Research Journal of Applied Sciences, Engineering and Technology 6(17): 3299-3303, 2013
ISSN: 2040-7459; e-ISSN: 2040-7467

Submitted: January 23, 2013 Accepted: February 25, 2013 Published: September 20, 2013

Standardization and Its Effects on *K*-Means Clustering Algorithm

Ismail Bin Mohamad and Dauda Usman
Department of Mathematical Sciences, Faculty of Science, Universiti Teknologi Malaysia, 81310, UTM Johor Bahru, Johor Darul Ta'azim, Malaysia

Abstract: Data clustering is an important data exploration technique with many applications in data mining. *K*-means is one of the most well known methods of data mining that partitions a dataset into groups of patterns, many methods have been proposed to improve the performance of the *K*-means algorithm. Standardization is the central preprocessing step in data mining, to standardize values of features or attributes from different dynamic range into a specific range. In this paper, we have analyzed the performances of the three standardization methods on conventional *K*-means algorithm. By comparing the results on infectious diseases datasets, it was found that the result obtained by the z-score standardization method is more effective and efficient than min-max and decimal scaling standardization methods.

Keywords: Clustering, decimal scaling, *k*-means, min-max, standardization, z-score

INTRODUCTION

One of the mosteasiest and generally utilized technique meant for creating groupings by optimizing qualifying criterion function, defined eitherglobally (totaldesign) or locally (on the subset from thedesigns), is the K-means technique (Vaishali and Rupa, 2011). *K*-means clustering is one of the older predictive n observations in d dimensional space (an integer d) is given and the problem is to determine a set of c points to minimize the mean squared distance from each data point to its nearest center with which each observation belongs. No exact polynomial-time algorithms are known for this problem. The problem can be set up as an integer programming problem but because solving integer programs with a large number of variables is time consuming, clusters are often computed using a fast, heuristic method that generally produces good (but not necessarily optimal) solutions (Jain *et al*., 1999). The *K*-means algorithm is one such method where clustering requires less effort. In the beginning, number of cluster c is determined and the centre of these clusters is assumed. Any random objects as the initial centroids can be taken or the first k objects in sequence can also serve as the initial centroids. However, if there are some features, with a large size or great variability, these kind of features will strongly affect the clustering result. In this case, data standardization would be an important preprocessing task to scale or control the variability of the datasets.

The *K*-means algorithm will do the three steps below until convergence

Iterate until stable (= no object move group):

- Determine the centroid coordinate
- Determine the distance of each object to the centroids
- Group the object based on minimum distance

The aim of clustering would be to figure out commonalities and designs from the large data sets by splitting the data into groups. Since it is assumed that the data sets are unlabeled, clustering is frequently regarded as the most valuable unsupervised learning problem (Cios *et al*., 2007).

A primary application of geometrical measures (distances) to features having large ranges will implicitly assign greater efforts in the metrics compared to the application with features having smaller ranges. Furthermore, the features need to be dimensionless since the numerical values of the ranges of dimensional features rely upon the units of measurements and, hence, a selection of the units of measurements may significantly alter the outcomes of clustering. Therefore, one should not employ distance measures like the Euclidean distance without having normalization of the data sets (Aksoy and Haralick, 2001; Larose, 2005).

Preprocessing Luai *et al*. (2006) is actually essential before using any data exploration algorithms to enhance the results' performance. Normalization of the dataset is among the preprocessing processes in data exploration, in which the attribute data are scaled tofall in a small specifiedrange. Normalization before clustering is specifically needed for distance metric,

Corresponding Author: Dauda Usman, Department of Mathematical Sciences, Faculty of Science, Universiti Teknologi Malaysia, 81310, UTM Johor Bahru, Johor Darul Ta'azim, Malaysia

like the Euclidian distance that are sensitive to variations within the magnitude or scales from the attributes. In actual applications, due to the variations in selection of the attribute's value, one attribute might overpower another one. Normalization prevents outweighing features having a large number over features with smaller numbers. The aim would be to equalize the dimensions or magnitude and also the variability of those features.

Data preprocessing techniques (Vaishali and Rupa, 2011) are applied to a raw data to make the data clean, noise free and consistent. Data Normalization standardize the raw data by converting them into specific range using a linear transformation which can generate good quality clusters and improve the accuracy of clustering algorithms.

There is no universally defined rule for normalizing the datasets and thus the choice of a particular normalization rule is largely left to the discretion of the user (Karthikeyani and Thangavel, 2009). Thus the data normalization methods includes Z-score, Min-Max and Decimal scaling. In the Z-score the values for an attribute *X* are standardized based on the mean and standard deviation of *X*, this method is useful when the actual minimum and maximum of attribute *X* are unknown. Decimal scaling standardized by moving the decimal point of values of attribute *X*, the number of decimal points moved depends on the maximum absolute value of *X*. Min-Max transforms the data set between 0.0 and 1.0 by subtracting the minimum value from each value divided by the range of values for each individual value.

MATERIALS AND METHODS

Let, Y = {X_1, X_2, …, X_n} denote the *d*-dimensional raw data set.
Then the data matrix is an *n*×*d* matrix given by:

$$X_1, X_2, ..., X_n = \begin{pmatrix} a_{11} & \cdots & a_{1d} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nd} \end{pmatrix}. \quad (1)$$

Z-score: The *Z*-score is a form of standardization used for transforming normal variants to standard score form. Given a set of raw data Y, the *Z*-score standardization formula is defined as:

$$x_{ij} = Z\left(x_{ij}\right) = \frac{x_{ij} - \bar{x}_j}{\sigma_j} \quad (2)$$

where, $\bar{x}_j$ and σ_j are the sample mean and standard deviation of the *jth* attribute, respectively. The transformed variable will have a mean of 0 and a variance of 1. The location and scale information of the original variable has been lost (Jain and Dubes, 1988). One important restriction of the *Z*-score standardization is that it must be applied in global standardization and not in within-cluster standardization (Milligan and Cooper, 1988).

Min-max: Min-Max normalization is the process of taking data measured in its engineering units and transforming it to a value between 0.0 and 1.0. Where by the lowest (min) value is set to 0.0 and the highest (max) value is set to 1.0. This provides an easy way to compare values that are measured using different scales or different units of measure. The normalized value is defined as:

$$MM(X_{ij}) = \frac{X_{ij} - X_{\min}}{X_{\max} - X_{\min}} \quad (3)$$

Decimal scaling:
Normalization by decimal scaling: Normalizes by moving the decimal point of values of feature *X*. The number of decimal points moved depends on the maximum absolute value of *X*. A modified value *DS (X)* corresponding to *X* is obtained using:

$$DS(X_{ij}) = \frac{X_{ij}}{10^c} \quad (4)$$

where, *c* is the smallest integer such that $max[|DS(X_{ij})|]<1$

***K*-means clustering:** Given a set of observations (x_1, x_2, …, x_n), where each observation is a *d*-dimensional real vector, *K*-means clustering aims to partition the *n* observations into *k* sets (*k*≤*n*) S = {S_1, S_2, …, S_k} so as to minimize the Within-Cluster Sum of Squares (WCSS):

$$\underset{\mathbf{S}}{\arg\min} \sum_{i=1}^{k} \sum_{\mathbf{x}_j \in S_i} \left\| \mathbf{x}_j - \boldsymbol{\mu}_i \right\|^2 \quad (5)$$

where, μ_i is the mean of points in S_i:

RESULTS AND DISCUSSION

In this section, details of the overall results have been discussed. A complete program using MATLAB has been developed to find the optimal solution. Few experiments have been conducted on three standardization procedures and compare their performances on *K*-means clustering algorithm with infectious diseases dataset having 15 data objects and 8 attributes as shown in Table 1. The eight datasets, Malaria dataset, Typhoid fever dataset, Cholera dataset, Measles dataset, Chickenpox dataset, Tuberculosis dataset, Tetanus dataset and Leprosy dataset for X1 to X8 respectively are used to test the performances of the three standardization methods on K-means clustering technique. The sum of squares error representing

Table 1: The original datasets with 15 data objects and 8 attributes

	X1	X2	X3	X4	X5	X6	X7	X8
Day 1	7	1	1	1	1	2	10	3
Day 2	8	2	1	2	1	2	1	3
Day 3	9	2	1	1	1	2	1	1
Day 4	10	4	2	1	1	2	1	2
Day 5	1	5	1	1	1	2	1	3
Day 6	2	5	4	4	5	7	10	3
Day 7	1	5	1	1	1	2	1	3
Day 8	2	5	4	4	5	4	3	3
Day 9	3	3	1	1	1	2	2	3
Day 10	4	6	8	8	1	3	4	3
Day 11	3	3	1	1	1	2	2	3
Day 12	4	6	8	8	1	3	4	3
Day 13	5	4	1	1	3	2	1	3
Day 14	6	8	10	10	8	7	10	9
Day 15	3	3	1	1	1	2	2	3

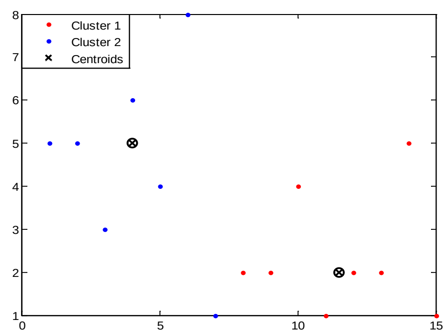

Fig. 1: Conventional K-means algorithm

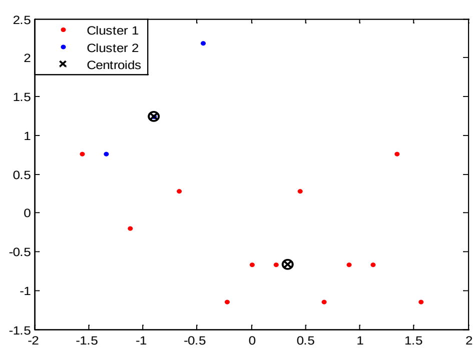

Fig. 2: K-means algorithm with a Z-score standardized data set

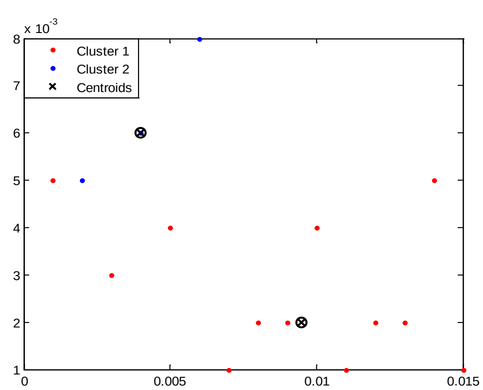

Fig. 3: K-means algorithm with the decimal scaling standardization data set

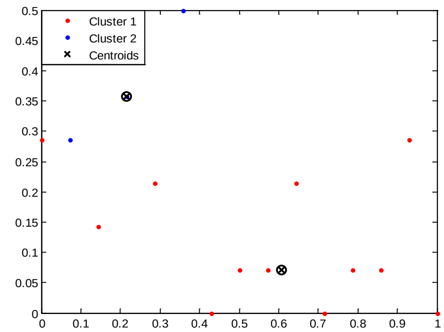

Fig. 4: K-means algorithm with min-max standardization data set

distances between data points and their cluster centers and the points attached to a cluster were used to measure the clustering quality among the three different standardization methods, the smaller the value of the sum of squares error the higher the accuracy, the better the result.

Figure 1 presents the result of the conventional *K*-means algorithm using the original dataset having 15 data objects and 8 attributes as shown in Table 1. Some points attached to cluster one and one point attached to cluster two are out of the cluster formation with the error sum of squares equal 141.00.

Z-score analysis: Figure 2 presents the result of the K-means algorithm using the rescale dataset with Z-score standardization method, having 15 data objects and 8 attributes as shown in Table 2. All the points attached to cluster one and cluster two are within the cluster formation with the error sum of squares equal 49.42

Decimal scaling analysis: Figure 3 presents the result of the K-means algorithm using the rescale dataset with the decimal scaling method of data standardization, having 15 data objects and 8 attributes as shown in Table 3. Some points attached to cluster one and one point attached to cluster two are out of the cluster formation with the error sum of squares equal 0.14 and converted to 140.00.

Min-max analysis: Figure 4 presents the result of the *K*-means algorithm using the rescale dataset with Min-Max data standardization method, having 15 data objects and 8 attributes as shown in Table 4. Some points attached to cluster one and one point attached to cluster two are out of the cluster formation with the error sum of squares equal 10.07

Table 5 shows the number of points that are out of cluster formations for both cluster 1 and cluster 2. The total error sum of squares for conventional *K*-means, *K*-means with z-score, *K*-means with decimal scaling and *K*-means with min-max datasets.

Table 2: The standardized datasets with 15 data objects and 8 attributes

	X1	X2	X3	X4	X5	X6	X7	X8
Day 1	-0.2236	-1.1442	-0.5192	-0.5192	-0.4913	-0.3705	1.8627	-0.0754
Day 2	0	-0.6674	-0.5192	-0.1652	-0.4913	-0.3705	-0.6519	-0.0754
Day 3	0.2236	-0.6674	-0.5192	-0.5192	-0.4913	-0.3705	-0.6519	-1.2070
Day 4	0.4472	0.2860	-0.1652	-0.5192	-0.4913	-0.3705	-0.6519	-0.6412
Day 5	-1.5652	0.7628	-0.5192	-0.5192	-0.4913	-0.3705	-0.6519	-0.0754
Day 6	-1.3416	0.7628	0.5428	0.5428	1.4739	2.4081	1.8627	-0.0754
Day 7	-0.2236	-1.1442	-0.5192	-0.5192	-0.4913	-0.3705	1.8627	-0.0754
Day 8	0	-0.6674	-0.5192	-0.1652	-0.4913	-0.3705	-0.6519	-0.0754
Day 9	0.2236	-0.6674	-0.5192	-0.5192	-0.4913	-0.3705	-0.6519	-1.2070
Day 10	0.4472	0.2860	-0.1652	-0.5192	-0.4913	-0.3705	-0.6519	-0.6412
Day 11	-1.1180	-0.1907	-0.5192	-0.5192	-0.4913	-0.3705	-0.3725	-0.0754
Day 12	-0.8944	1.2395	1.9587	1.9587	-0.4913	0.1852	0.1863	-0.0754
Day 13	-0.6708	0.2860	-0.5192	-0.5192	0.4913	-0.3705	-0.6519	-0.0754
Day 14	-0.4472	2.1930	2.6666	2.6666	2.9478	2.4081	1.8627	3.3193
Day 15	1.5652	-1.1442	-0.5192	-0.5192	-0.4913	-0.3705	-0.0931	-0.0754

Table 3: The standardized dataset with 15 data objects and 8 attributes

	X1	X2	X3	X4	X5	X6	X7	X8
Day 1	0.0070	0.0010	0.0010	0.0010	0.0010	0.0020	0.0100	0.0030
Day 2	0.0080	0.0020	0.0010	0.0020	0.0010	0.0020	0.0010	0.0030
Day 3	0.0090	0.0020	0.0010	0.0010	0.0010	0.0020	0.0010	0.0010
Day 4	0.0100	0.0040	0.0020	0.0010	0.0010	0.0020	0.0010	0.0020
Day 5	0.0010	0.0050	0.0010	0.0010	0.0010	0.0020	0.0010	0.0030
Day 6	0.0020	0.0050	0.0040	0.0040	0.0050	0.0070	0.0100	0.0030
Day 7	0.0070	0.0010	0.0010	0.0010	0.0010	0.0020	0.0100	0.0030
Day 8	0.0080	0.0020	0.0010	0.0020	0.0010	0.0020	0.0010	0.0030
Day 9	0.0090	0.0020	0.0010	0.0010	0.0010	0.0020	0.0010	0.0010
Day 10	0.0100	0.0040	0.0020	0.0010	0.0010	0.0020	0.0010	0.0020
Day 11	0.0030	0.0030	0.0010	0.0010	0.0010	0.0020	0.0020	0.0030
Day 12	0.0040	0.0060	0.0080	0.0080	0.0010	0.0030	0.0040	0.0030
Day 13	0.0050	0.0040	0.0010	0.0010	0.0030	0.0020	0.0010	0.0030
Day 14	0.0060	0.0080	0.0100	0.0100	0.0080	0.0070	0.0100	0.0090
Day 15	0.0150	0.0010	0.0010	0.0010	0.0010	0.0020	0.0030	0.0030

Table 4: The standardized dataset with 15 data objects and 8 attributes

	X1	X2	X3	X4	X5	X6	X7	X8
Day 1	0.4286	0	0	0	0	0.0714	0.6429	0.1429
Day 2	0.5000	0.0714	0	0.0714	0	0.0714	0	0.1429
Day 3	0.5714	0.0714	0	0	0	0.0714	0	0
Day 4	0.6429	0.2143	0.0714	0	0	0.0714	0	0.0714
Day 5	0	0.2857	0	0	0	0.0714	0	0.1429
Day 6	0.0714	0.2857	0.2143	0.2143	0.2857	0.4286	0.6429	0.1429
Day 7	0.4286	0	0	0	0	0.0714	0.6429	0.1429
Day 8	0.5000	0.0714	0	0.0714	0	0.0714	0	0.1429
Day 9	0.5714	0.0714	0	0	0	0.0714	0	0
Day 10	0.6429	0.2143	0.0714	0	0	0.0714	0	0.0714
Day 11	0.1429	0.1429	0	0	0	0.0714	0.0714	0.1429
Day 12	0.2143	0.3571	0.5000	0.5000	0	0.1429	0.2143	0.1429
Day 13	0.2857	0.2143	0	0	0.1429	0.0714	0	0.1429
Day 14	0.3571	0.5000	0.6429	0.6429	0.5000	0.4286	0.6429	0.5714
Day 15	1.0000	0	0	0	0	0.0714	0.1429	0.1429

Table 5: Summary of the results for cluster formations

	Cluster 1 points out	Cluster 2 points out	ESSs
Conventional *K*-means	2	2	159.00
K-means with Z-score	0	0	45.32
K-means with decimal scaling	3	1	130.00
K-means with Min-Max	4	1	09.21

CONCLUSION

A novel method of *K*-means clustering using standardization method is proposed to produce optimum quality clusters. Comprehensive experiments on infectious diseases datasets have been conducted to study the impact of standardization and to compare the effect of three different standardization procedures in conventional *K*-means clustering algorithm. It can be concluded that standardization before clustering algorithm leads to obtain a better quality, efficient and accurate cluster result. It is also important to select a specific standardization procedure, according to the nature of the datasets for the analysis. In this analysis we proposed Z-score as the most powerful method that will give more accurate and efficient result among the three methods in *K*-means clustering algorithm.

REFERENCES

Aksoy, S. and R.M. Haralick, 2001. Feature normalization and likelihood-based similarity measures for image retrieval. Pattern Recogn. Lett., 22: 563-582.

Cios, K.J., W. Pedrycz, R.W. Swiniarski and L.A. Kurgan, 2007. Data Mining: A Knowledge Discovery Approach. Springer, New York.

Jain, A. and R. Dubes, 1988. Algorithms for Clustering Data. Prentice Hall, NY.

Jain, A.R., M.N. Murthy and P.J. Flynn, 1999. Data clustering: A Review. ACM Comput. Surv., 31(3): 265-323.

Karthikeyani, V.N. and K. Thangavel, 2009. Impact of normalization in distributed K-means clustering. Int. J. Soft Comput., 4(4): 168-172.

Larose, D.T., 2005. Discovering Knowledge in Data: An Introduction to Data Mining. Wiley, Hoboken, NJ.

Luai, A.S., S. Zyad and K. Basel, 2006. Data mining: A preprocessing engine. J. Comput. Sci., 2(9): 735-739.

Milligan, G. and M. Cooper, 1988. A study of standardization of variables in cluster analysis. J. Classif., 5: 181-204.

Vaishali, R.P. and G.M. Rupa, 2011. Impact of outlier removal and normalization approach in modified k-means clustering algorithm. Int. J. Comput. Sci., 8(5): 331-336.