ISSN: 2321-7782 (Online)
Volume 2, Issue 9, September 2014
International Journal of Advance Research in Computer Science and Management Studies
Research Article / Survey Paper / Case Study
Available online at: www.ijarcsms.com

Implementation of an Entropy Weighted K-Means Algorithm for High Dimensional Sparse Data

S. Subha Indu[1]
MCA., M.Phil
Assistant Professor
Dept. of CA and SS
Sri Krishna Arts and Science College
Coimbatore, Tamil Nadu – India

K. Devika Rani Dhivya[2]
M.Sc (CS).,M.Phil.,M.B.A
Assistant Professor
Dept. of CA and SS
Sri Krishna Arts and Science College
Coimbatore, Tamil Nadu – India

Abstract: This paper contains a partitional based algorithm for clustering high-dimensional objects in subspaces for Iris gene dataset. In high dimensional data, clusters of objects often exist in subspaces rather than in the entire space. This is the data sparsity problem faced in clustering high-dimensional data. In the proposed algorithm, we extend the K-Means clustering process to calculate a weight for each dimension in each cluster and use the weight values to identify the subsets of important dimensions that categorize different clusters. This is achieved by including the weight entropy in the objective function that is minimized in the K-Means clustering process. An additional step is added to the K-Means clustering process to automatically compute the weights of all dimensions in each cluster. The experiments on both synthetic and real data have shown that the algorithm can generate better clustering results than other subspace clustering algorithms. The new algorithm is also scalable to large data sets

Keywords: K-Means, clustering, subspace clustering, high-dimensional data.

I. INTRODUCTION

High-dimensional data is a phenomenon in real-world data mining applications. The typical data clustering tasks are directly performed in the data space. However, the space is always of very high dimensionality, ranging from several hundreds to thousands. Due to the consideration of the curse of dimensionality, it is desirable to first project the data into a lower dimensional subspace in which the semantic structure of the data space becomes clear. In the low dimensional semantic space, the traditional clustering algorithms can be then applied. Clearly, clustering of high-dimensional sparse data requires special treatment. This type of clustering methods is referred to as subspace clustering, aiming at finding clusters from subspaces of data instead of the entire data space. In a subspace clustering, each cluster is a set of objects identified by a subset of dimensions and different clusters are represented in different subsets of dimensions. The results of extensive experiments on both synthetic and real data have demonstrated that the new algorithm outperformed the other subspace clustering algorithms in clustering accuracy. It was also effective in clustering sparse data and scalable in clustering large data with respect to the number of dimensions and the number of clusters the major challenge of subspace clustering, which makes it distinctive from traditional clustering, is the simultaneous determination of both cluster memberships of objects and the subspace of each cluster.

Consider that different dimensions make different contributions to the identification of objects in a cluster. The difference of contribution of a dimension is represented as a weight that can be treated as the degree of the dimension in contribution to the cluster. In subspace clustering, the decrease of the weight entropy in a cluster implies the increase of certainty of a subset of dimensions with larger weights in determination of the cluster. Therefore, in the clustering process, simultaneously minimize the within cluster dispersion and maximize the negative weight entropy to stimulate more dimensions to contribute to the identification of a cluster. This can avoid the problem of identifying clusters by a few dimensions with sparse data. The formula for computing a dimension weight is derived based on Liping Jing, Michael K. Ng, and Joshua Zhexue Huang [1] and added to the K-Means clustering process as an additional step in each iteration, so the cluster memberships of objects and the weights of dimensions in each cluster can be obtained simultaneously.

II. Related Work

A lot of work has been done in the area of clustering. In general, most of the common algorithms fail to generate meaningful results because of the inherent sparsity of the objects. In such high dimensional feature spaces, data does not cluster anymore. But usually, there are clusters embedded in lower dimensional subspaces. In addition, objects can often be clustered differently in varying subspaces. Local Dimensionality Reduction [9], like PROCLUS, projects each cluster on its associated subspace, which is generally different from the subspace associated with another cluster. The efficacy of this method depends on how the clustering problem is addressed in the first place in the original feature space. A potentially serious problem with such a technique is the lack of data to locally perform PCA on each cluster to derive the principal components, therefore, it is inflexible in determining the dimensionality of data representation.

Subspace clustering seeks to group objects into clusters on subsets of dimensions or attributes of a data set According to the ways with which the subsets of dimensions are identified, divide subspace clustering methods into two categories. The methods in the first category determine the exact subsets of dimensions where clusters are discovered. This is called as hard subspace clustering. The methods in the second category determine the subsets of dimensions according to the contributions of the dimensions in discovering the corresponding clusters. The contribution of a dimension is measured by a weight that is assigned to the dimension in the clustering process [2], [3]. This method called as soft subspace clustering because every dimension contributes to the discovery of clusters, but the dimensions with larger weights form the subsets of dimensions of the clusters. The method in this paper falls in the second category.

A. Hard Subspace Clustering

The subspace clustering methods in this category can be further divided into bottom-up and top-down subspace search methods. The bottom-up methods for subspace clustering consist of the following main steps:

1. Dividing each dimension into intervals and identifying the dense intervals in each dimension.

2. From the interactions of the dense intervals, identifying the dense cells in all two dimensions.

3. From the intersections of 2D dense cells and the dense intervals of other dimensions, identifying the dense cells in all three dimensions and repeating this process until all dense cells in all k dimensions are identified.

4. Merging the adjacent dense cells in the same subsets of dimensions to identify cluster.

B. Soft Subspace Clustering

Instead of identifying exact subspaces for clusters, this approach assigns a weight to each dimension in the clustering process to measure the contribution of the dimension in forming a particular cluster. However, the purpose is to select important variables for clustering. Extensions to some variable weighting methods, for example, the K-Means type variable weighting methods, can perform the task of subspace clustering[4]. A number of algorithms in this direction have been reported recently[7]. Unlike variable selection in which a weight is assigned to a dimension for the entire data set, assign a weight to each dimension for each cluster. As such, different clusters have different sets of weight values. To retain the scalability, the K-Means clustering process is adopted in these new subspace clustering algorithms. In each iteration, an additional step is added to compute the weight values[8]. The direct extension to the k-means type variable, weighting algorithm for variable selection results from the minimization of the following objective function.

Here, n, k, and m are the numbers of objects, clusters, and dimensions, respectively $\beta\ (>1)$ and $\eta\ (\geq 1)$ are two parameters greater than 1. w_{lj} is the degree of membership of the j_{th} object belonging to the l_{th} cluster. γ is the weight for the

i_{th} dimension in the l_{th} cluster. Xji is value of the i_{th} dimension of the jth object, and z_{li} is the value of the i_{th} component of the l_{th} cluster center. $\eta =$ 1 produces a hard clustering, whereas η > 1 results in a fuzzy clustering. There are three unknowns W, Z, and A that need to be solved.

III. ENTROPY WEIGHTING K-MEANS

The weight of a dimension in a cluster represents the probability of contribution of that dimension in forming the cluster. The entropy of the dimension weights represents the certainty of dimensions in the identification of a cluster [5],[6]. Therefore, modify the objective function by adding the weight entropy term to it so that simultaneously minimize the within cluster dispersion and maximize the negative weight entropy to stimulate more dimensions to contribute to the identification of clusters[10]. The positive parameter γ controls the strength of the incentive for clustering on more dimensions.

A. *PROPOSED ALGORITHM*

Input: The number of clusters k and parameter γ; Randomly choose k cluster centers and set all initial weights to 1=m;

REPEAT

Update the partition matrix W by equation;

$$\begin{cases} w_{lj} = 1, & \text{if } \sum_{i=1}^{m} \lambda_{li}(z_{li} - x_{ji})^2 \leq \sum_{i=1}^{m} \lambda_{ri}(z_{ri} - x_{ji})^2 \\ & \text{for } 1 \leq r \leq k, \\ w_{lj} = 0, & \text{otherwise.} \end{cases}$$

Update the cluster centers Z by eqn;

$$z_{li} = \frac{\sum_{j=1}^{n} w_{lj} x_{ji}}{\sum_{j=1}^{n} w_{lj}} \quad \text{for } 1 \leq l \leq k \text{ and } 1 \leq i \leq m.$$

Update the dimension weights A by eqn;

$$\lambda_{lt} = \frac{\exp\left(\frac{-D_{lt}}{\gamma}\right)}{\sum_{i=1}^{M} \exp\left(\frac{-D_{li}}{\gamma}\right)},$$

UNTIL (the objective function obtains its local minimum value);

The input parameter γ is used to control the size of the weights as follows:

- γ > 0. In this case, λ_{li} is inversely proportional to. The smaller D_{li}, the larger λ_{li}, the more important the corresponding dimension.

- $\gamma = 0$. $\lambda_{li'}$ is equal to one, indicating that the index i' has the smallest value of $D_{li'}$. The other weights λ_{li} for $i \neq i'$ are equal to zero. Each cluster contains only one important dimension. It may not be desirable for high-dimensional data sets.

- γ < 0. In this case, λ_{li} is proportional to D_{li}. The larger D_{li}, the larger λ_{li}. This is contradictory to the original idea of dimension weighting. Therefore, γ cannot be smaller than zero.

Partitioning the objects: After initialization of the dimension weights of each cluster and the cluster centers, a cluster membership is assigned to each object.

Cluster centers: Given the partition matrix W, updating cluster centers is to find the means of the objects in the same cluster. Thus, for k clusters, the computational complexity for this step is O(mnk)

Calculating dimensions weights: The last phase of this algorithm is to calculate the dimensions weights for all clusters based on the partition matrices W and Z. The computational complexity of this step is also O(mnk).

IV. EXPERIMENTAL RESULTS AND DISCUSSIONS

The proposed work is implemented using MATLAB for Iris gene dataset available in NCBI database. The K-means Algorithm was applied on Iris dataset. Due to the disadvantages arised in K-Means Algorithm, it is extended to calculate a weight for each dimension in each cluster of same dataset which is termed as EWK-Means algorithm. The cluster accuracy is achieved high in EWK-Means by assigning weight values to the dimensions due to this performance is high. One of the major advantages of EWK-Means algorithm is that Normal root Mean Squared Error (nrmse) value is minimized when compared to K-Means algorithm. With the same cluster values the accuracy of both K-Means and EWK-Means is reported here. Number of Dimensions and distribution of weight in EWK-Means is depicted in Fig 2.Fig.3 and fig 4 shows the output screen.

No of Dimensions	EWK-Means
100	**0.07**
150	**0.09**
200	**0.04**
250	**0.05**
300	**0.07**
350	**0.06**
400	**0.03**

Table 1: Weight Distribution

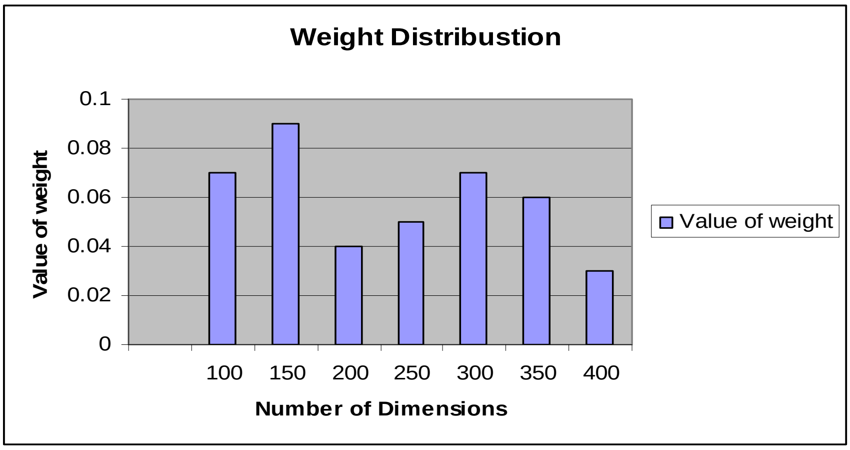


Fig.1: Weighted Distribution

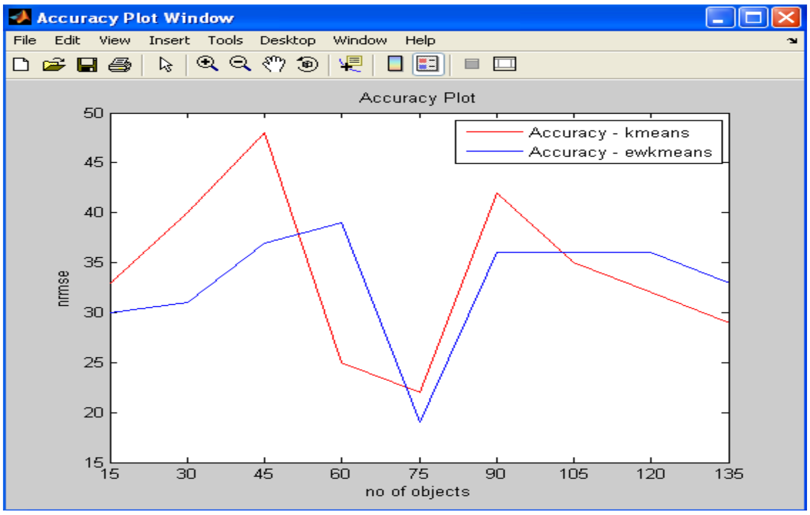
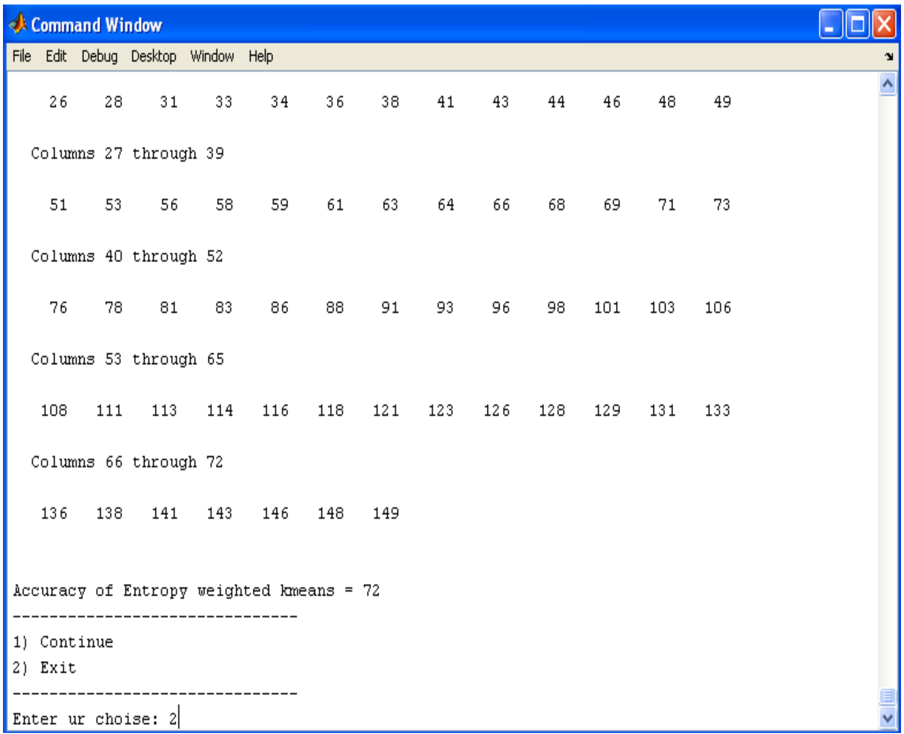


Fig.2 Comparison between the Methods Showing NRMSE



```
    26    28    31    33    34    36    38    41    43    44    46    48    49

  Columns 27 through 39

    51    53    56    58    59    61    63    64    66    68    69    71    73

  Columns 40 through 52

    76    78    81    83    86    88    91    93    96    98   101   103   106

  Columns 53 through 65

   108   111   113   114   116   118   121   123   126   128   129   131   133

  Columns 66 through 72

   136   138   141   143   146   148   149


Accuracy of Entropy weighted kmeans = 72
--------------------------------
1) Continue
2) Exit
--------------------------------
Enter ur choise: 2
```

Fig.3 Sample Screen Showing Implementation

V. Conclusion

In EWK-Means, K-Means type subspace clustering algorithm for high-dimensional sparse data. In this algorithm, we simultaneously minimize the within cluster dispersion and maximize the negative weight entropy in the clustering process. Because this clustering process awards more dimensions to make contributions to identification of each cluster, the problem of identifying clusters by few sparse dimensions can be avoided. As such, the sparsity problem of high-dimensional data is tackled.

References

1. L. Jing, M.K. Ng, J. Xu, and J.Z. Huang, “On the Performance of Feature Weighting k-Means for Text Subspace Clustering,” Proc. Sixth Int’l Conf. Web-Age Information Management, pp. 502-512, 2005.
2. Hore, Prodip, Hall, Lawrence O., Goldgof, Dmitry B., Gu, Yuhua, Maudsley, Andrew A., Darkazanli, Ammar, 2009b. A scalable framework for segmenting magnetic resonance images. J. Signal Process. Systems 54 (1–3), 183–203.
3. M. Zait and H. Messatfa, “A Comparative Study of Clustering Methods,” Future Generation Computer Systems, vol. 13, pp. 149- 159, 1997.
4. M. Steinbach, G. Karypis, and V. Kumar, “A Comparison of Document Clustering Techniques,” Proc. KDD Workshop Text Mining, 2000.
5. A.K. McCallum, “Bow: A Toolkit for Statistical Language Modeling, Text Retrieval, Classification and Clustering,” http://www.cs.cmu.edu/mccallum/bow, 1996.

6. Y. Zhao and G. Karypis, “Comparison of Agglomerative and Partitional Document Clustering Algorithms,” Technical Report #02-014, Univ. of Minnesota, 2002.

7. S. Zhong and J. Ghosh, “A Comparative Study of GenerativeModels for Document Clustering,” Proc. SDW Workshop ClusteringHigh-Dimensional Data and Its Applications, May 2003.

8. H. Frigui and O. Nasraoui, “Simultaneous Clustering and Dynamic Keyword Weighting for Text Documents,” Survey of Text Mining, Michael Berry, ed., pp. 45-70, Springer, 2004.

9. C. Domeniconi, D. Papadopoulos, D. Gunopulos, and S. Ma, “Subspace Clustering of High Dimensional Data,” Proc. SIAM Int’l Conf. Data Mining, 2004.

10. J.H. Friedman and J.J. Meulman, “Clustering Objects on Subsets of Attributes,” J. Royal Statistical Soc. B, vol. 66, no. 4, pp. 815-849, 2004.

11. J.Z. Huang, M.K. Ng, H. Rong, and Z. Li, “Automated Variable Weighting in k-Means Type Clustering,” IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 27, no. 5, pp. 1-12, May 2005.

12. Anil K. Jain ,”An Entropy Weight K- means Algorithm for Subspace Clustering of High-Dimensional Sparse data “, elixir journal publication , Pattern Recognition Letters 31 (2010) 651–666, Available online 9 September 2009.

13. Anil Kumar Tiwari, Lokesh Kumar Sharma, G. Rama Krishna, “Entropy Weighting Genetic K-means Algorithm for Subspac Clustering”, International Journal of Computer Applications, Number 7 - Article 5, 2010.

14. CAO Zhiguang, CHEN Wei, MA Rubao, “Application of Kmeans and maximum weighted entropy on color image segmentation”, ISSN-1002 8331, 2013.

15. HUANG Fen1, YU Qi1, YAO Xia2, SHANG Guiyan2, ZHU Yan2, WU Yanlian1, HUANG Yu2. “K-means clustering segmentation for H weight of wheat canopy image” [J]. CEA, 2014, 50(3): 129-134, 2013.

AUTHOR(S) PROFILE

Prof. S. Subha Indu, MCA., M.Phil., MBA., Asst. Professor, CA & SS Department, has received the MCA degree and M.Phil degree in Computer Science from Bharathiar university in 2011and 2013, respectively. She has 5 years of teaching. She has published papers in International journals. She has presented papers in various National and State level conferences.

Prof. K. Devika Rani Dhivya M.Sc., M.Phil., MBA., Asst. Professor, CA & SS Department, has received the M.Sc degree in Computer Science, M.Phil degree in Computer Science and M.B.A degrees in Information Systems from Bharathiar university in 2011, 2012 and 2013, respectively. She has 2 years of teaching. She has published papers in International journals. She has presented papers in various National and State level conferences.