American Journal of Applied Sciences 5 (9): 1247-1250, 2008
ISSN 1546-9239
© 2008 Science Publications

New Efficient Strategy to Accelerate k-Means Clustering Algorithm

Moh'd Belal Al- Zoubi, Amjad Hudaib, Ammar Huneiti and Bassam Hammo
Department of Computer Information Systems, University of Jordan,
Amman 11942, Jordan

Abstract: One of the most popular clustering techniques is the k-means clustering algorithm. However, the utilization of the k-means is severely limited by its high computational complexity. In this study, we propose a new strategy to accelerate the k-means clustering algorithm through the Partial Distance (PD) logic. The proposed strategy avoids many unnecessary distance calculations by applying efficient PD strategy. Experiments show the efficiency of the proposed strategy when applied to different data sets.

Key words: clustering, k-means algorithm, pattern recognition, partial distance

INTRODUCTION

Clustering techniques have become very popular in a number of areas, such as engineering, medicine, biology and data mining[1,2]. A good survey on clustering algorithms can be found in[3].

The k-means algorithm[4] is one of the most widely used clustering algorithms. The algorithm partitions the data points (objects) into C groups (clusters), so as to minimize the sum of the (squared) distances between the data points and the center (mean) of the clusters.

To apply the k-means algorithm, do the following:

- Choose C data points to initialize the clusters
- For each data point, find the nearest cluster center that is closest and assign that data point to the corresponding cluster
- Update the cluster centers in each cluster using the mean of the data points which are assigned to that cluster
- Repeat steps 2 and 3 until there are not more changes in the values of the means

In spite of its simplicity, the k-means algorithm involves a very large number of nearest neighbor queries. The high time complexity of the k-means algorithm makes it impractical for use in the case of having a large number of points in the data set. Reducing the large number of nearest neighbor queries in the algorithm can accelerate it. In addition, the number of distance calculations increases exponentially with the increase of the dimensionality of the data[5-7].

Many algorithms have been proposed to accelerate the k-means. In[5,6], the use of kd-trees[8] is suggested to accelerate the k-means. However, backtracking is required, a case in which the computation complexity is increased[7]. Kd-trees are not efficient for higher dimensions. Furthermore, it is not guaranteed that an exact match of the nearest neighbor can be found unless some extra search is done as discussed in[9].

Elkan[10] suggests the use of triangle inequality to accelerate the k-means. In[11-12], it is suggested to use R-Trees. Nevertheless, R-Trees may not be appropriate for higher dimensional problems.

In[13-15], 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

In this study, we propose a new algorithm to accelerate the k-means. The proposed algorithm avoids many unnecessary distance calculations by applying an efficient Partial Distance (PD) strategy.

Our approach proceeds by reducing the number of the nearest neighbor queries that the k-means algorithm executes. Thus speed is enhanced greatly. The time taken for a single iteration is greatly improved over the conventional k-means algorithms without degrading the clustering quality and until taking the same number of iterations for convergence as the conventional k-means. For simplicity, we report the results of running the k-means 20 iterations.

PARTIAL DISTANCE

The (conventional) Partial Distance (PD) algorithm[13-15] has been proposed to reduce the

Corresponding Author: Moh'd Belal Al- Zoubi, Computer Information Systems Dept., University of Jordan, Amman 11942, Jordan, Tel: +962-5355000

computation complexity of the exhaustive search. The PD algorithm allows early termination of the distance calculation between a data point (vector) and a cluster center by introducing a premature exit condition in the search process.

Let $C = \{c_i, i = 1,\ldots,N\}$ be a set of cluster centers of size *N*, where $(c_{ij}, j = 1,\ldots, K)$ is a K dimensional data point (vector). For a given data point $X = (x_j, j = 1,\ldots, K)$, it is required to find the cluster center with the minimum distance from the set C under the squared-error distance measure defined as follows:

$$d(X,c_i) = \sum_{j=1}^{K}(x_j - c_{ij})^2$$

The basic structure of the PD algorithm [13, 14] is illustrated below:

$d_{min} = \infty$

Loop A: For i = 1, …, N
d = 0

Loop B: For j = 1, …, K
d = d + $(x_j - c_{ij})^2$
if ($d > d_{min}$) Next i // (exit condition)
Next j
d_{min} = d
min = i
Next i

It can be observed that the partial distance search algorithm gains computation saving over the full search algorithm because of the provision for a premature exit from Loop B, on satisfying the condition $d>d_{min}$ (called the exit condition) before the completion of the distance computation $d(X, c_i)$.

The problem with the PD algorithm is that its efficiency is dependent on the current (initial) distance d_{min} found so far. The larger the distance is, the less useful this method becomes[15].

Different strategies have been suggested to obtain an efficient initial distance to the PD algorithm. Paliwal and Ramasubramanian[16] assigned probabilities to the cluster centers based on the number of data points matching a particular cluster center during the creation of the cluster centers. The cluster centers are then decreasingly ordered according to their probabilities. Given a data set, cluster centers that have large probabilities are tried first. When ordering is used with the PD algorithm, some improvement is gained. However, the problem with this strategy is that the authors assume that the data points follow the same match distribution as the cluster centers. This assumption is not always true, particularly when clustering image data. Moreover, there is no improvement when a large number of clusters is used, as reported in[17]. A similar approach has been reported in[18].

Recently, Yang and Wang[19] propose a partial search strategy which aims at finding the nearest cluster center in image data. The strategy is based on partitioning the space into partial regions, using Unbalanced-Binary-Tree (UBT). Although some speed improvement is achieved, the quality of the result is degraded.

THE PROPOSED ALGORITHM

In this study, we propose a new strategy to find efficient initial distances to the PD algorithm. Let x be any data point, let c be a cluster center and let c_{prev} be the previous location of the same cluster center (i.e., the cluster center in the previous iteration). Suppose that in the previous iteration we know that d_{prev} is the distance between x and c_{prev}, then we can use d_{prev} as an initial distance for the PD algorithm. In other words, if c_{prev} has moved a small distance, then d_{prev} is a good initial distance. Having these observations, the new proposed algorithm works as follows:

Step 0 assigns each data point, x, to its closest cluster center, c, using the PD algorithm. Use each resulting distance (d_{prev}) as an initial distance in the next step.

Step 1: $d_{min} = d_{prev}$ (from step 0)
Loop A: For i = 1, …, N
d = 0
Loop B: For j = 1, …, K
d = d + $(x_j - c_{ij})^2$
if ($d > d_{min}$) Next i // (exit condition)
Next j
d_{min} = d
min = i
Next i

It can be noticed that the new proposed strategy would gain more computation time saving than the conventional PD algorithm. This is because the initial distance, $d>d_{min}$ produced from the proposed strategy is very small, as shown in Step1 (first line) in the algorithm above.

Note that step 0 is applied only once to find previous distances for the next steps. If the PD

algorithm is used in step 0, then we still gain some CPU time saving.

RESULTS

In order to test the efficiency of our proposed strategy, two data sets have been tested. The first set is the Letter data set with 20,000 data points and 16 variables (dimensions) obtained from the UCI Repository of Machine Learning Databases[20]. The second set is the Speech set, which represents a data extracted from one minute of speech with 8 dimensions. The CPU time (Run time) is measured in seconds. The results are obtained after running the k-means algorithm 20 iterations in all cases.

Table 1 shows the performance of the proposed strategy (or the new algorithm) with a number of cluster centers (No. Clust) for the Letter data set. The Table shows that the new algorithm gave better results than both the exhaustive and PD algorithms in all cases. The time saving of the new algorithm exceed 40% compared to the exhaustive algorithm in cases where the number of clusters (No. Clust) was 76, as shown in Table 1.

Table 2 shows the performance of the new algorithm with different dimensions (D) for the Letter data set. The Table shows that the new algorithm gave better results than the ones given by both exhaustive and PD algorithms in all cases.

Table 1: Run time for exhaustive, PD and the new algorithm for the *Letter* data set with different number of clusters

No. Clust	EX	PD	New
26	108	90	78
52	217	174	162
76	403	288	235
104	427	345	313
130	534	414	384
156	641	501	454
182	748	595	532
208	855	671	609

Table 2: Run time for exhaustive, PD and the new algorithm for the *Letter* data set with different dimensions (D) and 26 clusters

D	EX	PD	New
4	31	22	20
8	57	43	37
12	82	44	36
16	108	86	78

Table 3: Run time for exhaustive, PD and the new algorithm for the *Speech* data with 8 dimensions and a number of clusters

No Clust	Ex	PD	New
100	99	69	58
200	205	136	120
300	300	203	180
400	408	271	240
500	500	339	301

The time savings of the new algorithm exceed 56% compared to the exhaustive algorithm in cases where the number of dimensions (D) was 12, as shown in Table 2.

Table 3 shows the performance of the new algorithm for the Speech data for 10,000 data points, 8 dimensions and a number of cluster centers (No. Clust). The Table shows that the new proposed algorithm gave the best results in all cases. The timesaving of the new algorithm exceeds 40%, which is a good percentage compared to the percentage of the time savings of the exhaustive algorithm in all cases.

It can be noticed from the tables above that the new algorithm outperformed both the conventional PD and exhaustive algorithms.

CONCLUSION

In this study, we propose a new algorithm to accelerate the k-means clustering algorithm. The proposed algorithm is based on the Partial Distance (PD) logic. The proposed algorithm avoids many unnecessary distance calculations by applying an efficient partial distance strategy. Experimental results show that the proposed algorithm gave better results than both the exhaustive and the conventional PD algorithms when applied to real data sets. The time saving (over the exhaustive algorithm) exceeded 50% in some cases.

REFERENCES

1. Lv, T., S. Huang, X. Zhang and Z. Wang, 2006. Combining Multiple Clustering Methods Based on Core Group. Proceedings of the Second International Conference on Semantics, Knowledge and Grid (SKG'06), pp: 29-29.
2. Nock, R. and F. Nielsen, 2006. On Weighting Clustering. IEEE Transactions and Pattern Analysis and Machine Intelligence, 28 (8): 1223-1235.
3. Xu, R. and D. Wunsch II, 2005. Survey of clustering algorithms. IEEE Trans. Neural Networks, 16 (3): 645-678.
4. MacQueen, J., 1967. Some methods for classification and analysis of multivariate observations. Proc. 5th Berkeley Symp. Math. Stat. and Prob, pp: 281-97.
5. Kanungo, T., D.M. Mount, N. Netanyahu, C. Piatko, R. Silverman and A.Y. Wu, 2002. An efficient k-means clustering algorithm: Analysis and implementation. IEEE Trans. Pattern Analysis and Machine Intelligence, 24 (7): 881-892.

6. Pelleg, D. and A. Moore, 1999. 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.
7. Sproull, R., 1991. Refinements to Nearest-Neighbor Searching in K-Dimensional Trees. Algorithmica, 6: 579-589.
8. Bentley, J., 1975. Multidimensional Binary Search Trees Used for Associative Searching. Commun. ACM, 18 (9): 509-517.
9. Friedman, J., J. Bentley and R. Finkel, 1977. An Algorithm for Finding Best Matches in Logarithmic Expected Time. ACM Trans. Math. Soft. 3 (2): 209-226.
10. Elkan, C., 2003. Using the Triangle Inequality to Accelerate k-Means. Proceedings of the Twentieth International Conference on Machine Learning (ICML-2003), pp. 609-616.
11. Hjaltason, R. and H. Samet, 1999. Distance Browsing in Spatial Databases. ACM Transactions on Database Systems, 24 (2): 26-42.
13. Proietti, G. and C. Faloutsos, 2000. 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.
14. Cheng, D., B. Gersho, Y. Ramamurthi and Y. Shoham, 1984. Fast Search Algorithms for Vector Quantization and Pattern Recognition. Proceeding of the IEEE International Conference on Acoustics, Speech and Signal Processing, 1, pp: 9, 11, 1-9, 11, 4.
15. Bei, C. and Gray, R., 1985. An Improvement of the Minimum Distortion Encoding Algorithm for Vector Quantization. IEEE Transactions on Communications, 33 (10): 1132-1133.
16. Chen, S. and W. Hsieh, 1991. Fast Algorithm for VQ Codebook Design. IEE Proc., 138 (5): 357-362.
17. Paliwal, K and V. Ramasubramanian, 1989. Effect of Ordering the Codebook on the Efficiency of the Partial Distance Search Algorithm for Vector Quantization. IEEE Trans. Commun., 37 (5): 538-540.
18. Hwang, W., B. Chen and S. Jeng, 1997. A fast Vector Quantization Encoding using Wavelet Transform. Pattern Recognition Letters, 18: 73-76.
19. Fizzore, L., O. Laface, P. Massafra and F. Ravera, 1993. Analysis and Improvement of the Partial Distance Search Algorithm. Int. Conf. Acoustics, Spach and Signal Processing (ICASSP-93), 2, pp: 315-318.
20. Yang, C.H. and S.J. Wang, 2005. Accelerating VQ-based Codeword Search on the Basis of Partial Search Strategy. Computer Standard and Interfaces, 28: 231-240.
21. Blake, C.L. and C.J. Merz, 1998. UCI Repository of Machine Learning Databases. http://www.ics.uci.edu/mlearn/MLRepository.html, University of California, Irvine, Department of Information and Computer Sciences.