Chapter 7
Visual Data Mining and Discovery with Binarized Vectors

Boris Kovalerchuk[1], Florian Delizy[1], Logan Riggs[1], and Evgenii Vityaev[2]

[1] Dept. of Computer Science, Central Washington University, Ellensburg, WA, 9896-7520, USA
[2] Institute of Mathematics, Russian Academy of Sciences, Novosibirsk, 630090, Russia

Abstract. The emerging field of Visual Analytics combines several fields where Data Mining and Visualization play leading roles. The fundamental departure of visual analytics from other approaches is in extensive use of visual analytical tools to discover patterns not only to visualize pattern that have been discovered by traditional data mining methods. High complexity data mining tasks often require employing a multi-level top-down approach, where first at the top levels a qualitative analysis of the complex situation is conducted and top-level patterns are discovered. This paper presents the concept of Monotone Boolean Function Visual Analytics (MBFVA) for such top level pattern discovery. This approach employs binarization and monotonization of quantitative attributes to get a top level data representation. The top level discoveries form a foundation for next more detailed data mining levels where patterns are refined. The approach is illustrated with application to the medical, law enforcement and security domains. The medical application is concerned with discovering breast cancer diagnostic rules (i) interactively with a radiologist, (ii) analytically with data mining algorithms, and (iii) visually. The coordinated visualization of these rules opens an opportunity to coordinate the multi-source rules, and to come up with rules that are meaningful for the expert in the field, and are confirmed with the database. Often experts and data mining algorithms operate at the very different and incomparable levels of detail and produce incomparable patterns. The proposed MBFVA approach allows solving this problem. This paper shows how to represent and visualize binary multivariate data in 2-D and 3-D. This representation preserves the structural relations that exist in multivariate data. It creates a new opportunity to guide the visual discovery of unknown patterns in the data. In particular, the structural representation allows us to convert a complex border between the patterns in multidimensional space into visual 2-D and 3-D forms. This decreases the information overload on the user. The visualization shows not only the border between classes, but also shows a location of the case of interest relative to the border between the patterns. A user does not need to see the thousands of previous cases that have been used to build a border between the patterns. If the abnormal case is deeply inside in the abnormal area, far away from the border between "normal" and "abnormal" classes, then this shows that

D.E. Holmes, L.C. Jain (Eds.): Data Mining: Found. & Intell. Paradigms, ISRL 24, pp. 135–156, 2011.
springerlink.com

this case is very abnormal and needs immediate attention. The paper concludes with the outline of the scaling of the algorithm for the large data sets and expanding the approach for non-monotone data.

Keywords: Data Mining, Visual discovery, Monotone chains, Multi-level Data Mining, Monotone Boolean Function, Visual Analytics.

1 Introduction

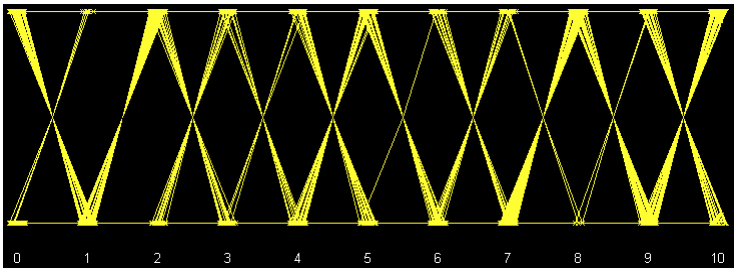
Visual data mining (VDM) assists a user in detecting interesting knowledge, and in gaining a deep visual understanding of the data in combination with advanced visualization [Beilken & Spenke, 1999; Keim et al., 2002; Schulz, et al, 2006; Badjio, Pouletm 2005; Zhao et al, 2005, Lim, 2009, 2010; Oliveira, Levkowitz, 2003, Pak, Bergeron, 1997; Wong et al, 1999,]. *Visualizing the border* between classes is one of the especially important aspects of visual data mining. The well-separated classes that are visually far away from each other with simple border between classes match our intuitive concept of the patterns. This simple separation serves as an important support for the idea that the data mining result is robust and not accidental. Moreover, for many situations, a user can easily catch a border visually, but its analytical form can be quite complex and difficult to discover. This visual simple border for a human may not be a simple mathematically.

VDM methods have shown benefits in many areas. However, available methods do not address the specifics of data, with little variability in the traditional visual representation of different objects such as parallel coordinates. VDM is an especially challenging task when data richness should be preserved without the excessive aggregation that often happens with simple and intuitive presentation graphics such as bar charts [Keim, Hao, et al., 2002]. Another challenge is that often such data lack the natural 3-D space and time dimensions [Groth, 1998] and instead require the visualization of an abstract feature.

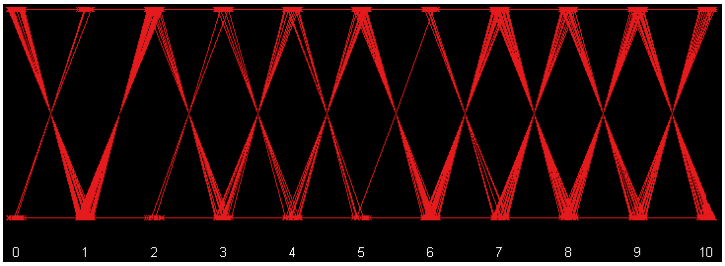
We begin with an analysis of the currently available methods of data visualization. Glyphs can visualize *nine attributes* (three positions *x*, *y*, and *z*; three size dimensions; color; opacity; and shape). Texture can add more dimensions. Shapes of the glyphs are studied in [Shaw, et al., 1999], where it was concluded that with large super-ellipses, about 22 separate shapes can be distinguished on the average. An overview of multivariate glyphs is presented in [Ward, 2002].

In this paper, we show that the placement based on the use of the *data structure* is a promising approach to visualize a border between classes for multidimensional data. We call this the **GPDS** approach (**Glyph Placement on a Data Structure**). It is important to note that in this approach, some attributes are *implicitly* encoded in the data structure while others are *explicitly* encoded in the glyph or icon. Thus, if the structure carries ten attributes and a glyph/icon carries nine attributes, nineteen attributes are encoded. Below to illustrate the use of the data structure concept, we consider simple 2-D icons as *bars* of different colors. Adding texture, motion and other icon characteristics can increase dimensions of the data visualized.

Alternative techniques such as Generalized Spiral and Pixel Bar Chart are developed in [Keim, Hao, et al., 2002]. These techniques work with large data sets without overlapping, but only with a few attributes, (these range from a single attribute to perhaps four to six attributes).



(a) '0'-class (benign)



(b) '1'-class (malignant)

Fig. 1. Breast cancer data in parallel coordinates

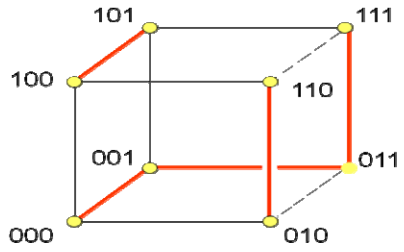
The parallel coordinate visualization [Inselberg, Dimsdale, 1990] can show ten or more attributes in 2-D, but suffers from record overlap and thus is limited to tasks with well-distinguished cluster records. In parallel coordinates, each vertical axis corresponds to a data attribute (x_i) and a line connecting points on each parallel coordinate corresponds to a record. Figure 1 (a)-(c) depicts about a hundred breast cancer cases (each of them is an 11-dimensional Boolean vector in Boolean space E^{11}). Classes '0' and '1' look practically the same as Figures 1 (a) and (b) show. Thus, parallel coordinates were are not able to discover visually the pattern that would separate classes 0' and '1' (benign and malignant) in these dataset. In this paper, we will show that the proposed GPDS method is able to do this.

Parallel coordinates belong to a class of methods that explicitly visualize *every* attribute x_i of an n-dimensional vector $(x_1,x_2,\ldots,x_n)$ and place the vector using *all*

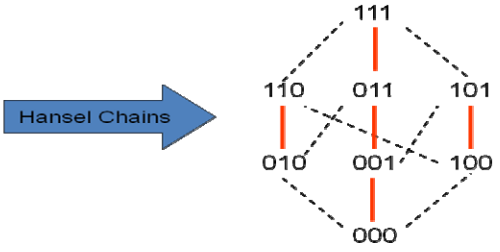
attributes x_i but each attribute is placed on its own parallel coordinate *independently* of the placing other attributes of this vector and other vectors. This is one of the major reasons of occlusion and overlap of visualized data. The GPDS approach constructs a data structure and can place objects using *attribute relations*.

2 Method for Visualizing Data

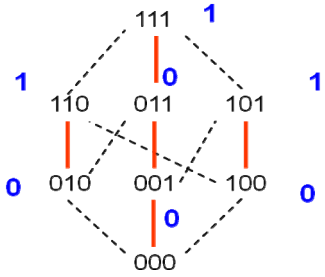
Below **we** describe Monotone Boolean Function Visual Analytics (**MBFVA**) method and its implementation called **VDATMIN** that exploit Glyph Placement on a Data Structure in combination with Monotone Boolean Functions approach. As was discussed above many data mining problems can be encoded using Boolean vectors, where each record is a set of binary values {0; 1} and each record belongs to one of two classes (categories) that are also encoded as 0 and 1. For instance, a patient can be represented as a Boolean vector of symptoms along with an indication of the diagnostic class (e.g., benign or malignant tumor) [Kovalerchuk, Vityaev, Ruiz, 2001, Kovalerchuk et al, 1996]. For Boolean vectors, our VDM method relies on **monotone structural relations** between them in the n-dimensional binary cube, E^n based on the theory of monotone Boolean functions.



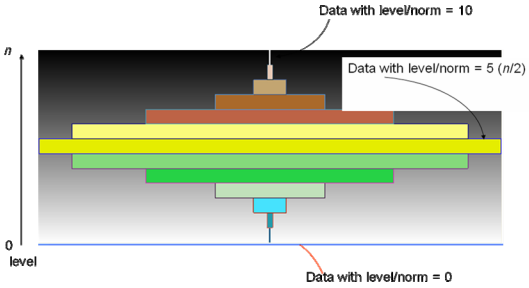
(a) Binary 3-D cube with Hansel chains



(b) Binary 3-D cube with Hansel chains as a lattice



(c) Binary 3-D cube with some nodes labeled by target class "0" or "1"



(d) generalization of (b) for n-dimensional Binary cube

Fig. 2. Visual representation of multidimensional Boolean data

Figure 2(a) illustrates the concept of monotonicity in 3-D Boolean cube, where red lines show three monotone chains (known as Hansel chains [Kovalerchuk et al, 1996; Hansel, 1966]):

chain 1: (000), (001), (011), (111),
chain 2: (100), (101),
chain 3: (011), (110).

In each Hansel chain, each next vector is greater than a preceding vector. In the next vector, exactly one attribute is greater than in the preceding vector. Together these Hansel chains cover the whole 3-D Boolean cube and none of vectors is repeated (chains do not overlap). There is a general recursive process [Kovalerchuk et al, 1996; Hansel, 1966] to construct Hansel chains for a Boolean cube, E^n of any dimension n without overlap of chains.

Figure 2 (b) shows the same Boolean cube as a lattice with Hansel chains drawn in parallel with the largest vector (111) on the top and the smallest vector (000) on the bottom. Figure 2(c) shows the same binary 3-D cube with some nodes labeled by target class "0" or "1". In this way, training and testing data can be shown. Figure 2(d) presents a generalization of the lattice visualization shown in Figure 29b) for n-dimensional Binary cube. This visualization is used for the user interface in the VDATMIN system.

The concept of the monotone Boolean function from discrete mathematics [Korshunov, 2003, Keller, Pilpel, 2009] is defined below. Let $E^n=\{0,1\}^n$ be a binary *n*- dimensional cube then vector $\mathbf{y}=(y_1,y_2,\ldots,y_n)$ is no greater than vector $\mathbf{x}=(x_1,x_2,\ldots,x_n)$ from E^n if for every i $x_i \geq y_i$, i.e.,

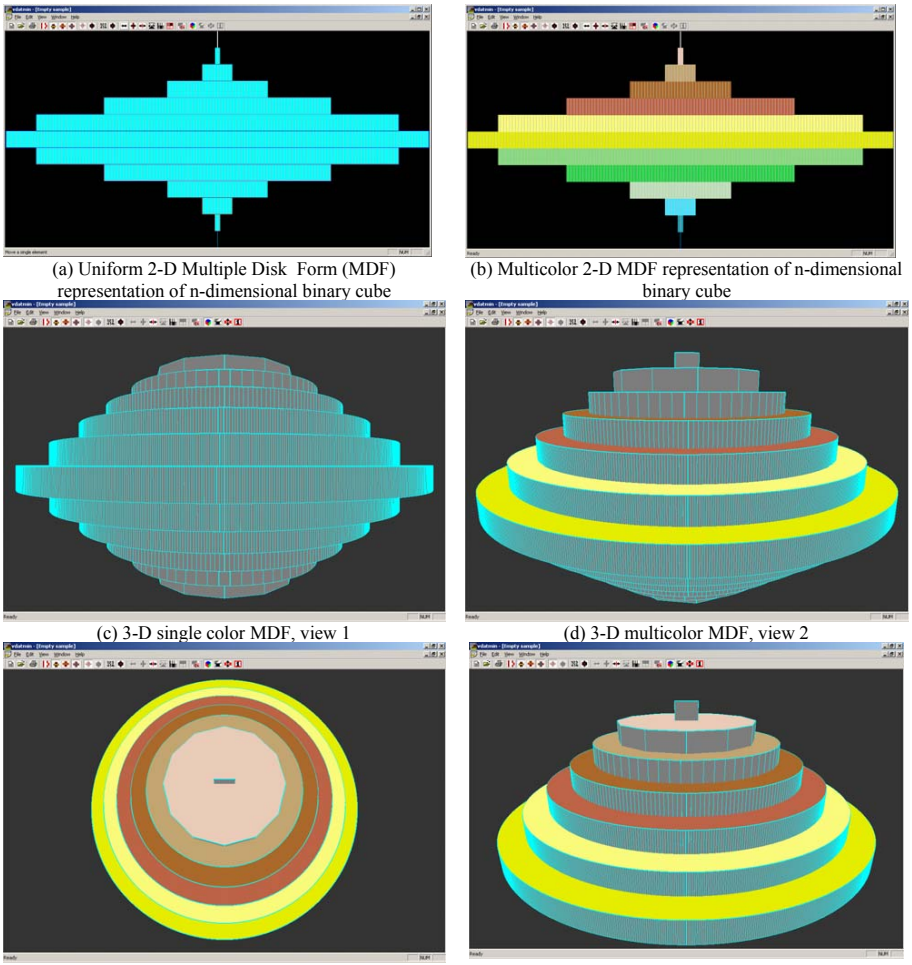
$$\mathbf{x} \geq \mathbf{y} \Leftrightarrow \forall i\ x_i \geq y_i$$

In other words, vectors **x** and **y** are ordered. In general relation $\geq$ for Boolean vectors in E^n is a *partial order* that makes E^n a *lattice* with a max element (1,1,…,1) and min element (0,0,…,0).

Boolean function f: $E^n \rightarrow E$ is called a *monotone Boolean function* if

$$\forall\ \mathbf{x} \geq \mathbf{y} \Rightarrow f(\mathbf{x}) \geq f(\mathbf{y}).$$

Figure 3 demonstrates the user interface of visual data mining system VDATMIN. Figure 3(a) shows all 2^{12}=4096 nodes of n-dimensional binary cube E^n for n=12. Each node (12-dimemtional Boolean vector) is represented as a blue bar. The bar that represents the vector **x** containing all zeros is located in the lowest layer in the middle of the picture. The bar representing the vector **x** that contains all "1" (|x|=12) is located at the top of the picture in the middle. All other bars are located in between.



(a) Uniform 2-D Multiple Disk Form (MDF) representation of n-dimensional binary cube

(b) Multicolor 2-D MDF representation of n-dimensional binary cube

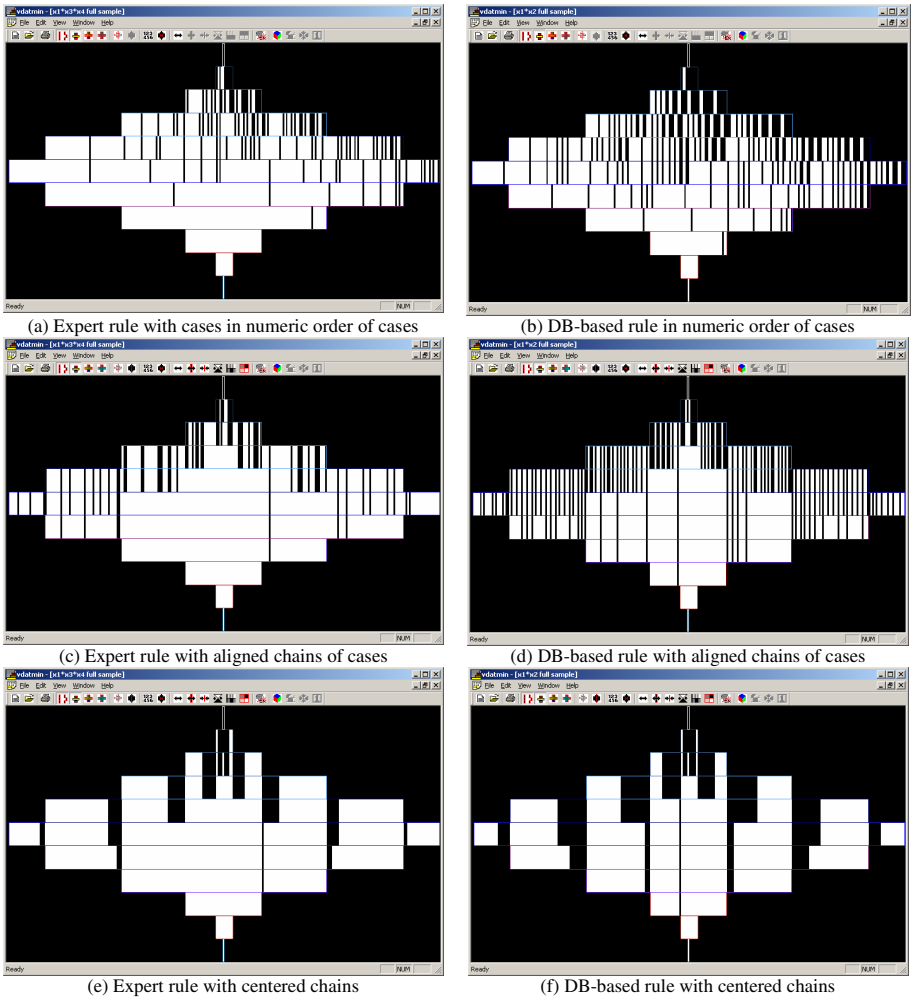
(c) 3-D single color MDF, view 1

(d) 3-D multicolor MDF, view 2

(e) 3-D multicolor MDF, view 3

(f) 3-D multicolor MDF, view 4

Fig. 3. VDATMIN user interface



(a) Expert rule with cases in numeric order of cases

(b) DB-based rule in numeric order of cases

(c) Expert rule with aligned chains of cases

(d) DB-based rule with aligned chains of cases

(e) Expert rule with centered chains

(f) DB-based rule with centered chains

Fig. 4. (a),(c),(e) Visualization of the expert cancer rule in 10-D feature space, (b), (d),(f) visualization of the closest cancer rule extracted from the data base 10-D feature space

The bar layer next from the bottom contains all 12 vectors that have norm $|\mathbf{x}|=1$. All vectors on the layers above it have norms from 2 to 12, respectively. The largest number or vectors is in the middle level (norm $|\mathbf{x}|=6$) for n=12. Therefore, that middle layer is the longest one. In 3-D, each layer is represented as a disk as shown in Figure 3 (c)-(f). This visual data representation is called *Multiple Disk Form* (**MDF**). It shows all 4096 12-D vectors without any overlap.

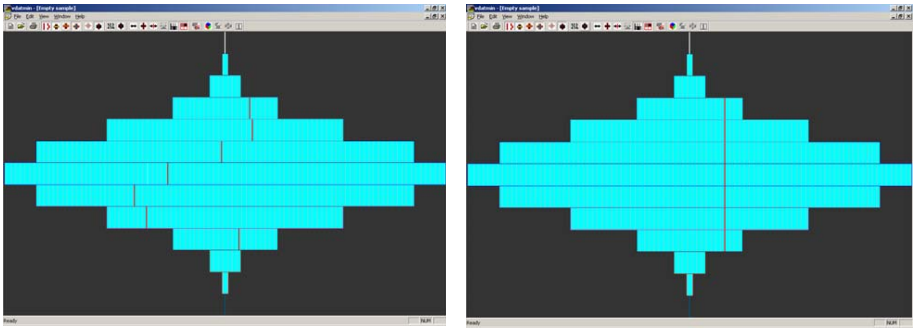
The MDF visualization is applied in Figure 4 to visualize cancer rules discovered by using relational data mining and by "expert" mining [Kovalerchuk, Vityaev, Ruiz, 2001]. The rule generated by the expert radiologist is shown in Figure 4 (a),(c),(e) and the cancer rule extracted from the database by a relational data mining algorithm MMDR [Kovalerchuk, Vityaev, 2000] is shown in Figure 4 (b), (d),(f). Each rule is described by showing all cases where it is true, as black bars and as white bars where it is false. In other words, each Boolean vector **x** (case, patient, element) is represented in MDF as a *black bar* if the target value for **x** is equal to 1 (cancer), $f(\mathbf{x})=1$, and it is a *white bar* if $f(x)=0$ (benign).

The VDATMIN also allows using other bar colors to indicate the status of the vector. For instance, Figure 3(b) shows each layer of vectors in different color. This system can indicate another status of the vector (case) which shows whether it is derived from target values (e.g., cancer, benign) of other cases using the monotonicity hypothesis. The vector **y** is rendered as a *light grey bar* if its target value $f(\mathbf{y})=0$ is *derived* from the target value for the vector **x**, such that $\mathbf{y} \leq \mathbf{x}$ and $f(\mathbf{x})=0$. Alternatively, the vector **y** is rendered as a *dark grey bar* if $\mathbf{y} \geq \mathbf{x}$ and $f(\mathbf{x})=1$. In this case $f(\mathbf{y})=1$. Vector **y** is called an *expanded vector*. The idea is that if the monotonicity hypothesis holds, then the value of $f(\mathbf{y})$ can be derived by expanding the value $f(\mathbf{x})$ as shown above. In other words, for white **x**, vector **y** is rendered as a light gray bar, which shows its similarity to **x** in the target variable value (e.g., cancer), and its status as derived from x is not directly observed. Similarly the dark gray color is used for vector **y** with the target value derived from $f(\mathbf{x})=1$. While grey scale metaphor with black and white extremes is a common one, sometimes it is better visually to use the darkness scale with other colors. Specifically VDATMIN uses the scale of blue color as well.

Figure 5 explains the visualization used in Figure 4 (b),(d),(f) related to aligning chains of vectors. Say we have a set of Boolean vectors:

(0000 0000 100) < (0000 1000 100) < (0001 1000 100) < (0001 1010 100) <
(0001 1010 110) < (0001 1011 110)

with a lexicographical order. This means that every coordinate in the previous vector is less than or equal to the same coordinate in the next vector. In Figure 5(a) all vectors are ordered in each layer independently from vectors on other layers. It is done by using their binary numeric value, while in Figure 5(b) it is done in accordance with their lexicographical order. They form a straight line of bars starting from the smallest one. This makes the visualization much clearer and simpler.



(a) Monotone increasing vectors rendered in the numeric order

(b) The same elements aligned as a vertical chain

Fig. 5. Alternative visualizations of increasing vectors

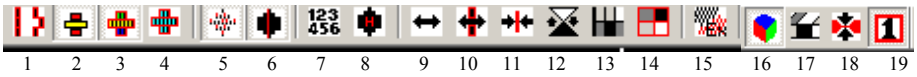
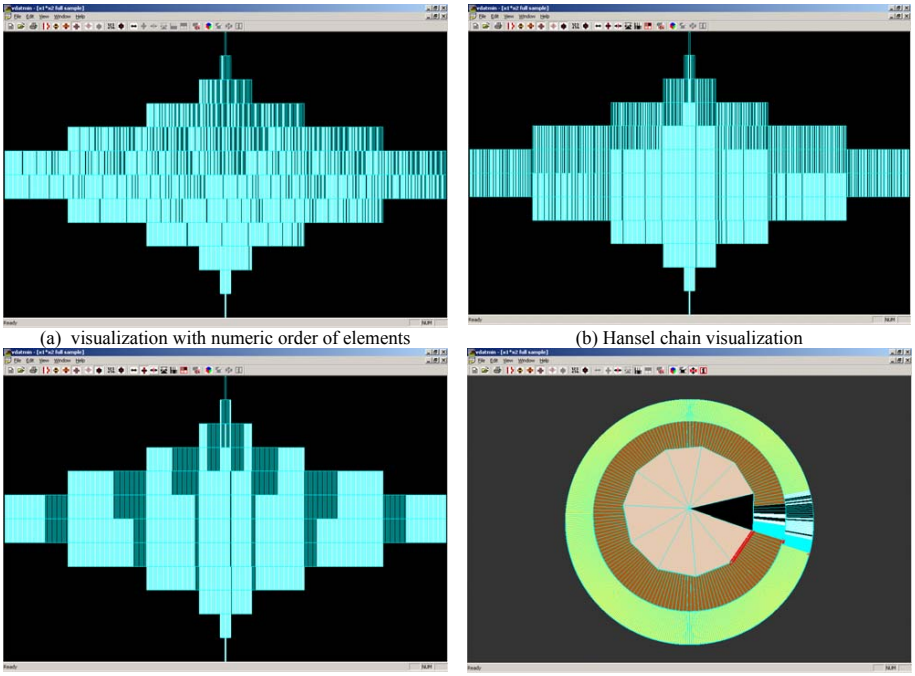


Fig. 6. Details of the user interface to support viewing monotone chains of elements. (1) Align Chains vertically, (2) Show Layer Borders, (, (3) Show Element Borders, (4) Show Element Borders in one color, (5) Show Bush Up (elements that are greater than a selected element) , (6) Highlight a chain, (7) Sort elements using their numerical values (natural order), (8) Align Chains (sort the data using the Hansel chains algorithm), (9) Move an element by dragging, (10) Move a chain by dragging, (11) Automatically center Hansel chains, (12) Expand Elements, (13) Show Monotonicity Violations, (14) Change expanded to real elements, (15) Expand Monotonicity, (16) Show 3D view, (17) Show 3D plot view, (18) Show 3D compressed view, (19) show initial position of disk.

Figure 6 shows the details of the user interface. Button 12 “Expand Elements” toggles the ability to click on a 2D element and expands down the chain if the element is white and expands up the chain if the element is black. Button 13 “Show Monotonicity Violations” will show violations as red elements. Button 14 “Change expanded to real elements” toggles the ability to click on an element and change it from expanded status (dark gray or light gray) to real (black or white). Button (16) “Show 3D view” toggles between 2D and 3D view. Button (16) “Show 3D plot view” is a 3D view that draws on the tops and bottoms of disks. Button (17) “Show 3D compressed view” is a view that compresses the data based on it being close to other data. Button (18) “Show the initial position of the disk” draws a red box around the 1st element in each layer. In the 3D a user has abilities to change the view of the MDF by controlling the camera that include rotating left - right, moving left-right, up-down, and zooming in and out.



(a) visualization with numeric order of elements (b) Hansel chain visualization

(c) Reordered (centered) Hansel chain visualization (d) Top 3-D view of (c)

Fig. 7. The visualization of the Boolean rule $y= x_1\&x_2$ in 11-dimensional space

Figure 7 shows the visualization of the Boolean rule $f(\mathbf{x}) = x_1\&x_2$ in 11-dimensional space, e.g. if in $x=(x_1,x_2,x_3,\ldots,x_{11})$ we have $x_1=x_2=1$, $x_i=0$, i= 3,4,…,11 then $f(\mathbf{x})=1$. In Figure 7, all vectors that have $f(\mathbf{x})=1$ are black bars in and all vectors that have $f(\mathbf{x})=0$ are white bars. There is no vector $\mathbf{x}$ with $f(\mathbf{x})$ expanded by monotonicity because all values of the target are given explicitly by the rule $f(\mathbf{x}) = x_1\&x_2$. This is the case when we have a complete rule. However, this is not the case in data-driven data mining where training data represent only a fraction of all vectors $\mathbf{x}$ in E^n. The first black bar (on the third layer from the bottom) represents the vector $\mathbf{x}$ with $x_1=x_2=1$, $x_i=0$, i= 3,4,…,11. All other vectors on the same layer with norm $|\mathbf{x}|=2$ are white because they cannot have $x_1=x_2=1$. The next layer ($|\mathbf{x}|=3$) contains 9 vectors and respectively 9 black bars.

In Figure 7(a), all vectors are ordered in each layer in accordance with their binary value (e.g., vector (000…111) is numerically smaller than (111…000)), where it is assumed that x_1 represents a lowest bit and x_{11} represents the highest bit. This our vector with $x_1=x_2=1$, $x_i=0$, i= 3,4,…,11 is shown on the right end of the layer with $|\mathbf{x}|=2$.

Figure 7(b) shows a border between two classes of $f(\mathbf{x})$ values 0 and 1 much better than (a) representation. It is based on monotone chains of elements of E^n called Hansel chains [Hansel, 1966]. Mathematical details how these layers are built are given in [Kovalerchuk, Delizy, 2005, Kovalerchuk, et al., 1996].

To be able to visualize data of the larger dimension we use grouping of Hansel chains and visualize groups of similar chains as a single chain. Thus, less area is needed to show the same data. The user has abilities to enter data to be visualized in two ways: (1) as formulas such as any disjunctive normal form (e.g., $x_1\&x_2 \vee x_3\&x_4\&x_5$) or as actual vectors in n-D. In the first case, the program parses the logical formulas.

3 Visualization for Breast Cancer Diagnistics

We already presented in Figure 4 MDF visualization of one of the expert cancer rule and the cancer rule extracted from the database by the data mining algorithm. Below we expand this analysis and show it in Figure 8.

A more complete **cancer rule** produced by the "expert mining" process that involves 11 features is as follows:

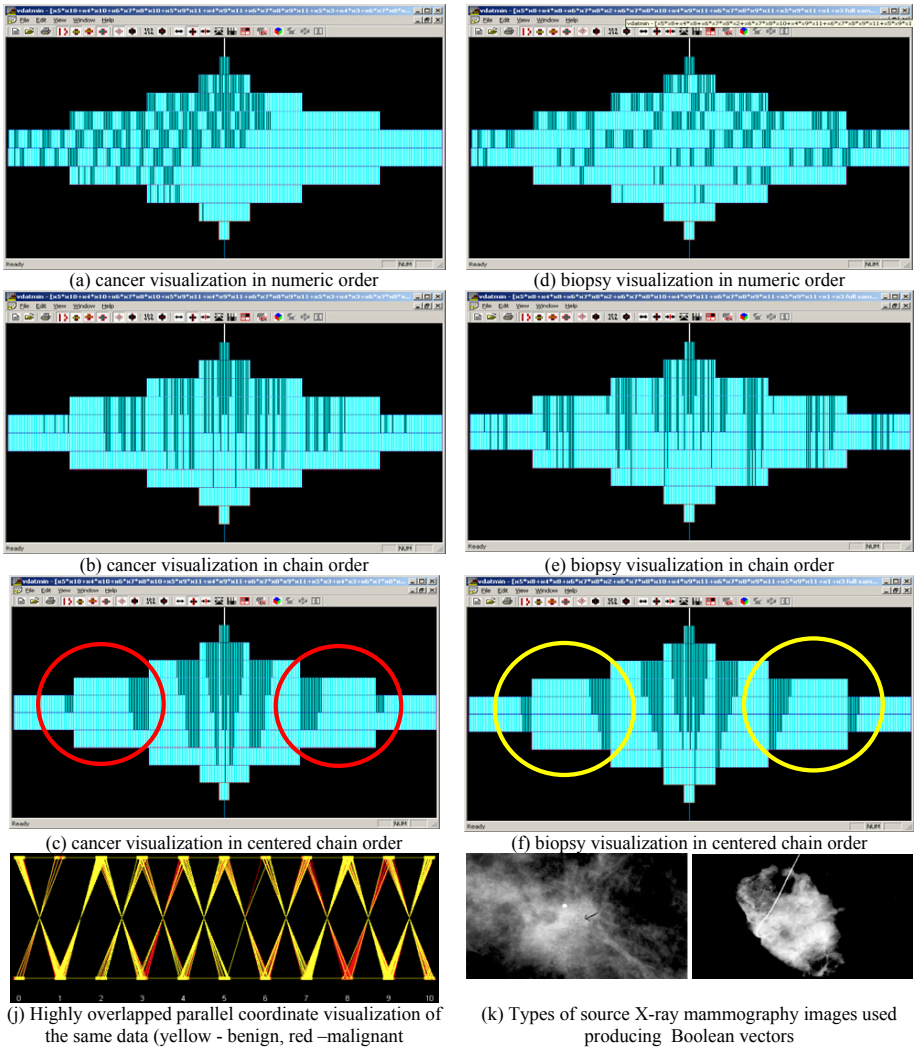
$$f(\mathbf{x}) = x_5x_{10} \vee x_4x_{10} \vee x_6x_7x_8x_{10} \vee x_5x_9x_{11} \vee x_4x_9x_{11} \vee x_6x_7x_8x_9x_{11} \vee x_5x_3 \vee x_4x_3 \vee x_6x_7x_8x_3 \vee x_2x_3 \vee x_1 \quad (2)$$

This formula is from [Kovalerchuk, Vityaev Ruiz, 2001] converted to the disjunctive normal form with the renaming variables to be able to feed VDATMIN directly. Figure 8 shows this rule with all three MDF visualization options.

Expert rules for the **biopsy** also have been developed in by using the expert mining technique based on Monotone Boolean Functions approach in [Kovalerchuk, Vityaev, Ruiz, 2001]. It is shown below again in a modified notation to be able to feed VDATMIN:

$$f(\mathbf{x}) = x_5x_{10} \vee x_4x_{10} \vee x_6x_7x_8x_{10} \vee x_5x_9x_{11} \vee x_4x_9x_{11} \vee x_6x_7x_8x_9x_{11} \vee x_5x_3 \vee x_4x_3 \vee x_6x_7x_8x_3 \vee x_2x_3 \vee x_1 \quad (3)$$

Figure 8 shows the advantages of chain-based MDF visualization relative to visualization that does not exploit monotone chains. The chain-based border between classes is much clearer. This advantage gives an immediate benefit: visual comparison of rules for biopsy and cancer. It also helps to identify the level of consistency of cancer and biopsy rules provided by the expert. It is expected that a biopsy rules should be less stringent than a cancer rules. For instance if the presence of $x_2\&x_3$ is a cancer indicator but only presence of x_3 can be sufficient to recommend biopsy test. In visual terms it means that the border of the biopsy class should be lower or at the same as cancer class for them to be consistent. This is exactly the case as Figure 8 (c) and (f) show. The black areas in the ovals in Figure 8 (f) for biopsy are lower than the same areas for cancer in Figure 8 (f). Figure 8 (j) shows highly overlapped parallel coordinate visualization of the same data (yellow - benign, red – malignant). The same classes are shown separately in Figure 1 (c). It shows advantages of VDATMIN relative to parallel coordinates for Boolean data. Figure 8(k) shows types of source X-ray mammography images used to derive Boolean vectors.



(a) cancer visualization in numeric order

(d) biopsy visualization in numeric order

(b) cancer visualization in chain order

(e) biopsy visualization in chain order

(c) cancer visualization in centered chain order

(f) biopsy visualization in centered chain order

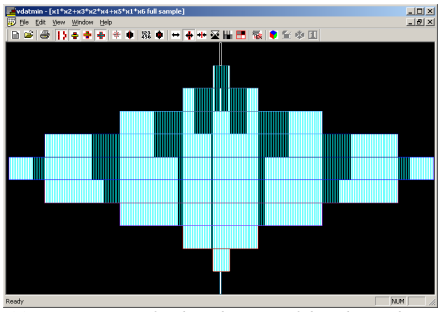
(j) Highly overlapped parallel coordinate visualization of the same data (yellow - benign, red –malignant

(k) Types of source X-ray mammography images used producing Boolean vectors

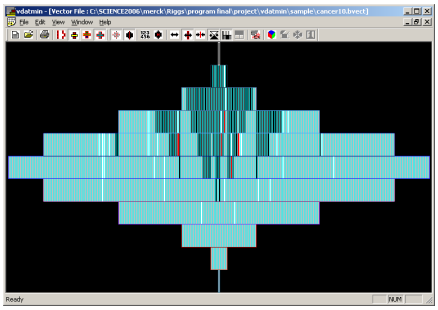
Fig. 8. Visualizations of expert cancer and biopsy rules

4 General Concept of Using MDF in Data Mining

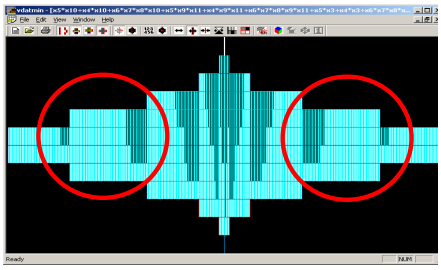
Figure 9 illustrates the general concept of simultaneous coordinated visualization of multiple components of the analytics: original available training data, rules extracted from these data by using data mining, rules extracted from the expert. Often the data and rules are in two categories: "final" and "warning". In many applications, final rules produce a "final" decision, e.g., cancer, crime, security breach, but "warning" rules produce warnings about possible final state, e.g., biopsy positive, crime warning, and security alert. There are must be consistency between final and warning rules from data mining and expert mining. The VDATMIN allows capturing discrepancies and consistency visually as Figure 9 shows on the illustrative examples for "final" and "warning" rules. Comparison of Figure 9(a) and 9(b) shows discrepancy between "final" data mining rules and monotonically expanded data. In the cancer example, this may lead to both missed cancer cases and benign cases diagnosed as malignant. In the center of (b) we see that the border of the monotone expansion is below than the border in (a). This means that some cancer case would be diagnosed as benign. In contrast, on the sides for both (a), (b) we see an opposite picture, which may lead to benign cases diagnosed as malignant. Similar interpretation will take place for crimes and security examples. Figure 9(c) and 9(d) show full consistency of expert 'final" and "warning" rules with each other. All "final" rules are nested in the "warning" rules. Both these rules also much more consistent with data (c) than pure data mining rules (a) as visual comparison shows in a very compact way in Figure 9.



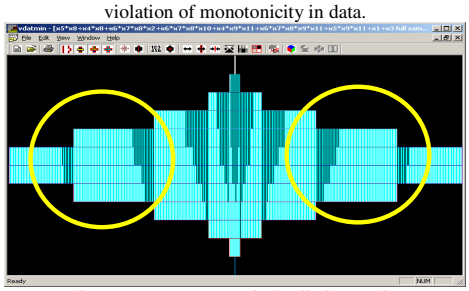
(a) *Data mining "final" rules - non-false alarm alert rules* extracted by a data mining algorithm



(b) *Actual data* (black and white) with expanded cases (light and dark gray) obtained by monotonicity. Red - violation of monotonicity in data.



(c) *Expert final rules* –non-false alarm



(d) *Expert "warning rules"*- all alarm rules

Fig. 9. Visual comparison of rules provided by the expert (a), (b), extracted from data (c) and "visual rule" (d)

5 Scaling Algorithms

5.1 Algorithm with Data-Based Chains

Our goal is to represent the MDF on a single screen with possibly some scrolling. The major factor that is limiting the visualization is the number of Hansel chains that can be visualized as vertical lines. We use two approaches:

(A1) grouping chains with similar height of the border to a cluster,
(A2) constructing chains from only vectors available in the database.

The steps of the algorithm to construct these chains are described below and illustrated in Figures 10 and 11:

Step 1: Order all vectors according their Hamming norm.

Step 2: Loop: for each vector $\mathbf{v}_i$ starting from the first one find all nodes that are greater than this node, $\mathbf{v}_i < \mathbf{v}_j$. This will create a matrix $M=\{m_{ij}\}$, where $m_{ij}=1$ if $\mathbf{v}_i < \mathbf{v}_j$ else $m_{ij}= \infty$. We can record only $m_{ij}=1$. Typically, this is a sparse matrix. This matrix can be interpreted as an incidence matrix of the graph G (directed acyclic graph, DAG), where 1 means that there is direct link between nodes with length 1 and $m_{ij}= \infty$ means the absence of the link and infinite length. Thus this step builds DAG, where arrow between nodes show the direction form smaller Boolean vector to the larger one.

Step 3: Find a longest directed path P in G using M. Call this path chain 1.

Step 4. Move C_1 to the center of MDF

Step 5: Remove all nodes of P from G and find a longest directed path in G with removed P. This path produces chain C_2. Locate C_2 vertically: one vector above another one in MDF.

Step 6: Repeat step 4 until every node of G will belong to some path. Steps 3-5 will produce k chains $\{C_i\}$ that do not overlap and cover all nodes of G.

Step 7: Compute distances D_{HC} (C_i, C_j) between all chains C_i.

Step 8: Move all other chains in accordance with their distance to C_1 in MDF. The chains with the shorter distance will be the closer to C_1.

Step 9: Assign color to vectors on each chain: black for f(x)=1 and white for f(x)=0.

Step 10: This step contains tree components: expanding chains to have vectors with equal Hamming norms on both chains; equalizing expanded vectors in color with given vectors to see the pattern of the border better, and hiding the empty part of the MDF form.

Consider an example of 200 given vectors in the E^{100}. What is the space needed to visualize them in MDF form? In the best case scenario we would have just two vertical chains that will contains all 200 vectors. Say the longest chain will have 101 vectors and the second chain will contain remaining 99 vectors. This is due to the fact

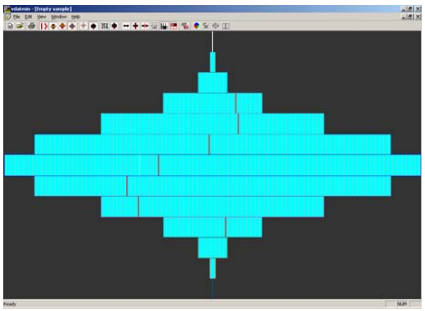
that in100-D the longest chain contains 101 vectors. In the worst case, we would need to visualize 200 chains, if each vector forms its own chain when 200 vectors are incomparable. Similarly, for a much larger set of 10^6 vectors we would need to visualize at least about 10^4 chains (10^6/101). For a screen with 2000 pixels, it will result in scrolling the screen 5 times to observe these 10^4 chains in MDF completely for 100-D space and 10^6 vectors. The combining of the scrolling with clustering of chains where each cluster will have about 5 chains per cluster allows to compress all 10^6 vectors in 100-D space into a single screen in the complete multiple disk form (MDF).

On step 7, a user can switch between different distances. To describe distance used we define necessary concepts. Let L(C) be the lower unit of the chain C (the vector **z** on the chain with the smallest norm such that $f(\mathbf{z})=1$) Next let $E(C_1,C_2)$ be the smallest element **z** of the chain C_2 with f(x)=1 that was obtained by monotone expansion of element $L(C_1)$. This means that is knowing the value $f(L(C_1))=1$ we can expand this value to $\mathbf{z}=E(C_1,C_2)$. Thus, this value $f(\mathbf{z})=1$ cannot be expanded by monotonicity to elements of chain C_2 that are below $E(C_1,C_2)$.

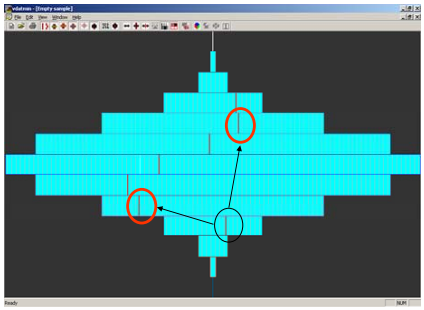
The Hamming distance D between lower units of two chains, $L(C_i)$ and L(Cj), D(L(Ci),L(Cj)) creates a smooth border, but it does not capture the monotone similarity between chains. The Hamming distance combined with Monotone Expansion, called *HME measure* captures both properties. In HME chain C_2 is placed closer to chain C_1 than chain C_3, if the smallest expanded element of C_2 from C_1, $E(C_1,C_2)$, is closer (in Hamming distance D) to $L(C_1)$, which is $D(E(C_1,C_2)) < D(E(C_1,C_3))$. HME is infinite if chain C_2 has no such expanded elements.

5.2 Algorithm with Pixel Chains

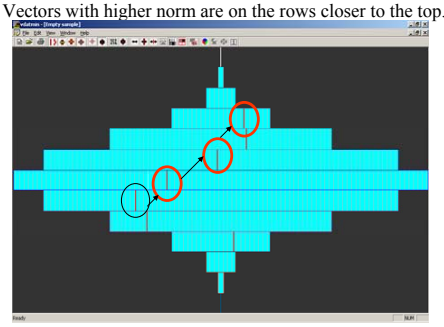
This algorithm modifies steps 8-10 from the previous algorithm. To visualize E^{100} it uses a window of 101x100 pixels. The x coordinate is the Hamming-based distance/measure from the longest chain to the current chain H. The y coordinate is the norm (height) of the *lower unit* on the chain H. In E^{100}, this size of the window follows from the fact that the largest Hamming distance is 100 and the longest chain has 101 vectors. See Figure 12. In general for E^n the window is $(n+1)\times n$. Thus a single screen has enough space for E^n with n=1000. This window is called a *Chain Pixel Space* (CPS). Chains are placed in CPS, where each pixel is empty or contains one or more chains. A user can change the visualization by switching the measures used in x (e.g., switching Hamming distance and HME). This visualization is very compact where each pixel can represent hundreds and thousands vectors, but with possible chain overlap. The number of chains overlapped in the pixel is shown by pixel color intensity in 2D or by a bar height when CPS is converted to its 3-D form. The spread of the border is shown in Figure 12(a) in each column. Figure 12(b) shows the lower edge of the border. Similarly, an upper border is visualized.



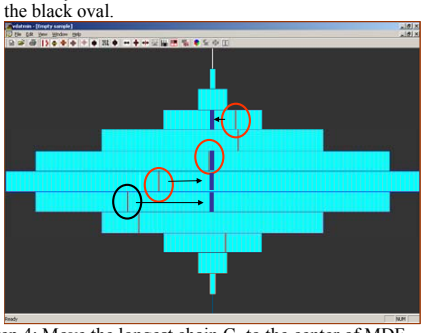
Step 1: Order given vectors according their Hamming norm. Show each vector as a bar in the row of its norm. Vectors with higher norm are on the rows closer to the top.



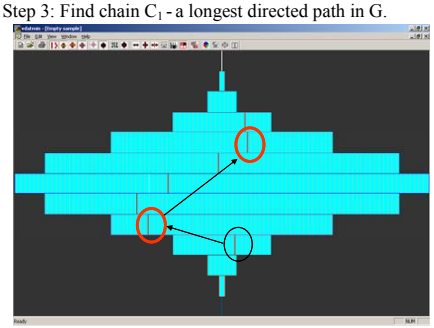
Step 2: Build a graph G of vectors. G has a link from $\mathbf{v}_i$ to $\mathbf{v}_j$. if $\mathbf{v}_i < \mathbf{v}_j$. Vectors in red ovals are greater than the vector in the black oval.



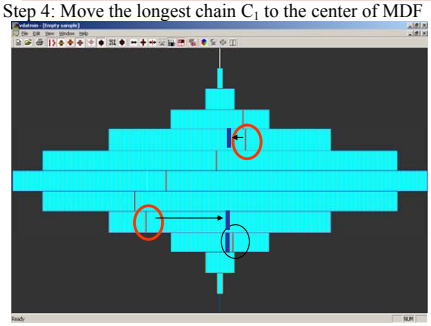
Step 3: Find chain C_1 - a longest directed path in G.



Step 4: Move the longest chain C_1 to the center of MDF

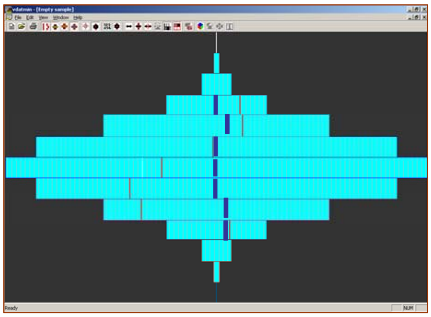


Steps 5-6: Remove all nodes of C_1 from G, find a longest directed path in G\C_1. Repeat to get all other chains

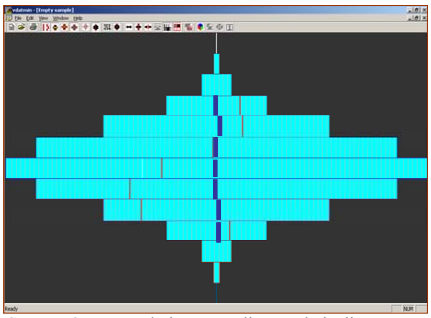


Steps 5-6: Locate chain vertically C_2: one vector above another one in MDF. Repeat this with other chains.

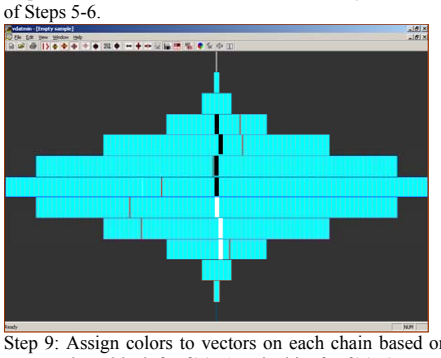
Fig. 10. Illustration of algorithm with data-based chains: part 1



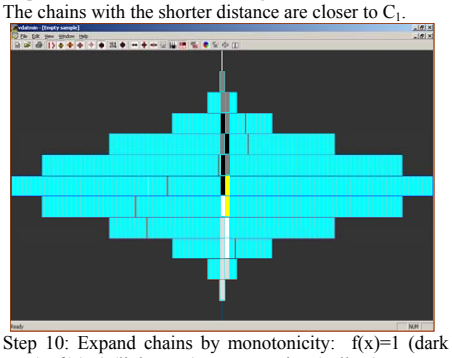
Steps 5-6: Chains C_1 and C_2 located vertically as a result of Steps 5-6.



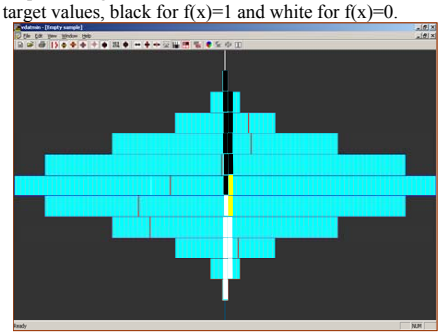
Steps 7-8: Move chains according to their distance to C_1. The chains with the shorter distance are closer to C_1.



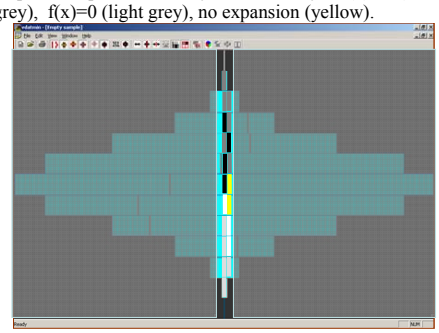
Step 9: Assign colors to vectors on each chain based on target values, black for f(x)=1 and white for f(x)=0.



Step 10: Expand chains by monotonicity: f(x)=1 (dark grey), f(x)=0 (light grey), no expansion (yellow).



Step 10: Expanded vectors equalized in color with given vectors to see the pattern of the border better.



Step 10: Hiding and removing the empty part of the MDF form.

Fig. 11. Illustration of the algorithm with data-based chains: part 2

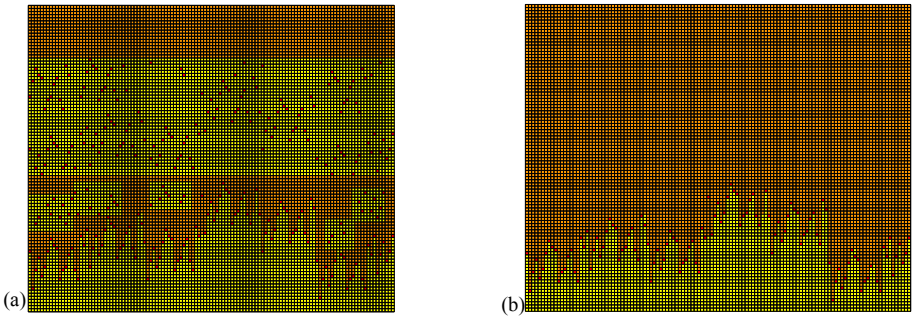


Fig. 12. Pixel-based chain visualization: (a) all chains, (b) lower border between classes

6 Binarization and Monotonization

Above we considered visual data mining with monotone binary data. Below we discuss generalization of this technique for the data that are not binary and not monotone. The simplest way to binarize data is use a threshold with δ-function: $\delta(x)=1$ if $x>=T$ else $\delta(x)=0$. The main reason for binarization is data simplification and getting a qualitative evaluation of the complex situation. Binarization of data is an extreme case of data discretization that is quite common in data mining with multiple techniques developed. Typically, we need to analyze the situation at the different levels of detail. We would prefer first to get a "bird view" of the situation and then 'zoom' to most interesting spots found in the 'bird view". This is not only a question of convenience, but a deep issue of artificial intelligence and brain modeling. In this way humans able to solve tasks that seems computationally intractable. These issues are studied in Dynamic Logic of Phenomena (DLP) [Kovalerchuk, Perlovsky, 2008, 2009].

Example 1: *Crime situation*. Multiple demographic factors contribute to crime level in specific areas. The collected data can be overwhelming, therefore initial qualitative analysis can be done with two target values "growing crime level", and "stable or declining crime level" that can be measured by threshold that the increase in the number of criminal cases is greater than threshold T.

Similarly, we can binarize features that impact the target feature –crime level. For the example 1 it can be: (1) population increase (no significant increase, significant increase); (2) Income (low, high);(3) Tax collected level (low, high), (4) Population percent of age 18-30 (low, high), (5) Mean age (no greater than T, greater than T), (6) Education level as a percent of population with college degree (low- no greater than T, high- greater than T), (7) Unemployment rate (low, high), (8) Law enforcement budget rate (low, high), (9) Crime level in the adjacent neighborhoods, (10) High school dropout level (low, high).

Thus, we can get 10-dimensional Boolean vectors of these exogenous attributes with low encoded as 0 and high as 1. Each available training vector is labeled by 0 or 1, that represent the crime level (low or high). Each such record will be for an individual neighborhood and a specific time. Such binary data representation allows

us to use the visual data mining technique described in this paper to discover a visual predictive rule for prediction crime level (low, high). We will also be able to test monotonicity of the attributes (1)-(10) relative to crime level. The lack of monotonicity may mean that some other impact factors are missing or data are corrupted. For instance, some neighborhoods already implemented the neighborhood watch program. This feature is not included in (1)-(10). Several iteration of adding attributes may be required to reach monotonicity. Thus, the proposed visual data mining approach can serve as a *feature augmenting mechanism* that compliment known feature selection mechanisms, which do not guide how to look at new attributes.

Example 2: *Security situation of the port*. Multiple factors within the port and outside contribute to port security level and the collected data can be overwhelming. Therefore initial qualitative analysis can be done with two target values "growing security threats", and "stable or declining security threats" that can be measured by threshold that the increase in the number of security alerts is greater than threshold T in the area.

Similarly, using thresholds we can binarize features that impact the target feature – security level: (1) Cargo in the area (down or no significant increase, significant increase), (2) Real estate value of the area (low, high), (3) Cargo value in the area (low, high), (4) Average number of people in the area (low, high), (5) Average number of non-port employees in the area (low, high), (6) Number of sensors in the area (low, high), (7) Average time a non-employee in the area per day (low, high), (8) Average time an employee in the area per day (low, high), (9) Average number of security people in the area (low, high), (10) Security budget rate in the area per $ of real estate in the area (low, high), (11) Security budget rate in the area per $ of cargo value in the area (low, high), (12) Incident level in the area (high, low).

In this example, final rules could indicate strong and urgent security measures that require significant extra security resource allocated and warning rules could indicate addition attention to the area with minimal additional resources allocated. As above in Figure 9 VDM allows to compare rules "extracted" from the security expert and with rules obtained by data mining and with the date expanded by monotonicity.

Obviously, it is not clear at the beginning how these attributes are related to the number of alerts generated in the area. Answering this question is a data mining task. To solve this task, we can get 10-dimensional Boolean vectors of these exogenous attributes with low encoded as 0 and high as 1. Each vector marked by the security alert rate (low or high encoded by 0 or 1 as well). Each such record will be for an individual area of the port and a specific time. This data binary representation allows us to use the visual data mining technique described above to discover visual predictive rules for prediction security alert level (low, high). We will also be able to test monotonicity of the attributes (1)-(10) relative to alert level. The lack of monotonicity may mean that some other impact factors are missing or data are corrupted. For instance, some port area already implemented the employee security training program, internal alert analysis, etc. These features are not included in (1)-(10). After several iteration of adding attributes, we can reach monotonicity. Thus, similar to the previous example, `VDATMIN` will serve here as a *feature augmenting mechanism* that compliment known feature selection mechanisms.

7 Monotonization

The algorithm below describes main steps of a monotonization process for the tasks where monotonicity is violated:

Step 1: Find a specific pair of vectors (**x**,**y**) with violation of monotonicity, that is **x**> **y** but f(**x**) < f(**y**)

Step 2: Find attributes of (**x**,**y**) that led to violation of monotonicity, **x**> **y** => f(**x**) < f(**y**).

Step 3: Find a subspace S and sub-vectors $\mathbf{x}_s$ and $\mathbf{y}_s$ in subspace S such that monotonicity holds: $\mathbf{x}_s < \mathbf{y}_s$ => f(**x**) < f(**y**).

Step 4: Get attributes U={u} of the total space W that do not belong to S, W\S = U. These attributes cause the violation of monotonicity, **x**> **y** => f(**x**) < f(**y**) in space W.

Step 5: Modify attributes U.

Example: If attribute (1) "Population increase" is a source of monotonicity violation, it can be modified to a new attribute (g): "*Growth of crime age population"*. The monotone link between this attribute and crime rate seems a reasonable hypothesis (to be tested) if *all other relevant factors* are the same. Under this assumption, the high growth in the crime age population in the area A may lead to a higher crime rate (Cr) than in area B with low growth of this category of the population. In contrast, if areas A and B have different other relevant factors, $F_A \neq F_B$, this may not be the case. Thus, we may have a very specific *restricted type of monotonicity* with the same other relevant factors:

$$[\ (F_A = F_B)\ \&\ g(A) \geq g(B)\ ] \Rightarrow Cr(A) \geq Cr(B), \qquad (1)$$

We will call it *OF-conditional monotonicity* because it holds only under condition that Other Factors (OF) are the same. Mathematically it means that $|\mathbf{a}| = |\mathbf{b}|+1$, that is Boolean vector **a** is obtained from Boolean vector **b** by changing one of its zero values to one. This is exactly how Hansel chains are built. In other words, this is a *one step up single-attribute monotonicity* because all other attributes of vectors **a** and **b** are the same.

Step 6: Test monotonicity of modified attributes. In the example above, it means testing (1) on the available data.

Step 7: If Step 6 test succeeded, we can use a modified attribute g instead of the original attribute in the `MBFVA` algorithm. If Step 6 test failed on monotonicity or OF-conditional monotonicity then go to step 8.

Step 8: Decide between two options (i) return to step 5 and make another modification of attributes and (ii) add a new attribute, that is go to step 9.

Note: The failed test result on step 6 can be a very useful result in spite being negative. It indicates that we may miss other relevant factors. Area A can be in the region with historically low crime due to a type population (low migration, high

church influence, etc). Thus, even negative test of monotonicity is helpful as a guidance to search new relevant attributes and improving data mining output.

Step 9: Add new attribute. In the example after adding migration level and church influence to $F_A = F_B$ we may generate a new OF-conditional monotonicity hypothesis for (1) and go to step 6 again.

Step 10. Continue looping steps 5-9 until monotonicity produced for all attributes in the original space W or time limit reached.

8 Conclusion

Monotone Boolean Function Visual Analytics (MBFVA) method allows the discovering of rules that are meaningful for the subject matter expert (SME) and are confirmed with the database. The efficiency of the method is illustrated with discovering breast cancer diagnostic rules that are produced by (i) Subject Metter Expert, (ii) the analytical data mining algorithm, and (iii) the visual means from data. The proposed coordinated visualization of these rules is a way to produce high quality rules. Multivariate binary data are visualized in 2-D and 3-D without occlusion. It preserves structural relations in multivariate data. As a result, the complex border between classes in a multidimensional space is converted into visual 2-D and 3-D forms. This decreases the user information overload. To expand the applicability of the described approach, this paper presented an outline of the scaling algorithm for large datasets where each chain of multidimensional vectors is compressed into a single pixel. The detailed development of this algorithm is a topic of future research.

References

[1] Beilken, C., Spenke, M.: Visual interactive data mining with InfoZoom-the Medical Data Set. In: 3rd European Conf. on Principles and Practice of Knowledge Discovery in Databases, PKDD (1999), http://lisp.vse.cz/pkdd99/Challenge/spenke-m.zip

[2] Groth, D., Robertson, E.: Architectural support for database visualization. In: Workshop on New Paradigms in Information Visualization and Manipulation, pp. 53–55 (1998)

[3] Hansel, G.: Sur le nombre des functions Bool'eenes monotones de n variables. C.R. Acad. Sci., Paris 262(20), 1088–1090 (1966)

[4] Inselberg, A., Dimsdale, B.: Parallel coordinates: A tool for visualizing multidimensional Geometry. In: Proceedings of IEEE Visualization 1990, pp. 360–375. IEEE Computer Society Press, Los Alamitos (1990)

[5] Keim, D., Hao Ming, C., Dayal, U., Meichun, H.: Pixel bar charts: a visualization technique for very large multiattributes data sets. Information Visualization 1(1), 20–34 (2002)

[6] Keim, D., Müller, W., Schumann, H.: Visual Data Mining. In: EUROGRAPHICS 2002 STAR (2002), http://www.eg.org/eg/dl/conf/eg2002/stars/s3_visualdatamining_mueller.pdf

[7] Keim, D.: Information Visualization and Visual Data Mining. IEEE TVCG 7(1), 100–107 (2002)

[8] Keller, N., Pilpel, H.: Linear transformations of monotone functions on the discrete cube. Discrete Mathematics 309(12), 4210–4214 (2009)
[9] Korshunov, A.D.: Monotone Boolean Functions. Russian Math. Surveys 58(5), 929–1001 (2003)
[10] Kovalerchuk, B., Delizy, F.: Visual Data Mining using Monotone Boolean functions. In: Kovalerchuk, B., Schwing, J. (eds.) Visual and Spatial Analysis, pp. 387–406. Springer, Heidelberg (2005)
[11] Kovalerchuk, B., Triantaphyllou, E., Despande, A., Vityaev, E.: Interactive Learning of Monotone Boolean Functions. Information Sciences. Information Sciences 94(1-4), 87–118 (1996)
[12] Kovalerchuk, B., Vityaev, E., Ruiz, J.: Consistent and complete data and "expert" mining in medicine. In: Medical Data Mining and Knowledge Discovery, pp. 238–280. Springer, Heidelberg (2001)
[13] Kovalerchuk, B., Vityaev, E.: Data Mining in Finance: Advances in Relational and Hybrid Methods. Kluwer/Springer, Heidelberg, Dordrecht (2000)
[14] Kovalerchuk, B., Perlovsky, L.: Fusion and Mining Spatial Data in Cyber-physical space with Phenomena Dynamic Logic. In: Proceedings of the 2009 International Joint Conference on Neural Networks, Atlanta, Georgia, USA, pp. 2440–2447 (2009)
[15] Kovalerchuk, B., Perlovsky, L.: Dynamic Logic of Phenomena and Cognition. In: Computational Intelligence: Research Frontiers, pp. 3529–3536. IEEE, Hong Kong (2008)
[16] Lim, S.: Interactive Visual Data Mining of a Large Fire Detector Database. In: International Conference on Information Science and Applications (ICISA), pp. 1–8 (2010), doi:10.1109/ICISA.2010.5480395
[17] Lim, S.: On A Visual Frequent Itemset Mining. In: Proc. of the 4th Int'l Conf. on Digital Information Management (ICDIM 2009), pp. 46–51. IEEE, Los Alamitos (2009)
[18] de Oliveira, M., Levkowitz, H.: From Visual Data Exploration to Visual Data Mining: A Survey. IEEE TVCG 9(3), 378–394 (2003)
[19] Pak, C., Bergeron, R.: 30 Years of Multidimensional Multivariate Visualization. In: Scientific Visualization, pp. 3–33. Society Press (1997)
[20] Shaw, C., Hall, J., Blahut, C., Ebert, D., Roberts, A.: Using shape to visualize multivariate data. In: CIKM 1999 Workshop on New Paradigms in Information Visualization and Manipulation, pp. 17–20. ACM Press, New York (1999)
[21] Ward, M.: A taxonomy of glyph placement strategies for multidimensional data visualization. Information Visualization 1, 194–210 (2002)
[22] Schulz, H., Nocke, T., Schumann, H.: A framework for visual data mining of structures. In: ACM International Conf. Proc Series, vol. 171; Proc. 29th Australasian Computer Science Conf., Hobart, vol. 48, pp. 157–166 (2006)
[23] Badjio, E., Poulet, F.: Dimension Reduction for Visual Data Mining. In: Stochastic Models and Data Analysis, ASMDA-2005 (2002), http://conferences.telecom-bretagne.eu/asmda2005/IMG/pdf/proceedings/266.pdf
[24] Wong, P., Whitney, P., Thomas, j.: Visualizing Association Rules for Text Mining. In: Proc. of the IEEE INFOVIS, pp. 120–123. IEEE, Los Alamitos (1999)
[25] Wong, P.C.: Visual Data Mining. In: IEEE CG&A, pp. 20–21 (September/October 1999)
[26] Zhao, K., Bing, L., Tirpak, T.M., Weimin, X.: A visual data mining framework for convenient identification of useful knowledge. In: Fifth IEEE International Conference on Data Mining, 8 p (2005), doi:10.1109/ICDM.2005.16