A Modified Fastmap K-Means Clustering Algorithm for Large Scale Gene Expression Datasets

Shital A. Raut and S. R. Sathe

Abstract— **Clustering is an important tool of data mining to extract the context or meaningful patterns from the datasets. To extract the patterns from gene expression datasets, cluster analysis is used. From last two to three years, there are noticeably increase in public datasets, like geo, arrayexpress, genebank, are noted. All these increase in public datasets are not only in numbers but also in dimensions. To analyze single experiment, 1000's of genes and 100's of samples are available. To handle these large dimensions, we have to moderate or modify the traditional clustering algorithms. K-means clustering algorithm is one of the most used and tested clustering algorithm not only for gene expression datasets but also for various different datasets. But, as the dimension goes on increasing, CPU time requirement and memory requirement also increasing. Here, we try to increase the speed of K-means algorithm by adding additional phase (by using moderate FastMap) before implementation of k-means algorithm on the datasets. So, Modified FastMap K-means Clustering Algorithm, is a two phase algorithm, which try to reduced CPU time and memory requirements as compared to tradition K-means requirements. We have shown tabular results for three datasets, which are downloaded from public repository, NCBI, geo. The algorithm can successfully generate good results for large as well as small datasets.**

Index Terms—**Gene expression analysis, MFKCA, Cluster analysis.**

I. Introduction

Gene is a basic unit of all the organisms. When the genes are active, they express coded information with them, this expression of genes is called gene expression. Gene expression is a required process to form complex structures, called as proteins, to directs functionality of cells.

Gene expression analysis, successfully found solutions for many core genetic problems. The significant contribution of this analysis is to discovers important biological information. It contributed, to get the functional roles of different genes and its participation in various cells, to analyze expression patterns of genes in various cells at various diseases and during treatment, to identify co-expressed genes which is required for critical biological observations.

Due to advancement of technologies, huge datasets (microarray generated) of gene expressions are available. There are number of methods exists, to analyze this data as noted in literature. Cluster analysis is wildly used for analysis of gene expression datasets. The goal of clustering is to group the objects which are similar in nature as per selected distance metrics. Clustering on gene expression databases, helps, to identify new classes of biological samples, to extract gene expression patterns from the sampled databases, to identify group of co-regulated genes, to find set of genes to get significant biological information.

If we look back for last two decades research papers, we will find number of clustering algorithms. Out of these, traditional clustering algorithms are general and need to moderate for particular application. Many improved algorithms are also available in the literature. For small datasets they can be used for generating the results, but for large dataset they cannot be efficiently applied. From last four to five years, due to evolution of new techniques, availability of new research tools in laboratories, and ample funding for research projects a huge new experimental databases are generated which are constantly added to the existing databases. This increase in databases are not only in number of databases but also in the size of databases. To handle these large databases, improved or more efficient algorithm need to be develop. As the experimental dimensions goes on increasing, computation calculations are getting more complexes and it directly effects the efficiency of an algorithms. The major affected parameters are larger CPU time and large memory requirement.

Here in this paper, we tried to improve the efficiency of one of the traditional algorithm, k-means clustering algorithm. The complete algorithm is divided into two phases. First phase is an implementation of moderate FastMap algorithm to reduce dimension and second phase is an implementation of conventional K-means algorithm. Formerly, we call Modified FastMap K-means Clustering Algorithm as MFKCA. We further, discuss the results of conventional K-means clustering and the results generated by MFKCA.

Outline of the paper

Sectional distribution of this paper is as follows, second section, is about microarrays generated datasets for gene expression analysis as background, third section is for general and recent advancement in clustering algorithms, The details of modified fastmap k-means algorithm is discussed in forth section. Experimental discussion and observation are followed in fifth Section, finally discussion cum summarization is done in sixth section.

II. Background

All the organisms are made up of DNA and its bio-product. As per central dogma of life, two steps are required to convert DNA to complex protein structure [1],[2]. This final complex protein structure is origin for functional activities of organisms cells. The two steps are famously known as 'transcription' and 'transformation'. When gene is active,

Manuscript received October 13, 2011; revised November 22, 2011.

Authors are with Department of CSE, VNIT, Nagpur (phone: 0712-2801259; e-mail: rautsa@gmail.com).

coding sequences are copied from DNA to RNA, called ‘transcription’. These RNA after further processing produces proteins, called ‘transformation’. Gene expression analysis helps to understand this complex procedure. The information can be further used for detection of diseases, cell functional regularity and drug design.

All over the world, microarrays are successfully used to measure gene expression. With the help of microarrays, we can observe thousands of genes simultaneously. It works as per principle of Watson-Crick base pairing rule [3],[4]. Microarrays exploit the preferential binding of complementary single-stranded nucleic acid sequences. In microarrays itself, there are number of technologies used to measure gene expression but most reliable is hybridization. There are two types of experiments. The first and common experiment is, we want to compare mRNA levels of one or more genes in cells from different sources, for example, it may be tumor vs. normal cells, or cells from genetically modified vs. normal cells, or a cell before and after drug treatment. Second type of experiment is time-course experiments, where cells are sampled at different times, e.g. after the administration of a drug, or as the cell cycle or development proceeds, and interest is in temporal patterns of gene expression. Here, we can take datasets of both experiment. For generating final datasets of gene expression, microarrays follows three steps:

a) Sample preparation and labeling: Two samples are required for processing. One is sample from tissue of interest and second is DNA probe on microarrays. It involves, extraction of mRNA from tissue of interest, conversion from mRNA to cDNA and labeling of this cDNA. Labeling is important as it founds detection of which cDNA are bound to microarray.

b) Sample hybridization and washing: In hybridization, DNA probes on the microarrays and labeled DNA target, forms heteroduplexes according to Watson-Crick base pairing rule. After hybridization, microarray chip is washed to eliminate any excess labeled sample other than DNA complementary probes.

c) Image scanning and its processing : A hybridized array is scanned to produce a microarray image. Labeled samples with dyes emit detectable light when simulated by a laser. Detectable emitted light by target DNA strands are bound to their complementary probes. This scanned output is a monochrome image.

This scanned image can be considered as an input for microarray information analysis. This analysis categorized as low level analysis and high level analysis. In low level analysis, spot quantization matrices are generated. The low level analysis includes identification of spots corresponding to genes, segmentation of spot for accuracy measure and convert valid spot information into image. The high level analysis includes all the methods which generate knowledge from gene expression matrix. Methods like Box Plot and Gene Pies are graphical analysis methods, Scatter Plot, Principal Component Analysis (PCA) Independent Component Analysis (ICA), Clustering Analysis are analysis methods considered in high level analysis. Estimated results from high level analysis is needed in almost all the biological connected fields.

The set of spot quantization matrices are further assembled as single gene expression data matrix (dataset). In gene expression data matrix, rows represents number of genes and column represents multiple experimental conditions (samples). Each cell position indicates absolute or relative abundance of particular gene at specific experimental condition. Mathematical representation is as follows:

$$E = E_{xy} \mid 1<=x<=g, 1<=y<=s \qquad (1)$$

In equation (1) ‘g’ represents number of genes in a matrix, ‘s’ represents number of samples, E_{xy} represents gene’x’ expression for ‘y’ sample.

III. Cluster Analysis

Cluster analysis is one of the high level analysis method. In data mining, to explore the knowledge from the dataset cluster analysis is used. This is one of the important tool to discover important and unknown patterns from the datasets.

We are using cluster analysis as analysis as a tool to identify gene expression patterns by clustering number of genes as per their expression values. This identification of clusters of genes can direct to find co-regulated genes, to discover abnormal patterns exists in different cells, to test drug effects on cells, to find new classic biological samples. There are number of algorithms for cluster analysis. The two basic aspect of clustering algorithms. First is, clustering algorithm forms cluster objects with similar properties. The inter cluster distance for different clusters must be maximum and intra cluster distance within cluster for different object must be minimum. Second is, distance metric must be selected as a qualification to group the objects based on similarity and dissimilarity. Next we further discuss major algorithms available for cluster analysis in short.

Categorization of major clustering methods include Partitioning methods, Hierarchical methods, Density-based methods, Grid-based methods, and Model-based methods.[5]

a) Partitioning methods: In partitioning algorithms, it organizes the ’n’ objects into ‘k’ partitions where each partition represents a cluster. Optimize clusters are formed based on dissimilarity function based on distance. Here user need to specify number of clusters. Two major algorithms in this category are K-means, where each cluster is represented by the mean value of the objects in the cluster and K-medoids, where each cluster is represented by one of the objects located near the center of the cluster.

b) Hierarchical methods: A hierarchical method creates a hierarchical decomposition of the given set of data objects either from top to bottom or from bottom to top. Thus, two approaches are called as divisive and agglomerative approach respectively. BIRCH is an advance hierarchical algorithm used for large databases with minimum resource management.

c) Density-based methods: These methods are developed based upon density. It consider clusters as dense regions of objects in the data space that are separated by regions of low density. The principle used to find clusters of any arbitrary shapes. Example of this method is DBSCAN algorithm.

d) Grid-based methods: Grid –based methods works on grid structure. They transform object space into finite grid structure so that all operation can be formed on same grid structure. STING and WaveCluster are examples of this methods.

e) Model-based methods: Model-based methods hypothesize a model for each of the clusters and find the best fit of the data to the given model it uses standard statistics to determine number of clustering automatically. COBWEB is a typical example of this category.

IV. Modified Fastmap k-Means Clustering Algorithm (MFKCA)

Nowadays, due to generation of large results from experimental research projects dimensions of datasets are very large. 100's of samples are available for single experiment. 10000's of genes are present per experiment, and hence huge databases are available. Due to complex computations and large memory requirement, traditional clustering algorithms cannot be used as it is on these huge datasets. Out of all traditional clustering algorithms, K-means is partitioned based algorithm, which consistently reported to be better than other algorithms. As the number of genes and number of samples on the chip increases, the size of dataset creates major problem for clustering algorithms. Most of the algorithm generate good results for small datasets, but failed in giving optimum results for large datasets. If we observe the working of most of the clustering algorithms, most of the CPU time is required for multiple scans of datasets and for generating repetitive computations. In MFKCA, we try to reduce these two important factors, so that we can get improved CPU time and optimum memory requirements. The phase one of algorithm will work out to reduced the 'n' samples value to '*k*' value using FastMap algorithm, input for phase one is imputed databases of size '*g*' × '*s*', where '*g*' represents number of genes for an experiment while 's' represents number of samples for the same experiment. FastMap converts original dimension '*g*' × '*s*' to '*g*' × '*k*', '*k*' represents transformed dimension from '*s*' to '*k*'. In second phase of algorithm, conventional K-means algorithm is applied on '*g*' × '*k*' to generate clusters for asked experiment. The detailed steps for an algorithm are as follows:

a) *Preprocessing*: Microarray generated datasets often suffer from multiple missing expression values. Causes of missing values are numerous. Major causes includes, problem while robotic methods employed in generating microarrays data, insufficient resolution, image corruption and slide contamination by dust. For most of the algorithms to work accurately, complete dataset is must. So in our algorithm we use simple approach to imputing missing values, is to replace a missing entry with the average expression over the rows[6].

b) *Phase-I(Moderate FastMap)*: For the desktop computers, it is difficult to cluster more than 15,000 genes and 100s samples per an experiment. Hence the idea is to map 'n' objects (number of genes with samples) into k-d space, using k-feature extraction function. The objects are the points in some unknown n-dimensional space and we try to project those into k mutually orthogonal directions [7],[8]. The proposed method is to project all the objects on a selected 'line'. Previously, all the objects will be a point in n-d space. The number of 'lines' are equal to value of 'k'. For each line, we choose two objects '*Oa*' and '*Ob*' called as 'pivot objects', the 'line' is that, passes through these pivot objects. For selection of '*Oa*' and '*Ob*', first we choose random object 'Ob' from the dataset, now '*Oa*' is the farthest object from 'Ob'. All the rest of the objects are need to be projected on the 'line' which passes from '*Oa*' and '*Ob*'. The projections of the objects on that 'line' are computed using cosine law, please refer Fig. 1. '*Oi*' are all the rest of the objects need to be projected.

In any triangle OaOiOb, the cosine law gives:

$$dbi^2 = dai^2 + dab^2 - 2xidab \qquad (2)$$

and hence, projection of '*Oi*' on 'line' '*Oa*' to '*Ob*' can be defined as:

$$X_i = \frac{dai^2 + dab^2 - dbi^2}{2dab} \qquad (3)$$

In the above equation, *dij* is the distance D(*Oi*,*Oj*) for *i*, *j* =1,….., *N*.

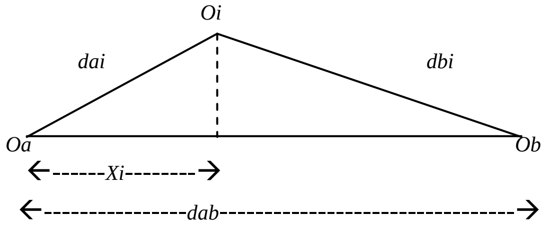

Fig. 1

The algorithm accepts input as '*N*' objects, the desired number of dimensions '*k*' and a distance function '**D**()'. It maps the objects into points in k-d space. In short, we map all objects in 2-*d* space when '*k*'=1, iteratively when *k*=2, we select new two pivot objects and all the objects are projected to the 'line' which passes from pivot objects. At end, when 'k^th^' iteration we map all the objects into k-d space.

c) *Phase-II:* In Phase–II, we are using conventional k-means algorithm. First, we are analyzing mapped k-d space, i.e. all '*k*' lines. We select that 'k^{th}' iteration which is having least difference between actual distance and projected distance of all the objects. After analysis, we have only one 2-*d* 'line' where all points are mapped from n-d to this 'k^{th}' 2-*d* space. This 'k^{th}' distance array is an input for k-means algorithm. Optimize clusters are formed which are calculated from distance array based on dissimilarity function. User need to specify number of the clusters. K-means is an iterative algorithm, till it reaches to its optimized clusters. Lastly, each cluster is represented by the mean value of the objects in the cluster.

V. Experimental Discussion and Observations

A. Implementation

From generated results, we can say that, the two-phase algorithm is more suitable to use for large datasets than conventional K-means algorithm. The program is implemented on windows based machines with general setup. All the software programs were written in MATLAB 7.0.

B. Experimental Datasets

All the experimental datasets are downloaded from NCBI Gene Expression Omnibus (GEO). Here, we are discussing three datasets in detail. First dataset, GSE9006, is Gene expression in PBMCs for children with diabetes [9]. The array platforms are two, Affymetrix Human Genome HG-U133A and Affymetrix Human Genome HG-U133B. Combined data from both the chips were used for result generations. Second dataset, GSE30, is Multiplex three dimensional brain gene expression mapping in a mouse model of Parkinson's disease [10]. The array platform is Mouse 9K UCLA. Third dataset, GSE28, is Diauxic shift, Exploring the metabolic and genetic control of gene expression on a genomic scale in yeast [11]. The detailed description of each of the datasets are given in the Table I, II, III.

C. Performance of the Algorithm and Observations

K-means is one of the most famous and widely used clustering algorithm for numerous applications. Using Modified FastMap K-means Clustering Algorithm (MFKCA), we try to speed up the k-means for large datasets. It reduces the n-d space of computation to 2-d space. We took different values for *k*=4,8,16,32, and 64 to analyze two datasets, where value of the 'k' need to taken from user side. It is shown that comparative speed up can be achieved with these two datasets using MFKCA. For third dataset, which is smaller than first two, the speed of MFKCA, with *k*=4, is better than traditional k-means algorithm. The algorithm produced the total time, in seconds, taken by the CPU for computing number of clusters C, for C=8, C=16,C=32,C=64 as output.

The GSE9006 observations is considered to be large set. The tabulated results are shown in Table I.

GSE30 is not large than GSE9006 but as seen to number of genes and number of samples, it can be said to be medium dataset. The tabulated results are shown in Table II.

GSE28 is a small dataset. But still if we look at the values generated by MFKCA and traditional k-means, speed up is more for MFKCA than K-means. Tabulated results are shown with Table III.

TABLE I: PERFORMANCE COMPARISON OF MFKCA AND K-MEANS CLUSTERING (KC) ON GSE9006

Number of Genes	No of samples		MFKCA	KC	MFKCA	KC	MFKCA	KC	MFKCA	KC
44928	117		C=8	C=8	C=16	C=16	C=32	C=32	C=64	C=64
		K=4	7.08	126.5	9.35	240.30	15.09	457.70	25.79	909.67
		K=8	9.15	126.5	11.90	240.30	17.43	457.70	31.72	909.67
		K=16	14.68	126.5	18.84	240.30	26.17	457.70	38.64	909.67
		K=32	25.57	126.5	28.03	240.30	36	457.70	49.17	909.67
		K=64	47.40	126.5	50.40	240.30	57.28	457.70	69.78	909.67

TABLE II: PERFORMANCE COMPARISON OF MFKCA AND K-MEANS CLUSTERING (KC) ON GSE30

Number of Genes	No. of Samples		MFKCA	KC	MFKCA	KC	MFKCA	KC	MFKCA	KC
9504	80		C=8	C=8	C=16	C=16	C=32	C=32	C=64	C=64
		K=4	1.4	17.9	2.1	29	3.2	163.1	5.4	328.1
		K=8	1.9	17.9	2.7	29	4.2	163.1	6.7	328.1
		K=16	3.1	17.9	3.8	29	5.1	163.1	6.8	328.1
		K=32	5.4	17.9	6.5	29	7.2	163.1	10.1	328.1
		K=64	9.7	17.9	11.1	29	12.4	163.1	14.8	328.1

TABLE III: PERFORMANCE COMPARISON OF MFKCA AND K-MEANS CLUSTERING (KC) ON GSE 28

Number of Genes	No. of Samples		MFKCA	KC	MFKCA	KC	MFKCA	KC	MFKCA	KC
6400	7		C=8	C=8	C=16	C=16	C=32	C=32	C=64	C=64
		K=4	0.43	0.25	1.21	2.38	2.3	6.59	3.22	20

VI. DISCUSSIONS

Nowadays, **s**cope of clustering is widespread. Many applications used clustering as one of the major tool for generating meaningful context from large datasets. Due to large requirement of memory and CPU time, available clustering algorithms are not suitable to use as it is for large datasets. Hence, some moderations are required in existing algorithm. Here, with MFKCA, we try to add certain speedup to existing K-means algorithm.

With MFKCA, we generate 'k' 2-d lines. On each selected line we mapped all the objects in the datasets. Out of 'k' lines we choose only one 2-d line, according to value of aggregated distance value. For choosing this 'line', we calculate summarization for differences of all objects actual distance and the projected or mapped distance. The 'line' which is having less difference between these two values is the most closest. We generate the distance array, which calculates the distance of all the objects with concern to this line. Traditional K-means algorithm is applied on this distance matrix as per given cluster value. The clusters are validated on its principle, that distance between inter-cluster must be maximum and distance of intra-cluster objects must be minimum. Cluster centroids are holding the mean value of respective clusters. We are taking the value of all these centroids for mutual comparison. In this, we also calculate the distance between two farthest objects of two different clusters for comparison.

To test the algorithm, we select three different sizes of datasets. For all these datasets, performance of MFKCA is good. The value of '*k*' can be taken from user-side. The value of '*k*' should not be less than 2. The selection of '*k*' value is important as with its increment, it increases CPU time, space and accuracy.

Here, we present the Moderate FastMap K-means Algorithm with two phases, to fasten up the process of clustering for large datasets. We demonstrate it on almost all the datasets which can be categorized as small, medium and large for gene expression analysis. The results are promisingly good and encouraging to use for other applications where clustering is required to find significant context.

ACKNOWLEDGEMENTS

This paper is a part and parcel of research work contributed for my (Shital Raut) PhD work on gene expression analysis. I express my sincere thanks to my guide Dr. S. R. Sathe, for his unconditional support, motivations and valuable advices. I sincerely thanks to all those are directly or indirectly give me encouragement to do my work.

REFERENCES

[1] S. A. Raut, S. R. Sathe, and A. Raut, "Bioinformatics: Trends in Gene Expression Analysis," *proceedings of 2010 International Conference On Bioinformatics and Biomedical Technology*, 16-18 April 2010, Chengdu, China.

[2] S. A. Raut, S. R. Sathe, and A. P. Raut, "Gene Expression Analysis-A Review for large datasets," *Journal of Computer Science and Engineering*, vol.4, Issue 1, November 2010.

[3] T. Speed, "Statistics And Gene Expression Analysis," 2004. Available: www.proba.jussieu.fr/bulletin/ArticleSpeed.pdf".

[4] T. Speed, "Statistical Analysis of Gene Expression Microarray Data. Chapman & Hall/CRC" ch1,2003.

[5] J. Han, M. Kamber, and J. Pei, "Data mining, Concepts and Techniques," 3rd Edittion, (The Morgan Kaufmann Series in Data Management Systems), ISBN-10: 1558604898/ISBN-13: 978-1558604896 ch 10.

[6] A. Zhang, "Advanced analysis of gene expression microarry data," Ch 4, World scientific publishing Co,2006.

[7] C. F. K. I. Lin, "FastMap: a fast algorithm for indexing, data-mining and visualization of traditional and multimedia datasets," *Proceeding SIGMOD '95 Proceedings of the 1995 ACM SIGMOD international conference on Management of data*, ISBN:0-89791-731-6.

[8] J. Tesic, "*Evaluating a Class of Dimensionality* Reduction," Available: vision.ece.ucsb.edu/~jelena/research/ 290I report.pdf.

[9] E. C. Kaizer ,C. L. Glaser, D. Chaussabel, and J. Banchereau , *Gene* expression in peripheral blood mononuclear cells from children with diabetes. J Clin Endocrinol Metab 2007 Sep;92(9):3705-11.PMID: 17595242, 2007

[10] V. M. Brown , A. Ossadtchi, A. H. Khan, Yee S *et al.*, "Multiplex three-dimensional brain gene expression mapping in a mouse model of Parkinson's disease," Genome Res 2002 Jun;12(6):868-84. PMID: 12045141, 2002

[11] J. L. DeRisi, V. R. Iyer, and P. O. Brown, " Exploring the metabolic and genetic control of gene expression on a genomic scale," *Science* 1997 Oct 24;278(5338):680-6. PMID: 9381177, 1997

Shital A. Raut B. E.in Comp.Tech, M.E. in Computer Sc.&Engg., and currently attach with Department of Computer Sc. & Engg. as Asst. Pofessor,VNIT, Nagpur,India. Registered as Ph.D scholer at VNIT. Current research interest includes bioinformatics, data mining, and algorithms. Published about more than 10 research papers in various reputed national and International conferences and Journal (01). She is having student membership for IEEE.

S. R. Sathe completed his M.Tech. form IIT, Bombay, and Ph.D. from R.T.M. Nagpur University. At present, he is Professor and head of theDeptt. Of Computer science and Engineering, VNIT,Nagpur. He had handled many national research projects on Image processing, Security and Parallel processing. Current research includes Bioinformatics, Image processing , Parallel processing, Algorithms etc. He had published more than 30 papers in various reputed International conferences and Journals.