A Coherent and Heterogeneous Approach to Clustering

Arian Maleki
Department of Electrical Engineering
Stanford University
arianm@stanford.edu

Nima Asgharbeigi
Department of Electrical Engineering
Stanford University
nimaa@stanford.edu

***Abstract*—Despite outstanding successes of the state-of-the-art clustering algorithms, many of them still suffer from shortcomings. Mainly, these algorithms do not capture *coherency* and *homogeneity* of clusters simultaneously. We show that some of the best performing spectral as well as hierarchical clustering algorithms can lead to incorrect clustering when the data is comprised of clusters with different densities or includes outliers. We introduce algorithms based on variants of geodesic distance that capture both coherency and homogeneity of clusters. Such choice of distance measure empowers simple clustering algorithms such as K-medoids to outperform spectral and hierarchical clustering algorithms. To show the theoretical merits of our approach, we present theoretical analysis of a simplified version of the algorithm. We also provide remarkable experimental evidence on the performance of our algorithms on a number of challenging clustering problems.**

I. Introduction

Clustering of data points is a task of high significance for many data analysis applications in various fields—from computer science and statistics [1] to biology and psychology [2], [3]. Early methods of data clustering suffered from harsh assumptions on the density of each cluster and other limitations. For example, EM algorithm which is based on generative models assumes each cluster to be normally distributed. Model-free algorithms such as K-means usually require the clusters to be linearly separable. These early algorithms were also extremely prone to outliers in the data and many local minima in their objective functions.

Extensive research in this area has led to modern clustering algorithms with remarkable performance. Spectral clustering (see [4],[5], [6] and [7]) is one of the most popular classes of such algorithms which relies on a weighted similarity graph. These methods then attempt to cut the similarity graph into a specified number of clusters with the least weighted cuts possible, while maintaining the "size" of the clusters balanced. Intuitively, the similarity graph captures the local connectivity structure of the data points and the the algorithm tries to cut through the least similar regions of the graph to form the clusters.

Another promising branch of work has focused on hierarchical clustering methods. This class of algorithms require a measure of similarity between disjoint subsets of data points to construct a hierarchy of clusters. At each level, each cluster is created by merging its children clusters in the hierarchy. The most popular of these algorithms take a bottom-up (or agglomerative) approach to construct this hierarchy. They begin at the bottom of the hierarchy with every data point as a singleton cluster and, at each step of the process, merge the two most similar clusters into a single cluster at the next level of the hierarchy. Depending on the specific similarity measure in use, there are various agglomerative clustering algorithms: single linkage, complete linkage, and average linkage to name a few [1].

Despite impressive successes of these modern approaches on a wide range of clustering problems, there are aspects of the algorithms that limit their performance on more challenging data sets. For instance, as we show in Section IV, normalized spectral clustering [5] can lead to incorrect clustering (for all values of σ) when the clusters vary significantly in size, density, and proximity to each other. This is mainly due to the fully connected similarity graph with Gaussian similarity weights in which one constant σ governs all similarities throughout the entire data set and thus it cannot adjust for variations of density in different clusters. A similar limitation arises when using ε-neighborhood graphs in which case one constant ε globally affects the structure of the similarity graph. Spectral clustering methods that employ k-nearest neighbor similarity graphs are less vulnerable to this effect. However, as we demonstrate empirically, they still suffer from the fact that the objective function attempts to balance the size (or volume) of different clusters in the data set. Although not a valid assumption for all clustering problems, having the clusters evenly balanced in size (or volume) is a common assumption among all spectral clustering methods. On the other hand, hierarchical clustering algorithms are usually very prone to error in the presence of misleading outliers.

In this paper we propose using a different class of distance measures called *geodesic distance* and introduce two clustering algorithms based on it. We claim that geodesic distance is much more capable of capturing the *coherency* of each cluster than Euclidean distance. By coherency we mean connectedness and togetherness of the points in a cluster which is one qualitative of a good cluster. Furthermore, by choosing the weights of the geodesic distance in a specific Gaussian form, our algorithms are very resistant to the *heterogeneity* of different clusters. Heterogeneity is the characteristic of a data set that includes clusters with completely different densities. Combining geodesic distance with the simple K-medoids algorithm results in a powerful algorithm that can handle heterogeneous clustering problems and capture coherency of the clusters simultaneously. We also introduce a variation on the normalized spectral clustering algorithm of Ng et al. [5] based on geodesic distance which enables the algorithm to perform very well on heterogeneous data sets in which the original algorithm fails. Despite this promising improvement, our geodesic spectral clustering algorithm still suffers from sensitivity to significant difference in cluster sizes. This, as we mentioned above, is a limitation common to nearly all spectral clustering methods.

Huo [8] has also attempted to utilize the geometric structures and proximity [8] to improve clustering. The presumption in his algorithm is that the data points are close to smooth manifolds and therefore the points are projected on the manifold by a denoising step. Then the new data points are clustered by an agglomerative clustering algorithm. This approach suffers from most of the problems of agglomerative algorithms such as dealing with outliers and is useful when the points of each cluster are actually close to a low dimensional manifold which is very restrictive. Our algorithm on the other hand uses a better approach for utilizing geoemtries and does not have any presumption about closeness to a low dimensional manifold. We should also mention that our approach is inspired by some of the ideas in manifold learning [9], but there are some major differences which will be noticed in the paper.

The organization of the paper is as follows. In the next section section we define a class of geodesic distances and then introduce our two algorithms built on top of it. In Section III we provide some theoretical results on a simplified version of our geodesic K-medoids algorithm. Section IV presents our experimental results comparing our geodesic clustering algorithms to a few competitive spectral and hierarchical clustering methods on a number of challenging clustering problems.

II. Algorithms

Consider a finite number of data points $X = \{x_1, x_2, \ldots, x_n\}$ in $\mathbb{R}^l$, and suppose the number of clusters k_c is given. Here we define a class of geodesic distances and use it as a foundation for the two algorithms that follow. In this paper, $\|\cdot\|$ always denotes Euclidean ℓ_2 norm.

A. Geodesic Distance

We first define two types of neighborhoods and their corresponding geodesic distances.[1]

Definition 1: The *ε-neighborhood* of a point $x \in X$ is defined as $N_\varepsilon(x) = \{y \in X : \|x - y\| \le \varepsilon\}$.

[1]Readers familiar with manifold theory [10] may notice that we are using the term "geodesic distance" for the distance approximated from data. This is to be consistent with the terminology of [9]. In this paper we refer to the real geodesic distance on a manifold as "continuous geodesic distance."

Definition 2: The *k-neighborhood* of a point $x \in X$ is defined as the set of k closest points to x in the ℓ_2 norm sense and is indicated by $N_k(x)$.

Definition 3: The *k-neighborhood graph* of X is an undirected graph $G = (V, E)$ with $V = X$ and $(x_i, x_j) \in E$ iff $x_i \in N_k(x_j)$ or $x_j \in N_k(x_i)$. Similarly, the *ε-neighborhood graph G* is defined with $N_\varepsilon(x)$.

Definition 4: Consider the k-neighborhood graph $G = (V, E)$ and suppose a matrix $W \in \mathbb{R}^{n\times n}$ of non-negative weights is assigned to the edges of G. The *k-geodesic distance* from any point x to any point y is defined as the total weight of the shortest weighted path from x to y on graph G, and is denoted by $d^k_{G,W}(x, y)$. Similarly, the *ε-geodesic distance* $d^\varepsilon_{G,W}(x, y)$ is defined over the undirected ε-neighborhood graph.

In [9] the weights W are taken as the Euclidean distances between the neighboring points. As will be seen shortly, we let W depend on the local density in addition to Euclidean distances. For the sake of simplicity, let $d_{G,W}$ represent either $d^k_{G,W}$ or $d^\varepsilon_{G,W}$, and similarly for $N(x)$. It is straightforward to show the following.

Theorem 2.1: If W is symmetric, then $d_{G,W}$ defines a distance on X.

Finally, notice that the geodesic distance can be efficiently calculated by applying Dijekstra's shortest path algorithm on graph G and weights W.

B. Geodesic K-medoids Clustering

1) Construct the (ε- or k-) neighborhood graph $G = (V, E)$ over the set of data points X.
2) Calculate
$$R_{ij} = \min\{\max_{y\in N_k(x_i)} \|x_i - y\|, \max_{y\in N_k(x_j)} \|x_j - y\|\}$$
for all $(x_i, x_j) \in E$, i.e., the lesser of the distance from x_i to its farthest neighbor and the distance from x_j to its farthest neighbor[2].
3) Set the weights $W_{ij} = \exp(R^l_{ij}/2\sigma^2)\|x_i - x_j\|$ for all $(x_i, x_j) \in E$.
4) Form the matrix of geodesic distances $D \in \mathbb{R}^{n\times n}$ by $D_{ij} = d_{G,W}(x_i, x_j)$.
5) Apply K-medoids to the distance matrix D to find k_c clusters.

In the above algorithm σ is a parameter that controls the importance of density in the clustering process. For example in the ideal case where no outliers exist in the data points large values of σ work well. As we will demonstrate in the simulations the performance of our algorithm is not very sensitive on the value of σ and it finds correct clusters for a range of values of σ.

The K-medoids algorithm is very similar to K-means except that, instead of using the means (centroids) of the clusters, it represents each cluster by its medoid point, i.e., the data point that is assigned to that cluster and has the minimum

[2]Even in the case of ε-neighborhood graph G, k-neighborhoods are used for the calculation of R_{ij}.

sum of distances to the rest of the points in the same cluster. Therefore, K-medoids itself suffers from drawbacks similar to those of K-means. For instance, it is unable to handle non-convex clusters as we demonstrate in Section IV. However, as we show in the theoretical analysis of Section III, geodesic distance empowers the algorithm to capture the coherence of each cluster even in the lack of convexity. K-medoids is also known to be more robust in the presence of outliers [1].

Here, the quantity R^l_{ij} is a measure of density around the data points x_i and x_j. The scaling applied in the weights W_{ij} enables the algorithm to adjust for the variations in local density and thus handle heterogeneous mixtures of clusters. The geodesic distance derived from the neighborhood graph G and this specific choice of weights W allows two data points to have long range connectivity if there exists a path between them that passes through regions of enough density relative to the density of surrounding regions. This observation highlights the interaction between coherency and heterogeneity from the perspective of the geodesic K-medoids algorithm. It is also worth noting that since the large geodesic distance between two points mean that they are in different clusters this helps us to use smart initialization to make the algorithm converge faster and in less iterations. As an example of such initialization we propose an initialization algorithm here. Choose the first medoid at random from the data points. Find p (usually around 5 percent of the cardinality of the data) points that have the maximum distance from the first chosen medoid and put them in set M. Select one of the points in M at random. For choosing th rth medoid find p points that have the maximum sum of distances from the previously chosen medoids and put them in M. Select one of the points in M at random. This initialization will help the algorithm converge to the correct clusters more quickly.

C. Geodesic Spectral Clustering

1) Construct the neighborhood graph $G = (V, E)$ over the set of points X.
2) Set the weights $W_{ij} = \|x_i - x_j\|$ for every $(x_i, x_j) \in E$.
3) Form the matrix of geodesic distances $D \in \mathbb{R}^{n\times n}$ by $D_{ij} = d_{G,W}(x_i, x_j)$.
4) Define the affinity matrix $A \in \mathbb{R}^{n\times n}$ by $A_{ij} = \exp(-D^2_{ij}/2\sigma^2)$ if $i \neq j$, and $A_{ii} = 0$.[3]
5) Construct the matrix $L = T^{-1/2}AT^{-1/2}$, in which T is a diagonal matrix whose T_{ii} element is the sum of A's i-th row.
6) Form the matrix $V = [v_1 v_2 \dots v_K] \in \mathbb{R}^{n\times k_c}$ of the k_c largest eigenvectors of L (choose them as orthogonal to each other for repeated eigenvalues).
7) Normalize the rows of V to have unit length by $Y_{ij} = V_{ij}(\sum_r V^2_{ir})^{-1/2}$.
8) Consider each row of Y as a point in $\mathbb{R}^{k_c}$ and apply K-means to cluster them into k_c clusters. At the end, assign each point x_i to the cluster that the i-th row of Y is assigned to.

This algorithm is a variation on the normalized spectral clustering algorithm due to Ng et al. [5]. Here we use geodesic distances in place of Euclidean distance in the original algorithm. As we show empirically in Section IV, the geodesic spectral algorithm widens the class of clustering problems handled by Ng et al.'s algorithm, outperforming the original method in presence of heterogeneity in the data set. Furthermore, our experiments reveal that the geodesic spectral clustering algorithm is far less sensitive to the scaling parameter σ.

On one hand, this success is due to the fact that geodesic distance is based on the neighborhood graph which is not fully connected and thus can adjust for local structure in the data. As we mention in Section IV, the normalized spectral clustering method that uses a k-nearest neighbors similarity graph instead of a fully connected one is also capable of dealing with heterogeneity of clusters. On the other hand, geodesic distance facilitates long range connectivity of points that belong to a coherent body of data points, and in so doing, naturally captures coherency of clusters. We should also mention that here the weights W are not scaled since the Gaussian similarity function at step 4 provides enough scaling.

III. Theoretical Analysis

This section will be devoted to the theoretical analysis of our algorithm in the asymptotic regime where the number of data points is very large. In our theoretical analysis we ignore the outliers. Since there are no outliers in the data set the density terms are not important any more and for the sake of simplicity of notations we will remove the density terms in our analysis. In the first subsection we just consider the case where data points lie on a manifold and the goal is to find some points that are representative of all the points on the manifold. Although the word representative seems vague at this point, it will be clear at the end of this section. It is clear that in most of the clustering problems the data points are not on a manifold. We will prove in Section III-B that this condition is not necessary for geodesic K-medoids and it works well in more general settings. We will define clusters and clustering in a more rigorous way and will prove that our algorithm can find the clusters correctly even if they are not on a manifold.

A. Vector Quantization Point of View

Let M be an m dimensional manifold in an l dimensional ambient space $\mathbb{R}^l$. Consider a C^1-smooth curve $\gamma(t) : [0, 1] \to M$. The length of this curve is denoted by $L(\gamma)$ and is equal to

$$L(\gamma) = \int_0^1 \|\dot{\gamma}(t)\| dt.$$

[3] Some may take a value of 1 for self-similarities A_{ii}, but we chose $A_{ii} = 0$ to be consistent with [5]. Furthermore, we didn't notice any significant difference in our experimental results by using either one.

Definition 5: The *continuous geodesic distance* [11] between any two points on a manifold M is defined as

$$d_{CG}(x,y) = \inf_{\gamma:[0,1]\to M, s.t. \gamma(0)=x, \gamma(1)=y} L(\gamma). \tag{1}$$

It is straightforward to prove the following theorem.

Theorem 3.1: Let the data points X lie on a Manifold M and assume that M is isometric to a subset of m-dimensional parameter space. Then the continuous version of Geodesic K-medoids (which uses d_{CG} instead of $d_{G,W}$) works similar to the K-Medoid in the parameter space, i.e. if the initialization points of K-medoids in the parameter space are the inverse image of the initialization points on the manifold, then the resulting set of clusters are the same.

Proof: K-medoids is just based on distances and not the coordinates of the points. Since the distances are preserved in the isometric mapping, the two algorithms will converge to the same set of clusters. ■

Tenenbaum et al. [9] have proved that under certain conditions on the distribution of the data, as the number of points increases the discrete geodesic distance converges to the continuous geodesic distance. There are two important conditions in their assumptions. The first one is that the inverse image of the data should lie on a connected and convex subset of the parameter space and the second one is that the density of the data on this set should be greater than zero. These two assumptions are not true in most of the clustering problems. But these results are interesting in a different perspective. They show that if the data lies on a manifold the geodesic K-medoids algorithm can give a few points that are representative of the data on the manifold. This is reminiscent of the goal of the Self Organizing Map (SOM) algorithm [12]. But since the parameters of a manifold are the actual representation of the data on the manifold our algorithm seems to be more natural to find the representatives of a manifold.

B. Clustering Point of View

In all the proofs in this section we assume that there are no outliers in our data set and the data set is composed of k_c clusters. Since there are no outliers considering the density term will not have any benefit for the algorithm. Therefore we will ignore that term in our analysis. Cluster is defined as a compact and connected subset of $\mathbb{R}^l$. Clusters should also satisfy the condition that for any two points in a cluster, there exists a finite length path between them in that cluster.

Definition 6: The *minimum inter-cluster distance* is defined as

$$d_c = \min_{i<j} \inf_{x\in C_i, y\in C_j} \|x-y\|. \tag{2}$$

Lemma 3.2: Consider a cluster C and assume that for every point $x \in C$, there exists a point $m \in C \cap X$ such that $\|x-m\| \leq \delta \leq \varepsilon/3$. If the geodesic distance is calculated by ε neighborhood rule, then for any two points in C the length of the discrete geodesic path is finite.

Proof: For any two points in cluster C, let ℓ denote the length of the shortest path between them in C. Divide the path into $n = \lceil \ell/\delta \rceil$ points, such that the length of the path between any two consecutive points is less than or equal to δ. Call these point $y_1, y_2, \ldots, y_n$. According to hypothesis, for each y_i there exist an m_i such that, $\|y_i - m_i\| \leq \delta$. We also know, $\|y_i - y_{i+1}\| \leq \delta$. The distance between m_i and m_{i+1} is less than ε by triangle inequality and therefore the graph will have an edge between these two points. Thus the discrete geodesic distance can be bounded above by:

$$\begin{aligned}
&d^{\varepsilon}_{G,W}(x,y) \leq \sum_{i=1}^{n-1} \|m_{i+1} - m_i\| = \\
&\sum_{i=1}^{n-1} \|m_{i+1} - y_{i+1} + y_{i+1} - y_i + y_i - m_i\| \leq \\
&\sum_{i=1}^{n-1} \|m_{i+1} - y_{i+1}\| + \|y_{i+1} - y_i\| + \|y_i - m_i\| \leq 3\ell
\end{aligned} \tag{3}$$

Since according to the definition of the cluster, ℓ is bounded, $d^{\varepsilon}_{G,W}(x,y)$ will also be bounded. ■

Theorem 3.3: Assume that the data is partitioned to k_c clusters and in each cluster the assumptions of the previous lemma hold. Also assume that $d_C > \beta > \varepsilon$. Under these hypotheses the geodesic K-medoids with ε neighborhood geodesic calculation can separate the clusters correctly and a local minimum with finite distortion would be global optimum.

Proof: Since the geodesic distance between points from different clusters are infinite (there is no path between two different clusters) and the rest are finite(according to the previous lemma), the clustering algorithm can be successful. In other words, if it is trapped in a local minimum with finite error, that will give the correct clusters. ■

Theorem 3.4: Let one of the clusters be C and the volume of this cluster be represented by V_C. Assume that the points are distributed in this cluster according to probability density function f. If $\inf_{x\in V_C} f(x) > 0$, then the conditions of Lemma 3.2 hold with probability one as $n \to \infty$.

Proof: Consider a point x in the cluster and let $\inf_{x\in V_C} f(x) = \beta$. Also let $B(x,\delta)$ represent a ball of radius δ which is centered at x and finally let A be the event: "$B(x,\delta)$ has no points in it". We have,

$$P(A) \leq (1 - \beta Vol(B(x,\delta)))^n \underset{n\to\infty}{\to} 0, \tag{4}$$

where n is the total number of points and Vol represents the volume (Lebesgue measure in $\mathbb{R}^l$) of a set.

■

By combining these theorems together one can reach to the conclusion that as the number of points in all the clusters increases, the probability of error of Geodesic K-medoids algorithm goes to zero, if we ignore the local minima of K-medoids algorithm.

Since the analysis of the k-neighborhood case is more elaborate we do not state it here and only briefly describe the intuition behind the proof. The proof that we explain here just considers uniform distribution in each cluster. It is straightforward to extend the results to the case of piecewise constant densities on each cluster. In a less straightforward way

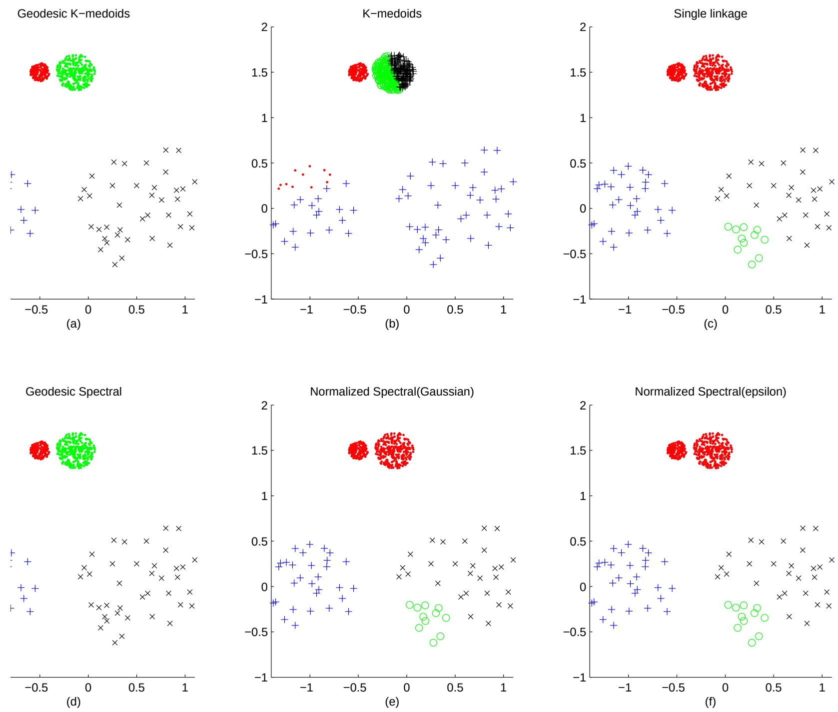

Fig. 1. Comparing different clustering algorithms on a heterogeneous data set. (a) geodesic K-medoids algorithm with k-geodesic distance, k=6, and $\sigma = 0.15$, (b) K-medoids, (c) Single linkage, (d) geodesic spectral algorithm with K-geodesic distance and k=6. Value of σ can range from 0.1 to 50, (e) Normalized spectral clustering of Ng et al. with Gaussian similarity matrix, $\sigma = 0.03$, (f) Normalized spectral clustering with ε-neighborhood similarity matrix and $\varepsilon = 0.25$.

piecewise constant results can be extended to general distributions. But since we just want to emphasize the guidelines we ignore general distributions and explain the proof for uniform distributions. Consider cluster C and let $B(x, r) \subset C$ be a ball of radius r with center $x \in C$. The average number of points that exist in this ball would be $k/n \sim \alpha r^l / V_C$. Therefore if we take $k/nV_C = \gamma^l$, the average radius would be equal to γ. As n goes to infinity the probability that ε is different from its mean decays exponentially and so this problem becomes exactly the same as ε-neighborhood problem. So, we can use the theorems we had for ε-neighborhood in order to prove the same theorems for the k-nearest neighbors.

IV. Experiments

In the previous section we showed theoretically that if the clusters are well separated our algorithm is able to detect the clusters in the asymptotic regime where the number of points goes to infinity. The goal of this section is to investigate the performance of our algorithm on a number of data sets in an experimental framework where the number of points is not very large and the presence of outliers prevents clusters from being completely separated. We also compare the performance of our algorithm with a number of other algorithms. These include K-medoids, single linkage clustering, complete linkage clustering, and the normalized spectral clustering of Ng et al. [5] with different types of similarity graphs, namely ε-neighborhood, k-nearest neighbors, and fully connected with Gaussian similarity function [6].

We observed that the performance of our algorithm is usually very robust to its parameters and the results are nearly the same for a range of values of k and σ. In order to find the best value of σ or ε for the variants of the normalized spectral clustering algorithms, we search over a range of values for the optimum value that minimizes the cost function. In our experiments we considered both Ncut [6] and K-means clustering distortion as the objective function. Finally after choosing the best value of the parameter by these automated methods, we run the algorithm on a number of values of the tuning constant and compare the performance visually with the automated method and choose the best. In order to avoid local minima, we run 100 iterations of each algorithm for each value of their tuning parameter and choose the best automatically.

Figure 1 shows the best performance of different algorithms on a data set of four clusters. The problem of K-medoids(or K-means) is not surprising and is a well known phenomenon. If the diameter of one cluster is larger than half of the distance between the medoids (or centroids) of the two clusters, K-medoids cannot distinguish the clusters correctly. Single linkage clustering cannot perform well here either, since it uses Euclidean distances to connect the points and this will lead to serious mistakes when the density of the points in different clusters varies significantly. In our simulations, two of the clusters have very high density and are located close to

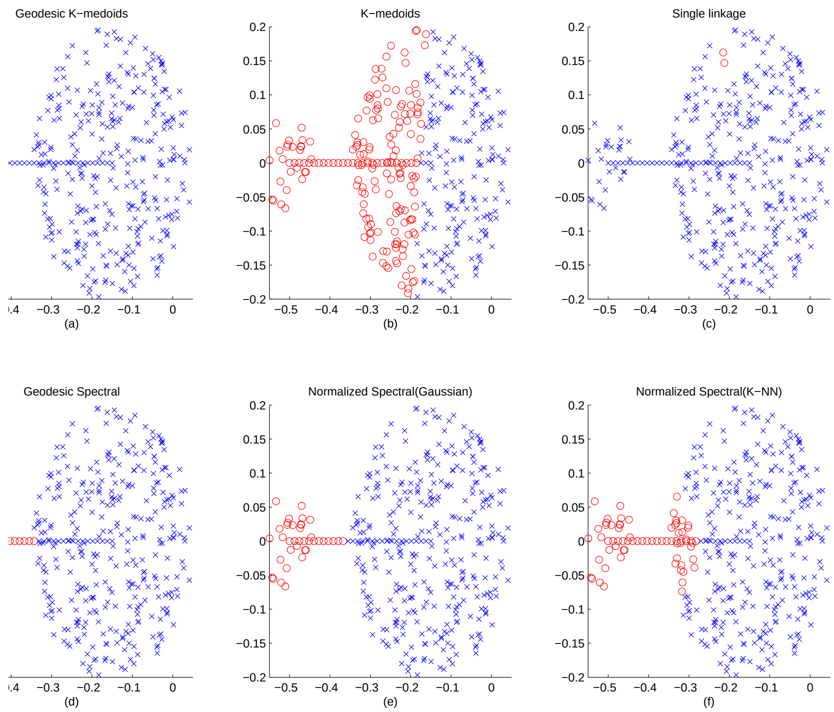

Fig. 2. Comparing different clustering algorithms when clusters differ considerably in size. (a) geodesic K-medoids algorithm with k-geodesic distance, k=10, and $\sigma = 0.01$, (b) K-medoids, (c) Single linkage, (d) geodesic spectral clustering with k-geodesic distance, $k = 10$, and $\sigma = 0.05$, (e) normalized spectral clustering of Ng et al. Gaussian similarity matrix, $\sigma = 0.025$, (f) normalized spectral clustering with k-nearest neighbor similarity matrix, $k = 10$, and $\sigma = 0.03$.

each other while the other two have much lower densities. The same problem happens for the spectral clustering algorithm. In the case of Gaussian similarity matrix, spectral clustering requires larger values of σ in order to find the low density clusters correctly. On the other hand, large values of σ prevents the algorithm from distinguishing between the clusters that are close to each other. The same problem exist for the spectral clustering with ε-neighborhood similarity matrix (Figure 1(e) and (f)).

Throughout our experiments we observed that our algorithms are not very sensitive to the value of parameters. For example, by changing k from 5 to 15 the resulting clusters do not change in (a). Also, in the cases where the clusters are well separated as in this data set, the value of σ does not play much role. In Figure 1(d) again the value of σ is not very important. We also noticed that spectral clustering with K-Nearest Neighbor performs very well in this case (not shown in the figure). This is due to the fact that for certain values of k, the similarities between the points from different clusters vanish. In the next experiment, however, we show the situation where K-Nearest Neighbor does not perform so well.

Now we turn our attention to a different situation in which the clusters vary significantly in size and are not very well separated. Figure 2 presents the results for different clustering algorithms on such a data set. Again, K-medoids and Single linkage suffer from their usual limitations in dealing with outliers. Spectral clustering algorithms show a different type of error in these figures. Spectral clustering algorithms try to balance the volume of different clusters. Therefore, when this assumption is not valid about two clusters in close proximity, spectral clustering tends to assign some of the points in the larger cluster to the smaller one. This phenomenon is observed to different degrees in all of the clusterings produced by spectral algorithms in Figure 2.

Finally, in order to demonstrate the robustness of our geodesic clustering algorithms in dealing with outliers as well as non-convex clusters, we performed experiments on the noisy bull's eye data set. The simulation results are presented in Figure 3. Our geodesic K-medoids algorithm, as well as geodesic spectral clustering and Ng et al.'s algorithm show impressive performance in this case.

V. Concluding Remarks

The ideal clustering algorithm should be capable of handling a heterogeneous mixture of coherently structured clusters while being robust to outliers in the data set. Empirically, we demonstrated that some of the most admired clustering methods fail to exhibit one or more of these desired traits. Toward a better class of clustering algorithms, we proposed a class of geodesic distances as the basis. Using two specific instances in this class of geodesic distances we introduced the geodesic K-medoids and the geodesic spectral clustering algorithms and discussed how they capture coherency each cluster while adjusting for heterogeneity of different clusters. The

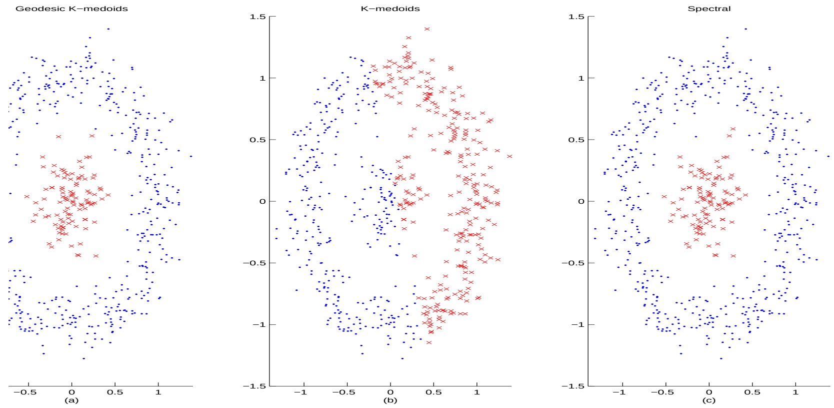

Fig. 3. Comparing different clustering algorithms on the noisy bull's eye data set. a) geodesic K-medoids algorithm with k-geodesic distance, k=6, and $\sigma = 0.15$, (b) K-medoids, (c) geodesic spectral clustering with $\sigma = 0.25$, and Ng et al.'s algorithm with $\sigma = 0.05$.

superior performance of these two algorithms on a number of difficult clustering problems is an encouraging indication that our approach has significant potential toward building better performing algorithms. The theoretical analysis of the geodesic K-medoids algorithm also indicates that in the asymptotic regime where the number of data points is very large the algorithm will find clusters with very high probability.

REFERENCES

[1] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of the Statistical Learning*. Springer, 2001.

[2] T. Sorlie, C. M. Perou, R. Tibshirani, T. Aas, and S. Geisler, "Gene expression pattern of breast carcinomas distinguish tumor subclass with clinical implications," *Proceedings of National Academy of Sciense*, vol. 98, pp. 10869–10874, 2001.

[3] P. Tamayo, D. Slonim, J. Mesirov, Q. Zhu, and S. Kitareewan, "Interpreting patterns of gene expression with self-organizing maps: Methods and application to hematopoietic differentiation," *Proceedings of National Academy of Sciense*, vol. 96, pp. 2907–2912, 1999.

[4] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, pp. 888–905, 2000.

[5] A. Ng, M. Jordan, and Y. Weiss, "On spectral clustering analysis and an algorithm," *Proceedings of the Neural Information Processing Systems Conference*, pp. 849–856, 2002.

[6] U. V. Luxburg, "A tutorial on spectral clustering," *Technical Report*, pp. 1–30, 2006.

[7] J. Malik, S. Belongie, T. Leung, and J. Shi, "Contour and texture analysis for image segmentation," *Perceptual organization for artificial vision systems*, 2000.

[8] X. Hue, "A local smoothing and geodesic distance based clustering algorithm for high dimensional noisy data; utilizing embedded geometric structures," *Workshop on Clustering High Dimensional Data and its Applications, in conjunction with SIAM International Conference on Data Mining*, 2003.

[9] J. B. Tenenbaum, V. D. Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 290, pp. 2319–2323, 2000.

[10] M. Spivak, "Calculus on manifolds: A modern approach to classical theorems of advanced calculus," *HarperCollins Publishers*, 1965.

[11] D. Donoho and C. Grimes, "Image manifolds which are isometric to euclidean spaces," *Journal of Mathematical Imaging and Vision*, vol. 23, pp. 5–24, 2005.

[12] T. Kohonen, "The self organizing map," *Proceedings of the IEEE*, vol. 78, pp. 1464–1479, 1990.