GF-CLUST: A NATURE-INSPIRED ALGORITHM FOR AUTOMATIC TEXT CLUSTERING

[1]Athraa Jasim Mohammed, [2]Yuhanis Yusof & [2]Husniza Husni
[1]University of Technology, Baghdad, Iraq
[2]Universiti Utara Malaysia, Malaysia

s94734@student.uum.edu.my; yuhanis@uum.edu.my;
husniza@uum.edu.my

ABSTRACT

Text clustering is a task of grouping similar documents into a cluster while assigning the dissimilar ones in other clusters. A well-known clustering method which is the K-means algorithm is extensively employed in many disciplines. However, there is a big challenge to determine the number of clusters using K-means. This paper presents a new clustering algorithm, termed Gravity Firefly Clustering (GF-CLUST) that utilizes Firefly Algorithm for dynamic document clustering. The GF-CLUST features the ability of identifying the appropriate number of clusters for a given text collection, which is a challenging problem in document clustering. It determines documents having strong force as centers and creates clusters based on cosine similarity measurement. This is followed by selecting potential clusters and merging small clusters to them. Experiments on various document datasets, such as 20Newgroups, Reuters-21578 and TREC collection are conducted to evaluate the performance of the proposed GF-CLUST. The results of purity, F-measure and Entropy of GF-CLUST outperform the ones produced by existing clustering techniques, such as K-means, Particle Swarm Optimization (PSO) and Practical General Stochastic Clustering Method (pGSCM). Furthermore, the number of obtained clusters in GF-CLUST is near to the actual number of clusters as compared to pGSCM.

Keywords: Firefly algorithm, text clustering, divisive clustering, dynamic clustering.

INTRODUCTION

Text documents, such as articles, blogs and news, are periodically increased in the Web. This increase has made online users to require more time and effort to obtain relevant information. In this context, manual analysis and manual discovery of beneficial information are very difficult. Hence, it is relevant to provide automatic tools for analysing large textual collections. Referring to such needs, data mining tasks, such as classification, association analysis and clustering are commonly integrated in the tools.

Clustering is a technique of grouping similar documents into a cluster and dissimilar documents in a different cluster (Aggarwal, & Reddy, 2014). It is a descriptive task of data mining where the algorithm learns by identifying similarities between items in a collection. According to Gil-Garicia and Pons-Porrata (2010), clustering algorithms are classified into two categories based on prior information (i.e., number of clusters): static and dynamic. In static clustering, a set of objects is classified into a determined number of clusters, while in the dynamic clustering, the objects are automatically grouped based on some criteria to discover the right number of clusters.

Clustering can also be carried out in two ways: soft clustering or hard clustering (Aliguliyev, 2009). In soft clustering, each object is grouped to be a member of any or all clusters with different membership grades, while in the case of hard clustering, the objects are members of a single cluster. Based on the mechanics of constructing the clusters, clustering methods can be divided into two categories: Hierarchical clustering and Partitional clustering (Forsati, Mahdavi, Shamsfard, & Meybodi, 2013; Luo, Li, & Chung, 2009). Hierarchical clustering methods create a tree of clusters, while Partitional clustering methods create a flat of clusters (Forsati, Mahdavi, Shamsfard, & Meybodi, 2013). This study focuses on dynamic, hard and hierarchical clustering.

LITERATURE REVIEW

Previous studies divided clustering algorithms into two main categories: Hierarchical and Partitional (Forsati et al., 2013; Luo et al., 2009). The Hierarchical clustering methods create a hierarchy of clusters (Forsati, Mahdavi, Shamsfard, & Meybodi, 2013). It is an efficient method for document clustering in information retrieval as it provides data-view at different levels and organizes the document collection in a structured manner. Based on the mechanics of constructing a hierarchy, it has two approaches:

Agglomerative and Divisive hierarchical clustering. The Agglomerative clustering approach operates from bottom to top, where every document is assumed as a single cluster. The approach attempts to merge any closest clusters based on dissimilarity matrix. There are three methods that are used for merging: single linkage, complete linkage and average linkage or also known as UPGMA (un-weighted pair group method with arithmetic mean). Details on these methods can be found in previous work, such as Yujian and Liye (2010). On the other hand, divisive clustering builds a tree of multi-level clusters (Forsati, Mahdavi, Shamsfard, & Meybodi, 2013). All objects are initially in a single cluster and at each level, the clusters are split into two clusters. Such an operation is demonstrated in Bisect K-means (Kashef, & Kamel, 2010, 2009). In the splitting process, it employs one of the partitional clustering algorithms that uses an objective function. This objective function has the ability to minimize the distance between a center and objects in one cluster and maximizes the distance between clusters.

A well-known example of a hard and static partitional clustering is the K-means (Jain, 2010). It initially determines a number of clusters and then assigns objects of a collection into the predefined clusters while trying to minimize the sum of squared error over all clusters. This process continues until a specific number of iterations is achieved. Figure 1 illustrates the steps involved in K-means. The K-means algorithm is easy to implement and efficient. However, it suffers from some drawbacks: random initialization of centers (including the determination of number of clusters) may cause the solution to be trapped into local optima. To overcome such weakness, a research area that employs meta-heuristics has been developed. It optimizes the solution to search for global optimal or near optimal solution using an objective function (Tang, Fong, Yang, & Deb, 2012). Problems are formulated as either a minimum or maximum function (Rothlauf, 2011).

There are two types of Meta-heuristic approaches: single meta-heuristic solution and population meta-heuristic solution (Boussaïd, Lepagnot, & Siarry, 2013). Single meta-heuristic solution initializes one solution and moves away from it, such as implemented in the Simulated Annealing (Kirkpatrick, Gelatt, & Vecchi, 1983) and Tabu Search (Glover, 1986). Population meta-heuristic solution initializes multi solutions and chooses the best solution based on evaluation of solutions at each iteration, such as in Genetic algorithm (Beasley, Bull, & Martin, 1993), Evolutionary programming (Fogel, 1994), Differential Evolution (Aliguliyev, 2009) and nature-inspired algorithms (Bonabeau, Dorigo, & Theraulaz, 1999). The nature-inspired algorithm, also known as Swarm intelligence, is related to the collective behavior of social insects or animals in solving hard problems (Rothlauf, 2011). Swarm intelligence,

includes Particle Swarm Optimization (PSO) (Kennedy, & Eberhart, 1995), Artificial Bee Colony (Mahmuddin, 2008; Mustaffa, Yusof, & Kamaruddin, 2013), and Firefly Algorithm (Yang, 2010).

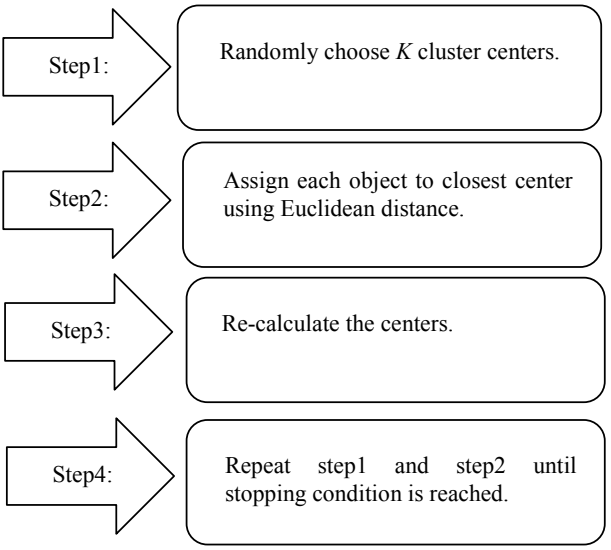


Figure 1. Steps of K-means (Jain, 2010).

With regards to the population meta-heuristic approach, Aliguliyev (2009) developed a modified differential evolution (DE) algorithm for text clustering that optimized various density based criterion functions: internal, external and hybrid functions. The result indicates that the proposed DE algorithm speeds up the convergence. On the other hand, the PSO, proposed by Kennedy and Eberhart (1995) was used for document clustering in Cui, Potok, & Palathingal (2005). The basic idea of PSO comes from the flock and foraging behavior where each solution has n dimensions search space. The birds do not have search space, so it is called "Particles". Each particle has a fitness function value that can be computed using a velocity of particles flight direction and distance. The basic PSO clustering algorithm (Cui, Potok, & Palathingal, 2005) is illustrated in Figure 2.

Initially, each particle randomly chooses a number of cluster centers from a vector of document dataset. Then, each particle performs three steps: creating clusters, evaluating clusters and creating new solutions. Creating clusters is done by assigning documents to the closest center. Evaluating clusters is done by evaluating the created clusters using fitness function (i.e., average distance

between documents and center (ADDC)) and selecting the best solution from multiple solutions. The last step is creating new solutions which is done by updating the velocity and position of particle. The previous three steps are repeated until one of the stopping conditions is reached; the maximum number of iterations or the average change in center is less than a threshold (predefined value).

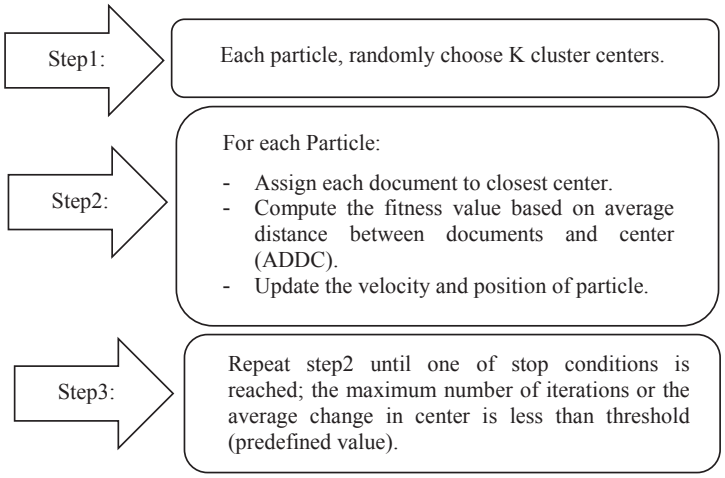


Figure 2. Steps in standard PSO clustering (Cui, Potok, & Palathingal, 2005).

Further work in clustering also includes Rashedi, Nezamabadi-pour and Saryazdi (2009) who proposed optimization algorithm that utilizew law of gravity and law of motion known as Gravitation Search Algorithm (GSA). They considered each agent as an object and determined their performance by their masses. The agents move towards heavier masses (the mass is calculated by map of fitness). The heavy masses represent good solutions. Later (Hatamlou, Abdullah, & Nezamabadi-pour, 2012), a hybrid Gravitational Search Algorithm with K-means (GSA-KM) for numerical data clustering was presented. GSA algorithm prevents K-means from trapping into local optima whereas the K-means algorithm speeds up the convergence of GSA. On the other hand, Fuzzy C-means (FCM) algorithm (which is a type of soft clustering) based on gravity and cluster merging is presented in Zhong, Liu, & Li (2010). It tries to find initial centers of clusters and solve the outlier's sensitivity problem. It also utilizes the law of gravity but the calculation of mass is different where the mass of object p is the number of neighbourhood objects of p.

Previous clustering algorithms (Cui, Potok, & Palathingal, 2005; Jain, 2010; Zhong, Liu, & Li, 2010) are denoted as static method which initially requires a pre-defined number of clusters. Hence, such algorithms are not appropriate to cluster data collections that are not accompanied with relevant information (i.e., number of classes or clusters). To date, such issues are solved using two approaches: estimation or by using dynamic swarm based approach. The first approach employs the validity index in clustering, which can drive to select the optimal number of clusters. Initially, it starts by determining a range of clusters (minimum and maximum number of clusters). Then, it performs clustering with the various numbers of clusters and chooses the number of clusters that produces the best quality performance. In the work of Sayed, Hacid and Zighed (2009), the clustering is of hierarchical agglomerative with validity index (VI) where at each level of merging step, it calculates the index of two closest clusters before and after merging. If the VI improves after merging, then merging of the clusters is finalized. This process continues until it reaches optimal clustering solution. Similarly, Kuo and Zulvia (2013) proposed an automatic clustering method, known as Automatic Clustering using Particle Swarm Optimization (ACPSO). It is based on PSO where it identifies number of clusters along with the usage of K-means that adjusts the clustering centers. The ACPSO determines the appropriate number of clusters in the range of [2, Nmax] . The result shows that ACPSO produces better accuracy and consistency compared to Dynamic Clustering Particle Swarm Optimization and Genetic (DCPG) algorithm, Dynamic Clustering Genetic Algorithm (DCGA) and Dynamic Clustering Particle Swarm Optimization (DCPSO). In the work of Mahmuddin (2008), a modified K-means and bees' algorithm are integrated to estimate the total number of clusters in a dataset. The aim of using bees' algorithm is to identify as near as possible the right centroids, while K-means is utilized to identify the best cluster. From previous discussions, it is learnt that the estimation approach is suitable for a problem that requires little or no knowledge of it; however, there is difficulty to determine the range of clusters for each dataset (lower and upper bound of number of clusters).

On the other hand, the dynamic swarm based approach can automatically find the appropriate number of clusters in a given data collection, without any support. Hence, it offers a more convenient cluster analysis. Dynamic swarm based approach adapts the mechanism of a specific insect or animal that is found in nature and converts it to heuristics rules. Each swarm employs it like an agent that follows the heuristic rules to carry out the sorting and grouping of objects (Tan, 2012). In literature, there are examples of such an

approach in solving clustering problems, such as Flocking based approach (Cui, Gao, & Potok, 2006; Picarougne, Azzag, Venturini, & Guinot, 2007) and Ant based clustering (Tan, Ting, & Teng, 2011a, 2011b). The Flocking based approach relates to behavior of swarm intelligence (Bonabeau et al., 1999) where a group of flocks swarm move in 2D or 3D search space following the same rules of flocks; get close to similar agents or far away from dissimilar agents (Picarougne, Azzag, Venturini & Guinot, 2007). This approach is computationally expensive as it requires multiple distance computations. On the other hand, the Ant based approach deals with behavior of ants, where each ant can perform sorting and corpse cleaning. This approach works by distributing the data object randomly in the 2D grid (search space), then determining a specific number of ants (agents) that move randomly in this grid to pick up a data item if it does not hold any object (item) and drop the object (item) if it finds similar object. This process continues until it reaches a specific number of iterations (Deneubourg et al., 1991).

Tan, Ting and Teng (2011b) proposed practical General Stochastic Clustering Method (pGSCM) that is a simplification of the Ant based clustering approach. The pGSCM is used to cluster multivariate real world data. The pseudo code of pGSCM is illustrated in Figure 3. The input of pGSCM is a dataset, D, that contains n objects and the output is the number of clusters discovered by pGSCM method, without any prior knowledge. In the initialization of pGSCM, the dissimilarity threshold for n objects is estimated. Then, it creates n bins where each bin includes one object from dataset D. Through the working of pGSCM, it selects two objects randomly from a dataset; if the distance between these two objects is less than their dissimilarity threshold, then the level of support of the two objects is compared. If object i has less support than j, then the lesser one is moved to the greater one and vice versa. At the end of iterations, a number of small and large bins are created. The large bins are selected as output clusters while the small bins are reassigned to large bins (objects in small bins assigned to similar center in large bins). The selected large bins process is based on threshold of 50, n/20 (means the threshold is 5% of the size of dataset, n); this threshold is based on criterion used by Picarougne, Azzag, Venturini, and Guinot (2007). This method performs well compared to the state-of-the-art methods; however, randomly selecting two objects in each iteration may create other issues. There is a chance that in some iterations, the same objects are selected or some objects are not selected at all. Furthermore, the selection process initially requires large number of iterations to increase the probability of selecting different objects.

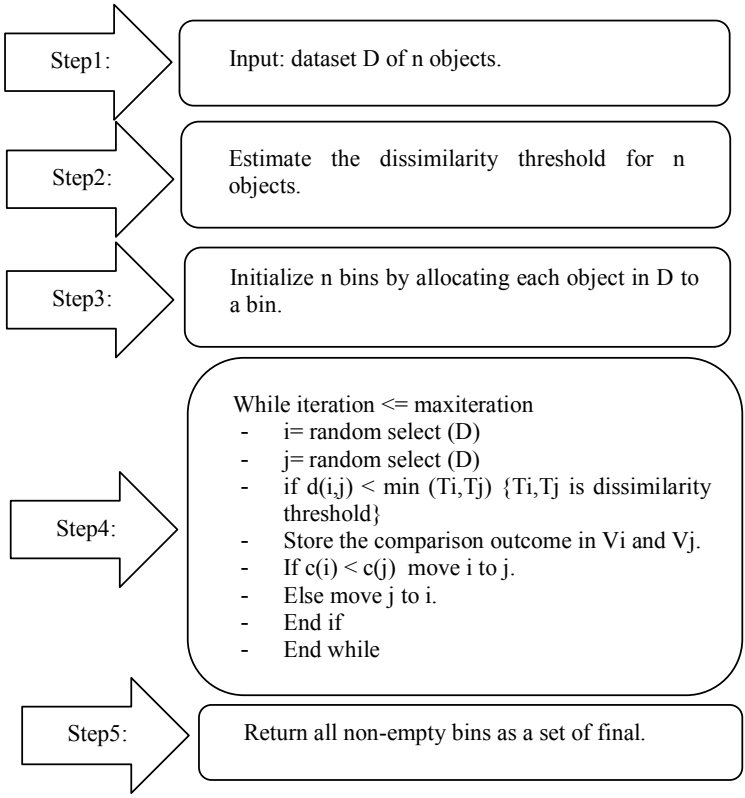


Figure 3. Pseudo code of pGSCM clustering algorithm (Tan, Ting, & Teng, 2011b).

Of late, a newly inspired meta-heuristic algorithm has appeared, known as Firefly Algorithm (FA). FA was developed and presented at Cambridge University by Xin-She Yang in 2008. Firefly algorithm is related to behavior of firefly insects that produce short and rhythmic flashes (flashing light), where, the rate of rhythmic flashes and amount of time brings two fireflies together. Further, the distance between two fireflies also affects the light, where the light becomes weaker and weaker when the distance increases. Xin-She Yang formulated this mechanism by associating the flashing light with objective function f(x). The value of x is represented by the position of the firefly, where every position has various values of flashing light. Based on the problem (maximization or minimization) that we want to solve, we can deal with the objective function. There is another factor in FA algorithm affected by the distance of two fireflies which is the attractiveness β. This factor is changed based on the distance of two fireflies; when two fireflies are attracted to each

other, the highest light will attract the lower light; this process will cause the changing in the position of two fireflies and lead to change in the value of β. The pseudo code of standard Firefly Algorithm is illustrated in Figure 4.

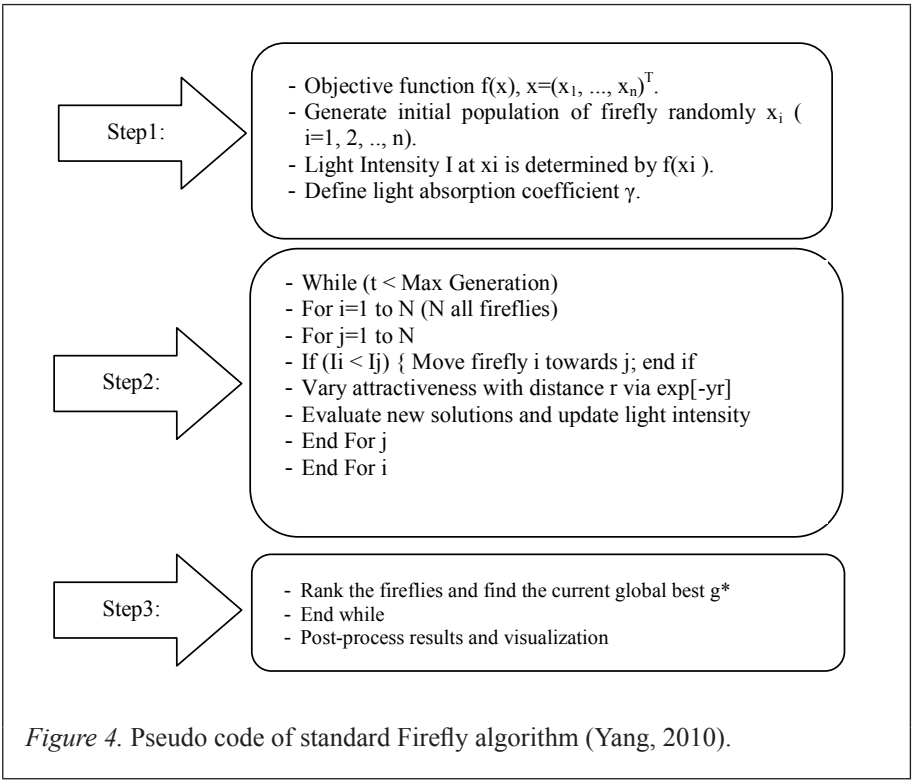


Figure 4. Pseudo code of standard Firefly algorithm (Yang, 2010).

Firefly Algorithm (Yang, 2010) has been applied in many disciplines and proven to be successful in image segmentation (Hassanzadeh, Vojodi, & Moghadam, 2011) and dispatch problem (Apostolopoulos & Vlachos, 2011). Additionally, the Firefly Algorithm was utilized in numeric data clustering and proven successful. Senthilnath, Omkar, and Mani (2011) used Firefly algorithm in supervised clustering (the class for each object is defined) and also in static manner (the number of clusters is defined). In the process of this algorithm, each firefly at specific location x in 2D search space evaluated the fitness using objective function related to the sum of Euclidean distance on all training data. The result demonstrates that FA can be efficiently used for clustering. But Banati and Bajaj (2013) implemented the Firefly algorithm differently. They used FA as an unsupervised learning (the class for each object is undefined). However, their implementation is still based on static manner, as shown in Senthilnath, Omkar, and Mani's (2011) work, where the number of clusters is pre-defined.

In this paper, Firefly Algorithm is proposed to cluster documents automatically using divisive clustering approach, and the algorithm is termed Gravity Firefly Clustering (GF-CLUST). The proposed GF-CLUST integrates the work on Gravitation Firefly Algorithm (GFA) (Mohammed, Yusof, & Husni, 2014a) with the criterion of selection clusters (Picarougne, Azzag, Venturini, & Guinot, 2007). GF-CLUST operates based on random positioning of documents that employs the law of gravity to find the force between documents which is used as the objective function.

METHODOLOGY

The proposed GF-CLUST works in three steps: data pre-processing, development of vector space model and data clustering, as shown in Figure 5.

Data Pre-processing

This step is very important in any machine learning system. It can be defined as the process of converting a set of documents from unstructured into a structured form. This process involves three steps as shown in Figure 5: Data cleaning, stop word remover and word stemming. Initially, it starts by selecting texts from each document. The extracted texts are cleaned of special characters and digits. Then, the text undergoes a splitting process that divides each cleaned text into a set of words. Later, it removes words that have length less than three characters, such as in, on, at, etc., and removes stop words, such as propositions, conjunctions, etc. The last step in pre-processing is the stemming process, where all the words are retained as the root (Manning, Raghavan, & Schütze, 2008).

Development of Vector Space Model

This step is commonly used in information retrieval and data mining approach where it represents the utilized data in a vector space. In this work, each cleaned document is represented in 2D space (columns and rows). The column denotes terms, m, extracted from documents, while the rows refer to the documents, n. The term frequency-inverse document frequency (TF-IDF) is an efficient scheme that is used to identify significance of terms in the text, The benefit of utilizing TF-IDF is the balance between the local and global term weighting in a document (Aliguliyev, 2009; Manning, Raghavan, & Schutze, 2008). Equation 1 is used to calculate the TF-IDF.

$$tfidf_{t,d} = tf_{t,d} * \log N / df_t \quad (1)$$

Data Clustering

This step includes two processes: identify centers and create clusters and selection of clusters. The identification of centers is achieved by using GFA (Mohammed, Yusof, & Husni, 2014a), where the GFA employs Newton's law of gravity to determine the force between documents and uses it as an objective function. Newton's law of gravity states that "Every point mass attracts every single other point mass by a force pointing along the line intersecting both points. The force is proportional to the product of the two masses and inversely proportional to the square of the distance between them" (Rashedi, Nezamabadi-pour, & Saryazdi, 2009). Equation 2 shows Newton's law of gravity.

$$F = G\frac{M_1 * M_2}{R^2} \tag{2}$$

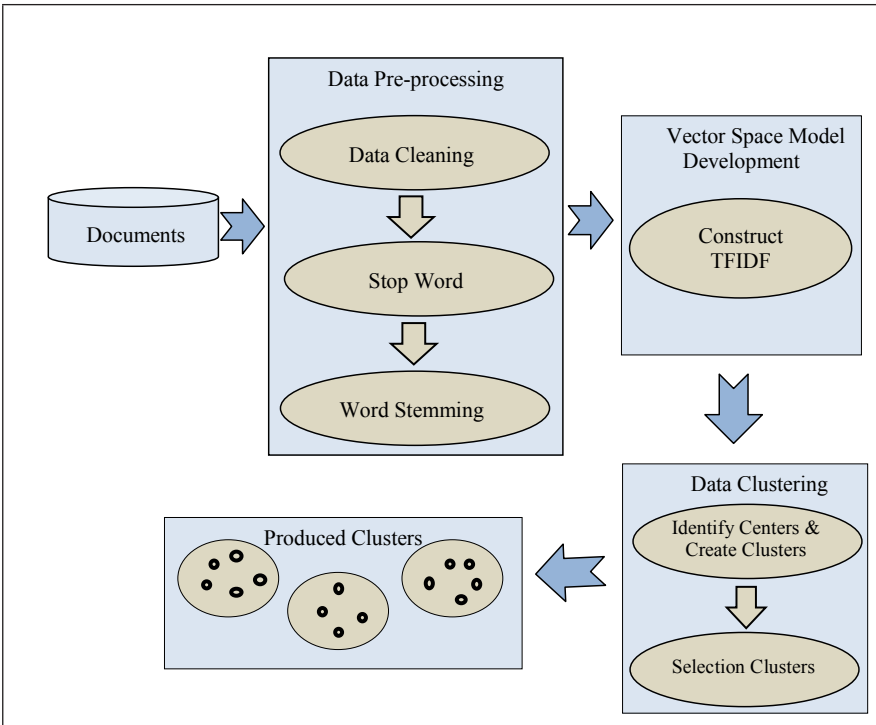


Figure 5. The process of Gravity Firefly clustering (GF-CLUST).

where F is the force between two masses, G is the gravitational constant, M_1 is the first mass, M_2 is the second mass, R is the distance between two masses.

In the GFA (Mohammed, Yusof, & Husni, 2014a), the *F* is the force between two documents as shown in Equation 3 while *G* represents the cosine similarity between two documents calculated using Equation 4 (Luo, Li, & Chung, 2009). The M_1 and M_2 represent the total weight of the first and second document, and is calculated using Equation 5 (Mohammed, Yusof, & Husni, 2014b). The value of *R* is based on Cartesian distance (Cdist) between the positions of two documents and is obtained using Equation 6 (Yang, 2010). The representation of documents position in GFA is illustrated in Figure 6, where *x* is a random value (for example, in 20 Newsgroups dataset, the value is in the range 1-300) and *y* is fixed at 0.5.

$$F(d_i * d_j) = Cosine Similarity(d_i * d_j) \frac{T(d_i) * T(d_j)}{Cdist^2} \quad (3)$$

$$Cosine Similarity(d_i * d_j) = \sum_{i=j=1}^{m} (d_i * d_j) \quad (4)$$

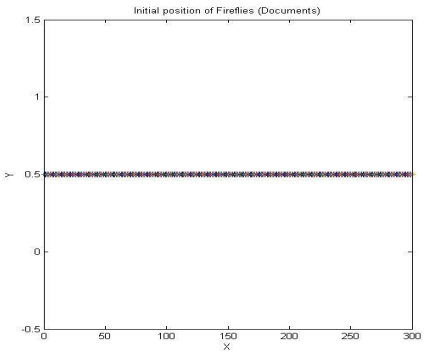
$$T(d_j) = \sum_{i=1}^{m} tfidf_{t_i, d_j} \quad (5)$$

$$Cdist(X_i * X_j) = \sqrt[2]{(X_i - X_j)^2} \quad (6)$$

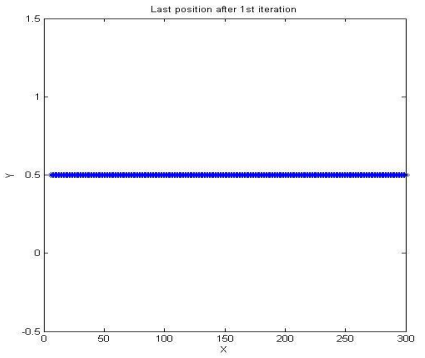
Later, the GFA assigns the value of force as initial light of each firefly (in this paper, the number of fireflies represents number of documents). Every firefly will compete with each other; if a firefly has a brighter light than another, then it will attract the ones with less bright light. The attraction value β between two fireflies changes based on the distance between these fireflies.

The position of the less bright firefly will change. Changes of the firefly position will then lead to the change of the force value (objective function in this algorithm that represents the light of each firefly). After a specific number of iterations, GFA identifies firefly (document) with the brightest light and denotes it as an initial cluster center. The pseudo-code for the identification of cluster centers in GFA is presented in Figure 7.

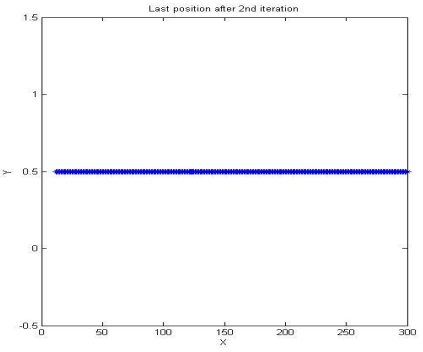
Once a center is identified, the process of creating the first cluster starts by finding the most similar documents (i.e., using cosine similarity as in equation 4). Documents that have high similarity to the centroid is located in the first cluster. This approach requires a specific threshold value (in this paper, different threshold value is used for different dataset). Documents that do not belong to the first cluster are ranked (step 17 in GFA process) based on it brightness. This is to find a new center and later, create a new cluster. Such a process is repeated until all documents are grouped accordingly.



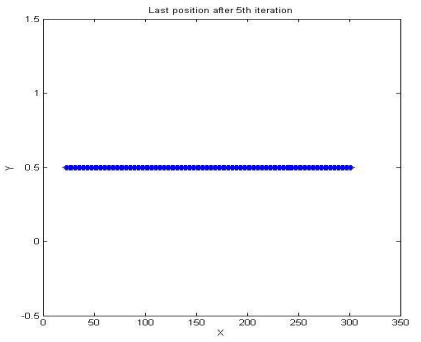
(a)Initial position of Fireflies (Documents)



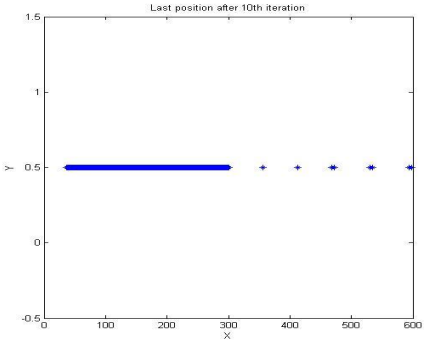
b)The Fireflies (documents) position after 1st iteration



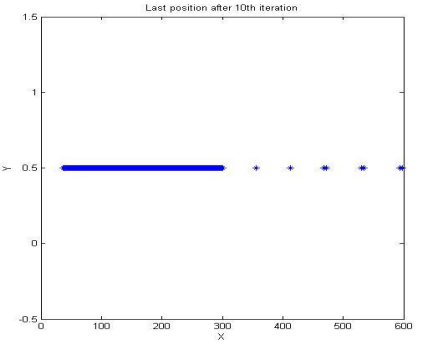
(c)The Fireflies (documents) position after 2nd iteration



(d)The Fireflies (documents) position after 5th iteration



(e) The Fireflies (documents) position after 10th iteration.



(f)The Fireflies (documents) position after 20 iterations.

Figure 6. The representation of documents position in GFA.

Step1: Generate Initial population of firefly xi where i=1, 2,.., n, n=number of fireflies (documents).
Step2: Initial Light Intensity, I=Force between two document using equation 3.
Step3: Define light absorption coefficient γ, initial γ=1
Step4: Define the randomization parameter α, α=0.2
Step5: Define initial attractiveness $\beta_0 = 1.0$
Step6: While t < Number of iterations
Step7: For i=1 to N
Step8: For j=1 to N
Step9: If(Force Ii< Force Ij){
Step10: Calculate distance between i,j using Equation 6.
Step11: Calculate attractiveness using equation below .

$$\beta = \beta_0 exp^{(-\gamma r_{ij}{}^2)}$$

Step12: Move document i to j using Equation

$$X^i = X^i + \beta * \left(X^j - X^i\right) + \alpha$$

Step13: Update force between two documents (light intensity).
Step14: End For j
Step15: End For i
Step16: Loop
Step17: Rank to identify center (brightest light).

Figure 7. Pseudo code of GFA.

Upon obtaining the clusters, cluster selection process is conducted. This process is carried out by choosing clusters (which include documents greater than 5% of the size of dataset n) that exceed an identified threshold. To date, it is set to 50, n/20 for normal distributed data, such as the 20Newsgroup (20NewsgroupsDataSet, 2006) and Reuters-21578 (Lewis, 1999) (normal distribution means every class includes the same number of documents). This threshold is based on the criteria used by Tan et al. (2011) and the idea of merging clusters is adopted from Picarougne, Azzag, Venturini, and Guinot (2007). The merging of clusters assigns smaller clusters to the bigger ones. On the other hand, the threshold value of 50, n/40 is used for dataset that is not normally distributed, such as the TR11 and TR12, retrieved from TREC collection (TREC, 1999).

RESULTS

Data Sets

Four datasets are used in evaluating the performance of GF-Clust. They were obtained from different resources: 20Newsgroup (20NewsgroupsDataSet, 2006), Reuters-21578 (Lewis, 1999) and TREC collection (TREC, 1999). Table 1 displays description of the chosen datasets.

Table 1

Description of Datasets

Datasets	Source	No. of Doc.	Classes	Min class size	Max class size	No. of Terms
20Newsgroups	20Newsgroups	300	3	100	100	2275
Reuters-21578	Reuters-21578	300	6	50	50	1212
TR11	TREC	414	9	6	132	6429
TR12	TREC	313	8	9	93	5804

The first dataset, named 20Newsgroups, contains 300 documents separated in three classes that include hardware, baseball and electronics. Each class involves 100 documents and 2,275 number of terms. The second dataset which is the Reuters-21578 contains 300 documents distributed in six classes which are the 'earn', 'sugar', 'trade', 'ship', 'money-supply' and 'gold'. Each class includes 50 documents and 1,212 number of terms. The third dataset is known as TR11 and is derived from TREC collection. It includes 414 documents distributed in nine classes. The smallest class size is six while the largest class includes 132 documents and the collection comprises 6,429 terms. The fourth dataset called TR12 is also obtained from TREC collection. It includes 313 documents with eight classes and 5,804 terms. The smallest class is nine documents while the largest contains 93 documents.

Evaluation Metrics

In order to evaluate the performance of the GF-Clust against state-of-the-art methods, K-means (Jain, 2010), PSO (Cui, Potok, & Palathingal, 2005) and pGSCM (Tan, Ting, & Teng, 2011a), four evaluation metrics are employed. These metrics include the ADDC, Purity, F-measure and Entropy (Forsati, Mahdavi, Shamsfard, & Meybodi, 2013; Murugesan, & Zhang, 2011).

The first metric, ADDC, measures the compactness of the obtained clusters. A smaller value of ADDC indicates a better cluster and it satisfies the optimization constrains (Forsati, Mahdavi, Shamsfard, & Meybodi, 2013). The ADDC can be defined as Equations 7 and 8.

$$ADDC = \sum_{j=1}^{k} \frac{\sum_{i=1}^{ni} \frac{Dis(ci * di)}{ni}}{K} \tag{7}$$

$$Dis(d_j, d_i) = \sqrt[2]{\sum_{n=1}^{m} (d_{jn} - d_{in})^2} \tag{8}$$

Where, K refers to number of clusters, n_i refers to number of documents in cluster i, c_i refers to center of cluster I, d_i refers to document in cluster i, and $Dis(c_i, d_i)$ is Euclidian distance (Murugesan, & Zhang, 2011) which is calculated using Equation 8.

Purity is defined as the weighted sum of all cluster purity as shown in Equation 9. Cluster purity is calculated based on the largest class of documents assigned to a specific cluster as shown in Equation 10. The larger the value of purity, the better a clustering solution is Entropy (Forsati, Mahdavi, Shamsfard, & Meybodi, 2013; Murugesan, & Zhang, 2011).

$$Purity = \sum_{\Theta k \in \{\Theta 1,..,\Theta c\}} \frac{P(\Theta_k * C_j)}{N} \tag{9}$$

$$P(\Theta_k * C_j) = Max_k \left| \Theta_k \cap C_j \right| \tag{10}$$

On the other hand, the F-measure metric measures the accuracy of the clustering solution as shown in Equation 11. It can be obtained by calculating two important metrics that are mostly used in evaluation of information retrieval system which are recall and precision. Recall is the division of the number of documents from specific class in specific cluster over the number of that class in whole dataset as shown in Equation 12;, while Precision is the division of the number of documents from specific class in specific cluster over size of that cluster as shown in Equation 13. Larger value of F-measure leads to a better clustering solution Entropy (Forsati, Mahdavi, Shamsfard, & Meybodi, 2013; Murugesan, & Zhang, 2011).

$$F(\theta_k) = \max_{C_j \in \{C_1,\ldots C_k\}} \left(\frac{2 * R(\theta_k, C_j) * P(\theta_k, C_j)}{R(\theta_k, C_j) * P(\theta_k, C_j)} \right. \tag{11}$$

$$R(\theta_k, C_j) = \frac{\text{The number of the members of class k in cluster j}}{\text{The number of the members of class k}} \tag{12}$$

$$P(\theta_k, C_j) = \frac{\text{The number of the members of class k in cluster j}}{\text{The number of the members of class j}} \tag{13}$$

The Entropy measures the goodness of clusters and randomness Entropy (Forsati, Mahdavi, Shamsfard, & Meybodi, 2013; Murugesan, & Zhang, 2011). It also can measure the distribution of classes in each cluster. The clustering solution reaches its high performance when clusters contain documents from a single class. In this situation, the entropy value of clustering solution will be zero. A smaller value of entropy demonstrates a better cluster performance. Equations 14 and 15 are used to compute the Entropy.

$$HC(j) = -\sum_{k=1}^{c} \frac{\left|\theta_k \cap C_j\right|}{\left|C_j\right|} \log \frac{\left|\theta_k \cap C_j\right|}{\left|C_j\right|} \tag{14}$$

$$H = \sum_{j=1}^{k} \frac{HC_j * \left|C_j\right|}{N} \tag{15}$$

Experimental Result and Discussion

The proposed GF-CLUST, K-means (Jain, 2010), PSO (Cui, Potok, & Palathingal, 2005) and pGSCM (Tan et al., 2011a) was executed 10 times before the average value of the evaluation metrics was obtained. All experiments were carried out in Matlab on windows 8 with a 2000 MHz processor and 4 GB memory. Table 2 tabularizes the results of Purity, F-measure, Entropy, ADDC and the number of obtained clusters.

Table 2

Performance Results and Standard Deviation: GF-CLUST vs. K-means vs. PSO vs. pGSCM algorithms

Validity Indices	Datasets	GF-Clust	K-means	PSO	pGSCM
Purity	20Newsgroups	**0.5667(0.00)**	0.3443(0.0314)	0.4097(0.0691)	0.3853(0.0159)
	Reuters	**0.4867(0.00)**	0.3240(0.0699)	0.3497(0.0697)	0.2357(0.0147)
	TR11	**0.4710(0.00)**	0.4087(0.0811)	0.4092(0.0606)	0.3372(0.0184)
	TR12	**0.4920(0.00)**	0.4096(0.0761)	0.3712(0.0405)	0.3029(0.0072)
F-measure	20Newsgroups	**0.5218(0.00)**	0.4974(0.0073)	0.5085(0.0390)	0.3571(0.0275)
	Reuters	**0.3699(0.00)**	0.3575(0.0664)	0.3522(0.0654)	0.2400(0.0108)
	TR11	0.3213(0.00)	**0.3792**(0.0861)	0.3492(0.0539)	0.2520(0.0346)
	TR12	**0.3851(0.00)**	0.3678(0.0958)	0.3161(0.0444)	0.2245(0.0171)
Entropy	20Newsgroups	**1.3172(0.00)**	1.5751(0.0272)	1.4847(0.0935)	1.5630(0.0132)
	Reuters	**1.6392(0.00)**	2.0964(0.2358)	2.1483(0.1563)	2.5144(0.0245)
	TR11	**2.0119(0.00)**	2.2620(0.3850)	2.2970(0.1913)	2.5660(0.0586)
	TR12	**1.9636(0.00)**	2.2768(0.2834)	2.4571(0.1551)	2.6647(0.0353)
ADDC	20Newsgroups	1.9739(0.00)	**0.5802**(0.2057)	1.8955(0.2166)	1.7517(0.0159)
	Reuters	1.5987(0.00)	**0.6957**(0.1341)	1.3806(0.1115)	1.4078(0.0142)
	TR11	1.1316(0.00)	**0.5600**(0.2648)	0.7283(0.1362)	1.0410(0.0115)
	TR12	1.1246(0.00)	**0.4592**(0.1001)	0.7486(0.1794)	1.0604(0.0065)
Number of clusters	20Newsgroups	**5**	3	3	6
	Reuters	7	6	6	**6.3**
	TR11	**8**	9	9	7.2
	TR12	**8**	8	8	6.3

Hint: The value highlighted in bold indicates the best value.

As can be seen from Table 2 and Figure 8, the Purity value for GF-CLUST is higher than K-means, PSO and pGSCM , in all datasets used in this paper (refers to 20Newsgroups, Reuters, TR11 and TR12). It is also noted that the PSO outperforms K-means and pGSCM in most datasets. On the other hand, pGSCM produces the lowest purity in all datasets excluding in 20Newsgroups dataset where it is better than K-means. Based on literature, a better clustering is when it produces high purity value (Forsati, Mahdavi, Shamsfard, & Meybodi, 2013; Murugesan, & Zhang, 2011).

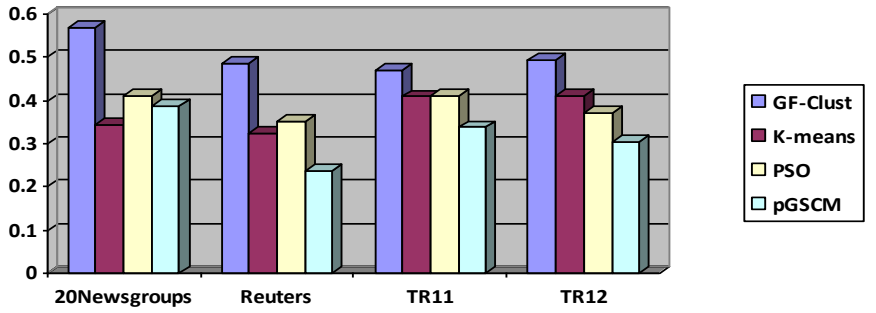


Figure 8. Purity Result: GF-CLUST vs. K-means vs. PSO vs. pGSCM Methods.

The comparative results of F-measure among GF-CLUST, K-means, PSO and pGSCM are tabularized in Table 2 and represented in a graph in Figure 9. The F-measure evaluates the accuracy of the clustering. It can be noticed from Figure 9 and Table 2 that the proposed GF-CLUST has higher F-measure value in most datasets (refers to 20Newsgroups, Reuters and TR12) despite not being supported with any information on the number of clusters. Nevertheless, for TR11 dataset, it can be seen that K-means outperforms GF-CLUST by generating 0.3792 compared to 0.3213.

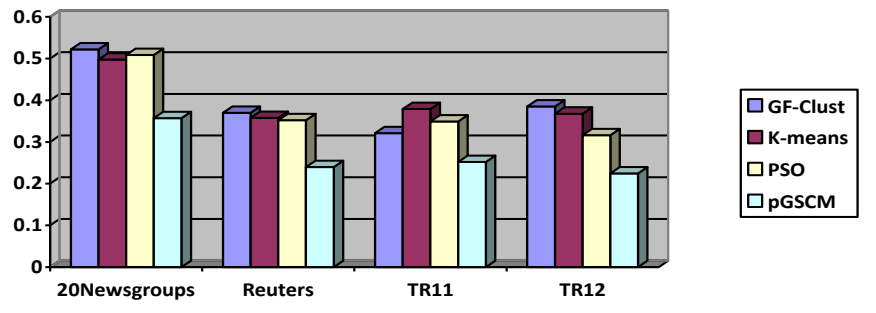


Figure 9. F-measure Result: GF-CLUST vs. K-means vs. PSO vs. pGSCM Methods.

Further, as can see in Table 2, the GF-CLUST has the best Entropy (lowest value) compared to K-means, PSO and pGSCM in all datasets tested in this work (refers to 20Newsgroups, Reuters, TR11 and TR12). The GF-CLUST produces 1.3172, 1.6392, 2.0119 and 1.9636 for 20Newsgroups, Reuters, TR11 and TR12, respectively. Further, it is also noted that K-means is a better algorithm compared to PSO and pGSCM in clustering most of the datasets (refers to Reuters, TR11 and TR12) as it produces less Entropy (i.e., 2.0964, 2.2620 and 2.2768). Figure 10 illustrates a graphical representation of the Entropy for the F-Clust, K-means, PSO and pGSCM.

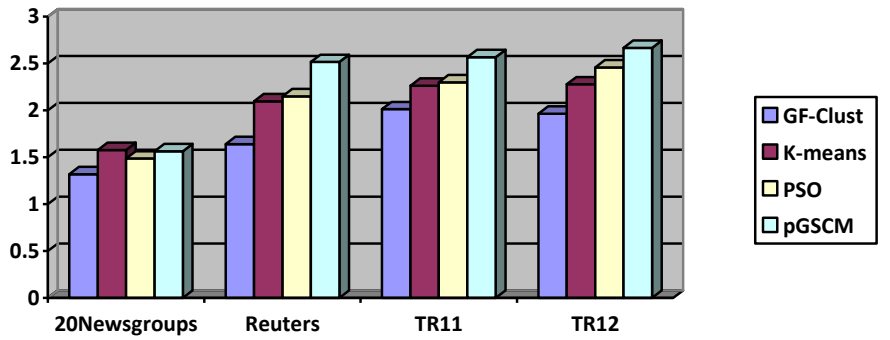


Figure 10. Entropy result: GF-CLUST vs. K-means vs. PSO vs. pGSCM Methods.

In terms of the distance between documents in a cluster, the ADDC value for Euclidian similarity is displayed in Table 2 for GF-CLUST, K-means, PSO and pGSCM. This value is used to show the algorithm satisfies the optimization constraints. The utilization of ADDC is similar to the Entropy metric where a smaller value demonstrates a better algorithm (Forsati, Mahdavi, Shamsfard & Meybodi, 2013). The ADDC results indicate that the K-means is a better clustering than PSO, pGSCM and the proposed GF-CLUST in all datasets. The reason behind this performance is that K-means utilizes Euclidean distance in assigning documents to clusters. On the other hand, the GF-CLUST employs Cosine similarity with a specific threshold to identify member of a cluster. Such an approach does not take into account the physical distance that exists between documents. Figure 11 illustrates a graphical representation of ADDC value among GF-CLUST, K-means, PSO and pGSCM.

Research in clustering is aimed at producing clusters that match the actual number of clusters. As can be seen in Table 2, the average number of obtained clusters by the GF-CLUST is 5, 7, 8 and 8 for the four datasets, 20Newsgroups,

Reuters-21578, TR11 and TR12. These values are near to the actual number of clusters which are 3, 6, 9 and 8. The result of the proposed GF-CLUST is better than the dynamic method, pGSCM, that generates 6, 6.3, 7.2 and 6.3 for the said datasets. Figure 12 shows a graphical representation of the number of obtained clusters with different algorithms using different datasets.

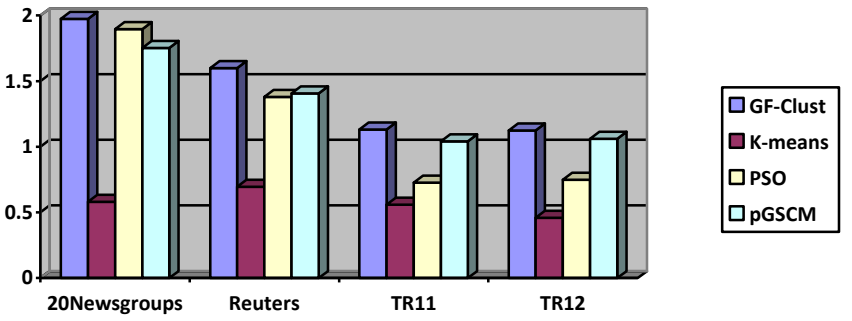


Figure 11. ADDC result: GF-CLUST vs. K-means vs. PSO vs. pGSCM Methods.

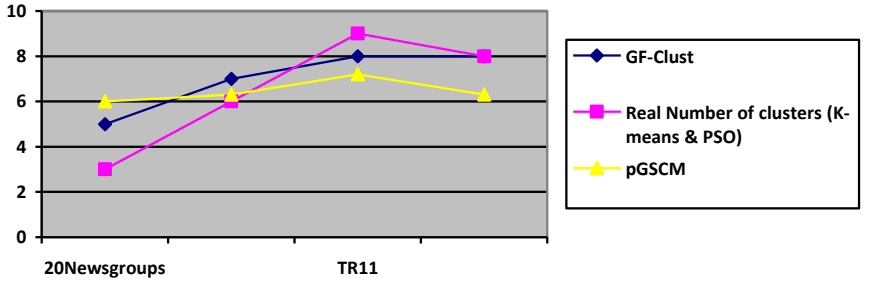


Figure 12. Results of the number of generated clusters by GF-Clust, pGSCM, K-means & PSO.

CONCLUSION

This paper presents a new method known as Gravity Firefly clustering method (GF-CLUST) for an automatic document clustering where the idea mimics the behavior of the firefly insect in nature. The GF-CLUST has the ability to identify a near optimal number of clusters and this is achieved in three steps: data pre-processing, development of vector space model and clustering. In the clustering step, GF-CLUST uses GFA to identify documents with

high force as centers and create clusters based on cosine similarity. It later chooses quality clusters and assigns small size clusters to them. Results of four benchmark datasets indicate that GF-Clust is a better clustering algorithm than the pGSCM. Furthermore, the GF-Clust performs better than K-means, PSO and pGSCM in terms of cluster quality; purity, F-measure and Entropy, hence, indicating that GF-Clust can become a competitor in the area of dynamic swarm based clustering.

REFERENCES

20NewsgroupsDataSet. (2006). Retrieved from http://www.cs.cmu.edu/afs/cs.cmu.edu/project/theo-4/text-learning/www/datasets.html.

Aggarwal, C. C., & Reddy, C. K. (2014). *Data clustering Algorithm and applications*. CRC Press, Taylor and Francis Group.

Aliguliyev, R. M. (2009). Clustering of document collection-A weighted approach. *Elsevier, Expert Systems with Applications*, *36*(4), 7904–7916. Retrieved from doi: 10.1016/j.eswa.2008.11.017

Apostolopoulos, T., & Vlachos, A. (2011). Application of the firefly algorithm for solving the economic emissions load dispatch problem. *International Journal of Combinatorics*, *201*, 23. Retrieved from doi:10.1155/2011/523806

Banati, H., & Bajaj, M. (2013). Performance analysis of Firefly algorithm for data clustering. *Int. J. Swarm Intelligence*, *1*(1), 19–35.

Beasley, D., Bull, D. R., & Martin, R. R. (1993). An overview of genetic algorithms : Part 1, fundamentals. *University Computing*, *15*(2), 58–69.

Bonabeau, E., Dorigo, M., & Theraulaz, G. (1999). *Swarm intelligence: From Natural to artificial systems*. New York, NY: Oxford University Press, Santa Fe Institute Studies in the Sciences of Complexity.

Boussaïd, I., Lepagnot, J., & Siarry, P. (2013). A survey on optimization metaheuristics. *Elsevier,Information Sciences*, *237*, 82–117.

Cui, X., Gao, J., & Potok, T. E. (2006). A flocking based algorithm for document clustering analysis. *Journal of Systems Architecture*, *52*(8-9), 505–515.

Cui, X., Potok, T. E., & Palathingal, P. (2005). Document clustering using particle swarm optimization. In *Proceedings 2005 IEEE Swarm Intelligence Symposium, SIS 2005*. (pp. 185–191). IEEEXplore. Retrieved from doi:10.1109/SIS.2005.1501621

Deneubourg, J. L., Goss, S., Franks, N., Sendova-Franks, A., Detrain, C., & Chrétien, L. (1991). The dynamics of collective sorting: robot-like ants and ant-like robots. In *Proceedings of the First International Conference on Simulation of Adaptive Behavior on from Animals to Animats* (pp. 356–363). MIT Press Cambridge, MA, USA.

Fogel, D. B. (1994). Asympototic convergence properties of gentic algorithms and evaluationary programming. *Cybernetics and Systems*, *25*(3), 389–407.

Forsati, R., Mahdavi, M., Shamsfard, M., & Meybodi, M. R. (2013). Efficient stochastic algorithms for document clustering. *Elsevier, Information Sciences*, *220*, 269–291. Retrieved from doi: 10.1016/j.ins.2012.07.025

Gil-Garicia, R., & Pons-Porrata, A. (2010). Dynamic hierarchical algorithms for document clustering. *Elsevier, Pattern Recognition Letters*, *31*(6), 469–477. Retrieved from doi: 10.1016/j.patrec.2009.11.011

Glover, F. (1986). Future paths for integer programming and links to artificial intelligence. *Computers and Operations Research*, *13*(No.5), 533–549.

Hassanzadeh, T., Vojodi, H., & Moghadam, A. M. E. (2011). An Image segmentation approach based on maximum variance intra-cluster method and Firefly algorithm. In *Seventh International Conference on Natural Computation (ICNC)* (Vol. 3, pp. 1817–1821). Shanghai: IEEE Explore. Retrieved from doi:10.1109/ICNC.2011.6022379

Hatamlou, A., Abdullah, S., & Nezamabadi-pour, H. (2012). A combined approach for clustering based on K-means and gravitational search algorithms. *Elsevier, Swarm and Evolutionary Computation*, *6*, 47–52. Retrieved from doi: 10.1016/j.swevo.2012.02.003

Jain, A. K. (2010). Data clustering: 50 years beyond K-means. *Elsevier, Pattern Recognition Letters*, *31*(8), 651–666. Retrieved from doi: 10.1016/j.patrec.2009.09.011

Kashef, R., & Kamel, M. (2010). Cooperative clustering. *Elsevier, Pattern Recognition*, *43*(6), 2315–2329. Retrieved from doi: 10.1016/j.patcog.2009.12.018

Kashef, R., & Kamel, M. S. (2009). Enhanced bisecting k-means clustering using intermediate cooperation. *Elsevier, Pattern Recognition*, *42*(11), 2557–2569.

Kennedy, J., & Eberhart, R. (1995). Particle swarm optimization. *Proceedings of IEEE International Conference on Neural Networks IV*. Perth, WA: IEEE. Retrieved from doi:10.1109/ICNN.1995.488968

Kirkpatrick, S., Gelatt, C. D., & Vecchi, M. P. (1983). Optimization by simulated annealing. *New Series*, *220*(No. 4598), 671–680.

Kuo, R. J., & Zulvia, F. E. (2013). automatic clustering using an improved particle swarm optimization. *Journal of Industrial and Intelligent Information*, *1*(1), 46–51.

Lewis, D. (1999). The reuters-21578 text categorization test collection. Retrieved from Available online at :http://kdd.ics.uci.edu/database/reuters21578/reuters21578.html

Luo, C., Li, Y., & Chung, S. M. (2009). Text document clustering based on neighbors. *Elsevier, Data & Knowledge Engineering*, *68*(11), 1271–1288. Retrieved from doi: 10.1016/j.datak.2009.06.007

Mahmuddin, M. (2008). *Optimisation using Bees algorithm on unlabelled data problems*. *Manufactoring engineering centre*. Cardiff: Cardif University , UK.

Manning, C. D., Raghavan, P., & Schütze, H. (2008). *Introduction to Information Retrieval* (1st ed.). New York, USA: Cambridge University Press.

Mohammed, A. J., Yusof, Y., & Husni, H. (2014). A Newton's universal gravitation inspired Firefly algorithm for document clustering. *Proceedings of Advanced in Computer Science and its Applications*, v. 279, Lecture Notes in Electrical Engineering, Jeong, H.Y., Yen, N. Y., Park, J.J. (Jong Hyuk), Springer Berlin Heidelberg, pp. 1259-1264.

Mohammed, A. J., Yusof, Y., & Husni, H. (2014). Weight-based Firefly algorithm for document clustering. *Proceedings of the First International Conference on Advanced Data and Information Engineering (DaEng-2013)*, v. 285, Lecture Notes in Electrical Engineering, Herawan, T., Deris, M. M., Abawajy, J., Springer Berlin Heidelberg, pp. 259-266.

Murugesan, K., & Zhang, J. (2011). *Hybrid hierarchical clustering : An expermintal analysis* (p. 26). University of Kentucky.

Mustaffa, Z., Yusof, Y., & Kamaruddin, S. (2013). Enhanced Abc-Lssvm for Energy fuel price prediction. *Journal of Information and Communication Technology*, *12*, 73–101. Retrieved from http://www.jict.uum.edu.my/index.php

Picarougne, F., Azzag, H., Venturini, G., & Guinot, C. (2007). A new approach of data clustering using a flock of agents. *Evolutionary Computation, 15*(3), 345–367. Cambridge: MIT Press.

Rashedi, E., Nezamabadi-pour, H., & Saryazdi, S. (2009). GSA: A gravitational search algorithm. *Elsevier, Information Sciences*, *179*(13), 2232–2248.

Rothlauf, F. (2011). *Design of modern heuristics principles and application*. Springer-Verlag Berlin Heidelberg. Retrieved from doi:10.1007/978-3-450-72962-4

Sayed, A., Hacid, H., & Zighed, D. (2009). Exploring validity indices for clustering textual data. *In Mining Complex Data*, *165*, 281–300.

Senthilnath, J., Omkar, S. N., & Mani, V. (2011). Clustering using Firefly algorithm: Performance study. *Elsevier, Swarm and Evolutionary Computation*, *1*(3), 164–171. Retrieved from doi: 10.1016/j.swevo.2011.06.003

Tan, S. C. (2012). Simplifying and improving swarm based clustering. In *IEEE Congress on Evolutionary Computation (CEC)* (pp. 1–8). Brisbane, QLD: IEEE.

Tan, S. C., Ting, K. M., & Teng, S. W. (2011a). A general stochastic clustering method for automatic cluster discovery. *Elsevier, Pattern Recognition*, *44*(10-11), 2786–2799.

Tan, S. C., Ting, K. M., & Teng, S. W. (2011b). Simplifying and improving ant-based clustering. In *Procedia Computer Science* (pp. 46–55).

Tang, R., Fong, S., Yang, X. S., & Deb, S. (2012). Integrating nature-inspired optimization algorithms to K-means clustering. In *Seventh International Conference on Digital Information Management (ICDIM), 2012* (pp. 116–123). Macau: IEEE. Retrieved from doi:10.1109/ICDIM.2012.6360145

TREC. (1999). Text REtrieval Conference (TREC). Retrieved from http://trec.nist.gov

Yang, X. S. (2010). *Nature-inspired metaheuristic algorithms* (2nd ed). United Kingdom: Luniver press.

Yujian, L., & Liye, X. (2010). Unweighted multiple group method with arithmetic mean. In *IEEE Fifth International Conference on Bio-Inspired Computing: Theories and Applications (BIC-TA)* (pp. 830–834). Changsha: IEEE. Retrieved from doi:10.1109/BICTA.2010.5645232

Zhong, J., Liu, L., & Li, Z. (2010). A novel clustering algorithm based on gravity and cluster merging. *Advanced Data Mining and Applications*, *6440*, 302–309. Retrieved from doi:10.1007/978-3-642-17316-5_30