Extend Table Lens for High-Dimensional Data Visualization and Classification Mining

CPSC 533c, "Information Visualization" Course Project, Term 2 2003

Fengdong Du *fdu@cs.ubc.ca*

University of British Columbia

Abstract:

Data mining and information visualization apply different techniques to solve the same problem of extracting useful information hidden in large amount of data. In this project, we focus on classification problem and developed an integrated information visualization environment, which plugs in classification rule-mining methods and extends the Table Lens techniques to handle high-dimensional data. This environment allows user to easily view the details of particular data tuples and in the meantime maintains an overview of the entire dataset. In this environment, users can flexibly choose interested subsets of data or subsets of attributes and interactively perform classification mining.

1. Introduction:

1.1 Classification Problem:

Given a set of training data, the classification problem is to generate a set of rules that can classify data tuples with high accuracy. Each training data tuple has a set of attributes. And one of them is called the class attribute and indicates which class the data tuple belongs to. An example of such kind of train dataset is shown in table 1.

Outlook	**Temperature**	**Humidity**	**Windy**	**Play_tennis**
sunny	hot	High	false	n
overcast	hot	High	false	y
rain	cool	Normal	true	n
sunny	cool	Normal	false	y

Table 1

Among several classification methods proposed over the years, ID3 decision tree algorithm is one of the most popular approaches because it not only presents good prediction accuracy but also provides the reasoning of classification. The possible classification rules produced from this training dataset using ID3 decision tree algorithm are shown in table 2.

Rule 1: Outlook = overcast ➔ Play_tennis = yes
Rule 2: Outlook = rain ➔ Play_tennis = no
Rule 3: Outlook = sunny and Temperature = hot ➔ Play_tennis = no
Rule 4: Outlook = sunny and Temperature = cool ➔ Play_tennis = yes

Table 2

One critical problem with the ID3 algorithm is that it needs to scan and split the training sample space multiple times until the entire tree is pure on class labels or all non-class attributes are exhausted. During the mining process, the splitting occurs for all remaining attribute lists because we don't know which attribute will be chosen to construct the rule after the splitting. And unfortunately, many of the attributes in the splitting are actually irrelevant to the final classification rules. As an example, the Humidity attribute and Windy attribute in the above train data example do not appear in any of the final rules and therefore are irrelevant to classification. However, these two attribute lists are split as well in the mining process together with those relevant attribute lists.

For high-dimensional large training dataset, this problem becomes even worse because of the I/O cost. One approach to solve this problem is to rely on the domain knowledge about the target dataset and remove some attributes that are believed irrelevant. Obviously, the reliability of domain knowledge highly affects the final classification rules.

1.2 Table Lens Techniques:

Table Lens is an information visualization technique for visualizing large relational dataset. The main advantage of this technique is "focus+context". Namely, users can view the details of an arbitrary data tuple and also keep displaying the overall view of the entire dataset. Focus changing and the consequent graphical re-presentation of the information structure are achieved through dynamic interactions and graphical distortions. Users can also change the graphical layout of the data tuples by sorting them according to a particular attribute. These techniques successfully support data exploration process and allow data analysts to visually detect the underlying relationship, e.g. correlation, association, among various attributes.

On the other hand, the original Table Lens technique is not capable of handling high-dimensional data. Given a relatively small screen size and datasets of high dimensionality (i.e. more than one hundred attributes), even though the Table Lens applies "fisheye view" kind of techniques and distorts the graphical representation in horizontal direction, it cannot accommodate the whole attributes in one screen view.

Another weakness of Table Lens, and also perhaps for all other data exploration approaches that rely on visual detection of data patterns, is that it is difficult to discover complicated patterns, e.g. the relationship between an attribute and the combination of several other attributes. And in addition, it is also not clear on the certainty of the patterns visually found by data analysts. Here, the key problem is that visual detection is the only tool in these kinds of systems. And obviously, visual detection is less powerful and less accurate than well-developed data mining methods.

1.3 Relationship Between Table Lens and Classification Mining:

The Table Lens technique works on relational datasets. The graphical representation it produced is a distorted table. Each row of this table represents a data tuple and each column represents an attribute. In other words, this technique produces graphical mappings for relational data tables or also called "cases-by-variable" table [1]. The classification mining methods take relational training dataset as input and tell users sort of dependency relationship between non-class attributes and class attribute. Namely, if some non-class attribute variables are bounded with particular values, the class attribute variable is bounded with particular class labels. And because the Table Lens technique produces graphical mapping of relational tables, the dependency relationship exists in the visual encoding of these relational tables.

The Table Lens technique provides interactive layout controls on the target dataset. The most important tool is to sort the data tuples by a particular attribute. This will allow users to directly detect some simple dependency relationship among attributes. As a concrete example, the sample data shown in table 1 has the following visual representation using Table Lens technique. The simple classification rules listed in table 2 can be visually detected directly from this graphical representation.

Figure 1

For complicated patterns, even though the dependency relationship still exists in the Table Lens encoding, users may not be able to detect it visually because of the lack of corresponding graphical layout controls. In this case, classification-mining methods can be plugged in the Table Lens environment to build a more powerful information extraction tool.

1.4 Extension of Table Lens:

Motivated by the above observations, we try to extend the Table Lens technique in two aspects in this project. First, we design a visual encoding for high dimensionality and therefore completely present the overall context of the high dimensional data within one screen space. We allow users to easily move one interested attribute from the context view to the detailed view, and therefore be able to see the details of this attribute. Second, we plug classification-mining methods into the integrated Table Lens visualization environment and solve the classification rule-mining problem in an interactive and exploratory process. In this environment, domain data analysts can first examine the overall context of the entire dataset and very likely obtain some idea on the potential patterns by viewing the overall context. This step will enrich the prior domain knowledge regarding the target dataset. Then, on basis of the prior knowledge, domain analysts can easily define a subset of interested data or subset of interested attributes and conduct classification mining. Since the mining process will only focus on the interested subset of data, we can successfully avoid many unnecessary computations and also significantly improve the goodness of output classification rules.

2. Related Work

To our best knowledge, the most directly related work to this project is Ankerst [2] [3]. Ankerst [2] [3] proposed a visual classification approach based on decision tree classifier. It maps each attribute value to a colored pixel, shows different attribute lists in the different areas of the screen, and yields a global view of large amount of data. The common idea between Ankerst [2] [3] and this project is to enable human involvement in the mining process. On the other hand, Ankerst [2] [3] relies on users to perceive the data pattern and control the mining process, e.g. choosing the best attributes to split the sample space; while this project focuses on high-dimensional data, presents the complete global view and detailed view simultaneously to enrich the prior knowledge of domain analysts, and then allow domain analysts to easily choose a subset of relevant attributes perceived from the context view and conduct classification mining. Another difference between Ankerst [2] [3] and this project is that in Ankerst [2] [3], attribute lists are visualized separately and independently; while in this project, we apply Table Lens as the visual encoding technique and preserve the unity of data tuples.

Rather than focusing on a particular data-mining task, Kreuseler [4] considers data mining and information visualization as a general integrated problem and proposed a framework for Visual Data Mining. The framework suggests an information-preprocessing step before performing the actual visualization. Various preprocessing and visualization techniques are discussed for different data exploration tasks.

Spenke [5] is an interactive user interface to view "cases-by-variable" tables. Compared to Table Lens, it has a query mechanism to support simple queries. On the other hand, it is dedicated for relatively small spreadsheet dataset and does not produce perceptible visual representation for large amount of data. Therefore, it can be logically categorized as a user interface project rather than an information visualization work.

3. Table Lens/Classification Mining Integrated Environment

3.1 High Dimensionality Encoding:

The original Table Lens technique applies color and position redundant encoding in the context view. This requires a minimum table cell space to achieve good perception. For high-dimensional data, this redundant encoding cannot fit all attributes in one single view. On the other hand, we usually want to maintain a complete and perceptible context view and avoid scrolling over many screen displays.

The fundamental task we want to complete in the context view is to show the distribution of the domain values of a particular attribute, given the target dataset is presented in some graphical pattern, e.g. sorted by another attribute. Therefore, the main purpose of visual encoding is to distinguish the different domain values. Here, color encoding could be sufficient to achieve this purpose and redundant encoding seems do not give us more expressive power. As a concrete example, let's consider the sample data set in section 1.1. Assume the dataset has the pattern: if outlook is overcast then playTennis is yes.

Let's also assume outlook=overcast is encoded as a blue bar and playTennis=yes is encoded as a yellow bar. In this case, when we sort the data tuples by outlook attribute, we can see the correspondence between the blue bars in outlook column and yellow bars in the playTennis column. Color encoding is sufficiently effective to show the correlation between these two attributes. In this project, we only use color encoding in the context view and therefore are able to accommodate as many as several hundreds attributes into one single context view. The geometric representation of an attribute value is a rectangle bar. When we fit high-dimensionality data into the global context view, the rectangle bars become colored pixels.

3.2 Focus Versus Context Views

To extend the Table Lens technique for handling large amount of high-dimensionality data, we divide the entire screen space into two parts. The right part is used to show the overall context view and the left part is used to show the detailed view. And the size of these two views can be adjusted by users. The layout of these two views is shown in figure 2.

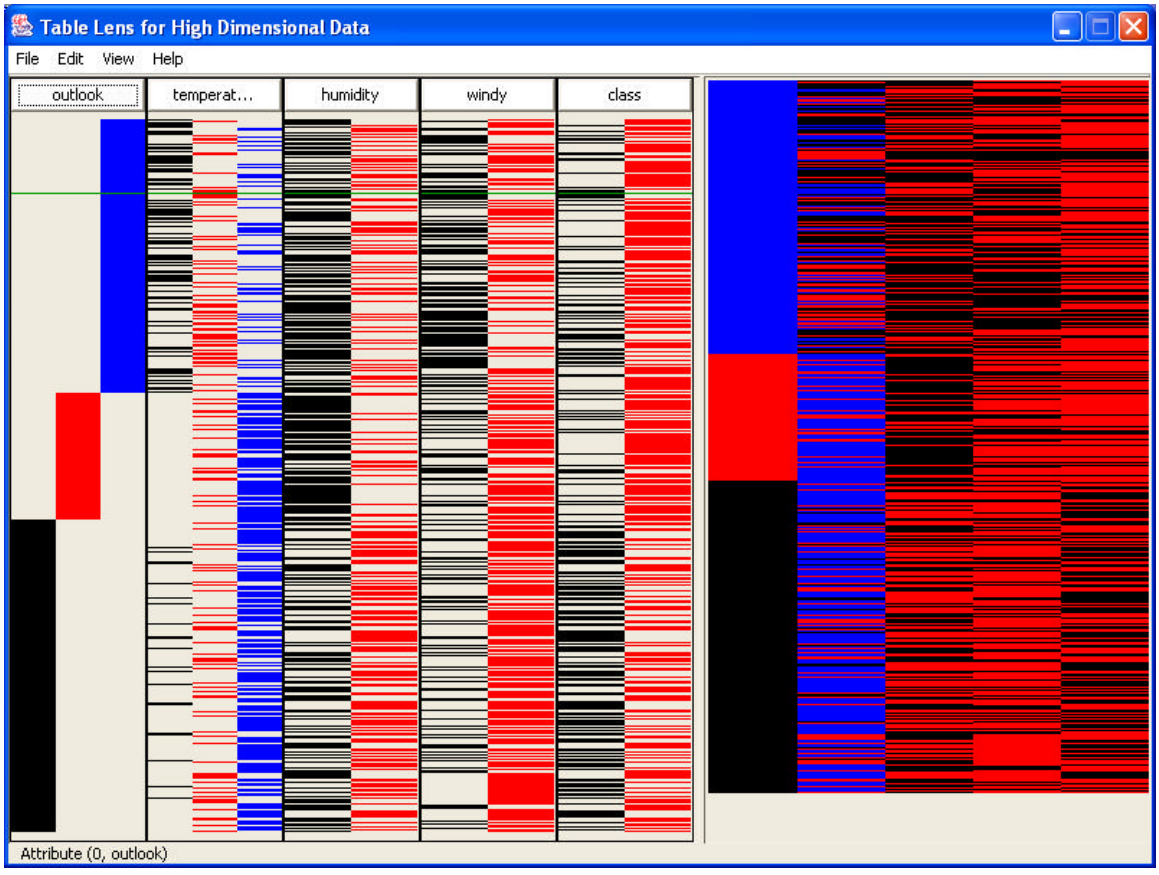


Figure 2

In the detailed view, we apply same visual encoding techniques used in the original Table Lens. Users can click a column head button and sort the entire dataset in ascending order or descending order of attribute values in this column. Then, the consequent overall

context view will automatically update. Users can also bring an attribute from the context view to the detailed view. This is done by first clicking an attribute in the context view, and then moving the mouse to a column in the detailed view and clicking the column head button. These interaction operations allow users to focus on an arbitrary attribute and see the details easily. Moreover, domain analysts can examine how the overall context view changes after sorting a particular attribute and very likely get better idea regarding the relationships of different attributes. This prior knowledge could be very important in classification mining.

3.3 Defining Interested Regions for Classification Mining:

Another useful interaction provided in the integrated environment is that users can define their interested regions in the context view and then perform classification mining on a particular region or the union of a set of regions. Because we eliminate the irrelevant attributes before the mining process, this approach successfully avoids the problem we described in section 1. As a consequence of defining the interested regions, we also restrict the classification rules that are output from our integrated environment to be interested rules. In other words, our integrated visualization environment can play the similar role of pruning uninterested rules in those non-visual data mining systems. And furthermore, in our integrated environment, domain analysts can obtain multiple sets of classification rules by defining different interested regions depending on their perceptions. On the contrary, non-visual classification mining systems do not have this kind of flexibility. Domain analysts do not have many controls on the mining process and the final output rules.

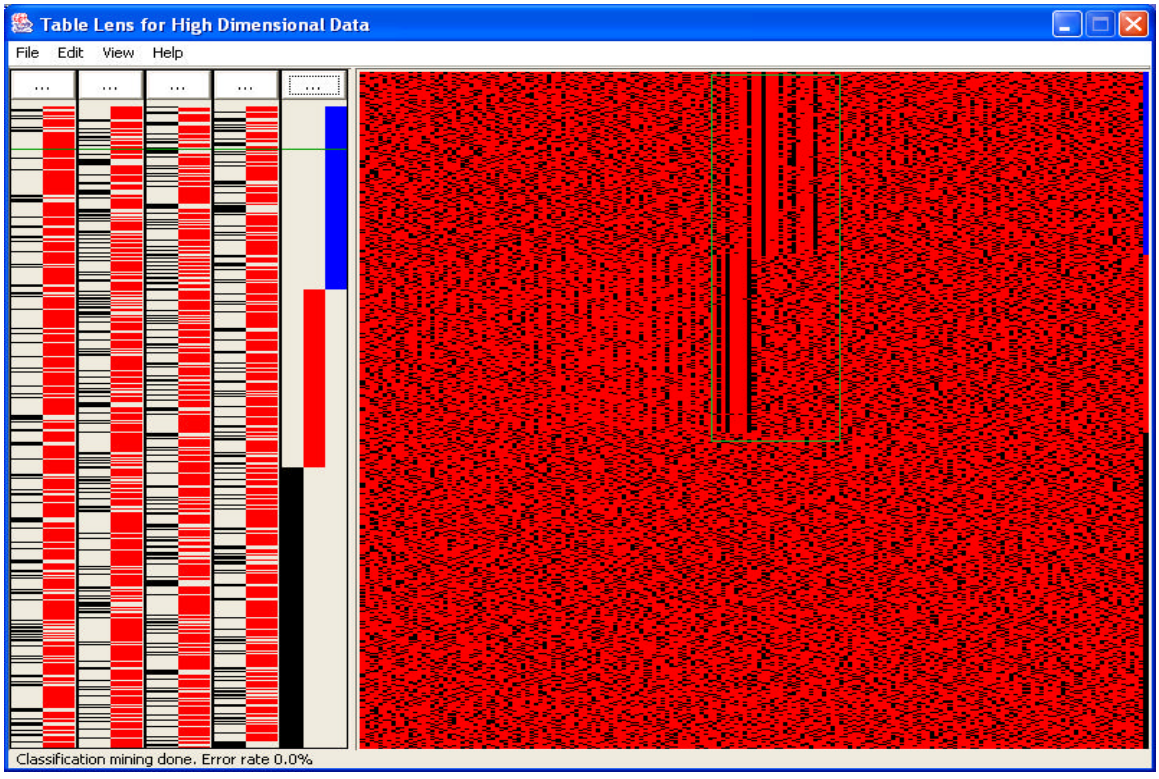


Figure 3

3.4 A Classification Mining Scenario:

In figure 3, we present a typical scenario of classification mining in our integrated visual data-mining environment. The dataset we chose here is a DNA sequence dataset that has 181 attributes including a class attribute that has three domain values. We fit 800 tuples in both the context view and the detailed view. When we sort the class attribute, we can see the pattern in the context view from the 80^{th} attribute to the 105^{th} attribute. Then we drag a rectangle area in the context view and define this interested region to perform classification rule mining. We click the classification menu item under the edit menu. Very quickly, we can see the mining error rate 0.0% in the bottom status bar. On the other hand, the running time performance of our implementation of the tradition decision tree algorithm is uncompetitive to this interactive visual classification-mining environment. And the prediction error rates of these two approaches on the same test dataset are almost same.

4. Implementation and Results:

This project is implemented in Java using standard Swing and AWT package. Graphics and Graphics2D are used to display all graphics components. To improve the graphics computation performance, all geometric shapes take integer parameters to define their sizes and positions. In the overall context view, because we allow user to adjust the width of the view, the number of available pixels is not necessarily dividable by the total number of attributes. To solve this problem, we make compensation for every pixel loss. The minimum width of the visual encoding of an attribute is 1 pixel. If the context view cannot accommodate all attributes with this minimum encoding width, only part of the attributes are displayed. In this case, users have to adjust the view size to obtain a complete display. Users can move the mouse to the right most attribute column showed in the context view, the corresponding attribute index and name will be displayed in the status bar.

Our current version of implementation works for the datasets with purely categorical attribute values. Because we use color encoding in the overall context view, to better perceive the graphical representation of the dataset, we restrict the maximum domain size of all attributes being 11. In many data mining applications, we typically have relatively small number of discrete values for categorical attributes. For numerical attributes or categorical attributes with large domain cardinality, we can partition the domain into small number of buckets and encode the different buckets. In practice, binary partition has been showed good accuracy using decision tree algorithm. Therefore, our encoding restriction is particularly suitable in classification mining and does not pose obstacle of using our integrated visual mining environment for more general cases.

5. Summary:

In this project, we extend the Table Lens techniques to handle high-dimensional datasets. Having observed that Table Lens produces graphical mapping of relational dataset and also encodes many hidden patterns existing in the dataset, we suggest using Table Lens as the visualization tool for classification mining problem. By plug-in classification mining methods, users can obtain more certainty and accuracy in addition to visual perceptions.

And many complex patterns that cannot be visually detected in the original Table Lens technique can now be discovered as classification mining problem.

On the other hand, the extended Table Lens visualization tool can be used as an important step before performing the classification mining computations. In this step, domain analysts can perceive the overall context and details of the target dataset very flexibly, and therefore enrich their prior knowledge. On basis of the prior knowledge, domain analysts have more controls on the mining process and therefore can get interested results very quickly. We provide interactive tools in this Table Lens/Classification Mining integrated environment to facilitate both the pattern perception and classification mining control.

Reference:

[1] R. Rao and S. Card, "The Table Lens: Merging Graphical and Symbolical Representations in an Interactive Focus+Context Visualization for Tabular Information", *Proc. Conf. Human Factors in Computing System*, Apr. 1994.

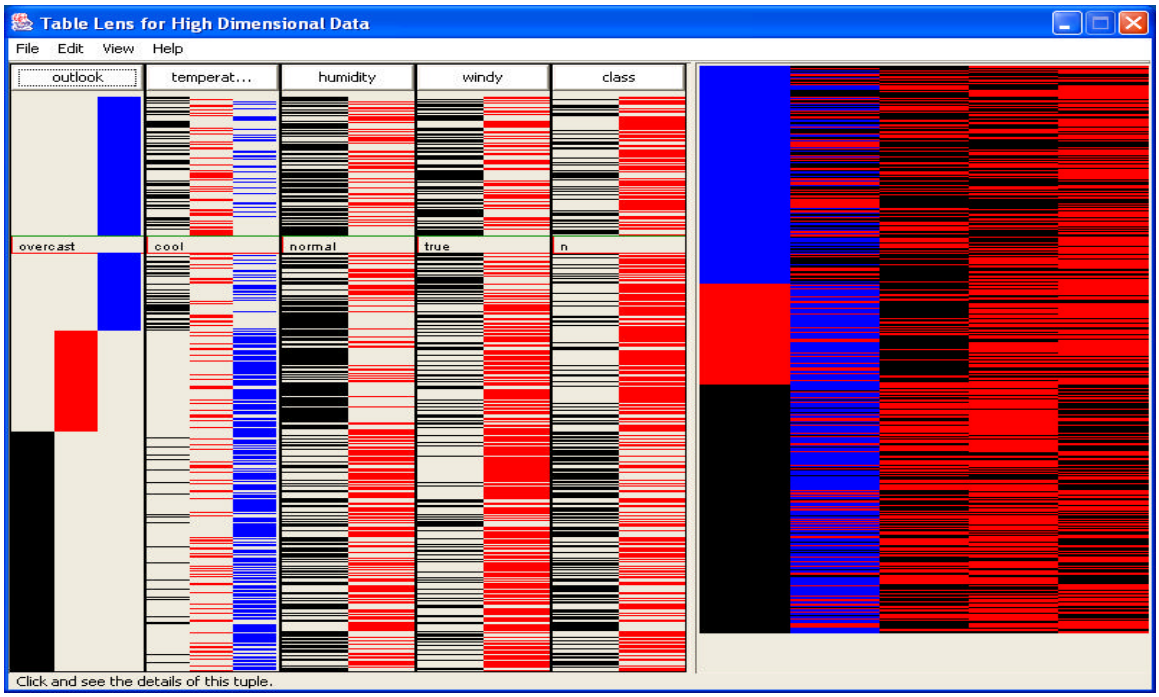
[2] M. Ankerst, C. Elsen, Ester M., Kriegel H.-P.: "Visual Classification: An Interactive Approach to Decision Tree Construction", *Proc. 5th Int. Conf. on Knowledge Discovery and Data Mining (KDD'99)*, San Diego, CA, 1999, pp. 392-396.

[3] M. Ankerst, "Visual Data Mining with Pixel-Oriented Visualization Techniques", *Proc. Workshop Visual Data Mining*, 2001

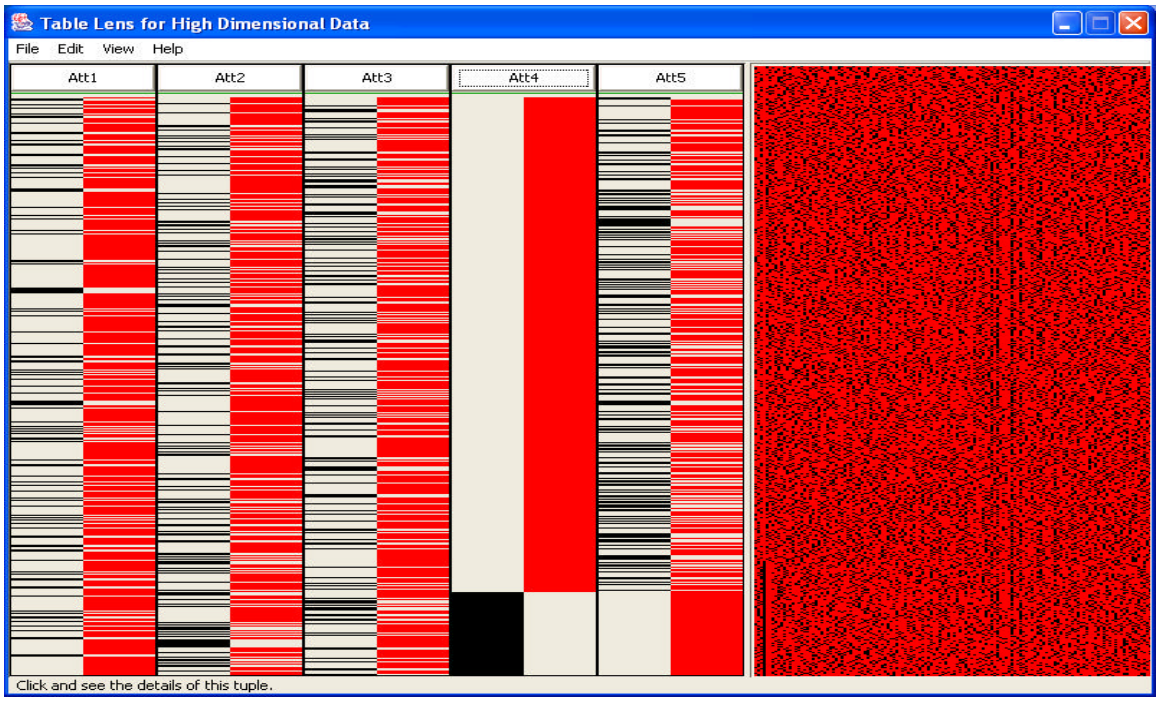
[4] M. Kreuseler, H. Schumann, "A Flexible Approach for Visual Data Mining", *IEEE Transactions on Visualization and Computer Graphics*. Vol. 8, No. 1, January-March 2002.

[5] M. Spenke, C. Beilken, and T. Berlage, "FOCUS: The Interactive Table for Product Comparison and Selection", *Proc. Ninth Ann. ACM Symp. User Interface Software and Technology*, 1996

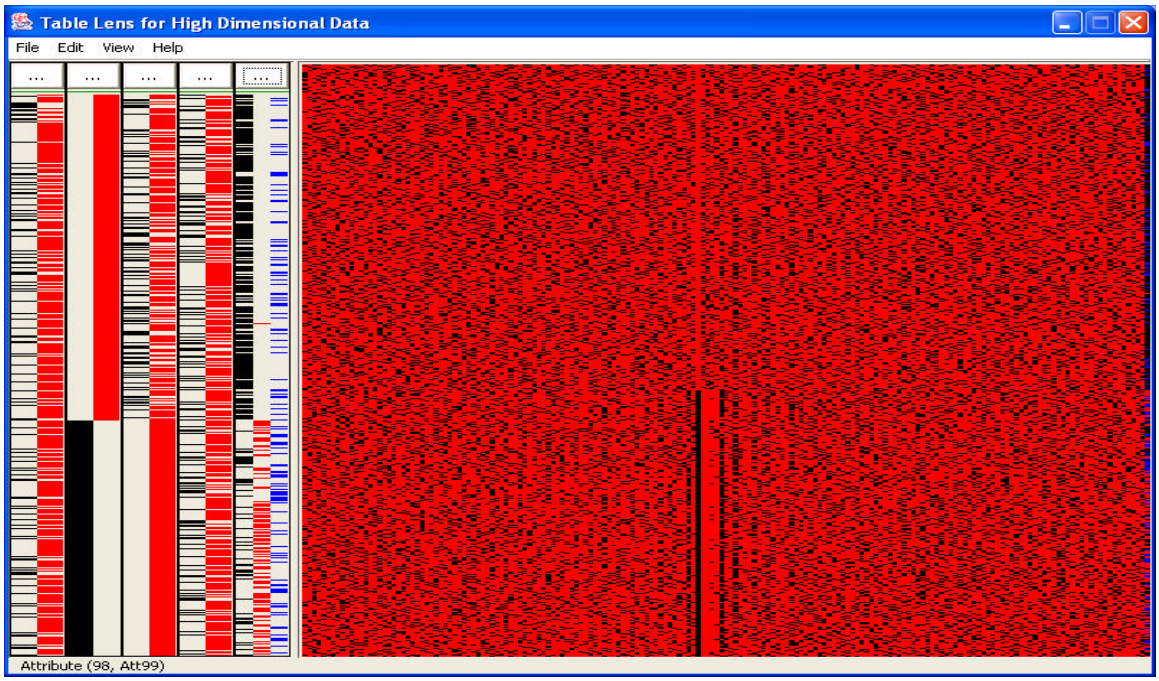
[6] M. Sarkar and M. Brown, "Graphical Fisheye Views", *Comm. ACM*, vol. 37, no. 12, pp. 73-84, Dec. 1994



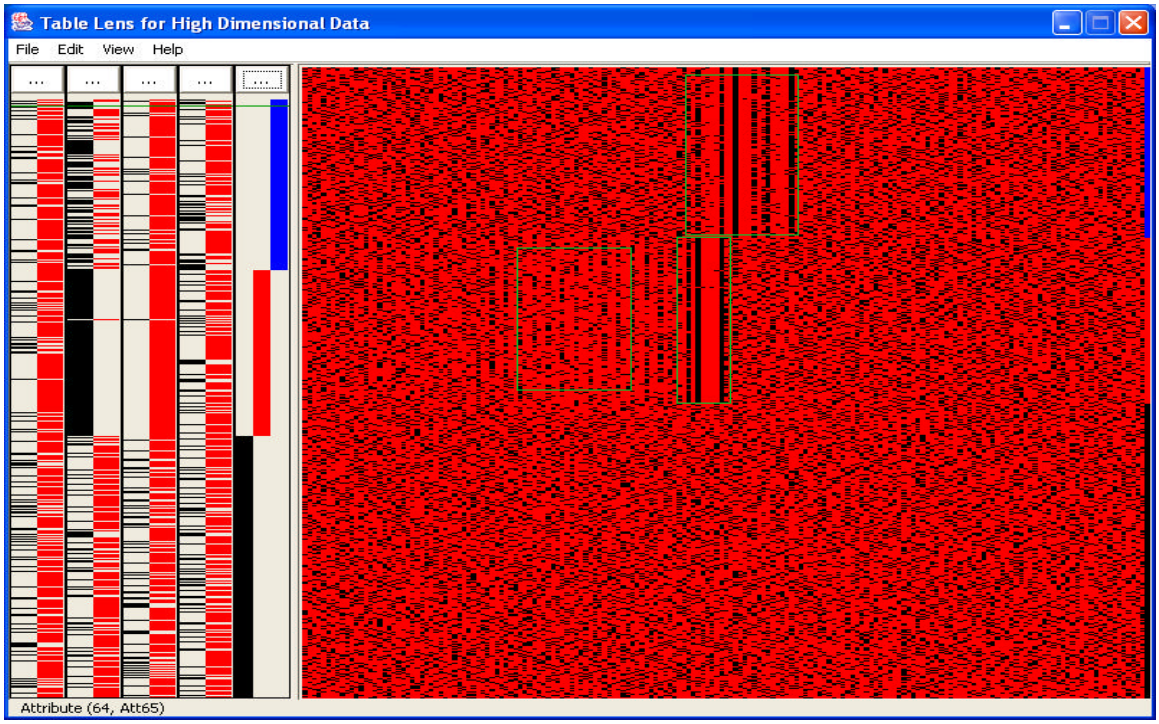
Snapshot 1: Showing detailed attribute values



Snapshot 2: Correspondence between the Detailed View and Context View



Snapshot 3: Bringing an Attribute from the Context View to Detailed View



Snapshot 4: Define Interested Classification Mining Regions