

Central Washington University
ScholarWorks@CWU

Computer Science Faculty Scholarship

College of the Sciences

1-18-2010

Visual Discovery in Multivariate Binary Data

Boris Kovalerchuk

Florian Delizy

Logan Riggs

Evgenii Vityaev

Follow this and additional works at: <https://digitalcommons.cwu.edu/compsci>



Part of the [Computer Sciences Commons](#)

Visual Discovery in Multivariate Binary Data

Boris Kovalerchuk ^{a*}, Florian Delizy ^a, Logan Riggs ^a, Evgenii Vityaev ^b

^a Dept. of Computer Science, Central Washington University, Ellensburg, WA, 9896-7520, USA;

^b Institute of Mathematics, Russian Academy of Sciences, Novosibirsk, 630090, Russia,

ABSTRACT

This paper presents the concept of Monotone Boolean Function Visual Analytics (MBFVA) and its application to the medical domain. 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 rules, and to come up with rules that are meaningful for the expert in the field, and are confirmed with the database. 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” patterns, then this shows that 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.

Keywords: Visualization, Data Mining, Visual discovery, Monotone chains.

1. INTRODUCTION

Visual data mining (VDM) assists a user in detecting interesting knowledge, and in gaining a deep visual understanding of the data [Beilken & Spenke, 1999; Schulz, et al, 2006; Badjio, Pouletm 2005; Zhao et al, 2005]. *Visualizing the border* between patterns is one of the especially important aspects of visual data mining. The simple border and patterns that are visually far away from each other match our intuitive concept of patterns. It 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.

VDM methods have shown benefits in many areas, but these methods do not address the specifics of data, where there is little variability in the visual representation of different objects. 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 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 patterns 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

* Further author information: (Send correspondence to Boris Kovalerchuk)
E-mail: borisk@cwu.edu, Telephone: 1 509 963 1438

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 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).

The parallel coordinate visualization [Inselberg, Dimsdale, 1990] can work with ten or more attributes, 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 depicts the following records:

$$\begin{aligned} &01010;11010;01110;01011;01111;11011; \\ &11111;10101;11101;10111 \end{aligned} \quad (1)$$

Can we discover a regularity that governs the dataset in Figure 1 that represents these data in parallel coordinates? It is difficult, but the regularity in (1) is a simple monotone Boolean function $(x_2 \& x_4) \vee (x_1 \& x_3 \& x_5)$. Parallel coordinates belong to a class of methods that explicitly visualize *every* attribute x_i of an n -dimensional vector (x_1, x_2, \dots, 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*.

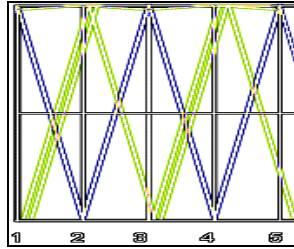


Figure 1. Ten Boolean records in parallel coordinates

2. METHOD FOR VISUALIZING DATA

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.

Figure 2 shows the user interface of visual data mining software VDATMIN. Figure 2(a) shows all $2^{12}=4096$ nodes of n -dimensional binary cube E^n for $n=12$. Each node (12-dimensional Boolean vector) is represented as a blue bar. The bar that represents the vector \mathbf{x} containing all zeros is located in the lowest layer in the middle of the picture. The bar that represents the vector \mathbf{x} that containing all ones is located at the top of the picture in the middle. All other bars are located in between.

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 of 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 2 (c)-(f). This visual data representation is called Multiple Disk Form (MDF). It shows all 4096 12-D vectors without any overlap.

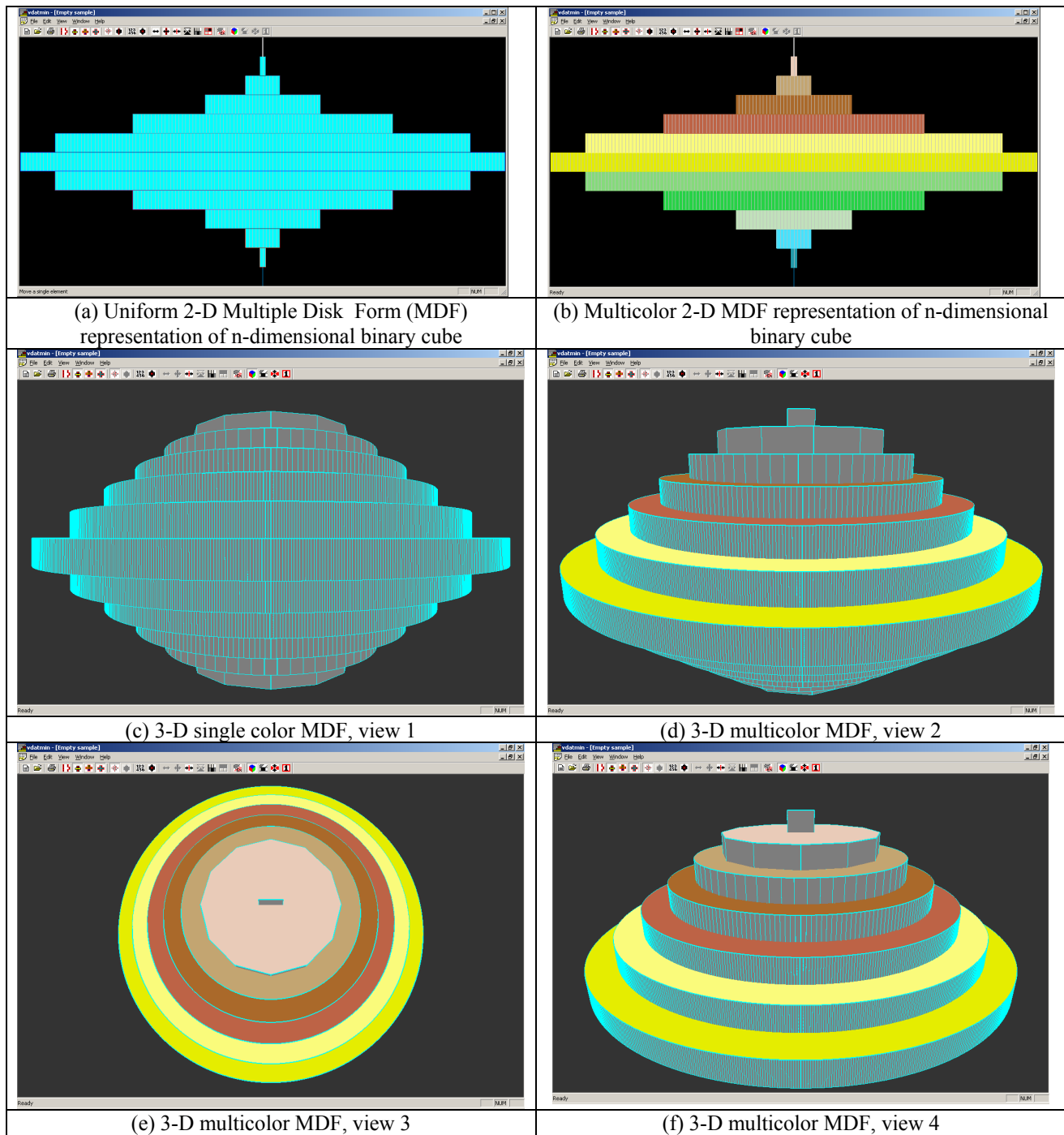


Figure 2. VATMIN user interface.

The MDF visualization is applied in Figure 3 to visualize cancer rules described analytically in [Kovalerchuk, Vityaev, Ruiz, 2001]. The rule generated by the expert radiologist is shown in Figure 3(a)-(c) and the cancer rule extracted from the database by a data mining algorithm is shown in Figure 3(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 \mathbf{x} (case, patient, element) is represented in MDF as a *black bar* if the target value for \mathbf{x} is equal to 1 (cancer), $f(\mathbf{x})=1$, and it is a *white bar* if $f(\mathbf{x})=0$ (benign).

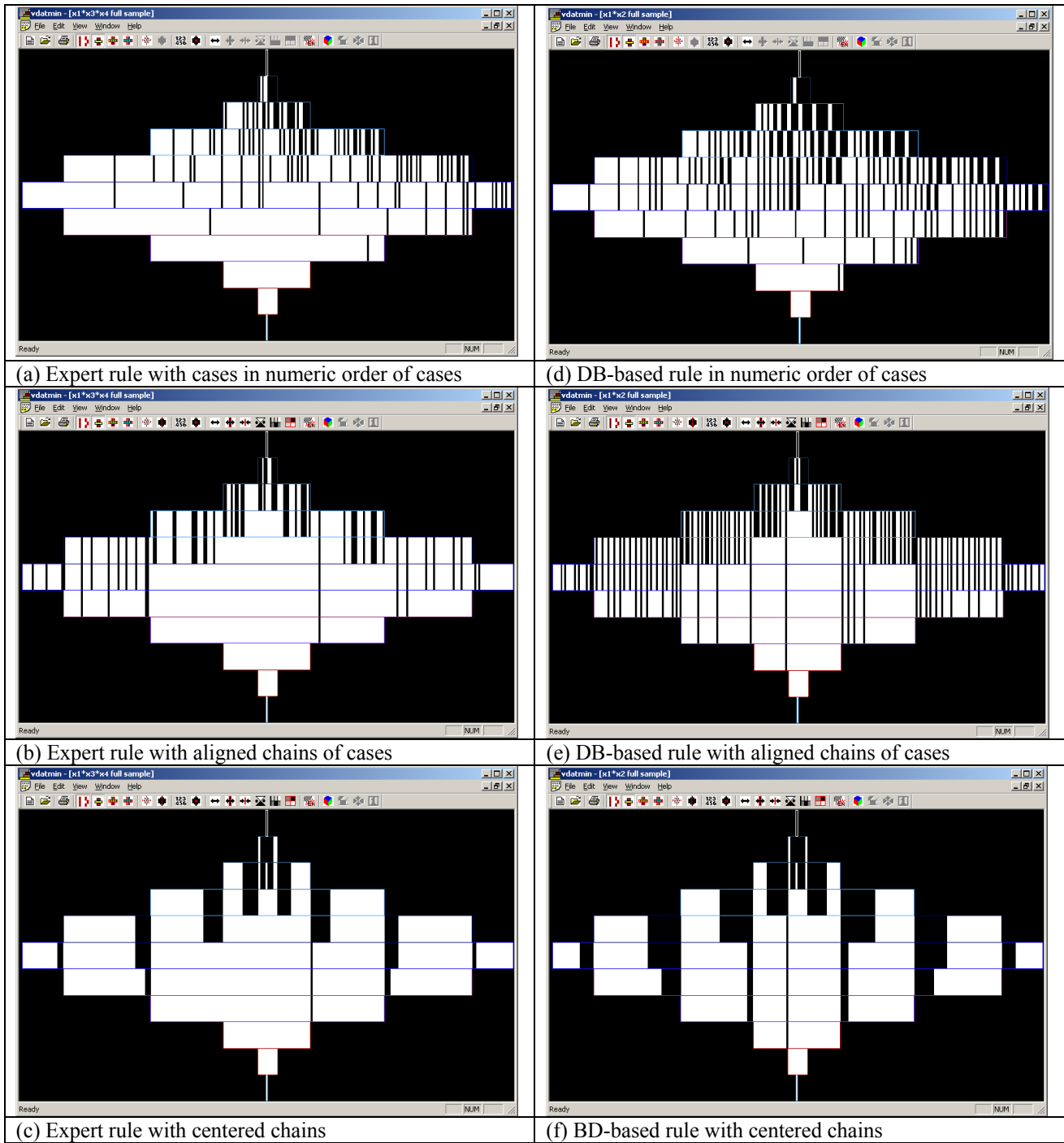


Figure 3:(a)-(c) Visualization of the expert cancer rule in 10-D feature space, (d)-(f) visualization of the closest cancer rule extracted from the data base 10-D feature space.

The VDATMIN also allows using other bar colors to indicate the status of the vector. For instance, 2(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 $f(y)=0$ is derived from the target value for the vector x , such that

$y \leq x$ and $f(x)=0$. Alternatively, the vector y is rendered as a *dark grey bar* if $y \geq x$ and $f(x)=1$. In this case $f(y)=1$. Vector y is called an *expanded vector*. The idea is that if the monotonicity hypothesis holds, then the value of $f(y)$ can be derived by expanding the value $f(x)$ 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 $f(x)=1$. 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 4 explains the visualization used in Figure 3 (b)-(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 4(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 4(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.

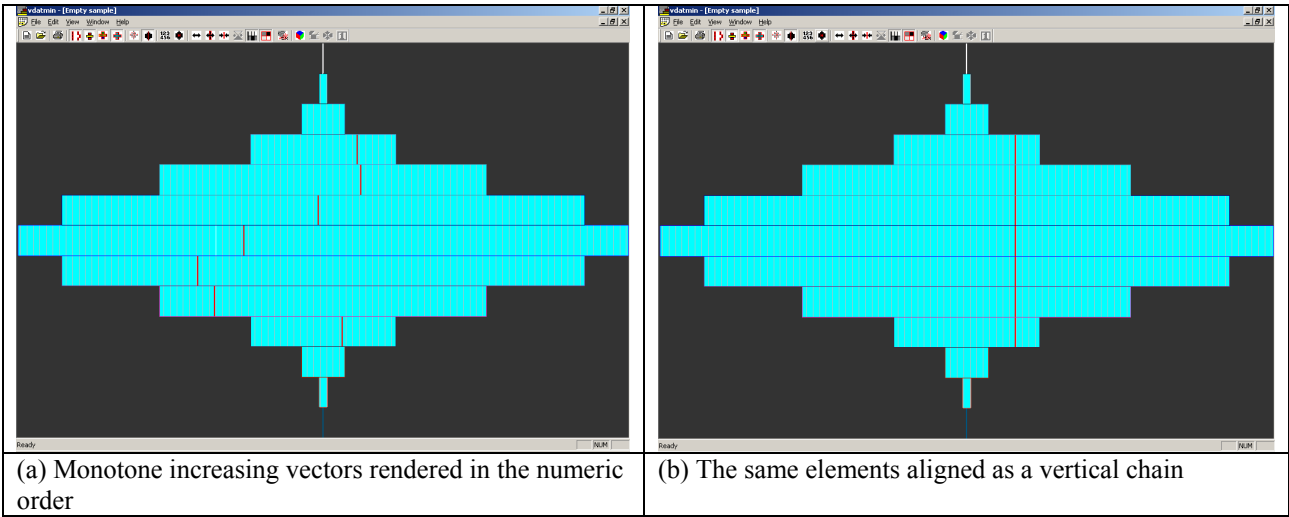


Figure 4. Alternative visualizations of increasing vectors

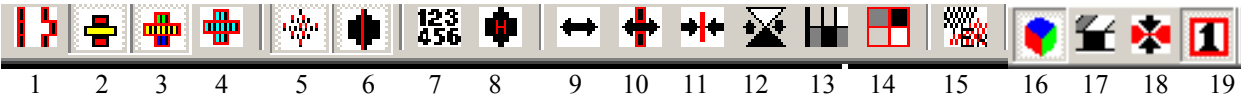


Figure 5. Details of the user interface to support viewing monotone chains of elements. (1) Align Chains vertically, (2) Show Layer Borders, (3) Show Element Borders, (4) Show Element Borders in one color, (5) Show Bush Up (elements that are greater than a selected element), (6) Highlight a chain, (7) 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 5 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.

Figure 6 shows the visualization of the Boolean rule $f(\mathbf{x}) = x_1 \& x_2$ in 11-dimensional space, e.g. if in $\mathbf{x}=(x_1, x_2, x_3, \dots, x_{11})$ we have $x_1=x_2=1$, $x_i=0$, $i=3,4,\dots,11$ then $f(\mathbf{x})=1$. In Figure 6, 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 $x_1=x_2=1$, $x_i=0$, $i=3,4,\dots,11$. All other vectors on the same layer with norm $|\mathbf{x}|=2$ are white because they can not have $x_1=x_2=1$. The next layer ($|\mathbf{x}|=3$) contains 9 vectors and respectively 9 black bars. In Figure 6(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 $x_1=x_2=1$, $x_i=0$, $i=3,4,\dots,11$ is shown on the right end of the layer with $|\mathbf{x}|=2$.

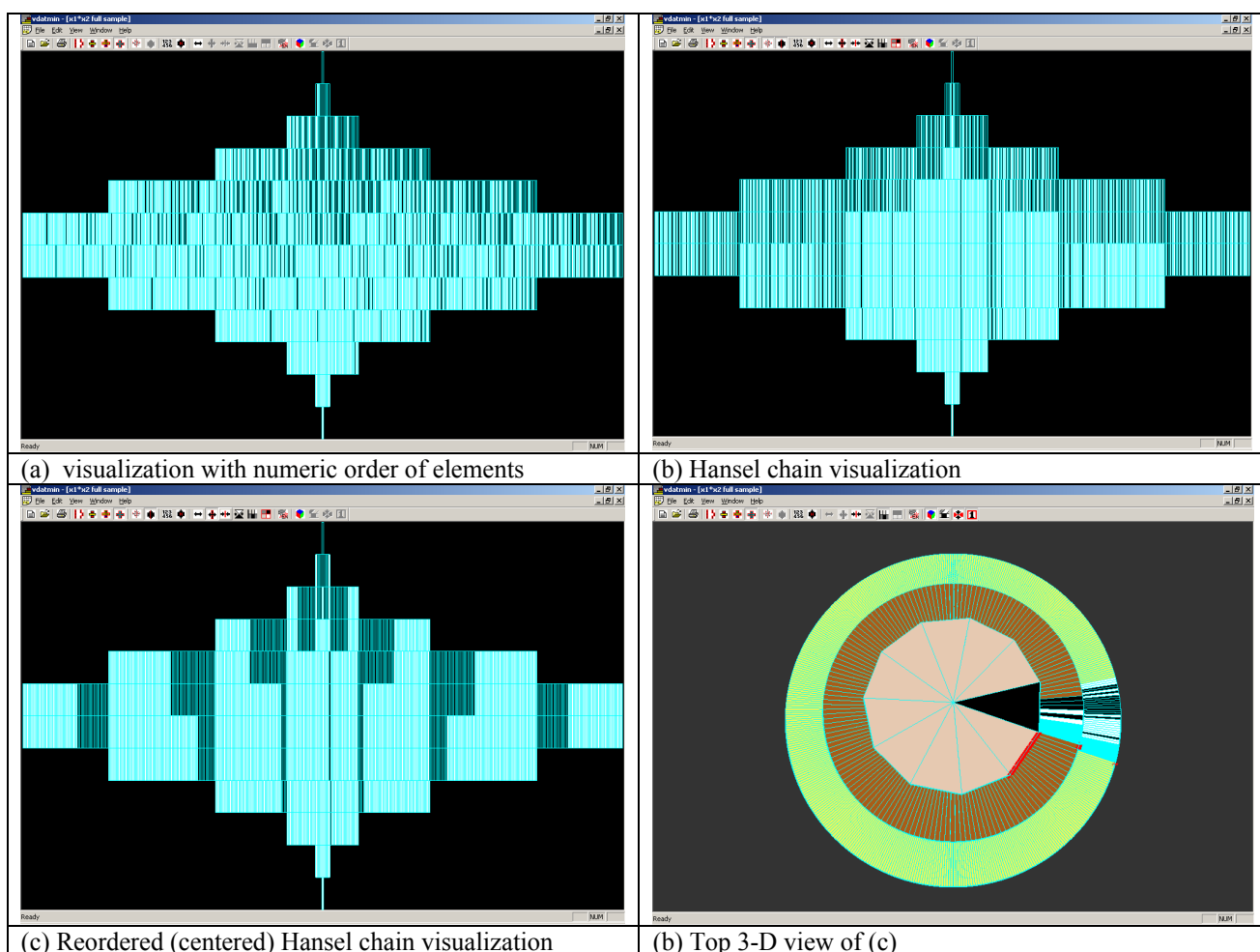


Figure 6. The visualization of the Boolean rule $y = x_1 \& x_2$ in 11-dimensional space.

Figure 6(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.

2 VISUALIZATION FOR BREAST CANCER DIAGNOSTICS

We already presented in Figure 3 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 7.

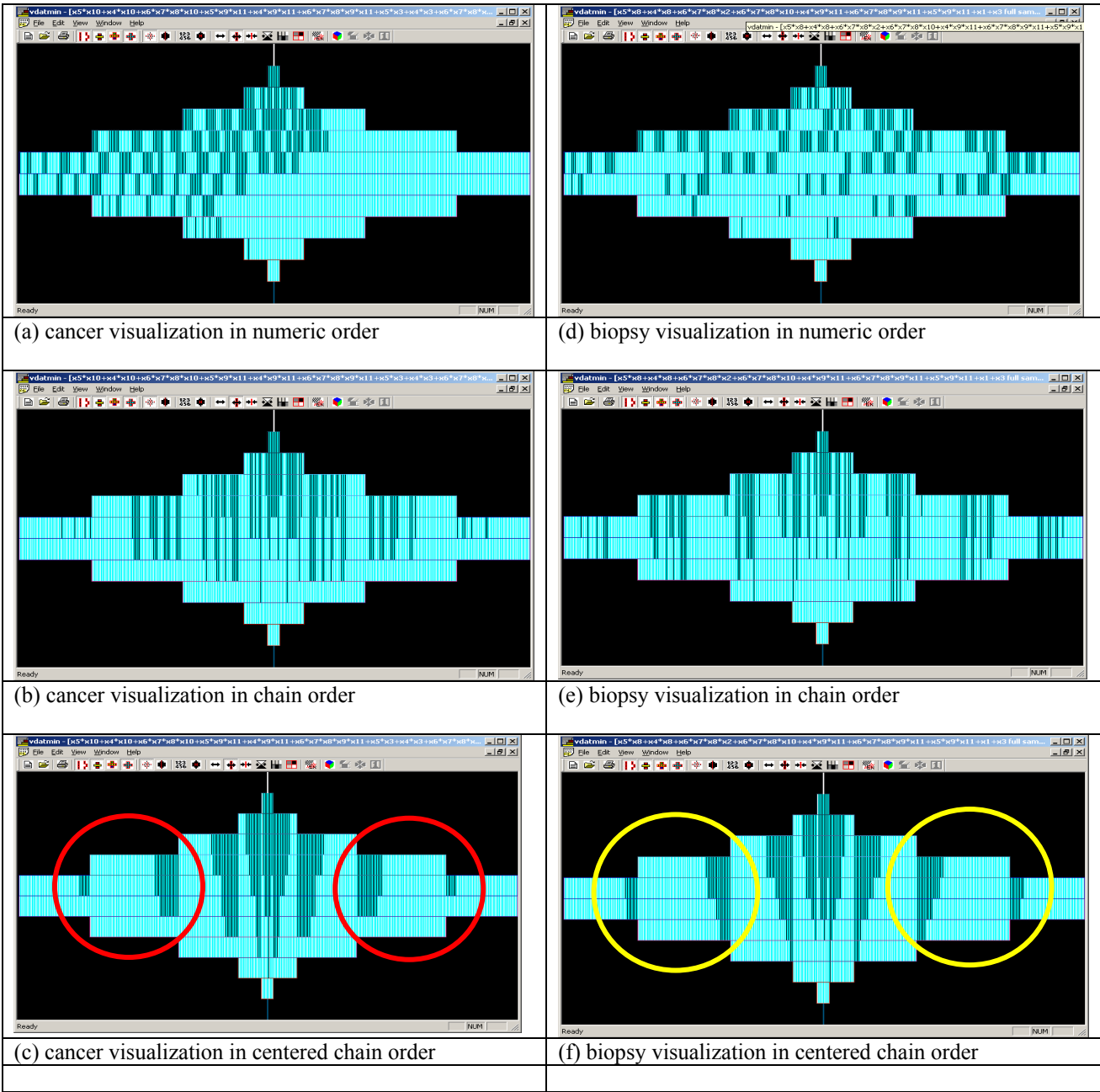


Figure 7. Visualization of expert cancer and biopsy rules

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 7 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 7 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 pattern should be lower or at the same as cancer pattern for them to be consistent. This is exactly the case as Figure 7 (c) and (f) show. The black areas in the ovals in Figure 7 (f) for biopsy are lower than the same areas for cancer in Figure 7 (f).

4. GENERAL CONCEPT OF USING MDF IN DATA MINING

Figure 8 illustrates the general concept of simultaneous coordinated visualization of multiple components of the analytics. This figure shows that rules “extracted” from the expert in this example are coordinated and aligned with each other, the data and the visual pattern better than the rule extracted from data by analytical data mining algorithm.

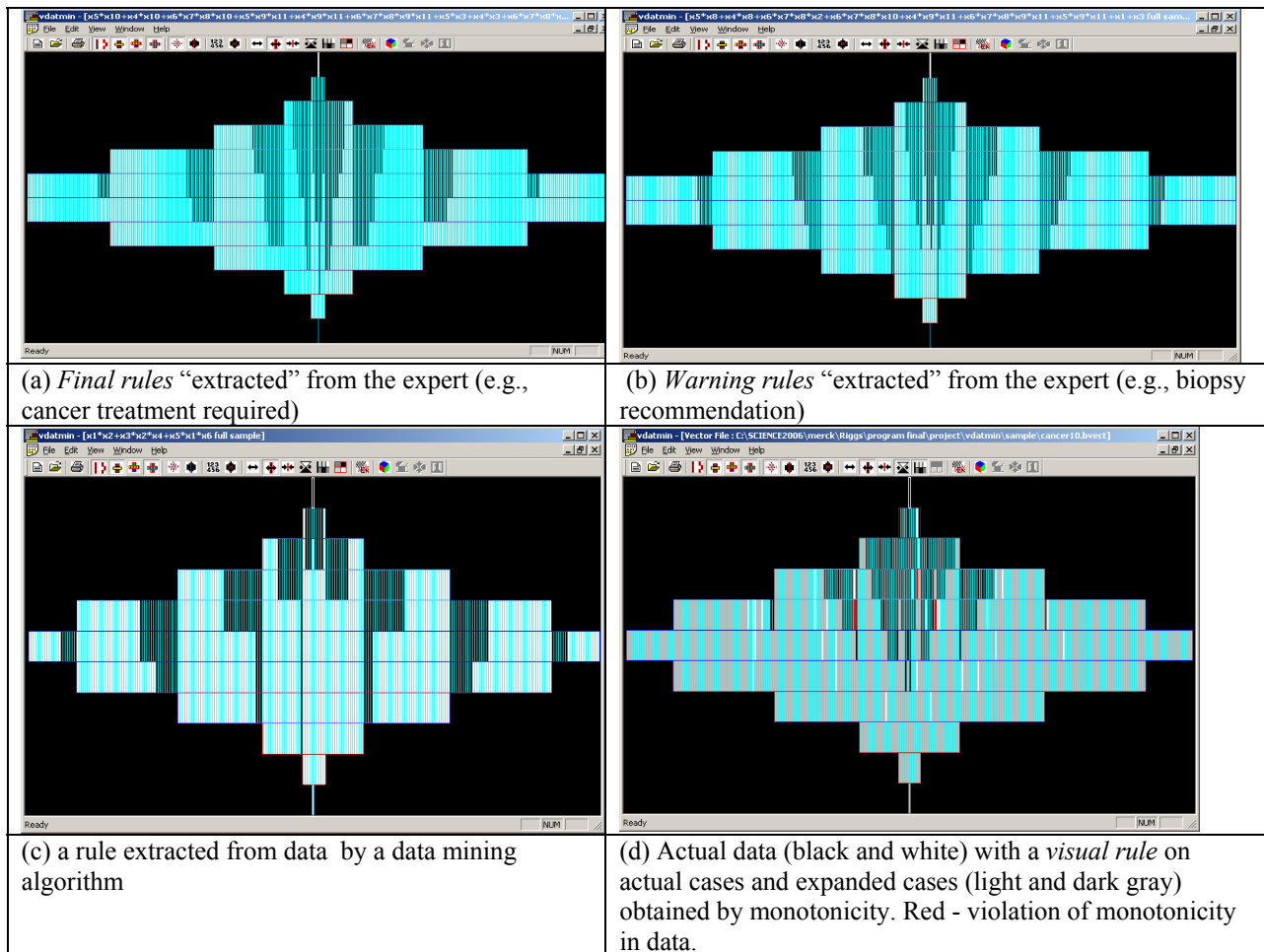


Figure 8. Visual comparison of rules provided by the expert (a), (b), extracted from data (c) and “visual rule” (d).

4. SCALING ALGORITHMS

4.1 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 9 and 10:

Step 1: Order all vectors according to their Hamming norm.

Step 2: Loop: for each vector v_i starting from the first one find all nodes that are greater than this node, $v_i < v_j$. This will create a matrix $M = \{m_{ij}\}$, where $m_{ij} = 1$ if $v_i < v_j$ 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 from 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 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 contain 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 in 100-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 $f(z)=1$). 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 knowing the value $f(L(C_1))=1$ we can expand this value to $z=E(C_1, C_2)$. Thus, this value $f(z)=1$ 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 $L(C_j)$, $D(L(C_i), L(C_j))$ 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.

4.2 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 11. 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 11(a) in each column. Figure 11(b) shows the lower edge of the border. Similarly, an upper border is visualized.

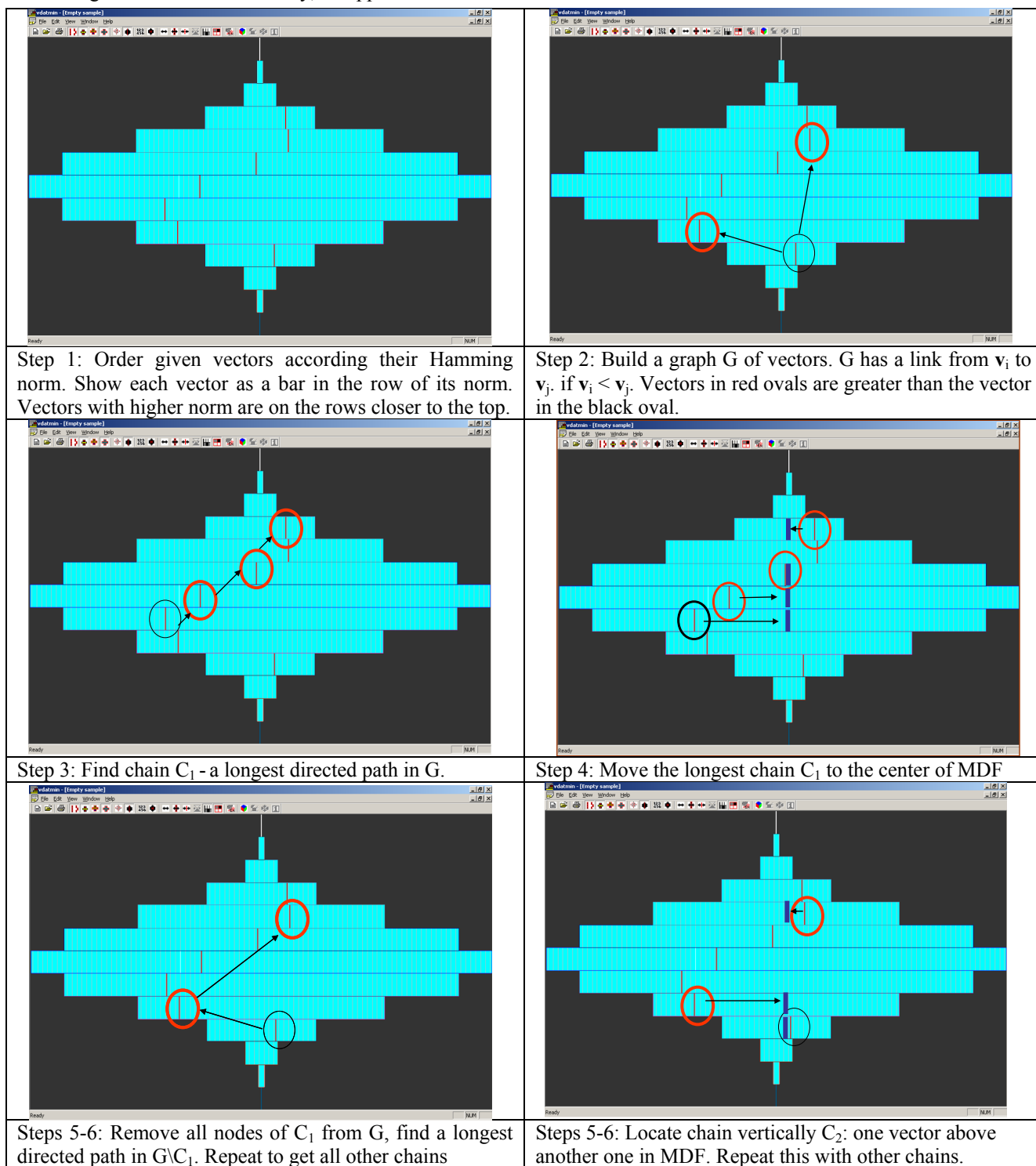
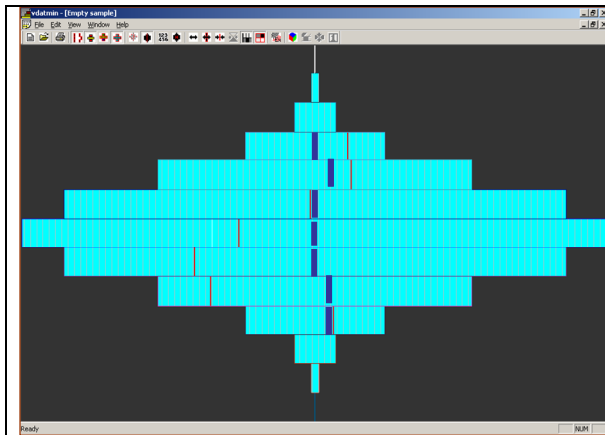
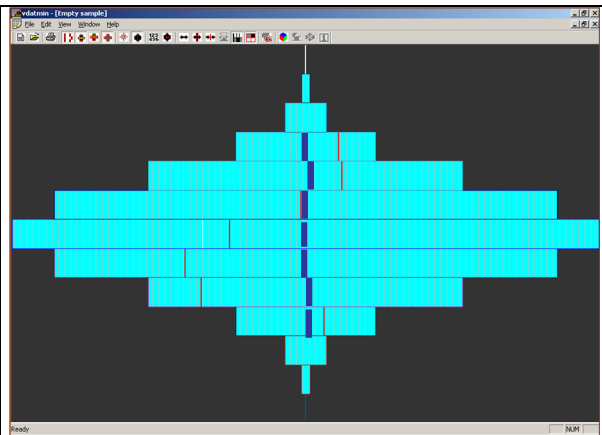


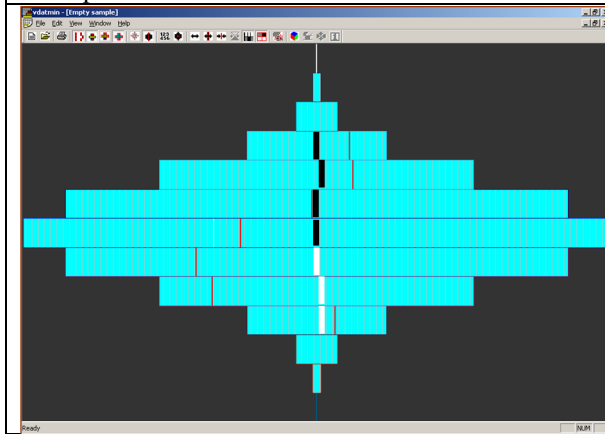
Figure 9. Illustration of algorithm with data-based chains: part 1



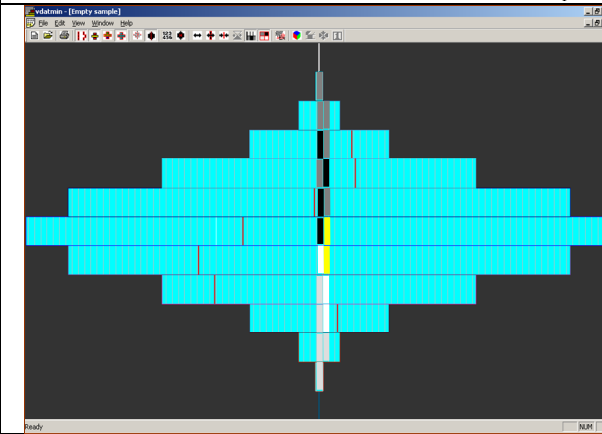
Steps 5-6: Chains C_1 and C_2 located vertically as a result of Steps 5-6.



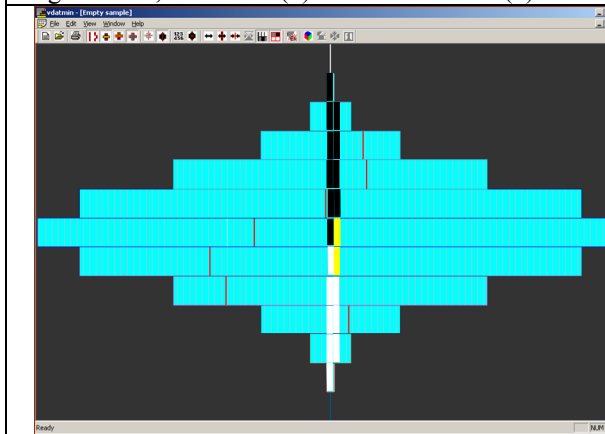
Steps 7-8: Move chains according to their distance to C_1 . The chains with the shorter distance are closer to C_1 .



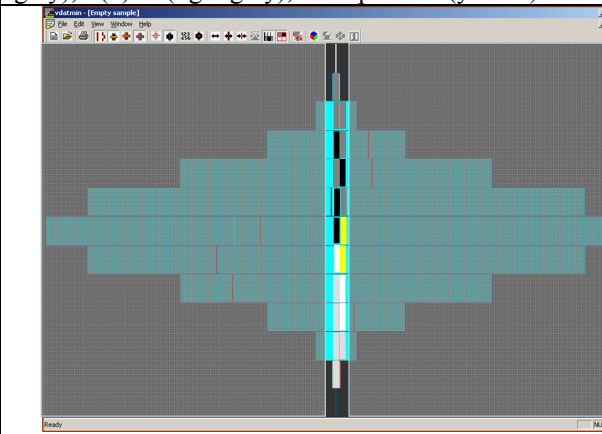
Step 9: Assign colors to vectors on each chain based on target values, black for $f(x)=1$ and white for $f(x)=0$.



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 border better.



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

Figure 10. Illustration of the algorithm with data-based chains: part 2

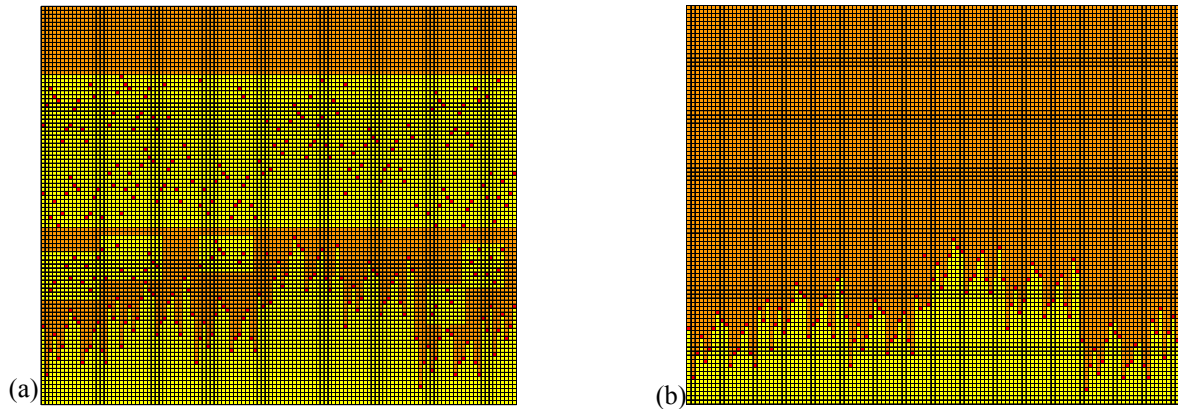


Figure 11 TTT. Pixel-based chain visualization: (a) all chains, (b) lower border between classes

5. CONCLUSION

This paper presented the concept of Monotone Boolean Function Visual Analytics (MBFVA). The 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) SME, (ii) the analytical data mining algorithm, and (iii) the visual means. 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 patterns 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., Spence, M., "Visual interactive data mining with InfoZoom-the Medical Data Set," 3rd European Conf. on Principles and Practice of Knowledge Discovery in Databases, <http://lisp.vse.cz/pkdd99/Challenge/spenke-m.zip>, PKDD '99, (1999).
- [2] Groth, D., Robertson, E., "Architectural support for database visualization," Workshop on New Paradigms in Information Visualization and Manipulation, 53-55 (1998).
- [3] Hansel, G., "Sur le nombre des fonctions Bool'eeenes 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", Proceedings of IEEE Visualization '90, Los Alamitos, CA, IEEE Computer Society Press, 360-375 (1990).
- [5] Keim, D., Hao Ming C., Dayal, U., Hsu Meichun. "Pixel bar charts: a visualization technique for very large multiattributes data sets," Information Visualization, 1(1), 20-34 (2002).
- [6] Kovalerchuk, B., Delizy, F., 2005, "Visual Data Mining using Monotone Boolean functions", Kovalerchuk, B., Schwing, J., (eds) Visual and Spatial Analysis, Springer, 387-406 (2005).
- [7] Kovalerchuk, B., Triantaphyllou, E., Despande, A., Vityaev, E., "Interactive Learning of Monotone Boolean Functions. Information Sciences," 94(1-4, 87-118 (1996).
- [8] Kovalerchuk, B., Vityaev, E., Ruiz, J., "Consistent and complete data and "expert" mining in medicine," Medical Data Mining and Knowledge Discovery, Springer, 238-280 (2001).
- [9] Shaw, C., Hall, J., Blahut, C., Ebert, D., Roberts. A., "Using shape to visualize multivariate data," CIKM'99 Workshop on New Paradigms in Information Visualization and Manipulation, ACM Press, 17-20 (1999).
- [10] Ward, M., "A taxonomy of glyph placement strategies for multidimensional data visualization," Information Visualization 1, 194-210 (2002).
- [11] Schulz, H., Nocke, T., Schumann, H., "A framework for visual data mining of structures", ACM International Conf. Proc Series; Vol. 171, Proc. 29th Australasian Computer Science Conf., v 48, Hobart, 157 - 166, (2006).
- [12] Badjio, E., Poulet, F., "Dimension Reduction for Visual Data Mining", Stochastic models and data analysis (ASMDA-2005), <http://conferences.telecom-bretagne.eu/asmda2005/IMG/pdf/proceedings/266.pdf>, (2002).
- [13] Zhao, Kaidi, Liu Bing, Tirpak, T.M., Xiao Weimin, "A visual data mining framework for convenient identification of useful knowledge," Data Mining, Fifth IEEE International Conference on, 8 pp. Digital Object Identifier: 10.1109/ICDM.2005.16 (2005).