Which Distance Metric is Right: An Evolutionary K-Means View

Chuanren Liu* Tianming Hu† Yong Ge‡ Hui Xiong§

Abstract

It is well known that the distance metric plays an important role in the clustering process. Indeed, many clustering problems can be treated as an optimization problem of a criterion function defined over one distance metric. While many distance metrics have been developed, it is not clear that how these distance metrics can impact on the clustering/optimization process. To that end, in this paper, we study the impact of a set of popular cosine-based distance metrics on K-means clustering. Specifically, by revealing the common order-preserving property, we first show that K-means has exactly the same cluster assignment for these metrics during the E-step. Next, by both theoretical and empirical studies, we prove that the cluster centroid is a good approximator of their respective optimal centers in the M-step. As such, we identify a problem with K-means: it cannot differentiate these metrics. To explore the nature of these metrics, we propose an evolutionary K-means framework that integrates K-means and genetic algorithms. This framework not only enables inspection of arbitrary distance metrics, but also can be used to investigate different formulations of the optimization problem. Finally, this framework is used in extensive experiments on real-world data sets. The results validate our theoretical findings on the characteristics and interrelationships of these metrics. Most importantly, this paper furthers our understanding of the impact of the distance metrics on the optimization process of K-means.

Keywords: Distance Metric, K-means, Genetic Algorithm, Document Clustering

1 Introduction

Data clustering aims to find intrinsic structures in data, and organize them into meaningful subgroups for further study [10]. It is ill-posed if no prior information is provided about the well-defined underlying data distributions. Thus, instead of designing a general purpose clustering algorithm, it is suggested that we should always study clustering in its application context [9]. For instance, document clustering is often used to enable automated categorization, where its performance can be measured against a human-imposed classification into different topical categories.

Over the years, while there have been many clustering algorithms proposed, K-means (and its variants) is still one of the most competitive algorithms for document clustering [16, 11]. The immense popularity can be attributed to its simplicity, understandability and scalability. With reasonably good results, it is fast and easy to combine K-means with other methods in larger systems. Although introduced more than half a century ago [12], K-means is still widely used in various real-world applications and has been identified as one of the top 10 algorithms in data mining [7]. In fact, its shadow can even be felt in many seemingly irrelevant latest developments, such as von Mises-Fisher model-based clustering, bipartite graph-based clustering, information theoretic co-clustering, clustering ensembles, and semi-supervised clustering.

Mathematically, a formal approach to clustering is to consider it as an optimization problem. Given a particular distance function to measure dissimilarity between objects and a corresponding criterion function, K-means essentially optimizes the criterion function alternately over its two parameters: a set of cluster assignments and a set of cluster centers. Here, similar to [21], for a set of vectors, we define the composite vector to be their sum and the centroid vector to be the arithmetic mean. The cluster center is defined to be the exact solution to optimize the criterion function over that cluster. Indeed, K-means has two steps: E-step and M-step, as shown in Algorithm 1. It is clear to see that the clustering solution depends on the underlying distance measure, so it is crucial to check whether the specified measure suits the intrinsic data structure.

For instance, the Euclidean distance is perhaps the most widely used measure, on which the traditional K-means was built with sum of squared error as the criterion function. However, for high-dimensional document data, the Euclidean distance is less meaningful in such a spherical space than the cosine similarity, the one used in the spherical K-means [6]. In addition, other cosine-like measures have been investigated as well, such as extended Jaccard and Pearson correlation [18].

*Rutgers University. chuanren.liu@rutgers.edu
†Dongguan University of Technology. tmhu@ieee.org
‡Rutgers University. yongge@rutgers.edu
§Contact Author. Rutgers University. hxiong@rutgers.edu

Algorithm 1 Standard K-means algorithm.

Randomly select K instances as initial cluster centers.
repeat
E-step: Form K clusters by assigning each instance to the cluster with the closet center c_k.
M-step: Compute the centroid of each cluster as new cluster center.
until Centers do not change

Recently, the studies in geometric algorithms have revived interest in Bregman divergences, an old class of distance measures that subsumes the Euclidean distance. In particular, the traditional K-means extended with Bregman divergences has proved to be able to handle the spherical data [3]. Moreover, Kullback-Leibler divergence, a special form of Bregman divergences which measure the difference between two probability distributions, has been shown to provide quality results on large real-world document data sets.

In summary, previous work on distance measures focuses on developing new measures or evaluating existing ones with respect to different clustering methods. In this paper, we study distance measures from a new perspective: how they affect the clustering solutions (both intermediate and final) during the optimization process of K-means. The reason to choose K-means is twofold. On one hand, we want to isolate their effect on the optimization process from those of different criterion functions. Thus, we need to concentrate on a single criterion function, which mostly depends on the definition of distance metric to perform cluster assignment. On the other hand, since K-means is widely used in practice, there is a need for such work from the application perspective as well. Surprisingly enough, our initial studies show that K-means lacks the ability to differentiate a set of popular cosine-based distance metrics. However, in reality, these distance metrics may suit different data scenarios respectively. To leverage the efficient optimization procedure of K-means to explore the strengths of these metrics, we propose an evolutionary K-means approach that integrates K-means and genetic algorithms (GAs). This framework not only enables the differentiation of these metrics, but also addresses some issues with K-means. Specifically, we make the following contributions in this paper.

- First, by solving a constrained optimization problem, we prove that the normalized cluster centroid (normalized to unit length) is the optimal center to the underlying criterion function in the spherical K-means. Since the cluster center is only involved in the scale invariant computation of cosine in the criterion function, the length "constraint" is actually a convenience rather than a constraint for seeking optimal solutions. In contrast, previous work had no such length constraint on the center.
- We identify a class of distance metrics that are monotonic with cosine and thus are equivalent to one another in terms of ranking order. In other words, given a set of cluster centers, they would produce the same cluster assignment as cosine does. Moreover, by both theoretical and empirical studies, we show that the cluster centroid is a good approximator to optimize their respective criterion functions in K-means. The above two points together speak to the inability of K-means to distinguish between these cosine-monotonic metrics. That is, given a set of initial cluster centers, K-means will produce the same clustering solution with them. To the best of our knowledge, this is the first work to study K-means for its ability to differentiate distance metrics.
- Moreover, we reveal some interesting interrelationships among these cosine-based distance metrics in terms of magnitude. First, the relationships in their own magnitude turn out to make fail two frequently used measures that evaluate how much a distance metric fits a data set. Second, the relationships in their slope magnitude provide theoretical evidences of their impact on the convergence process of K-means. Hence, these findings can serve as guidelines for the development of both new metrics and the adaptive selection strategy of metrics, which not only enables better clustering solutions, but faster convergence as well.
- Finally, we introduce a framework for integrating K-means and GAs. This framework not only enables inspection of arbitrary distance metrics, but also can be used to investigate different formulations of the criterion functions for K-means.

2 The Optimization Problem

In this section, K-means is presented as an optimization problem, where we show the normalized cluster centroid is the optimal center in the spherical K-means.

2.1 The Maximal Similarity Criterion

Let us use the vector space model to represent documents. In detail, each document, x_n, which has a total of M terms, is considered as a vector in M-dimensional space, i.e., $x_n = (x_{n1}, \cdots, x_{nM})^T$. A collection of documents are denoted by $X = (x_1, \cdots, x_N)^T$. The vectors in the collection are weighted by the standard term frequency-inverse document frequency scheme (TF-IDF).

In this paper, unless specified otherwise, we will assume that the vector representation of document vectors has been normalized to unit length. Since we only focus on cosine based metrics, such a normalization does not lose generality.

In document clustering, the popular criterion function to maximize is

$$\sum_{n=1}^{N} \cos(x_n, c_{I_n}) = \sum_{k=1}^{K} \sum_{x \in C_k} \cos(c_k, x) = \sum_{k=1}^{K} \frac{c_k^T}{|c_k|} \sum_{x \in C_k} x,$$

where K is the number of clusters, C_k is the set of documents in cluster k, and $|\cdot|$ is the Euclidean norm. $I_n \in \{1, 2, \cdots, K\}$ denotes the cluster assignment for x_n. Also referred to as the vector-space variant of the K-means algorithm [21], during the optimization process, documents are assigned to the cluster with the closet center and then the centers are updated in the next iteration. It is not hard to show the solutions take the form $c_k = a_0 \sum_{x \in C_k} x$, where a_0 is an arbitrary non-zero constant. Therefore, although the cluster centroid is often used in practice, it really does not matter which one we compute for c_k here, composite, centroid, or their normalized versions.

2.2 The Minimal Distance Criterion

The standard K-means uses the terminology of distance instead of similarity. Thus, for document clustering, its goal is to minimize the sum of distances between the document vectors and the center of the cluster that they are assigned to:

$$J = \sum_{n=1}^{N} d(x_n, c_{I_n}) = \sum_{k=1}^{K} \sum_{x \in C_k} d(c_k, x), \tag{2.1}$$

where d is the distance metric. Note that, it is not required that d is a well-defined metric. Semimetrics, which drop the triangle inequality requirement, are used in many practical cases. In the traditional K-means, the Euclidean distance $d(\mu, \nu) = |\mu - \nu|^2$ is employed. Since the centroid is the solution, it can also be called the Euclidean center.

2.3 The Equivalence Relationship

By converting similarity to distance with

$$d(\mu, \nu) = 1 - \cos(\mu, \nu)$$

for K-means, we can see the relationship between document clustering and the spherical K-means. By restricting to unit centers, we will show that the M-step of K-means optimizes the criterion function using normalized centroids as new cluster centers, hence establishing the equivalence relationship between document clustering and the spherical K-means.

Theorem 2.1 *The normalized centroid is the optimal center for the spherical K-means.*

Proof Note that, since only vectors in the k-th cluster matter when analyzing c_k, optimizing J over cluster centers can be decomposed into K separate optimization problems to minimize:

$$J_k = \sum_{x \in C_k} d(c_k, x).$$

For unit vectors, the cosine similarity is equal to dot-product, i.e.,

$$d(\mu, \nu) = 1 - \cos(\mu, \nu) = 1 - \mu^T \nu = \frac{1}{2} |\mu - \nu|^2,$$

Then we have $J_k = \frac{1}{2} \sum_{x \in C_k} |c_k - x|^2$. Compared with clustering in the traditional Euclidean space, the main difference here is that we need to consider the constraint of $|c_k| = 1$. By introducing the Lagrangian multiplier λ, we obtain the following Lagrangian function to minimize:

$$L_k = J_k - \lambda(c_k^T c_k - 1).$$

By solving the KKT optimal condition $\nabla_{c_k} L_k = 0$, the only feasible solution is

$$c_k = -\frac{1}{2\lambda} \sum_{x \in C_k} x.$$

Given the unit length constraint on c_k, we have $\lambda = -\frac{|\sum_{x \in C_k} x|}{2}$, and $c_k = \frac{\sum_{x \in C_k} x}{|\sum_{x \in C_k} x|}$. □

Theorem 2.1 shows that the spherical K-means for document clustering performs exactly coordinate descent on J. The loop of K-means repeatedly minimizes J over cluster assignment I_n with c_k fixed, and then minimizes J over c_k with I_n fixed. Since there is a finite number of clusterings, J must monotonically decrease and converge.

In the rest of this paper, since the cluster center only appears in the cosine operation in the criterion functions, if there is a solution, then there is a corresponding normalized solution of unit length that achieves the same value for the criterion. Without loss of generality, hereafter we confine our discussions to unit centers and unit centroids.

3 Cosine-Monotone Distance Metrics

In this section, we investigate a set of popular distance metrics for document clustering that are monotonic with respect to cosine. Such order-preserving property makes them indistinguishable to K-means during the E-step. Furthermore, we show that the centroid is, both in principle and in practice, a good solution to their respective criterion functions.

This is to say that the centroid is often the only choice of cluster center to K-means during the M-step. The above two points reveal a problem of K-means: as far as clustering solutions are concerned, K-means cannot distinguish between these cosine monotone metrics.

3.1 Distance Metrics

In addition to the unit distance defined by

$$d(\mu, \nu) = 1 - \cos(\mu, \nu) = 1 - \mu^T \nu, \tag{3.2}$$

there are other ways to convert similarity to distance. For example, a more complex formulation commonly used in the clustering toolkits, such as WEKA, is named by the Laplacian distance:

$$d(\mu, \nu) = \frac{1 - \cos(\mu, \nu)}{1 + \cos(\mu, \nu)} = \frac{1 - \mu^T \nu}{1 + \mu^T \nu}. \tag{3.3}$$

According to the meaning of cosine, similarity is indicated by the angle between two vectors. Thus a more natural way is directly using the angle, i.e.,

$$d(\mu, \nu) = \frac{2}{\pi} \arccos(\cos(\mu, \nu)) = \frac{2}{\pi} \arccos(\mu^T \nu). \tag{3.4}$$

For continuous or discrete non-negative features, [17] extended the binary definition of Jaccard similarity as

$$\text{Jaccard}(\mu, \nu) = \frac{\cos(\mu, \nu)}{|\mu|/|\nu| + |\nu|/|\mu| - \cos(\mu, \nu)} \in [0, 1].$$

From the extended Jaccard similarity, the corresponding Jaccard distance can be defined as

$$d(\mu, \nu) = 1 - \text{Jaccard}(\mu, \nu) = \frac{2(1 - \mu^T \nu)}{2 - \mu^T \nu}. \tag{3.5}$$

[11] showed that the extended Jaccard is better than cosine-based similarity, hence we also include it in our comparison.

However, since all the distance metrics above are converted from the same cosine similarity and thus are monotonic decreasing functions of cosine, they are really equivalent to one another in terms of ranking. That is, for four vectors $\mu_1, \mu_2, \nu_1, \nu_2$ in the space, if $d_1(\mu_1, \nu_1) < d_1(\mu_2, \nu_2)$, then we have $d_2(\mu_1, \nu_1) < d_2(\mu_2, \nu_2)$. Due to this monotonicity/equivalence, fitting these distance metrics in naive K-means will produce exactly the same clustering solutions *if* the centroid is always computed for the center in the underlying criterion functions. In such a sense, we just showed that

Theorem 3.1 *Naive K-means lacks the ability to distinguish between these cosine-monotone metrics.*

At first glance, one may argue why the centroid is always computed for the center. If we solve these metrics' respective criterion functions for the exact optimal centers, the solutions would probably be different, which, in turn, would lead to different cluster assignment afterwards. However, as we will see later, our studies show that 1) in practice their solutions are usually not available in closed form, which makes their computation costly; and 2) the centroid is not only a good approximator in terms of the criterion function, but also a good competitor in terms of the quality (measured against "ground truth") of the resultant clustering.

3.2 Optimal Center VS Cluster Centroid

Let us first determine the Laplacian center, i.e., the optimal center for the Laplacian distance. As in the case of the unit distance, to minimize the criterion function for the k-th cluster, we minimize the corresponding Lagrangian function:

$$L_k = J_k - \lambda(c_k^T c_k - 1).$$

It gives

$$c_k = -\frac{1}{\lambda} \sum_{x \in C_k} \frac{x}{(1 + x^T c_k)^2}, \tag{3.6}$$

where the Lagrangian multiplier

$$\lambda = -| \sum_{x \in C_k} \frac{x}{(1 + x^T c_k)^2} |$$

ensures $|c_k| = 1$. This formulation naturally leads to Algorithm 2, where it is supposed that ideally Equation 3.6 holds at convergence. However, Algorithm 2 turned out not to guarantee such convergence, as oscillation between several different points has been observed commonly in our experiments.

Furthermore, compared to cluster centroids, seeking exact Laplacian centers defined in Equation 3.6 is supported by neither internal nor external validation measures. For instance, the upper half of Table 1 reports the internal Laplacian distance for each cluster (true class) of la1, a document data set used in our experiments. The second and third columns show the average distances between the instances of each cluster to the centroid and the Laplacian center, respectively. The last column lists the distances between them in each cluster. One can see that the centroid is not only a competitive solution to the Laplacian criterion, but also very close to the Laplacian center. Also, extensive experiments show that the Laplacian centers make little improvement in the final clustering results from those by cluster centroids. Thus it is hardly justified to investigate more sophisticated methods to address the issue of convergence or to seek exact Laplacian centers.

Algorithm 2 The algorithm for the Laplacian center.

Initialize c_k
repeat
$c_k \leftarrow \sum_{x \in C_k} \frac{x}{(1+x^T c_k)^2}$
$c_k \leftarrow \frac{c_k}{|c_k|}$
until Convergence

Table 1: A comparison between the centroid c_E and the Laplacian center c_L in terms of the Laplacian distance $d(\mu, \nu)$ and the angle $\theta(\mu, \nu)$ (in degrees) for the la1 data set.

cluster	$\text{avg}_x d(x, c_E)$	$\text{avg}_x d(x, c_L)$	$d(c_E, c_L)$
0	0.7009	0.7008	0.0002
1	0.6497	0.6494	0.0005
2	0.6379	0.6374	0.0007
3	0.6605	0.6599	0.0009
4	0.6336	0.6331	0.0008
5	0.6835	0.6824	0.0017
cluster	$\text{avg}_x \theta(x, c_E)$	$\text{avg}_x \theta(x, c_L)$	$\theta(c_E, c_L)$
0	79.7167	79.7221	1.6758
1	77.4537	77.4717	2.6156
2	76.8228	76.8490	3.1019
3	77.9077	77.9363	3.3700
4	76.7007	76.7305	3.2342
5	78.7804	78.8333	4.6602

Similarly, the optimal center of cluster C_k for the angle distance is

$$c_k = -\frac{1}{\pi\lambda} \sum_{x \in C_k} \frac{x}{\sqrt{1-(x^T c_k)^2}}. \tag{3.7}$$

The optimal center for the Jaccard distance is

$$c_k = -\frac{1}{\lambda} \sum_{x \in C_k} \frac{x}{(2-x^T c_k)^2}. \tag{3.8}$$

Here the Lagrangian multiplier λ ensures $|c_k| = 1$. Table 2 shows a summary of these results.

In addition to empirical evidences, we also offer a theoretical explanation why the centroid is a fine approximator of the optimal center for the metrics discussed above.

Theorem 3.2 *Under a mild assumption, the cluster centroid is a fine approximator of the cluster center for these cosine-monotone metrics.*

Proof Note that all of the formulations of the optimal center c for the above metrics take the form $c = \sum_x w(x)x$, where $w(x)$ is the weight for x defined in each center formulation. Grouping by $w(x)$, it can be rewritten as

$$c = \sum_w w \sum_{w(x)=w} x.$$

A closer look into $w(x)$ indicates that $w(x_1) = w(x_2)$ implies $x_1^T c = x_2^T c$ and in turn $\cos(x_1, c) = \cos(x_2, c)$.

Table 2: Different metrics and their optimal centers.

metric	$d(\mu, \nu)$	c_k
Unit	$1 - \mu^T \nu$	$-\frac{1}{2\lambda} \sum_{x \in C_k} x$
Laplacian	$\frac{1-\mu^T \nu}{1+\mu^T \nu}$	$-\frac{1}{\lambda} \sum_{x \in C_k} \frac{x}{(1+x^T c_k)^2}$
Angle	$\frac{2}{\pi} \arccos(\mu^T \nu)$	$-\frac{1}{\pi\lambda} \sum_{x \in C_k} \frac{x}{\sqrt{1-(x^T c_k)^2}}$
Jaccard	$\frac{2(1-\mu^T \nu)}{2-\mu^T \nu}$	$-\frac{1}{\lambda} \sum_{x \in C_k} \frac{x}{(2-x^T c_k)^2}$

This is to say that all such x with $w(x) = w$ are positioned at the same angle from c. For example, on the unit-hypersphere in 3-dimensional space, all such x are located on the contour-like circle with center c. If these x are distributed "symmetrically enough" about c, we can make a mild assumption that $\sum_{w(x)=w} x \sim a_0 c$, where a_0 is a constant. That is, the composite vector of these x lies roughly in the same direction as c. Summing up all groups of x gives

$$\sum_w \sum_{w(x)=w} x \sim a_1 c,$$

where a_1 is a constant. For instance, the lower half of Table 1 shows that, while the average angle between the document vector and its cluster center (both centroid and the Laplacian center) is as large as about 80 degrees for the data set la1, the angle between the centroid and the Laplacian center is less than 5 degrees. Therefore, the above assumption of approximation generally holds in practice.

On the other hand,

$$\sum_w \sum_{w(x)=w} x = \sum_x x = Nm,$$

where N is the total data size and m is the unnormalized centroid. Therefore, we have $c \sim \frac{m}{|m|}$. □

3.3 The Magnitude Relationship

In addition to the impact on the optimization process of K-means, it is also important to examine these metrics' behavior during the validation of clustering solutions. Like K-means' criterion, many internal validation measures use a distance metric to evaluate intra-cluster similarity and inter-cluster dissimilarity, thus sharing the similar preference of clustering with K-means' criterion. Particularly, since we expect K-means, when equipped with the right metric, to favor true-class-like solutions, the same goes for the validation measures. In fact, as we will see later, the values of certain measures over the class structure can even be regarded as the degree of ease/usefulness for K-means to find true-class-like solutions. Therefore, to see which metric is most useful to K-means, we can compare them by how much they fit the class structure in terms of their validation values with respect to these measures.

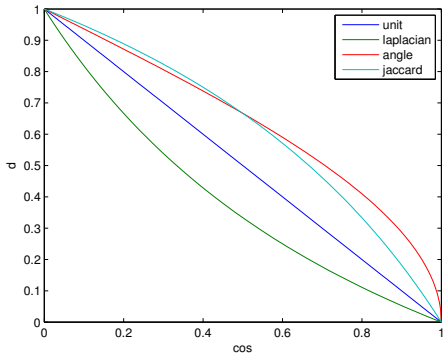

Figure 1: Plots of different metrics.

Surprisingly enough, although it is often assumed that different metrics may suit different types of data, these metrics' own relationship in magnitude, $d_{Laplacian} \leq d_{Unit} \leq d_{Angle} \leq d_{Jaccard}$ (demonstrated in Figure 1 and proved in Theorem 3.3), turns out to completely predetermine the preferences of two commonly used measures, regardless of which data set they are working on. In other words, the magnitude relationship between the metrics appear to make them two fail in validation.

Theorem 3.3 *For unit vectors μ and ν, we have $d_{Laplacian}(\mu,\nu) \leq d_{Unit}(\mu,\nu) \leq d_{Angle}(\mu,\nu)$, where equality holds when $s = \mu^T\nu = 0$ or 1. If $0 < s < \frac{1}{2}$, then $d_{Angle}(\mu,\nu) < d_{Jaccard}(\mu,\nu)$.*

Proof It is trivial to show that

$$d_{Laplacian} = \frac{d_{Unit}}{1+s} \leq d_{Unit}.$$

With the fact $\arcsin(s) < \frac{\pi}{2}s$ for $0 < s < 1$, we have

$$\begin{aligned} d_{Unit} &= 1 - s \\ &< 1 - \frac{2}{\pi}\arcsin(s) \\ &= \frac{2}{\pi}(\frac{\pi}{2} - \arcsin(s)) = \frac{2}{\pi}\arccos(s) = d_{Angle}. \end{aligned}$$

With the fact $\arcsin(s) > \frac{\pi}{2}\frac{s}{2-s}$ for $0 < s < \frac{1}{2}$, we also have

$$\begin{aligned} d_{Angle} &= 1 - \frac{2}{\pi}\arcsin(s) \\ &< 1 - \frac{s}{2-s} = 2\frac{1-s}{2-s} = d_{Jaccard}. \end{aligned}$$

The equality conditions can be verified in a straightforward manner. □

Adopted in clustering software CLUTO [11], one measure computes for each cluster the ratio of average internal similarity (average similarity between the instances in the same cluster) over average external similarity (average similarity between the instances in one cluster and the rest of the instances outside the cluster). In our case of distance, we can compute this ratio as $\frac{1-IDis}{1-EDis}$, where $IDis$ denotes the corresponding notion of average internal distance and $EDis$ denotes the average external distance. The higher this ratio is for a cluster, the more compact and isolated the cluster will be, which, in turn, makes it easier for K-means to recognize the cluster. Thus, we can use this ratio to see which distance metric fits the class structure best. However, as shown in Table 3, the order discovered by Theorem 3.3 completely agrees with the ratio ranking of the metrics on each class of data set la1. This is not coincidence, as such agreement is observed in all of the data sets in our experiments. It is also observed that the condition $0 < s < \frac{1}{2}$ in Theorem 3.3 holds generally in practice.

Table 3: The ratio of internal similarity over external similarity on data set la1.

cluster	Unit	Laplacian	Angle	Jaccard
0	1.73	1.66	1.77	1.82
1	2.76	2.57	2.83	2.97
2	3.49	3.22	3.56	3.74
3	2.82	2.53	2.95	3.17
4	2.96	2.70	3.07	3.25
5	1.90	1.72	2.02	2.15

Table 4: The performance lift of different metrics.

Data	Unit	Laplacian	Angle	Jaccard
fbis	1.640	1.447	1.684	1.832
la1	1.347	1.275	1.355	1.402
la2	1.359	1.283	1.367	1.417
re0	1.371	1.256	1.398	1.489
re1	1.632	1.438	1.676	1.827
wap	1.477	1.346	1.500	1.598

Another measure, performance lift [18], computes the ratio of the criterion function (in terms of similarity) of clustering solution over that of random clustering. In our case, by defining performance as the criterion of K-means, the lift is computed as $(N - J_t)/(N - J_r)$, where J_t and J_r are the criterion function values of true class structure and random clustering, respectively. Since this ratio actually represents the difference between class structure and random clustering to K-means, the higher the ratio, the easier it will be for K-means to find true-class-like solutions. Thus, we can use this lift to see which distance metric fits the class structure best. However, Table 4 shows that Theorem 3.3 determines the lift ranking of different metrics again.

To investigate the underlying reasons, we examine the distance values that constitute the numerator x and the denominator y of the ratio. It turns out that, due to the distribution of distance values of these document data sets, the difference $x - y$ is very small. In fact, compared to x and y, we can even assume that the difference is relatively fixed among the metrics. For instance, Table 5 gives the detailed values in performance lift for data set la1 with $N = 3204$.

Table 5: The criterion values used in performance lift for data set la1.

metric	Unit	Laplacian	Angle	Jaccard
J_t	2542	2128	2778	2828
J_r	2713	2360	2890	2936
$J_r - J_t$	170	232	111	107

One can see that compared to the numerator and the denominator of performance lift, $J_r - J_t$ is much smaller and does not vary much among the metrics. With such relationship between x and y, it is not hard to prove that the ratio becomes smaller with larger numerator and denominator. That is, for two ratios $\frac{x_1}{y_1}$ and $\frac{x_2}{y_2}$ with $x_1 > x_2$ and $y_1 > y_2$, if

$$x_1 - y_1 = x_1 - y_1 \sim e > 0,$$

then we have

$$\frac{x_1}{y_1} = 1 + \frac{e}{y_1} < 1 + \frac{e}{y_2} = \frac{x_2}{y_2}.$$

Intuitively, with $x - y$ fixed, $\frac{x}{y}$ would monotonically decrease and converge to 1 as x and y increase. As for the four metrics, the Laplacian distance is the smallest from Figure 1, which, in turn, leads to the smallest J_t and J_r. Since both the numerator and the denominator of $(N - J_t)/(N - J_r)$ are largest at this time, the performance lift of the Laplacian distance becomes the smallest among the four metrics.

In summary, to compare distance metrics in terms of true class structure, we should be careful with the choice of validation measures involving distance computation. Note that, in later experiments, we only use validation measures without distance computation.

4 Evolutionary K-means

To leverage the efficient optimization of K-means to explore those metrics, in this section, we present the GAK-means algorithm, a framework that combines GAs and K-means. This framework can be used to address not only the issue in Theorem 3.1, but also other major drawbacks of K-means.

4.1 A Strategy Overview

In short, GAs are randomized search and optimization techniques guided by the principles of evolution and natural genetics [13]. They are efficient, adaptive and robust search processes, performing multi-dimensional search in order to provide near optimal solutions of an evolution function in an optimization problem. Since clustering may be viewed as searching for a number of clusters such that the given criterion is optimized, GAs seem a natural choice.

The first problem in designing a GA-based K-means algorithm is how to encode solutions with string representation. Usually there are two choices, label-based encoding and center-based encoding. In the former, each string is an integer representing the cluster label of an object. In the latter, each string is a sequence of real numbers representing one of the K cluster centers. In our case of document clustering, as demonstrated in Table 7, the number of features M far exceeds the number of documents N. Hence, the N-dimensional search space of label-based encoding is much smaller than the KM-dimensional search space of center-based encoding. Based on these observations, GAK-means employs the label-based encoding scheme, where each chromosome represents a clustering solution. This scheme makes it more efficient to search for the solutions to optimize the criterion functions of various distance metrics.

The overview of GAK-means is provided in Algorithm 3. Roughly speaking, in each generation, the principles of K-means are first utilized for updating cluster centers and cluster assignments. Next, various operators of GAs are used to perturb the individual chromosomes to keep K-means from getting stuck at local optima. While such an integration addresses the problem of local optima with K-means, it inherits the efficiency of K-means by utilizing the problem specific knowledge. Most importantly, it can help us investigate the differences between the distance metrics.

Algorithm 3 GAK-means

```
Initialize Population
repeat
    for i = 1 → #Population do
        Decode the i-th chromosome
        Run one loop of naive K-means: compute new cluster centroids and then perform new cluster assignments
        Replace the old chromosomes by encoding new clustering solutions
        Calculate the criterion function as chromosome fitness
    end for
    Perform selection, crossover and mutation
until Termination condition attained
```

4.2 The Detailed Setting

Below we discuss the detailed setting of GAK-means used in the experiments. As for genetic operators, three are employed in this study.

Selection. The "selection" operator selects the fittest chromosomes to form the mating pool, thus mimicking the "survival of the fittest" concept of nature.

The probability of selecting each chromosome is proportional to the fitness value. Here, the fitness is defined as J^{-1}, where J is the clustering criterion function to be minimized. Thus, better clustering will be selected with a higher chance.

Crossover. The "crossover" operator exchanges information between two parent chromosomes and generates two children for the next generation. We use single-point crossover with a fixed probability q. For a pair of chromosomes, a random integer, i.e., the crossover point, is generated in the range from 1 to N. The portions of the pair lying to the right of the crossover point are exchanged to produce two offspring. In the present study q is taken to be 0.8.

Mutation. The "mutation" operator makes an occasional random alteration of a gene, thus introducing some extra variability into the population. Every gene in each chromosome has equal chance to undergo mutation. Although it is usually performed with very low probability q, it has an important role in the generation process. In this study q is chosen to be 0.25.

Besides the genetic operators, there are other issues that need to be taken into account.

Population Initialization. The population is initialized by randomly choosing $P = 10$ chromosomes. Each chromosome consists of N genes, representing a solution of cluster labels with N integer numbers between 1 and K.

Chromosome Validity. In the processes of initialization and some genetic operations, invalid chromosomes, theoretically, may be produced. At this time, we will repeat the specified step with a limited number of attempts. In the experiments, however, only a tiny number of invalid clustering solutions are observed in the process of crossover.

Convergence Guarantee. To guarantee the convergence of GAK-means, the elitist strategy is used, which aims to carry the best string from the previous iteration into the next. At the beginning of each iteration, we will keep a copy of the best clustering solution. If no better result is achieved at the end of this iteration, the kept result will replace the worst chromosome in the population.

Stopping Criterion. Usually, two stopping criteria are used in GAs. We can execute the process for a fixed number of iterations and the best solution obtained is taken to be the optimal one. For the other, the process is terminated if no further improvement in the fitness value of the best string is observed for a fixed number of iterations, and the best solution is taken to be the optimal one. In this study, we fix the number of iterations to 100.

Table 6: The optimization process on la1.

Iteration	unit	Laplacian	angle	Jaccard
1	2652.2844	2276.1666	2850.2987	2898.3847
2	2598.6688	2203.2961	2815.2190	2864.4407
3	2555.8096	2146.9709	2787.0330	2836.6687
4	2532.0347	2112.4579	2771.6221	2822.3008
5	2527.7898	2106.3936	2768.8720	2819.7240
6	2524.5607	2101.7390	2766.7815	2817.7743
7	2523.3578	2099.8371	2766.0146	2817.1023
8	2522.9968	2099.2909	2765.7822	2816.8915
9	2522.7260	2098.8492	2765.6112	2816.7461
10	2522.6356	2098.6896	2765.5551	2816.7018
11	2522.6237	2098.6722	2765.5474	2816.6949
12	2522.6159	2098.6620	2765.5425	2816.6903
13	2522.5950	2098.6372	2765.5288	2816.6766
14	2522.5649	2098.5953	2765.5095	2816.6586
15	2522.5476	2098.5668	2765.4987	2816.6496

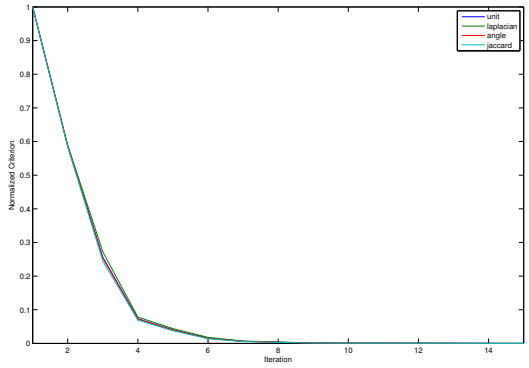

Figure 2: The optimization process on la1.

4.3 Discriminability Issues

At the end of this section, we show the effectiveness of GAK-means to distinguish between different distance metrics by Theorem 4.1.

Theorem 4.1 *GAK-means has the ability to distinguish between different distance metrics.*

Proof The proof is straightforward. Suppose I^1 and I^2 are two different clustering solutions (labels) by GAK-means for the same data set X, with two distance metrics d_1 and d_2 used in the fitness function, respectively. Now we can calculate the clustering criterion function for I^i using d_j, $J_j^i = \sum_{n=1}^{N} d_j(x_n, c_{I_n^i})$. All we need is to demonstrate there exist some I^1 and I^2 such that $J_1^1 < J_1^2$ but $J_2^1 > J_2^2$. As we will see later in the experiments, there are plenty of such examples. □

Based on the rationale above, one may argue that naive K-means (with unit distance) also has such discriminability. For example, we can record all the intermediate solutions from one run of K-means, where other distance metrics are likely to have different preferences. However, we have proven that the centroid is a fine approximator of their respective optimal centers, hence these distance metrics will probably share the same preference as unit distance does.

Table 7: The characteristics of data sets.

data	fbis	la1	la2	re0	re1	wap
#doc	2463	3204	3075	1504	1657	1560
#term	20000	6188	6060	2886	3758	8460
#class	17	6	6	13	25	20
MinClass	38	273	248	11	10	5
MaxClass	506	943	905	608	371	341

Indeed, as demonstrated in Table 6, all other criterion functions decrease along with the unit distance optimization. Even more, as shown in Figure 2, when normalized to $[0, 1]$ with $\frac{x-\min}{\max-\min}$, all of the criterion functions decrease at almost the same pace. These evidences prove again that naive K-means cannot distinguish between those monotonic metrics by any means. More importantly, instead of actively guiding K-means, those metrics play a role of passive watcher in that they cannot affect the optimization process.

5 Experimental Evaluation

In this section, we present an experimental evaluation of GAK-means. First we introduce the experimental data sets and cluster evaluation criteria. Then we analyze the impact of various metrics on both the clustering process and the clustering solutions.

5.1 Experimental Data Sets

For evaluation, we used six real data sets from different domains, all of which are available at the website of CLUTO [11]. Some characteristics of these data sets are shown in Table 7. One can see diverse characteristics in terms of size, number of clusters and cluster balance are covered by the investigated data sets.

5.2 Validation Measures

Since the true class labels of our data sets are available, we can measure the quality of the clustering solutions using external criteria that measure the discrepancy between the structure defined by a clustering and what is defined by the class labels. First we compute the confusion matrix C with entry C_{ij} as the number of documents from true class j that are assigned to cluster i. Then we calculate the following four external measures: normalized mutual information (NMI), conditional entropy (CE), purity and F-measure [19].

NMI and CE are entropy based measures. The cluster label can be regarded as a random variable with the probability interpreted as the fraction of data in that cluster. With T and C denoting the random variables corresponding to the true class and the cluster label, respectively, the two measures are defined as $NMI = \frac{H(T)+H(C)-H(T,C)}{\sqrt{H(T)H(C)}}$, $CE = H(T, C) - H(C)$, where $H(X)$ denotes the entropy of X. While NMI measures the shared information between T and C, CE tells the information remained in T after knowing C.

Purity, also called classification rate, computes the fraction of correctly classified data when all data in each cluster is classified as the majority class in that cluster. In contrast, F-measure still differentiates the remaining minority classes in each cluster by combining the precision and recall concepts from information retrieval. In detail, each cluster is treated as if it were the result of a query and each class as if it were the desired set of documents for a query. The recall and precision of that cluster for each given class can be computed as follows: $R_{ij} = C_{ij}/C_{+j}$, $P_{ij} = C_{ij}/C_{i+}$, where C_{+j}/C_{i+} is the sum of jth column/i-th row, i.e., j-th class size /i-th cluster size. The F-measure of cluster i and class j is then given by $F_{ij} = 2R_{ij}P_{ij}/(P_{ij} + R_{ij})$. Finally, the overall value for the F-measure is defined as a weighted average for each class, i.e., $F = \sum_j C_{+j} max_i\{F_{ij}\}/n$, where n is the total sum of all elements of matrix C. F-measure reaches its maximal value of 1 when the clustering is the same as the true classification.

5.3 Clustering Evaluation

In our experiments, GAK-means is run 10 times with the number of clusters set to the number of true classes. To investigate the difference among the distance metrics, we record every metric's criterion values and clustering solutions during the evolutionary process. The average results over the 10 runs are listed in Table 8, where the best results are highlighted in bold according to each validation measure (all measures prefer large values except CE). Apparently, Table 8 verifies the two points we made earlier. First, as validated by the four measures, GAK-means is really able to differentiate the metrics by generating their respective clustering solutions. Second, as illustrated with all four measures on the la1 data set, some metric, *angle* in this case, suits the data set better than other metrics.

In addition to simple comparison of averages, to account for the randomness in GAK-means, we also perform a statistical significance test, where *angle* was compared with the other metrics one by one for a paired t-test. In detail, for each metric, we collect the top 5 chromosomes from each run of GAK-means and hence generate a test sample of 50 top chromosomes. The t-test results in Table 9 proves again that *angle* is indeed the most suitable metric for GAK-means to cluster the la1 data set, especially in terms of CE.

Finally, using confusion matrix representation, we present two exemplar clustering solutions to validate the claim in Theorem 4.1. Ideally, in a quality clustering, every cluster should be as pure as possible. This means its corresponding row in the confusion matrix should contain as many zeros as possible. Moreover, every true class should also be as concentrated as possible.

Table 8: A comparison of clustering results.

data	distance	purity	F	NMI	CE
fbis	Unit	**0.6983**	0.5823	0.5966	1.3379
	Laplacian	0.6977	**0.5865**	**0.5971**	**1.3335**
	Angle	0.6977	0.5801	0.5949	1.3482
	Jaccard	0.6975	0.5800	0.5946	1.3492
la1	Unit	0.7827	0.7197	0.5623	1.0354
	Laplacian	0.7822	0.7198	0.5616	1.0370
	Angle	0.7829	0.7202	0.5624	1.0352
	Jaccard	0.7825	0.7196	0.5618	1.0367
la2	Unit	0.7695	0.7095	0.5510	1.0586
	Laplacian	0.7713	0.7118	0.5527	1.0539
	Angle	0.7691	0.7089	0.5505	1.0598
	Jaccard	0.7689	0.7087	0.5502	1.0606
re0	Unit	0.6689	**0.4807**	**0.4116**	1.3757
	Laplacian	**0.6888**	0.4733	0.4109	**1.3713**
	Angle	0.6695	0.4790	0.4067	1.3913
	Jaccard	0.6695	0.4790	0.4071	1.3901
re1	Unit	0.6654	0.4599	0.5507	1.4991
	Laplacian	0.6753	0.4672	0.5561	1.4731
	Angle	0.6717	0.4663	0.5507	1.5013
	Jaccard	0.6711	0.4670	0.5498	1.5056
wap	Unit	0.7093	0.6047	0.6071	1.3580
	Laplacian	0.7080	0.6022	0.6052	1.3628
	Angle	0.7099	0.6069	0.6081	1.3567
	Jaccard	0.7080	0.6040	0.6076	1.3578

Table 9: The comparison between *angle* and the other three metrics via t-tests. "**" means *angle* is better with significance level 0.05 and "*" means significance level 0.1.

-	unit	Laplacian	Jaccard
purity	*	*	*
F	*	*	**
NMI	*	**	**
CE	**	**	**

This means its corresponding column in the confusion matrix should contain as few nonzero values as possible. Table 10 plots the confusion matrices of two clustering solutions for the la1 data set, where the metrics' liking and disliking are marked with '+' and '-', respectively. One can see that although the Unit and Laplacian distances prefer the second solution, which is better according to the true class labels, the first solution is chosen by the Angle and Jaccard distances.

5.4 The Impact of Distance Metrics

To further our understanding of the nature of these metrics, in addition to final results, it is important to investigate their impact on the intermediate solutions during the evolutionary process. To that end, we show the evolutionary process on the la1 data set in Figure 3, where some interesting observations can be made about their behaviors.

First, as another showcase of Theorem 4.1, it is common that there is significant disagreement between the metrics over the "right" solutions at early iterations of GAs. Nevertheless, all four validation measures seem to unanimously support Laplacian's choice of chromosomes. This phenomenon can find its explanation in Theorem 3.3. In the beginning, the criterion function has not yet been fully optimized. Thus, the distance is still large between most of the instances and their cluster centroids. In other words, dot product s in Theorem 3.3 is small for most pairs involved in the distance computation. In this situation, as shown in the top left corner of Figure 1, the Laplacian distance goes down fastest with the largest slope magnitude $\frac{|d(s_1)-d(s_2)|}{|s_1-s_2|}$. In such a narrow range of small s, any perturbation in different directions by GAs will receive significantly different feedback from Laplacian. Hence, it is easier for GAs to find better solutions with the Laplacian criterion. Intuitively, we can say that the metric with larger slope magnitude can impose more penalty for the clustering error. Since Laplacian penalizes the clustering error greater than other metrics, especially when most of the chromosomes have not been optimized, GAs can benefit most from its guidance.

Table 10: The confusion matrix of la1 results.

Unit: - Laplacian: - Angle: + Jaccard: +						
cluster	0	1	2	3	4	5
0	7	0	13	28	444	2
1	1	327	9	69	9	28
2	254	12	110	39	23	4
3	1	1	4	8	1	656
4	74	3	72	370	7	11
5	4	11	65	429	71	37
Unit: + Laplacian: + Angle: - Jaccard: -						
cluster	0	1	2	3	4	5
0	8	0	16	24	463	0
1	1	330	9	71	8	26
2	259	13	107	44	20	4
3	0	1	4	7	1	658
4	66	1	69	342	7	11
5	7	9	68	455	56	39

Second, although the disagreement between the metrics reduces along the evolutionary process, it does not disappear at the end. On one hand, the decrease in disagreement can be explained by Theorem 3.2 again with illustrating examples in Table 6 and Figure 2. On the other hand, as demonstrated in Table 8, considerable disagreement persists till the end. While it may not look significant in Figure 3, the difference in their preferences is really statistically significant in Table 9.

6 Related Work

There are roughly three categories of work that are related to the main theme of this paper. In the following, we briefly review each of them.

6.1 GA-based Clustering

As introduced earlier, since GAs are general-purpose optimization techniques by randomized search, any clustering problem with a criterion function is open to them. In particular, GA-based K-means can be found in [14, 1, 2], where both label-based and center-based encoding approaches have been implemented.

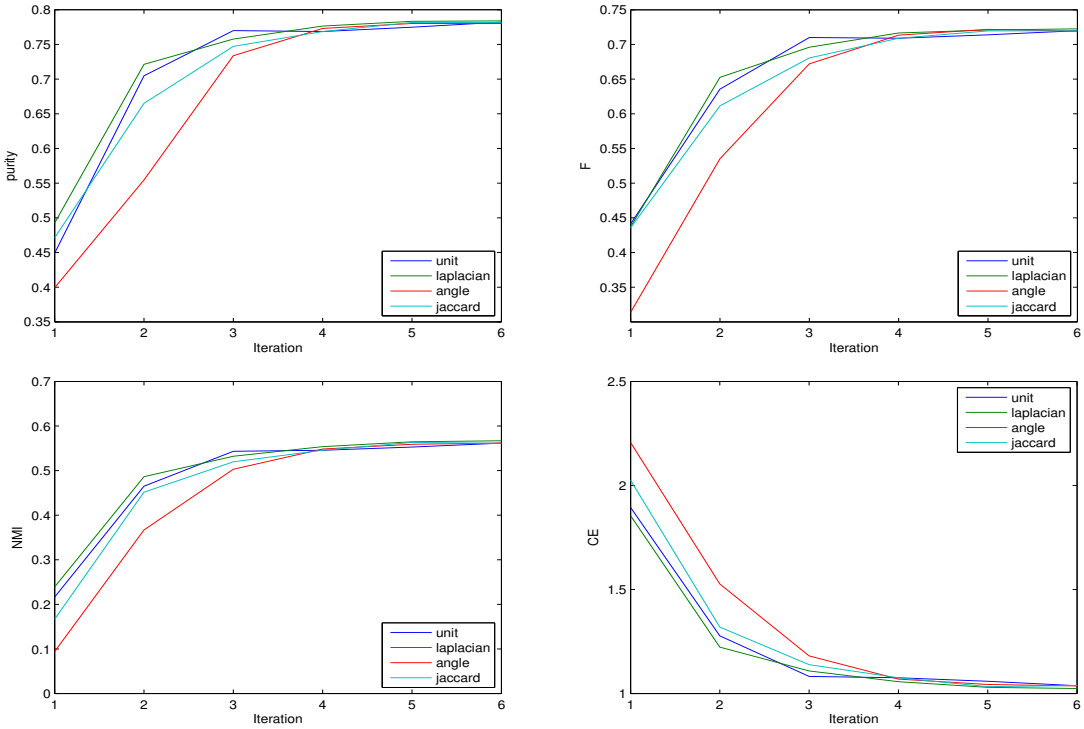

Figure 3: The validation measure curve for la1 during the GAK-means evolution.

While all of them computed the Euclidean distance in low dimensional spaces, Bandyopadhyay et al. [2] proposed to use the point symmetry distance, which is able to detect any shape of clusters with the characteristic of symmetry. As for document clustering, an interesting Minimum Spanning Tree (MST)-based encoding was reported in [4]. The edges of the MST are represented with a vector of binary elements, where a value of 1/0 means that the edge is eliminated/retained in the solution clustering.

However, due to various considerations, the data sets used in the above studies are very small. While [14, 1] only used data with dimensionality less than 5, [4] considered the data sets that are the output of a query, each less than 100 in size. In contrast, we used much larger data sets with high ratio of feature size over data size. More importantly, besides seeking better clustering solutions with GAs, our goal is to use GAs to explore the difference between the distance metrics.

6.2 Distance Metrics for Documents

Most works of single term analysis adopt the vector space model and hence focus on cosine-like measures, including Jaccard, Laplacian, Pearson coefficient, etc [16, 18, 21, 8]. Usually they would incorporate these measures into different types of clustering methods, such as partitional, hierarchical and graph-theoretic, and try to explain why certain combinations provide best results according to data characteristics, method specifics and validation measure biases.

Another class of study formulates more complex criterion functions based on certain properties of clustering. For instance, starting with the vector-space variant of K-means, CLUTO [11, 21] investigates a handful of different criterion functions for partitional clustering, which optimize various aspects of intra-cluster similarity and inter-cluster dissimilarity. Since these sophisticated criterion functions no longer lend themselves to the K-means style optimization, the greedy strategy is often employed as the choice of the optimizer.

In principle, although the metrics examined in this paper can be embedded into the criterion functions or the clustering processes of some of the works above, they help little to investigate the nature of the underlying distance metrics, for it is difficult to isolate the effect of metrics from others on the clustering process. Compared to these studies, we concentrated on a particular set of cosine-monotonic distance metrics that look "alike" to K-means. We performed both theoretical and empirical analyses regarding their impact on the optimization process of K-means.

6.3 Distance/Order-Preserving Metrics

Also related are Multidimensional Scaling (MDS) [5] and other studies on distance/order-preserving metrics. Particularly, metric MDS techniques take as input a matrix of dissimilarities for a data set and try to output a representation of the data set in d-dimensional space with a distance function.

The goal of metric MDS techniques is to minimize the difference between the derived distances in the d-dimensional space and the given dissimilarities in the matrix. Instead of preserving the exact dissimilarities input, the non-metric MDS seeks to maintain the rank order of the dissimilarities. Related are also techniques that learn a distance metric from absolute, qualitative feedback [20], or relative comparison [15].

Our work differs because the input is a set of distance metrics that all preserve the distance order of cosine. Instead of seeking a low dimensional projection or another distance metric, our goal is to differentiate these metrics with the clustering solutions using K-means style optimization.

7 Concluding Remarks

In this paper, we identified a set of popular cosine-monotone metrics which are equivalent to one another in terms of ranking order. Since K-means variants are a class of competitive clustering methods and different distance metrics may suit different types of data sets, it is a natural practice to substitute these cosine-based metrics for the unit distance in K-means for better clustering solutions. However, we showed that such a direct replacement did not work out for several reasons. First, due to their order-preserving property, K-means does exactly the same cluster assignment during the E-step. Second, by both theoretical and empirical studies, we showed that the cluster centroid is a good approximator of their respective optimal centers in the M-step. In other words, K-means itself is not able to differentially use these metrics. When searching for the above reasons, we also identified some interesting relationships between these metrics in terms of magnitude. Such relationships provide insight into the metrics' impact on the convergence process of K-means. Also, they shed light on the potential new metrics and adaptive use of existing ones, which enable better clustering solutions with faster convergence. Finally, to explore the potential strengths of these metrics, we developed an evolutionary K-means framework, which integrates K-means and genetic algorithms. This framework not only enables inspection and understanding of arbitrary distance metrics, but also can be used to investigate different formulations of the optimization problems for clustering.

In addition to clustering, the distance metrics studied in this paper are widely used in other high dimensional domains as well. Therefore, the results of this paper are likely to have an impact on the particular choice of the distance metrics and the way the metric is incorporated into the corresponding methods, which often arise from problems such as clustering, outlier detection, and similarity search.

Acknowledgement

This research was partially supported by grants from National Science Foundation (NSF) via grant numbers CCF-1018151 and IIP-1069258. Also, it was supported in part by Natural Science Foundation of China (61100136, 70890082, 71028002).

References

[1] S. Bandyopadhyay and U. Maulik. An evolutionary technique based on k-means algorithm for optimal clustering in R^N. *Information Sciences*, 146(1-4):221–237, 2002.

[2] S. Bandyopadhyay and S. Saha. GAPS: A clustering method using a newpoint symmetry-based distance measure. *Pattern Recognition*, 40:3430–3451, 2007.

[3] A. Banerjee, S. Merugu, I. S. Dhillon, and J. Ghosh. Clustering with bregman divergences. *J. Machine Learning Research*, 6:1705–1749, 2005.

[4] A. Casillas, M. González de Lena, and R. Martínez. Document clustering into an unknown number of clusters using a genetic algorithm. In *Text, Speech and Dialogue*, pages 43–49, 2003.

[5] T. Cox and M. Cox. Multidimensional scaling. *Chapman&Hall, London, UK*, 2004.

[6] I. S. Dhillon and D. S. Modha. Concept decompositions for large sparse text data using clustering. *Machine Learning*, 42(1):143–175, 2001.

[7] X. Wu et al. Top 10 algorithms in data mining. *Knowledge and Information Systems*, 14(1):1–37, 2008.

[8] Y. Ge et al. Multi-focal Learning and Its Application to Customer Service Support. In *KDD'09*.

[9] I. Guyon, U. Von Luxburg, and R. C. Williamson. Clustering: Science or art? In *NIPS'09 Workshop on Clustering Theory*.

[10] A. K. Jain. Data clustering: 50 years beyond k-means. *Pattern Recognition Letters*, 31:651–666, 2010.

[11] G. Karypis. CLUTO - Software for Data Clustering. http://glaros.dtc.umn.edu/gkhome/views/cluto.

[12] J. MacQueen. Some methods for classification and analysis of multivariate observations. In *Proc. 5th Berkeley Symp. Mathematical Statistics and Probability*, 1967.

[13] Z. Michalewicz. *Genetic Algorithms + Data Structures = Evolution Programs*. Springer-Verlag, 1992.

[14] C.A. Murthy and N. Chowdhury. In search of optimal clusters using genetic algorithms. *Pattern Recognition Letters*, 17(8):825–832, 1996.

[15] M. Schultz and T. Joachims. Learning a distance metric from relative comparisons. In *NIPS'03*.

[16] M. Steinbach, G. Karypis, and V. Kumar. A comparison of document clustering techniques. In *KDD'00 Workshop on Text Mining*.

[17] A. Strehl and J. Ghosh. Value based customer grouping from large retail data sets. In *Proc. 2000 SPIE Conf. Data Mining and Knowledge Discovery*.

[18] A. Strehl, J. Ghosh, and R. Mooney. Impact of similarity measures on web-page clustering. In *AAAI'00 Workshop of Artificial Intelligence for Web Search*.

[19] J. Wu, H. Xiong, and J. Chen. Adapting the right measures for k-means clustering. In *KDD'09*.

[20] E.P. Xing, A.Y. Ng, M.I. Jordan, and S. Russell. Distance metric learning, with application to clustering with side-information. In *NIPS'02*.

[21] Y. Zhao and G. Karypis. Empirical and theoretical comparisons of selected criterion functions for document clustering. *Machine Learning*, 55(3):311–331, 2004.