Clustering of Documents with Sequential Update in Distributed Environment

[1]Kaveri M. More, [2]Prof. R. P. Dahake

[1]Student, [2]Professor

[1] Department of Computer Engineering,

[1]MET Institute of Engineering, Nasik, India

Abstract : ***Co-clustering is powerful tool developed for 2D co-occurrences as well as bipartite data. There is need of significant computational resources for huge data processing. Traditional approaches of co-clustering suffered from issues of graph partitioning. From study of existing system expectation-maximization (EM) algorithms such as, k-means clustering algorithms gives the provable evidence with the sequential updates. Without affecting result, it reduces computational cost. In this proposed approach, sequential updates in a distributed environment is proposed using parallelize Alternate Minimization Co-Clustering (AMCC) algorithm. The AMCC algorithm is again extended using fast nonnegative matrix tri-factorization (FNMTF). The proposed work will also contribute attribute based filtering of input dataset using knn attribute filtration at the time of row and column shuffling process. Performance of the proposed system is evaluated on local machine and network of machines. Proposed AMCC algorithm with feature selection technique achieves better efficiency than existing algorithm.***

IndexTerms :- Co-Clustering, concurrent updates, sequential updates, distributed framework.

I. INTRODUCTION

In the domain of data mining Co-clustering is the capable tool for mining co-occurrence of 2D data. Clustering has lots of applications such as, in recommendation system, text mining, and gene expression. Clustering is an iterative method to refine data and collect data points into cluster. The most popular K-means clustering algorithm is based on current cluster information. Clustering assignments are gathered and utilized at the final stage of clustering. Assignments of clustering gathered at the time of refinement of clustering till clustering get stable.

There are two classes of clustering approach such as; first class clustering and updates in clusters. In first class update, the cluster information is updated after all input points are updated in cluster. Updates in cluster is referred as, concurrent updates. Second class of clustering, updates the cluster when points are changes its assignment of clusters. It is referred as, "Sequential updates".

Several techniques have been proposed previously which gives the proof of expectation maximization (EM) algorithms such as, K-means algorithm. With sequential updates it decreases the computational cost of processing without affecting to resulting solution. In this research work, sequential updates referred as, alternate minimization co-clustering (AMCC) algorithms [9]. It is variants of EM algorithms. The proposed AMCC algorithm is assemble of sequential updates. Convergence property of co-clustering algorithms cannot provide guarantee due to inconsistency in clustering information. It also brings synchronization overheads during information synchronization whenever cluster assignment get changed. These are reasons behind AMCC algorithm which cannot work with sequential updates in a distributed way.

Therefore, in this work we proposed a new approach to parallelize sequential updates for AMCC algorithms.

Dividing clusters:

In this approach, clustering problem is get divided into independent task and each task is allocated to the individual worker. The process independent task, row or column clusters are divided into multiple non-overlapping subsets at the beginning of iteration. Each worker then performed sequential updates with row or column cluster.

From performance analysis of approach, it can prove that this approach can preserve the convergence properties of AMCC algorithms. Therefore, proposed Co-clustering approach can support efficient implementations of AMCC algorithms with sequential updates. It can also provide abstraction for AMCC algorithms with sequential updates and allows programmers to specify the sequential update operations via simple APIs.

II. Related Work:

R.NEAL et al [1], discussed about expectation maximization (EM) algorithm for identification of maximum similar parameters in the process of clustering. A problem in which variables were unobserved is considered. EM algorithm started estimation from beginning determination. E is referred as expectations in EM algorithm which find the distribution for unobserved variables whereas, M is maximization step that re-estimates the parameters to be those with maximum likelihood.

I. Dhillon, et al [2], implemented a simple and top-down computationally efficient principled algorithm. It associates with the row and columns of clustering all stages. It gives the assurance of reaching towards finite number of steps. They discussed about co-clustering algorithm quiet issues of high dimensionality and sparsity by presenting results on joint-document clustering. To minimize the dimensionality, it estimated less parameters than the standard "one-dimensional" clustering approaches.

A. Banerjee et al.[3], represented partitional co-clustering algorithm. In proposed algorithm minimum Bregman information (MBI) principle is generalizes the principles of max entropy and standard list squares. Data is collected on the basis of relationship of multiple entities. These relational entities are represented as tensor. It is specific to the metrics. They implemented Meta co-clustering algorithm based on AM i.e. alternate minimization and then described applications of co-clustering such as, predicting missing values, and categorical data metrics compression.

B. Kwon et al [4] proposed scalable co-clustering approach. BCC approach i.e. Bregman co-clustering algorithm is also proposed by then which gives basic framework for co-clustering algorithm. They have parallelized twelve co-clustering algorithms by utilizing MPI i.e. message passing interface. To demonstrate the speedup performance in terms of different parameter settings scalability of synthetic datasets also has been validated. To describe batch update scenario of BCC framework, SBCC is the Sequential Bregman Co-clustering algorithm is proposed which contains two MSSRCC algorithms i.e. Minimum Sum-Squared Residue Co-clustering algorithms. It is possible to obtain near linear speedup for all the considered dense datasets using equal partitioning-based load balancing strategy.

M. Deodhar et al [5]., demonstrated the problem of predicting customer behavior across products. They represented model based co-clustering/meta algorithm, to improve both cluster assignment and fit of the models. The proposed approach not only enhance accuracy and reliability but also improved interpretability. The partitioning is based on apriori algorithm which separates the segmentation routine. The fulfillment of proposed work, the main aim is to extract synthetic and marketing data such as, to analyze microarray data with gene and experiment annotations, in the settings of social networking. Co-clustering model is used for co-clustering model as well as for predictor or classifier with the specific choices.

H. Wang, et al. [6], suggested Fast Nonnegative Matrix Tri-factorization (FNMTF) approach. It is cluster data side and feature side input data matrix. This approach is decoupled into number of smaller sub problem which required less metrics multiplication. Generally, it works on large-databases. In this work they defined a future work to incorporate manifold information and proposed Locality Preserved FNMTF (LP-FNMTF) method. Locality Preserved FNMTF (LP-FNMTF) approach, by which the clustering performance is improved. The promising results in extensive experimental evaluations validate the effectiveness of the proposed methods. Rather than applying traditional nonnegative constraints on the factor matrices of NMTF they inhibit them into cluster indicator matrices. Based on the distributions of features points are clustered together. To address the problem in clustering they have introduced two algorithms from both first is algorithm to solve J5 and second is algorithm to solve J7.

Y. Zhang, et al.[7], discussed about iMapReduce technique. Iterative algorithms under huge cluster environment are supported by iMapReduce. It discovers the features of iterative algorithm and provides the built-in support for them. The proposed persistent task reduces the initialization overheads by providing efficient data management avoiding shuffling among static data to various tasks. iMap allows asynchronous map task execution when possible. Due to iMap performance of system gets improved. SSSP gives the shortest path whereas, PageRank proposed to rank web pages. Map reduce is the function node, one-to-one mapping is get performed between mappers and reducers. But there are limitations over map reduce algorithm such as scheduling overheads because jobs have to load the input data from DFS and repeatedly depot the output data to DFS.

A. Narang ,et al[8], defined a real-time co-clustering and collaborative filtering approach with high prediction accuracy are computationally challenging issues. To addressed this problem hierarchical approach for online and offline distributed co-clustering as well as collaborative filtering by making theoretical analysis of parallel time complexity is proposed by them. They demonstrated the scalability and real time performance on Netlfix and Yahoo KDD Cup datasets. 3× better performance of baseline MP have been demonstrated by them. For general co-clustering formulation block-average co-clustering is analyzed. For distributed co-clustering a novel hierarchical an approach is described by them. Collaborative filtering is applied for distributed co-clustering for online as well as offline approach. For implementation of clustering they have proposed a very general framework.

Y. Cheng, et al [9], described a node delegation algorithm to identify sub-metrics in expression data which has low mean square residue scores. To identify co-regulation patterns in yeast and human node delegation algorithm can work efficiently. They have also suggested simultaneous clustering known as “bi-clustering”. In the application of biological data, bi-clustering discovers the both genes and conditions.

X. Cheng et al.[10], proposed Co-clusterD approach. It is defined as the binding of two or more types of servers. It can integrate the power of multiple servers and can utilize further to enhance the performance of data storage. Everyone is aware of k-means algorithm used for data clustering, it reduces the computational complexity with powerful provisional evidence that EM i.e. expectations maximization algorithm. With simultaneous updates k-means algorithm reduces the computational cost. Similarly co-clustering is an advanced technique is AMCC i.e. alternative minimization co-clustering algorithm used for sequential updates which can be alternative for EM algorithm and come up with AMCC algorithm for sequential updates. Co-clustering is two mode clustering strategy in data mining. It allows clustering (grouping) of rows and column data simultaneously so that data computational cost can be reduced. With proposed AMCC algorithms they want to prove efficiency of their proposed work.

III. OVERVIEW OF CLUSTERING APPROACH:

To designed a system which work efficiently into distributed environment with sequential updates and improved performance, Co-clustering is used. Clustering is the process of combining similar types of data. Several types of clustering algorithms are available for data clustering such as, k-means, k-menoid etc.

IV. System Architecture:

Following figure 1 represents the architecture of system.

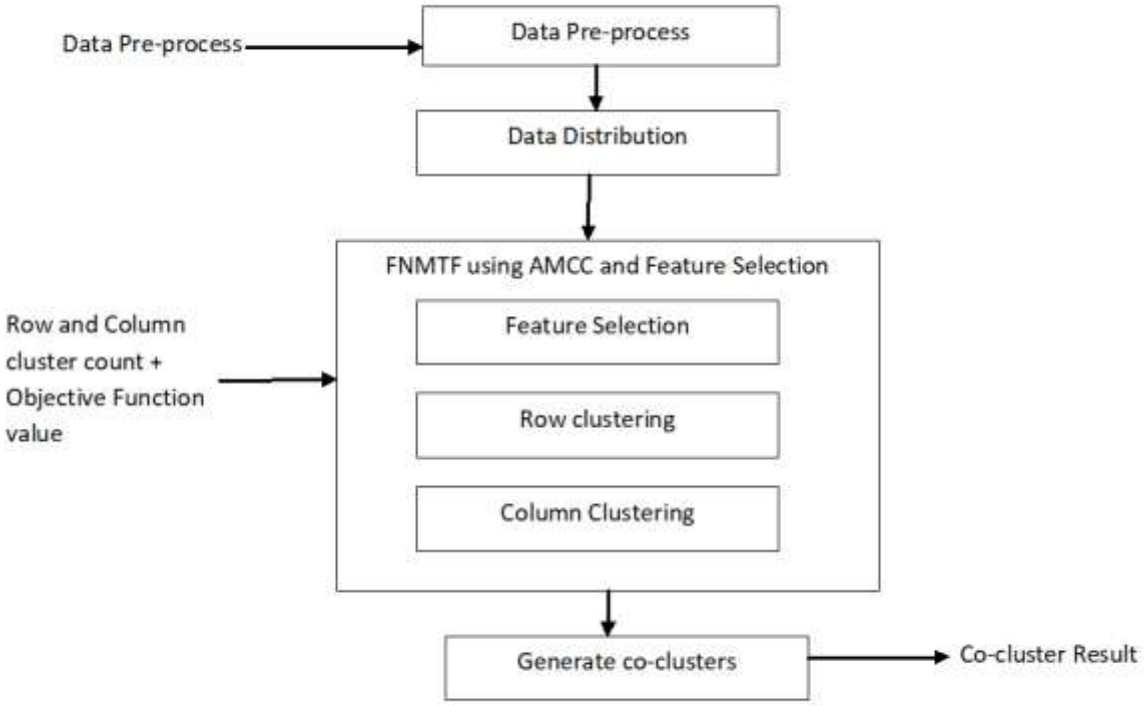


Fig 1.System Architecture

The FNMTF algorithm performs row and column clustering with the help of AMCC algorithm. To reduct the mathematical calculations Feature selection technique is used. To shuffle rows or columns in a dataset Euclidean distance between rows are calculated and using nearest neighbor technique the rows or columns are kept adjacent to each other. To improve the system efficiency, redundant rows and columns are identified and repetitive calculations of neighbor search are reduces using feature selection technique.

System flow of proposed system is given as below:

Step 1: Select the dataset
Step 2: Decide number of workers p (p <= mink/2 , l/2)
Step 3: Distribute subsets of rows Ri and Subsets of columns Ci to worker Wi
Group input data matrix in k rows and l columns
Step 4: AMCC Approach
Step 5. Attribute based filtering is done
Step 6: Worker performs row clustering
Step 7: Sequential update
Step 8: Worker performs column clustering
Step 9: Sequential update
Step 10: Subset of Scc and cluster indicators (Ir and Ic) is combined and synchronized using updater

V.Algorithms:

Let Z be the data matrix of size m X n. k and l are the row and column clustering count.

Let F and G be the factor matrices of size m*k and n*l respectively. S be the matrix of statistics

It can be initialize as:

$$s_{pq} = \frac{\sum_{\{u|\rho(u)=p\}} \sum_{\{v|\gamma(v)=q\}} z_{uv}}{|p| \cdot |q|},$$

----------(eq. 1)

Where |p| and |q| denotes the number of rows in row cluster and number of colunms in column clusters.

The matrix F is updated in each iteration as:

$$f_{up} = \begin{cases} 1, & argmin_p ||z_{u\cdot} - \tilde{\eta_{p\cdot}}||^2, \\ 0, & otherwise, \end{cases}$$

----------(eq. 2)

Where $\eta\sim_p$ is the row cluster prototype

The matrix G is updated in each iteration as:

$$g_{vq} = \begin{cases} 1, & argmin_q ||z_{.v} - \tilde{\mu_{.q}}||^2, \\ 0, & otherwise, \end{cases}$$

----------(eq. 3)

Where μ~q is the column cluster prototype.
The matrix S can be updated in each iteration as:

$$s_{pq} = argmin_{s_{pq}} \sum_{\{u|\rho(u)=p\}} \sum_{\{v|\gamma(v)=q\}} w_{uv} d_{\phi}(z_{uv}, s_{pq}).$$

----------(eq 4)

Where, dø is the distance measure.

Algorithm: FNMTF
5.1 FNMTF with Sequential Update
Input: Z: Dataset
p: Row Cluster Count
q: Column cluster count
Destination location
Output: co-cluster
Processing:
Step 1: Initialize G and F matrices
Step 2: Find Feature set
Step 3: Initialize S using eq. 1

Step 4: Repeat
//Fixing G
For each Zu in Z
Update F using eq. 2
Update S using eq. 4
end

//Fixing F
For each Zu in Z
update G using eq. 3
update S using eq. 4
end

until converges;
Step 5: Save Z as co-cluster information

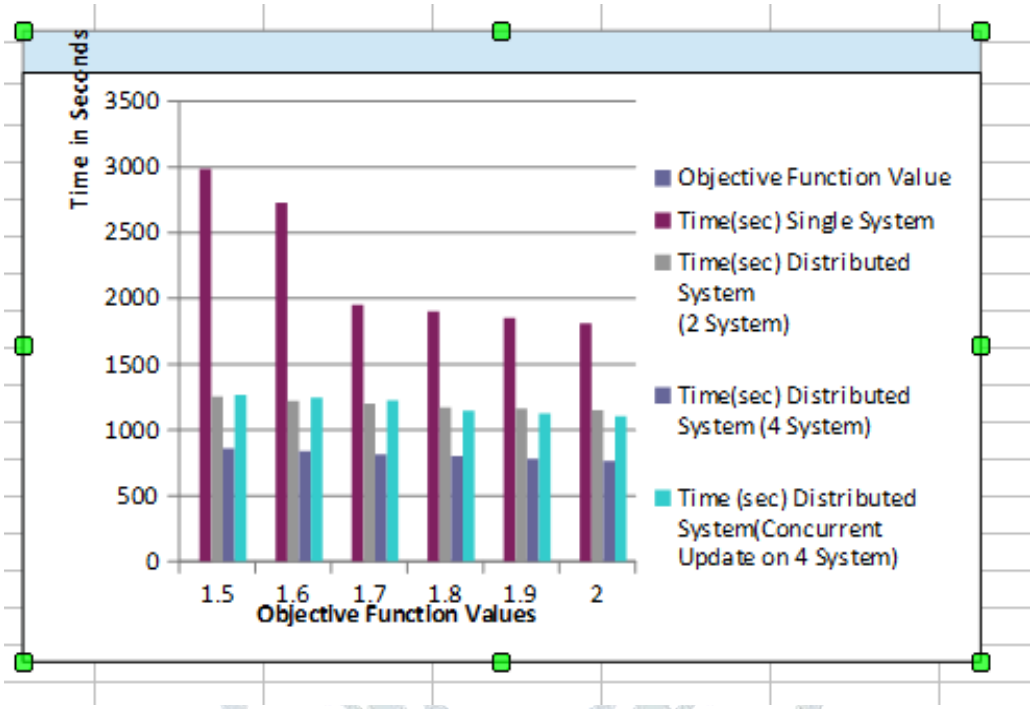
5.2 FNMTF with concurrent update
Input: Z: Dataset
p: Row Cluster Count
q: Column cluster count
Destination location
Output: co-cluster
Processing:
Step 1: Initialize G and F matrices
Step 2: Initialize S using eq. 1
Step 3: Repeat
//Fixing F and G
Update S using eq. 4
end
//Fixing F and S
update G using eq. 3
end
//Fixing G and S
Update F using eq. 2
until converges;
Step 5: Save Z as co-cluster information

VI.Results and Discussion:

Following table shows objective function values and time required for NIPS dataset on single, two and four systems with sequential update. Sequential updates with four systems are also compared with concurrent update with four systems.

Objective Function Value	Time(sec) Single System	Time(sec) Distributed System (2 System	Time(sec) Distributed System (4 System)	Time(sec) Distributed System with concurrent update (4 System)
1.5	2985	1255	860	1268
1.6	2730	1221	839	1247
1.7	1951	1199	814	1225
1.8	1902	1170	801	1148
1.9	1852	1162	783	1125
2	1810	1153	765	1103

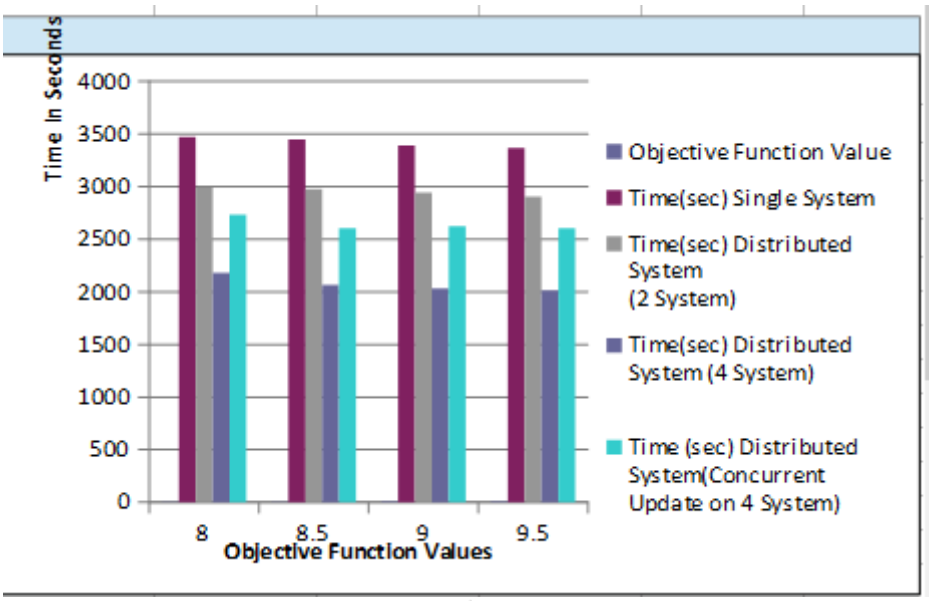
Fig. shows comparison between single, two and four systems with sequential update. Also shows comparison between sequential and concurrent update with four systems. Sequential update with four systems requires less time to create co-clusters on NIPS dataset.



Following table shows objective function values and time required for NIPS dataset on single, two and four systems with sequential update. Sequential updates with four systems are also compared with concurrent update with four systems on KOS dataset.

Objective Function Value	Time(sec) Single System	Time(sec) Distributed System (2 System)	Time(sec) Distributed System (4 System)	Time(sec) Distributed System with concurrent update (4 System)
8	3473	2989	2181	2731
8.5	3448	2975	2062	2602
9	3389	2942	2032	2621
9.5	3365	2901	2011	2602

Fig. shows comparison between single, two and four systems with sequential update. Also shows comparison between sequential and concurrent update with four systems. Sequential update with four systems requires less time to create co-clusters on KOS dataset.



Conclusion:

Co-clustering approach for mining multidimensional data is proposed. It is also called as,"Bi-clustering" which utilizes the row and column content. The sequential updates for alternate minimization coclustering (AMCC) algorithm which is variant of EM algorithms is used in fast nonnegative matrix tri-factorization FNMTF algorithm for cluster generation process. To improve system efficiency, the feature selection process is applied to find redundancy in row and column structure. The algorithm is tested on distributed environment. This approach maintains the convergence properties of AMCC algorithms. FNMTF algorithm supports efficient implementations of AMCC algorithms with sequential updates on distributed environment.

References:

[1] R. M. Neal and G. E. Hinton, "A view of the EM algorithm that justifies incremental, sparse, and other variants," Learning in Graphical Models, pp. 355–368, 1999

[2] I. Dhillon, "Co-clustering documents and words using bipartite spectral graph partitioning," in Proc. Knowl. Discovery Data Mining, 2001, pp. 269–274

[3] A. Banerjee, I. Dhillon, J. Ghosh, S. Merugu, and D. Modha, "A generalized maximum entropy approach to Bregman Co-clustering and matrix approximation," in Proc. 10th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, 2004, pp. 509–514.

[4] B.Kwon and H. Cho, "Scalable co-clustering algorithms," in Proc. ICA3PP, 2010, pp. 32–43.

[5] M. Deodhar, C. Jones, and J. Ghosh, "Parallel simultaneous coclustering and learning with map-reduce," in Proc. IEEE Int. Conf. Granular Comput., 2010, pp. 149–154.

[6] H. Wang, F. Nie, H. Huang, and F. Makedon, "Fast nonnegative matrix tri-factorization for large-scale data co-clustering," in Proc. 22nd Int. Joint Conf. Artif. Intell., 2011, pp. 1553–1558.

[7] Y. Zhang, Q. Gao, L. Gao, and C. Wang, "iMapreduce: A distributed computing framework for iterative computation," in Proc. IEEE Int. Symp. Parallel Distrib. Process. Workshops PhD Forum, 2011, pp. 1112–1121.

[8] A. Narang, A. Srivastava, and N. P. K. Katta, "High performance distributed co-clustering and collaborative filtering," in IBM Res., NY, United States, Tech. Rep. RI11019, 2011, pp. 1–28.

[9] Y. Cheng, G. Church, "Biclustering of Expression Data", Received 22 January 2015, Revised 22 June 2015

[10] X. Cheng, L. Gao, "Co-ClusterD: A Distributed Framework for Data Co-Clustering with Sequential Updates", IEEE transactions on knowledge and data engineering, vol.27, No.12, Dec,2015

[11] Kaveri More, Prof. R. P. Dahake "Clustering of Documents with Sequential Update in Distributed Environment", 6th post graduate conference of computer engineering cpgcon2017

[12] Kaveri More, Prof. R.P. Dahake, "Review on Clustering of Documents with Sequential Update in Distributed Environment", Journal of Emerging Technologies and Innovative Research, ISSN-2349-5162, Volume 5, Issue 6, June 2018