International Journal of Global Engineering (IJGE) E- ISSN: 2456-3099
VOL 2 ISSUE 1 (2017) PAGES 33 - 39
Received: 1/09/2017. Published: 26/09/2017

PFK-MEANS: A PARAMETER FREE K-MEANS ALGORITHM

N. Sundararajulu[1], M. Nandhini[2], P. Jayabharathi[3]
[1]Associate Professor, [2]Assistant Professor, [3]Assistant Professor
Department of Computer Science and Engineering
Dhanalakshmi Srinivasan College of Engineering and Technology, Tamil Nadu, India

Abstract

K-means clustering is broadly used for its efficiency. However, this algorithm suffers from two principal drawbacks: first, the user ought to specify in advance the right wide variety of clusters, which is normally a difficult task; second, its closing consequences depend on the initial starting points. The existing paper intends to overcome these issues by way of proposing a parameter free algorithm based on k-means (called pfk-means). We evaluated its overall performance by applying on several widespread datasets and evaluate with gmeans, a associated well be aware of computerized clustering method. Our overall performance studies have demonstrated that the proposed approach is fantastic in predicting the correct wide variety of clusters and producing consistent clustering results.
Keywords: k-means, gmeans, Parameter free, computerized clustering.

INTRODUCTION

In Data Mining, clustering consists of grouping a given dataset into a predefined variety of disjoint sets, called clusters, so that the elements in the same cluster are greater comparable to each other and more distinctive from the elements in the other cluster. This optimization trouble is acknowledged to be NP-hard, even when the clustering procedure offers with solely two clusters (Aloise 1980). Therefore, many heuristics and approximation algorithms have been proposed, in order to find near most advantageous clustering answer in reasonable computational time. The most outstanding clustering algorithm kmeans is a grasping algorithm which has two stages: Initialization, in which we set the seed set of centroids, and an iterative stage, known as Lloyd's algorithm (Lloyd., S. P.1982). Additionally, Lloyd's algorithm has two steps:

The undertaking step, in which every object is assigned to its closest centroid, and the centroid's update step. The essential gain of k-means is its quickly convergence to a neighborhood minimum, but k-means has two important drawbacks: first, the person ought to specifies in develop the correct wide variety of clusters, which is usually a difficult task; second, the algorithm is sensitive to the initial beginning points. In this paper, an alternative parameter free method for computerized clustering, called pfk-means, is proposed. Algorithm validation and comparative study with gmeans (Hamerly and Elkan 2003), a associated properly recognised algorithm, are carried out using countless real-world and artificial clustering records sets from he UCI Machine Learning Repository- UCIMLR (Asuncion et.al 2007). In the subsequent section, some associated works are briefly discussed. Then the proposed strategy is described in Section three Section 4 affords functions outcomes of this clustering approach to exclusive standard information sets and reviews its performance. Lastly, conclusion of this paper is summarized in Section 5.

2. RELATED WORK

Despite the reality that obtaining an most advantageous range of clusters k for a given records set is an NP-hard hassle (Spath 1980), numerous technique have been developed to find k automatically. Pelleg and Moore (2000) brought the X-means algorithm, which proceed through getting to know k with kmeans using the Bayesian Information Criterion (BIC) to rating each model, and chooses the model with the best possible BIC score. However, this approach tends to overfit when it deals with records that occur from non-spherical clusters. Tibshirani et al. (2001) proposed the Gap statistic, which compares the probability of a realized mannequin with the distribution of the probability of models trained on facts drawn from a null distribution. This method is appropriate for discovering a small variety of clusters, but has challenge when ok increases.

Cheung (2005) studied a rival penalized competitive studying algorithm, and Xu (1997, 1996) has established a very appropriate end result in finding the cluster number. Lee and Antonsson (2000) used an evolutionary approach to dynamically cluster a data set. Sarkar,et al,. (1997) and Fogel, Owens, and Walsh (1966) are proposed an method to dynamically cluster a statistics set the use of evolutionary programming, where two fitness features are concurrently optimized: one offers the ideal quantity of clusters, whereas the different leads to a perfect identification of every cluster's centroid. Recently Swagatam Das and Ajith Abraham (2008) proposed an Automatic Clustering the usage of Differential Evolution (ACDE) algorithm with the aid of introducing a new chromosome representation.

Hamerly and Elkan (2003) proposed the gmeans algorithm, primarily based on K-means algorithm, which uses projection and a statistical take a look at for the hypothesis that the facts in a cluster come from a Gaussian distribution. This algorithm works successfully if clusters are well-separated, and fails when clusters overlap and appear non-Gaussian. In our experiments, gmeans tends to overestimate the number of clusters, as mentioned in section 4 The majority of these strategies to decide the satisfactory range of clusters may additionally now not work very nicely in practice. In the current work, an alternative strategy is proposed, trying to overcome these issues.

3. PROPOSED APPROACH

The proposed algorithm starts by way of setting kmax=floor((n) 1/2), where n is the wide variety of objects in the given facts set. This choice is encouraged via the fact that the quantity of clusters lies in the vary from 2 to (n)1/2, as stated by means of Pal and Bezdek (1995). Then it applies a deterministic initialization method proposed by way of Kettani et al. (2013) (called KMNN) by using splitting the complete dataset into two clusters. K-means algorithm is then utilized with these two initial centroids. Again, the biggest cluster is then break up into two clusters by KMNN. This manner is repeated till k=kmax, and at each iteration, the maximum of CH cluster validity index (Calinski and Harabasz 1974) of the cutting-edge partition is stored. We used this index because it is rather cheaper to compute, and it commonly outperforms other cluster validity indices as stated by Milligan and Cooper (1985). Finally, the algorithm outputs the most appropriate k and partition corresponding to the most value of CH stored so far. This algorithm is outlined in the pseudo-code below:

Algorithm pfk-means

Algorithm pfk-means

Input: $X = \{x_1, x_2, \ldots, x_n\}$ in R^d

Output: k mutually disjoint clusters $C_1, \ldots, C_k$ such that $\bigcup_{j=1}^{k} C_j = X$

```
kmax ← ⌈ (n)^1/2 ⌉
[m1,m2] ← KMNN(X,2)
ko ←2
Io ← I
mo ← m
for h=2:kmax-1
      j← argMin(| C_i |)
           i<=k
      [p1,p2] ← KMNN(C_j,2)
       m_j ←  p1
       m_h+1 ←  p2
       [I,m] ← kmeans(X,h+1,'start',m)
       if CHo <CH(I) then
                    ko ←h+1
                    Io ← I
                    CHo ← CH(I)
               mo ← m
          end if
end for
Output: m, ko and Io
```

4 EXPERIMENTAL RESULTS

Algorithm validation is performed the use of distinctive data units from the UCI Machine Learning Repository [10]. We evaluated its overall performance by making use of on quite a few benchmark datasets and examine with gmeans (Hamerly and Elkan 2003). Silhouette index (Kaufman and Rousseeuw 1995) which measures the cohesion primarily based on the distance between all the points in the same cluster and the separation primarily based on the nearest neighbor distance, used to be used in these experiments in order to consider clustering accuracy. (bigger average silhouette value indicates a greater clustering accuracy). Silhouette index is based on distances between observations in the identical cluster and in one of a kind clusters. Given observation i , let ai be the average distance from point i to all different factors in equal cluster and d i, j represents the average distance from factor i to all points in any other cluster j . Finally, let bi denotes the minimal of these average distances d two i, j two . The silhouette width for the i-th statement is: silh(i) (bi two ai)/max two ai , bi two The average silhouette width can be locate through averaging silh(i) over all observations: The silhouette width silh(i) ranges from -1 to 1. If an observation has a cost shut to 1, then the facts point is nearer to its personal cluster than a neighboring one. If it has a silhouette width close to -1, then it is not very properly clustered. A silhouette width close to zero suggests that the observation ought to just belong to modern cluster or one that is near to it. Kaufman and Roussee use the common silhouette width to estimate the quantity of clusters in a facts set by the use of the partition with two or extra clusters that yields the greatest average silhouette width. Experimental outcomes are mentioned in desk 1 and determine 1, and some clustering outcomes are depicted in figure 2 to 7.

TABLE 1: Investigational results of gmeans and pfk-means application on dissimilar datasets in term of average Silhouette value.

		gmeans		pfk-means	
Data set	**k**	**k found**	**Mean sil.**	**k found**	**Mean sil.**
Iris	3	5	0.6744	**3**	**0.7541**
Ruspini	4	5	0.8772	**4**	**0.9086**
Breast	2	39	0.2644	**2**	**0.7541**
Aggregation	7	13	**0.6562**	28	0.5662
Compound	6	17	0.6193	2	**0.8302**
Pathbased	3	9	0.4499	12	0.5567
Spiral	3	**3**	0.5286	17	**0.5344**
D31	31	**31**	**0.9221**	**31**	**0.9221**
R15	15	16	0.9134	**15**	0.9360
Jain	2	17	0.6006	14	**0.6227**
Flame	2	4	**0.6302**	8	0.5572
Dim32	16	54	0.6244	**16**	**0.9961**
Dim64	16	49	0.8108	**16**	**0.9985**
Dim128	16	47	0.8331	**16**	**0.9991**
Dim256	16	48	0.755	**16**	**0.9996**
Dim512	16	45	0.8200	**16**	**0.9998**
Dim1024	16	47	0.6654	**16**	**0.9999**
a1	20	56	0.6057	**20**	**0.7891**
a2	35	74	0.6752	**35**	**0.7911**
a3	50	94	0.6570	**50**	**0.7949**
Thyroid	2	10	0.4726	3	**0.7772**
Glass	7	10	**0.7263**	15	0.6516
Wdbc	2	28	0.8388	4	**0.9983**
Wine	3	**3**	**0.5043**	**3**	**0.5043**
Yeast	10	55	0.4102	2	0.4102
S1	15	87	0.5027	**15**	**0.8802**
S2	15	87	0.5382	**15**	**0.8008**
S3	15	85	0.5371	**15**	**0.6661**
S4	15	92	0.5471	**15**	**0.6447**
t4.8k	6	108	0.5143	**35**	**0.5774**

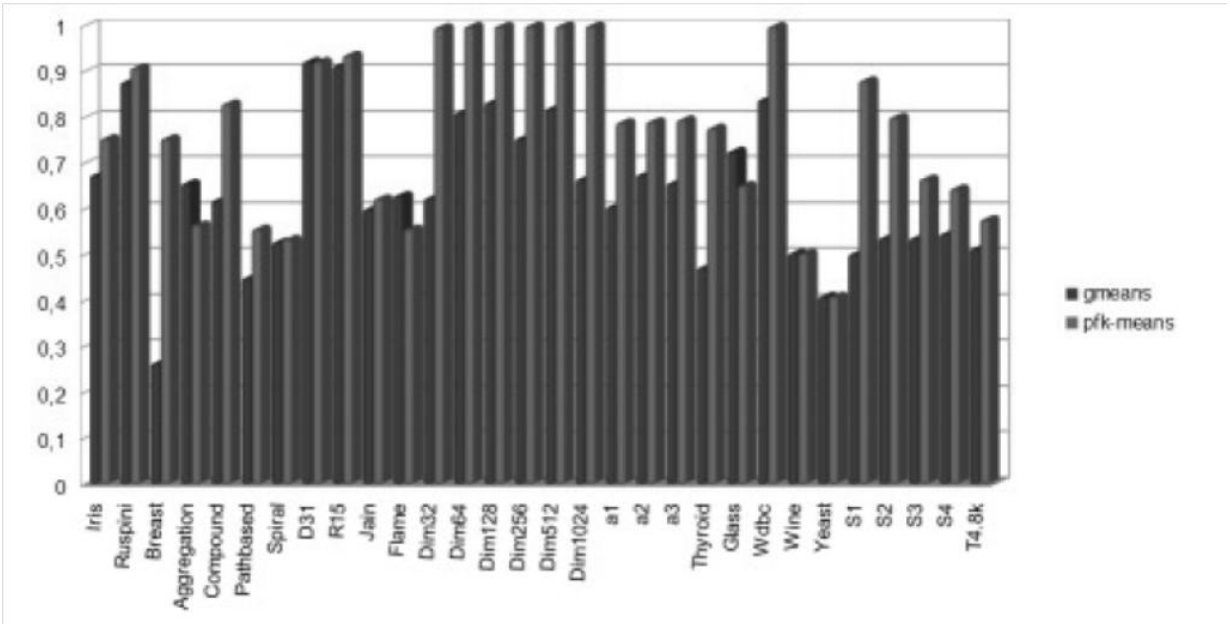

Fig 1: Diagram depicting of the mean Silhouette index for both pfk-means and gmeans applied on diverse datasets.

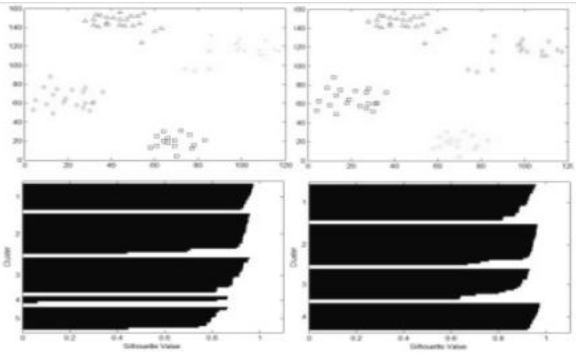

Fig 2: Clustering results of Ruspini dataset using gmeans (on left) and pfk-means (on right)

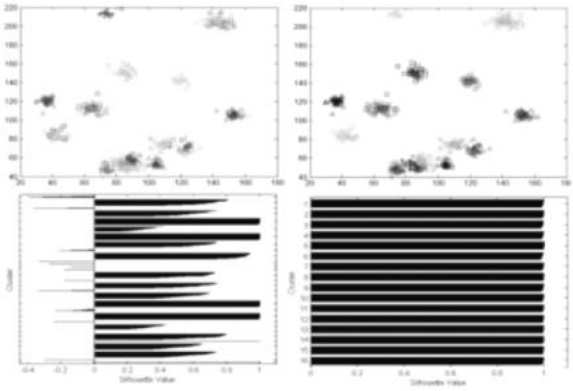

Fig 3: Clustering results of Dim32 dataset using gmeans (on left) and pfk-means (on right)

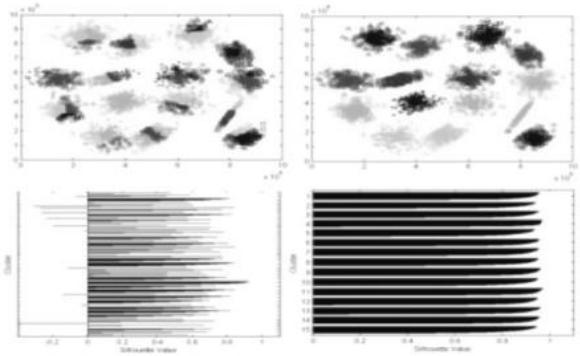

Fig 4: Clustering results of S1 dataset using gmeans (on left) and pfk-means (on right)

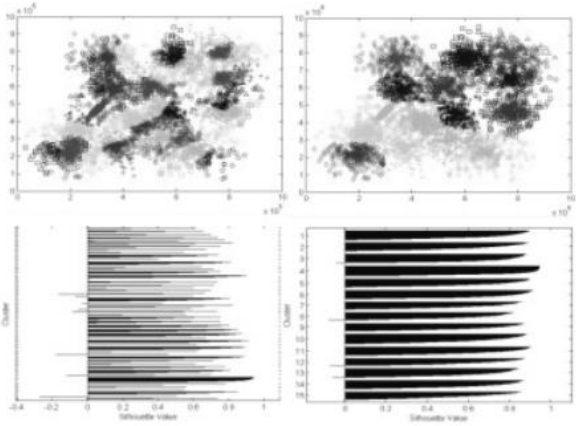

Fig 5: Clustering results of S2 dataset using gmeans (on left) and pfk-means (on right)

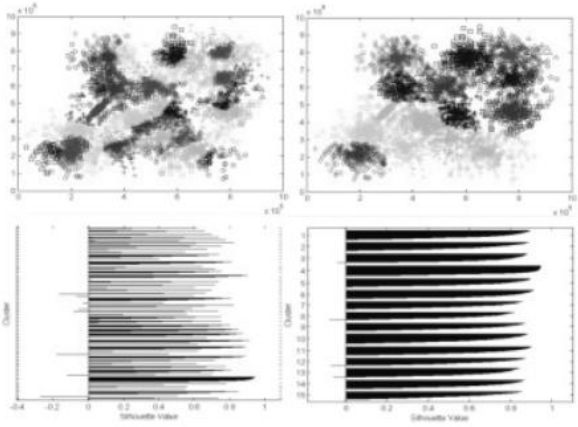

Fig 6: Clustering results of S3 dataset using gmeans (on left) and pfk-means (on right)

5 CONCLUSIONS

In this article, a parameter free k-means algorithm is recommended. The performance has been evaluated by applying on several standard datasets and compare with gmeans. The experimental study have established that it is effectual in producing consistent clustering results and have found the correct number of clusters with a successful rate of 63.33%. In the upcoming work, it will be of significance to find a tighter upper bound on the number of clusters, instead of n1/2 , in order to reduce the number of computations steps of the proposed approach. An additional probable augmentation will consist to choose a more appropriate similarity measure instead of Euclidian distance aiming to produce more accurate clustering results.

REFERENCES

[1] Aloise, D.; Deshpande, A.; Hansen, P.; Popat, P. (2009). "NP-hardness of Euclidean sumof-squares clustering". Machine Learning 75: 245–249. doi:10.1007/s10994-009-5103-0.

[2] Asuncion, A. and Newman, D.J. (2007). UCI Machine Learning Repository [http://www.ics.uci.edu/~mlearn/MLReposito ry.html]. Irvine, CA: University of California, School of Information and Computer Science.
[3] Lloyd., S. P. (1982). "Least squares quantization in PCM". IEEE Transactions on Information Theory 28 (2): 129–137. doi:10.1109/TIT.1982.1056489.
[4] T. Calinski and J. Harabasz. A dendrite method for cluster analysis. Communications in Statistics, 3:1–27, 1974.
[5] Cheung, Y. (2005) , "Maximum Weighted Likelihood via Rival Penalized EM for Density Mixture Clustering with Automatic Model Selection," IEEE Trans. Knowledge and Data Eng., vol. 17, no. 6, pp. 750-761.DOI:10.1109/TKDE.2005.97, http://ieeexplore.ieee.org/xpl/freeabs_all.jsp? arnumber=1423976
[6] Fogel, L.J, Owens, A. J. and Walsh, M. J (1996), "Artificial Intelligence Through Simulated Evolution". New York: Wiley.
[7] Greg Hamerly and Charles Elkan. Learning the k in k-means. In Proceedings of the seventeenth annual conference on neural information processing systems (NIPS), pages 281–288, 2003
[8] Kaufman and P. J. Rousseeuw. Finding groups in Data: "an Introduction to Cluster Analysis". Wiley, 1990.
[9] O.Kettani, B. Tadili and F. Ramdani. " A Deterministic K-means Algorithm based on Nearest Neighbor Search". International Journal of Computer Applications 63(15):33- 37, February 2013
[10] Lee, C.Y. and Antonsson, E.K. (2000), "Selfadapting vertices for mask-layout synthesis," in Proc. Model. Simul. Microsyst. Conf., M. Laudon and B. Romanowicz, Eds., San Diego, CA, Mar. pp. 83–86. http://www.design.caltech.edu/Research/Publicati ons/99h.pdf
[11] G. W. Milligan and M. C. Cooper. An examination of procedures for determining the number of clusters in a data set. Psychometrica, 50:159–179, 1985.
[12] Pal, N.R. and Bezdek, J.C. (1995) "On Cluster Validity for the Fuzzy C-Means Model," IEEE Trans. Fuzzy Systems, vol. 3, no. 3, pp. 370- 379. DOI: 10.1109/91.413225, ieeexplore.ieee.org/iel4/91/9211/00413225.pdf?ar number=413225
[13] Dan Pelleg and Andrew Moore. X-means: Extending k-means with efficient estimation of the number of clusters. In Proceedings of the 17th International Conf. on Machine Learning, pages 727–734. Morgan Kaufmann, 2000.
[14] Sarkar, M., Yegnanarayana, B. and Khemani, D. (1997), "A clustering algorithm using an evolutionary programming-based approach," Pattern Recognit. Lett., vol. 18, no. 10, pp. 975–986. DOI: 10.1016/S0167-8655(97)00122-0, http://speech.iiit.ac.in/svlpubs/article/Sarkar19979 75.pdf
[15] H. Spath, Clustering Analysis Algorithms for Data Reduction and Classification of Objects, Ellis Horwood, Chichester, 1980.
[16] Swagatam Das, Ajith Abraham (2008) "Automatic Clustering Using An Improved Differential Evolution Algorithm", Ieee Transactions On Systems, Man, And Cybernetics—Part A: Systems And Humans, Vol. 38, No. 1, Pp218-237. DOI: 10.1109/TSMCA.2007.909595 , www.softcomputing.net/smca-paper1.pdf
[17] Robert Tibshirani, Guenther Walther, and Trevor Hastie. Estimating the number of clusters in a dataset via the Gap statistic. Journal of the Royal Statistical Society B, 63:411–423, 2001.

[18] Xu, L. (1996). “How Many Clusters: A YingYang Machine Based Theory for a Classical Open Problem in Pattern Recognition,” Proc. IEEE Int’l Conf. Neural Networks ICNN ’96, vol. 3, pp.1546-1551 DOI: 10.1109/ICNN.1996.549130,ieeexplore.ieee.org/iel3/3927/11368/00549130.pdf?arnuber=549130

[19] Xu, L. (1997) “Rival Penalized Competitive Learning, Finite Mixture, and Multisets Clustering,” Pattern Recognition Letters, vol. 18, nos. 11- 13, pp. 1167-1178. DOI:10.1109/IJCNN.1998.687259,www.cse.cuhk.edu.hk/~lxu/papers/confchapters/XURPCLijcnn9 8.Pdf