A Visual Approach for External Cluster Validation

Ke-Bing Zhang, Mehmet A. Orgun, Senior Member, IEEE, and Kang Zhang, Senior Member, IEEE

Abstract—Visualization can be very powerful in revealing cluster structures. However, directly using visualization techniques to verify the validity of clustering results is still a challenge. This is due to the fact that visual representation lacks precision in contrasting clustering results. To remedy this problem, in this paper we propose a novel approach, which employs a visualization technique called HOV$^3$ (Hypothesis Oriented Verification and Validation by Visualization) which offers a tunable measure mechanism to project clustered subsets and non-clustered subsets from a multidimensional space to a 2D plane. By comparing the data distributions of the subsets, users not only have an intuitive visual evaluation but also have a precise evaluation on the consistency of cluster structure by calculating geometrical information of their data distributions.

I. INTRODUCTION

The goal of clustering is to distinguish objects into partitions/clusters based on given criteria. A large number of clustering algorithms have been developed for different application purposes [8, 14, 15]. However, due to the memory limitation of computers and the extremely large sized databases, in practice, it is infeasible to cluster entire data sets at the same time. Thus, applying clustering algorithms to sampling data to extract hidden patterns is a commonly used approach in data mining [5]. As a consequence of sampling data cluster analysis, the goal of external cluster validation is to evaluate a well-suited cluster scheme learnt from a subset of a database to see whether it is suitable for other subsets in the database. In real applications, achieving this task is still a challenge. This is not only due to the high computational cost of statistical methods for assessing the robustness of cluster structures between the subsets of a large database, but also due the non-linear time complexity of most existing clustering algorithms.

Visualization provides users an intuitive interpretation of cluster structures. It has been shown that visualization allows for verification of the clustering results [10]. However, the direct use of visualization techniques to evaluate the quality of clustering results has not attracted enough attention in the data mining community. This might be due to the fact that visual representation lacks precision in contrasting clustering results.

We have proposed an approach called HOV$^3$ to detect cluster structures [28]. In this paper, we discuss its projection mechanism to support external cluster validation. Our approach is based on the assumption that by using a measure to project the data sets in the same cluster structure, the similarity of their data distributions should be high. By comparing the distributions produced by applying the same measures to a clustered subset and other non-clustered subsets of a database by HOV$^3$, users can investigate the consistency of cluster structures between them both in visual form and in numerical calculation.

The rest of this paper is organized as follows. Section 2 briefly introduces ideas of cluster validation (with more details of external cluster validation) and visual cluster validation. A review of related work on cluster validation by visualization, and a more detailed account of HOV$^3$ are presented in Section 3. Section 4 describes our idea on verifying the consistency of cluster structure by a distribution matching based method in HOV$^3$. Section 5 demonstrates the application of our approach on several well-known data sets. Finally, section 6 summarizes the contributions of this paper.

II. BACKGROUND

A. Cluster Validation

Cluster validation is a procedure of assessing the quality of clustering results and finding a cluster strategy fit for a specific application. It aims at finding the optimal cluster scheme and interpreting the cluster patterns. In general, cluster validation approaches are classified into the following three categories [9, 15, 27].

Internal approaches: they assess the clustering results by applying an algorithm with different parameters on a data set for finding the optimal solution [1];

Relative approaches: the idea of relative assessment is based on the evaluation of a clustering structure by comparing it to other clustering schemes [8]; and

External approaches: the external assessment of a clustering approach is based on the idea that there exists known priori clustered indices produced by a clustering algorithm, and then assessing the consistency of the clustering structures generated by applying the clustering algorithm to different data sets [12].
B. External Cluster Validation

As a necessary post-processing step, external cluster validation is a procedure of hypothesis test, i.e., given a set of class labels produced by a cluster scheme, compare it with the clustering results by applying the same cluster scheme to the other partitions of a database, as shown in the Fig. 1.

![Fig. 1. External criteria based validation](image)

The statistical methods for quality assessment are employed in external cluster validation, such as Rand statistic [24], Jaccard Coefficient [7], Folkes and Mallows index [21], Huberts \( \Gamma \) statistic and Normalized \( \Gamma \) statistic [27], and Monte Carlo method [20], to measure the similarity between the priori modeled partitions and clustering results of a dataset. However, achieving these tasks is time consuming when the database is large, due to the drawback of high computational cost of statistics-based methods for assessing the consistency of cluster structure between the sampling subsets. Recent surveys on cluster validation methods can be found in the literatures [10, 12, 27].

C. Visual Cluster Validation

In high dimensional space, traditional clustering algorithms tend to break down in terms of efficiency as well as accuracy because data do not cluster well anymore [3]. Thus, introducing visualization techniques to explore and understand high-dimensional datasets is becoming an efficient way to combine human intelligence with the immense brute force computation power available nowadays [23]. Visual presentations can be very powerful in revealing trends, highlighting outliers, showing clusters, and exposing gaps in data [26].

Visual cluster validation is a combination of information visualization and cluster validation techniques. In the cluster analysis process, visualization provides analysts with intuitive feedback on data distribution and supports decision-making activities.

III. RELATED WORK

A. Previous Works

A large number of clustering algorithms have been developed, but only a small number of cluster visualization tools are available to facilitate researchers’ understanding of the clustering results [25]. Several efforts have been made in cluster validation with visualization [2, 4, 11, 13, 16, 18]. While, these techniques tend to help users have intuitive comparisons and better understanding of cluster structures, but they do not focus on assessing the quality of clusters.

For example, OPTICS [2] uses a density-based technique to detect cluster structures and visualizes them in “Gaussian bumps”, but its non-linear time complexity makes it neither suitable to deal with very large data sets, nor suitable to provide the contrast between clustering results. Kaski et. al [18] imposes the technique of Self-organizing maps (SOM) technique to project multidimensional data sets on a 2D space for matching visual models [17]. However, the SOM technique is based on a single projection strategy and not powerful enough to discover all the interesting features from the original data. Huang et. al [11, 13] proposed approaches based on FastMap [5] to assist users on identifying and verifying the validity of clusters in visual form. Their techniques are good on cluster identification, but are not able to evaluate the cluster quality very well.

The most prominent feature of techniques based on Star Coordinates, such as VISTA [4] and HOV\(^3\) [28], is their linear time computational complexity. This feature makes them suitable to be used as visual interpretation and detection tools in cluster analysis. However, the characteristic of imprecise qualitative analysis of Star Coordinates and VISTA limits them as quantitative analysis tools. In addition, VISTA adopts “landmark” points as representatives from a clustered subset and re-samples them to deal with cluster validation [4]. But its experience-based “landmark” point selection does not always handle the scalability of data very well, due to the fact that well-representative landmark points selected in a subset may fail in other subsets of a database.

Visualization techniques used in data mining and cluster analysis are surveyed in the literatures [22, 25].

B. Star Coordinates

The approach HOV\(^3\) employed in this research was inspired from the Star Coordinates [16]. For better understanding our work, we briefly describe it here.

Star Coordinates utilizes a point on a 2D surface to represent a set of points of \( n \)-dimensional data. The values of \( n \)-dimensional coordinates are mapped to the orthogonal coordinates \( X \) and \( Y \), as shown in Fig. 2.

![Fig. 2. Positioning a point by an 8 attribute vector in Star Coordinates](image)
The mapping from n-dimensional Star Coordinates to 2D X-Y coordinates is calculated as in Formula (1).

\[
p_f(x, y) = \sum_{i=1}^{n} \frac{d_{ji}}{m_i} - \min_j \sum_{i=1}^{n} \frac{d_{ji}}{m_i} \min_i
\]  

(1)

where \( p_f(x, y) \) is the normalized location of \( D_j = (d_{1j}, d_{2j}, \ldots, d_{nj}) \), and \( d_{ji} \) is the value of \( j \)th record of \( i \)th dimension of the data set on the \( i \)th coordinate \( C_i \) in Star Coordinates space; \( \bar{u}_{xj}(d_{ji} - \min_j) \) and \( \bar{u}_{yj}(d_{ji} - \min_j) \) are unit vectors of \( d_{ji} \) mapping to X direction and Y direction, \( \min_j = \min(d_{ji}, 0 \leq j < m) \) max\(_j\) = max \( (d_{ji}, 0 \leq j < m) \) are minimum and maximum values of \( i \)th dimension respectively; and \( m \) is the number of records in the data set.

C. HOV\(^3\) Model

The idea of HOV\(^3\) is based on hypothesis test by visualization. It treats hypotheses as measures to reveal the difference between hypotheses and real performance by projecting the test data against the measures [28]. Geometrically, the difference of a matrix \( D_i \) and a vector \( M \) can be represented by their inner product, \( D_i \cdot M \). Let \( D_i = (d_{i1}, d_{i2}, \ldots, d_{im}) \) be a data set with \( n \) attributes, and \( M = (m_1, m_2, \ldots, m_n) \). The inner product of each vector \( d_{ij} \) with \( M \) can be seen as a mapping from an \( n \)-dimensional data set to one measure \( F: R^n \rightarrow R \). It is written as:

\[
< d_{ij}, M > = m_{d_{i1}} + m_{d_{i2}} + \ldots + m_{d_{im}} = \sum_{k=1}^{n} m_{d_{ik}}
\]  

(2)

In order to enlarge the data analysis space, we introduce the complex number system into our study. Let \( z = x + i.y \), where \( i \) is the imaginary unit. According to the Euler formula, we have: \( e^{ix} = \cos x + i\sin x \). Let \( z_0 = e^{2\pi i/n} \); we see that \( z_0, z_0^2, z_0^3, \ldots, z_0^{n-1}, z_0^n \) (with \( z_0^n = 1 \)) divide the unit circle on the complex plane into \( n-1 \) equal sectors. Then the formula (1) can be simply written as:

\[
P_f(z_0) = \sum_{k=1}^{n} \left( d_{jk} - \min_{k} \right) / \left( \max_{k} d_{jk} - \min_{k} \right) \cdot z_0^k
\]  

(3)

where \( \min_{k} d_{jk} \) and \( \max_{k} d_{jk} \) represents the minimal and the maximal values of the \( k \)th coordinate respectively. This is the case of equally-divided circle surface. Then the more general form can be defined as:

\[
P_f(z_k) = \sum_{k=1}^{n} \left( d_{jk} - \min_{k} \right) / \left( \max_{k} d_{jk} - \min_{k} \right) \cdot z_k^k
\]  

(4)

where \( z_k = e^{i\theta} \); \( \theta \) is the angle of neighboring axes; and \( \sum_{k=1}^{n} \theta = 2\pi \). In any case equation (4) can be viewed as mappings from \( R^n \rightarrow C^2 \).

Given a non-zero measure vector \( m \) in \( R^n \), and a family of vectors \( P_i \), and the projections of \( P_i \) against \( m \) according to formulas (2) and (4), the HOV\(^3\) model is given as the following equation:

\[
P_f(z_k) = \sum_{k=1}^{n} \left( d_{jk} - \min_{k} \right) / \left( \max_{k} d_{jk} - \min_{k} \right) \cdot z_k \cdot m_k
\]  

(5)

where \( m_k \) is the \( k \)th attribute of measure \( m \).

As shown above, a hypothesis in HOV\(^3\) is a quantified measure vector. Thus HOV\(^3\) is also able to detect the consistency of cluster structures among the subsets of a database by comparing their data distributions, because cluster validation procedure is primarily a hypothesis test process.

D. The Axis Tuning Feature

Overlapping and ambiguities are inevitably introduced by projecting multidimensional data into 2D space. For mitigating the problem, Star Coordinates provides several visual adjustment mechanisms, such as axis scaling, axes angles rotation; coloring data points, etc [15]. We use Iris, a well-known data set in machine learning research, as an example to demonstrate the feature of axis scaling of techniques based on Star Coordinates as follows.
This axis-tuning feature is significant for our external cluster validation method based on distribution matching by HOV\(^3\). We give the detailed explain next.

IV. CLUSTER VALIDATION WITH HOV\(^3\)

The feature of tunable axis provides us a mechanism to quantitatively handle the external cluster validity by HOV\(^3\). Our approach is based on the assumption that by using a measure to project the data sets in the same cluster structure, the similarity of their data distributions should be high. Based on this idea, we have implemented an approach for external cluster validation based on distribution matching by HOV\(^3\).

A. Definitions

To explain our approach precisely, we first give a few definitions below.

**Definition 1:** A data projection from n-dimensional space to 2D plane by applying HOV\(^3\) to a data set \(\mathcal{P}\), as shown in formula (5), is denoted as \(D_p\mathcal{=HV}(\mathcal{P}, M)\), where

- \(\mathcal{P}\) is an n-dimensional data set, and \(\mathcal{P} = (p_1, p_2, \ldots, p_n)\), \(p_i\) (1 ≤ \(i\) ≤ \(n\)) is an instance of \(\mathcal{P}\);
- \(M = (w_1, w_2, \ldots, w_d)\) is a non-zero measure vector; \(w_d\) (1 ≤ \(i\) ≤ \(n\)) is the weight value of \(\mathcal{P}\) at \(t\) moment in the Star Coordinates plane;
- \(D_p\) is the geometrical distribution of \(\mathcal{P}\) in 2D space, \(D_p = (p(x_1,y_1), p(x_2,y_2), \ldots, p_n(x_n,y_n))\), \(p(x,y)\) is the location of \(p\) in X-Y Coordinates plane.

**Definition 2:** Let \(\mathcal{D}\) be a database of data points. A cluster \(C := (D, L)\) is a non-empty set \(D \subseteq \mathcal{D}\) on a label set \(L\), and the \(i\)th cluster \(C_i = \{p \epsilon \mathcal{P}, L \epsilon L \mid C_i \epsilon \mathcal{P}, L \epsilon L \epsilon \mathcal{P}\}\) where \(i\) is the cluster label of \(p\), \(L \epsilon \{-1, 0, 1, \ldots, k\}\), and \(k\) is the number of clusters.

As special case, an outlier point is an element of \(\mathcal{D}\) and with cluster label –1; a non-clustered element of \(\mathcal{D}\) has a cluster label of 0, i.e., it has not been clustered.

**Definition 3:** A spy subset \(\mathcal{P}_s\) is a clustered subset of \(\mathcal{D}\) produced by a clustering algorithm, where \(\mathcal{P}_s = \{C_1, C_2, \ldots, C_i, C_k\}\), \(C_i\) (1 ≤ \(i\) ≤ \(k\)) is a cluster in \(\mathcal{P}_s\); \(C_k\) is the outlier set of \(\mathcal{P}_s\).

A spy subset is used as a model to verify the cluster structure in the other partitions in the database \(\mathcal{D}\).

**Definition 4:** A subset \(\mathcal{P}_t \subseteq \mathcal{D}\) is a target subset of \(\mathcal{P}_s\), \(\mathcal{P}_t = \{p \epsilon \mathcal{D}, L \epsilon L \mid \mathcal{P}_t \epsilon \mathcal{D}, \mathcal{P}_t \epsilon \mathcal{D}\}\).

A target subset \(\mathcal{P}_t\) is a non-clustered subset of \(\mathcal{D}\) and has the same size of a spy subset \(\mathcal{P}_s\) of \(\mathcal{D}\). It is used as a target to investigate the similarity of cluster structure with the spy subset \(\mathcal{P}_s\).

**Definition 5:** A non-clustered point \(p_o\) is called an overlapping point of a cluster \(C_i\), denoted as \(C_o, p_o\) iff \((\exists p \epsilon C_i \wedge p_o \neq C_i \wedge |p - p_o| \leq \delta)\), where \(\delta\) is the threshold distance given by the user.

**Definition 6:** The overlapping point set of cluster \(C_i\) is composed as a quasi-cluster of \(C_i\), denoted as \(C_{qi}\), i.e., \(\{p \epsilon \mathcal{P}, |C_o, p_o\}\)

All overlapping points of \(C_i\) are composed a quasi-cluster \(C_{qi}\) of \(C_i\).

**Definition 7:** A cluster \(C_i\) is called a well-separated cluster visually, when it satisfies the condition that \((C_i \subseteq \mathcal{P}, C_i \subseteq \mathcal{P} \setminus \{C_i\} \setminus \{C_i, C_j\})\).

A well-separated cluster \(C_i\) in the spy subset implies that no points in \(C_i\) are within the threshold distance \(\delta\) in any other clusters in the spy subset.

Based on above the definitions, we present the application of our approach to external cluster validation based on distribution matching by HOV\(^3\) as follows.

B. The Stages of Our Approach

The stages of the application of our approach are summarized in the following steps:

1. Clustering

First, the user applies a clustering algorithm to a randomly selected subset \(\mathcal{P}_s\) from the given dataset \(\mathcal{D}\).

2. Cluster Separation

The clustering result of \(\mathcal{P}_s\) is introduced and visualized in HOV\(^3\). Then the user manually tunes the weight value of \(\mathcal{P}_s\) in HOV\(^3\). The clustered subset \(\mathcal{P}_s\) (now as a spy subset) and its target subset \(\mathcal{P}_t\) are projected together by HOV\(^3\) with vector \(M\) to detect the distribution consistency between \(\mathcal{P}_s\) and \(\mathcal{P}_t\).

3. Data Projection by HOV\(^3\)

The user samples another observation with the number of points as in \(\mathcal{P}_s\) as a target subset \(\mathcal{P}_t\). The clustered subset \(\mathcal{P}_s\) is projected by HOV\(^3\) with vector \(M\) to detect the distribution consistency between \(\mathcal{P}_s\) and \(\mathcal{P}_t\).

4. The Generation of Quasi-Clusters

The user gives a threshold \(\delta\), and then according to Definitions 5, 6 and 7, a quasi-cluster \(C_{qi}\) of a separated cluster \(C_i\) is computed. Then \(C_{qi}\) is removed from \(\mathcal{P}_s\) and \(C_i\) is removed from \(\mathcal{P}_t\). If \(\mathcal{P}_s\) has clusters then we go back to step 2, otherwise we proceed to the next step.

5. The Interpretation of Result

The overlapping rate of each cluster-and-quasi-cluster pair is calculated as \((C_{qi}, C_i) = |C_{qi} \setminus C_i| \div |C_{qi}|\). If the overlapping rate approaches 1, cluster \(C_i\) and its quasi-clusters \(C_{qi}\) have high similarity, since the amount ratio of the spy subset and the target subset is 1:1. Thus the overlapping analysis is simply transformed into a linear regression analysis, i.e., the points around the line \(C = C_{qi}\).
Corresponding to the procedure mentioned above, we give the algorithm of external cluster validation based on distribution matching by HOV$^3$ below, in Fig. 5.

\begin{verbatim}
DistributionMatching (target, spy)
{  s – threshold distance
   For (c\in cluster \subseteq spy) {
      c:\= clusterSeparate(spy);
      M\= (\forall \text{weightValue}\in\text{axes});
      \text{ds}\text{-}\text{PC}(\text{spy}\times\text{target}, M);
      c\text{\_}\text{target}\text{-}\text{ClusterGeneration} (\text{ds}\times\text{target}, c, \delta);
      \text{output}(c/c\text{\_}\text{target});
   }
}
Procedure clusterSeparate (spy)
{  For (\forall \text{overlapping cluster}\subseteq\text{spy}) {
      \text{axisScaling}(\text{spy});
      \text{c}\text{-}\text{noOverlappingCluster} ;
      \text{weightValue}\text{-}\text{axisWeight};
   }
}
Procedure QuasiClusterGeneration (ds, target, c, \delta)
{  For (\forall a \in \text{target}) {
      if (\exists b \text{\_}\text{target} \in (\text{ds}\times\text{a-b}) \& \text{\leq} \delta)
         c\text{\_}\text{target}\text{-}\text{add}(b);
   }
}
\end{verbatim}

Fig 5. The algorithm of external cluster validation based on distribution matching in HOV$^3$

In Fig. 5, the procedure \textit{clusterSeparate} responds the user’s axis tuning to separate the clusters in the spy subset, and to gather weight values of axes as a measure vector; the procedure \textit{quasiClusterGeneration} produces quasi clusters in the target subset corresponding to the clusters in the spy subset.

\section{Our Model}

In contrast to statistics-based external cluster validation model illustrated in Fig. 2, we exhibit our model for external cluster validation by visualization in HOV$^3$ in Fig. 6.

Fig. 6. External cluster validation by HOV$^3$

Comparing these two models, we may observe that instead of using a clustering algorithm to cluster another sampling data sets, in our model, we use a clustered subset from a database as a model to verify the similarity of cluster structure between the model and the other non-clustered subsets from the database. To handle the scalability on resampling datasets, we choose the non-cluster observations with the same size as the clustered subset, and then project them together by HOV$^3$. As a consequence, the user can easily utilize the well-separated clusters produced by scaling axes in HOV$^3$ as a model to pick out their corresponding quasi-clusters, where points in a quasi-cluster overlap its corresponding cluster. Also, instead of using statistical methods to assess the similarity between the two subsets, we simply compute the overlapping rate between the clusters and their quasi-clusters to explore their consistency.

\section{Examples and Explanation}

In this section, we present several examples to demonstrate the advantages of the external cluster validation in HOV$^3$. We have implemented our approach in MATLAB running under Windows 2000 Professional. The datasets in the examples are obtained from the UCI machine learning website: \url{http://www.ics.uci.edu/~mlearn/Machine-Learning.html}

Shuttle data set has 9 attributes and 4,3500 instances. We choose the first 5,000 instances of Shuttle as a sampling data and apply the K-means algorithm [19] to it. Then we utilize the clustered result as a \textit{spy subset}. We assumed that we have found the optimal cluster number k=5 for the sampling data. The original data distributions with and without cluster indices are illustrated in the diagrams of Fig.7 and Fig.8 respectively. It can been seen that there exists a cluster overlapping in Fig. 8.

To obtain well-separated clusters, we tuned the weight of each coordinate, and had a satisfied version of the data distribution as shown in Fig. 9. The weight values of axes are recorded as a measure vector \[0.80, 0.55, 0.85, 0.0, 0.40, 0.95, 0.20, 0.05, 0.459\], in this case. Then we chose the second 5,000 instances of Shuttle as a \textit{target subset} and projected the target subset and the spy subset together against the measure vector by HOV$^3$. 

Fig. 7. The original data distribution of the first 5,000 data points of Shuttle in MATLAB by HOV$^3$ (without cluster indices)
cluster are listed in Table 1, and their curves of linear regression to the line $C=C_q$ are illustrated in Fig. 11.

### TABLE I

<table>
<thead>
<tr>
<th>Subset</th>
<th>$C_q/C_1$</th>
<th>$C_q/C_2$</th>
<th>$C_q/C_3$</th>
<th>$C_q/C_4$</th>
<th>$C_q/C_5$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spy</td>
<td>318</td>
<td>773</td>
<td>513</td>
<td>2254</td>
<td>1142</td>
</tr>
<tr>
<td>Target 1</td>
<td>278/318</td>
<td>670/773</td>
<td>503/513</td>
<td>2459/2254</td>
<td>1123/1142</td>
</tr>
<tr>
<td>Target 2</td>
<td>279/318</td>
<td>897/773</td>
<td>626/513</td>
<td>2048/2254</td>
<td>1602/1142</td>
</tr>
<tr>
<td>Target 3</td>
<td>280/318</td>
<td>875/773</td>
<td>481/513</td>
<td>2093/2254</td>
<td>1455/1142</td>
</tr>
<tr>
<td>Target 4</td>
<td>261/318</td>
<td>713/773</td>
<td>368/513</td>
<td>2416/2254</td>
<td>1169/1142</td>
</tr>
</tbody>
</table>

*At current stage, we collect the quasi-clusters manually, thus $C_q$ here may have redundancy and misloading.

It is observed that the curves are well matched to the line $C=C_q$, i.e. the overlapping rate between the clusters and their quasi-clusters are high. The standard deviation is a good way to reflect the difference between the two vectors. Thus we have calculated the standard deviation of each $C_q-C_i$ pairs among the target ($k=1,...,4$) and the spy subsets. They are 0.0826, 0.1975, 0.1491 and 0.1304. This means that the similarity of cluster structure in the spy and the target subsets is high.

In summary, the experiments show that the same cluster structure in the spy subset of Shuttle also exists in the target subsets of Shuttle.

#### In these experiments, we have also measured the timing for both clustering and projection in MATLAB. The results are listed in the Table 2.

![Fig. 8](image1.png)

**Fig. 8.** The original data distribution of the first 5,000 data points of Shuttle in MATLAB by HOV (with cluster indices)

Their distributions are presented in Fig. 10, where we may observe that their data distributions are matched very well. We chose the points in the enclosed area in Fig. 10 as a “cluster” then obtained a quasi-cluster in the target subset corresponding to the cluster in the enclosed area. In the same way, we can find the other quasi-clusters from the target subset.

![Fig. 9](image2.png)

**Fig. 9.** A well-separated version of the spy subset distribution of Shuttle

![Fig. 10](image3.png)

**Fig. 10.** The projection of the spy subset and a target subset of Shuttle by applying a measure vector.

We have done the same experiment on 4 target subsets of Shuttle. The size of each quasi-cluster and its corresponding
TABLE 2
TIMING OF CLUSTERING AND PROJECTING

<table>
<thead>
<tr>
<th>Clustering by K-means (k=5)</th>
<th>Projecting by HOV^3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Subset Size</td>
<td>Time (Second)</td>
</tr>
<tr>
<td>Target 1</td>
<td>5,000</td>
</tr>
<tr>
<td>Target 2</td>
<td>5,000</td>
</tr>
<tr>
<td>Target 3</td>
<td>5,000</td>
</tr>
<tr>
<td>Target 4</td>
<td>5,000</td>
</tr>
</tbody>
</table>

Based on this calculation, it has been observed that the projection by HOV^3 is much faster than the clustering process by the K-means algorithm. It is particularly effective for verifying the clustering results within extremely huge databases. Although the cluster separation in our approach may incur some time, once the well-separated clusters are found, using a measure vector to project a huge data set will be a lot more efficient than re-applying a clustering algorithm to the data set.

VI. CONCLUDING REMARKS

In this paper we have proposed a novel visual approach to assist users to verify the validity of any cluster scheme, i.e., an approach based on distribution matching for external cluster validation by visualization. The HOV^3 visualization technique has been employed in our approach, which uses measure vectors to project a data set and allows the user to iteratively adjust the measures for optimizing the result of clusters.

By comparing the data distributions of a clustered subset and non-clustered subsets projected by HOV^3 with tunable measures, users can perform intuitive visual evaluation, and also have a precise evaluation on the consistency of the cluster structure by performing geometrical computation on their data distributions as well. By comparing our approach with existing visual methods, we have observed that our method is not only efficient in performance, but also effective in real applications.

REFERENCES