Review on Clustering of Documents with Sequential Update in Distributed Environment

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

[1] Student, Department of Computer Engineering, MET's Institute of Engineering, Nasik.
[2]Professor ,Department of Computer Engineering, MET's Institute of Engineering, Nasik.

Abstract: *Clustering is very effective tool developed for two dimensional 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. 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, therefore updates in the sequential manner in a distributed environment can be propose by using AMCC algorithms and by dividing clusters and batch updates approaches. AMCC algorithms can prove efficiency of Co-Clustering. The proposed work can also contributes attribute based filtering of input dataset.*

Keywords: **Co-Clustering, concurrent updates, sequential updates, cloud computing, distributed framework**

1. INTRODUCTION

In the domain of data mining, Co-clustering is the capable tool for mining co-occurrence of two dimensional data. Clustering has lots of applications such as, in recommendation system, text mining, and gene expression. 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; cluster information is get 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. Updates in sequential way can decreases the computational cost of processing without affecting to the result of solution. Sequential updates can be referred as alternate minimization co-clustering (AMCC) algorithms. It is different than EM algorithms. The AMCC algorithm is assemble of sequential updates. Convergence property of co-clustering algorithms cannot provide guarantee due to inconsistency in clustering information. It will also brings synchronization overheads during information synchronization whenever cluster assignment get changed. These are reasons behind AMCC algorithm which cannot works in a distributed way with sequential update. Parallelization of AMCC algorithm with updates in sequential way can be done by using two approaches, namely, 1. Dividing cluster and 2. Batch updates.

1. 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 and column clusters are divided into many non-overlapping subsets at the starting of iteration. Each worker then performed sequential updates with row or column cluster.

2. Batch Updates:

   In batching approach, requirement of sequential updates relaxing exact match, it performs sequential updates using batch updates. First, it performs bunch of row or column assignments of cluster and then updates information of clusters.

From performance analysis of both approaches it can prove that both approaches can preserve the convergence properties of AMCC algorithms. Therefore, proposed Clustering can develop AMCC algorithm in an efficient way with updates in a sequential manner. It can also provide abstraction for AMCC algorithms with sequential manner. It allows programmers to specify the sequential update operations via simple APIs.

2. REVIEW OF LITERATURE

R.NEAL et al, 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. EM algorithms can validate incremental variants in clustering process [1].

I. Dhillon, et al, 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. It is only suitable for noise removal to improve clustering performance [2].

A. Banerjee et al., 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. In this co-clustering and matrix approximation can preserve maximum amount of "information" in actual data [3].

B. Kwon et al, 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[4].

M. Deodhar et al, 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[5].

H. Wang, et al., 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. 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. Limitations could be incorporated by manifold information and proposed Locality Preserved FNMTF (LP-FNMTF) method [6].

Y. Zhang, et al., 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[7].

A. Narang ,et al, 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[8].

Y. Cheng, et al, 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. The proposed techniques affects on the performance of system as, complex computations required for square residue [9].

X. Cheng et al., proposed Co-clustering 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 [10].

3. OVERVIEW OF CLUSTERING APPROACH

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.

Different type of clustering techniqes are explained as below:

K-means and k-menoids are similarity based clustering algorithm and they required to specify k in advanced.

Hierarchical algorithms are useful in the applications where more n more searching is req. and it naturally generates the hierarchical tree which required k and threshold value. It cannot applicable in large scale data.

Density based algorithms mainly work to defined high density area than the remainder dataset. It does not required k mention in advanced; it can detect outlier from cluster. But it cannot discover the high density regions from low density regions. And also have limitation on text data.

AMCC algorithm is EM i.e. expectation maximization algorithm. It support for sequential updates in clustering process. A previous technique of clustering does not suitable for sequential updates. Alternate minimization co-clustering (AMCC) algorithm can be efficient in co-clustering process[10].

3.1 System Architecture:

Distributed framework is designed for efficient implementation of alternate minimization of co-clustering (AMCC) with sequential update. It is achieved by incorporation of two approaches

A] Dividing cluster approach

B] Batching Point approach

Fast non-negative matrix tri-factorization (FNMTC) is AMCC algorithm used to get proper convergence result. As a part of contribution before working on dataset, attribute based filtering process is done on dataset. It helps to fasten the cluster creation process. Execution of row and column clustering is carried out in distributed manner at different worker's end in concurrent manner. Hence statistics of co-cluster (Scc) should be changed whenever row (or column) clustering updates its cluster assignment. Thus, due to this concurrent process the convergence properties of co-clustering algorithms cannot be maintained and hence dividing cluster approach is adopted. This approach is better when numbers of workers are less than number of clusters. But this condition restricts the scalability of this approach. Hence batch update for AMCC algorithm is introduced. The difference between batch and sequential update is that batch updates perform the cluster information update after a batch of rows (or columns) have updated their cluster assignments, rather than after each change in cluster assignments.

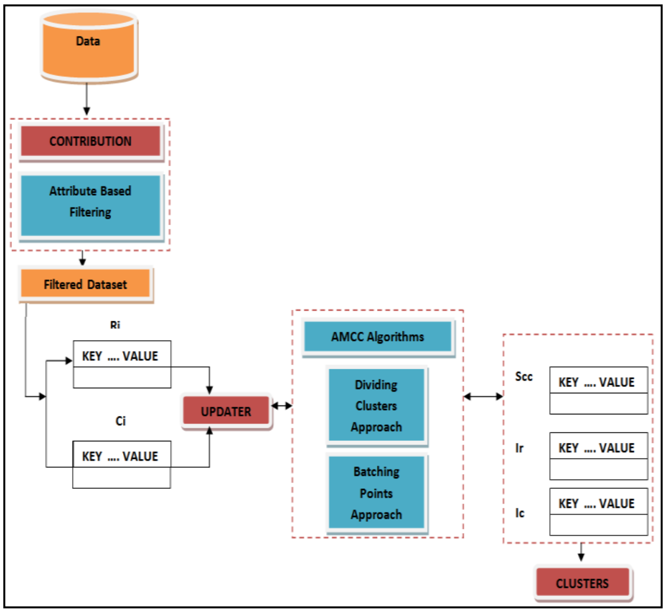

Figure 1: Architecture of Co-clustering

As shown in figure, it maintain five key-value pair tables in its memory to store the following variables: a non-overlapping subset of rows of the input data matrix R_i, a non-overlapping subset of columns of the input data matrix $C_{i,}$ row cluster indicators of the input data matrix I_r, column cluster indicators of the input data matrix I_c, and the statistics of co-clusters S_{cc}.

3.2 Processing Steps:

Step 1: Select the dataset

Step 2: Attribute based filtering is done

Step 3: Finalize the dataset

Step 4: Select Approach

If (A] Dividing cluster Approach) is selected

Step 5: Group input data matrix in k rows and l columns

Step 6: Decide number of workers p (p <= mink/2 ,l/2)

Step 7: Distribute subsets of rows Ri and Subsets of columns Ci to worker Wi

Step 8: Worker performs row clustering

Step 9: Sequential update

Step 10: Worker performs column clustering

Step 11: Sequential update

Step 12: Subset of Scc and cluster indicators (Ir and Ic) is combined and synchronized using updater

If([B] Batching Point Approach) is selected

Step 5: Distribute subsets of rows Ri and Subsets of columns Ci to worker Wi

Step 6: Batch for row is decided

Step 7: Batch for column is decided

Step 8: Row clustering is performed for each row batch

Step 9: Synchronization and sequential update is done

Step 10: Column clustering is performed for each column batch

Step 11: Synchronization and sequential update is done

3.3 Algorithm

Alternate Minimization (AM) Based Approach

Input:

- Data Matrix A (size: m x n) and k, l (# of row and column clusters)

Processing Steps:

Step 1: Initialize row and column cluster mappings, X (size: m x k) and Y (size: n x l)

-Random assignment of rows (or columns) to row (or column) clusters

-Any traditional one dimensional clustering can be used to initialize X and Y

Step 2: Objective function: $||A - \hat{A}||^2$, $\hat{A}$ is matrix approximation of A computed as follows:

-Each element of a co-cluster (obtained using current X and Y) is replaced by mean of co-cluster ($a_{I,J}$)

-Each element of a co-cluster is replaced by ($a_{i,J} + a_{I,j} - a_{I,J}$) i.e row mean + column mean – overall mean

Step 3: While (converged)

- Phase 1:

-Compute row cluster prototypes (based on current X and matrix A)

-Compute Bregman distance, $d_\Phi(r_i, R_r)$ - each row to each row cluster prototype

-Compute probability with which each of m rows fall into each of k row clusters

-Update row cluster X keeping column cluster Y same (some thresholding is required here to allow limited overlap)

- Phase 2:

-Compute column cluster prototypes (based on current Y and matrix A)

-Compute Bregman distance, $d_\Phi(c_j, C_c)$ - each column to each column cluster prototype

-Compute probability with which each of n columns fall into each of l column clusters

-Update column cluster Y keeping row cluster X same

Step 4: Compute objective function: $||A - \hat{A}||^2$

Step 5: Check convergence

Output: 'k' clusters

4. CONCLUSION

Co-clusteing approach for mining two dimensional data is discussed. It is also called as, "Bi-clustering" which utilizes the row and column content. There are two approaches such as, cluster dividing and batching approach are available for Co-clustering. Alternate minimization co-clustering (AMCC) algorithms which are variants of EM algorithms can maintain convergence properties with sequential updates. Two parallelize approaches can maintain the convergence properties of AMCC algorithms. Based on these two approaches, a new distributed framework called as, Co-Clustering can be introduced. It can support for efficient implementations of AMCC algorithms with sequential updates.

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] Columbia University Image Library. [Online]. Available:http://www.cs.columbia.edu/CAVE/software/softlib/coil-20.php,1996 UCI Machine Learning Repository. [Online]. Available: http:// archi