An Efficient Clustering Technique for Message Passing Between Data Points using Affinity Propagation

D. NAPOLEON
Assistant Professor
Department of Computer Science
Bharathiar University
Coimbatore, Tamil Nadu, INDIA

G.BASKAR
Research Scholar
Department of Computer Science
Bharathiar University
Coimbatore, Tamil Nadu, INDIA

S.PAVALAKODI
Research Scholar
Department of Computer Science
Bharathiar University
Coimbatore, Tamil Nadu, INDIA

Abstract **— A wide range of clustering algorithms is available in literature and still an open area for researcher's *k-means* algorithm is one of the basic and most simple partitioning clustering technique is given by Macqueen in 1967. A new clustering algorithm used in this paper is affinity propagation. The number of cluster k has been supplied by the user and the Affinity propagation found clusters with much lower error than other methods, and it did so in less than one-hundredth the amount of time between data point. In this paper we make analysis on cluster algorithm k-means, efficient k-means, and affinity propagation with colon dataset. And the result of affinity propagation shows much lower error when compare with other algorithm and the average accuracy is good.**

Keywords **- Data Mining, Clustering, k-means, Affinity propagation.**

I. INTRODUCTION

Clustering the data sets is a challenging problem. Different algorithms for clustering of genes have been proposed. The cluster analysis deals with the problems of organization of a collection of patterns into clusters the model is determining the number of clusters needed prior to learning. This is usually inputted by the user through a series of trial and error values. Also the usage of random initialization does not provide deterministic results. Different algorithms for clustering of genes have been proposed (Medvedovic & Sivaganesan 2002, Yeung et al. 2001, Yeung et al. 2003). However due to the large number of genes only a few algorithms can be applied for the clustering of samples ((Bagirov et al. 2003)). As the number of clusters increases the number of variables in the clustering problem increases drastically and most of clustering algorithms become inefficient for solving such problems. *K-means* algorithm and its different variations are among those algorithms which still applicable to clustering. *But k-means* algorithms in general can converge only to local minima and these local minima may be significantly different.

Affinity propagation takes a different approach to clustering. Rather than make hard decisions on the cluster centers at each iteration, soft information about cluster exemplars is propagated through the dataset by way of a message passing algorithm. Affinity Propagation performs the max-sum algorithm on a factor graph model of the data to solve for a good configuration of cluster members. The number of clusters need not be pre specified; instead, the message passing algorithm discovers the number of clusters automatically. Usually referred to simply as *k-means*, Lloyd's algorithm begins with *k* arbitrary centers, typically chosen uniformly at random from the data points. Each point is then assigned to the nearest center, and each center is recomputed as the center of mass of all points assigned to it. These two steps (assignment and center calculation) are repeated until the process stabilizes

To classify the various types of cancer into its different subcategories, different data mining techniques have been used over gene expression data. In this paper, we make a comparative analysis of *k-means* with affinity propagation, over colon dataset Comparison is made in respect of accuracy and convergence rate.[13]

II. K-MEANS ALGORITHM

The *k-means* algorithm (MacQueen, 1967) is one of a group of algorithms called *partitioning methods*. The k -means algorithm is very simple and can be easily implemented in solving many practical problems. The k-means algorithm is the best-known squared error-based clustering algorithm. Consider the

data set with 'n' objects,

$$S = \{xi : 1 ¡Ü i ¡Ü n\}.$$

Steps :

1) Initialize a k-partition randomly or based on some prior knowledge.

i.e. {C1, C2, C3,……., Ck }.

2) Calculate the cluster prototype matrix M (distance matrix of distances between k-clusters and data objects)

M = { m1 , m2 , m3, …….. , mk } where mi is a column matrix 1× n .

3) Assign each object in the data set to the nearest cluster - Cm

i.e. x j ,Cm if || x j - Cm || ¡Ü || x j – Ci || Í 1 ¡Ü j ¡Ü k , j , m where j=1,2,3,…….n.

4) Calculate the average of each cluster and change the k-cluster centers by their averages.

5) Again calculate the cluster prototype matrix M.

6) Repeat steps 3, 4 and 5 until there is no change for each cluster.

The process, which is called "*K-Means*", appears to give partitions which are reasonably efficient in the sense of within-class variance, corroborated to some extend by mathematical analysis and practical experience [6,8]. Also, the *K-Means* procedure is easily programmed and is computationally economical, so that it is feasible to process very large samples on a digital computer [14]. *K-Means* algorithm is one of first which a data analyst will use to investigate a new data set because it is algorithmically simple, relatively robust and gives "good enough" answers over a wide variety of data sets [9,11].

III. EFFICIENT K-MEANS ALGORITHM

Efficient *K-means* Algorithm (Zhang et al., 2003) is an improved version of k-means which can avoid getting into locally optimal solution in some degree, and reduce the probability of dividing one big cluster into two or more ones owing to the adoption of squared-error criterion.

Algorithm: Improved *K-means Algorithm*

(S, k), *S={x1,x2,...,xn }* [8]

Input: The number of clusters k1(k1> k) and dataset containing n objects(Xi)

Output: A set of clusters (Cj) that minimize the squared-error criterion.

Steps:

1. Draw multiple sub-samples {SI, S2, . . . ,Sj } from the orginal dataset;

2. Repeat step 3 for m=l to j

3. Apply *K-means* algorithm for subsample Sm for k1 clusters.

4. Compute $J_C(M) - \sum^{1}\text{-}1 \sum_{X\text{-}1} | X_1 - Z_1 |^2$

5. Choose minimum of as the refined initial points Zj

$$j_C, [1, k1]$$

6. Now apply *k-means* algorithm again on dataset S for k1 clusters.

7. Combine two nearest clusters into one cluster and recalculate the new cluster center for the combined cluster until the number of clusters reduces into k.

IV. AFFINITY PROPAGATION

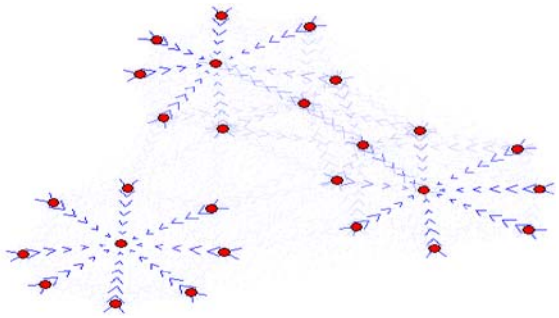

Fig 1. Affinity Propagatio

Affinity propagation is a new algorithm that takes as input measures of similarity between pairs of data points and simultaneously considers *all* data points as potential exemplars. Real-valued messages are exchanged between data points until a high quality set of exemplars and corresponding clusters gradually emerges. We have used affinity propagation to solve a variety of clustering problems and we found that it uniformly found clusters with much lower error than those found by other methods, and it did so in less than one-hundredth the amount of time. Because of its simplicity, general applicability, and performance, we believe affinity propagation will prove to be of broad value in science and engineering. The figure shows affinity propagation among a small set of two dimensional data point. Input consists of a collection of real-valued similarities between data points.

These are represented by s(i,k) which describes how well data point k is suited to be exemplar for data point i. In addition, each data point is supplied with a preference value s(k,k) which specifies apriori how likely each data point is to be an exemplar. This preference value can be set to something uninformative, allowing the clustering procedure to learn from the data an appropriate number of clusters. Alternative preference information can be utilized to minimize clusters or infuse the system with prior information.

In general the algorithm works in three steps:

***Step1*:** Update responsibilities given availabilities. Initially this is a data driven update, and over time lets candidate exemplars competition for ownership of the data.

Step2: Update availabilities given the responsibilities. This gathers evidence from data points as to whether a candidate exemplar is a good exemplar.

Step3: Monitor exemplar decisions by combining availabilities and responsibilities. Terminate if reach a stopping point (e.g. insufficient change). The update rules require simple local computations and messages are exchanged between pairs of points with known similarities

Affinity Propagation (Frey and Dueck, 2006; 2007) takes as input a collection of real-valued similarities between data points, where the similarity $s(i, k)$ indicates how well the data point with index k is suited to be the class center for data point i. When the goal is to minimize the squared error, each similarity is set to a negative Euclidean distance: for points $\boldsymbol{x}i$ and $\boldsymbol{x}k$, $s(i, k) = -||\boldsymbol{x}i - \boldsymbol{x}k||2$.Rather than requiring that the number of clusters be pre specified, Affinity Propagation

takes as input areal number $s(k, k)$ for each data point k so that data points with larger values of $s(k, k)$ are more likely to be chosen as class centers. These values are referred to as 'preferences'. Affinity Propagation can be viewed as searching configurations of the labels $c=\{c1, c2, \ldots, cn\}$ to minimize the energy:

$$E(c) = -\sum_{\iota=1}^{N} s(i, c_\iota)$$

The process of Affinity Propagation can be viewed as a message communication process on a factor graph (Kschischang *et al.*, 2001). There are two kinds of messages exchanged between data points, i.e., 'responsibility' and 'availability'. The responsibility $r(i, k)$, sent from data point i to candidate exemplar point k, reflects the accumulated evidence for how well-suited point k is to serve as the exemplar for point i, taking into account other potential exemplars for point i. The availability $a(i, k)$, sent from candidate exemplar point k to point i, reflects the accumulated evidence For how appropriate it would be for point i to choose point k as its exemplar, taking into account the support from other points that point k should be an exemplar. The messages need only be exchanged between pairs of points with known similarities

Input:

$s(i, k)$: the similarity of point i to point k.

$p(j)$: the preferences array which indicates the preference that data point j is chosen as a cluster center.

Output:

idx(j): the index of the cluster center for data point j.

dpsim: the sum of the similarities of the data points to their cluster centers.

netsim: the net similarity (sum of the data point similarities and preferences).

expref: the sum of the preferences of the identified cluster centers

netsim: the net similarity (sum of the data point similarities and preference)

Steps:

***Step1*:** Initialization the availability a (i.k) to zero

$$a(i, k) = 0 \qquad (1)$$

step2: update the responsibility using rule

$$r(i,k) \leftarrow s(i ,k) - \max \{a(i, k'). s(i, k')\}.$$

$$k's.t.\ k' \neq k \qquad (2)$$

***step3*:** update the availability using the rule

$$a(i,k) \leftarrow \min\{0, r(k,k) + \sum_{i's.t.\ i' \neq i,k} \max\{0, r(i',k)\}\}$$

$$(3)$$

The self-availability is updated differently

$$a(k,k) \leftarrow \sum_{i's.t.\ i' \neq k} \max\{0, r(i',k)\}. \qquad (4)$$

***Step 4*:** The message-passing procedure may be terminated after a fixed number of iterations, after changes in the messages fall below a threshold or after the local decisions stay constant for some number of iterations.

Availabilities and responsibilities can be combined to make the exemplar decisions. For point i, the value of *k* that maximizes a(i, k)+r(i, k) either identifies point *i* as an exemplar if *k=i* or identifies the data point thatis the exemplar for point *i*. When updating the messages, numerical Oscillations must be taken into consideration. As a result, each message is set to λ times its value from the previous iteration plus $1-\lambda$ times its prescribed updated value. The λ should be larger than or equal to 0.5 and less than 1. If λ is very large, numerical oscillation may be avoided, but this is not guaranteed. Hence a maximal number of iterations are set to avoid infinite iteration in AP clustering

V. DATASET

The colon dataset is a collection of gene expression measurements from 62 colon biopsy samples reported by Alon. It contains 22 normal and 40 colon cancer samples .the colon dataset consists of 2000 genes.

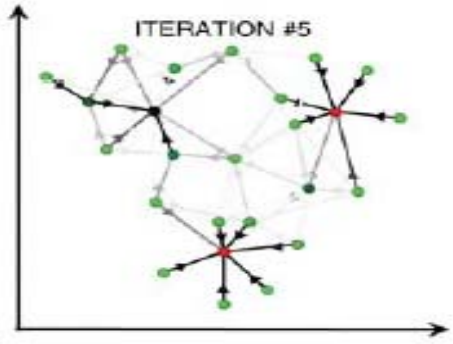

Fig 2. Iteration of affinity propagation on dataset.

TABLE 1

Result Over Clustering Algorithm Using 2000 Gene Colon Dataset (Total Number of Records Present In Data Set =62)

Clustering Algorithm	**Correctly Classified**	**Average accuracy**
K-means	33	53.23
Efficient k-means	37	58.06
Affinity Propagation	35	61.32

VI. CONCLUSION AND FUTURE WORK

The analysis of k-*means* algorithm is done with the help of cancer dataset (colon dataset). The *k-means* use in this study is efficient *k-means* and affinity propagation, the average accuracy rate of these is show below in table. Analysis of 2000 colon dataset the average accuracy of affinity propagation is better than *k-means* and efficient *k-means*. The convergence rate is also higher and speed of execution time is good however the variations of k-means required more trails to reach at a stable and good clustering solution. Performance of this algorithm can be improved with the help of variants global k-means, k++, fuzzy logic to get better quality of cluster. So these algorithm help to get good result.

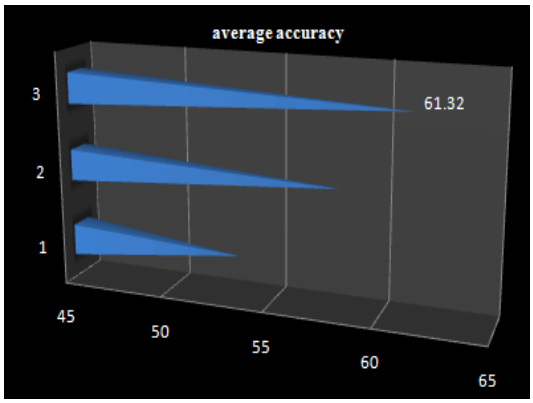

Fig 2.Average Accuracy Graph

REFERENCE

[1] Alizadeh A., Eisen M.B, Davis R.E, et al. Distinct types of diffuse large B-cell lymphoma identified by gene expression profiling. Nature. 2000; 403(6769):503–511.
[2] Brendan j.Frey and Delbert Duec clustering by passing message between data point science, 315(5814):972{976}
[3] Golub T.R, Slonim D.K, Tamayo P, et al. Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. Science. 1999;286(5439):531–53
[4] likas,A., Vlassis,M.& Verbeek,j.(2003), the global *k-means* clustering algorithm, pattern recognition,36,451-461.
[5] Nielsen T.O, West R.B, Linn S.C, et al. [11] Bell, R.M., Koren, Y., Volinsky, C., 2007. Modeling Relationships a Multiple Scales to Improve Accuracy of Large Recommender Systems. Proc. 13th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining, San Jose, California USA, p.95-104.
[6] Frey, B.J., Dueck, D., 2006. Mixture Modeling by Affinity Propagation. Neural Information Processing neural information processing system.
[7] Guha, S., Rastogi, R., Shim, K., 2001. CURE: an efficient clustering algorithm for large databases. Inf. Syst., 26(1): 35-58
[8] J.A. Lozano, J.M. Pena, P. Larranaga, An empirical comparison of four initialization methods for the k-means algorithm Lett. 20 (1999) 1027–1040
[9] G.W. Milligan, M.C. Cooper, An examination of procedures for determining the number of clusters in a data set, Psychometrika 50 (1985) 159–179.
[10] Anjan Goswami. Department of Computer Science and Engineering" Fast and Exact Out of-Core and Distributed K-Means Clustering 2001
[11] Bagirov, A.M.[Adil M.], Modified global k-means algorithm for minimum sum-of-squares clustering problems, October 2008
[12] E. Papageorgiou, I. Kotsioni, A. Linos, "Data Mining: A New Technique In Medical Research", Hormones 2005, 4(4):189-191
[13] Jaiwei Han, Michelinne Kamber, "Data Mining : Concepts and Techniques ", 2001, II Edition
[14] Jason T.L. Wang, Mohammed J. Zaki, Hannu T.T Toivonen and Dennis Shasha (EDS), "Data mining in bioinformatics" Pg. no: 654, Springer International Edition tumours: a gene expression study. Lancet2002
[15] MacQueen, J.B. (1967). Some Methods for Classification and Analysis of Multivariate Observations. In Proc. of 5th Berkley Symposium on Mathematical Statistics and Probability, Volume I: Statistics, pp. 281–297.
[16] Pawlak. Z. Rough Sets International Journal of Computer and Information Sciences, (1982), 341-356.
[17] P awan Lingras, Chad West. Interval set Clustering of Web users with Rough k-Means, submitted to the Journal of Intelligent Information System in 2002.
[18] Yeung K.Y, Haynor D.R, Ruzzo W.L. Validating clustering for gene expression data. Bioinformatics. 2001.
[19] Zhang Y. , Mao J. and Xiong Z.: An efficient Clustering algorithm, In Proceedings of Second International Conference on Machine Learning and Cyber netics, November 2003.
[20] L. Kaufman and P. J. Rousseeuw, Finding Groups in Data: an Introduction to Cluster Analysis, John Wiley & Sons, 1990.

AUTHOR PROFILE

D. Napoleon received the Master's Degree in Computer Applications from Madurai Kamaraj University, Tamil Nadu, India in 2002, and the M.Phil degree in Computer Science from Periyar University, Salem, Tamil Nadu, India in 2007. He has published articles in National and International Journals. He has presented papers both in National and International Conferences. His Current research interest includes: Knowledge discovery in Data Mining and Computer Networks.

G.Baskar received his Master's degree in Information Technology in K.S.Rangasamy College of Technology, Tiruchengode, Tamil Nadu India in 2008 and M.Phil Degree in Computer Science from Bharathiar University, Coimbatore, Tamil Nadu, India in 2010. His area of interest includes Data Mining**.**

S.Pavalakodi received Master's degree in Computer Science from Sri Ramalinga Sowdambigai college of Commerce and Scienc, Coimbatore in 2006, and M.Phil Degree in Computer Science from Bharathiar University, Coimbatore, Tamil Nadu, India in 2010. Her area of interest includes Data Mining