PaintingClass: Interactive Construction, Visualization and Exploration of Decision Trees

Soon Tee Teoh
Department of Computer Science
University of California, Davis
teoh@cs.ucdavis.edu

Kwan-Liu Ma
Department of Computer Science
University of California, Davis
ma@cs.ucdavis.edu

ABSTRACT

Decision trees are commonly used for classification. We propose to use decision trees not just for classification but also for the wider purpose of knowledge discovery, because visualizing the decision tree can reveal much valuable information in the data. We introduce PaintingClass, a system for interactive construction, visualization and exploration of decision trees. PaintingClass provides an intuitive layout and convenient navigation of the decision tree. PaintingClass also provides the user the means to interactively construct the decision tree. Each node in the decision tree is displayed as a visual projection of the data. Through actual examples and comparison with other classification methods, we show that the user can effectively use PaintingClass to construct a decision tree and explore the decision tree to gain additional knowledge.

Categories and Subject Descriptors

H.1.2 [**Information Systems**]: User/Machine Systems—*Human Information Processing*; H.2.8 [**Database Management**]: Database Applications—*Data Mining*; I.3.6 [**Computer Graphics**]: Methodology and Techniques

Keywords

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

1. INTRODUCTION

Classification of multi-dimensional data is one of the major challenges in data mining. In a classification problem, each object is defined by its attribute values in multi-dimensional space, and furthermore each object belongs to one class among a set of classes. The task is to predict, for each object whose attribute values are known but whose class is unknown, which class the object belongs to. Typically, a classification system is first trained with a set of data whose attribute values and classes are both known. Once the system has built a model based on the training, it is used to assign a class to each object. For example, neural networks have been used effectively for this purpose in algorithms such as [17] and [21].

A decision tree classifier first constructs a decision tree by repeatedly partitioning the dataset into disjoint subsets. One class is assigned to each leaf of the decision tree. Most classification systems, including most decision tree classifiers, are designed for minimal user intervention. More recently, a few classifiers have incorporated visualization and user interaction to guide the classification process. On one hand, visual classification makes use of human pattern recognition and domain knowledge. On the other hand, visualization gives the user increased confidence and understanding of the data [2, 3, 8].

A representative visual classification systems include Ankerst et al.'s PBC (Perception-Based Classifier) [2]. In [20], we proposed StarClass, a new interactive visual classification technique. StarClass allows users to visualize multi-dimensional data by projecting each data object to a point on 2-D display space using Star Coordinates [11]. 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.

In this paper, we introduce PaintingClass, a complete user-directed decision tree construction and exploration system. PaintingClass uses some ideas from StarClass, but has two main features as its contributions:

- PaintingClass introduces a new decision tree exploration mechanism, to give users understanding of the decision tree as well as the underlying multi-dimensional data. This is important to the user-directed decision tree construction process as users need to efficiently navigate the decision tree to grow the tree.
- PaintingClass extends the technique proposed to StarClass so that datasets with categorical attributes can also be classified. Many real-world applications use data containing both numerical and categorical attributes; therefore PaintingClass is much more useful than StarClass. We show the effectiveness of PaintingClass in classifying some benchmark datasets by comparing accuracy with other classification methods.

These features make PaintingClass an effective data mining tool. PaintingClass extends the traditional role of decision trees in classification to take on the additional role of identifying patterns, structure and characteristics of the dataset via visualization and exploration. This paradigm is a major contribution of PaintingClass. In this paper, we show some examples of knowledge gained from the visual exploration of decision trees.

2. PAINTINGCLASS OVERVIEW

SIGKDD '03, August 24-27, 2003, Washington, DC, USA

As is typical of decision tree methods, PaintingClass starts by accepting a set of training data. The attribute values and class of each object in the training set is known. In the root of the PaintingClass decision tree, every object in the training set is projected and displayed visually. In PaintingClass, each non-terminal node in the decision tree is associated with a *projection*, which is a definite mapping from multi-dimensional space into two-dimensional display. The user creates a projection that best separates the data objects belonging to different classes.

Each projection is then partitioned by the user into *regions* by *painting*. Next, for each region in the projection, the user can choose to re-project it, forming a new node. In other words, the user creates a projection for this new node in a way that best separates the data objects in the region leading to this node.

For every new node formed, the user has the option of partitioning its associated projection into regions. The user recursively creates new projection/nodes until a satisfactory decision tree has been constructed. Each projection thus corresponds to a non-terminal node in the decision tree, and each un-projected region thus corresponds to a terminal node. In this way, for each non-root node, only the objects projecting onto the chain of regions leading to the node are projected and displayed.

In the classification step, each object to be classified is projected starting from the root of the decision tree, following the region-projection edges down to an un-projected region, which is a terminal node (ie. a leaf) of the decision tree. The class which has the most training set objects projecting to this terminal region is predicted for the object.

The following sections will explain in more detail how projections are created and how regions are specified.

3. VISUAL PROJECTIONS USED IN PAINTINGCLASS

In StarClass [20], we used Star Coordinates [11] to project data defined in higher-dimensional to two-dimensional display. In PaintingClass, we use a similar but less restrictive paradigm. Instead of using only Star Coordinates, we conceptually can use any sensible projection method. This is important because each different projection method has its own advantages and drawbacks. Since each dataset has its unique characteristics, choosing the most appropriate method for a dataset or a subset of a dataset can better reveal the underlying structure of the data. In our current implementation, we allow the user to choose between Star Coordinates and Parallel Coordinates [9] projection methods. Star Coordinates is better at showing dimensions with numerical attributes and Parallel Coordinates is better at showing dimensions with categorical attributes. With just this simple choice between Star and Parallel Coordinates, PaintingClass can be used to classify datasets with both numerical and categorical attributes, giving it much wider applicability than StarClass.

3.1 Star Coordinates

Star Coordinates was first proposed as a method for visualizing multi-dimensional clusters, trends, and outliers. The two-dimensional screen position of an n-dimensional object is given by the vector sum of all unit vectors on each coordinate multiplied by the value of the data element in that coordinate. Each unit vector corresponds to one dimension and is shown by a line on the display. To edit a Star Coordinates 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.

3.2 Parallel Coordinates

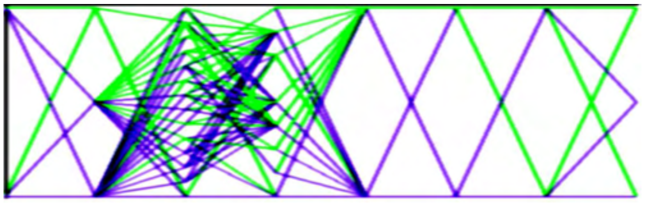

Figure 1: Parallel coordinates projection of the Australian dataset using the categorical dimensions. The categorical attribute values are enumerated. Since in some dimensions, the categorical attribute values can have only a few discrete values, many lines overlap. The resulting image does not convey much useful information.

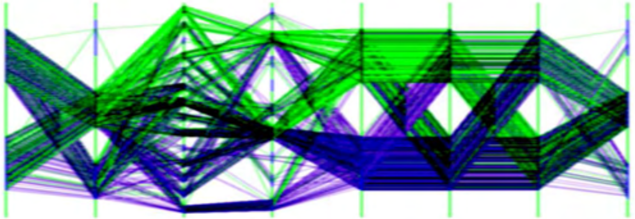

Figure 2: Modified parallel coordinates: Each integer value is mapped to an interval of length half the distance between one integer value and the next, and the mapping of an object to a parallel axis within this interval is determined by the ID number of the object. In this way, more lines are visible. More information is conveyed, for example the reader can verify that between the fifth and sixth axes, most of the objects in the purple class are projected to the lower half.

Parallel Coordinates [9] is a widely-used multi-dimensional data visualization method. Each dimension is represented by a vertical line, whose end-points represent the maximum and minimum value of the dimensionl. Each object therefore maps to one point on each of these lines, according to its attribute value in that dimension. The object is then displayed as a poly-line connecting all the points.

In PaintingClass, we slightly modify the standard Parallel Coordinates to better display the categorical dimensions. First, for each categorical dimension, all the possible attribute values are enumerated. Since the enumerated values are all integer values, and in some dimensions the number of distinct values is small, the resulting parallel projection contains many lines completely obsuring other lines. The information conveyed is therefore not useful. Since the enumerated categorical values are all integer values, in PaintingClass, each integer value is mapped to an interval of length half the distance between one integer value and the next, and the mapping of an object to a parallel axis within this interval is determined by the ID number of the object. Furthermore, lines are rendered in a semi-transparent manner. In this way, lines do not completely obscure each other. Figures 1 and 2 show the contrast between the original and the modified methods.

Another problem encountered in Parallel Coordinates visualization of categorical data is that when there are too many distinct values in a dimension, the interval of each value becomes too small, making it hard to select. Therefore, PaintingClass allows the user to specify a section on any axis to zoom in on.

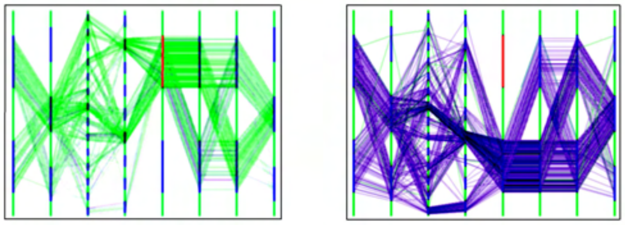

Figure 3: Painting regions in a parallel coordinates projection. The user clicks on an interval to toggle it between blue and red. In this figure, the top interval in the fifth dimension has been set to red. The objects belonging to the red region are shown in the left picture, and the objects belonging to the blue region are shown in the right picture. This separates the data shown in Figure 2 into two regions. In the blue region, nearly all the objects belong to the purple class, whereas in the red region, the majority of the objects belong to the green class, though there are still some purple class objects, so further separation is necessary.

4. PAINTING REGIONS

The main idea behind the decision tree construction method is to identify regions in the projected two-dimensional display that would separate the objects of different classes as much as possible. In a Star Coordinates projection, this is done by moving the axes around until a satisfactory projection is found. For example, in the middle image, the blue class is rather well-separated from the rest. When the user is satisfied with a projection, the user specifies regions by “painting” over the display with the mouse icon in the same way as in StarClass.

The “painting” paradigm can be extended to paint regions in the Parallel Coordinates projection. Initially, in a Parallel Coordinates projection, all intervals are set to belong to the blue region. The user clicks on an interval to change it to red. An object belongs to the red region if for every dimension with at least one red interval, the object has attribute value equal to a red interval. The object belongs to the blue class otherwise. An example is shown in Figure 3.

5. DECISION TREE VISUALIZATION AND EXPLORATION

Decision tree visualization and exploration is important for two mutually-complimentary reasons. First, to effectively and efficiently build a decision tree, it is crucial to be able to navigate through the decision tree quickly to find nodes that need to be further partitioned. Second, exploration of the decision tree aids the understanding of the tree and the data being classified. From the visualization, the user gains helpful knowledge about the particular dataset, and can more effectively decide how to further partition the tree.

The ultimate goal of PaintingClass is to maximize user understanding and knowledge. It is thus not the goal to display as much information as possible in the available screen space, because clutter may be detrimental to user understanding. The challenge then is to utilize the available display to let the user absorb as much useful information as possible. This is achieved through space-efficient visual metaphors which take advantage of human intuition. A problem encountered in attempting to visualize a PaintingClass decision tree is that there is too much data to be displayed coherently on a single screen. The visualization and exploration mechanism must therefore not only convey as much information as possible in a display but also allow the user to quickly navigate the tree to gain even more knowledge.

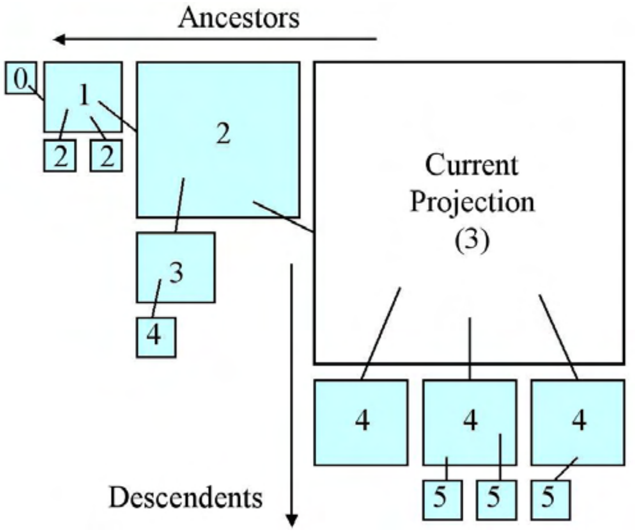

Figure 4: PaintingClass decision tree layout schematic. The current projection (ie. the projection in focus) is drawn as the largest square in the upper right corner of the display. The parent of the current projection is drawn to its immediate left, and the line of ancestors up to the root is drawn in this way. For every projection, its children are drawn directly below it. In this figure, each projection is labelled with the number representing its distance from the root.

Numerous tree visualization techniques including some specifically designed for visualizing decision trees [3, 6], have been proposed in the past. However, these methods are not appropriate for visualization decision trees generated by PaintingClass. This is because each node in PaintingClass is itself a picture requiring space. Furthermore, the display of the decision tree needs to convey clearly the relationship between regions and their corresponding nodes.

Figure 4 shows a schematic of the decision tree layout we designed for PaintingClass. It makes use of the focus+context concept, focusing on one particular projection, called the “current projection”, which is given the most screen space and shown in the upper right corner. The rest of the tree is shown as context to give the user a sense of where the current node is within the decision tree. Nodes that are close to the node in focus are allocated more space, because they are more relevant and the user is more likely to be interested in their contents. The ancestors (up to and including the root) of the node in focus are drawn in a line towards the left. The line including the focus node and all its ancestors is called the $ancestor\ line$. Except with both parent and child are in the ancestor line, the children of each node are drawn directly below it, in accordance with traditional tree drawing conventions. This layout is simple to understand, intuitive, and immediately portrays the shape and structure of the decision tree being visualized.

For exploration purposes, interactivity is of utmost importance. PaintingClass allows the user to easily navigate the tree by changing the node in focus. This is done either by clicking on the arrow on the upper right corner of a projection to bring it into focus, or by clicking on the arrow on the lower left corner of a projection in the ancestor line to bring it out of focus. In this case, the parent of the projection brought out of focus will be the new focus.

5.1 Auxiliary display

In the PaintingClass interface, the auxiliary display is shown in the lower left corner of the window. This portion of the window is not utilized in the display of the decision tree. PaintingClass thus makes use of this space to display some very important information that can aid the user's understanding of the data and the construction of the decision tree.

While decision tree visualization gives a good overview of the data, it is sometimes necessary for the user to see additional information. For example, in the Star Coordinates projection, many objects often map to the same pixel or nearby pixels. In fact, two objects with large Euclidean distance between them in high-dimensional space may map to nearby pixels, and it is hard to distinguish them. Therefore, as in StarClass, PaintingClass allows the user to zoom in on a region in the current projection, showing the region as auxiliary display using the "sticks" feature found in Star Coordiates.

When visualizing datasets with both numerical and categorical attributes, PaintingClass allows the user the option to use Star Coordinates or Parallel Coordinates. If the user chooses Star Coordinates, only the numerical attributes are used to determine the projected position of each point. Likewise, if the user chooses Parallel Coordinates, only the categorical attributes are used. This gives an incomplete picture of the data. Therefore, the auxiliary display can also be used to display the "dual" projection. In other words, when Star Coordinates is used to display the numerical attributes in the current projection, Parallel Coordinates can be used to display the categorical attributes in the auxiliary display, and vice versa. An example is shown in Figure 5.

The user clicks on the labels above the auxiliary display to choose between whether to show "sticks" zoom or to show the "dual" projection.

5.2 User interface for building the decision tree

As mentioned in Section 2, the decision tree construction process involves the repetition of three steps: (1) edit a projection, (2) specify regions, and (3) create a new node/projection for some region. In PaintingClass, interaction tools are provided so that users can easily perform each of these steps.

In Step 1, the PaintingClass control panel provides a check box for users to choose between Star Coordinates and Parallel Coordinates projection.

In Star Coordinates projection, the way the user moves the axes and paints a region is the same: holding down the left mouse button and dragging the mouse icon across the screen. To resolve the ambiguity, a control bar is displayed at the left edge of the display. The user clicks on the "edit axes" box, or one of the colored boxes to determine if the user wants to edit the axes or paint the region represented by the selected color.

In Parallel Coordinates projection, the user defines the red and blue regions simply by clicking on an interval to toggle it between red and blue. There are two additional boxes shown below the Parallel Coordinates display, one box is red and the other is blue. The user clicks on either box to toggle the display between showing the red or the blue region.

To create a new node from a region, the user needs to specify which region in the current projection to re-project. In Star Coordinates mode, this is simply done by clicking on one of the colored boxes in the same control bar at the left of the display. In Parallel Coordinates mode, the region selected by the two colored boxes below the display is used. The user then clicks on the "Create Projection" button in the control panel, and a new projection is created.

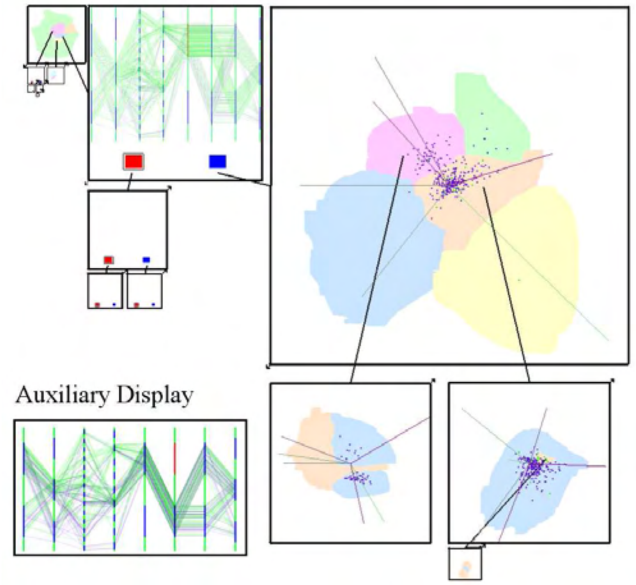

Figure 5: An auxiliary display is shown on the un-utilized space at the lower left of the display. This auxiliary display is used to supplement the user's exploration of the data. It can be used to show a detailed view of part of the data or the "dual" projection, which is the case in this figure. Since the current projection uses Star Coordinates to show the objects based on their numerical attributes, the "dual" projection uses Parallel Coordinates to show the categorical attributes.

5.3 Classification

PaintingClass counts the number of objects belonging 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 is predicted that class.

6. RESULTS

The dual objectives of PaintingClass are (1) to create a user-directed decision tree classification system, and (2) to enable users to explore and visualize multi-dimensional data and their corresponding decision trees to improve user understanding and to gain knowledge. We evaluate the success of PaintingClass in meeting the first goal by running experiments on well-known benchmark datasets and testing its accuracy, comparing it to other classification techniques. We also show some examples of how PaintingClass visualization is able to reveal important information in the datasets used.

6.1 Experimental Evaluation of accuracy

The design objective of PaintingClass is to create a decision-tree construction, visualization and exploration system so that the user can gain knowledge of the data being analyzed. Therefore, it is not the goal of PaintingClass to achieve superior classification accuracy. However, for PaintingClass to be a viable data mining tool, the decision trees generated by PaintingClass have to be "good". A good decision tree is defined as one that is able to effectively par-

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

	Algorithmic			Visual	
	CART	C4	SLIQ	PBC	**PaintingClass**
Satimage	85.3	85.2	86.3	83.5	**85.3**
Segment	84.9	95.9	94.6	94.8	**95.2**
Shuttle	99.9	99.9	99.9	99.9	**99.9**
Australian	85.3	84.4	84.9	82.7	**84.7**

Table 3: Accuracy of PaintingClass compared with other classification methods.

	CBA	C4.5	FID	Fuzzy	**PaintingClass**
Australian	85.0	82.6	58.0	88.9	**84.7**
adult	84.2	85.4	23.6	85.9	**85.1**
diabetes	74.4	73.8	62.0	77.6	**74.6**

tition the domain of the data, so to accurately predict the classes of objects. The structure of a good decision tree therefore reveals some underlying information about the data. Such a decision tree is worth visualization and exploration in PaintingClass. Therefore, we need to show that the decision trees generated by PaintingClass are good, and we do that by experimenting on well-known benchmark datasets and comparing accuracy results with popular existing classification systems.

We used the $Satimage$, $Segment$, $Shuttle$ and $Australian$ datasets from the benchmark Statlog database [15] for evaluating the accuracy of PaintingClass classification. We also use two additional well-known benchmark datasets, $diabetes$ [19] and $adult$ [12] in our evaluation. The description of these datasets is presented in Table 1.

Table 2 compares the accuracy of PaintingClass against the accuracy of popular algorithmic decision tree classifiers CART and C4 from the IND package [16], as well as SLIQ [14], and also visual classifier PBC. The results are taken from [3]. Table 3 shows how PaintingClass compares with CBA [13], C4.5 [18], FID [10], and Fuzzy [5]. The accuracy figures for these methods are taken from [5].

From the experimental results, PaintingClass performs well compared with the other methods. In particular, it appears that PaintingClass is slightly more accurate than PBC. Since the classification task is performed by the user in PaintingClass, the accuracy is highly dependent on the skill and patience of the user, and even the same user can produce some small variations in accuracy for different attempts. However, the accuracy produced by a competent user as shown in the results is sufficient to establish PaintingClass as a viable method for constructing decision tree. Small variations in accuracy do not change the conclusion of this evaluation. More importantly, the accuracy achieved shows that the decision trees generated by PaintingClass are effective in partioning the data, and therefore their visualization and exploration can yield valid and important information. The knowledge gained from the exploration is discussed in the next section.

6.2 Knowledge Gained

In [7], Buja and Lee mention that data mining, unlike traditional statistics, is not concerned with modeling all of the data, but with the search for interesting parts of the data. Therefore, the goal is not to achieve optimal performance in terms of global performance measures such as classification accuracy. In our design of PaintingClass, we follow the same principles. As a result, PaintingClass decision tree is a powerful tool in the discovery of knowledge in datasets for the following reasons:

- Exploration of the decision tree allows the user to see the hierarchy of split points. These split points reveal which dimensions, combinations of dimensions, and which values with each dimension are most correlated to different classes.

- The shape of the decision tree indicates certain characteristics of the data. For example, the unbalanced tree constructed for the $Australian$ dataset suggests that certain large clusters are "pure", meaning that they contain only objects belonging to one class and therefore need no further partitioning, whereas certain clusters are "mixed", meaning that they contain objects of different classes and therefore need to be partitioned further.

- The confidence in the prediction of the class of a data object can vary widely. This can be visualized clearly in PaintingClass, sometimes objects belonging to a single class appear clustered in the projection chosen by the user, far away from objects belonging to other classes. In this case, the user can have great confidence that an object projected to that area would belong to this class. This is contrasted with a "mixed" region, where the user is unable to edit the projection to separate objects belonging to different classes. Objects of different classes fall into the same region, and they are all predicted to belong to the majority class of the training-set objects projecting to this region. However, in this case, there is much less confidence in the prediction.

- Each node in the decision tree is itself a visual projection of a subset of the data. The projection used, whether Star Coordinates or Parallel Coordinates or any other method which may be incorporated into PaintingClass in the future, can reveal patterns, clusters (and their shapes) and outliers, as shown through various examples in this paper. These displays provide the user with rich information, not just statistical measures expressed as single numbers.

- Following the hierarchy of nodes in the decision tree allows the user to focus on subsets of the data. For example, the $adult$ dataset split by the married-civ-spouse category in the marital status dimension clearly indicates a strong correlation between married-civ-spouse and high income. To focus attention on only people who are married-civ-spouse, the user simply explores this subtree. Within this subtree, PaintingClass visualization reveals a strong correlation between higher education and high income.

7. FUTURE WORK

Although experimental evaluation has indicated the effectiveness of PaintingClass, we believe there are some areas where PaintingClass can be further improved. For example, just as Ankerst et al. have incorporated algorithmic support into PBC with significant success [3], we are also extending PaintingClass to include options for using automatic algorithms in cases where human judegment is difficult and also to aid in searching for optimal splits to relieve user tedium. We believe that with some algorithmic support, the accuracy and usability of PaintingClass can be further improved. We would like to apply PaintingClass to more datasets to further investigate its effectiveness and 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 PaintingClass to handle very large datasets. This could be done, for example, by random sampling.

Table 1: Description of the datasets.

Dataset	Training Set Size	Testing Set Size	num classes	numerical dimensions	categorical dimensions
Satimage	4435	2000	6	4	0
Segment	2310	10-fold	7	19	0
Shuttle	43500	14500	7	9	0
diabetes	768	3-fold	2	8	0
Australian	690	10-fold	2	6	8
adult	32561	16281	2	6	8

8. CONCLUSIONS

We have presented PaintingClass, a decision tree construction, visualization and exploration system. In PaintingClass, the user interactively edits projections of multi-dimensional data and paints regions to build a decision tree. We have experimentally verified the effectiveness of PaintingClass 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. With further improvements such as the incorporation of algorithmic techniques, PaintingClass can achieve even better results in terms of accuracy.

Yet it is not the sole purpose of PaintingClass to facilitate interactive construction of decision trees. The decision tree layout and navigation method introduced by PaintingClass allows users to explore decision trees. Exploration of decision trees fulfills some general goals of data mining beyond classification. We have shown that visualizing the structure of the decision tree, the individual nodes, and exploring through the hierarchy can reveal valuable knowledge about the dataset.

Besides introducing the decision tree exploration mechanism, PaintingClass also provides several useful extensions to StarClass, the most important of which is the use of Parallel Coordinates to display categorical values so that even datasets with categorical dimensions can be classified and visualized.

We believe that PaintingClass is a practical and effective data exploration and classification tool, and the idea of using decision tree visualization for knowledge discovery is an important contribution. Further features built on the PaintingClass platform may make it an even more powerful system.

9. 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)*, pp. 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] T. Barlow and P. Neville. Case Study: Visualization for Decision Tree Analysis in Data Mining. *Proc. IEEE Symposium on Information Visualization*, 2001.

[7] A. Buja and Y-S. Lee. Data Mining Criteria for Tree-Based Regression and Classification. *Proc. 7th Intl. Conf. on Knowledge Discovery and Data Mining (KDD '01)*, 2001.

[8] 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.

[9] A. Inselberg. The Plane with Parallel Coordinates. *Special Issue on Computational Geometry: The Visual Computer*, vol. 1, pp. 69–91, 1985.

[10] 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.

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

[12] R. Kohavi. Scaling Up the Accuracy of Naive-Bayes Classifiers: A Decision Tree Hybrid. *Proc. 2nd Intl. Conf. on Knowledge Discovery and Data Mining (KDD '96)*, Portland, Oregon, 1998.

[13] 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.

[14] M. Mehta, R. Agrawal, and J. Rissanen. SLIQ: A Fast Scalable Classifier for Data Mining *Proc. Intl. Conf. on Extending Database Technology (EDBT '96)*, Avignon, France, 1996.

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

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

[17] R. Parekh, J. Yang, and V. Honavar. Constructive Neural-Network Learning Algorithms for Pattern Classification. *IEEE Trans. on Neural Networks*, vol.11, no.2, 2000.

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

[19] 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.

[20] S.T. Teoh and K.L. Ma. StarClass: Interactive Visual Classification Using Star Coordinates. *Proc. 3rd SIAM Intl. Conf. on Data Mining (SDM '03)*, 2003.

[21] Z.-H. Zhou, Y. Jiang, and S.F. Chen. A General Neural Framework for Classification Rule Mining. *Intl. Journal of Computers, Systems, and Signals*, vol.1, no.2, pp.154–168, 2000.