Visual Data Mining in Closed Contour Coordinates

Boris Kovalerchuk, Dept. of Computer Science, Central Washington University, Ellensburg, WA, USA
Vladimir Grishin, View Trends, Ltd., Port St. Lucie, FL, USA

Abstract

This research is motivated by a long-standing problem of ineffective heuristic initial selection of a class of models, and its structures in modern data mining, machine learning, and other fields. Such heuristics usually are due to insufficient prior knowledge to select a class of models, and inability to represent visually and losslessly the complex high-dimensional data to explore the data for a model class selection. For instance, lossy visualization with different 2-D projections requires an unrealistic review of a vast amount of these projections and the abilities to reconstruct from them the n-D data structures. To make the selection of a class of models faster and more efficient in this paper new closed-contour-coordinate displays are proposed and explored both mathematically and experimentally. Such displays losslessly map all attributes of each n-D data point into a separate 2-D graph/figure. This allows using the unique power of human vision to compare in parallel the hundreds of features of these graphs, and proportionally speed up the selection of an appropriate class of models. This paper includes results of visual data mining for real data sets, including the experimental results of visual feature extraction using this approach. It expands our previous results demonstrated on simulated data and shows the radical advantages of these coordinates vs. parallel coordinates for data dimensions from 20 to 200.

1 Introduction

Challenges. A common approach for developing predictive and optimization models for real world tasks is selecting a class of mathematical models, and then identifying their parameters using available data. The prior knowledge of the field, task and data typically is *uncertain*, *insufficient* or *confusing* to make this selection on the solid scientific basis. For instance, using prior knowledge for a data mining supervised classification task a person may state that attribute x_1 is three times more important than attribute x_2. Does it mean exactly three times or about 3 times? What does it mean "about three times"? Does it mean that the weighted sum model $3x_1+x_2$ will express correctly this prior knowledge for correct classification of n-D points? As a result of this prior knowledge uncertainty selection of the class of models is rather an *art* than science. One of the major difficulties for scientifically sound selection of the class of models is that we cannot see the structure of data in multidimensional space by a naked eye, which is critical for identifying the model class.

As a result we cannot make sense of such Big high-dimensional data and **cannot select a correct class of predictive and/or optimization models** for such n-D data. In contrast we often successful in model selection with 2-D or 3-D data that we can observe with a naked eye. Thus in the multidimensional case for data mining we are in essence **guessing the class of models** in advance, e.g., linear regression, decision trees, SVM, linear discrimination, linear programming, SOM and so on. This is often is hidden when mass media claim that Data Mining/Machine Learning (DM/ML) methods take out guesswork. DM/ML methods take out guesswork only relative to *wild guesses* without using any data systematically. Therefore we deliberately use the "guesswork" term instead more common "scientific" wording "research on model class selection" to unhide the guess essence of this activity.

The guesswork is not limited by selecting a class of models but also selecting a set of internal components within each class, e.g., selecting a type of kernel functions in SVM, k in the k-Nearest Neighbors method, the number of hidden layers in Neural Networks, the procedure to choose the next splitting attribute in Decision Trees and so on.

Finding the optimal set of attributes from an n-D dataset analytically for a given data mining/machine learning method, in the worst case, requires testing 2^n subsets of attributes. This is not feasible even for relatively small n. Therefore we guess smaller subsets of attributes to be tested. It is done by using multiple heuristic methods to make the task computationally feasible.

The guesswork in the model class selection is a *long-time open problem* that can be traced for millennia from Ptolemy's model of the Sun and planets. Ptolemy did not have tools to see the larger universe to propose a better model class.

Today the guesswork often is equivalent to using methods that are simply *available* in some software tools, e.g., with random check of lossy 2-D or 3D displays of n-D data. Thus, often we put the task at hand into the Procrustean bed of specific methods, which were not designed for this task and its context, but just are handy. The most logical and beneficial approach seems to be the development of methods specifically for a given task. However, it is often much more difficult than using the available software and respectively requires more time, effort, and investments. It may require going far beyond the current research approaches, and even the change of the whole research paradigm.

The *overall goal* of this paper is to propose a visual analytics approach *to decrease the guess work* and make the selection of the predictive model more scientifically rigorous, task effective and faster for the same performance. Specifically we focus on the data mining/machine learning supervised and unsupervised learning tasks of classification and clustering of n-D data. The common part of model selection for these tasks is identifying the discriminating features. These features can be the initial attributes of n-D data or complex combinations of them. Therefore this paper focuses on visual data mining task of finding discrimination features using lossless closed contour visual representation of n-D data.

What is the benefit of such visual data mining to be used alongside traditional computational data mining approaches (e.g., finding similar points by their distance in the high-dimensional space)? While distances are a backbone of many Data Mining/Machine Learning algorithms, they suffer from the same deficiency of model guessing discussed above. It is just a specific form of model guessing. The common guess is the standard Euclidian distance with equal weights of all attributes. With an infinite number of weighted Euclidian distances, which one should be used, and how to justify any of them? Next how to justify the class of weighted Euclidian distances having multiple other classes of distances? Why should we be limited by this class? All these questions are open questions in Data Mining/Machine Learning for decades. The justification of the guess is commonly substituted by the later claim that a particular model guess gave a high accuracy of classification or prediction. Unfortunately, it does not help to be

successful with the different data next time. This explains the need to go beyond traditional DM/ML methodology, and the **Visual Data Mining (VDM)** is a promising one, to approach this fundamental challenge, which is critical for both theory of data mining and applications.

The next question is how Visual Data Mining can actually help to decrease or eliminate guesses? The idea is to mimic its success with 2-D data, which is well known for years. Consider 2-D data shown in Figure 1. The common guess without looking at this figure is to try a linear discrimination function to separate the blue and red points. It will obviously fail, while a quick look at these data in Figure 1, immediately gives a visual insight of a correct model class of "crossing" linear models, with one line going over blue points, and another one going over the red points. This example also clarifies to what extent we can and should guide a viewer to search for specific visual features, e.g., crossing lines.

When we have not seen similar data before, we likely have no prior knowledge of data features. In this case the guiding can be a guess of the visual features at the same level as for a class of analytical models discusses above. The whole motivation of our VDM is to *decrease guesses* and substitute them by *lossless observations* on n-D data in 2-D exploiting *unique human abilities of recognizing 2-D visual patterns*. Respectively, naïve attempts to build a unified "consistent approach" to extract features visually based on guesses of the real data structure will fail due to lack of a scientific basis for such guesses.

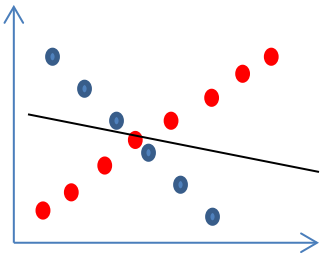


Figure 1. Two "crossing" classes that cannot be discriminated by a single straight line

We advocate more rigorous and practical ideas to build a unified "consistent approach":

(1) **Exploiting 2D-nD math links** -- establishing and using 2-D graph representation of specific mathematical n-D structures;

(2) **Deep structure analysis in 2-D** – analyzing lossless 2-D visual representation of an n-D point as a graph in 2-D instead of lossy representation of an n-D point as 2-D point that has no internal structure which limits the analysis to 2-D point clouds;

(3) **Exploiting perceptual abilities** -- selecting visual displays of n-D points as 2-D graphs (forms) derived from psychological studies of human visual form perception (Gestalt laws and others).

This approach does not guess unknown structures of given n-D data to be found, but attempts to use human abilities to detect some structures visually in specific visual representations of n-D data in 2-D. In this paper we focus on closed contours dictated by Gestalt laws. As a result only features of n-D structures that a human can detect in those 2-D representations will be found and some other important features *will not be discovered* due to inappropriate 2-D representation of them for the human for a given time. Respectively the major concern is on ensuring that the class of visual representations is large enough to enable human recognition of as many as possible real n-D structures.

This approach is motivated by extreme power and flexibility of parallel visual perception of 2-D forms, which allows simultaneously comparing hundreds of attributes of a high-dimensional point. It means that instead of one 2-D projection as in a simple lossy technique we can analyze, say, 100-D data relationships.

The success in exploiting these abilities in feature recognition depends on multiple factors such as previous experience, allotted time, order of presentation of the forms, a set of forms, locations of features on the form, specific AND or OR combination of features, projective and affine transformation of the form, "noise" in forms and in inter-relationships between them, their place in pattern hierarchy, and others. Due to such ultra-complexity, even after over 200 years of experimental research of form visual perception, psychological studies have estimated for extraction only of simple features and their attributes (figure size, orientation, concavities, etc.) for simple figures with a few features. Moreover, usually these results are correct only for a tested set of figures and the extension for another set could be questionable.

Thus, there is no available "universal" model of form visual recognition, which could answer our practical question needed for Visual Data Mining: "What is the set of form features that can be visually revealed on a certain set of complex figures for a certain time?" Such a model would be in fact, a model of human vision and brain. Note that, 50 years ago, John von Neumann said that evidently, brain model is not simpler than brain itself. So, if we want to use vision as the most powerful pattern recognition system for visual analytics and visual Data Mining, we have to experimentally test most prospective displays for some interesting n-D data structures. Gestalt laws seem to be the only common information about form perception, to allow us to select displays.

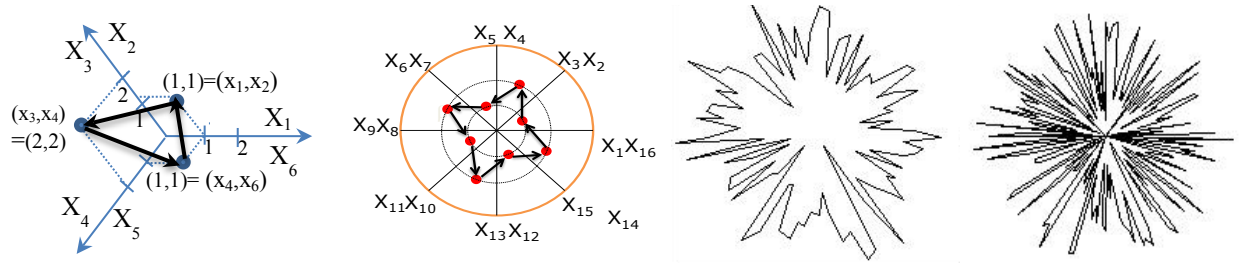
2 Approach

We propose to decrease the guess work by developing and exploiting **lossless visual representation of n-D data** in combination with analytical techniques. Visualization, visual analytics and visual data mining techniques with multidimensional data have been a subject of intensive research for years [4-6,11-14]. However many difficult problems are still open.

Principal components (PCA) in 2-D are lossy. The first two principal components of each n-D point do not contain all information that is contained in the complete n-D point. Recently we developed a large class of lossless methods to represent n-D points in 2-D [1,3,7,8] called **General Line Coordinates (GLC)** that include new Collocated Paired Coordinates (CPC) as well as well-known Parallel Coordinates and Radial (star) Coordinates.

The GLC allow coordinates go to any directions and be collocated. We had shown that several tasks including analysis of Challenger disaster data, World hunger data, and semantic meaning shift in jokes benefit from this approach [1,3,7-9]. Freeware [10] implements some of these methods. The current version of this freeware has three versions: standalone, web-based, and Excel-plugin. The approach detailed below is based on **selecting discrimination features** using lossless closed contour visual representation of n-D data. In concordance with the Gestalt laws, the closed contours such as *stars* in Radial Coordinates show the essential perceptual advantages over polylines in the Parallel Coordinates (PC), bar charts, pie charts, etc.

Shape perception features. Humans are able to detect, compare, and describe multiple figures by using hundreds of their local features such as concave, convex, angle, and wave, and combine them into a multilevel hierarchy [1,3].



- (a) (b) (c) (d)

Figure 2. Examples of n-D points as closed contours in 2-D: (a) 6-D point x=(1,1,2,2,1,1) in CPC Radial Coordinates with non-orthogonal Cartesian mapping, (b) 16-D point (1,1,2,2,1,1,2,2,1,1,2,2,1,1,2,2) in CPC Radial Coordinates with Cartesian encoding, (c) CPC star of a 192-D point in Polar encoring, (d) the same 192-D point as a traditional star in Polar encoding.

Each feature includes many attributes, e.g., size, orientation, location, and others. A term "holistic picture" denotes an image together with its description, which includes image statistics, textures, integral characteristics, forms, and coloring. Next, the holistic concept is appearing at multiple levels of image perception. First the image is considered as a set of "spot clusters", and relations between them as an overall structure of the image. Then each spot cluster is considered with the same aspects where elements are "spots", and the structure represents relations between these "spots". Next each "spot" is viewed at the holistic level in the same way, and at the levels of its elements. At these levels the features that are perceptually important include symmetry, elongation, orientation, compactness, convexity/ concavity, peaks, waves, sharp angle, inside/outside, etc.

Why do we focus on closed contours? It is based on **Gestalt laws** of human perception and 2-D organization of real world figures [1-3, 8, 19]. Almost century ago psychologists *many times experimentally revealed* fundamental Gestalt Laws of form perception and recognition by a human vision [19]. According to these laws a figure that *possesses a closure, symmetry, similarity, proximity, and continuity will be detected faster* in the presence of noise [2,19], their forms will be recognized faster and more accurately, and a *common pattern* will be specified better.

In accordance with these facts close contours (stars for short) are more effective for feature selection because:

- a star represents each n-D data point by a continuous, closed 2-D holistic figure, while Parallel Coordinates produce open "elongated" polyline, which is not percept as a closed figure;
- a star shows many invisible on Parallel Coordinates axial and central symmetries in the figure which facilitate detection similarities with other stars;
- Parallel Coordinates disrupt proximity due to discontinuity at the ends.
- Humans much easier detect similarity of different turns in stars than in Parallel Coordinates.

Gestalt Laws were verified for quite simple and *natural (common) figures* with a few features. Therefore in [1, 3] we experimentally confirmed the listed advantages for *artificial figures* with complex 2-D closed contours in comparison with open polylines in Parallel Coordinates. Traditional stars such as shown Figure 2c provided 2-3 times faster class or feature detection than polylines in Parallel Coordinates for data dimensions up to 100. CPC stars additionally extended dimension n up to 200. It was shown for n-D data *simulated* as hyper-tubes. Here we explore and verify such advantages for new General Line Coordinates displays and sets of *real* data from UCI Machine Learning repository.

For essentially better use of form perception for visual analytics we proposed in [3,8] a special type of collocated coordinates with *Polar* encodings non-orthogonal coordinates. In this paper we expand it to *Cartesian* encoding in non-orthogonal coordinates.

Figures 2 shows examples of 6-D, 16-D and 192-D points represented in 2-D as closed contours in **Radial Collocated Paired Coordinates**. For short, below we call these representations **CPC stars** and call these coordinates as **CPC Radial Coordinates**.

A 6-D point $\mathbf{x}=(x_1,x_2,x_3,x_4,x_5,x_6)=(1,1,2,2,1,1)$ with non-orthogonal Cartesian mapping is shown in Figure 2a in CPC Radial Coordinates. It is split to three 2-D pairs with the 2-D point $(x_1,x_2)=(1,1)$ located in the first sector in coordinates (X_1,X_2), point $(x_3,x_4)=(2,2)$ located in the second sector in coordinates $(_3,X_4)$, and point $(x_5,x_6)=(1,1)$ located in the third sector in coordinates (X_5,X_6). Then these points are connected sequentially to form an oriented graph. As a result a 6-D point is represented losslessly not by six 2-D points as in Parallel Coordinated but by three 2-D points, i.e., two times less 2-D points.

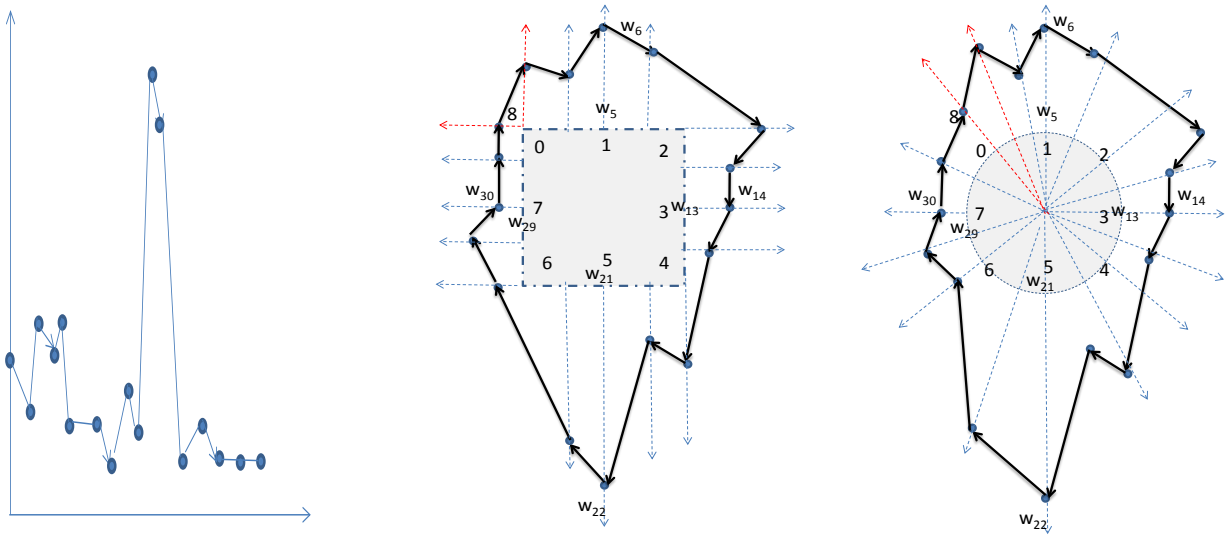
Respectively in general CPC Radial Coordinates use n/2 2–D points instead of n 2-D points and dramatically increase the abilities to visualize higher-dimensional data losslessly. In the mathematical terms, it is a theorem of half size that follows directly from the Radial CPC representation algorithm.

Theorem (half size). Any n-D point to be represented in the CPC Radial Coordinates requires n/2 2-D points for even n and (n+1)/2 for odd n.

As Figure 2 shows, in Radial CPC representation coordinates X_2 and X_3 are collocated. In the same way, X_4 and X_5 are collocated as well as all other X_j,X_{j+1} are collocated including X_1 and X_{16}. In general these CPC stars are generated as follows: a full 2π circle is divided on n/2 equal sectors. Each pair of values of coordinates (x_j, x_{j+1}) of an n-D point $\mathbf{x}$ is displayed in its own sector as a 2-D point. In the *polar mapping* this point is located at the distance $r = ({x_j}^2+{x_{j+1}}^2)^{1/2}$ from the star center, which is a Euclidean length of the projection of vector $\mathbf{x}$ on the plane of these two coordinates, with the angle α of the ray to this point from the sector start proportional to the value of x_j.

In this way we get n/2 points and connect them by straight lines (or arrows) to generate a star. This is a polar representation of *all* 2-D projections of $\mathbf{x}$ on plane. It is a lossless display forming a single connected figure without crossing lines. It satisfies all Gestalt Laws providing an effective application of form perception capabilities.

Other versions of this polar representation are produced when radius r represents x_j and angle α represents x_{j+1}, or vice versa. Alternatively, each pair (x_j,x_{j+1}) can be encoded in the (X_j,X_{j+1}) coordinates in *non-orthogonal Cartesian mapping* as show in Figure 2a.



(a) (b) (c)

Figure 3. (a) 34-D point in collocated coordinates, (b) X coordinate from (a) is mapped to the square and Y coordinate from (a) is mapped to the lines orthogonal to sides of the square, (c) X coordinate from (a) is mapped to the circle Y coordinate from (a) is et mapped to the lines orthogonal to the perimeter of the circle.

Figure 3 shows how to make closed contours for *non-radial (Cartesian) collocated paired coordinates* introduced in [7], where coordinates (X_j,X_{j+1}) are located orthogonally as in the Cartesian coordinates (X,Y), and where all odd coordinates are collocated in X, and all even coordinates are collocated in Y. Note that in the same way polylines in Parallel Coordinates can be represented as closed contours. We will call this class of coordinates as **Non-Radial Closed Contour (NRCC) Coordinates**.

The advantage of such closed contours shown in Figure 3 relative to traditional stars and CPC stars is that they do not *occlude the points near the center*, because all points are located outside of the circle or a square. Closed contours in Figure 3 also *improve visibility of some coordinates* relative other coordinates in comparison with the polar mapping of CPC Stars for high-dimensional data.

To be able to apply mapping of the X coordinate to a square or a circle shown in Figure 3 the lines in Figure 3a should not go backward. It is provided by ordering of all odd coordinates of n-D point **x** in the ascending order or by rescaling all odd coordinates to get this ascending order property. Below we present a theorem showing that this process guarantees that lines of the graph will not cross and will not go backward. This theorem is based in the following algorithm called **odd ordering algorithm:**

Step 1. Represent all nodes of n-D point **x** as a sequence of pairs $(x_1,x_2),(x_3,x_4),\ldots,(x_{n-1},x_n)$, e.g., (0.0), (1,0), (0,1), (1,1), (0,0).

Step 2. *Order* all nodes of the graph based on its first coordinate in ascending order, e.g., (0.0), (0,1), (0,0), (1,0), (1,1), i.e., $(x_1, x_2, x_5, x_6, x_9, x_{10}, x_3, x_4, x_7, x_8)$ and display a new CPC graph. This is equivalent to ordering of odd coordinates of an n-D point **x**. Figure 3a shows the result of applying of this algorithm to some n-D data point.

Theorem (planarity). The Odd ordering algorithm produces a planar CPC graph such that each edge is located on the right from the previous edges or on the same vertical line as the previous edge.

Proof. Let (x_i,x_{i+1}) and (x_{i+2},x_{i+3}) be two consecutive nodes of a CPC graph for an n-D point **x** after applying the odd ordering algorithm to **x**. As a result $x_i \le x_{i+2}$, i.e., each next edge of the CPC graph is located on the right from the previous edge or on the same vertical line as the current node. This location of the next node does not allow the next edge and further edges to cross the current edge and previous edges that are all on the left from the next edge. This means that if $x_i = x_{i+2}$ the next edge is on the same vertical line as the previous one.

The next algorithm is the **Complete odd ordering algorithm** where after odd ordering, all nodes with equal first coordinate are ordered in ascending order relative to its second coordinate, e.g., (0.0), (0,0), (0,1), (1,0), (1,1), i.e., $(x_1, x_2, x_9, x_{10}, x_5, x_6, x_3, x_4, x_7, x_8)$.

The **Complete ordering algorithm** orders *all coordinates* of **x** in the ascending order, e.g., for (0,0,1,0,0,1,1,1,0,0) the order is $(x_1, x_2, x_4, x_5, x_9, x_{10}, x_6, x_7, x_8)$= (0.0,0,0,0,0,1,1,1,1).

Corollary. The planarity theorem is true for both the complete **o**dd ordering**,** and the complete ordering algorithm.

Proof. Both orderings satisfy the requirements of odd ordering required for the planarity theorem.

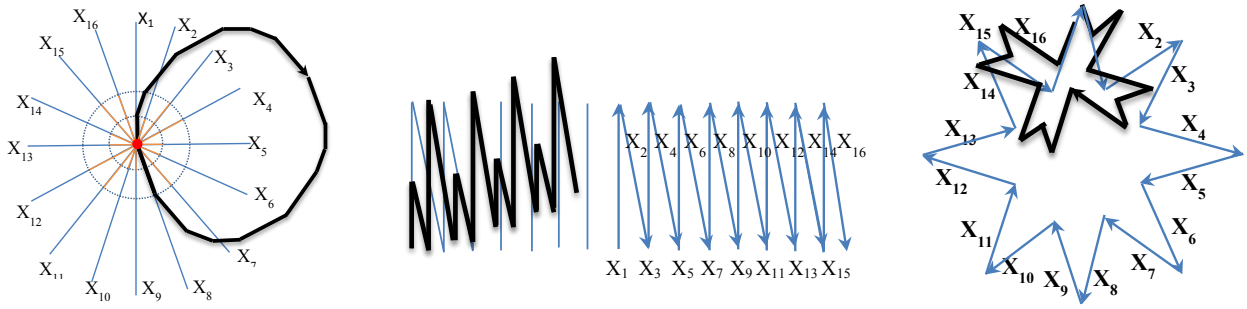
Different n -D points may have different orderings that avoid self-intersecting forms. Therefore for meaningful comparison we compare only 2-D forms for n-D points that have the same ordering.

We analyzed several real datasets. Some datasets have the same orderings practically for all n-D points. Some datasets have large subsets with the same orderings especially when strict ordering is relaxed by allowing violation of the ordering within some threshold.

For the datasets that are extremely diverse in terms of orderings we propose use original non-radial CPC graphs [7] without this reordering, or use Radial CPC Stars discussed above (see Figure 2) that have no self-crossings.

All the proposed versions of Collocated Paired Coordinates provide new visualizations, which show an n-D point in a way which could be especially beneficial for the naturally paired data, and could be easily interpretable.

All CPC representations have two times fewer break points on the contour of the figure than the traditional stars, which is significantly decreasing the complexity of forms.



(a) Black graph -lossless representation in KGRC. Red point -- lossy representation in KPRC

*(b) Left- lossless representation of point **x** in "Comb" Zigzag Coordinates, .Right - Zigzag-based Coordinates,*

*(c) Lossless representation of point **y** in Circular Zigzag Coordinates.*

*Figure 4. Examples of n-D data displays in alternative distance-based representations: (a) for 16-D point **x**=(1,1,2,2,1,1,2,2,1,1,2,2,1,1,2,2), (b) (c) for 16-D point **y** that is close to point **x***

.

It effectively doubles the representable data dimensions up to at least 200 dimensions for CPC stars [3] as Figure 2c illustrates. The expansion of the proposed approach for dimensions n up to 1000 is as follows: grouping coordinates x_i of **x** by 100-150 and representing them by separate or collocated colored stars, and/or mapping some x_i into colors. Lossy reduction of n can be applied after visual analysis of these lossless displays, which can reveal the least informative attributes. Another reduction is based on a priori domain knowledge.

To avoid occlusion each star can be displayed in its own coordinate system located in a separate cell. While this solves the occlusion issue, it creates another issue. This issue is switching gaze from one star to another one. It takes time, requires memorizing the first star before looking at another one, which complicates the comparison of stars

One of the solutions for this issue is considering one star as a base, and overlaying other stars with it one after another. The color of the overlaid star will differ from the color of the base star. The sections of two stars that are practically identical can be blinked or shown in a third color. The subject can use a mouse click to indicate that two stars are similar and potentially from the same class. An experimental study is needed to see whether the time to discover a pattern will shrink.

Related work. The radial arrangement of n coordinates with a common origin is used in several 2-D representations of n-D data. The first one has multiple names (e.g., star glyphs) [18], the name Radar plot is used in Microsoft Excel. It is based on the same idea a parallel coordinates with points on coordinate axes connected by a polyline. In addition points x_n and x_1 are connected to make a closed contour ("star"). In this paper we call this lossless representation of n-Da data as the **Traditional Star Coordinates (TSC)**.

Other visual representations of n-D data that are also called by their authors Star Coordinates (SC) [15, 20] are reviewed below. In addition we have Cartesian and Polar versions of the lossless CPC Star coordinates that we introduced and defined above.

To distinguish all of them we will use the following names and abbreviations: Graph Radial Coordinates (GRC) for lossless SC from [20] and Point Radial Coordinates (PRC) for lossy SC from [15,20]. In fact lossy SC in [15] are the same as lossy SC proposed in [20].

Note that our CPC Star Coordinates are graph-based, i.e., also Graph Radial Coordinates, therefore we denote Graph Radial Coordinates from [20] as **KGRC** to distinguish them from CPC Star Coordinates. Similarly several point-based Radial Coordinates exist, therefore Point Radial Coordinates from [20] are denoted as K**PRC.**

In KGRC a 2-D graph of an n-D point $\mathbf{x}=(x_1,x_2,\ldots,x_n)$ is created by connecting the consecutive edges. The first edge has length x_1 and is located on the first coordinate X_1, starting from origin and ending on point x_1 on X_1. The second edge starts at the end of the first edge and is going parallel to the second coordinate X_2. It has length x_2. Similarly the edge j is going parallel coordinate X_j starting at the end of edge j-1. In general this graph is not a closed contour. To make it a closed contour we can add an edge connecting the last node with the origin node. See an example in Figure 4a. In lossy KPRC only the last node is used to represent n-D point $\mathbf{x}=(x_1,x_2,\ldots,x_n)$. In the example in Figure 4a it is a red dot. In this example coincidently the graph not only starts in the origin but also ends in the origin.

In general in KGRC the n-D point **x** is represented by lengths consecutive edges with directions of edges reproducing directions of respective coordinate. This description of it gives us an opportunity to design the **generalized KPRC** using our General Line Coordinates. Respectively with GLC coordinates can have any directions and locations. For instance, coordinates can be located in different *zigzag arrangements* one after another as shown in Figure 4bc. They produce different lossless 2-D representations of n-D data that give more opportunities to discover visual pattern by different people for the same n-D data.

In the lossy point-based KPRC from [15, 20] the coordinates of the last node $L=(L_1,L_2)$ are weighed sums of $x_1,x_2,\ldots,x_n$, $L_1=w_1x_1+w_1x_2+\ldots w_nx_n$, and $L_1=u_1x_1+u_1x_2+\ldots u_nx_n$. These sums are many-to-one mapping of n-D points to 2-D points and therefore are lossy representations. In particular any scaling of 16-D point from Figure 4a will produce the same 2-D point, i.e., the origin. In [15] it is defined as a sum of vectors drawn along of each coordinates with respective lengths of x_j. In this example, each vector has the opposite vector. As a result the sum of these vectors is the zero vector which leads to the origin. While this visualization is lossy and respectively incomplete, it has important positive properties such as low occlusion and representing some integral information about the all attributes of an n-D point.

While the words "stars" and "radial" are present in the alternative visualization techniques such as the Radial Visualization (RadViz) [16, 17], and the already discussed Star Coordinates [15], they are radically different approaches. Both of them are lossy, representing each n-D data point by a **single 2-D point.** They lose **a** large part of the information of this n-D point, because of the many-to-one mapping of the attributes of an n-D point. In contrast our Stars are lossless, because they represent complete information of each n-D point by a graph. RadViz and lossy Star Coordinates from [15,20] can show clouds of points and roughly some attributes of this clouds (sizes, elongation, and

localization in data space). Moreover, these clouds will be meaningful mostly for compact classes and if they do not occlude each other. Thus the point-based approach has significant limitations being oriented mostly to visual classification and clustering tasks with *relatively simple compact data classes*. Due to absence of internal structure of a 2-D point in contrast with a 2-D graph the abilities to extract deep structural information from point clouds is practically impossible. In essence it prevents deep visual analytics from the very beginning of visual data representation.

In our approach, we attempt to maximally use the unique capabilities of the human vision system to extract the deep structural information. It allows detecting essentially *nonlinear, non-compact structures* in the n-D data space, and to understand their properties much better than lossy displays, such as RadViz, lossy Star Coordinates, projections, and others, that simplify the user's visual task by removing deep structural information. Thus, we have *two opposite approaches*: *lossy and lossless for visual knowledge discovery*.

3 Algorithm and Experimental Results

3.1 Closed contour lossless visual representation

Figure 5 and 6 show, respectively, traditional and CPC stars for 5 classes: healthy (black), and 4 diseases (colored) [data from UC Irvine Machine Learning Repository, Heart DB Hungary]. This dataset includes 14 attributes selected by a cardiologist from 47 registered attributes.

In Figure 5 some diseases have visible differences from healthy patients such as more fragments of rectangles and different symmetry axes. This first visual clue is a guide for the next analytical steps that check them on the whole dataset to provide confidence in the discovered pattern.

These figures also show that CPC stars are more compact than traditional stars. It is visible in Figure 6 where all not black cases are more "horizontal" and black cases are mostly vertical with Northwest orientation.

The difference between classes is less evident in traditional stars. CPC stars allow getting better patterns and finding them faster. Figure 7a shows a traditional star for an n-D point **p** from the black class. The traditional stars from Figure 5 that are close to **p** were found visually and are presented in Figures 7b-e from each colored class. Similarly Figure 8a shows the CPC star for the same point **p** and Figures 6b-e present respective close CPC stars from each colored class. The overlay of stars a and b from Figure 7 is captured in Figure 9 showing real closeness of these closed forms.

3.2 Feature Extraction Algorithm

Below we describe the algorithm for extraction of discrimination features using the data explained above. We start from an arbitrarily n-D point p_1 from class C_1 (e.g., black class), and find the n-D point p_2 in class C_2 (e.g., red class), which is most similar to p_1 using a lossless closed contour representation of points in 2-D. See Figure 7, where black Figure 7a represents p_1 and red Figure 7b represents p_2 for Figure 5. Then we search for the n-D points in both classes, which are most similar to p_1 and p_2. These points have been marked by stars in Figures 5 and 6. Next we evaluate distribution of these points between C_1 and C_2 classes. In Figure 3 it is 13:4 (76.5% in C_1) and in Figure 6 it is 14:3 (82.4% in C_1).

Respectively the **algorithm** steps are:

1. Randomly select an arbitrarily n-D point p_1 from class C_1
2. Find all the n-D points in both classes that most similar to p_1 and p_2.
3. Evaluate distribution of these points between C_1 and C_2 classes.
4. Remove these points from the dataset.
5. Select another point in C_1 from the remaining C_1 points and repeat the visual search for this point as we did for p_1 and p_2. This process continues until all points from C_1 and C_2 are processed.
6. Enhancing visual patterns to improve separation. In the case of points p_1 and p_2 this is finding features that differentiate them.
7. Formalizing found visual patterns to be able computing class of new objects without a human expert who needs to analyze visual patterns.

Below we discuss step 6 in more details. Consider p_1 and p_2 as shown in Figure 7ab. The upper line in the black case p_1 is going *down*, but in the red case p_2, it is *horizontal*. Next we test this visually discovered property on its ability to separate better those 17 cases. We have two cases with horizontal line in each class C_1 and C_2 among 17 cases that are similar to p_1 and p_2. Thus this feature is not a good feature to improve the separation of these 17 cases. Another visual feature must be found.

Having CPC star representation we can try to find separation features in CPC stars. We can see in Figure 8b (red case) a *very short line* on the right, which is almost vertical. This line is present in all 3 red cases and is not present in any of the 14 black cases that we try to separate. Thus, this is a perfect feature to improve the separation of 17 very visually close cases with 100% accuracy.

Next we turn to Step 7 to find an analytical form of that visual feature. Denote the start and end points of that line as w_s and w_e. Their distance $d(w_s,w_e)$ serves as a discrimination feature

$$\text{If } d(w_s,w_e) > d \text{ then class } C_1 \text{ else class } C_2, \quad (1)$$

where d is a distance threshold computed from Figure 8b. Let's for simplicity of notation assume that we started the graph in Figure 8 from this point w_s. In this case our start and end points are

$$w_{s1}=f(x_1,x_2),\ w_{s2}=g(x_1,x_2), \quad w_{e1}=f(x_3,x_4),\ w_{e2}=g(x_3,x_4) \quad (3)$$

where x_1-x_4 are first four original n-D coordinates of an n-D point that we consider. Here f and g are functions that are used to map x_1-x_4 to CPC star coordinates as we presented in section 2. Thus formula (1) will be rewritten as with use of (3):

$$((w_{s1}- w_{e1})^2 + (w_{s2}- w_{e2})^2)^{1/2} > d \text{ then class } C_1 \text{ else class } C_2 \quad (4)$$

$$((f(x_1,x_2)-f(x_3,x_4))^2 + (g(x_1,x_2)-g(x_3,x_4))^2)^{1/2} > d$$
$$\text{then Class } C_1 \text{ else Class } C_2 \quad (5)$$

Discovering (5) demonstrates the power of visual analytics, which combines visual and computational methods in Visual Data Mining. Discovering (5) purely analytically without a visual clue would be extremely difficult. We would need to guess somehow a class of models that includes (5). What could be the base for such a guess? It is hard to expect prior knowledge of this kind. In these particular data, we definitely did not have such prior knowledge. Next event if the guessed class will include (5) it is not necessary that (5) will be a winning model on the given training data.

How general is this algorithm? Why is it not an ad hoc one? Steps 1 and 4 are quite general for any training dataset with the classes of n-D points identified. The success in Steps 2, 3, 5 and 6 depends on 2-D representation of n-Data, perceptual abilities of the viewer, allotted time and amount of data. The step 7 is also quite general and its success depends on success in previous steps and

on mathematical skills of the analyst. So far experiments with CPC Stars show that all these steps are doable successfully for real data providing a consistent framework for visual analytics in Data Mining. Further research and experiments are needed to specify steps 1-7 more and data types where this algorithm will be efficient. It includes training data miners in visual features search.

3.3 Comparison with Parallel Coordinates

Figure 10 shows the same data in Parallel Coordinates as in Figure 6. We do not see a separation pattern between the classes in it, but a separation pattern is visible in CPC stars in Figure 6.

The difference between Traditional Stares, CPC stars, and Parallel Coordinates is even more visible in Figures 12 and 13 in the higher dimension (n=170). Figure 12 shows traditional 170-D stars in the first two rows: musk chemicals (first row, black), and non-musk chemicals (second row, red). Respectively the third and fourth rows in Figure 12 show CPC 170-D stars from the same dataset: musk chemicals (third row, black) and non-musk chemicals (forth row, red). A specific pattern on the right of each star is visible on rows 2 and 4, which represent non-musk chemicals. Multiple other distinct features can be extracted from Figure 2, which can assist in separating the two classes. In contrast in Figure 13, 4 points from the black class, and 5 points from the red class is very difficult to identify and separate from other features.

4. Prospects for higher data dimensions

The above advantages of CPC stars vs. traditional stars and parallel coordinates are even more essential for data of higher dimensions. We presented these three representations (Figures 12,13) for musk learning dataset from the UCI machine learning repository. It is an example of very practical design models of drugs and other chemicals without expensive experimental tests, such as clinical trials of the targeted properties. In these data each instance is described by their 170 physical, chemical, structural, etc. properties and its target attribute (musk class or non-musk class).

Although CPC stars show the same information in each cell as the traditional stars, they are better for visual analysis because they have: (1) less density of form features, (2) bigger sizes, (3) better separability, etc. In contract, Parallel Coordinates are unacceptable for such large data dimensions, while the stars above allow comparing data with hundreds of attributes. Open polylines in Parallel Coordinates of the same n-D data points as shown in Figure 13 are practically indistinguishable. These advantages of closed contours are consistent with Gestalt Laws making the need in new extensive user studies not necessary because we already conducted such studies on some data in [1,3] and abilities to rely on extensive previous experiments elsewhere that verified Gestalt Laws viewed as most universal information about form perception for display choice independently on specific data properties.

5. Conclusion

The new Visual Data Mining technique based on Closed Contour coordinates is proposed in this paper. The experimental results for the visual feature extraction from the multidimensional data using this technique show that it is a promising method for the visual analytics and the visual data mining. Advantages of the proposed visual data mining technique relative to the Parallel Coordinates have been shown. This technique also can be applied in cooperation with the analytical Data Mining methods to decrease the heuristic guesses in selecting a class of Data Mining models.

6 References

[1] Grishin V., Kovalerchuk, B., "Stars advantages vs, parallel coordinates: shape perception as visualization reserve", In: SPIE Visualization and Data Analysis 2014, Proc. SPIE 9017, 90170Q, 8 p.

[2] Elder, J., Goldberg, M., "Ecological statistics of Gestalt laws for the perceptual organization of contours", Journal of Vision (2002) 2, pp. 324-353 http://journalofvision.org/2/4/5/ 324

[3] Grishin V., Kovalerchuk, B., "Multidimensional collaborative lossless visualization: experimental study", CDVE 2014, Seattle, Sept 2014. Luo (Ed.): CDVE 2014, LNCS 8683, pp. 27–35, Springer, 2014

[4] Bertini,E., Tatu, A., Keim, D., "Quality metrics in high-dimensional data visualization: An overview and systematization", IEEE Tr. on Visualization and Computer Graphics, 17 (12), pp. 2203–2212, 2011.

[5] Hoffman, P. Grinstein, G., "Survey of Visualizations for High-Dimensional Data Mining", In: Information Visualization in Data Mining and Knowledge Discovery, Eds. U. Fayyad, A. Wierse, G. Grinstein, pp. 44-82, Academic Press, 2002.

[6] Inselberg, A., Parallel Coordinates: Visual Multidimensional Geometry and its Applications. Springer, 2009.

[7] Kovalerchuk, B., "Visualization of multidimensional data with collocated paired coordinates and general line coordinates", In: SPIE Visualization and Data Analysis 2014, Proc. SPIE 9017, Paper 90170I, doi: 10.1117/12.2042427, 15 p. http://www.cwu.edu/%7Eborisk/pub/BK3_2014_Visual.pdf

[8] Kovalerchuk B., Grishin V., "Collaborative lossless visualization of n-D data by collocated paired coordinates", in: CDVE 2014, Seattle, UW, Sept 2014, Y. Luo(Ed.):LNCS 8683, pp. 19–26, Springer, 2014.

[9] Kovalerchuk, B., Smigaj A., "Computing with words beyond quantitative words: incongruity modeling", in: Proc. of NAFIPS, 08-17-19, 2015, Redmond, WA, IEEE, 2015, pp. 226-233.

[10] Lots of Lines software (standalone, web-based, and Excel-plugin, http://www.cwu.edu/~Imaglab/programs/CS480_2015/WebBuild/Assets/Documentation/index.html.

[11] Simov S., Bohlen M., Mazeika A. (Eds), Visual Data Mining, Springer, 2008

[12] Tergan. S. Keller, T. (eds) Knowledge and Information Visualization, Springer, 2005

[13] Ward, M. Grinstein, G., Keim, D. Interactive Data Visualization: foundations, techniques, and applications, A K Peters, 2010.

[14] Wong, P. Bergeron, R., "30 Years of Multidimensional Multivariate Visualization", in G. M. Nielson, H. Hagan, and H. Muller (Eds), Scientific Visualization - Overviews, Methodologies and Techniques, pages 3-33, IEEE Computer Society Press, 1997.

[15] M. Rubio-Sánchez, L. Raya, F. Díaz, A. Sanchez, "A comparative study between RadViz and Star Coordinates, Visualization and Computer Graphics", IEEE Transactions on (Vol.22, Issue: 1), pp. 619 – 628, 2015. DOI: 10.1109/TVCG.2015.2467324

[16] Sharko,J., Grinstein, G., Marx K., "Vectorized Radviz and Its Application to Multiple Cluster Datasets", IEEE Trans. Vis. Comput. Graph. 14(6): pp.1444-1427, 2008.

[17] Daniels, K. Grinstein, G., Russell, A., Glidden, M., "Properties of normalized radial visualizations", Information Visualization, 11(4):pp.273– 300, 2012.

[18] Ward, M., "Multivariate Data Glyphs: Principles and Practice," Handbook of data visualization, Springer, pp. 179–198, 2008.

[19] Wertheimer, M. "Gestalt theory", Social Research, 11, 78-99,l 1944.

[20] Kandogan E., "Visualizing multi-dimensional clusters, trends, and outliers using star coordinates", in Proc. 7th ACM SIGKDD conf. on Knowledge discovery and data mining, pp.107-116, 2001.

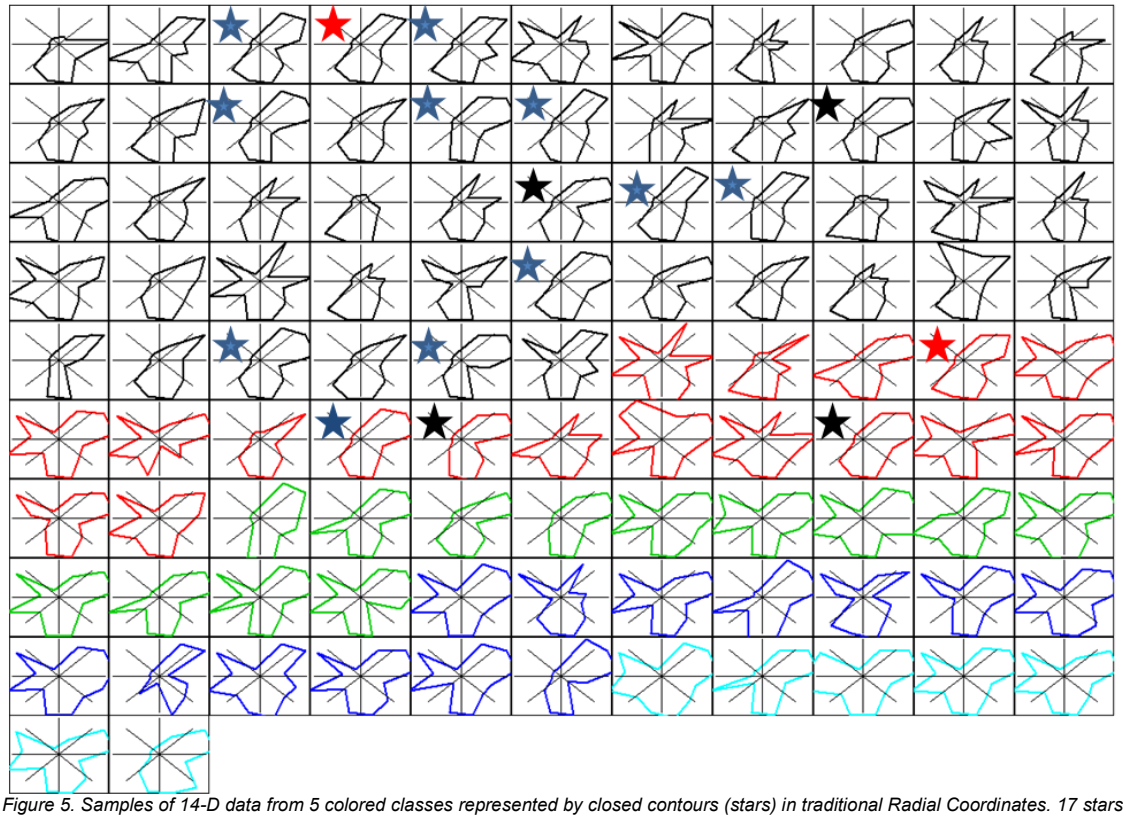


Figure 5. Samples of 14-D data from 5 colored classes represented by closed contours (stars) in traditional Radial Coordinates. 17 stars mark similar forms found in the black and red classes (13 in black class and 4 in red class). The found pattern is dominant in the black class (76.5% accuracy). Red stars mark most similar forms found in these opposite classes.

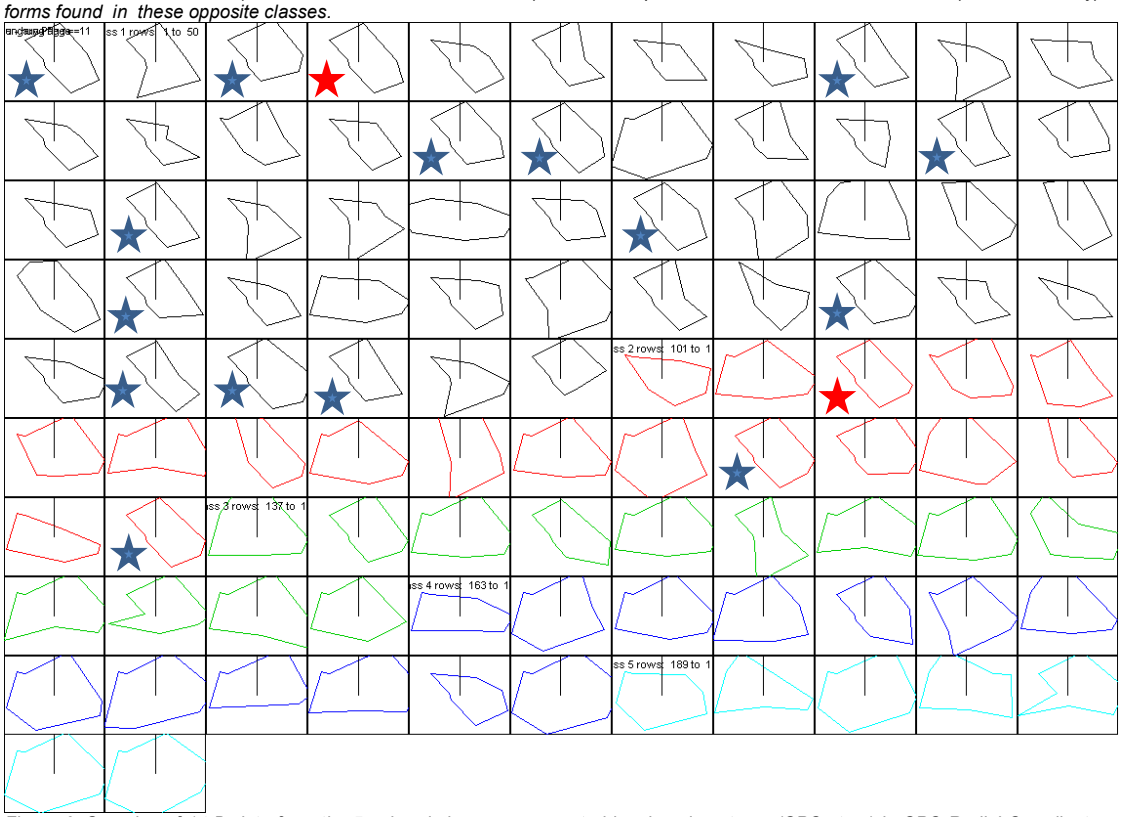


Figure 6. Samples of 14-D data from the 5 colored classes represented by closed contours (CPC stars) in CPC Radial Coordinates. 17 stars mark similar forms found in black and red classes (14 in black class and 3 in red class). The found pattern is dominant in the black class (82.4% accuracy). Red stars mark the most similar forms found in these opposite classes.

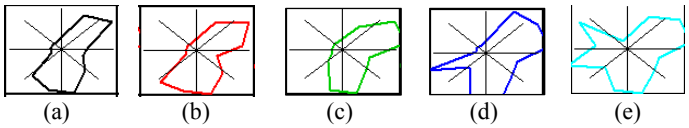


Figure 7. Closest traditional stars from 5 classes from Figure 5

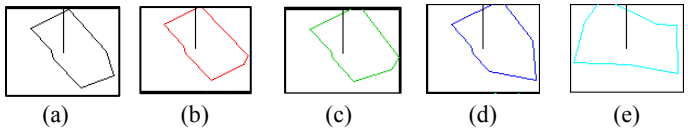


Figure 8. Closest CPC stars from 5 classes *from Figure 6*

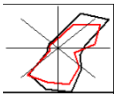


Figure 9. Overlay of stars a and b from Figure 5.

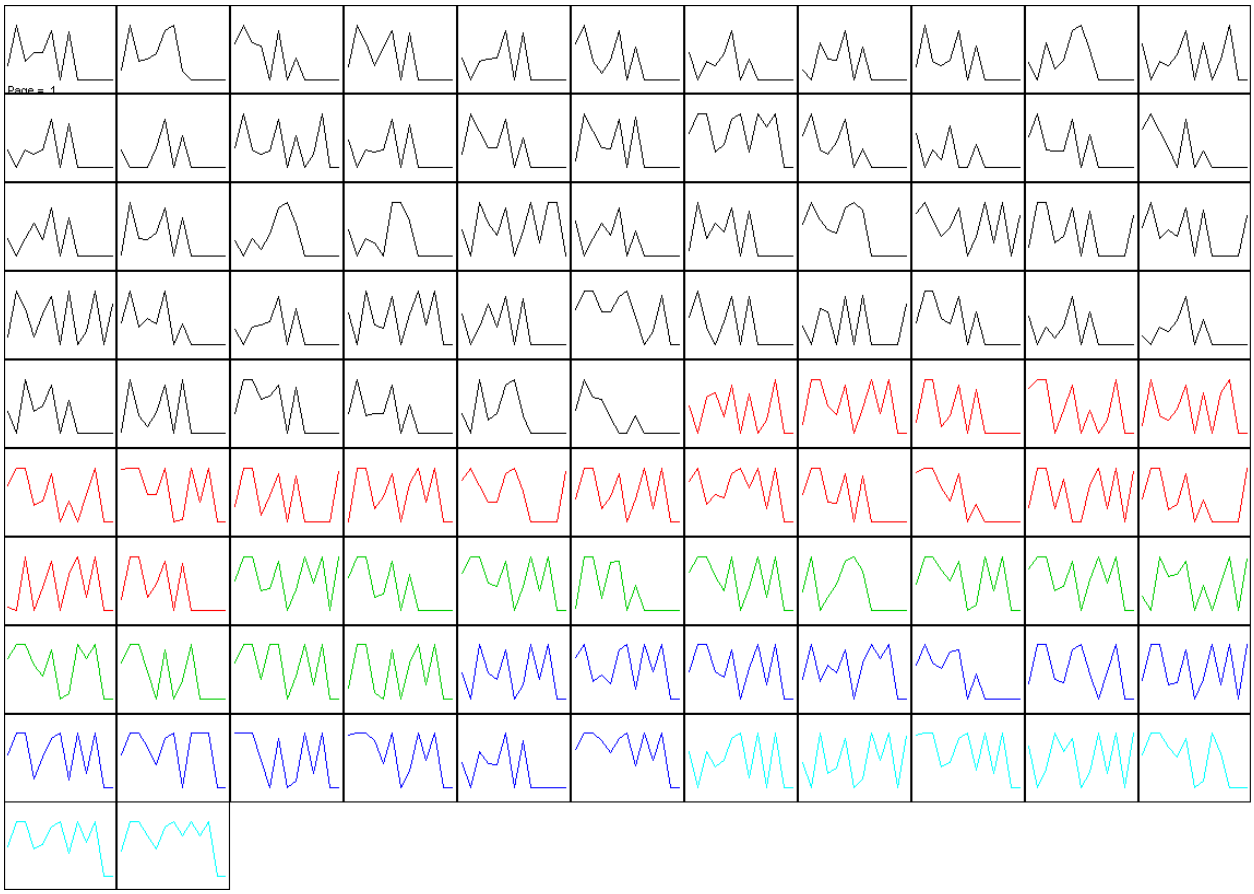


Figure 10. Samples of 14-D data from 5 colored classes in Parallel Coordinates.

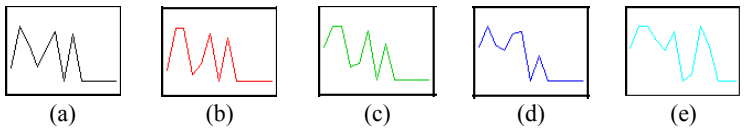


Figure 11. Similar Parallel Coordinates curves from Figure 10.

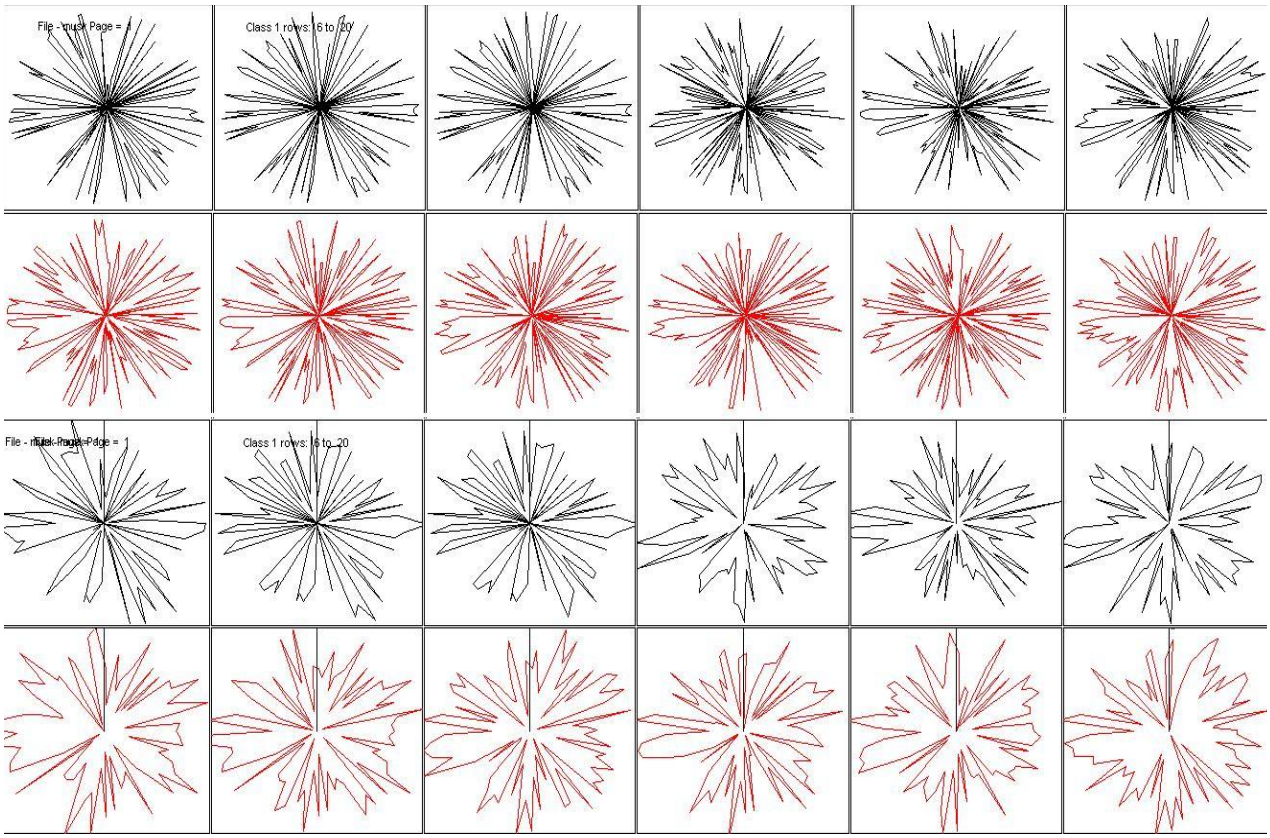


Figure 12. Traditional 170-D stars: class "musk" (first row, black) and class "non-musk chemicals" (second row, red). CPC 170-D stars from the same dataset: class "musk" (third row, black) and class "non-musk chemicals" (forth row, red).

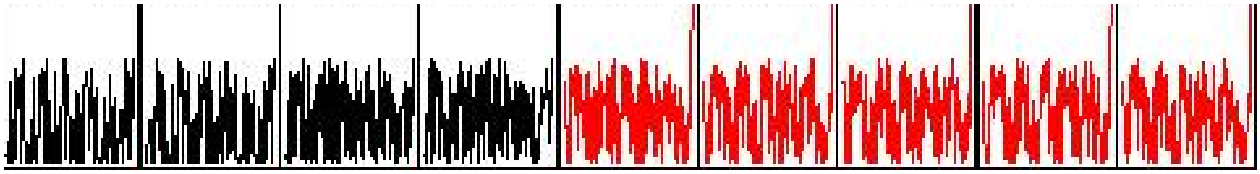


Figure 13. Nine 170-dimensional points of two classes in Parallel Coordinates.