Fast and Simple Deterministic Seeding of KMeans for Text Document Clustering

Ehsan Sherkat, Julien Velcin, Evangelos E. Milios

Fast and Simple Deterministic Seeding of KMeans for Text Document Clustering

Ehsan Sherkat[1(✉)], Julien Velcin[2], and Evangelos E. Milios[1]

[1] Dalhousie University, Halifax, Canada
ehsansherkat@dal.ca, eem@cs.dal.ca
[2] University Lyon 2, Lyon, France
julien.velcin@univ-lyon2.fr

Abstract. KMeans is one of the most popular document clustering algorithms. It is usually initialized by random seeds that can drastically impact the final algorithm performance. There exists many random or order-sensitive methods that try to properly initialize KMeans but their problem is that their result is non-deterministic and unrepeatable. Thus KMeans needs to be initialized several times to get a better result, which is a time-consuming operation. In this paper, we introduce a novel deterministic seeding method for KMeans that is specifically designed for text document clustering. Due to its simplicity, it is fast and can be scaled to large datasets. Experimental results on several real-world datasets demonstrate that the proposed method has overall better performance compared to several deterministic, random, or order-sensitive methods in terms of clustering quality and runtime.

AQ1

Keywords: Document clustering · Text · KMeans initialization
Deterministic

1 Introduction

The objective of KMeans is to assign similar data points to the same cluster while they are dissimilar to other clusters. The gradient descent method is usually used for optimizing the objective function and due to the non-convex nature of KMeans, the initial seeds play an important role in the quality of the clustering. There are several research works that try to provide good seeds for the KMeans. These methods can be divided into two major categories of non-deterministic and deterministic methods [12].

The non-deterministic methods are random or order-sensitive in nature. KMeans++ is a well known seeding method that incrementally selects initial seeds one at a time [3]. In each step, a data point is selected with a probability proportional to the minimum distance to the previously selected seeds. Because the first seed in KMeans++ is determined randomly and next seeds are selected based on a probabilistic method, the initial seeds are not repeatable. The KMC2 method improves the KMeans++ sampling step by Markov chain Monte Carlo

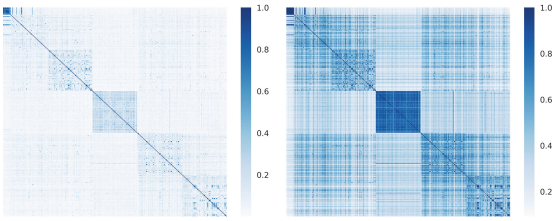
N. Ferro et al. (Eds.): CLEF 2018, LNCS 11018, pp. 1–13, 2018.
https://doi.org/10.1007/978-3-319-98932-7_7

based approximation [4]. Similarly to KMeans++, KMC2 starts with a uniformly random seed then the next seeds are selected by Markov chains of size m. The key factor for speeding up the KMC2 is that for each seed selection, it does not need to fully pass through all the data points and it only needs to compute the distance between m data points and previously selected seeds. The m is a fixed value, independent of the number of data points.

While there are many non-deterministic seeding methods, there exist few deterministic ones. The deterministic approaches need to be run only once and it makes them more practical for larger datasets. The comparison between different deterministic methods is presented by [11]. The KKZ method is one of the first deterministic seeding methods for KMeans [17]. It first sorts the data points by their vector's norm and the one with the highest value is selected as the first seed. The next seeds will be selected from data points that have the largest distance to the closest previously selected seeds. The most important drawback of this method is that it is sensitive to outliers. To avoid selecting an outlier as the initial seed, the ROBIN approach [16] uses local outlier factor (LOF) method [9]. This method first starts with a reference point r that usually is the origin of data points. Then it sorts the data points in decreasing order of their minimum distances from r. It then traverses the sorted list and selects the first non-outlier node, based on its LOF value. For the next steps, it sorts the data points in decreasing order by their minimum distance to the previous seeds and, again, the first non-outlier node is the next seed. The LOF method is not applicable to high dimensional and sparse datasets, which is an important issue in textual document collections [2].

The PCA-part and VAR-part are two popular deterministic hierarchical initialization methods for KMeans [21]. They start with all data points as a single cluster and then divide the data point into two halves based on Principle Component Analysis (PCA) [1]. This process continues and at each step, the half with largest average distance to its centroid is divided into two parts until the required number of seeds is reached. The result of the previous steps is an approximate clustering of data points; the centroid of the clusters are used for initializing KMeans.

There are some applications that require determinism. Interactive document clustering is a task that involves a human domain expert in the clustering procedure [7]. First, the clustering algorithm provides the user with initial clustering results, then the user provides feedback to reflect her idea of a meaningful clustering. If the initial result is non-deterministic, the user may get confused by the inconsistent clustering result. It is possible to store the initial data points to make the result of a non-deterministic method repeatable, but it may lead to a bad quality solution unless one initializes the clustering algorithm several times and then consider the one which has optimized the objective function the most, which is a very time-consuming process. In a medical domain, such as cancer subtype prediction, it is essential to have deterministic clusters for making a consistent decision and for being able to compare the clustering results with other clustering algorithms [20]. There is a particular treatment plan for each



(a) Pairwise cosine similarity (b) Pairwise *dsim* similarity

Fig. 1. The comparative result of pairwise cosine and *dsim* similarity of Newsgroup5 dataset. The darker color indicates the higher similarity between two documents. The documents are sorted by their class labels and five clusters are clearly detectable in both heatmaps. (Color figure online)

cancer subtype and in case that a subtype is clustered differently with different seeds it may impact the patients treatment procedure.

In this paper, we introduce a simple deterministic seeding method for KMeans algorithm, called DSKM (Deterministic Seeding KMeans), with the target of text document clustering. The proposed method is not only deterministic and reproducible but also improves the overall clustering results. The proposed method tries to find initial seeds that are as diverse as possible which consequently lead to a better clustering result. The KMeans need to be initialized by DSKM only once and this makes it fast and can be applied on large datasets. The code to the paper is publicly available[1].

2 Proposed Method

The key idea of the proposed method is to select k data points that are far from each other and, at the same time, have a high L_1 norm. These data points are used to initialize the KMeans algorithm. Steps of the proposed method are described in the following.

Step 1. First the document vectors are created based on terms of document collection after removing numbers, punctuations and stop-words. The document-term matrix produced as a result of this step is the input of the Algorithm 1. Let D be the set of documents and d a document in D. The TF-IDF weight of term w in document d is defined as Eq. 1, which has smoothed variant of the IDF.

$$TF\_IDF(w, d, D) = f(w, d) \times log \frac{|D| + 1}{|x \in D : w \in x| + 1} + 1. \tag{1}$$

where $f(w, d)$ is the frequency of term w in document d. Each document vector is then normalized by the L_2 norm. The high dimension of vectors may impact the results of the clustering algorithm. To reduce the dimension, we use a

[1] https://github.com/ehsansherkat/DSKM.

Algorithm 1. Deterministic seeding KMeans (DSKM)

input : k: Number of clusters, Data$^{|D|\times|W|}$: document-term matrix `// Step1`
output: S:$\{s_1, s_2, ..., s_k\}$ = Set of seed documents index

Function $T(s_i)$: `// Threshold function`
  return $\frac{1}{|D|}\sum_{j=1}^{|D|} dsim(d_j, s_i)$;
end
$C^{|D|\times|D|} \leftarrow$ pairwise-similarity(Data, 'cosine');
A:$\{a_1, a_2, ..., a_{|D|}\} \leftarrow$ sort(Data, 'L_1 norm');
$s_0 \leftarrow$ C$[a_1]$ `// Set starting point. C[i] is row vector. Step2;`
S $\leftarrow \{\}$
for $i \leftarrow 1$ **to** $|D|$ **do** `// Step 3`
  if *dsim(C[s_0], C[a_i])* $<$ *T(s_0)* **then**
    S $\leftarrow a_i$;
    break;
  end
end
while $|S| < k$ **do** `// Step 4`
  found $\leftarrow$ False;
  for $i \leftarrow 1$ **to** $|D|$ **do**
    if *dsim(C[s_j], C[a_i])* $<$ *T(s_j)*, $\forall s_j \in S, a_i \notin S$ **then**
      S $\leftarrow a_i$;
      found $\leftarrow$ True;
      break;
    end
  end
  if *found == False* **then**
    S $\leftarrow argmin(\sum_{j=1}^{|S|} dsim(a_i, s_j)), \forall a_i \in A, a_i \notin S$
  end
end
return S

simple but effective approach for pruning: the terms with a lower *mean-TF-IDF score* than the average mean-TF-IDF of all terms. For each term, the *mean-TF-IDF score* is calculated based on Eq. 2.

$$mean\_TF\_IDF(w, D) = \frac{1}{|D|} \times \sum_{d \in D} TF\_IDF(w, d, D). \tag{2}$$

Step 2. The rows of the document-term matrix are sorted by L_1 norm in a way that the first row of the matrix is the document with the highest L_1 norm. Documents with a higher L_1 norm have more impact on grouping similar documents because of having more key-terms. Therefore, we select the document with the highest L_1 norm as the starting data point (s_0). This procedure will generally not select an outlier document as a seed document.

Step 3. In the third step, we find a data point that is far from the starting data point and consider it as the first seed. Let $C^{|D|\times|D|}$ be the pairwise cosine similarity matrix between each pair of documents. Let c_{d_i} be the i-th row of C. c_{d_i} corresponds to the vector of similarities of document d_i with every other document. It has been shown that the cosine similarity is a better metric than the Euclidean distance for comparing textual documents [6]. We define the double similarity (*dsim*) between the document d_i to document d_j as Eq. 3.

$$dsim(d_i, d_j) = \frac{c_{d_i} \cdot c_{d_j}}{\|c_{d_i}\|_2 \left\|c_{d_j}\right\|_2}. \tag{3}$$

The insight for using *dsim* is that not only two documents, but also their similar documents, should be far from each other. Using *dsim* can help to achieve this goal. The comparison between heatmaps of pairwise cosine and *dsim* similarity of Newsgroup5 dataset is depicted in Fig. 1. The darker colors in the *dsim* heatmap indicates that two documents may have considerable number of common similar documents. It means that two documents may be more similar to each other if we compare their similar documents with each other than directly comparing them.

Let A be the list of document indexes sorted in decreasing order by their L_1 norm. The goal of the third step is finding the first document which has *dsim* similarity less than a specific threshold from the starting point (s_0) by traversing from the first of list A (Lines 8–13 Algorithm 1). Let S be the set of seed documents and $s_i \in S$ be the document index of seed i. The similarity threshold (Lines 1–3 Algorithm 1) is calculated based on Eq. 4.

$$T(s_i) = \frac{1}{|D|} \sum_{j=1}^{|D|} dsim(d_j, S_i). \tag{4}$$

$T(s_0)$ is the threshold for finding the first seed based on the starting data point s_0. We do not consider the starting data point as the first seed but we will give the chance for it to be selected in the next steps. Using Eq. 4 as the threshold prevents to select documents that are at the very end of list A which have low L_1 norm and less impact on grouping similar documents. After having found the first document s_1 that passes the threshold, we stop considering other documents and we add it to the seed document set S. Now, the seed documents set has the size of 1.

Step 4. We find $k-1$ seed documents in this step. Starting from the beginning of set $A - S$ and find the first document which is far from every seed in set S based on the threshold defined by Eq. 4. We iterate this step until k seeds are determined (Lines 16–22 Algorithm 1). If there is no document far from all the seeds in S, the following objective function is considered, with the goal of finding the document, which has the lowest cumulative *dsim* to every other seed document (Lines 23–25 Algorithm 1).

$$argmin(\sum_{j=1}^{|S|} dsim(d_i, s_j)), \quad 1 \leq i \leq |D|, \quad d_i \notin S. \tag{5}$$

This step ensures that the proposed method can always find k seed documents in every document collection.

After finding the initial seeds, we can directly initialize the KMeans algorithm. Based on our experiments, we can achieve a higher quality of result if for each seed document we find a few similar documents based on cosine similarity and then consider their centroid as the final seed. In our experiments, we extended each seed document with first 15 most similar documents to it by calculating the cosine similarity.

Complexity Analysis: Let $|D| = n$ be the number of documents and m the number of unique terms after applying Eq. 2 filter. The time complexity of sorting document-term-matrix and calculating the cosine similarity matrix is $O(n \log n)$ and $O(n^2 m/2)$ while the time complexity of finding seed documents based on $dsim$ is $O(n^2 k)$. Calculating the cosine similarity matrix is the most time-consuming step of the proposed method but it could easily be processed in parallel. In reality, the size of m will be less than a few thousand even for large textual datasets after selecting important terms, which makes the proposed approach practically feasible.

3 Experiments

In this section, first we introduce the baseline methods including state-of-the-art deterministic and non-deterministic initialization algorithms. The datasets' description and the evaluation metrics are in Sect. 3.2. Finally, the extensive experimental results is reported in Sect. 3.3.

Table 1. Description of datasets. The Eq. 2 is used for feature selection for the first 7 datasets and for the rest only stop-words and words with frequency less than 20 are removed.

#	Dataset	#Samples	#Dim.	#Classes	#	Dataset	#samples	#Dim.	#classes
1	Newsgroup5	400	1450	5	8	BBCsport	737	969	5
2	Yahoo6	600	2206	6	9	BBC	2225	3121	5
3	R8	7674	1997	8	10	Wikilow	4986	15441	10
4	Newsgroup20	18846	11556	20	11	WikiHigh	5738	17311	6
5	WebKB	4199	1578	4	12	Guardian	6520	10801	6
6	NewsSeparate	381	380	13	13	Irishtimes	3246	4823	7
7	SMS	5549	858	2					

3.1 Baseline Methods

We compared three random or order-sensitive seeding methods, Points, KMeans++, and KMC2 with the proposed method. In the Points method, uniformly k randomly selected data points are considered as the initial seeds for the KMeans algorithm. The KMeans++ is one of the most widely used seeding methods which has been demonstrated to achieve better performance result than the Points method [3]. KMeans++ starts with a random seed, then it tries to find the next one as far as possible from the first seed based on a probability sampling method called D^2-sampling. In this sampling method, data points that have higher distance to the previously selected seeds will more likely be selected as the next seed. This process continues until k initial seeds are detected. The KMC2 method is speeding up the KMeans++ algorithm by Markov chain Monte Carlo sampling based approximation [4]. It has been reported that the KMC2 has a better quality of results and computational cost than the KMeans++ algorithm. In our experiments, we used the assumption-free version of KMC2 with m equals to 200.

Two widely used deterministic seeding methods of PCA-part and VAR-part are compared with the proposed method. The PCA-part method hierarchically divides the data points into two halves based on PCA. First, it starts with calculating the centroid of all data points as a single cluster, and the principal eigenvector of the cluster covariance matrix. Second, it passes an hyperplane orthogonal to the principal eigenvector of the cluster which passes from the cluster centroid to create two sub-clusters. The sum distance of each data points in each sub-cluster to its centroid is calculated and the sub-cluster with a higher value is divided in the next step. Finally, this procedure is continued until k clusters are obtained. The VAR-part (variance partitioning) is an approximation to the PCA-part method [21]. In VAR-part the covariance matrix of the cluster is assumed to be diagonal. In each partitioning stage, the hyperplane is diagonal to the dimension with the largest variance. Based on our experiments, using the Euclidean distance leads to similar initialized seeds compared to cosine distance for VAR-par and PCA-part in all datasets; therefore we used the Euclidean distance for both methods.

In our experiments, we used the Spherical version of the KMeans algorithm. In Spherical KMeans the feature vectors is projected to the unit sphere equipped with the cosine similarity which performs better than Euclidean distance for text document clustering [14]. We compared the Spherical KMeans with different seeding methods with Fuzzy CMeans and Von Mises-Fisher Mixture methods. In the Fuzzy CMeans algorithm the data points can belong to more than one cluster with different membership values rather than distinct membership to only one cluster [8]. In our experiments, we used cosine similarity for the distance measure of the Fuzzy CMeans. The Von Mises-Fisher Mixture methods is a mixture model for clustering data distributed on the unit hypersphere based on Von Mises-Fisher distribution [5].

Table 2. Comparing precision of seeds. The average (±std) over 50 runs is reported for the Points, KMeans++, and KMC2 methods.

Dataset	DSKM	Points	KMeans++	KMC2
Newsgroup5	**0.800**	0.684 ± 0.145	0.636 ± 0.182	0.692 ± 0.134
Yahoo6	**1.000**	0.700 ± 0.115	0.613 ± 0.131	0.677 ± 0.070
R8	**0.750**	0.393 ± 0.120	0.495 ± 0.135	0.443 ± 0.137
Newsgroup20	**0.700**	0.634 ± 0.064	0.617 ± 0.072	0.638 ± 0.060
WebKB	**1.000**	0.660 ± 0.179	0.610 ± 0.151	0.655 ± 0.165
NewsSeparate	**0.846**	0.582 ± 0.084	0.563 ± 0.089	0.614 ± 0.103
SMS	**1.000**	0.620 ± 0.214	0.630 ± 0.219	0.610 ± 0.207
BBCsport	**0.800**	0.660 ± 0.140	0.576 ± 0.148	0.656 ± 0.133
BBC	**0.800**	0.668 ± 0.153	0.580 ± 0.146	0.688 ± 0.145
Wikilow	**0.800**	0.646 ± 0.090	0.556 ± 0.098	0.676 ± 0.111
WikiHigh	**0.833**	0.653 ± 0.152	0.627 ± 0.131	0.687 ± 0.123
Guardian	**1.000**	0.643 ± 0.105	0.577 ± 0.138	0.667 ± 0.120
Irishtimes	**0.857**	0.611 ± 0.114	0.509 ± 0.149	0.643 ± 0.112

3.2 Datasets and Evaluation Metrics

Datasets. The description of datasets is provided in Table 1. We obtained dataset Newsgroup5 by selecting 5 categories of the Newsgroup20[2] dataset each containing 80 randomly chosen documents. The Newsgroups20 dataset consists of nearly 20,000 messages of Internet news articles with 20 categories. The Yahoo6 is a sub-collection of questions and answers extracted from the Yahoo! Answers website [13]. We used 6 sub-categories with 100 randomly selected question and answer pairs. R8 is a subset of *Reuters-21578* dataset containing 8 categories and can be downloaded from Ana Cachopo's homepage[3]. The WebKB dataset consists of 4199 faculty, student, project, and course websites collected from the four universities on January 1997[4]. The NewsSeparate dataset is a subset of RSS news feeds from BBC, CNN, Reuters and Associated Press manually categorized into 13 categories [19]. The SMS dataset is a set of labeled SMS messages for spam research[5].

Datasets number 8 to 13 are taken from [15] and can be downloaded from their web-page[6]. The BBCsport, BBC, Irishtimes, and Guardian are news articles and WikiHigh and Wikilow are a subset of a Wikipedia dump from January 2014.

[2] http://qwone.com/~jason/20Newsgroups/.
[3] http://ana.cachopo.org.
[4] https://www.cs.cmu.edu/afs/cs/project/theo-20/www/data/.
[5] http://www.dt.fee.unicamp.br/~tiago/smsspamcollection/.
[6] http://mlg.ucd.ie/howmanytopics/index.html.

Table 3. Comparing clustering accuracy. For the deterministic approaches the McNemar's test is used. The P-value less than 0.05 indicates that the clustering algorithm does not have the same error rate as DSKM approach. The average over 50 runs with standard deviation is reported for the random or order-sensitive methods in which the m shows the minimum and the M shows the maximum of 50 runs.

Dataset	KMeans (DSKM)	KMeans (PCA-part)	KMeans (VAR-part)	KMeans (Points)	KMeans (KMeans++)	KMeans (KMC2)	Fuzzy CMeans (Points)	Von Mises Fisher Mixture
Newsgroup5	**0.850**	0.740 $p < 0.05$	0.525 $p < 0.05$	0.687 ± 0.082 m:0.522 M:0.91	0.696 ± 0.095 m:0.555 M:0.922	0.706 ± 0.080 m:0.542 M:0.912	0.719 ± 0.056 m:0.505 M:0.785	0.666 ± 0.073 m:0.497 M:0.820
Yahoo6	**0.850**	0.827 $p > 0.05$	0.803 $p < 0.05$	0.756 ± 0.079 m:0.553 M:0.847	0.740 ± 0.072 m:0.577 M:0.850	0.746 ± 0.070 m:0.620 M:0.843	0.798 ± 0.062 m:0.633 M:0.830	0.645 ± 0.052 m:0.457 M:0.757
R8	**0.688**	0.411 $p < 0.05$	0.537 $p < 0.05$	0.468 ± 0.060 m:0.332 M:0.605	0.476 ± 0.052 m:0.381 M:0.585	0.474 ± 0.064 m:0.361 M:0.612	0.457 ± 0.045 m:0.368 M:0.539	0.431 ± 0.054 m:0.271 M:0.513
Newsgroup20	0.485	**0.517** $p < 0.05$	0.386 $p < 0.05$	0.478 ± 0.037 m:0.399 M:0.565	0.496 ± 0.041 m:0.378 M:0.605	0.484 ± 0.039 m:0.410 M:0.595	0.119 ± 0.003 m:0.114 M:0.126	0.343 ± 0.024 m:0.303 M:0.407
WebKB	**0.65**	0.609 $p < 0.05$	0.529 $p < 0.05$	0.609 ± 0.029 m:0.521 M:0.669	0.604 ± 0.033 m:0.539 M:0.661	0.605 ± 0.039 m:0.529 M:0.692	0.603 ± 0.041 m:0.514 M:0.660	-
NewsSeparate	**0.861**	0.711 $p < 0.05$	0.766 $p < 0.05$	0.727 ± 0.072 m:0.562 M:0.89	0.713 ± 0.059 m:0.583 M:0.861	0.748 ± 0.066 m:0.622 M:0.864	0.747 ± 0.048 m:0.627 M:0.874	0.679 ± 0.066 m:0.507 M:0.824
SMS	0.597	0.904 $p < 0.05$	**0.907** $p < 0.05$	0.675 ± 0.142 m:0.502 M:0.907	0.646 ± 0.139 m:0.502 M:0.907	0.667 ± 0.143 m:0.505 M:0.907	0.797 ± 0.037 m:0.721 M:0.839	-
BBCsport	0.856	0.670 $p < 0.05$	**0.951** $p < 0.05$	0.783 ± 0.115 m:0.521 M:0.961	0.789 ± 0.117 m:0.620 M:0.958	0.800 ± 0.124 m:0.514 M:0.958	0.869 ± 0.123 m:0.626 M:0.955	0.803 ± 0.122 m:0.528 M:0.955
BBC	0.956	**0.958** $p > 0.05$	0.953 $p > 0.05$	0.870 ± 0.116 m:0.654 M:0.962	0.817 ± 0.133 m:0.493 M:0.965	0.833 ± 0.142 m:0.443 M:0.965	0.948 ± 0.035 m:0.704 M:0.953	0.809 ± 0.108 m:0.539 M:0.953
Wikilow	0.763	**0.969** $p < 0.05$	0.834 $p < 0.05$	0.803 ± 0.101 m:0.466 M:0.968	0.771 ± 0.096 m:0.581 M:0.964	0.793 ± 0.097 m:0.477 M:0.967	0.843 ± 0.075 m:0.702 M:0.964	0.751 ± 0.067 m:0.590 M:0.870
WikiHigh	0.715	**0.861** $p < 0.05$	0.658 $p < 0.05$	0.774 ± 0.087 m: 0.544 M:0.88	0.774 ± 0.087 m:0.487 M:0.890	0.785 ± 0.069 m:0.655 M:0.874	0.851 ± 0.026 m:0.712 M:0.867	0.629 ± 0.062 m:0.496 M:0.730
Guardian	**0.951**	0.951 $p > 0.05$	0.950 $p > 0.05$	0.834 ± 0.104 m: 0.583 M:0.954	0.832 ± 0.108 m:0.574 M:0.955	0.837 ± 0.121 m:0.554 M:0.954	0.945 ± 0.013 m:0.856 M:0.947	0.851 ± 0.097 m:0.661 M:0.945
Irishtimes	**0.871**	0.772 $p < 0.05$	0.626 $p < 0.05$	0.695 ± 0.083 m: 0.518 M:0.871	0.671 ± 0.085 m:0.505 M:0.837	0.678 ± 0.084 m:0.498 M:0.827	0.784 ± 0.074 m:0.625 M:0.877	0.704 ± 0.059 m:0.574 M:0.837

Evaluation Metrics. The clustering quality is measured by two widely used document clustering evaluation metrics of Normalized Mutual Information (NMI) and Accuracy (Acc) [10]. These metrics generate values between 0 and 1 in which values closer to 1 shows better performance. To match the predicted labels with actual labels for calculating the accuracy, we used the Hungarian method [18].

We compare the precision of initial seeds of methods defined by Eq. 6. The true label of each initial seed is used to find the diversity of seed labels. The method with more diverse (their true labels be different) initial seeds is better because it is able to introduce a better representative seed for each cluster. The comparative result of seed precision of evaluation methods is given in Table 2. The PCA-part and VAR-part produce initial centroids instead of initial seeds so it is not possible to evaluate their seed precision.

$$SeedPrecision = \frac{\#diverse\ labels}{k}. \tag{6}$$

The NMI score of the proposed method compared to other methods is summarized in Table 4. The DSKM outperforms in most of the datasets. The same trend of performance similar to the accuracy score can be observed for NMI score as well. KMC2 has slightly better NMI score compared to KMeans++ and Points.

Table 4. Comparing clustering NMI score. The average 50 runs with standard deviation is reported forthe random or order-sensitive approaches in which the m shows the minimum and the M shows the maximum of 50 runs.

Dataset	KMeans (DSKM)	KMeans (PCA-part)	KMeans (VAR-part)	KMeans (Points)	KMeans (KMeans++)	KMeans (KMC2)	Fuzzy CMeans (Points)	Von Mises Fisher Mixture
Newsgroup5	**0.781**	0.742	0.513	0.663 ± 0.075 m:0.437 M:0.829	0.667 ± 0.074 m:0.511 M:0.821	0.665 ± 0.066 m:0.550 M:0.815	0.663 ± 0.032 m:0.537 M:0.706	0.622 ± 0.069 m:0.442 M:0.777
Yahoo6	**0.704**	0.684	0.645	0.631 ± 0.043 m:0.492 M:0.700	0.621 ± 0.044 m:0.538 M:0.693	0.629 ± 0.041 m:0.532 M:0.694	0.664 ± 0.028 m:0.585 M:0.678	0.538 ± 0.03 m:0.449 M:0.615
R8	**0.575**	0.534	0.509	0.515 ± 0.032 m:0.420 M:0.567	0.520 ± 0.027 m:0.460 M:0.580	0.527 ± 0.029 m:0.453 M:0.600	0.480 ± 0.032 m:0.425 M:0.548	0.397 ± 0.057 m:0.260 M:0.495
Newsgroup20	**0.539**	0.533	0.467	0.498 ± 0.023 m:0.453 M:0.554	0.509 ± 0.028 m:0.439 M:0.578	0.501 ± 0.024 m:0.456 M:0.567	0.234 ± 0.002 m:0.232 M:0.239	0.412 ± 0.018 m:0.365 M:0.45
WebKB	**0.388**	0.320	0.353	0.362 ± 0.017 m:0.324 M:0.396	0.362 ± 0.017 m:0.316 M:0.395	0.363 ± 0.016 m:0.322 M:0.394	0.349 ± 0.023 m:0.307 M:0.377	-
NewsSeparate	**0.872**	0.809	0.829	0.819 ± 0.035 m:0.729 M:0.899	0.813 ± 0.033 m:0.742 M:0.882	0.833 ± 0.031 m:0.777 M:0.893	0.826 ± 0.022 m:0.767 M:0.894	0.77 ± 0.036 m:0.679 M:0.868
SMS	0.123	0.409	**0.414**	0.120 ± 0.128 m:0.000 M:0.414	0.135 ± 0.115 m:0.002 M:0.413	0.140 ± 0.119 m:0.000 M:0.414	0.267 ± 0.043 m:0.165 M:0.317	-
BBCsport	0.761	0.716	**0.858**	0.742 ± 0.077 m:0.583 M:0.881	0.742 ± 0.079 m:0.578 M:0.876	0.752 ± 0.089 m:0.570 M:0.876	0.816 ± 0.066 m:0.692 M:0.869	0.743 ± 0.091 m:0.461 M:0.876
BBC	0.865	**0.871**	0.857	0.806 ± 0.072 m:0.663 M:0.880	0.772 ± 0.091 m:0.536 M:0.891	0.774 ± 0.090 m:0.557 M:0.889	0.851 ± 0.020 m:0.708 M:0.856	0.718 ± 0.086 m:0.494 M:0.859
Wikilow	0.867	**0.934**	0.897	0.862 ± 0.040 m:0.730 M:0.933	0.853 ± 0.037 m:0.781 M:0.930	0.862 ± 0.039 m:0.740 M:0.931	0.879 ± 0.027 m:0.825 M:0.927	0.774 ± 0.031 m:0.713 M:0.832
WikiHigh	0.723	**0.740**	0.642	0.707 ± 0.047 m:0.580 M:0.761	0.702 ± 0.043 m:0.548 M:0.764	0.704 ± 0.037 m:0.633 M:0.759	0.721 ± 0.016 m:0.620 M:0.727	0.552 ± 0.041 m:0.479 M:0.667
Guardian	**0.862**	**0.862**	0.861	0.805 ± 0.054 m:0.627 M:0.870	0.803 ± 0.056 m:0.647 M:0.872	0.807 ± 0.062 m:0.644 M:0.871	0.852 ± 0.015 m:0.748 M:0.856	0.786 ± 0.055 m:0.639 M:0.848
Irishtimes	**0.783**	0.720	0.642	0.681 ± 0.049 m:0.575 M:0.759	0.672 ± 0.055 m:0.564 M:0.761	0.680 ± 0.052 m:0.573 M:0.761	0.741 ± 0.032 m:0.666 M:0.780	0.672 ± 0.036 m:0.595 M:0.740

3.3 Experimental Results

The accuracy result of the DSKM in comparison to other methods is summarized in Table 3. For random or order-sensitive methods, we report the average over 50 runs with its standard deviation, the minimum, and the maximum result. In order to have a fair comparison, we only initialize KMeans once for the non-deterministic methods. For the PCA-part and VAR-part methods, the McNemar's test is applied to determine whether their clustering result has the same error rate as DSKM. The Hungarian algorithm is used to map the cluster labels to actual labels. The deterministic approaches are superior in accuracy score compared to the average score of random or order-insensitive methods. Better performance result for deterministic methods on non-textual and Synthetic datasets has been reported by [12]. A possible reason is that the deterministic methods are running once and the seeding step can be viewed as an approximate clustering of data points. The DSKM method has similar or even better accuracy compared to the maximum accuracy score of the random or order-sensitive methods on Yahoo6, R8, WebKB, NewSeparate, BBC, Guardian, and Irishtimes. The SMS dataset is an unbalanced dataset and DSKM does not perform well on it although it was able to find 100% diverse initial seeds (Table 2). PCA-part, and VAR-part performed well on the SMS dataset which demonstrates their effectiveness for unbalanced datasets. Fuzzy CMeans has the best average and Von Mises Fisher Mixture the lowest accuracy score on most of the datasets

Table 5. Running time (seconds) of seeding methods. A random single run of KMeans++ and KMC2 is reported. Datasets are sorted by the sample size.

Dataset	DSKM	PCA-part	VAR-part	KMeans++	KMC2
NewsSeparate	0.03	0.74	0.03	0.02	**0.01**
Newsgroup5	**0.03**	5.27	**0.03**	0.05	**0.03**
Yahoo6	0.02	10.08	0.04	**0.01**	0.02
BBCsport	0.06	4.79	0.03	**0.01**	**0.01**
BBC	0.38	94.93	0.39	0.08	**0.07**
Irishtimes	0.99	410.3	1.25	0.22	**0.14**
WebKB	0.72	-	-	0.11	**0.06**
Wikilow	7.02	6849.23	5.62	1.45	**0.70**
SMS	0.77	7.92	0.11	**0.02**	0.03
WikiHigh	8.75	8725.6	5.59	1.21	**0.71**
Guardian	6.02	3681.96	3.88	0.82	**0.44**
R8	3.22	172.67	0.90	**0.25**	0.47
Newsgroup20	55.96	19712.72	39.56	8.28	**6.28**

among random or order-sensitive methods. On the Newsgroup20 dataset, Fuzzy CMeans does not perform well, which indicates that this method has difficulty on large datasets with a high number of clusters. The Points, KMeans++, and KMC2 have similar average accuracy results on most datasets. This shows that KMeans++ and KMC2 are performing better for very large datasets which is a case for Newsgroup20 and R8 datasets.

We compared the running time of the seeding methods in Table 5. Although the PCA-part has better performance result than the VAR-part, its running time makes it not practical for large datasets. The DSKM method has acceptable running time even for large datasets. The KMC2 is the fastest seeding algorithm compared to the others and based on its accuracy and NMI performance, it is the best random or order-sensitive method. Due to the random nature of the KMeans++ and KMC2, the Kmeans is initialized several times by them and the clustering which optimizes the KMeans objective function is selected. The impact of the number of initializations on the accuracy performance of the KMeans++ and KMC2 for NewsSeparate is depicted in Fig. 2. In order to have stable results, we reported the average of 50 runs for KMC2 and KMeans++. As the number of initialization increases, the accuracy of the KMC2 and KMeans++ increases and converges to a stable value. On the other hand, the running time increases as the number of initializations is increased. This indicates that the DSKM method could be even faster than the random or order-sensitive methods in practice because it does not need to run several times.

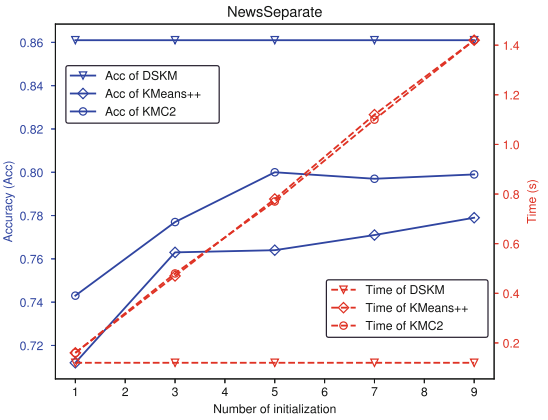


Fig. 2. The impact of number of initialization on the Accuracy performance and running time. Each initialization of the KMeans++ and KMC2 is the result of average 50 runs.

4 Conclusion

In this paper, we propose a new deterministic seeding algorithm for the KMeans algorithm called DSKM. The key idea of the DSKM is that the initial seeds should be as far as possible from each other. Two data points that not only themselves but their similar documents are less similar to each other are good candidates and that is why we defined the *dsim* similarity. For finding seeds we start from documents with higher L_1 norm. Experimental results on several real world textual datasets shows that DSKM outperforms the other deterministic, random or order-sensitive methods in terms of clustering accuracy and NMI score. The proposed methods have an acceptable running time even for large datasets.

References

1. Abdi, H., Williams, L.J.: Principal component analysis. Wiley Interdiscip. Rev.: Comput. Stat. **2**(4), 433–459 (2010)
2. Aggarwal, C.C., Yu, P.S.: Outlier detection for high dimensional data. In: Proceedings of the 2001 ACM SIGMOD International Conference on Management of Data, SIGMOD 2001, pp. 37–46. ACM, New York (2001)
3. Arthur, D., Vassilvitskii, S.: K-means++: the advantages of careful seeding. In: Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2007, pp. 1027–1035. Society for Industrial and Applied Mathematics, Philadelphia (2007)
4. Bachem, O., Lucic, M., Hassani, H., Krause, A.: Fast and provably good seedings for k-means. In: Lee, D.D., Sugiyama, M., Luxburg, U.V., Guyon, I., Garnett, R. (eds.) Advances in Neural Information Processing Systems 29, pp. 55–63. Curran Associates, Inc. (2016)

5. Banerjee, A., Dhillon, I.S., Ghosh, J., Sra, S.: Clustering on the unit hypersphere using von mises-fisher distributions. J. Mach. Learn. Res. **6**, 1345–1382 (2005)
6. Basu, S., Davidson, I., Wagstaff, K.: Constrained Clustering: Advances in Algorithms, Theory, and Applications, 1st edn. Chapman & Hall/CRC, Boca Raton (2008)
7. Bekkerman, R., Raghavan, H., Allan, J., Eguchi, K.: Interactive clustering of text collections according to a user-specified criterion. In: Proceedings of the 20th International Joint Conference on Artifical Intelligence, IJCAI 2007, pp. 684–689. Morgan Kaufmann Publishers Inc., San Francisco (2007)
8. Bezdek, J.C.: A convergence theorem for the fuzzy ISODATA clustering algorithms. IEEE Trans. Pattern Anal. Mach. Intell. **2**(1), 1–8 (1980)
9. Breunig, M.M., Kriegel, H.P., Ng, R.T., Sander, J.: LOF: identifying density-based local outliers. In: Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data, SIGMOD 2000, pp. 93–104. ACM, New York (2000)
10. Cai, D., He, X., Han, J.: Document clustering using locality preserving indexing. IEEE Trans. Knowl. Data Eng. **17**(12), 1624–1637 (2005)
11. Celebi, M.E., Kingravi, H.A.: Linear, deterministic, and order-invariant initialization methods for the k-means clustering algorithm. In: Celebi, M.E. (ed.) Partitional Clustering Algorithms, pp. 79–98. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-09259-1_3
12. Celebi, M.E., Kingravi, H.A., Vela, P.A.: A comparative study of efficient initialization methods for the k-means clustering algorithm. Expert Syst. Appl. **40**(1), 200–210 (2013)
13. Chang, M., Ratinov, L., Roth, D., Srikumar, V.: Importance of semantic representation: dataless classification. In: AAAI, July 2008
14. Dhillon, I.S., Modha, D.S.: Concept decompositions for large sparse text data using clustering. Mach. Learn. **42**(1–2), 143–175 (2001)
15. Greene, D., O'Callaghan, D., Cunningham, P.: How many topics? stability analysis for topic models. In: Calders, T., Esposito, F., Hüllermeier, E., Meo, R. (eds.) ECML PKDD 2014 Part I. LNCS (LNAI), vol. 8724, pp. 498–513. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-662-44848-9_32
16. Hasan, M.A., Chaoji, V., Salem, S., Zaki, M.J.: Robust partitional clustering by outlier and density insensitive seeding. Pattern Recogn. Lett. **30**(11), 994–1002 (2009)
17. Katsavounidis, I., Kuo, C.C.J., Zhang, Z.: A new initialization technique for generalized Lloyd iteration. IEEE Sig. Process. Lett. **1**(10), 144–146 (1994)
18. Kuhn, H.W.: The Hungarian method for the assignment problem. Nav. Res. Logist. Q. **2**(1–2), 83–97 (1955)
19. Martins, R.M., Coimbra, D.B., Minghim, R., Telea, A.: Visual analysis of dimensionality reduction quality for parameterized projections. Comput. Graph. **41**, 26–42 (2014)
20. Nidheesh, N., Nazeer, K.A., Ameer, P.: An enhanced deterministic k-means clustering algorithm for cancer subtype prediction from gene expression data. Comput. Biol. Med. **91**, 213–221 (2017)
21. Su, T., Dy, J.G.: In search of deterministic methods for initializing k-means and Gaussian mixture clustering. Intell. Data Anal. **11**(4), 319–338 (2007)