VISTA: Validating and Refining Clusters via Visualization

Keke Chen
College of Computing
Georgia Institute of Technology
801 Atlantic Dr., Atlanta, Georgia 30332, USA
Tel: (404) 633-6594
kekechen@cc.gatech.edu

Ling Liu
College of Computing
Georgia Institute of Technology
801 Atlantic Dr., Atlanta, GA 30332, USA
lingliu@cc.gatech.edu

Abstract

Clustering is an important technique for understanding of large multi-dimensional datasets. Most of clustering research to date has been focused on developing automatic clustering algorithms and cluster validation methods. The automatic algorithms are known to work well in dealing with clusters of regular shapes, e.g. compact spherical shapes, but may incur higher error rates when dealing with arbitrarily shaped clusters. Although some efforts have been devoted to addressing the problem of skewed datasets, the problem of handling clusters with irregular shapes is still in its infancy, especially in terms of dimensionality of the datasets and the precision of the clustering results considered. Not surprisingly, the statistical indices works ineffective in validating clusters of irregular shapes, too. In this paper, we address the problem of clustering and validating arbitrarily shaped clusters with a visual framework (VISTA). The main idea of the VISTA approach is to capitalize on the power of visualization and interactive feedbacks to encourage domain experts to participate in the clustering revision and clustering validation process. The VISTA system has two unique features. First, it implements a linear and reliable visualization model to interactively visualize multi-dimensional datasets in a 2D star-coordinate space. Second, it provides a rich set of user-friendly interactive rendering operations, allowing users to validate and refine the cluster structure based on their visual experience as well as their domain knowledge.

Keywords: Data Clustering, Cluster Analysis Framework, Information Visualization, Interactive Cluster Visualization, Cluster Validation and Refining

Submission Category: Regular Paper
Total number of pages: 23 including the title page
Contact Author: Keke Chen

VISTA: Validating and Refining Clusters via Visualization

Keke Chen Ling Liu
College of Computing, Georgia Institute of Technology
{kekechen, lingliu}@cc.gatech.edu

Abstract

Clustering is an important technique for understanding of large multi-dimensional datasets. Most of clustering research to date has been focused on developing automatic clustering algorithms and cluster validation methods. The automatic algorithms are known to work well in dealing with clusters of regular shapes, e.g. compact spherical shapes, but may incur higher error rates when dealing with arbitrarily shaped clusters. Although some efforts have been devoted to addressing the problem of skewed datasets, the problem of handling clusters with irregular shapes is still in its infancy, especially in terms of dimensionality of the datasets and the precision of the clustering results considered. Not surprisingly, the statistical indices works ineffective in validating clusters of irregular shapes, too. In this paper, we address the problem of clustering and validating arbitrarily shaped clusters with a visual framework (VISTA). The main idea of the VISTA approach is to capitalize on the power of visualization and interactive feedbacks to encourage domain experts to participate in the clustering revision and clustering validation process. The VISTA system has two unique features. First, it implements a linear and reliable visualization model to interactively visualize multi-dimensional datasets in a 2D star-coordinate space. Second, it provides a rich set of user-friendly interactive rendering operations, allowing users to validate and refine the cluster structure based on their visual experience as well as their domain knowledge.

1 Introduction

Over the past decades most of the clustering research has been focused on automatic clustering algorithms and statistical validity indices. The automatic methods are known to work well in dealing with clusters of regular shapes, e.g. compact spherical shapes, but incur high error when dealing with arbitrarily shaped clusters. Concretely, problems with the automatic clustering/validation algorithms can be briefly summarized as follows:

- It is hard to handle the arbitrarily shaped clusters, which are common in applications. Some new algorithms like CURE [15], WaveCluster [13] and DBSCAN [10], have addressed this problem and try to solve it in restricted situations, such as in low dimensional datasets, or the cluster shapes are elongated/enlarged regular ones. Yet it is still considered as an unsolved hard problem due to the complexity in multi-dimensional (>3D) space and the unpredictable skewed cluster distributions.

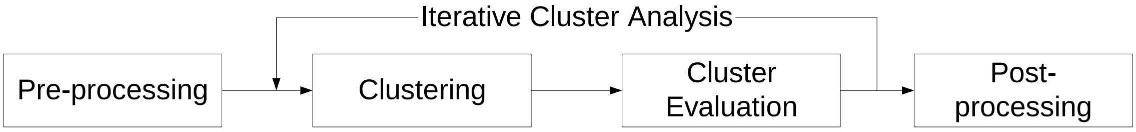


Figure 1: Iterative cluster analysis

- The arbitrarily shaped clusters also make the traditional statistical cluster validity indices ineffective [16], which leaves it difficult to determine the optimal cluster structure. For example, the compactness index, designed for spherical clusters, of an elongated shape is not high but the quality of cluster could be considered as good in practice.

- In applications, some irregularly shaped clusters may be formed by combining two regular clusters or by splitting one large cluster with the incorporation of domain knowledge. However, it is inconvenient to incorporate domain knowledge in or allow the user to steer the clustering process with automatic algorithms.

One feature of the automatic clustering algorithms is that it almost excludes human from the clustering process, which is good in terms of reducing user's workload, but which is not so good since the user cannot easily manipulate the process. What the user can do is usually setting the parameters before the clustering algorithm running, waiting for the algorithm producing the results, validating the results and repeating the entire process if the results are unsatisfactory. Once the clustering algorithm starts running, the user cannot monitor or steer the cluster process, which also makes it hard to incorporate domain knowledge into the clustering process and especially inconvenient for large-scale clustering since the iterative cycle is long. This exclusion makes the existing clustering framework inefficient and unintuitive for the user to deal with the application-specific clustering tasks.

Since clustering is an unsupervised process, the quality of clustering result needs to be evaluated by "cluster validity methods". The validity methods are heavily related to the geometry or density nature of clusters, such as the compactness or density of clusters, the distances between clusters, and so on [19, 16]. Particularly, cluster validity indices can be used to decide the optimal number of clusters. Some typical indices includes root-mean-square standard deviation (RMSSTD) for compactness of clusters, R-squared (RS) for dissimilarity between clusters, and S_Dbw for compound evaluation of compactness and dissimilarity [29, 16]. Smaller RMSSTD and S_Dbw values or larger RS values suggest better cluster quality. Although these indices were proved effective in determining the optimal number of *compact well-separated spherical* clusters, they do not work well for arbitrarily shaped clusters. A simple example in Figure 2−4 shows that the perfect clustering result of non-spherical clusters is often not consistent with the evaluation result. The 2D synthetic dataset shown in the figures consists of two clusters, both containing the same number of items and one of which is an elongated ellipse. With Euclidean distance the indices RMSSTD, RS and S_Dbw all suggest the k-means clustering (Figure 3) is better, which is, however, not recommended intuitively. A partition of three clusters (Figure 4) even gives better index values than partitions of two, which is not correct at all. These indices simply do not work in evaluating the irregular clusters. Although it is possible to find appropriate indices to deal with certain shapes, the statistical methods are not flexible enough to adapt any unanticipated shapes, especially when you do not know the sketch of clusters.

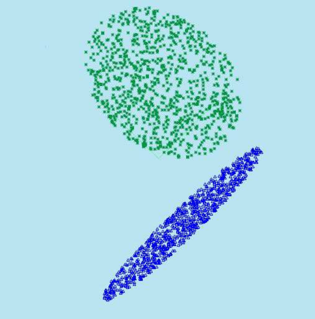


Figure 2: Correct clustering. $RMSSTD = 4.6516$, $RS = 0.5923$, $S\_Dbw = 1.5870$

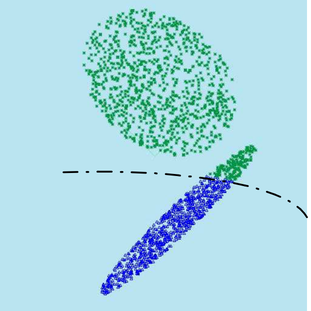


Figure 3: k-means result for $k = 2$. $RMSSTD = 4.5536$, $RS = 0.6093$, $S\_Dbw = 1.4605$

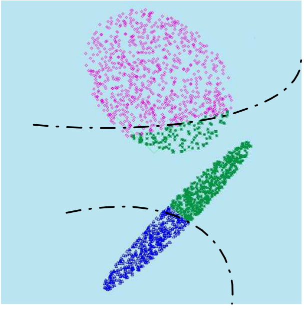


Figure 4: k-means result for $k = 3$. $RMSSTD = 3.4837$, $RS = 0.7715$, $S\_Dbw = 1.2022$

Since the geometry and density features of clusters derived from the distance (similarity) relationship, determines the validity of clustering results, no wonder that visualization is the most intuitive method for validating clusters, especially the clusters in irregular shape. Many clustering algorithms in literature employ the 2D-plot of the clustering results to validate their effectiveness on 2D experimental datasets. However, the cluster visualization is not commonly used in practice because of the difficulty in visualizing multi-dimensional (>3D) datasets.

Therefore, in general, clustering algorithms and validity indices have to answer the two question to deal with the arbitrarily shaped clusters: “How to recognize the special structure of each particular dataset?” and “Is it possible to refine a given imprecise cluster definition provided by algorithms through certain approaches efficiently?” To answer these questions, we propose a visual framework that allows the user to be involved into the clustering process via interactive visualization. The core of the visual framework is the visual cluster rendering system VISTA. VISTA can work with any algorithmic results - at the beginning, VISTA imports the algorithmic clustering result into the visual cluster rendering system, and then lets the user participate in the following “clustering-evaluation” iterations interactively. With the reliable mapping mechanism employed by VISTA system, the user can visually validate the defined clusters via interactive operations. The interactive operations also allow the user to refine the clusters or incorporate domain knowledge to define better cluster structure.

Combining with the algorithmic clustering results, VISTA works well in improving the understanding of the cluster structure and the performance of validating and refining the arbitrarily shaped clusters. We will demonstrate the power of VISTA with two concrete examples - one is about how to validate and refine the algorithmic results with visual cluster rendering and the other is how to incorporate domain knowledge into the clustering process via visualization.

We organize the paper as following. The visual framework and VISTA system are introduced in section 2; in section 3, two empirical examples are demonstrated in details to show the power of

VISTA in validating and refining clusters for real datasets. More experimental results are given in terms of the performance of cluster visual rendering. The related work is discussed in section 4. Finally, we discuss and conclude the VISTA approach.

2 VISTA Visual Framework

Most frequently, the clustering is not finished when the computer/algorithm finishes unless the user has evaluated, understood and accepted the patterns or results, therefore, the user has to be involved in the “clustering – analysis/evaluation” iteration (Figure 1). In many cases, a simplified process that employs automatic algorithms is like the following:

1. Run the algorithms with initial parameters, after pre-processing, such as feature extraction, and sampling or summarization for large datasets.

2. Evaluate the cluster quality and analyze the clustering results with statistical indices and domain knowledge.

3. If the result is not satisfactory, adjust the parameters and re-run the clustering algorithms, then do step 2 again until the satisfactory result is found.

4. If the result is satisfactory, do post-processing, which may label all of the items in the entire dataset or just output the cluster description.

Concrete discussion about this process can be found in [20]. Our discussion will focus on steps 2 and 3. In step 2, it is often ineffective to validate the arbitrarily shaped clusters with the traditional cluster validity indices. And it is also difficult for human to verify the result with the domain knowledge. Most clustering algorithms also need to set appropriate parameters. For example, CURE [15] requires the parameters of the number of representative points and shrink factor, and DBSCAN [10] needs a proper Eps and $MinPts$ to get satisfactory clusters. In step 3, it is usually very time-consuming to find appropriate parameters for a new run. The user needs to understand the meaning of the parameters, and often has to try several sets of parameters before finding the appropriate ones, which may increase the number of the analysis cycles. We observed that with automatic clustering algorithms the steps 2 and 3 can only be done in sequence – The user can only tune the parameters before the algorithm runs and then wait for the results coming and evaluate the results. We propose that if we can interweave these two steps, e.g. the user can participate in the clustering process, monitoring and steering the process, the entire process would be more efficient. Instead of achieving this interweaving by improving the existing automatic algorithms – which could be very hard – we develop an interactive visual cluster rendering system to get human involved in. The entire visual framework is like Figure 5.

Former studies [25, 24, 30] in the area of visual data exploration support the notion that visual exploration can help in cognition. Visual representations, especially interactive visualization, can be very powerful in revealing trends, highlighting outliers, showing clusters, and exposing gaps.

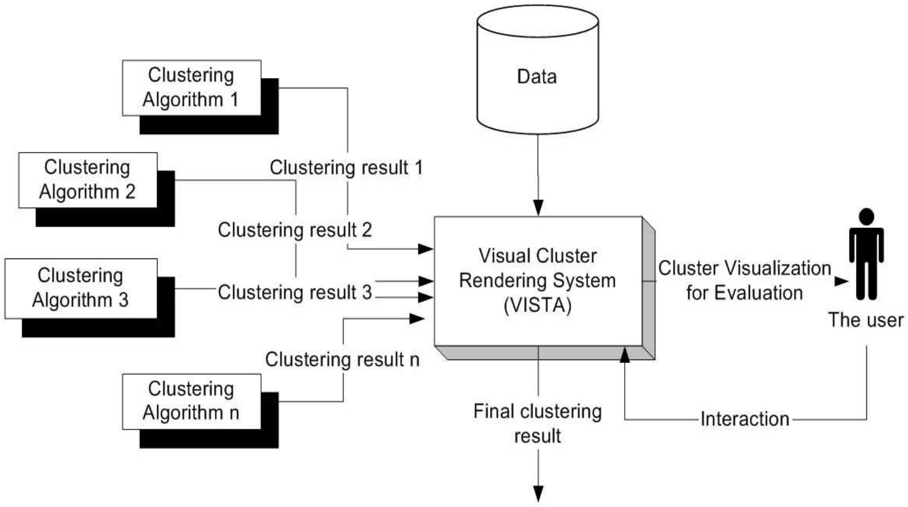


Figure 5: Visual framework for validating and refining clusters

Previous research shows that, with the right coding, human pre-attentive perceptual skills can enable users to recognize patterns, spot outliers, identify gaps and find clusters in a few hundred milliseconds [30]. For example, in a scatter-plot based visualization, the human visual ability is adept at finding the clusters – the point-dense area very quickly, and the shape of the cluster is identified at the same time too. All of the advantages make the interactive cluster visualization systems very attractive. However, there are some challenges for cluster visualization techniques, among which the most challenging one is cluster preserving– the clusters appearing in the 2D/3D visualization should be the real clusters in k-D ($k > 3$) space. Since a k-D to 2D/3D mapping inevitably introduces visual bias, such as broken clusters, overlapping clusters, or fake clusters formed by outliers, static visualization is not sufficient to find all cluster information, and therefore, additional interactive rendering techniques may be needed to improve the visual quality.

In VISTA cluster rendering system, we use a linear (or affine) mapping [12] – α-mapping to avoid the breaking of clusters after mapping, where “gaps” between point clouds in visualization are the *real gaps* in high dimensional space, but the overlapping and fake clusters may still exist. The compensation technique is interactive dynamic visualization. The interactive operations are used to change the projection plane, which allows the user to observe the datasets from different perspectives continuously. Continuously changed visualization usually provides important clues for the user to discriminate the overlapping and the fake clusters effectively.

While the visual cluster rendering system is combined with the algorithmic result, the two can improve each other. The colored algorithmic result in visualization provides visual clustering clues – the points in same color, i.e. in the same cluster, should be grouped into the same area by large, which can guide the user to find a satisfactory visualization very quickly. On the other side, the satisfactory cluster visualization after rendering can validate the algorithmic results by visually checking the match of the visual cluster distribution and the algorithmic distribution. In summary, the best way for visual cluster rendering is to combine the algorithmic results with the interactive visualization system.

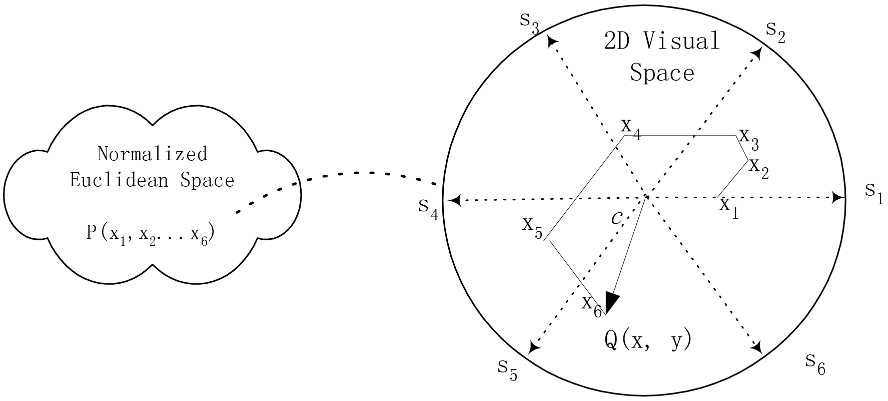


Figure 6: Illustration of α-mapping with $k = 6$

The basic methodology employed in visual cluster validating and refining follows the steps:

Step1. Load the dataset, (and algorithmic clustering result if available), into the VISTA visual cluster rendering system, after pre-processing.

Step2. Use the interactive operations to find a satisfactory visualization,

Step3. Import domain knowledge if available, mark the visual boundaries between clusters and refine the algorithmic result if applicable.

Step4. Output the refined result.

To illustrate how the VISTA system works, we will briefly introduce the α-mapping and the associated interactive operations. The initial version of VISTA is used to render Euclidean datasets, where the similarity is defined by Euclidean distance, since the Euclidean datasets are the most common datasets in applications and human vision is only able to discern Euclidean distances. By default, we will not mention this again in the following discussion.

2.1 α-mapping

We invent a linear mapping α-mapping that partially preserves k-dimensional (k-D) information in 2D space and is used to build a k-parameter-adjustable interactive visualization system. α-mapping maps k-D data to 2D star coordinates [22] and normalizes the visualization into the designated display area. It utilizes star coordinates to establish the visualization. A k-axis 2D star coordinates is defined by an origin $\vec{o}(x_0, y_0)$ on screen and the k coordinates S_1, S_2, $\ldots$, S_k , which represent the k dimensions in 2D spaces. The k coordinates are equidistantly distributed on the circumference of the circle $\mathbb{C}$, as in Figure 6, where the unit vectors are $\vec{s_i} = (\cos(2\pi i/k), \sin(2\pi i/k)), i = 1..k$. The radius c of the circle $\mathbb{C}$ is the scaling factor to the entire visualization. Changing c will change the effective size and the detail level of visualization.

We formally describe α-mapping as follows. Let a 2D point $Q(x, y)$ represent the image of a k-dimensional max-min normalized [26] (with normalization bounds [-1, 1]) data point $P(x_1, \ldots, x_k)$

on the 2D star coordinates. Q(x, y) is determined by the average of the vector sum of the k vectors $\alpha_i \cdot x_i \cdot \vec{s}_i, (i = 1..k)$, where α_i are the k adjustable parameters. This sum can be scaled by the radius c.

α-mapping :

$$A(x_1, \ldots, x_k, \alpha_1, \ldots, \alpha_k) = (c/k) \sum_{i=1}^{k} \alpha_i x_i \vec{s}_i - \vec{o} \tag{1}$$

i.e.

$$\{Q_x, Q_y\} = \{(c/k) \sum_{i=1}^{k} \alpha_i x_i \cos(2\pi/i) - x_0, (c/k) \sum_{i=1}^{k} \alpha_i x_i \sin(2\pi/i) - y_0\} \tag{2}$$

α-mapping has two important properties:

1. The mapping is linear. Without loss of generality, we set $\vec{o}$ to (0, 0), since it only translates the visualization. It is easy to see α-mapping is a linear mapping, given a set of constant α_i. It is known that the linear mapping does not break clusters [12]. Therefore, each gap seen in visualization confirms a $real$ gap in the original k-D space, and α-adjustment does not create any false “gaps”. However, overlapping of clusters may still happen, and sometimes, overlapping outliers may form fake clusters in visualization. What we need to do is to percept the overlapping and separate them interactively.

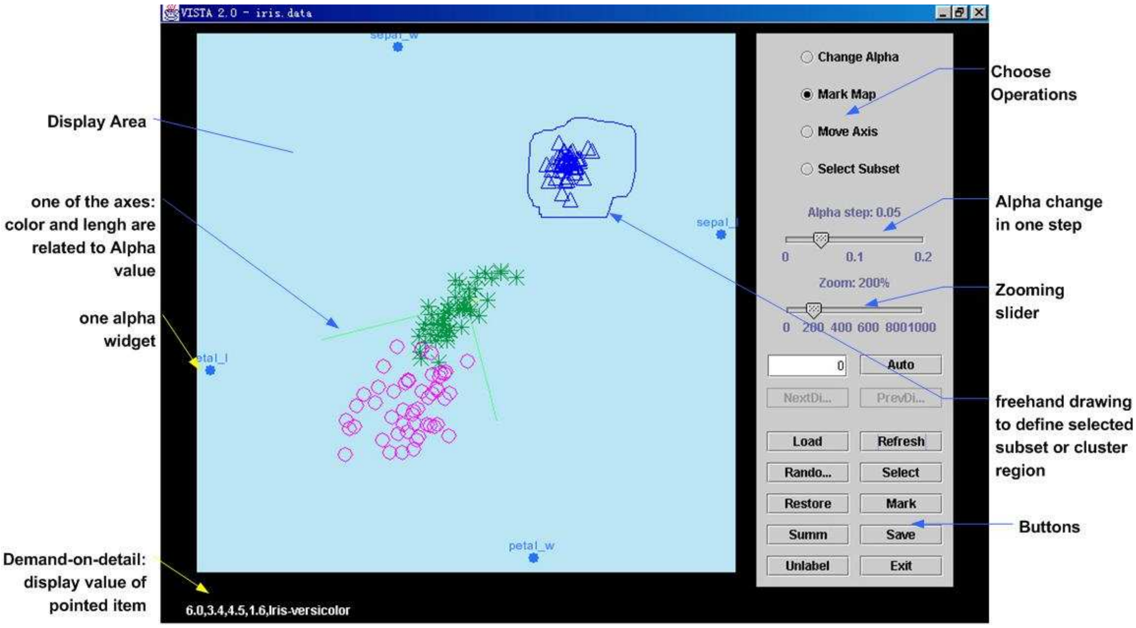
2. The mapping is adjustable. α_i can be regarded as the weight of the i-th dimension, which means how significant the i-th dimension is in the visualization. By changing α_i continuously, we can see the effect of the i-th dimension on the cluster distribution. In addition, when one value or several values are changed continuously at the same time, the k-D dataset is mapped to a series of smoothly changed projections, which provide important cluster clues.

2.2 The visual rendering operations

The VISTA system looks like Figure 7. The task of the VISTA cluster rendering system is to provide the interactive visualization techniques to help the users find and separate the overlapping clusters through continuously changed visualization. We have designed and implemented a set of interactive rendering operations in VISTA.

2.2.1 α-parameter adjustment

The most important operation in VISTA system is α-parameter adjustment (or simply, α-adjustment). This operation changes the parameters defined in formula (1). Each change refreshes the visualization in real time (about several hundred milliseconds, depending on different hardware conditions and the size of dataset). α-parameter adjustment enables the user to find the dominating dimensions, to observe the dataset from different perspectives and to distinguish the real clusters from overlapping with a series of continuously changed visualizations.



Alpha step: 0.05

Zoom: 200%

6.0,3.4,4.5,1.6,Iris-versicolor

Figure 7: VISTA visual rendering system

Continuous α-parameter adjustment of one dimension looks at the effect of this dimension on the entire visualization. Let $X(x_1, \ldots, x_k)$ and $Y(y_1, \ldots, y_k)$, $x_i, y_i \in [-1, 1]$ represent any two normalized points in k-D space. Let $\|\vec{v}\|$ represent the length of vector $\vec{v}$. We define the visual distance between X and Y is:

$$\begin{aligned} dist(X,Y) &= \|A(x_1,\ldots,x_k,\alpha_1,\ldots,\alpha_i,\ldots,\alpha_k) - A(y_1,\ldots,y_k,\alpha_1,\ldots,\alpha_i,\ldots,\alpha_k)\| \\ &= \|(c/k)\sum_{i=1}^{k}\alpha_i(x_i - y_i)\vec{s}_i\| \qquad (3) \end{aligned}$$

which means if x_i and y_i are close, changing α_i does not change the distance between X and Y a lot – the dynamic visual result is that X and Y are moving together when α_i changes. Thus, we can conclude that close points in k-D space, which should belong to one cluster, tend to move together in any α-adjustment, while those points that are far away from each other in k-D space may move together in some α-adjustment but definitely not in all α-adjustment. This property makes α-adjustment very effective in revealing the main visual bias introduced by α-mapping, which is two distinguished clusters may be mapped to the same visual area by some specific α settings. A set of heuristic dimension-by-dimension rendering rules [4] are based on this dynamic feature. Experiments showed these rules are very effective in validating and refining the cluster boundary.

In addition, the point movement also reveals the distribution of each dimension by large. If we adjust the α value of the dimension i, the point movement can be represented by:

$$\begin{aligned}\Delta(i) &= A(x_1,\ldots,x_k,\alpha_1,\ldots,\alpha_i,\ldots,\alpha_k) - A(x_1,\ldots,x_k,\alpha_1,\ldots,\alpha'_i,\ldots,\alpha_k) \\ &= (c/k)(\alpha_i - \alpha'_i)x_i\vec{s}_i \end{aligned} \tag{4}$$

which means that the points having larger x_i will be moving faster along the i-th coordinate, similar x_i moving in a similar way. An illustration of finding "gaps" by one-dimensional α-adjustment is shown in Figure 8. By looking at the moving point cloud along the i-th axis when we are changing the α_i value, we can easily estimate the value distribution in i-th dimension. In Figure 8, we demonstrate that point moving and point distribution can be interpreted with each other.

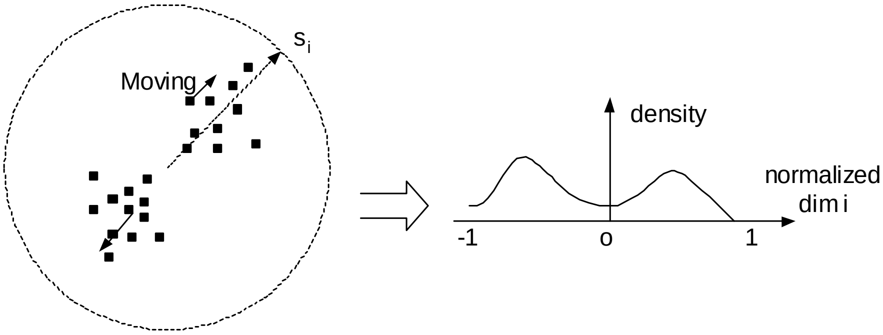


Figure 8: α adjustment, dimensional data distribution and point movement

Compared to the basic rendering techniques "axis scaling" and "axis rotation" in the original star coordinates system [22], VISTA α-adjustment, powered by the VISTA visualization model, can solely support the user to find satisfactory visualization. We argue that, in fact, α-adjustment together with the zooming operation are more powerful than the combination of "axis scaling" and "axis rotation". The zooming factor c does not change the structure of the content in visualization, hence, only is the α-adjustment enough to provide the informative projection planes.

Concretely, the mapping described in the original star coordinates [22] can be actually rephrased and included in VISTA visualization model, where the original k-dimensional point $P(x_1,\ldots,x_k)$ is normalized to [0, 1] and $\alpha_i, (i = 1, 2, \ldots, k)$ is limited to [0, 1], instead. The positive-only α-adjustment shows the dynamic visual effect just as the "axis scaling" described by the author. Nevertheless, α-adjustment in [-1, 1] defines more than "axis scaling" – the actual effect can be viewed as "rotation around the visual center".

In general, α-adjustment in different range can be intuitively understood as the projection of "visual rotation [1]" of point cloud around the perpendicular axis of coordinate i. Different α-value ranges define different rotation angles (Figure 9$-$12). For example, "axis scaling" can be regarded as the projection of rotation from 0^o to 90^o around the perpendicular axis as shown in Figure 9.

While the range of α-values determines the ability of interactive operations, the scope of the nor-

[1] observed as rotation in continuous α-adjustment

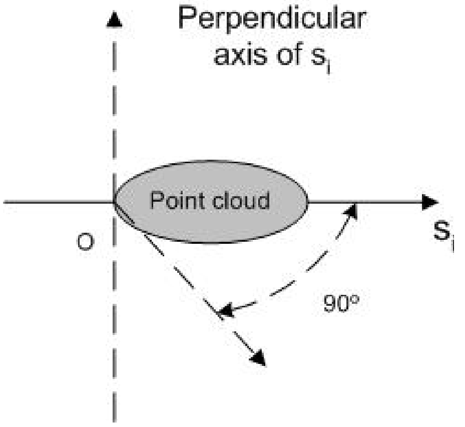


Figure 9: $\alpha \in [0, 1]$, $normalized \in [0, 1]$

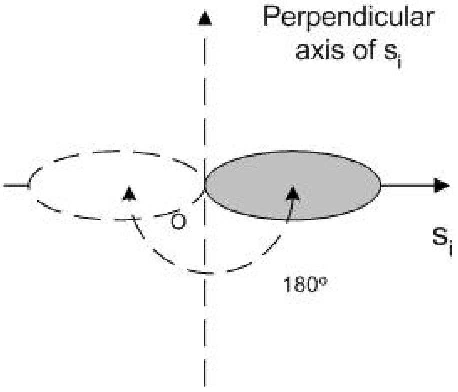


Figure 10: $\alpha \in [-1, 1]$, $normalized \in [0, 1]$

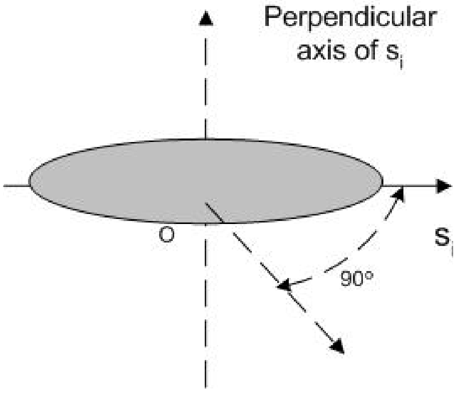


Figure 11: $\alpha \in [0, 1]$, $normalized \in [-1, 1]$

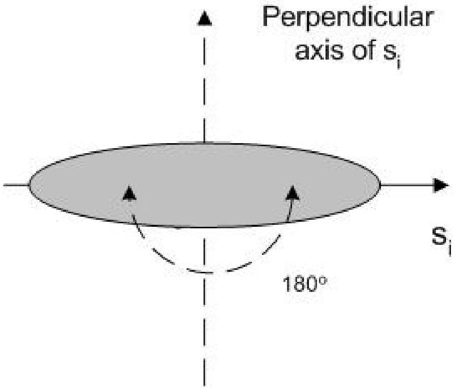


Figure 12: $\alpha \in [-1, 1]$, $normalized \in [-1, 1]$

malized original values influences the coverage of effective visualization. To make the demonstration clearer, without loss of generality, we consider the situation where dimension i dominates the visualization with a set of α-values, e.g. $(c/k)\sum_{j=1}^{k}\alpha_j x_j \vec{s}_j \approx (c/k)\alpha_i x_i \vec{s}_i$. When the dimension i are normalized to [0, 1] as in the original star coordinates, the point cloud tends to distribute over the positive direction of $\vec{s}_i$, which squeezes the effective visualization onto one side of the display area and shows less details inside the point cloud (Figure 9). In contrast, normalizing to [-1, 1] elongates the point cloud to cover the entire effective visualization space and therefore, allows the user to observe more details (Figure 11).

To sum up, with the different scopes of normalized values and α values, we can observe different effects of α-parameter adjustment. We list four typical intuitive rendering effects as in Figure $9-12$. α-mapping in VISTA model is the case in Figure 12, which efficiently utilizes the entire display area and enables the "180^o rotation" of data points along the perpendicular axis. The mapping in star coordinates is the case in Figure 9, which tends to squeeze the point clouds and thus limit the observation angles. Different parameter scope setting of VISTA visualization model determines the ability of corresponding visualization system. We choose the set of ranges that maximize the power of the underlying visualization model in VISTA system.

2.2.2 Other operations

A bunch of interactive operations are designed and implemented in VISTA cluster rendering system for revealing cluster overlapping, validating algorithmic clustering results, and refining the definition of clusters visually. Since α-parameter adjustment is the most frequently used one, random rendering and automatic rendering, are used to increase the efficiency of α-parameter adjustment [4]. Another set of operations support point-set-oriented operations and are used to validate and refine visual cluster definition after we get the initial cluster visualization with α-parameter adjustment or load the algorithmic result. These operations include subset selection, cluster marking, cluster splitting, cluster merging, and hierarchical structure defining. Domain knowledge in form of labeled items can also be incorporated into VISTA visualization.

To facilitate the description, we define a dataset as $\mathbb{D} = \{X_i \mid X_i =< x_{i1}, \ldots, x_{ik} >, 1 \leqslant i \leqslant N\}$, where k is the number of dimensions and N is the number of instances in the dataset. We also define a subset $\mathbb{S}$ is a subset of $\mathbb{D}$, i.e. $\mathbb{S} \subseteq \mathbb{D}$, and a cluster is also a subset. Therefore, operations on clusters are the operations on subsets.

Subset selection This operation defines a subset of points by freehand drawing an enclosed region on screen or selecting a range of one dimension. The selected subset can be used for further processing, such as cluster marking, merging and splitting. Initially, we have one subset, which is the entire dataset. The clusters are defined as subsets from then on. We name the i-th subset as $\mathbb{S}_i$. Suppose before selection, we have had m subsets ordered as $(\mathbb{S}_1, \mathbb{S}_2, \ldots, \mathbb{S}_m)$. The $(m + 1)$-th subset is selected from one or more subsets. We define subset selection as following, where '-' is set difference operation.

$$SS(m): \quad (\mathbb{S}_1, \mathbb{S}_2, \ldots, \mathbb{S}_m) \rightarrow (\mathbb{S}_1 - \mathbb{S}_{m+1}, \mathbb{S}_2 - \mathbb{S}_{m+1}, \ldots, \mathbb{S}_m - \mathbb{S}_{m+1}, \mathbb{S}_{m+1})$$

Merging and splitting clusters These two operations enable the user to refine the visualized algorithmic clustering result. If the user finds a part of a cluster should be semantically separated from the cluster, she/he can use selection operation to select this part and then excludes it from the cluster. If two nearby clusters should be regarded as one cluster from the domain knowledge, the user just selects them and merges them into one cluster. A cluster boundary can be refined by merging and splitting operations, too. Splitting subset i to subset i_1 and i_2, and merging subset i to j are defined as following, where '$\cup$' is set union operation.

$Split(i, i_1, i_2, m):$
$(\mathbb{S}_1, \ldots, \mathbb{S}_i, \ldots, \mathbb{S}_m) \rightarrow (\mathbb{S}_1, \ldots, \mathbb{S}_i - \mathbb{S}_{i_2}, \ldots, \mathbb{S}_m, \mathbb{S}_{i_2})$, where $\mathbb{S}_{i_1} = \mathbb{S}_i - \mathbb{S}_{i_2}$
$Merge(i, j, m):$
$(\mathbb{S}_1, \ldots, \mathbb{S}_i, \ldots, \mathbb{S}_j, \ldots, \mathbb{S}_m) \rightarrow (\mathbb{S}_1, \ldots, \mathbb{S}_{i-1}, \mathbb{S}_{i+1}, \ldots, \mathbb{S}_{j-1}, \mathbb{S}_{j+1}, \ldots, \mathbb{S}_m, \mathbb{S}_i \cup \mathbb{S}_j)$

Defining hierarchical cluster structure With the operations of defining the cluster hierarchy, the user can group the clusters together to form a higher level cluster structure, or the user can zoom into one large cluster and find the fine cluster structure in the cluster iteratively. These operations together define a layered cluster structure. With the operations of zooming in or zooming out, the user can find the cluster details at different level. We define the operations as follows. Hierarchical structure is a tree structure. A node contains all subsets presented in its children. Suppose we are at a leaf node j, which has m clusters, and we want to define a sub-layer which contains l clusters among the m clusters, $l \leqslant m$. We have the following definition.

$$H_j(m, \mathbb{S}_{i1}, \mathbb{S}_{i2}, \ldots, \mathbb{S}_{il}): \quad (\mathbb{S}_1, \ldots, \mathbb{S}_m) \rightarrow (\mathbb{S}_{j1}, \ldots, \mathbb{S}_{j,m-l}, (\mathbb{S}_{i1}, \mathbb{S}_{i2}, \ldots, \mathbb{S}_{il}))$$

$i1 \ldots il$ are the index of the selected l subsets among the m subsets, and $j1 \ldots j, m-l$ are the index of the rest subsets. We use embedded "$(\ldots(\ldots))$" to represent the layer and its sub-layer(s). Cluster exploration on hierarchical structure can provide an efficient way in dealing with very large datasets[2].

Importing domain knowledge A set of domain knowledge is transferred to a set of k-D items with different group identities. These items are imported into the visual rendering system and rendered in different colors with different groups. These colored items act as the guidance to re-define the cluster partition with domain knowledge. If domain knowledge is represented by l groups of items$\{g_1, \ldots, g_l\}$, these items form l new subsets after they are loaded, which then used to direct further cluster splitting or merging. For example, if g_1 covers $\mathbb{S}_1$ and $\mathbb{S}_2$, then we can consider merging $\mathbb{S}_1$ and $\mathbb{S}_2$. A more specific example will be given in section 3.2.

$$D(m, g_1, g_2, \ldots, g_l): \quad (\mathbb{S}_1, \ldots, \mathbb{S}_m) + (g_1, g_2, \ldots, g_l) \rightarrow (\mathbb{S}_1, \ldots, \mathbb{S}_m, g_1, g_2, \ldots, g_l)$$

3 Empirical Study

In this section, we will introduce two examples of visual rendering. The first one demonstrates the ability of VISTA visual validating and interactive refining. The second one shows how to incorporate domain knowledge into VISTA visual cluster rendering. The datasets used in the examples can be found at UCI machine learning database (http://www.ics.uci.edu/m̃learn/MachineLearning.html).

3.1 Analyzing the "iris" dataset

In this example, we will use the most popular clustering algorithm – k-means [19] to produce the clustering result on the dataset "iris", and then import the result into VISTA system. With VISTA system, we will validate the k-means result visually and then try to refine the clusters and improve the quality of the k-means clusters. The quality of clusters will also be evaluated by statistical

[2] Combining with some tree structures for data summarization, such as CF-tree in BIRCH [33], we can implement efficient visualization for very large datasets

indices RMSSTD, RS, and S_Dbw [29, 16] at the same time to see if the statistical indices are consistent with the visual improvement.

“Iris” dataset is a famous dataset widely used in pattern recognition and clustering. It is a 4-D dataset containing 150 instances, and there are three clusters, each has 50 instances. One cluster is linearly separable from the other two; the latter two are not exactly linearly separable from each other.

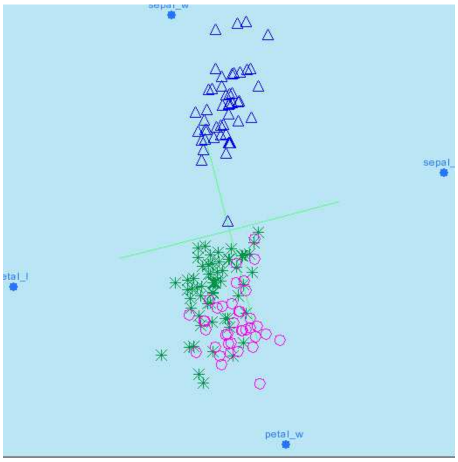


Figure 13: the initial visualization with k-means labels

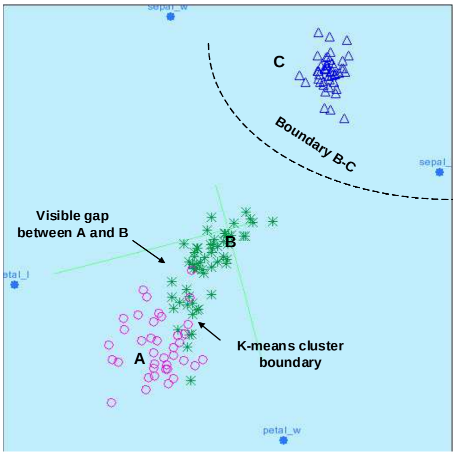


Figure 14: after α-adjustment.

We demonstrate the operations of validating and refining k-means clustering result of “iris” data as following steps. Firstly, we load the dataset and import the k-means labels for “iris” dataset into the visualization. Different clusters are visualized in different colors and shapes (Figure 13). In the initial visualization, we have already found that one cluster has been separated from the other two. After interactive cluster rendering, mainly the α-parameter adjustment, the visual boundaries become clearer (Figure 14). The boundary B-C clearly separates cluster C from the other two clusters. The gap between cluster A and B can be visually perceived. α-mapping model confirms that this gap does exist in the 4-D space since α-mapping does not break clusters. We make this gap as the visual boundary A-B. This visually perceived boundary A-B is not consistent with the k-means boundary, but we have more confidence with it since it has been intuitively confirmed. There is a principle in visual cluster rendering – *we prefer visual perception rather than statistical information because we believe the visual ability is better than statistical methods in dealing with arbitrarily shapes*.

Considering this visual boundary, we want to edit the k-means result with visual cluster editing operations. First, we split the points that belong to cluster A but visualized in cluster B, from cluster A. These points are then merged into cluster B. Do the same operation on the B points in cluster A as shown in Figure 15. After the editing operations, the points in the clusters are shown more homogeneously (Figure 16). The visual partition exactly reflects the real cluster distribution (compare Figure 16 and 17).

We check the validating results of the widely used cluster validity indices RMSSTD, RS and

S_Dbw, to see if the statistical validation is consistent with the visual improvement. RMSSTD is used to estimate the homogeneity of the clusters. Smaller RMSSTD indicates that the clusters are more compact. RS is used to estimate the dissimilarity between clusters. Larger RS indicates higher dissimilarity between groups. S_Dbw is a compound evaluation of compactness and dissimilarity, e.g. the overall quality of clusters. The smaller S_Dbw implies the better quality. The statistical evaluation shows RMSSTD is increased from 0.4421 to 0.4614, RS is decreased from 0.8254 to 0.8098, and S_Dbw rises from 1.4158 to 1.5115 after visual rendering. This means the compactness of clusters and the dissimilarity between clusters are decreased at the same time – the quality of clustering after visual improvement is worse than the k-means result statistically, which is not appropriate in practice! The irregular shapes of A and B, together with the closeness to each other, makes the statistical methods ineffective in this scenario.

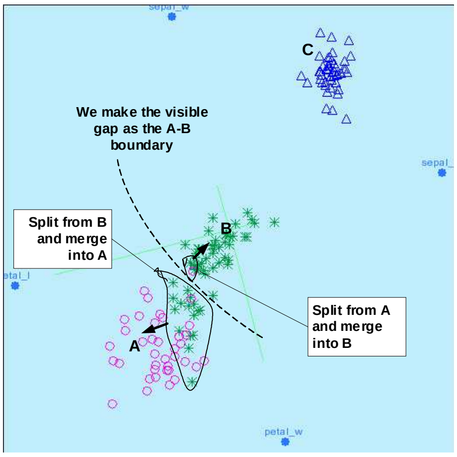


Figure 15: editing the clusters.

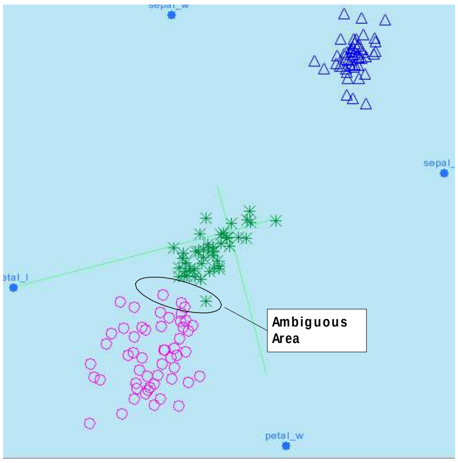


Figure 16: after editing.

As the literature of the “iris” dataset mentioned, the clusters A and B are not linearly separable. To further refine the cluster definition, we can also informally define a small “ambiguous area” around the gap between A and B, the points in which have equal probability of belonging to A or B. In extended experiments with trained users, all users can find the visualization like Figure 16, which means visual validity could be very practical in exploring certain datasets. We will support this assertion with more experimental result in section 3.3.

In conclusion, we believe that the VISTA system is better than the statistical indices, in terms of validating arbitrarily shaped clusters. In this example, we have seen that sometimes the vague boundary between the two clusters is easily checked by human visual ability but it is not so easy for the automatic algorithms. In addition, this example also shows the power of online refining ability of the VISTA system – after validation, the user can improve the quality of clusters immediately by editing the clusters, which effectively combines the two steps “re-clustering” and “evaluation” together.

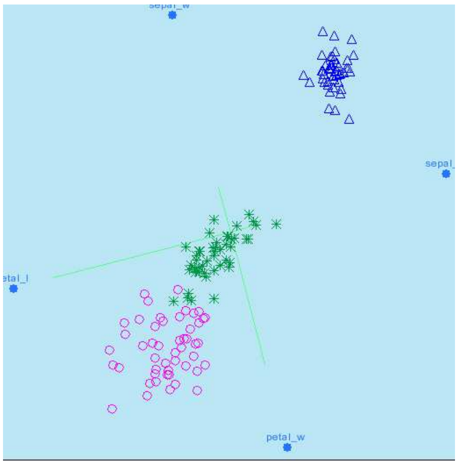


Figure 17: documented cluster definition

3.2 Incorporating domain knowledge

In this example, we will demonstrate that the VISTA system can conveniently incorporate the domain knowledge into the clustering process and provide intuitive clues for the user to define the application-specific clusters. We first define the “domain knowledge” utilized in VISTA system, and then show how to use the domain knowledge to distinguish the application-specific cluster distribution in rendering the “shuttle” dataset.

Domain knowledge plays a critical role in the clustering process [20]. It is the semantic explanation to the data groups, which may be different from the structural clustering criteria, which is based on distance definition between points. Domain knowledge usually indicates a high-level cluster distribution, which may be different from the structural clustering results, for example, the original clusters may be grouped to form larger clusters or split to form finer cluster structure. It is always desired to *see* if the domain knowledge is consistent with the cluster structure and if the domain knowledge can be incorporated into clustering process to refine the cluster structure. However, splitting or grouping basic clusters heavily depends on the understanding of the particular cluster structure. It is really unintuitive and inconvenient for automatic-algorithm-based clustering process. VISTA system provides the possibility of visually checking the matching between the cluster structure and domain knowledge, and conveniently manipulating the basic clusters based.

Domain knowledge can be represented in various forms in Artificial Intelligence [20, 28]. In clustering process, the domain knowledge is simply represented as a few labeled instances. We define the domain knowledge used in VISTA system as follows. Suppose the dataset $\mathbb{D} = \{X_i \mid X_i =< x_{i1}, \ldots, x_{ik} >, 1 \leqslant i \leqslant N\}$ and the user have some domain knowledge about the application. The domain knowledge could come from the specific properties of the application, the experimental results, or any hypotheses the application holds. However, because of various reasons, typically, time/money costly to get the experimental results, we can only get a small number of labeled interesting instances $Y_1, Y_2, , Y_n$ ($n \ll N$), which reflect the properties, or the experimental results. According to the labels, this set of instances should be partitioned into m

groups. Using (instance, group ID) to represent a labeled instance, we have the n instances labeled as

$(Y_{1,1}, 1)(Y_{1,2}, 1) \ldots (Y_{1,n_1}, 1)$

$\ldots$

$(Y_{m,1}, m)(Y_{m,2}, m) \ldots (Y_{m,n_m}, m)$

where $n_1 \ldots n_m$ are the number of instances in each group. They are regarded as the domain-related "landmarks" in VISTA system. The number of the instances is often so small that they cannot work effectively as a training dataset for classification algorithms [27] classifying the entire datasets.

There are two rendering methods for incorporating the domain knowledge. The first method is as follows. When visualizing a dataset, the landmark points are loaded and visualized in different colors according to their group ID. This guiding information can direct the user to define the high-level cluster structure, or to refine the algorithmic clustering results a little bit.

The alternative method is to visualize the dataset first and then sample some points from the "critical areas" on the visualization, such as the connective area of two point clouds. The small number of sample points are easy to be classified with the domain knowledge and then re-imported into the visualization as the "landmarks" to direct rendering.

We use the "shuttle" dataset and the second method to demonstrate how the VISTA system incorporates the domain knowledge into the clustering process. "Shuttle" dataset is a 9-D dataset. There are three large clusters and some tiny clusters in the dataset. Approximately 80% of the data belongs to one cluster. The other two large clusters have about 15% and 5% points, respectively. We use the testing dataset, which has 14500 items, for visualization.

After loading the dataset and adjusting the parameters, we get the initial visualization, which shows the cluster distribution is highly irregular. Roughly, there are six homogenous segments (Figure 18). We have no idea whether they are six individual clusters, or they should be grouped or split to form higher-level clusters.

We then pick several points from the visualization, which should ideally cover the connective areas. Suppose the "expert", according to experiments or any domain knowledge (use the labels from the original dataset to mimic), tells us the sample points should be grouped into three clusters. Using them as the "landmarks", we find a possible cluster structure as Figure 19. To observe the landmarks clearly, we visualized other data points in white color. The points in triangle, round, and star shapes are the landmarks of different groups. The result clearly shows that this dataset probably should be partitioned in the suggested way according to the domain knowledge. The real cluster distribution of the "shuttle dataset" is visualized in Figure 20 for comparison.

In conclusion, since the automatic algorithms exclude the human from the clustering process, the domain knowledge cannot be easily incorporated into the clustering process. With the help of VISTA system, the user is able to incorporate the domain knowledge into the clustering process and define the application-specific cluster distribution online. This combination of human-based analysis/evaluation and clustering process breaks the gap between human and the machines, and thus improves the efficiency of the entire cluster analysis process.

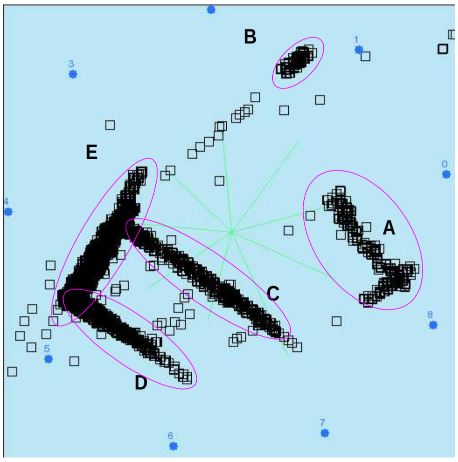


Figure 18: after initial rendering

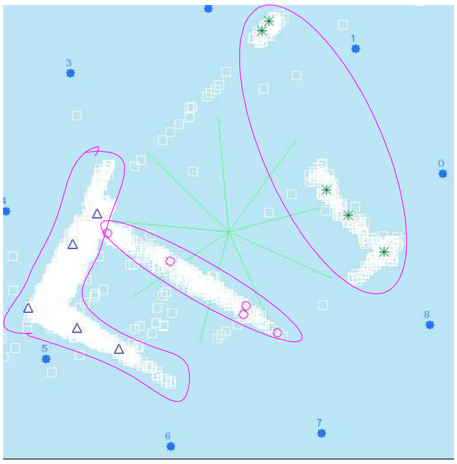


Figure 19: loading the domain knowledge

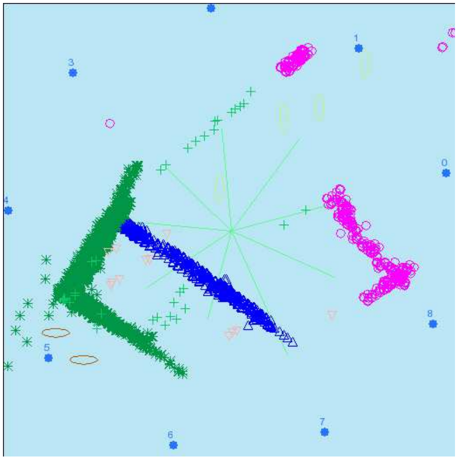


Figure 20: documented cluster distribution

Some research on classification were also seeking help from the unlabeled data to reinforce the learning ability with the small number of training instances [2]. However, we believe if the domain knowledge (the labeled data) can be consistently incorporated into the basic cluster structure by splitting or grouping clusters, the result is much easier to interpret. Therefore, VISTA system can be an effective alternative in dealing with the combination of labeled and unlabeled data.

3.3 More Experimental Results

The VISTA visual clustering system was implemented in Java. In this section we will introduce more experimental results to show the power of visual cluster rendering system in finding clusters individually or in combining any external information to provide better clustering results. These experiments were conducted on a number of well-known datasets that can also be found in UCI machine learning database. These datasets, although small or median in size, have irregular cluster distribution, which is an important factor for testing the effectiveness of the VISTA system.

Five well-trained users use the VISTA cluster rendering system to find satisfactory visualization for each of the ten datasets. After we use the interactive visual operations to find the satisfactory visualization, either solely by visual rendering or incorporated by algorithmic result, we mark the areas which are regarded as clusters and the items in each area are respectively labeled with the cluster ID. With the original labels in the datasets, we define the items that are wrongly clustered as the errors, the number of which divided by the size of the dataset is the error rate of visual cluster rendering on this dataset.

We firstly use unguided visual rendering (UGV) to find the visual partition. Unguided visual rendering does not rely on any external information and only depends on the visually observed dense-point areas and the gaps between the areas. Since there is visual bias on the visualization, the visual rendering sometimes may trap in local minima, where the user thinks the visualization is good enough for defining cluster boundaries. We could avoid trapping in local minima by incorporating some external information, either from other clustering algorithms or domain knowledge. In our experiments, 10% of labeled items randomly selected from the original datasets are used as the "landmarks". We also compare the results of k-means and CURE algorithms on the experimental datasets. CURE clustering is recognized as one that can deal with irregular cluster shapes in some level, and k-means is the most popular algorithm commonly used in research or applications. The experiment shows that individually CURE or K-means cannot deal the arbitrarily shaped clusters very well and UGV may trap into some local minima, but by combining with the external information we can improve the UGV result more or less. The result also shows the visualization result, either UGV or combined rendering, is often better than algorithmic result for arbitrarily shaped clusters. In addition, the interaction time for most datasets is less than 5 minutes, which means it is not difficult for a trained user to find a satisfactory visualization.

We list the experimental results in Table 3.3, where N is the number of rows in the given dataset, k is dimensionality of the dataset, and c is the number of clusters in the dataset. "UGV(%)" is the error rates (%) of unguided visual rendering result. "Combo(%)" is the error rates(%) of the combining of UGV with "landmark points". "Time(min)" is the time (minutes) spent in interaction in the "combo" method, which begins at loading a dataset and ends with saving the interaction result. "CURE(%)" is the error rates (%) of CURE clustering algorithm. "KM(%)" is the error rates (%) of K-means clustering algorithm. The results show that the combined rendering can improve the algorithmic cluster definition greatly, and even the unguided visual rendering can give better result in many cases.

4 Related Work

The common cluster analysis framework is described in the clustering review paper [20]. Recently, some algorithms have been developed to deal with arbitrarily shaped clusters. CURE [15] uses a set of representative points to describe the boundary of a cluster in its hierarchical algorithm. But the number of representative points increases dramatically with the increase of the complexity of cluster shapes in order to maintain the precision. CHAMELEON [23] employs a multilevel graph partitioning algorithm on the k-Nearest Neighbour graph, which may produce better results than

Dataset	N	k	c	UGV(%)	Combo(%)	Time (min)	CURE (%)	KM(%)
Bre-canc-wisc	699	10	2	16.7	3.3 ± 0.4	1.6 ± 0.3	8.6	5.1
Crx	690	15	2	20.2	14.5 ± 0.3	2.3 ± 0.8	46.8	48.0
Iris	151	4	3	5.5	0.7 ± 1.2	1.9 ± 0.3	32.7	11.3
Page-blocks	5473	10	5	13.0	7.8 ± 0.2	2.6 ± 0.4	32.7	45.6
Hepatitis	155	19	2	21.9	21.7 ± 2.4	2.7 ± 0.2	35.7	42.6
Heart	270	12	2	24.0	18.6 ±1.5	2.3 ± 0.5	47.4	21.1
Mushroom	8124	21	2	24.7	8.4 ± 0.3	5.5 ± 0.4	31.9	40.2
Australian	690	14	2	15.4	14.6 ± 0.7	2.6± 0.9	36.8	14.5
Wine	178	12	3	7.9	2.1 ± 0.5	3.1 ± 0.4	32.0	5.6
Shuttle.test	14500	9	7	10.2	4.0 ± 0.4	1.7 ± 0.2	20.5	23.2

Table 1: More experimental results on typical datasets having irregular cluster distribution

CURE on complex cluster shapes for spatial datasets. But the high complexity of the algorithm prevents its application on higher dimensional datasets. DBSCAN [10] is a density-based algorithm but it is very sensitive to the parameter Eps and $MinPts$. The distribution-based algorithm DBCLASD [31] and the wavelet transformation based algorithm WaveCluster [13] were also reported as being efficient only in spatial datasets. In conclusion, the automatic algorithms can deal with the arbitrarily shaped clusters in some special situations, but the results are not generally applied to any applications.

The most difficult problem is, while it is easy to validate the arbitrarily shaped clusters in 2D or 3D space (visually or algorithmically), it is hard to validate the irregular clusters for high-dimensional ($>$3D) datasets, since the commonly used statistical indices are only effective for regular shapes [16], such as spherical or elongated shapes. Information visualization is commonly recognized as a useful method for understanding sophistication in datasets. Many efforts have been made to analyze the datasets in a visual way. We discuss the scatter-plot-based techniques only because it is the most intuitive techniques for cluster visualization. The early research on general plot-based data visualization is Grand Tour and Projection Pursuit [7]. Since there are numerous projections from a multidimensional data space to a 2D space, the purpose of the Grand Tour and the Project Pursuit is to guide the user to find the “interesting projections” automatically by interpolating the intermediate projections to connect several pre-selected projections. L.Yang [32] utilizes the Grand Tour technique to show projections in an animation. VISTA system has proved that the property of dataset can be utilized in user interaction to help find the satisfactory visualization quickly [4], rather than mathematically interpolating most uninteresting projections, which cost a lot computational power. Dhillon [9] aimed at precise visualization of clusters, but the technique is effective for 3 clusters. When more than 3 clusters exist, his method needs the help of Grand Tour techniques. Other techniques, such as Scatterplot matrices, coplots, prosection [6] and FastMap based visualization [11, 18] only create static visualization, which inevitably distorts the cluster structure but have no complementary methods to rectify it, thus do not provide enough information for either clustering or cluster validation. In KDD2002 tutorial [14], more visualization methods were discussed.

Star Coordinates [22] is a visualization system designed to visualize and analyze the clusters interactively. We utilize the form of Star Coordinates and build a normalized α-mapping model in our system. We have discussed that α-mapping model extends the ability of the original mapping in star coordinates paper and demonstrated the particular ability of VISTA system in visually validating and refining clusters. HD-Eye [17] is another interesting interactive visual clustering system. HD-Eye visualizes the density-plot of the interesting projection of any two of the k dimensions. It uses icons to represent the possible clusters in each projection and the relationship between the clusters. However, it is hard for users to synthesize all of the interesting 2 of k-D projections to find the general pattern of the clusters. In fact, visually determining the basic cluster distribution solely through user interaction is not necessary. The semi-automatic 1D visualization based algorithm OPTICS [1] actually works well in finding the 1D sketch of arbitrarily shaped clusters, the result of which can be utilized by some high-level visualization systems, such as VISTA. However, OPTICS itself cannot be easily applied to analyze or validate the shape of clusters and the distance relationship between clusters, and to incorporate any domain knowledge in cluster analysis, as VISTA does.

5 Discussion and Conclusion

Most of researchers have focused on automatic clustering algorithms, but very few have addressed the human factor in the clustering process. Although the existing clustering algorithms and cluster validity methods are working well on spherical clusters with the normal statistical assumptions, they have encountered the difficulty in dealing with arbitrarily shaped clusters. In order to solve this problem, we should check the human factor in the clustering process more carefully. The VISTA system demonstrates some possible ways to introduce the users into the clustering process. In this paper, we proposed the VISTA visual framework to combine the algorithmic results with visual cluster rendering system. The power of VISTA visual cluster rendering system enables the users to visually validate and interactively refine the clusters. It also allows the users to incorporate domain knowledge into the clustering process in a convenient way. The empirical study shows that the VISTA framework/system works very well in visually validating and refining the algorithmic clustering results. A short version of this paper [5] was presented in international conference of data mining (2003). We have published the downloadable VISTA system at the web site http://disl.cc.gatech.edu/VISTA/.

The current VISTA system can handle datasets with dimensionality less then 50 interactively. Dimensionality higher than or close to 50 will cause the difficulty in human visual understanding and operations. In such situations, linear dimensionality reduction techniques, such as Principal Component Analysis (PCA) [21], or non-linear techniques [8] are needed to reduce the dimensionality. It is commonly agreed that the higher dimensionality remained, the less the dataset is distorted, and the more precise the clustering result is. For instance, higher dimensionality tends to have more *variance explained* in PCA. The theoretical result [3] shows that the first few principal components (PCs) given by PCA may not be the 'best' PCs for clustering. Therefore, 2/3D visualizations directly based on the first 2/3 PCs may not visualize the clusters correctly. However, to find the best few PCs for clustering is very difficult and complicated. Visualizations based on transformed dataset having higher dimensionality ($>$ 3D) will have higher chance to preserve the original dataset without involving complicated computing. Investigating the effect of dimensionality reduction to VISTA system for very high dimensional datasets is one of the main subject for VISTA approach.

Computer system capability also restricts the number of data items that can be handled. In current experimental system (Pentium4 1.6G, 256M), the VISTA system can handle about 100K points while refreshing the visualization in real-time ($<$several hundreds of milliseconds). Huge datasets may bring up several new issues in terms of the effect of preprocessing to visual cluster rendering, understanding and manipulating the visualization, and effective post-processing with visualization. In conclusion, VISTA system for very high dimensional datasets and very large datasets will be the focus of the future work.

References

[1] M. Ankerst, M. M. Breunig, H.-P. Kriegel, and J. Sander. Optics: Ordering points to identify the clustering structure. *Proc. of ACM SIGMOD*, 1999.

[2] A. Blum and T. Mitchell. Combining labeled and unlabeled data with co-training. In *COLT: Proceedings of the Workshop on Computational Learning Theory*.

[3] W. Chang. On using principal components before separating a mixture of two multivariate normal distributions. *Applied Statistics*, 32, 1983.

[4] K. Chen and L. Liu. Cluster rendering of skewed datasets via visualization. *Proc. of ACM SAC*, 2003.

[5] K. Chen and L. Liu. Validating and refining clusters via visual rendering. *Proc. of Intl. Conf. on Data Mining*, 2003.

[6] W. S. Cleveland. Visualizing data. *AT&T Bell Laboratories*, 1993.

[7] D. Cook, A. Buja, J. Cabrera, and C. Hurley. Grand tour and projection pursuit. *Journal of Computational and Graphical Statistics*, 23, 1995.

[8] D. DeMers and G. Cottrell. Non-linear dimensionality reduction. In S. J. Hanson, J. D. Cowan, and C. L. Giles, editors, *Advances in Neural Information Processing Systems*, volume 5, pages 580–587. Morgan Kaufmann, San Mateo, CA, 1993.

[9] I. S. Dhillon, D. S. Modha, and W. S. Spangler. Visualizing class structure of multidimensional data. *the 30th Symposium on the Interface: Computing Science and Statistics*, 1998.

[10] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. *Second International Conference on Knowledge Discovery and Data Mining*, 1996.

[11] C. Faloutsos and K.-I. D. Lin. Fastmap: A fast algorithm for indexing, data-mining and visualization of traditional and multimedia datasets. *Proc. of ACM SIGMOD*, 1995.

[12] J. Gallier. Geometric methods and applications for computer science and engineering. 2000.

[13] S. C. Gholamhosein Sheikholeslami and A. Zhang. Wavecluster: A multi-resolution clustering approach for very large spatial databases. *Proc. of VLDB*, 1998.

[14] G. Grinstein, M. Ankerst, and D. A. Keim. Visual data mining: Background, applications, ad drug discovery applications. *Proc. of ACM SIGMOD*, 1999.

[15] S. Guha, R. Rastogi, and K. Shim. Cure: An efficient clustering algorithm for large databases. *Proc. of ACM SIGMOD*, 1998.

[16] M. Halkidi, Y. Batistakis, and M. Vazirgiannis. Cluster validity methods: Part i and ii. *SIGMOD Record*, 31(2), 2002.

[17] A. Hinneburg, D. A. Keim, and M. Wawryniuk. Visual mining of high-dimensional data. *IEEE Computer Graphics and Applications*, 1999.

[18] Z. Huang, D. W. Cheung, and M. K. Ng. An empirical study on the visual cluster validation method with fastmap. *the proc. of DSFAA*, 2001.

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

[20] A. K. Jain and R. C. Dubes. Data clustering: A review. *ACM Computing Surveys*, 1999.

[21] I. Jolliffe. *Principal Component Analysis*. Spring Verlag, 1986.

[22] E. Kandogan. Visualizing multi-dimensional clusters, trends, and outliers using star coordinates. *Proc. of ACM SIGKDD*, 2001.

[23] G. Karypis, E.-H. S. Han, and V. Kumar. Chameleon: hierarchical clustering using dynamic modeling. *IEEE Computer*, (8), 1999.

[24] D. Keim. Visual exploration of large data sets. *Comm. ACM*, 44(8), 2001.

[25] J. Larkin and H. Simon. Why a diagram is (sometimes) worth ten thousand words. *Cognitive Science*, 1987.

[26] H. Liu and H. Motoda. *Feature Extraction, Construction and Selection: A Data Mining Perspective*. Kluwer Academic Publishers, 1998.

[27] T. Mitchell. *Machine Learning*. McGraw Hill, 1997.

[28] S. Russel and P. Norvig. *Artificial Intelligence: a Modern Approach*. Prentice Hall, 1995.

[29] S. Sharma. *Applied Multivariate Techniques*. Wiley&Sons, 1995.

[30] B. Shneiderman. Inventing discovery tools: Combining information visualization with data mining. *Information Visualization*, 2002.

[31] X. Xu, M. Ester, H.-P. Kriegel, and J. Sander. A distribution-based clustering algorithm for mining in large spatial databases. *Proc. of IEEE Intl. Conf. on Data Eng. (ICDE)*, 1998.

[32] L. Yang. Interactive exploration of very large relational datasets through 3d dynamic projections. *Proc. of ACM SIGKDD*, 2000.

[33] T. Zhang, R. Ramakrishnan, and M. Livny. Birch: An efficient data clustering method for very large databases. *Proc. of ACM SIGMOD*, 1996.