

# Geodesic K-means Clustering

Nima Asgharbeygi, Arian Maleki

Department of Electrical Engineering, Stanford University

nimaa@stanford.edu, arianm@stanford.edu

## Abstract

*We introduce a class of geodesic distances and extend the K-means clustering algorithm to employ this distance metric. Empirically, we demonstrate that our geodesic K-means algorithm exhibits several desirable characteristics missing in the classical K-means. These include adjusting to varying densities of clusters, high levels of resistance to outliers, and handling clusters that are not linearly separable. Furthermore our comparative experiments show that geodesic K-means comes very close to competing with state-of-the-art algorithms such as spectral and hierarchical clustering.*

## 1 Introduction

One of the most well-known and widely used algorithms for clustering is K-means [2]. Typically K-means is used with Euclidean distance in which case centroids become component-wise mean of cluster points. This leads to a very low computational complexity and makes K-means a suitable candidate to try on very large datasets. However, K-means often fails when clusters are not linearly separable or when the data is cluttered with outliers. There are many extensions of K-means attempting to overcome these issues, most of which can be found in [11].

A central element in almost all clustering methods is the distance (dissimilarity) measure. The desirable distance measure should capture the connectivity of the points in each cluster, while separating different clusters in presence of outliers in between. In this paper we take a step toward this ideal by proposing a new distance metric for clustering purposes. We introduce the class of *geodesic distances* which rely on a sparse neighborhood graph over the data points together with local density weights on its edges. We claim that the shortest weighted paths on such neighborhood graphs provide the desirable properties mentioned above.

To prove our point, we pair geodesic distance with the simplest clustering algorithm, K-means, and show

that the resulting algorithm is competitive with state-of-the-art approaches such as spectral [9, 7] and hierarchical [3] clustering. Toward that end, we first formulate a general-distance extension of K-means in Section 3.1. Then in Section 3.2 we derive *geodesic K-means* as an efficient instance of general-distance K-means that uses geodesic distance. Unlike classical K-means, geodesic K-means can robustly handle datasets with nonlinearly separable clusters and outliers and does not require a Euclidean representation of data points.

Independently, two recent lines of work have explored related approaches. Feil and Abonyi [1] apply geodesic distances to a fuzzy variant of K-means, while Kim et al. [5] make a similar attempt with fuzzy K-medoids. However, their notions of geodesic distance do not incorporate local density information and thus are susceptible to outliers.

## 2 Geodesic Distance

Consider a finite number of data points  $X = \{x_1, x_2, \dots, x_n\}$  in  $\mathbb{R}^q$ , and suppose the number of clusters  $K$  is given. The following definitions carry through with a general metric space  $(\mathcal{M}, d)$ , but here we use the more familiar notation of  $(\mathbb{R}^q, \|\cdot\|)$ .

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

**Definition.** The  $k$ -neighborhood of a point  $x \in X$  is the set of  $k$  closest points to  $x$  in the  $\ell_2$  norm sense. In other words,  $N_k(x) \subseteq X$  such that  $|N_k(x)| = k$  and  $\max_{z \in N_k(x)} \|x - z\| \leq \min_{z \in X \setminus N_k(x)} \|x - z\|$ .

**Definition.** The  $k$ -neighborhood graph of  $X$  is an undirected graph  $G_k = (V, E)$  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  $\varepsilon$ -neighborhood graph  $G_\varepsilon$  is defined with  $N_\varepsilon(x)$ .

**Definition.** Assume a symmetric matrix  $W \in \mathbb{R}^{n \times n}$  of non-negative weights on the edges of  $G_k$ . 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  $G_k$ , and is denoted by  $d_{G_k, W}(x, y)$ .

(set to  $n \max_{x_i, x_j \in X} W_{ij}$ , effectively  $\infty$ , if there is no such path). Similarly, the  $\varepsilon$ -geodesic distance  $d_{G_\varepsilon, W}$  is defined over the undirected  $\varepsilon$ -neighborhood graph  $G_\varepsilon$ .

**Theorem 2.1.** *If  $G$  is an  $\varepsilon$  or  $k$ -neighborhood graph over  $X$  weighted by a symmetric matrix  $W$  of non-negative weights, then  $d_{G, W}$  defines a distance on  $X$ .*

In practice, geodesic distance can be efficiently computed using Dijkstra's algorithm. A special case of geodesic distance has been used in [10] for finding manifold embeddings, with the Euclidean distances between neighbor points as the weights  $W$ . Since we are concerned with robustly separating clusters in a data set in the presence of outliers, we define  $W$  such that it captures the local density of data points. Given a density estimator  $\hat{f}$ , we define the *density scaling weights* as

$$W_{ij} = \exp(1/2\sigma^2 \max\{\hat{f}(x_i), \hat{f}(x_j)\} \|x_i - x_j\|), \quad (1)$$

for all  $(x_i, x_j) \in E$  with  $G = (V, E)$ . Intuitively, this choice of weights on the edges of our original neighborhood graph introduces an exponential increase in the total weight of paths that pass through low density regions of the data set. Therefore, the intra-cluster shortest paths would tend to pass through the corresponding cluster, while the inter-cluster shortest paths would have much larger total weights compared to them.

We use a nonparametric density estimation [4] based on neighborhood graphs. For some integer  $k_d$ , suppose we have the  $k_d$ -neighborhood graph  $G_{k_d}$  over  $X$ . For any  $x_i \in X$ , let  $R(N_{k_d}(x_i))$  be the distance from  $x_i$  to its farthest neighbor in its neighborhood  $N_{k_d}(x_i)$ , i.e.,  $R(N_{k_d}(x_i)) = \max_{x_j \in N_{k_d}(x_i)} \|x_i - x_j\|$ . Moreover, let  $\text{Vol}(r)$  represent the volume (Lebesgue measure) of a Euclidean ball of radius  $r$  in  $\mathbb{R}^q$ , i.e.,  $\text{Vol}(r) = 2r^q \pi^{q/2} / q \Gamma(q/2)$ . Then the  $k$ -NN estimator at any point  $x_i \in X$  is given by

$$\hat{f}(x_i) = \frac{k_d - 1}{n \cdot \text{Vol}(R(N_{k_d}(x_i)))}. \quad (2)$$

The  $k$ -NN estimator converges in probability if  $k_d \rightarrow \infty$  and  $k_d/n \rightarrow 0$  as  $n \rightarrow \infty$  [6].

### 3 Algorithms

#### 3.1 General-Distance K-means

We propose a different formulation of K-means that, unlike classical K-means, does not rely on Euclidean coordinates. Let  $(X, d)$  be a metric space in which  $X = \{x_1, x_2, \dots, x_n\}$  is a set of points and  $d : X \times X \rightarrow [0, \infty)$  defines a metric distance over  $X$ . Furthermore, suppose  $\gamma : X \rightarrow \{1, 2, \dots, K\}$  is a clustering function that assigns each data point to one of the  $K$  clusters.

**Definition.** The *General-Distance K-means loss function* is defined as

$$W_{GD}(X, \gamma) = \sum_{l=1}^K \sum_{\{i: \gamma(x_i)=l\}} \sum_{\{j: \gamma(x_j)=l\}} d^2(x_i, x_j). \quad (3)$$

If  $d$  is the Euclidean distance, this will reduce to the loss function for classical K-means [3], i.e.,  $W_E(X, \gamma) = \sum_{l=1}^K \sum_{\{i: \gamma(x_i)=l\}} \sum_{\{j: \gamma(x_j)=l\}} \|x_i - x_j\|^2$ . However, there is a deeper connection between the two objectives. A well known result in multivariate analysis [8] states the following.

**Theorem 3.1.** *Let  $X$  and  $d$  be defined as before,  $D_{ij} = d^2(x_i, x_j)$ , and  $H = -\frac{1}{n} \mathbf{1}_n \mathbf{1}_n^T$ . If  $-\frac{1}{2} H D H$  is positive semidefinite, then there exists a set of  $n$  points  $Y = \{y_1, y_2, \dots, y_n\}$  in  $\mathbb{R}^p$ , for some  $p \leq n - 1$ , such that  $\|y_i - y_j\|_2 = d(x_i, x_j)$  for all  $1 \leq i, j \leq n$ .*

Also, the inner product over  $Y$  can be written in terms of metric distance  $d$  over  $X$ . It is straightforward to verify the following result.

**Theorem 3.2.** *Under the assumptions of Theorem 3.1, for any clustering function  $\gamma$ , denote the cluster centroids by  $\bar{y}_l = \frac{1}{n_l} \sum_{i: x_i \in C_l} y_i$  for  $1 \leq l \leq K$  with  $C_l = \{x_i : \gamma(x_i) = l\}$  and  $n_l = |C_l|$ . Then for any  $1 \leq i, j \leq n$  and  $1 \leq l \leq K$  we have:*

$$\begin{aligned} \langle y_i - \bar{y}_l, y_j - \bar{y}_l \rangle &= -\frac{1}{2} (d^2(x_i, x_j) - \frac{1}{n_l} \sum_{x_r \in C_l} d^2(x_r, x_j) \\ &\quad - \frac{1}{n_l} \sum_{x_r \in C_l} d^2(x_i, x_r) + \frac{1}{n_l^2} \sum_{x_r, x_{r'} \in C_l} d^2(x_r, x_{r'})). \end{aligned} \quad (4)$$

This theorem enables us to interpret the general-distance loss function  $W_{GD}$  in a different way.

**Theorem 3.3.** *With  $X, Y$ , and  $d$  as before, we have*

$$W_{GD}(X, \gamma) = W_E(Y, \gamma) = 2 \sum_{l=1}^K n_l \sum_{\{i: x_i \in C_l\}} \|y_i - \bar{y}_l\|^2.$$

This equivalence means that minimization of the general-distance loss over  $X$  can be achieved by minimizing the Euclidean loss over  $Y$ . The classical K-means is clearly fit to do the latter. However, computing the coordinates of all  $y_i \in \mathbb{R}^p$  can be a computational burden since the dimension  $p$  can be as large as  $n - 1$ . Theorem 3.2 enables us to calculate the Euclidean distances from any point  $y_i$  to cluster centroids in terms of the metric distance  $d$  on  $X$ , i.e.,

$$\|y_i - \bar{y}_l\|^2 = \frac{2}{n_l} \sum_{x_r \in C_l} d^2(x_i, x_r) - \frac{1}{n_l^2} \sum_{x_r, x_{r'} \in C_l} d^2(x_r, x_{r'}). \quad (5)$$

---

**Algorithm 1** : General-Distance K-means

---

**Require:** Set of  $n$  points  $X$ , metric distance  $d$  over  $X$ , an initial assignment  $\gamma_0$  with no empty clusters,  $t = 0$ .

**repeat**

$C_l(t) = \{x_i : \gamma_t(x_i) = l\}$ ,  $n_l(t) = |C_l(t)|$ ,  $1 \leq l \leq K$ .

Update cluster assignments for all  $x_i \in X$  as

$$\gamma_{t+1}(x_i) = \arg \min_{1 \leq l \leq K} \left( \frac{2}{n_l(t)} \sum_{x_r \in C_l(t)} d^2(x_i, x_r) - \frac{1}{n_l^2(t)} \sum_{x_r, x_{r'} \in C_l(t)} d^2(x_r, x_{r'}) \right). \quad (6)$$

$t \leftarrow t + 1$

**until**  $|W_{GD}(X, \gamma_{t+1}) - W_{GD}(X, \gamma_t)|$  is small enough.

---

Therefore, we can perform K-means clustering on  $Y$  in  $\mathbb{R}^p$  without explicitly calculating the coordinates of  $y_i$ 's. Putting everything together, the general-distance K-means algorithm is outlined in Algorithm 1.

**Theorem 3.4.** *Assuming that  $(X, d)$  can be embedded in some Euclidean space (i.e., assumptions of Theorem 3.1 hold), in each iteration of general-distance K-means algorithm the loss function improves. In other words,  $W_{GD}(X, \gamma_{t+1}) \leq W_{GD}(X, \gamma_t)$ .*

### 3.2 Geodesic K-means Clustering

The general-distance K-means is computationally expensive due to the fact that cluster centroids are implicitly defined by intra-cluster distances according to  $d$ . Therefore, calculating the distance from any point  $x_i$  to each cluster centroid as in (5) requires manipulation of many pairwise distances. In the case of graph-based distances, such as the class of geodesic distances in Section 2, it is possible to overcome this issue by making centroids more explicit as follows.

**Definition.** Given a geodesic distance  $d_{G,W}$  defined over  $X$ , the *virtual centroid* of any cluster  $C_l$  is a representation of the true centroid  $\bar{y}_l$  (according to Theorem 3.1) in graph  $G$  as a new vertex  $\tilde{x}_{n+l}$  that augments  $G$  along with edges  $\{(\tilde{x}_{n+l}, x_i) | x_i \in M_l\}$  for a set of *landmark points*  $M_l \subseteq C_l$ .

This enables us to approximate the distance from any point  $x_j \in X$  to the centroid of any cluster  $C_l$  by the geodesic distance  $d_{\tilde{G},W}(x_j, \tilde{x}_{n+l})$  in the augmented graph  $\tilde{G}$ . We choose the weights on the augmented edges to be  $W_{(n+l)i} = \|y_i - \bar{y}_l\|$  given by (5).

Now the question is how to choose the landmark points in each cluster. Since  $G$  is a neighborhood graph, one natural idea is to set the landmark points of cluster  $C_l$  to be the  $(\varepsilon$  or  $k$ ) neighborhood of  $\tilde{x}_{n+l}$  according to  $\|y_i - \bar{y}_l\|$ . This would require the computation

---

**Algorithm 2** : Geodesic K-means

---

**Require:** Set of  $n$  points  $X$ , number of clusters  $K$ , neighborhood parameter  $\varepsilon$  or  $k$ , sampling rate  $r_s \in (0, 1]$ , an initial clustering assignment  $\gamma_0$  with no empty clusters,  $t = -1$ . Construct the  $(\varepsilon$  or  $k$ )-neighborhood graph  $G$  over  $X$ . Calculate the matrix of weights  $W$  according to (1).

**repeat**

$\tilde{G} \leftarrow G$ ,  $t \leftarrow t + 1$

**for** all  $1 \leq l \leq K$  **do**

$C_l = \{x_i : \gamma_t(x_i) = l\}$ ,  $n_l = |C_l(t)|$ .

Take a random sample  $S_l$  of  $\lceil r_s n_l \rceil$  points from  $C_l$ .

Calculate the distances from virtual centroid  $\tilde{x}_{n+l}$  to all  $x_i \in S_l$  by

$$\ell(\tilde{x}_{n+l}, x_i) = \frac{2}{\lceil r_s n_l \rceil} \sum_{x_r \in S_l} d_{G,W}^2(x_i, x_r) - \frac{1}{\lceil r_s n_l \rceil^2} \sum_{x_r, x_{r'} \in S_l} d_{G,W}^2(x_r, x_{r'}). \quad (7)$$

Let  $N(\tilde{x}_{n+l})$  be the  $(\varepsilon$  or  $k$ )-neighborhood of  $\tilde{x}_{n+l}$  in  $S_l$  according to  $\ell(\tilde{x}_{n+l}, \cdot)$ . Augment  $\tilde{G}$  by adding vertex  $\tilde{x}_{n+l}$  and the edges  $(\tilde{x}_{n+l}, x_i)$  for all  $x_i \in N(\tilde{x}_{n+l})$  with weights  $W_{(n+l)i} = \ell(\tilde{x}_{n+l}, x_i)$ .

**end for**

Update cluster assignments for all  $x_i \in X$  as

$$\gamma_{t+1}(x_i) = \arg \min_{1 \leq l \leq K} d_{\tilde{G},W}(x_i, \tilde{x}_{n+l}). \quad (8)$$

**until**  $|W_{GD}(X, \gamma_{t+1}) - W_{GD}(X, \gamma_t)|$  is small enough.

---

in (5) for all  $x_i \in C_l$ , as opposed to all  $x_i \in X$  required by the general-distance K-means. We can, however, go further than that. Recall the definition of cluster centroids in  $\mathbb{R}^p$  given by  $\bar{y}_l = \frac{1}{n_l} \sum_{i: x_i \in C_l} y_i$ . The law of large numbers allows us to approximate  $\bar{y}_l$  by  $\hat{\bar{y}}_l = \frac{1}{|S_l|} \sum_{i: x_i \in S_l} y_i$  with a random sample set (without replacement)  $S_l \subseteq C_l$ . When  $|S_l|$  is large enough,  $\hat{\bar{y}}_l$  will be a good approximation of  $\bar{y}_l$ . Algorithm 2 outlines the resulting approximate geodesic K-means.

Although  $(X, d_{G,W})$  may not always be exactly embedded in some Euclidean space (as required by Theorem 3.4), our simulation results show that geodesic K-means yields very good clustering performance. This verifies the fact that  $d_{G,W}$  is not far from being embedded in some Euclidean space, in the sense that, by adding some small constant to all geodesic distances, they will be embedded in some Euclidean space. Deriving conditions under which geodesic distance is approximately embedded in a Euclidean space is underway.

## 4 Experiments

Here we present empirical evidence to demonstrate the power of geodesic K-means despite its simplicity. Figure 1 shows geodesic K-means (with  $k = 6$ , and

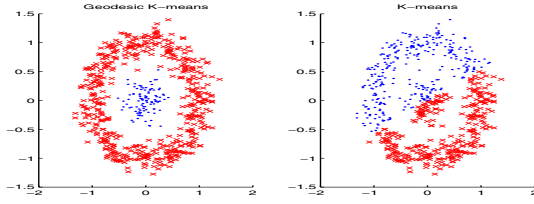


Figure 1. Noisy bull's eye dataset.

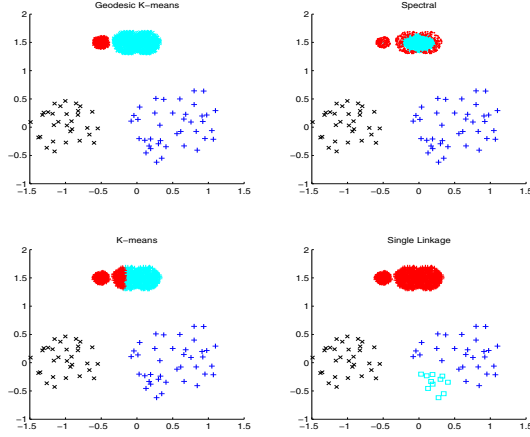


Figure 2. Four-heterogenous dataset.

$\sigma = 0.15$ ) handling the famous noisy bull's eye dataset, which involves clusters that are not linearly separable as well as outliers. The next synthetic dataset, four-heterogenous in Figure 2, consists of a mixture of clusters with different sizes and densities. Figure 2 draws comparison between the performance of geodesic K-means and that of classical K-means, normalized spectral clustering of Ng et al. [9], and single linkage algorithm [3]. Clearly both spectral (with Gaussian similarity matrix and the best  $\sigma = 0.025$ ) and single linkage algorithms perform poorly on this dataset, while geodesic K-means handles it with  $k = 10$  and  $\sigma = 0.01$ . Figure 3 presents a similar comparison between geodesic K-means ( $k = 6$  and  $\sigma = 0.03$ ), spectral clustering with Gaussian similarity matrix (best  $\sigma = 0.02$ ), and spectral clustering with  $k$ -nearest neighbor similarity, (with best  $k$  and  $\sigma$  being 50 and 0.02 respectively).

Finally, we have compared geodesic K-means to several clustering algorithms on the Iris dataset. Table 1 summarizes the number of errors for different algorithms, from left to right: spectral with Gaussian similarities ( $\sigma = 1$ ), spectral with  $k$ -neighborhood ( $k = 5$ ,  $\sigma = 4$ ), geodesic K-means ( $k = 4$ ,  $\sigma = 40$ ), K-means, single linkage, complete linkage, and average linkage.

Table 1. The Iris dataset.

SG	SK	GK	KM	SL	CL	AL
16	8	10	16	52	24	14

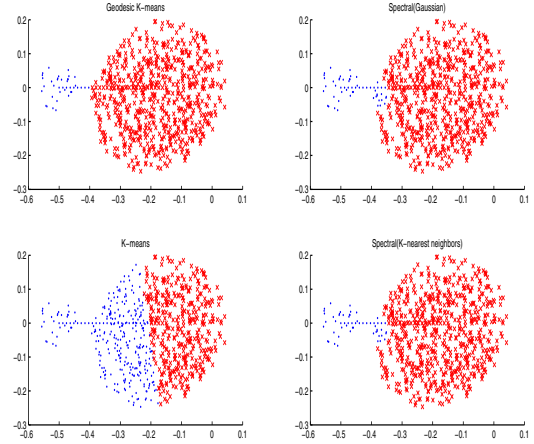


Figure 3. Two-heterogenous dataset.

## 5 Concluding Remarks

We demonstrated that the class of geodesic distances introduced in this paper has great potential to address many of the challenges in clustering problems. Specifically, we developed the geodesic K-means algorithm and showed that it claims a competitive position in comparison with spectral and hierarchical clustering.

## References

- [1] B. Feil and J. Abonyi. Geodesic distance based fuzzy clustering. *Lecture Notes in Computer Science, Soft Computing in Industrial Applications*, 39:50–59, 2007.
- [2] J. Hartigan and M. Wong. A k-means clustering algorithm. *Applied Statistics*, 28:100–108, 1979.
- [3] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of the Statistical Learning*. Springer, 2001.
- [4] A. J. Izenman. Recent developments in nonparametric density estimation. *Journal of the American Statistical Association*, 86(413):205–224, 1991.
- [5] J. Kim, K.-H. Shim, and S. Choi. Soft geodesic kernel k-means. *ICASSP 2007*, 2:429–432, April 2007.
- [6] D. O. Loftsgaarden and C. P. Quesenberry. A nonparametric estimate of multivariate density function. *Annals of Mathematical Statistics*, 36:1049–1051, 1965.
- [7] U. V. Luxburg. A tutorial on spectral clustering. *Technical Report*, pages 1–30, 2006.
- [8] K. V. Mardia, J. T. Kent, and J. M. Bibby. *Multivariate Analysis*. Academic press, 1979.
- [9] A. Ng, M. Jordan, and Y. Weiss. On spectral clustering analysis and an algorithm. pages 849–856. NIPS, 2002.
- [10] J. B. Tenenbaum, V. D. Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290:2319–2323, 2000.
- [11] R. Xu and D. Wunsch. Survey of clustering algorithms. *IEEE Trans. Neural Networks*, 16:645–678, 2005.