StarClass: Interactive Visual Classification Using Star Coordinates

Soon Tee Teoh* Kwan-Liu Ma*

Abstract

Classification operations in a data-mining task are often performed using decision trees. The visual-based approach to decision tree construction has gained increasing popularity. We present StarClass, a new interactive visual classification method. This method maps multi-dimensional data to the visual display using star coordinates, allowing the user to interact with the display to create a decision tree. Preliminary evaluation indicates that this new technique is as effective as state-of-the-art algorithmic classification methods, and more effective than the previous visual-based methods. StarClass also offers additional advantages such as improving the user's understanding of the data.

Keywords : visual data mining, classification, decision trees, information visualization, interactive visualization

1 Introduction.

Classification [12] of multi-dimensional data is one of the major tasks in data mining. In the classification problem, there is a set of classes that objects can belong to. Based on an object's attribute values, a classification method assigns the object to one class among the set of classes. Typically, a classification system is first trained with a set of data whose attribute values and classes are known. Once the system has built a model based on the training, it is used to assign a class to each object whose class is as yet unknown.

Decision trees have been well-established as an effective tool for classification. A decision tree is constructed by repeatedly partitioning the dataset into disjoint subsets. One class is assigned to each leaf of the decision tree. Reasons for the popularity of decision trees include fast construction [15], simplicity [1] and easy conversion to SQL statements [14].

Most classification systems are designed for minimal user intervention. However, recently, Ankerst et al. [3] argue for increased user involvement in the classification process, with three important reasons: (1) the use of powerful human pattern recognition capabilities in decision tree construction, (2) the increase in confidence, and (3) the possibility of incorporating domain knowledge to the algorithm. In addition, visualization of the data also improves the user's understanding of the data [2, 6]. For example, the user can see whether certain classes are clearly defined whereas other classes have much overlap with one another, and whether objects in a certain class spread over large spaces or form small clusters.

A representative visual classification system is Ankerst et al.'s PBC (Perception-Based Classifier) [2] based on Circle Segments. One limitation of PBC is that partitioning of each node in the tree is based only on one attribute at a time, whereas some class boundaries may be a function of more than one attribute. More recently, Wang et al. [18] proposed methods for visualizing nearest neighbor, decision tree and ensemble classification.

In terms of accuracy, the current visual classification systems are still slightly inferior to the best algorithmic methods. There is potential for better visual representations and better interactive methods to provide the user even better understanding of the data than current visual methods can.

Therefore, we propose StarClass, a new interactive visual classification technique, characterized by simplicity, intuitiveness and effectiveness. StarClass allows users to visualize multi-dimensional data by projecting each data object to a point on 2-D display space using Star Coordinates [9]. When a satisfactory projection has been found, the user then partitions the display into disjoint regions; each region becomes a node on the decision tree. This process is repeated for each node in the tree until the user is satisfied with the tree and wishes to perform no more partitioning. This decision tree construction process is explained in more detail in Section 3. We also discuss some visual and interaction features and techniques to aid the efficient construction of decision trees.

Algorithmic classification methods often require users to set parameters or threshold values. Depending of the values or models chosen by the user, very different class assignments can result. The selection of a good threshold value or model (such as making the Gaussian assumption) is often a very challenging problem because the user has little basis for making choices. Therefore, threshold values are often chosen in an arbitrary manner, often leading to problems like over-fitting, yielding sub-optimal results. StarClass tries to avoid this pitfall by giving the user valuable information regarding the underlying distribution of the data via visual inspection. This information is not expressed in terms of aggregated statistical values, but much richer, as we will show with examples throughout the paper. Human judgement is thus used to help achieve more accurate classification.

Experimental evaluation of StarClass is presented in Section 4. The same section also explains the contributions that StarClass makes in handling multi-variate class

*Department of Computer Science, University of California, Davis

boundaries, greater accuracy and improved user understanding compared to PBC. StarClass is intended as a foundation on which to build more advanced classification. Some ideas which we are currently exploring and developing are described in Section 5.

2 Star Coordinates.

Star Coordinates, similar to ShapeVis [17] and RadViz [7], was first introduced by Kandogan [9] as a method for visualizing multi-dimensional clusters, trends, and outliers. In this section, we give a brief description of Star Coordinates, and in Section 3, we explain how we utilize various features of Star Coordinates in StarClass, our classification system.

2.1 Star Coordinates projection and interaction. In Star Coordinates, each dimension is represented as an axis radiating from the center of a circle to the circumference of the circle. The values of each dimension are first normalized to the 0-1 range. As an intermediate step, a multi-dimensional object is mapped onto one point on each axis based on its value in the corresponding dimension. For example, if an object A has a normalized value v in dimension D and dimension D is represented in display space as an axis from (0,0) to (x,y), then the position of A on the axis representing D is given by (vx,vy). The final position of an object will be the average of all the projections onto individual axes.

To edit a projection, the user manually moves an axis around by clicking on the end-point of the selected axis and dragging it to the desired position. This interaction allows the user to scale or rotate an axis, as described in [9]. In addition, our implementation can automatically generate a new and different dimension-axis mapping. Figure 1 shows such changes applied to the projection of the *Satimage* training data from the Statlog [11] database. Each data object is shown as a point at the final position defined by the projection given above. Each point is colored according to its class.

The desirable features of Star Coordinates include the following: (1) the mapping is conceptually and computationally simple: it uses only simple averaging, (2) data objects with similar attribute values always map to nearby positions on the display (Data objects that map to nearby screen positions do not always have similar attribute values, but this is handled by the detail view feature described in Section 2.2.), (3) the user can edit the axes and change the projection very easily, and (4) the position of each data object can be influenced by all its attribute values or any combination of them according to the user's choice.

2.2 Detail view. While the displays in Figure 1 give a good overview of the data, it is clear that many objects map to the same pixel or nearby pixels. We allows the user to zoom into a small area of the display in order to obtain a more detailed view. Figure 2 shows zooming into a "crowded" area to resolve individual data objects.

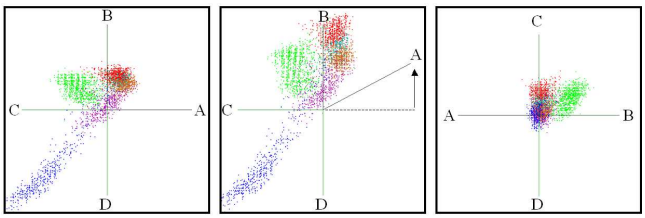

Figure 1: Left: Star Coordinates projection of the **Satimage** dataset. The four dimensions used are represented as axes labelled A through D. Each data object is projected to one point on the display. Middle: The user moves the A axis with the mouse. The projected points move accordingly. Right: A projection with a different assignment of the axis yields a significantly different picture.

Using a feature found in Star Coordiates [9] as well as ShapeVis [17], we draw objects in the detailed view not as simple points but with lines indicating the attribute values in each dimension. The orientation of each line is aligned with the direction of the axis whose attribute dimension it represents. The length of each line of an object is directly proportional to the normalized attribute value of the corresponding dimension. The projection of high-dimensional objects into 2-D space may cause objects far apart in high-dimension space to have close proximity in 2-D space. The drawing of lines for each object serves to differentiate these objects.

3 Decision tree construction.

The main idea behind the decision tree construction method is to identify regions in the projected 2-D space such that ideally each region would contain only projected points belonging to one class. The user thus begins by exploring an overview of the entire training dataset, attempting to find a projection which separates objects belonging to different classes clearly into different regions in the 2-D display. The decision tree construction algorithm is built on this assumption: Given a region R containing mostly objects belonging to a certain class C, any as yet unclassified data object that projects onto R is likely to belong to class C.

3.1 Specifying regions. When the user is satisfied with a projection, the user specifies regions by "painting" over the display with the mouse icon (see Figure 3). StarClass allows the user to control the size of the paint-brush for more efficient user interaction. An "Eraser" operation is also provided. Each region is shown with a background with a unique color.

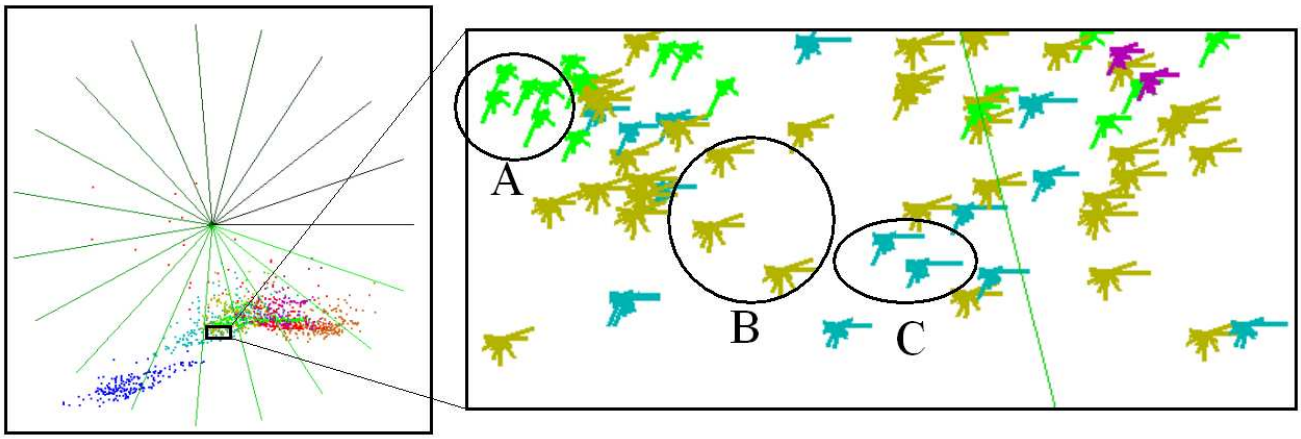

Figure 2: A zoom rectangle provides a detailed view of a subset of the **Segment** dataset. In the detailed view, lines (called "sticks") are drawn for each object. Each stick represents one dimension, with the direction aligned with the corresponding axis, and the length dependent on the attribute value in that dimension. In RadClass, a major purpose of the sticks is to distiguish between objects far apart in high-dimensions but mapping to nearby positions in 2-D. In this example, the objects in class A have a elongated stick in the 7-o'clock position, and objects in class B have a relatively long stick in the 2-o'clock position compared to objects in class C. It is thus visually obvious to the user that the three circled regions are different in high-dimensional space although they are projected to similar position in 2-D. This provides clues regarding how to move the coordinate axes so as to separate the objects of the different classes.

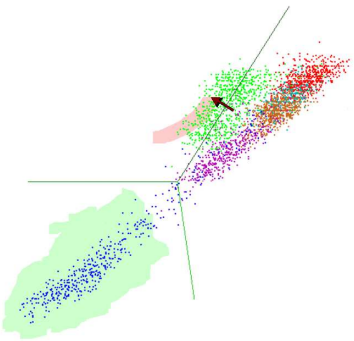

Figure 3: Painting regions. The user holds down a mouse button and drags the mouse to "paint" a region. The size of the "paint-brush" can be controlled. In this figure, a green region has been painted and the user is painting a red region.

3.2 Building the decision tree. Typically, for a dataset with a fair amount of complexity, one projection is not sufficient to completely resolve all the different classes. In this case, the user should define regions each having different characteristics. For each such region, the user can define a new "conditional projection" to further resolve only the subset of data objects which project onto that region. An example is shown in Figure 4. The user then paints regions on each conditional projection. The creation of regions and conditional projections is repeated until the user is satisfied that any further conditional projections will not significantly improve the decision tree model. Regions which do not have conditional projections thus constitute the leaves of the decision tree.

StarClass counts the number of objects beloning to each class mapping to each terminal region (i.e., the leaf of the decision tree). The class with the most number of objects mapping to a terminal region is elected as the region's "expected class". During classification, any object which finally projects to the region will be assigned that class.

3.3 Guidelines for improving accuracy. Achieving a high success rate in the classification task requires some skill on the part of the user. Nevertheless, if one follows the following guidelines, even a user with limited experience can perform very effective classification.

- When possible, find a projection where one class is clearly separated from the rest. If not, find a projection

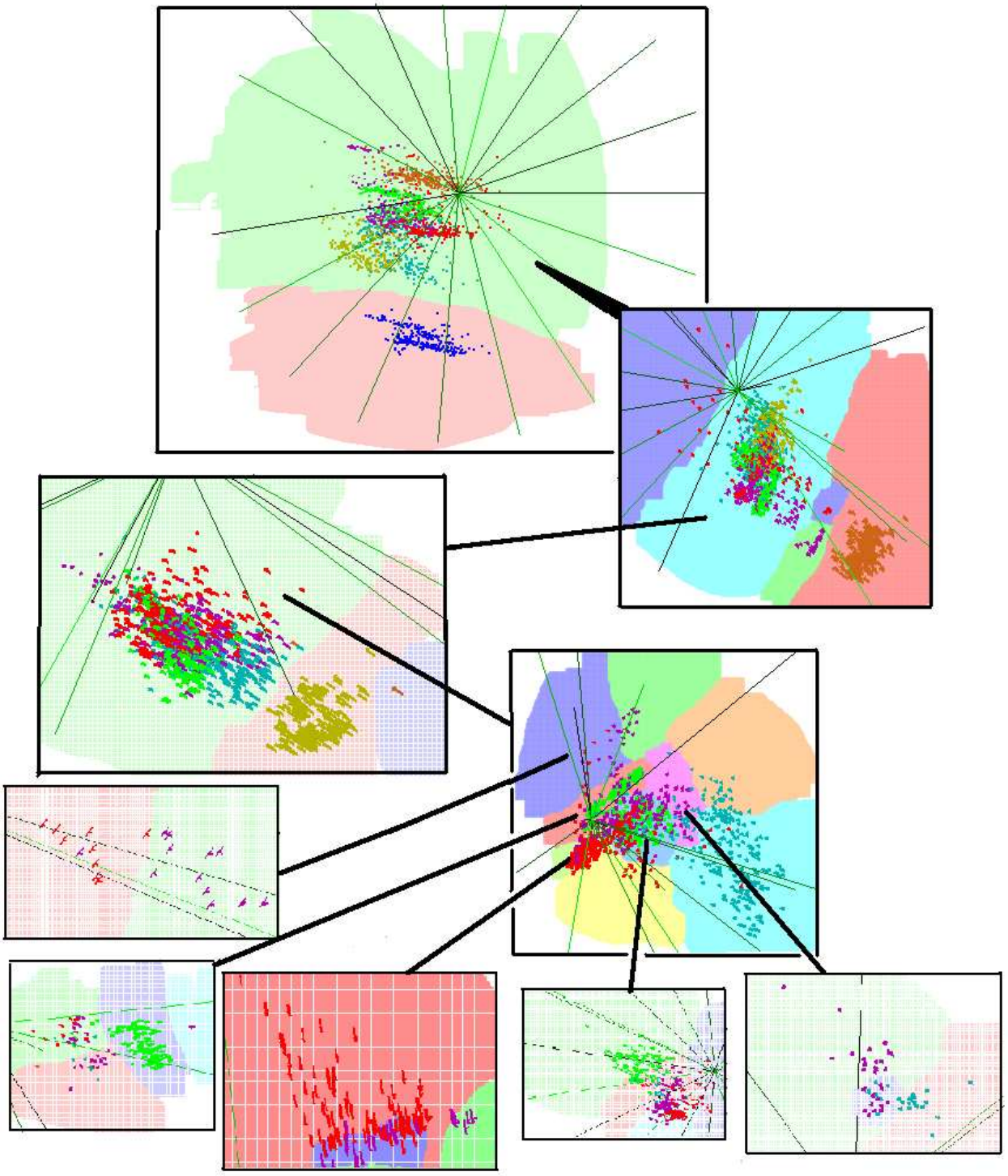

Figure 4: Top levels of the decision tree of the Statlog **Segment** dataset. The top box shows the projection of the entire dataset, and is therefore the root of the decision tree. It is split into two regions: green and red. The red region contains only objects of the blue class, so the user makes it a terminal region(leaf node). The green region is then subject to a different projection and further partitioned.

where classes are separated such that there are clear regions where certain classes are absent. In this way, regions can be defined so that they only contain objects belonging to a subset of all classes, simplifying the task for the next node.

- When defining regions, the user should make sure enough training data samples are present in each region. This is to avoid overfitting the data.

- In Section 2.2, we have described the drawing of data objects with lines (see Figure 2) in the detail view. The length of each line indicates the attribute values of the objects in the corresponding dimension. This helps the user decide how to adjust the projection to separate the objects. An effective way to separate two objects close together in screen space but with very different appearance in the zoom-view is to identify a dimension where their lines have different lengths, and move the axis corresponding to that dimension.

- The dragging of the coordinate axes with the mouse causes an immediate response in the position of the projected points. Based on the motion of the points, the user can get a good hint about where to position the axis.

4 Experimental Evaluation.

There are three motivating factors in the design of StarClass: (1) effectiveness in separating classes with a multi-variate boundary, (2) accuracy of classification, and (3) effectiveness in improving user understanding. We have performed an evaluation of StarClass by using it to classify actual data, and analyzing the process and result of the classification using those three factors as criteria.

4.1 Multi-variate class boundary. StarClass is designed with the goal of being particularly advantageous in separating classes with a multi-variate boundary. To test this claim, we created a dataset containing two classes separated by a linear function $2x + 3y - z - 1 = 0$. In a very short amount of time, the user is able to find a projection on which the two classes are separated clearly. This is presented in Figure 5, which also shows an example of separating two classes with a non-linear multi-variate boundary. Although this is only a dataset we artificially created, we believe that class boundaries in real multi-dimensional data are often a function of multiple variables. Unfortunately, functional expressions of class boundaries in real data are usually not known. However, the greater accuracy of StarClass compared to PBC, as presented in the next section, may be due to advantages in detecting multi-variate class boundaries.

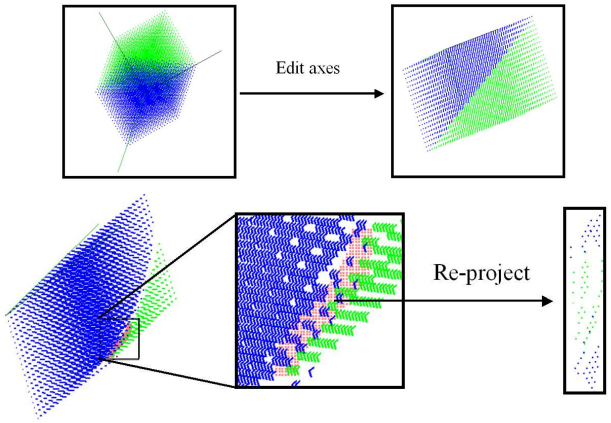

Figure 5: Top: Separating two classes with a linear multi-variate boundary. The green and blue classes are separated by the function $2x + 3y - z - 1 = 0$. By moving the axes around, the user finds a projection effectively separating the two classes. Unlike other methods such as PBC, Radclass is not limited to split-points on one dimension at a time. Bottom: Separating two classes with a non-linear multi-variate boundary $2x^2 + 3xy - z - 2 = 0$ requires the re-projection of a small region of overlap in the first projection.

4.2 Accuracy. The Statlog database [11] has been used as a benchmark in the evaluation of many previous classification methods. We choose the *Satimage*, *Segment* and *Shuttle* datasets from the database for comparison because they contain objects with only numerical attributes. The description of these datasets is presented in Table 1. The accuracy of popular algorithmic decision tree classifiers CART and C4 from the IND package [13], as well as SPRINT [15] and CLOUDS [4], and also visual classifier PBC, on these datasets is known. Table 2 shows the results of performing classification using StarClass compared with the above-mentioned methods.

From the experimental results, StarClass performs well compared with the other methods. In particular, it appears that StarClass is slightly more accurate than PBC. Since the classification task is performed by the user in StarClass, the accuracy is highly dependent on the skill and patience of the user, and even the same user can introduce some small variations in accuracy for different attempts.

We have also run StarClass on the *diabetes* dataset [16] to test the effectiveness of StarClass on noisy data. In this dataset, there are 768 data objects with 9 numerical attribute values each. Each object belongs to one of two classes: (1) tested positive for diabetes, or (2) tested negative for diabetes. We randomly partitioned the objects into three groups of 256. We pick one group as testing data and use the other two groups as training data. We repeat this for each of the groups, and take the average accuracy achieved. A

Table 1: Description of the datasets.

Dataset	Size	Num classes	Num dimensions
Satimage	4435	6	4
Segment	2310	7	19
Shuttle	43500	7	9

Table 2: Accuracy of StarClass compared with algorithmic approaches and visual approach PBC.

	Algorithmic				- Visual	
	CART	C4	SPRINT	CLOUDS	PBC	**StarClass**
Satimage	85.3	85.2	86.3	85.9	83.5	**85.3**
Segment	84.9	95.9	94.6	94.7	94.8	**95.2**
Shuttle	99.9	99.9	99.9	99.9	99.9	**99.9**

resulting decision tree is shown in Figure 6. The accuracy results are shown in Table 3. The other approaches used for comparison are CBA [10], C4, FID [8], and Fuzzy [5]. The accuracy figures for these methods are taken from [5]. From the accuracy rates, StarClass performs better than all the methods except Fuzzy, which is designed specifically for classifying such noisy data.

4.3 Improved user understanding. One of the major advantages of visual classification compared to algorithmic classification is that during the classification process, as the user interacts with the visual representation of the data, the user gains a much better understanding of the data. This important point was made and demonstrated in [2]. However, we believe that with the visualization method used in StarClass, user understanding of the data can be even further improved.

The major limitation of the Circle Segments visualization method used in PBC is that a data object is not viewed as one single entity, but is separated into its individual attribute values. With Star Coordinates, the position of a data object can be influenced by some or all of its attribute values, thereby giving the user a more complete impression. Because of that, important insights into the data can be gained, such as these examples from the three test cases we used for studying accuracy:

- In the overview projection of the *Satimage* dataset shown in Figure 3, the blue class appears well-separated from the rest, and diagonally there is a change from the purple to the brown to the grey and finally to the red class, with significant overlap between adjacent classes. The green class is off to the side, with some overlap with four other classes.

- In the *Segment* dataset shown in Figure 4, the blue, brown and olive green classes have clear boundaries whereas the red class contains objects with wide range of attribute values.

- One of the interesting features of the *Shuttle* dataset shown in Figure 7 is the existence of different clusters within classes, for example two clusters belonging to the red class are boxed and two clusters belonging to the purple class are circled.

5 Future work.

Although experimental evaluation has indicated the effectiveness of StarClass, we are currently working on a number of areas where we believe StarClass can be further improved. Just as Ankerst et al. have incorporated algorithmic support into PBC with significant success [3], we are also extending StarClass to include options for using automatic algorithms in cases where Star Coordinates has limited effectiveness in clearly separating objects of different classes. We believe that with some algorithmic support, the accuracy of StarClass can be further improved.

We would like to apply StarClass to other datasets to further investigate its effectiveness. We would also like to conduct an extensive user study to find out how effectively different users perform the classification task with this tool.

Finally, we plan to extend StarClass to handle very large datasets. This could be done for example by random sampling. It would also be desirable to expand the scope of StarClass to handle data with not just numerical attributes but also categorical ones.

6 Conclusions.

We have presented StarClass, a decision tree construction method based on interactive multidimensional visualization. In StarClass, the user interacts with Star Coordinates to find appropriate projections that best separate objects of different classes. A major advantage of Star Coordinates is that it is flexible enough to visually separate two classes with a boundary dependent on multiple variables.

Visual classification facilitates the use of domain knowledge to guide the classification process. It is not always convenient, or even possible, to program this domain knowledge

Table 3: Accuracy of StarClass for classifying the **diabetes** dataset compared with other classifi cation methods.

CBA	C4	FID	Fuzzy	**StarClass**
74.4	73.8	62.0	77.6	**74.6**

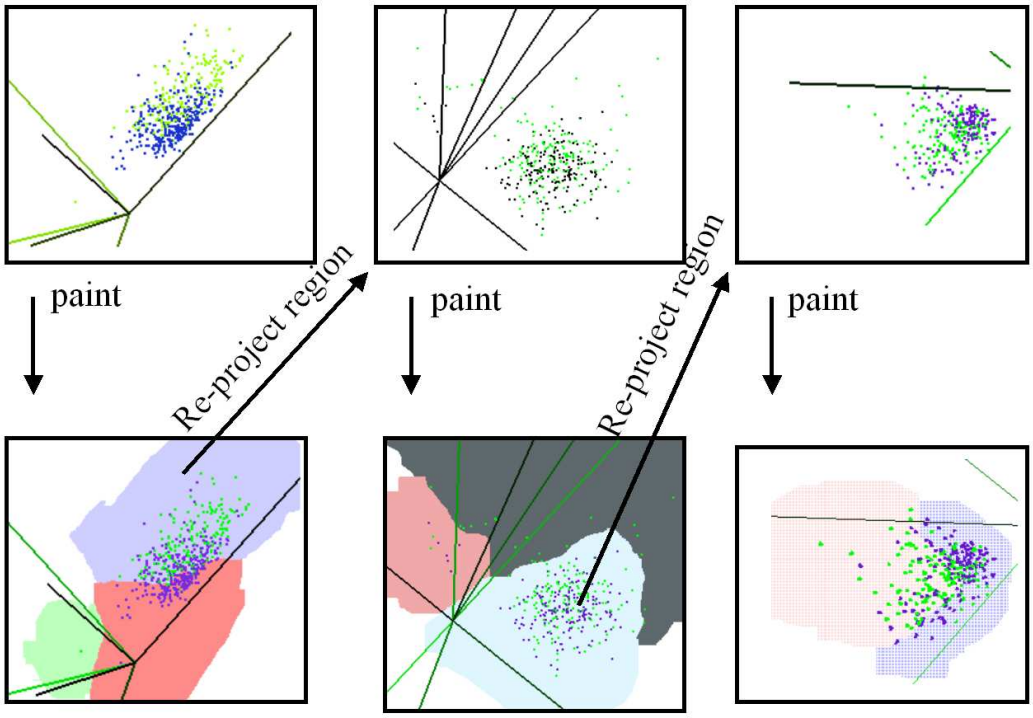

Figure 6: Classifi cation process of the **diabetes** dataset using StarClass. Compared to the Statlog **Segment** dataset shown in Figure 4, the boundary between the classes in the **diabetes** dataset is much less clear because of the noisy nature of the data.

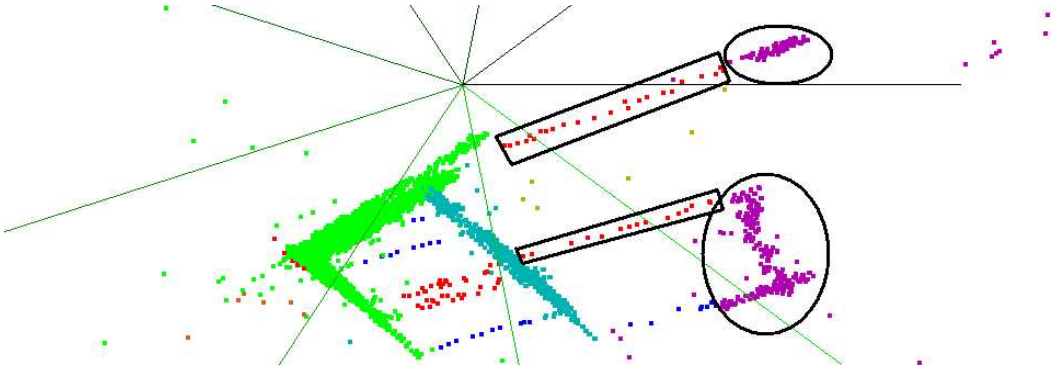

Figure 7: Star Coordinates projection of the **Shuttle** dataset. Characteristics of different classes can be observed, for example, there are distinct clusters within the same class: two clusters belonging to the red class are boxed and two clusters belonging to the purple class are circled.

into an algorithmic classification system. Other advantages of visual classification include the increase in the user's understanding of the data as a result of having a visual image of the data, and the confidence gained in the result of the classification because the user can see the way the classes are separated.

We have experimentally verified the effectiveness of StarClass by applying it to the classification of actual data with up to 19 dimensions, and comparing its performance to that of well-known algorithmic and visual classification methods. Because of advantages in handling multi-variate class boundaries, in accuracy and in better user understanding, we believe that StarClass can make a substantial contribution to the classification effort. With further improvements such as the incorporation of algorithmic techniques, StarClass can achieve even better results.

7 Acknowledgements.

his work has been sponsored in part by Department of Energy SciDAC program (Memorandum agreements No. DE-FC02-01ER41202), and the National Science Foundation PECASE award (ACI 9983641), ITR program (ANI-0220147), and LSSDSV program (ACI 9982251).

References

[1] R. Agrawal, S. Ghosh, T. Imielinski, B. Iyer, and A. Swami. An Interval Classifier for Database Mining Applications. *Proc. 18th Intl. Conf. on Very Large Databases (VLDB '92)*, pp. 560–573, Vancouver, B.C., Canada, August 1992.

[2] M. Ankerst, C. Elsen, M. Ester, and H.-P. Kriegel. Visual classification: An interactive approach to decision tree construction. *Proc. 5th Intl. Conf. on Knowledge Discovery and Data Mining (KDD '99)*, pages 392–396, 1999.

[3] M. Ankerst, M. Ester, and H.-P. Kriegel. Towards an effective cooperation of the user and the computer for classification. *Proc. 6th Intl. Conf. on Knowledge Discovery and Data Mining (KDD '00)*, 2000.

[4] K. Alsabti, S. Ranka, and V. Singh. CLOUDS: A Decision Tree Classifier for Large Datasets. *Proc. 4th Intl. Conf. on Knowledge Discovery and Data Mining (KDD '98)*, New York, 1998, pp. 2–8.

[5] W.-H. Au and K.C.C. Chan Classification with Degree of Membership: A Fuzzy Approach. *Proc. 2nd IEEE Intl. Conf. on Data Mining (ICDM '02)*, 2002.

[6] U. Fayyad, G. Piatetsky-Shapiro, and P. Smyth. The KDD Process for Extracting Useful Knowledge from Volumes of Data *Communications of the ACM 39*, 11, 1996.

[7] P. Hoffman and G. Grinstein. Dimensional anchors: A graphic primitive for multidimensional multivariate information visualizations. *Proc. New Paradigms in Information Visualization and Manipulation*, 1999.

[8] C.Z. Janikow. Fuzzy Decision Trees: Issues and Methods. *IEEE Trans. on Systems, Man, and Cybernetics - Part B: Cybernetics*, vol. 28, no. 1, pp 1-14, 1998.

[9] E. Kandogan. Visualizing Multi-Dimensional Clusters, Trends, and Outliers using Star Coordinates. *Proc. ACM SIGKDD '01*, pp. 107-116, 2001.

[10] B. Liu, W. Hsu, and Y. Man. Integrating Classification and Association Rule Mining. *Proc. 4th Intl. Conf. on Knowledge Discovery and Data Mining (KDD '98)*, New York, 1998.

[11] D. Michie, D.J. Spiegelhalter, and C.C. Taylor. *Machine Learning, Neural and Statistical Classification*. Ellis Horwood, 1994.

[12] T.M. Mitchel. *Machine Learning*. McGraw Hill, 1997.

[13] NASA Ames Research Center *Introduction to IND Version 2.1*, 1992.

[14] J.R. Quilan. *C4.5: Programs for Machine Learning* Morgan Kaufman, 1993.

[15] J.C. Shafer, R. Agrawal, and M. Mehta. SPRINT: A Scalable Parallel Classifier for Data Mining. *Proc. 22nd Intl. Conf. on Very Large Databases (VLDB '96)*, Bombay, India, 1996, pp. 544–555.

[16] J.W. Smith, J.E. Everhart, W.C. Dickson, W.C. Knowler, and R.S. Johannes. Using the ADAP Learning Algorithm to Forcast the Onset of Diabetes Mellitus. *Proc. Symp, on Computer Applications and Medical Cares*, pp. 422–425, 1983.

[17] H. Theisel and M. Kreuseler. An enhanced spring model for information visualization. *Computer Graphics Forum*, 17(3), September 1998.

[18] J. Wang, B. Yu, and L. Gasser. Concept Tree Based Ordering for Shaded Similarity Matrix *Proc. 2nd IEEE Intl. Conf. on Data Mining (ICDM'02)*, 2002.