REFINEMENT OF CLUSTERS FROM K-MEANS WITH ANT COLONY OPTIMIZATION

[1]C.IMMACULATE MARY, [2]DR. S.V. KASMIR RAJA
[1]Lecturer(S.G.) in Computer Science ,Sri Sarada College, Salem-636016, Tamil Nadu, India.
[2]Dean (Research), S.R.M. University , Chennai, Tamil Nadu, India.
E-mail: cimmaculatemary@gmail.com, svkr@srmuniv.ac.in

ABSTRACT

Clustering is a distribution of data into groups of similar objects. In this paper, Ant Colony Optimization (ACO) is proposed to improve k-means clustering. Though the k-means is one of the best clustering algorithm, the quality is based on the starting condition and it may converge to local minima. And an important point is, so far, the researchers have not contributed to improve the cluster quality after grouping. Our proposed method has two phases: in the first step, the initial seeds for k-means clustering are selected based on statistical modes to converge to a "better" local minimum. And in the second step, we have proposed a novel method to improve the cluster quality by ant based refinement algorithm. The proposed algorithm is tested in medical domain and shows that refined initial starting points and post processing refinement of clusters indeed lead to improved solutions.

Keywords: *k-Means Clustering, Initial Centroid Selection, Ant Colony Optimization (ACO).*

1. INTRODUCTION

Clustering have been used in a number of applications such as engineering, biology, medicine and data mining [1,2]. Xu and Wunsch (2005) presented a brief survey on clustering algorithm [3]. One of the most widely used algorithms is k-means clustering [4]. It partitions the objects into clusters by minimizing the sum of the squared distances between the objects and the centroid of the clusters. The k-means clustering is simple but it has high time complexity, so it is not suitable for large data set [5-7]. Many algorithms have been proposed to accelerate the k-means. Bentley (1975) suggested kd-trees to accelerate the k-means. However, backtracking is required, a case in which the computation complexity is increased [8]. And the kd-trees are not efficient enough for higher dimensions. Furthermore, it is not guaranteed that an exact match of the nearest neighbour can be found unless some extra search is done as discussed in [9]. Elkan (2003) suggests the use of triangle inequality to accelerate the k-means [10]. Hjaltason and Samet (1999) suggested to use RTrees [11]. Nevertheless, R-Trees may not be appropriate for higher dimensional problems. In [12-14], the Partial Distance (PD) algorithm has been proposed. The algorithm allows early termination of the distance calculation by introducing a premature exit condition in the search process.

As seen in the literature, the researchers contributed only to accelerate the algorithm; there is no contribution in cluster refinement. In this paper, we proposed an Ant Colony Optimization (ACO) algorithm to improve the k-means. The proposed algorithm consists of two steps. In the first step, to avoid local minima, we presented a simple and efficient method to select initial centroids based on mode value of the data vector. And the k-means algorithm is applied to cluster the data vectors. Then in the second step, an ant colony optimization algorithm is applied to refine the cluster to improve the quality.

The paper is organized as follows: the following section presents the general k-means algorithm. Section 3 presents our proposed initial refinement procedure. Section 4 discusses the proposed cluster refinement algorithm with ant colony optimization. Section 5 presents the results and the work is concluded in section 6.

2. STANDARD K-MEANS ALGORITHM

One of the most popular clustering techniques is the k-means clustering algorithm. Starting from a random partitioning, the algorithm repeatedly (i) computes the current cluster centers (i.e. the average vector of each cluster in data space) and (ii) reassigns each data item to the cluster whose centre is closest to it. It terminates when no more reassignments take place. By this means, the intra-cluster variance, that is, the sum of squares of the

differences between data items and their associated cluster centers is locally minimized. k -means' strength is its runtime, which is linear in the number of data elements, and its ease of implementation. However, the algorithm tends to get stuck in suboptimal solutions (dependent on the initial partitioning and the data ordering) and it works well only for spherically shaped clusters. It requires the number of clusters to be provided or to be determined (semi-) automatically. In our experiments, we run k-means using the correct cluster number. The algorithm for the standard k-means clustering is given as follows:

a. Choose a number of clusters k
b. Initialize cluster centers $\mu_1,\ldots \mu_k$
    i. Could pick *k* data points and set cluster centers to these points
    ii. Or could randomly assign points to clusters and take means of clusters
c. For each data point, compute the cluster center it is closest to (using some distance measure) and assign the data point to this cluster.
d. Re-compute cluster centers (mean of data points in cluster)
e. Stop when there are no new re-assignments.

3. INITIAL REFINEMENT

The initial cluster centers are normally chosen either sequentially or randomly as given in the standard algorithm. The quality of the final clusters based on these initial seeds. It may lead to local minimum; this is the disadvantage in k-means clustering. To avoid this, in our proposed method, we are selecting the modes of the data vector as initial cluster centers. Based on the number of clusters, the modes are selected one after another. Initially the first mode value is selected as the center for the first cluster and the next highest frequently occurred value is (next mode value) assigned as the center for next cluster. With this modification, the k-means algorithm is tested with medical data, as Table 1 shows, the quality is not improved but the processing time is reduced. Thus the time complexity is reduced in k-means algorithm also the local minima can be avoided.

Table 1. Clustering results for Wisconsin Breast Cancer Dataset.

Methods	Quality Measures		Time Complexity (in secs)
	Entropy	F Measure	
K-means with Mode	0.2373	0.9599	0.516
K-means	0.2373	0.9599	0.953

4. ACO BASED CLUSTER REFINEMENT

Ant-based clustering and sorting was originally introduced for tasks in robotics by Deneubourg et al. [15]. Lumer and Faieta [16] modified the algorithm to be applicable to numerical data analysis, and it has subsequently been used for data-mining [17], graph-partitioning [18]-[20] and text-mining [21]-[23].

Such ant-based methods have shown their effectiveness and efficiency in some test cases [24]. However, the ant-based clustering approach is in general immature and leaves big space for improvements. With these considerations, however, the standard ant-based clustering performs well; the algorithm consists of lot of parameters like pheromone, agent memory, number of agents, number of iterations and cluster retrieval etc. For these parameters more assumptions have been made in the previous works. So far, ants are used to cluster the data points. Here, for the first time we have used ants to refine the clusters. The clusters from the above section are considered as input to this ACO based refinement step.

The basic reason for our refinement is, in any clustering algorithm the obtained clusters will never give us 100% quality. There will be some errors known as misclustering . That is, a data item can be wrongly clustered. These kinds of errors can be avoided by using our refinement algorithm.

In our proposed method, only one ant is used to refine the clusters. This ant is allowed to go for a random walk on the clusters. Whenever it crosses a cluster, it will pick an item from the cluster and drop it into another cluster while moving. The pick and drop probability is calculated as given:

$$\text{Picking up probability, } P_p = \left(\frac{k_1}{k_1 + f} \right)^2$$

$$\text{Dropping probability, } P_d = \left(\frac{f}{k_2 + f} \right)^2$$

Here, f is the entropy value of the clusters calculated before the item was picked and dropped, while k_1 and k_2 are threshold constants (picking-up threshold and dropping threshold, respectively). If the dropping probability is lower than the picking probability then the item is dropped into another cluster and the entropy value is calculated again. This random walk is repeated for N number of times. From the following section, it is shown that our refinement algorithm improves the cluster quality. The algorithm is given as:

a. Choose a number of clusters k
b. Initialize cluster centers $\mu_1,\ldots \mu_k$ based on **mode**
c. For each data point, compute the cluster center it is closest to (using some distance measure) and assign the data point to this cluster.
d. Re-compute cluster centers (mean of data points in cluster)
e. Stop when there are no new re-assignments.
f. Ant based refinement
    i. Input the clusters from improved k-means.
    ii. For i = 1 to N do
        a. Let the ant go for a random walk to pick an item
        b. Calculate the pick and drop probability
        c. Decide to drop the item.
        d. Re-calculate the entropy value to check whether the quality is improving or not.
    iii. Repeat

5. EXPERIMENTS & RESULTS

The quality of the clusters can be analyzed using two measures called entropy [25] and F-measure [26]. The definitions are given below.

Entropy – For each cluster, the class distribution of the data is calculated first, i.e., for cluster j we compute p_{ij}, the "probability" that a member of cluster j belongs to class i. Then using this class distribution, the entropy of each cluster j is calculated using the standard formula

$$E_j = -\sum_i p_{ij} \log(p_{ij})$$

where the sum is taken over all classes. The total entropy for a set of clusters is calculated as the sum of the entropies of each cluster weighted by the size of each cluster:

$$E_{CS} = \sum_{j=1}^{m} \frac{n_j * E_j}{n}$$

where n_j is the size of cluster j, m is the number of clusters, and n is the total number of data points.

F measure – it combines the precision and recall ideas from information retrieval [27]. We treat each cluster as if it were the result of a query and each class as if it were the desired set of data items for a query. We then calculate the recall and precision of that cluster for each given class. More specifically, for cluster j and class i

$$\text{Recall}(i, j) = n_{ij} / n_i$$
$$\text{Precision}(i, j) = n_{ij} / n_j$$

where n_{ij} is the number of members of class i in cluster j, n_j is the number of members of cluster j and ni is the number of members of class i.

The F measure of cluster j and class i is then given by

$$F(i, j) = (2 * \text{Recall}(i, j) * \text{Precision}(i, j)) / ((\text{Precision}(i, j) + \text{Recall}(i, j))$$

For an entire hierarchical clustering the F measure of any class is the maximum value it attains at any node in the tree and an overall value for the F measure is computed by taking the weighted average of all values for the F measure as given by the following.

$$F = \sum_i \frac{n_i}{n} \max\{F(i, j)\}$$

where the max is taken over all clusters at all levels, and n is the number of data items. Table 2 presents the results, shows that our proposed method outperforms the standard method.

Table 2. Cluster Quality Analysis

Wisconsin Breast Cancer Dataset

	K-Means	K-Means with Mode	K-Means with ACO
No. of Classes	2	2	2
No. of Clusters	2	2	2
Entropy	0.2373	0.2373	0.1502
F measure	0.9599	0.9599	0.9799

Dermatology Dataset

	K-Means	K-Means with Mode	K-Means with ACO
No. of Classes	6	6	6
No. of Clusters	6	6	6
Entropy	0.0868	0.0868	0.0103
F measure	0.8303	0.8303	0.8841

6. CONCLUSION

In this paper, we have proposed Ant Colony Optimization (ACO) algorithm to improve the cluster quality from k-means algorithm. At first, the initial cluster centers are selected based on statistical mode based calculation to converge to a

better local minimum. And in the second step, we have proposed a novel method to improve to cluster quality by ant based refinement algorithm. The proposed algorithm is tested in medical domain and the experimental results show that refined initial starting points and post processing refinement of clusters provides better results than the conventional algorithm.

REFERENCES

[1] Lv T., Huang S., Zhang X., and Wang Z, Combining Multiple Clustering Methods Based on Core Group. Proceedings of the Second International Conference on Semantics, Knowledge and Grid (SKG'06), pp: 29-29, 2006.

[2] Nock R., and Nielsen F., On Weighting Clustering. IEEE Transactions and Pattern Analysis and Machine Intelligence, 28(8): 1223-1235, 2006.

[3] Xu R., and Wunsch D., Survey of clustering algorithms. IEEE Trans. Neural Networks, 16 (3): 645-678, 2005.

[4] MacQueen J., Some methods for classification and analysis of multivariate observations. Proc. 5th Berkeley Symp. Math. Stat. and Prob, pp: 281-97, 1967.

[5] Kanungo T., Mount D.M., Netanyahu N., Piatko C., Silverman R., and Wu A.Y., An efficient k-means clustering algorithm: Analysis and implementation. IEEE Trans. Pattern Analysis and Machine Intelligence, 24 (7): 881-892, 2002.

[6] Pelleg D., and Moore A., Accelerating exact k-means algorithm with geometric reasoning. Proceedings of the fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, New York, pp. 727-734, 1999.

[7] Sproull R., Refinements to Nearest-Neighbor Searching in K-Dimensional Trees. Algorithmica, 6: 579-589, 1991.

[8] Bentley J., Multidimensional Binary Search Trees Used for Associative Searching. Commun. ACM, 18 (9): 509-517, 1975.

[9] Friedman J., Bentley J., and Finkel R., An Algorithm for Finding Best Matches in Logarithmic Expected Time. ACM Trans. Math. Soft. 3 (2): 209-226, 1977.

[10] Elkan, C., Using the Triangle Inequality to Accelerate k-Means. Proceedings of the Twentieth International Conference on Machine Learning (ICML-2003), pp. 609-616, 2003.

[11] Hjaltason, R. and Samet H., Distance Browsing in Spatial Databases. ACM Transactions on Database Systems, 24 (2): 26-42, 1999.

[12] Proietti, G. and Faloutsos C., Analysis of Range Queries and Self-spatial Join Queries on Real Region Datasets Stored using an R-tree. IEEE Transactions on Knowledge and Data Engineering, 5 (12): 751-762, 2000.

[13] Cheng D., Gersho B., Ramamurthi Y., and Shoham Y., 1984. Fast Search Algorithms for Vector Quantization and Pattern Recognition. Proceeding of the IEEE International Conference on Acoustics, Speech and Signal Processing, 1, pp:1-9, 1984.

[14] Bei C., and Gray, R., An Improvement of the Minimum Distortion Encoding Algorithm for Vector Quantization. IEEE Transactions on Communications, 33 (10): 1132-1133, 1985.

[15] Deneubourg J.L., Goss S., Franks, N. Sendova-Franks A., Detrain C., and Chétien L. The Dynamics of Collective Sorting: Robot-like Ants and Ant-like Robots, In Proceedings of the 1st International Conference on Simulation of Adaptive Behavior: From Animals to Animats,. MIT Press, Cambridge, MA, USA, 1:356-363, 1991.

[16] Lumer E., and Faieta B. Diversity and adaptation in populations of clustering ants. In Proceedings of the Third International Conference on Simulation of Adaptive Behaviour: From Animals to Animats. MIT Press, Cambridge, MA, 3:501–508, 1994.

[17] Lumer E., and Faieta B. Exploratory database analysis via self-organization, 1995.

[18] Kuntz P., and Snyers D. Emergent colonization and graph partitioning. In Proceedings of the Third International Conference on Simulation of Adaptive Behaviour: From Animals to Animats. MIT Press, Cambridge, MA, 3:494–500, 1994.

[19] Kuntz P., and Snyers D. New results on an ant-based heuristic for highlighting the organization of large graphs. In Proceedings of the 1999 Congress on Evolutionary Computation, IEEE Press, Piscataway, NJ, 1451–1458, 1999.

[20] Kuntz P., Snyers D., and Layzell P. A stochastic heuristic for visualizing graph clusters in a bi-dimensional space prior to partitioning. Journal of Heuristics, 5(3):327–351, 1998.

[21] Ramos V., and Merelo JJ. Self-organized stigmergic document maps: Environments as a mechanism for context learning. In Proceedings of the First Spanish Conference on Evolutionary and Bio-Inspired Algorithms, Centro Univ. M´erida, Spain, 284–293, 2002.

[22] Handl J., and Meyer B. Improved ant-based clustering and sorting in a document retrieval

interface. In Proceedings of the Seventh International Conference on Parallel Problem Solving from Nature, Springer-Verlag, Berlin, Germany, 2439:913–923, 2002.

[23] Hoe K., Lai W., and Tai T. Homogeneous ants for web document similarity modeling and categorization. In Proceedings of the Third International Workshop on Ant Algorithms, Springer-Verlag, Heidelberg, Germany, 2463:256–261, 2002.

[24] Handl J., Knowles J., and Dorigo M. "On the Performance of Ant-based Clustering", In Design and Application of Hybrid Intelligent Systems, Frontiers in Artificial Intelligence and Applications,. Amsterdam, the Netherlands: IOS Press, 104:204-213, 2003.

[25] Shannon CE., A mathematical theory of communication, Bell System Technical Journal, 27:379-423 and 623-656, July and October, 1948.

[26] Larsen B., and Aone C. Fast and Effective Text Mining Using Linear-time Document Clustering, KDD-99, San Diego, California, 1999.

[27] Kowalski G, Information Retrieval Systems – Theory and Implementation, Kluwer Academic Publishers, 1997.