ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 12, December 2015

Map-Reduce and Relationship Miner for Big Data

K.Yogeshwaran, R.Gowri, K.Monika

Assistant Professor, Dept. of ECE, Kalaignar Karunanidhi Institute of Technology, Coimbatore, India

Assistant Professor, Dept. of ECE, Tejaa Shakthi Institute of Technology, Coimbatore, India

Research Scholar, Dept. of Computer Science, PSG college of Arts and Science, Coimbatore, India

ABSTRACT: Big Data concern 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 are now rapidly expanding in all science and engineering domains, including physical, biological and biomedical sciences. Data mining and big data are both of them relate to the use of large data sets to handle the collection or reporting of data. Data mining with big data to retrieve the large amount of data set in the database. The term often refers simply to the use of predictive analytics or other certain advanced methods to extract value from data. Accuracy in big data may lead to more confident decision making and better decisions can mean greater operational efficiency, cost reduction and reduced risk. An aim of this research work, Big Data starts with large-volume, heterogeneous, autonomous sources with distributed and decentralized control and seeks to explore complex and evolving relationships among data. This paper presents a new data mining method that analyzes the time-persistent relations or states between the entities of the dynamic networks and captures all maximal non-redundant evolution paths of the stable relational states.

KEYWORD: Big Data, data mining, k-means clustering, HACE theorem, heterogeneity, autonomous sources, complex and evolving associations.

I. INTRODUCTION

This study proposes a HACE (Heterogeneous, Autonomous, Complex, Evolving) theorem to model Big Data characteristics. These characteristics make it an extreme challenge for discovering useful knowledge from the Big Data. In this research work, HACE theorem can be used to characterizes the features of the Big Data revolution, and proposes a Big Data processing model, from the data mining perspective. The Data Driven model is aggregation of information sources, mining and analysis, user interest modelling and security and privacy. Existing methods can only work in an offline fashion and are incapable of handling this Big Data scenario in real time. In proposed system to build a stream-based Big Data analytic framework for fast response and real-time decision making. The key challenges and research issues include: - designing Big Data sampling mechanisms to reduce Big Data volumes to a manageable size for processing to prediction models from Big Data streams. A knowledge indexing framework to ensure real time data monitoring and classification for Big Data applications.

II. RELATED WORK

Dynamic networks have recently being recognized as a powerful abstraction to model and represent the temporal changes and dynamic aspects of the data underlying many complex systems. Significant insights regarding the stable relational patterns among the entities can be gained by analysing temporal evolution of the complex entity relations. This can help identify the transitions from one conserved state to the next and may provide evidence to the existence of external factors that are responsible for changing the stable relational patterns in these networks. Experimental results based on multiple datasets from real world applications show that the method is efficient and scalable. As the capacity to exchange and store information has soared, so has the amount and diversity of available data. To represent the relations between various entities in diverse applications and to capture the temporal changes and dynamic aspects of

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 12, December 2015

the underlying data, dynamic networks have been used as generic model due to its flexibility and availability of theoretical and applied tools for efficient analysis.Examples of some widely studied networks includes the friend networks of popular social networking sites like Facebook, the Enron email network, co-authorship and citation networks, and protein-protein interaction networks. Analysis of temporal aspects of the entity relations in these networks can provide significant insight about the conserved relational patterns and their evolution over time. Considerable effort has been made towards the development of efficient methods to analyze and extract useful information from static networks. Although the existing techniques can detect the frequent patterns in a dynamic network or track related patterns over time, they are not designed to identify stable relational patterns and do not focus on tracking the changes of these conserved relational patterns over time. Our contribution in this paper is two folds. Firstly, we introduce a new class of patterns referred as the evolving induced relational states that are designed to analyze the time-persistent relations or states between the entities of the dynamic networks. Secondly, we present an algorithm to efficiently mine all maximal non-redundant evolution paths of the stable relational states of a dynamic network. We experimentally evaluate our algorithm using three real world datasets. First, we evaluate the performance and scalability of the algorithm on a large patent citation network by varying different input parameters. Second, we investigate some discovered evolving induced relational states from a trade network, an email communication network, and a patent citation network and provide a qualitative analysis of the information captured in them.

DISADVANTAGES OF EXISTING SYSTEM:

- It provides 100 times more sensitive vision than any existing radio telescopes, answering fundamental questions about the Universe. However, with a 40 gigabytes (GB)/second data volume, the data generated from the SKA are exceptionally large.
- Although researchers have confirmed that interesting patterns, such as transient radio anomalies can be discovered from the SKA data.
- Existing methods can only work in an offline fashion and are incapable of handling this Big Data scenario in real time. As a result, the unprecedented data volumes require an effective data analysis and prediction platform to achieve fast response and real-time classification for such Big Data.

III. PROPOSED ALGORITHM

In proposed system to build a stream-based Big Data analytic framework for fast response and real-time decision making. The key challenges and research issues include: -designing Big Data sampling mechanisms to reduce Big Data volumes to a manageable size for processing; - building prediction models from Big Data streams. Such models can adaptively adjust to the dynamic changing of the data. A knowledge indexing framework to ensure real-time data monitoring and classification for Big Data applications.

Advantages

- Hug data store and retrieve
- Adapted all environments
- More reliable
- User friendly
- Avoid collusions (eg. Dead lock)
- Ignore network traffics.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 12, December 2015

IV. LITERATURE SURVEY

Briefly, Alpaydin mentions optical character recognition, speech recognition, and encoding/decoding as example applications of k-means. However, a survey of the current literature on the subject offers a more in depth treatment of some other practical applications, such as "data detection" for burst-mode optical receiver and recognition of musical genres (Turnbull and Elkan), which are specialized examples of what Alpaydin mentions. As Zhao, Nehorai, and Porat describe "burst-mode data-transmission systems," a "significant feature of burst-mode data transmissions is that due to unequal distances between" sender and receivers, "signal attenuation is not the same" for all receivers. Because of this, "conventional receivers are not suitable for burst-mode data transmissions." The importance, they note, is that many "high-speed optical multi-access network applications, [such as] optical bus networks [and] WDMA optical star networks" can use burst-mode receivers (Zhao, 1492). In their paper, they provide a "new, efficient burst-mode signal detection scheme" that utilizes "a two-step data clustering method based on a K-means algorithm." They go on to explain that "the burst-mode signal detection problem" can be expressed as a "binary hypothesis," determining if a bit is 0 or 1. Further, although they could use maximum likelihood sequence estimation (MLSE) to determine the class, it "is very computationally complex, and not suitable for high-speed burst-mode data transmission." Thus, they use an approach based on k-means to solve the practical problem where simple MLSE is not enough (Zhao, 1493).

The new approach Turnbull and Elkan use to initialize k-means is what they call Subset Furthest First (SFF). They note that a problem with (normal) Furthest First is "that it tends to find the outliers in the data set." By using only a subset, they found that there are less outliers that can be found, and "thus, the proportion of nonoutlier points obtained as centers is increased" (Turnbull, 581). Finally, Turnbull and Elkan also show that using the multiple initialization methods of Gaussian maximum likelihood, k-means, and in-class k-means, they get about the same classification accuracy as one would obtain by using the method of gradient descent. However, they note, "in each trial, creating a network without gradient descent takes seconds, whereas applying gradient descent takes hours." In the end, they cite a study that "found that humans achieved 70 percent music classification accuracy," and compared that to their own result - which was 71.5 percent (Turnbull, 583). Reading Turnbull and Elkan, as well some other current literature, one can see some potential difficulties in using k-means. Some of those below are discussed below.

Simply put, k-Means Clustering is an algorithm among several that attempt to find groups in the data (Alpaydin 139). In pseudo code, it is shown by Alpaydin (139) to follow this procedure:

Initialize $\mathbf{m}_i$, $i = 1,\ldots,k$, for example, to k random $\mathbf{x}^t$
Repeat
  For all $\mathbf{x}^t$ in X
    $b_i^t \leftarrow 1$ if $\| \mathbf{x}^t - \mathbf{m}_i \| = \min_j \| \mathbf{x}^t - \mathbf{m}_j \|$
    $b_i^t \leftarrow 0$ otherwise
  For all $\mathbf{m}_i$, $i = 1,\ldots,k$
    $\mathbf{m}_i \leftarrow$ sum over t $(b_i^t \mathbf{x}^t)$ / sum over t (b_i^t)
Until $\mathbf{m}_i$ converge

V. K-MEANS CLUSTERING

K-means clustering aims to partition n observations into k clusters in which each observation belongs to the cluster with the nearest mean, serving as a prototype of the cluster. This results in a partitioning of the data space into Voronoi cells. Data mining involves exploring and analyzing large amounts of data to find patterns for big data. The techniques came out of the fields of statistics and artificial intelligence (AI), with a bit of database management thrown into the mix. Generally, the goal of the data mining is either classification or prediction.

Let $X = \{x_1, x_2, x_3, \ldots\ldots\ldots, x_n\}$ be the set of data points and $V = \{v_1, v_2, \ldots\ldots\ldots, v_c\}$ be the set of centers.

Step 1: Randomly select *'c'* cluster centers.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 12, December 2015

Step 2: Calculate the distance between each data point and cluster centers.

Step 3: Assign the data point to the cluster center whose distance from the cluster center is minimum of all the cluster centers..

Step 4: Recalculate the new cluster center using:

$$v_i = (1/c_i) \sum_{j=1}^{c_i} x_i$$

where, 'c_i' represents the number of data points in i^{th} cluster.

Step 5: Recalculate the distance between each data point and new obtained cluster centers.

Step 6: If no data point was reassigned then stop, otherwise repeat from step 3.

In classification, the idea is to sort data into groups. "new, efficient burst-mode signal detection scheme" that utilizes a two-step data clustering method based on a K-means algorithm. The burst-mode signal detection problem can be expressed as a binary hypothesis determining if a bit is 0 or 1. Further, although they could use maximum likelihood sequence estimation (MLSE) to determine the class, it "is very computationally complex, and not suitable for high-speed burst-mode data transmission." There are some difficulties in using k-means for clustering data. We can see several mentions of this in current and past research, and an oft-recurring problem has to do with the initialization of the algorithm. In order to be reusable and vary the parameters, the experiment starts by specifying a minimum and maximum number of clusters and a minimum and maximum size for a Sample. Further, it needs a constant to transform a random mean, and another for random standard deviation. These just multiply a pseudorandom number between 0...1 to increase the range of parameters available. It then generates a random number of Samples with random mean and variance between the supplied parameters discussed above. Finally, it combines all the samples into one and passes control to the two algorithms to find k. The algorithms attempted to use nothing but the combined Sample as input. Gives best result when data set are distinct or well separated from each other.

Two ideas struck me as plausible ways to find k, given nothing but a sample, although both of them relied on using standard deviation. The first idea sorts the examples in the combined sample and simply iterates over each of them, adding examples to a new working sample if they are within 3 standard deviations of the current mean, since about 99 percent of the points in a given normal distribution should be within that range. When an example falls outside that range, another working sample is created, and the process continues. When all the examples have been iterated over, the function returns the number of working samples it used. Further, it assumes a minimum of 50 examples in each sample (to calculate the initial mean and standard deviation), and this parameter can be varied depending on knowledge of the data. The Java source code for this function (or the entire experiment) is available upon request.

The second idea takes the standard deviation of a sample, and recursively calls itself using smaller ones as the standard deviation decreases. Since it is easier to understand a recursive function by viewing it as opposed to reading a description, the source code for it is provided in Fig below. Hadoop is leading open source map reduce. Data access via Map Reduce streaming. Map Reduce-This segment will retrieve data from storage, process it, and transfer its results to the storage. Extracting the result-Once processing is completed, for the result to be useful to humans, it must be retrieved from the storage and presented. Storage-Map Reduce requires storage from which to retrive data and in which to store the obtained results of the computation. The data predicted by Map Reduce is not the relational data as

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 12, December 2015

generally used by conventional database system. Instead, data is consumed in chunks, which are then divided among nodes and fed to the map phase as key value pairs. This data does not need a schema, and may be formless.

```
private static int findK(Sample s, double minSD)
{
    // Idea 2: split sample while standard deviation decreases
    double tolerance = 0.001, thisSD;

    s.sort();

    if (minSD < 0) minSD = s.getStandardDeviation();

    Sample workingSample1 = getHalf(1, s);
    Sample workingSample2 = getHalf(2, s);

    thisSD = workingSample1.getStandardDeviation();
    if (thisSD+tolerance < minSD)
        return 1 + findK(workingSample1, thisSD);

    thisSD = workingSample2.getStandardDeviation();
    if (thisSD+tolerance < minSD)
        return 1 + findK(workingSample2, thisSD);

    return 0;
}
```

Fig 1: Recursive FindK() function

At first glance it looks like the function expects an initial standard deviation, but in fact, it is used such that on the first run, that parameter is initialized to -1. Therefore, it will calculate the initial standard deviation based on the entire sample. Within it there is a variable, tolerance, which is used to indicate how close the smaller samples' standard deviation should be to the last standard deviation to continue splitting the sample. This value is chosen arbitrarily, and it is likely the experiment results could have been better had I attempted to let the program "learn" a good value to use.

VI. RESULT AND DISCUSSION

Presented here in tabular form are the results of 20 experimental runs. More were done, of course, but due to space requirements, only 20 are presented. In both sets, the number of samples was uniformly random between 5 and 12, the size of each sample varied from 50 to 100 and the mean and standard deviation of each sample spanned 0-100 and 0-5, respectively. In the first set of ten runs, the standard deviations of the initial samples were allowed to vary. The second set used a fixed standard deviation of two.

Set 1: Variable Standard Deviation										
Run:	1	2	3	4	5	6	7	8	9	10
k:	6	7	11	6	11	6	10	7	7	8
Iterative:	3	11	2	7	8	8	6	3	22	6
Recursive:	8	8	9	8	9	8	8	8	8	8

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 12, December 2015

Set 2: Fixed Standard Deviation = 2										
Run:	1	2	3	4	5	6	7	8	9	10
k:	9	7	5	11	11	9	7	7	6	10
Iterative:	8	8	14	21	10	11	7	4	2	5
Recursive:	8	8	8	9	8	8	8	8	8	8

As the data show, neither algorithm was particularly successful at correctly finding k. The recursive algorithm is a lot less erratic, and seems to be closer to finding k than its iterative counterpart. It should be noted, however, that in several of the cases where the recursive algorithm found too few clusters, particularly when it was one or two off, looking at the individual generated clusters reveals that actually, it was very good - just that two or three of the means were incredibly close, and the algorithm could not differentiate between all the sets. For example, in the first run of the fixed standard deviation, one of the means was 87.68 with another at 89.78. In this case, it is unlikely even a human could have differentiated between the two. Similarly, in the sixth run of the same set, two of the generated means were less than a standard deviation apart. However, a glaring issue with the recursive trials is that it appears to always hover around eight or nine - indicating that perhaps the tolerance was so low, it encouraged splitting the sample all the time. However, even varying the side of the equation in which the tolerance was applied (or subtracting it instead of adding it) did not change the variety of results obtained.

Analysis of multiple runs of this experiment shows unexciting results as well. Using 100 trials, it was found that about 10 percent of the time, the iterative algorithm guesses correctly. With only 8 possible values to choose from, this is worse than guessing if the algorithm had known how many it had to choose from. Among the times that it counted the wrong number of clusters, it was off by an astounding average of 45.3 percent. On the other hand, the recursive algorithm fared slightly better. Using the same 100 trials, it correctly identified k 16 percent of the time, and averaged only 19.3 percent difference among the times it was wrong. Clearly however, this is nothing to write home about. At this point, it was clear that at least some of the time, the recursive algorithm appeared to be finding k correctly, except that some of the means were too close to each other to distinguish between them. Therefore, another 100 trials were run, this time randomly generating the mean of the first sample, and adding a fixed number to each successive one to ensure a minimum distance between each sample. Discouragingly, the results weren't much better. The iterative algorithm again only correctly found k nine percent of the time, and averaged a whopping 71 percent difference when it was wrong. The performance of the recursive algorithm did improve slightly - this time getting k correct 26 percent of the time, and only being "off" by an average of 16.5 percent the times it was incorrect.

VI. CONCLUSION AND FUTURE WORK

CONCLUSION

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 heterogeneous and diverse data sources, 2) autonomous with distributed and decentralized control, and 3) complex and evolving in data and knowledge associations. Such combined characteristics suggest that Big Data require a “big mind” to consolidate data for maximum values. Regard Big Data as an emerging trend and the need for Big Data mining is arising in all science and engineering domains. With Big Data technologies, we will hopefully be able to provide most relevant and most accurate social sensing feedback to better understand our society at real time.

- To conclude about the data set ,
- Accuracy of clustered data. Out of the total data set
- Accuracy of classification HACE theorem

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 12, December 2015

SCOPE FOR FUTURE WORK

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. In other situations, privacy concerns, noise, and errors can be introduced into the data, to produce altered data copies. Developing a safe and sound information sharing protocol is a major challenge. At the model level, the key challenge is to generate global models by combining locally discovered patterns to form a unifying view. This requires carefully designed algorithms to analyze model correlations between distributed sites, and fuse decisions from multiple sources to gain a best model out of the Big Data. At the system level, the essential challenge is that a Big Data mining framework needs to consider complex relationships between samples, models, and data sources, along with their evolving changes with time and other possible factors. A system needs to be carefully designed so that unstructured data can be linked through their complex relationships to form useful patterns, and the growth of data volumes and item relationships should help form legitimate patterns to predict the trend and future.

REFERENCES

[1] R. Ahmed and G. Karypis, "Algorithms for Mining the Evolution of Conserved Relational States in Dynamic Networks", Knowledge and Information Systems, vol. 33, no.3, pp.603-630, Dec 2012.
[2] M.H. Alam, J.W. Ha, and S.K. Lee, "Novel Approaches to Crawling Important Pages Early", Knowledge and Information Systems, vol.33, no.3, pp.707-734, Dec 2012.
[3] S. Aral and D. Walker, "Identifying Influential and Susceptible Members of Social Networks" , Science, vol. 337, pp.337-341, 2012.
[4] A. Machanavajjhala and J.P. Reiter, "Big Privacy: Protecting Confidentiality in Big Data" , ACM Crossroads, vol.19, no.1, pp.20-23, 2012.
[5] S. Banerjee and N. Agarwal, "Analyzing Collective Behavior from Blogs Using Swarm Intelligence", Knowledge and Information Systems, vol. 33, no.3, pp.523-547, Dec 2012.
[6] E. Birney, "The Making of ENCODE: Lessons for Big-Data Projects", Nature, vol.489, pp.49-51, 2012
.[7] J. Bollen, H. Mao, and X. Zeng, "Twitter Mood Predicts the Stock Market", J. Computational Science, vol.2, no.1, pp.1-8, 2011.
[8] S. Borgatti, A. Mehra, D. Brass, and G. Labianca, "Network Analysis in the Social Sciences", vol.323, pp.892-895, 2009.
[9] J. Bughin, M. Chui, and J. Manyika, Clouds, Big Data, and Smart Assets: Ten Tech-Enabled Business Trends to Watch. McKinSey Quarterly, 2010.
[10] D. Centola, "The Spread of Behavior in an Online Social Network Experiment", Science, vol.329, pp.1194-1197, 2010.
[11] E.Y. Chang, H. Bai, and K. Zhu, "Parallel Algorithms for Mining Large-Scale Rich-Media Data", Proc. 17th ACM Int'l Conf. Multimedia, pp.917-918, 2009.
[12] R. Chen, K. Sivakumar, and H. Kargupta, "Collective Mining of Bayesian Networks from Distributed Heterogeneous Data", Knowledge and Information Systems, vol.6, no.2, pp.164-187, 2004.
[13] Y.-C. Chen, W.-C. Peng, and S.-Y. Lee, "Efficient Algorithms for Influence Maximization in Social Networks", Knowledge and Information Systems, vol.33, no.3, pp.577-601, Dec 2012.
[14] C.T. Chu, S.K. Kim, Y.A. Lin, Y. Yu, G.R. Bradski, A.Y. Ng, and K. Olukotun, "Map-Reduce for Machine Learning on Multicore", Proc. 20th Ann. Conf. Neural Information Processing Systems (NIPS'06), pp. 281-288, 2006.
[15] G. Cormode and D. Srivastava, "Anonymized Data: Generation, Models, Usage", Proc. ACM SIGMOD Int'l Conf. Management Data, pp. 1015-1018, 2009.

BIOGRAPHY

K.Yogeshwaran is a Assistant Professor in the ECE Department, Kalaignar Karunanidhi Institute of Technology, Coimbatore, India. He received Master of Engineering (ME-VLSI DESIGN) degree in 2013 and His research interests are VLSI Design, communication system, networking, Data Mining, Big Data Analysis etc...

R. Gowri is a Assistant Professor in the ECE Department, Tejaa Shakthi Institute of Technology, Coimbatore, India. She received Master of Engineering (ME-VLSI DESIGN) degree in 2013 and Her research interests are VLSI Design, communication system, Data Mining, Big Data Analysis, Image Processing etc...