Article

JIS

Journal of Information Science
2017, Vol. 43(2) 275–292

DOI: 10.1177/0165551516638784
journals.sagepub.com/home/jis
SAGE

An improved ant algorithm with LDA-based representation for text document clustering

Aytug Onan
Celal Bayar University, Turkey

Hasan Bulut
Ege University, Turkey

Serdar Korukoglu
Ege University, Turkey

Abstract
Document clustering can be applied in document organisation and browsing, document summarisation and classification. The identification of an appropriate representation for textual documents is extremely important for the performance of clustering or classification algorithms. Textual documents suffer from the high dimensionality and irrelevancy of text features. Besides, conventional clustering algorithms suffer from several shortcomings, such as slow convergence and sensitivity to the initial value. To tackle the problems of conventional clustering algorithms, metaheuristic algorithms are frequently applied to clustering. In this paper, an improved ant clustering algorithm is presented, where two novel heuristic methods are proposed to enhance the clustering quality of ant-based clustering. In addition, the latent Dirichlet allocation (LDA) is used to represent textual documents in a compact and efficient way. The clustering quality of the proposed ant clustering algorithm is compared to the conventional clustering algorithms using 25 text benchmarks in terms of F-measure values. The experimental results indicate that the proposed clustering scheme outperforms the compared conventional and metaheuristic clustering methods for textual documents.

Keywords
Latent Dirichlet allocation; text clustering; text mining

1. Introduction

Clustering (also known as cluster analysis) is an important data analysis technique which divides data into groups of similar objects based on a similarity measure such that each cluster contains objects that are similar to other objects within the same cluster and dissimilar to objects out of the cluster [1]. Clustering is an interdisciplinary research field which combines fields, such as statistics, pattern recognition and machine learning [2]. The main difference between clustering and classification comes from the unsupervised nature of the clustering process. Clustering has been successfully applied in many different scientific disciplines, such as marketing, geography, sociology and biology. Since it has been applied in so many different fields, there are lots of different techniques and algorithms in the literature devoted to the clustering. Clustering algorithms can be broadly categorised into two classes as hierarchical clustering algorithms and partitional clustering algorithms based on the obtained clusters [3]. Hierarchical clustering algorithms group data with a sequence of nested partitions, either from singleton clusters to a cluster including all individuals or vice versa. The former is called agglomerative hierarchical clustering and the latter is referred as divisive hierarchical clustering [4]. In contrast,

Corresponding author:
Hasan Bulut, Department of Computer Engineering, Ege University, 35100, Izmir, Turkey.
Email: hasan.bulut@ege.edu.tr

partitional clustering algorithms construct k clusters from a set of data points without using any hierarchical structure, such that each group contains at least one object and each object belongs to exactly one group [1]. Partitional clustering algorithms can be classified into two categories, namely hard and soft partitional clustering [5]. The requirement of assigning each object to only one cluster is relaxed in fuzzy partitional clustering algorithms [1]. K-means and fuzzy c-means algorithms are two widely used representatives of hard and fuzzy partitional clustering algorithms, respectively [5]. In partitional clustering algorithms, data objects are divided into several clusters, where points are iteratively reassigned between k clusters. Given the number of clusters (k), the partitional clustering algorithms aim to find a partition of clusters so that the chosen criterion (such as minimizing the sum of squared error) is optimized. Partitional clustering algorithms require the enumeration of all possible partitions to achieve the global optimum [6]. Since checking all possible partitions is not computationally feasible, some heuristic methods are used in partitional clustering. These methods include K-means, K-modes and K-medoids based algorithms (such as PAM, CLARA and CLARANS methods) [1].

The clustering process can be modelled as an optimisation problem [7]. Metaheuristic algorithms are well established, recognised and efficient methods to solve optimisation problems. Metaheuristic algorithms can be broadly divided into two groups as single-solution based and population-based methods [8]. Tabu search, simulated annealing and local search are some representatives of single-solution based metaheuristics, whereas genetic algorithms, particle swarm optimisation, ant colony optimisation and artificial bee colony optimisation are some representatives of population-based metaheuristics. The conventional clustering algorithms can suffer from several shortcomings such as being very sensitive to the initial state and converging to a local optimum [9]. In contrast, the stochastic characteristics of metaheuristics make them viable techniques in exploring the search space and identifying the optimal partitions [9]. Hence, ant colony optimisation, particle swarm optimisation and hybrid evolutionary algorithms have been successfully utilised in clustering [9]. A detailed survey of ant-based clustering algorithms and other hybrid evolutionary algorithms can be found in [9, 11].

Document clustering (also known as text clustering) is an important field in text mining. In document clustering, clustering techniques are applied to textual data so that documents can be organised, navigated, summarised or retrieved in an efficient manner [12]. The Web is a rich and widely distributed source of information with a progressively expanding volume of data [13]. As a result, the volume of electronic text documents available has been progressively increasing. Document clustering is one technique to identify useful knowledge from textual documents. Document clustering can be utilised in document organisation, systematic browsing of documents, corpus summarisation and document classification [14]. The clustering of text documents can be a challenging task due to the high dimensionality and the sparsity of features [15]. The vector space model is one of the most widely utilised representation schemes in text mining. However, this scheme suffers from high dimensionality of feature vectors [12]. Hence, feature selection methods or dimension reduction methods are generally utilised in text mining applications. Topic models can also be used for dimensionality reduction in text collections. The latent topics in documents are identified with the use of topic modelling. In this way, the curse of dimensionality problem in text corpora can be eliminated, while enhancing the classification or clustering quality.

Taking all these issues into account, this paper presents an improved ant clustering based method for document clustering. In the proposed clustering scheme, a hybrid ant-based clustering algorithm (referred to as AntClass) is utilised as the base model [16]. This algorithm is a hybrid clustering algorithm which combines the stochastic and exploratory features of ant colony optimisation with the deterministic and heuristic features of K-means algorithm [16]. One of the main problems encountered by the algorithm is that the number of clusters (referred to as heaps in the algorithm) generated when the algorithm terminates is not relatively close to the number of clusters in the original dataset. The algorithm tends to generate more clusters (heaps). In order to enhance the clustering quality, two heuristic approaches for merging heaps are proposed in conjunction to AntClass algorithm. The improved ant-clustering based scheme is applied to a document clustering domain with the latent Dirichlet allocation (LDA) based representation of text documents. The experiments are conducted on 25 text benchmarks.

The rest of the paper is organised as follows. Section 2 briefly reviews the existing work on metaheuristic clustering in textual documents. Section 3 discusses the motivation and contribution of the study. Section 4 presents the LDA model. Section 5 briefly describes the clustering algorithms used in the experimental evaluations. Section 6 presents the proposed clustering scheme. Section 7 presents the experimental results and we provide the concluding remarks in Section 8.

2. Literature review

This section briefly reviews the use of metaheuristic algorithms and the LDA in document clustering.

2.1. Metaheuristics in clustering

Song and Park [17] presented a clustering scheme based on genetic algorithm and the latent semantic indexing. In this scheme, the latent semantic indexing is utilised for representing documents due to the high dimensionality of feature space in textual data. As a genetic algorithm, a variable string length model is used for clustering task. The experimental results on textual benchmark indicated that the utilisation of genetic algorithm in conjunction with the latent semantic indexing can enhance the clustering quality on textual documents. Hasanzadeh et al. [18] presented a particle swarm optimization based approach for text document clustering. In the proposed scheme, the latent semantic indexing is utilised for obtaining a corpus-based document representation. In particle swarm optimisation, an adaptive inertia weight is used to enhance the exploration and exploitation of search space and the convergence rate. The clustering algorithm is empirically evaluated with different dimensions of feature space. The experimental results indicated that the latent semantic indexing can enhance the clustering quality compared to vector space model. Besides, particle swarm optimisation can yield better results compared to K-means clustering algorithm.

Vaijayanthi et al. [19] presented a hybrid evolutionary clustering algorithm for text document clustering. In the proposed method, vector space model with TF-IDF weighting is utilised to represent text documents. The presented hybrid clustering algorithm combines ant colony optimisation, tabu search and K-means algorithms. First, K-means algorithm is applied on the textual data. The results obtained by the algorithm are regarded as the initial positions for the ants. Besides, a tabu list is used to prevent ants from revisiting the same documents.

Dziwinski et al. [20] introduced a fully controllable novel ant colony algorithm for text document clustering. To represent text documents, vector space model is utilised. The presented clustering algorithm is a variant of basic ant clustering model with a basic heuristic decision function to improve the convergence of the algorithm significantly.

Azaryuon and Fakhar [21] presented an improved ant clustering based algorithm for text document clustering. To represent text documents, vector space model with TF-IDF weighting is utilised. In this algorithm, a heuristic based approach is proposed to guide the movements of ants. The performance of the proposed clustering algorithm is compared to K-means algorithm and conventional ant-based clustering in terms of F-measure.

Avanija and Ramar [22] presented a hybrid evolutionary scheme for clustering text documents. In this scheme, an ontology based clustering algorithm is utilised in conjunction with the semantic similarity measure and particle swarm optimisation. Document clustering is utilised for recovering relevant documents. The experimental comparisons with K-means indicate the superiority of the proposed scheme for text domain.

Forsati et al. [23] presented a hybrid evolutionary algorithm for document clustering. The presented approach combines the exploratory features of harmony search with the refining features of K-means algorithm. In this way, the dependability of K-means algorithm to the parameters, such as initial cluster centres, is reduced and the entire search space can be searched within a reasonable time.

Cagnina et al. [24] presented a particle swarm optimisation based algorithm for clustering short text documents. A discrete particle swarm optimisation algorithm (referred to as CLUDIPSO) is utilised as the base algorithm. The algorithm is enhanced with the use of an efficient representation of particles, an improved evaluation function and a modified mutation operator. The experimental results are conducted on short texts, such as scientific abstracts, news and short legal documents.

Forsati et al. [25] developed an improved bee colony optimisation algorithm for text document clustering. In the presented approach, cloning and fairness are integrated into the bee colony optimisation algorithm to improve the exploration and exploitation capabilities. Besides, the improved bee colony optimisation algorithm is hybridised with K-means algorithm in several different ways.

Judith and Jayakumari [26] presented a hybrid evolutionary algorithm which combines particle swarm optimisation and K-means algorithms for document clustering. In this scheme, latent semantic indexing is utilised for text representation and MapReduce is utilised for distributed processing of the algorithm.

Song et al. [27] presented a quantum-behaved particle swarm optimisation algorithm for text document clustering. In this scheme, an improved position updating approach is developed to confine the search areas of particles to a particular range. The experimental results are conducted on several benchmark datasets and models are evaluated in terms of F-measure values.

2.2. Latent Dirichlet allocation in clustering

Probabilistic topic models can be used to summarise large text collections based on the distribution of words over topics and distribution of topics over documents [28–31]. The basic utilisation of probabilistic topic modelling methods is to identify main topics of text documents. In addition to this basic utilisation, LDA and other probabilistic topic modelling

approaches have been widely applied in a large number of natural language processing applications, e.g. for the temporal topic detection [32], attribute authorship to text documents [33], summarising the opinions about product reviews [34], understanding topic evolution [35], detecting aspects in review documents [36], building a knowledge organisation system [37], dividing text documents into semantically coherent segments [38], fine-grained sentiment analysis [39] and identifying the hidden topic structures of changes in dynamic text collections [40].

Newman and Block [41] examined the performance of the probabilistic latent semantic analysis, the latent semantic analysis and K-means algorithms for identifying topics and topic trends in a series of text documents. The experimental analysis indicates that the probabilistic latent semantic analysis is a viable tool for identifying meaningful topics in a large corpus.

Omar et al. [42] presented a LDA based approach for topic representation. In this scheme, LDA is utilised to extract topics. Then, thesaurus and corpus-based similarity measures are utilised to compute words' similarities. Based on these similarities, the descriptive features for topics are obtained. Besides, human annotators provide coherence-based topic annotation. The topic coherence obtained from annotators is used as the class labels. Based on word similarity features and topic annotation, two datasets are prepared. The proposed scheme is evaluated on topic classification using classifiers, such as support vector machines, K-nearest neighbour and Random Forest.

The LDA method can also be utilised to cluster text documents. The rest of this section presents the review of the applications of the LDA method with a special emphasis on document clustering.

Rafi et al. [43] presented a document clustering model based on topic modelling. In this model, topic modelling was utilised to represent text documents in an efficient way. In this way, the dimensionality of text documents was reduced, while the semantic relations of the text can be captured.

Ma et al. [44] presented a three-phased scheme for text document clustering. In this scheme, the text document was represented by topics obtained by the LDA method. The most significant topics were identified based on the significance degrees of topics. Then, the initial centres of clusters were determined by the K-means++ algorithm. Finally, the K-means algorithm was utilised to cluster documents based on the latent topics.

Yau et al. [45] examined the performance of the LDA method and its variants (the hierarchical LDA, correlated topic models and hierarchical Dirichlet process) for clustering scientific documents.

Xu et al. [46] presented a hierarchical LDA based method for clustering text documents. In the conventional bag-of-words representation of text documents, the relationship between important terms and co-occurrence relations cannot be captured. Hence, the model presented in [43] utilises the LDA for representation of text documents.

Qiu and Xu [47] presented a word clustering method based on the LDA and K-means algorithms. In this model, the LDA is used to extract topics from the texts and the centroids of the K-means algorithm are selected among the nouns with the highest probability values.

Tang et al. [48] examined the performance of the vector space model and the LDA method for cross-lingual document clustering.

Vulic et al. [49] presented a multilingual probabilistic topic model based on the LDA. They applied the presented method to a several real-life tasks, such as cross-lingual event-centred news clustering, cross-lingual document classification, cross-lingual semantic similarity and cross-lingual information retrieval.

Savoy [50] applied the principal component analysis, the part-of-speech (POS) frequencies and topical features for clustering and authorship attribution to the state of the union addresses. In the topical features based text clustering, the POS frequencies were utilised to derive similarities between the styles of presidents. In the stylistic features based text clustering, an inter-textual distance measure was applied. Finally, an authorship attribution was obtained by clustering the styles of the different presidents.

3. Motivation and contribution of the study

As mentioned in advance, the high dimensionality is an important challenge encountered in text document clustering. In order to tackle the problem of high dimensionality, several methods are proposed, such as obtaining a lower-dimensional representation for the text datasets and working on a transformed space. For instance, the non-negative matrix factorisation method can be used to approximate the term-document matrix [51]. Unsupervised dimensionality reduction methods can be used in conjunction with the clustering methods. For instance, four dimension reduction techniques (i.e. independent component analysis, latent semantic indexing, document frequency and random projection) have been examined for text document clustering [52]. Ding and Li [53] combined linear discriminant analysis and the K-means algorithm for adaptive dimension reduction in text document clustering. Zhu and Allen [54] presented a latent semantic indexing based approach for representing the text documents in a semantically coherent way. Ma et al. [42] utilised the LDA for representing text documents as a collection of topics. Recent studies on the use of metaheuristic clustering algorithms on text

domain indicate that metaheuristics can yield promising results for document clustering [17–27]. Though the use of the LDA method and metaheuristic clustering algorithms takes great research attention in the literature, the number of works that utilises the LDA for representing text documents in text document clustering is very limited. To fill this gap, this paper presents an empirical analysis and benchmark results for 25 widely utilised text datasets in text document clustering with LDA-based text representation scheme. Besides, the paper presents an enhanced ant-based clustering scheme. To our knowledge, this is the first comprehensive analysis of LDA-based representation and metaheuristic clustering methods in text document clustering.

4. The latent Dirichlet allocation

The LDA is a very popular generative probabilistic topic model, where each document is represented as a random mixture of latent topics and each topic is represented as a distribution over fixed set of words [55]. In LDA, each document can exhibit multiple topics with different degrees. In LDA, the words in each document are the observed data and the main objective is to infer the underlying latent topic structure based on the observed data. For each document in the corpus, the words are generated with a two-staged procedure. First, a distribution over topics is randomly chosen. Then, for each word of the document, a topic from the distribution over topics is randomly chosen and a word from the particular distribution is randomly chosen [56]. LDA can be modelled as a three-level Bayesian graphical model. This graphical model of LDA is presented in Figure 1. In Figure 1, nodes are random variables and edges represent possible dependencies between the variables. In this representation, α denotes the Dirichlet parameter, Θ denotes document-level topic variables, z denotes per-word topic assignment, w denotes the observed word and β denotes topics. As seen from the three-layered representation in Figure 1, α and β parameters are sampled only once while generating the corpus, document-level topic variables are sampled for each document and word-level variables are sampled for each word of the document [56]. In LDA, a word is a discrete data from a vocabulary indexed as $\{1, \ldots, V\}$. A document is a sequence of N words $w=(w_1, w_2, \ldots, w_n)$. A corpus consists of M documents and represented as $D=\{w_1, w_2, \ldots, w_M\}$. Based on this notation, the generative process of LDA is summarised in Figure 2.

The generative procedure of LDA indicates a joint distribution over random variables. The probability density function of a k-dimensional Dirichlet random variable, the joint distribution of a topic mixture and the probability of a corpus are computed using Equation 1, Equation 2 and Equation 3, respectively [55]:

$$p(\Theta|\alpha) = \frac{\Gamma(\sum_{i=1}^{k}\alpha_i)}{\prod_{i=1}^{k}\Gamma(\alpha_i)}\Theta_1^{\alpha_1-1}\ldots\Theta_k^{\alpha_k-1} \tag{1}$$

$$p(\Theta, z, w|\alpha, \beta) = p(\Theta|\alpha)\prod_{n=1}^{N}p(z_n|\Theta)p(w_n|z_n, \beta) \tag{2}$$

$$p(D|\alpha, \beta) = \prod_{d=1}^{M}\int p(\Theta_d|\alpha)\left(\prod_{n=1}^{N_d}\sum_{Z_{dn}}p(z_{dn}|\Theta_d)p(w_{dn}|z_{dn}, \beta)\right)d\Theta_d \tag{3}$$

Given a document, the main inferential problem of LDA is to compute the posterior distribution of the hidden variables. The computation of the posterior distribution of the hidden variables for exact inference is an intractable problem. In order to deal with this intractability, approximation algorithms are widely utilised as the inference methods. These

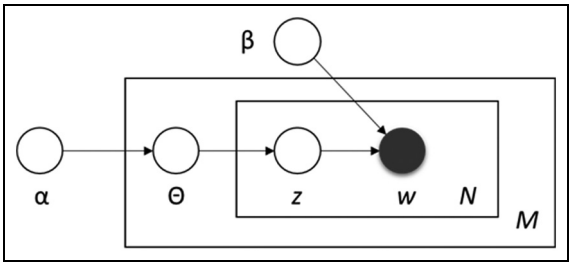

Figure 1. The graphical model of LDA [55].

For each document w in a corpus D:

(1) Choose $N \sim$ Poisson (ξ).
(2) Choose $\Theta \sim$ Dir (α).
(3) For each of the N words w_n:

(3.1) Choose a topic $z_n \sim$ Multinomial (Θ).

(4) Choose a word w_n from $p(w_n|z_n, \beta)$, a multinomial probability conditioned on the topic z_n.

Figure 2. The generative process of LDA [55].

methods include Laplace approximation, variational approximation, Gibbs sampling and Markov chain Monte Carlo [55, 57].

5. Clustering algorithms

This section briefly presents the clustering algorithms utilized in the experimental evaluations. In order to evaluate the clustering quality of the proposed clustering scheme, three conventional clustering algorithms (K-means, K-means++ and expectation maximisation) and two metaheuristic clustering algorithms (ant-based clustering and particle swarm optimisation based clustering) are taken into account.

5.1. K-means algorithm

The K-means algorithm is one of the most popular algorithms used in clustering due to its easy implementation, simplicity and efficiency [58]. The algorithm takes the number of clusters as input parameter and initiates with the random selection of k objects as cluster centres. The remaining objects are assigned to the clusters with the closest centres. The algorithm continues to compute new mean of clusters till stopping criterion. One critical issue in clustering is the determination of an appropriate similarity measure to identify the degree of closeness or separation between the target values [59]. Euclidean distance, cosine similarity, Jaccard coefficient, Pearson correlation coefficient and some other metrics can be used to compute the distance. The K-means algorithm gives good results, especially for clusters with well-aligned and compact shapes [1]. Though it is an efficient and scalable algorithm, it suffers from several weaknesses, such as being able to deal with only numerical attributes, being highly dependent on the position of randomly selected initial cluster centres [4]. The algorithm is very sensitive to data with noise and outliers [59]. Besides, the algorithm can terminate at a local optimum solution [61].

5.2. K-means++ algorithm

The clustering quality obtained by the K-means algorithm may be greatly affected based on the selection of initial cluster centres. In this regard, the K-means++ algorithm uses a heuristic to find the initial centres for the K-means algorithm. This heuristic utilises a probability distribution obtained from the distances of data points to the already selected initial centres [62]. Then, following cluster centres for each cluster is chosen among the other data points in the dataset based on the probability proportional to its squared distance from the point's closest existing cluster centre. Once the selection of k centres is completed, the process proceeds with the use of the standard K-means algorithm.

5.3. Expectation maximisation algorithm

Expectation maximisation (EM) is an iterative refinement algorithm for clustering [63]. It finds maximum likelihood parameter estimates in the probabilistic methods. EM consists of two main steps, namely expectation and maximisation steps. In clustering, EM assigns each object to a particular cluster based on weight values indicating the probability of membership [1]. In EM, a finite Gaussian mixtures model is utilised to estimate parameters set in an iterative manner [64]. As mentioned in advance, the method iterates in two steps for clustering. First, the initial guess of parameter vectors is obtained. Then, these parameters are iteratively refined in expectation and maximisation steps by computing the

probability of membership for each instance according to the initial parameters and re-estimating the parameter values according to the adjusted parameter values [1, 64].

5.4. Particle swarm optimisation

Particle swarm optimisation (PSO) is a stochastic population-based metaheuristic algorithm, which is inspired from the social behaviour of organisms, such as birds within a flock [65]. In PSO, each candidate solution is regarded as a particle in the search space. In the basic model of PSO, the optimisation problem is solved by examining the search space via the particles of the swarm. In this model, each particle has its own position and velocity. The position represents the direction of a particle, whereas the velocity corresponds to the current step. The position of each particle is updated based on its best position and the best position of the swarm [8]. In this way, the search behaviour of a particle is modified in a cooperative way [66]. In the basic model of PSO-based clustering, a fixed set of clusters is used and a set of cluster centres are determined by examining the search space with particles [67]. There are several variants of PSO for clustering [9].

5.5. Ant-based clustering

The basic model to use ant-based algorithms in cluster analysis and classification was introduced by Deneubourg et al. [68]. In this model, ant behaviour is modelled for clustering and classification. Ants move randomly and pick up or drop off an item according to the density of environment. The probability of picking up an item is high when the item is in a less dense area and the probability of dropping an item is high when the item is in a dense area with similar objects that should be clustered within the same cluster [66]. The probability of picking up and dropping off an item is determined by the following equalities, respectively [68]:

$$P_{pick} = \left(\frac{k_p}{(k_p + f)} \right)^2 \tag{4}$$

$$P_{drop} = \left(\frac{k_d}{(k_d + f)} \right)^2 \tag{5}$$

where f is the number of similar items in the neighbourhood perceived by the ant, k_p is the pick-up constant and k_d is the drop-off constant. When the number of similar items in the neighbourhood perceived by the ant is high, it is unlikely to pick up the item. Similarly, the probability of dropping off an item is high when the number of similar items in the neighbourhood is high. The following subsections briefly present two ant-based clustering algorithms extending the basic model. The basic model presented in [68] is extended by Lumer and Faieta [69]. In this clustering scheme, data objects are scattered randomly into a two-dimensional grid such that a grid cell contains at most one data object. The movements of an ant are randomly determined. An ant makes a decision on picking up and dropping off an item i based on the following probabilities [69]:

$$P_{pick}(i) = \left(\frac{k_p}{(k_p + f(i))} \right)^2 \tag{6}$$

$$P_{drop}(i) = \begin{Bmatrix} 2f(i), if\, f(i) < k_d \\ 1 \qquad otherwise \end{Bmatrix} \tag{7}$$

where k_p and k_d are the pick-up and the drop-off constants, respectively, and $f(i)$ is local density function given as by Equation 8 [69]:

$$f(i) = \begin{Bmatrix} \frac{1}{d^2} \sum_j \left(1 - \frac{d(i,j)}{\alpha} \right), \; if\, f > 0 \\ 0 \qquad\qquad\qquad otherwise \end{Bmatrix} \tag{8}$$

where d^2 is the size of ant's neighbourhood, $d(i,j)$ is the distance between data objects i and j, and α is the data-dependent scaling parameter [60].

6. An improved ant clustering algorithm based on heaps merging

The improved ant clustering algorithm presented in this paper enhances AntClass algorithm by heap-merging approaches. Hence, we first present the basic concepts of AntClass algorithm. It is a hybrid method which combines the stochastic and exploratory features of ant colony optimisation with the deterministic and heuristic features of the K-means algorithm to improve the convergence. The algorithm mainly consists of four steps. Initially, the algorithm starts with an ant-based algorithm for clustering objects. This step is followed by the K-means algorithm. Then, the ant-based clustering algorithm is applied once again and the algorithm terminates with applying the K-means algorithm on objects once more [16]. In this scheme, each data contains a vector of n real values. The distance between two data objects is measured by a distance measure. Data objects are scattered randomly on a grid similar to other ant-based clustering algorithms. One basic difference of the algorithm is that ants can create, build or destroy heaps. Heap is a data structure which contains two or more objects. A heap can be located on a single grid cell. Given a particular heap H consisting of n_H objects, maximum distance between two data objects (denoted as $D_{max}(H)$) is determined as follows [16]:

$$D_{max}(H) = max_{\gamma_i, \gamma_j \in E} D(\gamma_i, \gamma_j) \tag{9}$$

where γ_i and γ_j denote data object i and j, respectively. The centre of mass of all objects in H, $\gamma_{center}(H)$ and the mean distance between all objects in H ($D_{mean}(H)$) are determined by the following equations [16]:

$$\gamma_{center}(H) = \frac{1}{n_H} \sum_{\gamma_i \in H} \gamma_i \tag{10}$$

$$D_{mean}(H) = \frac{1}{n_H} \sum_{\gamma_i \in H} D(\gamma_i, \gamma_{center}(H)) \tag{11}$$

where γ_i refers to a particular data object i. The most dissimilar object in a heap, $\gamma_{dissim}(H)$ is an object which maximises the distance to centre of mass of all objects in the heap. The ant-based clustering stage of AntClass algorithm starts with random initialisation of ants' positions on the grid. Ants can perform several actions depending on the state. Each ant moves at each iteration. If an ant does not carry any object, then eight cells in the ant's neighbourhood are examined and a single object from the neighbourhood is picked up according to the pick-up probability. Otherwise, if an ant carries a data object (γ), then eight cells in the ant's neighbourhood are examined and the object is dropped according to the dropping off probability [16]. The two main actions performed by the ants are picking up and dropping off objects. Since ants are able to build or destroy heaps, there are several conditions to be considered in picking up and dropping off objects. An unladen ant looks for a possible object by examining the eight cells in its neighbourhood and if one object or heap is found, then there are three cases to consider. The first case is that the ant may find one object alone. In this case, data object is picked up according to a fixed probability. The second case is that the ant finds a heap with two objects. In this case, one of the objects from the heap is removed with a probability ($P_{destroy}$). The other case is that the grid cell contains a heap H with more than two objects. In this case, the ant picks up the most dissimilar object in the heap if the condition given by Equation 12 is satisfied [16]:

$$\frac{D(\gamma_{dissim}(H), \gamma_{center}(H))}{D_{mean}(H)} > T_{remove} \tag{12}$$

where T_{remove} represents a threshold parameter for removing an object. Similarly, an ant carrying an object considers eight cells in its neighbourhood and there are three conditions to examine. The cell may not contain any object, may contain only one object or may contain a heap. If the cell is empty, the carried object will be dropped off with a probability (P_{drop}). If the cell contains only one object, carried object will be dropped off and a heap with two objects will be generated if the condition given by Equation 13 is satisfied [16]:

$$\frac{D(\gamma, \tau)}{D_{max}} < T_{create} \tag{13}$$

where τ is the object in the cell, γ is the object carried by the ant and T_{create} is a threshold parameter. If the cell contains a heap H, then the carried object is dropped off if the condition given by Equation 14 is satisfied:

$$D(\gamma, \gamma_{center}(H)) < D(\gamma_{dissim}(H), \gamma_{center}(H)) \tag{14}$$

At the end of the ant-based clustering stage of the AntClass algorithm, there are some objects which are not assigned to any heap at all or wrongly assigned to any heap. In order to overcome this problem, the K-means algorithm is applied to the partition obtained by the ant-based clustering stage. The algorithm operates on grid positions. The first two steps of the AntClass algorithm end with applying ant-based clustering and K-means algorithms. At the end of this step, k heaps of objects are generated. After the K-means stage, the ant-based clustering algorithm is applied once more but instead of single objects, the algorithm runs on heaps. In this step, ants are able to pick up or drop off previously created heaps [16]. Since ants should deal with heaps of objects instead of objects in this stage, the algorithmic steps are adapted so that ants can carry entire heaps of objects [16]. The picking up a heap is handled in the same way as the picking up an object. In this stage, the basic metric of heaps merging is presented to improve the convergence of the algorithm. Based on this metric, two heaps are merged if the distance between the centroid of the heap carried by the ant and the centroid of the heap on the board is less than a threshold value ($T_{createforheap}$) [16]. Kanade [70] presented a new metric for heaps merging in this stage. In this scheme, the heaps are merged if the distance between the centroid of the heap carried by the ant and the centroid of the heap on the board is less than a fixed percent of the mean distance of all the objects in the heap from the centre of the heap on the board. At the end of this step, the K-means algorithm is applied once again and the algorithm terminates. The experimental results indicate that the number of heaps (clusters) generated by the AntClass algorithm at the end of its four steps tends to exceed the actual number of clusters in the original dataset [11, 16]. The motivation of this research stems from this deficiency. To determine the appropriate number of clusters is an essential task for obtaining robust/efficient clustering models [71, 72]. In this section, two different heap merging approaches are presented in order to solve the problem efficiently. The heaps merging approaches presented in [16, 70] are utilised in the third stage of the AntClass algorithm (ant-based clustering). However, the approaches presented in Sections 6.1 and 6.2 can be regarded as the further refinements of the clustering results after the application of the four basic stages of the AntClass algorithm. The heaps merging approaches presented in [16, 70] determine whether the heap carried by a particular ant and the heap on the board should be merged or not. In contrast, the approaches presented in Section 6.1 (HeapsMerge1) and Section 6.2 (HeapsMerge2) consider all the heaps generated at the end of the fourth stage of the AntClass algorithm to improve the quality of the final clustering results. The first approach determines the heap merging based on the most dissimilar objects and the other approach determines the heap merging based on any objects in heaps [73]. To summarise, the basic metric of heaps merging is applied at the third stage of the AntClass, HeapsMerge1 and HeapsMerge2. However, the heaps obtained at the end of the AntClass algorithm are further merged in HeapsMerge1 and HeapsMerge2 according to the schemes presented in Figures 4 and 5, respectively.

6.1. HeapsMerge1: The most dissimilar objects based heaps merging

In this approach, the first four steps of the AntClass algorithm are preserved. The distances between centres of heaps (clusters) generated by the K-means stage are determined. The process starts with heaps with the closest centres. The pair of heaps are sorted according to the pairwise distance of centres. Let H_1 and H_2 denote two heaps to be examined for merging and $\gamma_{center}(H_1)$ and $\gamma_{center}(H_2)$ denote the centres for corresponding heaps, $\gamma_{dissim}(H_1)$ and $\gamma_{dissim}(H_2)$ represent the most dissimilar objects in each heap. These two heaps are merged into one heap only if the conditions given by Equation 15 and Equation 16 are satisfied:

$$D(\gamma_{center}(H_1), \gamma_{center}(H_2)) < D(\gamma_{dissim}(H_1), \gamma_{center}(H_1)) \tag{15}$$

$$D(\gamma_{center}(H_1), \gamma_{center}(H_2)) < D(\gamma_{dissim}(H_2), \gamma_{center}(H_2)) \tag{16}$$

The merging criterion for the HeapsMerge1 approach with two clusters is illustrated in Figure 3, where blue spots and red spots denote objects of two different heaps (H_1 and H_2), green points indicate the centres of clusters and γ_{dissim} is used to represent the most dissimilar objects of heaps. The steps of the algorithm are presented in Figure 4. The algorithm continues until there are no heaps generated by the K-means stage left satisfying the required conditions for merging, i.e. the new generated heaps by merging the other heaps are not taken into account for further processing. HeapsMerge1 is a heuristic approach which was developed to deal with an excessive heap generation problem. Heuristic approaches are viable tools for solving problems in artificial intelligence and optimisation. This approach is developed based on the observations obtained by experimental datasets. In order to formulate this heuristic approach, an extensive empirical analysis on datasets has been conducted.

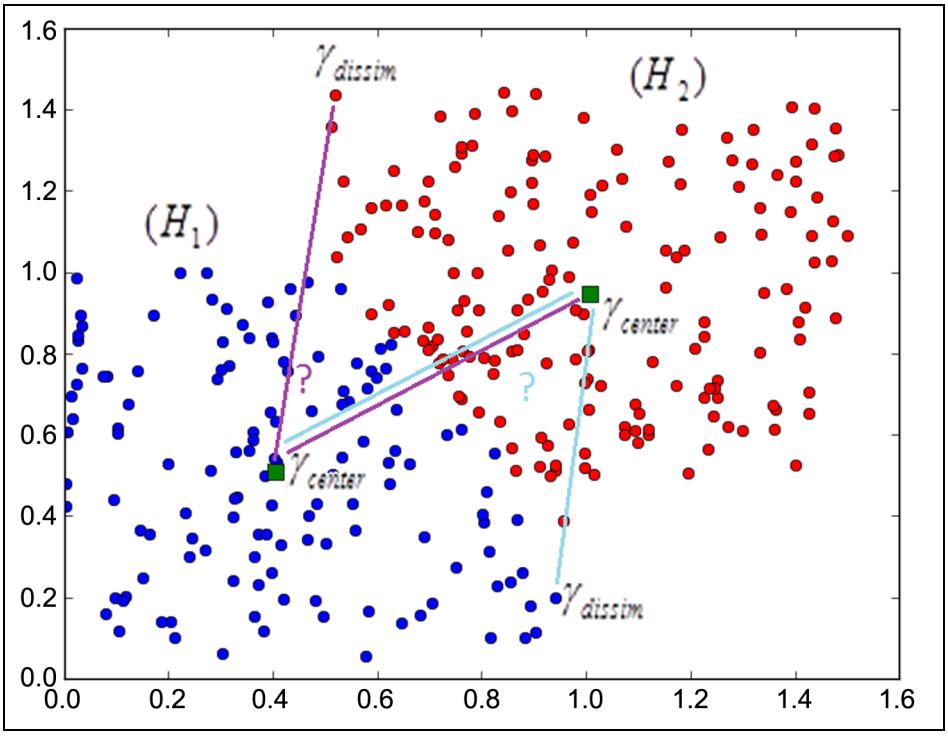

Figure 3. An illustration for HeapsMerge1.

1. Take the partition of dataset generated by K-means algorithm and k heaps (H_1,…, H_k),

2. Repeat until there are any heaps satisfying the required conditions for merging given by Eq. (15) and Eq. (16):
- (a) Compute the center of each heap ($\gamma_{center}(H_1)$,…, $\gamma_{center}(H_k)$),
- (b) For each heap, compute the distance between the most dissimilar object in the heap and center of the heap,
- (c) Compute the distances between the centers of heaps,
- (d) Sort heaps according to distances between centers of heaps in ascending order,
- (e) Starting with heaps with the closest centers, merge two heaps into one heap only if Eq. (15) and Eq. (16) are valid for these heaps and proceed with other heaps in the order of closer centers,

Figure 4. The general structure of HeapsMerge1.

6.2. HeapsMerge2: Object–centre distances based heaps merging

To deal with the problem of excessive heap generation, we introduce another heap merging approach based on distances among the objects and the centres of heaps. The AntClass algorithm contains two consecutive steps of ant-based clustering and the K-means algorithm. Due to this scheme, an object already assigned to a particular cluster with the closest centre may be deployed in a worse cluster. Hence, HeapsMerge2 aims to place objects to the closest clusters by examining the distance between heaps and cluster centres. Similar to the other heap merging approach presented (HeapsMerge1), the first four steps of the AntClass algorithm are preserved and the heaps generated by the K-means algorithm are kept for further processing. Let H_1,…, H_k be the heaps generated by the earlier steps of the algorithm, each with a number of objects. The basic merging criterion is merging the two heaps if there are a number of objects in any heaps whose distances to their heap's centre are larger than their distances to other heap's centre. For each heap, several closer heaps are examined for merging. One critical issue here is to determine the appropriate number of objects satisfying the merging criterion and the appropriate number of heaps in the neighbourhood to be examined for merging. Since the characteristics, attributes and number of clusters differ in various datasets, the number of heaps generated and the number of objects in each heap may vary accordingly. For the datasets presented in the experimental design, the number of objects, the

1. Take the partition of dataset generated by K-means algorithm and k heaps (H_1,…, H_k),
2. Determine the size of the neighborhood based on number of heaps (k) obtained:
- If k<8: All heaps generated will be taken as the neighborhood.
- If 8<k<16: The neighborhood size is equal to 8.
- If k>16: The neighborhood size is equal to 16.

3. Determine the number of objects in the heap with maximum objects ($n_{H_{max}}$) and the average number of objects in heaps (n_{average}). Calculate the appropriate number of objects for merging criterion by Equation (17).
4. Repeat until there are any heaps satisfying the required conditions for merging:
(a) Calculate the distance of each object in a heap to its center and the center of another heap,
(b) Compare distances calculated in (a).
(c) If there are at least n_{appr} objects whose distances to their heap's center are larger than their distances to the other heap's center, then merge two heaps. Otherwise, proceed with next heap.

Figure 5. The general structure of HeapsMerge2.

number of heaps, the average heap sizes and the largest heap sizes after the two consecutive steps of the algorithm for each dataset is determined. In order to obtain a better approximation to the number of objects for merging criterion, a number of different configurations (with different number of heaps and neighbourhood sizes) are examined. In HeapsMerge2, two heaps are merged if a number of objects (denoted by n_{appr}) satisfy the requirement of having larger distances to their heap's centre compared to the heap's centre of the other examined heap. In order to formulate the number of objects for merging criterion (n_{appr}), different values of n_{appr} are evaluated based on the results obtained from the datasets presented in Section 7.1. Based on the observations on the results of HeapsMerge2 for different values of n_{appr}, a relation between the number of objects in heap with maximum objects ($n_{H_{max}}$) and the average number of objects in heaps ($n_{average}$) is formulated. The merging criterion is formulated as given by Equation 17:

$$n_{appr} = \frac{n_{H_{max}} - n_{average}}{5} \tag{17}$$

where n_{appr} is number of objects for merging criterion, $n_{H_{max}}$ is number of objects in heap with maximum objects and $n_{average}$ is average number of objects in heaps. The proposed heaps merging approach based on object–centre distances is presented in Figure 5.

7. Experimental design and results

We conduct a set of experiments with Java on a computer with Intel Core i7 CPU 3.40 GHz with 8.00 GB RAM. The experimental results of conventional clustering algorithms, metaheuristic based clustering algorithms and the proposed clustering schemes (HeapsMerge1 and HeapsMerge2) on 25 text collections are examined based on F-measure. To set the number of cluster parameters in the K-means, K-means++ and expectation maximisation clustering algorithms, the actual number of clusters are given as the initial parameters. The cluster numbers and other characteristics of datasets utilised in experimental evaluations are presented in Section 7.1. The basic parameters of ant-based clustering algorithm and their initial values are selected according to [16]. The basic parameter values of particle swarm optimisation based clustering are selected according to [25]. In the algorithms, cosine similarity is utilised to compute the distance between two vectors owing to its widespread use in text domain. The results reported in the empirical analysis present the average results of 10 different runs for each algorithm.

7.1. Datasets

In order to evaluate the effectiveness of proposed clustering schemes on text document clustering, we have used 25 text benchmarks from several domains, such as sentiment analysis, emails, scientific documents and news articles. In Table 1, the descriptive information regarding the text collections utilised in the experimental analysis is presented. The number of features listed in Table 1 presents the number of terms extracted when the vector space model scheme is utilised to represent text documents [74]. As mentioned in advance, the clustering of text documents can be difficult due to the high dimensionality of the feature space in text domain. In order to tackle this problem, we have modelled each text collection with the use of LDA and Gibbs sampling. In this representation, text documents are represented by latent topics. Hence,

Table 1. Descriptive information for the datasets [74].

Dataset	Domain	Documents (n)	Features (n)	Clusters (n)
Multi-Domain-Sentiment	Sentiment analysis	8000	13,360	2
Review-Polarity	Sentiment analysis	2000	15,698	2
SpamAssassin	Emails	9348	97,851	2
SpamTrec-3000	Emails	3000	100,464	2
Irish-Sentiment	Sentiment analysis	1660	8659	3
Classic3	Abstracts	7095	7749	4
CSTR	Scientific	299	1726	4
SyskillWebert	Web pages	334	4340	4
Reviews	News articles	4069	22,927	5
Hitech	News articles	2301	12,942	6
La1s	News articles	3204	13,196	6
La2s	News articles	3075	12,433	6
LATimes	News articles	6279	10,020	6
Tr21	TREC documents	336	7903	6
Tr23	TREC documents	204	5833	6
Tr31	TREC documents	927	10,129	7
Re8	News articles	7674	8901	8
Tr12	TREC documents	313	5805	8
Tr11	TREC documents	414	6430	9
Oh0	Medical documents	1003	3183	10
Oh10	Medical documents	1050	3239	10
Oh15	Medical documents	3101	54,142	10
Oh5	Medical documents	918	3013	10
Ohscal	Medical documents	11162	11,466	10
Tr41	TREC documents	8778	7455	10

different number of features in the range of 50–200 are examined. Based on the empirical analysis, the highest F-measure values are obtained when 50 topics are used as features in each of the text collection. Hence, we provide results for a feature set of 50 topics. In order to build an efficient clustering scheme with LDA-based representation, the identification of coherent topics is essential [44]. Since topic distributions are not always identical, efficient filtering of incoherent (low quality) topics is important. To optimise the topic coherence, we utilised the topic coherence approach presented in Mimno et al. [75].

7.2. Evaluation measure

In order to evaluate the clustering quality of the proposed clustering algorithms, F-measure is used as the evaluation metric. F-measure is an external cluster validation measure based on precision and recall [68]. In external validation, the clustering results are evaluated based on the data that were not used for clustering, such as known class labels. External validation aims to analyse how close a clustering to a reference is [76]. In the context of document clustering, the clustering is measured in terms of how close the clustering results to the class labels assigned in the documents [77]. To summarize, F-measure evaluates how well a set of generated clusters match the actual classes in the dataset [77]. F-measure is a widely utilised technique in the evaluation of metaheuristic-based clustering algorithms [23, 25, 78, 79]. Hence, we utilised this metric as the evaluation measure. It takes both data objects in the original dataset and data objects generated by the clustering algorithm into account. The precision and recall are calculated by Equations 18 and 19, respectively:

$$p(i,j) = \frac{n_{ij}}{n_j} \tag{18}$$

$$r(i,j) = \frac{n_{ij}}{n_i} \tag{19}$$

where each class i (as given by the class labels of datasets) is regarded as the set of n_i items and each cluster j (generated by clustering algorithm) is regarded as the set of n_j items and n_{ij} represents the number of data items of class i within cluster j [73]. F-measure is calculated as follows, where b is set to one for equal weighting of the precision and recall:

Table 2. Average F-measure values for the algorithms on text collections.

Dataset	K-means	K-means++	EM	PSO	AntClass	HeapsMerge1	HeapsMerge2
Multi-Domain-Sentiment	0.502	0.502	0.497	0.605	0.688	*0.712*	**0.743**
Review-Polarity	0.645	0.602	0.591	0.678	0.716	*0.727*	**0.812**
SpamAssassin	0.638	0.608	0.677	0.667	0.685	*0.797*	**0.816**
SpamTrec-3000	0.532	0.454	0.432	0.692	0.714	*0.734*	**0.788**
IrishEcnomic	0.413	0.389	0.477	0.512	0.526	*0.604*	**0.64**
Classic3	0.736	0.746	0.748	0.787	0.728	*0.755*	**0.896**
CSTR	0.382	0.44	0.622	0.725	0.745	*0.768*	**0.796**
SyskillWebert	0.634	0.605	0.625	0.686	0.82	*0.83*	**0.92**
Reviews	0.582	0.654	0.651	0.759	0.768	*0.815*	**0.835**
Hitech	0.374	0.384	0.427	0.608	0.628	*0.642*	**0.718**
La1s	0.442	0.388	0.612	0.608	0.671	*0.721*	**0.733**
La2s	0.405	0.413	0.541	0.646	0.675	*0.731*	**0.737**
LATimes	0.377	0.382	0.482	0.696	0.732	*0.733*	**0.745**
Tr21	0.544	0.523	0.598	0.784	0.812	*0.827*	**0.853**
Tr23	0.447	0.474	0.545	0.685	0.694	*0.712*	**0.727**
Tr31	0.613	0.612	0.691	0.619	0.64	*0.714*	**0.804**
Re8	0.557	0.511	0.556	0.719	0.764	*0.852*	**0.869**
Tr12	0.381	0.405	0.534	0.675	0.712	*0.786*	**0.819**
Tr11	0.631	0.597	0.666	0.672	0.684	*0.698*	**0.691**
Oh0	0.476	0.445	0.573	0.604	0.625	*0.771*	**0.804**
Oh10	0.519	0.532	0.53	0.572	0.552	*0.625*	**0.641**
Oh15	0.464	0.448	0.478	0.512	0.522	*0.621*	**0.665**
Oh5	0.48	0.472	0.488	0.531	0.517	*0.58*	**0.592**
Ohscal	0.467	0.473	0.371	0.422	0.592	*0.598*	**0.664**
Tr41	0.434	0.42	0.589	0.692	*0.721*	0.718	**0.737**

$$F(i,j) = \frac{(b^2+1)p(i,j)r(i,j)}{b^2 p(i,j) + r(i,j)} \tag{20}$$

The range of F-measure is [0, 1]. The higher F-measure value indicates the better quality of clustering [80].

7.3. Results and discussion

The experimental results are obtained from 25 different text datasets. In Table 2, average F-measure values on conventional clustering algorithms (K-means, K-means++ and EM), metaheuristic-based clustering algorithms (PSO for clustering and AntClass algorithm) and the proposed improved ant-based clustering algorithms (HeapsMerge1 and HeapsMerge2) are presented. In Table 2, the best results achieved by a particular algorithm on the datasets are highlighted as boldface, whereas the second highest results are presented in italics. For the clustering algorithms compared in the experimental analysis, the highest F-measure values on text collections are obtained by the proposed improved ant-based clustering scheme, referred to as HeapsMerge2. Except for the Tr41 dataset, the second highest F-measure values are obtained by the proposed ant-based clustering scheme, referred to as HeapsMerge1. For the Tr41 dataset, the second highest F-measure value is obtained by the AntClass algorithm. However, the F-measure values for HeapsMerge1 and AntClass are close to each other. As can be observed from the F-measure values obtained by the ant-based clustering schemes (AntClass, HeapsMerge1 and HeapsMerge2), the performance of these algorithms outperform the conventional clustering algorithms and PSO based clustering for text document clustering. Regarding the results obtained by K-means, K-means++ , EM and PSO, the highest F-measure values are generally obtained by the PSO based algorithm. The PSO based clustering algorithm outperforms the conventional clustering algorithms in 21 out of 25 datasets. For the results of K-means, K-means++ and EM, the highest results among these algorithms vary according to the datasets utilised. However, EM generally yields better results.

The performance of the algorithms in the document collections in terms of the F-measure values presented in Table 2 indicate that metaheuristic-based clustering algorithms outperform the conventional clustering algorithms. This finding supports the results reported in earlier studies in metaheuristic-based clustering [23, 25]. In addition, the approaches presented in this paper (HeapsMerge1 and HeapsMerge2) yield better F-measure values than the conventional clustering

Table 3. Two-way ANOVA test results.

Source	DF	SS	MS	F	P
Dataset	24	0.87639	0.036516	11.16	0.000
Algorithms	6	1.63616	0.272694	83.34	0.000
Error	144	0.47116	0.003272		
Total	174	2.98371			

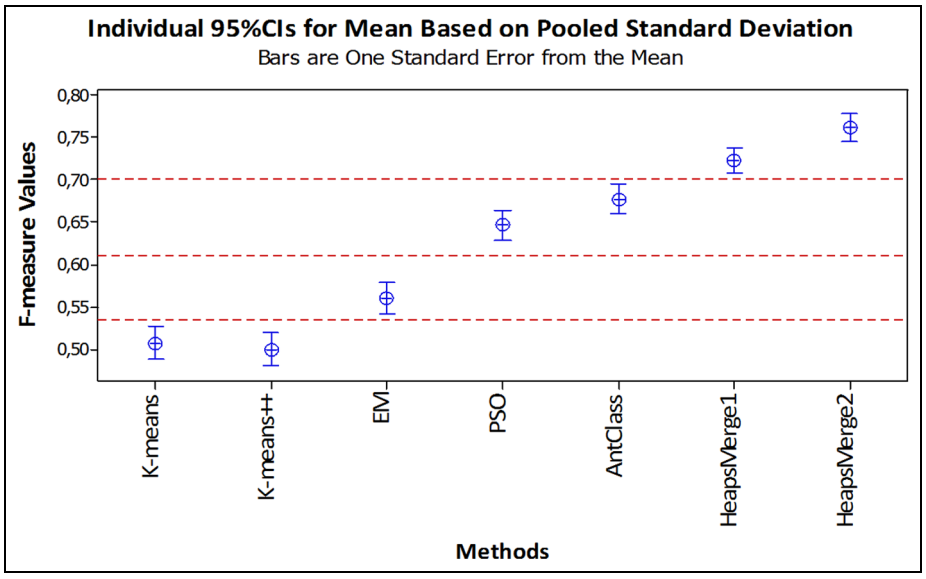

Figure 6. Confidence interval for the compared algorithms.

algorithms and other metaheuristic-based clustering algorithms. Conventional clustering algorithms, such as K-means, can be very sensitive to the initial centres and converge to the local optimum. Metaheuristic algorithms are well-established techniques to find solutions to the optimisation problems. While exploring the search space, they utilise more than one candidate solution. They iteratively improve these candidate solutions by moving to a better nearby solution. Hence, they improve the possibility of finding a better solution in every iteration and if one of the candidate solutions gets trapped in a local optimum, other candidate solutions can be improved by a better nearby solution. Metaheuristic-based clustering algorithms can be utilised as viable methods in text document clustering.

To further evaluate the results obtained in the empirical analysis, we performed two-way ANOVA test in Minitab statistical program. The results for two-way ANOVA test of overall results obtained by the algorithms are summarized in Table 3, where DF, SS, MS, F and P denote degrees of freedom, adjusted sum of squares, mean square, F-statistics and probability value, respectively. Degrees of freedom (DF) are the amount of information in the data. The adjusted sum of squares term (SS Term) represents the amount of variation in the response data that is explained by each term of the model. The adjusted sum of squares error (SS error) represents the variation in the data that the predictors do not reveal. The adjusted sum of squares total (SS Total) represents the total variation in the data. F-statistics (F) is the test statistic for identifying whether a term is associated with the response. Finally, the probability value (P) is used to make a decision about the statistical significance of the terms and model.[1] For the analysis results listed in Table 3, it can be observed that the F-statistics value of 11.6 for datasets indicates that there is a statistically significant difference ($P < 0.001$) for the means of at least two datasets and the F-statistics value of 83.34 for algorithms indicates that there is a statistically significant difference ($P < 0.001$) for the means of at least two algorithms (the critical values for dataset and algorithms are $F_{(24,144,0.01)} = 1.923 < 11.16$ and $F_{(6,144,0.01)} = 2.93 < 83.34$, respectively). The 95% confidence interval for the compared algorithms based on the pooled standard deviation is presented in Figure 6, which supports the statistical analysis results given in Table 3. Based on the statistical significances between the results of algorithms, Figure 6 is divided into four regions denoted by red dashed lines. Hence, the difference between the K-means and K-means++ algorithms is not statistically significant, whereas the F-measure values obtained by the expectation maximisation algorithm are

statistically significant compared to the results of K-means and K-means++. Similarly, the difference between particle swarm optimisation and AntClass algorithm is not statistically significant. As can be observed from Figure 9, HeapsMerge1 and HeapsMerge2 are located in another region of the interval plot. Hence, the higher values obtained by the proposed algorithms are statistically more significant than the results obtained by other algorithms.

8. Conclusion

Document clustering is an important research direction which applies clustering methods to textual documents. Document clustering can be useful in organising, browsing, classifying and summarising documents. Clustering can be modelled as an optimisation problem and metaheuristic algorithms are successfully applied for clustering. In this paper, we presented two new heaps merging approaches based on the most dissimilar objects in heaps and objects-centre distances to enhance the clustering quality of ant-based clustering. The proposed heaps merging approaches solve the problem of generation of excessive number of clusters (heaps) by the AntClass algorithm. Besides, the LDA is utilised to represent text collections as a set of latent topics. The proposed clustering schemes are compared to conventional and metaheuristic clustering algorithms on 25 text benchmarks in terms of F-measure. The experimental results indicate that heaps merging approach based on the most dissimilar objects in heaps outperforms the other clustering methods utilised in the empirical analysis. Besides, the experimental results show that the proposed heap merging approach based on objects-centre distances performs better than the K-means and AntClass algorithms.

Funding

This research received no specific grant from any funding agency in the public, commercial or not-for-profit sectors.

Note

1. http://support.minitab.com/

References

[1] Han J and Kamber M. *Data mining: concepts and techniques*. 2nd ed. San Francisco, CA: Morgan Kaufmann, 2006.
[2] Berkhin P. A survey of clustering data mining techniques. In: Kogan J, Nicholas C and Teboulle M (eds) *Grouping Multidimensional Data*. 1st ed. Berlin: Springer-Verlag, 2006, pp. 25–71.
[3] Alridge M. Clustering: an overview. In: Berry MW and Browne M (eds) *Lecture Notes in Data Mining*. 1st ed. Singapore: World Scientific, 2006, pp. 99–109.
[4] Xu R and Wunsch D. Survey of clustering algorithms. *IEEE Transactions on Neural Networks* 2005; 16(3): 645–678.
[5] Ouadfel S, Batouche M and Ahmed-Taleb A. ACPSO: A novel swarm automatic clustering algorithm based image segmentation. In: Ali S, Abbadeni N and Batouche M (eds) *Multidisciplinary Computational Intelligence Techniques: Applications in Business, Engineering and Medicine*. 1st ed. Hershey, PA: IGI-Global, 2012, pp. 226–239.
[6] Lanzi PL. 'Clustering: partitioning data methods: data mining and text mining lecture notes', Online Referencing, http://www.pierlucalanzi.net (2013, accessed November 2015).
[7] Hruschka ER, Campello RJGB, Freitas AA and De Carvalho ACPLF. A survey of evolutionary algorithms for clustering. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews* 2009; 39(2): 133–155.
[8] Talbi EG. *Metaheuristics from design to implementation*. 1st ed. Hoboken, NJ: Wiley, 2009.
[9] Hasan MJA and Ramakrishnan S. A survey: hybrid evolutionary algorithms for cluster analysis. *Artificial Intelligence Review* 2011; 36: 179–204.
[10] Nanda SJ and Panda G. A survey on nature inspired metaheuristic algorithms for partitional clustering. *Swarm and Evolutionary Computation* 2014; 16: 1–18.
[11] Jafar OAM and Sivakumar R. Ant-based clustering algorithms: a brief survey. *International Journal of Computer Theory and Engineering* 2010; 2: 787–796.
[12] AlSumait L and Domeniconi C. Text clustering with local semantic kernels. In: Berry MW and Castellanos M (eds) *Survey of Text Mining II*. 1st ed. Berlin: Springer-Verlag, 2008, pp. 87–105.
[13] Bhatia MPS and Khalid AK. Information retrieval and machine learning: support technologies for web mining research and practice. *Webology* 2008; 5: 2.
[14] Aggarwal CC and Zhai CX. A survey of text clustering algorithms. In: Aggarwal CC and Zhai CX (eds) *Mining Text Data*. 1st ed. Berlin: Springer-Verlag, 2012, pp. 77–128.
[15] Aggarwal CC and Zhai CX. A survey of text classification algorithms. In: Aggarwal CC and Zhai CX (eds) *Mining Text Data*. 1st ed. Berlin: Springer-Verlag, 2012, pp. 163–222.

[16] Monmarche N, Slimane M and Venturini G. On improving clustering in numerical databases with artificial ants. In: *Proceedings of the 5th European Conference on Advances in Artificial Life*. Springer, 1999, pp. 626–635.
[17] Song W and Park SC. Genetic algorithm for text clustering based on latent semantic indexing. *Computers and Mathematics with Applications* 2009; 57: 1901–1907.
[18] Hasanzadeh E, Poyanrad M and Rokny HA. Text clustering on latent semantic indexing with particle swarm optimization (PSO) algorithm. *International Journal of the Physical Sciences* 2012; 7(1): 116–120.
[19] Vaijayanthi P, Natarajan AM and Murugadoss R. Ants for document clustering. *International Journal of Computer Science* 2012; 9(2): 493–499.
[20] Dziwinski P, Bartczuk L and Starczewski JT. Fully controllable ant colony system for text data clustering. *Lecture Notes in Computer Science* 2012; 7269: 199–205.
[21] Azaryuon K and Fakhar B. A novel document clustering algorithm based on ant colony optimization algorithm. *Journal of Mathematics and Computer Science* 2013; 7: 171–180.
[22] Avanija J and Ramar K. A hybrid approach using pso and k-means for semantic clustering of web documents. *Journal of Web Engineering* 2013; 12(3–4): 249–264.
[23] Forsati R, Mahdavi M, Shamsfard M and Meybod MR. Efficient stochastic algorithms for document clustering. *Information Science* 2013; 220:269–291.
[24] Cagnina L, Errecalde M, Ingaramo D and Rosso P. An efficient particle swarm optimization approach to cluster short texts. *Information Sciences* 2014; 265: 36–49.
[25] Forsati R, Keikha A and Shamsfard M. An improved bee colony optimization algorithm with an application to document clustering. *Neurocomputing* 2015; 159: 9–26.
[26] Judith JE and Jayakumari J. An efficient hybrid distributed document clustering algorithm. *Scientific Research and Essays* 2015; 10(1): 14–22.
[27] Song W, Qiao Y, Park SC and Qian X. A hybrid evolutionary computation approach with its application for optimizing text document clustering. *Expert Systems with Applications* 2015; 42: 2517–2534.
[28] Hofmann T. Probabilisitc latent semantic indexing. In: *Proceedings of the 22nd Annual International SIGIR Conference on Research and Development in Information Retrieval*. ACM, 1999, pp. 50–57.
[29] Griffiths TL and Steyvers M. Finding scientific topics. In: *Proceedings of the National Academy of Sciences of the United States of America*. PNAS, 2004, pp. 5228–5235.
[30] Griffiths TL, Stevyers M and Tenenbaum JB. Topics in semantic representation. *Psychological Review* 2007; 114(2): 211–244.
[31] Rathore AS and Roy D. Performance of LDA and DCT models. *Journal of Information Science* 2014; 40(3): 281–292.
[32] Guo X, Xiang Y, Chen Q, Huang Z and Hao Y. LDA-based online topic detection using tensor factorization. *Journal of Information Science* 2013; 39(4): 459–469.
[33] Savoy J. Authorship attribution based on a probabilistic topic model. *Information Processing and Management* 2013; 49(1): 341–354.
[34] Zhai Z, Liu B, Xu H and Jia P. Constrained LDA for grouping product features in opinion mining. In: *Proceedings of the 15th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining*. PAKDD, 2011, pp. 448–459.
[35] Wu Q, Zhang C, Hong Q and Chen L. Topic evolution based on LDA and HMM and its application in stem cell research. *Journal of Information Science* 2014; 40(5): 611–620.
[36] Bagheri A, Saraee M and de Jong F. ADM-LDA: An aspect detection model based on topic modelling using the structure of review sentences. *Journal of Information Science* 2014; 40(5): 621–636.
[37] Hu Z, Fang S and Liang T. Empirical study of constructing a knowledge organization system of patent documents using topic modelling. *Scientometrics* 2014; 100(3): 787–799.
[38] Misra H, Yvon F, Cappe O and Jose J. Text segmentation: a topic modelling perspective. *Information Processing and Management* 2011; 47(4): 528–544.
[39] Chen Z, Huang Y, Tian J, Liu X, Fu K and Huang T. Joint model for sub sentence-level sentiment analysis with Markov logic. *Journal of the Association for Information Science and Technology* 2015; 66(9): 1913–1922.
[40] Kar M, Nunes S and Ribeiro C. Summarization of changes in dynamic text collections using Latent Dirichlet Allocation model. *Information Processing and Management* 2015; 51(6): 809–833.
[41] Newman DJ and Block S. Probabilistic topic decomposition of an eighteenth-century American newspaper. *Journal of the American Society for Information Science and Technology* 2006; 57(6): 753–767.
[42] Omar M, On BW, Lee I and Choi GS. LDA topics: representation and evaluation. *Journal of Information Science* 2015; 41(5): 662–675.
[43] Rafi M, Shaikh SM and Farooq A. Document clustering based on topic maps. International *Journal of Computer Applications* 2010; 12(1): 32–36.
[44] Ma Y, Wang Y and Jin B. A three-phase approach to document clustering based on topic significance degree. *Expert Systems with Applications* 2014; 41(18): 8203–8210.
[45] Yau CK, Porter A, Newman N and Suominen A. Clustering scientific documents with topic modelling. *Scientometrics* 2014; 100: 767–786.

[46] Xu JG, Zhou SL, Qiu L, Liu SY and Li PF. A document clustering algorithm based on semi-constrained hierarchical latent Dirichlet allocation. In: Buchmann R, Kifor CV and Yu J (eds) *Knowledge Science, Engineering and Management*. 1st ed. Berlin: Springer-Verlag, 2014, pp. 49–60.
[47] Qiu L and Xu JG. A Chinese word clustering method using latent Dirichlet allocation and K-means. In: *Proceedings of the 2nd International Conference on Advances in Computer Science and Engineering*. CSE 2013, 2013, pp. 267–270.
[48] Tang G, Xia Y, Cambria E, Jin P and Zeng TF. Document representation with statistical word senses in cross-lingual document clustering. *International Journal of Pattern Recognition and Artificial Intelligence* 2015; 29(2): 1–26.
[49] Vulic I, Smet WD, Tang J and Moens MF. Probabilistic topic modelling in multilingual settings: an overview of its methodology and applications. *Information Processing and Management* 2015; 51(1): 111–147.
[50] Savoy J. Text clustering: an application with the state of the union addresses. *Journal of the Association for Information Science and Technology* 2015; 66(8): 1645–1654.
[51] Wei X, Xiu L and Gong Y. Document clustering based on non-negative matrix factorization. In: *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 2003, pp. 267–273.
[52] Tang B, Shepher M, Heywood MI and Luo X. Comparing dimension reduction techniques for document clustering. *Lecture Notes in Computer Science* 2005; 3501: 292–296.
[53] Ding C and Li T. Adaptive dimension reduction using discriminant analysis and k-means. In: *Proceedings of the 24th International Conference on Machine Learning*. ACM, 2007, pp. 521–528.
[54] Zhu WZ and Allen RB. Document clustering using the LSI subspace signature model. *Journal of the American Society for Information Science and Technology* 2013; 64(4): 844–860.
[55] Blei DM, Ng AY and Jordan MI. Latent dirichlet allocation. *Journal of Machine Learning Research* 2003; 3: 993–1022.
[56] Blei DM. Probabilistic topic models. *Communications of the ACM* 2012; 55(4): 77–84.
[57] Jordan M. *Learning in graphical models*. 1st ed. Cambridge: MIT Press, 1999.
[58] Jain AK. Data clustering: 50 years beyond k-means. *Pattern Recognition Letters* 2010; 31: 651–666.
[59] Huang A. Similarity measures for text document clustering. In: *Proceedings of the New Zealand Computer Science Research Student Conference*. NZCSRCS, 2008, pp. 1–8.
[60] Theodoridis S and Koutroumbas K. *Pattern Recognition*. New York: Academic Press, 1999.
[61] Huang Z. Extensions to the k-means algorithm for clustering large data sets with categorical values. *Data Mining and Knowledge Discovery* 1998; 2(2): 283–304.
[62] Arthur D and Vassilvitskii S. K-means++: the advantage of careful seeding. In: *Proceedings of the Eigheenth Annual Symposium on Discrete Algorithms*. ACM-SIAM, 2007, pp. 1027–1035.
[63] Dempster AP, Laird NM and Rubin DB. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society* 1977; 39(1): 1–38.
[64] Jin X and Han J. Expectation maximization clustering. In: Sammut C and Webb GI (eds) *Encyclopedia of Machine Learning*. 1st ed. Berlin: Springer-Verlag, 2010, pp. 382–383.
[65] Kennedy J and Eberhart RC. Particle swarm optimization. In: *Proceedings of the International Conference on Neural Networks*. IEEE, 1995, pp. 1942–1948.
[66] Engelbrecht AP. *Computational intelligence: an introduction*. 2nd ed. New York: Wiley, 2007.
[67] Omran M, Salman A and Engelbrecht AP. Image classification using particle swarm optimization. In: *Proceedings of the fourth Asia-pacific conference on simulated evolution and learning*. IEEE, 2002, pp. 370–374.
[68] Deneubourg JL, Goss S, Franks N, Sendova-Fransk A, Detrain C and Chretien L. The dynamics of collective sorting: robot-like ants and ant-like robots. In: *Proceedings of the First International Conference on Simulation of Adaptive Behaviour*. MIT Press, 1991, pp. 356–365.
[69] Lumer E and Faieta B. Diversity and adaptation in populations of clustering ants. In: *Proceedings of the Third International Conference on Simulation of Adaptive Behaviour*. MIT Press, 1994, pp. 499–508.
[70] Kanade PM. *Fuzzy ants as a clustering concept*. MSc Thesis. Tampa, FL: University of South Florida, 2004.
[71] Handl J and Meyer B. Ant-based and swarm-based clustering. *Swarm Intelligence* 2007; 1: 95–113.
[72] Sugar CA and Garth MJ. Finding the number of clusters in a data set: an information theoretic approach. *Journal of the American Statistical Association* 2003; 98: 750–763.
[73] Onan A. *A study of hybrid evolutionary algorithms for cluster analysis*. MSc Thesis. Izmir: Ege University, Turkey, 2013.
[74] Rossi RG, Maraccini RM and Rezende SO. *Benchmarking text collections for classification and clustering tasks*. Technical Report. Sao Paulo: University of Sao Paulo, 2013.
[75] Mimno D, Wallach HM, Talley E, Leenders M and McCallum A. Optimizing semantic coherence in topic models. In: *Proceedings of the conference on empirical methods in natural language processing*. Stroudsburg, PA: Association for Computational Linguistics, 2011, pp. 262–272.
[76] Meyer S and Stein B. Analysis of clustering algorithm for web-based search. *Lecture Notes in Computer Science* 2002; 2569: 168–178.
[77] Tan SC, Ting KM and Teng SW. A comparative study of a practical clustering method with traditional methods. *Lecture Notes in Computer Science* 2010; 6464: 112–121.

[78] Handl J, Knowles J and Dorigo M. *Ant-based clustering: a comparative study of its relative performance with respect to k-means, average link and 1d-som*. Technical Report. Brussels: Universite Libre de Bruxelles, 2003.
[79] Handl J, Knowles J and Dorigo M. On the performance of ant-based clustering. In: *Proceedings of the 3rd International Conference on Hybrid Intelligent Systems*. IOS Press, 2003, pp. 204–213.
[80] Dalli A. Adaptation of the F-measure to cluster-based lexicon quality evaluation. In: *Proceedings of 10th Conference of the European Chapter for the Association for Computational Linguistics*. ACL, 2003, pp. 51–56.