A Scale Dependent Model for Clustering by Optimization of Homogeneity and Separation *

Andy M. Yip † Tony F. Chan† Tarek P. Mathew ‡

Abstract

We present a model for clustering a set of high dimensional data into subsets of homogenous clusters which are well-separated from each other. A novel feature of our model is that it allows the user to directly control the scale of the clusters. This is realized by formulating the clustering problem as an optimization problem whose objective function combines two measures of the quality of a clustering, namely *homogeneity* and *separation*, and a parameter which controls the scale. As an illustration to the use of our general framework, an application on clustering data based on pair-wise similarity measured by the (uncentered) Pearson correlation coefficient (cosine of the angle between two data vectors in an Euclidean space) is presented. In this case, for a dataset of size n in a p-dimensional space, our algorithm will output a clustering in $O(npd + mpd^2)$ floating point operations while typical hierarchical and partitioning algorithms require $O(n^2p)$ and $O(npkm)$ respectively. Here d is the number of clusters determined by our algorithm, k is the number of clusters specified by the user, and m is the number of iterations. Experimental results on synthetic, biological and textured image data are presented to demonstrate the usefulness of the proposed model.

Keywords: clustering algorithms, unsupervised learning, optimization, gene expression analysis, image segmentation, texture.

---

*This work was partially supported by grants from the NSF under contract DMS-9973341, the ONR under contract N00014-96-1-0277 and the NIH under contract MH65166.

†Department of Mathematics, University of California, 405 Hilgard Avenue, Los Angeles, CA 90095-1555. Email: {*mhyip,chan*}*@math.ucla.edu*

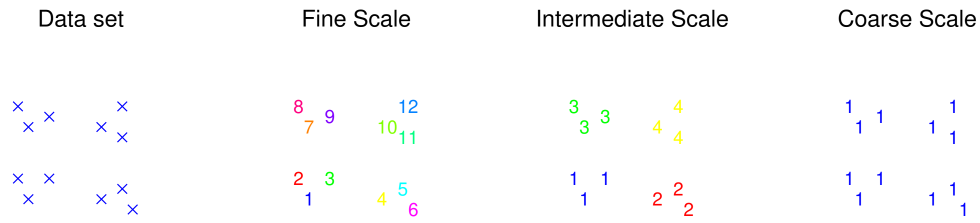
‡Email:*tmathew@mindspring.com*

1 Introduction

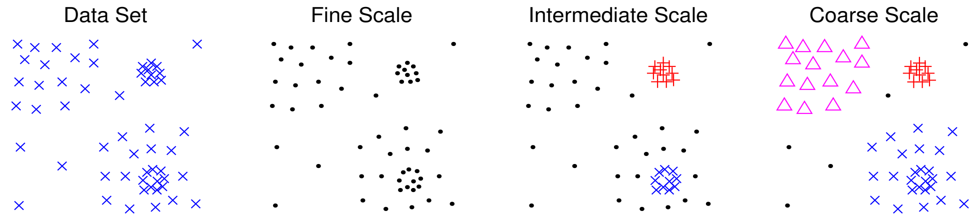
Large samples of data are routinely collected in diverse disciplines within applied and experimental sciences. The data of interest may vary from time series recordings of economic indices, geophysical or atmospheric variables, gene expression levels in cells, to pixel values of images. Clustering aims at discovering groups of similar data and interesting patterns for further processing and analysis. See [8, 10] for surveys of general purpose clustering algorithms and their applications.

Generally, clustering is not a well-defined problem unless we specify *a priori* what kind of clustering is desirable. Given a dataset, many clusterings are possible. Indeed, different clustering algorithms usually result in different clusterings. This is not surprising as different algorithms assume different *a priori* knowledge about the clusters. In partitioning algorithms such as the k-means algorithm [12], a good clustering is chosen as one which minimizes the sum of all distances between each data vector and the centroid of the cluster to which the data vector belongs. In the other words, the k-means algorithm is based on the belief that clusters should be as compact as possible. It works well when the clusters are well-separated from each other or when the clusters are of similar size and of approximately spherical shape. DBSCAN [4], a density-based algorithm, defines a cluster by the notion of density-reachability. A basic assumption is that clusters are dense regions which are separated from each other such that points within the same cluster are density-reachable to each other while points in different clusters are not. Thus, clusters can be found by identifying dense regions. CHAMELEON [11], a hierarchical algorithm, is capable of identifying clusters of various sizes and densities and of arbitrary shapes. Its merging decisions are based on the relative closeness and relative inter-connections of the clusters to be merged. It basically assumes that clusters are separated by a distance which is large compared with distances between neighborhood points within individual clusters and are of low relative inter-connectivity.

Despite the different assumptions in the above-mentioned clustering algorithms, a common assumption is that datasets have natural clusterings. In the two dimensional examples in [4, 11], natural clusterings are identified by human perception. However, natural clusterings are not always clearly defined. For example, in the dataset $\{1/n\}_{n=1}^{100}$ in one-dimension Euclidean space, it is unclear what forms a natural clustering based on human perception. In this paper, however, we take a different point of view that clusterings can be defined by a *scale* (resolution) which is intrinsic to the data. If we look at the data at different scales, we should obtain different clusterings. Thus, our *Scale Dependent Model* (SDM) defines clusterings in a dynamic way. Figure 1 illustrates the notion of scale. In Figure 1(a), each point forms an individual cluster at the fine scale. At the intermediate scale, each cluster consists of a group of three. When we look at the data at the coarse



(a)



(b)

Figure 1: Clusterings at various scales. (a) The dataset forms a hierarchical structure in which three clusterings are possible. (b) A black dot represents a singleton. We observe that for each clustering at certain scale, points that do not form a cluster at that scale are identified as singletons.

scale, all data vectors collapse to a single cluster. Figure 1(b) illustrates the same idea in terms of noise. In the figure, each dot corresponds to a cluster having only one point — we may regard these singletons as outliers. At the fine scale, every data vector is a cluster itself. At the intermediate scale, two dense clusters are identified. Finally, at the coarse scale, three clusters are identified and five points are considered as outliers. In practice, clusterings at different scales may correspond to groupings at different levels of abstraction.

Our goal in this paper is to address the problem of scale directly in our formulation, so that users can control the scale of a clustering directly through a scale parameter. The notion of scale also arises naturally in image segmentation. Our model is motivated by the work of Chan and Vese [1] where different partitions of an image can be obtained through different choices of a scale parameter. In their model, the scale parameter serves as a measure of trade-off between surface area and volume of segments. In total variation based image restoration problem, Strong and Chan gave explicitly the definition of the scale of a piecewise constant image feature Ω in [16],

$$\text{scale} \;:= \frac{|\Omega|}{|\partial\Omega|}$$

where $|\Omega|$ and $|\partial\Omega|$ are the area and the length of the boundary of Ω respectively. The main result of Strong and Chan is that the difference δ between the intensities of a feature of the original and the restored images is inversely proportional to the scale of the feature

$$\delta = \frac{\mu}{\text{scale}}.$$

Here μ is said to be a scale parameter which directly controls the intensity of a feature in the restored image. This means that intensities of features of the same scale are changed by the same amount. In addition, loosely speaking, features of very small scale relative to μ (noise) are essentially removed while other features of larger scale are preserved. It is this concept of scale in image restoration problems that we seek to carry over to the clustering problem, i.e., points that do not form a cluster in a user specified scale are identified as noise, whereas points that form a cluster in that scale are grouped together.

To measure the quality of a clustering, we employ the notions of *homogeneity* and *separation*. These two measures have been widely used in the data mining community [14, 15, 3, 17, 2]. For example, in [15], Sharan and Shamir employed these measures to assess the accuracy of their algorithm and used these measures as indicators to compare their results with other clustering algorithms. In [17, 2], Zhao and Karypis, and Dhillon and Modha studied clustering of text documents by means of optimizing functions of these two measures while keeping the number of clusters fixed. While Sharan and Shamir compared clustering results by measuring homogeneity and separation, a natural question is whether we can obtain homogeneous and well-separation clusters by optimizing both quantities. The answer to this question is complicated by the fact that homogeneity and separation are conflicting measures; optimizing one will be destroying the other. An "external force" must be imposed to balance these two repelling forces. In [17, 2], an external force is exerted by specifying the number of clusters—*a priori* knowledge from the user. So, we may see that a central question is that, given a suitable definition for homogeneity and separation measures, how to combine them and incorporate *a priori* knowledge in a meaningful way such that both measures can optimized simultaneously within the given *a priori* specifications. Our approach formulates a clustering problem as an optimization problem whose objective function combines these two measures and a scale parameter so that a combination of homogeneity and separation is directly optimized while keeping the resulting clusters at the user specified scale. Moreover, the scale parameter plays a similar role as in image segmentation — it determines features to be preserved or removed. As an illustration to the use of our general framework, an application on clustering data based on pair-wise similarity measured by the (uncentered) Pearson correlation coefficient (cosine of the angle between two data vectors in an Euclidean space) is presented. Such a similarity measure is appropriate when we are only interested in the angle between each pair of data vectors. For instance, when clustering text documents [17], we may only be interested in the distribution of the terms in a document but not its total number of terms (documents of similar contents may have very different length); when clustering gene expression data [15, 5], relative change of gene expression levels may be more interesting than the actual amount of change (co-expressed genes may have very different expression levels). In this case, for a dataset consisting of n points in a p-dimensional space, our algorithm will output a clustering in $O(npd + mpd^2)$ floating point operations while typical hierarchical and partitioning algorithms require $O(n^2p)$ and $O(npkm)$ respectively. Here d is the number of clusters determined by our algorithm, k is the number of clusters specified by the user and m is the number of iterations. In our experiments, m is usually less than 5.

One can also obtain a sequence of clusterings by using different choices of k with the k-means algorithm or different choices of threshold value in the dendrogram produced by a hierarchical clustering algorithm. However, clusters obtained may not be of the same scale, see Examples 3 and 4 in §5. On the other hand, our approach differs from the density-based algorithm DBSCAN in the sense that our objective function scores a clustering using global information across different clusters while the DBSCAN algorithm constructs clusters based on local density only (c.f. Example 6 in §5).

It is worth noting that the choice of an appropriate clustering algorithm may depend on what a user is looking for and how much *a priori* knowledge is available. For example, if the number of clusters is known and if the user is looking for clusters of approximately spherical in shape, then the k-means algorithm may be a reasonable choice. If the user is looking for clusters of the same scale, then our SDM model may be applied.

The paper is outlined as follows. In §2, we discuss the notions of homogeneity and separation. We also give definitions of average homogeneity and average separation as measures of overall homogeneity and separation of a clustering. In §3, we present our optimization formulation of clustering problems and a heuristic algorithm for the optimization problem. In §4, we study an application of our model to data whose pair-wise similarity is measured by the Pearson correlation coefficients. Experimental results on synthetic and textured image data are presented in §5. In §6, we discuss how to adapt the objective function to specific applications. An example of clustering gene expression data using an adapted objective function is also presented. Finally, concluding remarks are given in §7.

2 Measuring the Quality of a Clustering

A good measure of the quality of a clustering is a key ingredient in a clustering algorithm. Intuitively, a desirable clustering should consist of homogeneous and well-separated clusters. Therefore, it is natural to define homogeneity within a cluster and separation between different clusters and to use them to measure the quality of a clustering.

In this section, we discuss how to define the homogeneity and the separation of a clustering given a definition of the homogeneity H of individual cluster and the separation S between pairs of clusters. We then present some guidelines for defining H and S.

We first fix some notations that will be used throughout the paper. Let $\{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n\}$ be a given set of data in $\mathbb{R}^p$. A clustering of a dataset into d clusters is denoted by $\mathcal{C}^{(d)} = \{\mathcal{C}_1, \cdots, \mathcal{C}_d\}$. The number of points in a cluster $\mathcal{C}_i$ and the number of points in a clustering $\mathcal{C}^{(d)}$ are denoted by $|\mathcal{C}_i|$ and $|\mathcal{C}^{(d)}| = \sum_{i=1}^{d} |\mathcal{C}_i|$ respectively. Homogeneity of an individual cluster $\mathcal{C}_i$ is denoted by $H(\mathcal{C}_i)$. Separation between a pair of clusters $\mathcal{C}_i$ and $\mathcal{C}_j$ is denoted by $S(\mathcal{C}_i, \mathcal{C}_j)$. The homogeneity and the separation of a clustering will be defined as a function of the homogeneity of individual clusters and a function of the separation of individual clusters respectively, i.e.,

$$\begin{aligned} H(\mathcal{C}^{(d)}) &= H(\mathcal{C}_1, \cdots, \mathcal{C}_d) := f(H(\mathcal{C}_1), \cdots, H(\mathcal{C}_d)) \\ S(\mathcal{C}^{(d)}) &= S(\mathcal{C}_1, \cdots, \mathcal{C}_d) := g(S(\mathcal{C}_1, \mathcal{C}_2), \cdots, S(\mathcal{C}_{d-1}, \mathcal{C}_d)) \end{aligned}$$

for some functions f and g. In most applications, we choose the functions f and g to be weighted average functions. Average homogeneity of a set of d clusters $\mathcal{C}^{(d)} = \{\mathcal{C}_1, \cdots, \mathcal{C}_d\}$ is defined as the weighted average of the homogeneity of all clusters

$$H_{\mathrm{WA}}(\mathcal{C}^{(d)}) = H_{\mathrm{WA}}(\mathcal{C}_1, \cdots, \mathcal{C}_d) := \frac{1}{\sum_{i=1}^{d} |\mathcal{C}_i|} \sum_{i=1}^{d} |\mathcal{C}_i| H(\mathcal{C}_i).$$

Average separation of a set of d ($d > 1$) clusters $\mathcal{C}^{(d)} = \{\mathcal{C}_1, \cdots, \mathcal{C}_d\}$ is defined as the weighted average of all pair-wise separation between two clusters

$$S_{\mathrm{WA}}(\mathcal{C}^{(d)}) = S_{\mathrm{WA}}(\mathcal{C}_1, \cdots, \mathcal{C}_d) := \frac{1}{\sum_{i<j} |\mathcal{C}_i||\mathcal{C}_j|} \sum_{i<j} |\mathcal{C}_i||\mathcal{C}_j| S(\mathcal{C}_i, \mathcal{C}_j).$$

Here the subscript "WA" denotes "weighted average".

An appropriate choice of definition of $H(\cdot)$ and $S(\cdot, \cdot)$ is problem dependent. Our SDM allows some flexibility in defining $H(\cdot)$ and $S(\cdot, \cdot)$. However, they should be defined such that H_{WA} and S_{WA} possess the following properties.

1. $H_{\mathrm{WA}}(\mathcal{C}) \geq 0$ should be maximized when each cluster in $\mathcal{C}$ is a singleton.

2. $S_{\mathrm{WA}}(\mathcal{C}) \geq 0$ should be maximized when every data vector belongs to the same cluster.

3. There must be a trade-off between maximizing $H_{\mathrm{WA}}(\mathcal{C})$ and maximizing $S_{\mathrm{WA}}(\mathcal{C})$.

For example, in Figure 1(a), the H_{WA} (respectively S_{WA}) value of the clustering in the fine scale is the largest (respectively smallest) while that in the coarse scale is the smallest (respectively largest). To name a few, some widely used definitions of H and S are as follows,

$$\begin{aligned} H(\mathcal{C}_i) \quad &: \quad \frac{1}{\binom{|\mathcal{C}_i|}{2}} \sum_{\mathbf{x},\mathbf{y}\in\mathcal{C}_i} s(\mathbf{x},\mathbf{y}); \quad \max_{\mathbf{x},\mathbf{y}\in\mathcal{C}_i} s(\mathbf{x},\mathbf{y}); \quad \frac{1}{|\mathcal{C}_i|}\sum_{\mathbf{x}\in\mathcal{C}_i} s(\mathbf{x},\mathbf{m}(\mathcal{C}_i)); \quad \frac{1}{|\mathcal{C}_i|}\sum_{\mathbf{x}\in\mathcal{C}_i} d_{\mathrm{knn}}(\mathbf{x},\mathcal{C}_i) \\ S(\mathcal{C}_i,\mathcal{C}_j) \quad &: \quad \frac{1}{|\mathcal{C}_i||\mathcal{C}_j|} \sum_{\mathbf{x}\in\mathcal{C}_i,\mathbf{y}\in\mathcal{C}_j} s(\mathbf{x},\mathbf{y}); \quad \min_{\mathbf{x}\in\mathcal{C}_i,\mathbf{y}\in\mathcal{C}_j} s(\mathbf{x},\mathbf{y}); \quad s(\mathbf{m}(\mathcal{C}_i),\mathbf{m}(\mathcal{C}_j)) \end{aligned}$$

where $\mathbf{m}(\mathcal{C}_i)$ is the mean vector of $\mathcal{C}_i$. In the above definitions, $s(\mathbf{x},\mathbf{y})$ is a similarity (dissimilarity) measure between $\mathbf{x}$ and $\mathbf{y}$ and $d_{\mathrm{knn}}(\mathbf{x},\mathcal{C}_i)$ is a distance which depends on the distance between $\mathbf{x}$ and each of its k-nearest neighborhoods in $\mathcal{C}_i$. If the data are in an Euclidean space $\mathbb{R}^n$, then $s(\mathbf{x},\mathbf{y})$ may be chosen as the Minkowski distance $\|\mathbf{x}-\mathbf{y}\|_p$. If the data are in the unit sphere of $\mathbb{R}^n$, then $s(\mathbf{x},\mathbf{y})$ may be chosen as the Pearson correlation coefficient $\mathbf{x}\cdot\mathbf{y}/(\|\mathbf{x}\|\|\mathbf{y}\|)$ or the Minkowski distance. In addition to the above commonly used ones, many other definitions of H and S are possible. Relative closeness may be used as a measure of separation, see [11]. To deal with clusters of complicated shape, several vectors may be used to represent a cluster instead of its mean vector alone [7]. We remark that in the above definitions, when the Minkowski distance is used, the value $H(\mathcal{C}_i)$ (and hence $H_{\mathrm{WA}}(\mathcal{C})$) may actually decrease as $\mathcal{C}_i$ becomes tighter. In order to make H increase as $\mathcal{C}_i$ becomes tighter, we may simply replace H by the composition of a decreasing function with H.

3 Mathematical Formulation

3.1 Formulation

We now formulate the selection of an optimal clustering in the scale dependent model.

Let $\mu \geq 0$ be a fixed number (scale parameter). Let $h(x,y,z)$ be a function which is increasing both in x, in y and in z for $x,y,z \geq 0$. The scale dependent model seeks a clustering $\mathcal{C}^*$ such that

$$h(H_{\mathrm{WA}}(\mathcal{C}^*), S_{\mathrm{WA}}(\mathcal{C}^*), \mu) = \max_{\mathcal{C}} h(H_{\mathrm{WA}}(\mathcal{C}), S_{\mathrm{WA}}(\mathcal{C}), \mu).$$

Here the maximization is taken over all possible clustering $\mathcal{C}$ of the dataset. In the sequel, we employ the function $h(x, y, z) = x + zy$, yielding that an optimal choice of clustering $\mathcal{C}^*$ satisfies

$$\mathcal{C}^* = \arg\max_{\mathcal{C}} J_\mu(\mathcal{C}) \qquad \text{for } \mu \geq 0 \tag{1}$$

where $J_\mu(\mathcal{C}) \equiv H_{\mathrm{WA}}(\mathcal{C}) + \mu S_{\mathrm{WA}}(\mathcal{C})$. In this formulation, since H_{WA}, S_{WA} and μ are nonnegative, we observe that if $\mu \to 0$, then we are effectively maximizing H_{WA} and we allow S_{WA} to be small. The resulting clustering will tend to have many small clusters (singletons if $\mu = 0$). If $\mu \to \infty$, then we are effectively maximizing S_{WA} which results in a single large cluster. If we choose an intermediate μ, we obtain clusters at an intermediate scale. The following proposition says that our formulation produces clusterings of decreasing homogeneity and increasing separation as μ is increased.

Proposition 1 *Denote an optimal solution of (1) with respect to μ by $\mathcal{C}(\mu)$. Then, $H_{\mathrm{WA}}(\mathcal{C}(\mu))$ is a decreasing function of μ, and $S_{\mathrm{WA}}(\mathcal{C}(\mu))$ is an increasing function of μ.*

3.2 A Heuristic Algorithm

The optimization problem (1) is an NP-hard problem. To solve it efficiently, we use a two-stage algorithm which is greedy in nature. We call this algorithm as the SDM algorithm. In the first stage, we begin by assigning the first data vector as the first cluster. Then, we sequentially consider each unassigned point and decide whether to merge the point with an existing cluster or to create a new cluster on its own. The decision is based on the choice which maximizes the objective function. After all data vectors have been assigned a cluster label, we proceed to the second stage. In the second stage, we try to merge the clusters formed in the first stage. Again, the decision is based on the choice which maximizes the objective function. The algorithm works for any choice of $H(\cdot)$ and $S(\cdot,\cdot)$. The heuristic algorithm proceeds as in Figure 2.

The time complexity of stage 1 is $O(ncd)$ and that of stage 2 is $O(mcd^2)$ where d is the number of clusters determined by the algorithm, m is the number of passes of the **While**-loop and $c = c(n, p, d)$ is the number of floating operations required for evaluating the objective function. Thus, the total time complexity is $O(ncd + mcd^2)$.

When $H^{\mathrm{P}}_{\mathrm{WA}}$ and $S^{\mathrm{P}}_{\mathrm{WA}}$ (defined in §4) are used as measures of clustering quality, the objective function can be updated and evaluated effectively. In this case, we have $c = O(p)$. Hence, the total complexity reduces to $O(npd + mpd^2)$. In general, without an efficient mechanism to update and evaluate the objective function, c can be as large as $O(npd)$.

Input: $\{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n\}$, μ
Output: d, $\{\mathcal{C}_1, \cdots, \mathcal{C}_d\}$

Stage 1
Initialization: $d \leftarrow 1$, $\mathcal{C}_1 \leftarrow \{\mathbf{x}_1\}$
For $i = 2$ **to** n **do**:
  For $j = 1$ **to** d **do**:
    $f_j \leftarrow J_\mu(\mathcal{C}_1, \cdots, \mathcal{C}_j \cup \{\mathbf{x}_i\}, \cdots, \mathcal{C}_d)$
  EndFor
  $f_{d+1} \leftarrow J_\mu(\mathcal{C}_1, \cdots, \mathcal{C}_d, \{\mathbf{x}_i\})$
  $j_* = \arg\max_{1 \le j \le d+1} f_j$
  If $j_* \neq d + 1$ * merge $\{\mathbf{x}_i\}$ with $\mathcal{C}_{j_*}$ *\
    $\mathcal{C}_{j_*} \leftarrow \mathcal{C}_{j_*} \cup \{\mathbf{x}_i\}$
  Else * create a new cluster $\mathcal{C}_{d+1} = \{\mathbf{x}_i\}$ *\
    $\mathcal{C}_{d+1} \leftarrow \{\mathbf{x}_i\}$
    $d \leftarrow d + 1$
  EndIf
EndFor

Stage 2
CHANGE $\leftarrow 1$, $J_* \leftarrow f_{j_*}$
While CHANGE **do**:
  CHANGE $\leftarrow 0$
  For $i = d$ **to** 2 **do**:
    For $j = i - 1$ **to** 1 **do**:
      $f_j \leftarrow J_\mu(\mathcal{C}_1, \cdots, \mathcal{C}_j \cup \mathcal{C}_i, \cdots, \mathcal{C}_d)$
    EndFor
    $j_* = \arg\max_{1 \le j \le i-1} f_j$
    If $f_{j_*} > J_*$ * merge $\mathcal{C}_i$ with $\mathcal{C}_{j_*}$ *\
      $\mathcal{C}_{j_*} \leftarrow \mathcal{C}_{j_*} \cup \mathcal{C}_i$
      $J_* \leftarrow f_{j_*}$
      $\mathcal{C}_{j-1} \leftarrow \mathcal{C}_j$ for $j = i + 1, \cdots, d$
      Delete $\mathcal{C}_d$
      $d \leftarrow d - 1$, CHANGE $\leftarrow 1$
    EndIf
  EndFor
EndWhile

Figure 2: The SDM algorithm: a heuristic optimization algorithm.

We note that in many of our experiments, the **While**-loop in the second stage requires less than 5 iterations. This indicates that the greedy approach in the first stage identifies homogeneous and separated clusters well and that the second stage performs little refinement.

The proposed greedy algorithm is designed for speed and simplicity. It is heuristic and does not guarantee an optimal clustering. Moreover, once a data vector is merged with some cluster, it cannot be undone. When undoing merging is desired, cluster refinement techniques can be easily incorporated into the SDM algorithm to undo merging and improve the quality of a solution. However, in our experiments, the greedy based SDM algorithm usually produces clustering solutions of good quality.

Finally, we remark that the algorithm may depend on the ordering of data vectors. However, we may run the algorithm several times and pick the one with the largest objective function value. See also Example 9 in §5 for experiments on the robustness of the algorithm to the ordering of the data vectors.

4 An Application of the SDM

In this section, we apply our general framework of the SDM to a specific case when pair-wise similarity of data is measured by the Pearson correlation coefficient. We first present a definition of homogeneity and separation based on the Pearson correlation coefficients and its properties. Next, we study the role of the scale parameter and how it controls the scale of the resulting clustering through some simple but important examples.

4.1 Homogeneity and Separation based on the Pearson Correlation Coefficient

In this subsection, we study some properties of homogeneity and separation defined based on pair-wise similarity measured by the (uncentered) Pearson correlation coefficient. In the rest of the paper, we will just use the term "Pearson correlation coefficient" without mentioning it being uncentered. Such a similarity measure is found useful for clustering gene expression data [5, 9, 15] and clustering text documents [17]. An underlying assumption is that similarity of a pair of data vectors does not depend on the length of the data vectors. In the context of text documents, this corresponds to the fact that documents having similar contents may have very different lengths. Of course, if the length of data vectors should be taken into account, then the Pearson correlation coefficient may not be an appropriate choice.

Given 2 points $\mathbf{x}$ and $\mathbf{y}$ in $\mathbb{R}^p$. We can measure their similarity using the Pearson correlation coefficient

$$\rho^{\mathrm{P}}(\mathbf{x},\mathbf{y}) = \frac{\mathbf{x}^T\mathbf{y}}{\|\mathbf{x}\|\|\mathbf{y}\|}.$$

It is clear that $-1 \le \rho^{\mathrm{P}}(\mathbf{x},\mathbf{y}) \le 1$. The superscript "P" denotes the use of the Pearson correlation coefficient. A correlation coefficient of 1 indicates a maximal similarity between $\mathbf{x}$ and $\mathbf{y}$ while that of -1 indicates a minimal similarity. We remark that in some applications, one may want to use $|\rho^{\mathrm{P}}|$ as a similarity measure when negatively correlated variables are as important as those positively correlated. Since correlation coefficients do not depend on the length of the data, we assume in the sequel that all the data are normalized to unit vectors in a preprocessing step. In the following, we denote the usual inner product in $\mathbb{R}^p$ by ρ, i.e., $\rho(\mathbf{x},\mathbf{y}) = \mathbf{x}^T\mathbf{y}$. For unit vectors $\mathbf{x}$ and $\mathbf{y}$, we have $\rho^{\mathrm{P}}(\mathbf{x},\mathbf{y}) = \rho(\mathbf{x},\mathbf{y})$. We define the representative or profile of a cluster $\mathcal{C}_i$ by its mean vector $\mathbf{m}(\mathcal{C}_i) = \sum_{\mathbf{x}\in\mathcal{C}_i} \mathbf{x}/|\mathcal{C}_i|$ which captures the main characteristic of a cluster.

With ρ^{P} as a measure of pair-wise similarity, homogeneity of a cluster $\mathcal{C}_i$ is defined by the average of the cosine of the angle between each point in the cluster and the profile (mean)

$$H^{\mathrm{P}}(\mathcal{C}_i) := \frac{1}{|\mathcal{C}_i|}\sum_{\mathbf{x}\in\mathcal{C}_i} \rho^{\mathrm{P}}(\mathbf{x},\mathbf{m}(\mathcal{C}_i)) = \frac{1}{|\mathcal{C}_i|}\sum_{\mathbf{x}\in\mathcal{C}_i} \rho\left(\mathbf{x}, \frac{\mathbf{m}(\mathcal{C}_i)}{\|\mathbf{m}(\mathcal{C}_i)\|}\right) = \|\mathbf{m}(\mathcal{C}_i)\|. \tag{2}$$

The average homogeneity of a clustering $\mathcal{C}^{(d)}$ is then given by

$$H_{\mathrm{WA}}^{\mathrm{P}}(\mathcal{C}^{(d)}) := \frac{1}{\sum_{i=1}^{d}|\mathcal{C}_i|}\sum_{i=1}^{d}|\mathcal{C}_i|H^{\mathrm{P}}(\mathcal{C}_i) = \frac{1}{|\mathcal{C}^{(d)}|}\sum_{i=1}^{d}|\mathcal{C}_i|\|\mathbf{m}(\mathcal{C}_i)\|. \tag{3}$$

Since all the $\mathbf{x}_i$'s are unit vectors, we have $\|\mathbf{m}(\mathcal{C}_i)\| \leq 1$ for all i and hence $0 \leq H_{\mathrm{WA}}^{\mathrm{P}} \leq 1$. If all the data in each $\mathcal{C}_i$ are very close together, then $\|\mathbf{m}(\mathcal{C}_i)\| \to 1$, and $H_{\mathrm{WA}}^{\mathrm{P}} \to 1$. On the other hand, if all the data in $\mathcal{C}_i$ are scattered, then $\|\mathbf{m}(\mathcal{C}_i)\| \to 0$ and $H_{\mathrm{WA}}^{\mathrm{P}} \to 0$. We also note that the value of $H_{\mathrm{WA}}^{\mathrm{P}}$ is independent of the position of the profiles (depends only on their norms). Hence $H_{\mathrm{WA}}^{\mathrm{P}}$ does not depend on the separation between clusters.

Similarly, based on ρ^{P}, we define separation of a pair of clusters by

$$S^{\mathrm{P}}(\mathcal{C}_i,\mathcal{C}_j) := \frac{1}{2}\left[1-\rho^{\mathrm{P}}(\mathbf{m}(\mathcal{C}_i),\mathbf{m}(\mathcal{C}_j))\right] = \frac{1}{2}\left[1-\rho\left(\frac{\mathbf{m}(\mathcal{C}_i)}{\|\mathbf{m}(\mathcal{C}_i)\|},\frac{\mathbf{m}(\mathcal{C}_j)}{\|\mathbf{m}(\mathcal{C}_j)\|}\right)\right].$$

It is easy to see that $0 \leq S^{\mathrm{P}} \leq 1$. For $d > 1$, the average separation of $\mathcal{C}^{(d)}$ becomes

$$S_{\mathrm{WA}}^{\mathrm{P}}(\mathcal{C}^{(d)}) := \frac{1}{2}\left[1-\frac{1}{\sum_{i<j}|\mathcal{C}_i||\mathcal{C}_j|}\sum_{i<j}|\mathcal{C}_i||\mathcal{C}_j|\rho\left(\frac{\mathbf{m}(\mathcal{C}_i)}{\|\mathbf{m}(\mathcal{C}_i)\|},\frac{\mathbf{m}(\mathcal{C}_j)}{\|\mathbf{m}(\mathcal{C}_j)\|}\right)\right]. \tag{4}$$

For $d = 1$, we define $S_{\mathrm{WA}}^{\mathrm{P}}(\mathcal{C}^{(1)}) := 1$. If θ_{ij} is the angle between $\mathbf{m}(\mathcal{C}_i)$ and $\mathbf{m}(\mathcal{C}_j)$, then we also have

$$S_{\mathrm{WA}}^{\mathrm{P}}(\mathcal{C}_i,\mathcal{C}_j) \equiv \sin^2(\theta_{ij}/2). \tag{5}$$

Note that $0 \leq S_{\mathrm{WA}}^{\mathrm{P}} \leq 1$. We observe that the larger the $S_{\mathrm{WA}}^{\mathrm{P}}$-value, the more the separation between clusters. Again, the superscript "P" on $H_{\mathrm{WA}}^{\mathrm{P}}$ and $S_{\mathrm{WA}}^{\mathrm{P}}$ denotes the use of the Pearson correlation coefficient.

We remark that, in general, for a fixed dataset, a clustering with larger homogeneity value usually has smaller separation value and vice versa. This is a desirable property which makes our formulation in §3.1 successful.

In Figure 3, we show an example how $H_{\mathrm{WA}}^{\mathrm{P}}$ and $S_{\mathrm{WA}}^{\mathrm{P}}$ change with different clusterings. A dataset with 13 points is shown in Figure 3(a). We evaluated the values of $H_{\mathrm{WA}}^{\mathrm{P}}$ and $S_{\mathrm{WA}}^{\mathrm{P}}$ of 12 different clusterings. In Figure 3(b), 12 different clusterings are shown. Each row represents a clustering of the 13 points. Points having the same symbol belong to the same cluster. We note that in clustering 2, the 13-th point is merged with the 12-th point. In clustering 6, the 8-th point is merged with the points $\{5, 6, 7\}$. These mergings result in undesirable clusterings. Similarly unnatural mergings occur in clusterings 4 and 5. The $H_{\mathrm{WA}}^{\mathrm{P}}$ and $S_{\mathrm{WA}}^{\mathrm{P}}$ values of the 12 clusterings are shown in Figure 3(c) and 3(d) respectively. We note that both $H_{\mathrm{WA}}^{\mathrm{P}}$ and $S_{\mathrm{WA}}^{\mathrm{P}}$ of clustering 2 are less than that of clustering 3. Also, both $H_{\mathrm{WA}}^{\mathrm{P}}$ and $S_{\mathrm{WA}}^{\mathrm{P}}$ of clustering 7 are greater than that of clusterings 4, 5 and 6. Thus, the measures $H_{\mathrm{WA}}^{\mathrm{P}}$ and $S_{\mathrm{WA}}^{\mathrm{P}}$ favor clustering 3 compared to clustering 2 and favor clustering 7 compared to clusterings 4, 5 and 6. This is consistent with our goal of identifying homogeneous and well-separated clusters. We can also see that for other more natural clusterings (e.g. clusterings 1, 7, 8, 9, 12), there is a trade-off between maximizing $H_{\mathrm{WA}}^{\mathrm{P}}$ and maximizing $S_{\mathrm{WA}}^{\mathrm{P}}$.

According to our formulation in §3.1, clusterings 2, 4, 5 nor 6 will not be a maximizer of the objective function for any scale parameter $\mu \geq 0$. Thus our formulation would pick clusterings at various scales as well as rule out those undesirable clusterings.

4.2 The Scale Parameter

In this subsection, we discuss the role of μ in the clustering problem by considering two simple but important examples. In the first example, we illustrate that the model in (1) is capable of capturing clusters at various scales by adjusting the scale parameter μ. In the second example, we illustrate that, for a fixed μ, the clusters obtained are approximately of the same scale.

The dataset of the first example is shown in Figure 4. The figure shows 3 different clusterings of a dataset having 4 data vectors at 3 different scales. The coordinates of the data vectors are $\mathbf{x}_1 = (1, 0)$, $\mathbf{x}_2 = (\cos\epsilon, \sin\epsilon)$, $\mathbf{x}_3 = (\sin\sqrt{\epsilon}, \cos\sqrt{\epsilon})$ and $\mathbf{x}_4 = (0, 1)$. Our aim is to show that for any small $\epsilon > 0$, there exists a range of values of μ such that clustering 2 in Figure 4, which is the most interesting case, will maximize the objective function. To analyze the relationship between μ and ϵ, we calculate the objective function value for each clustering separately.

Fine Scale: $\mathcal{C}_1 = \{\mathbf{x}_1\}$ $\mathcal{C}_2 = \{\mathbf{x}_2\}$ $\mathcal{C}_3 = \{\mathbf{x}_3\}$ $\mathcal{C}_4 = \{\mathbf{x}_4\}$

$$\begin{aligned} H_1(\epsilon) &:= H_{\mathrm{WA}}^{\mathrm{P}}(\mathcal{C}_1, \mathcal{C}_2, \mathcal{C}_3, \mathcal{C}_4) = 1 \\ S_1(\epsilon) &:= S_{\mathrm{WA}}^{\mathrm{P}}(\mathcal{C}_1, \mathcal{C}_2, \mathcal{C}_3, \mathcal{C}_4) = \frac{1}{2} - \frac{1}{12}\left[\cos\epsilon + \sin\sqrt{\epsilon} + \sin(\epsilon + \sqrt{\epsilon}) + \sin\epsilon + \cos\sqrt{\epsilon}\right] \end{aligned}$$

Intermediate Scale: $\mathcal{C}_1 = \{\mathbf{x}_1, \mathbf{x}_2\}$ $\mathcal{C}_2 = \{\mathbf{x}_3\}$ $\mathcal{C}_3 = \{\mathbf{x}_4\}$

$$\begin{aligned} H_2(\epsilon) &:= H_{\mathrm{WA}}^{\mathrm{P}}(\mathcal{C}_1, \mathcal{C}_2, \mathcal{C}_3) = \frac{1}{4}\left[\sqrt{(1+\cos\epsilon)^2 + (\sin\epsilon)^2} + 2\right] \\ S_2(\epsilon) &:= S_{\mathrm{WA}}^{\mathrm{P}}(\mathcal{C}_1, \mathcal{C}_2, \mathcal{C}_3) = \frac{1}{2} - \frac{1}{10}\left[2\frac{(1+\cos\epsilon)\sin\sqrt{\epsilon} + \sin\epsilon\cos\sqrt{\epsilon} + \sin\epsilon}{\sqrt{(1+\cos\epsilon)^2 + (\sin\epsilon)^2}} + \cos\sqrt{\epsilon}\right] \end{aligned}$$

Coarse Scale: $\mathcal{C}_1 = \{\mathbf{x}_1, \mathbf{x}_2\}$ $\mathcal{C}_2 = \{\mathbf{x}_3, \mathbf{x}_4\}$

$$H_3(\epsilon) := H_{\mathrm{WA}}^{\mathrm{P}}(\mathcal{C}_1, \mathcal{C}_2) = \frac{1}{4}\left[\sqrt{(1+\cos\epsilon)^2 + (\sin\epsilon)^2} + \sqrt{(\sin\sqrt{\epsilon})^2 + (1+\cos\sqrt{\epsilon})^2}\right]$$

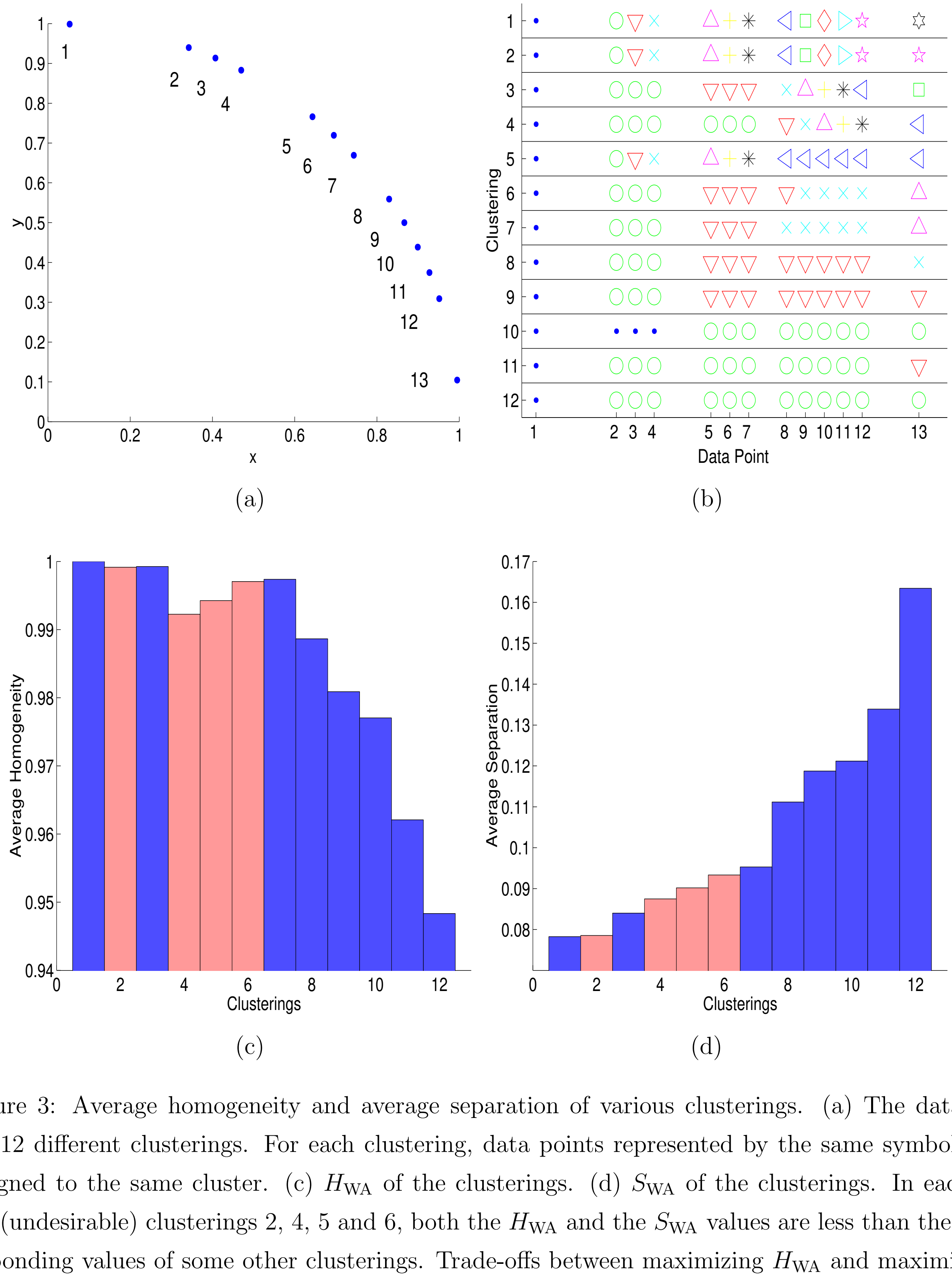


Figure 3: Average homogeneity and average separation of various clusterings. (a) The dataset. (b) 12 different clusterings. For each clustering, data points represented by the same symbol are assigned to the same cluster. (c) H_{WA} of the clusterings. (d) S_{WA} of the clusterings. In each of the (undesirable) clusterings 2, 4, 5 and 6, both the H_{WA} and the S_{WA} values are less than the corresponding values of some other clusterings. Trade-offs between maximizing H_{WA} and maximizing S_{WA} are observed among other more natural clusterings (e.g. clusterings 1,7,8,9,12).

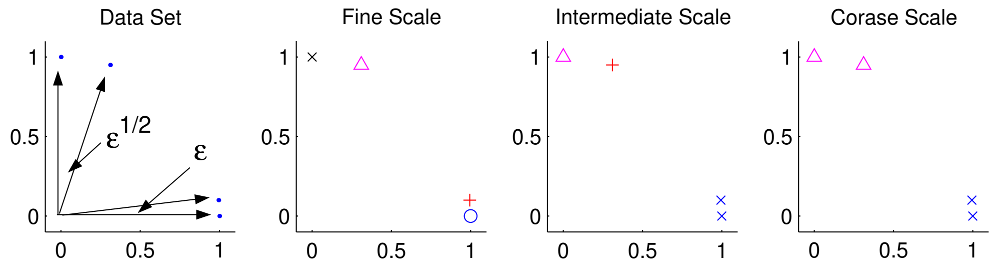


Figure 4: Clusterings at various scales. For each $\epsilon > 0$, there exists a range of values of μ such that the clustering at the intermediate scale is obtained by the SDM. Thus, the SDM is able to resolve clusterings at different scales.

$$S_3(\epsilon) \quad := \quad S^{\mathrm{P}}_{\mathrm{WA}}(\mathcal{C}_1, \mathcal{C}_2) = \frac{1}{2} - \frac{1}{2} \times \frac{(1+\cos\epsilon)\sin\sqrt{\epsilon} + \sin\epsilon(1+\cos\sqrt{\epsilon})}{\sqrt{(1+\cos\epsilon)^2 + (\sin\epsilon)^2}\sqrt{(\sin\sqrt{\epsilon})^2 + (1+\cos\sqrt{\epsilon})^2}}$$

We are now in position to analyze the relationship between μ and ϵ. By Taylor expansion, we obtain

$$\begin{aligned} \frac{H_1(\epsilon) - H_2(\epsilon)}{S_2(\epsilon) - S_1(\epsilon)} &= \frac{15}{16}\epsilon^2 + \frac{15}{32}\epsilon^{\frac{5}{2}} + O(\epsilon^3) \\ \frac{H_2(\epsilon) - H_3(\epsilon)}{S_3(\epsilon) - S_2(\epsilon)} &= \frac{5}{8}\epsilon + \frac{5}{16}\epsilon^{\frac{3}{2}} + \frac{295}{384}\epsilon^2 + \frac{215}{256}\epsilon^{\frac{5}{2}} + O(\epsilon^3). \end{aligned}$$

In summary, if $15\epsilon^2/16 \lesssim \mu \lesssim 5\epsilon/8 + 5\epsilon^{3/2}/16 + 295\epsilon^2/384$ (for ϵ small enough), then we have

$$H_2(\epsilon) + \mu S_2(\epsilon) > \max\{H_3(\epsilon) + \mu S_3(\epsilon), H_1(\epsilon) + \mu S_1(\epsilon)\}$$

as desired. This shows that our formulation can always identify three clusters (with $\mathbf{x}_1$ and $\mathbf{x}_2$ being merged) no matter how small ϵ is, for a suitable choice of μ. Note that the width of the corresponding interval for μ is of $O(\epsilon)$ — it does not go to zero quickly.

In the second example, a dataset with 4 groups of points X_i for $i = 1, \cdots, 4$ is used, see Figure 5. We study how the formulation in (1) obtains clusters of approximately the same scale by considering 3 different clusterings

$$\mathcal{C}^1 = \{X_1 \cup X_2, X_3 \cup X_4\}; \qquad \mathcal{C}^2 = \{X_1 \cup X_2, X_3, X_4\}; \qquad \mathcal{C}^3 = \{X_1, X_2, X_3, X_4\}.$$

For simplicity, we assume the following

- $|X_i| = c$ and $H^{\mathrm{P}}_{\mathrm{WA}}(X_i) = h$ for $i = 1, \cdots, 4$;
- $S^{\mathrm{P}}_{\mathrm{WA}}(X_1, X_2) = S^{\mathrm{P}}_{\mathrm{WA}}(X_3, X_4) = \sin^2\theta$ (c.f. equation (5));

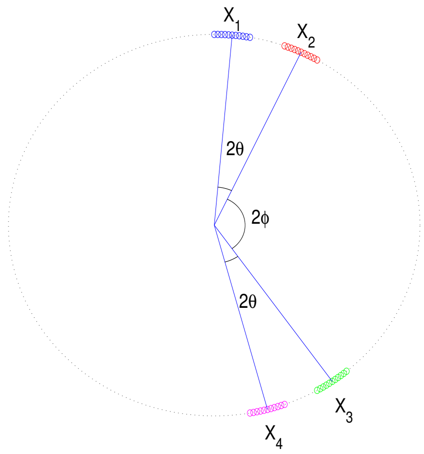


Figure 5: A dataset with 4 groups of points is used to show that, for a given μ, clusters of the same scale are obtained, i.e. X_1 is merged with X_2 if and only if X_3 is merged with X_4

- $0 < \theta \leq \pi/4$ and $\theta \leq \phi \leq \pi/2 - \theta$.

Under the above assumptions, X_i for $i = 1, \cdots, 4$ are of same scale, and, $X_1 \cup X_2$ and $X_3 \cup X_4$ are of the same scale. In view of this, $\mathcal{C}^1$ and $\mathcal{C}^3$ have clusters of same scale while $\mathcal{C}^2$ does not. By direct calculation, the objective function value of each of the 3 clusterings can be obtained

- $J_\mu(\mathcal{C}^1) = h|\cos\theta| + \mu \sin^2(\theta + \phi)$;
- $J_\mu(\mathcal{C}^2) = \frac{h}{2}(1 + |\cos\theta|) + \frac{\mu}{5}\left[2\sin^2(\frac{\theta}{2} + \phi) + 2\sin^2(\frac{3\theta}{2} + \phi) + \sin^2\theta\right]$;
- $J_\mu(\mathcal{C}^3) = h + \frac{\mu}{6}\left[2\sin^2\theta + 2\sin^2(\theta + \phi) + 2\sin^2(2\theta + \phi) + \sin^2\phi\right]$.

Then, it can be shown that $J_\mu(\mathcal{C}^2)$ is not a maximizer of (1), i.e.,

$$J_\mu(\mathcal{C}^2) < \max\{J_\mu(\mathcal{C}^1), J_\mu(\mathcal{C}^3)\}$$

for any $\mu \geq 0$, $0 < \theta \leq \pi/4$ and $\theta \leq \phi \leq \pi/2 - \theta$. Thus, using the SDM, we never obtain $\mathcal{C}^2$ as an optimal clustering. In the other words, X_1 is merged with X_2 if and only if X_3 is merged with X_4.

In summary, the above examples illustrate that by varying μ, the SDM is capable of capturing different clusterings while each clustering contains clusters of the same scale.

5 Experimental Results

In this section, we perform several tests on our clustering model. All datasets are assumed (or transformed) to be on the unit sphere of an Euclidean space. As mentioned in §4, this assumption

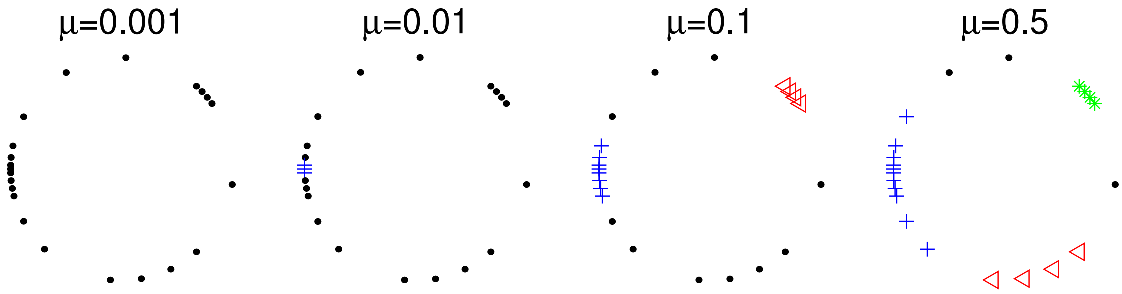


Figure 6: Clusterings at various scales obtained by the SDM algorithm (a dot denotes a singleton). For each μ clusters of the same scale are identified while clusters of small scale are split into singletons.

holds for data such as text documents and gene expression levels. Thus, similarity of data vectors can be reasonably measured the Pearson correlation coefficient. All experiments are done with the SDM algorithm presented in §3.2.

Example 1: Our aim in this example is to show that the parameter μ controls the scale and that our model allows identification of outliers. For ease of visualization, a dataset consisting of 22 data vectors on the unit circle of $\mathbb{R}^2$ is employed, see Figure 6. The data are randomly permuted before running the SDM algorithm. A dot in the figure denotes a cluster having only one point. When $\mu = 0.001$, the clustering corresponds to a fine scale, every cluster is a singleton. When $\mu = 0.01$, a cluster of three points is identified. When μ is increased to 0.1, two clusters are identified. Note that the clusters detected are of the same scale, in the sense that the spacings between adjacent points in each cluster are comparable. When we further increase μ to 0.5, three clusters are detected.

Example 2: A natural question to ask is whether the optimal clustering $\mathcal{C}^*(\mu)$ is sensitive with respect to μ. In Figure 7, we employ the SDM algorithm with μ ranging from 0 to 0.5 to determine the optimal clustering $\mathcal{C}^*(\mu)$ of the dataset used in Figure 6. We see that the graphs of $H_{\mathrm{WA}}(\mathcal{C}^*(\mu))$, $S_{\mathrm{WA}}(\mathcal{C}^*(\mu))$ and d are approximately piecewise constant. This shows that clusterings produced are stable for a range of values of μ. Moreover, natural clusterings (e.g. when $\mu = 0.1$ and 0.5) have a wide range of μ at which the clusterings are stable. Hence, it is easy to identify most of the possible clusterings by running the SDM algorithm with several randomly chosen values of μ. The values of μ at which the three graphs in Figure 7 have discontinuity are exactly the same. This is due to the fact that the clusterings (and hence H_{WA}, S_{WA} and d) produced are stable with respect to μ and discontinuities in the graphs appear only when the clusterings are changed. Finally, the values

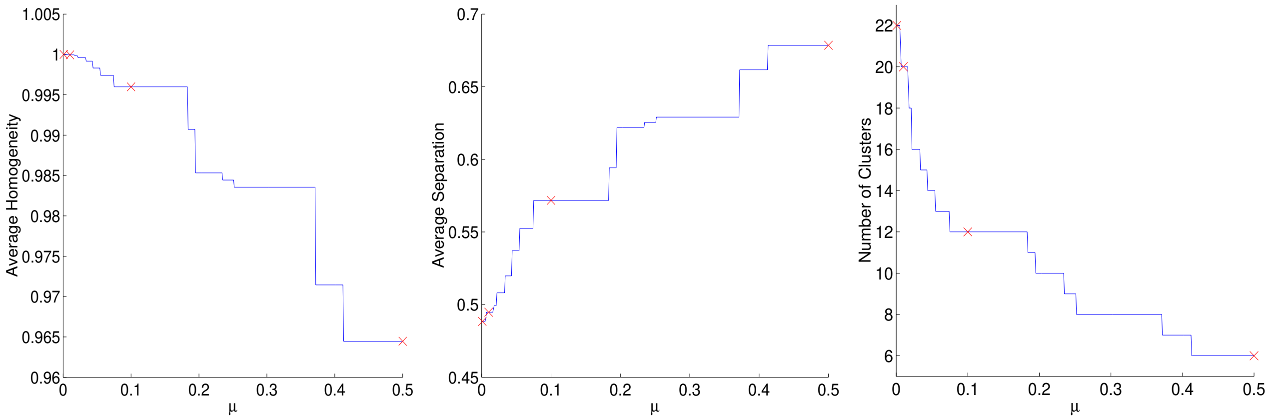


Figure 7: Effect of μ on H_{WA}^{P}, S_{WA}^{P} and the number of clusters obtained by the SDM algorithm. Clusterings produced are stable for a range of values of μ.

of H_{WA}^{P}, S_{WA}^{P} and d of the four clusterings in Example 1 are marked with a symbol "x". From the figure, we may see approximately the range of μ at which each clustering occurs.

Example 3: In this example, we show the ability of the SDM algorithm to separate outliers. For reference proposes, we include the results of the k-means algorithm which is well-known to be sensitive to outliers. The dataset consists of 500 points on the unit sphere of the 10-dimensional Euclidean space. The dataset has 5 natural clusters and 20 random outliers. The dataset is constructed by first generating 5 random 10-dimensional *source vectors* $\mathbf{s}_l$ for $l = 1, \cdots, 5$. Then each data vector $\mathbf{x}_i$ is generated by adding random perturbation to a randomly selected source vector

$$\mathbf{x}_i = \mathbf{s}_l + \alpha \mathbf{e}_i \qquad \text{for some} \quad 1 \leq l \leq 5 \tag{6}$$

where $\alpha = 0.2$ and the entries of $\mathbf{e}_i$ are i.i.d. and uniformly distributed on $[-0.5, 0.5]$. Finally, we add 20 outliers, which do not fall into any of the clusters but at the same time are not far away from the main clusters, to the dataset. The data are normalized to unit vectors before running our algorithm and the k-means algorithm. The similarity matrix of the dataset is visualized as a grey scale image in Figure 8(a). The (i, j)-th entry of the matrix is the Pearson correlation coefficient between the i-th and the j-th data vectors. In the image, a correlation coefficient of 1 is displayed as white color while that of 0 is displayed as black color. For visualization, we ordered the data such that adjacent points belong to the same cluster and the 481st-500th data vectors are outliers. In Figure 8(b)-(d), the (i, j)-th pixel in the figure is colored if and only if the corresponding two data vectors belong to the same cluster. In Figure 8(b), we show the natural clustering of the dataset.

Data belonging to the same cluster are displayed with the same color. It can be seen that there are 5 main clusters (the 5 large diagonal blocks) with 20 outliers (on the top right corner). We tested the k-means algorithm with $k = 5, \cdots, 25$. For $k = 5, 6, 7$, the results are quite similar, we only display the results for $k = 5$ in Figure 8(c). The k-means algorithm correctly identified the five main clusters but as expected, since the outliers are not very well-separated from the main clusters, some outliers are assigned to the main clusters. For $k > 7$, the main clusters are broken down into subclusters. Even though with $k = 25$, the outliers cannot be singled out. This is again reflected by the fact that k-means algorithm is sensitive to outliers. It also reflects that the outliers are not far away from the main clusters, and therefore, it is nontrivial to separate them. Our SDM (with $\mu = 0.5$), however, successfully identifies the main clusters and separates outliers from the main clusters, see Figure 8(d). Additionally, the SDM doesn't merge any of the outliers to form small subclusters but creates a new cluster for each outlier. This allows the SDM algorithm to identify outliers easily.

Example 4: The aim of this example is to contrast the difference between the SDM and classical hierarchical methods. We employ a dataset of 1000 points in $\mathbb{R}^3$ which are normalized to unit vectors. The results of the SDM algorithm ($\mu = 100$) and the average-linkage hierarchical method are shown in Figure 9(a) and (b) respectively. The parameters of the 2 methods are tuned such that 2 clusters are obtained. The SDM recovers the 2 natural clusters, an elongated one and a circular one. However, the hierarchical method splits the elongated cluster into 2 pieces in which one of the pieces is merged with the circular cluster. This phenomena is attributed to the fact that after 3 clusters are formed, the next merging decision made by the hierarchical method is solely based on the distance between the centers of the 3 clusters, but not on the distribution of the points within each cluster. Other hierarchical methods such as single-linkage and complete-linkage methods also exhibit similar behavior since each of their merging decisions depend only on a single representative from each cluster. The SDM, however, takes both intra-cluster and inter-cluster information into account, and hence, results in discovering the 2 natural clusters.

Example 5: We now indicate how to cluster a dataset in p-dimensional Euclidean space (not on the unit sphere of p-dimensional Euclidean space) by mapping the dataset into the unit sphere in $(p + 1)$-dimensional Euclidean space. Thus, similarity can be measured by the Pearson correlation coefficients. Note that in general other similarity measures such as Minkowski distance is preferred; our propose here is merely to illustrate another possibility. We used the Mesh Partitioning Toolbox developed by Gilbert et al [6] to map the 2-dimensional dataset in Figure 1(b) into the unit sphere

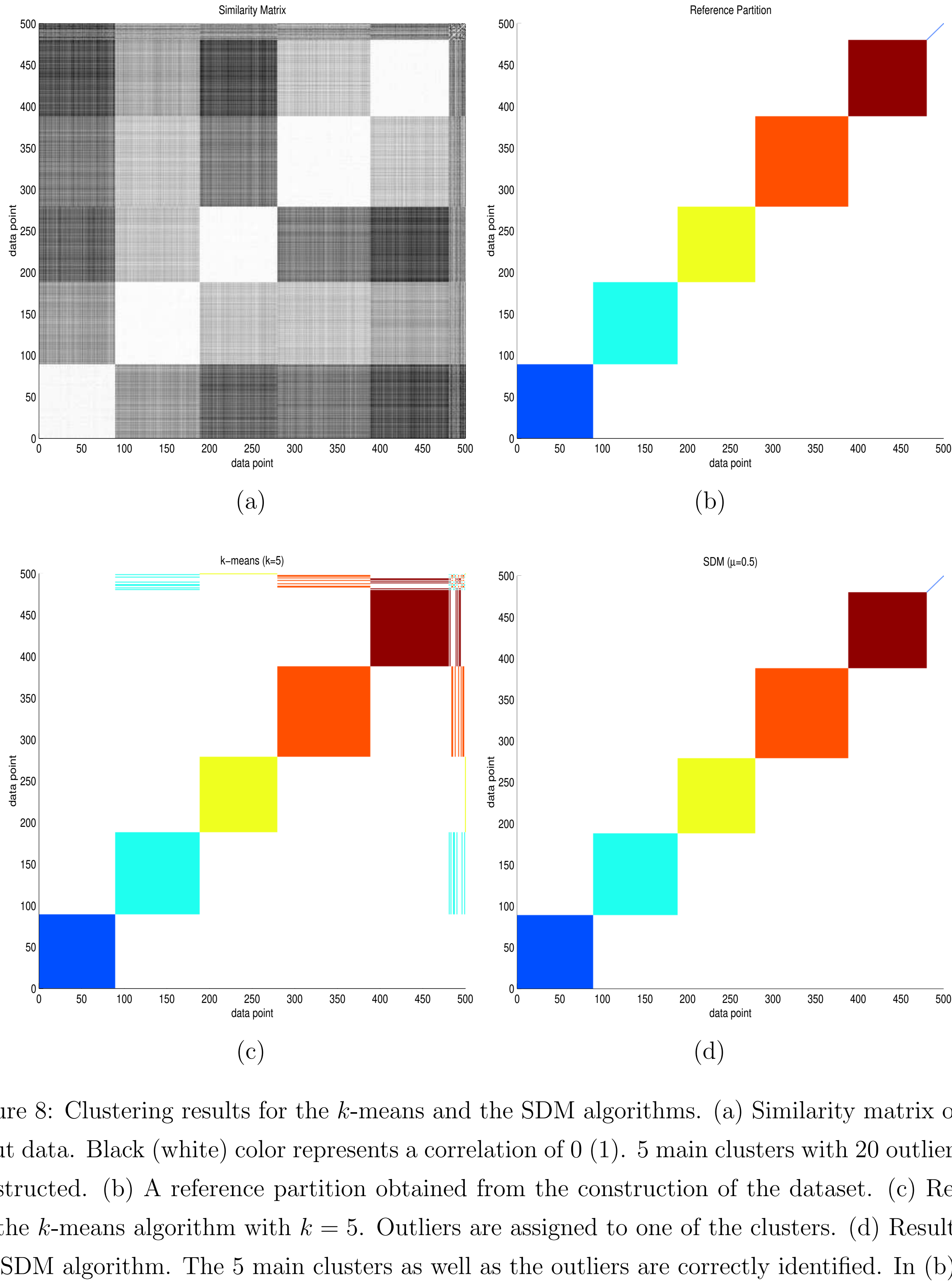
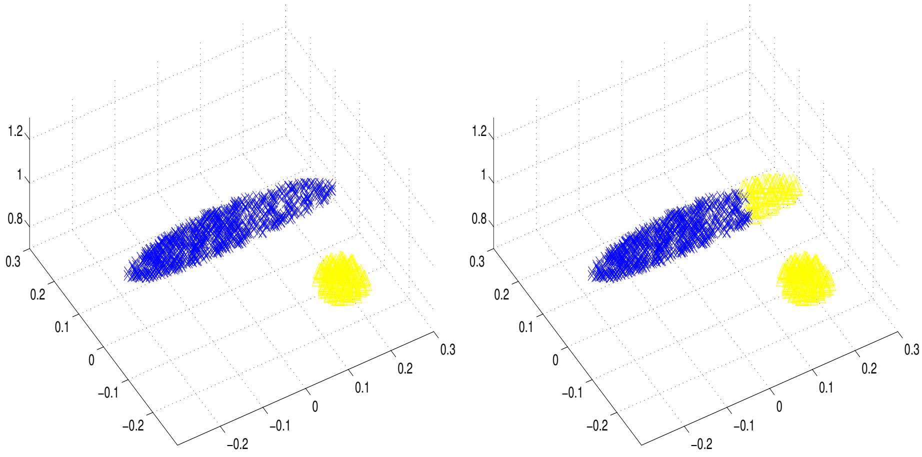


Figure 8: Clustering results for the k-means and the SDM algorithms. (a) Similarity matrix of the input data. Black (white) color represents a correlation of 0 (1). 5 main clusters with 20 outliers are constructed. (b) A reference partition obtained from the construction of the dataset. (c) Results for the k-means algorithm with $k = 5$. Outliers are assigned to one of the clusters. (d) Results for the SDM algorithm. The 5 main clusters as well as the outliers are correctly identified. In (b)-(d), two data vectors $\mathbf{x}_i$ and $\mathbf{x}_j$ belong to the same cluster if and only if the (i, j)-th pixel in the figure is colored.



(a) (b)

Figure 9: (a) Clustering results of the SDM algorithm. (b) Clustering results of the averag-linkage hierarchical method. The merging decisions of the average-linkage hierarchical methods are based on distance between cluster centers only. The lack of intra-cluster information results in break down of the elongated natural cluster. The SDM successfully recovers the 2 natural clusters by taking inter-cluster and intra-cluster information into account.

in $\mathbb{R}^3$. Figures 10(a) and (c) show the results of using the SDM algorithm with $\mu = 0.2$ and $\mu = 0.5$ in 2-dimensional space respectively. The corresponding results on the unit sphere in $\mathbb{R}^3$ is shown in Figures 10(b) and (d). It can be seen that clusters of approximately the same scales are identified for each μ. Moreover, clusters obtained with $\mu = 0.2$ have smaller scale than that with $\mu = 0.5$. We observe that when $\mu = 0.2$, two tight clusters (located in the right hand side of Figure 10 (a)) are identified while all other points are identified as outliers, except 5 points on the upper left corner. When $\mu = 0.5$, three clusters of coarser scale are identified. Moreover, outliers are also identified.

Example 6: In this example, we emphasize the difference between the SDM algorithm and the DBSCAN algorithm, a local density-based clustering algorithm. The SDM algorithm scores a clustering based on the global distribution of the data while the DBSCAN merges data vectors based on local densities. The dataset consists of 6000 random points in the two rhombuses

$$\left\{(x,y) : |x + 100/\sqrt{2}| + |y| \le 100/\sqrt{2}\right\} \cup \left\{(x,y) : |x - 100/\sqrt{2}| + |y| \le 100/\sqrt{2}\right\}.$$

The data points are distributed so that the density near the origin (i.e. the common corner of the two rhombuses) is higher than elsewhere. The high density near the origin makes the DBSCAN fail to separate the two natural clusters. In Figure 11(a), we list the clustering obtained by the SDM

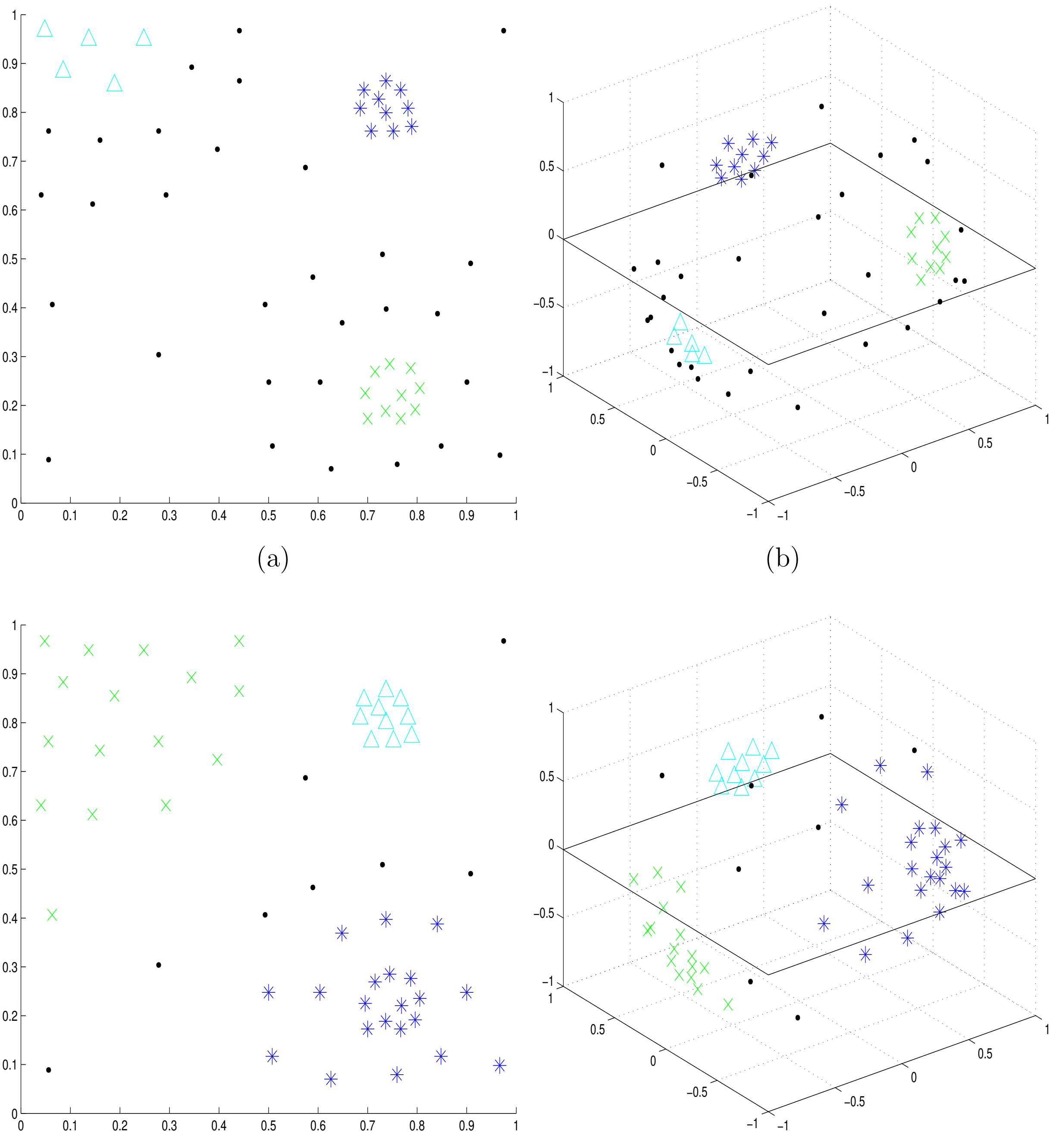


Figure 10: Clustering of a 2-dimensional dataset at two different scales. (a) Clustering results in $\mathbb{R}^2$ using the SDM algorithm with $\mu = 0.2$. (b) Corresponding results for (a) on the unit sphere in $\mathbb{R}^3$. (c) Clustering results in $\mathbb{R}^2$ using the SDM algorithm with $\mu = 0.5$. (d) Corresponding results for (c) on the unit sphere in $\mathbb{R}^3$.

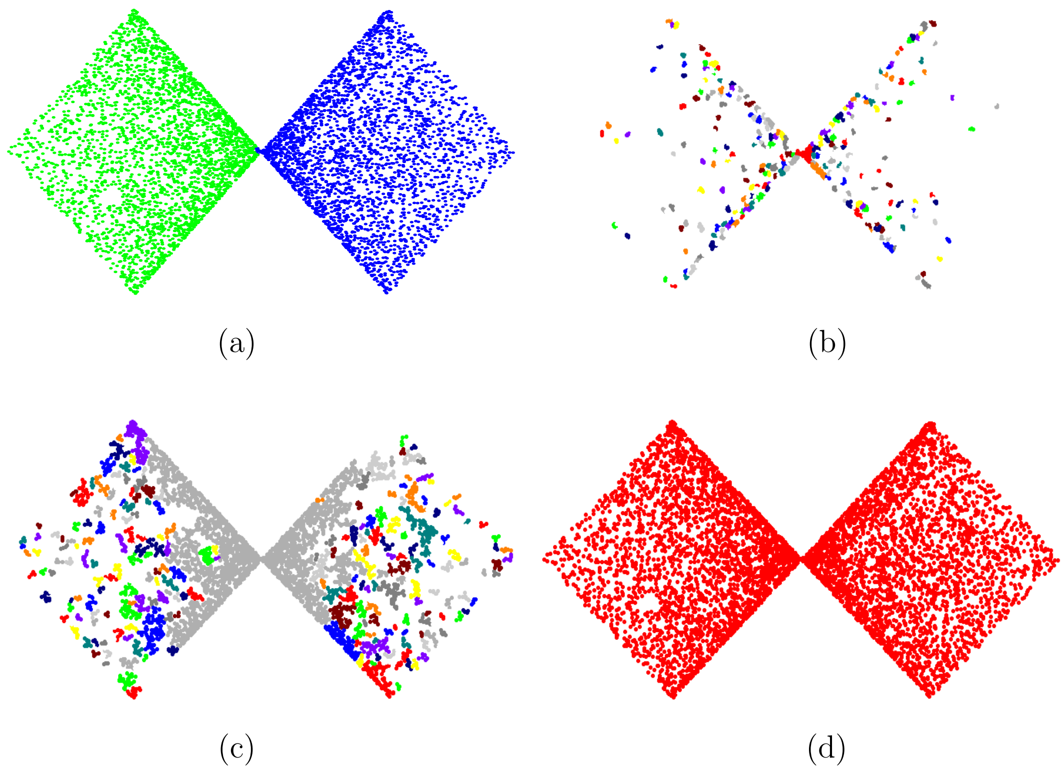


Figure 11: Local density-based clustering algorithm cannot separate the two natural clusters. (a) SDM with $\mu = 0.9$. (b) DBSCAN with $Eps = 1$. (c) DBSCAN with $Eps = 2$. (d) DBSCAN with $Eps = 4$. In (b)–(d), all singletons are not shown.

algorithm for $\mu = 0.9$. We observe that the SDM is capable of separating the two natural clusters. The points near the origin are separated into two clusters because of the presence of the points away from the origin. The points in the right and left rhombuses exert opposite forces on the points near the origin and thus they are pulled to separate. In Figures 11(b)–(d), the results of the DBSCAN algorithm for $Eps = 1$, 2 and 4 are shown. All singletons are not displayed. The $MinPts$ parameter is fixed at 4 as recommended by the authors of [4]. Here the parameter Eps specifies the radius of a neighborhood around a data vector, and $MinPts$ specifies the minimum number of points in a neighborhood so that an Eps-neighborhood containing at least $MinPts$ points is considered to be 'dense enough'. We observe that when $Eps = 1$, corresponding to a very high density, most clusters are of very small size. When $Eps = 2$, corresponding to a density in between the densities near the origin and away from the origin, points near the origin are merged, while points away from the origin form clusters of small size. When $Eps = 4$, corresponding to a very low density, all points are merged into a single cluster.

Example 7: We illustrate in this example that when several datasets are similar to each other, a fixed μ will discover all clusters having the same scale among the datasets. This implies that μ is not a generic parameter; if we are given a reference partition corresponding to some μ_0, then μ_0 can be applied to other similar datasets to discover clusters having the same scale as the clusters in the reference partition. In Figure 12, we show the clustering results of the SDM algorithm (with μ fixed to be 0.1) on 4 datasets having 1, 2, 3, and 4 cluster(s) of the same scale respectively. We observe that, using the fixed μ, all the clusters are identified and the noise vectors are singled out (except in Figure 12(a) where 5 noise vectors are grouped to the large cluster).

Example 8: In this example, we apply the SDM algorithm to identify different textures in an image at different scales. The original 512-by-512 image is shown in Figure 13(a). There are four different textures in the image. These textures form a natural hierarchy; four textures (vertical line segments, horizontal line segments, random noises in a white background and random noises in a black background) in the lower hierarchy and two textures (line segments and random noises in a flat background) in the higher hierarchy. To cluster portions of the image with similar textures together, we divide the image into 256 blocks of size 32-by-32, as shown in Figure 13(b). Each block is mapped into a 10-dimensional feature space. Thus, the dataset consists of 256 data vectors in a 10-dimensional feature space. The feature space is depicted in Table 1. We normalize the data as follows. First, for each feature, the maximum value across all data vectors is normalized to 1. Next, we normalize each data vector to a unit vector. We employ the SDM algorithm with $\mu = 0.8$ and $\mu = 20$. The dataset is randomly permuted beforehand. The clustering results for $\mu = 0.8$ and $\mu = 20$ are shown in Figures 13(c) and (d) respectively. In Figures 13(c) and (d), the color of each pixel represents the cluster label of the corresponding block in Figure 13(b). We observe in Figure 13(c) that blocks having the same texture in the lower hierarchy are perfectly clustered together. Similarly, Figure 13(d) shows that blocks having the same texture in the higher hierarchy are also perfectly clustered together.

We remark that we divide the image into blocks because of its simplicity. One may also cluster pixels instead of blocks.

Example 9: In this example, we test the robustness of our heuristic algorithm to the ordering of the data. We use 3 datasets where each of them consists of 400 data in 20-dimensional space. They are generated using the model in (6) with $\alpha = 0.2$, 0.25 and 0.3 respectively. All results are done with $\mu = 2$. A reference clustering $\{\mathcal{C}_1, \cdots, \mathcal{C}_5\}$ is defined such that $\mathbf{x}_i \in \mathcal{C}_l$ if $\mathbf{x}_i$ is generated by the l-th source $\mathbf{s}_l$. For each dataset, we run the algorithm 500 times where each time the dataset is

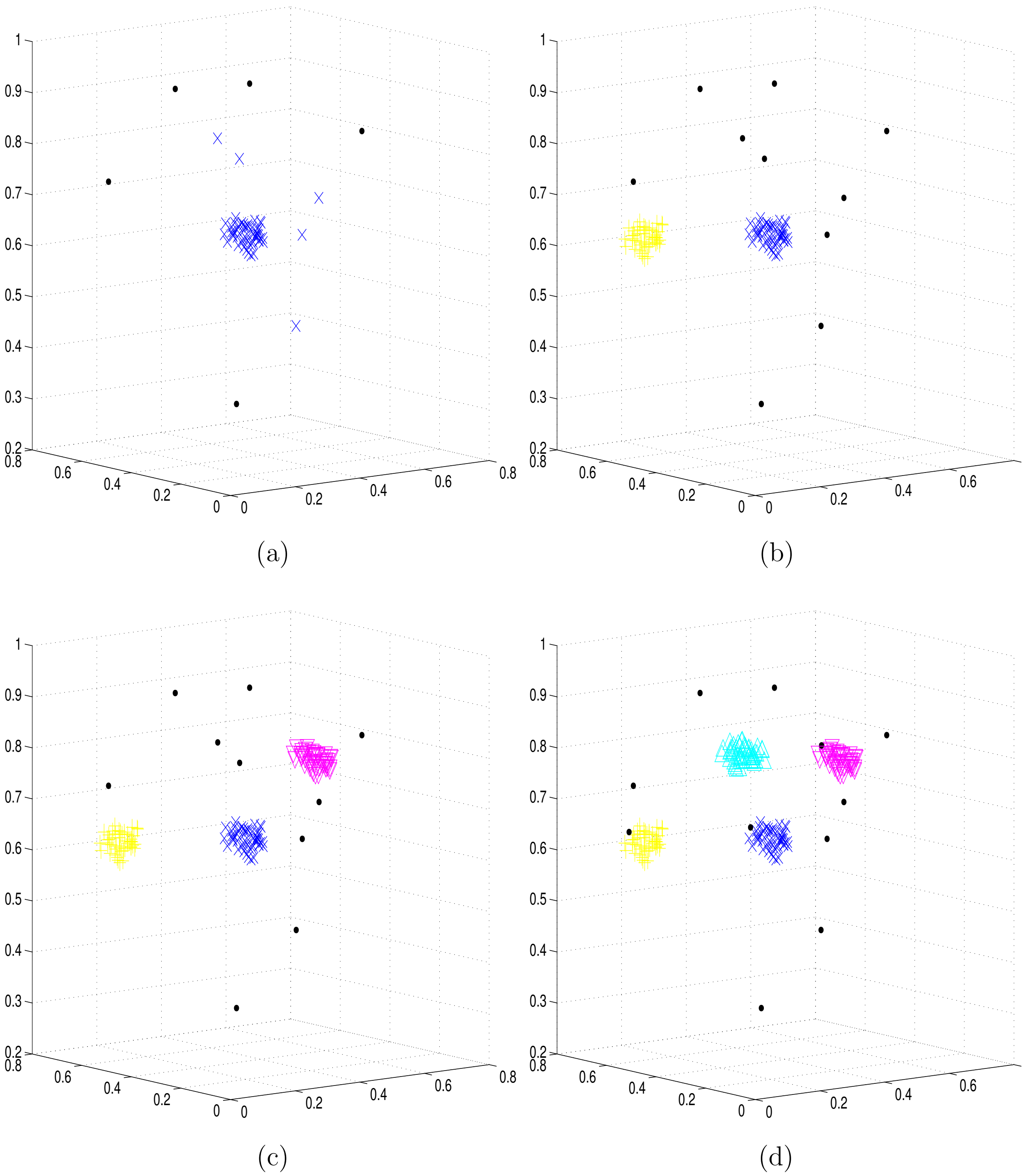


Figure 12: Using the same value of μ to cluster 4 datasets having 1, 2, 3, and 4 cluster(s) of the same scale respectively. We observe that, using the fixed μ, all the clusters are identified with noise singled out (except in (a) where 5 noise vectors are grouped to the large cluster).

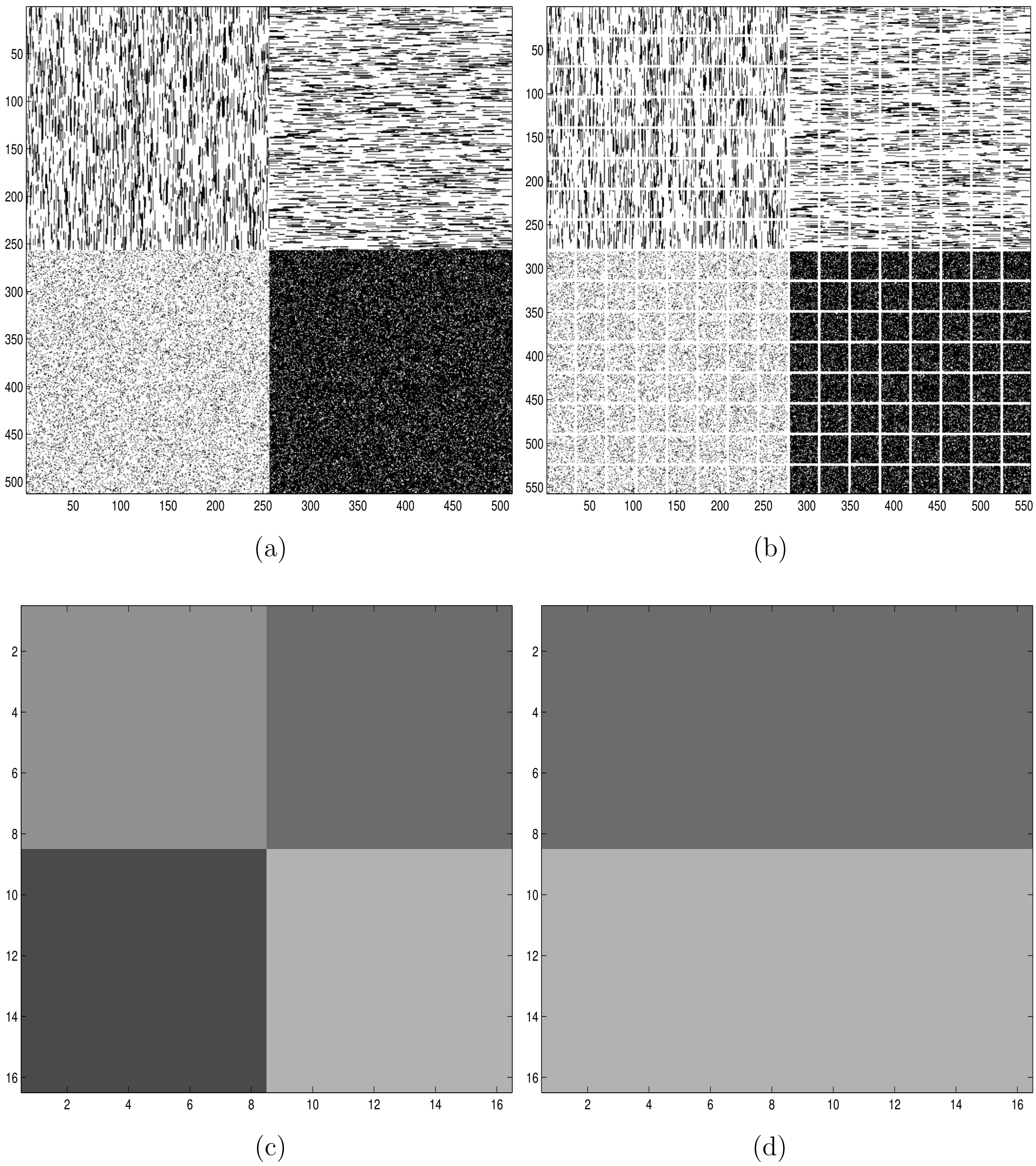


Figure 13: Identifying different textures in an image at different scales. (a) An image with 4 different textures. (b) Dividing the image into 256 blocks. (c) The SDM algorithm with $\mu = 0.8$. (d) The SDM algorithm with $\mu = 20$. In (c) and (d), the color of each pixel represents the cluster label of the corresponding block. Blocks with the same textures (with respect to certain scale) are perfectly clustered together.

Dimension	Feature
1	Number of pixels with grey level between 1 and 25 in a block
2	Number of pixels with grey level between 26 and 75 in a block
3	Number of pixels with grey level between 76 and 125 in a block
4	Number of pixels with grey level between 126 and 175 in a block
5	Number of pixels with grey level between 176 and 225 in a block
6	Number of pixels with grey level between 226 and 256 in a block
7	Total variation of grey levels of a block in horizontal direction
8	Total variation of grey levels of a block in vertical direction
9	Total variation of grey levels of a block in diagonal direction
10	Total variation of grey levels of a block in anti-diagonal direction

Table 1: The feature space used to capture texture information in Example 8.

randomly permuted beforehand. We compare the resulting clusterings with the reference partition by counting the number of mismatches. Table 2 lists the results. We only show the number of mismatches that have actually occurred. We can see from the results that when $\alpha = 0.2$ and 0.25, the algorithm is very robust to the ordering of the data. There is no mismatch in most of the tests. When $\alpha = 0.3$ (corresponding to very noisy data), 67% of the tests have no mismatch. We observe that when $\alpha = 0.25$ and 0.30, number of mismatches between 61 and 87 are of relatively high frequencies. In each of these clusterings, either two clusters are merged or a cluster is split into singletons, resulting in the number of mismatches close to the size of the corresponding cluster (which is between 61 and 87).

# mismatches	0	1	2–4	38	61–69	75–79	80–87	143–152	165–171	Total
$\alpha = 0.20$	488	5	4		1	2				500
$\alpha = 0.25$	455	5	2	1	27	2	7	1		500
$\alpha = 0.30$	335	2	1		73	11	63	12	3	500

Table 2: Frequency table of the number of mismatches after running the algorithm for 500 times for different random permutations of the input data. The results show that the SDM algorithm is robust to the ordering the data.

6 Adapting the Objective Function to Specific Applications

In this section we discuss two situations in which maximizing J_μ may not give a clustering in an intermediate scale of interest. We also discuss how to adapt the objective function to solve such problems. Finally, we present an example on clustering gene expression data to illustrate that the SDM model can be adapted to suit specific applications.

In Figures 14(a) and (b), we show schematically two datasets which may be problematic when they are clustered by maximizing J_μ. Each dataset contains 6 groups of points, denoted by $\mathcal{C}_i$ for $i = 1, \cdots, 6$. These two datasets can be characterized by (a) groups of clusters that are far away from each other relative to distances between clusters within the same group; (b) clusters of unbalanced size. Consider three different clusterings

$$\mathcal{C}^1\text{: all singletons}; \qquad \mathcal{C}^2 = \{\mathcal{C}_1, \mathcal{C}_2, \mathcal{C}_3, \mathcal{C}_4, \mathcal{C}_5, \mathcal{C}_6\}; \qquad \mathcal{C}^3 = \{\mathcal{C}_1 \cup \mathcal{C}_2 \cup \mathcal{C}_3,\ \mathcal{C}_4 \cup \mathcal{C}_5 \cup \mathcal{C}_6\}.$$

In Figure 14(c), we illustrate typical objective function values of the above three clusterings for both datasets. We observe that the clustering $\mathcal{C}^2$ of an intermediate scale is never a maximizer for any $\mu \geq 0$. This is attributed to the definition of $S^{\mathrm{P}}_{\mathrm{WA}}$ in (4):

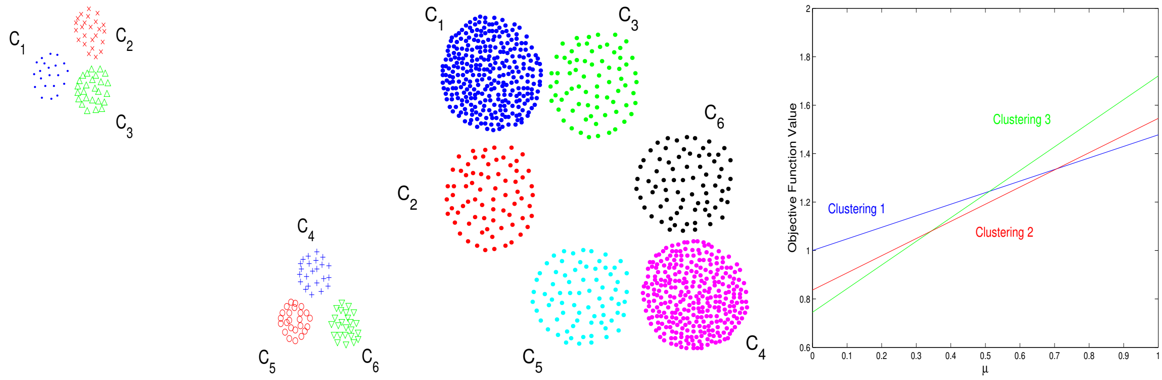
$$S^{\mathrm{P}}_{\mathrm{WA}}(\mathcal{C}^{(d)}) := \frac{1}{2}\left[1 - \frac{1}{\sum_{i<j} |\mathcal{C}_i||\mathcal{C}_j|} \sum_{i<j} |\mathcal{C}_i||\mathcal{C}_j| \rho\left(\frac{\mathbf{m}(\mathcal{C}_i)}{\|\mathbf{m}(\mathcal{C}_i)\|}, \frac{\mathbf{m}(\mathcal{C}_j)}{\|\mathbf{m}(\mathcal{C}_j)\|}\right)\right]. \tag{7}$$

For the dataset in Figure 14(a), the $S^{\mathrm{P}}_{\mathrm{WA}}$ value of $\mathcal{C}^2$ is much smaller than that of $\mathcal{C}^3$. If we split $\mathcal{C}^3$ into $\mathcal{C}^2$, then the $H^{\mathrm{P}}_{\mathrm{WA}}$ value will only increase slightly that it cannot compensate the large decrease in $S^{\mathrm{P}}_{\mathrm{WA}}$. However, if we split $\mathcal{C}^2$ into $\mathcal{C}^1$, then the value of $H^{\mathrm{P}}_{\mathrm{WA}}$ will greatly increase to 1 with a relatively small decrease in $S^{\mathrm{P}}_{\mathrm{WA}}$ (especially when μ is small). For the dataset in Figure 14(b), the size of the clusters $\mathcal{C}_1$ and $\mathcal{C}_4$ are much larger than that of the rest. Because of the weighting factor $|\mathcal{C}_i||\mathcal{C}_j|$ in the definition of $S^{\mathrm{P}}_{\mathrm{WA}}$, the model prefers to merge clusters together, especially for clusters which are relatively close together and of very different sizes. Such a merging will increase $S^{\mathrm{P}}_{\mathrm{WA}}$ more than it will decrease in $H^{\mathrm{P}}_{\mathrm{WA}}$.

Despite the fact that $\mathcal{C}^2$ may not be a maximizer for any $\mu \geq 0$, the following monotonicity properties, which are at the heart of our model, do generally hold.

$$H^{\mathrm{P}}_{\mathrm{WA}}(\mathcal{C}^1) > H^{\mathrm{P}}_{\mathrm{WA}}(\mathcal{C}^2) > H^{\mathrm{P}}_{\mathrm{WA}}(\mathcal{C}^3); \qquad S^{\mathrm{P}}_{\mathrm{WA}}(\mathcal{C}^1) < S^{\mathrm{P}}_{\mathrm{WA}}(\mathcal{C}^2) < S^{\mathrm{P}}_{\mathrm{WA}}(\mathcal{C}^3).$$

This indicates that the problems may be resolved by redefining the objective function as some nonlinear combination of $H^{\mathrm{P}}_{\mathrm{WA}}$, $S^{\mathrm{P}}_{\mathrm{WA}}$ and μ, i.e., $J_\mu(\mathcal{C}^{(d)}) = h(H^{\mathrm{P}}_{\mathrm{WA}}(\mathcal{C}^{(d)}), S^{\mathrm{P}}_{\mathrm{WA}}(\mathcal{C}^{(d)}), \mu)$ for some function h. For example, let $J_\mu(\mathcal{C}^{(d)}) = H^{\mathrm{P}}_{\mathrm{WA}}(\mathcal{C}^{(d)}) + \mu[S^{\mathrm{P}}_{\mathrm{WA}}(\mathcal{C}^{(d)})]^m$ for some $m > 1$. Besides



(a) (b) (c)

Figure 14: Situations in which the objective function $J_\mu(\mathcal{C})$ fails to identify some clusterings of intermediate scale. (a) Groups of clusters that are far apart from each other. (b) Unbalanced clusters. (c) Clustering 2 is not a maximizer of $J_\mu(\mathcal{C})$ for any $\mu \geq 0$.

redefining $J_\mu(\mathcal{C}^{(d)})$, we describe another remedy for each of the above two situations. For the problem in Figure 14(a), we may first cluster the dataset into 2 clusters. Then, for each cluster obtained, we may recursively cluster the data vectors again. For the problem in Figure 14(b), we may simply employ an unweighted average of pair-wise separation as a measure of the overall separation. More precisely, for a clustering with d clusters, we may define

$$S^{\mathrm{P}}_{\mathrm{UWA}}(\mathcal{C}^{(d)}) := \frac{1}{2}\left[1 - \frac{1}{\binom{d}{2}}\sum_{i<j}\rho\left(\frac{\mathbf{m}(\mathcal{C}_i)}{\|\mathbf{m}(\mathcal{C}_i)\|}, \frac{\mathbf{m}(\mathcal{C}_j)}{\|\mathbf{m}(\mathcal{C}_j)\|}\right)\right].$$

This gives another objective function

$$K_\mu(\mathcal{C}^{(d)}) = H^{\mathrm{P}}_{\mathrm{WA}}(\mathcal{C}^{(d)}) + \mu S^{\mathrm{P}}_{\mathrm{UWA}}(\mathcal{C}^{(d)}). \tag{8}$$

We observe from experiments that K_μ usually produces more cluster solutions than J_μ as μ is varied.

We remark that occurrence of the above two problems can be identified from the values of H and S after clustering with J_μ.

Next, we present an example on clustering gene expression data. The particular dataset that we used has the two characteristics illustrated in Figure 14. We employ the objective function K_μ defined in (8) and cluster the dataset in 2 levels.

Example 10: In this example, we test the SDM algorithm in clustering gene expression data. We use the dataset reported by Iyer et al in [9] which studied the response of human fibroblasts to serum. Primary cultured human fibroblasts were induced into a quiescent state after serum deprivation for 48 hours. Next, they were stimulated by adding a medium which provided growth factors to them. DNA microarrays were then used to monitor the temporal changes in mRNA levels of 8613 human genes at 12 different time points (0 hr, 15 min, 30 min, 1 hr, 2 hr, 4 hr, 8 hr, 12 hr, 16 hr, 20 hr and 24 hr) after stimulation. A separate unsynchronized sample was also added as the 13-th experimental condition. A subset of 517 genes whose expression changed substantially were selected. Their data matrix is constructed by taking the ratio of the value of each time point to the value of the first time point. In our test, we use these 517 selected genes over the above mentioned 13 experimental conditions. In a preprocessing step, we log transform (base 2) the original data matrix so that inductions or repressions of the same magnitude have the same absolute value but with different signs. Finally, each data vector (gene) is normalized to unit length.

Due to the issues mentioned in this section, we use K_μ in (8) as the objective function. We first run the SDM algorithm with $\mu = 0.65$ to obtain 2 clusters. The resulting clustering is shown in Figure 15(a). In the figure, each row represents normalized expression levels of a gene across 13 experimental conditions. Thus the Euclidean norm of the expression levels of each row is 1. A cell in red (green) represents the corresponding gene is overexpressed (underexpressed) at the corresponding experimental condition. The coloring scheme is shown in Figure 15(d). We denote the first cluster (in the upper half of Figure 15(a)) by X_1 and the second cluster (in the lower half of Figure 15(a)) by X_2. The clusters X_1 and X_2 consist of genes which are overexpressed and underexpressed at most of the time points. Next, we run the SDM algorithm with $\mu = 1.0$ on X_1 and with $\mu = 0.8$ on X_2. This yields a total of 11 clusters (excluding singletons) which are denoted by $\mathcal{C}_i$ for $i = 1, \cdots, 11$. The resulting clusterings are shown in Figures 15(b) and (c) respectively. In Figure 16, we show the average profile of each of the 11 clusters. We observe that all of the profiles have very distinct patterns which shows that the clusters are well-separated.

In Table 3, we summarize the results quantitatively. We also include the results by the CLICK algorithm [15] for reference. In the table, "Combined" denotes a clustering obtained by combining the results of clustering X_1 and X_2. Among the two values of $H^{\mathrm{P}}_{\mathrm{WA}}$, $S^{\mathrm{P}}_{\mathrm{UWA}}$ and $S^{\mathrm{P}}_{\mathrm{WA}}$ obtained from the two algorithms, the better value is shown in bold face. We observe that for the combined results, the SDM algorithm gives both better $H^{\mathrm{P}}_{\mathrm{WA}}$ and $S^{\mathrm{P}}_{\mathrm{UWA}}$ values. For the rest of the results, trade-offs between maximizing $H^{\mathrm{P}}_{\mathrm{WA}}$ and $S^{\mathrm{P}}_{\mathrm{UWA}}/S^{\mathrm{P}}_{\mathrm{WA}}$ are observed.

In [5], Eisen et al also performed cluster analysis on this dataset. Eisen et al gave a biological interpretation of five of the clusters. These five clusters are also approximately discovered by the SDM algorithm. They correspond to $\mathcal{C}_1$, $\mathcal{C}_2$, $\mathcal{C}_6$, $\mathcal{C}_8$ and $\mathcal{C}_9$ in the results of the SDM algorithm. Interpretation, according to Eisen et al, of these five clusters is given in Table 4.

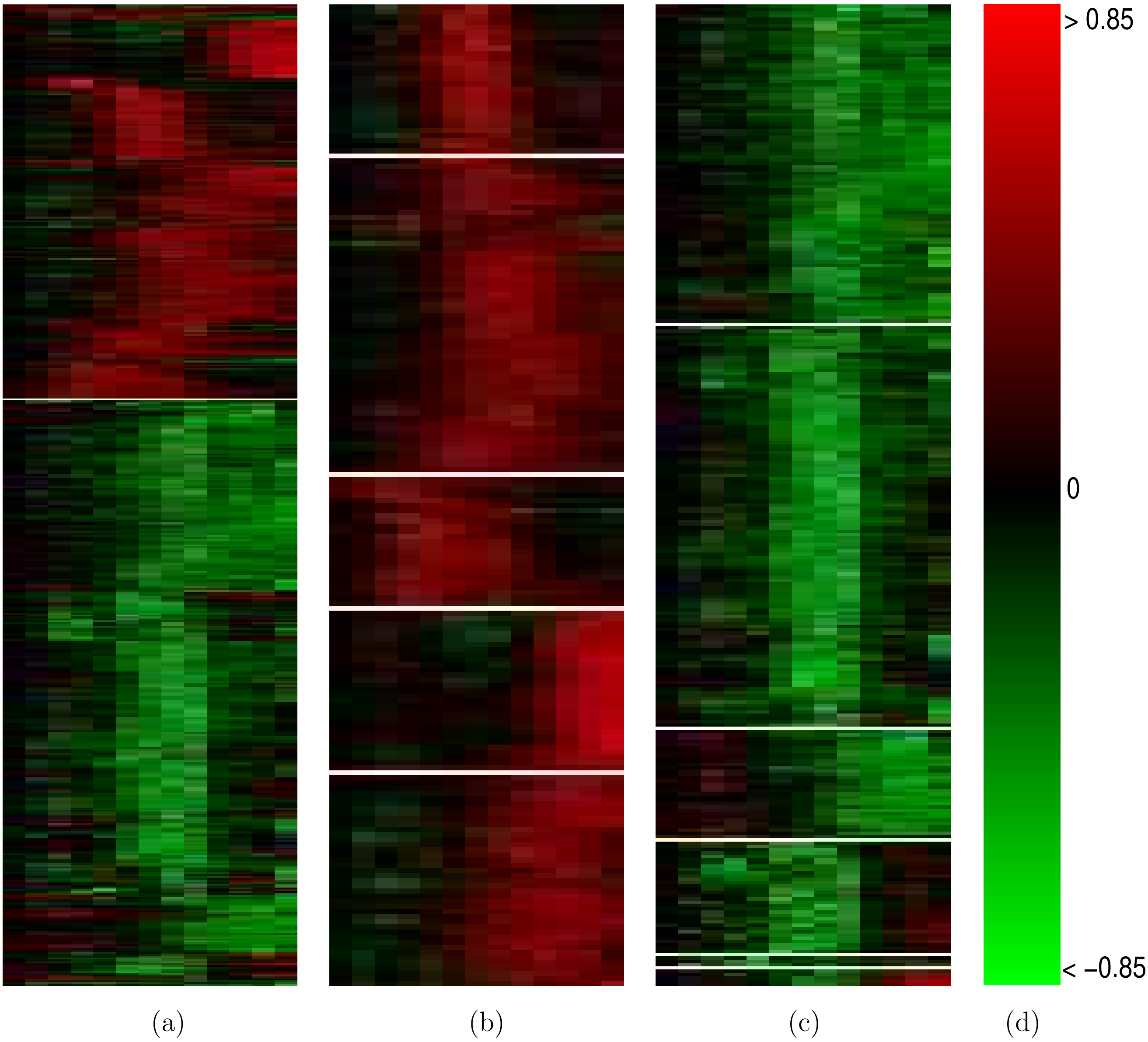


Figure 15: Clustering of gene expression data with the SDM algorithm. (a) 2 clusters X_1, X_2 are obtained by the SDM algorithm for $\mu = 0.65$ on the full dataset. They correspond to genes that are overexpressed and underexpressed in most of the time points respectively. (b) 5 clusters ($\mathcal{C}_i$ for $i = 1, \cdots, 5$) are obtained for $\mu = 1.0$ on the first cluster X_1 in (a). (c) 6 clusters ($\mathcal{C}_i$ for $i = 6, \cdots, 11$) are obtained for $\mu = 0.8$ on the second cluster X_2 in (a). (d) Correspondence between the color of a cell and the normalized expression level. In (a)-(c), each row represents the normalized expression levels of a gene at 13 different conditions.

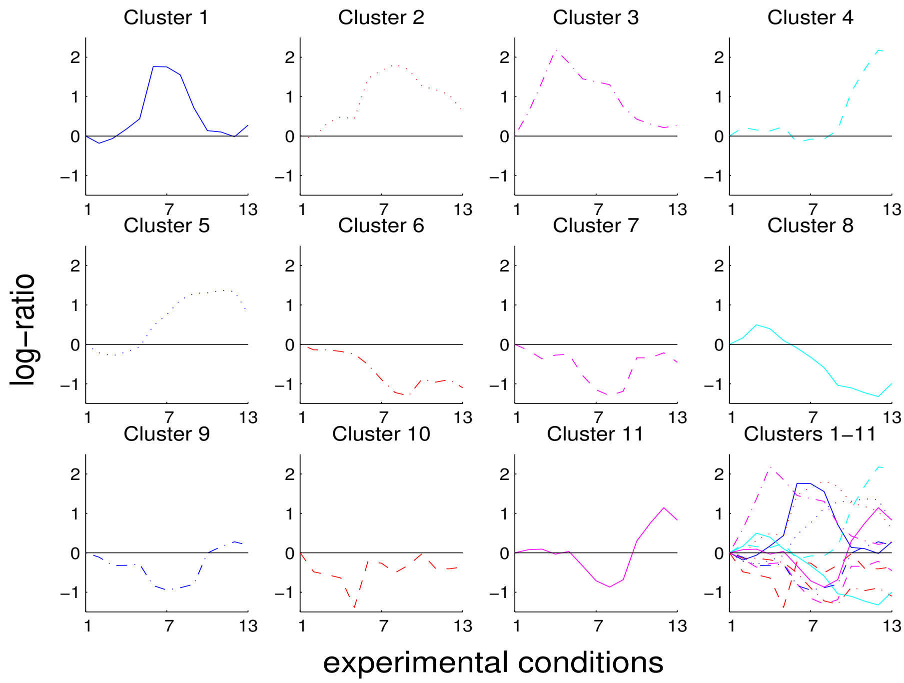


Figure 16: Average profile of the 11 clusterings obtained in Figures 15 (b) and (c). The profiles show very distinct patterns indicating that the clusters obtained are well-separated from each other.

To quantitatively measure the difference between two clusterings, define $S_{ij} = 1$ if the SDM algorithm assigns $\mathbf{x}_i$ and $\mathbf{x}_j$ to the same cluster and $S_{ij} = 0$ otherwise. Likewise, define C_{ij} for the corresponding results of the CLICK algorithm. Let $n_{kl} = |\{(i,j) : S_{ij} = k \text{ and } C_{ij} = l\}|$ for $k,l = 0,1$. The Jaccard coefficient is defined as $n_{11}/(n_{11} + n_{10} + n_{01})$. In Table 5, we list the Jaccard coefficients between the results of the SDM and the CLICK algorithms. A larger coefficient indicates a better match between two clusterings and a perfect match has a coefficient of 1. Thus, we observe that the clusterings produced by the two methods are quite different. We are unable to compare the 2 clusterings qualitatively in terms of their interpretability as the clustering produced by the CLICK algorithm has no interpretation reported (as far as the authors know).

7 Concluding Remarks

In this paper, we have formulated a model for identifying clusters of different scales based on the maximization of an objective function which seeks to maximize the homogeneity of individual clusters and the separation between different clusters, with a weighting parameter μ which assigns a weighting to these two quantities. By varying the parameter μ, clusters of different scales can be

	SDM			
Dataset	X ($\mu = 0.65$)	X_1 ($\mu = 1$)	X_2 ($\mu = 0.8$)	Combined
# Clusters	2	5	6	11
Clusters size	208, 309	29, 61, 25 31, 41	97, 122, 33 34, 3, 5	29, 61, 25, 31, 41 97, 122, 33, 34, 3, 5
# Singletons	0	21	15	36
$H^{\mathrm{P}}_{\mathrm{WA}}$	0.74546	**0.91814**	0.90431	**0.90987**
$S^{\mathrm{P}}_{\mathrm{UWA}}$	**0.97705**	**0.46649**	**0.46457**	**0.50623**
$S^{\mathrm{P}}_{\mathrm{WA}}$	**0.97705**	0.25546	0.17385	0.54021

	CLICK			
Dataset	X	X_1	X_2	Combined
# Clusters	6	4	3	7
Clusters size	254, 147, 47 25, 23, 21	73, 69 35, 26	148, 110, 20	73, 69, 35, 26 148, 110, 20
# Singletons	0	5	31	36
$H^{\mathrm{P}}_{\mathrm{WA}}$	**0.83692**	0.88638	**0.91055**	0.90082
$S^{\mathrm{P}}_{\mathrm{UWA}}$	0.57478	0.41969	0.39352	0.46315
$S^{\mathrm{P}}_{\mathrm{WA}}$	0.70855	**0.26434**	**0.20049**	**0.57243**

Table 3: Summary of the clustering results by the SDM and CLICK algorithms. The values $H^{\mathrm{P}}_{\mathrm{WA}}$, $S^{\mathrm{P}}_{\mathrm{UWA}}$ and $S^{\mathrm{P}}_{\mathrm{WA}}$ serve as figures of merit. For each of these values, the algorithm which yields a better result is shown is bold face.

Cluster 1	Signaling and angiogenesis	Cluster 8	The immediate-early response
Cluster 2	Wound healing and tissue remodeling	Cluster 9	The cell cycle
Cluster 6	Cholesterol biosynthesis		

Table 4: Interpretation of the clustering results by the SDM algorithm according to [5].

identified.

In all clustering algorithms, some parameters must be specified, either explicitly or implicitly, in order to pick one of the clusterings that an algorithm produces. In the k-means algorithm, the number of clusters must be specified. In the SDM model, the scale parameter specifies the scale of the desired clusters. Thus, in applications where we seek for clusterings in different scales or clusters of a known scale, the SDM algorithm is an appropriate choice. We illustrate an application of the SDM to a special case when pair-wise similarity is measured by the Pearson correlation coefficient. Analysis, through two representative examples, of the SDM is given to illustrate that (a) by varying the scale parameter μ, clusters of different scale can be identified; (b) for a fixed μ, clusters of the same scale are identified. In general, a single configuration of a clustering algorithm may not be suitable for all kinds of data. In view of this, we present strategies on adapting the objective

	X	X_1	X_2	Combined
Jaccard coeff.	0.6375	0.3759	0.4262	0.4136

Table 5: Jaccard coefficients between the clusterings obtained by the SDM and the CLICK algorithms. We observe that the clusterings produced by the two methods are quite different.

function to suit different applications within our SDM framework.

We also successfully apply the SDM to a wide range of clustering problems, including problems with synthetically generated data, clustering of image textures, and clustering gene expression profiles. Numerical tests demonstrate the usefulness of such formulations.

Acknowledgements. The authors wish to thank the *Institute for Pure and Applied Mathematics* (IPAM) at the University of California, Los Angeles. This work was undertaken while we were participants in the *Functional Genomics* program held at IPAM during the Fall quarter of 2000.

We would also like to thank Roded Sharan and Ron Shamir for providing the CLICK program and Martin Ester, Hans-Peter Kriegel, Jorg Sander and Xiaowei Xu for providing the DBSCAN program.

References

[1] Chan, T.F. and Vese, L. 2001. Active contours without edges. IEEE Transactions on Image Processing, 10:266–277.

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

[3] Duda, R.O., Hart, P.E. and Stork, D.G. 2001. Pattern classification. New York: Wiley-Interscience.

[4] Ester, M., Kriegel, H., Sander, J. and Xu, X. 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. Proceedings of 2nd Int. Conf. On Knowledge Discovery and Data Mining, Portland, OR, pp.226–231.

[5] Eisen, M.B., Spellman, P.T., Brown, P.O. and Botstein, D. 1998. Cluster analysis and display of genome wide expression patterns. Proc. of National Academy of Sciences, 85:14863–14868.

[6] Gilbert, J.R., Miller, G.L. and Teng, S. 1998. Geometric mesh partitioning: implementation and experiments. SIAM J. Sci. Comp, 19(6):2091–2110.

[7] Guha, S., Rastogi, R. and Shim, K. 1998. CURE: An efficient clustering algorithm for large databases. Proceedings of 1998 ACM-SIGMOD Int. Conf. on Management of Data.

[8] Han, J. and Kamber, M. 2001. Data mining: concepts and techniques. San Francisco: Morgan Kaufmann Publishers.

[9] Iyer, V.R. et al. 1999. The transcriptional program in the response of human fibroblasts to serum. Science, 283:83–87.

[10] Jain, A.K., Murty, M.N. and Flyn, P.J. 1999. Data clustering: a review. ACM Computing Surveys, 31(3):264–323.

[11] Karypis, G., Han, E.H. and Kumar, V. 1999. CHAMELEON: a hierarchical clustering algorithm using dynamic modeling. COMPUTER, 32:68–75.

[12] MacQueen, J.B. 1967. Some methods for classification and analysis of multivariate observations. Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability, Volume I: Statistics, pp.281–297.

[13] Mumford, D. and Shah, J. 1989. Optimal approximation by piecewise smooth functions and associated variational problems. Comm. Pure Appl. Math., 42:577–685.

[14] Shamir, R. and Sharan, R. 2002. Algorithmic approaches to clustering gene expression data. MIT Press: Current Topics in Computational Biology, pp.269–299.

[15] Sharan, R. and Shamir, R. 2000. CLICK: A clustering algorithm for gene expression analysis. Proc. Intl. Conf. on Iintelligent Systems for Molecular Biology, pp. 260–268.

[16] Strong, D.M. and Chan, T.F. 2000. Edge-preserving and scale-dependent properties of total Variation Regularization. Department of Mathematics, University of California, Los Angeles, CA, UCLA CAM Report 00-38.

[17] Zhao, Y. and Karypis, G. 2001. Criterion functions for document cluster: experiments and analysis. Department of Computer Science and Enginneering, University of Minnesota, Technical Report # 01-40.