suitable for handling the big data due to their well-known 5V's characteristics. Over the past few years, several systems and applications have developed to use cluster, cloud or grid computing to manage and analyze big data so as to support data science (e.g., knowledge discovery and data mining). In this paper, we present a knowledge-based system for social network analysis so as to support big data mining of interesting patterns from big social networks that are represented as graphs.

Peer-review under responsibility of KES International

Keywords: Knowledge discovery and data mining; big data; big data management; big data analysis; graph data, data and knowledge representation; knowledge technologies

1. Introduction and Related Work

In the current era of big data, a wide variety of valuable data can be easily collected and generated from a broad range of data sources of different veracities at a high velocity in various real-life applications (e.g., bioinformatics, sensor and stream systems, smart worlds, Web, social networks[13,14,16,22,23,32]). Moreover, volumes of these big data are also beyond the ability of commonly-used software to manage, query, process, and analyze within a tolerable elapsed time. In general, characteristics of these big data can be described by the well-known 5V's:

1. *value*, which focuses on the usefulness of data (e.g., knowledge that can be discovered from the big data);
2. *variety*, which focuses on differences in types, contents, or formats of data;
3. *velocity*, which focuses on the speed at which data are collected or generated;
4. *veracity*, which focuses on the quality of data (e.g., precise data, uncertain and imprecise data); and
5. *volume*, which focuses on the quantity of data.

* Corresponding author
E-mail address: kleung@cs.umanitoba.ca (C.K. Leung)

Peer-review under responsibility of KES International
doi:10.1016/j.procs.2016.08.250

Embedded in the big data[7,9]—such as web logs, texts, documents, business transactions, banking records, financial charts, biological data, medical images, surveillance videos, as well as streams of advertisements, marketing, telecommunication, life science, and social media data—are rich sets of useful information and knowledge. Due to the aforementioned 5V's characteristics, many traditional data management and analytic approaches may not be suitable for handling the big data. New forms of techniques are needed for managing, querying, processing, and analyzing big data so as to enable enhanced decision making, insight, process optimization, knowledge discovery and data mining. This drives and motivates research and practices in big data management, big data analytics, and data science.

Efficient and effective *management of big data* in distributed environments supports a wide range of data science activities including analytics, cybersecurity, knowledge discovery and data mining. To manage big data, many recent applications and systems use cluster, cloud, or grid computing[17]. Once the big data are managed, they can then be analyzed (e.g., inspected, cleaned, transformed, and modelled) and mined by data science techniques.

Data science, generally, aims to develop systematic or quantitative data mining and analytic algorithm[18] for mining and analyzing big data. *Big data analytics*[1,20,21], in particular, incorporates various techniques from a broad range of fields, which include cloud computing, knowledge discovery and data mining, machine learning, mathematics, as well as statistics. With the 5V's characteristics of big data, it is natural to handle the big data in a distributed computing environment such as a cloud environment because it represents a "natural" context for big data by providing high performance, reliability, availability, transparency, abstraction, and/or virtualization.

Social networks are examples of big data. These networks are generally made of social entities (e.g., individuals, corporations, collective social units, or organizations) that are linked by some specific types of interdependencies (e.g., friendship, kinship, common interest, beliefs, or financial exchange). In these networks, a social entity is connected to another entity as his friend, next-of-kin, collaborator, co-author, classmate, co-worker, team member, and/or business partner. *Social network analysis* applies big data mining and analytics techniques to social networks so as to (i) computationally facilitate social studies and human-social dynamics in these big data networks, as well as (ii) design and use information and communication technologies for dealing with social context.

Over the past few years, several data mining algorithms and techniques[29,30] have been proposed for social network analysis. However, many[24,26,35] of them aim to detect communities by using *clustering* techniques. In contrast, we focus on applying *association rule or frequent pattern mining* techniques to analyze and mine big social graph data for interdependencies or connections among social entities in a big social network.

Our *key contributions* of the paper include the following: We present a knowledge-based system for big data representation, management, and analysis of big data from social networks. We represent such big social data as graph data so as to support knowledge discovery and data mining for interesting patterns from big social networks.

The remainder of this paper is organized as follows. The next section provides some background. Section 3 presents our knowledge-based system for mining and analyzing big social network data represented as a big social graph. Experimental results and conclusions are given in Sections 4 and 5, respectively.

2. Background

In this section, we present some background materials on (i) big data (e.g., big data management) and (ii) social network analysis.

2.1. Big data

High volumes of valuable data (e.g., web logs, texts, documents, business transactions, banking records, financial charts, biological data, medical images, surveillance videos, as well as streams of advertisements, marketing, telecommunication, life science, and social media data[2,8,19,27]) can be easily collected or generated from different data sources, in different formats, and at high velocity in many real-life applications in modern organizations and society. This leads us into the new era of *big data*[28], which refer to a wide variety of valuable data collected and generated from a wide range of data sources of different veracities at a high velocity in various real-life applications and with volumes beyond the ability of commonly-used software to manage, query, process, and analyze within a tolerable

elapsed time. This drives and motivates research and practices in big data management, big data analytics, and data science.

To manage big data, many recent applications and systems that use cluster, cloud or grid computing [17] have been developed. *Cluster computing* [33] involves a group of distributed or parallel computers that are interconnected through high-speed networks such as local area networks. These computers work together as a single computing group to manage, query and process data. *Grid computing* [4,5], on the other hand, can be considered as a form of distributed or parallel computing that coordinates heterogeneous networked loosely coupled computers. Each computer in the grid may perform a different task. *Cloud computing* [3,15] can be considered as another form of distributed or parallel computing. Public, private or hybrid cloud involves a group of interconnected and virtualized computers to provide on-demand services such as infrastructure-as-a-service (IaaS), platform-as-a-service (PaaS) and software-as-a-service (SaaS) [12].

2.2. Social network analysis

In the current era of big data (including big social network data), various social networking sites or services—such as Facebook, Google+, LinkedIn, Twitter, and Weibo [10,31]—are commonly in use. For instance, a LinkedIn user can create a professional profile, establish connections to other LinkedIn users, and exchange messages. In addition, he can also join common-interest user groups and tag his connections (e.g., first-degree, second-degree, and third-degree connections) according to some overlapping categories (e.g., colleagues, classmates, business partners, friends). Moreover, a LinkedIn user A can view another user B's profile, send him messages, endorse his skills, and/or recommend him. When user A joins a common-interest user group, many of his friends may also be interested in joining the group too.

Similarly, a Facebook user C can create a personal profile, add other Facebook users as friends, exchange messages, and join common-interest user groups. Many of these Facebook users are linked to some other Facebook users via their *mutual friendship* (i.e., if a Facebook user C is a friend of another Facebook user D, then user D is also a friend of user C). The number of (mutual) friends may vary from one Facebook user to another. It is not uncommon for a user C to have hundreds or thousands of friends. Some of these friends are more important to user C than others. As a preview, in this paper, we will present a knowledge-based system that mines and analyzes big social graphs so as to discover some interesting knowledge about these friends (in Facebook) or connections (in LinkedIn).

Moreover, although many of the Facebook users are linked to some other Facebook users via their mutual friendship, there are situations in which such a relationship is *not* mutual. To handle these situations, Facebook added the functionality of "follow", which allows a Facebook user to subscribe or follow public postings of some other Facebook users without the need of adding them as friends. So, for any user C, if many of his friends followed some individual users or groups of users, then user C might also be interested in following the same individual users or groups of users. Furthermore, the "like" button allows users to express their appreciation of content such as status updates, comments, photos, and advertisements.

As another instance, a Twitter user can read the tweets of other users by "following" them. Relationships between social entities are mostly defined by following (or subscribing) each other. Each user (social entity) can have multiple followers, and can follow multiple users at the same time. The *follow/subscribe relationship* between follower and followee is not the same as the friendship (in which each pair of users usually know each other before they setup the friendship). In contrast, in the follow/subscribe relationship, a user E can follow another user F while user F may not know user E in person. We use E→F to represent the follow/subscribe (i.e., "following") relationship that user E is following user F.

In recent years, the number of users in these social networking sites has grown rapidly (e.g., more than 433 million registered Linked users [25], 1.65 billion monthly active Facebook users [11], and 310 million monthly active Twitter users [34] at the end of first quarter of 2016). These big numbers of users in social networks create an even more massive number of linkages (e.g., connections, friendships, follow/subscribe relationships) among users. Hence, having a knowledge-based system for mining and analyzing big social graphs for the discovery of some interesting knowledge (e.g., popular users) about these users is desirable. To elaborate, discovery of popular users helps an individual user new to the network to make connection or follow the same popular users. Moreover, many businesses have used social network media to either (i) reach the right audience and turn them into new customers or (ii) build

a closer relationship with existing customers. Hence, discovery of customers who follow or subscribe to products or services provided by a business helps the business identify its targeted or preferred customers.

3. Our Graph-Based Social Network Analysis System

In this section, we present our knowledge-based system for social network analysis on big social data, which are represented in the form of a *big social graph*. Such a graph representation supports data science—in particular, big data mining and analytics—for the discovery of interesting patterns from big social networks.

3.1. Social network analysis on directed social graphs capturing follow/subscribe relationships

In social networking sites like Twitter and Google+, social entities (users) are linked by the *follow/subscribe* (*i.e., "following"*) *relationships* such that a user A (i.e., *follower*) follows another user B (i.e., *followee*), which can be denoted as A→B. Moreover, recall from Section 2.2 that, in addition to the usual "add friend" feature, Facebook also provides users with the "follow" feature. Hence, social entities in Facebook can also be linked by the follow/subscribe relationships too. Note that these follow/subscribe relationships are directional. Consider Scenario 1.

Scenario 1. For an illustrative purpose, let us consider a small portion of a big social network. Here, there are $|V|$=12 users (Albert, Betty, Charles, Doris, Ed, Fiona, George, Helen, Ivan, Jane, Ken, and Lisa). Each user is following some others as described below:

- Albert is following Betty.
- Betty is following Albert and Charles.
- Charles is following Albert and Ivan.
- Doris is following Albert, Betty, Charles and Ed.
- Ed is following Albert, Betty, Charles and Doris.
- Fiona is following Ed and George.
- George is following Fiona.
- Helen is following George.
- Ivan is following Lisa.
- Jane is following Ivan.
- Ken is *not* following anyone.
- Lisa is following Charles, Ivan and Ken. □

We represent these big social network data in Scenario 1 by using a *directed graph* $G = (V, E)$, where

1. each node/vertex $v \in V$ represents a user (i.e., a social entity) in the social network, and
2. each directed edge $e=(u, v) \in E$ represents the follow/subscribe relationship between a pair of users $u, v \in V$ such that user u (i.e., follower) is "following" user v (i.e., followee).

The arrow on an edge represents the "following" direction. For instance, a directed arrow "Betty→Charles" represents that Betty is following Charles on a social networking site. In contrast, a bi-directed arrow "Albert↔Betty" represents that Albert and Betty are following each other (i.e., Albert is following Betty, and Betty is following Albert) on a social networking site. See Fig. 1.

Let us consider the space requirements for this directed graph representation of big social network data. Theoretically, given $|V|$ social entities, there are potentially $|V| \cdot (|V| - 1)$ directed edges for follow/subscribe relationships. Practically, the number of directed edges is usually lower than its maximum $|V| \cdot (|V| - 1)$ unless for the extreme case where everyone is following everyone in a social network. In Fig. 1, there are only $|E|$=22 directed edges (cf. possibly 132 edges for $|V|$=12 users), where E = {(Albert, Betty), (Betty, Albert), (Betty, Charles), (Charles, Albert), (Charles, Ivan), (Doris, Albert), (Doris, Betty), (Doris, Charles), (Doris, Ed), (Ed, Albert), (Ed, Betty), (Ed, Charles), (Ed, Doris), (Fiona, Ed), (Fiona, George), (George, Fiona), (Helen, George), (Ivan, Lisa), (Jane, Ivan), (Lisa, Charles),

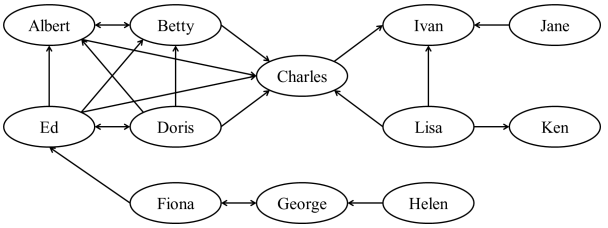

Fig. 1. A directed social graph capturing follow/subscribe relationships in Scenario 1.

(Lisa, Ivan), (Lisa, Ken)} for V = {Albert, Betty, Charles, Doris, Ed, Fiona, George, Helen, Ivan, Jane, Ken, Lisa}. With this directed graph representation of big social network data, our graph-based social network analysis system can easily answer the following questions.

Question 1. *Who are the most popular followees?* The most popular followees can be found by first counting the number of incoming edges of every node (i.e., the number of followers of every social entity v) and then picking those with the highest number of incoming edges. Let $\#followers(v) = |\{u \in V \mid (u, v) \in E\}|$. Then, $\text{argmax}_{v \in V}\ \#followers(v)$ gives the answer.

Example 1. For Scenario 1, *both Albert* (who is followed by Betty, Charles, Doris and Ed) *and Charles* (who is followed by Betty, Doris, Ed and Lisa) *are the two most popular followees*. This answer is supported by the four incoming edges pointing to the node of Albert and of Charles in Fig. 1, i.e., #*followers*(Albert) = 4 and #*followers*(Charles) = 4 indicating that Albert and Charles are each followed by four followers. □

Question 2. *Who are the most active followers?* The most active followers can be found by first counting the number of outgoing edges of every node (i.e., the number of followees of every social entity u) and then picking those with the highest number of outgoing edges. Let $\#followees(u) = |\{v \in V \mid (u, v) \in E\}|$. Then, $\text{argmax}_{u \in V}\ \#followees(u)$ gives the answer.

Example 2. For Scenario 1, *both Doris* (who is following Albert, Betty, Charles and Ed) *and Ed* (who is following Albert, Betty, Charles and Doris) *are the two most active followers*. This answer is supported by the four outgoing edges pointing from the node of Doris and of Ed in Fig. 1, i.e., #*followees*(Doris) = 4 and #*followees*(Ed) = 4 indicating that Doris and Ed are each following four followees. □

3.2. Social network analysis on bi-directed or undirected social graphs capturing mutual friendships

In social networking sites like Facebook and LinkedIn, social entities (users) are usually linked by the *mutual friendships* such that a user A is a friend (or first-degree connection) of another user B meaning that user B is also a friend of user A. Such a mutual friendship can be denoted as A↔B. Unlike those directional follow/subscribe relationships described in Section 3.1, the mutual friendships are bi-directional. Consider Scenario 2.

Scenario 2. For an illustrative purpose, let us reconsider the same $|V|$=12 users (Albert, Betty, Charles, Doris, Ed, Fiona, George, Helen, Ivan, Jane, Ken, and Lisa) as in Scenario 1. However, each user is a friend of some others as described below:

- Albert is a friend of Betty, Charles, Doris and Ed.
- Betty is a friend of Albert, Charles, Doris and Ed.
- Charles is a friend of Albert, Betty, Doris, Ed, Ivan and Lisa.
- Doris is a friend of Albert, Betty, Charles and Ed.
- Ed is a friend of Albert, Betty, Charles, Doris and Fiona.
- Fiona is a friend of Ed and George.
- George is a friend of Fiona and Helen.

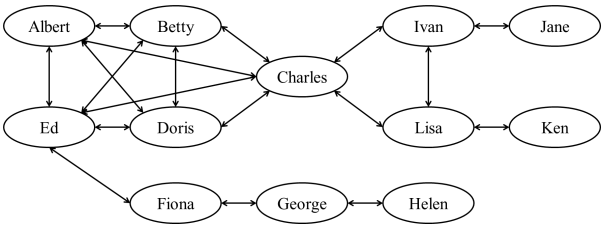

Fig. 2. A bi-directed social graph capturing mutual friendships in Scenario 2.

- Helen is a friend of George.
- Ivan is a friend of Charles, Jane and Lisa.
- Jane is a friend of Ivan.
- Ken is a friend of Lisa.
- Lisa is a friend of Charles, Ivan and Ken. □

These big social network data in Scenario 2 can be represented by using a *bi-directed graph* $G = (V, E)$, where

1. each node/vertex $v \in V$ represents a user (i.e., a social entity) in the social network; and
2. as each directed edge $e=(u, v) \in E$ represents the follow/subscribe relationship between a pair of users $u, v \in V$ such that user u is following user v, a pair of directed edges (u, v) and (v, u) represents the mutual friendship between a pair of users $u, v \in V$ such that users u and v are mutual friends.

A bi-directional edge represents the mutual friendships. For instance, a bi-directed arrow "Albert↔Betty" represents that Albert and Betty are mutual friends. See Fig. 2.

Let us consider the space requirements for this bi-directed graph representation of big social network data. Theoretically, given $|V|$ social entities, there are potentially $|V| \cdot (|V| - 1)$ directed edges for mutual friendships. Practically, the number of edges is usually lower than its maximum $|V| \cdot (|V| - 1)$ unless for the extreme case where everyone is a friend of everyone in a social network. In Fig. 2, there are only $|E|$=36 directed edges (cf. possibly 132 edges for $|V|$=12 users), where E = {(Albert, Betty), (Albert, Charles), (Albert, Doris), (Albert, Ed), (Betty, Albert), (Betty, Charles), (Betty, Doris), (Betty, Ed), (Charles, Albert), (Charles, Betty), (Charles, Doris), (Charles, Ed), (Charles, Ivan), (Charles, Lisa), (Doris, Albert), (Doris, Betty), (Doris, Charles), (Doris, Ed), (Ed, Albert), (Ed, Betty), (Ed, Charles), (Ed, Doris), (Ed, Fiona), (Fiona, Ed), (Fiona, George), (George, Fiona), (George, Helen), (Helen, George), (Ivan, Charles), (Ivan, Jane), (Ivan, Lisa), (Jane, Ivan), (Ken, Lisa), (Lisa, Charles), (Lisa, Ivan), (Lisa, Ken)} for V = {Albert, Betty, Charles, Doris, Ed, Fiona, George, Helen, Ivan, Jane, Ken, Lisa}.

An observant reader may notice that the mutual friendships are symmetric in nature. Hence, for space efficiency, we do not need to use bi-directed edges. Instead, we can use undirected edges so that an edge A—B indicates a user A is a mutual friend (or first-degree connection) of another user B, and vice versa. Consequently, we represent the big social network data in Scenario 2 by using an *undirected graph* $G = (V, E)$, where

1. each node/vertex $v \in V$ represents a user (i.e., a social entity) in the social network, and
2. each undirected edge $e=(u, v) \in E$ represents the mutual friendship between a pair of users $u, v \in V$ such that users u and v are mutual friends.

See Fig. 3.

With this undirected graph representation of big social network data, the potential number of edges is reduced by half from $|V| \cdot (|V| - 1)$ required by bi-directed graphs to $\frac{|V| \cdot (|V|-1)}{2}$ required by undirected graphs. For example, there are only $|E|$=18 edges in Fig. 3 (cf. $|E|$=36 edges in the bi-directed graph in Fig. 2). An advantage of this undirected graph representation of big social network data is space efficiency. Moreover, another advantage is that, with this undirected graph representation, our graph-based social network analysis system can easily answer the following questions.

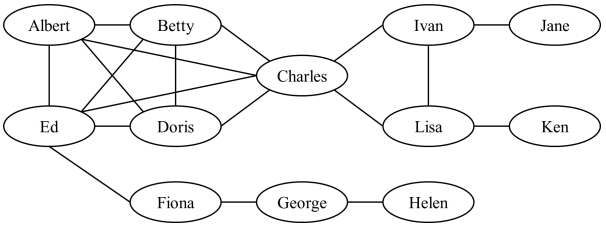

Fig. 3. An undirected social graph capturing mutual friendships in Scenario 2.

Question 3. *Who is the most popular user?* The most popular user can be found by first counting the number of incoming and outgoing edges of every node (i.e., the number of friends, or first-degree connection, of every social entity v) and then picking the one with the highest number of edges. Let $\#friends(v) = |\{u \in V \mid (u, v) \in E \ \vee \ (v, u) \in E\}|$. Then, $\text{argmax}_{v \in V}\ \#friends(v)$ gives the answer.

Example 3. For Scenario 2, *Charles* (who is a friend of Albert, Betty, Doris, Ed, Ivan and Lisa) *is the most popular user*. This answer is supported by the six incoming and outgoing edges pointing to/from the node of Charles in Fig. 3, i.e., $\#friends(\text{Charles}) = 6$ indicating that Charles has six friends . □

Question 4. *Who have the highest number of friends-of-friends* (*or second-degree connections*)*?* This question can be answered by first finding all second-degree connections of every node u and then picking those with the highest numbers of second-degree connections. Let $shortestDist(u, v)$ return the distance of the shortest path between nodes u and v in G (computed by algorithms like Dijkstra's algorithm[6]). Then, $\text{argmax}_{u \in V}\ |\{v \in V \mid shortestDist(u, v) = 2\}|$ gives the answer.

Example 4. For Scenario 2, *Fiona, Ivan and Lisa all have the highest number of friends-of-friends* (*or second-degree connections*). This answer is supported by the five second-degree connections of Fiona, Ivan or Lisa. Specifically,

- $shortestDist(\text{Fiona}, v) = 2$ for each $v \in \{\text{Albert, Betty, Charles, Doris, Helen}\}$ because Albert is a friend of Ed, who happens to be a friend of Fiona. Similarly, Betty, Charles and Doris are friends of Ed, who happens to be a friend of Fiona. Moreover, Helen is a friend of George, who happens to be another friend of Fiona.
- Similarly, $shortestDist(\text{Ivan}, v)=2$ for each $v \in \{\text{Albert, Betty, Doris, Ed, Ken}\}$. On the one hand, Albert, Betty, Doris and Ed are friends of Charles, who happens to be a friend of Ivan. On the other hand, Ken is a friend of Lisa, who happens to be a friend of Ivan.
- $shortestDist(\text{Lisa}, v)=2$ for each $v \in \{\text{Albert, Betty, Doris, Ed, Jane}\}$. On the one hand, Albert, Betty, Doris and Ed are friends of Charles, who happens to be a friend of Lisa. On the other hand, Jane is a friend of Ivan, who happens to be a friend of Lisa. □

Question 5. *Who have the highest number of k^{th}-degree connections?* This question can be answered by first finding all k^{th}-degree connections of every node u and then picking those with the highest numbers of k^{th}-degree connections. This question can be considered as an extension or generalization to Question 4. Conversely, Questions 3 and 4 can be considered as special cases of the current question where k=1 and k=2, respectively. Let $shortestDist(u, v)$ return the distance of the shortest path between nodes u and v in G. Then, $\text{argmax}_{u \in V}\ |\{v \in V \mid shortestDist(u, v) = k\}|$ gives the answer.

Example 5. For Scenario 2, *both Jane and Ken have the highest number of third-degree connections*. This answer is supported by the five third-degree connections of Jane or Ken. Specifically,

- $shortestDist(\text{Jane}, v)=3$ for each $v \in \{\text{Albert, Betty, Doris, Ed, Ken}\}$ because Albert, Betty, Doris and Ed are friends of Charles, who happens to be a friend of Ivan, who in turn is a friend of Jane. As for Ken, he is a friend of Lisa who also happens to be a friend of Ivan.

- *shortestDist*(Ken, v)=3 for each $v \in$ {Albert, Betty, Doris, Ed, Jane} because Albert, Betty, Doris and Ed are friends of Charles, who happens to be a friend of Lisa, who in turn is a friend of Ken. As for Jane, she is a friend of Ivan who also happens to be a friend of Lisa.

For Scenario 2, *Helen has the highest number of fourth-degree connections*. This answer is supported by the four fourth-degree connections of Helen. Specifically, *shortestDist*(Helen, v)=4 for each $v \in$ {Albert, Betty, Charles, Doris} because Albert, Betty, Charles and Doris are friends of Ed, who happens to be a friend of Fiona, who in turn is a friend of George. George is a friend of Helen.

For Scenario 2, *George and Helen have the highest number of fifth-degree connections*. This answer is supported by the two fifth-degree connections of George or Helen. Specifically,

- *shortestDist*(George, v)=5 for each $v \in$ {Jane, Ken} because Jane is a friend of Ivan, whereas Ken is a friend of Lisa. Both Ivan and Lisa are friends of Charles, who happens to be a friend of Ed, who in turn is a friend of Fiona. Fiona is a friend of George.
- *shortestDist*(Helen, v)=5 for each $v \in$ {Ivan, Lisa} because Ivan and Lisa are both friends of Charles, who is a friend of Ed. Ed happens to be a friend of Fiona, who in turn is a friend of George. George is a friend of Helen.

For Scenario 2, *Helen has the highest number of sixth-degree connections*. This answer is supported by the two sixth-degree connections of Helen. Specifically, *shortestDist*(Helen, v)=6 for each $v \in$ {Jane, Ken} because Jane is a friend of Ivan, whereas Ken is a friend of Lisa. Both Ivan and Lisa are friends of Charles, who happens to be a friend of Ed, who in turn is a friend of Fiona. Fiona is a friend of George, who is a friend of Helen. □

Question 6. *Who are the most isolated users?* The most isolated users can be found by first computing all k^{th}-degree connections of every node and then picking those with the highest k, i.e., $\text{argmax}_{u \in V} \textit{shortestDist}(u, v)$ gives the answer.

Example 6. For Scenario 2, *Helen, Jane and Ken are the most isolated users*. This answer is supported by the existence of six-degree connections of Helen, Jane and Ken, i.e., *shortestDist*(Helen, Jane)=6 and *shortestDist*(Helen, Ken)=6 in Example 5. □

4. Experimental Results

To evaluate the performance of our knowledge-based system by using the following two real-life social network datasets:

1. The Stanford Network Analysis Project (SNAP) ego-Twitter dataset, which contains 81,306 social entities and 1,768,149 follow/subscribe relationships among these social entities; and
2. the SNAP ego-Facebook dataset, which contains 4,039 social entities and 88,234 mutual friendships among these social entities.

These two datasets were downloaded from `http://snap.stanford.edu/data/`. All experiments were run using either

1. a single machine with an Intel Core i7 4-core processor (1.73 GHz) and 8 GB of main memory running a 64-bit Windows 7 operating system, or
2. the Amazon Elastic Compute Cloud (EC2) cluster—specifically, 11 High-Memory Extra Large (m2.xlarge) computing nodes (http://aws.amazon.com/ec2/).

We implemented our knowledge-based system—which mines and analyzes big social graphs—in the Java programming language. The stock version of Apache Hadoop 0.20.0 was used. With it, the big social graph data are divided into several partitions and assigned to different processors. Each processor executes the map and reduce functions. Once the data are properly partitioned and assigned to each processor, the processor handles the assigned data without reliance on the results from other processors.

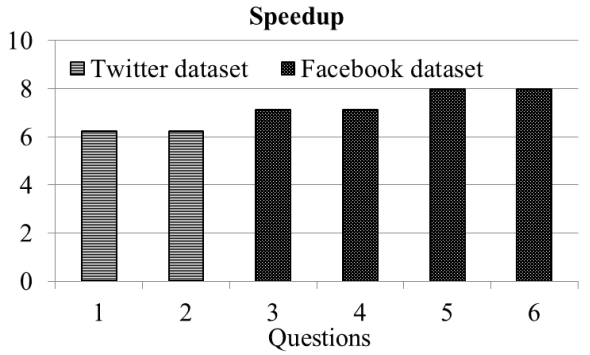

Fig. 4. Experimental results on SNAP ego-Twitter and ego-Facebook datasets

The results shown in Fig. 4 are based on the average of multiple runs. Runtime includes CPU and I/Os. In particular, Fig. 4 shows that the use of our knowledge-based system running on the cloud cluster to conduct social network analysis on big graph data led to a speedup of above 6 times when compared with that running on a single machine for the SNAP ego-Twitter dataset when answering Questions 1 and 2. Fig. 4 also shows that the use of our knowledge-based system running on the cloud cluster to conduct social network analysis on big graph data led to a speedup of around 7 to 8 times when compared with that running on a single machine for the SNAP ego-Facebook dataset when answering Questions 3–6.

Higher speedup is expected when using more processors. Moreover, our knowledge-based system is also shown to be scalable with respect to the number of social entities in the big social network. As ongoing work, we are conducting more experiments, including an in-depth study on the quality of our system in supporting data science, big data management, big data analytics, knowledge discovery and data mining.

5. Conclusions

High volumes of a wide variety of valuable data can be easily collected and generated from a broad range of data sources of different veracities at a high velocity. In the current era of big data, many traditional data management and analytic approaches may not be suitable for handling the big data due to their well-known 5V's characteristics. Over the past few years, several systems and applications have developed to use cluster, cloud or grid computing to manage and analyze big data so as to support data science (e.g., knowledge discovery and data mining). In this paper, we presented a knowledge-based system for social network analysis so as to support big data mining of interesting patterns from big social networks that are represented as graphs. In particular, our system conducts social network analysis on (i) directed graphs capturing follow/subscribe (i.e., "following") relationships (e.g., in Twitter) as well as (ii) bi-directed and undirected graphs capturing mutual friendships (e.g., in LinkedIn, Facebook). Experimental results show effectiveness of our system for social network analysis in support data science (e.g., knowledge discovery and data mining) of big social network data that are represented as graphs.

Acknowledgment

This project is partially supported by Natural Sciences and Engineering Research Council of Canada (NSERC) and the University of Manitoba.

References

1. Agrawal D, Chawla S, Elmagarmid AK, Kaoudi Z, Ouzzani M, Papotti P, Quiané-Ruiz JA, Tang N, Zaki M. Road to freedom in big data analytics. In: *Proceedings of the EDBT 2016*. OpenProceedings.org; 2016, p. 479–484.

2. Braun P, Cameron JJ, Cuzzocrea A, Jiang F, Leung CK. Effectively and efficiently mining frequent patterns from dense graph streams on disk. *Procedia Computer Science* 2014; **35**: 338–347.
3. Buyya R, Yeo CS, Venugopal S, Broberg J, Brandic I. Cloud computing and emerging IT platforms: vision, hype, and reality for delivering computing as the 5th utility. *Future Generation Computer Systems* 2009; **25**(6):599–616.
4. Chen J, Yang Y. Grid and workflows. In: *Encyclopedia of Database Systems*, Springer; 2009, p. 1276–1279.
5. Chetty M, Buyya R. Weaving computational grids: how analogous are they with electrical grids? *Computing in Science & Engineering* 2002; **4**(4):61–71.
6. Cormen TH, Leiserson CE, Rivest RL, Stein C. *Introduction to Algorithms, 3rd ed.* Cambridge, MA, USA: MIT Press; 2009.
7. Cuzzocrea A, Lee W, Leung CK. High-recall information retrieval from linked big data. In: *Proceedings of the IEEE COMPSAC 2015, vol. 2*, IEEE; 2015, p. 712–717.
8. Cuzzocrea A, Han Z, Jiang F, Leung CK, Zhang H. Edge-based mining of frequent subgraphs from graph streams. *Procedia Computer Science* 2015; **60**:573–582.
9. Cuzzocrea A, Leung CK. Upper bounds to expected support for frequent itemset mining of uncertain big data. In: *Proceedings of the ACM SAC 2015*, ACM; 2015, p. 919–921.
10. Ellis K, Goldszmidt M, Lanckriet GRG, Mishra N, Reingold O. Equality and social mobility in Twitter discussion groups. In: *Proceedings of the ACM WSDM 2016*. ACM; 2016, p. 523–532.
11. Facebook newsroom - company info. http://newsroom.fb.com/company-info/
12. Han Z, Leung CK. FIMaaS: scalable frequent pattern mining-as-a-service on cloud for non-expert miners. In: *Proceedings of the BigDAS 2015*, ACM; 2015, p. 84–91.
13. Jiang F, Leung CK, Liu D. Efficiency improvements in social network communication via MapReduce. In: *Proceedings of the IEEE DSDIS 2015*. IEEE; 2015, p. 161–168.
14. Jin X, Zong S, Li YJ, Wu S, Yin W, Ge W. A domain knowledge based method on active and focused information service for decision support within big data environment. *Procedia Computer Science* 2015; **60**:93–102.
15. Kaaouache MA, Bouamama S. Solving bin packing problem with a hybrid genetic algorithm for VM placement in cloud. *Procedia Computer Science* 2015; **60**:1061–1069.
16. Kawagoe K, Leung CK., Similarities of frequent following patterns and social entities. *Procedia Computer Science* 2015; **60**:642–651.
17. Le T, Anciaux N, Guilloton S, Lallali S, Pucheral P, Sandu Popa I, Chen C. Distributed secure search in the personal cloud. In: *Proceedings of the EDBT 2016*. OpenProceedings.org; 2016, p. 652–655.
18. Leung CK. Big data mining applications and services. In: *Proceedings of the BigDAS 2015*. ACM; 2015, p. 1–8.
19. Leung CK, Cuzzocrea A. Frequent subgraph mining from streams of uncertain data. In: *Proceedings of the C3S2E 2015*. ACM; 2015, p. 18–27.
20. Leung CK, Hayduk Y. Mining frequent patterns from uncertain data with MapReduce for big data analytics. In: *Proceedings of the DASFAA 2013, Part I*. Springer; 2013, p. 440–455.
21. Leung CK, Jiang F. Big data analytics of social networks for the discovery of 'following' patterns. In: *Proceedings of the DaWaK 2015*. Springer; 2015, p. 123–135.
22. Leung CK, Jiang F, Pazdor AGM, Peddle AM. Parallel social network mining for interesting 'following' patterns. *Concurrency and Computation: Practice and Experience* 2016. DOI:10.1002/cpe.3773
23. Leung CK, Tanbeer SK, Cuzzocrea A, Braun P, MacKinnon RK. Interactive mining of diverse social entities. *International Journal of Knowledge-based and Intelligent Engineering Systems* 2016; **20**(2):97–111.
24. Lin W, Kong X, Yu PS, Wu Q, Jia Y, Li C. Community detection in incomplete information networks. In: *Proceedings of the WWW 2012*, p. 341-350.
25. LinkedIn newsroom - about us. https://press.linkedin.com/about-linkedin
26. Ma L, Huang H, He Q, Chiew K, Wu J, Che Y. GMAC: a seed-insensitive approach to local community detection. In: *Proceedings of the DaWaK 2013*, p. 297-308.
27. MacKinnon RK, Leung CK. Stock price prediction in undirected graphs using a structural support vector machine. In: *Proceedings of IEEE/WIC/ACM WI-IAT 2015, vol. 1*. IEEE; 2015, p. 548–555.
28. Madden S. From databases to big data. *IEEE Internet Computing* 2012; **16**(3):4–6.
29. Nguyen-Thi AT, Nguyen PQ, Ngo TD, Nguyen-Hoang TA. Transfer AdaBoost SVM for link prediction in newly signed social networks using explicit and PNR features. *Procedia Computer Science* 2015; **60**:332–341.
30. Podobnik V, Lovrek I. Implicit social networking: discovery of hidden relationships, roles and communities among consumers. *Procedia Computer Science* 2015; **60**:583–592.
31. Rader E, Gray R. Understanding user beliefs about algorithmic curation in the Facebook news feed. In: *Proceedings of the ACM CHI 2015*, ACM; 2015; p. 173–182.
32. Rahman QM, Fariha A, Mandal A, Ahmed CF, Leung CK. A sliding window-based algorithm for detecting leaders from social network action streams. In: *Proceedings of the IEEE/WIC/ACM WI-IAT 2015, vol. 1*. IEEE; 2015, p. 133–136.
33. Rosà A, Chen LY, Binder W. Predicting and mitigating jobs failures in big data clusters. In: *Proceedings of the IEEE/ACM CCGrid 2015*. IEEE; 2015, p. 221–230.
34. Twitter - about company. https://about.twitter.com/company
35. Wei EHC, Koh YS, Dobbie G. Finding maximal overlapping communities. In: *Proceedings of the DaWaK 2013*. Springer; 2013, p. 309-316.