International Journal of Current Engineering and Technology
E-ISSN 2277 – 4106, P-ISSN 2347 – 5161

Available at http://inpressco.com/category/ijcet

Research Article

Count based K-Means Clustering Algorithm

Wilson Joseph†*, W. Jeberson† and Klinsega Jeberson†

†Computer Science and Information Technology, SHIATS, Allahabad, India

Accepted 15 April 2015, Available online 25 April 2015, **Vol.5, No.2 (April 2015)**

Abstract

The k-means clustering algorithm is a premier algorithm for data clustering. However, one of its limitations is the need to specify the number of clusters, K, before the algorithm is implemented. In this paper, we present a novel technique which assists us in determining K while performing data clustering on one dimensional data using enhanced k-means algorithm. The technique is based upon a Count test, which is performed on the dataset. The elements which pass the test become the initial cluster heads for k-means clustering. The experimental results suggest that the proposed technique is efficient, produces better results than rule of thumb technique used for determining K. The technique also helps in addressing the problem of empty clusters.

Keywords: *Clustering algorithm, K-means algorithm, Initial cluster heads, Count Test, Rule of thumb, Cluster Analysis.*

1. Introduction

Clustering is an attempt to group similar objects in same groups. This is done such that patters in the same cluster are similar, and patterns belonging to two different clusters are different. From the machine learning perspective, Clustering can be viewed as unsupervised learning of concepts. Unsupervised machine learning means that clustering does not depend on predefined classes and training examples while classifying the data objects (Zhang Chen and Xia Shixiong, 2009), (Yedla *et al*, 2010). Clustering algorithms are mainly divided into two categories: Hierarchical algorithms and Partition algorithms. A hierarchical clustering algorithm divides the given data set into smaller subsets in hierarchical fashion. A partition clustering algorithm partition the data set into desired number of sets in a single step. Numerous methods have been proposed to solve clustering problem (Fahim A. M., *et al*, 2006), (S. Revathi and Dr. T. Nalini, 2013). The most popular technique for clustering is K-means developed by Mac Queen in 1967. The simplicity of K-means made this algorithm to be used in many different domains. K-means is a partition clustering technique that separates data into k mutually excessive groups. By iterative such partitioning, K-means minimizes the sum of distance from each data to its clusters. K-means technique is very famous because of its ability to cluster huge data, and also outliers, quickly and efficiently (Kohei Arai and Ali Ridho Barakba, 2007).

*Corresponding author: *Corresponding author: **Wilson Joseph**

K-means is simple and can be used for a wide range of data types; it is quite sensitive to initial positions of cluster heads. The final cluster heads may not be optimal ones as the algorithm can converge to local optimal solutions. An empty cluster can be obtained if no points are allocated to the cluster during the assignment step. Therefore, it is quite necessary for K-means to have good initial cluster heads (S. Deelers, and S. Auwatanamongkol, 2007). Some algorithms for initializing cluster heads for K-means have been proposed in the past (Likas, *et al*, 2003), (S. S. Khan and A. Ahmad, 2004), (K. A. Abdul Nazeer and M. P. Sebastian, 2009).

But what remains a major challenge in cluster analysis is the estimation of optimal number of K. In this paper we propose the Count technique for estimating the number of clusters and their respective cluster heads. We have modified an enhanced K-means Algorithm (G. Sathiya and P. Kavitha, 2014) by applying our technique in it to predict the number of K. We also have compared our technique with Rule of Thumb technique and found it works better.

The rest of this paper is organized as follows. Section II reviews Rule of Thumb technique. In Section III, we review the Enhanced K-means Algorithm. In Section IV the proposed Count technique is discussed. In Section V we experimentally demonstrate the performance of the proposed method, followed by conclusion and future scope in Section VI.

2. Rule of thumb technique

It is simple and one of the most classical techniques for prediction of K. This technique can be applied to any

type of data set. In this technique we calculate K based upon the number of elements given (Trupti M. Kodinariya and Dr. Prashant R. Makwana, 2013).

$$K \approx \sqrt{n/2}$$

Where n is the total number of elements in the data set and K is the number of Clusters.

3. Enhanced K-means technique

Enhanced k-means technique was given by Kavitha *et al*, 2014. It's a modification of k-means clustering for better initial cluster heads.
The pseudo code for this technique is.

Input:
$R = \{r1, r2, \ldots rn\}$ // set of n elements
Output:
Set of k clusters.

Working:
Part 1: Find the initial heads of the clusters by applying Algorithm 1.

Part 2: Move each element to its appropriate cluster by using Algorithm 2.

In the first part, data set is checked, if it has any negative values in it or not. If the data set has negative value attributes, then we transform all the elements in the data set to the positive space by subtracting each element attribute with the minimum attribute value in the given data set. If data set contains only positive value attributes then the transformation is not needed. The two part process of enhanced method is described below as Algorithm 1 and Algorithm 2.

In the following step, the distance from origin for each element was calculated. Then, the original elements are sorted in accordance with the sorted distances. After sorting, we partition the sorted elements into k equal sets. In each set take the middle point as the initial head. The chosen initial heads lead to better clustering results. Next, for each element, the distance is calculated from all the initial heads. The next stage is an iterative process which makes use of a heuristic approach to reduce computational time.

Algorithm 1: Finding the initial heads

Input:
$R = \{r1, r2, \ldots rn\}$ // set of n elements

Output:
A set of k clusters.

Steps:

- If the dataset R has both positive and negative attribute values go to step 2, otherwise go to step 4.
- Pick the minimum attribute value in the given data set R.
- For each element attribute, calculate the difference with the minimum attribute value.
- For each element calculate the distance from origin.
- Sort the distances obtained in step 4. Sort the elements in accordance with the distances.
- Partition the sorted elements into k equal sets.
- In each set, take the middle point as the initial head for that set.
- Compute the distance between each element $ri(1 <= i <= n)$ to all the initial heads $hj\,(1 <= j <= k).$
- Repeat

In the next phase, the elements in the data set are assigned to the clusters having the closest head value. ClusterId of an element denotes the cluster to which it belongs. NearestDist of an element denotes the present nearest distance from closest head. Next, we have to recalculate the head value, for each cluster.

Algorithm 2: Moving elements to clusters

Input:
$R = \{r1, r2, \ldots rn\}$ // set of n elements.
hj // Initial head value.

Output: A set of k clusters.

Steps:

- For each element ri, find the closest head value hj and assign ri to cluster j.
- Set ClusterId[i] = j // j: Id of the closest cluster. Set NearestDist[i] = d(ri, hj)
- For each cluster $j\,(1 <= j <= k)$, recalculate the head values.
- For each element ri,

Compute its distance from the head of the present nearest cluster.

If this distance is less than or equal to the present nearest distance, the data point stays in the same cluster. Else

For every head $hj(1 <= j <= k)$ compute the distance $d(ri, hj).$

End for;

Until the convergence criteria is met.

Then calculate its distance from the head of the present nearest cluster for each element. If this distance is less than or equal to the present nearest distance, the data

point stays in the same cluster. Otherwise calculate the distance for every cluster head.

Count based technique

The idea of count based technique is influenced by the enhanced k-means technique described in the previous section. The technique is a three part process.

In the first part, dataset is sorted using heap sort.

Input:
$R = \{r1, r2, \ldots rn\}$ // set of n elements
Output:
Sorted data set

In the second part, the occurrence of an element in the dataset is observed and is stored in count. And based on the count value; an array of elements is made whose count value surpasses the threshold.

The elements which surpass the threshold in their occurrence become the initial cluster heads. As threshold parameter, we have HighVal and LowVal.
The assumption is, if the range of a data set is greater than the number of data points then the data set is well spread, in this case HighVal is used as the threshold value. Otherwise data set is less spread, and LowVal is used as the threshold value.

Minfrq and Maxfrq are two other parameters used in the technique.

Minfrq stores the least occurrence value corresponding to an element which occurred least in the data set and Maxfrq stores the maximum occurrence value corresponding to an element which occurred most in the data set. Using minfrq and maxfrq, threshold value is calculated.

To calculate HighVal:

```
if ((maxfrq - minfrq) < 10))
alpha = (maxfrq + minfrq)/2;
else
alpha = minfrq + ((maxfrq - minfrq)/3);
```

And to calculate LowVal:

```
if ((maxfrq - minfrq) < 10))
beta = (maxfrq + minfrq)/2;
else
beta = minfrq + ((maxfrq - minfrq)/2.5);
```

Input:
$S = \{s1, s2, \ldots. sn\}$ //set of n sorted elements

Output:
$P = \{p1, p2, \ldots pk\}$ //set of k cluster heads

Steps:

- If there are no more unique elements left. Go to step 3. Otherwise go to step 2
- Check if the count for the element crosses the threshold value.
    - If yes, then move the element to data set P and go to step 1 and check for the next unique element.
    - If no, proceed to step 1 and check for the next unique element.
- Initialize the elements in P to be cluster heads.
- Calculate the difference between each element $ri(1 <= i <= n)$ to all the initial heads $pj(1 <= j <= k)$.

In the third part, elements are moved to appropriate clusters based on the distance between initial cluster heads and the element.

Input:
$P = \{p1, p2, \ldots, pk\}\}$ //set of k initial cluster heads
$R = \{r1, r2, \ldots rn\}$ //set of n elements

Output:
A set of k clusters.

Steps:

Repeat

- Assign each element ri to cluster with the closest cluster head value.
- Calculate the new average value for each cluster

Until

- Convergence criteria are met.

So when the third part is over with, we get k set of clusters.

Results

We have implemented the Rule of thumb and Count Techniques in Java language and tested them with three kinds of synthetic data sets, HighExtremes, EquiThrough and Random. HighExtremes data set has elements with either high counts or vice versa. EquiThrough data set has all the elements with almost similar counts. Random data set has elements with random counts. We have used data set of three different sizes, 100, 500 and 1000 of three kinds HighExtremes, EquiThrough and Random.

In figure 1, the number of Cluster heads formed in both the techniques is compared. In figure 2, the Cluster head average of all the clusters formed in both the techniques is compared. Both the figures are based on the data in Table1.
Table1 shows that, count technique reduces the average value of each cluster, which means elements of each cluster, are more tightly bound to one other and thus resultant clusters are denser (Ravindra Jain, 2012).

Table 1: Comparison between Count based and Rule of thumb technique

Clustering Technique	Data set Size	Data Set Kind	NI: No. of Iterations	NC: No. of Clusters	SC: Sum of cluster heads	Ratio of SC/NC
Rule of Thumb	100	HighExtremes	5	7	229.4	32.8
		EquiThrough	6	7	253.3	36.2
		Random	7	7	247.5	35
Count	100	HighExtremes	9	5	187.8	37.6
		EquiThrough	10	15	412.6	27.5
		Random	4	5	210.9	42.2
Rule of Thumb	500	HighExtremes	6	15	650	32.8
		EquiThrough	4	15	543.2	36.2
		Random	4	15	606.5	40.4
Count	500	HighExtremes	7	5	180.2	36
		EquiThrough	6	16	412.8	25.8
		Random	6	19	587.2	30.9
Rule of Thumb	1000	HighExtremes	5	22	982.9	44.7
		EquiThrough	4	22	881.5	40
		Random	4	22	885	40.2
Count	1000	HighExtremes	9	5	180.1	36
		EquiThrough	10	28	857.2	30.6
		Random	5	28	824.2	29.4

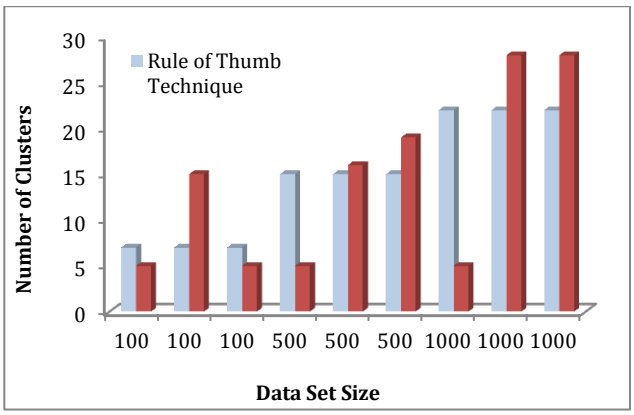

Figure 1: Comparison of Number of Cluster Heads

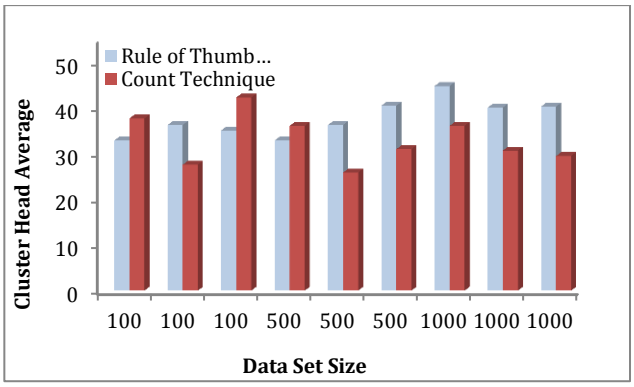

Figure 2: Comparison of Cluster Head Average Value

Conclusions

This paper presents a novel technique for clustering one dimensional data set. We have compared count technique with an older technique. From the experiments, it has been observed that new technique does give good results in a number of cases and also there weren't any cases, in which empty clusters were formed. The proposed technique is also found to be more precise and accurate in cluster formation

Future study will include modification of the present technique to be used for higher dimensions.

References

Zhang Chen and Xia Shixiong, (2009),K-means Clustering Algorithm with improved Initial Center, Second International Workshop on Knowledge Discovery and Data Mining (WKDD), pp: 790-792.

Madhu Yedla , Srinivasa Rao Pathakota, T M Srinivasa, (2010), Enhancing K-means Clustering Algorithm with Improved Initial Center, IJCSIT, Vol. 1 (2) , 2010, 121-125

Fahim A. M., Salem A. M., F.A. Torkey and M.A. Ramadan, (2006, An Efficient enhanced k-means clustering algorithm, Journal of Zhejiang University, 10(7): 6261633.

S. Revathi and Dr. T. Nalini, (2013), Performance Comparison of Various Clustering Algorithm, IJARCSSE, Volume 3, Issue 2, February 2013

Kohei Arai, Ali Ridho Barakba, (2007), Hierarchical K-means: an algorithm for centroids initialization for K-means, Rep. Fac. Sci. Engrg. , Saga Univ. 36-1 (2007),25-31

S. Deelers, and S. Auwatanamongkol, (2007), Enhancing K-Means Algorithm with Initial Cluster Centers Derived from Data Partitioning along the Data Axis with the Highest Variance, World Academy of Science, Engineering and Technology Vol:1 2007-11-27

A. Likas, N. Vlassis and J.J. Verbeek, (2003), The Global k-means Clustering algorithm, Pattern Recognition , Volume 36, Issue 2, 2003, pp. 451-461.

S. S. Khan and A. Ahmad, (2004), Cluster Center Initialization for K-mean Clustering, Pattern Recognition Letters, Volume 25, Issue 11, 2004, pp. 1293-1302

K. A. Abdul Nazeer, M. P. Sebastian, (2009), Improving the Accuracy and Efficiency of the k-means Clustering Algorithm. Proceedings of the World Congress on Engineering 2009, Vol I WCE 2009, July 1 - 3, 2009, London, U.K.

G. Sathiya and P. Kavitha, (2014), An Efficient Enhanced K-Means Approach with Improved Initial Cluster Centers, Middle-East Journal of Scientific Research 20 (4): 485-491

Trupti M. Kodinariya and Dr. Prashant R. Makwana, (2013), Review on determining number of Cluster in K-Means Clustering, IJARCSMS, Volume 1, Issue 6

Ravindra Jain, (2012), A Hybrid Clustering Algorithm for Data Mining, CS & IT-CSCP, CS & IT 05, pp. 387–393, 2012

Jiawei Han, Micheline Kamber and Jian Pei, (2011), Data Mining Concepts and Techniques, Morgan Kaufmann Publishers, First Indian Edition

Margaret H. Dunham, (2006), Data Mining-Introductory and Advanced Concepts, Pearson Education.

Elmasri, Navathe and Somayajulu, Gupta, (2006), Fundamentals of Database Systems, Pearson Education, First edition.