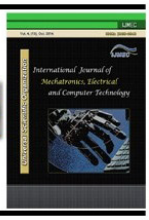
Vol. 7(23), Jan. 2017, PP. 3255-3262

Contents list available at IJMEC

International Journal of Mechatronics, Electrical and Computer Technology (IJMEC)

Journal Homepage: www.aeuso.org

PISSN: 2411-6173 - EISSN: 2305-0543



Text Document Clustering by Using Semi-supervised Learning and Outlier Detection

Bahman Askari[1*] and Sattar Hashemi[2]

[1,2]*Department of Computer Science and Engineering, Shiraz University, Shiraz, Iran*

***Corresponding Author's E-mail:** bmn.askari@gmail.com*

Abstract

Text document clustering is the process of grouping similar documents into clusters. Clustering is a technique of unsupervised categorization which divides the objects of dataset into a specific number of clusters based on the criterion of similarity or dissimilarity. This categorization is in a way that the resulted clusters are distinct as possible and with the maximum of inside cluster similarity. K-means algorithm is one of the most famous and most liked techniques of clustering because it is easy to understand and perform. Also it has a kind of linear complexity. K-means suffers from outliers in data sets, high sensitivity to the initial centers and also correct number of clusters. In order to overcome this drawbacks we propose a novel method in three stages. In the first stage ODBD algorithm is applied to detect the outliers and divide data sets to two groups; normal objects and outlier objects. Then FICBC algorithm is run on normal objects to calculate the cluster centers intelligently by using additional information and initial knowledge such as K; the number of clusters, cannot and must linked sets. Finally we use the centers obtained from the previous phase and with an iteration, we calculate the distance between each of outliers and these centers. Then, each outlier is assigned to the closest cluster. The Euclidean distance criterion is used for the calculation of this distance. The proposed method is run on the UCI dataset and the obtained results are compared to the other clustering methods. Experiments show that the proposed method achieves significantly better results than previous clustering approaches.

***Keywords*:** Text Document Clustering, Outlier Detection, Semi-supervised Learning, K-means

1. Introduction

Cluster analysis aims to identify homogeneous groups of units called clusters within data [3]. Clustering is an unsupervised technique in which the data sets which are usually vectors in multi-dimension space are divided into a certain number of clusters based on a similarity or dissimilarity criteria. For example, if the number of clusters is K, and there exist n number of m-dimension data, the clustering algorithm will assign each object to a cluster. This assignment takes place according to this rule that the assigned data to a certain cluster are more similar to each other rather than the other clusters. The K-means algorithm is one of the most well-known clustering algorithms and is being used in various types of data mining. The K-means categorizes data set objects in certain numbers of clusters. This method is one of the most attractive and widely used operations in clustering techniques; because it is simple and understandable and its time complexity is linear. What follows is the pseudo-code of K-means:

K-means algorithm

Inputs: Data set D = {d_1, d_2, … d_n }, where di =data points, n= number of data points
K = number of cluster centers
Outputs: K clusters with their centers

Article History:
IJMEC DOI: 649123/10.225116
Received Date: Aug. 29, 2016
Accepted Date: Dec. 16, 2016
Available Online: Jan. 18, 2017

Step 1:
Randomly select k data object from dataset D as initial cluster centers.
Step 2:
Repeat step 3 to step 5 till no new cluster centers are found
Step 3:
Calculate the distance between each data object d_i (1<=i<=n) and all K cluster centers C_j (1<=j<=K) and assign data object di to the nearest cluster.
Step 4:
For each cluster j (1<=j<=K), recalculate the cluster center.

The K-means algorithm has some pitfalls. For instance, it stops in local optimums and is sensitive to the initial values of cluster centers and outliers in dataset. In K-means algorithm, the initial cluster centers are selected randomly, and consequently it can cause nondeterministic results. In each dataset, an outlier is an object which its distance is not normal comparing to the other objects. In other words, an outlier is an object that has less similarity with the other objects. These problems have an undesirable effect on the efficiency and accuracy of the K-means algorithms.

In machine learning, using both labeled and unlabeled data for the purpose of learning is called semi-supervised learning, the main goal of which is to incorporate unlabeled data in order to improve generalization when insufficient training information is available. Recently, semi-supervised learning has been become an important topic in both theory and practice since there are many classifications where the labeled data may be very scarce or expensive, while unlabeled data may be more available [7]. Initial knowledge and additional information can be used in K-means clustering to calculate the cluster centers intelligently.

In real applications, groups that we want to extract can be too complex to be discovered by strictly unsupervised algorithms [1]. Thus, in order to support analysis or visualization of data, the user often provides additional information to indicate the crucial values, parts of a graph, etc. Consequently, clustering process is supposed to take an advantage of such background knowledge to provide better results. Constrained clustering is a part of semi-supervised learning [2]. It incorporates equivalence constraints between some pairs of elements to enforce which of them belong to the same group (positive constraints or must linked constraints) and which do not (negative constraints or cannot-linked constraints) [13]. Numerous clustering algorithms have been modified to aggregate additional information from equivalence constraints.

Most of adopted methods, including K-means [13], Gaussian mixture model [12, 10], hierarchical algorithms [4], spectral methods [6], generate partitions, which are fully consistent with imposed restrictions (they define hard-type of constraints). A two layer semi-supervised clustering method was proposed by [8]. They used space level constraints clustering, positive and negative constrains to improve the text clustering accuracy. A novel constrained clustering method, Constrained clustering with a complex cluster structure (C4s), was presented by [9] which incorporates equivalence constraints, both positive and negative, as the background information. C4s is capable of discovering groups of arbitrary structure, e.g. with multi-modal distribution, since at the initial stage the equivalence classes of elements generated by the positive constraints are separated into smaller parts. That provides a detailed description of elements, which are in positive equivalence relation. In order to automatically detect the number of groups, the cross-entropy clustering is applied for each partitioning process.

In this paper we propose a constrained clustering algorithm with outlier detection called ODBD-FICBC-K means to improve the accuracy of K-means algorithm for text retrieval.

2. Methodology

To improve the efficiency of the K-means algorithm, we present a novel method in three stages. In first stage, the dataset should be investigated for specifying and detecting the outliers. In this step we use ODBD (Outlier detection based on dissimilarity) algorithm to divide dataset into two subsets of

normal object and outliers. In second layer we use the number of clusters, positive and negative constrains as initial knowledge to calculate the initial centers intelligently by FICBC algorithm. Finally the K-means clustering process should be performed separately for normal object and outliers.

2.1. Outlier detection in dataset

In this stage the ODBD algorithm, which is based on the dissimilarity of objects, is employed for specifying and detecting the outliers. According to this algorithm, a value is calculated for all of the dataset objects which is called the dissimilarity degree. The objects that their dissimilarity degrees are higher than the threshold value are considered as the outlier. Assume that a data set DS is defined in the form of: DS= (D,A) in which D={d_1,d_2,… d_n} is the set of n objects and A={a_1,a_2,… a_m} is the set of attributes with the order of m. The dissimilarity degree of two objects d_i, $d_j \in$ D on the attribute of f $\in$ A is calculated as following:

$$ad_{ij}^{f} = \frac{\left(\left\| d_{if} - d_f \right| - \left| d_{jf} - d_f \right\| \right)^2}{d_{\max f} - d_{\min f}} \quad (1)$$

Where d_{if} and d_{jf} are the values of attribute f in the objects i and j, respectively. d_f, d_{max} and d_{min} are the average, the maximum and the minimum value of attribute f on all objects of the dataset, respectively. The dissimilarity degree of two objects can be obtained from the average value of the dissimilarity of the objects on each of attributes, as following:

$$od(i,j) = \frac{\sum_{ak=1}^{m} ad_{ij}^{ak}}{m} \quad (2)$$

Where ad_{ij}^{ak} is the dissimilarity value of the objects i, j on the a_k-th attribute. According to the equations (1) and (2), the dissimilarity matrix dm, which is an order N square matrix, is created to calculate the dissimilarity of each object with respect to the other objects. By adding the row elements of this matrix, the objects synergic dissimilarity matrix is made which shows the dissimilarity degree of each object with respect to all other objects of the data set. For the sake of simplicity and simplification of comparisons, the synergic dissimilarity degree matrix is normalized and then, the average of elements of this matrix with an impact factor value in the range of [0,1] is considered as the threshold similarity value. What follows is the pseudo-code of ODBD:

ODBD algorithm

inputs: data set D = { d_1, d_2, … d_n }, where di =data points, n= number of data points
impact factor value IFV $\in$ [0,1]
outputs: outlier and normal objects
step 1:
step 1-1:
for each data object di from D
for each data object d_j from D
calculate od(i,j) by using equation (1) and (2)
dm(i,j)=od(i,j)
end for
end for
step 1-2:
for each row r_i in dm
calculate sum of elements and assign in sd_i
end for
calculate d_{max} = maximum value in sd

step 1-3:
for each data value sd_i in sd
$dd_i = (d_{max} - sd_i) / d_{max}$
end for
td=mean(dd) *IFV
step 2:
for each data object di from D
if dd_i < td
assign d_i to outlier objects
else
assign d_i to normal objects
end if
end for

2.2. Calculating the initial cluster centers by using additional information

The initial points of K-means are selected randomly, so the accuracy is not deterministic and depends on the initial points. In this stage, we use the number of clusters, positive and negative constrains as additional information to propose an algorithm to compute the cluster centers more intelligently.

In this part, FICBC algorithm are explained. First, SLCC and its other versions are introduced and then, K-means is briefly discussed and the combinatorial structure is introduced.

Space Level Constraints Clustering (SLCC) is a semi-supervised clustering method which uses a hybrid method that benefits from both constrained information and intelligent distance measuring criterion. There are two types of constraints; must linked and cannot linked. In the must linked constraints, each two marked samples should be grouped in one cluster and this grouping scheme is based on the prior knowledge of the problem. In contrast, for cannot linked constraints, prior knowledge is employed to avoid grouping of two samples in one cluster.

Incidentally, FICBC uses the shortest path as its metric which can be considered as an adaptive for intelligent metric instance. Regarding to the mentioned properties, SLCC is an efficient and flexible scheme to form the primary clusters. The distinct point of SLCC in comparison with other clustering methods, is utilizing the prior knowledge to form the significant clusters [5, 11]. SLCC implementation can be described as follows: first; the distance of those sentences which should be grouped in one cluster is set to zero. Next, according to cannot linked constraints the distance of those samples which should not be in one cluster, should be infinite. In the third stage; propagation method is employed. This method acts according to shortest path metric [4] that uses the user defined constraints. What follows is the pseudo-code of FICBC:

FICBC algorithm

Inputs: Data set D = { d_1, d_2, … d_n }, where di =data point, n= number of data points
K = number of cluster centers, MLS: must linked set, CLS: cannot linked set
Outputs: initial center of clusters
for each data object di from D
for each data object d_j from D
sp(i,j)=euclidean distance(d_i,d_j)
end for
end for
for (i,j) $\in$ MLS
sp(i,j) = 0
sp(j,i) = 0
end for
for k=1:size(MLS)
for i=1:size(sp)

```
            for j=1:size(sp)
                sp(i,j) = MIN(sp(i,j),sp(i,k)+sp(k,j))
        end for
    end for
    end for
    foreach (i,j) ∈ sp
        if sp(i,j) == 0
        add {(i,j)} to MLS
    end for
    for (i,j) ∈ CLS
        sp(i,j) = inf
      sp(j,i) = inf
    end for
    for (i,j) ∈ CLS and (j,k) ∈ MLS
        sp(i,k) = inf
        sp(k,i) = inf
    end for
    clusters = {Cj, for each data object di}
    linkage starts empty
    distance(i,j) = sp(i,j)
    while (size(clusters)> K)
        [min-row,min-col] = min(sp)
        add (min-row,min-col) to linkage
        merge (min-row, min-col) to Cnew in clusters
        for Ci in clusters
                Distance(Ci, Cnew) = MAX(distance(Ci, C1), distance(Ci , C2))
        end for
    end while
```

The general algorithm is as follows. We have some datasets as input. When we have the instances, we can create the proximity matrix, and with any constraints we give from user, we should propagate the constraints on the matrix and update the proximity matrix. Then we supply this new matrix to a proximity-based clustering algorithm. In SLCC algorithm it would like specific instances that want to be in the same class to be very close together, and two instances that want to put in different classes should be very far apart, and for using this change in proximity matrix, it should increase the distance between two cannot linked points and decrease the distance between two must linked points, and propagate the distances between other points with shortest-path algorithm, we know this phase as imposing constraints. After this phase the rest of the algorithm is lookalike the instance level algorithm. After applying the algorithm, if points d_i and d_j are very close together, then points that are very close to d_i are close to d_j and if points d_i and d_j are very far apart, then points that are very close to d_i are far from d_j. To apply the constraints on the proximity matrix, we interpret the proximity matrix as weights for a complete graph over the data points, and we impose must linked constraints by decreasing the distance between the must linked points to zero and impose all cannot linked entries to or ∞, and allow all other entries to vary.

2.3. ODBD-FICBC-K-means

As mentioned above, the K-means algorithm suffers from outliers in data sets, high sensitivity to the initial centers and also correct number of clusters. In order to overcome this two drawbacks, we propose a novel method in three stages.in the first stage ODBD algorithm is applied to detect the outliers and divide the objects into two groups; normal objects and outlier objects. Then FICBC algorithm is run on normal objects to calculate the cluster centers intelligently by using k; the number of clusters, cannot and must linked sets. Finally, by using the centers obtained from the previous phase and with an iteration, the distance between each of outliers and these centers is calculated. Then, each

outlier is assigned to the closest cluster. The Euclidean distance criterion is used for calculating this distance. What follows is the pseudo-code of ODBD-FICBC-K-means:

ODBD-FICBC-K-means algorithm

inputs: data set D = { d_1, d_2, … d_n }, where d_i =data point, n= number of data points
k = number of cluster centers
outputs: k clusters with their centers
step 1:
find outlier and normal objects by using ODBD algorithm
step 2:
centers= initial center of clusters for normal objects by using FICBC algorithm
step3:
step 3-1:
consider centers from step 2 as k initial cluster centers
step 3-2:
repeat step 3-3 to step 3-4 till no new cluster centers are found
step 3-3:
calculate the distance between each data object d_i (1=< i <= size(normal objects)) and all k cluster centers C_j (1=<j<=k) and assign data object d_i to the closest cluster.
step 3-4:
for each cluster j (1=<j<=k), recalculate the cluster center.
step 4:
for each data object d_i from outlier objects
step 4-1
calculate the distance of di to all k final cluster centers C from step 3 by using euclidean distance
step 4-2
find the closest center C_j and assign d_i to the cluster with closest center C_j
end for

3. Experimental Results and Discussion

To evaluate the algorithm presented in this study, the algorithm is implemented using MATLAB 2013 programming software and the results are compared with the K-means algorithm, SLCC and SLCC-K-means. The Iris, Bupa and Glass data sets from UCI are used in the experiments. The Iris data set is a categorization of iris flowers in which there exist three different classes of iris and each class contains 50 objects. Each object has 4 attributes. In the Bupa data set, 345 objects exist each having 6 attributes. The attributes are gathered from blood tests concerning the diagnosis of liver hampering caused by irregular drink of alcohol. Each object of this data set is the record of a male person. In the glass data set, 214 objects exist and each object has 9 attributes and this set has 6 classes. The properties of these data sets are listed in table 1.

Table 1: The datasets and their properties

dataset	#objects	#features	#clusters
Iris	150	4	3
Bupa	345	6	2
Glass	214	9	6

Selecting the impact factor and consequently the similarity threshold value is very important to detect and determine the number of outliers. This factor value might be different for each data set. According to the ODBD algorithm, if the impact factor assumed to be zero, the number of outliers would be zero. This means that the ODBD-k-means algorithm changes to the normal k-means

algorithm. Fig, 1, illustrates the number of outliers for different values of impact factor in the range [0, 1] by using ODBD algorithm.

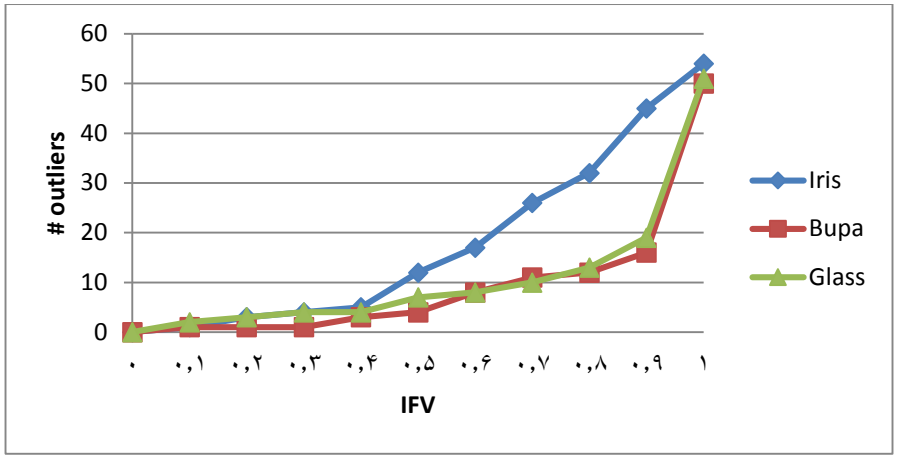


Figure 1: Comparison of the number of Outliers by Using ODBD

ODBD algorithm was run with several values of IFV and the best IFV for each data set was found. The number of outliers are shown in table 2.

Table 2: The number of outliers in data set

Dataset	Impact factor Value	#Outlier objects
Iris	0.4	5
Bupa	0.5	12
Glass	0.2	9

For calculating the algorithm accuracy, we can use accuracy indicator which is obtained using to the following relation:

$$Accuracy = \frac{True\ Positive}{TruePositive + False\ Postive} \quad (3)$$

Finally, the ODBD-FICBC-K-means was run on data sets.

Table 3: Comparison of Proposed Algorithm with similar Algorithm

Dataset	K-means	SLCC	SLCC-K-means	ODBD-FICBC-K-means
Iris	92.66	89.33	96.00	97.33
Bupa	52.43	60.28	52.46	54.55
Glass	45.79	50.93	51.87	55.14

According to the results in table 3, our proposed algorithm could improve the accuracy of K-means in comparison with the other methods especially on Iris and Glass datasets. It is helpful for clustering algorithm to discuss about the situations that use constraints. As we know, if the data are form clusters that are well separated, there is no need for prior knowledge at all. In Bupa dataset the clusters are well separated so we can see a slight improvement on this dataset. Likewise, if no distinction can be

made between classes in feature space, then by using prior knowledge, we cannot separate data correctly, and using constraints is not so helpful. Prior knowledge will therefore be most useful when patterns are at least partially separable, but a clustering algorithm will not detect them correctly without using background knowledge. This situation can arise in many ways.

Conclusion

In this paper, general cluster analysis methods have been reviewed with a special focus on K-means algorithm. The K-means algorithm suffers from outliers in data sets, high sensitivity to the initial cluster centers and also correct number of clusters. A three-layer method has employed to improve the accuracy of K-means. In the first layer, the outliers have been detected from datasets and have been temporarily removed. Then, additional information about data sets has been used as constraints to intelligently find the cluster centers in second layer. Finally, in the third layer a new version of K-means has been used to overcome drawbacks. Applying our developed method to cluster the UCI dataset gives significant results in comparison with similar methods. Currently, we are investigating other data sets and expect to improve the accuracy of K-means on them in the near future.

References

[1] Bar-Hillel A, Hertz T, Shental N, Weinshall D, “Learning distance functions using equivalence relations,” in Machine Learning, Proceedings of the Twentieth International Conference (ICML 2003), August 2003. DC, USA, AAAI Press, Washington, pp. 11–18.

[2] Basu S, Banerjee A, Mooney RJ, “Semi-supervised clustering by seeding,” in: Machine Learning, Proceedings of the Nineteenth International Conference (ICML 2002), July 2002. Australia, Morgan Kaufmann, Sydney, pp. 27–34

[3] Cristina Tortora, Mireille Gettler Summa, Marina Marino, Francesco Palumbo, “Factor probabilistic distance clustering (FPDC): a new clustering method,” Advances in Data Analysis and Classification, 2015

[4] D. Klein, S. D. Kamvar, and C. Manning, “From instance-level constraints to space-level constraints: Making the most of prior knowledge in data clustering,” in Proceedings of ICML, pp. 307–314, Sydney, Australia, 2002.

[5] K. Wagstaff, C. Cardie, S. Rogers, and S. Schroedl. “Constrained k-means clustering with background knowledge,” in Proceeding of the 18th International Conference on Machine Learning (ICML-01), pp. 577–584, 2001.

[6] Li Z, Liu J, Tang X, “Constrained clustering via spectral regularization,” in: Computer Vision and Pattern Recognition, (CVPR 2009), IEEE Conference on, pp. 421–428, 2009

[7] Liming Yang and Laisheng Wang, “A class of semi-supervised support vector machines by DC programming,” Advances in Data Analysis and Classification, pp. 417-433, 2013.

[8] Mohammad D.P, Eghbal M, Reza B, “A Two layer semi-supervised Clustering method for text retrieval,” International Journal of Computer Technology & Applications, Vol 3 (6), pp. 1971-1978, Nov-Dec 2012

[9] Marek Smieja, Magdalena Wiercioch, “Constrained clustering with a complex cluster structure,” Advances in Data Analysis and Classification, pp. 1–26, 2016.

[10] Melnykov V, Melnykov I, Michael S, “Semi-supervised model-based clustering with positive and negative constraints,” Advances in Data Analysis and Classification, pp. 1–23, 2015.

[11] S. Basu, “Semi-Supervised Clustering with Limited Background Knowledge,” in Proceeding of the Ninth AAAI/SIGART Doctoral Consortium, pp. 979- 980, San Jose, CA, July 2004.

[12] Shental N, Bar-Hillel A, Hertz T, Weinshall D, “Computing Gaussian mixture models with EM using equivalence constraints,” Adv Neural Inf Process Syst, 16(8), pp. 465–472, 2005.

[13] Wagstaff K, Cardie C, Rogers S, Schrödl S, “Constrained k-means clustering with background knowledge,” in: Machine Learning, Proceedings of the Eighteenth International Conference (ICML 2001), Williams College, Williamstown, MA, USA, Morgan Kaufmann, pp. 577–584.