International Journal of Computer science engineering Techniques-– Volume 2 Issue 5, July - Aug 2017

RESEARCH ARTICLE OPEN ACCESS

pfk-means: A Parameter Free K-means Algorithm

Omar Kettani*,
*(Scientific Institute, Physics of the Earth Laboratory/Mohamed V- University, Rabat)

Abstract:

K-means clustering is widely used for its efficiency. However, this algorithm suffers from two major drawbacks: first, the user must specifies in advance the correct number of clusters, which is generally a difficult task; second, its final results depend on the initial starting points. The present paper intends to overcome these issues by proposing a parameter free algorithm based on k-means (called pfk-means). We evaluated its performance by applying on several standard datasets and compare with gmeans, a related well know automatic clustering method. Our performance studies have demonstrated that the proposed approach is effective in predicting the correct number of clusters and producing consistent clustering results.

Keywords — **Parameter free, automatic clustering, k-means, gmeans.**

INTRODUCTION

In Data Mining, clustering consists of grouping a given dataset into a predefined number of disjoint sets, called clusters, so that the elements in the same cluster are more similar to each other and more different from the elements in the other cluster. This optimization problem is known to be NP-hard, even when the clustering process deals with only two clusters (Aloise 1980). Therefore, many heuristics and approximation algorithms have been proposed, in order to find near optimal clustering solution in reasonable computational time. The most prominent clustering algorithm k-means is a greedy algorithm which has two stages: Initialization, in which we set the seed set of centroids, and an iterative stage, called Lloyd's algorithm (Lloyd., S. P.1982). Additionally, Lloyd's algorithm has two steps: The assignment step, in which each object is assigned to its closest centroid, and the centroid's update step. The main advantage of k-means is its fast convergence to a local minimum, but k-means has two major drawbacks: first, the user must specifies in advance the correct number of clusters, which is generally a difficult task; second, the algorithm is sensitive to the initial starting points.

In this paper, an alternative parameter free method for automatic clustering, called pfk-means, is proposed. Algorithm validation and comparative study with gmeans (Hamerly and Elkan 2003), a related well known algorithm, are conducted using several real-world and artificial clustering data sets from the UCI Machine Learning Repository (Asuncion and Newman 2007).

In the next section, some related work are briefly discussed. Then the proposed approach is described in Section 3. Section 4 presents applications results of this clustering method to different standard data sets and reports its performance. Finally, conclusion of the paper is summarized in Section 5.

2. RELATED WORK

Despite the fact that obtaining an optimal number of clusters k for a given data set is an NP-hard problem (Spath 1980), several method have been developed to find k automatically.

Pelleg and Moore (2000) introduced the X-means algorithm, which proceed by learning k with k-means using the Bayesian Information Criterion (BIC) to score each model, and chooses the model with the highest BIC score. However, this method tends to overfit when it deals with data that arise from non-spherical clusters. Tibshirani et al. (2001) proposed the Gap statistic, which compares the likelihood of a learned model with the distribution of the likelihood of models trained on data drawn from a null distribution. This

method is suitable for finding a small number of clusters, but has difficulty when k increases. Cheung (2005) studied a rival penalized competitive learning algorithm, and Xu (1997, 1996) has demonstrated a very good result in finding the cluster number. Lee and Antonsson (2000) used an evolutionary method to dynamically cluster a data set. Sarkar,et al,. (1997) and Fogel, Owens, and Walsh (1966) are proposed an approach to dynamically cluster a data set using evolutionary programming, where two fitness functions are simultaneously optimized: one gives the optimal number of clusters, whereas the other leads to a proper identification of each cluster's centroid. Recently Swagatam Das and Ajith Abraham (2008) proposed an Automatic Clustering using Differential Evolution (ACDE) algorithm by introducing a new chromosome representation. Hamerly and Elkan (2003) proposed the gmeans algorithm, based on K-means algorithm, which uses projection and a statistical test for the hypothesis that the data in a cluster come from a Gaussian distribution. This algorithm works correctly if clusters are well-separated, and fails when clusters overlap and look non-Gaussian. In our experiments, gmeans tends to overestimate the number of clusters, as reported in section 4.

The majority of these methods to determine the best number of clusters may not work very well in practice. In the present work, an alternative approach is proposed, attempting to overcome these issues.

3. Proposed approach

The proposed algorithm starts by setting kmax=floor((n) $^{1/2}$), where n is the number of objects in the given data set. This choice is motivated by the fact that the number of clusters lies in the range from 2 to $(n)^{1/2}$, as reported by Pal and Bezdek (1995).

Then it applies a deterministic initialization procedure proposed by Kettani et al. (2013) (called KMNN) by splitting the entire dataset into two clusters. K-means algorithm is then applied with these two initial centroids. Again, the largest cluster is then split into two clusters by KMNN. This process is repeated until k=kmax, and at each iteration, the maximum of CH cluster validity index (Calinski and Harabasz 1974) of the current partition is stored. We used this index because it is relatively inexpensive to compute, and it generally outperforms other cluster validity indices as reported by Milligan and Cooper (1985). Finally, the algorithm outputs the optimal k and partition corresponding to the maximum value of CH stored so far. This algorithm is outlined in the pseudo-code below:

Algorithm pfk-means

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

Output: k mutually disjoint clusters $C_1 ,..., C_k$ 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 conducted using different data sets from the UCI Machine Learning Repository [10]. We evaluated its performance by applying on several benchmark datasets and compare with gmeans (Hamerly and Elkan 2003).

Silhouette index (Kaufman and Rousseeuw 1995) which measures the cohesion based on the distance between all the points in the same cluster and the separation based on the nearest neighbor distance, was used in these experiments in order to evaluate clustering accuracy. (bigger average silhouette value indicates a higher clustering accuracy). Silhouette index is based on distances between observations in the same cluster and in different clusters. Given observation i , let a_i be the average distance from point i to all other points in same cluster and d □ i, j □ represents the average distance from point i to all points in any other cluster j . Finally, let b_i denotes the minimum of these average distances d □ i, j □ . The silhouette width for the i-th observation is:

silh(i) □ (b_i □ a_i)/max□ a_i , b_i □

The average silhouette width can be find by averaging silh(i) over all observations:

$$silh \square - \square \frac{1}{n} \sum_{i \square 1}^{n} silh(i)$$

The silhouette width silh(i) ranges from -1 to 1. If an observation has a value close to 1, then the data point is closer to its own cluster than a neighboring one. If it has a silhouette width close to -1, then it is not very well clustered. A silhouette width close to zero indicates that the observation could just belong to current cluster or one that is near to it. Kaufman and Rousseeuw use the average silhouette width to estimate the number of clusters in a data set by using the partition with two or more clusters that yields the largest average silhouette width.

Experimental results are reported in table 1 and figure 1, and some clustering results are depicted in figure 2 to 7.

TABLE 1:Experimental results of pfk-means and gmeans application on different 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**

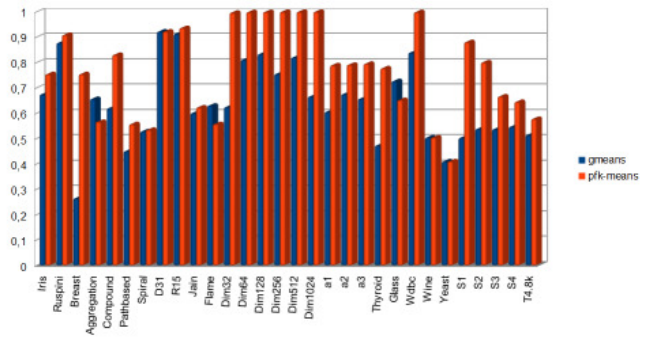

Fig 1: Chart of mean Silhouette index for both gmeans and pfk-means applied on different datasets.

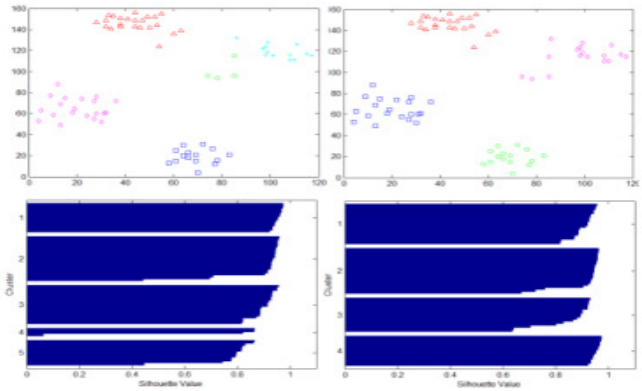

Fig 2: Clustering results of Ruspini dataset using gmeans (on left) and pfk-means (on right)

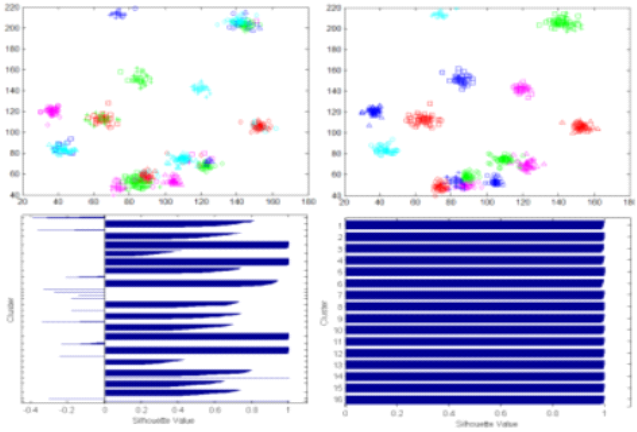

Fig 3: Clustering results of Dim32 dataset using gmeans (on left) and pfk-means (on right)

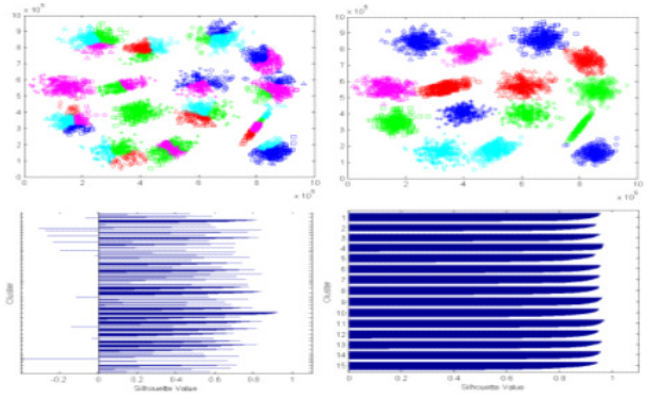

Fig 4: Clustering results of S1 dataset using gmeans (on left) and pfk-means (on right)

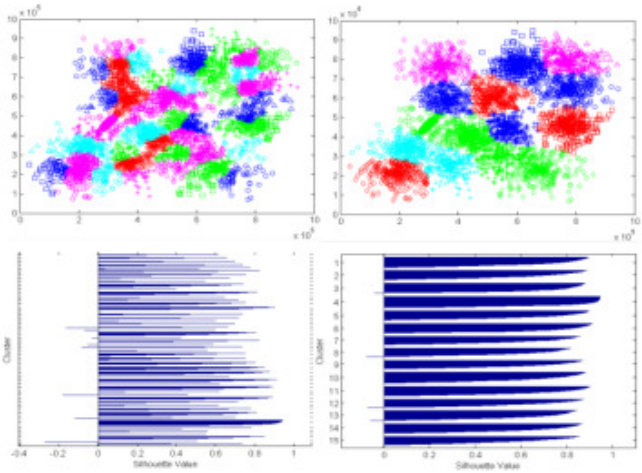

Fig 5: Clustering results of S2 dataset using gmeans (on left) and pfk-means (on right)

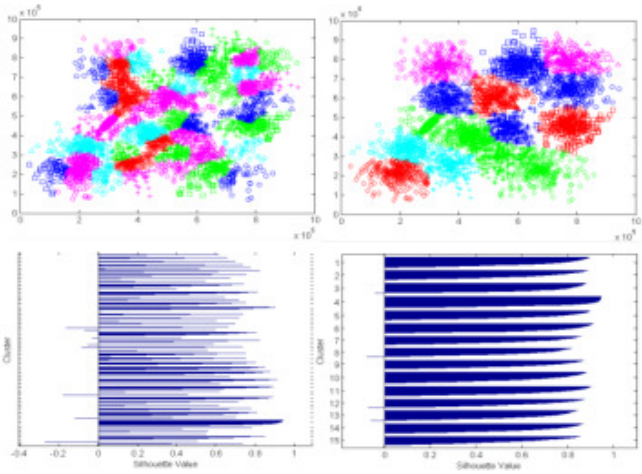

Fig 6: Clustering results of S3 dataset using gmeans (on left) and pfk-means (on right)

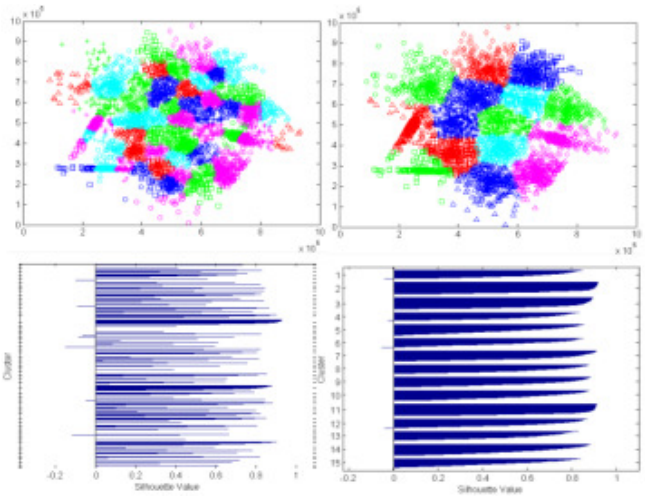

Fig 7: Clustering results of S4 dataset using gmeans (on left) and pfk-means (on right)

5 Conclusion

In this paper, a parameter free k-means algorithm was suggested. We evaluated its performance by applying on several standard datasets and compare with gmeans. Our experimental study have demonstrated that it is effective in producing consistent clustering results and have found the correct number of clusters with a successful rate of 63.33%.

In future work, it will be of interest to find a tighter upper bound on the number of clusters, instead of $n^{1/2}$, in order to reduce the number of computations steps of the proposed approach. Another possible enhancement 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 sum-of-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/MLRepository.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), "Self-adapting 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/Publications/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?arnumber=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/Sarkar199797 5.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 Ying-Yang 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?arnumber=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/XURPCLijcnn98.Pdf