ISSN 2278-3091

International Journal of Advanced Trends in Computer Science and Engineering, Vol. 3 , No.1, Pages : 122– 127 (2014)
Special Issue of ICETETS 2014 - Held on 24-25 February, 2014 in Malla Reddy Institute of Engineering and Technology, Secunderabad– 14, AP, India

ENHANCING DATA PRIVACY IN DATA EXTRACTION WITH BIG DATA

MELWIN DEVASSY[1], GERA.PRAVEEN KUMAR[2]
[1]*Student, Malla Reddy Institute of Engineering and Technology,melwindevassy@gmail.com* ,
[2]*Asst. Professor, Malla Reddy Institute of Engineering and Technology, praveenger@gmail.com*

Abstract

Big Data concerns large-volume, complex, growing data sets with multiple, autonomous sources. With the fast development of networking, data storage, and the data collection capacity, Big Data is now rapidly expanding in all science and engineering domains, including physical, biological and bio-medical sciences. In this article we concentrate on attacking one of the issues in big data which is Data Privacy. We analyze the one of the techniques that can be used for addressing the Data Privacy concern (SQL Implementation).

1. Introduction

Big Data is a new term used to identify the datasets that due to their large size and complexity, we can not manage them with our current methodologies or data mining software tools. Big Data mining is the capability of extracting useful information from these large datasets or streams of data, that due to its volume, variability, and velocity, it was not possible before to do it. The Big Data challenge is becoming one of the most exciting opportunities for the next years.

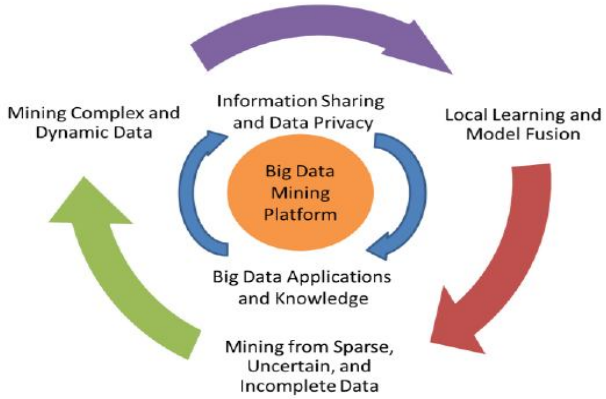

Figure 1: A Big Data processing framework

2 Data Mining Challenges with Big Data

For an intelligent learning database system to handle Big Data, the essential key is to scale up to the exceptionally large volume of data. Figure 1 shows a conceptual view of the Big Data processing framework, which includes three tiers from inside out with considerations on data accessing and computing (Tier I), data privacy and domain knowledge (Tier II), and Big Data mining algorithms (Tier III).

The challenges at Tier I focus on data accessing and actual computing procedures. Because Big Data are often stored at different locations and data volumes may continuously grow, an effective computing platform will have to take distributed large-scale data storage into consideration for computing.The challenges at Tier II center around semantics and domain knowledge for different Big Data applications. Such information can provide additional benefits to the mining process, as well as add technical barriers to the Big Data access (Tier I) and mining algorithms (Tier III). At Tier III, the data mining challenges concentrate on algorithm designs in tackling the difficulties raised by the Big Data volumes, distributed data distributions, and by complex and dynamic data characteristics. The circle at Tier III contains three stages.

2.1 Tier I: Big Data Mining Platform

In typical data mining systems, the mining procedures require computational intensive computing units for data analysis and comparisons. A computing platform is therefore needed to have efficient access to, at least, two types of resources: data and computing processors. For small scale data mining tasks, a single desktop computer, which contains hard disk and CPU processors, is sufficient to fulfill the data mining goals. Indeed, many data mining algorithm are designed to handle this type of problem settings. For medium scale data mining tasks, data are typically large (and possibly distributed) and cannot be fit into the main memory. Common solutions are to rely on parallel computing (Shafer et al. 1996; Luo et al. 2012) or collective mining (Chen et al. 2004) to sample and aggregate data from different sources and then use parallel computing programming (such as the Message Passing Interface) to carry out the mining process.

For Big Data mining, because data scale is far beyond the capacity that a single personal computer (PC) can handle, a typical Big Data processing framework will rely on cluster computers with a high performance computing platform, where a data mining task is deployed by running some parallel programming tools, such as MapReduce or ECL (Enterprise Control Language), on a large number of computing nodes (*i.e.*, clusters). The role of the software component is to make sure that a single data mining task, such as finding the best match of a query from a database with billions of samples, is split into many small tasks each of which is running on one or multiple computing nodes. For example, as of this writing, the world most powerful super computer Titan, which is deployed at Oak Ridge National Laboratory in Tennessee, USA, contains 18,688 nodes each with a 16-core CPU.

Such a Big Data system, which blends both hardware and software components, is hardly available without key industrial stockholders' support. In fact, for decades, companies have been making business decisions based on transactional data stored in relational databases. Big Data mining offers opportunities to go beyond their relational databases to rely on less structured data: weblogs, social media, email, sensors, and photographs that can be mined for useful information. Major business intelligence companies, such IBM, Oracle, Teradata etc., have all featured their own products to help customers acquire and organize these diverse data sources and coordinate with customers' existing data to find new insights and capitalize on hidden relationships.

ISSN 2278-3091

ternational Journal of Advanced Trends in Computer Science and Engineering, Vol. 3 , No.1, Pages : 122– 127 (2014)
ecial Issue of ICETETS 2014 - Held on 24-25 February, 2014 in Malla Reddy Institute of Engineering and Technology, Secunderabad– 14, AP, India

2.2 Tier II: Big Data Semantics and Application Knowledge

Semantics and application knowledge in Big Data refer to numerous aspects related to the regulations, policies, user knowledge, and domain information. The two most important issues at this tier include (1) data sharing and privacy; and (2) domain and application knowledge. The former provides answers to resolve concerns on how data are maintained, accessed, and shared; whereas the latter focuses on answering questions like "what are the underlying applications ?" and "what are the knowledge or patterns users intend to discover from the data ?".

2.2.1 Information Sharing and Data Privacy

Information sharing is an ultimate goal for all systems involving multiple parties (Howe et al. 2008). While the motivation for sharing is clear, a real-world concern is that Big Data applications are related to sensitive information, such as banking transactions and medical records, and so simple data exchanges or transmissions do not resolve privacy concerns (Duncan 2007, Huberman 2012, Schadt 2012). For example, knowing people's locations and their preferences, one can enable a variety of useful location-based services, but public disclosure of an individual's movements over time can have serious consequences for privacy.

To protect privacy, two common approaches are to (1) restrict access to the data, such as adding certification or access control to the data entries, so sensitive information is accessible by a limited group of users only, and (2) anonymize data fields such that sensitive information cannot be pinpointed to an individual record (Cormode and Srivastava 2009). For the first approach, common challenges are to design secured certification or access control mechanisms, such that no sensitive information can be misconducted by unauthorized individuals. For data anonymization, the main objective is to inject randomness into the data to ensure a number of privacy goals. For example, the most common k-anonymity privacy measure is to ensure that each individual in the database must be indistinguishable from k−1 others.

Common anonymization approaches are to use suppression, generalization, perturbation, and permutation to generate an altered version of the data, which is, in fact, some uncertain data.One of the major benefits of the data annomization based information sharing approaches is that, once anonymized, data can be freely shared across different parties without involving restrict access controls. This naturally leads to another research area namely privacy preserving data mining (Lindell and Pinkas 2000), where multiple parties, each holding some sensitive data, are trying to achieve a data mining goal without sharing any sensitive information inside the data.

This privacy preserving mining goal, in practice, can be solved through two types of approaches including (1) using some communication protocols, such as Yao's protocol (Yao 1986), to request the distributions of the whole dataset, rather than requesting the actual values of each record, or (2) to design some special data mining methods to derive knowledge from anonymized data (this is inherently similar to the uncertain data mining methods).

DATA PRIVACY EXPERIMENT

In this section we applied a combination of *k-anonymity, suppression, and generalization* techniques on a Ugandan data set of about 1200 records, to implement data privacy. In our lab experiment, we employed the *k-anonymity* methodology to deidentify data from a Makerere University admission tabular data published publicly by the University as Student Admission Records and posted online [Makerere University Admission List, 2010]. Our reason for choosing *k-anonymity* is the ease of implementation for tabular data privacy. Our concept is that academia, businesses, non governmental organizations with no highly skilled computational experts in Uganda, could easily implement *k-anonymity* to provide basic data privacy for tabular data;

Steps taken to implement Data privacy

Our initial step was to de-identify the data set by removing PII as defined by the US data privacy laws. While no explicit data privacy laws exist in Uganda, we utilized the definitions of what constitutes PII as defined by the US data privacy laws (HIPAA), considering that they are universally acceptable. After we removed PII, we identified attribute values that we could suppress and other attribute values we could generalize. We then applied *k-anonymity* to the de-identified data set and rechecked if there was need reapply *suppression and generalization* to satisfy *k-anonymity*. We then output the de-identified tabular set satisfying *k-anonymity*. We checked for data utility to see if data to be published is meaningful to the user while not compromising privacy [Rastogi et al, 2007; Sramka et al, 2010].

PROCESS TAKEN TO ACHIEVE DATA PRIVACY

INPUT: Data from relation or schema
OUTPUT: Data privacy preserving published tabular data set
STEP 1. Identify PII Attributes
STEP 2. Remove PII Attributes
STEP 3. Identify none explicitly identifying or quasi-identifier attributes
STEP 4. Generalize or Suppress quasi-identifier attributes
STEP 5. Sort or order data
STEP 6. Check that ***k>1*** in tuples
STEP 7. Check for single values in attributes that cannot be grouped together to achieve ***k>1***
STEP 8. If single values and outliers that cannot be grouped together still exist in attributes, then continue to Generalize or Suppress quasi-identifier attribute values until k-anonymity is achieved at ***k>1***
STEP 9. Check for utility
STEP 10. Publish tabular data set

The original published data set included the following attributes, in which we let:

- ***A*** = { *RegNo, StudentNo, Lname, Fname, Mname, Sex, BirthDate, Nationality, Hall, Program, IndexNo,Year },* the relation *admission list* that included all attributes in the published data set.
- ***B*** = { *Lname, Fname, Mname, StudentNo, IndexNo, RegNo*}, the set of all PII attributes that we identified in the published data set.

ISSN 2278-3091

ternational Journal of Advanced Trends in Computer Science and Engineering, Vol. 3 , No.1, Pages : 122– 127 (2014)
ecial Issue of ICETETS 2014 - Held on 24-25 February, 2014 in Malla Reddy Institute of Engineering and Technology, Secunderabad– 14, AP, India

• ***C*** = { *Nationality, Sex, BirthDate,*}, the set of all quasi-identifier attributes identified in the data set.
• ***D****={Hall, Program, Year}*, the set of all non-sensitive attributes.
• ***E={ }***, the set of all sensitive attributes.
• Thus, we have: ***B*** ⊂ ***A, C*** ⊂ ***A, D*** ⊂ ***A*** and ***E*** ⊂ ***A***
• *Therefore* ***A=B*** ∪ ***C*** ∪ ***D*** ∪ ***E, A ={ B, C, D, E}.***
• Removing PII yields ***A ={ C, D, E}.***
• The de-identification of the *Admission List* set involves a complement of the *PII* set: ***(B)c = U – B = A –B = C + D + E.***
• Thus, therefore we remained with the *Quasi attributes, Non-Sensitive attributes, and Sensitive Attributes; w*here ***U*** is the universal set, which in this case is all the *Admission List attributes.*
• We suppressed or generalized the *Quasi Attributes:* suppress or generalize (***C***);
• Then, we applied *k-anonymity*: *k-anonymity((B)c);*
• ***Finally,*** we ordered values of ***(B)c;*** If ***k = 1***, we suppressed or generalized ***C*** until ***k >1***.

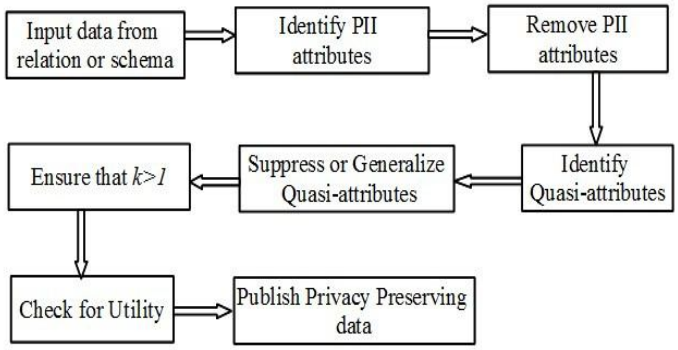

Figure 2. A data privacy process utilizing k-anonymity

Domain and Application Knowledge

Domain and application knowledge provides essential information for designing Big Data mining algorithms and systems. In a simple case, domain knowledge can help identify right features for modeling the underlying data (e.g., blood glucose level is clearly a better feature than body mass in diagnosing Type II diabetes). The domain and application knowledge can also help design achievable business objectives by using Big Data analytical techniques. For example, stock market data are a typical domain which constantly generates a large quantity of information, such as bids, buys, and puts, in every single second. The market continuously evolves and is impacted by different factors, such as domestic and international news, government reports, and natural disasters etc. An appealing Big Data mining task is to design a Big Data mining system to predict the movement of the market in the next one or two minutes.

Data utility challenges of removing PII

With the Makerere University data set, removing names and student numbers entirely kills utility. The data becomes meaningless to students who simply want to view it to see if their names are on the university admission list. One way this problem can be dealt with, is by publishing a list with just the *student number* or *student names* while obscuring other data as illustrated in the following two scenario:

· Scenario 1: we include *student number* publication of the university admission list:*Admission List = {StudentNo,Hall, Program, Year}.*

· Scenario 2: we include *student names* for publication of the university admission list: *Admission List = {Fname,Lname, Hall, Program, Year}.*

In both scenarios, the issue of balancing data utility and data privacy remain quite challenging and demand tradeoffs.

Relational model view

For a formal relational model view,
• we let ***π <attribute list>(R)***
• *w*here ***π*** is the projection or selecting of attributes from a relation (Table),
• ***<attribute list>*** is the list of attributes from *Admission List,*
• ***(R)*** is the relation from which we select attributes.
• The original projection with all attributes is:
• ***π****<RegNo, StudentNo, Lname, Fname, Mname, Sex, BirthDate, Nationality, Hall, Program, IndexNo, Year>****(Admission List)***.
The projection void of PII attributes is:
• *To_Be_Published_List* ← *π< Sex, BirthDate, Nationality, Hall, Program, Year >****(Admission List)***.
• We applied k-anonymity to the list that is to be published:
• *k-anonymity(To_Be_Published_List).*

The SQL Implementation

We implemented de-identification in SQL by creating a SQL View and doing SELECT on the view by choosing only attributes that remain in the *Admission List* after removing PII. We created SQL Views that are void of PII attributes:CREATE VIEW V2 AS SELECT Sex, BirthDate, Nationality, Hall, Program, Year FROM Admission_List;

RegNo	StudentNo	Lname	Fname	Mname	Sex	BirthDate	Nationality	Hall	Program	IndexNo	Year
09/U//EVE	20900	Annet	Anna		F	01/01/67	UGANDAN	AFRICA	LIS	U0166	2008
09/U//EVE	20901	Green	RICE		F	01/01/80	UGANDAN	MARY STUART	ARM	U0763	2008
09/U//EVE	20902	Timothy	NICE		F	01/01/81	KENYAN	MARY STUART	BLE	U0063	2007
09/U//EVE	20903	Jones	Jane	GRACE	F	01/01/73	TANZANIA	MARY STUART	LIS	U0198	2007
09/U//EVE	20904	Carter	James		M	01/01/74	UGANDAN		RAM	U0160	2007
09/U//EVE	20905	Brown	Britain	N	F	01/01/83	KENYAN	AFRICA	ARM	U0715	2008
09/U//EVE	20906	Sams	Sam		F	01/01/84	TANZANIA	MARY STUART	RAM	U0725	2007
09/U//EVE	20907	Faster	Master		M	01/01/85	UGANDAN		BLE	U1148	2008
09/U//EVE	20908	Uhuru	Kenya		F	01/01/90	UGANDAN	COMPLEX	ARM	U0062	2007
09/U//EVE	20909	Vineyard	Martha		M	01/01/88	KENYAN	AFRICA	ARM	U1017	2008

Table 1. Admission List with PII – Data is fictitious for illustrative purposes

Generalization

We generalized the *BirthDate* attribute to further prevent any reconstruction attacks by first developing a domain generalization hierarchy(DGH), as shown below, after which we

ISSN 2278-3091

ternational Journal of Advanced Trends in Computer Science and Engineering, Vol. 3 , No.1, Pages : 122– 127 (2014)
ecial Issue of ICETETS 2014 - Held on 24-25 February, 2014 in Malla Reddy Institute of Engineering and Technology, Secunderabad– 14, AP, India

implemented the generalization in SQL. We choose the DGH based on the oldest person in the data set, and built our *DGH to B4 = {196*}*, giving protection for the individuals born in 1967

Figure 3. Domain Generalization Hierarchy for the BirthDate Attribute.

SQL Implementation

CREATE table V2_Generalize1 SELECT Sex, BirthDate, Nationality, Hall, Program, Year FROM V2;

UPDATE V2_Generalize1 set BirthDate ='196*' WHERE BirthDate BETWEEN 1967-01-01 AND 1999-12-31';

Sex	BirthDate	Nationality	Hall	Program	Year
F	196*	UGANDAN	AFRICA	LIS	2008
F	196*	UGANDAN	MARY STUART	ARM	2008
F	196*	KENYAN	MARY STUART	BLE	2007
F	196*	UGANDAN	MARY STUART	LIS	2008
M	196*	UGANDAN		RAM	2007
F	196*	KENYAN	AFRICA	ARM	2008
F	196*	TANZANIA	MARY STUART	RAM	2007
M	196*	UGANDAN		BLE	2008
F	196*	UGANDAN	COMPLEX	ARM	2007
M	196*	TANZANIA	AFRICA	ARM	2008

Table 2. PII Attributes removed, BirthDate Attribute generalized to DGH to B4 = {196*}

Suppression

In the case of achieving *k-anonymity*, we had to suppress some values that appear once, yet still we had to ensure the utility of the data set.

Sex	BirthDate	Nationality	Hall	Program	Year
F	196*	UGANDAN	AFRICA	LIS	2008
F	196*	UGANDAN	MARY STUART	ARM	2008
F	196*	KENYAN	MARY STUART	BLE	2007
F	196*	TANZANIA	MARY STUART	LIS	2008
M	196*	UGANDAN		RAM	2007
F	196*	KENYAN	AFRICA	ARM	2008
F	196*	TANZANIA	MARY STUART	RAM	2007
M	196*	UGANDAN		BLE	2008
F	196*	UGANDAN	COMPLEX	ARM	2007
M	196*	KENYAN	AFRICA	ARM	2008

Table 3. Highlighted values to be suppressed

SQL:UPDATE V2_Generalize1 set Hall =' 'WHERE Hall ='Complex';

In **Table 3**, ***k!>1*** for Hall attribute. We suppressed the value 'Complex' in the Hall attribute, to achieve *k-anonymity* at ***k>1***for all values in the attributes. Yet still even though the Year attribute satisfies *l-diversity*, still an attacker could single out a single record of a female from Kenya, a resident of Mary Stuart Hall, enrolled in 2007. Therefore, we employed suppression to further conceal such records while achieving *k-anonymity > 1* as illustrated in **Table 4**.

Check for *k-anonymity* that ***k >1*** by ordering data and counting that attribute values satisfy condition ***k>1***:

SELECT Sex, BirthDate, Nationality, Hall, Program, Year FROM V2 ORDER BY Sex, Program, Hall;

Sex	BirthDate	Nationality	Hall	Program	Year
F	196*	UGANDAN	AFRICA	LIS	2008
F	196*	UGANDAN	MARY STUART	ARM	2008
F	196*	KENYAN	MARY STUART	BLE	
F	196*	TANZANIA	MARY STUART	LIS	
M	196*	UGANDAN		RAM	2007
F	196*	KENYAN	AFRICA	ARM	2008
F	196*	TANZANIA	MARY STUART	RAM	
M	196*	UGANDAN		BLE	2008
F	196*	UGANDAN		ARM	2007
M	196*	KENYAN	AFRICA	ARM	2008

Table 4. We achieve k-anonymity at k>1

2.3 Tier III: Big Data Mining Algorithms

2.3.1 Local Learning and Model Fusion for Multiple Information Sources

As Big Data applications are featured with autonomous sources and decentralized controls, aggregating distributed data sources to a centralized site for mining is systematically prohibitive due to the potential transmission cost and privacy concerns. On the other hand, although we can always carry out mining activities at each distributed site, the biased view of the data collected at each different site often leads to biased decisions or models, just like the elephant and blind men case. Under such a circumstance, a Big Data mining system has to enable an information exchange and fusion mechanism to ensure that all distributed sites (or information sources) can work together to achieve a global optimization goal.

Model mining and correlations are the key steps to ensure that models or patterns discovered from multiple information sources can be consolidated to meet the global mining objective. More specifically, the global mining can be featured with a two-step (local mining and global correlation) process, at data, model, and at knowledge levels. At the data level, each local site can calculate the data statistics based on the local data sources and exchange the statistics between sites to achieve a global data distribution view. At the model or pattern level, each site can carry out local mining activities, with respect to the localized data, to discover local patterns. By exchanging patterns between multiple sources, new global patterns can be synthetized by aggregating patterns across all sites (Wu and Zhang 2003). At the knowledge level, model correlation analysis investigates the relevance between models generated from different data sources to determine how relevant the data sources are correlated to each other, and how to form accurate decisions based on models built from autonomous sources

2.3.2 Mining from Sparse, Uncertain, and Incomplete Data

Spare, uncertain, and incomplete data are defining features for Big Data applications. Being sparse, the number of data points is too few for drawing reliable conclusions. This is normally a complication of the data dimensionality issues, where

ISSN 2278-3091

ternational Journal of Advanced Trends in Computer Science and Engineering, Vol. 3 , No.1, Pages : 122– 127 (2014)
ecial Issue of ICETETS 2014 - Held on 24-25 February, 2014 in Malla Reddy Institute of Engineering and Technology, Secunderabad– 14, AP, India

data in a high dimensional space (such as more than 1000 dimensions) does not show clear trends or distributions. For most machine learning and data mining algorithms, high dimensional spare data significantly deteriorate the difficulty and the reliability of the models derived from the data. Common approaches are to employ dimension reduction or feature selection (Wu et al. 2012) to reduce the data dimensions or to carefully include additional samples to decrease the data. scarcity, such as generic unsupervised learning methods in data mining.

Uncertain data are a special type of data reality where each data field is no longer deterministic but is subject to some random/error distributions. This is mainly linked to domain specific applications with inaccurate data readings and collections. For example, data produced from GPS equipment is inherently uncertain, mainly because the technology barrier of the device limits the precision of the data to certain levels (such as 1 meter). As a result, each recording location is represented by a mean value plus a variance to indicate expected errors. For data privacy related applications (Mitchell 2009), users may intentionally inject randomness/errors into the data in order to remain anonymous. This is similar to the situation that an individual may not feel comfortable to let you know his/her exact income, but will be fine to provide a rough range like [120k, 160k]. For uncertain data, the major challenge is that each data item is represented as some sample distributions but not as a single value, so most existing data mining algorithms cannot be directly applied.

Common solutions are to take the data distributions into consideration to estimate model parameters. For example, error aware data mining (Wu and Zhu 2008) utilizes the mean and the variance values with respect to each single data item to build a Naïve Bayes model for classification. Similar approaches have also been applied for decision trees or database queries. Incomplete data refers to the missing of data field values for some samples. The missing values can be caused by different realities, such as the malfunction of a sensor node, or some systematic policies to intentionally skip some values (*e.g.*, dropping some sensor node readings to save power for transmission).

While most modern data mining algorithms have inbuilt solutions to handle missing values (such as ignoring data fields with missing values), data imputation is an established research field which seeks to impute missing values in order to produce improved models (compared to the ones built from the original data). Many imputation methods (Efron 1994) exist for this purpose, and the major approaches are to fill most frequently observed values or to build learning models to predict possible values for each data field, based on the observed values of a given instance.

.

2.3.3 Mining Complex and Dynamic Data

The rise of Big Data is driven by the rapid increasing of complex data and their changes in volumes and in nature (Birney 2012). Documents posted on WWW servers, Internet backbones, social networks, communication networks, and transportation networks etc. are all featured with complex data. While complex dependency structures underneath the data raise the difficulty for our learning systems, they also offer exciting opportunities that simple data representations are incapable of achieving. For example, researchers have successfully used Twitter, a well-known social networking facility, to detect events such as earthquakes and major social activities, with nearly online speed and very high accuracy.

In addition, the knowledge of people's queries to search engines also enables a new early warning system for detecting fast spreading flu outbreaks (Helft 2008). Making use of complex data is a major challenge for Big Data applications, because any two parties in a complex network are potentially interested to each other with a social connection. Such a connection is quadratic with respect to the number of nodes in the network, so a million node network may be subject to one trillion connections. For a large social network site, like Facebook, the number of active users has already reached 1 billion, and analyzing such an enormous network is a big challenge for Big Data mining. If we take daily user actions/interactions into consideration, the scale of difficulty will be even more astonishing.

Inspired by the above challenges, many data mining methods have been developed to find interesting knowledge from Big Data with complex relationships and dynamically changing volumes. For example, finding communities and tracing their dynamically evolving relationships are essential for understanding and managing complex systems (Aral and Walker 2012, Centola 2010). Discovering outliers in a social network (Borgatti et al. 2009) is the first step to identify spammers and provide safe networking environments to our society.

If only facing with huge amounts of structured data, users can solve the problem simply by purchasing more storage or improving storage efficiency. However, Big Data complexity is represented in many aspects, including complex heterogeneous data types, complex intrinsic semantic associations in data, and complex relationship networks among data. That is to say, the value of Big Data is in its complexity.

Complex heterogeneous data types: In Big Data, data types include structured data, unstructured data, and semi-structured data etc. Specifically, there are tabular data (relational databases), text, hyper-text, image, audio and video data etc. The existing data models include key-value stores, bigtable clones, document databases, and graph database, which are listed in an ascending order of the complexity of these data models. Traditional data models are incapable of handling complex data in the context of Big Data. Currently, there is no acknowledged effective and efficient data model to handle Big Data.

3. Conclusions

Driven by real-world applications and key industrial stakeholders and initialized by national funding agencies, managing and mining Big Data have shown to be a challenging yet very compelling task. While the term Big Data literally concerns about data volumes, our HACE theorem suggests that the key characteristics of the Big Data are (1) huge with **h**eterogeneous and diverse data sources, (2) **a**utonomous with distributed and decentralized control, and (3) **c**omplex and **e**volving in data and knowledge associations. Such combined characteristics suggest that Big Data requires a "big mind" to consolidate data for maximum values (Jacobs 2009). In order to explore Big Data, we

have analyzed several challenges at the data, model, and system levels. To support Big Data mining, high performance computing platforms are required which impose systematic designs to unleash the full power of the Big Data. At the data level, the autonomous information sources and the variety of the data collection environments, often result in data with complicated conditions, such as missing/uncertain values.

4.References

1) Ahmed and Karypis 2012, Rezwan Ahmed, George Karypis, Algorithms for mining the evolution of conserved relational states in dynamic networks, *Knowledge and Information Systems*, December 2012, Volume 33, Issue 3, pp 603-630

2) Alam et al. 2012, Md. Hijbul Alam, JongWoo Ha, SangKeun Lee, Novel approaches to crawling important pages early, *Knowledge and Information Systems*, December 2012, Volume 33, Issue 3, pp 707-734

3) Aral S. and Walker D. 2012, Identifying influential and susceptible members of social networks, *Science*, vol.337, pp.337-341.

4) Machanavajjhala and Reiter 2012, Ashwin Machanavajjhala, Jerome P. Reiter: Big privacy: protecting confidentiality in big data. *ACM Crossroads*, 19(1): 20-23, 2012.

5) Banerjee and Agarwal 2012, Soumya Banerjee, Nitin Agarwal, Analyzing collective behavior from blogs using swarm intelligence, *Knowledge and Information Systems*, December 2012, Volume 33, Issue 3, pp 523-547

6) Birney E. 2012, The making of ENCODE: Lessons for big-data projects, *Nature*, vol.489, pp.49-51.