# A New Topology-Preserving Distance Metric with Applications to Multi-dimensional Data Clustering

Konstantinos K. Delibasis

# A new topology-preserving distance metric with applications to multi-dimensional data clustering

Konstantinos K. Delibasis[1]

[1] Dept. of Computer Science and Biomedical Informatics, University of Thessaly, Lamia, Greece
kdelibasis@gmail.com

**Abstract.** In many cases of high dimensional data analysis, data points may lie on manifolds of very complex shapes/geometries. Thus, the usual Euclidean distance may lead to suboptimal results when utilized in clustering or visualization operations. In this work, we introduce a new distance definition in multi-dimensional spaces that preserves the topology of the data point manifold. The parameters of the proposed distance are discussed and their physical meaning is explored through 2 and 3-dimensional synthetic datasets. A robust method for the parameterization of the algorithm is suggested. Finally, a modification of the well-known k-means clustering algorithm is introduced, to exploit the benefits of the proposed distance metric for data clustering. Comparative results including other established clustering algorithms are presented in terms of cluster purity and V-measure, for a number of well-known datasets.

**Keywords:** distance metric, data manifolds, clustering.

## 1 Introduction

Clustering high-dimensional data is an area that has attracted considerable research interest over the past two decades [1, 3, 6]. The existence of irrelevant features and correlations between subsets of features, which are commonly encountered in such datasets, renders the task of identifying clusters much harder as distances between observations become less informative about the cluster structure. Dimensionality reduction and Feature Embedding is widely used to improve clustering performance and to enable the visualization of the resulting cluster structure in such data. Although well-established methods like Principal Component Analysis (PCA) and metric Multi-Dimensional Scaling (MDS) [2] have been successfully applied on a plethora of high-dimensional applications, there is no guarantee that the cluster structure in the high-dimensional space will be preserved in the low-dimensional subspace since in many cases clusters could be defined by highly non-linear structure. For this purpose non-linear dimensionality reduction techniques have been explicitly designed to identify a lower dimensional manifold along which the data lie, and are therefore appropriate to distinguish nonlinearly separable clusters.

Kernel-based clustering is amongst the most popular methods for nonlinear clustering, based on the projection of the input data points into a high dimensional kernel space in order to make nonlinear clusters linearly separable [5]. In particular, kernel k-means combines the k-means method with the kernel trick in an attempt to deal with nonlinearly separable data, however specifying a suitable kernel function and appropriate parameters most of the times is a hard task. Another widely used manifold learning method is isometric mapping (ISOMAP). Instead of using the Euclidean distance, Isomap is based on approximating geodesic distance along the manifold [7]. However, isomap operates on neighboring data-points defined by Euclidean threshold distance, which accelerates the algorithm but presents problems in case of outlier points. In [8] the authors applied the k-means clustering algorithm after Isomap and proposed a modified definition of the geodesic distances but concluded that even their modified method was unsatisfactory in real-data cases where the data is noisy or the clusters are highly nonlinear.

In this work we propose a new topology preserving distance that follows the geodetic of the underlying manifold. Instead of imposing a threshold on distance between points, we construct a graph with all available points and impose a penalty function that penalizes distant points. The definition of distant points uses a characteristic distance parameter whose value is automatically estimated from the available dataset. Furthermore, we propose a modification of the k-means algorithm incorporating benefits of a newly introduced distance metric that preserves the topology of the data point manifold. A critical advantage of the proposed approach is the feasible robust parameterization of the algorithm. Extensive experiments on both simulated and real data sets employing the Purity and V-measure metrics for comparison, as described in [4] provide further evidence on the wide applicability of the proposed method.

## 2 Methodology

### 2.1 The proposed topology-preserving distance metric

Let $\mathbf{P}$ be a data matrix of dimensions $N \times K$, each row of which is a feature vector (equivalently a data point) $\mathbf{p} = (x_1, x_2, ..., x_K)$ of dimensionality $K$. Any given set of such points may be arranged on an unknown manifold in the $\Re^K$ space. Thus, the Euclidean distance metric between any two points may not represent their actual distance.

Let us define an *auxiliary* distance metric between any pair of data points as:

$$D_{ij} = \begin{cases} \lambda d_{ij}, d_{ij} > d_0 \\ d_{ij}, d_{ij} \le d_0 \end{cases} \quad (1)$$

where $d_{ij} = \left\| \mathbf{p_i} - \mathbf{p_j} \right\|$ is the Euclidean distance between the two points, $\lambda$ is a sufficiently big value and $d_0$ is a characteristic distance, whose value is estimated for the

current dataset, as it will be described later. Let us stretch that $D_{ij}$ is not the distance metric proposed in this work, rather an auxiliary definition.

For a given set of data points $P=\{\mathbf{p_i}\}, i=1,2,...,N$ in $\Re^K$, a fully connected graph $G=(P,E)$ is defined with vertices $P$ as the set of all data points and edges $E$ as the set of all possible connections between vertices, $E=\{(i,j)\}, i,j=1,2,...,N$. Thus, each point is connected to all other points in the data set. The cost of the connection (edge) between any pair of points $\mathbf{p_i},\mathbf{p_j}$ is set equal to their auxiliary distance $D_{ij}$, as defined in Eq.(1). The proposed topology preserving distance between $\mathbf{p_i},\mathbf{p_j}$ is defined as the cost of the minimum-cost path $\pi_{\mathrm{ij}}$ according to the well-known Dijktra's algorithm, between the two points.

$$A(\mathbf{p_i},\mathbf{p_j})=A_{ij}=\mathrm{cost}(\pi_{\mathrm{ij}}), i,j=1,2,...,N \tag{2}$$

Since any generated path $\pi_{\mathrm{i,j}}$ between $\mathbf{p_i},\mathbf{p_j}$ consists of an ordered series of data points with indices $(i_1,i_2,...,i_M)$, where $i_1=i, i_M=j$ the proposed topology preserving distance between $\mathbf{p_i},\mathbf{p_j}$ is calculated as

$$A_{i,j}=\mathrm{cost}(\pi_{\mathrm{i,j}})=\sum_{m=1}^{M-1}D_{m,m+1}, i=i_1, j=i_M \tag{3}$$

The parameter $d_0$ is a characteristic length in $\Re^K$ that defines the scale of local linearity in a given set of data points. It is self-evident that any two points with Euclidean distance less than or equal to $d_0$ will be connected without any intermediate points. On the other hand, for any two points with Euclidean distance greater than $d_0$, the proposed algorithm will generate a connecting path with intermediate points, if sufficient pairs of these points have Euclidean distance not greater than $d_0$.

Fig. 1 shows the paths generated by the Dijkstra's algorithm using the auxiliary distance $D_{ij}$, in the case of data points generated using the Swiss roll data set [11], for $\lambda=10^8$, $d_0=2$, using one randomly selected point $i_0$ as the source for Dijkstra's algorithm (the point from which all other distances are calculated). The dataset is constructed to contain 2 classes ($N_c=2$), 400 points each, denoted by different color. The points lie on a manifold that is defined by a parametric equation. The paths $\pi_{\mathrm{i_0,j}}$ are also plotted as blue lines for all points $j=1,2,\ldots, 800, j\neq i_0$. As it can be observed, the selected value of $d_0$ generates shortest paths that lie on the manifold, rather than crossing the gap as it would be dictated by the Euclidean distance. The use of this propose distance metric in any clustering or classification process on this dataset is expected to significantly increase the achieved accuracy.

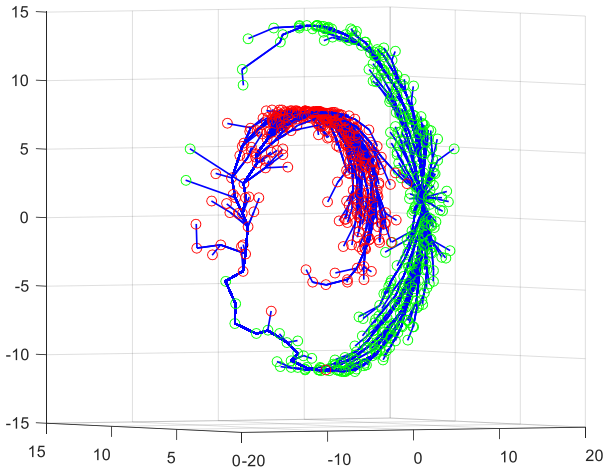


**Fig. 1.** The paths generated by the proposed distance definition in the case of the Swiss Roll dataset, for $d_0$=2. The points lie on a manifold and consist of two classes shown in green and red color

### 2.2 Determining the value of $d_0$ parameter

The parameter $d_0$ is very important for the efficient operation of the algorithm. A low value of $d_0$ would cause $D_{ij} = \lambda d_{ij}$ for almost all pairs of points $\mathbf{p_i}, \mathbf{p_j}$. Consequently, the Dijkstra's algorithm would return a single edge from $\mathbf{p_i}$ to $\mathbf{p_j}$, rather than generating a path with intermediate points. Eq.(1) would be simplified to $A_{ij} = D_{ij}$, thus the proposed metric would be equivalent to a Euclidean one. On the other hand, a high value for $d_0$ would cause $D_{ij} = d_{ij}$ for almost all pairs of points $\mathbf{p_i}, \mathbf{p_j}$, also resulting in a single-edge path, just as described previously.

Let $i_0$ be the index of a randomly selected point. The proposed algorithm is executed $N$-1 times connecting $i_0$ with all the rest $N$-1 points $\mathbf{p_j}$, calculating distance $A_{i_0 j}$ and generating the corresponding paths $\pi_{i_0 j}$. Let us denote the sequence of points that constitute the path from $i_0$ to $j$ as $(i_1, i_2, ..., i_M)$ with $i_0 = i_1$, $j = i_M$ and the series of Euclidean distances $\{d_{i_m, i_{m+1}}\}, m = 1,2,...,M$ . To simplify notation, let us use $d_{i_m, i_{m+1}} = d_{m,m+1}$. Let $d_{i_0 j}^{\max} = \max\{d_{m,m+1}\}, m = 1,2,...,M$ be the maximum Euclidean length of the steps in the path from $i_0$ to $j$. Then the average $d_{i_0 j}^{\max}$ can be calculated for the selected point $i_0$ over all other points $j$ in the data set:

$$\langle d_i \rangle = \frac{1}{N-1} \sum_{\substack{j=1, \\ j \neq i}}^{N} d_{i,j}^{\max} \tag{4}$$

By its definition, $\langle d_i \rangle$ is calculated for a selected point $i_0$ and it is a function of $d_0$. It is easily proven that when $d_0$ has very low values, below the minimum Euclidean distance $d_{\min}$ in the dataset, $\langle d_i \rangle$ is equal to the mean Euclidean distance between $i_0$ and all data points. In the case of data points being equally distributed (eg. on a regular grid), the quantity $\langle d_i \rangle$ is expected to be monotonically increasing when $d_0$ in $[d_{\min}, d_{\max}]$. When $d_0$ approaches values equal to $d_{\max}$, the $\langle d_i \rangle$ becomes equal to the mean Euclidean distance between $i_0$ and all data points and remains constant for larger values of $d_0$. In the case however of anisotropic data point distribution, $\langle d_i \rangle$ drops sharply when $d_0$ has an appropriate intermediate value, since the proposed algorithm generates connecting paths between data points that consist of steps with smaller distances. Finally, when $d_0$ approaches values equal to the maximum Euclidean distance in the dataset $d_{\max}$, the $\langle d_i \rangle$ becomes equal to the mean Euclidean distance between $i_0$ and all data points. Thus, for any point in the dataset $i$, the quantity is calculated for different values of $d_0$ and the value of $d_0 = d_{\min}^i$ that produces the minimal value is determined

$$d_{\min}^i = \arg\max_{d_0} \langle d_i \rangle, d_0 \in [d_{\min}, d_{\max}] \tag{5}$$

In the special case of data points lying on a manifold with shape of large scale concavities, then $d_{\min}^i$ has a value that indicates the characteristic length of the concavities. Thus, setting $d_0$ to a value less than $d_{\min}^i$ will cause the proposed algorithm to produce connecting paths between data points that do not cross the concavities, but lie on the manifold, thus behaving like geodetic curves. In order to obtain a good estimation of $d_{\min}^i$, the calculation is repeated for many randomly selected points in the dataset and the average $\bar{d}_{\min}^i$ is obtained.

Figure 2 shows the $\langle d_i \rangle$ calculated for 50 points randomly distributed in a two-dimensional dataset for $d_0$ in $[d_{\min}, d_{\max}]$. Two different datasets are used to demonstrate the aforementioned behavior: first 2601 points are put on a regular grid on the plane (a) and 2601 points are randomly placed on the plane (following the flat distribution). In both cases the [0,10]×[0,10] part of the plane is used. The $\langle d_i \rangle$ is plotted against the values of $d_0$. The average $\bar{d}_{\min}^i$ (over all 50 points) is plotted as a green circle, whereas the average thick curve also overlaid. The aforementioned behavior is obvious in both data sets, whereas the consistency for all different points is evident.

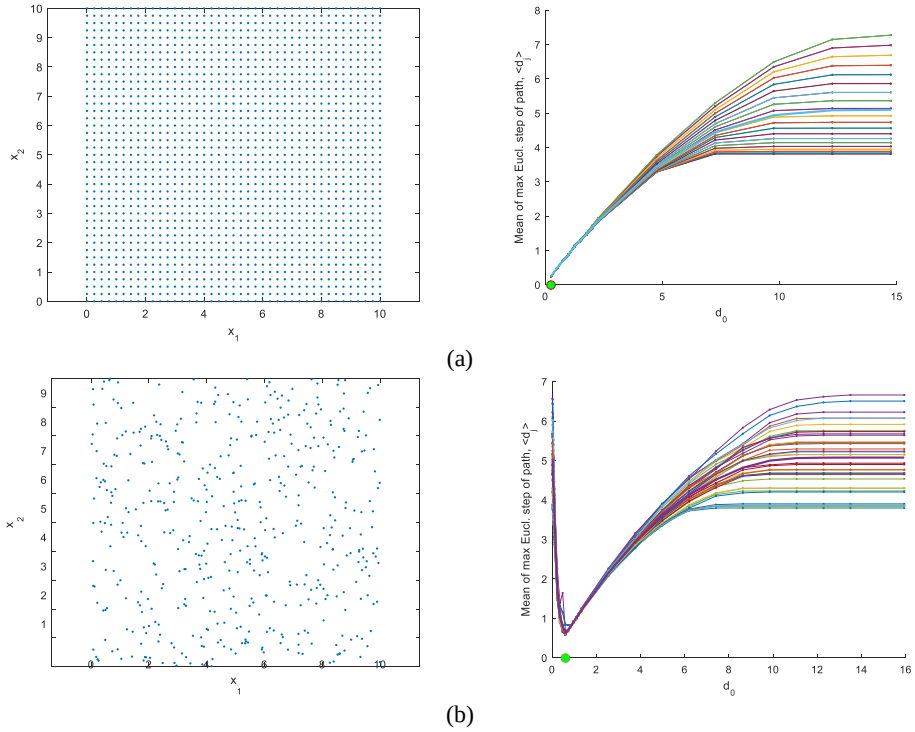


**Fig. 2.** The $\langle d_i \rangle$ as a function of $d_0$, for a number of data points and the determined $\bar{d}^{i}_{\min}$ plotted as a green circle, for data points a) on a 2D regular grid and (b) randomly distributed.

The same process is applied to the Swiss Roll dataset [11] using 800 points in 2 classes. The quantity $\langle d_i \rangle$ is plotted for $d_0$ in [$d_{\min}$, $d_{\max}$] =[ 0.0072, 48.4663]. The average $\bar{d}^{i}_{\min}$ (over all 40 points) was found equal to 3.6431.

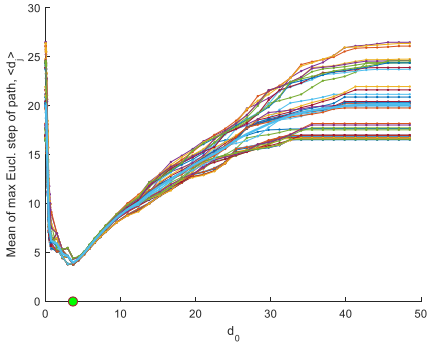


**Fig. 3.** The shape of $\langle d_i \rangle$ as a function of $d_0$, for the Swiss roll data set. The estimated $\bar{d}^{i}_{\min}$ is plotted as green circle.

The effect of $d_0$ is demonstrated in Fig. 4. One point is randomly selected from the dataset and the minimum cost connections with all other points are shown. The cost of the connecting path is calculated according to Eq. (3), as a function of $d_0$. It can be observed that for very low or very high values of $d_0$, the connecting paths cross the topological gap between points, whereas for intermediate values, the paths follow the manifold, as geodesic curves.

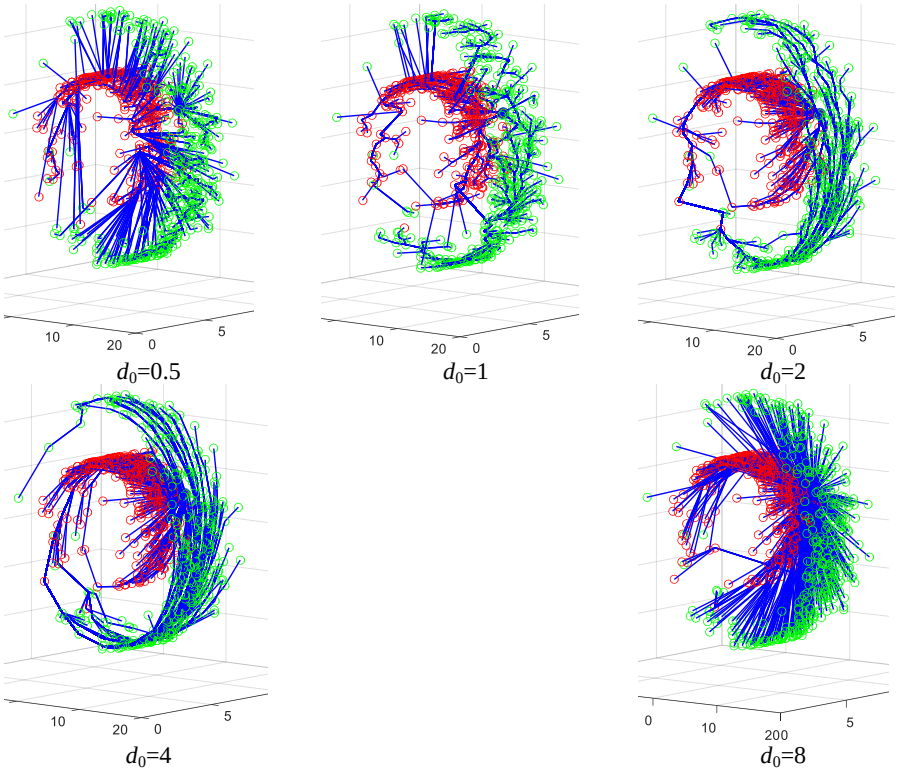


**Fig. 4.** The minimum cost paths generated by the proposed algorithm applied on the Swiss roll dataset, for different values of $d_0$. Intermediate values of $d_0$, as suggested by the estimation of the average $\bar{d}^{i}_{\min}$ , produce paths that follow the underlying manifold.

### 2.3 A k-means variant for the proposed topology-preserving distance metric

In this subsection a variant of the k-means clustering algorithm is proposed, that utilizes the proposed topology preserving distance metric. The main differences from the classic k-means algorithm can be summarized as following:

The $N \times N$ distance matrix is calculated using the proposed distance metric in Eq.(2) (it requires the characteristic length $d_0$): $\mathbf{A} = \{A(\mathbf{p_i}, \mathbf{p_j})\} = \{A_{ij}\}, i, j = 1,2,...,N$ .

The class centers in each iteration are selected from the data points, so that they minimize the average (proposed metric) distance from the members of the specific

class. The algorithm is terminated when all class centers remain unchanged in two consecutive iterations. The details of the proposed algorithm are given below.

Input: the data matrix $P$, the number of classes $N_c$, the $N \times N$ distance matrix **A** using the proposed metric.

```
Select randomly N_c class centers from P  {k_c}, c = 1,2,...,N_c
while NOT(termination condition)
  For each class center c
    Determine points P_c ⊆ P closer to k_c.
    Identify x_0 ∈ P_c, with minimal mean distance from all
members of P_c, (1/M_c) Σ_{y∈P_c} A(x_0, y) is (M_c: the population of P_c )
    Set k_c = x_0.
  end
end
```

Experimentation shows (see Results section) that the proposed variant of the k-means method produces consistently optimal results for values of $d_0$ slightly smaller than the estimated $\bar{d}^i_{\min}$.

## 3 Results

The proposed k-means variant that uses the proposed distance metric is evaluated against the classic k-means, the kernel k-means (implemented as in [9]) and the spectral clustering (implemented in [10] and [12]) in terms of purity of clustering, as well as V-measure. The proposed method has been executed for 20 times with random initialization and the resulting average purity, as well as the standard deviation are plotted for different values of $d_0$ in Fig. 5. The same quantities achieved by the classic k-means clustering, the kernel k-means and the spectral clustering are also shown. It can be observed that the proposed algorithm clearly outperforms the classic and the kernel k-means. The behavior of the proposed algorithm with respect to parameter $d_0$ is consistent with the estimated value of $\bar{d}^i_{\min}$: for values of less than the achieved clustering is consistently high. For values of $d_0 > \bar{d}^i_{\min}$, the proposed algorithm behaves very similarly to the classic k-means. This is expected, since as described above the proposed distance definition becomes similar to the Euclidean one.

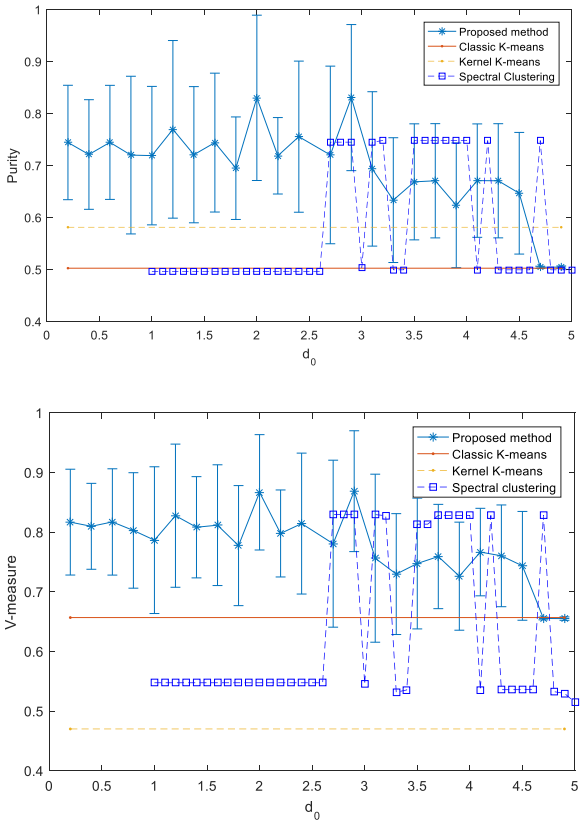


**Fig. 5.** The achieved purity and V-measure using the proposed method, applied to the Swiss roll dataset, for different values of $d_0$, against the classic k-means, the kernel k-means [9] and the spectral clustering [10].

Fig. 6 shows the same results for the COIL dataset. The determination of $\bar{d}^{i}_{\min}$ as shown in Fig. 6(a) is unambiguous. The behavior of the proposed k-means variant with respect to $d_0$, is also very consistent, with best performance occurring at values of $d_0$ slightly smaller than the estimated $\bar{d}^{i}_{\min}$. The proposed k-means variant with the suggested distance metric outperforms the other methods in comparison.

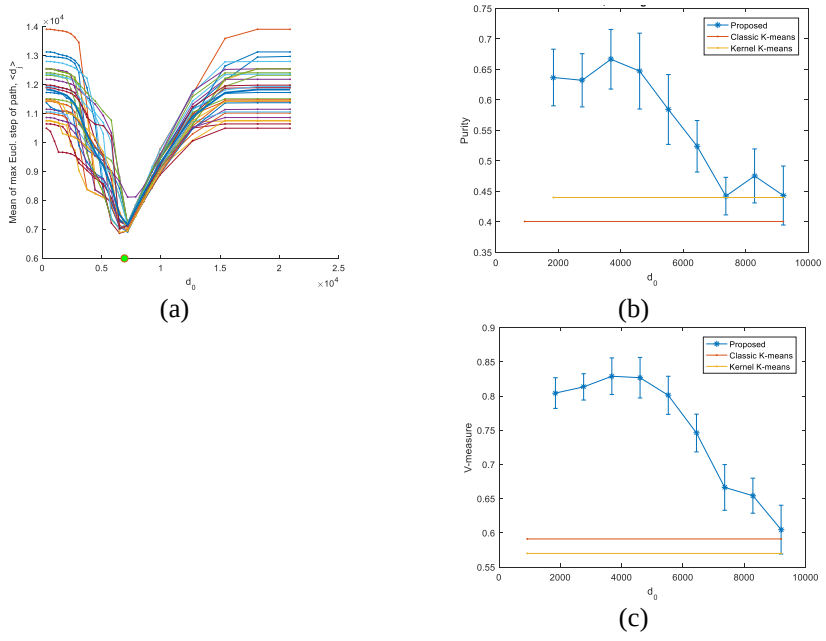


**Fig. 6.** The achieved purity and V-measure using the proposed method applied to the COIL dataset, for different values of $d_0$, against the classic k-means, the kernel k-means [9]. The spectral clustering [9] produced worse results and it was therefore not included in the graph.

Table 1 shows the clustering purity and V-measure achieved by the proposed method, k-means, kernel k-means and spectral clustering. The values for the proposed method were calculated by using the corresponding value for $d_0$ slightly less than the estimated $\overline{d}^{i}_{\min}$. The standard Matlab implementation was used for the k-means method. Kernel k-means was used as provided in [9]. Spectral clustering was used as provided in [10] and/or in [12] that implements the algorithm described in [13].

**Table 1.** Dataset description, with the achieved clustering purity and V-measure by the proposed method, classic and kernel k-means and spectral clustering.

| Dataset | Classes, Dim, Num. | $\overline{d}^{i}_{\min}$ | Proposed | | K-means | | Kernel K-m | | Spectral Cl. | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | Pur. | V-m | Purity | V-m. | Purity | V-m | Purity | V-m |
| Swiss Roll | 4 3 1600 | 3.64 | 0.75 | 0.83 | 0.505 | 0.67 | 0.59 | 0.47 | 0.775 | 0.85 |
| Olivetti | 10 4096 400 | 2057 | 0.57 | 0.75 | 0.47 | 0.68 | 0.25 | 0.43 | 0.30 | 0.551 |
| Umist | 20,1030 4414 | 3808 | 0.7 | 0.84 | 0.58 | 0.66 | 0.23 | 0.251 | 0.19 | 0.22 |
| Isolet | 26 617 1567 | 9.38 | 0.48 | 0.67 | 0.425 | 0.59 | 0.269 | 0.402 | 0.155 | 0.258 |
| Coil | 20 $2^{16}$ 1440 | 6950 | 0.64 | 0.81 | 0.407 | 0.595 | 0.44 | 0.565 | 0.285 | 0.480 |
| Phoneme | 4 256 4509 | 40.27 | 0.65 | 0,61 | 0.565 | 0.395 | 0.779 | 0.691 | 0.302 | 0.104 |

## 4 Conclusions

A new distance metric for high-dimensional data has been presented that preserves the topology of the underlying manifold. A variant of the k-means clustering algorithm has been suggested that utilizes this metric. The value of the main parameter of the proposed distance metric can be obtained with a standard and efficient process. The performance of the proposed method has been analyzed theoretically and validated experimentally, on a number of benchmark datasets. Comparative results with well-established clustering algorithms show that the proposed method is a competent alternative with consistent behavior that systematically performs equally well or better than the other techniques under comparison. Future work includes the algorithmic fine tuning of the proposed k-means variant and the extension of the application of the distance metric to visualization and dimensionality reduction techniques. Comparative results will also be expanded to include more optimized implementations of other state of the art methods.

## Acknowledgement

The author wishes to acknowledge the partial support by the Interdepartmental Postgraduate Program “Computer Science and Computational Biomedicine” of the School of Science of the University of Thessaly

## References

1. R. Agrawal, J. Gehrke, D. Gunopulos, and P. Raghavan. Automatic subspace clustering of high dimensional data, 2005
2. M. A. Cox and T. F. Cox. Multidimensional scaling. In Handbook of Data Visualization, Springer Handbooks Comp.Statistics, pages 315{347. Springer Berlin Heidelberg, 2008
3. H.-P. Kriegel, P. Kro3ger, and A. Zimek. Clustering high-dimensional data: A survey on subspace clustering, pattern-based clustering, and correlation clustering. ACM Transactions on Knowledge Discovery from Data, 3(1):1{58, 2009
4. N. G. Pavlidis, D. P. Hofmeyr, and S. K. Tasoulis. Minimum density hyperplanes. Journal of Machine Learning Research, 17(156):1{33, 2016
5. B. Schoelkopf, A. Smola, and K.-R. Mueller. Nonlinear component analysis as a kernel eigenvalue problem. Neural Computation, 10(5):1299{1319, 1998
6. S. K. Tasoulis, D. K. Tasoulis, and V. P. Plagianakos. Enhancing principal direction divisive clustering. Pattern Recognition, 43(10):3391{3411, 2010
7. J. B. Tenenbaum, V. Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. Science, 290(5500):2319{2323, 2000.

8. H. Yu, X. Zhang, Y. Yang, X. Zhao, and L. Cai. An extended isomap by enhancing similarity for clustering. In H. Jiang, W. Ding, M. Ali, and X. Wu, editors, Advanced Research in Applied Arti_cial Intelligence, volume 7345 of Lecture Notes in Computer Science, pages 808{815. Springer Berlin Heidelberg, 2012
9. Mehmet Gonen and Adam A. Margolin. Localized Data Fusion for Kernel k-Means Clustering with Application to Cancer Biology. Advances in Neural Information Processing Systems 27 (NIPS 2014), Montrıal, Quıbec, Canada, 2014
10. Mathworks Homepage, https://www.mathworks.com/matlabcentral/fileexchange/46733-spectral-clustering, last accessed 2/3/2019
11. http://people.cs.uchicago.edu/~dinoj/manifold/swissroll.html
12. https://www.mathworks.com/matlabcentral/fileexchange/34412-fast-and-efficient-spectral-clustering
13. Ulrike von Luxburg, "A Tutorial on Spectral Clustering", Statistics and Computing 17 (4), 2007