DBMD implementation for Big Data Mining in the CEASE method

[1]Dr.N.Krishnaveni, [2]Ms.A.Ishwariya, [3]Ms.R.Priyanka

Assistant Professor
Department of Computer Science and Engineering
P.S.R.Engineering College

***Abstract*: The fundamental tools to discover knowledge from big data was matrix composition. Here data generated by modern applications via cloud Computing. However, it is still inefficient or infeasible to process very big data using such a method in a single machine or through virtual machines. Moreover, big data are often distributedly collected data from various data centers and stored on different machines via scheduling algorithms. Thus, such data generally bear strong heterogeneous noise. It is essential and useful to develop distributed matrix decomposition for big data analytics. Such a method should scale up well, model the heterogeneous noise, and address the communication issue in a distributed system. To this end, we propose a distributed Bayesian matrix decomposition model (DBMD) for big data mining and clustering. Specifically, we adopt three strategies to implement the distributed computing including 1) the accelerated gradient descent, 2) the alternating direction method of multipliers (ADMM), and 3) the statistical inference. We investigate the theoretical convergence behaviors of these algorithms. To address the heterogeneity of the noise, we propose an optimal plug-in weighted average that reduces the variance of the estimation. Finally Comparison made between these algorithms to understand the result between them.**

INTRODUCTION

Data visualization Technology was used in this project to handle the data processing. Data visualization technique that combines graph-based topology representation and dimensionality reduction methods to visualize the intrinsic data structure in a low-dimensional vector space. The application of graphs in clustering and visualization has several advantages. A graph of important edges (where edges characterize relations and weights represent similarities or distances) provides a compact representation of the entire complex data set. This text describes clustering and visualization methods that are able to utilize information hidden in these graphs, based on the synergistic combination of clustering, graph-theory, neural networks, data visualization, dimensionality reduction, fuzzy methods, and topology learning. The work contains numerous examples to aid in the understanding and implementation of the proposed algorithms.

The recent development of methods for extracting precise measurements of spatial gene expression patterns from three-dimensional (Input) image data opens the way for new analyses of the complex gene regulatory networks controlling animal development. We present an integrated visualization and analysis framework that supports user-guided data clustering to aid exploration of these new complex data sets. The interplay of data visualization and clustering-based data classification leadsto improved visualization and enables a more detailed analysis than previouslypossible. We discuss 1) the integration of data clustering and visualization into one framework, 2) the application of data clustering to Input gene expression data, 3) the evaluation of the number of clusters k in the context of Input gene expression clustering, and 4) the improvement of overall analysis quality via dedicated post processing of clustering results based on visualization. We discuss the use of this framework to objectively define spatial pattern boundaries and temporal profiles of genes and to analyze how mRNA patterns are controlled by their regulatory transcription factors.

RELATED WORK

Matrix decomposition is one of the fundamental tools to discover knowledge from big data generated by modern applications. However, it is still inefficient or infeasible to process very big data using such a method in a single machine. Moreover, big data are often distributedly collected and stored on different machines. Thus, such data generally bear strong heterogeneous noise. It is essential and useful to develop distributed matrix decomposition for big data analytics. Such amethod should scale up well, model the heterogeneous noise, and address the communication issue in a distributed system. To this end, we propose a distributed Bayesian matrix decomposition model (DBMD) for big data mining and clustering. Specifically, we adopt three strategies to implement the distributed computing including 1) the accelerated gradient descent, 2) the alternating direction method of multipliers (ADMM), and 3) the statistical inference. We investigate the theoretical convergence behaviors of these algorithms. To address the heterogeneity of thenoise, we propose an optimal plug-in weighted average that reduces the variance of the estimation. Synthetic experiments validate our theoretical results, and real-world experiments show that our algorithms scale up well to big data and achieves superioror competing performance compared to two typical distributed methods including Scalable-NMF and scalable k-means++.

ALGORITHM
k-means clustering algorithm

K-means is one of the simplest unsupervised learning algorithms that solvethe well-known clustering problem. The procedure follows a simple and easy way to classify a given data set through a certain number of clusters (assume k clusters) fixed apriori. The main idea is to define k centers, one for each cluster. These centersshould be placed in a cunning way because of different location causes different result. So, the better choice is to place them as much as possible far away from each other. The next step is to take each point belonging to a given data set and associateit to the nearest center. When no point is pending, the first step is completed and an early group age is done. At this point we need to re-calculate k new centroids as barycenter of the clusters resulting from the previous step. After we have these k new centroids, a new binding has to be done between the same data set points and the nearest new center. A loop has been generated. As a result of this loop we may notice that the k centers change their location step by step until no more changes aredone or in other words centers do not move any more. The matrix decomposition methods mentioned above have little relevance to the underlying computational architecture. They assumed that the program is running on a single machine, and an arbitrary number of data points are accessible instantaneously. However, the huge size of data often makes it impossible to handle all of them on a single machine. Many applications collect data distributedly from different sources (e.g., labs, hospitals). The communication between them is expensive due to the limited bandwidth, and direct data sharing also raise privacy concern. Moreover, data collected from different sources often bear strong heterogeneous noise. Therefore, developing efficient matrix decomposition methods in a distributed system is essential. are sequential, which limits its applicability to big data. Generally, scaling the k-means algorithm to distributed data is relatively easy due to its iterative nature.Distributed k-means algorithms often split data by samples. The distributed versions of k-means often focus on reducing the number of passes needed to obtain a good initialization by sampling, e.g., DKEM [24] and scalable k-means++ [25]. Recently, Yu *et al*. proposed a distributed BPMF by splitting the data by samples and employed a stochastic alternating direction method of multipliers (ADMM) to solve it [26]. But it is common in the filtering collaborative that the user-item data *X 2 Rm×n* are very spare and of large *m* and *n*. Splitting *X* by rows is only efficient for the tall-and-skinny matrix due to the communication load. To reduce thecommunication load, some studies split the data matrix *X* over both columns and rows, and then store the blocks of *X* distributedly on nodes. Then they employed distributed Monte Carlo Markov Chain methods (MCMC) for inference. There also exist distributed NMF variants that split *X* into blocks. Moreover, Benson *et al*.proposed an approximated and scalable NMF algorithm for tall-and-skinny matrices. Finally, this algorithm aims at minimizing an objective function know assquared error function given by:

$$J(V) = \sum_{i=1}^{c} \sum_{j=1}^{c_i} \left(\left\| x_i - v_j \right\| \right)^2$$

where,

'$||x_i - v_j||$' is the Euclidean distance between x_i and v_j.

'c_i' is the number of data points in i^{th} cluster.

'c' is the number of cluster centers.

Algorithmic steps for k-means clustering

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

Calculate the distance between each data point and cluster centers.

Assign the data point to the cluster center whose distance from the cluster centeris minimum of all the cluster centers.
Recalculate the new cluster center using:

where, 'c_i' represents the number of data points in i^{th} cluster.
Recalculate the distance between each data point and new obtained clustercenters.
If no data point was reassigned then stop, otherwise repeat from step 3).

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

RESULT

A continual testing of individual software components when developed, involved evaluating and reviewing the software prototype identifying problem situations. With the project methodology using an evolutionary development strategy to systematically progress between each iteration, a standard set of small scale testing procedures were in place to deal with erroneous defects with the source code, filtering out unintended functionality. With each characteristic continually incorporated and verified this ensured the software prototype deliverable retainedand prioritised further development attributes. To ascertain the prototype produced throughout the development phase was correct, a series of test were formalised and conducted.
Involvement and creation of small scale throw away prototyping werethe heart of the project quickly identifying and defining

pathways for the development to maintain. This method vastly improved and distinguished cleardirections to take, with the minimum requirements and possible extensions reflected upon within each small prototype using the preceding design phase. Although the design phase in the initial prototypes did not provide an accurate correlation to what was actually produced in the UML models, it still enabled identification and analysisof the user interface and output documents produced needing to be factored into the prototype.

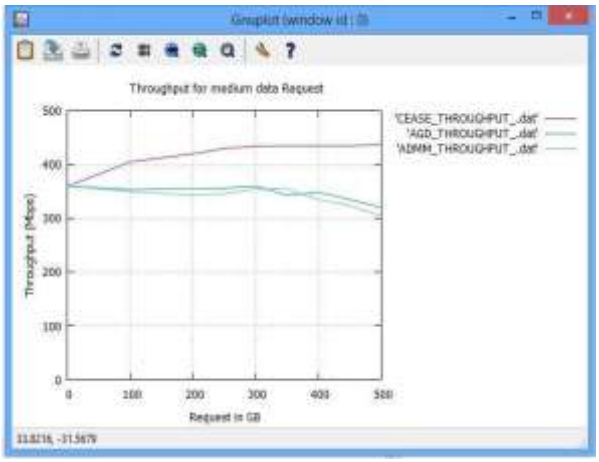

Fig: Throughput Comparision For Medium Data Request

Fig. show the throughput consumption and data size utilization at the cloud data center, respectively. The energy consumption in the case of our proposed K-MEAN CLUSTER based CEASE algorithm has high throughput when compared to ADG and ADMM migration because we set the dynamic upper threshold value by computing the median absolute deviation, and interquartile range of past data respectively. The throughput and the data size was high in the data mining and smartclustering.

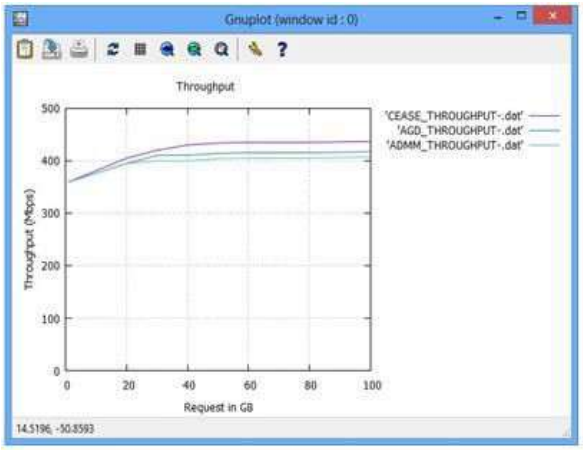

Fig: Throughput Comparison For Small Data Request

Fig. show the throughput consumption and small data size utilization at the cloud data center, respectively. The energy consumption in the case of our proposed K-MEAN CLUSTER based CEASE algorithm has high throughput when comparedto ADG and ADMM migration because we set the dynamic upper threshold value bycomputing the median absolute deviation, and interquartile range of past data respectively. The throughput was high even at low data size was high in the data mining and smart clustering.

Below fig show the throughput consumption and small data size utilization at the cloud data center, respectively. The energy consumption in the case of our proposed K-MEAN CLUSTER based CEASE algorithm has high throughput when compared to ADG and ADMM migration because we set the dynamic upper threshold value bycomputing the median absolute deviation, and interquartile range of past data respectively. The throughput was high even at high data size in the data mining and smart clustering.. it reach around 420 MbpS even at 1500 GB data requested by many users

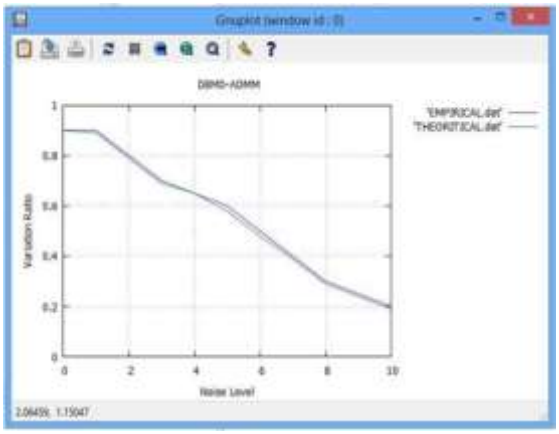

Fig :Throughput Comparision For Large Data Request

Fig :The variance ratio var($W\sim$)= var(W^{-}) on a series of synthetic datasets**T**here is variation in therotical and the pratical implemention in Bayesian matrix decomposition model (DBMD) for big data mining and clustering in the alternating direction method of multipliers (ADMM) method. When noise level was increasing its change in value make a huge impact in the data mining and throughput.

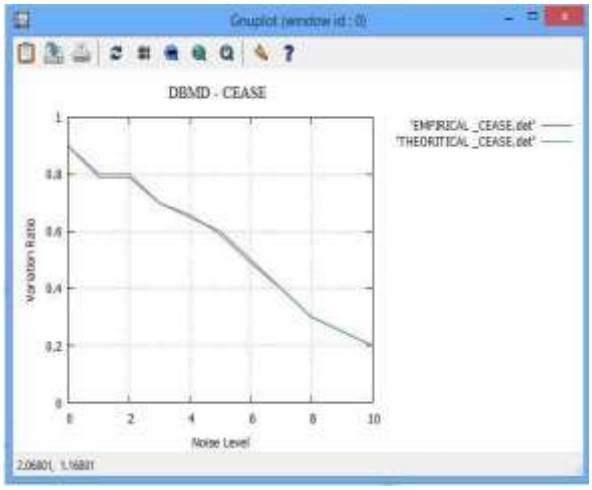

Fig : The variance ratio var($W\sim$)= var(W^{-}) on a series of synthetic datasets

There is only minor variation in therotical and the pratical implemention in Bayesianmatrix decomposition model (DBMD) for big data mining and clustering in the the communication-efficient accurate statistical estimation (CEASE) method. When noise level was increasing there is no change in value which make a great in the datamining and throughput even at the high data request. When number of users increases the number of data request in size also increases. It tends to creating a noise level at some point our algorithm clearly handle the noise level hence it gives high QoS factors.

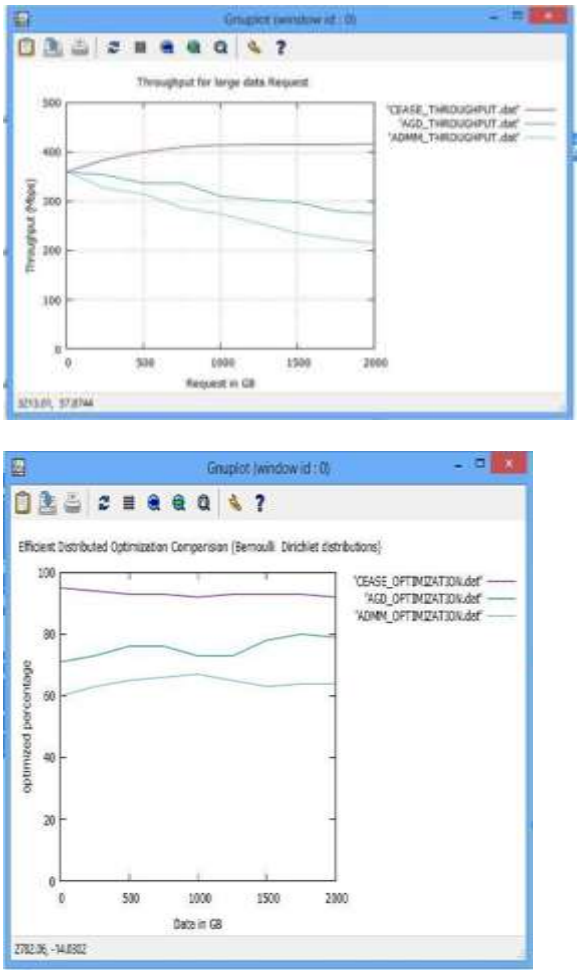

Fig: Optimization Comparison of Each Algorithm

Optimization of each algorithm was plotted between optimization percentage vs data in GB. These algorithm was tested under heavy data requirement hence it contain many number of user. Each user requesting different type of file with size variation. We should then consider which optimization strategy is suitable for current partitioned data. However, few studies explored different optimization strategies and elaborated their difference. Even more serious is that few methods tackle the heterogeneity of noise among the distributed data. CEASE algorithm perform good optimization when compared to the other AGD and ADMM even at high data requirement at various VMs.

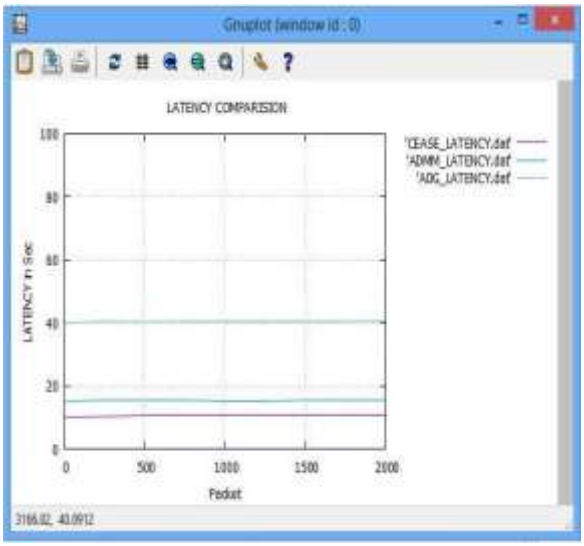

Fig: Optimization Comparison of Each Algorithm

When the latency between the central machine and nodes is high the algorithmmake traffic in the network. Moreover it affect the QoS in the entire system. So herewe plotted the latency vs packet in the graph. The implemeded result shows that CEASE has a very low latency compared to the other algorithm. So as far as all concern our algorithm perform well in all aspects.

CONCLUSION

We proposed a distributed Bayesian matrix decomposition model for big data mining and clustering. Three distributed strategies (i.e., AGD, ADMM and CEASE) were adopted to solve it. Convergence rates of AGD and ADMM depend on different structural parameters and thus have different behaviors. In short, CEASE converges faster with the number of instances on each node machine increasing, the convergence rate of CEASE doesn't change much, but the convergence rate of AGD and ADMM change much. Empirically, CEASE also converges faster with thenumber of instances growing. To tackle the heterogeneous noise in the data, we propose an optimal plug-in weighted average scheme that significantly reduces the variance of the estimation. The proposed algorithms scale up well. The real-world experiments demonstrate that the proposed algorithms achieve superior or competitive performance. Both the Bayesian prior and the weighted average strategies reduce the influence of the highly noisy data.

REFERENCES

[1] A Ismail, "Utilizing Big Data Analytics as a solution for Smart Cities," in 3rd MEC International Conference on Big Data and Smart City, 2016, pp. 1-5.
[2] A Sharif, J Li, M Khalil, R Kumar, and M.I Sharif, "Internet of Things- Smart Data minig Management System for Smart Cities using Big Data Analytics," in IEEE, 2017, pp. 281-284.
[3] C Xu, X Huang, J Zhu, and K Zhang, "Reseach on the Construction of Sanya Smart Tourism City based on Internet and Big Data," in International Conference on Intelligent Transportation, Big Data & Smart City, 2018, pp. 125-128.
[4] P Papadimitriou and H.G Molina, "Data Leakage Detection," IEEE Transaction on Knowledge and Data Engineering, vol. 23, no.1, pp. 51-63, January 2011.
[5] J Croft and M Caesar, "Towards Practical Avoidence of Information Leakage in Enterprise Networks," in 6th USENIX conference Hot Topics Securty (HotSec), CA,USA, 2011, p. 7.
[6] I Gupta and A.K. Singh, "A Probabilistic Approach for Guilty Agent Detection using Bigraph after Distribution of Sample Data," in Procedia Computer Science, vol. 125, 2018, pp. 662-668.
[7] K Kaur, I Gupta, and A.K. Singh, "A Comparative Evaluation of Data Leakage/Loss prevention Systems (DLPS)," in 4th International Conference on Computer Science & Information Technology (CS & IT-CSCP), Dubai, UAE, 2017, pp. 87-95.
[8] M Backes, N Grimm, and A Kate, "Lime: Data Lineage in the Malicious Environment," in 10th International Workshop Security Trust Management, 2014, pp. 183-187.
[9] A. Kumar, A. Goyal, A. Kumar, N. K. Chaudhary, and S., S.Kamath, "Comparative Evaluation of Algorithms for Effective Data Leakage Detection," in IEEE Conference on Information and Communication Technologies (ICT 2013),vol. 13, 2013, pp. 177-182.
[10] S Sholla, R Naaz, and M.A Chishti, "Semantic Smart City: Context Aware Application Architecture," in 2nd International Conference on Electronics, Communication and Technology (ICECA), 2018, pp. 721-724.
[11] A. Shabtai, Y. Elovici, and L. Rokach,. NewYork: Springer, 2012, ch. Introduction to Information Security and Data Leakage, pp. 1-87.
[12] X. Shu and D. Yao, "Data Leak Detection as a Service," in Springer, International Conference on Security and Privacy in Communication Systems, 2012,pp. 222-240.
[13] F Liu, X Shu, D Yao, and A.R. Butt, "Privacy- Preserving Scanning of Big Content for Sensitive Data Exposure with MapReduce," in 5th ACM Conference Data Application Security, Privacy (CODASPY), Texas, USA, 2015, pp. 195-206.
[14] X. Shu, J. Zhang, D. Yao, and W. C. Feng, "Fast Detection of Transformed Data Leaks," IEEE Transactions on Information Forensics and Security, vol. 11, no. 3, pp. 528-542, March 2016.
[15] M Gafny, A Shabtai, L Rokach, and Y Elovici, "Detecting Data Misuse by Applying Context- Based Data Linkage," ACM workshop Insider Threats, pp. 3-12, 2010.
[16] K Kaur, I Gupta, and A.K. Singh, "A Comparative Study of the Approach Provided for Preventing the Data Leakage," vol. 9, no. 5, pp. 21-33, 2017.
[17] X. Shu and D. Yao, "Privacy-Preserving Detection of Sensitive Data Exposure,"IEEE Transactions on Information forensics and Security, vol. 10, no. 5, pp. 1092- 1103, May 2015.
[18] A Harel, A Shabtai, L Rokach, and Y Elovici, "M-Score: A MiuseabilityWeight Measure," IEEE: Dependable Secure Comput., vol. 9, no. 3, pp. 414-428, 2012.
[19] K Gupta and A Kush, "A Review on Data Leakage Detection for Secure," International Journal of Engineering and Advanced Technology (IJEAT), vol. 7, no. 1, pp. 153-159, October 2017.
[20] K Gupta and A Kush, "Performance Evaluation on Data Leakage Detection for Secure Communication," in 5th International Conference on " Co mputing for Sustainable Global Develop ment: INDIACom, New Delhi, India, 2018, pp. 3957-3960.
[21] K Kaur, I Gupta, and A.K. Singh, "Data Leakage Prevention: E-Mail Protectionvia Gateway," in IOP Co.