# *An efficient K-Means Algorithm integrated with Jaccard Distance Measure for Document Clustering*

Mushfeq-Us-Saleheen Shameem
Lecturer, Dept. of Computer Science and Engineering
University of Development Alternative (UODA)
Dhaka, Bangladesh
shameem95@yahoo.com

Raihana Ferdous
Master in Computer Science
University of Trento
Trento, Italy
raihana.ferdous@yahoo.com

*Abstract*— **Document Clustering is a widely studied problem in Text Categorization. It is the process of partitioning or grouping a given set of documents into disjoint clusters where documents in the same cluster are similar. K-means, one of the simplest unsupervised learning algorithms, solves the well known clustering problem following a simple and easy way to classify a given data set through a certain number of clusters (assume k clusters) fixed a priori. The main idea is to define k centroids, one for each cluster. This clustering algorithm uses an iterative procedure which converges to one of numerous local minima. We have found that these iterative techniques are especially sensitive to initial starting conditions of the centroid of each cluster and the more the distance among the cluster centroid the better the clustering performance. In simple K-means algorithm the way to initialize the centroid is not specified and one popular way to start is to randomly choose k points of the samples as k centroids but this process does not guarantee to choose the maximum dissimilar documents as the centroid point for k-cluster. In this paper we proposed a modified k-means algorithm which uses Jaccard distance measure for computing the most dissimilar k documents as centroids for k clusters. Our experimental results demonstrate that our proposed K-means algorithm with Jaccard distance measure for computing the centroid improves the clustering performance of the simple K-means algorithm.**

***Keywords-Document Clustering, K-Means algorithm, Precission, Recall, F1-Measure, Entropy.***

## I. INTRODUCTION

Document clustering or Text categorization is closely related to concept of data clustering. Document clustering is a more specific technique for unsupervised document organization, automatic topic extraction and fast information retrieval or filtering. For example, as amount of online information are increasing rapidly, users as well as Information retrieval system needed to classify the desired document against a specific query. Generally two types of clustering approaches are used where one is bottom up and the other one is top down. In this paper we have focused on the performance of K-means clustering algorithm, a top down clustering algorithm which assigns each document to the cluster whose center (also called centroid) is nearest. Here the documents are represented in vector space model as document vector and the center is the average of all the documents in the cluster. It is an unsupervised algorithm where "K" stands for number of clusters and from a set of documents; K-means attempts to classify them into K clusters by approaching an iterative way.

K-means is based on the idea to cluster n documents based on terms into k partitions so that the intra-document similarity is high rather than inter-document similarity. However, the clustering performance of the K-means algorithm depends on the initial evaluation of the centroid point for the cluster. These centroids shoud be placed in a cunning way because of different location causes different result. So, the better choice is to place them as much as possible far away from each other. But in case of simple K-means algorithm we have found that at the initial evaluation of the centroid are done randomly. So sometimes it may produce very poor performance as it fails to classify the documents in disjoint sets. In this paper, we have indicated this problem and proposed a technique to measure the initial guess for the centroid points for K clusters. Here the documents are represented in the vector space model and some dissimilarity measurement techniques can be applied over the document set to find out the most dissimilar K documents. We have used the Jaccard distance measure for finding the K most dissimilar documents. Then these K points should be used as K centroid which guarantees to classify the document in K disjoint sets. In the following subsections we have described our proposed modification in the K-means algorithm and we have also implement a document clustering model with our modified K-means algorithm to evaluate the clustering efficiency using our proposed K-means algorithm.

## II. BASIC DEFINITION

### *A. Simple K-Means Algorithm*

K-means algorithm follows a simple and easy way to classify a given document set through a certain number of clusters (assume k clusters). The main idea is to define k centroids, one for each cluster. The simple K-means algorithm chooses the centroid randomly from the document set. The next step is to take each document belonging to a given data set and associate it to the nearest centroid. The K-means clustering partitions a data set by minimizing a sum-of-squares cost function.

978-1-4244-4570-7/09/$25.00 ©2009 IEEE

$$J = \sum_{j=1}^{k} \sum_{i=1}^{x} \left\| x_i^{(j)} - c_j \right\|^2 \quad (1)$$

where $\left\| x_i^{(j)} - c_j \right\|^2$ is a chosen distance measure between a document $x_i^{(j)}$ and the cluster centre $c_j$, is an indicator of the distance of the n documents from their respective cluster centroids.

### *B. Jaccard Distance Measure*

The Jaccard index, also known as the Jaccard similarity coefficient, is a statistic used for comparing the similarity and diversity of sample sets. The Jaccard coefficient measures similarity between sample sets, and is defined as the size of the intersection divided by the size of the union of the sample sets. The Jaccard distance, which measures dissimilarity between sample sets, is complementary to the Jaccard coefficient.

$$J_\delta(A,B) = 1 - J(A,B) = \frac{|A \cup B| - |A \cap B|}{|A \cup B|} \quad (2)$$

## III. MODIFIED K-MEANS ALGORITHM

Our experimental study shows that the clustering performance using K-means algorithm depends on the initial definition of the centroid point for k clusters. These centroids shoud be placed in a cunning way because it produces different results for different location. So, the better choice is to place them as much as possible far away from each other. But we have found that in the simple version of the k-means procedure can be viewed as a greedy algorithm for partitioning the n samples into k and it's main weakness is that the way to initialize the centroid is not specified. One popular way to start is to randomly choose k of the samples which does not always guarantee to choose the most dissimilar k points as centroid points. So sometimes it chooses centroid points which are similar and this indicates poor clustering performance. We have proposed a modified K-means algorithm which uses Jaccard distance co-efficient among the documents in the corpus to find the most dissimilar k documents and these k document points act as centroid point for k cluster. This technique assures to classify the corpus in disjoint document sets and increase the clustering performance drastically.

### *A. Modified K-Means clustering algorithm*

Suppose that we have n sample documents vectors d1, d2, ..., dn all from the same class, and we know that they fall into k compact clusters, k < n. Let mi be the initial centroid of the of the vectors in cluster i. If the clusters are well separated, we can use a minimum-distance classifier to separate them. That is, we can say that x is in cluster i if || d - mi || is the minimum of all the k distances. This suggests the following procedure for finding the k means:

1. User Jaccard coefficient to find k most dissimilar document and assign them as k centroid as m1, m2, ..., mk
2. Until there are no changes in any mean
   a. Use the estimated means to classify the samples into clusters
   b. For i from 1 to k
      i. Replace mi with the mean of all of the samples for cluster i
   c. end_for
3. end_until

## IV. THE ANALYTICAL MODEL OF DOCUMENT CLUSTERING USING MODIFIED K-MEANS ALGORITHM

This section describes the details procedure for a document clustering model using K-means algorithm. We have implemented this model to find out the clustering performance of our modified K-means algorithm.

### *A. Corpus Preprocessing*

Corpus preprocessing is the first phase of document clustering. It includes tokenization and stemming. We have retrieved a set of tokens by removing non relevant features that occur uniformly across all documents in the corpus. We have seen that many words contain the canonical form of morphologically rich syntactic categories, like nouns or verb. For this purpose we have used Suffix Porter's Stemming algorithm. The error rate of stemming are measured around 5%. [2].

### *B. Feature Selection*

The idea of feature selection is to selects a subset of the token occurring in the training set, and uses only this subset as features for clustering purpose. Frequency based feature selection, Information Gain, Chi-square method X2, and Mutual Information are some popular feature selection techniques. Among these techniques, Frequency based feature selection provides significance performance in text categorization and we have used these technique. The idea is to sort the retrieved features by highest number of occurrence and select the m-best that's mean we remove terms which appears in fewer than m documents. Our feature selection procedure is based on the assumption that relevant feature will be selected which are free form local minima problem. So we used term frequency for getting relevant feature and also Inverse Document Frequency to avoid this feature getting stacked into local minima.

### *C. Weighting Schemes*

We have used vector space model to represent the document where the documents are defined as vectors in a multi-dimensional Euclidean space. Among several different ways of computing the document vectors, also known as (term) weights, we have used tf-idf weighting scheme which is one of the best known schemes. In tf-idf coordinate of document d in direction of term t is determined by term frequency which indicates the number of times t occurs in

document d, scaled in a variety of ways to normalize document length.

$$TF(d,t_i) = \frac{n(d,t_i)}{\sum_i n(d,t_i)} \quad (3)$$

Where n(d,ti) denote the number of occurrences of ti in a document and Σi n(d, ti) denote the total number of tokens in document. And Inverse document frequency IDF(t) is used to scale down the terms that occur in many documents.

$$IDF(t_i) = \log(\frac{D}{D_i}) \quad (4)$$

Where Di denotes the number of documents containing ti and D denotes the total number of documents in the collection. Finally weights can be normalized. So the weight f in a document d is:

$$W'^{d}_{f} = TF(d,t_i) * IDF(t_i) = \frac{W^{d}_{f}}{\sqrt{\sum_{t\in d}(W^{d}_{f})^2}} \quad (5)$$

*D. K-means Clustering Algorithm*

In the simple K-means algorithm the way to initialize the cluster centroid is generally performed randomly from the document set. We have proposed a technique to initialize the centroids by using jaccard distance measure which is a measure of dissimilarity between two document vectors of n dimensions space.

First we measure the disssimilarity matrix between all pair of document where documents are represented as vector in n dimensional space (each co-ordinate represent a feature). Let T(d) be the set of features appearing in document d.

$$r'(d_1,d_2) = \frac{|T(d_1)\cap T(d_2)|}{|T(d_1)\cup T(d_2)|} \quad (6)$$

Where, is a similarity measure: and . $r'(d_1,d_2)=1$ and $r'(d_1,d_2)=r'(d_2,d_1)$.

Jaccard dissimilarity metrix, $J'=1-r'$ .Using the complimentary of jaccard similarity matrix we get matrix which indicates the dissimilarity between the documents. From this dissimilarity matrix we have chosen k most dissimilar documents and initialize them as K cluster centroid.

The next step is to assign rest of the documents into the appropriate cluster. We have used the cosine similarity techniques to measure the similarity between the documents and the cluster centroid. And then we assign each document to the most similar cluster.

$$similarity = \cos(\theta) = \frac{|A.B|}{\|A\|\|B\|} \quad (7)$$

If all the documents are distributed over the cluster and there is no more update in the cluster centroids then end the iterative procedure of clustering.

## V. EXPERIMENT AND RESULT

To measure the performance of our modified K-means clustering algorithm we have implemented two clustering model one of which used the previous simple K-means algorithm and the other one with modified K-means algorithm using Jaccard distance measure. For experiment we have used a famous corpus which is Reuters-21578 collection(http://kdd.ics.uci.edu/databases/reuters21578/reuters21578.html). It includes 12,902 documents for 91 classes and split a corpus between 9603 training documents and 3299 test documents. We have used a subset of these 90 Categories which includes 3 categories with 3000 documents. And as K-means is a un-supervised learning algorithm we have combined all the documents of training and test set of these 3 category and treated them as one mixed category.

As Document Clustering using K-means algorithm is an unsupervised learning method so we used the sum of square method to evaluate the performance of clustering which is the most used method for unsupervised learning. But in our dataset we found the documents distributed into some class/category which we can consider the pre-knowledge about the document distribution. And finally we can use this pre-knowledge of document distribution for some supervised performance evaluation method such as Precision, Recall and F1-Measure. To evaluate the clustering performance we have run both simple K-means and our modified K-means algorithm several times over Reuters-21578 corpus and then we have used several techniques for measuring the clustering performance.

*A. Sum of square*

Sum of square error measure the squared error between each document and its cluster centroid. The main aim of K-means clustering algorithm is to minimize the sum of square values which indicates higher clustering performance. Let ni be the number of samples in cluster Di and μi be the cluster sample mean and the Sum of squared errors is defined as:

$$\mu_i = \frac{1}{n_i}\sum_{X\in Di} X \quad \text{And} \quad (8)$$

$$E = \sum_{i=1}^{k}\sum_{X\in D_i}\|X-\mu_i\|^2$$

We run our modified K-means clustering procedure and also the simple K-means clustering techniques and found that modified K-means reduced the sum of square value.

Table 1: Sum of Square value for Modified K-means and simple K-means algorithm

| # No | Random assignment for the first document in three cluster | Sum_of_sqare |
|---|---|---|
| 1 | Modified K-means | 294.396 |
| 2 | Simple K-means | 504.665 |

*B. Precision and Recall*

Precision and Recall which two are widely used statistical classifications. Precision is the measure of exactness or fidelity, whereas Recall is a measure of completeness. Here the Precision and Recall are defined as follows:

$$\text{Re}\,call = \frac{|\{relevantdocuments\} \cap \{documentsretrieved\}|}{|\{relevantdocuments\}|}$$

$$\text{Pr}\,ecision = \frac{|\{relevantdocuments\} \cap \{documentsretrieved\}|}{|\{documentsretrieved\}|}$$

In our Document Clustering project we have treated each cluster as if it were the result of a query and each class/category in our dataset as if it were the desired set of documents for a query. We calculate the recall and precision of that cluster for each given class. More specifically, for cluster j and class i as follows:

$$\text{Re}\,call(i,j) = \frac{n_{ij}}{n_i} \quad \text{And} \qquad (9)$$

$$\text{Pr}\,ecision(i,j) = \frac{n_{ii}}{n_j}$$

Where is the numbers of members of class/category i in cluster j and is the number of members of cluster j and is the number of members of class/category i.

In our corpus we have the following data distribution:

Table 2: Description of Document Sets

| Class/category Name | Acq | Crude | Ship |
|---|---|---|---|
| Number of Document | 2369 | 578 | 197 |

After running our implemented classifier, we got the following document distribution over the cluster as depicted in the Table3. Here our modified K-menas clustering algorithm first define the most dissimilar 3 documents and then assign those document vector as the centroid of 3 cluster. Table 3 shows the most dissimilar 3 documents are 10, 2000 and 3000 in our corpus of 3100 documents.

Table 3: Description of Document Sets after clustering using Modified K-means

| Cluster Name | Dissimilar document | Total Document | Total Retrieved document in Category with Modified K-menas |
|---|---|---|---|
| Cluster 1 | 10 | 412 | Acq: 366, Crude: 46, Ship: 0 |
| Cluster 2 | 2500 | 2089 | Acq: 1980, Crude: 77, Ship: 32 |
| Cluster 3 | 3000 | 643 | Acq:23, Crude:455, Ship: 165 |

Table 4: Precision and Recall value

| Classes | Cluster 1 | Cluster 2 | Cluster 3 |
|---|---|---|---|
| Acq | 0.89 | 0.95 | 0.04 |
| Ship | 0.11 | 0.04 | 0.71 |
| Crude | 0.0 | 0.02 | 0.26 |

| Classes | Cluster 1 | Cluster 2 | Cluster 3 |
|---|---|---|---|
| Acq | 0.15 | 0.84 | 0.01 |
| Ship | 0.08 | 0.13 | 0.79 |
| Crude | 0.0 | 0.16 | 0.84 |

From the above Precision and Recall value we get the following graph:

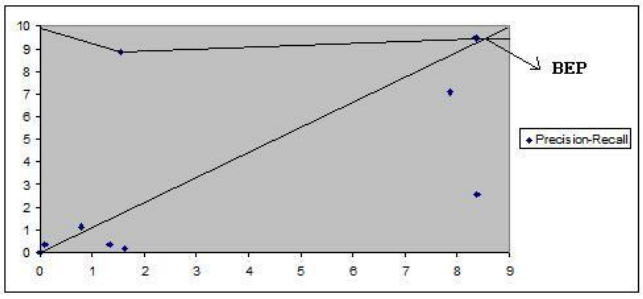


Fig 1: Precision-Recall curve with BEP

*C. F1-Measure*

We have used the F –Measure which is the weighted harmonic mean of Precision and Recall. F-Measure is the most popular performance evaluation measure that assesses the precision/recall tradeoff.

To evaluate the F1-Measure of our project the F measure of cluster j and class i is then given by

$$F = \frac{2.(\Pr ecision.\operatorname{Re} call)}{(\Pr ecision + \operatorname{Re} call)} \quad (10)$$

We have computed the weighted average of all values for F measures is using the following equation.

$$F = \sum_i \frac{n_i}{n} \max F(i, j) \quad (11)$$

To evaluate the maximum performance we have run the simple K-means algorithm on Reuters-21578 corpus 10 times and we have got maximum efficiency with this algorithm as 68%. Then we use our modified K-means clustering algorithm on the same data corpus and we have found a dramatic increase in the clustering efficiency. As we use the most dissimilar documents as the initial centroid for the clusters, we got the clustering efficiency increases upto 83% with our modified algorithm.

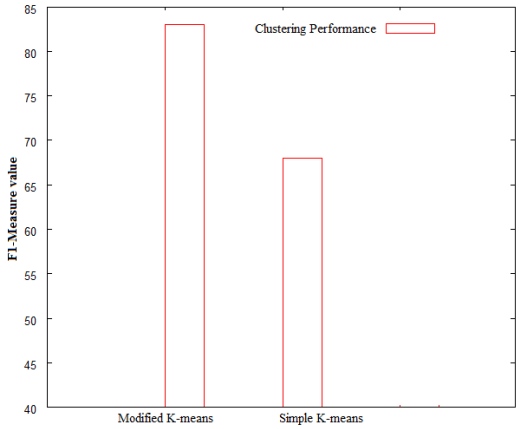


Fig 2: F1-Measure value (percentage) of clustering using modified K-means and Simple K-means

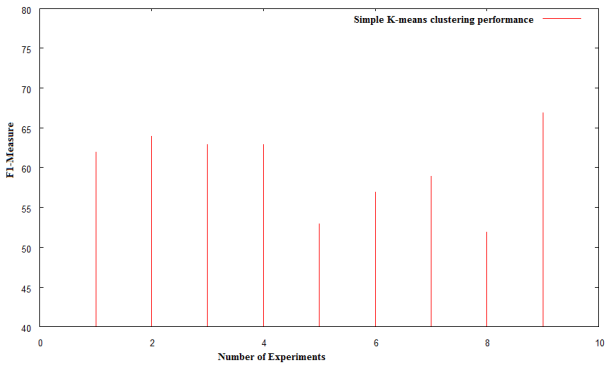


Fig 3: F1-Measure value (percentage) of clustering using Simple K-means algorithm.

### D. *Entropy*

Entropy is a measure of the uncertainty associated with a random variable. It refers to the Shannon entropy, which quantifies, in the sense of an expected value, the information contained in a message, usually in units such as bits. The general form of Entropy is:

$$E_j = -\sum_i p_{ij} \log(p_{ij}) \quad (12)$$

Where for cluster j we compute pij, the "Probability" that a member of cluster j belongs to class i.

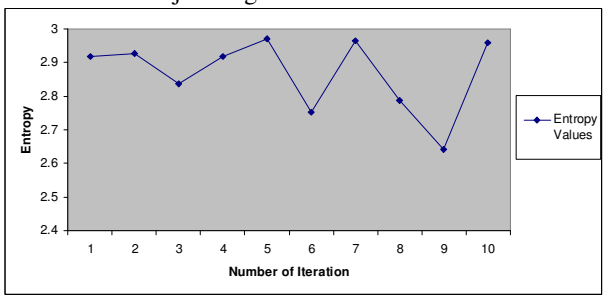


Fig 4: Entropy over 10 Iterations.

### E. *Efficiency*

To find the efficiency we tried to find how fast our project can find an appropriate cluster for a document. We have done this experiment for both the modified K-means and the simple K-means algorithm by varying the feature selection threshold. The results show that by selecting more features the efficiency of the modified K-means can be improved. But in case of simple K-means algorithm it does not guarantee to improve the efficiency by selecting more features.

Table 5: Efficiency Test on dataset containing 3144 documents

| Algorithm Type | Term Frequency Threshold | IDF Threshold | F1-Measure |
|---|---|---|---|
| Modified K-means | 5 | 1 | 0.86 |
| Modified K-means | 15 | 2 | 0.82 |
| Simple K-means | 5 | 1 | 0.69 |
| Simple K-means | 15 | 3 | 0.62 |

## VI. CONCLUSION

We have experiment the unsupervised learning with our designed classier based on the Reuter's collection data set. The Reuter's collection data set are already classified. We have proposed a modification in the simple K-means algorithm and the experiments prove that with this modification the clustering performance drastically increase.

## REFERENCES


[1] Moises Goldszmidt and Mehran Sahami, A Probabilistic Approach to Full-Text Document Clustering, Techreport (Technical Report), ID Code:357, 2008, Stanford University.

[2] Michael Steinbach, George Karypis , Vipin Kumar, A comparison of Document Clustering Technique, Department of Computer Science and Egineering, University of Minnesota, Technical Report #00-034, KDD Workshop on Text Mining, 2000.

[3] Eui-Hong (Sam) Han, Daniel Boley, Maria Gini, Robert Gross, Kyle Hastings, George Karypis, Vipin Kumar, B. Mobasher, and Jerry Moore, "WebAce: A Web Agent for Document Categorization and Exploration". Proceedings of the 2nd International Conference on Autonomous Agents (Agents'98).

[4] Daphe Koller and Mehran Sahami, "Hierarchically classifying documents using very few words", Proceedings of the 14th International Conference on Machine Learning (ML), Nashville, Tennessee, July 1997, Pages 170-178.

[5] Gerald Kowalski, "Information Retrieval Systems – Theory and Implementation", Kluwer Academic Publishers, 1997.

[6] Cutting, D. R.; Karger, D.; Pedersen, J.; and Tukey, J. W. 1992. "Scatter/Gather: A cluster-based approach to browsing large document collections ". In Proceedings of SIGIR-92. pp. 318–329. Copenhagen, Denmark.

[7] Guha, S.; Rastogi, R.; and Kyuseok, S. 1999. ROCK: "A robust clustering algorithm for categorical attributes", In Proceedings of ICDE'99. pp. 512–521. Sydney, Australia.