Big Data and Its Predictive Analysis

Thara D.K
Assistant Professor
Dept. of ISE, CIT, Gubbi, Tumkur, Karnataka, India
Email ID:tarakmurthy18@gmail.com

Veena.A
Assistant Professor
Dept. of CSE, CIT, Gubbi, Tumkur, Karnataka, India
Email ID:veenarammehtry@gmail.com

***Abstract*--- Big Data Everywhere! The pervasiveness of computers in everyday life has already increased and keeps increasing the available digital data both in volume and variety/disparity. The volume of the data generated in an organization is increasing everyday. An efficient and scalable storage system is required to manage data growth. The first quarter part of the article talks about Map Reduce, which is a software framework that processes vast amount of data in parallel on large clusters. Predictive analytics is one of the best ways to use data in order to improve decision making. Hence the remaining part of the article demonstrates that the sparse fine grained data is the basis for predictive analytics. Predictive analytics is performed by one of the best modeling technique Naive Bayes. The results based on Naive Bayes are conservative as one would expect theoretically and empirically.**

Keywords: Big Data; Map Reduce; Predictive analytics

1. INTRODUCTION

The large and dynamic availability of digital data is referred to as Big Data. Lots of data is being collected and warehoused from sources like, Web data, e-commerce, purchases at department/grocery stores, Bank/Credit Card transactions, Social Networks. Big Data refers to huge amount of data that is generally in peta bytes.

Big Data is a buzzword that is associated with volumes of data that cannot be processed in traditional environments [2]. The quantity of data that is created every two days is estimated to be 5 exabytes. This amount of data is similar to the amount of data created from the dawn of time up until 2003. Moreover, it was estimated that 2007 was the first year in which it was not possible to store all the data that we are producing. This massive amount of data opens new challenging discovery tasks. The challenges in big data processing in real time include handling of streams that come with certain velocity, parallel processing of data, and even correlation [3]. Big Data and its processing are attracting lot of attention recently. Hadoop is open source software that makes use of MapReduce for distributed processing. It has attracted worldwide attention for processing big data in the real world [1]. Distributed parallel processing is done by Hadoop and over few years it has become a reliable system for processing big data.

Big data processing has certain phases that include data acquisition and recording, information extraction and cleaning, data representation, aggregation and integration, data modeling, analysis and query processing, and interpretation. The challenges in big data processing include heterogeneity and incompleteness, scale, timeliness, privacy and human collaboration [4]. Big data is measured in terabytes as shown in figure 1.

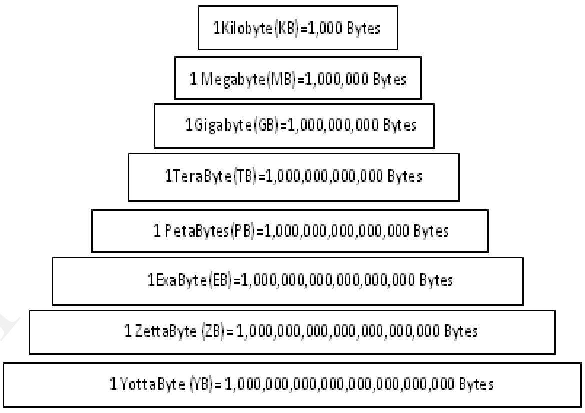
1Kilobyte(KB)=1,000 Bytes
1 Megabyte(MB)=1,000,000 Bytes
1Gigabyte(GB)=1,000,000,000 Bytes
1TeraByte(TB)=1,000,000,000,000 Bytes
1 PetaBytes(PB)=1,000,000,000,000,000 Bytes
1ExaByte(EB)=1,000,000,000,000,000,000 Bytes
1 ZettaByte (ZB)= 1,000,000,000,000,000,000,000 Bytes
1 YottaByte (YB)= 1,000,000,000,000,000,000,000,000 Bytes

Fig 1: Mountain of Data [5]

Big data typically refers to the following types of data:

- Traditional enterprise data – includes customer information from Customer Relationship Management (“CRM”) systems, transactional Enterprise Resource Planning (“ERP”) data, web store transactions, and general ledger data.
- Machine-generated /sensor data – includes Call Detail Records (“CDR”), weblogs, smart meters, manufacturing sensors, equipment logs (often referred to as digital exhaust), and trading systems data.
- Social data – includes customer feedback streams, micro-blogging sites like Twitter, social media platforms like Facebook

There are three key characteristics that define big data:

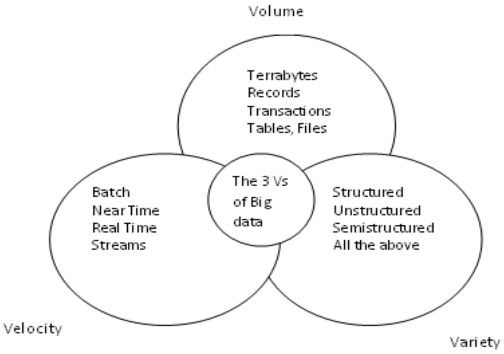

Fig 2: The Three Vs of Big Data

- Volume: Volume refers to larger amounts of data being generated from a range of sources For example; big data can include data gathered from the Internet of Things (Iota). As originally conceived, Iota referred to the data gathered from a range of devices and sensors networked together, over the Internet. RFID tags appear on inventory items capturing transaction data as goods are shipped through the supply chain. Big data can also refer to the exploding information available on social Media such as Face book and Twitter.

- Variety: Variety refers to using multiple kinds of data to analyze a situation or event. On the Iota, millions of devices generating a constant flow of data results in not only a large volume of data but different types of data characteristic of different situations. For example, in addition to WSN, heart monitors in patients and Global position System all generate different types of structured data However, devices and sensors aren't the only sources of data. Additionally, people on the Internet generate a highly diverse set of structured and unstructured data. Web browsing data, captured as a sequence of clicks, is structured data. However, there's also substantial unstructured data. For example, in 2012 there were 600 million websites and more than 125 million blogs, with many including non structured multidimensional data base.

- Velocity: Velocity of data also is on demand rapidly over time for semi structure data band there's a need for more frequent decision making about that data. As the world becomes more global and developed, and as the Iota builds, there's an increasing frequency of data capture and decision making about those —things as they move through the world. Further, the velocity of social media use is increasing. For example, there are more than 250 million face book per day. Face book lead to decisions about other Face book, escalating the velocity. Further, unlike classic data warehouses that generally —store data, big data is more dynamic. As decisions are made using big data, those decisions ultimately can influence the next data that's gathered and analyzed, adding another dimension to velocity.

- Value. The economic value of different data varies significantly. Typically there is good information hidden amongst a larger body of non-traditional data; the challenge is identifying what is valuable and then transforming and extracting that data for analysis.

A. Big Data Research Challenges

Main objective of Big Data analysis is to generate value from huge amount of unorganized data. Privacy and security issues, data ownership, heterogeneity, timeliness, maintaining cloud service for Big Data, machine learning algorithm for Big Data, scalability and complexity are the major research challenges for Big Data analysis. Due to high rate of data growth, due to huge volume and unstructured nature of data, traditional RDBMS and SQL can't be used for Big Data analysis.

To store fast growing huge information generating from various data source, the data processing system should have a scalable architecture. Scalability means ability to add more node to the cluster as the data grow, without affecting the performance of the system. Traditional RDBMS is not suitable enough for the Big Data. First reason is RDBMS or traditional Distributed Data Base System cannot expand to a cluster having thousands of nodes due to restrictions imposed by ACID constraints. In case of cluster with large number of nodes there involves significant network delay and maintaining consistency becomes very difficult. Second reason is traditional RDBMS cannot operate on unstructured or semi structured data.

A good Big Data analysis system should have two characteristics. Firstly, it should able to store and access huge volume of data in a small time. Though the storage devices becoming cheaper day by day, the data access speed is not improving in that way. So the data storage architecture should be smart enough to access huge data in small time from many slow devices. Google Distributed File system and Hadoop Distributed File System are two very efficient frameworks for storage and access of huge data. Second characteristics of Big Data analysis system is it should be able to process huge amount of data in small time to draw some conclusion from it. But there is a limitation in micro processor speed. Processor speed cannot be increased beyond certain limit due generation of uncontrollable heat. Therefore parallel data processing is an alternative solution for data intensive operation. Map Reduce is an innovative idea for data intensive computation which ultimately does parallel processing of huge data [19].

B. Tools: open source revolution

The Big Data phenomenon is intrinsically related to the open source software. Large companies as Facebook, Yahoo!, Twitter, and LinkedIn benefit and contribute working on open source projects. Big Data infrastructure deals with Hadoop, and other related software as:

- Apache Hadoop [15]: software for data-intensive distributed applications, based in the MapReduce programming model and a distributed file system called Hadoop Distributed Filesystem (HDFS). Hadoop allows writing applications that rapidly process large amounts of data in parallel on large clusters of compute nodes.

- Apache Pig [16]: software for analyzing large data sets that consists of a high-level language similar to SQL for expressing data analysis programs, coupled with infrastructure for evaluating these programs. It contains a compiler that produces sequences of Map-Reduce programs.

- Cascading [15]: software abstraction layer for Hadoop intended to hide the underlying complexity of MapReduce jobs. Cascading allows users to create and execute data processing workflows on Hadoop clusters using any JVM-based language.

- Scribe [16]: server software developed by Facebook and released in 2008. It is intended for aggregating log data streamed in real time from a large number of servers.

- Apache HBase [4]: non-relational columnar distributed database designed to run on top of Hadoop Distributed Filesystem (HDFS). It is written in Java and modeled after Google's BigTable. HBase is an example if a NoSQL data store.

- Apache Cassandra [2]: another open source distributed database management system developed by Facebook. Cassandra is used by Netflix, which uses Cassandra as the back-end database for its streaming services.

- Apache S4 [29]: platform for processing continuous data streams. S4 is designed specifically for managing data streams. S4 apps are designed combining streams and processing elements in real time.

- Storm [34]: software for streaming data-intensive distributed applications, similar to S4, and developed by Nathan Marz at Twitter.

2. MAP REDUCE

MapReduce is the new programming model that is used for generating and processing huge data known as big data. The results of big data processing can be used in various sectors such as financial services, education, health, agriculture, and so on [6].There are two models for processing big data. They are known as MapReduce and Data Flow Graphs (DFGs). The MapReduce model was originally proposed by Google. This model is as shown in figure 3.

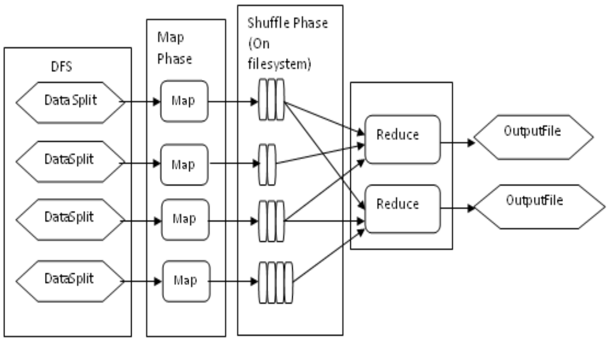

Fig 3 : Map Reduce Model [18]

As can be seen in figure 3, the MapRedue model has many phases involved. Important phases are Map phase and Reduce phase. First of all, the DFS component takes big data and splits the data. Such data is mapped in the Map phase. Afterwards, the maps are processed using Shuffle phase on the file system. Afterwards, the Reduce phase generates final output. The MapReduce model ha some drawbacks. They include mandated shuffle phase is not efficient, and joins are also cumbersome.

MapReduce is a software framework for easily writing applications which process vast amounts of data (multi-terabyte data-sets) in-parallel on large clusters (thousands of nodes) of commodity hardware in a reliable, fault-tolerant manner.

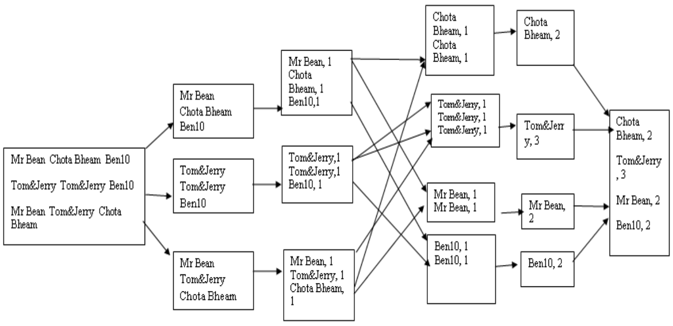

Fig 4 : Map Reduce example

MapReduce has become a dominant parallel computing paradigm for big data, i.e., colossal datasets at the scale of tera-bytes or higher. Ideally, a MapReduce system should achieve a high degree of load balancing among the participating machines, and minimize the space usage, CPU and I/O time, and network transfer at each machine.

A significant amount of recent research work has addressed the problem of solving various data management problems in the cloud. The major algorithmic challenges in map-reduce computations involve balancing a multitude of factors such as the number of machines available for mappers/reducers, their memory requirements, and communication cost (total amount of data sent from mappers to reducers). Most past work provides custom solutions to specific problems, e.g., performing fuzzy joins in map-reduce, clustering, graph analyses, and so on. While some problems are amenable to very efficient map-reduce algorithms, some other problems do not lend themselves to a natural distribution, and have provable lower bounds. Clearly, the ease of "map-reducability" is closely related to whether the problem can be partitioned into independent pieces, which are distributed across mappers/reducers [14].

3. SPARSE, FINE-GRAINED (BEHAVIOR) DATA

In this article we focus on one particular sort of data: sparse, fine-grained feature data, such as that derived from the observation of the behaviors of individuals. In the context of behavior data, we can draw a contrast: we specifically do not mean data that are summaries of individuals' behaviors, as used traditionally, but data on the actual fine-grained behaviors themselves.

For example, data on individuals' visits to massive numbers of specific web pages are used in predictive analytics for targeting online display advertisements. 8–10 Data on individual geographic locations are used for targeting mobile advertisements.[7] Data on the individual merchants with which one transacts are used to target banking advertisements.[8] A key aspect of such datasets is that they are sparse: for any given instance, the vast majority of the features have a value of zero or ‘‘not present.’’ For example, any given consumer has not transacted with the vast majority of merchants, has not visited the vast majority of geographic locations, has not visited the vast majority of web pages, etc.

Predictive modeling based on sparse, finegrained (behavior) data is not a new phenomenon. More data do not necessarily lead to better predictive performance. It has been argued that sampling (reducing the number of instances) or transformation of the data to lower dimensional spaces (reducing the number of features) is beneficial [9] whereas others have argued that massive data can lead to lower estimation variance and therefore better predictive performance[8]. The bottom line is that, not unexpectedly, the answer depends on the type of data, the distribution of the signal (the information on the target variable) across the features, as well as the signal-to-noise ratio. Therefore, we will focus on a certain sort of data: sparse, fine-grained data, such as data created by the detailed behavior of individuals. Such data from different domains have similar characteristics that would lead one to expect increasing benefits with very large data, an expectation that does not come with traditional data for predictive modeling. Modern information systems increasingly are recording fine-grained actions of individuals. As we use our telecommunications devices, make financial transactions, surf the Web, send e-mail, tag online photos, ‘‘like’’ postings, and so on, our behaviors are logged.

When data are drawn from human actions, noise rates often are high because of the ceiling imposed by behavioral reliability [10]. Social scientists have long argued that one way to circumvent the poor predictive validity of attitudes and traits is to aggregate data across occasions, situations, and forms of actions [11]. This provides an early suggestion that more (and more varied) data might indeed be useful when modeling human behavior data. The implication for predictive analytics based on data drawn from human behaviors is that by gathering more data over more behaviors or individuals (aggregated by the modeling), one could indeed hope for better predictions.

4. PREDICTIVE TECHNIQUES FOR ANALYSING FINE GRAINED BEHAVIOR DATA

Prediction can be viewed as construction and use of a model to assess the class of an unlabelled sample or to assess the value or value ranges of an attribute that a given sample is likely to have. Predictive modeling is based on one or more data instances for which we want to predict the value of a target variable. Data-driven predictive modeling generally induces a model from training data, for which the value of the target (the label) is known.

Bayesian classifiers are statistical classifiers. They can predict class membership probabilities such as the probability that a given sample belongs to particular class. Comparing classification algorithms have found a simple Bayesian classifier known as Naïve Bayesian classifier.

Naïve Bayesian classifier assumes that the effect of an attribute value on a given class is independent of the values of the other attributes. This assumption is called class conditional independence. It is made to simplify the computations in world and in this sense is considered “Naïve”.

For the experiments we report, we use the Naıve Bayes classifier to learn models and to make predictions. Despite its simplicity, Naı¨ve Bayes has been shown to have surprisingly good performance on sparse datasets. A very attractive property of Naı¨ve Bayes is that it is extremely fast to run on a large, sparse dataset when it is formulated well. The main speedup stems from the fact that Naı¨ve Bayes completely ignores interfeature dependencies by assuming that the within-class covariances of the features are zero. Combining this “naïve” assumption with Bayes' rule for conditional probability produces the following formulation:

$$P(C = c|x_{1,\ldots,}x_m) = \frac{P(C = c) \cdot P(x_1, \cdots, x_m|C = c)}{P(x_1, \cdots, x_m)}$$

$$\propto P(C = c) \cdot \prod_{j=1}^{m} P(X_j = x_j|C = c)$$

There are a several variants of Naı¨ve Bayes in common use, which differ in their calculation of the conditional probabilities. Each of these variants assumes a different event model, which postulates how the features from the data were generated [12]. The two most popular choices are the multivariate and the multinomial event models. The main idea behind the multinomial event model is that each input sample results from a series of independent draws from the collection of all features. The main advantage of this model is that it requires only computing over those features that have a count that is nonzero. For a sparse dataset, this results in a huge reduction in run time since all the zeros can be ignored. The multivariate event model takes the point-of-view that the features are generated according to an independent Bernoulli process with probability parameters θj for class C.

5. MULTIVARIATE BERNOULLI NAÏVE BAYES FOR BIG DATA

In the case of classification of high-dimensional sparse data, we can achieve computational time reductions for Naı¨ve Bayes by reformulating the calculation all of the probabilities in terms of only the active (non-zero) elements of the data matrix [17]. Specifically, let us denote the i-th instance in the dataset as xi and the j-th feature of the i-th instance as xi,j. An element in this data matrix is said to be inactive if its value is equal to zero (xi,j = 0) and active otherwise. For the sparsedata problems (with binary features) on which this paper focuses, the average number of active elements per instance jxj is much smaller than the total number of features m. Thus, it makes sense to use a sparse formulation.

The Multivariate Bernoulli event model assumes that the data are generated according to a Bernoulli process. This is in fact a compound assumption consisting of the following three components:

1. The problem is binary classification.
2. The data are generated i.i.d.
3. The features can only take on a distinct number of Discrete levels (e.g. 0 and 1). We will assume they are Binary.

These assumptions enable very fast calculation of the empirical probabilities encountered when computing the class probability of an instance. If the data were generated by independent Bernoulli processes with probability parameters hj for class C, we ought to see that:

$$P(x_i|C=c)$$

$$=\prod_{j=1}^{m} P\left(X_j=x_{i,j}|C=c\right)^{x_{i,j}}.\left(1P(X_j=x_{i,j}|C=c)\right)^{(1-x_{i,j})}$$

$$=\prod_{j=1}^{m}(\theta_j)^{x_{i,j}}\left(1-\theta_j\right)^{(1-x_{i,j})}$$

We can then determine the log likelihood of the combination of all of the probabilities gathered in probability vector h, given all of the training instances x as:

$$L(\theta)=\log P(x|C=c)$$

$$=\sum_{i=1}^{n}\sum_{j=1}^{m} x_{i,j}\log\theta_j+\left(1-x_{i,j}\right)\log\left(1-\theta_j\right)$$

Where L(θ) represents the log likelihood and xi,j again represents the value of feature j of training instance i. Maximizing the log likelihood yields:

$$\hat{\theta}_{X_j=1|C=c} = P(X_j=1|C=c;\theta_c)$$

$$= \frac{|X_j=1\wedge C=c|}{|C=c|}$$

For computational reasons, this is usually reformulated as:

$$\hat{\theta}_{X_j}=1|C=c=\frac{1+|X_j=1\wedge C=c|}{2+|C=c|}$$

This corresponds to assuming a Laplacian prior on the probability in a Bayesian setting. Similarly, we can derive that:

$$\hat{\theta}_c=P\left(C=c\right)=\frac{|C=c|}{n}$$

This leads to the most common formulation of the Naı¨ve Bayes classifier. The Bernoulli event model for sparse data: For the Bernoulli event model, a sparse formulation is also possible by expanding the summands in the calculation of the log likelihood of an input vector into two parts. This is possible due to the assumption that
a feature value Xj can only take on two values (either one or zero). The log conditional probability of the instance xi to belong to class C, given its values for the binary features then becomes:

$$\log P(X_i|C)=\log\left(\prod_{j=1}^{m}P\left(X_{j=x_{i,j}}|C\right)\right)$$

$$=\sum_{j=1}^{m}\log\left(P\left(X_{j=x_{i,j}}|C\right)\right)$$

$$=\sum_{j|x_{i,j=1}}\log\left(P(X_j=1|C)\right)+\sum_{j|x_{i,j=0}}\log\left(P(X_j=0|C)\right)$$

$$=\sum_{j|x_{i,j=1}}\log\left(P(X_j=1|C)\right)+\sum_{j|x_{i,j=0}}\log[P(C)-P(X_j=1|C)]$$

Revealing the following log likelihood for an instance xi:

$$\log P(C|x)\propto\log P(C)+\sum_{j|X_{i,j=1}}\log\left(P(X_j=1|C)\right)+\sum_{j|X_{i,j=0}}\log[P(C)-P(X_j=1|C)]$$

With:

$$\sum_{(j|x_{i,j}=0)} \log[P(C)-P(X_j=1|C)] = \sum_{j=0}^{m} \log[P(C)-P(X_j=1|C)] - \sum_{j|x_{i,j=1}} [\log P(C)-P(X_j=1|C)]$$

For a dataset with $\overline{m}$ active elements per instance, this formulation of log likelihood only needs O($\overline{m} \bullet n$) amortized time to finish for n instances (as opposed to O(m● n) for the original formulation) under the assumption that m is of the same order as n. This leads to a reduction in computational time proportional to $\rho = {}^{m}/_{\overline{m}}$ which increases as the sparseness of the dataset increases. Note that a separate problem with computing and using Naı¨ve Bayes estimates in practice is that often the exponentials to be calculated to recover the probabilities are so small that they cannot be represented by a computer using double precision. One example of avoiding the exponentials is to compare the ratio of the probabilities instead of the actual probabilities [13] computing a score such as:

$$S(x) = \frac{P(C=1|x)}{P(C=0|x)}$$

$$\alpha \log P(C=0|x) - \log P(C=1|x)$$

CONCLUSION

We have discussed big data and the mapreduce programming model that is used for generating and processing the huge data. In this paper we have focused on a particular sort of data as being a reason for continued attention to scaling up. Predictive modeling is based on on or more data instances for which we want to predict the value of a target variable. Our main contribution is to use sparse fine-grain data to predict the target variable.

We have used an implementation of multivariate naïve bayes that can mine massive, sparse data extremely efficiently. Predictive modeling with large, transactional data can be made substantially more accurate by increasing the data size to a massive scale.

Big Data Impact on Society

Emergence of Big Data technologies made it possible for a wide range of people including researchers from social science and humanities, educational institute, government organization, and individual to produce, share, organize and interact with large scale data. With what motive and perspective do people from different groups use mass volume of data using latest technology is crucial. If it is used for decision making or opinion making or enforcement of new policies, it will have considerable long term impact on society and individual. The market sees Big Data as pure opportunity to target advertising towards right kind of people, which may bother an individual with flood of advertisements. Business and governments may exploit Big Data without concern for issue of legality, data quality. This may leads to poor decision makings. The threat of use of Big Data without a legal structure and strict law can hamper both individual and society as a whole.

Big data does not always mean as better data. A few Social scientists and policy maker sees big data as a representative of society. Which is not necessarily be true as a large portion of world population still does not dump data into Big Data repository by using internet or by any other means. For instance Twitter or Facebook does not represent all people, all though many sociology researchers and journalist treat them as if they are representative of global population. More over number of accounts on social networking sites does not necessarily represent same number of people, as individuals can fake their identity and can create multiple accounts. A large mass of raw information in form of Big Data is not self-explanatory. And the specific methodologies for interpreting the data are open to all sorts of philosophical and ethnical debate. It may or may not represent the truth and an interpretation may be biased by some ethnic views or personal opinions.

Personal data can be sensitive and may have some privacy issue. It is valid and serious issue whether privacy can be maintained with increasing storage and usages of Big Data.

For example there are huge data on health care system available today which can are used extensively for research purpose. And an individual can be identified from it and can be monitored periodically who is suffering from a disease without his or her knowledge. But it may emotionally or socially harm the person once his or her health information made public by people with evil intention. Many dataset contains identifier for individual such as name, date of birth or unique code issued by government agencies. So an individual can be spied with good or bad intention. Big data aggregator assumes that they have rights to the whole data which may include private and sensitive data as well. But in case of company failure or company take over, the data set may go to some other hand and any existing privacy protection policy are unlikely to survive in a hand of a new owner.

REFERENCES

[1] BogdanGhit¸ AlexandruIosup and Dick Epema (2005). Towards an Optimized Big Data Processing System. USA: IEEE. P1-4.

[2] SriramRao, Raghu Ramakrishnan and Adam Silberstein (2012). Sailfish: A Framework for Large Scale Data Processing. USA: Microsoft. p1-
14.

[3] TAKAHASHI Chieko, SERA Naohiko, TSUKUMOTO Kenji, OSAKI Hirotatsu (2012). OSS Hadoop Use in Big Data Processing. USA: NEC TECHNICAL JOURNAL. p1-5.

[4] LiranEinav and Jonathan Levin (2013). The Data Revolution and Economic Analysis. USA: Prepared for the NBER Innovation Policy and the Economy Conference. p1-29.

[5] Yingyi Bu, Bill Howe, Magdalena Balazinska and Michael D. Ernst (2010).HaLoop: Efficient Iterative Data Processing on Large Clusters. USA: IEEE. p1-12.

[6] Ling LIU (2012). Computing Infrastructure for Big Data Processing. USA: IEEE. P1-9.

[7] Provost F. Geo-social targeting for privacy-friendly mobile advertising: Position paper. Working paper CeDER-11-06A. Stern School of Business, New York University, 2011.

[8] Martens D, Provost F. Pseudo-social network targeting from consumer transaction data. Working paper CeDER- 11-05. Stern School of Business, New York University, 2011.

[9] Kohavi R, John GH. Wrappers for feature subset selection.Artif Intell 1997; 97:273–324.

[10] Ajzen I. The theory of planned behavior. Theor Cogn Self Regul 1991; 50:179–211.

[11] Fishbein M, Ajzen I. Attitudes towards objects as predictors of single and multiple behavioral criteria. Psychol Rev 1974; 81:59–74.

[12] McCallum A, Nigam K. A comparison of event models for naive bayes text classification. AAAI Workshop on Learning for Text Categorization, 1998.

[13] Bickel PJ, Levina E. Some theory for Fisher’s linear discriminant function, naive Bayes, and some alternatives when there are many more variables than observations.Bernoulli 2004; 10:989–1010.

[14] Towards an Understanding of the Limits of Map-Reduce Computation, Foto N. Afrati, Anish Das Sarma

[15] Apache Hadoop, http://hadoop.apache.org.

[16] Apache Pig, http://www.pig.apache.org/.

[17] Enric Junque de Fortuny, David Martens and Foster Provost (2013) Predictive modeling with big data.

[18] Dr. Shoban Babu Sriramaju (2014) A Review on processing on Big Data

[19] Surajit Das, Dr. Dinesh Gopalani (2014) Big Data Analysis Issues and Evolution of Hadoop : IJPRET, Volume 2(8).