Dual Clustering: Integrating Data Clustering over Optimization and Constraint Domains

Cheng-Ru Lin, Ken-Hao Liu, and Ming-Syan Chen, *Fellow*, *IEEE*

Abstract—Spatial clustering has attracted a lot of research attention due to its various applications. In most conventional clustering problems, the similarity measurement mainly takes the geometric attributes into consideration. However, in many real applications, the nongeometric attributes are what users are concerned about. In the conventional spatial clustering, the input data set is partitioned into several compact regions and data points which are similar to one another in their nongeometric attributes may be scattered over different regions, thus making the corresponding objective difficult to achieve. To remedy this, we propose and explore in this paper a new clustering problem on two domains, called dual clustering, where one domain refers to the optimization domain and the other refers to the constraint domain. Attributes on the optimization domain are those involved in the optimization of the objective function, while those on the constraint domain specify the application dependent constraints. Our goal is to optimize the objective function in the optimization domain while satisfying the constraint specified in the constraint domain. We devise an efficient and effective algorithm, named Interlaced Clustering-Classification, abbreviated as ICC, to solve this problem. The proposed ICC algorithm combines the information in both domains and iteratively performs a clustering algorithm on the optimization domain and also a classification algorithm on the constraint domain to reach the target clustering effectively. The time and space complexities of the ICC algorithm are formally analyzed. Several experiments are conducted to provide the insights into the dual clustering problem and the proposed algorithm.

Index Terms—Data mining, data clustering, dual clustering.

1 Introduction

Data clustering has been identified as an important technique for many applications, including similarity search, pattern recognition, trend analysis, marketing analysis, grouping, classification of documents, and so forth [4], [6], [10], [12]. In general, there are two types of attributes associated with the data points in data clustering, i.e., numerical attributes and categorical attributes. Numerical attributes are those with ordered values, such as the height of a person and the speed of a moving vehicle. Categorical attributes are those with unordered values, such as the kind of a drink and the brand of a car.

Since the early work in the k-means algorithm [20], data clustering has been studied for years and several technologies have been developed, including the nearest neighbor clustering [19], fuzzy clustering [2], partitional clustering [8], hierarchical clustering [23], hybrid clustering [18], artificial neural networks for clustering [13], and support vector clustering [1], to name a few. Among others, spatial clustering is an important technique in many applications. In most conventional clustering problems, the similarity measurement mainly takes the geometric attributes into consideration. However, in many real applications, the nongeometric attributes are what users are concerned about. Data points which are similar to one another in their nongeometric attributes may be scattered geometrically. By using a conventional clustering algorithm, it is infeasible to partition the geometric region so that the data points in each subregion are similar to each other in their nongeometric attributes. To remedy this, we propose and explore in this paper a new clustering problem in two domains, named dual clustering, where one domain refers to the optimization domain and the other refers to the constraint domain. Attributes in the optimization domain are involved in the optimization of the objective function, while those in the constraint domain need to comply with the geometric constraints. In the dual clustering problem, we try to partition the data set into several groups, so that these groups form nonoverlapping compact regions in the constraint domain while minimizing the dissimilarity of the data points in a group on the optimization domain. A formal description of the dual clustering problem will be given in Section 2. The dual clustering problem can be best understood by the example shown in Fig. 1. Each data point in Fig. 1 consists of four attributes of which two are in the constraint domain and the other two are in the optimization domain. For the dual clustering problem, a possible solution of the input data set is shown in Fig. 1. Note that points of different clusters are shown with different symbols. The projection of the input data set in the constraint domain is shown in Fig. 1a and that in the optimization domain is shown in Fig. 1b. Note the data point A in Fig. 1b. In conventional clustering, A will most likely to be grouped into cluster $\mathbf{C}_2$. However, in this example, data point A belongs to cluster $\mathbf{C}_1$ instead. Consider the projection on constraint domain, i.e., Fig. 1a. Data point A is at the center of cluster $\mathbf{C}_1$. If data point A is assigned to cluster $\mathbf{C}_2$, then clusters $\mathbf{C}_2$ and $\mathbf{C}_1$ will not form nonoverlapping compact regions any more. If we move not only point A but also

• *The authors are with the Department of Electrical Engineering, National Taiwan University, Taipei, Taiwan, ROC.*
E-mail: mschen@cc.ee.ntu.edu.tw, {owenlin, kenliu}@arbor.ee.ntu.edu.tw.

Manuscript received 4 Jan. 2004; revised 17 Oct. 2004; accepted 10 Dec. 2004; published online 17 Mar. 2005.
For information on obtaining reprints of this article, please send e-mail to: tkde@computer.org, and reference IEEECS Log Number TKDE-0002-0104.

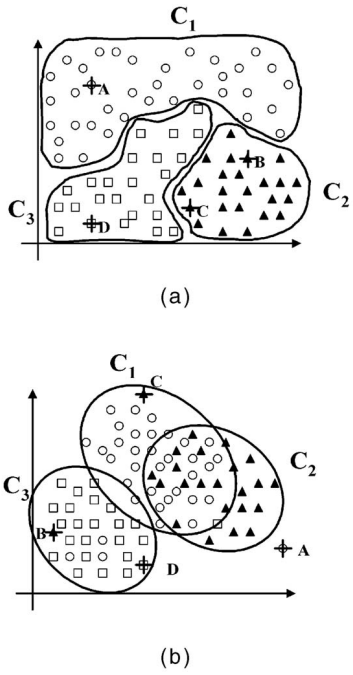


Fig. 1. An illustrative example of the dual clustering problem. (a) Projection on constraint domain. (b) Projection on optimization

other necessary points from cluster $\mathbf{C}_1$ to cluster $\mathbf{C}_2$ just to comply with the compact region constraint, the clustering cost in the optimization domain will be increased. The compact region constraint accounts for the reason that those points, such as A, B, and C, do not belong to their closest clusters in the optimization domain.

In our opinion, the dual clustering problem cannot be dealt with by any direct extension of existing clustering algorithms as is evident from the above example. Consequently, new methods are called for to handle this new clustering problem. In this paper, we devise a new approach to solve the problem, named Interlaced Clustering and Classification algorithm, abbreviated as ICC. Algorithm ICC integrates the information in both domains by an interlaced process of executing clustering and classification on the input data set. Specifically, for better stability and quality, we employ the complete-link algorithm as our clustering method and the Support Vector Machine (SVM) as our classification method. In essence, the ICC algorithm, on one hand, aims at having the optimal clustering results in the optimization domain and, on the other hand, gradually shapes the resulting clustering better in accordance with spatial clustering in the constraint domain. The ICC algorithm effectively reaches the target clustering by iteratively performing the clustering (by the complete-link algorithm) and the classification (by SVM). The time and space complexities of the ICC algorithm are analyzed as $O(l \cdot (n^2 \log n + k^2 \cdot n \cdot t))$, where k is the number of clusters, l is a small integer specified by users, and t is the number of iterations taken by the SVM algorithm to converge at the training phase [5], [15]. In addition, we conduct several experiments to exhibit the properties of the proposed the ICC algorithm. To the best of our knowledge, there were no prior works on the dual clustering. We hence deliberately design two alternative algorithms in this paper for comparison purposes. In contrast to the ICC algorithm which moves back and forth between the optimization domain and the constraint domain, another approach to solve the dual clustering problem is to modify the similarity measure in the optimization domain so as to explicitly specify the penalty in the constraint domain. This algorithm is referred to as a cost revision algorithm. By doing so, conventional clustering algorithms can be applied in the optimization domain. In addition, we devise another algorithm for this dual clustering problem, called k-NN Clustering, which is in essence an extension to KNN classification algorithm. The performance comparison results provide insights into the dual clustering problem and also show the advantages of the ICC algorithm.

We mention in passing that several prior works have been conducted on data clustering with some constraints. The work in [24] defines a taxonomy of constraints for clustering with the focus on exploring the constraints which can be formulated with SQL aggregates and imposed on individual clusters. Some works are proposed to cope with the total mass constraints [21], [22]. In the situation of high dimensions and large cluster numbers, the phenomena of empty clusters or clusters with very few items are observed. The constraint that each cluster has to contain a minimum number of points is added to the k-means clustering algorithm to deal with such phenomena [3]. Also, the work in [17] focuses on the continuous constraint: All the data points in each cluster form a continuous region on the time sequence. The work in [7] studies a problem that the data points of each group must be within a specified range in those constraint attributes. The clustering techniques for spatial data in presence of physical constraints are discussed in [9], [25], [29]. The work in [11] proposed a kernel function for structured data and applied the kernel with a modified k-means algorithm on spatial data clusters. However, by that kernel function, the attributes in the constraint domain are only treated with higher weights, but the clusters are not constrained to form compact regions. These methods are, however, not applicable to the dual clustering problem addressed in this paper which, in fact, occurs in many real applications, such as public facility allocation, electioneering planning, and market analysis, to name a few. As pointed out earlier, despite its importance, no prior work explicitly explored this dual clustering problem, let alone developing solution algorithms for it. This fact distinguishes this paper from others.

The rest of this paper is organized as follows: Some related works and the problem description are given in Section 2. Section 3 presents the proposed ICC algorithm to deal with this dual clustering problem. The experimental studies are presented in Section 4. This paper concludes with Section 5.

2 Preliminaries

We will first introduce related technologies in Section 2.1 and then present the formal problem definition of the dual clustering problem in Section 2.2.

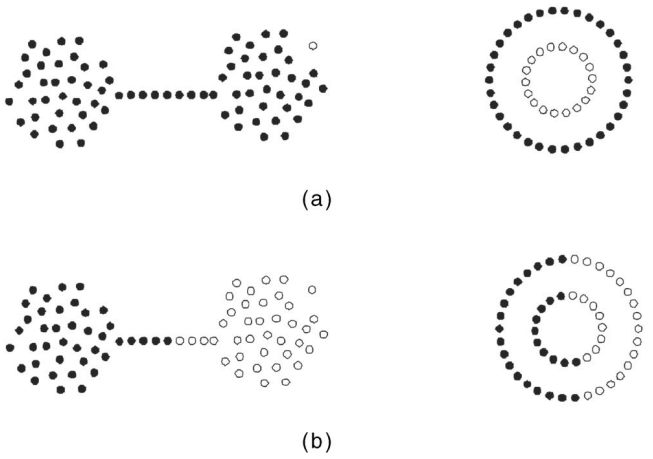


Fig. 2. Examples of the clustering capabilities of the single-link and complete-link clustering algorithms. (a) Clustering results of the single-link algorithm. (b) Clustering results of the complete-link algorithm.

2.1 Related Technologies

2.1.1 Clustering Techniques

A hierarchical clustering algorithm starts with every input data instance as a subcluster and then continuously merges the two closest subclusters until a predefined number of clusters are found. Most existing hierarchical clustering algorithms are variations of the single-link and complete-link algorithms [16], [23]. Both algorithms require the time complexity of $O(n^2 \log n)$, where n is the number of data points. The space complexity of the single-link algorithm is $O(n)$, while that of the complete-link algorithm is $O(n^2)$. The outline of a general hierarchical clustering algorithm is given below.

Hierarchical Clustering Algorithm
1. Initially, each data point forms a cluster by itself.
2. The algorithm repetitively merges the two closest clusters.
3. Output the hierarchical structure constructed.

A single-link clustering algorithm and a complete-link one differ in the intercluster distance measurement, i.e., Step 2. The single-link algorithm uses the distance between the two closest points of the two clusters as the intercluster distance, i.e.,

$$d(C_i, C_j) = \min\{d(o_i, o_j) | o_i \in C_i, o_j \in C_j\},$$

while the complete-link algorithm uses the distance of two farthest points as the intercluster distance, i.e., $d(C_i, C_j) = \max\{d(o_i, o_j) | o_i \in C_i, o_j \in C_j\}$.

As shown in Fig. 2a, the single-link algorithm suffers from so-called chaining effect and the complete-link clustering algorithm has problems in dealing with particular shapes such as circles shown in Fig. 2b. Because of the difference of the intercluster distance, the single-link algorithm can find the clusters of any shape while the complete-link algorithm finds isotropic clusters.

2.1.2 Classification Techniques

While clustering is called an unsupervised learning, classification is said to be a supervised learning. In general, the classification has two phases, i.e., training phase and testing phase. In the training phase, we build a classifier from the training data set. In the testing phase, we classify the testing data with the classifier. Several techniques have been proposed for classification, including neural network, SVM, decision tree, regression model, and so on.

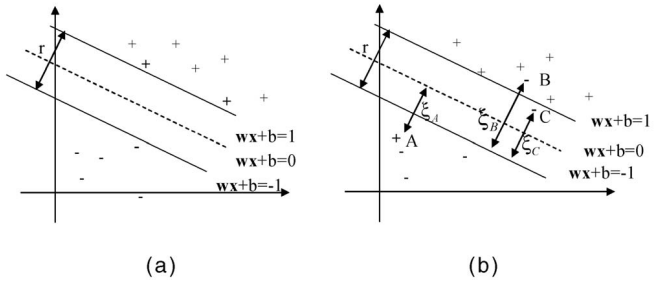


Fig. 3. Illustrative examples of the SVM algorithm with linear separable and linear inseparable data sets. (a) Linear separable and (b) linear inseparable.

The k-nearest neighbor (abbreviated as KNN) classification algorithm is one of the most intuitional algorithms [28]. In the training phase, it simply records all the training data. In the testing phase, KNN selects the k most similar cases in the training data set for the given tested entry and use their categories as votes to decide the category of the tested entry. The KNN algorithm is quite simple while having high accuracy at testing. The main drawback of the KNN algorithm is the need of keeping the whole training set while testing, which makes KNN infeasible when the training set becomes large.

In recent years, the Support Vector Machines (abbreviated as SVM) has been developed and applied to many cases [1], [11]. It also has proven itself as one of the most accurate classifiers. The SVM is a new generation of learning system based on statistical learning theory [26], [27]. The SVM model only does the binary classification. However, it is easy to extend SVM to do multicategory classification. Among various methods, the "one-against-one" approach [14] constructs a classifier for each pair of classes and then uses a voting strategy to designate a point to be in the class with the maximum number of votes. The idea behind SVM is to map the input data set to another feature space in order to find an optimal hyperplane which separates the two-category input data set. More specifically, given a data set of n points $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$ and their category vector $\{y_1, y_2, \ldots, y_n\}$, where $y_i = 1$ or -1, SVM tries to find an optimal hyperplane, $\mathbf{wx} + b = 0$, which separates the two categories of input data set, i.e., $y_i(\mathbf{wx}_i + b) \geq 1$, as shown in Fig. 3a. The hyperplane is said to be optimal if it separates the two categories with the maximum margin, i.e., r. It can be proved that $r = \frac{2}{\|w\|}$, so maximizing the margin is minimizing the value of $\|w\|$. However, the input data set could be linearly inseparable. In that case, SVM adds a soft margin, ξ_i, for each data point $\mathbf{x}_i$, and the criteria becomes $y_i(\mathbf{wx}_i + b) \geq 1 - \xi_i$. An example of three misclassified points is shown in Fig. 3b. The SVM is then trying to

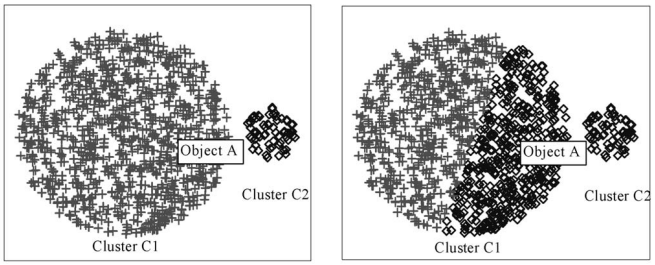


Fig. 4. Possible results of spatial clustering and general clustering. (a) Result of spatial clustering and (b) result of general clustering.

minimize the cost: $\frac{1}{2}\|w\|^2+C\sum_{i=1}^{n}\xi_i^2$, where C is a positive value representing the tolerance of such misclassified data points. It has been proven that with a larger margin, i.e., smaller $\|w\|$, the classifier has better generality, meaning it can be applied to unseen data set better. Thus, by controlling the value of C, we can find the balance between the training error and the testing error. In this paper, we control the number of those *outliers* by controlling the value of C.

In concept, SVM maps the data points to another feature space. However, it is not necessary to find the mapping, $\Phi(x_i)$, explicitly. Instead, one only has to find the *kernel* of the feature space which defines the inner product of two input data points on the feature space, i.e., $K(x_i,x_j)=\langle\Phi(x_i),\Phi(x_j)\rangle$. Moreover, a kernel can be obtained by a combination of other kernels. This property allows us of mapping the data points to the various feature space directly.

2.2 Problem Description

The conventional spatial clustering problem is trying to partition the input data into several compact regions. The general clustering problem is trying to partition the data into several groups so that the data instances in a group are similar to one another. Though being similar to each other, these two clustering problems differ in a subtle aspect. Specifically, by spatial clustering, we form clusters as shown in Fig. 4a, while the result of a general clustering is the one shown in Fig. 4b. These figures are expected to be viewed in color. Bigger versions of figures in this paper can be found in http://arbor.ee.ntu.edu.tw/dualclus. In the spatial clustering problem, we require each cluster to form a compact region. As a result, although being closer to the points of cluster $C2$, the data point A in Fig. 4a still belongs to cluster $C1$. However, in the general clustering problem, we try to keep data points in a cluster as similar to each other as possible. The data point A hence belongs to cluster $C2$ as shown in Fig. 4b. The dual clustering problem we studied in this paper is to conduct the general clustering in the optimization domain and the spatial clustering in the constraint domain. Explicitly, we use the following formula as our clustering cost measurement.

Definition. *Given a clustering partition $P=\{C_1,C_2,\ldots,C_k\}$ of the input data set of n points $\{o_1,o_2,\ldots,o_n\}$, we define the clustering cost as*

$$Cost(P)=\frac{\sum_{C_i}\sum_{o\in C_i}(o-C_i.center)^2}{n},$$

where $C_i.center$ is the center of cluster C_i.

The problem of the dual clustering can be stated below.

Problem of Dual Clustering. *Given a data set of n points: $\{o_1,o_2,\ldots,o_n\}$, where each point consists of several attributes of two domains, constraint domain and optimization domain, the dual clustering problem is the process to partition the input data set into several groups in such a way that each group forms a compact region in the constraint domain while minimizing the clustering cost in the optimization domain. A compact region is a connected region of a simple shape whose boundary is a simple closed curve.*

3 Algorithm of Dual Clustering

To solve the problem of dual clustering, the Interleaved Clustering-Classification (ICC) algorithm is developed in Section 3.1. We also explore the properties of the ICC algorithm in Section 3.2. Finally, an illustrative example is presented in Section 3.3.

3.1 Interlaced Clustering-Classification

To facilitate our presentation, we use $dist_{opt}(o_i,o_j)$ to represent the distance between data points o_i and o_j in the optimization domain and $dist_{cons}(o_i,o_j)$ for the distance in the constraint domain. Then, the ICC algorithm for dual clustering can be presented as follows:

ICC Algorithm

//Given a data set, $\{o_1,o_2,\ldots,o_n\}$, a desired cluster number in optimization domain, k, weight in optimization domain, w, and the execution level, l.

1. Perform the complete-link clustering algorithm with the distance measurement

$$dist_{hybrid}(o_i,o_j)=\sqrt{w\cdot dist_{opt}(o_i,o_j)^2+(1-w)\cdot dist_{cons}(o_i,o_j)^2}$$

   to form a partition P of k clusters.
2. Set $t=1$.
3. Assign different labels to the clusters found in P. Mark each data points with the same label of its own cluster.
4. Apply the SVM training algorithm to the attributes in the constraint domain along with the labels marked in previous step to obtain a classifier C.
5. Apply the classifier C on the whole data set to form a clustering result. The data points classified to the same group belong to the same cluster.
6. Split each disconnected cluster into two or more connected clusters.
7. If $t=l$, then output the clustering result obtained in previous step.
8. Perform the complete-link algorithm in the optimization domain to form a k-cluster partition P with distance definition:

$$dist_{ICC}(o_i, o_j) = \sqrt{\alpha \cdot dist_{opt}(o_i, o_j)^2 + (1-\alpha) \cdot \delta(C(o_i), C(o_j))^2}$$

$$\text{where } \delta(x,y) = \begin{cases} dist_{cons}(o_i, o_j) & \text{if } x = y \\ 1 & \text{otherwise} \end{cases}$$

$$\alpha = 0.5 + \frac{t-1}{2l}, \text{ and } C(o_i) \text{ is the category of } o_i.$$

9. Set $t = t + 1$ and goto Step 3.

The ICC algorithm first partitions the input data set with a hybrid distance measure which uses the information in both domains. The weight in the optimization domain w is used to specify an initial partition for later interlaced classification and clustering. After determining the clusters by the hybrid distance measurement, the ICC algorithm tries to mark the scope of each cluster in the constraint domain. In Step 3, each data point in the same cluster is marked with a unique label. In Step 4, the ICC algorithm trains a SVM classifier to learn the scope of each cluster on constraint domain. The one-against-one approach is adopted to do multicategory classification. In Step 5, the knowledge learned by the SVM model is outputted by applying the SVM classifier on input data set. However, the SVM classifier C may generate some disconnected regions for a cluster. Thus, in Step 6, the ICC algorithm further refines the partition results by splitting those disconnected regions with a single-link clustering algorithm. In this splitting procedure, the distance between two data points is defined as

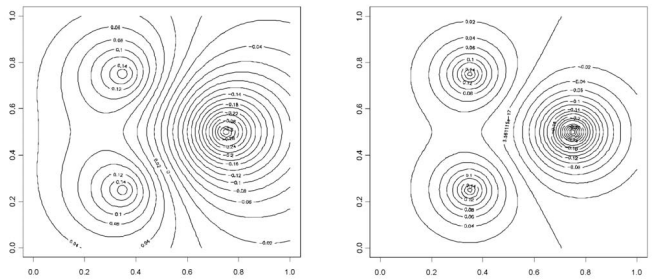
$$dist_{split}(o_i, o_j) = \begin{cases} \infty & \text{if } C(o_i) \neq C(o_j) \\ dist_{cons}(o_i, o_j) & \text{otherwise.} \end{cases}$$

The single-link clustering algorithm merges the two closest subclusters until a significant increase in the merged distances is being detected. After that, the ICC algorithm integrates the information of the compact region constraint into the clustering process, i.e., the distance measured at Step 8. The execution level l specifies the granularity level of the interlace loop. Then, in the subsequent iterations, the ICC algorithm gradually and linearly increases the importance of the information found in the constraint domain by increasing the value of α. The concept behind the design of the function for the value of α is trying to integrate the information in the constraint domain gradually and linearly into the optimization domain. By doing so, the ICC algorithm tries to reduce oscillating range and, thus, results in a stable clustering solution in optimization domain. In essence, the ICC algorithm, on one hand, aims to have the optimal clustering results in the optimization domain and, on the other hand, gradually shapes the resulting clustering better in accordance with spatial clustering in the constraint domain. The ICC algorithm effectively reaches the target clustering by iteratively performing the above two interlaced procedures (i.e., Step 4, 5, and Step 8). Finally, it outputs the clustering results at Step 7.

3.2 Properties of the ICC Algorithm

In our implementation, we use the Gaussian kernel in the SVM algorithm, i.e.,

$$K(o_i, o_j) = \exp\left(-\frac{\|o_i - o_j\|^2}{2\sigma^2}\right).$$



(a) (b)

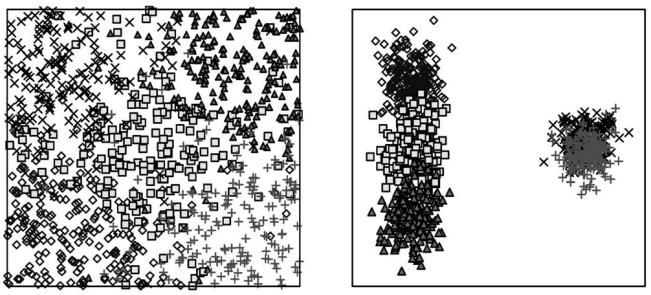
Fig. 5. The projections of hyperplanes in the feature space of Gaussian kernel.

Note that the parameter σ plays an important role in this algorithm. When σ is small, we will have many sparse and small clusters in the constraint domain; when σ grows these small clusters will be merged to form a larger cluster. The parameter σ is called the width of the Gaussian kernel. As shown in the formula, the Gaussian kernel has the form $\|o_i - o_j\|^2$ in the exponential term. Thus, all the points with the same distance from o_j will have the same value of the kernel function. Also, points far from each other will have a very small value of the kernel function. These properties are indeed instrumental to the clustering.

Furthermore, omitting its proof, we can show that the normal vector of the hyperplane in SVM, i.e., w, is a linear combination of some of input data points on the feature space. More specifically, $w = \sum_{i=1}^{l} \alpha_i \cdot \phi(x_i)$. Those vectors with nonzero α are called support vectors. Thus, the hyperplane in the feature space, $w \cdot \phi(x_i) + b = 0$, can be rewritten as $\sum_{i=1}^{m} \alpha_i \cdot K(sv_i, x) + b = 0$, where m is the number of support vector, and sv_i, $i \in [1, m]$, are the support vectors. The projection on the original space of a hyperplane with three support vectors is shown in Fig. 5. The left two support vectors belong to a class and the right one belongs to the other. As shown in Fig. 5, the support vectors are the centers of these circles of these contour plots. The contour plots are formed by different values of b. It can be verified that the value of b controls the balance of the margin of the two classes, while the value σ controls the granularities of the partition.

Theorem 1. *The time complexity of the ICC algorithm is $O(l \cdot (n^2 \log n + k^2 \cdot n \cdot t))$, where k is the number of clusters and t is the number of iterations taken by the SVM algorithm to converge at the training phase.*

Proof. During the clustering phase, we use the complete-link clustering technique with only a modified distance measure, which has time complexity of $O(n^2 \log n)$. During the classification phase, the time complexity of SVM to train a single classifier is $O(n \cdot t)$, where t is the number of iterations in the SVM training algorithm [5], [15] and a total of $k(k-1)/2$ support vector classifiers are constructed to perform the k-class classification. Therefore, the time complexity of the classification phase is $O(k^2 \cdot n \cdot t)$. In addition, the time complexity of SVM to apply the classifier to the input data set is $O(mn)$, where m is the number of support vectors. Since the number of support vectors is always



(a) (b)

Fig. 6. Data Set 1. (a) Projection on the constraint and (b) projection on the optimization domain.

TABLE 1
The Clusters of Data Set 1

Size	Center on Cons.		Center on Opt.	
170	0.7696	0.2486	0.8013	0.4467
180	0.2215	0.7333	0.7980	0.5390
203	0.4888	0.4986	0.2093	0.4924
221	0.7682	0.7620	0.2012	0.1041
226	0.2477	0.2186	0.1999	0.8936

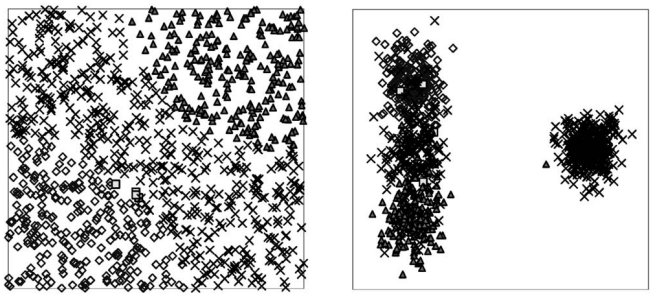
smaller than the size of input data set, the time complexity is subsumed by the term $O(n^2 \log n)$. With the number of iterations being l, the time complexity of the ICC algorithm is $O(l \cdot (n^2 \log n + k^2 \cdot n \cdot t))$. □

Theorem 2. *The space complexity of the ICC algorithm is* $O(n^2)$.

Proof. The space complexities of both complete-link and SVM are $O(n^2)$. This theorem follows. □

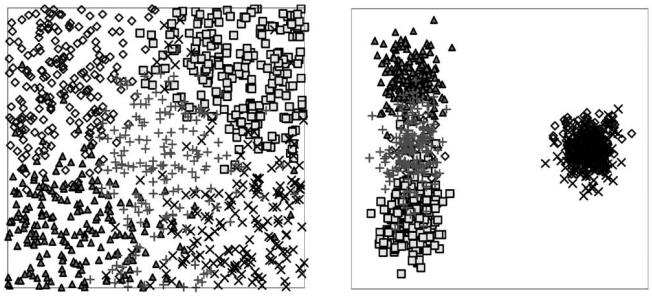
3.3 An Illustrative Example

Consider the Data Set 1 shown in Fig. 6 where each point consists of four attributes, two in the constraint domain and the other two in the optimization domain. There are 1,000 data points in this data set, i.e., $n = 1,000$. Since it is difficult to present a 4-dimensional data set on a plane, we display the data set by showing the projection of each cluster in both domains. Note that the clusters shown in Fig. 6 are the output of our data generation procedure, which will be described in Section 4, and can be taken as the target answer of this example. Detail information, such as the number of points and center of each group, is shown in Table 1. We execute the ICC algorithm with $k = 5$, $w = 0.7$, and $l = 5$. After executing a complete-link clustering algorithm, i.e., Step 1 of the ICC algorithm, five clusters are found as shown in Fig. 7. It is seen that this intermediate clustering result is not ideal because two of them are very close in our input data set. Then, we execute the SVM algorithm with $C = 1$ by the Gaussian kernel with $\sigma = \sqrt{n}$ in the constraint domain (i.e., Fig. 7a) and try to locate the scope of each cluster. The SVM model is able to identify the scopes of only four clusters, as shown in Fig. 8. We next execute the clustering algorithm with modified distance measurement described in the Step 6 of the ICC algorithm with $\alpha = 0.5$. The partition is shown in Fig. 9 where the input data set is accurately partitioned. Then, we use SVM algorithm to identify the scope of each cluster more clearly in the constraint domain and employ a complete-link clustering



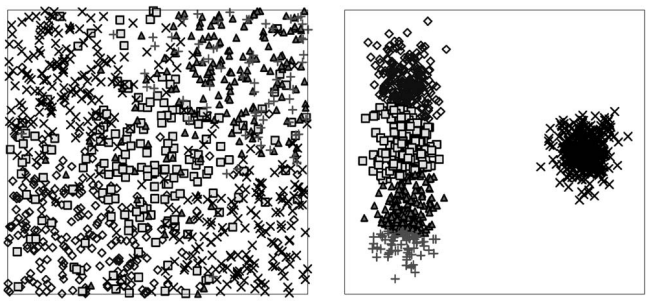
(a) (b)

Fig. 8. The classification result of Step 4 in the ICC algorithm. (a) Projection on the constraint domain and (b) projection on the optimization domain.



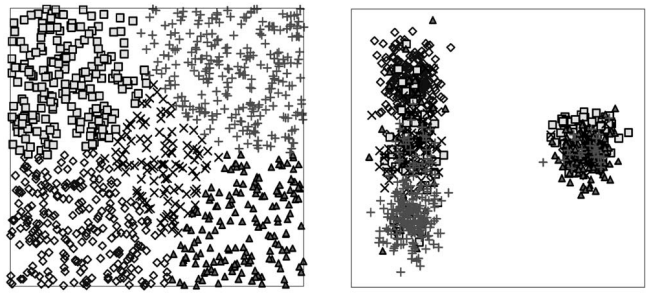
(a) (b)

Fig. 9. The output of Step 6 in the ICC algorithm. (a) Projection on the constraint domain and (b) projection on the optimization domain.



(a) (b)

Fig. 7. The output of Step 1 in the ICC algorithm. (a) Projection on the constraint domain and (b) projection on the optimization domain.



(a) (b)

Fig. 10. The final result of algorithm. (a) Projection on the constraint domain and (b) projection on the optimization domain.

TABLE 2
The Clusters found by ICC in Data Set 1

Size	Center on Cons.		Center on Opt.	
113	0.5143	0.4356	0.2857	0.4905
170	0.7977	0.2190	0.6876	0.4844
207	0.1956	0.7608	0.6543	0.5356
245	0.2357	0.2150	0.2379	0.8224
265	0.7688	0.7655	0.2607	0.1930

algorithm to adjust the partition in the optimization domain with $\alpha = 0.75$. Finally, we complete the ICC algorithm by using SVM to get the final partition, as shown in Fig. 10. The detail information of the clusters found by the ICC algorithm is shown in Table 2.

4 Experimental Studies

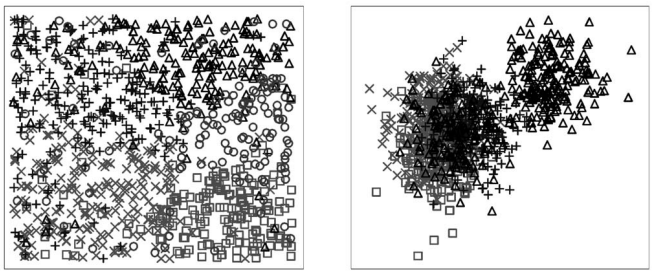
In this section, we conduct a series of experiments to assess the performance of the ICC algorithm. These experiments are performed on a computer with a P4 1.7 Ghz Intel CPU and 896MB of memory. Section 4.1 describes the generation of the synthetic data. In Section 4.2, we show the clustering quality by some visualization output. The scaling-up experiments are performed in Section 4.3. We also explore the impact of parameter w in Section 4.4. We present performance comparison results between the ICC algorithm and other algorithms in Section 4.5. The SVM module used in our algorithm is implemented by our colleagues, and can be accessed by http://www.csie.ntu.edu.tw/~cjlin/libsvm.

4.1 Data Generation

We use synthetic data as the input of our experiments so as to control the features of the data generated. The parameters used in data generation are listed in Table 3. There are two attributes in the constraint domain and another two attributes in the optimization domain. First, for each cluster, we select the cluster centers in both domains, denoted by C_i^{opt} and C_i^{cons}, where $i = 1 \ldots k$ and k is the number of clusters. These data points are selected uniformly from a square of width one. Then, we randomly generate a point, p, in the constraint domain and decide the cluster to which this point belongs by the following formula:

$$P(p \in Cl_i) = \frac{\frac{1}{dist\left(p, C_i^{cons}\right)^f}}{\sum_{j=1}^{k} \frac{1}{dist\left(p, C_j^{cons}\right)^f}},$$

where $dist(p, q)$ is the distance between p and q in the constraint domain. With this probability distribution, point p is inclined to belong to a cluster whose center is close to p. Note that, parameter f is used to control the randomness. If $f = 0$, then the point will join any of the k clusters with same probability. For a large value of f, the point will be assigned to the cluster with the closest center with a probability close to one. Suppose that point p is assigned to Cl_i. We next decide the attributes of p in the optimization domain by a normal distribution with C_i^{opt} being the center and r as the standard deviation. We repeat this procedure to generate n points. An example of the generated data is shown in Fig. 11. The detail information of the generated clusters is shown in Table 4.



(a) (b)

Fig. 11. Data Set 2 (generated with $k = 5$, $f = 4$, $r = 0.15$, and $n = 1,000$). (a) Projection on the constraint domain and (b) projection on the optimization domain.

4.2 Experiment I: Clustering Quality

In this experiment, we apply the ICC algorithm on a complex data set, referred to as Data Set 2, in Fig. 11 in order to demonstrate the clustering qualities achieved. It can be noted that in this data set, three clusters are very close to each other and the other two are also close to each other in the optimization domain. As shown in Fig. 12, although clusters are heavily overlapped with one other, the ICC algorithm can still partition them into a considerably good clustering result. The detail information of the clusters found is also shown in Table 5. The ICC algorithm is performed with parameters settings as $k = 5$, $w = 0.5$, and $l = 3$. The equal weights in both domain are chosen such that our initial partition does not show bias toward either the constraint domain or the optimization domain. For our data set, the selected execution level suffices and any further increase in the value generates clustering results of similar quality to Fig. 12.

4.3 Experiment II: Scaling Up

The execution times of the ICC algorithm against different data set sizes are shown in Fig. 13. It can be seen that the ICC algorithm scales approximately quadratically with the size of the input data set. This conforms to our analysis on the complexity of ICC by Theorem 1.

4.4 Experiment III: On Parameter w

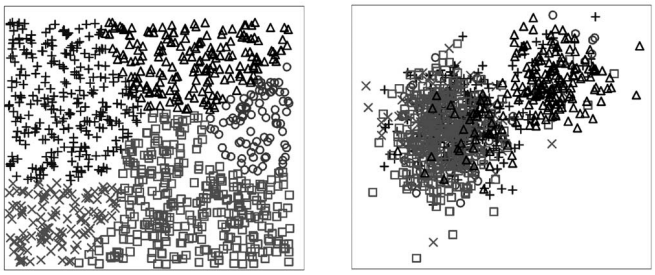
In this experiment, we execute the ICC algorithm with the parameter setting as $w = 0.9$ and $l = 1$. Step 1 of the ICC algorithm is the data preprocessing stage, where the user specifies the weight in optimization domain, i.e., w, to

TABLE 3
Parameters of Data Generation

Parameter	Description
n	Number of generated data points
k	Number of generated clusters
f	Randomness factor on the constraint domain
r	Standard deviation of the normal distribution on the optimization domain

TABLE 4
The Clusters in Data Set 2

Size	Center on Cons.		Center on Opt.	
143	0.7034	0.5434	0.2421	0.6274
187	0.7160	0.2119	0.1348	0.5148
200	0.6058	0.7961	0.8496	0.9118
232	0.2752	0.6553	0.3449	0.6600
238	0.3034	0.3071	0.1115	0.7057



(a) (b)

Fig. 12. The clustering result of the ICC algorithm on Data Set 2. (a) Projection on the constraint domain and (b) projection on the optimization domain.

produce a starting point for the later interlaced optimization. The clustering result obtained by this preprocessing stage is shown in Fig. 14. The result is not desirable since the initial overweight in the optimization domain is biasing the ICC optimization process toward the optimization domain. However, in the next step, algorithm SVM still tries to identify the geometric scope of each cluster. Note that there are too many noises on the constraint domain. After this SVM stage, these noises will be assigned to different clusters on constraint domain and, thus, greatly increase the clustering cost on the optimization domain. Finally, the results of the ICC algorithm is shown in Fig. 15. Thus, although with greater value of w, the final clustering cost will not be smaller. Similarly, with smaller value of w, a more clear clustering results on constraint domain will be obtained after the preprocessing stage. However, the clustering cost on the optimization domain will be much larger. A suitable value of w is needed to obtain a good clustering result on both domains.

4.5 Experiment IV: Comparison with Other Algorithms

As mentioned before, there were no prior works on dual clustering. Therefore, we deliberately designed two alternative algorithms here for comparison purposes. In contrast to the ICC algorithm which moves back and forth between the optimization domain and the constraint domain, another approach to solve the dual clustering problem is

TABLE 5
The Clusters found by Algorithm ICC in Data Set 2

Size	Center on Cons.		Center on Opt.	
85	0.8838	0.5896	0.2974	0.6642
125	0.1730	0.1825	0.1738	0.6638
194	0.6534	0.8379	0.7418	0.8389
274	0.2055	0.6785	0.3303	0.6959
322	0.6580	0.2494	0.1704	0.6106

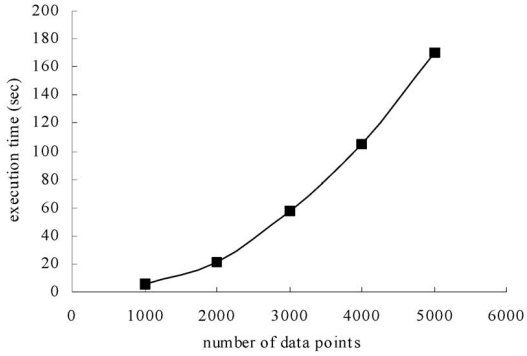


Fig. 13. Execution time against various sizes of data set.

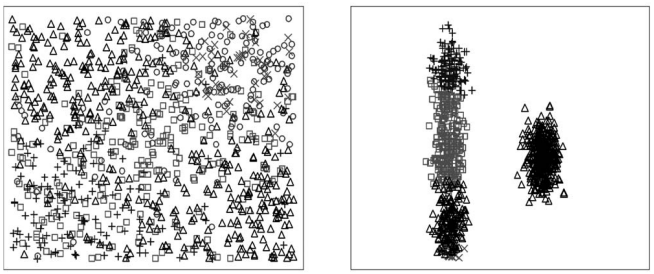


Fig. 14. The clustering output of Step 1 of the ICC algorithm with $w = 0.9$.

to modify the similarity measure in the optimization domain so as to explicitly specify the penalty factor p for the similarity measure in the constraint domain, i.e.,

$$dist_{RC}(o_i, o_j) = \sqrt{dist_{opt}(o_i, o_j)^2 + p \cdot dist_{cons}(o_i, o_j)^2}.$$

This algorithm is referred to as a cost revision algorithm (abbreviated as CR). By doing so, conventional clustering algorithms can then be applied in the optimization domain. Note that $p = \frac{1}{w}$, where w is the weight of optimization domain as in $dist_{hybrid}$ of the ICC algorithm. Different values of p is used to evaluate the clustering results on optimization domain and the respective projected clustering in the constraint domain, as shown in Fig. 16. In addition, we devise another algorithm for this dual clustering problem, called k-NN Clustering (abbreviated as KNNC). The KNNC algorithm first finds k nearest neighbors in the constraint domain for each point. Next, the KNNC algorithm uses the average values of the k neighbors in the optimization domain as the new attributes of the point and then applies a complete-link algorithm to the new attributes for obtaining a solution clustering.

We apply the above two algorithms, i.e., CR and KNNC, to the data set used in Section 3.3, and the clustering results

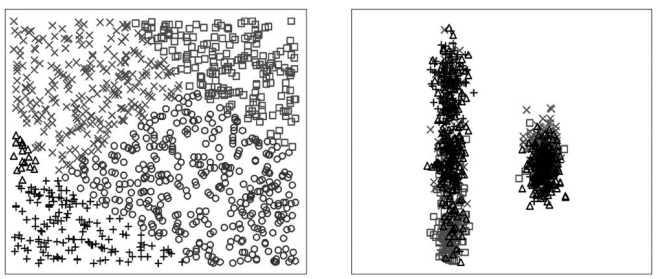


Fig. 15. The result of the ICC algorithm with $w = 0.9$.

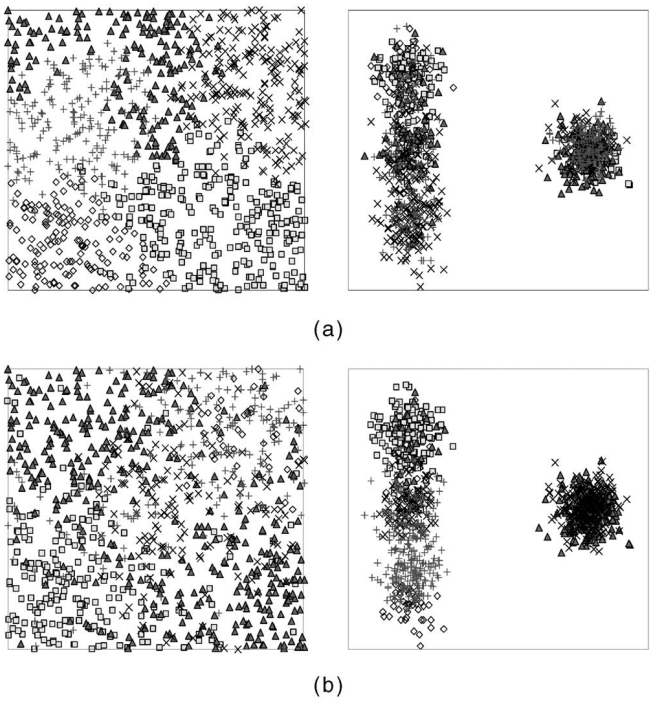


Fig. 16. Results of the CR algorithm with different values of p. (a) Clustering results with $p = 3$ and (b) clustering results with $p = 1/3$.

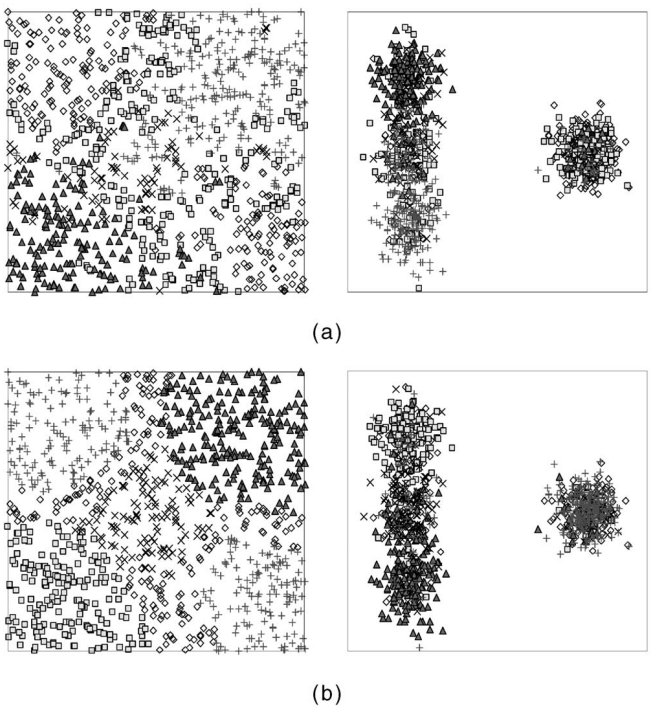


Fig. 17. Results of the KNNC algorithm with different numbers of neighbors. (a) Clustering results with $k = 5$ and (b) clustering results with $k = 30$.

in the constraint domain are shown in Figs. 16 and 17. The resulting clustering, however, tends to show bias toward either the optimization domain or the constraint domain if one of corresponding terms in the modified similarity measure dominates. In contrast to the good clustering result by the ICC algorithm in Fig. 10, the resulting partitions shown in Figs. 16 and 17 do not meet the requirement of compact region in the constraint domain, meaning that though with our best effort, conventional clustering algorithms are not applicable to solving the dual clustering problem. More precisely, in the CR algorithm, the points near the boundaries of two clusters are mingled with those of the neighboring cluster even for a large value of p. For a large value of p, the clustering cost on the optimization domain will increase. In the clustering results of the KNNC algorithm, the points at boundaries of two clusters are inclined to form a new cluster by themselves. This phenomenon can be seen in Fig. 17.

We also compare the clustering cost defined in Section 2.2 in the optimization domain obtained by these algorithms with that obtained by ICC. The results are shown in Fig. 18. The clustering cost of the ICC algorithm is very close to that of the others even after we consider the requirement in the constraint domain, showing the very advantage of the ICC algorithm. For the purpose of comparison, we choose the parameters of the alternative algorithms that produce similar results in constraint domain, i.e., $p = 3$ for the CR algorithm, and $k = 50$ for the KNNC algorithm.

5 Conclusion

We proposed and explored in this paper a new clustering problem on two domains, called dual clustering, where one domain refers to the optimization domain and the other refers to the constraint domain. Our goal is to optimize the objective function in the optimization domain while satisfying the constraints in the constraint domain. The proposed the ICC algorithm combines the information in both domains and iteratively performs a clustering algorithm on the optimization domain and also a classification algorithm on the constraint domain to reach the target clustering effectively. The time and space complexities of the ICC algorithm have been formally analyzed. Several experiments have also been conducted to provide the insights into the dual clustering problem and the proposed algorithm.

Acknowledgements

This work was supported in part by the National Science Council of Taiwan, R.O.C., under Contracts NSC93-2752-E-002-006-PAE.

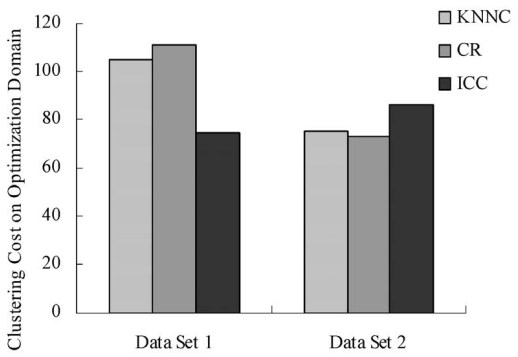


Fig. 18. Clustering costs in the optimization domain of the KNNC, CR, and ICC algorithms.

REFERENCES

[1] A. Ben-Hur, D. Horn, H.T. Siegelmann, and V. Vapnik, "Support Vector Clustering," *J. Machine Learning Research,* vol. 2, pp. 125-137, 2001.

[2] J.C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms.* New York: Plenum Press, 1981.

[3] P.S. Bradley, K.P. Bennett, and A. Demiriz, "Constrained K-Means Clustering," Technical Report MSR-TR-2000-65, Microsoft Research, May 2000.

[4] A.G. Buchner and M. Mulvenna, "Discovery Internet Marketing Intelligence through Online Analytical Web Usage Mining," *ACM SIGMOD Record,* vol. 27, no. 4, pp. 54-61, Dec. 1998.

[5] C.-C. Chang, C.-W. Hsu, and C.-J. Lin, "The Analysis of Decomposition Methods for Support Vector Machines," *IEEE Trans. Neural Networks,* pp. 1003-1008, 2000.

[6] M.-S. Chen, J. Han, and P.S. Yu, "Data Mining: An Overview from Database Perspective," *IEEE Trans. Knowledge and Data Eng.,* vol. 5, no. 1, pp. 866-883, Dec. 1996.

[7] B.-R. Dai, C.-R. Lin, and M.-S. Chen, "On the Techniques for Data Clustering with Numerical Constraints," *Proc. SIAM Int'l Conf. Data Mining (SDM '03),* 2003.

[8] R.C. Dubes, "How Many Clusters Are Best?—An Experiment," *Pattern Recognition,* vol. 20, no. 6, pp. 645-663, 1987.

[9] V. Estivill-Castro and I. Lee, "Autoclust+: Automatic Clustering of Point-Data Sets in the Presence of Obstacles," *Proc. Int'l Workshop on Temporal, Spatial and Spatio-Temporal Data Mining (TSDM '00),* pp. 133-146, 2000.

[10] U.M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurasamy, *Advances in Knowledge Discovery and Data Mining.* Cambridge, Mass: MIT Press, 1996.

[11] T. Gaertner, J.W. Lloyd, and P.A. Flach, "Kernels for Structured Data," *Proc. Int'l Conf. Inductive Logic Programming (ILP '02),* July 2002.

[12] J. Han and M. Kamber, *Data Mining: Concepts and Techniques.* Morgan Kaufmann, 2000.

[13] J. Hertz, A. Krogh, and R.G. Palmer, *Introduction to the Theory of Neural Computation.* Reading, Mass.: Addison-Wesley, 1991.

[14] C.-W. Hsu and C.-J. Lin, "A Comparison of Methods for Multiclass Support Vector Machines," *IEEE Trans. Neural Networks,* pp. 415-425, 2002.

[15] D. Hush and C. Scovel, "Polynomial-Time Decomposition Algorithms for Support Vector Machines," *Machine Learning,* pp. 51-71, 2003.

[16] B. King, "Step-Wise Clustering Procedures," *J. Am. Statistical Assoc.,* vol. 69, pp. 86-101, 1967.

[17] C.-R. Lin and M.-S. Chen, "On the Optimal Clustering of Sequential Data," *Proc. Second SIAM Int'l Conf. Data Mining,* Apr. 2002.

[18] C.-R. Lin and M.-S. Chen, "A Robust and Efficient Clustering Algorithm Based on Cohesion Self-Merging," *Proc. Eighth ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining,* Aug. 2002.

[19] S.Y. Lu and K.S. Fu, "A Sentence-to-Sentence Clustering Procedure for Pattern Analysis," *IEEE Trans. Systems, Man, and Cybernetics,* vol. 8, pp. 381-389, 1978.

[20] J. McQueen, "Some Methods for Classification and Analysis of Multivariate Observations," *Proc. Fifth Berkeley Symp. Math. Statistics and Probability,* 1967.

[21] K. Rose, E. Gurewitz, and G. Fox, "Deterministic Annealing, Constrained Clustering, and Optimization," *Proc. IEEE Int'l Joint Conf. Neural Networks,* pp. 2515-2520, 1991.

[22] K. Rose, E. Gurewitz, and G. Fox, "Constrained Clustering as an Optimization Method," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 15, no. 8, pp. 785-794, Aug. 1993.

[23] P.H.A. Sneath and R.R. Sokal, *Numerical Taxonomy.* London: Freeman, 1973.

[24] A.K.H. Tung, J. Han, L.V.S. Lakshmanan, and R.T. Ng, "Constraint-Based Clustering in Large Databases," *Proc. 2001 Int'l Conf. Database Theory,* Jan. 2001.

[25] A.K.H. Tung, J. Hou, and J. Han, "Spatial Clustering in the Presence of Obstacles," *Proc. Int'l Conf. Data Eng. (ICDE),* 2001.

[26] V. Vapnik, *The Nature of Statistical Learning Theory.* Springer Verlag, 1995.

[27] V. Vapnik, *Statistical Learning Theory.* Wiley, 1998.

[28] C. Yu, B.-C. Ooi, K.-L. Tan, and H.V. Jagadish, "Indexing the Distance: An Efficient Method to KNN Processing," *The VLDB J.,* pp. 421-430, 2001.

[29] O.R. Zaiane, A. Foss, C.-H. Lee, and W. Wang, "On Data Clustering Analysis: Scalability, Constraints, and Validation," *Proc. Pacific-Asia Conf. Knowledge Discovery and Data Mining (PAKDD '02),* pp. 28-39, 2002.

Cheng-Ru Lin received the BS and PhD degrees in electrical engineering from National Taiwan University, Taipei, in 1999 and 2003, respectively. He is currently a researcher at Arcadyan Advanced Technology Center, Taipei, Taiwan. His research interests include data mining, distributed system, network, and multimedia applications.

Ken-Hao Liu received the BS degree in electrical engineering from the National Taiwan University, Taipei, in 2001. He is currently a PhD candidate in the Electrical Engineering Department, National Taiwan University, Taipei. His research interests include data clustering and data mining.

Ming-Syan Chen received the BS degree in electrical engineering from National Taiwan University, Taipei, and the MS and PhD degrees in computer, information, and control engineering from The University of Michigan, Ann Arbor, in 1985 and 1988, respectively. Dr. Chen is currently a professor and the chairman of the Graduate Institute of Communication Engineering and a professor in EE Department and also CSIE Department, National Taiwan University, Taipei, Taiwan. He was a research staff member at IBM Thomas J. Watson Research Center, Yorktown Heights, New York, from 1988 to 1996. His research interests include database systems, data mining, mobile computing systems, and multimedia networking, and he has published more than 200 papers in his research areas. Dr. Chen served as an associate editor of the *IEEE Transactions on Knowledge and Data Engineering* (*TKDE*) from 1997 to 2001, is currently on the editorial boards of several journals, and was a Distinguished Visitor of the IEEE Computer Society Asia-Pacific from 1998 to 2000. He served as the program chair of PAKDD-02 (Pacific Area Knowledge Discovery and Data Mining), international vice chair of INFOCOM 2005, and program vice-chair of IEEE ICDCS 2005, ICPP 2003, and VLDB-2002. He was a keynote speaker on Web data mining at the International Computer Congress in Hong Kong, 1999, a tutorial speaker on Web data mining at DASFAA-1999 and on parallel databases at the 11th IEEE ICDE in 1995 and also a guest coeditor for the *IEEE Transactions on Knowledge and Data Engineering* special issue on data mining in December 1996. He holds, or has applied for, 18 US patents and seven ROC patents in the areas of data mining, Web applications, interactive video playout, video server design, and concurrency and coherency control protocols. He is a recipient of the NSC (National Science Council) Distinguished Research Award and K.-T. Li Research Penetration Award for his research work, and also the Outstanding Innovation Award from IBM Corporate for his contribution to a major database product. He also received numerous awards for his research, teaching, inventions, and patent applications. Dr. Chen is a fellow of the IEEE and a member of the ACM.