

Visual Data Mining and Discovery in Multivariate Data using Monotone n-D Structure

Boris Kovalerchuk¹, Alexander Balinsky²

¹Department of Computer Science, Central Washington University, 400 E. University Way, Ellensburg, WA 98926-7520, USA

²Cardiff School of Mathematics, Cardiff University, Senghennydd Road, Cardiff, CF24 4AG, UK

Abstract. Visual data mining (VDM) is an emerging research area of Data Mining and Visual Analytics to gain a deep visual understanding of data. A border between patterns can be recognizable visually, but its analytical form can be quite complex and difficult to discover. VDM methods have shown benefits in many areas, but these methods often fail in visualizing highly overlapped multidimensional data and data with little variability. We address this problem by combining visual techniques with the theory of monotone Boolean functions and data monotonization. The major novelty is in visual presentation of structural relations between n-dimensional objects instead of traditional attempts to visualize each attribute value of n-dimensional objects. The method relies on n-D tone structural relations between vectors. Experiments with real data show advantages of this approach to uncover a visual border between classes.

Keywords: Data Mining, Visualization, Structural relations, Monotonicity, Multidimensional data.

1 Introduction

Visual data mining is an emerging research area of Data Mining and Visual Analytics [Thomas et al., 2005] to gain a deep visual understanding of data. The purpose of this paper is to develop a technique for visualizing and discovering patterns and relations from *multidimensional binary data* using the technique of monotone of Boolean functions. *Visualizing the border* between patterns is one of especially important aspects of visual data mining. In many situations, a user can easily catch a border visually, but its analytical form can be quite complex and difficult to discover. The simple borders of patterns that are visually far away from each other match our intuitive concept of the pattern and increase confidence to the robustness of data mining result as discovered patterns.

Deep visual understanding is a goal of visual data mining [Beilken & Spenke, 1999]. Known visualization techniques such as parallel coordinates have been successfully used in visual data mining, but for highly overlapped multidimensional

data, the resulting visualization often is unsatisfactory with a high-level occlusion. This problem is especially challenging in medical applications tracked with binary symptoms and complex dynamic optimization problems.

VDM is especially challenging task when data richness should be preserved without the excessive aggregation [Keim et al., 2002]. Another challenge is a lack of natural 3-D space and time dimensions in many tasks [Groth, 1998] that requires the visualization of abstract features. Quite often visual representations suffer from subjectivity, poor scalability, inability to perceive more than 6-8 dimensions, and the slow interactive examination of complex data [Last, Kandel, 1999, Keim et al., 2002].

Glyph or iconic visualization is an attempt to encode multidimensional data using parameters such as the shape, color, transparency, and orientation of 2-D or 3-D objects (2-D icons, cubes, or more complex “*Lego-type*” 3-D objects) [Ebert et al, 1996; Post, van Walsum et al., 1995; Ribarsky et al., 1994]. 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]. Glyph methods commonly use data dimensions as positional attributes (x,y,z) to place glyphs, which often result in occlusion.

Other glyph methods place glyphs using *implicit or explicit structure* within the data set. 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*). In this approach, some attributes are *implicitly* encoded in the data structure while others are *explicitly* encoded in the glyph/icon. Thus, if the structure carries ten attributes and a glyph/icon carries nine attributes, nineteen attributes are encoded. We use simple 2-D icons as *bars* of different colors. Adding texture, motion and other icon characteristics can increase dimensions of data visualized. On the other hand collapsing of the bar to a single pixel makes GPDS approach more scalable.

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 (≤ 6). Other visualization methods, known as Scatter, Splat, Map, Tree, and Evidence Visualizer, that are implemented in MineSet (Silicon Graphics), permit up to eight dimensions to be shown on the same plot by using color, size, and animation of different objects [Last & Kandel, 1999].

The *parallel coordinate technique* [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. Parallel coordinates visualize explicitly *every* attribute x_i of an n -dimensional vector (x_1, x_2, \dots, x_n) in 2-D and place the vector using *all attributes* x_i but each attribute is placed on its own parallel coordinate *independently* of 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*.

A typical example of current research in high-dimensional Visual Analytics is the work of Wilkinson et al. [2006], which uses the *2D distributions of orthogonal*

pairwise projections of the multidimensional Euclidean space. 2-D projections may not discover real n -D patterns and glyphs often lead to occlusion. We attempt to show only relevant structural relation between n -D vectors in 2-D or 3-D not actual n -D vectors. For more extensive representation of related work, see [Peng et al, 2004, Mackinlay, 1997, de Oliveira, Levkowitz, 2003, Yang et al, 2007].

The proposed method to represent n -D data in 2-D or 3-D is based on the following principles of visual representations, learning, and discovery:

Principle 1: *Represent complexity not abstract multiplicity.* The visual system was developed evolutionary to deal efficiently with dynamics of complex concrete objects in 3-D world. This ability does not imply the same efficiency to deal with multiple abstract multidimensional objects.

Principle 2: *Represent concrete complexity in low dimensions.* The human visual system has unique abilities to understand concrete complex information received via visual channels in low dimensions (2-D and 3-D). Therefore, it is desirable to transform the n -D data to concrete 2-D or 3-D entities.

Principle 3: *Represent individual attributes of n -D data in 2-D/3-D only if necessary.* This will help to minimize occlusion and loss of information. The parallel coordinates method that represents all attributes of n -D data suffers from occlusion.

Principle 4: *Represent structural relations between n -D data in 2-D or 3-D that have a clear meaning for the analyst.* Such an attempt can avoid occlusion and loss of information.

Principle 5: *A human can capture structural relations such as order (hierarchy, generalization) and monotonicity in n -D.* Therefore, these relations are first candidates to be a base for n -D data representation in 2-D and 3-D.

Principle 6: *If n -D data have no recorded natural hierarchy and n -D monotonicity, modify data representation, learn, discover, and build these structures in a new representation with clear meaning for the analyst.* Our suggested hierarchical generalization and monotonization process is derived from this principle. Also our process of learning new cases from training data by using monotone extension is derived from this principle.

This paper is organized as follows. Section 2 contains definitions and mathematical statements. Section 3 describes the proposed visual representation n -D Boolean space in 2-D. Section 4 describes a learning process. The paper concludes with computational experiments and an outline of the further research and summary.

2. Chains and similarity distances between chains

We denote a set of all n -D binary vectors E^n , (n -D binary cube), which is a partially ordered set that form a lattice with a max node $(1,1,\dots,1)$ and min node $(0,0,\dots,0)$ for the relation \leq on vectors $\mathbf{a}=(a_1,\dots,a_n) \in E^n$ and $\mathbf{b}=(b_1,\dots,b_n) \in E^n$,

$$\mathbf{a} \leq \mathbf{b} \Leftrightarrow \forall a_i \leq b_i$$

Only some vectors in E^n are ordered in this way. Those ordered vectors form chains.

Definition. A chain H in E^n is an ordered set of vectors (*lattice nodes*) $\{h_i\}$ such that $h_1 \leq h_2 \leq \dots \leq h_k$

Definition. A set of n -D Boolean vectors, \mathbf{a} , \mathbf{b} , \mathbf{c} , and \mathbf{d} is called a *square* (see Fig. 1a) if $\mathbf{a} < \mathbf{b} < \mathbf{d}$, $\mathbf{a} < \mathbf{c} < \mathbf{d}$ and $|\mathbf{a}|+1=|\mathbf{b}|=|\mathbf{c}|=|\mathbf{d}|-1$, where $|\cdot|$ is the *Hamming norm*,

$$|\mathbf{x}| = \sum_{i=1}^n x_i, \text{ that is the number of 1s in the vector.}$$

Definition. Squares $S_1 = \langle \mathbf{a}_1, \mathbf{b}_1, \mathbf{c}_1, \mathbf{d}_1 \rangle$ and $S_2 = \langle \mathbf{a}_2, \mathbf{b}_2, \mathbf{c}_2, \mathbf{d}_2 \rangle$ are called *adjacent* if $\mathbf{c}_1 = \mathbf{b}_2$. See Fig. 1b.

Definition. Chains H_1 and H_2 are *adjacent* if one of them contains three nodes \mathbf{a} , \mathbf{b} and \mathbf{d} of the square S and another one contains the fourth node \mathbf{c} of the square S . We will also say that H_1 and H_2 are *adjacent in square S* , or *S -adjacent*.

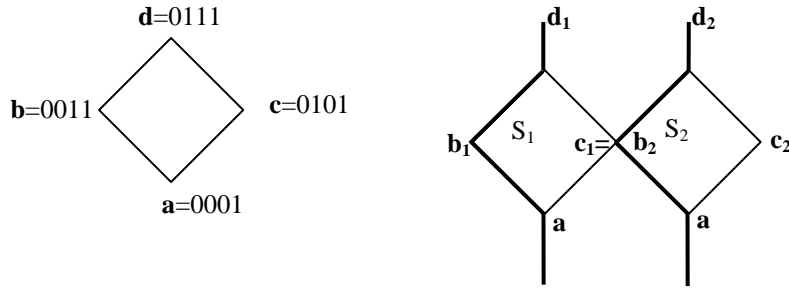


Fig. 1. (a) Example of Boolean squares, (b) adjacent squares and chains

Here node \mathbf{a} is the min node, \mathbf{d} is max node, and \mathbf{b} and \mathbf{c} are intermediate nodes. The square is the simplest *lattice* with incomparable nodes \mathbf{b} and \mathbf{c} . We assume that \mathbf{a} , \mathbf{b} , \mathbf{c} , and \mathbf{d} belong to two chains, where \mathbf{a} , \mathbf{b} and \mathbf{d} belong to the first chain H_1 and \mathbf{c} belongs to the second chain H_2 . Below we define Hansel chains [Hansel, 1966, Kovalerchuk et al, 1996]

Definition. A Hansel chain H in E^1 consists of vectors 1-D vectors (0) and (1).

Definition. A set of Hansel chains $\{H\}$ in E^2 consist of chains $1H = \langle (10) \rangle$ and $0H = \langle (00), (10), (11) \rangle$. These chains do not overlap and cover E^2 completely. To simplify notation we will write $0H = \langle 00, 10, 11 \rangle$ and $1H = \langle 10 \rangle$.

Definition. A set of Hansel chains $\{H\}$ in E^3 consist of chains presented in table 1. These chains do not overlap and cover E^3 completely.

Table 1. Hansel chain nodes in E^3

Chain	Norm 0	Norm 1	Norm 2	Norm 3
Chain 0H1	000	001	011	111
Chain 1H1		100	101	
Chain 0H2		010	110	

The *general process* of defining *Hansel chains* for E^{n+1} that cover E^{n+1} completely and without overlap is as follows [Hansel, 1966, Kovalerchuk et al, 1996]. It uses recursively two processes of cloning and growing-cutting of chains already constructed in E^n . If H is a chain in E^n then it produces two chains in E^{n+1} . First H is *cloned*, that is we produce two copies of H . Then the first copy is used to produce a chain $0H$ in E^{n+1} where each element of H is expanded by adding a leading zero, e.g.

Having a single chain $H = \langle 0, 1 \rangle$ in E^1 : all chains for higher n are produced by cloning and growing H . The application of these steps in E^3 produces chains in E^4 (see table 2), then repeated to get chains in E^5 from shown chains in E^4 (see Table 3).

In particular, cloned chains in E^2 are: $0H = \langle 00, 01 \rangle$ and $1H = \langle 10, 11 \rangle$. Each of these chains contains two Boolean vectors. Thus, in E^2 we have two Hansel chains after the grow-cut process is applied: $0H = \langle (00), (10), 11 \rangle$ and $1H = \langle 10 \rangle$. For building Hansel chains in E^3 having $H = \langle 00, 01, 11 \rangle$ in E^2 we get first $0H = \langle 000, 001, 011 \rangle$. Next, we produce a chain $1H$ with leading 1 from the second copy of H , $1H = \langle 100, 101, 111 \rangle$. The *growth-cut step* is cutting the max node from $1H$ and growing $0H$ by adding the max node to $0H$. This produces chains $0H = \langle 000, 001, 011, 111 \rangle$ and $1H = \langle 100, 101 \rangle$ with 4 and 2 elements respectively. In E^2 , we have two chains $\langle 00, 01, 11 \rangle$ and $\langle 10 \rangle$. We have shown how chain $\langle 00, 01, 11 \rangle$ produces chains in E^3 . Similarly, chain $\langle 10 \rangle$ that consists of a single node produces first two chains in E^3 : $0H = \langle 010 \rangle$ and $1H = \langle 110 \rangle$ that also consist of single nodes. Then the grow-cut process produces $H0 = \langle 010, 110 \rangle$ and empty $1H$.

 Table 2. Hansel chains in E^4

Chain	Norm 0	Norm 1	Norm 2	Norm 3	Norm 4
0H1	0000	0001	0011	0111	1111
1H1		1000	1001	1011	
0H2		0100	0101	1101	
1H2			1100		
0H3		0010	0110	1110	
1H3			1010		

 Table 3. Hansel chains in E^5

Chain	Norm 0	Norm 1	Norm 2	Norm 3	Norm 4	Norm 5
00H1	00000	00001	00011	00111	01111	11111
10H1		10000	10001	10011	10111	
01H1		01000	01001	01011	11011	
11H1			11000	11001		
00H2		00100	00101	01101	11101	
10H2			10100	10101		
01H2			01100	11100		
00H3		00010	00110	01110	11110	
10H3			10010	10110		
01H3			01010	11010		

*Chains 11H2 and Chain 11H3 are empty and omitted.

Statement 1. If chains H_1 and H_2 are adjacent then $|a_1 - a_2| = |b_1 - c_1| = |b_1 - b_2| = |d_1 - d_2| = 2$.

Proof. According to the definition of square S_i , $|a_i| = |c_i| - 1 = |b_i| - 1$, therefore the Hamming distance between a_i and c_i is 1, $|a_i - c_i| = |a_i - b_i| = 1$. Also $c_i = b_2$, thus $|c_i| = |b_2|$ and $|a_i| = |a_2|$. Therefore, $|a_i| = |a_2| = |c_i| - 1$. Also $a_1 \neq a_2$ because they belong to different chains. Vector a_1 alters c_1 in one attribute and vector a_2 alters c_1 in another attribute. Say, $c_{1i} = 1$ and $c_{1j} = 1$, then alterations are $a_{1i} = 0$, and $a_{2j} = 0$. Thus, the distance between a_1 and a_2 is 2, $|a_1 - a_2| = 2$. Similarly, d_1 and d_2 alter c_1 in two positions, thus, $|d_1 - d_2| = 2$.

Statement 2 The minimal distance between two unequal Boolean vectors x and y , $x \neq y$, with equal norms $|x| = |y|$ is 2, $|x - y| = 2$.

Proof. If this distance would be equal to 1, $|\mathbf{x}-\mathbf{y}| = 1$, then it will be only one attribute i such that $x_i \neq y_i$, say $x_i=0$ and $y_i=1$. This will imply that $|\mathbf{x}| < |\mathbf{y}|$ which contradicts the condition that $|\mathbf{x}| = |\mathbf{y}|$.

Statements 1 and 2 tell us that adjacent chains H_1 and H_2 are closest chains (measured by the Hamming distance between nodes of adjacent squares).

Theorem. If H_1 and H_2 are adjacent Hansel chains then for every chain $\mathbf{h}_1 \in H_1$ and chain $\mathbf{h}_2 \in H_2$ with equal norms, $|\mathbf{h}_1|=|\mathbf{h}_2|$ the Hamming distance is equal to 2, $|\mathbf{a}_1-\mathbf{a}_2| = 2$, that is

$$\forall \mathbf{h}_1 \in H_1 \forall \mathbf{h}_2 \in H_2 (|\mathbf{h}_1| = |\mathbf{h}_2| \Rightarrow |\mathbf{h}_1-\mathbf{h}_2| = 2). \quad (1)$$

In other words, the distance between vectors of the same norm in the adjacent Hansel chains is the constant 2 and this is the smallest distance if $H_1 \neq H_2$.

Proof. It follows from the recursive design of the Hansel chains in E^{n+1} by using chains already constructed in E^n as described above. Consider chains $0H=\{0\mathbf{h}_i\}$ and $1H=\{1\mathbf{h}_i\}$ in E^{n+1} produced by cloning chain $H=\{\mathbf{h}_i\}$, $i=1:k$, in E^n before the grow-cut process is applied. We will use notation $\mathbf{h}_i=(h_{i1}, \dots, h_{in})$. Nodes are numbered in such way that $i=1$ for the node with the smallest norm, $|\mathbf{h}_i|=\min\{|\mathbf{h}_j|\}, i=1:k$. Chains $0H$ and $1H$ have the equal number of elements k . Also $|1\mathbf{h}_i|=|0\mathbf{h}_i|+1$ and the norm of the i -th node of $1H$, $1\mathbf{h}_i$ is the norm of $0\mathbf{h}_{i+1}$, $|1\mathbf{h}_i|=|0\mathbf{h}_{i+1}|$, because adding leading 0 in $0H$ does not change the norm, but adding 1 in $1H$ increases it by 1. For $i=1$, $|1\mathbf{h}_1|=|0\mathbf{h}_1|+1$, thus, norms of all nodes in $1H$ are greater than the norm of the first node of $0H$, $0\mathbf{h}_1$. Similarly, the largest node of $1H$, $1\mathbf{h}_k$ has no node with equal norm in $0H$ (before it will be transferred to $0H$ by the grow-cut process and become node $0\mathbf{h}_{k+1}$). This consideration tells us that $|1\mathbf{h}_i|=|0\mathbf{h}_{i+1}|$, $i=2, \dots, k-1$.

Let us compute the Hamming distance between $1\mathbf{h}_i$ and $0\mathbf{h}_{i+1}$ using decomposition:

$$\begin{aligned} |1\mathbf{h}_i - 0\mathbf{h}_{i+1}| &= \sum_{j=1}^{n+1} |1h_{ij} - 0h_{i+1,j}| = |1h_{i,1} - 0h_{i+1,1}| + \sum_{j=2}^{n+1} |1h_{ij} - 0h_{i+1,j}| = \\ |1-0| + \sum_{j=2}^{n+1} |h_{ij} - h_{i+1,j}| &= 1 + |\mathbf{h}_i - \mathbf{h}_{i+1}| = 1 + 1 = 2 \end{aligned}$$

Here we separated from the sum the leading digit ($j=1$). The remaining parts are \mathbf{h}_i and \mathbf{h}_{i+1} . We used the property of Hansel chains that $|\mathbf{h}_{i+1}| = |\mathbf{h}_i|+1$ and $|\mathbf{h}_i - \mathbf{h}_{i+1}|=1$.

This proof is for chains before the growth-cut process is applied. This process does not change the norm of any element of the chains; it only relocates the largest node $1\mathbf{h}_k$. The theorem is not applicable for this node, because the condition of the theorem in (1) was $|\mathbf{h}_1| = |\mathbf{h}_2|$ which is not true for $1\mathbf{h}_k$, as its norm is greater than the norm of any other node in $0H$ and $1H$. This theorem is illustrated in Tables 4-6.

Statement 3. The distance between n -D Boolean vectors $\mathbf{a}=\{a_j\}$ and $\mathbf{b}=\{b_j\}$ with equal norms, $|\mathbf{a}|=|\mathbf{b}|$, is an even number.

Proof. If r is the number of bits j where $a_j = 1$ and $b_j = 0$ then there should be r other bits where $a_j = 0$ and $b_j = 1$ to have $|\mathbf{a}_1|=|\mathbf{a}_3|$. In this case the total number of bits where \mathbf{a} and \mathbf{b} differ is $2r$, which is their Hamming distance, $|\mathbf{a}_1-\mathbf{a}_3|=2r$.

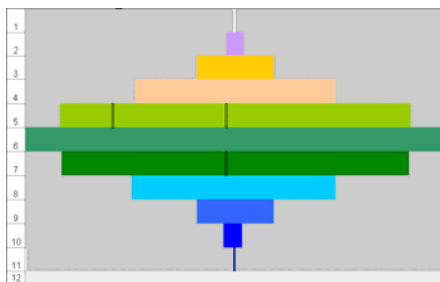
Statement 4. If H_1 and H_2 are adjacent Hansel chains and H_2 and H_3 are also adjacent Hansel chains, then the Hamming distance $|\mathbf{a}_1-\mathbf{a}_3|$ between every \mathbf{a}_1 from H_1 and every \mathbf{a}_3 from H_3 such that $|\mathbf{a}_1|=|\mathbf{a}_3|$ is equal to 2, or 4.

Table 7. Distances D between Hansel chains in E^5

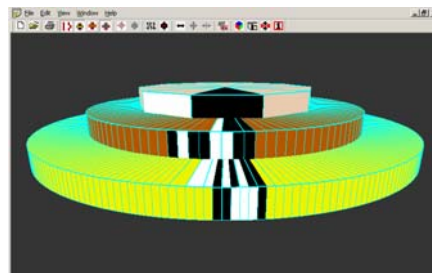
	00H1	10H1	01H1	11H1	00H2	10H2	01H2	00H3	10H3	01H3
00H1	0	2	2	4	2	3	4	2	2	3
10H1		0	2	2	2.5	3	4	2.5	2	3
01H1			0	2	2	3	2	2.5	3	2
11H1				0	4	3	2	4	3	2
00H2					0	2	2	2	4	4
10H2						0	2	3	2	2
01H2							0	2	2	2
00H3								0	2	2
10H3									0	2
01H3										0

3. Representing and drawing of n-D Boolean space in 2-D

Below we describe a structure to allocate Boolean vectors in 2-D. Vectors are ordered vertically by their Boolean norm (sum of “1”s) with the largest vectors rendered on the top, starting from (11111). This is a way to *use relations between attributes* for placing Boolean vectors. Each vector will be first placed in the MDF and then drawn as a colored bar: white for the 0 class, black for the 1 class. Fig. 3 depicts the hierarchy of levels of Boolean vectors in E^{10} , that is 11 levels from 0 to 10, where level 0 contains a single vector (0000000000) and level 10 contains a single vector (1111111111). Several vectors are shown in Fig. 3 as **small bars**. The top row bar shows vector (1111111111). The vector on the third row cannot be identified from this figure without additional assumptions, but we can tell for sure that it has eight “1”s, because of its location on the third line. To specify it more we link each vertical position (column) with a specific vector. We can use decimal values of Boolean vectors for this. Assume that Boolean vectors with the same norm are ordered as decimal numbers, where the leftmost column presents the largest decimal vector for the given norm, (1111111100) and the rightmost position is used for vector (0011111111) in the third row. Each level in Fig. 3a is called a disk and the entire visualization is called the **multiple disk form (MDF)**. In the MDF, every vector has a fixed horizontal and vertical position as shown in Fig. 3.



(a)



(b)

Fig. 3. (a) 2-D Multiple Disk Form (MDF) representation of 10-dimensional Boolean space without occlusion. (b) A 3-D version of MDF with grouping Hansel chains.

The fundamental advantage of MDF representation in comparison with methods reviewed in section 1 is that there is *no occlusion* of vectors in MDF. In Fig. 3 all 1024 vectors are completely visible and their attributes are represented implicitly but unambiguously. The advantage of this procedure is to allow the user to compare more than one binary dataset or Boolean function at a time. This representation uses a distance between vectors as numbers, $D_N(n(\mathbf{a}),n(\mathbf{b})) = |n(\mathbf{a})-n(\mathbf{b})|$, where $n(\mathbf{a})$ and $n(\mathbf{b})$ are numbers that vectors \mathbf{a} and \mathbf{b} represent.

The disadvantage of such MDF implementation called *process P₀* is that distance D_N may not be relevant to the application domain. Thus, the task is to locate vectors in MDF in way that will capture relations/structure that are *relevant* to the domain. The *Hamming distance* D_H captures relevant relations in many domains. A partial order relation \leq between vectors $\mathbf{a} \leq \mathbf{b} \Leftrightarrow \forall a_i \leq b_i$ is one of them that represent the structure of the attribute space. The following statement shows the link between D_H and this partial order.

Statement: If $D_H(\mathbf{a},\mathbf{b})=1$ then $\mathbf{a} < \mathbf{b}$ or $\mathbf{b} < \mathbf{a}$. If $\mathbf{a} < \mathbf{b}$ and $D_H(\mathbf{a},\mathbf{b})=k$ then a chain exist such that there are $k-1$ nodes between \mathbf{a} and \mathbf{b} on this chain.

Algorithm. The idea of the algorithm for visual *n*-D Boolean data representation is to locate chains on the multiple disc form (MDF). In the previous consideration [Kovalerchuk, Delizi, 2005] any relocation of chains in MDF was allowed to make a visually appealing border between classes. This captures relations between nodes along the chains but not between chains.

There is also a requirement to include “transversal” dependences between classes. Horizontal dependence can appear from similarity between chains and from the user’s input. A user can provide additional information on dependence between different measurements/attributes in a form of a weighted graph of dependences.

Now we impose limitations to ensure that *similar chains are located next to each other* not only in the vertical position of their lower units but also in similarity of other chains’ nodes. We call this requirement a *Local Similarity Principle* (LSP). After that, it will be possible to combine the LSP and measurement dependences by combining the Hamming norm and the distance on the graph of measurements.

A **new algorithm** uses the statements and the theorem from section 2 and has three major steps:

- (1) putting the largest chain H to the center of MDF,
- (2) ordering other chains relative to their closeness to H in the *averaged Hamming chain similarity measure*, $D_{HC}(U, H)$ defined in section 2,
- (3) ordering chains with equal *similarity measure* D_{HC} to H relative to the location of the borders between patterns on these chains.

For data that satisfy the monotonicity test it is the closeness of the height (Hamming norm) of their lower unit (see section 4 for detail). Tables 8-11 show chains for E^1 - E^5 located using this algorithm.

Table 8. Horizontal drawing of Hansel chains in E^1

Chain	Norm 0	Norm 1
H	0	1

Table 9. Horizontal drawing of Hansel chains in E^2

Chain	<i>Norm 0</i>	<i>Norm 1</i>	<i>Norm 2</i>
0H	00	01	11
1H		10	

Table 10. Horizontal drawing of Hansel chains in E^3

Chain	<i>Norm 0</i>	<i>Norm 1</i>	<i>Norm 2</i>	<i>Norm 3</i>
0H2		010	110	
0H1	000	001	011	111
1H1		100	101	

Table 11. Horizontal drawing of Hansel chains in E^4 in accordance with chain distances

Chain	<i>Norm 0</i>	<i>Norm 1</i>	<i>Norm 2</i>	<i>Norm 3</i>	<i>Norm 4</i>
1H3			1010		
0H2		0100	0101	1101	
0H1	0000	0001	0011	0111	1111
1H1		1000	1001	1011	
0H3		0010	0110	1110	
1H2			1100		

Table 11. Horizontal drawing of Hansel chains in E^5 in accordance with chain distances

Chain U	Distance $D_{HC}(U, 00H1)$	<i>Norm 0</i>	<i>Norm 1</i>	<i>Norm 2</i>	<i>Norm 3</i>	<i>Norm 4</i>	<i>Norm 5</i>
01H2	4			01100	11100		
01H3	3			01010	11010		
00H3	2			10010	10110		
00H3	2		00010	00110	01110	11110	
01H1	2		01000	01001	01011	11011	
00H1	0	00000	00001	00011	00111	01111	11111
10H1	2		10000	10001	10011	10111	
00H2	2		00100	00101	01101	11101	
10H2	3			10100	10101		
11H1	4			11000	11001		

Tables 7-12 can be converted to the visual representation as shown in Fig 4 for E^{10} . It is MDF from Fig. 3a rotated 90^0 , with Hansel chains real 10-D vectors shown white bars (benign tumors) and black bars (cancer tumor cases).

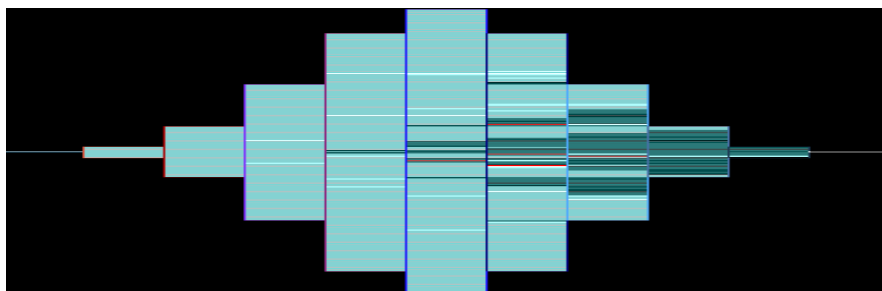


Fig. 4. MDF in E^{10} with Hansel chains and real data shown as white and black bars

4. Learning process by monotone extension and monotonicity

Learning algorithms in data mining and machine learning include: (1) discovering a regularity R using training and testing data for specific classes, (2) generalizing R to new data $\{\mathbf{x}\}$, and (3) computing $R(\mathbf{x})$ to get a class of \mathbf{x} . Often learning algorithms do not separate (1)-(3), but only produce a border between classes as a hyperplane or other discriminate surface in n-D. The lack of explicitly formulated regularity R is a flaw of such algorithms. In this section, we describe a process of learning monotone regularities R (in n-D Boolean space) that are explicitly defined.

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

$$\forall \mathbf{x} \geq \mathbf{y} \Rightarrow f(\mathbf{x}) \geq f(\mathbf{y}) \quad (2)$$

Definition. A Boolean function is called a *binary regularity* R if $R: E^n \rightarrow E$, where $E = \{0,1\}$ is a set of two labels (label of class 0, and label of class 1).

Definition. A binary regularity is called a *monotone regularity* if R is a monotone Boolean function.

Discovering monotone regularity. At first, we *discover monotonicity* on training data T_r by testing property (2) on all pairs of training data (vectors \mathbf{x}, \mathbf{y} with known R values. The computational process can be shorten by using Hansel chains. If monotonicity of R is confirmed on T_r , then it is tested with test vectors T_t . If this test is also successful, then we *generalize* R , that is we assume that R is a monotone Boolean function (monotone regularity) and test this generalization. Then we apply the generalized R as a monotone Boolean function to new cases by using **monotone extension** that holds for monotone functions

$$(\mathbf{x} \geq \mathbf{a} \ \& \ R(\mathbf{a})=1) \Rightarrow R(\mathbf{x})=1; \quad (\mathbf{a} \geq \mathbf{y} \ \& \ R(\mathbf{a})=0) \Rightarrow R(\mathbf{y})=0,$$

Here \mathbf{a} is a vector from training data with known class $R(\mathbf{a})$, e.g., benign or malignant and \mathbf{x} is a new vector with unknown $R(\mathbf{x})$. We do this expansion using Hansel chains. If \mathbf{a} belongs to chain \mathbf{h} , then for all $\mathbf{x} > \mathbf{a}$ on this chain we assign $R(\mathbf{x})=1$. Similarly if $\mathbf{a} > \mathbf{x}$ and $R(\mathbf{a})=0$, then we assign $R(\mathbf{x})=0$ for all \mathbf{x} below \mathbf{a} on that chain. Next, the same process is applied to expand to \mathbf{x} on other Hansel chains.

Border between classes. Let H be a chain, and $X = \{\mathbf{x}\}$ be a subset of Boolean vectors from chain H such that $\forall \mathbf{x} R(\mathbf{x})=1$. Set X can be a set of training data of class 1 on chain H .

Definition. Boolean vector $\mathbf{x}_{\min 1} \in X$ is a *lower one in X for $R, X \subseteq H$* , if

$$\forall \mathbf{x} \in X, \mathbf{x} \neq \mathbf{x}_{\min 1} \quad \mathbf{x}_{\min 1} < \mathbf{x}.$$

Similarly, let H be a chain, and $Y = \{\mathbf{y}\}$ is a subset of Boolean vectors from chain H such that $\forall \mathbf{x} R(\mathbf{x})=0$. Set X can be a set of training data of class 1 on chain H .

Definition. Boolean vector $\mathbf{x}_{\max 0} \in X$ is an *upper zero in X for $R, X \subseteq H$* , if

$$\forall \mathbf{x} \in X \quad \mathbf{x} \neq \mathbf{x}_{\max 0} \quad \mathbf{x}_{\max 0} > \mathbf{x}.$$

If $X=H$ then $\mathbf{x}_{\max 0}$ is an upper zero in of the chain and $\mathbf{x}_{\min 1}$ is a lower unit of the chain H for R . In the case of $X=H$, $\mathbf{x}_{\max 0} < \mathbf{x}_{\min 1}$ and $|\mathbf{x}_{\max 0} - \mathbf{x}_{\min 1}|=1$, that is $\mathbf{x}_{\min 1}$ is the next node on the chain after $\mathbf{x}_{\max 0}$. These two nodes form a border between classes, $R(\mathbf{x})=0$ and $R(\mathbf{x})=1$ on chain H .

Definition. A set of all lower ones and upper zeros, $\{\mathbf{x}_{\max 0}\} \cup \{\mathbf{x}_{\min 1}\}$ for X and R on all Hansel chains in E^n is called a *border* between classes in E^n .

Definition. A width of the border between classes in E^n on chain H for R is the Hamming distance between $\mathbf{x}_{\max 0}$ and $\mathbf{x}_{\min 1}$ on H , $|\mathbf{x}_{\max 0} - \mathbf{x}_{\min 1}|$.

The smallest width of this border is $|\mathbf{x}_{\max 0} - \mathbf{x}_{\min 1}|=1$. A wider border between classes indicates a better separation of classes for a given training dataset.

Monotonization. The monotonicity test may end up with discovering its violations. There are four such cases in Fig. 4. If 5% threshold was set up this can be an acceptable generalization of data having only 4 violations out of 100 training cases. Otherwise, the process of data monotonization is applied. It can be done by: (1) changing or modifying attributes (e.g., by changing scales of attributes or splitting a single non-monotone attribute to two monotone attributes), (2) discovering limits of local monotonicity and decomposing a non-monotone Boolean function to a logical combination of monotone Boolean functions.

5 Computational Experiments

We conducted several computational experiment that show wide and narrow borders between patterns with different level of border clarity. This depends on both data and VDM process.

Different way of locating Boolean vectors on MDF form produce very different results relative to the shape of the border between classes (see Fig. 5). They range from no border visible at all (Fig. 5a) to a very clear border (Fig 5d).

Experiments with real breast cancer data show advantages of this approach to uncover a visual border between benign and malignant cases in breast cancer. Fig. 4 shows analysis of multivariate monotonicity of breast cancer cases that is almost perfect with four exclusions that are shown in red. The MDF design with Hansel chains located horizontally expands scalability of this visualization. To be able see a large number of attributes a user can use zooming or scrolling MDF.

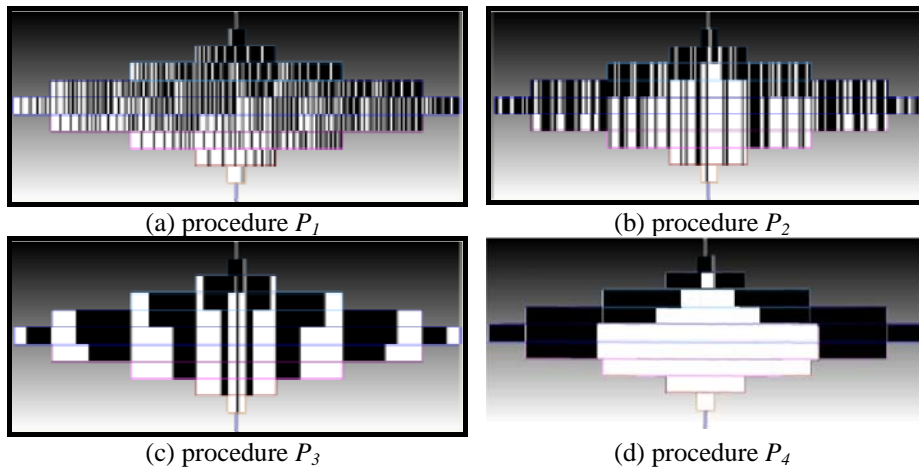


Fig. 5. Different visual border representations of vectors of two classes in MDF for the same simple monotone Boolean function.

6. Conclusion

Visual data mining had shown benefits in many areas. However, classical VDM methods do not address the specific needs for processing data that are highly overlapped in the visual space. This paper proposed a structural VDM approach and shown its effectiveness in applications. To use this approach for Boolean vectors of a high dimensionality n we visualize only a part of the data structure that is actually needed for given data.

We exploit monotonicity properties to build the *complete visual border* between two classes such as expected and unexpected cases. Note that traditional machine learning and data mining methods do not built a complete border with high confidence, because they cannot exploit monotonicity for arbitrary features.

The proposed method allows discovering the border between expected and unexpected cases for monotonized set of diagnostic features. In mathematical terms, this task is equivalent to the task known in discrete math as *restoration of a monotone Boolean function* [Hansel, 1966]. The main idea of this process is based on decomposition of the binary cube E^n into Hansel chains that cover all nodes of E^n lattice without overlap of chains. Each Hansel chain consists of set of ordered n -D Boolean vectors ($\mathbf{x} > \mathbf{y} \Leftrightarrow \forall i x_i > y_i$). The set of upper zeros for all chains form a *border* of the *expected cases* and a set of lower ones form a border of the all *unexpected cases*.

By further developing these procedures for non-monotone Boolean data and k -valued data structures, this approach can be used in variety of applications including tasks, where data dynamically changed/updated and the visual border between patterns also dynamically changes.

References

1. Beilken, C., Spenke, M., Visual interactive data mining with InfoZoom -the Medical Data Set. The 3rd European Conference on Principles and Practice of Knowledge Discovery in Databases, PKDD '99,
2. Card, S. K., Mackinlay, J., The structure of the information visualization design space. In Proc. IEEE Symposium on Information Visualization, pages 92-99, 1997
3. Ebert, D., Shaw, C., Zwa, A., Miller, E., Roberts, D. Two-handed interactive stereoscopic visualization. IEEE Visualization '96 Conference.1996, 205-210.
4. Hansel, G. (1966), "Sur le nombre des fonctions Boolenes monotones den variables". C.R. Acad. Sci. Paris, v. 262, n. 20, 1088-1090.
5. Groth, D., Robertson, E., Architectural support for database visualization, Workshop on New Paradigms in Information Visualization and Manipulation, 53-55,1998, 53-55.
6. Inselberg, A., Dimsdale, B., Parallel coordinates: A tool for visualizing multidimensional Geometry. Proceedings of IEEE Visualization '90, Los Alamitos, CA, IEEE Computer Society Press, 1990, 360-375.
7. Keim, D., Ming C. Hao, Dayal, U., Meichun Hsu. Pixel bar charts: a visualization technique for very large multiattributes data sets. Information Visualization, March 2002, Vol. 1, N. 1, pp. 20-34.

14 **Boris Kovalerchuk¹, Alexander Balinsky²**

8. Kovalerchuk, B., Triantaphyllou, E., Despande, A., Vityaev, E., Interactive Learning of Monotone Boolean Functions. *Information Sciences*, Vol. 94, issue 1-4, pp. 87–118, 1996.
9. Kovalerchuk, B., Vityaev, E., Ruiz, J., Consistent and complete data and “expert” mining in medicine. *Medical Data Mining and Knowledge Discovery*, Springer, 2001:238–280.
10. Kovalerchuk, B., Delizy F., Visual Data Mining using Monotone Boolean Functions, In: *Visual and Spatial Analysis* (Eds. Kovalerchuk B., Schwing J.), Springer 2005, 387-406.
11. Last, M., Kandel, A., Automated perceptions in data mining, invited paper. *IEEE International Fuzzy Systems Conference Proc. Part I*, Seoul, Korea, 1999, pp. 190–197.
12. de Oliveira, M., Levkowitz, H., From Visual Data Exploration to Visual Data Mining: A Survey. *IEEE Trans. On Visualization and Computer Graphics*, 2003, Vol. 9, No. 3, pp. 378-394
13. W. Peng, M.O. Ward, and E.A. Rundensteiner, “Clutter Reduction in Multi-Dimensional Data Visualization Using Dimension Reordering,” *Proc. IEEE Information Visualization*, pp. 89-96, 2004.
14. Post, F., T. van Walsum, Post, F., Silver, D., Iconic techniques for feature visualization. In *Proceedings Visualization '95*, pp. 288–295, 1995.
15. Ribarsky, W., Ayers, E., Eble, J., Mukherja, S., Glyphmaker: creating customized visualizations of complex data. *IEEE Computer*, 27(7), 57–64, 1994.
16. 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, 1999, 17-20.
17. J. Thomas, K. Cook, eds. *Illuminating the Path: The Research and Development, Agenda for Visual Analytics*. National Visualization and Analytics Center, 2005.
<http://nvac.pnl.gov/agenda.stm>
18. Ward, M., A taxonomy of glyph placement strategies for multidimensional data visualization. *Information Visualization* 1, 2002, 194–210.
19. Wilkinson, L. Anand A., Grossman R., "High-Dimensional Visual Analytics: Interactive Exploration Guided by Pairwise Views of Point Distributions," *IEEE Transactions on Visualization and Computer Graphics*, vol. 12, no. 6, pp. 1363-1372, Nov/Dec, 2006
20. Di Yang, Rundensteiner E., Ward, M., Analysis Guided Visual Exploration of Multivariate Data, *Proceedings of the IEEE Symposium on Visual Analytics*, 2007.