Multiobjective Data Clustering*

Martin H. C. Law Alexander P. Topchy Anil K. Jain

Department of Computer Science and Engineering, Michigan State University
East Lansing, MI 48824 {lawhiu,topchyal,jain}@cse.msu.edu

Abstract

Conventional clustering algorithms utilize a single criterion that may not conform to the diverse shapes of the underlying clusters. We offer a new clustering approach that uses multiple clustering objective functions simultaneously. The proposed multiobjective clustering is a two-step process. It includes detection of clusters by a set of candidate objective functions as well as their integration into the target partition. A key ingredient of the approach is a cluster goodness function that evaluates the utility of multiple clusters using re-sampling techniques. Multiobjective data clustering is obtained as a solution to a discrete optimization problem in the space of clusters. At meta-level, our algorithm incorporates conflict resolution techniques along with the natural data constraints. An empirical study on a number of artificial and real-world data sets demonstrates that multiobjective data clustering leads to valid and robust data partitions.

1 Introduction

Data clustering is a challenging task, whose difficulty is caused by a lack of unique and precise definition of a cluster. It is generally acknowledged that clustering is an ill-posed problem when prior information about the underlying data distributions is not well defined. Inability to detect clusters with diverse shapes and sizes is a fundamental limitation of every clustering algorithm irrespective of the clustering criterion (objective function) used. Discovery of a majority or all of the clusters (of arbitrary shapes) present in the data is a long-standing goal of exploratory pattern analysis. A framework is needed that integrates the outputs of multiple clustering algorithms, corresponding to various clustering objectives, applied to the same data, to meet this goal.

A related problem is that virtually all existing clustering algorithms assume a homogeneous clustering criterion over the entire feature space. As a result, all the clusters detected tend to be similar in shape and often have similar data density. A single clustering algorithm cannot find all the clusters if different regions of the feature space contain clusters of diverse shapes, because its intrinsic criterion may not fit well with the data distribution in the entire feature space.

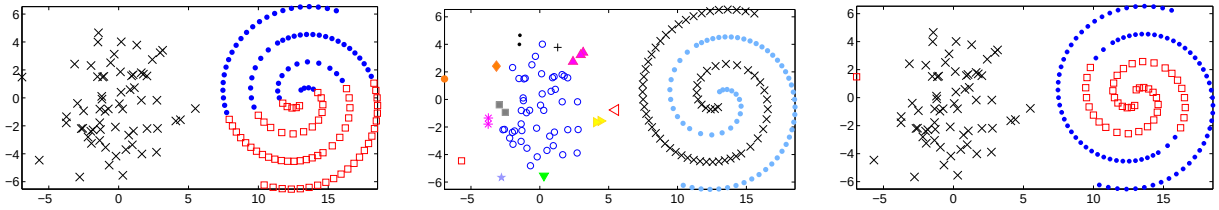
Consider the example in Fig. 1 that contains two spiral clusters and one globular cluster with a total of 192 points. While the k-means algorithm can recover the globular cluster using an appropriate value of k, it cannot identify the two spirals no matter what value of k is used. In contrast, the single-link algorithm (SL) is able to detect the two spirals, but fails to place the remaining points into a single globular cluster. The single-link dendrogram reveals that the three clusters cannot be recovered, no matter at what similarity level we decide to cut the dendrogram to obtain a partition. Our experiments with spectral clustering [10] also demonstrate that the three desired clusters cannot be obtained.

A key observation is that although none of the clustering algorithms considered above can recover all the three clusters, *all* the target (individual) clusters in the example have been detected, albeit by different clustering criteria. The globular cluster is detected by the k-means algorithm with $k = 3$, while SL can locate the two spiral clusters. Assuming that diverse clustering algorithms can provide all the desired clusters, we can use a cluster fitness function to identify which clusters from specific clustering algorithms are the most meaningful to be included in the final clustering solution. However, evaluation of cluster utility (goodness) is a non-trivial procedure closely related to calibration of clustering objective functions. This leads to the multiobjective data clustering approach introduced in this paper.

1.1 Multiobjective Clustering

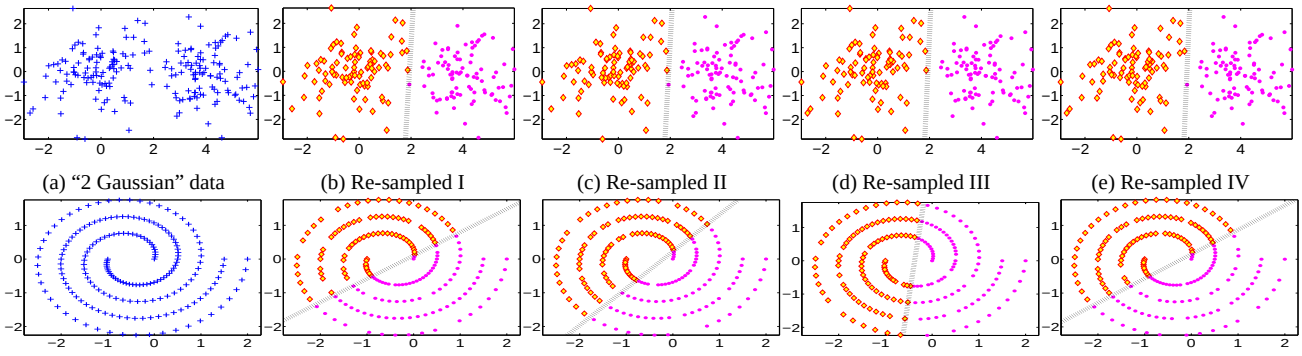
The goal of multiobjective clustering is to find clusters in a data set by applying several clustering algorithms corresponding to different objective functions. We propose a clustering approach that integrates the output of different clustering algorithms into a single partition. More precisely, given different clustering objective functions, we seek a partition that utilizes the appropriate objective functions for different parts of the data space. This framework can be

*This work is supported by ONR grant no. N000140410183.



(a) k-means with $k = 3$ (b) Single-link, $k = 15$ (c) Spectral with $\sigma = 0.6$ and $k = 3$

Figure 1: The resulting partitions by (a) k-means, (b) single-link and (c) spectral clustering on this "globular-spiral" data set.



(a) "2 Gaussian" data (b) Re-sampled I (c) Re-sampled II (d) Re-sampled III (e) Re-sampled IV

(f) "2 spiral" data (g) Re-sampled I (h) Re-sampled II (i) Re-sampled III (j) Re-sampled IV

Figure 2: Results of k-means with $k = 2$ for different re-sampled versions of two data sets. Dotted lines in the figures correspond to the cluster boundaries. The partitions of "2 Gaussian" data set are almost the same for different re-sampled versions, suggesting that k-means with $k = 2$ gives good clusters. The same cannot be said for the "2 spiral" data set.

viewed as a meta-level clustering since it operates on multiple clustering algorithms simultaneously. The final partition not only contains meaningful clusters but also associates a specific objective function with each cluster.

Multiobjective clustering is a two-step process: (i) independent or parallel discovery of clusters by different clustering algorithms, and (ii) construction of an "optimal" partition from the discovered clusters. The second step is a difficult conceptual problem, since clustering algorithms often are not accompanied by a measure of the goodness of the detected clusters. The objective function used by a clustering algorithm is not indicative of the quality of the partitions found by other clustering algorithms. The goodness of each cluster should be judged not only by the clustering algorithm that generated it, but also by an external assessment criteria. In the special case where all partitions come from the same family of probability models, methods such as Bayesian information criteria (BIC) or minimum description length (MDL) can be used to decide in favor of a particular partition. Unfortunately, many clustering algorithms do not admit probabilistic interpretations and these criteria are inapplicable. A more general goodness function that can be adopted for all clustering algorithms is needed.

A possible candidate for the goodness function is the stability of a cluster under the re-sampling of the data set [8, 6]. Fig. 2 gives an intuition why stable clusters are preferable. Another issue is that adopting clusters from different partitions may not produce a valid data partition. Multiobjective clustering frequently encounters conflicting criteria in the sense that detected clusters are incompatible (due to cluster overlap, for example). Such spatial constraints must be taken into account at the meta-level. In addition to the goodness function, we need to provide a conflict resolution technique for handling competing clusters.

1.2 Related Work

Multiobjective clustering is not the only approach that operates with multiple clustering solutions. For example, clustering ensembles combine different partitions of the data using the so-called consensus functions [12]. Recent studies of clustering ensembles have dealt with all aspects of partition generation, their diversity as well as methods of their combination, e.g., see [1, 4, 3, 7]. In addition, a specialized combination method for k-means and agglomerative linkage algorithms was proposed in [11].

It must be emphasized that clustering with multiple objective functions is not equivalent to clustering ensembles. The distinction lies in both the choice and the integration of the objective functions. Clustering ensembles operate with

homogenous objective functions. Therefore, good clusters may become diluted by weak clusters in an ensemble. In general, solutions delivered by consensus functions may violate the criteria of optimality inherent to the contributing partitions. On the other hand, while the proposed multiobjective approach includes conflicting objectives, it indirectly localizes the objectives and optimizes them in distinct regions of the feature space.

A separate issue is the choice of the clustering objective functions to be combined. Here we assume that the chosen set of clustering algorithms ensures that each of the true clusters is detected by at least one of the algorithms.

2 Integration of Partitions

2.1 Problem Statement

Suppose we are given a data set $\mathcal{D} = \{\mathbf{x}_1, \ldots, \mathbf{x}_m\}$ and L clustering algorithms $\mathcal{A}_i$, $i = 1, \ldots, L$, such that each algorithm $\mathcal{A}_i$ returns a partition $P^{(i)}$ of $\mathcal{D}$ which maximizes the corresponding objective function f_i. Formally,

$$P^{(i)} = \{S_1^{(i)}, S_2^{(i)}, \ldots, S_{M_i}^{(i)}\} = \underset{P(\mathcal{D})}{\operatorname{argmax}}\, f_i(P(\mathcal{D})), \quad (1)$$

where $P(\mathcal{D})$ denotes an arbitrary partition of $\mathcal{D}$ and $S_j^{(i)}$ is the j-th cluster in $P^{(i)}$. Let $\mathcal{S} \equiv \cup_i P^{(i)}$ be a collection of all the clusters generated by the candidate algorithms $\{\mathcal{A}_i\}$:

$$\mathcal{S} = \{S_1^{(1)}, \ldots, S_{M_1}^{(1)}, \ldots, S_1^{(L)}, \ldots, S_{M_L}^{(L)}\}. \quad (2)$$

The goal of multiobjective clustering is to find a "compromise" partition $T \equiv \{\mathcal{C}_1^*, \ldots, \mathcal{C}_K^*\}$ based on the partitions $\{P^{(i)}\}$. In other words, the target collection of clusters, T, is derived from the clusters in $\mathcal{S}$:

$$T = \{\mathcal{C}_1^*, \ldots, \mathcal{C}_K^* : \forall i\ \exists j, k,\ \mathcal{C}_i^* \approx S_j^{(k)}\}. \quad (3)$$

The key assumption of the proposed approach is that all the underlying clusters in $\mathcal{D}$ can be identified (approximately) by at least one of the L candidate clustering algorithms. In order to interpret T as a partition, the following conditions must be satisfied:

Absence of conflicts. No data point $\mathbf{x}_i$ is assigned to more than one cluster in T, i.e., $\mathcal{C}_i^* \cap \mathcal{C}_j^* = \emptyset$ for all $i \neq j$.

Complete coverage. Each data point in $\mathcal{D}$ is assigned to at least one cluster in partition T.

In practice, these two properties may be violated, as it is hard for clusters produced by different algorithms to be exactly non-conflicting and cover all the data. We can still interpret T as an approximate partition if the number of conflicting and unassigned data points is small. If necessary, a post-processing step can assign such data points to one of the clusters $\mathcal{C}_i^*$, thereby ensuring that T is a valid partition.

In practice, we must compare and select the clusters obtained by the clustering algorithms applied to the entire feature space. For this purpose, we need to introduce an additional objective function that is external to the criteria used by the clustering algorithms. Let $g_j(\mathcal{C}_i, \mathcal{D})$ be a goodness function that measures the quality of the cluster $\mathcal{C}_i$ in a manner consistent with the objective function f_j. By comparing the values of goodness functions $\{g_j(\mathcal{C}_i, \mathcal{D})\}$, we can indirectly adopt the most appropriate clustering criteria $\{f_j\}$ for different data subsets (clusters).

2.2 Stability As Goodness Function

The goodness function $g_j(\mathcal{C}_i, \mathcal{D})$ depends on both the cluster $\mathcal{C}_i$ and the entire data set $\mathcal{D}$, instead of $\mathcal{C}_i$ alone, because, in general, the goodness of a cluster acquires its meaning in the context of all the data points. A reasonable goodness function should possess the following properties. First, it should be related to the clustering criterion f_j, or the clustering algorithm $\mathcal{A}_j$ that optimizes f_j. The larger the value of $g_j(\mathcal{C}_i, \mathcal{D})$, the better the cluster $\mathcal{C}_i$ is with respect to f_j or, equivalently, $\mathcal{A}_j$. Second, it should be comparable for different clustering objection functions. In other words, if $g_j(\mathcal{C}_i, \mathcal{D}) > g_l(\mathcal{C}_i, \mathcal{D})$, then the quality of cluster $\mathcal{C}_i$ is better with respect to f_j than f_l. Finally, the values of the goodness function must be comparable across different clusters. For example, $g_j(\mathcal{C}_i, \mathcal{D}) = g_j(\mathcal{C}_l, \mathcal{D})$ implies that clusters $\mathcal{C}_i$ and $\mathcal{C}_l$ are equally good with respect to the criterion f_j.

We suggest that the goodness function should be based on cluster stability [8]. Cluster stability reflects the variation in the clustering solution under perturbation of the data and can be used with different clustering algorithms. The perturbation is done by data re-sampling, either with or without replacement. Stable clusters are usually preferable, because if the same clusters are formed irrespective of minor changes in the data set, the clusters are robust and hence reliable. Stable clusters can be a consequence of good isolation or compactness of a cluster. Note that direct optimization of stability is a hard problem, without known efficient methods. The pseudo-code for calculating $g_j(\mathcal{C}_i, \mathcal{D})$ is given by Algorithm 1. Here, $sim(\mathcal{C}_i, P(\mathcal{D}))$ is a similarity measure that compares cluster $\mathcal{C}_i$ with an arbitrary data partition $P(\mathcal{D})$. Algorithm 1 runs the clustering algorithm many times with different re-sampled versions of the data set, and uses the average of the similarity between $\mathcal{C}_i$ and the resulting partition as the value of the goodness function.

Finding the similarity measure $sim(\mathcal{C}_i, P(\mathcal{D}))$ between a cluster and a partition is far from straightforward. Most previous work only compares a partition with another partition [5, 9]. However, it is possible to derive the similarity between a cluster and a partition by transforming the cluster

for $l := 1$ to M **do**
Re-sample $\mathcal{D}$ either with or without replacement to obtain the perturbed data set $\mathcal{D}'$
Run $\mathcal{A}_j$ using $\mathcal{D}'$ as input and obtain $P(\mathcal{D}')$
$P(\mathcal{D}')$ is converted to $P(\mathcal{D})$ by labelling the data in $\mathcal{D}/\mathcal{D}'$ according to the semantics of the clusters
Compute $score[l] = sim(\mathcal{C}_i, P(\mathcal{D}))$
end for
$g_j(\mathcal{C}_i, \mathcal{D})$:= average of $score[l]$

Algorithm 1: Goodness function $g_j(\mathcal{C}_i, \mathcal{D})$ evaluation.

$\mathcal{C}_i$ to a partition and adopting one of the known partition-to-partition distance definitions. Specifically, we construct $\mathcal{P}_1$, a partition consisting of two clusters, as $\mathcal{P}_1 = \{\mathcal{C}_i, \mathcal{D}/\mathcal{C}_i\}$, where $\mathcal{D}/\mathcal{C}_i$ denotes the set of data points in $\mathcal{D}$ that are not in $\mathcal{C}_i$. $P(\mathcal{D})$ is transformed to $\mathcal{P}_2$, a partition of two clusters, by $\mathcal{P}_2 = \{\mathcal{C}^*, \mathcal{D}/\mathcal{C}_1^*\}$, where $\mathcal{C}^*$ denotes the union of all "positive" clusters in $P(\mathcal{D})$. A cluster $\mathcal{C}'_j$ in $P(\mathcal{D})$ is positive if more than half of its data points are in $\mathcal{C}_i$. Intuitively, $\mathcal{C}^*$ is the "best" approximation of $\mathcal{C}_i$ in $P(\mathcal{D})$. Mutual information, which is a common similarity measure between two partitions [12], can be used to compare $\mathcal{P}_1$ and $\mathcal{P}_2$.

$$MI(\mathcal{P}_1, \mathcal{P}_2) = \sum_{i=0}^{1}\sum_{j=0}^{1} \frac{r_{ij}}{m^2} \log \frac{m p_{ij}}{r_{ij}}, \qquad (4)$$
$$r_{ij} = p_{i\cdot}\, p_{\cdot j}, \quad p_{i\cdot} = p_{i0} + p_{i1}, \quad p_{\cdot j} = p_{0j} + p_{1j},$$

where p_{11} denotes the number of data points in both $\mathcal{C}^*$ and $\mathcal{C}_i$, p_{01} is the number of data points in both $\mathcal{D}/\mathcal{C}_i$ and $\mathcal{C}^*$, p_{10} is the number of data points in both $\mathcal{C}_i$ and $\mathcal{D}/\mathcal{C}^*$, p_{00} is the number of data points in both $\mathcal{D}/\mathcal{C}_i$ and $\mathcal{D}/\mathcal{C}^*$, and m is the total number of data points. One drawback of mutual information is that its maximum value depends on the cluster sizes $|\mathcal{C}_i|$ and $|\mathcal{C}^*|$. Therefore, we normalize the mutual information with its largest possible value. This leads to the normalized mutual information (NMI) criteria [4].

$$NMI(\mathcal{P}_1, \mathcal{P}_2) = \frac{MI(\mathcal{P}_1, \mathcal{P}_2)}{\frac{-1}{2m}\Big(\sum_{i=0}^{1} p_{i\cdot} \log \frac{p_{i\cdot}}{m} + \sum_{j=0}^{1} p_{\cdot j} \log \frac{p_{\cdot j}}{m}\Big)}. \qquad (5)$$

Since the normalized mutual information, as well as other similarity measures between partitions, depends on the number of clusters, $P(\mathcal{D})$ should have a fixed number of clusters for a fair comparison. This explains why both $\mathcal{P}_1$ and $\mathcal{P}_2$ are constructed with two clusters only.

2.3 Selection of Clusters

Given the list of candidate clusters $\mathcal{S} = \{\mathcal{C}_1, \ldots, \mathcal{C}_M\}$, we find $T = \{\mathcal{C}_1^*, \ldots, \mathcal{C}_K^*\}$, the set of target clusters, using the goodness function. Let $\mathbf{u} = (u_1, \ldots, u_M)^t$ be a vector of indicator variables, where $u_i = 1$ if the i-th cluster in the collection is selected in T, and 0 otherwise. The superscript t denotes the transpose of a vector. The set T can be constructed by finding the optimal $\mathbf{u}$. We shall first assume K, the number of clusters in T, is known.

Let q_{ij} denote the penalty for violating the no-conflict property if $\mathcal{C}_i$ and $\mathcal{C}_j$ are selected in T, and let $\mathbf{Q} \equiv \{q_{ij}\}$. The overall penalty of violating the no-conflict property for $\mathbf{u}$ can be written as the quadratic form $\frac{1}{2}\mathbf{u}^t\mathbf{Q}\mathbf{u}$. Let n_{ij} be the number of data points that are in both $\mathcal{C}_i$ and $\mathcal{C}_j$, i.e., $n_{ij} = |\mathcal{C}_i \cap \mathcal{C}_j|$, and let $\mathbf{N} = \{n_{ij}\}$. One reasonable definition of q_{ij} is:

$$q_{ij} = \frac{n_{ij}}{\max(|\mathcal{C}_i|, |\mathcal{C}_j|)}. \qquad (6)$$

Intuitively, q_{ij} represents the proportion of data points from the smaller cluster that are also assigned to the larger cluster.

Let $\xi(\mathbf{u})$ be the ratio of the data points that remain unassigned by the clusters present in $\mathbf{u}$. If complete-coverage property is satisfied, $\xi(\mathbf{u})$ is zero. Since we want to minimize $\xi(\mathbf{u})$ and $\mathbf{u}^t\mathbf{Q}\mathbf{u}$, as well as maximize the sum of cluster goodness functions, we introduce two positive parameters γ_1 and γ_2 and consider the following discrete quadratic programming problem.

$$\begin{aligned} &\text{Minimize } J(\mathbf{u}) = -\mathbf{s}^t\mathbf{u} + \gamma_1\mathbf{u}^t\mathbf{Q}\mathbf{u} + \gamma_2\xi(\mathbf{u}) \\ &\text{subject to } u_i \in \{0,1\} \text{ and } \sum_{i=1}^{M} u_i = K. \end{aligned} \qquad (7)$$

Here, $\mathbf{s}$ is the vector of goodness function values, $\mathbf{s}_i = g_j(\mathcal{C}_i, \mathcal{D})$, and j is the index of the algorithm $\mathcal{A}_j$ that creates $\mathcal{C}_i$. The first constraint on u_i reflects that $\mathbf{u}$ consists of indicator variables. The second constraint on the sum of u_i corresponds to the fixed number of clusters to be selected. The function $\xi(\mathbf{u})$ can be approximated as $\xi(\mathbf{u}) \approx 1/m(\mathbf{d}^t\mathbf{u} - \frac{1}{2}\mathbf{u}^t\mathbf{N}\mathbf{u})$, where $\mathbf{d} \equiv (d_1, \ldots, d_M)^t$ and d_i is the size of $\mathcal{C}_i$. Since this approximation is quadratic in $\mathbf{u}$, the corresponding objective function in (7) will also be quadratic. In practice, though, we compute the number of unassigned points exactly. Since the Hessian matrix of the objective function may be non-definite, the relaxed continuous version of this optimization problem is non-convex and may have multiple local minima. At the meta-level, multiobjective clustering translates to a NP-hard combinatorial optimization problem.

A simple but efficient heuristic relying on local descent search with multiple re-starts is adopted to solve problem (7). Our optimization procedure starts with an initial random vector $\mathbf{u}_0$ that satisfies the constraint $\sum_{i=1}^{M} u_i = K$. Hill-climbing proceeds by iteratively improving $\mathbf{u}$ until a local minimum is reached. Let ψ be a move operator that generates a set of modified vectors $\mathbf{u}_{\text{new}}$ from the current

solution $\mathbf{u}_0$. The improved $\mathbf{u}$ is obtained as

$$\mathbf{u}_{\text{new}} = \arg \max_{\mathbf{u} \in \psi(\mathbf{u}_0)} J(\mathbf{u}). \quad (8)$$

In the current context, $\psi(.)$ modifies $\mathbf{u}_0$ by swapping a randomly selected "1" in $\mathbf{u}_0$ with a randomly selected "0". As a result, the number of 1's in $\mathbf{u}$ is not changed, thereby satisfying $\sum_{i=1}^{M} u_i = K$. Intuitively, each step in hill-climbing attempts to change the current clustering by discarding one existing cluster and including a new one.

We need to modify $J(\mathbf{u})$ if the number of target clusters, K, is not known. A natural modification is to consider the average, instead of the sum, of the goodness functions. This leads to the following minimization problem:

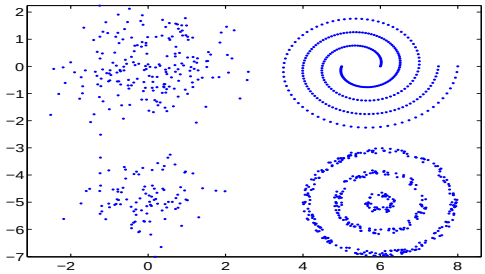
$$\text{Minimize } J'(\mathbf{u}) = -\frac{\mathbf{s}^t\mathbf{u}}{\sum_{i=1}^{M} u_i} + \gamma_1 \mathbf{u}^t\mathbf{Q}\mathbf{u} + \gamma_2 \xi(\mathbf{u}) \quad (9)$$
$$\text{subject to } u_i \in \{0, 1\}.$$

Again, hill climbing can be used to solve this optimization problem. The operator $\psi(.)$ is extended so that, beside moving "1" in $\mathbf{u}$ to a new location, it can also flip a value $1 \leftrightarrow 0$ in random components of $\mathbf{u}$. The flipping operation is effectively an addition or deletion of a cluster from the current solution. In our experiments both versions of multiobjective clustering (Equations (7) and (9)) have been studied.

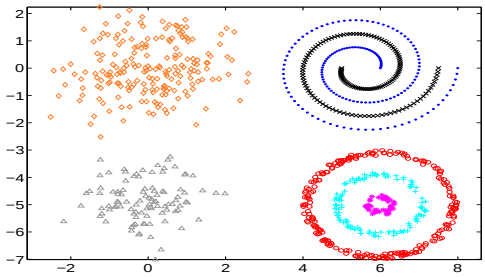
3 Experiments

In order to ensure that the target clusters are found by at least one of the $\mathcal{A}_j$, we consider several "independent" clustering algorithms, i.e., based on different principles. The k-means algorithm minimizes the total within-cluster variance and tends to find spherical clusters. EM is an example of model-based clustering algorithm and detects hyperellipsoidal clusters that may be overlapping. Single-link (SL) [13] clustering, being based on minimum spanning tree, can find chained clusters. Spectral clustering [10] finds clusters based on the spectral properties of the similarity graph constructed from inter-pattern distances. All these algorithms require a parameter k, the number of desired clusters[1]. Note that k is different from K, the number of clusters in the target partition. A wide range of reasonable values of k and σ is used in the experiments. A clustering algorithm run with different parameter values results in two different algorithm instantiations. For each data set, we re-sample 90% of the data without replacement 50 times in order to compute the goodness function. For both EM and k-means, multiple restarts are used to alleviate the local minimum problem. γ_1 and γ_2 are set to 0.4 and 5, respectively, when K is fixed (Eq. (7)). When K is variable (Eq. (9)), γ_1 and

[1]For spectral clustering, an extra width parameter σ is needed.



(a) Synthetic data



(b) Clusters discovered by multiobjective clustering

Figure 3: Clustering results on the synthetic data set.

Data set	k, EM	k, k-means	k, SL	k, spectral	σ, spectral
`iris`	3-5	3-6	10,20,30	3	0.2,0.25
`dermat`	4-10,12	4-10,12	20,40	6,7,8	1,1.4,1.8
`image`	5-12,14	5-12,14	400,450,500	7,9,11,13	3,5,7

Table 1: Parameters for the candidate clustering algorithms.

γ_2 are divided by the initial guess of K. This is due to the difference in the two objective functions. The initial value of K is not important, as we obtain the same target clusters in all of our experiments when K is initialized close to the true number of clusters. The class labels (if available) are only used to evaluate the clustering results by comparing the detected clusters with the known classes.

3.1 Synthetic Data Set

1000 data points in 7 clusters of different shapes are generated (Fig. 3(a)). The clustering algorithms used include k-means with k from 4 to 8, EM with $k \in \{4, 6, 8, 10\}$, SL with $k = 40$ and spectral clustering with $k = 7$ and $\sigma \in \{0.1, 0.3, 0.5, 0.7\}$. Clusters with less than 10 data points (1% of data) are discarded. Five clusters from SL, one cluster from spectral clustering with $\sigma = 0.3$, and one cluster from spectral clustering with $\sigma = 0.5$, are selected when we optimize Eq. (7) (Fig. 3(b)). The "globular" clusters in Fig. 3(b) are identified by spectral clustering, while the spirals and the rings are identified by single-link. Although none of the clustering algorithms by themselves can recover all these seven clusters, the proposed multiobjective clustering has successfully identified them. Identical result is obtained if K is allowed to vary by optimizing Eq. (9).

3.2 Real World Data Sets

We have also performed experiments on three real world data sets from the UCI machine learning repository[2]. The parameter values for the candidate clustering algorithms are listed in table 1. For Iris data (`iris`), we fix $K=3$ and optimize Eq. (7) only; if K is allowed to vary, the "natural" but uninteresting 2-cluster solution is obtained. Multiobjective clustering selects the three clusters generated by EM with $k=3$. This turns out to be the best clustering result among all the algorithms considered; only 2 out of 150 data points in `iris` have different class and cluster labels. So, if a clustering algorithm (EM in this case) is suitable for the data set in the entire feature space, multiobjective clustering selects all the clusters generated by this algorithm.

The dermatology data set (`dermat`) contains 366 data points of 34 features from six classes. One feature with missing values is discarded. Clusters with size less than five are ignored. Four of the underlying classes are completely recovered. 19 out of 91 points in the remaining two classes are misclassified. None of the clustering algorithms alone can achieve this performance. With $k = 6$, k-means misclassified 49 points, EM misclassified 64 points, and the best spectral clustering can do is to misclassify 49 points. Single-link can only identify two significant clusters. When K is allowed to vary, one cluster contains exactly these two overlapping true classes, suggesting that they are better represented by a single cluster. We also observe that clusters from different algorithms (k-means and spectral clustering) are selected in the final partition. Hence, different clustering objectives are indeed adopted in different regions of the feature space.

For the image segmentation data set (`image`), each feature is normalized to have zero mean and unit variance. The constant feature is discarded. In total there are 2310 data points with 18 features in seven classes. Clusters containing less than 2% of data points are discarded. The multiobjective clustering solution does not match the true class labels very well. However, a scan of the confusion matrices produced by different clustering algorithms suggest that some of the true classes cannot be discovered by any of the algorithms, possibly because the true classes are highly overlapping. Comparison of the result of multiobjective clustering with the true class labels gives the confusion matrices in Fig. 5. We notice that classes 2 and 7 can be reasonably recovered. Class 1 is being split, but this is consistent with the structure of the data, as there are indeed two different blue regions (corresponding to class 1) in Fig. 4(a). Classes 3 and 5 (red and magenta) and classes 4 and 6 (cyan and gray) are also observed to be grouped together in Fig. 4(a). It is not surprising that multiobjective clustering cannot distinguish these classes. Indeed, when K is not fixed, a much more natural clustering structure is found (Fig. 4(c) and 5(b)). Note also the outlier data points are identified as unassigned data points (green in Fig. 4(b) and magenta in Fig. 4(c)).

[2] `http://www.ics.uci.edu/~mlearn/MLSummary.html`.

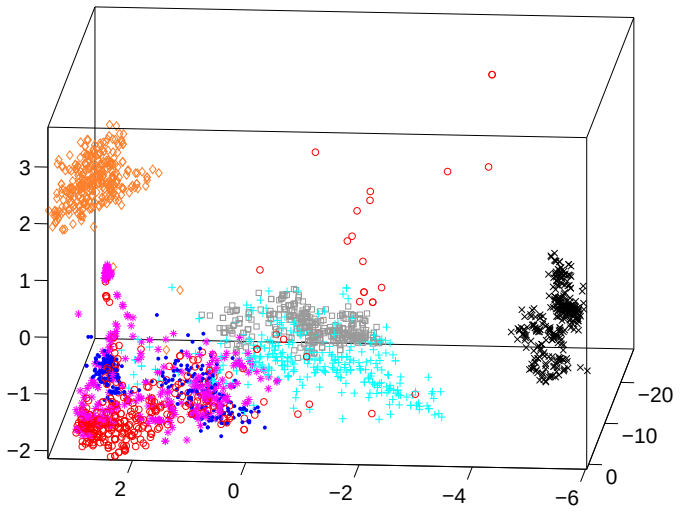
4 Summary and Future Work

We have proposed a data clustering approach based on multiple clustering objectives. By using an independent goodness function to evaluate the clusters, the proposed algorithm picks the best set(s) of objective functions for different parts of the feature space, instead of relying on a single objective function (via choosing a specific clustering algorithm). Our experimental results on both synthetic and real world data sets show that the clusters obtained can be of superior quality when compared with the partitions generated by individual clustering algorithms.

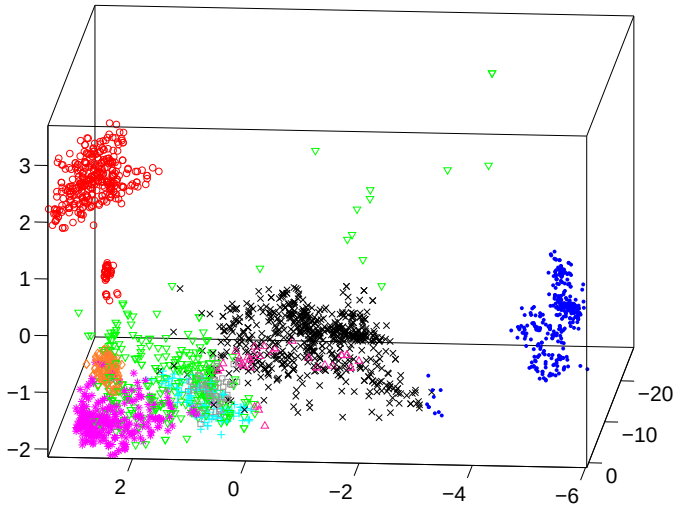
The proposed method is based on the principle of data space partitioning, where different learning algorithms are applied to different parts of the data space. This is desirable in data clustering because clusters in different regions can be of different shapes. Moreover, clusters in different parts of the data space can have different data densities. Some clustering algorithms (e.g., k-means) can have difficulties in identifying low and high density clusters simultaneously. By applying different algorithms, or the same algorithm with different parameters, we have a much higher chance to recover those clusters.

There are two types of parameters in our approach. The parameters of the candidate clustering algorithms should be set in such a way that the algorithms have a reasonable chance to detect the underlying clusters. This is achieved by using a wide range of parameter values: if the parameters lead to a partition with a single big cluster, or the number of significant clusters considerably deviates from the prior estimate of the number of clusters, the parameters should be adjusted. The second type of parameters includes γ_1 and γ_2, which control the penalty for conflicting and unassigned data points. If the multiobjective clustering solution contains too many conflicting data points, γ_1 should be increased. If too many data points are not assigned to any cluster, γ_2 should be increased. On the other hand, if clusters from a single partition are selected despite their low quality, then the penalty for conflict and incomplete coverage is too large and γ_1 and γ_2 need to be decreased.

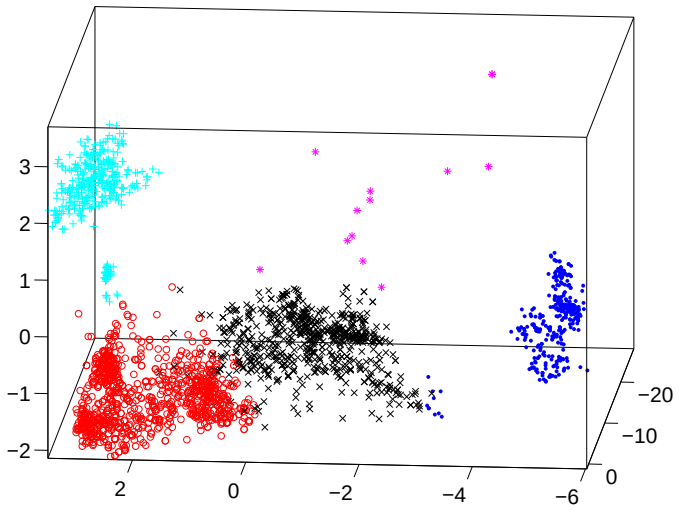
There are several directions for future work. The number of conflicting and unassigned data points can be expressed differently in the meta-objective functions in Equations (7) and (9). Goodness functions that are based on principles other than stability can be considered. If side-information like constraints on cluster labels is available, they can be incorporated into the goodness function. Our algorithm breaks down when the "effective" regions of different clustering objectives overlap significantly. How to perform mul-



(a) `image`, true class labels



(b) `image`, K fixed to 7, optimize Eq. (7)



(c) `image`, variable K, optimize Eq. (9)

Figure 4: Clusters obtained by multiobjective clustering for the `image` data set when Equations (7) and (9) are optimized. The first three principal components are shown.

$$\left[\begin{array}{cccccccc|cc} 126 & 134 & 0 & 8 & 2 & 0 & 0 & & 56 & 4 \\ 0 & 0 & 330 & 0 & 0 & 0 & 0 & & 0 & 0 \\ 0 & 0 & 0 & 179 & 27 & 1 & 6 & & 117 & 0 \\ 0 & 0 & 10 & 2 & 260 & 11 & 0 & & 20 & 27 \\ 0 & 0 & 0 & 125 & 19 & 54 & 38 & & 87 & 7 \\ 0 & 0 & 0 & 0 & 330 & 0 & 0 & & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 328 & & 1 & 0 \end{array}\right]$$

(a) K is fixed to 7

$$\left[\begin{array}{cccc|cc} 0 & 6 & 324 & 0 & 0 & 0 \\ 330 & 0 & 0 & 0 & 0 & 0 \\ 0 & 27 & 284 & 6 & 13 & 0 \\ 10 & 287 & 33 & 0 & 0 & 0 \\ 0 & 26 & 266 & 38 & 0 & 0 \\ 0 & 330 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 328 & 0 & 0 \end{array}\right]$$

(b) K varies; 4 clusters are detected

Figure 5: Confusion matrices between the cluster labels and the true class labels in Fig. 4. Row: true classes; column: cluster labels; first italic column: no. of unassigned points; second italic column: no. of conflicting points.

tiobjective clustering in this very hard setting is an interesting problem. Finally, other search methods like genetic algorithms should be explored to select the target clusters.

References

[1] S. Dudoit and J. Fridlyand. Bagging to improve the accuracy of a clustering procedure. *Bioinformatics*, 19(9):1090–1099, 2003.

[2] B. Efron and R. J. Tibshirani. *An Introduction to the Bootstrap*. Chapman & Hall, 1993.

[3] B. Fischer and J. M. Buhmann. Path-based clustering for grouping of smooth curves and texture segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(4):513–518, 2003.

[4] A. Fred and A. K. Jain. Robust data clustering. In *Proc. of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. II–128–133, 2003.

[5] A. K. Jain and R. C. Dubes. *Algorithms for Clustering Data*. Prentice Hall, 1988.

[6] A. K. Jain and J. Moreau. Bootstrap techniques in cluster analysis. *Pattern Recognition*, 20(5):547–568, 1987.

[7] E. Johnson and H. Kargupta. Collective, hierarchical clustering from distributed, heterogeneous data. In *Large-Scale Parallel KDD Systems*, pp. 221–244. Springer-Verlag, 1999.

[8] T. Lange, M. L. Braun, V. Roth, and J. M. Buhmann. Stability-based model selection. In *Advances in Neural Information Processing Systems 15*. MIT Press, 2003.

[9] M. Meila. Comparing clusterings by the variation of information. In *Proc. of Computational Learning Theory*, 2003.

[10] A. Y. Ng, M. I. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In *Advances in Neural Information Processing Systems 14*, pp. 849–856. MIT Press, 2002.

[11] Y. Qian and C. Suen. Clustering combination method. In *Proc. International Conference on Pattern Recognition-Vol. 2*, pp. 736–739, 2000.

[12] A. Strehl and J. Ghosh. Cluster ensembles – a knowledge reuse framework for combining multiple partitions. *Journal of Machine Learning Research*, 3:583–617, December 2002.

[13] C. T. Zahn. Graph-theoretic methods for detecting and describing gestalt clusters. *IEEE Transactions on Computing*, 20(31):68–86, 1971.