Expert Systems With Applications 150 (2020) 113288

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Improving spherical k-means for document clustering: Fast initialization, sparse centroid projection, and efficient cluster labeling

Hyunjoong Kim[a], Han Kyul Kim[a], Sungzoon Cho[a,b,*]

[a] Department of Industrial Engineering, Seoul National University, South Korea
[b] Institute for Industrial Systems Innovation, Seoul National University, South Korea

ARTICLE INFO

Article history:
Received 30 September 2018
Revised 23 July 2019
Accepted 5 February 2020
Available online 6 February 2020

Keywords:
Spherical k-means
Document clustering
k-means initialization
Sparse vector projection
Clustering labeling

ABSTRACT

Due to its simplicity and intuitive interpretability, spherical k-means is often used for clustering a large number of documents. However, there exist a number of drawbacks that need to be addressed for much effective document clustering. Without well-dispersed initial points, spherical k-means fails to converge quickly, which is critical for clustering a large number of documents. Furthermore, its dense centroid vectors needlessly incorporate the impact of infrequent and less-informative words, thereby distorting the distance calculation between the document vectors.

In this paper, we propose practical improvements on spherical k-means to overcome these issues during document clustering. Our proposed initialization method not only guarantees dispersed initial points, but is also up to 1000 times faster than previously well-known initialization method such as k-means++. Furthermore, we enforce sparsity on the centroid vectors by using a data-driven threshold that is capable of dynamically adjusting its value depending on the clusters. Additionally, we propose an unsupervised cluster labeling method that effectively extracts meaningful keywords to describe each cluster.

We have tested our improvements on seven different text datasets that include both new and publicly available datasets. Based on our experiments on these datasets, we have found that our proposed improvements successfully overcome the drawbacks of spherical k-means in significantly reduced computation time. Furthermore, we have qualitatively verified the performance of the proposed cluster labeling method by extracting descriptive keywords of the clusters from these datasets.

1. Introduction

Clustering is one of the most commonly used unsupervised algorithms for exploring or finding patterns in a dataset. Without any label information, it relies solely on the similarity or distance measure between the data points to group similar data points. Beside numerical data, clustering is also often applied on documents regardless of their different vector representations (Xie, Girshick, & Farhadi, 2016; Xu & Tian, 2015; Yang, Fu, Sidiropoulos, & Hong, 2017).

Previously, numerous document clustering algorithms such as density-based clustering (Ester, Kriegel, Xu, & thers, 1996), agglomerative clustering (Sibson, 1973) and graph-based clustering (Clauset, Newman, & oore, 2004) have been proposed. Although each of these algorithms use its own unique method to define a distance measure, they all require a huge computation time of $O(n^2)$ or more for calculating their distance measures. Therefore, the k-means algorithm is commonly used for clustering high dimensional and large document vectors as it requires significantly less computation time of $O(nk)$.

$$argmin_C \sum_{i=1}^{k} \sum_{x \in C_i} |x - c_i|^2 = argmin_C \sum_{i=1}^{k} |C_i| Var(C_i) \quad (1)$$

C : cluster index
C_i : cluster i
c_i : the centroid of cluster i
x : a vector of a data point
k : a number of clusters

By minimizing the variance of distance between the vectors within each cluster as in Eq. 1, the k-means algorithm represents each cluster by its centroid vector. Although finding an optimal solution for Eq. 1 is an NP problem, Lloyd's k-means algorithm (Lloyd, 1982) efficiently searches for local optimal solutions by following the procedures listed in Fig. 1. Although calculating pair-

* Corresponding author.
E-mail addresses: hyunjoong@dm.snu.ac.kr (H. Kim), hank@dm.snu.ac.kr (H.K. Kim), zoon@snu.ac.kr (S. Cho).

https://doi.org/10.1016/j.eswa.2020.113288
0957-4174/

D: input dataset
k: a number of clusters

def kmeans (D, k):
  $C \leftarrow$ Initialize k centroids with random sampling
  $L \leftarrow$ Assign every points to its closest centroid
  while (not converged):
    $C \leftarrow$ Update centroids by averaging assigned data points
    $L \leftarrow$ Reassign all the points to its closest centroid
  return C, L

Fig. 1. Lloyd's k-means.

wise distance between k centroids and n data points is a computational bottleneck, this cost is relatively small, requiring only $O(kn)$ (Coates & Ng, 2012). With low computation time and quick convergence, Lloyd's k-means algorithm, therefore, is frequently used for clustering large input data or generating representations based on intrinsic similarity measures (Coates, Ng, & Lee, 2011; He, Wen, & Sun, 2013; Jin, Li, Lin, & Cai, 2013).

Among numerous variants of k-means clustering, a spherical k-means algorithm (Dhillon & Modha, 2001) is often used for document clustering. Instead of Euclidean distance, it defines the distance between the clusters with cosine distance. As cosine similarity has previously been found to be effective in clustering sparse vectors (Huang, 2008), spherical k-means is especially effective in clustering sparse document vectors that are characterized by few defining features (Buchta, Kober, Feinerer, & Hornik, 2012; Dhillon, Guan, & Kogan, 2002).

In this paper, we propose an improved spherical k-means algorithm for effective document clustering. As the convergence rate and the results of spherical k-means algorithm is heavily influenced by the initial centroids, our enhanced spherical k-means algorithm proposes a computationally faster and well-dispersed initialization method. Furthermore, we preserve the sparsity of the centroid vectors, and suggest a cluster labeling method for intuitively interpreting the document clusters. Despite various document representation methods, we will limit our exploration to sparse document vectors such as term frequency - inverse document frequency (TF-IDF) (Sparck Jones, 1972) and Bag-of-Concepts (BOC) (Kim, Kim, & Cho, 2017).

The rest of this paper is structured as follows. In Section 2, we discuss widely known issues of the spherical k-means clustering algorithm, and previous attempts to resolve them. In Section 3 and 4, we propose our new methods to overcome these problems, and we verify their performances, respectively.

1. Select point c_1 randomly
2. Select next point c_t with probability $\frac{d(x)^2}{\sum_{x \in D} d(x)^2}$
3. Repeat step 2 until k points are chosen as initial points.

Fig. 3. k-means++.

2. Related works

In this section, we will explore some widely known issues of the spherical k-means clustering algorithm in document clustering. Before discussing these limitations, we will first define notations that will be used throughout this paper. C_i will indicate the i_{th} cluster, while c_i will represent its centroid vector. D will be used to denote the input dataset, while x_i will designate a single instance of D. Furthermore, k will indicate the number of clusters.

Initial centroids are crucial in the k-means clustering algorithm as well-chosen initial points will guarantee faster convergence (Arthur & Vassilvitskii, 2007). For example, dispersed initial points such as those in Fig. 2 (a) will provide faster and more stable clustering results as compared to those concentrated in a local neighboring region as in Fig. 2 (b).

Therefore, various initialization methods have been previously proposed to generate well dispersed initial centroids. The global k-means clustering algorithm sequentially discovers initial points from its sub k-means problems (Bagirov, 2008; Likas, Vlassis, & Verbeek, 2003). Starting with a sub k-means problem with k=1, it finds one optimal initial centroid. With this initial centroid fixed, it finds the subsequent optimal initial point by solving a sub k-means problem with k=2. Based on the initial points from previous sub k-means problems, the global k-means clustering algorithm repeats clustering k times to discover optimal initial points. However, its huge computation cost of $O(nk^2)$ renders it impractical when k is large.

Similarly, the k-means++ algorithm (Arthur & Vassilvitskii, 2007) overcomes the initialization issue by sequentially selecting a dispersed initial centroid. As summarized in Fig. 3, the first centroid is chosen randomly. The subsequent centroid is selected based on the value that is proportional to the square of $d(x)$, the minimal distance between the previously selected centroids and other possible candidates. However, k-means++ still requires $O(nk)$ computation time, incurring large computation cost when n and k are large. Furthermore, it is difficult to apply this algorithm to high-dimensional document vectors as most of pairwise cosine distance between the sparse vectors are close to one. For example, Table 1 shows that most pairwise cosine distance between the document vectors of various text datasets are indeed close to one.

Despite this critical computational bottleneck of initialization in k-means, recent research in the k-means algorithm has failed to address this fundamental issue. To reduce the training time of k-

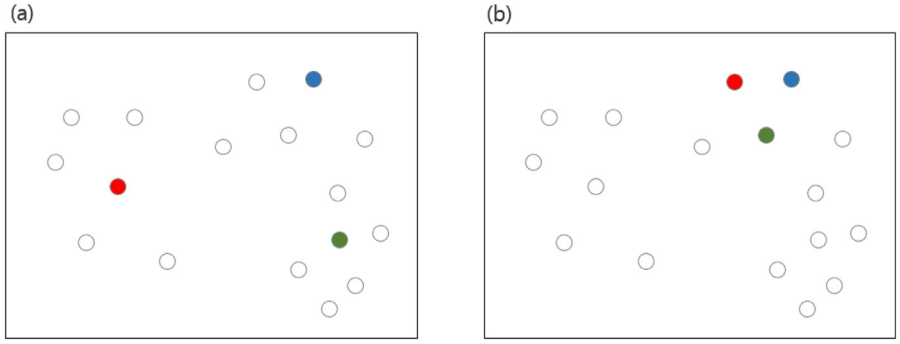

Fig. 2. (a) Good initial points, (b) Bad initial points.

Table 1
Distribution of Cosine distance between high-dimensional sparse document vectors from various text datasets. D1: A6 blogs, D2: Tuscon blogs, D3: Sonata blogs, D4: IMDb reviews, D5: Reuters RCV1, D6: MovieLens 20M, D7: Yelp reviews (in percentage).

Distance range	Dataset						
	D1	D2	D3	D4	D5	D6	D7
<= 0.7	0.249	0.59	0.323	0.272	0.045	1.456	0.01
0.7 - 0.8	0.378	1.121	0.455	7.751	0.067	2.386	0.286
0.8 - 0.9	1.628	3.89	1.984	57.271	0.316	12.458	11.073
0.9 - 1.0	97.745	94.399	97.239	34.706	99.573	83.7	88.632

1. Construct set D_{init}, $\alpha \times k$ randomly selected data from data D
2. Select c_i randomly from D_{init}
3. Remove x_j where $x_j \in D_{init}$ and $\cos(c_i, x_j) \geq t_{init}$
4. Repeat $2 \sim 3$ until k centroid points are selected or D_{init} becomes empty
5. If the number of selected centroid is less than k, then select remaining points from $D - D_{init}$ randomly

Fig. 4. Summary of the proposed initialization method.

means clustering for large datasets, the recent papers on k-means clustering propose new methods of approximating the algorithm (Bachem, Lucic, Hassani, & Krause, 2016; Capó, Pérez, & Lozano, 2017; Shen, Liu, Tsang, Shen, & Sun, 2017) or adapting it to computationally more powerful hardware systems (Bahmani, Moseley, Vattani, Kumar, & Vassilvitskii, 2012; Li, Zhao, Chu, & Liu, 2013; Shahrivari & Jalili, 2016). However, all of these recent advancements will be truly effective only when the fundamental initialization issue of k-means clustering itself is resolved.

During each iteration of spherical k-means for document clustering, the centroid vectors become dense as the frequencies of infrequent words are also being averaged to the centroid vectors. However, these infrequent words should not have a significant impact on defining the distance between the documents and centroids. Therefore, dense centroid vectors reduce the importance of more frequent and descriptive words, distorting the distance calculation during the clustering (Abualigah, Khader, Al-Betar, & Alomari, 2017). Furthermore, the sparse centroid vector provides better interpretability and requires less memory.

$$w_i \leftarrow sign(w_i) \times max(w_i - \theta, 0) \tag{2}$$

Previously, L1 ball projection based methods have been proposed to enforce the sparsity of the centroid vectors (Duchi, Shalev-Shwartz, Singer, & Chandra, 2008; Sculley, 2010; Shalev-Shwartz, Singer, Srebro, & Cotter, 2011). As expressed in Eq. 2, these methods reduce the L1 norm of a dense centroid vector by subtracting a threshold value θ from each element of the vector. However, sparsity is not guaranteed as a uniform θ is set for all of the clusters. In document clustering, a uniform θ is especially detrimental as word frequencies within a text dataset follows Zipf's law (Jurafsky, 2000). As the number of words with low frequency increases exponentially, finding an optimal value of θ is heavily dependent on the number of documents designated in each cluster. Therefore, applying a uniform θ that ignores the size of the cluster will create overly sparse centroid vectors for some clusters, while failing to enforce sparsity on other clusters. Furthermore, manual effort or subjective judgment is needed for selecting an appropriate value of θ.

3. Proposed methods

In this section, we propose our improved spherical k-means clustering algorithm for document clustering. For a large number of documents, our enhancement provides computationally an efficient initialization method, and ensures the sparsity of the centroid vectors. Additionally, we provide a method for interpreting the clusters.

3.1. Selecting initial points for high-dimensional sparse data

If initial centroids are located within neighboring regions as in Fig. 2 (b), the k-means algorithm fails to converge quickly. Although k-means++ overcomes this initialization problem by selecting distant initial points, it inefficiently requires $n \times k$ distance computations. Furthermore, it fails to select effective initial points for high-dimensional data as the majority of pairwise cosine distance calculations become meaningless as shown in Table 1. To ensure that distant sparse data points are selected during initialization, we propose our unique initialization method for document clustering as in Fig. 4.

Our proposed initialization algorithm begins by constructing D_{init}, a subset of the entire dataset D. It contains randomly selected $\alpha \times k$ data points, in which α is a predefined parameter. Empirically, we find α between 1.5 and 10 to be effective. Within D_{init}, an initial centroid c_i is randomly selected, and the remaining data points in D_{init} with cosine similarity equal to or greater than the predefined threshold t_{init} are removed. This process is repeated until k centroids are selected. If D_{init} becomes empty prior to selecting k centroids, the remaining centroids are randomly chosen from the data points within $D - D_{init}$.

As shown in Table 1, the majority of pairwise cosine distance between any two sparse document vectors are close to one. Exploiting this characteristic of a high-dimensional document vector space, our proposed initialization method successfully generates dispersed initial points while limiting its search space to D_{init}. Due to its limited search space, it requires at most $\alpha \times k^2$ distance computations, which is computationally cheaper than k-means++ as $\alpha \times k \ll n$. Furthermore, k-means++ does not guarantee dispersed initial points as it fails to incorporate the distance between previously selected and potential initial points. However, our proposed initialization method ensures dispersed initial points as it removes the neighboring regions of the previously selected initial point from further exploration (Fig. 5).

3.2. Preserving centroid sparsity

Since a centroid vector is an average of all document vectors within the cluster, the frequencies of rarely occurring words are

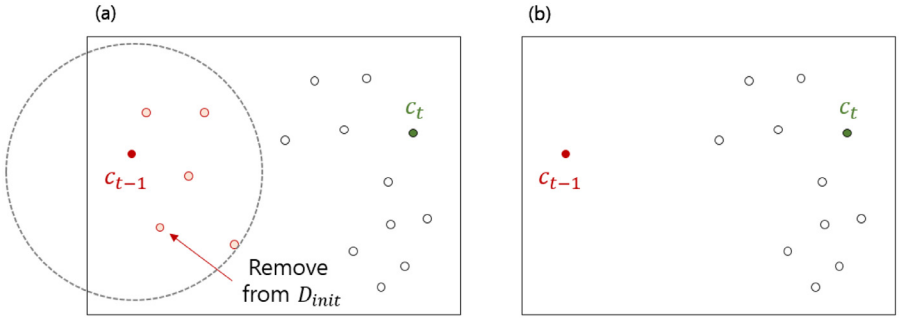

Fig. 5. Intuition behind the proposed initialization method. (a) Once an initial point is selected, the data points with their distance less than threshold t_{init} are removed (b) Subsequently selected initial point is bound to be distant.

D: input dataset
k: a number of clusters
α: a factor for determining the search space of initial points
t_{init}: minimum distance between initial points
β: sparsity threshold factor
k_0: a number of keyword candidates
k_1: a number of selected keywords

def improved_spherical_kmeans $(D, k, \alpha, t_{init}, \beta, k_0, k_1)$:
  $C \leftarrow$ Initialize k centroids with sparse initializer (D, k, α, t_{init})
  $L \leftarrow$ Assign every points to its closest centroid
  while (not converged):
    $C \leftarrow$ Update centroid. Averaging their comprising data points
    $C \leftarrow$ Project centroids to sparse vector (C, L, β)
    $L \leftarrow$ Re-assign all the points to its closest centroid
  $Z \leftarrow$ cluster labeling (C, L, k_0, k_1)
  return C, L, Z

Fig. 6. Pseudocode for our improved spherical k-means.

Table 2
Our proposed enhancements and criteria for validating their respective performances.

Enhancements	Criteria for validation
Dispersed initialization	* Initialization speed * Stability of the clustering result
Centroid sparsity	* Sparsity of centroids * Stability of the clustering result after applying sparse centroid representation
Cluster labeling	* Qualitative analysis

also used as its features. Since these infrequent words are less informative than frequent words, we propose a method for removing their effects from the centroid vector, thereby increasing its sparsity.

$$c_{ij} \leftarrow sign(c_{ij}) \times max\left(|c_{ij}| - \frac{\beta}{DF(C_i)}, 0\right) \quad (3)$$

In order to preserve the sparsity of the centroid vectors, we first define $DF(C_i)$, the number of documents in cluster i. If the absolute value of each feature c_{ij} within the centroid vector of c_i is less than $\frac{\beta}{DF(C_i)}$, the value is reduced to zero as summarized by Eq. 3. By relying on the number of documents in each cluster for calculating $\frac{\beta}{DF(C_i)}$, our method is capable of dynamically defining the level of infrequency based on the size of each cluster. Therefore, our proposed method successfully prevents infrequent and trivial words from affecting the result of document clustering without generating excessively sparse or unchanged centroid vectors as in the case of L1 projection.

3.3. Interpreting document clusters

Document clustering often occurs in unsupervised learning settings. As there are no true class labels, unsupervised methods for labeling document clusters are crucial for extracting useful insights from the input data. Previously, unsupervised document labeling has been actively researched in the field of topic modeling (Blei, Ng, & Jordan, 2003; Chuang, Manning, & Heer, 2012a; Chuang, Ramage, Manning, & Heer, 2012b; Newman, Noh, Talley, Karimi, & Baldwin, 2010; Sievert & Shirley, 2014; Snyder, Knowles, Dredze, Gormley, & Wolfe, 2013). Similar to these previous studies in topic

Table 3
Seven datasets used for our experiment.

Data	Number of documents	Total number of words	Number of nonzero elements	Percentage of nonzero elements
A6 blogs	63,153	91,302	18,051,341	0.313 %
Tucson blogs	105,755	81,497	29,192,999	0.339 %
Sonata blogs	229,253	85,129	60,861,803	0.312 $
IMDb reviews	1,228,348	68,049	181,411,713	0.217 %
Reuter RCV1	804,414	47,236	60,915,113	0.160 %
MovieLens 20M	138,493	131,262	20,000,263	0.110 %
Yelp reviews	5,261,669	27,247	365,341,887	0.255 %

Table 4
Relative speed-up of our proposed method's initialization compared to k-means++ (in seconds).

Data	α	k			
		10	20	50	100
A6 blogs	1	x 288	x 268	x 260	x 233
	3	x 265	x 257	x 213	x 150
	5	x 248	x 226	x 166	x 99
	10	x 217	x 159	x 100	x 54
	k-means+	5 s	10 s	25 s	52 s
Tucson blogs	1	x 464	x 487	x 397	x 367
	3	x 388	x 440	x 306	x 244
	5	x 358	x 376	x 261	x 172
	10	x 312	x 279	x 160	x 102
	k-means+	7 s	16 s	41 s	82 s
Sonata blogs	1	x 777	x 941	x 860	x 777
	3	x 785	x 841	x 614	x 495
	5	x 707	x 770	x 534	x 376
	10	x 600	x 615	x 330	x 208
	k-means+	16 s	33 s	86 s	175 s
IMDb review	1	x 1165	x 1257	x 2137	x 2253
	3	x 803	x 714	x 1988	x 1787
	5	x 1180	x 1172	x 1715	x 1381
	10	x 815	x 1062	x 1301	x 866
	k-means+	41 s	84 s	214 s	431 s
Reuters RCV1	1	x 511	x 686	x 819	x 892
	3	x 439	x 713	x 850	x 772
	5	x 520	x 685	x 672	x 678
	10	x 518	x 639	x 622	x 425
	k-means+	13 s	28 s	71 s	146 s
MovieLens 20M	1	x 193	x 215	x 218	x 210
	3	x 202	x 213	x 214	x 184
	5	x 202	x 204	x 186	x 145
	10	x 189	x 172	x 144	x 103
	k-means+	4 s	9 s	24 s	49 s
Yelp reviews	1	x 484	x 876	x 1535	x 3092
	3	x 368	x 908	x 1508	x 2917
	5	x 362	x 903	x 1877	x 1595
	10	x 351	x 598	x 1143	x 2120
	k-means+	81 s	164 s	421 s	848 s

modeling, we propose a new unsupervised method for labeling document clusters by discovering keywords within each cluster.

We extract these keywords by comparing their frequencies within different document clusters. Although the terms with high frequencies might initially seem important, not all of them will be discriminative enough to capture the unique characteristics of each document cluster. For example, conjunctions will have high frequencies in all of the document clusters. However, they are not ideal candidates for keywords as they merely serve grammatical purposes. Therefore, we propose a method for finding a keyword by scoring the word's relative frequency within each document cluster (Eq. 4).

$$s(w, C_i) = \frac{p_i(w)}{p_i(w) + p_{-i}(w)} \tag{4}$$

The discriminative score of word w in cluster i, denoted as $s(w, C_i)$, is calculated by comparing $p_i(w)$, its relative frequency within cluster C_i, and $p_{-i}(w)$, its frequencies within other remaining clusters. As a document vector is represented by the frequencies of its comprising words, a feature of the centroid vector corresponding to w can also be used as $p_i(w)$. By comparing the identical word's frequencies within different clusters, our method successfully discovers truly insightful keywords that are frequent only in a few document clusters.

Unlike previous supervised document cluster labeling methods (Onan, Korukoğlu, & Bulut, 2016; Zhang, Xu, Tang, & Li, 2006), our proposed method does not require any model training. Without having to worry about the computational cost of training a model, our proposed method can be easily applied to interpreting large document clusters.

3.4. Overall summary

In Fig. 6, we provide an overall summary of our improved spherical k-means for document clustering. Instead of selecting random initial points, it quickly selects dispersed initial points using extra hyperparameters α and t_{init}. Furthermore, it uses an additional hyperparameter β to enforce sparsity in all of the centroid vectors during clustering. Finally, it extracts keywords for interpreting these document clusters.

4. Experiment results

For each of our enhancements on document clustering, we validate their performances based on numerous criteria listed in Table 2.

We test our proposed improvements on seven different datasets (Table 3) with the number of documents ranging from 63,000 to

Table 5
Average ratio of clustering quality measurements between the proposed initialization method and k-means++.

Data	Intra-cluster distance	Inter-cluster distance	Silhouette score
A6 blogs	0.986	1.001	1.124
Tucson blogs	1.006	1.000	0.998
Sonata blogs	1.005	1.000	1.038
IMDb reviews	0.999	1.000	0.984
Reuters RCV1	0.999	1.001	1.019
MovieLens 20M	1.002	0.996	1.102
Yelp reviews	1.001	0.991	0.980

Table 6
Average percentage of nonzero elements in the centroids from original spherical k-means.

Dataset	k			
	10	20	50	100
A6 blogs	47.5%	37.1%	24.8%	17.8%
Tuscon blogs	45.1%	35.0%	24.6%	17.9%
Sonata blogs	57.2%	48.1%	37.2%	28.1%
IMDb reviews	85.8%	69.0%	55.7%	44.9%
Reuters RCV1	46.5%	37.7%	26.4%	20.6%
MovieLens 20M	8.4%	6.7%	4.8%	3.7%
Yelp reviews	89.6%	87.8%	81.5%	73.8%

Table 8
Average cosine distance between our sparse centroids and original dense centroids ($\times 10^2$).

Dataset	k			
	10	20	50	100
A6 blogs	0.79	1.08	1.64	2.21
Tuscon blogs	0.53	0.69	1.10	1.51
Sonata blogs	0.33	0.48	0.83	1.13
IMDb reviews	0.08	0.16	0.29	0.40
Reuters RCV1	0.05	0.09	0.15	0.24
MovieLens 20M	0.10	0.19	0.40	0.84
Yelp reviews	0.01	0.01	0.03	0.06

5,261,000. A6 blogs, Tucson blogs, and Sonata blogs are blog posts containing the names of each respective types of automobiles that have been crawled from Naver, a Korean online portal site. On the other hand, the Internet Movie Database (IMDb) contains 2,514 movie reviews from a popular movie review website. Furthermore, we also test our improvements on publicly available text datasets such as Reuters Corpus Volume 1 (RCV1), Yelp review data and the MovieLens 20M dataset. Although MovieLens 20M dataset is not composed of documents, its user-item matrix is also high-dimensional and sparse. Therefore, we have included MovieLens 20M to verify the performance of our proposed methods on clustering other general sparse vectors.

4.1. Validating the proposed initialization method

To verify the effectiveness of our proposed initialization method, we test its computation speed and effect on the clustering results.

Table 4 confirms that our proposed initialization method is significantly faster than k-means++. As the computation cost of our method is proportionate to α instead of the number of documents, it outperforms k-means++ in all seven datasets. As the computation time decreases by $n / (\alpha \times k)$, our proposed initialization method achieves relatively greater speed-up in the bigger datasets. For example, the speed-ups achieved in the larger datasets such as IMDb and Yelp reviews are greater than those achieved in other datasets for any given value of k. Therefore, our proposed method is especially useful for clustering a large number of documents.

To measure the impact of our proposed initialization method on the clustering result, we compare the intra-cluster distance, inter-cluster distance and the silhouette score of our proposed method to those of k-means++. For effective clustering, each of the resulting clusters must be compact and clearly different from other clusters. Therefore, lower intra-cluster distance and higher inter-cluster distance usually indicate better clustering outcome. Using this intuition, the silhouette score (Lewis, Ackerman, & de Sa, 2012; Rousseeuw, 1987) uses these two measures to compute the score for measuring the quality of generated clusters (Equation 5).

$$s(x) = \frac{b(x) - a(x)}{max(a(x), b(x))} \tag{5}$$

$a(x)$: average distance between a data point x and other data points within the same cluster

Table 7
Average percentage of nonzero elements in the centroids (Percentage decrease from those of original spherical k-means).

Dataset	β	k			
		10	20	50	100
A6 blogs	0.01	7.1% (85.5%)	4.4% (88.2%)	2.3% (90.8%)	1.5% (92.1%)
	0.02	5.6% (88.7%)	3.1% (91.4%)	1.7% (93.4%)	1% (94.7%)
	0.05	3.4% (92.6%)	2.2% (94.4%)	1% (96.0%)	0.6% (96.9%)
	0.1	2.3% (94.9%)	1.4% (96.2%)	0.6% (97.4%)	0.4% (97.9%)
Tuscon blogs	0.01	7.3% (82.5%)	4.6% (86.4%)	2.7% (89.4%)	1.7% (90.8%)
	0.02	5.7% (86.9%)	3.6% (89.8%)	2% (92.1%)	1.2% (93.5%)
	0.05	4.2% (91.4%)	2.4% (93.3%)	1.2% (95.0%)	0.7% (95.8%)
	0.1	2.9% (93.8%)	1.6% (95.3%)	0.8% (96.6%)	0.5% (97.3%)
Sonata blogs	0.01	12.2% (79.9%)	7.9% (83.8%)	4.3% (88.7%)	2.8% (90.5%)
	0.02	9.2% (83.5%)	5.6% (88.1%)	3% (91.7%)	1.9% (93.4%)
	0.05	6% (89.7%)	3.7% (92.1%)	2% (94.7%)	1.2% (95.8%)
	0.1	4.6% (91.5%)	2.8% (94.5%)	1.3% (96.3%)	0.8% (97.1%)
IMDb reviews	0.01	13.2% (84.7%)	9.2% (87.1%)	6.3% (88.9%)	4.6% (89.9%)
	0.02	11% (87.9%)	6.8% (89.8%)	4.8% (91.5%)	3.6% (92.3%)
	0.05	6.8% (91.7%)	5% (92.9%)	3.3% (94.2%)	2.3% (94.9%)
	0.1	5.5% (93.6%)	3.6% (94.7%)	2.3% (95.8%)	1.6% (96.3%)
Reuters RCV1	0.01	19.3% (58.5%)	14.7% (62.4%)	8.8% (66.6%)	6.3% (69.3%)
	0.02	17.2% (64.9%)	11.4% (68.8%)	7.3% (72.4%)	5.1% (75.4%)
	0.05	12.3% (72.4%)	9.5% (75.4%)	5.5% (79.2%)	3.7% (82.0%)
	0.1	10.7% (76.8%)	7.3% (80.3%)	4.4% (83.5%)	2.8% (86.3%)
MovieLens 20M	0.01	2.3% (71.6%)	2% (71.7%)	1.5% (70.4%)	1.1% (71.2%)
	0.02	2.2% (75.2%)	1.6% (76.1%)	1.2% (76.4%)	0.9% (76.7%)
	0.05	1.7% (79.7%)	1.3% (81.2%)	0.9% (81.9%)	0.6% (83.3%)
	0.1	1.4% (83.7%)	1% (84.1%)	0.7% (86.4%)	0.5% (87.4%)
Yelp reviews	0.01	37.4% (60.4%)	27.2% (69.2%)	19.2% (76.3%)	15.1% (79.6%)
	0.02	28.8% (67.8%)	22.4% (74.3%)	15.8% (80.6%)	12.1% (83.7%)
	0.05	21.3% (75.2%)	17.6% (80.3%)	11.9% (85.4%)	8.8% (88.1%)
	0.1	18.3% (79.4%)	13.8% (84.1%)	9.4% (88.6%)	6.9% (90.6%)

Table 9
Average ratio of clustering quality measurements between the proposed centroid sparsity preservation method and k-means++.

data	Intra-cluster distance	Inter-cluster distance	Silhouette score
A6 blogs	1.000	0.996	0.975
Tucson blogs	1.008	1.002	1.010
Sonata blogs	1.004	1.002	1.042
IMDb reviews	0.999	1.001	1.067
Reuters RCV1	0.999	1.000	1.002
MovieLens 20M	1.002	1.000	1.068
Yelp reviews	0.999	0.994	0.965

Table 10
Summary of clustering results with IMDb dataset.

k	Average pairwise distance between the centroids	Percentage of nonzero elements in the centroids (%)
1,000	0.525	0.46

Table 11
Examples of extracted cluster labels from IMDb reviews.

Clusters	Cluster labels
"Titanic"	iceberg, zane, sinking, titanic, rose, winslet, camerons, 1997, leonardo, leo, ship, cameron, dicaprio, kate, tragedy, jack, disaster, james, romance, love, effects, story
Heros of Marvel comics	zemo, chadwick, boseman, bucky, panther, holland, cap, infinity, mcu, russo, civil, bvs, antman, winter, ultron, airport, avengers, marvel, captain, superheroes, stark, evans, america, iron, spiderman
Alien sci-fi	skyline, jarrod, balfour, strause, invasion, independence, cloverfield, angeles, district, los, worlds, aliens, alien, la, budget, scifi, battle, cgi, day, effects, war
Horror	gayheart, loretta, candyman, legends, urban, witt, campus, tara, reid, legend, alicia, englund, leto, scream, murders, slasher, helen, killer, student, teen, summer, cut, horror, final, sequel, scary
"Matrix"	neo, morpheus, neos, oracle, trinity, zion, architect, hacker, reloaded, revolutions, wachowski, fishburne, machines, agents, matrix, keanu, smith, reeves, agent, jesus, machine, computer, fighting, fight, real

$b(x)$: average distance between a data point x and other data points in the closest cluster

In Table 5, we have divided the intra-cluster distance, the inter-cluster distance and the silhouette score of our proposed method from those of k-means++, respectively. For a given dataset, we have initially averaged the values of these three criteria on the same k, and subsequently calculated the overall average of each criterion. As the values of these three criteria are close to 1, it indicates that our proposed initialization method successfully preserves the quality of the clustering outcome. Therefore, these two experiments confirm that our initialization method provides a significant computational advantage without any loss in clustering quality.

4.2. Validating the proposed centroid sparsity preservation method

To verify the effectiveness of our proposed centroid sparsity preservation method, we compare the centroid vectors generated from our method with dense centroids created from original spherical k-means.

Tables 6 and 7 show the percentage of nonzero elements in the centroid vectors from the original spherical k-means and those from our proposed method, respectively. From all seven datasets, our proposed method successfully creates sparse centroid vectors with 99.66% ~ 99.89% of their elements as zeros. For example, original spherical k-means clustering with k=10 in A6 blogs creates centroid vectors with 47.51% of their elements as nonzero (Table 6). However, the percentage of nonzero elements significantly drop to 7.07% for even a small β of 0.01.

Additionally, Table 8 computes the average pairwise cosine distance between the dense centroids from the original spherical k-means and the sparse centroids from our proposed method. Surprisingly, the pairwise distance between two different representations of centroids is considerably small regardless of k, suggesting that the centroid vectors themselves have not significantly changed. By enforcing sparsity, our proposed method, however, successfully removes the impact of irrelevant words from the centroid vectors, thereby enhancing their interpretability without changing the outcome of document clustering.

To evaluate the impact of our proposed centroid sparsity preservation method on the clustering quality, we have once again computed the intra-cluster distance, the inter-cluster distance and the silhouette score of the clusters generated from the sparse centroid vectors (Table 9). For all seven datasets, our method successfully creates sparse centroid vectors without any loss in the clustering quality.

4.3. Validating the proposed cluster labeling method

In this subsection, we qualitatively verify the effectiveness of our proposed document cluster labeling method with IMDb reviews and Sonata blogs. As the cluster labels of other datasets show similar trends, we have attached their results in Appendix C. However, the results of Reuters RCV1 and MovieLens 20M are not included as their input data are provided as pre-computed vectors.

4.3.1. IMDb reviews

To generate meaningful document clusters for IMDb reviews, we have increased the values of k to 1,000, and set β as 0.1. With an appropriately large enough k, the pairwise distances between the centroids are now significantly greater (Table 10).

Table 11 lists five examples of cluster labels extracted by our proposed document cluster labeling method as described by Eq. 4. With these discriminative keywords, we can easily conclude that the first and second document clusters are related to the "Titanic"

Table 12
Summary of the clustering results of Sonata blogs.

k	Average pairwise distance of the centroids	Percentage of nonzero elements in the centroids (%)
500	0.963	0.34

Table 13
Examples of extracted cluster labels from Sonata blogs.

Clusters	Cluster labels
Renting Sonata during trips to Jeju Island	Car rental in Jeju Island, Busan departure Jeju, Jeju Ole Road, Roundtrip flight, Lotte Hotel, free travel, room, Trip to Jeju, Jeju, gas, airline ticket, breakfast, Jeju Airport
Used car sales	YF Sonata, Premium Sonata 2011, Complete option, YF Sonata PR, No-accident, Sold out, Gunpo City, Black, Hi-pass, No Warranty, Rating, Panorama, fake offerings
Classical music	brass instruments, horn, trumpet, brass, Telemann, Eb, oboe, concerto, Haydn, instrument, performing, orchestra, solo, movement, composer
"Sonata of Temptation" by Korean singer Ivy	Song, Listening while studying, lyrics, singing, singer, vocal, voice, ballad, masterpiece, Ivy, temptation, title, listening, albums
"Flaming Sonata" a Korean novel	Gwangsu Lee (Korean author), naturalism, Japanophilism, Pyongyang, wild, beauty, Dongin Kim (Korean author), light, resembles, realism, madness, 1920, "Potato" (novel by Dongin Kim), Korean literature

and the Marvel movies, respectively. Similarly, we can clearly understand that the third, fourth and fifth document clusters contain movie reviews about alien-related science fiction movies, horror movies and the "Matrix", respectively. As shown by these examples, our proposed document cluster labeling method extracts intuitive keywords, providing a simple yet data-driven approach for interpreting the clusters.

4.3.2. Sonata blogs

As the word Sonata can refer to a type of a Korean automobile, a form of classical music or a name of a Korean pop song, Sonata blogs can be clustered into several unique topics. Therefore, we have applied our cluster labeling method to 230,000 blog posts to analyze these different meanings of the term Sonata. A basic summary of the clustering result is shown in Table 12, while some of its labeled clusters are listed in Table 13.

By observing the extracted keywords in Table 13, we notice that the first and second cluster grouped the blog posts that discuss renting a Sonata or selling a used Sonata, respectively. However, the keywords extracted from other document clusters successfully reflect the multiple definitions of the term Sonata. The keywords of the third, fourth and fifth document clusters suggest that they contain blog posts related to classical music, Korean pop singer Ivy and the Korean literature A Flame Sonata, respectively. Therefore, our proposed method is also effective in distinguishing and interpreting documents that contain terms with multiple different meanings. (Tables 14, 15, 16, 17, 18, 19, 20, 21 and 22)

5. Conclusions

Due to its simplicity and intuitive interpretability, spherical k-means is often used for clustering a large number of sparse document vectors. In this paper, we propose several modifications to improve the performance of spherical k-means.

To ensure well-dispersed initial points, we suggest a new initialization method that is more computationally efficient than the initialization method of k-means++. By choosing a new initial point such that its distance to other initial points are greater than a certain threshold, our method ensures that all initial points are dissimilar. Furthermore, its fast computation time and convergence speed render it ideal for clustering a large number of document vectors.

Furthermore, we propose a method of enforcing sparsity in the centroid vectors. Since the centroid vectors of the conventional spherical k-means are calculated as the average of their comprising vectors, the frequencies of rare and meaningless words are inevitably captured by them, thereby losing their intuitive interpretability and distorting the distance measure between the data points. Unlike L1 ball projection that requires a manually predefined parameter, our proposed method relies on the number of documents assigned to each cluster, providing a dynamic and data-driven solution for enforcing sparsity.

Additionally, we provide a novel cluster labeling method for interpreting document clusters. Instead of training an additional classifier for labeling, our method effectively extracts keywords from each cluster based on a simple scoring mechanism that utilizes both the coverage and the discriminative power of a word.

However, this paper does not address other fundamental issues of k-means clustering such as finding an optimal value of k or overcoming the uniform effect. Throughout this paper, we have simply assumed that the adequate values of k have always been given, thereby generating topically independent document clusters. However, finding an appropriate value of k for a given dataset is crucial in extracting meaningful insights from its topically independent clusters. If k is set to be too large, topically similar documents will be needlessly divided into several different clusters. On the other hand, topically dissimilar documents will be clustered into a single cluster if k is set to be too small. Despite the importance of finding an appropriate value of k, previously suggested measures for approximating optimal k such as silhouette score are ineffective for high dimensional sparse document vectors (Almeida, Guedes, Meira, & Zaki, 2011). In the future, measures for approximating optimal k in a high dimensional sparse vector space will further improve our methods proposed in this paper.

Despite these limitation, our proposed methods provide practical improvements on spherical k-means for document clustering. As our methods provide significant computational speed up during initialization, and require less memory for storing sparse centroids, it will especially be effective in clustering a large number of documents. With ever-increasing text data being generated in various industries, the importance of quick, simple and intuitive document clustering cannot be overemphasized.

Declaration of Competing Interest

None.

Credit authorship contribution statement

Hyunjoong Kim: Conceptualization, Methodology, Software, Validation, Investigation, Resources, Data curation, Writing - original draft, Visualization, Project administration, Funding acquisition. **Han Kyul Kim:** Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Resources, Data curation, Writing - original draft, Writing - review & editing, Visualization. **Sungzoon Cho:** Formal analysis, Investigation, Supervision, Funding acquisition.

Acknowledgments

This work was supported by the BK21 Plus Program (Center for Sustainable and Innovative Industrial Systems, Department of Industrial Engineering and Institute for Industrial Systems Innovation, Seoul National University) funded by the Ministry of Education, Korea (no. 21A20130012638), the National Research Foundation (NRF) grant funded by the Korea government (MSIP) (no. 2011-0030814), and the Institute for Industrial Systems Innovation of SNU.

Appendix A. Intra-cluster distance, inter-cluster distance and silhouette score of the clusters generated from k-means++ and our proposed initialization method

Table 14
Intra-cluster distance of k-means++ and our proposed initialization method with varying α and k.

Data	α	k			
		10	20	50	100
A6 blogs	1	0.677	0.645	0.591	0.546
	3	0.680	0.640	0.588	0.544
	5	0.671	0.647	0.589	0.554
	10	0.672	0.639	0.594	0.544
	k-means+	0.676	0.645	0.589	0.547
Tuscon blogs	1	0.617	0.578	0.514	0.473
	3	0.626	0.573	0.518	0.474
	5	0.617	0.574	0.517	0.475
	10	0.617	0.581	0.515	0.478
	k-means+	0.612	0.571	0.511	0.479
Sonata blogs	1	0.680	0.635	0.587	0.545
	3	0.683	0.639	0.588	0.549
	5	0.676	0.651	0.587	0.548
	10	0.682	0.639	0.584	0.551
	k-means+	0.682	0.633	0.582	0.548
IMDb reviews	1	0.240	0.239	0.235	0.236
	3	0.243	0.237	0.236	0.235
	5	0.243	0.238	0.236	0.236
	10	0.243	0.236	0.236	0.236
	k-means+	0.243	0.241	0.235	0.234
Reuters RCV1	1	0.788	0.760	0.720	0.684
	3	0.788	0.761	0.717	0.683
	5	0.793	0.762	0.719	0.684
	10	0.788	0.758	0.719	0.683
	k-means+	0.786	0.763	0.724	0.682
MovieLens 20M	1	0.576	0.555	0.529	0.512
	3	0.579	0.559	0.529	0.511
	5	0.576	0.557	0.532	0.512
	10	0.576	0.555	0.529	0.511
	k-means+	0.575	0.554	0.531	0.511
Yelp reviews	1	0.502	0.491	0.480	0.475
	3	0.499	0.487	0.478	0.469
	5	0.500	0.486	0.481	0.473
	10	0.500	0.489	0.479	0.473
	k-means+	0.495	0.490	0.479	0.474

Table 15
Inter-cluster distance of k-means++ and our proposed initialization method with varying α and k.

Data	α	k			
		10	20	50	100
A6 blogs	1	0.882	0.895	0.899	0.907
	3	0.888	0.894	0.898	0.900
	5	0.890	0.897	0.902	0.908
	10	0.895	0.891	0.905	0.905
	k-means+	0.901	0.895	0.898	0.909
Tuscon blogs	1	0.869	0.870	0.881	0.880
	3	0.866	0.874	0.874	0.880
	5	0.867	0.883	0.876	0.879
	10	0.866	0.869	0.880	0.880
	k-means+	0.862	0.877	0.878	0.880
Sonata blogs	1	0.902	0.909	0.921	0.921
	3	0.910	0.907	0.917	0.919
	5	0.909	0.917	0.920	0.920
	10	0.905	0.913	0.920	0.921
	k-means+	0.906	0.914	0.919	0.920
IMDb reviews	1	0.291	0.286	0.288	0.285
	3	0.286	0.288	0.285	0.285
	5	0.287	0.288	0.286	0.285
	10	0.285	0.292	0.286	0.285
	k-means+	0.289	0.284	0.287	0.287
Reuters RCV1	1	0.924	0.926	0.929	0.937
	3	0.931	0.926	0.931	0.937
	5	0.916	0.925	0.931	0.937
	10	0.923	0.927	0.931	0.936
	k-means+	0.923	0.924	0.928	0.937
MovieLens 20M	1	0.786	0.793	0.798	0.804
	3	0.779	0.792	0.801	0.808
	5	0.789	0.793	0.800	0.807
	10	0.790	0.788	0.799	0.806
	k-means+	0.791	0.795	0.803	0.807
Yelp reviews	1	0.647	0.628	0.621	0.615
	3	0.612	0.630	0.623	0.619
	5	0.648	0.631	0.616	0.619
	10	0.661	0.646	0.624	0.615
	k-means+	0.654	0.646	0.623	0.614

Table 16
Silhouette score of k-means++ and our proposed initialization method with varying α and k.

Data	α	k			
		10	20	50	100
A6 blogs	1	0.142	0.153	0.149	0.184
	3	0.140	0.152	0.161	0.187
	5	0.143	0.151	0.165	0.171
	10	0.157	0.148	0.164	0.189
	k-means+	0.146	0.148	0.171	0.190
Tuscon blogs	1	0.169	0.148	0.205	0.218
	3	0.160	0.185	0.179	0.223
	5	0.173	0.191	0.186	0.210
	10	0.157	0.146	0.195	0.210
	k-means+	0.168	0.172	0.187	0.212
Sonata blogs	1	0.154	0.173	0.193	0.183
	3	0.162	0.169	0.176	0.183
	5	0.172	0.172	0.174	0.172
	10	0.150	0.166	0.183	0.181
	k-means+	0.147	0.171	0.170	0.179
IMDb reviews	1	0.061	0.042	0.021	0.009
	3	0.052	0.043	0.026	0.009
	5	0.047	0.036	0.023	0.008
	10	0.050	0.038	0.022	0.012
	k-means+	0.049	0.038	0.023	0.012
Reuters RCV1	1	0.088	0.100	0.127	0.145
	3	0.095	0.102	0.130	0.144
	5	0.078	0.102	0.125	0.151
	10	0.085	0.105	0.128	0.151
	k-means+	0.090	0.095	0.120	0.152
MovieLens 20M	1	0.122	0.112	0.090	0.079
	3	0.158	0.148	0.084	0.078
	5	0.163	0.107	0.087	0.081
	10	0.120	0.117	0.107	0.076
	k-means+	0.123	0.102	0.089	0.075
Yelp reviews	1	0.073	0.061	0.041	0.029
	3	0.069	0.060	0.045	0.023
	5	0.069	0.062	0.040	0.026
	10	0.073	0.062	0.039	0.025
	k-means+	0.074	0.053	0.044	0.029

Appendix B. Intra-cluster distance, inter-cluster distance and silhouette score of the clusters generated from k-means++ and our proposed centroid sparsity preservation method

Table 17
Intra-cluster distance of k-means++ and our centroid sparsity preservation method with varying β and k.

Data	β	k			
		10	20	50	100
A6 blogs	0.01	0.670	0.638	0.601	0.550
	0.02	0.671	0.647	0.589	0.554
	0.05	0.669	0.642	0.602	0.553
	0.1	0.675	0.641	0.590	0.549
	k-means	0.677	0.645	0.589	0.548
Tuscon blogs	0.01	0.620	0.577	0.513	0.477
	0.02	0.622	0.575	0.515	0.478
	0.05	0.624	0.579	0.518	0.483
	0.1	0.618	0.572	0.526	0.474
	k-means	0.612	0.571	0.512	0.479
Sonata blogs	0.01	0.686	0.642	0.591	0.548
	0.02	0.680	0.643	0.579	0.547
	0.05	0.679	0.639	0.589	0.551
	0.1	0.679	0.643	0.585	0.545
	k-means	0.683	0.633	0.582	0.549
IMDb reviews	0.01	0.241	0.239	0.236	0.236
	0.02	0.239	0.241	0.236	0.236
	0.05	0.244	0.240	0.236	0.236
	0.1	0.242	0.240	0.237	0.236
	k-means	0.243	0.241	0.235	0.235
Reuters RCV1	0.01	0.786	0.764	0.718	0.682
	0.02	0.792	0.762	0.721	0.681
	0.05	0.785	0.766	0.723	0.682
	0.1	0.789	0.762	0.719	0.684
	k-means	0.786	0.764	0.724	0.682
MovieLens 20M	0.01	0.578	0.556	0.530	0.512
	0.02	0.584	0.556	0.529	0.512
	0.05	0.577	0.556	0.529	0.512
	0.1	0.580	0.556	0.529	0.513
	k-means	0.576	0.555	0.531	0.512
Yelp reviews	0.01	0.502	0.490	0.477	0.472
	0.02	0.498	0.488	0.479	0.473
	0.05	0.500	0.490	0.477	0.473
	0.1	0.497	0.486	0.479	0.472
	k-means	0.496	0.490	0.480	0.474

Table 18
Inter-cluster distance of k-means++ and our centroid sparsity preservation method with varying β and k.

Data	β	k			
		10	20	50	100
A6 blogs	0.01	0.896	0.894	0.905	0.900
	0.02	0.878	0.902	0.897	0.905
	0.05	0.885	0.880	0.907	0.909
	0.1	0.889	0.903	0.903	0.912
	k-means	0.901	0.895	0.898	0.910
Tuscon blogs	0.01	0.869	0.874	0.879	0.881
	0.02	0.873	0.882	0.874	0.883
	0.05	0.859	0.883	0.881	0.876
	0.1	0.867	0.876	0.886	0.884
	k-means	0.862	0.878	0.878	0.880
Sonata blogs	0.01	0.909	0.913	0.917	0.923
	0.02	0.906	0.920	0.924	0.924
	0.05	0.908	0.918	0.919	0.922
	0.1	0.910	0.908	0.923	0.923
	k-means	0.906	0.915	0.920	0.920
IMDb reviews	0.01	0.291	0.289	0.287	0.287
	0.02	0.293	0.286	0.286	0.286
	0.05	0.287	0.286	0.287	0.287
	0.1	0.290	0.288	0.287	0.286
	k-means	0.289	0.285	0.288	0.287
Reuters RCV1	0.01	0.923	0.922	0.932	0.938
	0.02	0.914	0.927	0.931	0.938
	0.05	0.926	0.922	0.932	0.937
	0.1	0.921	0.927	0.932	0.939
	k-means	0.923	0.924	0.929	0.937
MovieLens 20M	0.01	0.802	0.788	0.799	0.807
	0.02	0.786	0.794	0.803	0.811
	0.05	0.788	0.797	0.800	0.813
	0.1	0.794	0.795	0.807	0.815
	k-means	0.792	0.795	0.803	0.807
Yelp reviews	0.01	0.613	0.629	0.625	0.616
	0.02	0.651	0.631	0.624	0.618
	0.05	0.666	0.626	0.625	0.617
	0.1	0.656	0.651	0.623	0.621
	k-means	0.655	0.646	0.623	0.615

Table 19
Silhouette score of k-means++ and our centroid sparsity preservation method with varying β and k.

Data	β	k			
		10	20	50	100
A6 blogs	0.01	0.147	0.145	0.162	0.178
	0.02	0.136	0.135	0.165	0.181
	0.05	0.148	0.128	0.159	0.181
	0.1	0.150	0.161	0.179	0.196
	k-means	0.146	0.148	0.171	0.190
Tuscon blogs	0.01	0.167	0.159	0.189	0.215
	0.02	0.169	0.188	0.190	0.211
	0.05	0.144	0.171	0.209	0.211
	0.1	0.173	0.182	0.192	0.218
	k-means	0.168	0.172	0.187	0.212
Sonata blogs	0.01	0.146	0.161	0.172	0.183
	0.02	0.162	0.183	0.190	0.190
	0.05	0.167	0.173	0.181	0.188
	0.1	0.164	0.148	0.184	0.184
	k-means	0.147	0.171	0.170	0.179
IMDb reviews	0.01	0.053	0.043	0.026	0.008
	0.02	0.062	0.038	0.024	0.006
	0.05	0.051	0.040	0.026	0.011
	0.1	0.067	0.047	0.027	0.014
	k-means	0.049	0.038	0.023	0.012
Reuters RCV1	0.01	0.090	0.100	0.129	0.149
	0.02	0.076	0.102	0.132	0.153
	0.05	0.089	0.093	0.125	0.146
	0.1	0.079	0.098	0.125	0.145
	k-means	0.090	0.095	0.120	0.152
MovieLens 20M	0.01	0.104	0.119	0.105	0.085
	0.02	0.156	0.122	0.090	0.078
	0.05	0.119	0.103	0.085	0.081
	0.1	0.141	0.110	0.087	0.077
	k-means	0.123	0.102	0.089	0.075
Yelp reviews	0.01	0.073	0.065	0.040	0.024
	0.02	0.075	0.059	0.038	0.025
	0.05	0.076	0.052	0.042	0.026
	0.1	0.078	0.056	0.039	0.024
	k-means	0.074	0.053	0.044	0.029

Appendix C. Examples of extracted cluster labels from A6 blogs, Tuscon blogs, and Yelp reviews

Table 20
Examples of extracted cluster labels from from A6 blogs.

Clusters	Cluster labels
Gwanggyo New Town	riverside, offer, riverside city, new town, distributor, Han River, winner, model house, civilian, urban, BM-motors, Buddha, 6th, location, public, parcel out
Chipset of Samsung Galaxy	Optimus, Vega, TSMC, Samsung, ARM, Qualcomm, Tegra, Galaxy Note, Galaxy Note 2, Galaxy, Galaxy S, Galaxy S3, Nexus, cellphone, Samsung, Samsung Electronics
iPad chipset	Retina, face, Siri, mini, iPad, iPhone, iPhone5, Apple, MHz, call, 4S, iOS, phone, iOS-6, 16GB, 3G, 5S, network, 32G, app, fixel, connector, finterprint, LTE
Audi A6	Delco, vehicle, used A6, imported car, trip, Lincoln Continental, belkin, free visit, belkin air, car battery, audio, generator, Audi A4, Audi A6, Audi A7, Audi A8, development
Car repair	oil filter, ignition, air cleaner, drain, upper, suction, castrol, upper arm, exchange, oil, exchange operation, exchange, injection, anti-freeze, pad

Table 21
Examples of extracted cluster labels from Tuscon blogs.

Clusters	Cluster labels
Rental car and locations	repairing wheel, genuine, QM3, Kia-motors, Hyundai-motors, Kimje (city of Korea), Yongjong-island, The-new Avante (car model name), glossy black, Iksan (city of Korea), Gwacheon (city of Korea), carrier, water-repellent coating
Long term rental service	long term, rental car, car lease, rental fee, auto-lease, quotation, KT Kumho rental car, return, home shopping, new car long term rent, initial cost, tax, lease, rental, Corporate, non-guarantee, acquisition, navigation, promotion, insurance, quote
Used car sales in Ulsan city	Ulsan (city of Korea), fast, car key, Jinjang-dong, Hyundai, heat block, 3M sun tanning, wheel, Audi, imported car
Tire sales	365 days, cheapest, wear, 60R1, change tire, Kumho-tire, Majesty, Dynapro, Ventus, air pressure, Tire-tech, tire price, snow-chain, Michelin, 55R, Nexen Tire, Wheel Alignment, Hankook Tire
Renting Tuscon during trips to Jeju Island	Mom, eat, coffee, delicious, Jeju Island, dad, home, my, friend, morning, travel, real, dawn, arrival, picure, us, departure, people, jeju, car wash, here, wait, ride, so, Tucson

Table 22
Examples of extracted cluster labels from Yelp reviews.

Clusters	Cluster labels
Vegetarians	vegetarians, carnivores, alike, vegans, eaters, vegetarian, options, veggie, tofu, non, vegan, choices, meat, option, variety, lots, dishes, menu, buffet, especially, delicious, selection, fresh, tasty, for, many, food, everyone, great
Seafoods	crabs, crab, cakes, shell, snow, stone, king, cake, seafood, lobster, soft, shrimp, appetizer, fried, meat, sauce, steak, ordered, salad, dinner, taste, meal, delicious, fresh
Korean foods	banchan, bibimbap, kimchi, korean, kalbi, bulgogi, bone, tofu, dishes, bbq, spicy, pork, soup, rice, seafood, noodles, dish, side, restaurants, beef, meat, fried, restaurant, hot, sauce, ordered, other, menu
Mexican foods	lard, refried, tortillas, beans, fat, bean, burritos, vegetarian, burrito, mexican, vegan, oil, greasy, salsa, chips, tacos, taste, fries, fried, meat, eat, rice, free, cheese, fresh,
Restaurants for anniversary	anniversary, celebrated, celebrating, celebrate, occasion, romantic, wedding, complimentary, reservation, husband, dessert, year, our, evening, wife, view, special, dinner, waiter, wonderful, steak, server, wine

References

Abualigah, L. M., Khader, A. T., Al-Betar, M. A., & Alomari, O. A. (2017). Text feature selection with a robust weight scheme and dynamic dimension reduction to text document clustering. *Expert Systems with Applications, 84*, 24–36.

Almeida, H., Guedes, D., Meira, W., & Zaki, M. J. (2011). Is there a best quality metric for graph clusters? In Joint European conference on machine learning and knowledge discovery in databases (pp. 44–59). Springer.

Arthur, D., & Vassilvitskii, S. (2007). k-means++: The advantages of careful seeding. In Proceedings of the eighteenth annual ACM-SIAM symposium on discrete algorithms (pp. 1027–1035). Society for Industrial and Applied Mathematics.

Bachem, O., Lucic, M., Hassani, H., & Krause, A. (2016). Fast and provably good seedings for k-means. In *Advances in neural information processing systems* (pp. 55–63).

Bagirov, A. M. (2008). Modified global k-means algorithm for minimum sum-of-squares clustering problems. *Pattern Recognition, 41*(10), 3192–3199.

Bahmani, B., Moseley, B., Vattani, A., Kumar, R., & Vassilvitskii, S. (2012). Scalable k-means++. *Proceedings of the VLDB Endowment, 5*(7), 622–633.

Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent dirichlet allocation. *Journal of machine Learning research, 3*(Jan), 993–1022.

Buchta, C., Kober, M., Feinerer, I., & Hornik, K. (2012). Spherical k-means clustering. *Journal of Statistical Software, 50*(10), 1–22.

Capó, M., Pérez, A., & Lozano, J. A. (2017). An efficient approximation to the k-means clustering for massive data. *Knowledge-Based Systems, 117*, 56–69.

Chuang, J., Manning, C. D., & Heer, J. (2012a). Termite: Visualization techniques for assessing textual topic models. In *Proceedings of the international working conference on advanced visual interfaces* (pp. 74–77). ACM.

Chuang, J., Ramage, D., Manning, C., & Heer, J. (2012b). Interpretation and trust: Designing model-driven visualizations for text analysis. In *Proceedings of the sigchi conference on human factors in computing systems* (pp. 443–452). ACM.

Clauset, A., Newman, M. E., & oore, C. (2004). Finding community structure in very large networks. Physical Review E, *70*(6), 66111.

Coates, A., Ng, A., & Lee, H. (2011). An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics* (pp. 215–223).

Coates, A., & Ng, A. Y. (2012). Learning feature representations with k-means. In *Neural networks: Tricks of the trade* (pp. 561–580). Springer.

Dhillon, I. S., Guan, Y., & Kogan, J. (2002). Iterative clustering of high dimensional text data augmented by local search. In Data mining, 2002. ICDM 2003. proceedings. 2002 ieee international conference on (pp. 131–138). IEEE.

Dhillon, I. S., & Modha, H. S. (2001). Concept decompositions for large sparse text data using clustering. Machine Learning, *42*(1), 143–175.

Duchi, J., Shalev-Shwartz, S., Singer, Y., & Chandra, T. (2008). Efficient projections onto the l 1-ball for learning in high dimensions. In *Proceedings of the 25th international conference on machine learning* (pp. 272–279). ACM.

Ester, M., Kriegel, J., Hans-Peter nd Sander, Xu, X., & thers (1996). A density-based algorithm for discovering clusters in large spatial databases with noise.. In *Proceeding of the 2nd international conference of knowledge discovery and data mining* (pp. 226–231).

He, K., Wen, F., & Sun, J. (2013). K-means hashing: An affinity-preserving quantization method for learning binary compact codes. In *Proceedings of the ieee conference on computer vision and pattern recognition* (pp. 2938–2945).

Huang, A. (2008). Similarity measures for text document clustering. In Proceedings of the sixth new zealand computer science research student conference (NZCSRSC 2008), christchurch, new zealand (pp. 49–56).

Jin, Z., Li, C., Lin, Y., & Cai, D. (2013). Density sensitive hashing. IEEE Transactions on Cybernetics, *44*(8), 1362–1371.

Jurafsky, D. (2000). *Speech & language processing*. Pearson Education India.

Kim, H. K., Kim, H., & Cho, S. (2017). Bag-of-concepts: Comprehending document representation through clustering words in distributed representation. *Neurocomputing, 266*, 336–352.

Lewis, J., Ackerman, M., & de Sa, V. (2012). Human cluster evaluation and formal quality measures: A comparative study. In *Proceedings of the annual meeting of the cognitive science society: 34*.

Li, Y., Zhao, K., Chu, X., & Liu, J. (2013). Speeding up k-means algorithm by gpus. *Journal of Computer and System Sciences, 79*(2), 216–229.

Likas, A., Vlassis, N., & Verbeek, J. J. (2003). The global k-means clustering algorithm. Pattern Recognition, *36*(2), 451–461.

Lloyd, S. (1982). Least squares quantization in pcm. IEEE Transactions on Information Theory, *28*(2), 129–137.

Newman, D., Noh, Y., Talley, E., Karimi, S., & Baldwin, T. (2010). Evaluating topic models for digital libraries. In *Proceedings of the 10th annual joint conference on digital libraries* (pp. 215–224). ACM.

Onan, A., Korukoğlu, S., & Bulut, H. (2016). Ensemble of keyword extraction methods and classifiers in text classification. *Expert Systems with Applications, 57*, 232–247.

Rousseeuw, P. J. (1987). Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. Journal of Computational and Applied Mathematics, *20*, 53–65.

Sculley, D. (2010). Web-scale k-means clustering. In *Proceedings of the 19th international conference on world wide web* (pp. 1177–1178). ACM.

Shahrivari, S., & Jalili, S. (2016). Single-pass and linear-time k-means clustering based on mapreduce. *Information Systems, 60*, 1–12.

Shalev-Shwartz, S., Singer, Y., Srebro, N., & Cotter, A. (2011). Pegasos: Primal estimated sub-gradient solver for svm. Mathematical Programming, *127*(1), 3–30.

Shen, X., Liu, W., Tsang, I., Shen, F., & Sun, Q.-S. (2017). Compressed k-means for large-scale clustering. *Thirty-first aaai conference on artificial intelligence*.

Sibson, R. (1973). Slink: an optimally efficient algorithm for the single-link cluster method. The Computer Journal, *16*(1), 30–34.

Sievert, C., & Shirley, K. E. (2014). Ldavis: A method for visualizing and interpreting topics. In *Proceedings of the workshop on interactive language learning, visualization, and interfaces* (pp. 63–70).

Snyder, J., Knowles, R., Dredze, M., Gormley, M., & Wolfe, T. (2013). Topic models and metadata for visualizing text corpora. *Proceedings of the 2013 NAACL HLT Demonstration Session*, 5–9.

Sparck Jones, K. (1972). A statistical interpretation of term specificity and its application in retrieval. Journal of Documentation, *28*(1), 11–21.

Xie, J., Girshick, R., & Farhadi, A. (2016). Unsupervised deep embedding for clustering analysis. In *International conference on machine learning* (pp. 478–487).

Xu, D., & Tian, Y. (2015). A comprehensive survey of clustering algorithms. *Annals of Data Science, 2*(2), 165–193.

Yang, B., Fu, X., Sidiropoulos, N. D., & Hong, M. (2017). Towards k-means-friendly spaces: Simultaneous deep learning and clustering. In *Proceedings of the 34th international conference on machine learning-volume 70* (pp. 3861–3870). JMLR. org.

Zhang, K., Xu, H., Tang, J., & Li, J. (2006). Keyword extraction using support vector machine. In *International conference on web-age information management* (pp. 85–96). Springer.