Simulation Modelling Practice and Theory 54 (2015) 49–63

Contents lists available at ScienceDirect

Simulation Modelling Practice and Theory

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

A novel approach for initializing the spherical *K*-means clustering algorithm

Rehab Duwairi [a,*], Mohammed Abu-Rahmeh [b]

[a] *Department of Computer Information Systems, Jordan University of Science and Technology, Irbid, Jordan*
[b] *Department of Computer Science, Jordan University of Science and Technology, Irbid, Jordan*

ARTICLE INFO

Article history:
Received 4 December 2014
Received in revised form 19 March 2015
Accepted 20 March 2015

Keywords:
Spherical *K*-means clustering
K-means initialization
Intra-cluster similarity
Cluster compactness

ABSTRACT

In this paper, a novel approach for initializing the spherical *K*-means algorithm is proposed. It is based on calculating well distributed seeds across the input space. Also, a new measure for calculating vectors' directional variance is formulated, to be used as a measure of clusters' compactness. The proposed initialization scheme is compared with the classical *K*-means – where initial seeds are specified randomly or arbitrarily – on two datasets. The assessment was based on three measures: an objective function that measures intra cluster similarity, cluster compactness and time to converge. The proposed algorithm (called initialized *K*-means) outperforms the classical (random) *K*-means when intra cluster similarity and cluster compactness were considered for several values of *k* (number of clusters). As far as convergence time is concerned, the initialized *K*-means converges faster than the random *K*-means for small number of clusters. For a large number of clusters the time necessary to calculate the initial clusters' seeds start to outweigh the convergence criterion in time. The exact number of clusters at which the proposed algorithm starts to change behavior is data dependent (=11 for dataset1 and = 15 for dataset2).

1. Introduction

The *k*-means algorithm is an iterative partitioning algorithm that starts with a set of *n* points in R^w, and ends up with a grouping of these points in clusters, by determining a set of *K* points, which are considered as centers of the resulting clusters. The problem is to find the optimal set of such points that minimizes the mean of squared distances from each data instance to its nearest center [30], this objective function is variance based and known as the squared error distortion [27].

Other minimization objective functions include the sum of distances, as in the Euclidean *K*-medians problem [4,33], or to minimize the maximum distance between any point and its nearest center, which is adopted in the Geometric *K*-center problem [1]. Other objective functions describe the goodness of clusters, and these functions are subject to maximization; as in the case of document clustering [17,18], where the function measures the sum of intra cosine similarities within clusters.

The simplicity of the algorithm made it a very attractive choice for clustering, and its known for its efficiency, the algorithm described above is known as Lloyd's algorithm [21]. Obviously; it's similar to the fitting routines; starting with an initial state, then optimizing the parameters subsequently. It has been used in many applications such text categorizations [3], consensus clustering [42,45] and the segmentation of images [5].

[*] Corresponding author. Fax: +962 2 7201077.
E-mail address: rehab@just.edu.jo (R. Duwairi).

http://dx.doi.org/10.1016/j.simpat.2015.03.007
1569-190X/© 2015 Elsevier B.V. All rights reserved.

Even though the algorithm is simple and efficient, it is not exempt to drawbacks, such as the selection of bad initial centers (slower convergence) [48], which motivated our modifications to the algorithm, and the hill-climbing problem that results in local optimum solutions [7], where local optimality corresponds to the centroidal voronoi problem [19,21,30]. Other problems include the determination of the optimal number of clusters and sensitivity to outliers [12,26,27].

Generally, the application of the *K*-means algorithm requires determining a proper proximity measure, an assessment metric of the clusters' quality, the number of clusters (k), the values of initial means, and a convergence condition. Usually the algorithm terminates when the centroids become stable, but the proximity measure and the objective function are dependent on the type of data being clustered. Initializing the algorithm proved to be a sufficient approach for overcoming the problem of getting stuck at bad local optimum solutions [22,24,25,31].

Clustering algorithms have targeted numerous data types such as textual data [3,20,28,37], images [15] and communication data [44]. Text clustering requires preprocessing the documents, so they are represented via constructing a vector space; a well known procedure to facilitate information retrieving process. A set of terms among all the terms that occur in the corpora are selected to comprise the axis, and the documents are vectors in the space, each of which has term weights as its components [6,11], distinct weights indicate the significance of the corresponding term to the document. Beside the huge size of document collections present, the performance requirements are confronted by the large number of dimensions that occur in such contexts, which is described as the curse of dimensionality, thus simple techniques are desirable.

As text documents are converted into numeric data vectors, clustering can take place by applying the *K*-means, but a number of considerations have to be made, for instance, choosing the Euclidean distance as a proximity measure is not appropriate to cluster document vectors [40], since the location of a pair of points cannot decide their relevance subjectively, rather it is the angle between any two documents that defines their similarity [6,11,39], therefore, it is the notion of direction that must be the foremost rule that guides the process.

A variant of the *K*-means that uses the cosine similarity is known as the spherical *K*-means, this algorithm can be applied to document vectors or any type of directional data. In addition to representing documents in a vector space, the obtained vectors can be normalized to be of unit length in the space, resulting in a set of points that occur on the surface of the unit sphere about the origin, after which the algorithm was named [17,18]. The logical interpretation of the produced clusters that a single cluster is expected to contain documents that belong to semantically related subjects. In a typical data set of documents, large number of terms may not occur in a single document, and the document vector would contain large number of zeros, causing the documents to be sparse [13,17]. Spherical *K*-means has shown its capability of taking advantage of the sparsity of documents.

The authors of this work present a new technique to enhance the performance of the spherical *K*-means to cluster sets of documents efficiently, and propose a new assessment metric that measures the clusters' compactness. First; we will introduce a procedure to initialize the algorithm by finding the seeds by which the algorithm starts. The initialization process relies on perturbing the space systematically, such that these initial points are vectors distributed among the document vectors as evenly as possible, following Anderberg's observation [41]. Starting with such seeds gives the opportunity for the algorithm to find better clusters, and faster convergence conditions.

Secondly; the centroids (mean vectors) of the spherical *K*-means tend to be orthonormal at the time of convergence [18], so we are interested in investigating the compactness of the resulting clusters by presenting a new measure for the vectors' dispersion about the centroids in the directional sense. The proposed measure can be represented as a special case of the moment generating function $G^m(t) = E(e^{t(\cos_i\theta - |c_i|)})$. The formula is obtained by setting $t = 0$ and $m = 2$. Further discussion and interpretation of the formula will be provided in Section 4.

The proposed algorithm and quality function were extensively tested on two different datasets. The first dataset consists of 21,826 documents acquired from [52]. The second dataset is the 20 News Group collection and consists of approximately 20,000 documents [53]. The assessment was based on comparing the proposed algorithm with the standard randomly initialed *K*-means algorithm on three metrics: the quality of generated clusters (intra cluster similarity), the cluster compactness and the time necessary of achieving the convergence function. The proposed algorithm straightforwardly outperforms the randomly initialized *K*-means algorithm when cluster quality and compactness are considered. However, the time necessary to achieve convergence was lesser in the case of the proposed algorithm for small number of clusters but started to increase (even become larger than the random *K*-means) for large number of clusters. The details of the experimentations and the results obtained are explained in Section 5.

The rest of this paper is organized as follows: Section 2 presents documents clustering using spherical *K*-means. Section 3, by comparison, describes existing initialization schemes for the *K*-means algorithm. Section 4 explains the proposed initialization technique and the proposed cluster quality measure. Section 5 summarizes the properties of the datasets used in this research and describes the experiments and the results that were obtained. Finally, Section 6 presents the conclusions of this work.

2. Document clustering using the spherical *K*-means

The *K*-means algorithm can be applied to cluster text documents as well. Normally the proximity measure will be the cosine similarity, when combined with the *K*-means, the resulting algorithm is known as the spherical *K*-means. This variant is isomorphic to the standard version, but differs in the associated accessories such as the specifically designated objective function and the nature of the vectors being processed. The vectors have to be normalized by their length, to have the vectors lying on the surface of the unit hypersphere about the origin. The normalization does not affect the documents since the original directions are preserved, a key issue concerning documents' domain.

The locations of data points in the space are not the key factor in deciding clusters memberships, whereas the direction of the subspace spanned by the 1-D dimensional lines of vectors is the driving criterion for defining relations among data instances. This fact is reflected by the choice for the proximity measure as being the cosine of the angle between any two vectors; the cosine similarity has proved to be far more efficient than the Euclidean distance in the process of cluster analysis of high dimensional directional data, such as textual data [40]. However, data points with directional characteristics can be treated as being generated by models of different distributions. Such model based algorithms iterate alternatively between a model re-estimation step and data re-assignment step, usually the MLE (Maximum Likelihood Estimation) method is used for the latter step [50]. A comprehensive assortment of model-based algorithms for clustering directional data were presented in [8–10,51], an empirical comparison of model-based clustering algorithms was conducted in [50] by providing a generic prototype algorithm that can be used to generate different combinations of models and data assignment methods. For more information regarding directional data, we refer the reader to [36].

The underlying transformation of documents to numeric vectors was exploited for information retrieval purposes, as well as for clustering paradigm. Dhillon and others [17] inspected the efficiency of clustering very large document collections preprocessed similarly, using the spherical *K*-means. They exhibited the efficiency of the algorithm by achieving a 23 min clustering of 113,716 NSF award abstracts on a single sequential workstation. However; they made use of some modifications on the technical and implementation issues, such as deploying multithreading to accelerate the preprocessing step, which involves too many I/O operations, with the support of scalable data structures such as local and global hash tables. Afterwards, the spherical *K*-means was applied, as an efficient method for clustering, with the capability of exploiting the sparseness of vectors (for large document collections, vectors are expected to have large proportion of zero entries, due to the huge diversity). The initialization used a perturbing scheme similar to the BS (Binary Splitting) of [34].

In a separate research [18], Dhillon and colleagues contemplated the effect of using the resulting clusters' centroids for matrix approximation of the input data, a procedure for projecting input points on a smaller subspace, retaining as much of the information embedded in the data, while discarding the noise. Beforehand, their experiments showed that the spherical *K*-means produces compact clusters, and then an astonishing breakthrough was achieved for matrix approximation when the converged centroids were used as a basis for projection. Compared to other effective schemes such as the truncated SVD (Singular Value Decomposition), they observed that concept vectors (centroids) are localized in the space, with a notable tendency to being orthonormal, a perfection that is longed for in such cases. This anomaly owes to the nature of the spherical *K*-means in exploiting the sparseness of documents. Concept vectors decomposition is a discovery rather than an innovation.

A clustering refinement algorithm is presented in [16], by suggesting a first variation principal to the algorithm, yielding to a ping-pong strategy that moves data points between clusters, to capture higher objective function values. The technique is suggested to be combined with the spherical *K*-means, applied as intermediate phases between the iterations. Formally, the first variation partition nextFV $\left(\{\pi_l\}_{l=1}^k\right)$ is obtained by moving a vector x from its nearest centroid's cluster π_i to another cluster π_j, where $\{\pi_l\}_{l=1}^k$ is the current partitioning. Merely moving a point from one cluster to another is simple and intuitive, this operation is carried out only as clusters become stable from the spherical *K*-means' perspective.

The authors defined the stopping criterion for the spherical *K*-means as a numeric difference between the objective function values for two successive iterations (in the case of checking the objective function values for convergence), expressed as Δ_k. Apparently the iterations of the algorithm are controlled by this numeric quantity; when $\Delta_k \leqslant 0$ for all points, the algorithm halts. Then they defined Δ as the criterion for the first variation (FV) iterations, as if they were applied in a stand alone fashion. They proved that the FV has higher values for Δ than for the spherical *K*-means, such that $\Delta \geqslant \Delta_k$. This implies that when the spherical *K*-means is about to stop when $\Delta_k \leqslant 0$, the FV may have a $\Delta \geqslant 0$, so applying the FV at this point can dig the obstacles off the spherical *K*-means' track, and the latter may continue. The combined refinement algorithm proved to be superior, by allowing the spherical *K*-means to escape from local maximum, and to capture higher objective function values that are beyond the vision of the plain algorithm.

Supposedly, the FV can aid the spherical *K*-means as the vector x_i to be moved is known in prior, but the study did not expound how to choose x_i, eventually the FV will encounter $\Delta \leqslant 0$, but examining different points may give different objective function values. However, $O(k)$ evaluations of objective function values for the $O(d)$ vectors are out of question for computational complexity, and yet approximations are required for optimal performance. Nevertheless, the algorithm mainly suits moderately sized document collections, and small clusters (fine grain clustering) as admitted by the authors.

The spherical *K*-means is applied predominantly in a batch mode, where centroids are updated only after assigning all vectors to their closest cluster. The batch mode refers to Forgy's [23] approach for assigning points to clusters, in contrast, we have seen the incremental mode in MaQueen's [35] variant of the *K*-means, where centroids are recomputed after allocating each point to a cluster. Generalizing the dialogue, incremental mode has a competitive learning nature; as a data point is processed, centroids are updated correspondingly. Soft competitive learning implicates updating each centroid, with a rate of adjustment proportional to the centroid's similarity with the point being processed, but a well known strategy; the WTA (Winner-Takes-All) where only one centroid (the closest to the point) will be updated, ascertained to be more efficient [9,49].

The benefits of competitive learning paradigms were investigated in the spherical *K*-means context. Banerjee and Gosh [9] used incremental WTA competitive learning to achieve balanced clustering, in order to avoid empty clusters. Their approach was combined with a frequency sensitive competitive learning (refer to [9]) as a counter mechanism that penalizes the assignment of a data point to a crowded cluster. The principle is intuitively appealing, but it was approached without the emphasis on efficiency or quality, time results and corresponding empirical comparisons were not provided.

An effort to ameliorate these shortcomings, Zhong attempted to refine their technique using different learning rate schedulers in [49]. He verified the supremacy of using a gradually decreasing learning rate to the flat rate, relating the incremental mode to the gradient ascent approach. The suggested OSKM (Online spherical *K*-means) with a WTA strategy updates the centroids using an exponentially decreasing rate. The OSKM incurs in a complexity of $O(MNd)$, where M is the number of batch iterations, N is the number of data points and d is the dimensionality (not to be confused with d mentioned earlier: the number of documents), obviously this is a high complexity. The author adjusted the running time with a neat trick; by exploiting the sparsity of textual data and reducing the complexity to $O(MKN_{nz})$ where the N_{nz} refers to the number of non-zero entries in the document-by-word matrix. The OSKM can be efficient, but as for MacQueens's [35] variant, incremental mode algorithms suffer the sensitivity to the instance order, so the clustering quality would be susceptible to this matter.

As it has been mentioned earlier, *K*-means clustering is a popular algorithm and it has been used in numerous applications. For example, Ayech and Ziou [5] have proposed an original version of *K*-means which is suitable for the segmentation of Terahertz images. This algorithm is called ranked-*K*-means which is less sensitive to the initialization problem. The authors reformulate the *K*-means clustering algorithm under ranked set sampling to overcome settling for local optima or for giving different results based on the initialization points.

Consensus clustering aims to find a single partitioning from multiple existing partitions. Too many algorithms have been suggested to achieve the previous goal. Of particular interest to us are the algorithms which utilize the *K*-means clustering algorithm such as the work reported in [42]. This work laid out the ground for utilizing the *K*-means algorithm for consensus clustering. However, in their work, they have focused on one objective function which is very restrictive for real life applications. The work reported in [45], by comparison, uses multiple utility functions to build a theoretic framework for *K*-means based consensus clustering. They transformed consensus clustering to *K*-means based clustering. The authors have also handled incomplete basic partitions. Their suggested algorithm performs very well on multiple datasets and was comparable to the state of art algorithms in this field.

3. Existing initialization schemes for the *K*-means algorithm

Previous work has shown that the *K*-means in its standard forms is appropriate and efficient for large sizes of data with high dimensionality, but the approaches to overcome the associated problems still suffer some serious disadvantages. However, since the *K*-means converges to local optimum solutions, it is widely agreed that the final result depends on the initial state (centers) by which the algorithm starts [38,41,46,43], hence, starting by better initial states would yield better local optima, and probably could be near global, thus, refining the initial state approach was considered in this paper.

Different perspectives for the starting state of the *K*-means exist; these perspectives differ in the initialization of the parameters so that the *K*-means can start. Classical methods for initialization include the random start, Forgy's [23], MacQueen's [35] and Kaufman's [32] initializations. These methods are compared by Pena and others [38] in terms of quality of clustering, and the sensitivity of the *K*-means to initial starting conditions (robustness), when each of the four initializations is adapted. Their study also considered the effect of the initialization methods on the convergence speed.

Their problem statement emphasizes that the *K*-means is sensitive to the initial state; thereby the running of the algorithm is a deterministic mapping from the initial state of parameters toward the final setting of these parameters, which typify the final local minimum solution. This is true since the final result of the same input data differs when restarted with deferent starting states, while the result remains the same when the *K*-means is re-invoked with similar initialization.

Non-trivial solutions approached the initialization issue, such as the strategy of MaxMin heuristic or the construction of hierarchical clustering as an initial clustering to the *K*-means [38]. These methods result in hybrid algorithms, which suffer the very same problem of the *K*-means. However, traditional approaches mentioned above are worthy to be examined by studying their behavior, which is achieved in [38].

The Random initialization gives a random partitioning of the data into k clusters, Forgy's approach (FA) chooses k initial seeds rather than partitions, these seeds are chosen randomly from the input data as initial centroids. MacQueen approach (MA) is similar to Forgy's, but differs in the re-computation of centroids, on each iteration, including the initial one, a centroid is recalculated incrementally as a data point is assigned to that centroid's cluster, thus the initial centroids (as well as the final centroids) are affected by the instance order by which the data points are examined. Kaufman's approach (KA) is a heuristic based initialization carried out by successive selection of points until k centroids are found, the first centroid is the point lying on the central location of the database, subsequent selections are based on the heuristic of choosing points with higher probability of having large number of instances around them.

The conclusions obtained by comparing the four approaches in [38] is that the Random and Kaufman's initialization outperform the other two in terms of final clustering quality and the robustness of the *K*-means. The MacQueen approach has proven to be the fastest converging approach; however, concerning the Random and Kaufman's initialization (since these are of better clustering quality) the Kaufman's initialization has a higher convergence speed when compared to the Random.

Bradley and Fayyad [14] introduced a refinement method that finds initial points for the K-means; their methodology can be generalized to other iterative clustering algorithms. The clustering framework adopted considers the data to be drawn from a mixture model, and the objective is to maximize the probability of data items, such that each item is assigned a probability as follows

$$\Pr(x|M) = \sum W_i \cdot \Pr(x|C_i, M), \ 1 \leqslant i \leqslant k$$

where M is the mixture model, W_i is a weight associated with each center in C_i, and the objective now is to determine the parameters of center locations C and their weights W that maximize the given probability.

They stated that parameters can be refined iteratively via K-means or other iterative techniques, but the resulting refinement is sensitive to the initial parameters, so a refinement algorithm is suggested, which relies on sub-sampling the data set extensively and clustering each subset using the K-means independently, given that empty clusters are not allowed for consistency issues. Having j clustering results with k centroids for each, the set of $j * k$ points are now clustered via K-means repeatedly using the j subsets of points as centroids each time. The reason for clustering the $j * k$ points is to overcome the problem of noisy estimates associated with sub-sampling methods, we could choose the centers of one of the j subsets with least distortion (or highest joint probability), but noisy estimates especially in skewed distributions and high dimensional data occur frequently.

The results of this method were quite impressive, notice that the essence of the procedure is to capture the joint probability distribution by estimating the modes' locations of the data. Eventually, the centroids will be located on these modes by the K-means in few iterations (faster convergence), with higher clustering accuracy. But this method requires much information priory, for example the algorithm requires many points to refine, of course these points are relatively small to the data size, but these points are not selected trivially, rather these are computed via sub-sampling the data severely. Furthermore, the K-means is evoked several times as part of the refinement, which is supposed to be a preprocessing step to the K-means algorithm itself, thus there is no guarantee that the resulting initial points would be efficient unless the number of sub-samples drawn is sufficiently large, which increases the time complexity in an undesirable manner, in other words, their method needs initialization too. All in all, density estimation of high dimensional data remains to be a difficult task.

He and others [24] presented a comparative study for different initialization methods; they categorized these methods into three major families: random based, distance optimization based and density estimation methods. Random based such as Forgy's random selection of k seeds, abbreviated as R-SEL, and MacQueens's initialization R-MEAN as in [24], are described earlier. Distance optimization methods include the SCS (Simple Cluster Seeking) which starts by the first input x_1 as the initial seed c_1, for the rest of the input, a point x_j becomes a seed if $|x_j - c_k| > \rho$ for all selected seeds c_k. This method is sensitive to the value of ρ and the instance order. Another distance based method is the KZZ [24,31], starts by an initial seed c_1 from the input whose norm is maximum (maximal length vector), for the rest of the input, choose the input point which has the maximum distance from all seed points chosen already. The KZZ is more flexible than the SCS since it does not require the determination of a threshold ρ, but it's sensitive to the instance order, and incurs in too many distance calculations.

The third family, density estimation methods, includes the Kaufman initialization, which is also described earlier, the KR's main drawback (as for Kaufman & Rousseeuw in [32]) is the high computational complexity. Another method, by Al-Daoud and Roberts [2], relies on dividing the space R^d into M smaller subspaces, each of which spanned by a proportion of the data points, and the seeds are distributed evenly across these subspaces. For each subspace, initial seeds are chosen randomly, the given method is sensitive to the number of subspaces M, which has to be compatible with k somehow, or else it would affect the density estimation. To avoid this problem the authors in [2] refined the scheme by subdividing the space into disjoint sets, and the number of seeds in each set is evaluated using a polynomial equation in the order of number of dimensions, the validity of this refinement is not proven yet.

The study in [24] considered the inter-cluster separation as a factor in evaluating the goodness of clustering methods along the compactness (intra-cluster similarity) of individual clusters. Captivating conclusions were drawn according to their experiments; distance optimization methods (SCS&KZZ) were superior in the sense of cluster separability, the KZZ gave higher performance than the SCS as expected. On the other hand, The Kaufman (KR) density estimation method did not seem to be more attractive than the random methods, except for a modest speedup in the convergence toward the final solution, a finding that goes with the conclusions drawn in [38], the authors argued that this observation owes to the nature of the K-means, which solely behave toward refining the intra-cluster's connectivity, with no regard to separating distinct clusters. Finally, random methods were comparable against each other, their performance is affected by some parameters such as the number of clusters k, and the nature of data input (noisy vs. clean).

A divisive hierarchical partitioning based approach for initializing the K-means was performed by Su and Dy in [41]. Actually their work relies on a smart choice for splitting the clusters, an approach that bears a resemblance to the PCA (Principal Component Analysis). The PCA constructs a covariance matrix; this matrix characterizes the unseen relationships between pairs of dimensions regarding the input data, therefore the eigenvector that corresponds to the highest eigenvalue is considered to be the direction that contributes the most to the SSE (distortion), and it is a good candidate to project a cluster for splitting.

The PCA-Part (Principal Component Analysis Partitioning) algorithm suggested in [41] starts with a single cluster of all data, the direction that will be chosen for projection is evaluated in a simpler manner than for the PCA (which evaluates the eigenvectors), the technique examines a basis for the space R^d, wherein the input data exist, the vectors of the basis are orthonormal, meaning that they are orthogonal to each other and normalized by their lengths, so they form the orthogonal matrix. The most straightforward basis is simply the standard basis of the identity I. Now each vector e in I will be examined to choose the projection direction: the vector that maximizes the value $a = \sum_{x_i \in c}(y_{ip}e_p - \alpha_p e_p)^2$, y_{ip} is the projection of a data point y_i on the vector e_p in the basis, where α_p is the projection of the mean μ of the cluster C at hand. Now the point x_i whose projection is y_i will be assigned to the new cluster $C_1 \subset C$ if $y_i \leqslant \alpha_p$, otherwise assign x_i to c_2. The two new clusters C_1 and C_2 with their new means μ_1 and μ_2 are supposed to have aggregate $SSE_{\text{new}} = SSE\,(C_1) + SSE\,(C_2) < SSE\,(C)$. The process continues on partitioning clusters down the hierarchy, choosing the cluster with largest *SSE* at each stage to split, until k clusters are obtained. These k clusters in turn form the initial partitioning for the K-means.

The results of the given technique in [41] are desirable in terms of quality and speed of convergence, but the time complexity of the PCA-Part when combined with the K-means is relatively high, the authors claim that it takes less time than restarting the K-means 10 times. Moreover, choosing the direction of splitting considers only one dimension, computationally such evaluation is efficient, but does not promise to result in the real first principal direction (eigenvector of the largest eigenvalue), as depicted by their study. The examinations for finding the projection direction are simple that it's restricted to one dimensional subspaces of projection, the probability holds for having subspaces of 2-D or more that may contribute to the *SSE* significantly, since the examination is distance based not covariance matrix based. However, new research directions are established, and since the technique is deterministic, the study of its behavior is possible, and enhancements are prospective.

An intuitive method for initialization also presented recently by Yuan and others in [47], they improved the K-means by finding a near optimal estimation of centroids locations, which in turn treated as the initial points for the K-means. Their method exhaustively computes the distances between all pairs of data points, to construct compact sets in terms of distances among them. The algorithm starts by a pair wise comparison of distances between all data points in the input, and the pair of closest points to each other (that correspond to the smallest distance in the input set) are joined in a set A_1, the remaining points are inspected, and the point that has a minimal distance from any point in A_1 will be added, and so on. The set A_1 grows incrementally until the number of points incorporated in A_1 exceeds a given threshold, which is proportional to the total number of points n and the number of clusters k. similarly, k sets are formed in the same manner, the partitioning of points (k sets) is the initial clustering whose computed centroids become the initial points for the K-means.

This method uses the Euclidean distances as the proximity measure, and each set A_k has a number of points limited by $\alpha * n/k$, where $0 < \alpha \leqslant 1$, n is the input size and k is the number of clusters. The algorithm gave better clustering quality than the randomly initialized version of the K-means, but the algorithm requires some modifications due to the vagueness of some issues, for example the algorithm is sensitive to the value of α, the authors claim that 0.75 is a setting with satisfactory efficiency, but the value is experimentally obtained according to the data sets under experimentation, whereas such factors have to be artificially set as being data specific, to scale the scheme for different data topologies. Obviously, the algorithm incurs in a high computational complexity due to the extensive distance calculations, which seem to follow a Brute-force discipline, the required distance computations can be found in a static proximity matrix constructed once, but the creation of such matrix is prohibitive.

4. The proposed initialization methodology and objective function

This section provides the details of the proposed initialization technique for the spherical K-means and the new proposed objective function which measures documents' dispersion from the clusters' centroids (i.e. it measures cluster compactness).

4.1. The initialization process

Through this work, we suggest a novel approach to initialize the spherical K-means algorithm. Initialization approaches presented so far attempt to capture the density of data, or to optimize the distance criteria between initial points. These approaches encountered problems such as the need for initialization themselves, or the infiltration of the standard K-means' shield against the curse of dimensionality. Obviously; these issues are not useful for documents' domain, which engages very high dimensionality. Besides, these approaches follow Euclidean distance disciplines that are not suitable for directional data.

In our suggested technique, we define some criteria for defining the most significant term or dimension to be the basis for distributing the initial points. The process starts by defining two hypothetical vectors: the minimum vector hv_1 and the maximum vector hv_2; the hv_1 vector contains minimum weights of all terms across documents, and the hv_2 vector contains maximum weights. These vectors are supposed to act like boundaries surrounding the input vectors. Notice that the spectrum of vectors in the document's domain occur solely in the positive portion of the space R^w (words cannot have negative occurrence). Formally, $hv_1 = (hv_{11}, \ldots, hv_{w1})$ and $hv_2 = (hv_{12}, \ldots, hv_{w2})$ such that

$$hv_{j1} = \min(x_{ji}),\ 1 \leqslant i \leqslant d$$

$$hv_{j2} = \max(x_{ji}),\ 1 \leqslant i \leqslant d$$

where j corresponds to the j-th term in the hv_1 or hv_2 vectors.

The weights of the hypothetical vectors play the major role in specifying the initial vectors' weights. Each term is handled as an independent scale, then partitioned equally into $(k+1)$ intervals. The frontiers of these intervals will comprise the weights of the k vectors for the given term. The evaluation of these intervals is illustrated in the following.

The first mean m_1 consists of the weights

$$m_1 = \begin{bmatrix} hv_{11} + \frac{(hv_{12}-hv_{11})}{k+1} \\ \cdot \\ \cdot \\ \cdot \\ hv_{w1} + \frac{(hv_{w2}-hv_{w1})}{k+1} \end{bmatrix}$$

Whereas the second mean m_2 will consist of the weights

$$m_2 = \begin{bmatrix} hv_{11} + 2 * \frac{(hv_{12}-hv_{11})}{k+1} \\ \cdot \\ \cdot \\ \cdot \\ hv_{w1} + 2 * \frac{(hv_{w2}-hv_{w1})}{k+1} \end{bmatrix}$$

And the k-the mean m_k's weights are

$$m_k = \begin{bmatrix} hv_{11} + k * \frac{(hv_{12}-hv_{11})}{k+1} \\ \cdot \\ \cdot \\ \cdot \\ hv_{w1} + k * \frac{(hv_{w2}-hv_{w1})}{k+1} \end{bmatrix}$$

The initial means are obtained using the terms with no regard of any interrelation evaluation between them, so we do not expect them to be located on the unit sphere. The set of initial means have to be normalized:

$$c_j = \frac{m_j}{|m_j|} \quad \text{where } 1 \leqslant j \leqslant k,$$

These initial points are to be the starting parameters, along the input value for k, for the spherical K-means. Now the new algorithm is depicted in Fig. 1.

According to this technique, the initial points are distributed in correspondence to the weights present in the input vectors. This algorithm follows a deterministic perturbing discipline, which is akin to the BS (Binary Splitting) algorithm of [34]. The BS finds the first seed as the mean of the input, and randomly perturbs each seed twice, then the iterations cease when k seeds are found. On the contrary, our algorithm is systematically deterministic, and the perturbation is performed in a single step. Our chief objective is to propose an initialization technique without the need of being initialized itself, but with the emphasis on retaining the algorithm's efficiency and its immunity against the curse of dimensionality, which is crucial in documents clustering context.

4.2. Cluster compactness

As mentioned earlier, the quality metric for the clusters cannot follow the distance based objective functions, due to the relevance issues in document's domain. An objective function that maintains the integrity of such analysis is the intra similarity of clusters, where clusters are expected to contain relevant documents, thus the quality of a cluster:

$$f(\pi_j) = \sum_{x_i \in \pi_j} x_i^T c_j$$

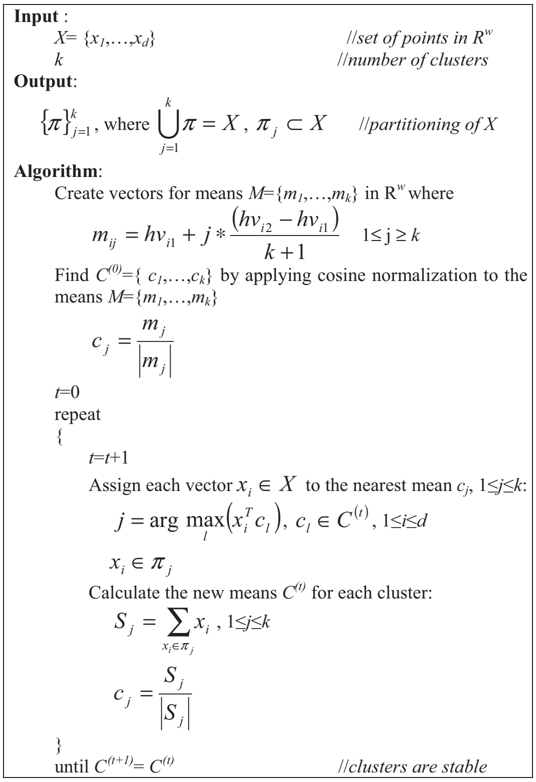
where c_j is the centroid of the cluster, evaluated by normalizing the sum of vectors S_j:

$$S_j = \sum_{x_i \in \pi_j} x_i$$

$$c_j = \frac{S_j}{|S_j|}$$

Notice that since all vectors $x_i \in X$ and centers c_j, $1 \leqslant j \leqslant k$ are normalized, the $f(\pi_j)$ is exactly the sum of cosine similarities in the given cluster. Now the objective function that is subject to maximization is:

$$f\{\pi_j\}_{j=1}^{k} = \sum_{j=1}^{k} f(\pi_j) = \sum_{j=1}^{k} \sum_{x_i \in \pi_j} x_i^T c_j$$



```
Input :
    $X = \{x_1,\ldots,x_d\}$                //set of points in $R^w$
    $k$                                     //number of clusters
Output:
    $\{\pi\}_{j=1}^{k}$, where $\bigcup_{j=1}^{k} \pi = X$ , $\pi_j \subset X$      //partitioning of X
Algorithm:
    Create vectors for means $M=\{m_1,\ldots,m_k\}$ in $R^w$ where
        $m_{ij} = hv_{i1} + j * \frac{(hv_{i2} - hv_{i1})}{k+1}$     $1 \leq j \geq k$
    Find $C^{(0)}=\{c_1,\ldots,c_k\}$ by applying cosine normalization to the means $M=\{m_1,\ldots,m_k\}$
        $c_j = \frac{m_j}{|m_j|}$
    $t=0$
    repeat
    {
        $t=t+1$
        Assign each vector $x_i \in X$ to the nearest mean $c_j$, $1\leq j\leq k$:
            $j = \arg\max_l (x_i^T c_l)$, $c_l \in C^{(t)}$, $1\leq i\leq d$
            $x_i \in \pi_j$
        Calculate the new means $C^{(t)}$ for each cluster:
            $S_j = \sum_{x_i \in \pi_j} x_i$ , $1\leq j\leq k$
            $c_j = \frac{S_j}{|S_j|}$
    }
    until $C^{(t+1)} = C^{(t)}$             //clusters are stable
```

Fig. 1. The initialized spherical *K*-means.

In addition to the above widely used objective function, we propose a new assessment metric that measures the variance of the documents about the corresponding centroids in the directional sense. As the cosine similarity for each document and its centroid is already calculated, we utilize these values, since the cosine represents the directional relevance in the space, which is not served by the Euclidean distance measures.

Using the cosine values, we induce a reasonable measure as follows: for each document, it can form a 2-D subspace along with the centroid, on the surface of this subspace; we create an imaginary unit circle about the origin as shown in Fig. 2.

Now the projection of a vector on the corresponding centroid results in a new vector on the line of the centroid. It can be verified that the length of the projected vector is affected directly by the angle in-between, such that the smaller the angle between the vector and the centroid, the larger the length of the projected vector becomes. In Fig. 2, the projected vector is evaluated directly as

$$x'_1 = c_j^T x_1 c_j$$

x'_2 is evaluated similarly. When a vector is projected onto the centroid, the process involves calculating a scalar multiplier that multiplies the centroid to obtain the new vector. However, we are now interested in the length of the new vector $|x'_1|$, which is less than 1, according to the triangle $Ox_1x'_1$, it can be indicated by the following trigonometric identity

$$|x'_1| = \cos\theta \cdot |x_1| \Rightarrow |x'_1| = \cos\theta$$

This relation applies to any normalized vector when projected on the corresponding normalized centroid, and the length of this projection is exactly the cosine similarity calculated earlier. As vectors of the cluster are projected, resulting in a set of points on the centroid's line, these points have to be closer to the point c_j in the space as the original vectors are directionally closer to the centroid, thus the directional variance can be represented as the dispersion of the projected vectors' lengths about $|c_j|$.

Having this intuition in mind, we carry out the general formulation: we define a discrete random variable Y that assumes any vector $x_i \in X$, and the function $g(Y) = \cos_i\theta - 1$, where

$$\cos_i\theta = \arg\max_l (x_i^T c_l), \ 1 \leqslant i \leqslant d, \ 1 \leqslant l \leqslant k, \ x_i \in X$$

Since the random variables defined above are of the discrete type, then the moment generating function

$$G(t) = E(e^{t \cdot g(Y)}) = E(e^{t(\cos_i\theta - 1)})$$

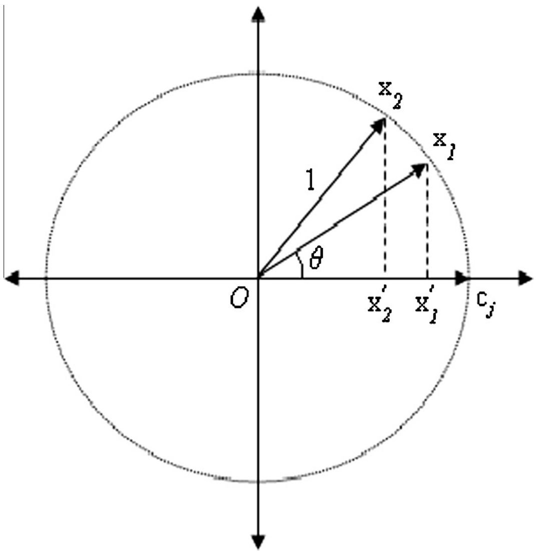


Fig. 2. The projection of a vector on the centroid.

can be redefined as the following expectation

$$G(t) = \sum_{i=1}^{d} e^{t(\cos_i\theta - 1)} f(x_i)$$

If this expectation was differentiated m times, then

$$G^m(t) = \frac{d^m G}{dt^m} = \sum_{i=1}^{d} (\cos_i\theta - 1)^m e^{t(\cos_i\theta - 1)} f(x_i)$$

For the time being, we refer to the directional variance as the second moment about $|c_l| = 1,\ 1 \leqslant l \leqslant k$ for the random variable $g(Y)$, which can be generated by the setting of $m = 2$ and $t = 0$

$$G^2(0) = \frac{d^2 G}{dt^2}(t = 0) = \sum_{i=1}^{d} (\cos_i\theta - 1)^2 f(x_i)$$

As Y is a discrete random variable that has an outcome space of d equally likely cases, then the underlying distribution function is uniform, which implies that $f(x) = 1/d$ for all $x_i \in X$ for which Y can assume a value with positive probability. However, due to the degrees of freedom $d - 1$ that the calculation of the variance possesses, the final formula becomes

$$\text{var} = \frac{1}{d-1} \sum_{i=1}^{d} (\cos_i\theta - 1)^2$$

where d is the number of documents, $\cos_i\theta = \arg\max_l(x_i^T c_l),\ 1 \leqslant i \leqslant d,\ 1 \leqslant l \leqslant k$, and the term 1 represents the length of the centroids $|c_l|$.

This formula is the second moment of the random variable $g(Y)$ about the $|c_l| = 1$, notice that the second moment is evaluated about the directional center of gravity, so it can be regarded as the second *central* moment, which is the directional variance. Finally, it is noteworthy that the proposed directional variance is a minimization function, but by generating the first central moment (setting $m = 1$ and $t = 0$), then we will acquire a maximization function, because $G'(0) \leqslant 0$ for all $x_i \in X$. Expressly; regardless of the choices made for m and t, the resulting moment is optimized by approaching 0.

5. Experimentation and result analysis

The proposed algorithm was implemented using the C#.NET environment; all experiments were carried out on a Pentium 2.4 GHz sequential platform, with 1 GB of main memory.

5.1. Datasets

Through the course of clustering analysis development, it is agreed that there is no specific clustering algorithm for which it can be claimed that it is the optimal algorithm across all application domains, and the performance of an algorithm is data dependent [10]. In order to avoid the effect of this dependency on the reliability of the experiments, they were conducted on two different datasets.

The first dataset is a collection of 21,826 documents acquired from different resources such as LISA (Library and Information Science Abstracts) group, abstracts from medical and other journals [52], the given dataset occupies 12.9 Mbytes of disk space. The second dataset, the 20 newsgroup [53], consists of approximately 20,000 documents from different Usenet newsgroups, occupying 43.9 Mbytes of disk size.

5.2. Preprocessing and dataset representation

A preprocessing step for document clustering is to create a space of features that comprises a reference from which document vectors are drawn. Each feature denotes a term that occurs in the document collection, these terms are selected as words with unique content bearing potential [17], such that having certain capability of discrimination between documents. Each term will be assigned an identifier, and a document becomes a weighted vector of these terms, each weight measures some characteristics of the corresponding term, as the occurrence frequency along a global measure of the term's discrimination power across the collection [6]. Several feature selection and weighting schemes exist; the adopted data preparation schemes in this work are as follows:

1. Lexical Analysis: consists of extracting terms with great deal of importance, and omitting terms for which measurements cause noisiness to the data. This will be carried out in two steps:
   (a) Stopwords removal: A predefined set of words are removed from the documents since they have low discrimination power such that they are likely to be frequent across all the documents in the corpus.
   (b) Stemming: Words with identical morphological roots are considered isomorphic, thus these are represented as one word; their stem. A number of stemming techniques are presented through the literature, a well known algorithm; the Porter algorithm is adopted from the work of [29].
2. Representing documents as vectors: at this stage, w terms are selected; each term is assigned an identifier between 1...w. Documents are represented as $X = \{x_1 \ldots x_d\}$ where X is the word-by-document matrix, whose rows are the terms and the columns are the document vectors, x_i is a vector in R^w corresponding to the i-th document, and d is the number of documents. For each vector x_i, the j-th component, related to the j-th term, is a numeric weight obtained by the product:

$$x_{ji} = tf_{ji} * IDF_j$$

where tf_{ji} is the number of occurrences of the j-th term in the i-th document, and IDF_j is a global measure of the j-th term's importance across all documents, defined by $IDF_j = \log(d/d_j)$; d_j is the number of documents containing the j-th term. The described notions and general framework are almost similar to the ones adopted in [6,17].

3. Normalization of vectors: after defining the space (terms) and representing the documents as vectors in the space, these vectors are normalized by their length, such that:

$$\forall x_i \in X : x_i \leftarrow \frac{x_i}{|x_i|} \quad \text{where } |x_i| = (x_i^T x_i)^{\frac{1}{2}} = \left(\sum_{j=1}^{w} (tf_{ji} IDF_j)^2\right)^{\frac{1}{2}}$$

Now each document has unit length:

$$|x_i| = 1 \text{ or } x_i^T x_i = 1, \quad 1 \leqslant i \leqslant d$$

Normalizing document vectors to be of unit length implies that documents lie on the surface of the unit sphere around the origin, thus the application of the K-means to normalized document vectors is known as the spherical K-means.

5.3. Experiments

Through the experiments, the proposed method is compared against the randomly initialized variant of the spherical K-means, which has been used as a rule of thumb for starting the algorithm. Generic initialization methods presented in the context of the K-means disregard directional considerations. Our attempt is to validate initializing the spherical K-means by proving its superiority over the randomly initialized version.

The initialized K-means versus the randomly-initialized version are compared by different assessment metrics; namely, the objective function, the proposed directional variance of the vectors, which is a disguise for clusters' compactness, and the time of execution. Time measurements are taken solely for the clustering part of the algorithms, whereas the time required for documents preprocessing: establishing the vector space and building inverted files, is not taken into account, since the two algorithms are applied to the same set of vectors, thus preprocessing time does not affect time judgments. Finally, no prior assumptions are made about the necessary number of clusters, which is specified by the user.

5.3.1. Objective function comparisons

The objective function is a maximization function that measures the similarity of documents in clusters, where higher objective function values indicate higher relevance between the documents in each cluster.

Fig. 3 compares the obtained values of the objective function for the two algorithms applied to the first dataset acquired from [52]. As the figure shows; the initialized algorithm demonstrates its superiority over the random algorithm. As the number of clusters (k) increases, the objective function values increase accordingly, which is natural since additional centroids imply increased cosine similarities.

Fig. 4 elaborates the difference in objective function values again for the two algorithms applied upon the 20 newsgroups dataset. The figure shows the superiority of the proposed algorithm in terms of quality. It is also notable the ongoing increase of the objective function values as the number of clusters k increases. However, the optimal number of clusters is subjective since the abstract conceptualization of a clustering quality in such numerical values may increase up to d (total number of documents), where each document matches its own centroid with a cosine similarity equals 1.

5.3.2. Compactness comparisons

The directional variance is a minimization function (when $m = 2$ and $t = 0$), it measures the dispersion of the directions of the vectors about their corresponding centroids, simulating clusters' compactness.

Fig. 5 shows compactness evaluation of the resulting clusters for the first dataset. The initialized algorithm yielded clusters of smaller directional variance, which is desirable as this indicates more compact clusters. The values of the directional variance behave inversely to the number of clusters, as the latter increases, the directional variance decreases correspondingly, where the additional centroids would reduce the directional dispersion in the space.

The investigation of clusters' compactness for the 20 newsgroups dataset is depicted in Fig. 6, the proposed algorithm has achieved improved clustering in terms of this criterion. Similarly, the directional variance decreases as the number of clusters k increases. As mentioned before, the directional variance is a minimization function, but yet, this is subjective to the choices of m and t, for example; the directional first central moment ($m = 1$ and $t = 0$) would be a maximization function. For brevity, different adjustments of the moment generating function $G^m(t)$ are optimized by reaching 0, which cannot be reached unless we have the d documents matching their d centroids, in other words; as the objective function reaches d.

5.3.3. Time comparisons

The following figures demonstrate and compare the execution times required by the initialized and the random versions of the spherical K-means. Preprocessing documents was not accounted for in time evaluation, since the two algorithms will operate on the same data. The two algorithms were applied for different number of clusters (k) for the two datasets described above.

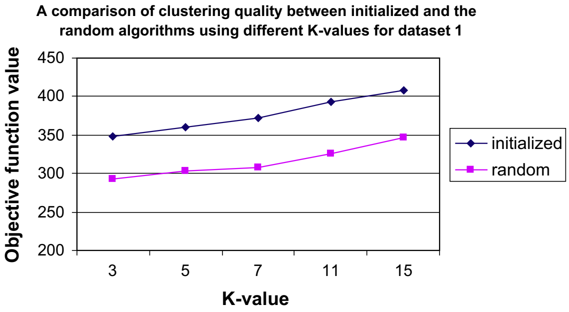


Fig. 3. Objective function values for dataset1.

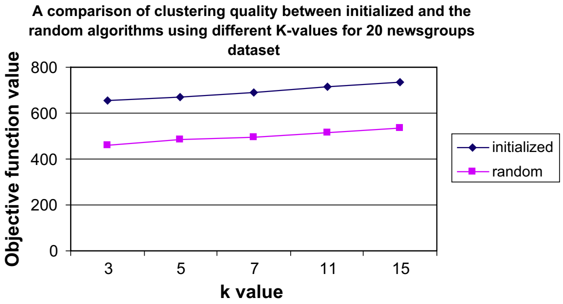


Fig. 4. Objective function values for 20 newsgroups dataset.

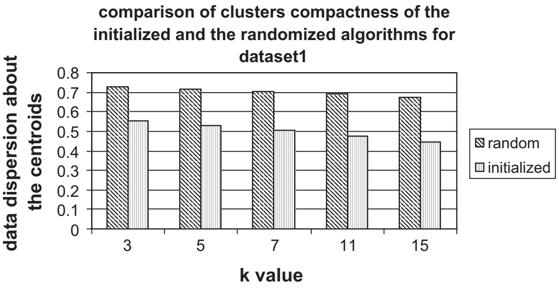


Fig. 5. Directional variance values for dataset1.

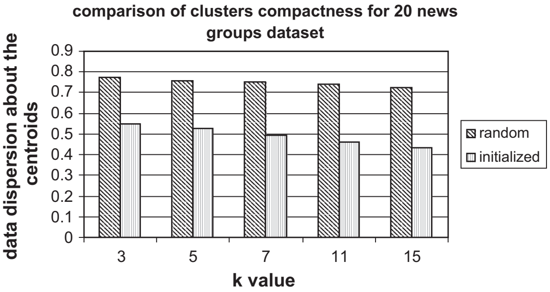


Fig. 6. Directional variance values for 20 newsgroups dataset.

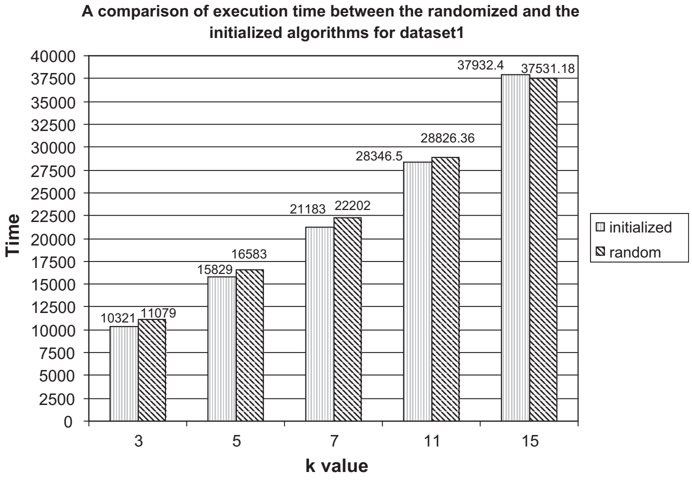


Fig. 7. Execution times for dataset1 (in seconds).

Execution times in Fig. 7 are evaluated for the initialized and the randomly initialized algorithms applied on dataset1 with different values for number of clusters *k*. The figure shows that the initialized version outperforms the random algorithm for smaller number of clusters, despite the additional calculations of the initial centroids, but this can only be justified by the faster convergence, since the initial centroids are chosen artificially. However, as the number of clusters increase, the difference in execution time becomes less, specifically, at $k = 15$, the performance of the initialized algorithm degraded against the randomly initialized algorithm, a potential explanation for this behavior is that increased computations for additional clusters' seeds seem to start outweighing the convergence criteria in time evaluation.

Fig. 8 demonstrates the execution time comparisons of the two algorithms as they are applied on the 20 newsgroups dataset. Similar to Fig. 7, the initialized algorithm has shown the same behavior, where it gained better execution times for small number of clusters due to faster convergence, but at the values $k = 11$ and $k = 15$, the speed of convergence became less significant compared to the growing size of computations, and the random algorithm started to outperform the proposed algorithm.

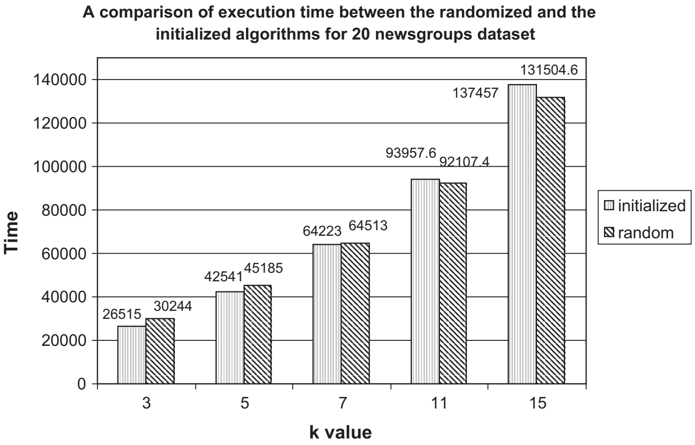


Fig. 8. Execution times for 20 newsgroups dataset (in seconds).

5.4. Final remarks

As it was noticed from the above experiments, the proposed *K*-means algorithm (referred to as initialized *K*-means) clearly outperforms the randomly initialized *K*-means when the objective function and cluster compactness were considered. As for when time of convergence is considered, the initialized *K* means outperforms the randomly-initialized *K* means when the number of clusters is small. However, when the number of clusters increases the random *K*-means starts to converge faster than the initialized *K*-means. The value of *K* at which the initialized *K*-means changes its behavior is dataset dependent. It was 15 for dataset1 and 11 for the 20 Newsgroups dataset.

6. Conclusions

In this paper, we proposed a novel approach for initializing the spherical *K*-means algorithm, and introduced a new assessment metric for clusters' compactness, which measures the directional dispersion of vectors about their corresponding centroids. The spherical *K*-means is a variant of the *K*-means whose input is a set of unit vectors that lie on the surface of the unit hypersphere about the origin, and uses the cosine similarity as its proximity measure. The data to be clustered is regarded as directional data, such that the magnitude of a vector in the space is insignificant, whereas the direction of the line (1-D subspace) spanned by that vector is the key factor in deciding the shapes of clusters. As the problem of clustering is dominated by such directional considerations, the cosine similarity proved to be superior to other measures in revealing the relations among the data, as shown by the study in [40]. In the generative models of cluster analysis, directional data are assumed to be generated from a set of vMF (von Mises-Fisher) distributions [8], in analogy to the Gaussian mixtures (normally distributed groups) of the Euclidean types of data with location related considerations.

Initialization of the *K*-means algorithm is regarded as a sufficient approach to remedy the problem of local optimality [24,31,38,41]. The various initialization schemes presented through the literature were inappropriate for document clustering, either in the directional sense, or their vulnerability to high dimensionality due to the extensive operations involved; such as repetitive subsampling or prohibitive pair wise comparisons of data instances, moreover, many of these methods require initialization themselves.

For the best of our knowledge, the issue of estimating initial means for the spherical *K*-means was not inspected, thus our attempt is to suggest a novel approach for initializing the spherical *K*-means, in order to achieve higher quality clustering according to the assessment metrics mentioned in section 4. The proposed algorithm follows a perturbation discipline, where the input space is subdivided into equivalent intervals, based on the weights present in the vectors, and the boundaries of these intervals comprise the set of initial means. This subdivision follows Anderburg's observation that good initial means are distributed as evenly as possible. We focused on simplicity to retain the algorithm's efficiency, by avoiding severe computations associated with pair wise comparisons of data instances, additionally the initialization is carried out in a deterministic manner, and does not require initialization for itself.

Clustering algorithms are judged using various quality metrics, or objective functions, where many of these rely on the proximity measure. The spherical *K*-means' clusters are assessed using a maximization objective function that measures the intra-cosine similarity for the clusters; this is viable since each centroid have the desired intermediate direction of the cluster. In the absence of visualization ability, it is difficult to speculate the shapes of clusters; however, we introduced a new assessment metric that simulates cluster's compactness as a measure of the directional dispersion of the vectors about their centroid. The choices for basic evaluations were made by projecting the vectors on the centroid's 1-D subspace. Then the formal equation was derived from the moment generating function $G^2(0)$, yielding to a directional second central moment of the random variable denoting the set of cosine similarities, with the center of gravity $|c_l| = 1$.

Finally, our experiments were performed on different datasets and different values of k (number of clusters), comparing the initialized variant against the randomly initialized variant of the spherical K-means algorithm. Quality results according to the sum of intra cosine similarities as well as clusters' compactness (proposed measure) proved that the initialized variant outperforms the randomly initialized variant, whereas time results have shown that the initialized variant's execution times increase at a faster rate with respect to the value k, than for the randomly initialized variant; this behavior can be explained by the increase in computations required for larger number of centroids.

References

[1] P. Agarwal, C.M. Procopiuc, Exact and approximation algorithms for clustering, in: Proceedings of the 9th ACM-SIAM Symposium on Discrete Algorithms, 1998, pp. 658–667.

[2] M. Al-Daoud, S. Roberts, New methods for the initialization of clusters, Pattern Recogn. Lett. 17 (5) (1994) 451–455.

[3] V.M. Amala-Bai, D. Manimegalai, An analysis of document clustering algorithms, in: Proceedings of the IEEE Conference on Communication Control and Computing Technologies (ICCCCT), 7–9 October, India, 2010.

[4] S. Arora, P. Raghavan, S. Rao, Approximation schemes for Euclidean K-median and related problems, in: Proceedings of the 30th Annual ACM Symposium on Theory of Computing, 1998, pp. 106–113.

[5] M.W. Ayech, J. Ziou, Segmentation of Terahertz imaging using k-means clustering based on ranked set sampling, Expert Syst. Appl. 42 (6) (2015) 2959–2974.

[6] R. Baeza-Yates, B. Ribeiro-Neto, Modern Information Retrieval, Addison Wesley, 1999.

[7] L. Bai, J. Liang, C. Sui, C. Dang, Fast global k-means clustering based on local geometrical information, Inf. Sci. 245 (2013) 168–180.

[8] A. Banerjee, I. Dhillon, J. Ghosh, S. Sra, Clustering on the unit hypersphere using von Mises-Fisher distributions, J. Mach. Learn. 6 (2005) 1345–1382.

[9] A. Banerjee, J. Ghosh, Frequency-sensitive competitive learning for scalable balanced clustering on high-dimensional hyperspheres, IEEE Trans. Neural Networks 15 (3) (2004) 702–719.

[10] A. Banerjee, I. Dhillon, J. Ghosh, S. Sra, Generative model-based clustering of directional data, in: Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining KDD, 2003.

[11] F. Beil, M. Ester, X. Xu, Frequent term-based text clustering, in: Proceedings of the 8th International Conference on Knowledge Discovery and Data Mining (KDD), 2002, pp. 436–442.

[12] Berkhin P. Survey of clustering data mining techniques. Technical report, Accrue Software, 2002.

[13] E. Bingham, H. Mannila, Random projection in dimensionality reduction: applications to image and text data, in: Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2001, pp. 245–250.

[14] P. Bradley, U. Fayyad, Refining initial points for k-means clustering, in: Proceedings of the 15th International Conference on Machine Learning (ICML98), 1998, pp. 91–99.

[15] J. Cao, Z. Wu, J. Wu, W. Liu, Towards information-theoretic K-means clustering for image indexing, Signal Process. 93 (2013) 2026–2037.

[16] I. Dhillon, Y. Guan, J. Kogan, Iterative clustering of high dimensional text data augmented by local search, in: Proceedings of the 2002 IEEE International Conference on Data Mining, 2002.

[17] I. Dhillon, J. Fan, Y. Guan, Efficient clustering of very large document collections, in: Data Mining for Scientific and Engineering Applications, Kluwer Academic Publishers, 2001, pp. 357–381.

[18] I. Dhillon, D. Modha, Concept decompositions for large sparse text data using clustering, Mach. Learn. 42 (1) (2001) 143–175.

[19] Q. Du, V. Faber, M. Gunzburger, Centroidal voronoi tessellations: applications and algorithms, SIAM Rev. 41 (4) (1999) 637–676.

[20] F. Dzogang, C. Marsala, M.J. Lesot, M. Rifqi, An Ellipsoidal K-means for document clustering, Proceedings of the 12th IEEE International Conference on Data Mining (ICDM), Brussels, Belgium | December 10–13, 2012.

[21] V. Faber, Clustering and the continuous K-means algorithm, Los Alamos Sci. 22 (1994) 138–144.

[22] U. Fayyad, C. Reina, P. Bradley, Initialization of iterative refinement clustering algorithms, in: Proceedings of the 4th International Conference on Knowledge Discovery & Data Mining (KDD98), 1998, pp. 194–198.

[23] E. Forgy, Cluster analysis of multivariate data: efficiency vs. interpretability of classifications, Biometrics 21 (1965) 768.

[24] J. He, M. Lan, C. Tan, S. Sung, H. Low, Initialization of cluster refinement algorithms: A review and comparative study, in: proceedings of the IEEE International Joint Conference on Neural Networks, 2004, pp. 297–302.

[25] K. Hornik, I. Feinerer, M. Kober, C. Buchta, Spherical k-means clustering, J. Stat. Softw. 50 (10) (2012) 1–22.

[26] A. Jain, M. Murty, P. Flynn, Data clustering: a review, ACM Comput. Surv. 31 (3) (1999) 264–323.

[27] A. Jain, R. Dubes, Algorithms for Clustering Data, Prentice Hall, 1988. Book available online at <http://www.cse.msu.edu/~jain/clustering_jain_dubes.pdf>.

[28] A. Kalogeratos, A. Likas, Document clustering using synthetic cluster prototypes, Data Knowl. Eng. 70 (2011) 284–306.

[29] M. Kantrowitz, B. Mohit, V. Mittal, Stemming and its effects on TFIDF ranking, in: Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 2000, pp. 357–359.

[30] T. Kanungo, D. Mount, N. Netanyahu, C. Piatko, R. Silverman, A. Wu, The analysis of a simple K-means clustering algorithm, in: Proceedings of the 16th ACM Symposium on Computational Geometry, 2000, pp. 100–109.

[31] I. Katsavounidis, C. Kuo, Z. Zhang, A new initialization technique for generalized Lloyd iteration, IEEE Signal Process. Lett. 1 (10) (1994) 144–146.

[32] L. Kaufman, L. Rousseeuw, Finding Groups in Data: An Introduction to Cluster Analysis, John Wiley and Sons, 1990.

[33] S. Kolliopoulos, S. Rao, A nearly linear-time approximation scheme for the Euclidean K-median problem, in: Proceedings of the 7th Annual European Symposium on Algorithms, 1999.

[34] Y. Linde, A. Buzo, R. Gray, An algorithm for vector quantizer design, IEEE Trans. Commun. 28 (1) (1980) 84–95.

[35] J. MacQueen, Some methods for classification and analysis of multivariate observations, in: Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability, vol. 1, 1967, pp. 281–296.

[36] K.V. Mardia, P. Jupp, Directional Statistics, 2nd ed., John Wiley and Sons Ltd., 2000.

[37] J.P. Mei, L. Chen, Proximity-based k-partitions clustering with ranking for document categorization and analysis, Expert Syst. Appl. 41 (2014) 7095–7105.

[38] J. Pena, J. Lozano, P. Larranaga, An empirical comparison of four initialization methods for the K-means algorithm, Pattern Recogn. Lett. 20 (1999) 1027–1040.

[39] M. Steinbach, G. Karypis, V. Kumar, A comparison of document clustering techniques, in: Proceedings of the 6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2000.

[40] A. Strehl, J. Ghosh, R. Mooney, Impact of similarity measures on web-page clustering, in: Proceedings of the AAAI Workshop on AI for Web Search, 2000, pp. 58–64.

[41] T. Su, J. Dy, A deterministic method for initializing K-means clustering, in: Proceedings of the 16th IEEE International Conference on Tools with Artificial Intelligence, 2004, pp. 784–786.

[42] A. Topchy, A. Jain, W. Punch, Combining Multiple Weak Clusterings, ICDM, 2003, pp. 331–338.

[43] G. Tzortzis, A. Likas, The MinMax k-Means clustering algorithm, Pattern Recogn. 47 (2014) 2505–2516.

[44] T. Velmurugan, Performance based analysis between k-Means and Fuzzy C-Means clustering algorithms for connection oriented telecommunication data, Appl. Soft Comput. 19 (2014) 134–146.
[45] J. Wu, H. Liu, H. Xiong, J. Cao, J. Chen, K-means-based consensus clustering: a unified view, IEEE Trans. Knowl. Data Eng. 27 (1) (2015) 155–169.
[46] I. Yoo, X. Hu, A Comprehensive comparison study of document clustering for a biomedical digital library MEDLINE, in: Proceedings of the 6th ACM/IEEE-CS joint conference on Digital libraries, 2006, pp. 220–229.
[47] F. Yuan, Z. Meng, H. Zhang, C, Dong, A new algorithm to get the initial centroids, in: Proceedings of the 3rd IEEE International Conference on Machine Learning and Cybernetics, 2004, pp. 1191–1193.
[48] S. Zeng, X. Tong, N. Sang, Study on multi-center fuzzy C-means algorithm based on transitive closure and spectral clustering, Appl. Soft Comput. 16 (2014) 89–101.
[49] S. Zhong, Efficient online spherical K-means clustering, in: Proceedings of the IEEE International Joint Conference on Neural Networks IJCNN, vol. 18, 2005, pp. 790–798.
[50] S. Zhong, J. Ghosh, A comparative study of generative models for document clustering, in: Proceedings of the workshop on Clustering High Dimensional Data and Its Applications in SIAM Data Mining Conference, 2003.
[51] S. Zhong, J. Ghosh, A comparative study of generative models for document clustering, in: proceedings of the Workshop on Clustering High Dimensional Data: Third SIAM Conference on Data Mining, 2003.
[52] http://ir.dcs.gla.ac.uk/resources/test_collections/ (last access: November 2014).
[53] http://people.csail.mit.edu/jrennie/20Newsgroups (last access: November 2014).