

JRV: An Interactive Tool for Data Mining Visualization

Danyu Liu
Department of Computer &
Information Sciences
University of Alabama
at Birmingham
liudy@cis.uab.edu

Alan Sprague
Department of Computer &
Information Sciences
University of Alabama
at Birmingham
sprague@cis.uab.edu

Upender Manne
Department of Pathology
University of Alabama
at Birmingham
manne@path.uab.edu

ABSTRACT

In this paper, we demonstrate *JRV*, a new data mining visualization tool for the knowledge discovery process where the user and computer can cooperate with each other. First, the computer can be instructed by the user interactively to compute values of several evaluation functions. Then, the user can take advantage of domain knowledge and assess the intermediate results obtained. Furthermore, by providing effective and efficient data visualization, the pattern recognition capacities of users can be greatly improved. Instead of being limited to two attributes at a given time in independence diagrams, this novel tool will allow simultaneous analyses of multiple attribute dependencies using four different drawing panels. Also, by utilizing the existing techniques of data visualization, we design a general model which can handle both categorical and numerical attributes in an intuitive way. With this model, we can identify patterns of interests efficiently. Through actual examples, we show that it might help users to find novel attribute relationships. This work is supported by NIH grant # RO1-CA98932-01.

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

General Terms

Algorithms, Design, Experimentation, Human Factors

Keywords

visual data mining, information visualization, interactive visualization, panda-index

1. INTRODUCTION

With the exponential accumulation of data from several domains such as business, medicine, science and government, it is very important to develop new data mining techniques and tools to help knowledge experts to extract novel and useful information from these raw data. As one of the powerful knowledge discovery techniques, data visualization has received increasing attention recently [3], [8] and [10]. The data visualization combines several techniques from different fields, such as data mining, cognitive

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

ACMSE'04, April 2–3, 2004, Huntsville, Alabama, USA.
Copyright 2004 ACM 1-58113-870-9/04/04...\$5.00.

science, graphics design and interactive computer graphics. Fayyad et al. [4] notice the importance of integrating user interaction and data visualization into the whole knowledge discovery process to help in finding understandable patterns for humans.

However, most existing tools are developed for limited interaction with users and without human domain knowledge. Generally, the user chooses datasets and sets of some parameters related to algorithms; however the selection of these parameters is tedious and difficult to determine a priori. Recently, Ankerst et al. [10] propose strategies to assimilate human domain knowledge and integrate data visualization and user interaction into knowledge discovery procedure for three important reasons:

- (1) With the help from data visualization, the capacities of human to find useful patterns can be greatly improved.
- (2) The users will trust the created patterns from this interaction process.
- (3) Domain knowledge of users can steer the data mining process.

In addition, visualization techniques act as complements to the data mining procedure and aid in deciding the appropriate data mining technique to use and appropriate subsets of the data to be considered.

One of the most important issues in current data mining research is to find complicated dependencies among attributes. Several techniques, including *independence diagrams* [1], *equiwidth histograms*, *correlation coefficients* and *scatterplots*, have been proposed to pinpoint the underlying attribute dependence. [1] creates the independence diagrams to find the underlying relationship between two attributes using visualization techniques. They define a two-dimension grid and display the grid in the following way: the more brightness of a cell, the more data items each cell contains. Independence diagrams is one of the few visualization techniques that use graphics methods to represent the underlying raw data rather than data distribution in the dataset or discovered knowledge. It can only compare two attributes at the same time, so this limitation severely affects the efficiency of the tool. As another simple and efficient technique, *Line Graph* can represent data trend in the dataset by drawing connecting lines between data points, however it has two major shortcomings: First, the knowledge the user can derive from it depends on the degree of overlap. Second, this technique can only represent limit number of records due to the width of computer screen. [9] presents a novel technique, called *circle segments*, for visualizing large amounts of high-dimensional data; however it can not visualize categorical attributes.

Since most of the data collected from business, medicine, science and government is multi-dimensional; i.e. each dataset contains at least three attributes for each entry, the task to visualizing multi-dimensional data is not trivial, because we have to project these

multi-dimensional data onto the computer screen which is based on a two-dimensional space.

The motivation for this paper is to develop JRV, a visualization tool, for knowledge discovery which has the following desired characteristics:

- (1) User friendly interface.
- (2) Representing all data dimensions using single visual cue.
- (3) Capacity to deal with large amount of multi-dimensional data.
- (4) Handling both categorical attributes and numerical attributes.
- (5) Providing facilities to compute the values of selection measures to help users to approximate the quality of attributes as a reference in finding the relationship among these attributes.

The inspiration of the present work come from Ankerst et al.'s *Bar Visualization* [10], where each color cell inside the bar represents a sorted attribute value and the color is determined by its corresponding classification label. However, users can not perceive any attribute relationship from this representation directly. We extend this existing visualization model and propose a new algorithm to compute the purity of generated bars, so users can use this new technique to scrutinize the underlying attribute dependence in datasets efficiently. The basic idea of the new model – as we shall explain later, is to represent the selected attribute (source attribute) graph bar, which is generated using the bar visualization technique, by reorganizing it into several sub-bars based on the values of another specified attribute (group-by attribute) where we try to find the direct attribute relationship from *source attribute* to *group-by attribute*. We argue that the stronger the relationship two attributes have, the purer the created sub-bar graphs will be. The purity of the graph bar is defined by a novel evaluation function, called *panda-index*, which has been discussed in detail elsewhere in this paper.

The rest of this paper is organized as follows. Section 2 reviews some well known attribute selection measures: information gain, gain ratio and gini index. In section 3, we introduce our techniques of visualizing the underlying dataset to represent the attribute dependence. Section 4 presents the user interface of JRV and reports the results of some experiment evaluation on the STATLOG benchmark [12]. The last section concludes this paper and discusses some future work.

2. SELECTION MEASURES

Several selection measures have been proposed to evaluate the quality of attributes, including numerical attributes and categorical attributes. Besides using visualization techniques to represent the underlying raw data, JRV also provides facilities to compute the values of selection measures, such as information gain, gain ratio and gini index, of the specified attribute to help users to estimate the quality of attributes where these values will provide users some accessories in finding the attribute relationship among datasets. In

this section we briefly review the three important selection measures.

2.1 Information Gain

Information gain is proposed in the ID3 algorithm of Quinlan [14] as a measure to choose the best splitting attribute at each tree node during the construction of decision trees. It is alternatively referred to as an *attribute selection measure* or a *measure of the goodness of split* [5]. The attribute with the maximum entropy reduction is generally chosen as the best splitting attribute in current classification procedure. This attribute minimizes not only the information required to classify the samples in the resulting partitions but also the expected number of tests required to classify an object.

Let S be a dataset where $s = |S|$ represents the number of data samples inside this dataset. Suppose the class label attribute has n distinct values which determine n distinct classification results, C_i (for $i = 1, \dots, n$) where s_i is the number of data samples of S that belong to class C_i . The expected information, also known as the entropy, of set S is defined as

$$I(s_1, s_2, s_3, \dots, s_n) = - \sum_{i=1}^n p_i \log_2(p_i),$$

where P_i is the probability that an arbitrary sample belongs to class C_i and is given by s_i/s . Suppose attribute A has m distinct values, $\{a_1, a_2, a_3, \dots, a_m\}$ where it can be used to divide dataset S into m subsets, $\{S_1, S_2, S_3, \dots, S_m\}$, where S_j contains all those samples in S whose value of attribute A is a_j . The entropy, or expected information based on the partitioning into subsets by attribute A , is defined by

$$E(A) = \sum_{j=1}^m \frac{s_{1j} + s_{2j} + \dots + s_{nj}}{s} I(s_{1j}, s_{2j}, s_{3j}, \dots, s_{nj}),$$

where s_{ij} is the number of samples of class C_i in a subset S_j . Note that the smaller the entropy value, the great the purity of subset partitions. Also for a given subset S_j ,

$$I(s_{1j}, s_{2j}, s_{3j}, \dots, s_{nj}) = - \sum_{i=1}^m p_{ij} \log_2(p_{ij}),$$

where $p_{ij} = \frac{s_{ij}}{|S_j|}$ and is the probability that a sample in S_j belongs

to class C_i . The information gain that would be obtained by partitioning data set S on attribute A is given by

$$Gain(A) = I(s_1, s_2, s_3, \dots, s_n) - E(A)$$

In the ID3 algorithm, the attribute with the maximum information gain is chosen as the best splitting attribute to branching current datasets.

Table 1 Converting raw data records to entry lists

Income	Credit	Class
100	10	A
87	7	B
90	8	B
104	2	C
120	9	A

→

Income	Class
87	B
90	B
100	A
104	C
120	A

Credit	Class
2	C
7	B
8	B
9	A
10	A

Class	Color
A	Red
B	Green
C	Blue

2.2 Gain Ratio

Although information gain is a quality measure, it still has some limitations. One shortcoming is: it tends to prefer attributes with many values. An improvement [6] can be obtained by taking into consideration the cardinality of each division. The optimized approach uses gain ratio instead of information gain. The gain ratio is generated using the following two steps: First the split info is given by,

$$\text{split info}(X) = -\sum_{i=1}^n \frac{|T_i|}{|T|} \times \log_2 \left(\frac{|T_i|}{|T|} \right)$$

where T is a training set. The *split info* represents the potential information which is generated by dividing T into n subsets, whereas the information gain measures the information relevant to classification that arises from the same division. Then, the gain ratio is define by

$$\text{gain ratio}(X) = \text{gain}(X) / \text{split info}(X).$$

The gain ratio compensates for the number of attributes by normalizing by the information encoded in the split itself. This approach is easily understood for categorical attributes, but in the situation of numerical attributes a binary testing should be introduced in order to utilize the above formulas. Given A is a numerical attribute, we can separate values of attribute A into group A_1 and group A_2 by two inequalities $A \leq Z$ and $A > Z$ respectively, where Z is a threshold. Then we can use the above formulas to compute the gain ratio corresponding with the specified threshold Z by considering A as a categorical attribute, which has two values, A_1 and A_2 . Finally, the maximum gain ratio among all obtained gain ratio where each matches one specified threshold, is returned as the gain ratio of attribute A . It might look like that tests on numerical attributes would be difficult to perform, because they have arbitrary threshold. Several algorithms [7], [13] and [6] have been proposed to find appropriate thresholds against which to compare the values of numerical attributes. First, the training set T is sorted on the values of the attribute A . A has only a limited number of distinct values in T , so we can symbolize them in non-decreasing order as $\{V_1, V_2, \dots, V_m\}$. It is generally to select the midpoint of each interval as the tested threshold, thus there are only $m - 1$ thresholds splits on A . So the computational complexity of obtaining gain ratio of numerical attributes is $O(k)$, where k is the number of entries(records) in the training dataset.

2.3 Gini Index

Another popular evaluation function is the gini function, which is proposed by Breiman et. al in CART [7], a binary decision tree algorithm. The gini index for a database D is defined as

$$\text{gini}(D) = 1 - \sum p_j^2$$

where p_j is the probability of class j in D . The quality of a split of D into two subsets D_1 and D_2 is given by

$$\text{gini}_{\text{split}}(D) = \frac{n_1}{n} (\text{gini}(D_1)) + \frac{n_2}{n} (\text{gini}(D_2))$$

where n_1 and n_2 are the size of subset D_1 and D_2 respectively, and $n = n_1 + n_2$. For each possible split the impurity of the subgroups is summed and the split with the maximum reduction in impurity chosen [11]. According to the evaluated attribute type, numerical or

categorical, CART adopts different strategies to compute the gini index.

2.3.1 Methods for Numerical Attributes

First, the training dataset is sorted based on the numerical attribute being evaluated. Let us denote the sorted values in non-decreasing order as $\{V_1, V_2, \dots, V_m\}$. Second, the midpoint of each interval of the generated value sequence is chosen as the split point z . Third, an inequality of the form $A \leq z$, where A is the evaluated numerical attribute, is used to divide the ordered value sequence into two subsets. Because totally there exists $m - 1$ intervals, we only need to inspect $m - 1$ available split points. Thus the cost of assessing splits for a numerical attribute is determined by the cost of sorting the values of numerical attribute. SLIQ [11] proposes a pre-sorting technique which could reduce the cost of evaluating numerical attributes. Because gini index indicates impurity of dataset, the smaller the number, the better. After $m - 1$ computation, the minimum number is returned as the gini index of attribute A .

2.3.2 Methods for Categorical Attributes

Suppose $S(A)$ is the set of distinct values of a categorical attribute A . For n values of the attribute, there are $2^{n-1} - 1$ splits and $2^{n-1} - 1$ possible gini index values each corresponding one split. For each attribute, CART searches all splits and output the value of the best split as the attribute gini index.

3. VISUALIZATION FOR ATTRIBUTES RELATIONSHIP

The *Bar Visualization* technique proposed in [10] has two major advantages: First, both numerical attributes and categorical attributes can be visualized. Second, it can easily visualize a large amount of multi-dimension data. For example, it can efficiently represent the DNA data from STATLOG datasets [12] including 2000 data entries each with 180 attributes. In this section, we extend the bar visualization model where the induced new model can help users to find the novel attributes relationship by creating new patterns.

The bar technique for data visualization is built from two major ideas:

- (1) Each attribute of datasets is represented as bar graph and every square cell in the bar graph is based on the value of the represented attribute.
- (2) Each class label is indicated by different color.

3.1 Data Structure of Entry Lists

To efficiently represent the raw data, we use several entry lists to store converted data records. For each attribute of a dataset, we generate an independent attribute entry list which consists of two columns: one contains attribute values; the other contains their matching class label. In addition to attribute entry lists, we also create a separate list, named class entry list, where users can choose different color for each distinct class label. Therefore, the class entry list can be used to identify the class to which an attribute value belongs. Table 1 illustrates the state of the data structures before and after pre-processing. Also the values in the attribute entry lists are sorted independently and the generated order determines the rendering sequence of cells inside bar graph. In the situation of sorting numerical attribute values which have definite numerical order, this approach can be easily implemented. However, for categorical attribute values which do not have internal order, the same approach can not be directly applied.

Table 3: Income entry lists after splitting by Gender

Class	C	B	A	C	A	Gender = 'M'
Income	78	87	100	104	120	

Class	C	B	A	A	Gender = 'F'
Income	56	90	111	156	

Table 4: Income entry lists after splitting by Credit

Class	C	C	B	C	A	Credit <= 7
Income	56	78	87	104	156	

Class	B	A	A	A	Credit > 7
Income	90	100	111	120	

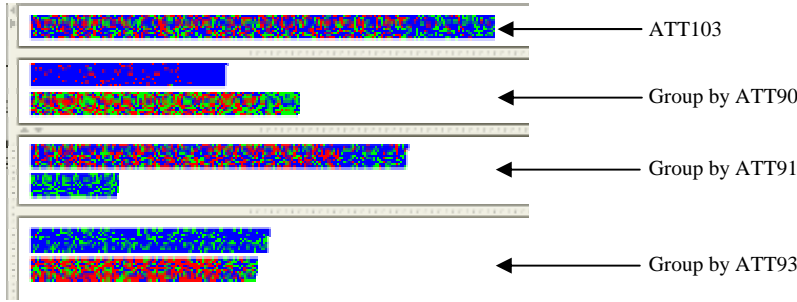


Figure 1. Visualization of DNA training data from STATLOG

Although, we could randomly assign different numbers to different categorical attribute values, this artificial order might seriously degrade the users' perception to find useful patterns in bar graph. It is imperative for us to find an efficient algorithm to sort categorical attribute values. Coppersmith et al. propose SLIQext [2], an algorithm for searching the best splitting point for categorical attributes, with good results reported. An order for sorting the categorical attribute values can be induced from this algorithm. In our JRV system, we adopt this induced order to arrange the values in entry lists of categorical attributes.

3.2 Visualization Techniques for Bar Graph

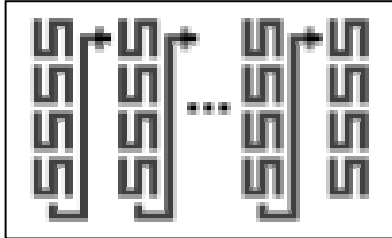


Figure 2. Cells Rendering Order Inside Bar Graph

We apply a new rendering pattern, called *zigzag* pattern, to represent the bar graph. Inside a bar, the sorted attribute values are arranged to different cells according to their position in the ordered sequence with a column by column *zigzag* style, which is illustrated in Figure 2. JRV system provides users interactions to adjust the height and width of this rendering pattern, so it helps users to effectively find useful patterns from generated bar graph.

3.3 Visualization Techniques for Attribute Relationship

The basic idea of our novel model to find attribute relationship by using bar visualization techniques is to represent the selected attribute (*source attribute*) graph bar, which is generated using the above bar visualization techniques, by splitting it into several sub-bars based on the values of another specified attribute (*group-by attribute*) where we try to find the direct attribute relationship

between *source attribute* and *group-by attribute*. We argue that the stronger the relationship two attributes have, the purer the created sub-bar graphs will be. The purity of the bar graph is measured by a novel evaluation function, called *panda-index*, which will be detailed in section 3.4.

3.3.1 Methods for Categorical Attributes

Table 2: Artificial training data tuples

Class	A	B	B	C	A	A	A	C	C
Income	100	87	90	104	120	111	156	56	78
Gender	M	M	F	M	M	F	F	F	M
Credit	10	7	8	2	9	8	7	5	3

When the *group-by attribute* is categorical, we split the original bar of *source attribute* into several sub-bars where the number of sub-bars is determined by the number of distinct values of the *group-by attribute*. Given the artificial training data tuples shown in Table 2, *Income* is a numerical attribute, *Gender* is a categorical attribute and *Class* is the classification label. If we need to find the relationship between *Income* as the *source attribute* and *Gender* as the *group-by attribute*, we should divide the *Income* attribute entry list into two sub-bars, one for Male and the other for Female, which are both attribute values of *Gender*. The generated sub-bars are illustrated in Table 3. Then we may use the same bar visualization technique to draw these created sub-bars.

3.3.2 Methods for Numerical Attributes

When the *group-by attribute* is numerical, we should use a binary split of the form $A \leq v$, where v is a real number, to divide the *source attribute* bar graph into two sub-bars. For example, if we need to find the relationship between *Income* as the *source attribute* and *Credit* as the *group-by attribute*, we should apply a inequality $A \leq v$ to divide the *Income* attribute entry list into two sub-bars, one for the case where *Credit* value less than or equal to v and the other larger than v . If v is chose as 7, the generated sub-bars are illustrated in Table 4. Figure 1 sketches the visualization of attribute relationship for the DNA training data from the STATLOG benchmark [13] which has only categorical attributes.

In order to find the attribute relationship between ATT103 and other three attributes, we take the values of ATT90, ATT91 and ATT93 to split the first bar graph respectively by using attribute relationship visualization techniques. Because all these attributes are categorical attributes with values 0 or 1, the group size of all induced sub-bars is two.

3.4 Panda Index

Table 5 gives two interesting patterns extracting from bar graph where different shades represents different class labels. We argue *Pattern B* is more useful than *Pattern A*, because we can easily find a split point, which is located on the boundary between the fifth cell and the sixth cell. In order to help users to evaluate the created bar graph, we design a novel function, called *Panda Index*, to evaluate the purity of bar graph where the stronger the relationship two attributes have, the purer the created sub-bar graphs will be. Suppose L is the length of the generated bar graph. The *panda index* for a bar graph is defined as:

$$Panda\ Index = \sum_{i=1}^L W_i$$

where W_i is the weight of each cell. We define the cell weight as one less than the number of contiguous cells with same class label i.e. same color. We give an example for computing cell weight in Table 6. The computation complexity of *Panda Index* is $O(n)$ where n is the length of the bar graph. So the panda index for pattern A is 0 and pattern B is 18. Because these numbers represent purity by counting the cell weight, the larger the number, the better.

Table 5. Two Interesting Patterns

Pattern A									
Pattern B									

Table 6. Cell Weight Computation

Pattern A								
Cell Weight	0	0	0	0	0	0	0	0
Pattern B								
Cell Weight	0	3	3	3	3	2	2	2

Given a group of sub-bar A which consists of M sub-bar graph $\{S_1, S_2, S_3, \dots, S_m\}$, the corresponding formula for panda index is defined as:

$$Panda\ Index(A) = \sum_{i=1}^M \frac{|S_i|}{|A|} P_i$$

where $|S_i|$ represents the number of data sample which sub-bar S_i contains, $|A|$ represents the number of data samples in training datasets and P_i represents the panda index of sub-bar S_i .

4. JRV INTERFACE OVERVIEW AND EXPERIMENT RESULTS

Figure 3 illustrates screen shots of the JRV system when representing the DNA training data from STATLOG datasets. The main panel, located in the center of the window, displays the visualization for the specified attribute groups which are defined by

source attributes and group-by attributes chosen in the bottom-left panel. In the left panel of the window, the *JRV-System* tree illustrates several values of evaluation functions for indicated attributes. Users can use the left-bottom panel to interactively act with JRV system where the right-bottom panel prints system information, such as the summary information for values from evaluation functions. JRV takes in flat file data in Comma Separated Value format. Each line of the file should be a list of attribute values separated by commas. Through interactively operating the JRV system, users can find useful information among raw data and save the interesting and useful patterns as image files for later usage. We implement the JRV system in java and build the whole system with JDK 1.4.2. JRV system integrate both some existing algorithms, such as information gain, gain ratio, gini index, SLIQ and SLIQext and our new model, panda index. For evaluation purpose, we choose well-known benchmark datasets as our training data. All experiments are proceeded in a Pentium 4 1.52GHz with 512 MB main memory.

Table 7 presents some panda index values obtained from training data of DNA sequence from STATLOG, which contains 2000 records where each record consists of 180 attributes. Because it is not meaningful to compute the attribute relationship with itself, the table diagonal has no experiment results. The shaded cell indicates the maximum panda index in each row.

5. CONCLUSIONS

In this paper, we introduce the 'attribute relationship' visualization technique as an approach for extracting useful knowledge from underlying datasets. Using our technique, users can obtain several evaluation function values interactively, represent large amount of multi-dimension data and find useful relationship among attributes. Our experiment show that the Panda Index can well represent the attribute relationship by visualization technique. In our future work, we will try to use panda index to help users to do classification tasks which is another import issue in data mining research.

Table 7. Experiment result of panda index for DNA training data

		Group-by Attributes									
		A85	A86	A87	A88	A89	A90	A91	A92	A93	A94
Source Attributes	A85		4557	4743	5667	5284	7359	9315	10099	5980	37468
	A86	3655		3149	2810	4100	7366	2990	3121	1446	73019
	A87	3731	3133		2902	3611	6178	3038	3047	2082	33058
	A88	5463	5004	3877		6567	6147	4889	3968	1046	93944
	A89	8926	7109	9600	10963		10197	6886	6150	2819	28471
	A90	9215	9671	11716	10625	7484		13627	12006	5728	512068
	A91	17406	2448	2699	3221	3363	12933		2398	3207	2066
	A92	20555	3048	2995	2943	3889	12511	2529		2874	2779
	A93	72988	5903	9552	5582	9121	54964	4838	3318		4246
	A94	12380	3034	2962	3269	3537	13931	2275	2135	2965	

Visualization of attribute relationship for DNA training data from STATLOG

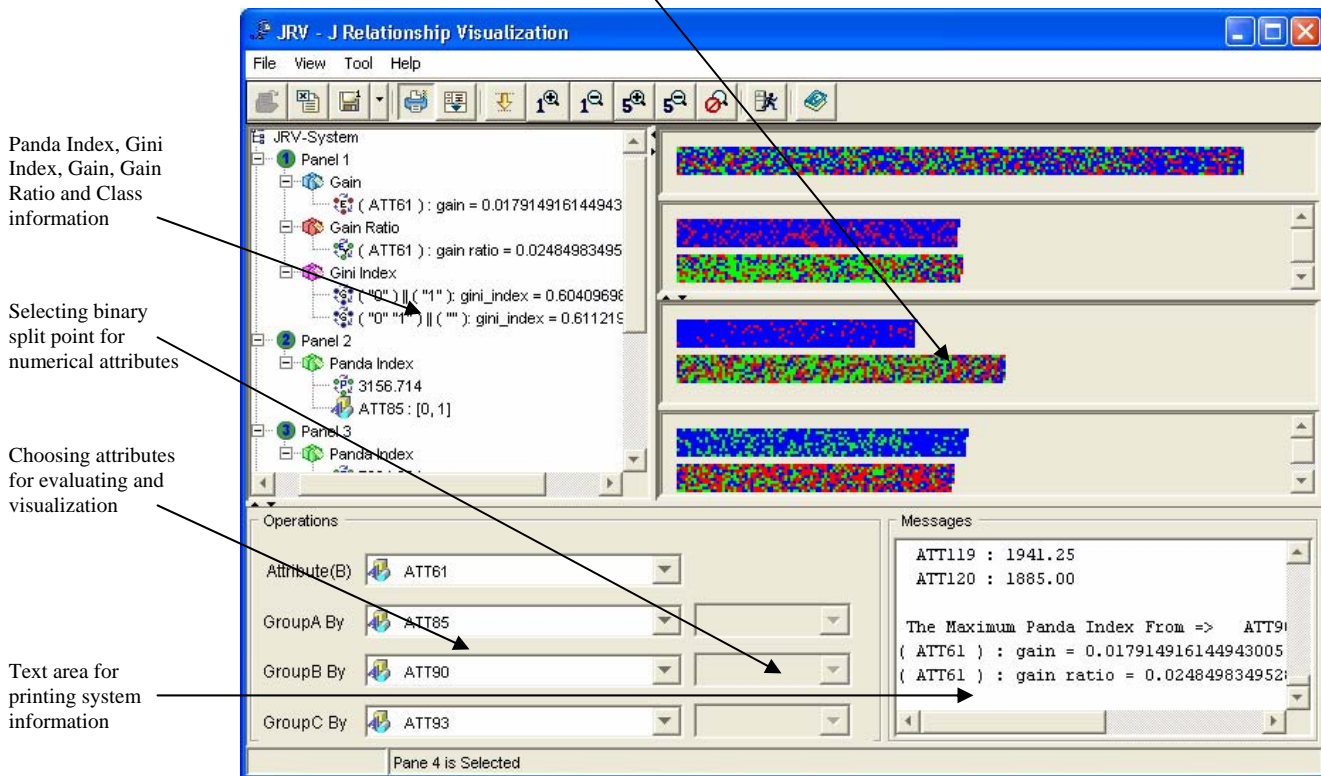


Figure 3. Screen Shots of the JRV System

6. REFERENCES

- [1] Berchtold S., Jagadish H.V., Ross K.A.: "Independence Diagrams: A Technique for Visual Data Mining", Proc. 4th Intl. Conf. on Knowledge Discovery and Data Mining (KDD'98), New York City, 1998, pp. 139-143.
- [2] Coppersmith D., Hong S.J., Hosking J.R.M.: "Partitioning Nominal Attributes in Decision Trees", Data Mining and Knowledge Discovery, an International Journal, Kluwer Academic Publishers, Vol.3, 1999, pp. 197-21
- [3] E. Kandogan. Visualizing Multi-Dimensional Clusters, Trends, and Outliers using Star Coordinates. Proc. ACM SIGKDD '01, pp. 107-116, 2001.
- [4] Fayyad, U., Piatetsky-Shapiro, G., Smyth, P., The KDD process for extracting useful knowledge from volumes of data. Communications of the ACM 39, 11.
- [5] J. Han and M. Kamber. Data Mining: Concepts and Techniques. Morgan Kaufmann, 2000.
- [6] J. Ross Quinlan. C4.5: Programs for Machine Learning. Morgan Kaufman, 1993.
- [7] L. Breiman et. al. CART, Classification and Regression Trees. Wadsworth, Belmont, 1984.
- [8] 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.
- [9] M..Ankerst, Keim D. A. and Kriegel H.-P.: "Circle Segments: A Technique for Visually Exploring Large Multidimensional Data Sets", Proc. Visualization '96, Hot Topic Session, San Francisco, CA, 1996.
- [10] Data Sets", Proc. Visualization '96, Hot Topic Session, San Francisco, CA, 1996.
- [11] 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.
- [12] Mehta M., Agrawal R., Rissanen J.: "SLIQ: A Fast Scalable Classifier for Data Mining", Proc. of the Int. Conf. on Extending Database Technology (EDBT '96), Avignon, France, 1996.
- [13] Michie D., Spiegelhalter D.J., Taylor C.C.: "Machine Learning, Neural and Statistical Classification", Ellis Horwood, 1994.
See also <http://www.ncc.up.pt/liacc/ML/statlog/datasets.html>.
- [14] Paterson, A., and Niblett, T.B. (1982). ACLS Manual. Edinburgh: Intelligent Terminals Ltd.
- [15] Quinlan, J. R. (1986). Induction of decision trees. Machine Learning, 1, 81-106.