

PolyCluster: An Interactive Visualization Approach to Construct Classification Rules

Danyu Liu, Alan P. Sprague and Jeffrey G. Gray
Department of Computer & Information Science
University of Alabama at Birmingham 153 Campbell Hall
1300 Univ. Blvd Birmingham AL 35294-1170
{liudy, sprague, gray}@cis.uab.edu

Abstract

This paper introduces a system, called PolyCluster, which adopts state-of-the-art algorithms for data visualization and integrates human domain knowledge into the construction process of classification rules. By utilizing PolyCluster, users can obtain the visual representation for underlying datasets, and utilize that information to draw polygons to encompass well-formed clusters. Each polygon, along with its corresponding projection plane and associated attributes (or dimensions), will be saved as a classification rule, called a PolyRule, for later prediction tasks. Experimental evaluation shows that PolyCluster is a visual-based approach that offers numerous improvements over previous visual-based techniques. It also can help users to obtain additional knowledge from current datasets.

1. Introduction

We are living in the age of digital information where all kinds of electronic data are growing rapidly without end in sight. However, the ability to analyze and extract useful knowledge from multifarious datasets lags far behind the capacity to create and store the data. It is critical to develop new data mining techniques and tools to help knowledge experts extract novel and useful information from raw data.

As a burgeoning technique, visualization has received much attention recently, as shown in [1], [2], and [12]. Visualization may assist in the identification of structures, features, patterns, and relationships in datasets by illustrating the underlying data in various kinds of graphical figures. Most classification systems are not integrated with human intervention, but an emerging trend is to focus more on this important feedback mechanism. As an example, Fayyad et al. [7]

noticed the importance of integrating user interaction and data visualization into the whole knowledge discovery process in order to help find understandable patterns for humans. Recently, Ankerst et al. [2] proposed strategies to assimilate human domain knowledge and integrate data visualization and user interaction into a knowledge discovery procedure. They identified three important reasons for including human domain knowledge:

- With help from data visualization, the human capacities to find useful patterns can be greatly improved.
- The users will have more confidence in the trust that they place in the created patterns generated from this interactive process.
- Domain knowledge of users can steer the data mining process. In addition, visualization techniques act as a complement to the data mining procedure. Domain knowledge can assist in deciding the appropriate data mining technique to use, and the appropriate subsets of the data to be considered [6].

Visualization techniques have been applied predominantly to datasets that are multidimensional, where each record in the dataset has more than three distinct attributes. Due to the limitation of our living space, we can only imagine and illustrate datasets, which are less than or equal to three dimensions. In order to resolve this obstacle, many methods have been invented to represent multidimensional data on a two-dimensional computer screen (e.g., scatter plots, parallel coordinates [11], star coordinates [18], and bar visualization [2]). The process for converting multidimensional data into a two-dimensional space is called data projection.

Alternatively, decision tree is a robust and effective classification technique that has been extensively

adopted by several visual-based classification systems [1], [2], [18] and [17]. A representative visual classification system is PBC (Perception-Based Classifier), which is based on circle segments [1]. PBC provides an interactive decision tree construction mechanism to help users build classification models. More recently, Teoh et al [17] proposed PaintClass, a new interactive visual classification technique that can classify both categorical and numeric data. PaintClass allows users to visualize multidimensional data by projecting each record to a color line on a two-dimensional display space using parallel coordinates [11]. The parallel coordinate system is a widely-used method for visualizing multidimensional data. For n -dimensional data there are “ n ” equally spaced parallel axes (horizontal or vertical), and each data point is rendered as a line that crosses each axis at a position proportional to its value for that dimension. Additionally, PaintClass incorporates a new decision tree exploration method to give users understanding of the decision tree in addition to finding additional knowledge from underlying data.

In this paper, an interactive classification rules construction and exploration system is introduced, called PolyCluster. The motivation for PolyCluster is taken from several existing popular visual-based visualization systems [1],[2],[9],[18]and[17]. PolyCluster offers several unique features as novel contributions:

- PolyCluster uses classification rules as its classification mechanism. Classification rules are similar to decision trees and are a viable alternative. Compared to decision trees, classification rules have many advantages that will be enumerated in a later section.
- PolyCluster introduces a new classification construction mechanism that can help users build a classification model interactively, as well as visualize records in multi-dimensional spaces.

These features enable PolyCluster to be an effective and efficient data mining solution. PolyCluster applies classification rules to finding structure, identifying patterns, and pinpointing relationships via multidimensional data visualization techniques. This framework is a major contribution of PolyCluster.

The rest of this paper is organized as follows. Section 2 reviews the visualization model of class-preserving projections. The classification rules (PolyRules) construction process is introduced in Section 3. In Section 4, quality measurements of clusters are described and the analysis of PolyRules applied to Iris and segment datasets is documented.

Finally, future work and a concluding summary are offered in the last two sections of the paper.

2. Class-preserving projections

Class-preserving projections were first introduced by Dhillon et al. [5] as a method for visualizing multidimensional clusters, centroids, and outliers by projecting multidimensional data onto a two-dimensional plane that can then be illustrated on a visual display device. This projection method can maintain the multidimensional cluster distribution and class structures, and are quite similar to Fisher’s linear discriminants [8].

When projecting data from multidimensional space to two-dimensional space, useful information is inevitably lost. This disadvantage can be alleviated by deliberately choosing an optimal two-dimensional plane of projection that could preserve original information as much as possible. In the clustering process, the intra-class distance should always be minimized, but the inter-class distance should be maximized. A key goal is to find those planes that can best preserve inter-class distance.

In the canonical case, the data is classified into three classes, i.e. C_1, C_2 and C_3 . Let $x_1, x_2, x_3, \dots, x_n$ be the d -dimensional data points where each point belongs to one of the three classes. The class means m_j are given by the following formula:

$$m_j = \frac{1}{n_j} \sum_{x_i \in C_j} x_i \quad j=1,2,3, \text{ where } n_j \text{ is the number of data}$$

points in class C_j (i.e., mathematically represented

$$\text{as } n_j = |C_j|).$$

For the purpose of visualization, a two-dimensional plane should be ascertained onto which all data points can be projected. The position of the candidate projection plane can be determined by a pair of orthogonal unit vectors on the plane. Generally, an orthonormal basis of the candidate projection plane can be chosen as these two vectors. Given an orthonormal basis $\omega_1, \omega_2 \in R^d$, where R^d is d -dimensional real space, a point x gets projected to the pair $(\omega_1^T x, \omega_2^T x)$, and likewise, each class mean m_j is mapped to $(\omega_1^T m_j, \omega_2^T m_j)$, where $j=1,2,3$.

In order to obtain superior contour of the projected classes, the distance between the projected class means (centroids) should be maximized. This can be accomplished by choosing vectors ω_1, ω_2 such that the objective objective functions, as the following,



Figure 1. (a) An optimal projection of the Iris dataset. (b) A poor projection of the Iris dataset.

can be maximized.

$$Q(\omega_1, \omega_2) = \sum_{i=1}^2 \sum_{j=2}^3 \sum_{k=1}^{j-1} \left\{ \left| \omega_i^T (m_j - m_k) \right|^2 \right\}.$$

Given A is a column vector and m_j & m_k are class means, the alternative expression for objective function can be rewritten as:

$$\begin{aligned} Q(\omega_1, \omega_2) &= \sum_{i=1}^2 \left\{ \omega_i^T \{ (m_2 - m_1)(m_2 - m_1)^T \right. \\ &\quad + (m_3 - m_1)(m_3 - m_1)^T \\ &\quad \left. + (m_3 - m_2)(m_3 - m_2)^T \right\} \omega_i \\ &= \omega_1^T S_B \omega_1 + \omega_2^T S_B \omega_2 \\ &= \text{trace}(W^T S_B W), \end{aligned}$$

where

$$W = [w_1, w_2], \quad \omega_i^T \omega_2 = 0, \quad \omega_i^T \omega_i = 1, \quad i = 1, 2,$$

$$\begin{aligned} S_B &= (m_2 - m_1)(m_2 - m_1)^T \\ &\quad + (m_3 - m_1)(m_3 - m_1)^T \\ &\quad + (m_3 - m_2)(m_3 - m_2)^T. \end{aligned}$$

The positive semi-definite matrix S_B can be explained as the inter-class scatter matrix. Due to $m_3 - m_2 \in \text{span}\{m_3 - m_1, m_2 - m_1\}$, scatter matrix S_B has $\text{rank} \leq 2$. Searching for the maximizing ω_1 and ω_2 can be limited to the column space of S_B . In general, the vectors $m_2 - m_1$ and $m_3 - m_1$ determine the optimal ω_1 and ω_2 which in turn compose an orthonormal basis of the visualization plane of projection. However, if class means m_1, m_2 and m_3 are collinear, then scatter matrix S_B degenerates to be of rank one and ω_1 should be in the direction of $m_2 - m_1$ while any unit vector, which is orthonormal to ω_1 , can be chosen as ω_2 . It is easy to understand that geometrically, the optimal projection plane, determined by orthonormal vectors ω_1 and ω_2 , is parallel to the plane that contains the three class means m_1, m_2 and m_3 .

For computation purpose, two eigenvectors (or

principal components) can be used to correspond to the two largest eigenvalues of S_B as vectors ω_1 and ω_2 , which uniquely determines the position of a projection plane. The interested reader can look at [14] for discussion on how to compute eigenvectors and eigenvalues. Note that projecting class means to this optimal plane can precisely preserve the distances between the class means. In other words, the distances between the projected class means are exactly equal to the corresponding distances in the original d-dimensional space. Thus, this kind of projection is defined as a class-preserving projection [5]. In Figure 1(a), an illustration is given of an optimal class-preserving projection on the Iris plant dataset [8]. In Figure 1(a), it should be noted that the three classes have their own clear contour and each class is well separated from the other two classes. However, if the chosen plane can not accomplish a class-preserving projection, induced class projections may be intermingled with each other. For instance, in Figure 1(b), the distance between Iris versicolor and Iris virginica is lost and the contour of each class is difficult to distinguish. Thus, the desire is to find an optimal class-preserving plane on which data can be illustrated.

As mentioned before, in the canonical case, the projection distance among three class means are exactly equal to the corresponding distances in the original d-dimensional space if the plane is chosen optimally. However, in most practical situations, it is more likely to find datasets that have more than three classes. The question is: ‘Can we examine the interplay among these classes?’ Fortunately, the answer can be given affirmatively. Similarly, a generalized method, which is similar to the method mentioned above, to differentiate more than three classes in the same view is presented in [5]. However, if projecting more than three class means in a plane, the distances between those class means will probably not be preserved faithfully.

Alternatively, the other solution is to examine all classes three at a time by using the methods mentioned above. Thus, there are a total of C_3^k such 2-dimensional projections, where k is the number of classes and each projection is determined absolutely by three class means. Considering the real datasets, the number of classes each dataset has is not too large, so the algorithm complexity of this method is acceptable. In PolyCluster, the boundary and contour of classes are found by exploring the projected class distribution figures. Thus, the latter option was chosen for PolyCluster because it can be applied to create much clearer regions of classes.

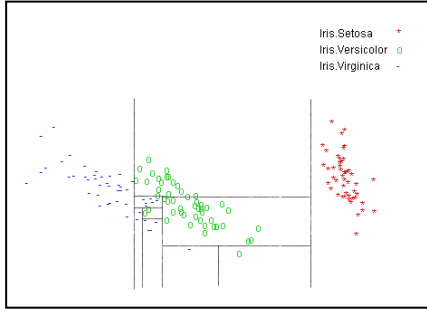


Figure 2. Tree generated by ID3

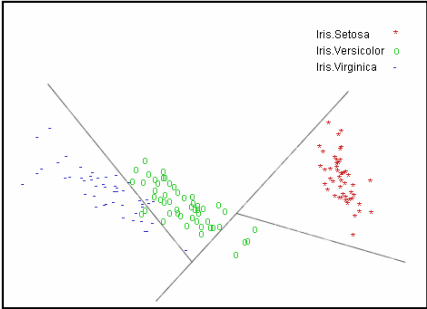


Figure 3. Tree generated by SADT

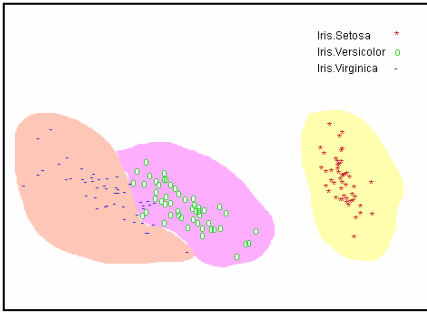


Figure 4. Tree generated by StarClass

However, there still exists a problem in this method. Note that if a given dataset has less than three classes, it is not possible to obtain three class means m_1, m_2 and m_3 to compute the position of projection plane. To solve this problem, the process can be initiated by clustering the given dataset and applying a state-of-the-art clustering algorithm to group records according to similarity. Then, it is feasible to use those centroids of generated clusters to compute optimal projection planes. Meanwhile, even for datasets with more than three classes, it is still possible to adopt this solution to provide more options and flexibility to find planes of class-preserving projection.

3. Polyrules construction

The major function of a classifier is to assign a class label to the newly created unlabeled objects. Training

data is the data on which the classifier was built. Meanwhile, the unlabelled data, or the data whose class label is unknown to the classifier, is called testing data. Compared with decision trees, each classification rule represents a nugget of knowledge. New rules can be integrated into an existing rule system without destroying those already discovered, whereas to add a new branch to an existing tree structure, decision trees may need to reorganize the whole tree. This might be the major reason why classification rules are popular. In PolyCluster, classification rules are created, called PolyRules, as a classifier.

3.1. Identify class regions

After visualizing training data in PolyCluster with the method in Section 2, we construct a classifier by generating a group of PolyRules where each rule is composed of a pair of orthonormal vector ω_1 and ω_2 , which determine the position of projection plane, position of a polygon (drawn by users), which identify a certain class region, and used attributes (or dimensions). For each class region, one or several polygons can be drawn to identify the class contour such that in ideal cases each polygon encircles points exclusively belonging to the same class label.

There exist several methods to identify the class regions. For example, in Figure 2, ID3 [16] partitions a set of points with axis-parallel hyperplanes. However, due to axis-parallel limitation, ID3 generally creates a decision tree that is much larger and less accurate than the SADT tree [10] in Figure 3, which is created with oblique lines. However, decision trees generated by oblique lines easily include outliers and noise. Another method to identify class regions is introduced by StarClass [18], which uses a brush to paint the regions in Figure 4. There are two disadvantages to this method. First, if the class region is very large, the whole painting process will be very boring and time consuming because users have to brush all areas within that region. Second, after painting the original image with a different color for identifying the class region, the new color will blur the underlying color points. By contrast, PolyCluster, which is more convenient and efficient compared with the above three existing approaches for identifying the class regions, uses polygons to mark the class regions. PolyCluster allows users to adjust the polygon contour by using the mouse to drag the polygon corner to a different location. If users are not satisfied with the created polygons, they can use the provided operation on a pop-up menu to delete selected polygons.

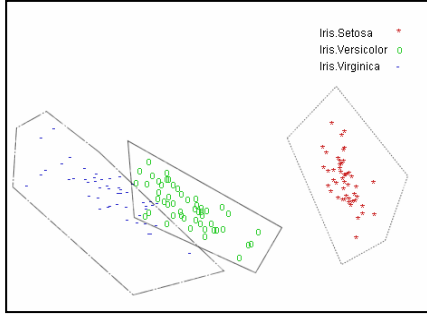


Figure 5. Based on a selected projection plane, the user draws three polygon areas where each polygon illustrates a class contour.

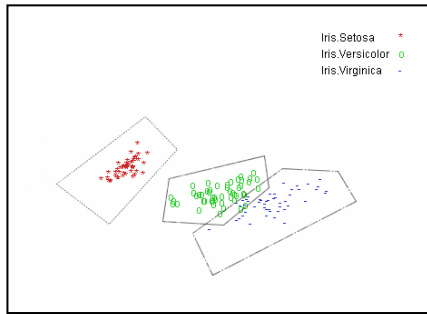


Figure 6. After projecting dataset onto a different plane, the user draws the second group of polygons to identify the area of classes. Each of generated polygons combined with the position of used projection plane and attributes composes a PolyRule for classification purpose.

3.2. Polyrules construction and classification

PolyCluster is founded on the following assumption: Given a region ψ , which includes mostly objects belonging to a certain class σ , any new coming object, without class label, which is mapped on region ψ , is likely to be assigned to class σ . As mentioned in section 3.1, a PolyRule is composed of a pair of orthonormal vectors ω_1 and ω_2 determining the position of a projection plane, position of a drawn polygon and used attributes. If the current projection figure cannot differentiate the class boundary clearly, it is possible to redo this procedure and re-project data onto different projection planes for creating more effective PolyRules. Based on different projection planes, two groups of polygons are drawn in Figure 5 and 6.

In PolyCluster, two kinds of PolyRules are defined: primary and associate. A primary rule is the rule that

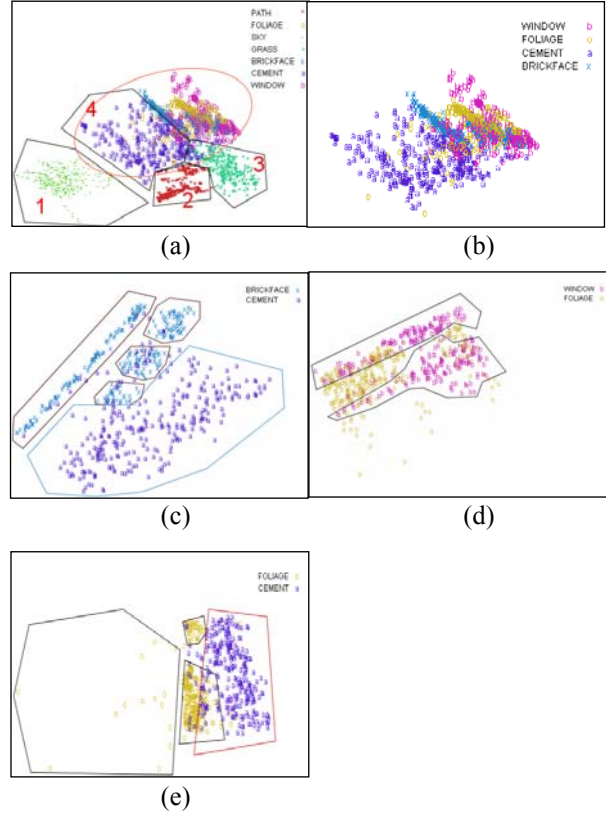


Figure 7. Illustration of the Statlog Segment dataset. In (a), each polygon of 1, 2 and 3 combined with the position of the used projection plane and attributes represents a primary PolyRule. Although Polygon 4 can only compose an associate PolyRule because of its poor classification performance. The intermingled area, marked by a red ellipse in (a) image, is filtered out and reprojected onto the same plane without samples belonging to class PATH, SKY and GRASS and shown in (b) image. The images (c), (d) and (e) represent the conditional projections which resolve only part of intermingled samples by choosing subset of classes and reprojecting data samples onto different planes.

can perform a classification task with high precision beyond a certain predefined threshold in the training dataset. PolyCluster normally chooses 95% as the threshold. Alternatively, if a rule performs a classification task with low precision compared with that threshold, then it belongs to associate rules. For instance, in the (a) of Figure 7, PolyRules created by polygons 1, 2 and 3 are primary rules, but those created by polygon 4 are associate rules. Given a newly arrived unlabeled data, if there does not exist a classification contradiction among all primary rules, the new data is assigned the class label by the applied

PolyRule. However, if a rule set gives multiple classifications for a testing sample, the majority vote method can be used. For example, if 5 classification rules assign class label “A” to a testing sample \check{D} and 3 rules assign class label “B” to \check{D} , then the sample \check{D} should choose “A” as its class label. On the other hand, we can choose the rule with the maximum coverage for classifying the new data sample. Finally, if a tie still remains, one class label can be randomly selected for the new data sample. Meanwhile, there is another major difference between primary rules and associate rules. Primary rules can be applied independently if they don’t introduce contradiction into classification results. However, associate rules must be used with other rules, and different groups of associate rules have to be applied in a certain sequence that can be called its priority order.

Generally, for a complicated dataset, one optimal projection can not find clear boundaries for all classes. In such circumstance, intermingled areas should be filtered and re-projected onto different projection planes. For instance, in (a) of Figure 7, the area marked by the red ellipse is re-projected onto different projection planes with a different subset of class. The split process is repeated until users are satisfied with the induced results, or no more improvement can be made by continuing this procedure.

4. Empirical evaluations

The objectives of PolyCluster are: (1) to create a user-directed, interactive, and visual-based classification system, (2) to integrate the powerful human pattern recognition capabilities into a classification system, and (3) to enable users to trust the created patterns from this interaction process.

4.1. Measurement for quality of clusters

After obtaining induced clusters from the interactive procedure, a method is needed to evaluate the quality of clusters. In [17], Song et al. propose a compactness measurement for evaluating the quality of a cluster by computing the ratio of external connecting distance (ECD) and internal connecting distance (ICD). However, a goal is to evaluate the overall quality of clusters and place greater emphasis on the size of clusters. Thus, an extended model of ECD and ICD was designed to evaluate the quality of clusters by computing the ratio of average external connecting distance (AECD) and average internal connecting distance (AICD) which are introduced below.

A graph G consists of a finite set V of points where $V = \sum_{i=1}^n C_i$, $C_i \subseteq V$, $C_i \cap C_j = \emptyset$ if $i \neq j$ and each C_i is

a cluster in graph G , a finite set E of edges, and a function γ that assigns a positive number to a pair of points (p, q) as the distance between point p and q . Let $Dist_{ij} = \{\gamma(p, q) \mid (p, q) \in E, p \in C_i, q \in C_j, i \neq j\}$. The

$ECD(C_i, G, \gamma)$ is defined as $Min(Dist_{ij})$ where $i \neq j, i \& j = 1 \dots n$, and n is the number of generated clusters in current dataset. Let $Dist_i = \{l \in R^+ \mid p, q \in C_i, p \neq q, \gamma(p, q) \leq l\}$. The

$ICD(C_i, G, \gamma)$ is defined as $Min(Dist_i)$ and ICD is also named as the diameter of cluster C_i . The AECD and AICD of Graph G is given by

$$AECD(G, \gamma) = \sqrt{\frac{1}{n-1} \times \sum_{i=1}^n N_i \times N_j \times ECD(C_i, G, \gamma)}$$

$$AICD(G, \gamma) = \sqrt{\frac{1}{n-1} \times \sum_{i=1}^n N_i \times ICD(C_i, G, \gamma)}$$

where N_i is the number of points in cluster C_i and N_j is the number of points in the nearest neighbor cluster of cluster C_i . $\frac{AECD(G, \gamma)}{AICD(G, \gamma)}$ is used as the measurement to

evaluate quality of generated clusters for given dataset. Users can use this measurement as a criterion to choose a best clustering combination from several available options.

4.2. Experimental setup

The PolyCluster system is implemented in Java and compiled with Java JDK 1.4.2. An experimental evaluation of PolyCluster on two well-known benchmark datasets (Iris and Segment) was performed. Fisher’s Iris dataset is one of the most famous datasets used in data mining. It contains fifty records each of three types of plant: Iris setosa, Iris versicolor, and Iris virginica where each record has four distinct attributes: sepal length, sepal width, petal length and petal width. All attributes have values that are numeric. Meanwhile, records in the segment dataset are drawn randomly from a database of 7 outdoor images. The images are hand-segmented to create a classification for every pixel. It contains 36 numerical attributes and 2,310 records where each record belongs to one of 7 distinct classes. All experiments were performed on a Pentium IV 2.53GHz Dell Desktop with 512 MB main memory.

Table 1. Datasets descriptions

Dataset	Size	Classes	Attributes
Iris	150	3	4
Segment	2310	7	19

Table 2. Accuracy of PolyCluster compared with C4.5, PBC and PaintClass

	C4.5	PBC	PaintClass	PolyCluster
Iris	94.6%	n/a	n/a	94.0%
Segment	95.9%	94.8%	95.2%	94.8%

4.3. Error measurement

In PolyCluster, sum-minority error measurement was used to evaluate the experimental results. Consider a set of samples S , belonging to 3 distinct classes each of them has $|S_1|, |S_2|, |S_3|$ samples respectively

where $|S| = \sum_{i=1}^3 |S_i|$ and $S_i \cap S_j = \emptyset, i \neq j, i \& j = 1, 2, 3$. The

class C_i that appears most often in set S is found and chosen as majority category. If set S has most samples in its majority category, then it is relatively pure. It is always desirable to draw polygons that can encompass data samples to form pure subsets (or clusters). To guide users to generate relatively pure splits, the sum-minority error measure is defined to be $|S| - \text{Max}(|S_i|)$.

4.4. Experimental results

The famous Statlog database [13] has been extensively used as a benchmark for evaluating numerous data mining classification tools. Iris and Segment [13] have been chosen as experimental datasets to evaluate PolyCluster. The description for these two datasets is illustrated in Table 1. The accuracy of the decision tree algorithm C4.5 is from [15], and visual-based classifiers PBC and PaintClass from [2] and [17], respectively.

There are two common methods for evaluating the accuracy of a classifier. One is the holdout method, which reserves a certain amount of data for testing and uses the remainder for training. In practical circumstances, it is common to hold one-third of the data for testing and use the remaining two-thirds for training. However, this method is not applicable if the given dataset is too small, because the generated training dataset will contain inadequate data for building the classifier. In this situation, a cross validation method can be applied. All except one data subset are used to construct the classifier and the left

one is used as testing dataset. Generally, the standard way of predicting the error rate of a learning technique given a single, fixed sample of data is to use stratified tenfold cross-validation. When evaluating PolyCluster, holdout method was used for the Segment dataset, and the ten-folder cross validation was used for the Iris dataset.

Table 2 illustrates the results of performing a classification using PolyCluster compared with C4.5, PBC and PaintClass. From the experimental results, we can find PolyCluster perform quite well compared with those state-of-the-art classifiers. Because the classification rules are built by users manually rather than automatically built by underlying algorithms, the precision of the PolyCluster is quite dependent on the pattern-perception capabilities of humans. In particular, it seems that PolyCluster can obtain the same accuracy as that of the famous visual classifier PBC.

Compared to traditional non-visual based classifiers, visual-based classifiers can help users obtain extra knowledge for current datasets and improve users' trust into the induced classification results. With the techniques of class-preserving projections of the multidimensional data onto two-dimensional planes, PolyCluster can give users a clear overview of the space distribution of high-dimensional data and the relationship among different classes. For example, in Figure 5, the class Iris Setosa appears well-separated from the other two classes, but there is a tiny overlap between the boundary of the class Iris Versicolor and Iris Virginica. As another example, in (a) of Figure 7, the class PATH, SKY and GRASS can be found to have obvious boundaries, whereas the class FOLIAGE, BRICKFACE and WINDOW intermingle with each other.

Note that visualization is not a substitute for quantitative analysis; however, it is a very good complementary tool to understand fully the underlying dataset. It is a viable alternative to the application of traditional tools, such as C4.5 and CART [3]. Although experimental results from PolyCluster are not superior to that of some state-of-the-art classification techniques, PolyCluster demonstrates a different approach to explore and classify multi-dimensional datasets in an effective and efficient way.

5. Future work

Although experimental results have shown that PolyCluster is an effective classification system, it is desirable to improve the capacities of PolyCluster by integrating facets of other state-of-the-art visual-based

classifiers. For example, just as PBC [1] has included several levels of cooperation between human and computer (i.e. automatic, automatic-manual, manual-automatic and manual), PolyCluster can be extended to support those options in cases where users can not apply PolyCluster to generate an effective classifier. Likewise, PolyCluster can be applied to other datasets, such as DNA sequences and microarray datasets, to extend its application areas. Finally, PolyCluster should be extended to handle both numerical and categorical attributes [4].

6. Conclusions

The paper introduces PolyCluster, an interactive multidimensional data visualization and classification tool. In PolyCluster, users can obtain the visual representation for underlying datasets, utilize that information to draw polygons to encompass well-formed clusters, and then build a classifier interactively. Experimental results have shown that PolyCluster is an effective and efficient approach to find structures, features, patterns, and relationships in underlying datasets. In addition, PolyCluster integrates a pair of novel and robust measurements, called AECD and AICD which users can adopt as a criterion to choose a best clustering combination from several available options. With further improvement such as the integration of automatic algorithms to build classifiers and the capabilities to handle categorical attributes, PolyCluster can become an even more powerful visual-based classification system.

7. References

- [1] M. Ankerst, C. Elsen, M. Ester and H.-P. Kriegel, *Visual Classification: An Interactive Approach to Decision Tree Construction*, *Proceedings of the 5th ACM SIGKDD International Conference on Knowledge Discovery and Data mining*, 1999, pp. 392-396.
- [2] M. Ankerst, M. Ester and H.-P. Kriegel, *Towards an Effective Cooperation of the User and the Computer for Classification*, *Proceedings of the 6th ACM SIGKDD International Conference on Knowledge Discovery and Data mining*, Boston, MA, 2000.
- [3] L. Breiman, *Classification and Regression Trees*, Kluwer Academic Publishers, 1984.
- [4] D. Coppersmith, S. J. Hong and J. R. M. Hosking, *Partitioning Nominal Attributes in Decision Trees*, *Data Mining and Knowledge Discovery*, 3 (1999), pp. 197-217.
- [5] I. S. Dhillon, D. S. Modha and W. S. Spangler, *Visualizing Class Structure of Multidimensional Data*, *Proceedings of the 30th Symposium on the Interface: Computing Science and Statistics*, Interface Foundation of North America, Minneapolis, MN, 1998, pp. 488-493.
- [6] P. DOMINGOS, *The Role of Occam's Razor in Knowledge Discovery*, *Data Mining and Knowledge Discovery*, 3 (1999), pp. 409-425.
- [7] U. Fayyad, G. P. Shapiro and P. Smyth, *The KDD Process for Extracting Useful Knowledge from Volumes of Data*, *Communications of the ACM*, 39 (1996), pp. 27-34.
- [8] R. A. Fisher, *The Use of Multiple Measurements in Taxonomic Problems*, *Annals of Eugenics*, 7 (1936), pp. 179-188.
- [9] J. Han, L. V. S. Lakshmanan and R. T. Ng, *Constraint-Based Multidimensional Data Mining*, *IEEE Computer*, 32 (1999), pp. 46-50.
- [10] D. Heath, S. Kasif and S. Salzberg, *Induction of Oblique Decision Tree*, *Proceedings of the 13rd International Joint Conference on Artificial Intelligence*, Morgan Kaufmann, San Mateo, CA, 1993, pp. 1002-1007.
- [11] A. Inselberg, *The Plane with Parallel Coordinates*, *Special Issue on Computational Geometry: The Visual Computer*, 1 (1985), pp. 69-91.
- [12] E. Kandogan, *Visualizing Multi-dimensional Clusters, Trends, and Outliers Using Star Coordinates*, *Proceedings of the 7th ACM SIGKDD International Conference on Knowledge Discovery and Data mining*, San Francisco, California, 2001, pp. 107-116.
- [13] D. Michie, D. J. Spiegelhalter and C. C. Taylor, *Machine Learning, Neural and Statistical Classification*, Ellis Horwood, 1994.
- [14] W. H. Press, S. A. Teukolsky, W. T. Vetterling and B. P. Flannery, *Numerical Recipes in C++: The Art of Scientific Computing*, Cambridge University Press, 2002.
- [15] J. R. Quinlan, *C4.5: Programs for Machine Learning*, Morgan Kaufman, 1993.
- [16] J. R. Quinlan, *Induction of decision trees*, *Machine Learning*, 1 (1986), pp. 81-106.
- [17] S. T. Teoh and K.-L. Ma, *PaintingClass: Interactive Construction, Visualization and Exploration of Decision Trees*, *Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data mining*, 2003.
- [18] S. T. Teoh and K.-L. Ma, *StarClass: Interactive Visual Classification Using Star Coordinates*, *Proceedings of the 3rd SIAM International Conference on Data Mining*, 2003.