

# SMGKM: An Efficient Incremental Algorithm for Clustering Document Collections

Adil Bagirov<sup>1</sup>, Sattar Seifollahi<sup>2,3,4</sup>, Massimo Piccardi<sup>2</sup>, Ehsan Zare Borzeshi<sup>3</sup>,  
and Bernie Kruger<sup>4</sup>

<sup>1</sup> Faculty of Science and Technology,

Federation University Australia, VIC, Australia

<sup>2</sup> Faculty of Engineering and Information Technology,

University of Technology Sydney, NSW, Australia

<sup>3</sup> Capital Markets Cooperative Research Centre (CMCRC),

Sydney, NSW, Australia

<sup>4</sup> Transport Accident Commission (TAC), VIC, Australia

**Abstract.** Given a large unlabeled document collection, the aim of this paper is to develop an accurate and efficient algorithm for solving the clustering problem over this collection. Document collections typically contain tens or hundreds of thousands of documents, with thousands or tens of thousands of features (i.e., distinct words). Most existing clustering algorithms struggle to find accurate solutions on such large data sets. The proposed algorithm overcomes this difficulty by an incremental approach, incrementing the number of clusters progressively from an initial value of one to a set value. At each iteration, the new candidate cluster is initialized using a partitioning approach which is guaranteed to minimize the objective function. Experiments have been carried out over six, diverse datasets and with different evaluation criteria, showing that the proposed algorithm has outperformed comparable state-of-the-art clustering algorithms in all cases.

**Keywords:** Document clustering, Incremental clustering, Spherical  $k$ -means

## 1 Introduction

Text document clustering has received considerable attention in the literature due to the huge amount of information generated in an electronic form in areas such as text mining and information retrieval. Given the size of such document collections, it is vital to be able to bring them into more a structured form. To this aim, *cluster analysis* can play an important role in organizing such a huge amount of documents into meaningful clusters. A cluster can be simply defined as a collection of data objects (documents here) that are ‘similar’ (under a suitable similarity definition) to one another and dissimilar from objects in other clusters.

Most clustering methods applied to unstructured document collections start with creating a vector space known as a bag-of-words (BoW) model [24]. In this

model, each document is represented by a vector with the frequencies of each word in the document. Such a representation is typically very sparse due to the large number of distinct words. The set of all documents in the vector space is usually called the document-to-term matrix.

Document clustering can be seen as a specialization of general data clustering. It was initially used for improving the precision or recall in information retrieval systems [14, 26] and as an efficient way of finding the nearest neighbors of a document [7]. Document clustering can be organized over two main stages: in the first stage, the documents are preprocessed into a usable data representation. Preprocessing may include some of the following tasks: the exclusion of words without informational value, known as stop words; the reduction of the words to their radicals, known as stemming; the uppercase/lowercase conversion, known as case-folding; and the eventual transformation into vector space. The second stage is the clustering of the vectorial representations.

Using the vector representation, various classical clustering algorithms such as the  $k$ -means algorithm and its variants, hierarchical agglomerative clustering and graph-theoretic methods have been applied in the text mining literature; for detailed reviews, see [1, 8, 11].

The similarity measure is fundamental to formalize a clustering problem. This measure, in particular, can be defined using distance(-like) functions. Clustering problems based on the squared Euclidean norm as the similarity measure are called the *minimum sum-of-squares clustering* (MSSC) problems. To date, many different algorithms have been proposed to solve this problem. Amongst them, the  $k$ -means algorithm and its variants have been amongst the most popular (see, for example, [12, 13] and references therein; and [2–4, 15, 22]).

Another similarity measure, which is widespread in text mining, is the cosine measure. The spherical  $k$ -means (SKM) algorithm [8] is the variant that uses the cosine measure. The SKM, in fact, is equivalent to a  $k$ -means algorithm using the Euclidean distance over the projection of the vector space onto the unit sphere. It has been found to work well for text clustering. The work of [5] has shown that SKM can also be derived as an EM algorithm for Maximum Likelihood Estimation of the mean direction parameters of a uniform mixture of von Mises-Fisher (or Langevin) distributions.

In the literature, there are two main categories of feature extraction methods: term frequency-based methods [8, 9, 20, 23] and semantic methods [17, 18, 28, 29]. A term frequency-based method is simply based on counting words' number, whereas a semantic method attempts to construct an ontology containing words and their relations. Term frequency-based methods tend to be simpler and more effective and, for this reason, we adopt them in this work. In particular, we leverage the term frequency-inverse document frequency (tf-idf) representation which is the product of the term frequency (tf) (the raw count of the terms appearing in the document) and the inverse document frequency (idf) which increases the importance of terms that appear in only a few documents and conversely decreases the importance of terms appearing in many documents.

The tf-idf representation has been widely utilized thanks to its computational efficiency and effectiveness [9, 20, 25].

Many document clustering algorithms such as  $k$ -means and SKM provide a means for effectively navigating, summarizing and organizing the information in the collection. In this paper, we propose an algorithm to improve the solution provided by SKM with a modest increase of computational load still within the range of a single-processor machine. Thus, our emphasis is on high accuracy with a limited increase of computational resources. The algorithm is an extension of the strategy of the incremental algorithm presented in [2], with the main difference that it projects each solution to the unit sphere to satisfy the spherical constraint. In our approach, the documents are first converted to a tf-idf representation [8, 23]. Normalizing the data vectors helps remove the bias induced by the length of a document and has generally provided superior results [8, 23, 24]. The vector space model is, then, used as an input to our two-stage algorithm. We find a better initial solution to the clustering problem in the first stage, and improve the result iteratively in the second stage by starting from the initial solution. The main contribution of our algorithm is a procedure to select better initial cluster centroids on the unit sphere in the first stage (which is different from the one in [2]), for the benefit of the ensuing MSSC optimization. The experimental results presented in Section 4 show that the proposed algorithm outperforms comparable one-stage algorithm spherical  $k$ -means. Unlike the spherical  $k$ -means algorithm the proposed algorithm as an incremental algorithm solves not only the clustering problem with the given number of clusters but also all intermediate clustering problems. Moreover, the proposed algorithm requires significantly less computational time than the spherical  $k$ -means algorithm in solving all clustering problems.

The rest of this paper is organized as follows. Section 2 provides a brief discussion on related works with special focus on the spherical  $k$ -means. The proposed method and related algorithms are presented in Section 3. Data and numerical results are reported and discussed in Section 4, and finally Section 5 contains some concluding remarks.

Throughout this paper we will use symbols  $m$ ,  $n$ , and  $k$  to denote the number of documents, the number of terms (or feature), and the number of clusters, respectively. We will use symbol  $X$  to denote the set of  $m$  documents that we wish to cluster, and  $X^1, X^2, \dots, X^k$ , to denote each of the  $k$  clusters.

## 2 Problem Formulation

Given the vector space model, the document vectors may be represented as  $x^1, x^2, \dots, x^m$ , with each  $x^i \in R^n$ . Recall that  $m$  is the total number of documents and  $n$  stands for the number of unique words in the vector space model. A clustering of the document collection is its partitioning into the disjoint subsets

$X^1, X^2, \dots, X^k$ , i.e.

$$\bigcup_{j=1}^k X^j = \{x^1, x^2, \dots, x^m\} \quad \& \quad X^j \cap X^l = \emptyset, \quad j \neq l.$$

In SKM, the data are projected onto the unit sphere. Dhillon et al. [8] have used the popular tf-idf scheme which reads out as (normalized) term frequency-inverse document frequency [23]. The tf-idf normalization implies that  $\|x^i\| = 1$ , i.e., each document vector lies on the surface of the unit sphere in  $R^n$ . The  $k$ -clustering (or  $k$ -partition) problem is formulated as the following optimization problem:

$$\begin{cases} \min & f_k(c) \\ \text{subject to} & c = (c^1, \dots, c^k) \in R^{nk}, \end{cases} \quad (1)$$

where

$$f_k(c^1, \dots, c^k) = \sum_{x \in X} \min_{j=1, \dots, k} d(c^j, x). \quad (2)$$

Here,  $c^1, \dots, c^k \in R^n$  are cluster centers and the function  $d : R^n \times R^n \rightarrow R_+$  is the similarity measure,  $R_+$  is the set of nonnegative numbers.

The function  $f_k$  is called the  $k$ -th clustering objective function. The similarity measure  $d$  is defined using the cosine measure, that is for  $c, x \in R^n$

$$d(c, x) = 1 - \cos(x, c) = 1 - \frac{\langle x, c \rangle}{\|x\| \|c\|},$$

where  $\langle x, c \rangle$  stands for the inner product of  $x$  and  $c$  and  $\|\cdot\|$  is the Euclidean norm in  $R^n$ .

Since all document vectors are normalized it is also required that for each cluster center  $c^i \in R^n$  is also normalized that is:  $\|c^i\| = 1$ ,  $i = 1, \dots, k$ . In this case one has the following similarity measure  $d$ :

$$d(c, x) = 1 - \langle x, c \rangle.$$

Then the  $k$ -clustering can be reformulated as the following constrained optimization problem:

$$\begin{cases} \min & f_k(c) \\ \text{subject to} & c = (c^1, \dots, c^k) \in R^{nk}, \\ & \|c^i\| = 1, i = 1, \dots, k. \end{cases} \quad (3)$$

The problem (3) is a nonconvex constrained optimization problem and its objective function is piecewise linear. Due to the minimum operation used in the definition of this function (see (2)), it is also nonconvex. Since the document collections usually contain hundreds of thousands of documents, objective function  $f_k$  has many local solutions. Furthermore, the typical number of words in

these collections is thousands or even tens of thousands. Therefore, Problem (3) is a large-scale optimization problem. Finally, the feasible set in this problem is nonconvex and it is a thin set in the  $n$ -dimensional space. Such problems are highly challenging not only for global optimization techniques, but also for local optimization methods. In this paper, we use the spherical  $k$ -means algorithm as our algorithm of choice.

### 3 The proposed algorithm

The proposed algorithm is based on an incremental approach. The main idea is the following: instead of working with  $k$  clusters from the start, we add the clusters one by one in successive iterations. At each iteration, we select the initial position of the added cluster by using a partitioning approach that is described in Section 3.1. This approach enjoys a performance guarantee that proves key for the accuracy of the overall algorithm. After the addition of the new initial cluster, a conventional  $k$ -means algorithm is called to re-optimize all the clusters (3). Then, the algorithm proceeds to the next iteration, until all clusters have been added.

#### 3.1 Calculation of starting cluster centers

The incremental algorithm proposed in this paper solves the clustering problem gradually by starting with one cluster and adding a new cluster at a time, up to the set number. Hereafter we describe the algorithm for determining the initial position of the cluster added at the  $k$ -th iteration.

Assume that the solution  $c^1, \dots, c^{k-1}$ ,  $k \geq 2$  to the  $(k-1)$ -clustering problem is known. Denote by  $d_{k-1}^i$  the distance between  $x^i$ ,  $i = 1, \dots, m$  and the closest cluster center among  $k-1$  centers  $c^1, \dots, c^{k-1}$ :

$$d_{k-1}^i = \min \{d(c^1, x^i), \dots, d(c^{k-1}, x^i)\}. \quad (4)$$

We will also use the notation  $d_{k-1}^x$  for  $x \in \{x^1, \dots, x^m\}$ .

Consider the following two sets:

$$S_1 = \{y \in R^n : d(y, x^i) \geq d_{k-1}^i, \forall i \in \{1, \dots, m\}\},$$

$$S_2 = \{y \in R^n : \exists i \in \{1, \dots, m\} \text{ such that } d(y, x^i) < d_{k-1}^i\}.$$

The set  $S_1$  contains all points  $y \in R^n$  which do not attract any point from the set  $X$  and the set  $S_2$  contains all points  $y \in R^n$  which attract at least one point from  $X$ . It is obvious that cluster centers  $c^1, \dots, c^{k-1} \in S_1$ . Since the number  $k$  of clusters is less than the number of data points in the set  $X$  all data points which are not cluster centers belong to the set  $S_2$  (because such points attract at least themselves) and therefore this set is not empty. Note that  $S_1 \cap S_2 = \emptyset$  and  $S_1 \cup S_2 = R^n$ .

$$f_{k-1}(c^1, \dots, c^{k-1}) = \frac{1}{m} \sum_{i=1}^m d_{k-1}^i, \forall y \in S_1.$$

This means by taking any point  $y \in S_1$  as a starting point for the  $k$ -th cluster center will not decrease the value of the clustering function  $f_k$ . Therefore, starting points should not be chosen from the set  $S_1$ .

Take any  $y \in S_2$ . Then one can divide the set  $X$  into two subsets as follows:

$$\begin{aligned}\bar{B}_1(y) &= \{x \in X : d(y, x) \geq d_{k-1}^x\}, \\ \bar{B}_2(y) &= \{x \in X : d(y, x) < d_{k-1}^x\}.\end{aligned}$$

The set  $\bar{B}_2(y)$  contains all data points  $x \in X$  which are closer to the point  $y$  than to their cluster centers and the set  $\bar{B}_1(y)$  contains all other data points. Since  $y \in S_2$  the set  $\bar{B}_2(y) \neq \emptyset$ . Furthermore,  $\bar{B}_1(y) \cap \bar{B}_2(y) = \emptyset$  and  $X = \bar{B}_1(y) \cup \bar{B}_2(y)$ .

The difference  $z_k(y)$  between the value of the  $k$ -th auxiliary cluster function with the  $k$ -th cluster center  $y$  and the value  $f_{k-1}(c^1, \dots, c^{k-1})$  for the  $(k-1)$ -clustering problem is:

$$z_k(y) = \frac{1}{m} \sum_{x \in \bar{B}_2(y)} (d_{k-1}^x - d(y, x))$$

which can be rewritten as

$$z_k(y) = \frac{1}{m} \sum_{x \in X} \max\{0, d_{k-1}^x - d(y, x)\}. \quad (5)$$

The difference  $z_k(y)$  shows the decrease of the value of the  $k$ -th cluster function  $f_k$  comparing with the value  $f_{k-1}(c^1, \dots, c^{k-1})$  if the point  $(c^1, \dots, c^{k-1}, y)$  is chosen as the cluster center for the  $k$ -clustering problem.

If a data point  $x \in A$  is the cluster center then this point belongs to the set  $S_1$ , otherwise it belongs to the set  $S_2$ . Therefore we choose a point  $y$  from the set  $X \setminus S_1$ . We take any  $y = x \in X \setminus S_1$ , compute  $z_k(x)$  and introduce the following number:

$$z_{max}^1 = \max_{x \in X \setminus S_1} z_k(x). \quad (6)$$

Let  $\gamma_1 \in [0, 1]$  be a given number. We compute the following subset of  $X$ :

$$\bar{X}_1 = \{x \in X \setminus S_1 : z_k(x) \geq \gamma_1 z_{max}^1\}. \quad (7)$$

If  $\gamma_1 = 0$  then  $\bar{X}_1 = X \setminus S_1$  and if  $\gamma_1 = 1$  then the set  $\bar{X}_1$  contains data points with the largest decrease  $z_{max}^1$ .

For each  $x \in \bar{X}_1$  we compute the set  $\bar{B}_2(x)$  and its center  $c(x)$ . We replace the point  $x \in \bar{X}_1$  by the point  $c(x)$  because the latter is better representative of the set  $\bar{B}_2(x)$  than the former. Denote by  $\bar{X}_2$  the set of all such centers. For each  $x \in \bar{X}_2$  we compute the number  $z_k^2(x) = z_k(x)$  using (5). Finally, we compute the following number:

$$z_{max}^2 = \max_{x \in \bar{X}_2} z_k^2(x). \quad (8)$$

The number  $z_{max}^2$  represents the largest decrease of the values  $f_l(c^1, \dots, c^{l-1}, x)$  among all centers  $x \in \bar{X}_2$  comparing with the value  $f_{k-1}(c^1, \dots, c^{l-1})$ .

Let  $\gamma_2 \in [0, 1]$  be a given number. We define the following subset of  $\bar{X}_2$ :

$$\bar{X}_3 = \{x \in \bar{X}_2 : z_l^2(x) \geq \gamma_2 z_{max}^1\}. \quad (9)$$

If  $\gamma_2 = 0$  then  $\bar{X}_3 = \bar{X}_2$  and if  $\gamma_2 = 1$  then the set  $\bar{X}_3$  contains only centers  $x$  with the largest decrease of the cluster function  $f_k$ .

All points from the set  $\bar{X}_3$  are considered as starting points for solving problem (3). Therefore, their selection guarantees to maximally decrease the objective.

The algorithm for finding the initial cluster centers in solving Problem (3) can be summarized as follows:

**Algorithm 1** Algorithm for finding the set of starting cluster centers.

**Input:** The solution  $(c^1, \dots, c^{k-1})$  to the  $(k-1)$ -clustering problem.

**Output:** The set of starting cluster centers for the  $k$ -th cluster center.

*Step 0.* (Initialization). Select  $\gamma_1, \gamma_2 \in [0, 1]$ .

*Step 1.* Compute  $z_{max}^1$  using (6) and the set  $\bar{X}_1$  using (7).

*Step 2.* Compute  $z_{max}^2$  using (8) and the set  $\bar{X}_3$  using (9).

### 3.2 An incremental clustering algorithm and its implementation

In this subsection we present an incremental algorithm for solving Problem (3).

**Algorithm 2** An incremental clustering algorithm.

**Input:** The collection of documents  $X = \{x^1, \dots, x^m\}$ .

**Output:** The set of  $k$  cluster centers  $\{c^1, \dots, c^k\}, k > 0$ .

*Step 1.* (Initialization). Compute the center  $c^1 \in R^n$  of the set  $X$ . Set  $l := 1$ .

*Step 2.* (Stopping criterion). Set  $l := l + 1$ . If  $l > k$  then stop. The  $k$ -partition problem has been solved.

*Step 3.* (Computation a set of starting points for the next cluster center). Apply Algorithm 1 to compute the set  $\bar{X}_3$  of starting point for the  $l$ -th cluster center.

*Step 4.* (Computation a set of cluster centers). For each  $\bar{y} \in \bar{X}_3$  take  $(c^1, \dots, c^{l-1}, \bar{y})$  as a starting point, solve Problem (3) and find a solution  $(\hat{y}^1, \dots, \hat{y}^l)$ . Denote by  $\bar{X}_4$  a set of all such solutions.

*Step 5.* (Computation of the best solution). Compute

$$f_l^{min} = \min \{f_l(\hat{y}^1, \dots, \hat{y}^l) : (\hat{y}^1, \dots, \hat{y}^l) \in \bar{X}_4\}$$

and the collection of cluster centers  $(\bar{y}^1, \dots, \bar{y}^l)$  such that

$$f_l(\bar{y}^1, \dots, \bar{y}^l) = f_l^{min}.$$

*Step 6.* (Solution to the  $l$ -partition problem). Set  $c^j := \bar{y}^j, j = 1, \dots, l$  as a solution to the  $l$ -th partition problem and go to Step 2.

We call the proposed Algorithm 2 the Spherical Modified Global  $k$ -means (SMGKM). The most important and time consuming step in this algorithm is Step 4 where Problem (3) is solved starting from many initial cluster centers. For this problem, we use a conventional spherical  $k$ -means algorithm which allows us to automatically take into account the constraints of Problem (3).

## 4 Experimental results

In this section we present numerical results on the evaluation of the proposed method and compare it with the Spherical  $k$ -means algorithm (SKM), described in [8]. We do not include comparison with hierarchical clustering algorithms as these algorithms have not been widely used in text mining. The reason of using the SKM for comparison is the similarity of the SKM with our proposed method in which both algorithms use spherical space and  $k$ -means as the base.

### 4.1 Datasets

To test and compare the proposed algorithm, we have carried out experiments with six datasets. A brief description of these datasets is given in Table 1 and more details of the datasets and preprocessing are given below.

**Table 1.** Dataset summary.

Datasets	$m$	$n$
1. Cora	2,240	2,319
2. Associated Press (APress)	2,246	4,994
3. WebKB	4,199	2,153
4. Reuters	12,902	1,313
5. Phone Calls (PCalls)	13,937	2,696
6. 20 Newsgroups (20Newsg)	18,774	3,103

The Cora data set [19] consists of the abstracts and references of approximately 34,000 computer science research papers; of these, we selected a subset of 2410 papers categorized into one of seven subfields of machine learning.

The Associated Press (APress) collection [10] contains Associated Press news stories from 1988 to 1990. The original data includes over 200,000 documents with 20 categories. The sample AP data set from [6], which is sampled from a subset of the TREC AP collection contains 2,246 documents.

The WebKB data set [27] consists of approximately 6000 web pages from computer science departments of various university, divided into seven categories: student, faculty, staff, course, project, department and other. In this paper, we use the four most populous entity-representing categories: student, faculty, course, and project, which all together contain 4,199 pages.

The Reuters data set [16] was originally collected by Carnegie Group, Inc. and Reuters, Ltd. in the course of developing the CONSTRUE text categorization system. It consists of 21,578 news stories appeared on the Reuters newswire in 1987. We use a subset of data containing 12,902 documents which are manually assigned to 135 categories.

The 20 Newsgroups dataset (20Newsg) [21] contains postings to Usenet newsgroups. The postings are organized by content into 20 different newsgroups with about 1000 messages from each newsgroup and are therefore well suited for text clustering. This collection consists of 18,774 non-empty documents distributed evenly across 20 newsgroups.

Finally, The Phone Calls dataset (PCalls) is from the Transport Accident Commission (TAC) which is a major accident compensation agency of the Victorian Government in Australia. It consists of a collection of 593,433 phone calls from 13,937 single TAC clients recorded by various operators over 5 years. The phone calls are made for different purposes including, but not limited to: compensation payments, recovery and return to work, different type of services, medications and treatments, pain, solicitor engagement and mental health issues.

The following preprocessing steps have been applied to all datasets before their use in the experiments: 1) removal of numbers, punctuation, symbols and “stopwords”; 2) synonyms and misspelled words have been replaced with the base and actual words (for the PCalls dataset); 3) sparse terms (95% sparsity or more) and infrequently occurring words have been removed; 4) we have also removed generic words (for the PCalls dataset) such as names and addresses based on a predefined list. The data have then been projected to a vector space by using the popular term frequency-inverse document frequency (tf-idf) scheme.

## 4.2 Discussion and evaluation

Tables 2 and 3 show the best objective function value,  $f_{best}$ , and relative errors,  $E_1$  and  $E_2$  for the SKM and SMGKM respectively, where the relative error is defined as:

$$E_i = \frac{f_i - f_{best}}{f_{best}} \times 100$$

where  $f_i$  is the value of the clustering function obtained by  $i$ -th algorithm. In most cases, SMGKM demonstrate better performance, i.e., low values for the objective function in terms of relative errors, and in some cases the differences are significant. When the number of clusters is small, SKM performs slightly better than SMGKM (in some cases) but the differences are not significant.

To further compare and assess the quality of the clusters generated by the algorithms, we apply two well-known cluster validity indices: the Dunns and DaviesBouldin validity indices.

The Dunn’s validity index is defined as

$$I(D) = \max_{i=1, \dots, k} \left\{ \min_{j=1, \dots, k; j \neq i} \left\{ \frac{d(c^i, c^j)}{\max_{l=1, \dots, k} r(c^l)} \right\} \right\} \quad (10)$$

**Table 2.** The function value and relative errors

$k$	Associated Press			Cora			WebKB		
	$f_{best}$	$E_1$	$E_2$	$f_{best}$	$E_1$	$E_2$	$f_{best}$	$E_1$	$E_2$
10	1509.66	0.00	0.15	1475.39	0.00	0.06	2607.65	0.00	0.45
12	1481.68	0.00	0.25	1455.01	0.16	0.00	2567.58	0.00	0.27
15	1456.63	0.00	0.26	1423.41	0.00	0.29	2505.60	0.00	0.09
17	1435.98	0.56	0.00	1406.54	0.00	0.21	2455.62	0.34	0.00
20	1411.64	0.66	0.00	1384.11	0.44	0.00	2408.00	0.65	0.00
25	1380.19	0.95	0.00	1351.01	0.49	0.00	2342.28	0.30	0.00
30	1355.94	0.97	0.00	1324.20	0.89	0.00	2285.21	0.25	0.00
35	1337.06	0.90	0.00	1300.97	1.28	0.00	2229.02	0.62	0.00
40	1316.64	0.95	0.00	1279.14	1.61	0.00	2183.48	0.68	0.00
45	1300.28	0.53	0.00	1262.07	1.79	0.00	2143.49	0.68	0.00
50	1286.99	1.03	0.00	1248.15	1.73	0.00	2099.22	1.01	0.00
55	1274.34	0.70	0.00	1235.44	1.95	0.00	2064.24	1.05	0.00
60	1263.40	0.85	0.00	1223.60	2.48	0.00	2035.21	1.16	0.00
65	1254.37	1.06	0.00	1213.49	1.85	0.00	2009.59	1.64	0.00
70	1245.54	0.82	0.00	1204.21	1.96	0.00	1986.34	1.58	0.00
75	1235.14	0.52	0.00	1195.76	2.13	0.00	1960.96	1.67	0.00
80	1227.43	0.78	0.00	1187.62	1.63	0.00	1939.72	1.74	0.00
85	1220.38	0.04	0.00	1180.06	1.67	0.00	1915.72	2.42	0.00
90	1213.26	0.30	0.00	1172.98	1.72	0.00	1896.69	2.43	0.00
95	1203.35	0.00	0.21	1165.74	1.88	0.00	1879.04	2.64	0.00
100	1198.26	0.00	0.13	1159.05	2.01	0.00	1862.89	2.90	0.00

where  $d(c^i, c^j)$  is the distance between centers  $c_i$  and  $c^j$ . The  $r(c^l)$  is the radius of the  $l$ -th cluster center and is defined as

$$r(c^l) = \max_{x \in X^l} \|c^l - x\|, \quad (11)$$

where  $k$  is the number of clusters. The Dunns cluster validity measure maximizes the inter-cluster distances and minimizes the intra-cluster distances. Therefore, the number of clusters that maximizes  $I(D)$  can demonstrate the optimal number of the clusters.

The Davies-Bouldin validity index is a measure of within-cluster to between-cluster separation

$$I(DB) = \frac{1}{k} \sum_{i=1}^k \max_{j=1, \dots, k; j \neq i} \frac{S_k(X^i) + S_k(X^j)}{d(c^i, c^j)} \quad (12)$$

where  $k$  is the number of clusters,  $S_k(X^l)$  is the average distance of all data points from the cluster  $X^l$  to their cluster center  $c^l$  and  $d(c^i, c^j)$  is the distance between  $i$ -th and  $j$ -th cluster centers. Smaller values for the  $I(DB)$  means that clusters are compact and far from each other. Therefore, the smaller  $I(DB)$ , the better clustering.

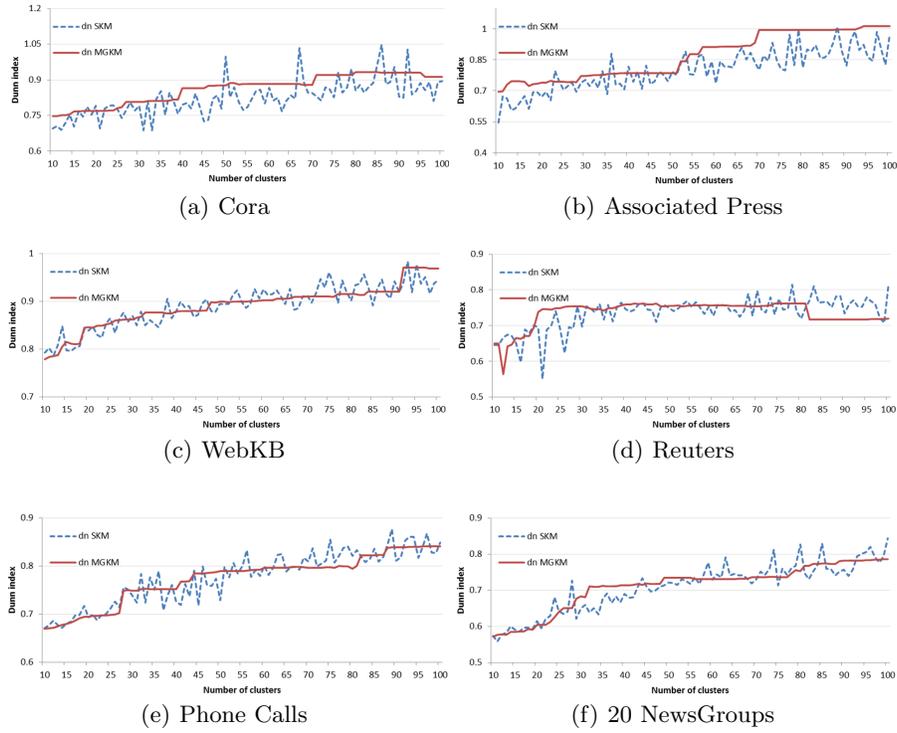
**Table 3.** The function value and relative errors

$k$	Reuters			Phone Calls			20 NewsGroups		
	$f_{best}$	$E_1$	$E_2$	$f_{best}$	$E_1$	$E_2$	$f_{best}$	$E_1$	$E_2$
10	4586.69	0.00	0.06	9493.95	0.00	0.01	12910.54	0.01	0.00
12	4458.53	0.00	0.37	9388.49	0.00	0.16	12772.45	0.00	0.01
15	4289.97	0.00	0.02	9237.82	0.10	0.00	12590.80	0.12	0.00
17	4215.26	0.24	0.00	9144.10	0.53	0.00	12488.07	0.32	0.00
20	4119.90	0.87	0.00	9029.94	0.43	0.00	12367.28	0.15	0.00
25	3984.14	0.69	0.00	8867.60	0.20	0.00	12182.07	0.00	0.09
30	3871.80	1.17	0.00	8729.13	0.23	0.00	12042.95	0.24	0.00
35	3785.61	1.37	0.00	8614.72	0.05	0.00	11898.11	0.00	0.13
40	3719.43	1.41	0.00	8491.54	0.34	0.00	11777.05	0.00	0.27
45	3663.73	0.63	0.00	8385.29	0.23	0.00	11690.82	0.00	0.05
50	3607.86	0.84	0.00	8297.62	0.20	0.00	11559.44	0.13	0.00
55	3555.87	1.20	0.00	8214.63	0.21	0.00	11455.58	0.34	0.00
60	3512.71	1.43	0.00	8139.41	0.15	0.00	11362.46	0.30	0.00
65	3471.35	0.72	0.00	8065.04	0.00	0.06	11282.29	0.30	0.00
70	3431.43	0.95	0.00	7976.79	0.00	0.36	11200.54	0.53	0.00
75	3402.23	1.26	0.00	7931.58	0.00	0.07	11125.10	0.66	0.00
80	3374.66	0.80	0.00	7862.77	0.04	0.00	11044.67	0.60	0.00
85	3344.92	1.07	0.00	7806.99	0.18	0.00	10982.90	0.66	0.00
90	3321.59	1.16	0.00	7756.88	0.14	0.00	10926.60	0.72	0.00
95	3293.53	1.47	0.00	7707.74	0.08	0.00	10874.15	0.49	0.00
100	3271.34	1.15	0.00	7640.79	0.36	0.00	10823.80	0.63	0.00

Figures 1(a) – 1(f) display the Dunn’s cluster validities for SKM and SMGKM as the number of clusters increases from 10 to 100. Here, the dot lines correspond to the SKM and solid lines to the SMGKM. Terms “dn SKM” and “dn SMGKM” stand for Dunn index values using the SKM and SMGKM, respectively. Graphs for the SMGKM are much more stable than graphs for the SKM when the number of clusters increases. The reason is likely that the SMGKM exploits an incremental scheme which adds one cluster each time, while the SKM calculates all clusters from scratch.

Figures 2(a) – 2(f) show the Davies-Bouldin indices for SKM and SMGKM as the number of clusters increases. Terms “db SKM” and “db SMGKM” stand for Davies-Bouldin index values using the SKM and SMGKM, respectively. The graph for SMGKM is more stable, confirming stability in Figures 1(a) and 1(f). These figures also demonstrate significant improvements of SMGKM over the SKM, as the graphs for SMGKM are below (much, when the number of clusters increases) those for SKM in almost all cases.

Table 4 reports the optimal number of clusters using the Dunn and Davies-Bouldin measures as well as the total CPU time spent by SKM and SMGKM.  $c_1^*$  and  $c_2^*$  stand for the optimal number of clusters using the SKM and SMGKM and  $t_1$  and  $t_2$  are the times (cumulative CPU) consumed by SKM and SMGKM,

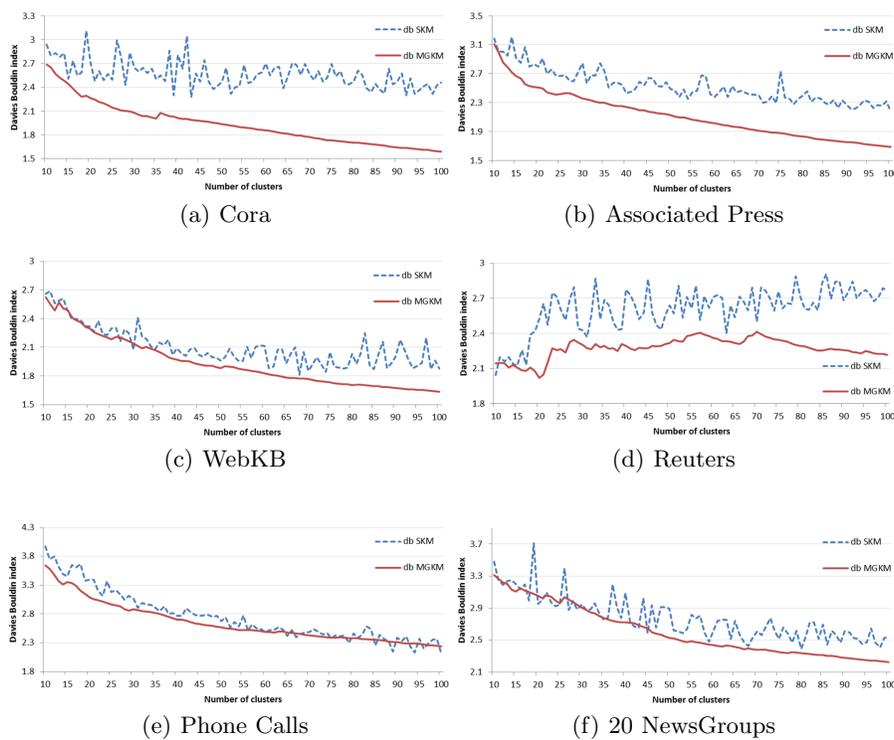


**Fig. 1.** Cluster validity (Dunn) index for datasets 3 and 4

**Table 4.** The optimal number of clusters and cumulative CPU time for computing up to 100 clusters.  $c_1^*$  and  $c_2^*$  stand for the number of clusters using SKM and SMGKM, respectively, and  $t_1$  and  $t_2$  for the time

dataset	Dunn index		DB index		Total CPU time	
	$c_1$	$c_2$	$c_1$	$c_2$	$t_1$	$t_2$
Cora	50	40	40	36	1024	1568
APress	77	70	72	74	2601	2088
WebKB	93	92	68	81	2501	1862
Reuters	29	39	10	20	5327	4745
PCalls	89	90	89	88	15861	11710
NewsG	79	81	68	77	27873	20280

respectively. Despite using a less efficient environment for coding, the times for SMGKM have been lower than those for SKM, except on Cora.



**Fig. 2.** Cluster validity (Davies Bouldin) for datasets 1 – 6

## 5 Conclusions

In this paper, we have presented an incremental algorithm for document clustering that is capable of finding “deeper” solutions. In the algorithm, a new cluster is added in turn starting from an initial position that is guaranteed to maximally decrease the objective function value. Clustering is performed in a spherical space, meaning that each solution is projected to the unit sphere to mitigate the potential bias from more frequent words.

In the experiments, we have thoroughly compared our method with the spherical  $k$ -means algorithm (SKM) that can be regarded as state-of-the-art for document clustering. The results over six challenging datasets have shown that the proposed algorithm has consistently outperformed SKM under a number of clustering indices (objective function value, Dunn and Davies-Bouldin). This gives us ground to believe that the proposed algorithm can prove beneficial for large-scale document clustering applications.

**Acknowledgement.** This project was funded by the Capital Market Cooperative Research Centre in combination with the Transport Accident Commission of Victoria. Acknowledgements and thanks to industry partner David Attwood

(Lead Research Partnerships). This research has received ethics approval from University of Technology Sydney (UTS HREC REF NO. ETH16-0968).

## References

1. D. Arthur and S. Vassilvitskii, “ $k$ -means++: The advantages of careful seeding,” in H. Gabow (Ed.) Proceedings of the 18th Annual ACM-SIAM Symposium on Discrete Algorithms [SODA07], Philadelphia, pp. 1027–1035, 2007.
2. A. M. Bagirov, “Modified global  $k$ -means algorithm for minimum sum-of-squares clustering problems,” Pattern Recognition, vol. 41 (10), pp. 3192–3199, 2008.
3. A. M. Bagirov, J. Ugon, and D. Webb, “Fast modified global  $k$ -means algorithm for incremental cluster construction,” Pattern Recognition, vol. 44 (4), pp. 866–876, 2011.
4. L. Bai, J. Liang, C. Sui, and C. Dang, “Fast global  $k$ -means clustering based on local geometrical information,” Information Sciences, vol. 245, pp. 168 – 180, 2013.
5. A. Banerjee, I.S. Dhillon, J. Ghosh, and S. Sra, “Clustering on the unit hypersphere using Von Mises-Fisher distributions,” Journal of Machine Learning Research, vol. 6, pp. 1345–1382, 2005.
6. D. Blei, T. Griffiths, M.I. Jordan, and J. Tenenbaum, “Hierarchical Topic Models and the Nested Chinese Restaurant Process,” Advances in Neural Information Processing Systems, vol. 16 (106), pp. 168 – 180, 2004.
7. C. Buckley and A.F. Lewit, “Optimizations of inverted vector searches,” SIGIR ’85, pp. 97–110, 1985.
8. S. Dhillon, J. Fan, and Y. Guan, “Efficient clustering of very large document collections,” in Data Mining for Scientific and Engineering Applications, Kluwer Academic Publishers, Oxford, 2001.
9. U. Erra, S. Senatore, F. Minnella, and G. Caggianese, “Approximate TF-IDF based on topic extraction from massive message stream using the GPU,” Information Sciences, vol. 292, pp. 143–161, 2015.
10. D. Harman, “Overview of the first text retrieval conference (TREC-1),” in Proceedings of the First Text Retrieval Conference (TREC-1), DIANE Publishing, pp. 1–20, 1979.
11. J. A. Hartigan and M.A. Wong, “A  $k$ -means clustering algorithm,” Applied Statistics, vol. 28, pp. 100–108, 1979.
12. A. K. Jain, M. N. Murty, and P. J. Flynn, “Data Clustering: A Review,” ACM Comput. Surv., vol. 31 (3), pp. 264–323, 1999.
13. J. Kogan, “Introduction to Clustering Large and High-dimensional Data,” Cambridge University Press, 2007.
14. G. Kowalski, “Information Retrieval Systems Theory and Implementation,” Kluwer Academic Publishers, 1997.
15. J.Z.C. Lai and T.-J. Huang, “Fast global  $k$ -means clustering using cluster membership and inequality,” Pattern Recognition, vol. 43 (5), pp. 1954 – 1963, 2010.
16. D. D. Lewis, “Reuters-21578 Text Categorization Collection Distribution 1.0,” 1997. <http://kdd.ics.uci.edu/databases/reuters21578/reuters21578.html>, 1997.
17. Y. Liu, S. Xiao, X. Lv, and S. Shi, “Research on  $k$ -Means text clustering algorithm based on semantic,” in Proc. 10th International Conference on Computing, Control and Industrial Engineering (CCIE’10), vol. 1, pp. 124–127, 2010.

18. J. Ma, "Improved k-Means algorithm in text semantic clustering," *The Open Cybernetics Systemics Journal*, vol. 8, pp. 530–534, 2014.
19. A. McCallum, K. Nigam, J. Rennie, and K. Seymore, "Automating the construction of internet portals with machine learning," *Information Retrieval*, vol. 3(2), pp. 127 – 163, 2000.
20. C. Qimin, G. Qiao, W. Yongliang, and W. Xianghua, "Text clustering using VSM with feature clusters," *Neural Computing and Applications*, vol. 26 (4), pp. 995–1003, 2015.
21. J. Rennie, "The 20 newsgroups data set," 2008. <http://qwone.com/~jason/20Newsgroups>, 1997.
22. B. Ordin and A. M. Bagirov, "A heuristic algorithm for solving the minimum sum-of-squares clustering problems," *Journal of Global Optimization*, vol. 61, pp. 341–361, 2015.
23. S. Salton and C. Buckley, "Term-weighting approaches in automatic text retrieval," *Information Processing and Management*, vol. 24 (5), pp. 513–523, 1988.
24. G. Salton and M.J. McGill, "Introduction to Modern Retrieval," McGraw-Hill Book Company, New York, 1983.
25. S. Seifollahi, A. Bagirov, R. Layton, and I. Gondal, "Optimization based clustering algorithms for authorship analysis of phishing emails," *Neural Processing Letters*, pp. 1–15, 2017.
26. C. J. Van Rijsbergen, "Information Retrieval," Butterworth, London, second edition, 1989.
27. WebKB, Available electronically at <http://www.cs.cmu.edu/~WebKB>.
28. J. Yi, Y. Zhang, X. Zhao, and J. Wan, "A novel text clustering approach using deep-learning vocabulary network," *Mathematical Problems in Engineering*, vol. 1, pp. 1–13, 2017.
29. W. Zhang, T. Yoshida, and X. Tang, "A comparative study of TF-IDF, LSI and multi-words for text classification," *Expert Systems with Applications*, vol. 38, pp. 2758–2765, 2011.