# A Novel Path-Based Clustering Algorithm Using Multi-dimensional Scaling

Uyen T.V. Nguyen, Laurence A.F. Park, Liang Wang, and Kotagiri Ramamohanarao

Department of Computer Science and Software Engineering
The University of Melbourne, Victoria, Australia 3010
{thivun, lapark, lwwang, rao}@csse.unimelb.edu.au

**Abstract.** Data clustering is a difficult and challenging task, especially when the hidden clusters are of different shapes and non-linearly separable in the input space. This paper addresses this problem by proposing a new method that combines a path-based dissimilarity measure and multi-dimensional scaling to effectively identify these complex separable structures. We show that our algorithm is able to identify clearly separable clusters of any shape or structure. Thus showing that our algorithm produces model clusters; that follow the definition of a cluster.

**Keywords:** Unsupervised learning, path-based clustering.

## 1 Introduction

Data clustering, or cluster analysis, is the process of finding a natural partition of a set of patterns, points or objects [1]. The clustering task plays a very important role in many areas such as exploratory data analysis, pattern recognition, computer vision, and information retrieval. Although cluster analysis has a long history, there are still many challenges, and the goal of designing a general purpose clustering algorithm remains a challenging task [2]. Intuitively, the clustering task can be stated as follows: given a set of $n$ objects, a clustering algorithm tries to partition these objects into $k$ groups so that objects within the same group are alike while objects in different groups are not alike. However, the definition of similarity is application dependent and sometimes unknown, which makes clustering an ill-posed problem.

Despite many clustering algorithms being proposed, K-means is still widely used and is one of the most popular clustering algorithms [2]. This is because it is an efficient, simple algorithm and provides successful results in many practical applications. However, K-means is only good at clustering compact and Gaussian shaped clusters and fails in capturing elongated clusters, or clusters that are non-linearly separable in the input space [3]. In order to tackle this problem, kernel K-means was introduced [4]. This method maps the data into a higher dimensional feature space defined by a non-linear function (intrinsic in the kernel function) so that the possibility of separating the data linearly becomes feasible. However, the task of choosing a suitable kernel function and its parameters for a given dataset is difficult. Another emerging approach is to use a spectral clustering algorithm, which performs

A. Nicholson and X. Li (Eds.): AI 2009, LNAI 5866, pp. 280–290, 2009.

the clustering on a set of eigenvectors of the affinity matrix derived from the data. It has been shown that results obtained by spectral clustering often outperform traditional clustering algorithms like K-means [5]. Although there are many different points of view to explain why spectral clustering works [6-8], it is still not completely understood yet. Moreover, spectral clustering leaves the users many choices and parameters to be set such as the similarity metric and its parameters, the type of graph Laplacian matrix, and the number of eigenvectors to be used [5]. Unfortunately, the success of spectral clustering depends heavily on these choices which make using spectral clustering a difficult task for the user.

In this paper, we propose a new clustering method that is capable of capturing clusters with different shapes that are non-linearly separable in the input space. This is not a new problem and there are two main approaches that address this problem that can be found in the literature [9-12]. In [9-11], a new path-based dissimilarity measure was proposed to embed the connectedness information between objects and a cost function based on this new dissimilarity measure was introduced. Optimization techniques were then used to find a set of clusters that minimizes this cost function. However, finding an optimal partition that minimizes this new cost function is a computationally intensive task and many different optimization techniques have been considered to address this problem. Another approach to this problem [12] is to improve a spectral clustering algorithm by using a path-based similarity measure. Instead of performing the spectral analysis directly on the similarity matrix, they modify the similarity matrix to include the connectedness among the objects. However, this approach is based on spectral clustering, which has many disadvantages as mentioned above.

We address the same problem but approach it differently. Instead of finding a new cost function like [9-11] or improve an existing algorithm like [12], we transform the original data into a new representation that takes into account the connection between objects so that the structures inherent in the data are well represented. This is achieved by a combination of the path-based dissimilarity measure and multi-dimensional scaling as described in Section 3. Compared with other methods, our method is much simpler yet produces very impressive results. The results prove that our new method is able to identify complex and elongated clusters in addition to the compact ones.

## 2 Background

In this section, we present the two main theories that are used by our algorithm. The first is the path-based dissimilarity measure, which gives a new way to identify the dissimilarity between two objects by taking into account the connection among objects. The second is the multi-dimensional scaling technique which is used to find a set of data points that exhibit the dissimilarities given by a dissimilarity matrix.

### 2.1 Path-Based Dissimilarity Measure

The path-based dissimilarity measure was first introduced in [9]. The intuitive idea behind this is that if two objects $x_i$, $x_j$ are very far from each other (reflected by a large distance value $d_{ij}$ with respect to metric $m$), but there is a path through them consisting

of other objects such that the distances between any two successive objects are small, then $d_{ij}$ should be adjusted to a smaller value to reflect this connection. The adjustment of $d_{ij}$ reflects the idea that no matter how far the distance between two objects may be, they should be considered as coming from one cluster if they are connected by a set of successive objects forming density regions. This is reasonable and reflects the characteristic of elongated clusters.

The path-based dissimilarity measure can be described in a more formal way. Suppose that we are given a dataset of $n$ objects $X$ with each object $x_i$ consisting of $m$ features, $x_i = (x_{i1}, x_{i2},\ldots x_{im})$ and an $n \times n$ distance matrix $D$ holding the pair-wise distances of all pairs of objects in $X$. The objects and their distance matrix $D$ can be seen as a fully connected graph, where each vertex in this graph corresponds to an object and the edge weight between two vertices $i$ and $j$ is the distance between the corresponding objects $x_i$ and $x_j$, or $d_{ij} = dis(x_i, x_j)$. The path-based distance between $x_i$ and $x_j$ is then defined as follows.

Suppose that $P_{ij}$ is the set of all possible paths from $x_i$ to $x_j$ in the graph, then for each path $p \in P_{ij}$, the effective dissimilarity between $x_i$ and $x_j$ along $p$ is the maximum of all edge weights belonging to this path. The path-based distance $d_{ij}'$ between $x_i$ and $x_j$ ($pbdis(x_i, x_j)$), is then the minimum of effective dissimilarities of all paths in $P_{ij}$, or:

$$d_{ij}^{'} = pbdis(x_i, x_j) = \min_{p \in P_{ij}} \{ \max_{1 \le h < |p|} (dis(p[h], p[h+1])) \} \tag{1}$$

where $p[h]$ denotes the object at the $h^{th}$ position in the path $p$ and $|p|$ denotes the length of path $p$.

### 2.2 Multi-dimensional Scaling

Multi-dimensional scaling (MDS) is a technique that allows us to visually explore the data based on its dissimilarity information. In general, given a pair-wise distance matrix of a set of objects, the MDS algorithm finds a new data representation, or a configuration of points, that preserves the given pair-wise distances for a given metric as well as possible. Many MDS algorithms are available [13] and they can be divided into two main categories: metric MDS and non-metric MDS. For the sake of completeness, the theory behind classical multi-dimensional scaling is presented here to show how an MDS algorithm works. Classical multi-dimensional scaling is an attractive MDS method as it provides an analytical solution using an eigen-decomposition.

To start with, the process of deriving the matrix of squared pair-wise distances from a coordinate matrix (also known as data or pattern matrix) in terms of matrix operations is presented. Let $X$ be an $n \times m$ coordinate matrix with each row $i$ containing the coordinates of point $x_i$ on $m$ dimensions ($x_i = [x_{i1}, x_{i2},\ldots, x_{im}]$) and $D^{(2)}(X)$ the squared distance matrix where each element at $(i, j)$ is the squared distance between $x_i$ and $x_j$. Suppose that Euclidean distance is used, then:

$$d_{ij}^2 = \sum_{k=1}^{m} (x_{ik} - x_{jk})^2 \tag{2}$$

After some simple transformations, the squared distance matrix $D^{(2)}(X)$ can be computed using a compact expression:

$$D^{(2)}(X) = \tilde{c}\tilde{1}^T + \tilde{1}\tilde{c}^T - 2XX^T \tag{3}$$

where $\tilde{c}$ is a vector with the diagonal elements of $XX^T$, and $\tilde{1}$ is a vector of ones.

Classical multi-dimensional scaling reverses this composition. It takes a dissimilarity matrix $\Delta$ (with each element $\delta_{ij}$ the dissimilarity between two unknown objects $x_i$ and $x_j$) as input and finds a set of points $Z = \{z_1, z_2, \ldots, z_n\}$ so that each pair-wise distance $d_{ij}$ (the distance between $z_i$ and $z_j$) is as close to $\delta_{ij}$ as possible. This can be done using an eigen-decomposition.

Suppose that $Z$ is the $n \times m'$ coordinate matrix that best matches $\Delta$, then $Z$ and $\Delta$ should be related by (3), or:

$$\Delta^{(2)} = \tilde{c}\tilde{1}^T + \tilde{1}\tilde{c}^T - 2ZZ^T \tag{4}$$

where $\Delta^{(2)}$ is the squared dissimilarity matrix, $\tilde{c}$ is now the vector with diagonal elements of $ZZ^T$.

Because distances are invariant under translation, we assume that $Z$ has column means equal to *0*. Then multiplying the left and right sides of (4) by the centering matrix $J (J = I - (1/n)\tilde{1}\tilde{1}^T)$ and *-1/2*, and after some reductions, we have:

$$B = ZZ^T = (-1/2)J\Delta^{(2)}J \tag{5}$$

So the scalar product matrix $B$ of $Z$ can be derived from the dissimilarity matrix $\Delta$ as above. From the scalar product matrix $B$, the coordinate matrix $Z$ is easily computed by using an eigen-decomposition. Let $Q$ and $\Lambda$ be the eigenvector and eigenvalue matrices of $B$ respectively. Since $B$ is a real and symmetric matrix (because of the symmetry of $\Delta$), we have:

$$B = Q\Lambda Q^T \tag{6}$$

If $\Delta$ is a Euclidean distance matrix, which means that it is constructed from the pair-wise distances of a set of points, then $\Lambda$ contains only positive and zero eigenvalues. Otherwise, there might be some negative eigenvalues, and classical scaling ignores them as error. Let $\Lambda_+$ be the matrix of positive eigenvalues and $Q_+$ the matrix of the corresponding eigenvectors, then the coordinate matrix $Z$ is calculated as:

$$Z = Q_+\Lambda_+^{1/2} \tag{7}$$

One point should be noted is if $\Lambda$ contains only positive and zero eigenvalues, then $Z$ will provide an exact reconstruction of $\Delta$. Otherwise, the distance matrix $\Delta$ will be an approximation. Another point is that the relative magnitudes of those eigenvalues in $\Lambda$ indicate the relative contribution of the corresponding columns of $Z$ in reproducing the original distance matrix $\Delta$. So, if $k'$ eigenvalues in $\Lambda$ are much larger than the rest, then the distance matrix based on the $k'$ corresponding columns of $Z$ nearly reproduces the original dissimilarity matrix $\Delta$. In this sense, we can reduce the

number of dimensions of $Z$ by choosing only the principle eigenvalues with only a small loss of information.

## 3 A New Algorithm

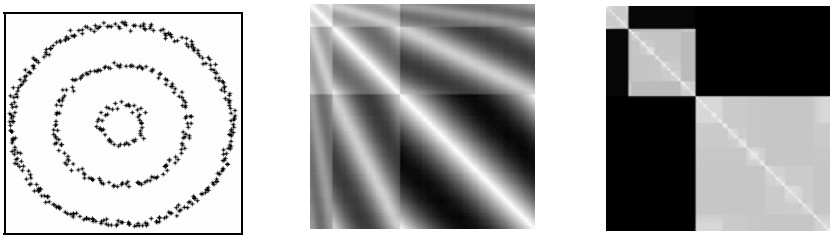
In this section, the details of our proposed algorithm are presented. With a good data representation, any simple clustering algorithm like K-means can be applied successfully. In order to achieve this goal, we first use the path-based dissimilarity measure to change the dissimilarities (or distances) between all pairs of objects. This transformation is performed once at the beginning on the whole dataset. As the path-based dissimilarity measure takes into account the connection relationships among the objects, this transformation will embed the cluster structure information into the new dissimilarity matrix. We then find a set of objects, or a new data representation, that reflects these new pair-wise dissimilarities by using a multi-dimensional scaling algorithm. After that, K-means is employed to do the clustering on this new data representation.

**Algorithm.** Path-based clustering using multi-dimensional scaling

- Input: $n \times m$ data matrix $X$, number of clusters $k$
- Algorithm:
  1. Compute the $n \times n$ pair-wise distance matrix $D$ from data matrix $X$
  2. Transform $D$ into $D'$ using path-based dissimilarity measure
  3. Perform classical MDS on $D'$ to get a $n \times m'$ new data matrix $Z$
  4. Identify $k'$ - the number of principle dimensions of $Z$.
     Let $Y$ the $n \times k'$ matrix of $k'$ first columns of $Z$.
  5. Apply K-means on $n$ rows $y_i$ of $Y$ to get a partition $C_1, C_2, \ldots C_k$
- Output: Clusters $A_1, \ldots A_k$ with $A_i = \{x_j | y_j \in C_i\}$

In step 2, the path-based distances between all pairs of objects are computed using an algorithm similar to the algorithm of Floyd [14]. In step 3, the classical multi-dimensional scaling algorithm will return a configuration of points whose pair-wise distances approximate the new distance matrix $D'$. Because of this, the number of dimensions $m'$ of the MDS configuration $Z$ may be very large. However, only some of them are important and the distance matrix can be reconstructed using only these principle dimensions with very small error. So the number of principle dimensions needs to be identified and only those important ones should be used to represent a new data matrix. This can easily be done by an analysis on the eigenvalues of the scalar product matrix which is also returned by classical multi-dimensional scaling algorithm. In our experiments, we choose the number of dimensions as the number of clusters minus one (as done in spectral clustering) and the results showed that this is a suitable setting.

One of the advantages of our algorithm is that it operates based on the dissimilarity matrix which often arises naturally from a data matrix. A distance metric is used to calculate the dissimilarity between objects. Among many distance metrics available,



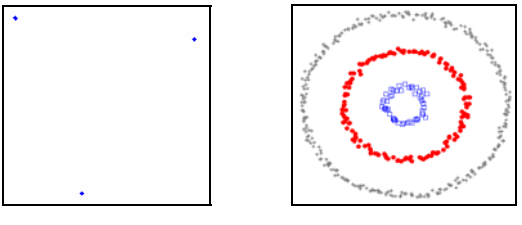
(a) (b) (c)

**Fig. 1.** Distance matrices of three-circle dataset: (a) input data; (b) original distance matrix; (c) transformed distance matrix

Euclidean distance is a simple and popular metric that proves successful in many applications. Another advantage is that our algorithm is more efficient than the original path-based clustering algorithm [9-11] as it calculates the path-based distances of all pairs of objects only one time at the beginning. Moreover, the algorithm requires only one parameter from the user, the number of clusters. Finally, experimental results prove its strong ability to detect complex structures inherent in the data.

To show the effectiveness of our algorithm, we analyze the clustering process on a commonly used three-circle synthetic dataset. Fig. 1 shows the original dataset in two-dimensional space and its distance matrices before and after the path-based transformation. The distance matrices are displayed on gray scale images with white for *0* and darker for higher values. To emphasize the utility of the path-based dissimilarity, the points in this example are ordered so that the points within each circle form a block of successive pixels on the image. It is shown that after the transformation, the distance matrix is nearly block-diagonal with each cluster corresponding to a block. This indicates that applying path-based dissimilarity transformation enhances the cluster structures on the distance matrix.

After performing path-based dissimilarity transformation, the classical MDS is performed on the transformed distance matrix. The new data representation $Y$ with two principle dimensions is obtained and plotted in Fig. 2(a). From this plot, we can see that the data points of the original dataset are transformed into this new space and form three very compact clusters, each of which represents a circle of the original dataset. With this new representation, simple clustering algorithm like K-means can easily detect and correctly identify the three circles as shown in Fig. 2(b).



(a) (b)

**Fig. 2.** Results obtained on three-circle dataset: (a) new data representation on two-dimensional space; (b) K-means result on three-circle dataset

After the path-based transformation, the distance between data points within a cluster becomes very small compared to those of different clusters. This obeys the rule that the points belonging to the same cluster should be similar while points from different clusters should be dissimilar in the new space, which is clearly shown by the results in Fig. 2.

## 4 Experiment Results

In order to evaluate the performance of our algorithm, a number of experiments on both synthetic and real datasets were performed. In all these experiments, the results of our algorithm were compared with those of two popular clustering methods, K-means and spectral clustering. With spectral clustering, we used Ncut normalization on the Laplacian. Also, to avoid manually setting the scaling parameter, we employed the local scaling setting proposed in [15] as it has been shown that this setting gives better results than a global scaling parameter.

### 4.1 Results on Synthetic Datasets

To demonstrate the power of our algorithm on separable data, the first comparison was performed on four synthetic datasets with different data structures: three-circle, face, two-moon, and two-spiral datasets. The results are presented in Fig. 3. We can see that K-means is unable to identify any of the clusters because the clusters are elongated in nature. Spectral clustering is able to identify the clusters in two of the data sets (three-circle and two-moon). Interestingly, our algorithm is able to identify

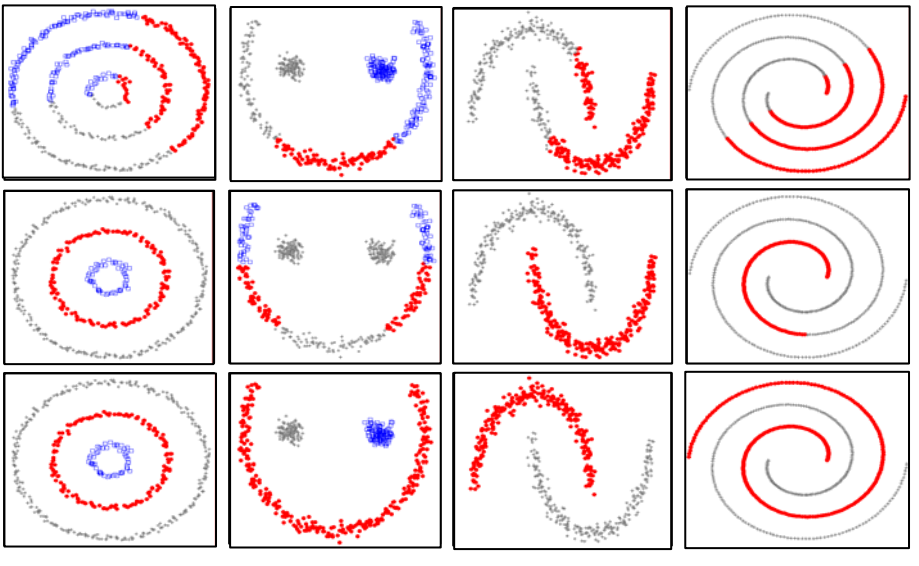
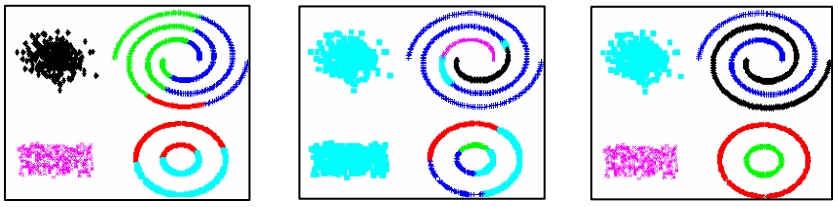


**Fig. 3.** Results on 4 synthetic datasets: first row: results of K-means; second row: results of spectral clustering; third row: results of our algorithm



(a) (b) (c)

**Fig. 4.** Results on a complex synthetic dataset: (a) result of K-means; (b) result of spectral clustering; (c) result of our algorithm

each cluster in all of the data sets. To understand these results, we examined the path-based data representation of each dataset and learned that each cluster in the original space forms a very compact cluster in the new space (similar to the case of three-circle dataset explained above). With such good representation, the data space can easily be partitioned using K-means and produce correct results as presented.

The second comparison was performed on a more complex synthetic dataset, which consists of six clusters: two circles, two spirals, a rectangular, and a Gaussian-shaped cluster. The results obtained by K-means, spectral clustering and our algorithm are shown in Fig. 4, which indicate that neither K-means nor spectral clustering can correctly identify all clusters in this dataset. On the contrast, our algorithm detected all the clusters despite of their differences in shape and size.

### 4.2 Results on Real Datasets

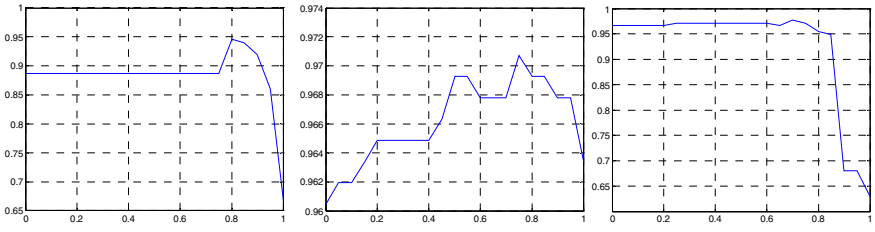
In order to test the performance on real data, we performed a comparison on three commonly used datasets from UCI repository [16]: Iris, Breast-cancer, and Wine. The descriptions of these datasets are presented in Table 1. To measure the performance of each clustering algorithm, the accuracy metric [17] was used. The results are summarized and presented on the same table.

We can see that K-means provides high accuracy on each of the data sets, implying that each of the data sets contain radial clusters. We can also see that our path-based algorithm provides high accuracy on the Breast-cancer data set implying that is has two distinct clusters. Our algorithm gave an accuracy of approximately 2/3 for the other two data sets, which leads us to believe that one cluster is distinct and the other two are overlapping. The overlapping would cause our method to place both clusters in to one, giving an accuracy of 2/3. This overlapping property can be easily seen when each data set is projected into a two-dimensional space.

To deal with the case of overlapping clusters, we will examine a mixed clustering method. We define the new distance as a function of $\alpha$ ($0 \le \alpha \le 1$), original distance, and path-based distance as (8). With $\alpha = 0$, the result obtained is equal to the result of K-means while with $\alpha = 1$, the result is of our original algorithm. The remaining values of $\alpha$ give a weighted combination of K-means and our algorithm. The accuracies obtained on three datasets when $\alpha$ changes from *0* to *1* are displayed in Fig. 5.

**Table 1.** UCI data descriptions and clustering results

| Dataset | Descriptions | | | Accuracy (%) | | |
|---|---|---|---|---|---|---|
| | # of instances | # of attributes | # of classes | K-means | Spectral | Our algorithm |
| Iris | 150 | 4 | 3 | 89.33 | **90.67** | 66.67 |
| Breast-cancer | 683 | 9 | 2 | 96.04 | 69.25 | **96.34** |
| Wine | 178 | 13 | 3 | **96.63** | **96.63** | 62.92 |



(a) (b) (c)

**Fig. 5.** Accuracies obtained on three datasets when α changes: (a) on Iris dataset; (b) on Breast-cancer dataset; (c) on Wine dataset

$$d_{ij}(\alpha) = (1-\alpha) \times dis(x_i, x_j) + \alpha \times pbdis(x_i, x_j) \tag{8}$$

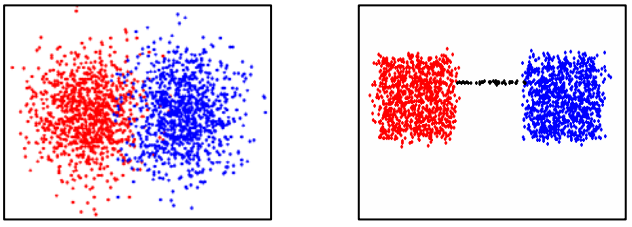
We can see that the mixing has increased the accuracy of our algorithm by allowing it to identify overlapped clusters. In two of the three data sets, we can see that the mixing has produced an increase over K-means as well, implying that K-means has also profited from our algorithm identifying the clearly separable clusters.

By applying the mixed algorithm to the synthetic data in Fig. 3 we obtain the greatest accuracy at $\alpha = 1$ (when using our algorithm only), with a descent in accuracy as α is reduced to 0. This result is obvious since our algorithm is suited for separable clusters, and each of the synthetic data sets is clearly separable in Euclidean space.

## 5 Discussions

As part of our analysis, we examine two cases where our algorithm cannot separate clusters, but K-means is able to provide high accuracy. The first case is when there are overlapping regions between clusters and the second is when separated clusters are connected by small bridges as shown in Fig. 6.

In these cases, our algorithm will consider the data as one cluster since the distances between any two points in different clusters is small due to the path-based transformation. K-means identifies these clusters with only small error. However, these are difficult cases for the clustering task in general. If we removed the class information from the data (remove the color from Fig. 6), there is no reason why we should identify the two cases shown as two clusters. There is also no reason why the



(a) (b)

**Fig. 6.** Examples of two cases when the algorithm fails: (a) two Gaussian clusters with an overlapping region; (b) two separated clusters connected by a small bridge

data could not contain one or three clusters. The beauty of our path-based clustering algorithm is that it identifies each clearly separable cluster (independent of shape and structure) and makes no assumptions about inseparable data.

## 6 Conclusions

In this paper, we have proposed a new clustering method that is capable of capturing complex structures in data. With the combination of the path-based dissimilarity measure and multi-dimensional scaling, we can produce a good data representation for any given dataset, which makes it possible to detect clusters of different shapes that are non-linearly separable in the input space.

We showed that our path-based clustering method clearly identifies separable clusters. We also showed that our algorithm is unable to identify inseparable clusters, but also explained that identifying clusters in such data is in the eye of the beholder. This behavior makes our path-based clustering algorithm produce model clusters; that follow the definition of a cluster.

## References

1. Jain, A., Law, M.: Data Clustering: A User's Dilemma. In: Pal, S.K., Bandyopadhyay, S., Biswas, S. (eds.) PReMI 2005. LNCS, vol. 3776, pp. 1–10. Springer, Heidelberg (2005)
2. Jain, A.: Data Clustering: 50 Years Beyond K-means. In: Daelemans, W., Goethals, B., Morik, K. (eds.) ECML PKDD 2008, Part I. LNCS (LNAI), vol. 5211, pp. 3–4. Springer, Heidelberg (2008)
3. Dhillon, I.S., Guan, Y., Kulis, B.: Kernel k-means: spectral clustering and normalized cuts. In: 10th Int. Conf. on Knowledge Discovery and Data Mining, pp. 551–556. ACM, New York (2004)
4. Scholkopf, B., Smola, A., Muller, K.: Nonlinear component analysis as a kernel eigenvalue problem. J. Neu. Com. 10, 1299–1319 (1998)
5. Von Luxburg, U.: A tutorial on spectral clustering. J. Sta. and Com. 17, 395–416 (2007)
6. Meila, M., Shi, J.: A random walks view of spectral segmentation. In: International Conference on AI and Statistics (2001)

7. Ng, A., Jordan, M., Weiss, Y.: On spectral clustering: Analysis and an algorithm. In: Adv. in Neu. Inf. Pro. Sys. 14, vol. 2, pp. 849–856 (2001)
8. Shi, J., Malik, J.: Normalized cuts and image segmentation. IEEE Trans. on Pattern Analysis and Machine Intelligence 22, 888–905 (2000)
9. Fischer, B., Zoller, T., Buhmann, J.: Path based pairwise data clustering with application to texture segmentation. In: Figueiredo, M., Zerubia, J., Jain, A.K. (eds.) EMMCVPR 2001. LNCS, vol. 2134, pp. 235–250. Springer, Heidelberg (2001)
10. Fischer, B., Buhmann, J.M.: Path-based clustering for grouping of smooth curves and texture segmentation. IEEE Trans. on Pattern Analysis and Machine Intelligence 25, 514–519 (2003)
11. Fischer, B., Buhmann, J.M.: Bagging for path-based clustering. IEEE Trans. on Pattern Analysis and Machine Intelligence 25, 1411–1415 (2003)
12. Chang, H., Yeung, D.: Robust path-based spectral clustering. J. Pat. Rec. 41, 191–203 (2008)
13. Borg, I., Groenen, P.: Modern multidimensional scaling: Theory and applications. Springer, Heidelberg (2005)
14. Cormen, T.H., Leiserson, C.E., Rivest, R.L.: Introduction to Algorithms. MIT Press, Cambridge (1989)
15. Zelnik-Manor, L., Perona, P.: Self-tuning spectral clustering. J. Adv. in Neu. Inf. Pro. Sys. 17, 1601–1608 (2004)
16. UCI Machine Learning Repository, `http://www.ics.uci.edu/~mlearn/MLRepository.html`
17. Cai, D., He, X., Han, J.: Document clustering using locality preserving indexing. IEEE Trans. on Knowledge and Data Engineering 17, 1624–1637 (2005)