Lossless Interpretable Glyphs for Visual Knowledge Discovery in High-Dimensional Data

Nicholas Cutlip
Dept. of Computer Science
Central Washington University
Ellensburg, WA 98926, USA
Nicholas.Cutlip @cwu.edu

Boris Kovalerchuk
Dept. of Computer Science
Central Washington University
Ellensburg, WA 98926, USA
BorisK@cwu.edu

***Abstract*— The expansion of the use of a machine learning (ML) technology in multiple domains heavily depends on acceptance of these models by end users and their abilities to understand the patterns discovered by machine learning algorithms. The simplicity and understandability of those patterns by the domain expert is critical for success of such ML explorations. This paper proposes a method for designing lossless compact and interpretable machine learning visual patterns to be easily captured and remembered by the domain experts due to their compactness and simplicity. This glyph approach is illustrated with a "bird" glyph. This glyph is a combination of Stick Figures and a type of General Line Coordinates known as Shifted Paired Coordinates. The efficiency of the approach is demonstrated on several benchmark machine learning data sets.**

Keywords— Interpretable Machine learning, classification, glyph, lossless visualization, high dimensional data, general line coordinates. Shifter Paired Coordinates, Stick figures.

I. Introduction

A. Motivation

The expansion of the use of machine learning (ML) technology in multiple domains heavily depends on acceptance of these models by end users and their abilities to understand the pattern discovered by ML algorithms. The simplicity and understandability of those patterns by the domain expert is crucial for success of such ML explorations, where the data visualization plays the critical role. For the data visualization, a key goal is making visualization *lossless* and reversible, so that a user may create the visualization from raw data and then recreate the raw data from that visualization (no information is lost). The next goal is making of observations and discovering of patterns using the visualized data as *simple*, as opposed to working with the raw data.

A technique to meet these goals is proposed in this paper. It allows discovering simple and compact patterns, which can be quickly recognized and recalled by subject experts. It integrates Stick Figures (SF) glyphs and Shifted Paired Coordinates (SPC), a subset of General Line Coordinates (GLC). We use benchmark ML data to show the effectiveness of the method.

B. State-of-the-art

Existing methods accomplish the described goals with varying success. In general, it is difficult to accomplish all of them in a balanced manner. Furthermore, the higher the data dimension is, the more difficult it becomes to visualize those data effectively. Another issue is the difficulty with which users glean information from a visualization, e.g., a doctor examining a visualization to diagnose cancer. There is a need for **simple yet informative data visualization** methods, accessible to users.

Glyph-based visualization methods [1,2] have been applied in machine learning [3], where an icon, or symbol are used for each n-D point. The classic example of data visualization using glyphs is that of Chernoff faces, in which the n-D point $\mathbf{x}=(x_1,x_2,\ldots,x_n)$ is represented as a cartoon image of a face (glyph) and each attribute x_i is encoded to some facial feature such as eyebrow angle or mouth position [4]. In Stick Figures [5], data attributes are encoded to the angles between segments of the stick figure and lengths of the segments. Fig. 1 shows a few samples of the glyphs from [1].

These common glyphs do one-to-one mapping. They encode one attribute value x_i to one visual element like a line or a face part. The glyph method proposed in this paper breaks this one-to-one mapping allowing mapping two or more attributes to a single visual element. The advantage is that it simplifies glyphs. It is based on a recently developed approach which has expanded the breadth of higher dimensional data visualization options. It is a set of General Line Coordinates (GLC) [6-8], where the commonly used GLC methods are Radial Coordinates, Shifted Paired Coordinates (SPC), and GLC-Linear coordinates.

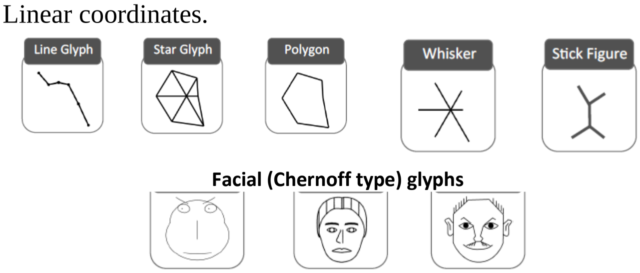


Fig. 1. Examples of glyph types [1].

C. Approach outline

In the proposed method each data point is represented as a "**bird**" glyph, consisting of encoded angles and lengths. Shifted Paired Coordinate (SPC) axes are then arrayed around the vertices of the bird. The attributes of an n-D data point are used to encode all these features of the glyph. Sets of glyphs representing clusters, hyperblocks, classes, or other subsets of the data set may then be visualized in a variety of manners. The **juxtaposition** of glyphs allows for simple and efficient comparison between data points in the dataset. These glyphs are simple enough to contain patterns recognizable by non-ML experts. We analyze glyphs produced using n-D rectangles (hyperblocks) with the UCI ML benchmark data to determine their accuracy of as classification tools including in high-risk classification scenarios.

II. METHODS

A. *Stick Figures*

The glyphs that we explore were first proposed in [6] as an option for future studies for expanding GLC to glyphs for encoding more attributes in GLC. We first describe the two visualization methods **Stick Figures** (SF) [5] and **Shifted Paired Coordinates** (SPC) before describing their hybridized version, SPC-SF.

Fig. 2 demonstrates encoding a 4-D point $x = \{x_1, x_2, x_3, x_4\}$ into a two-segment SF, or "bird". Each line segment of the bird represents two data attributes, encoded as an angle and as a length. Thick black lines represent the SF. Thin blue lines demonstrate how the SF angles are encoded. The method is expandable to higher dimensional data by adding more segments.

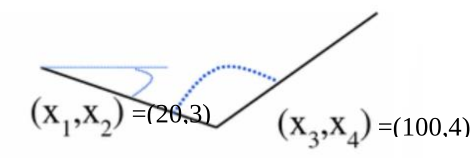


Fig. 2. 4-D data point visualized as a two segment Stick Figure (SF).

B. *Shifted Paired Coordinates*

Shifted Paired Coordinates (SPC) make use of paired orthogonal coordinates to losslessly visualize n-D data. Fig. 3 shows a 6-D point $\boldsymbol{x} = (3{,}2{,}3.5{,}3.2{,}5{,}4)$ represented in SPC as three pairs of 2-D points (3,2), (3.5,3.2) and (5,4) in the respective pairs of coordinates.

The 6-D point $\boldsymbol{x}$ is visualized as a two-segment line strip (graph) with only three vertices to show 6 values of the 6-D point. This reduction of the data representation's dimensionality by $n/2$ is a common feature of paired orthogonal coordinates in GLC [6].

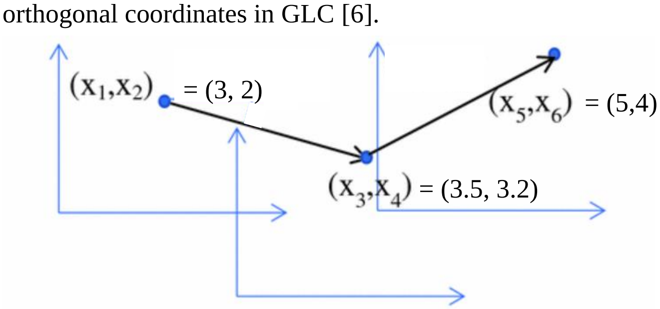


Fig. 3. 6-D data point visualized in Shifted Paired Coordinates.

The shift of one pair of coordinates relative to another one can be controlled and made dynamic. In [6] it was used to make the visualization simpler. Here we use it to accommodate more features in the form of glyph features.

C. *Hybridized SPC-SF "birds"*

In the shifted paired coordinates, we define three 2-D points for any 6-D point as shown in Fig 3. To embed a SF glyph to SPC we locate those three pairs of coordinates by encoding the lengths of two edges and the values of two angle in the bird. Through this, the bird conveys additional 4 attributes.

In general, in SPC without glyphs we encode n-D data with $n/2$ 2-D points. Embedding glyphs allows us to convey four more attributes, as in Fig. 4. This illustrates the advantage of embedding glyphs to GLC because it allows increasing the dimensionality of the data visualized without adding two more shifted paired coordinates. This also opens an opportunity to apply glyphs to other general line coordinates and expand these methods' dimensionality.

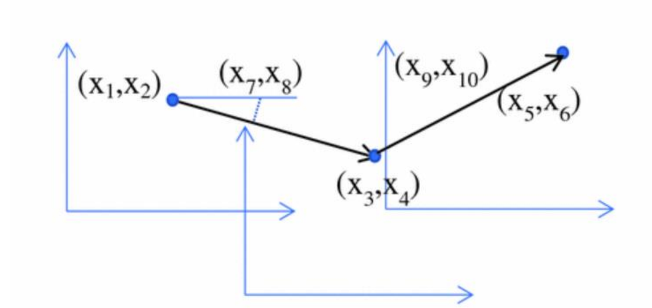


Fig. 4. Hybridized SPC-SF visualization method. The Stick Figure itself, shown as a thick black line, encodes 4 attributes of the data: x_7, x_8, x_9, x_{10}. The remaining 6 data attributes, $x_1 - x_6$, are encoded to the x and y shift values of the SPC axes.

In Fig. 4, attribute pairs (X_1, X_2), (X_3, X_4) and (X_5, X_6) are the same as in Fig. 3, presented as 3 nodes. The remaining four attribute encode the glyph, which we already call the "**bird**". For the left wing of the bird, X_7 will be encoded to the initial angle and X_8 will be encoded to the length of the wing. For the right wing, X_9 will be encoded to the initial angle, and X_{10} will be encoded to the length of the wing.

The duplication of the attributes required by SPCs or the bird is also a way to **emphasize** attributes. For example, for 9-D data, SPC requires 10 attributes (5 pairs) where attributes X_9 and X_{10} in the visualization may be the same, with attribute X_9 duplicated and encoded twice. Alternatively,, the length and angle of the right wing of the bird could both represent this attribute. This implies that the repeated attribute will be emphasized. One way of selecting the most significant attributes to emphasize is by exploring which attributes in a dataset have the highest variability and putting them to the most distinct parts of the glyph.

As was suggested above, we will refer to this hybridized combination of SF and SPC that encodes 10 attributes as a "bird", due to its similarity to the 'V' shape of a bird in flight.

The combination of SPC-SF for the other dimensions can be memorized using other analogies.

The hybrid glyphs were formed through a combination of Stick Figure (SF) and Shifted Paired Coordinate (SPC) methods. The visualization method is readily extensible to an **arbitrary number of dimensions**. This can be accomplished through adding more segments to the SF, along with sets of shifted paired coordinates. Each addition of a SF segment and SPC coordinate pair will increase the dimensionality of the visualization by 4. However, it will eliminate the “bird” metaphor and will require another metaphor.

The angles can be encoded in two ways: *independently* of prior angles (“*statically*”) or *dependently* (“*dynamically*”). It can be done by using different ranges of the angles for attributes, say, full 360, 180, or other degree ranges. Statically assigned angles are measured from the *horizontal* line. Dynamically assigned angles are measured as an angle relative to the *previous segment*. These produce two different classes of glyphs as shown in Fig. 5, with dynamic angles in the top row of glyphs and static angles in the bottom row.

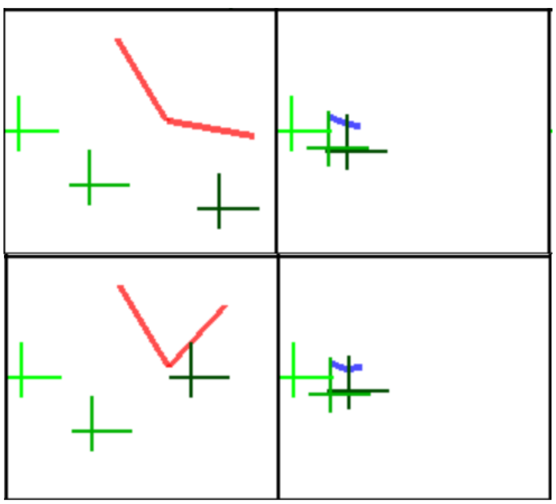


Fig. 5. Bird angles encoded dynamically (top row) and statically (bottom row).

Mapping with a greater range may result in further visual distinction between glyphs of different classes. However, too great an angle range may increase the difficulty of identifying visual similarities between points.

As is visible in Fig. 5, angle encoding method (static or dynamic) has a greater effect on the appearance of larger glyphs than smaller glyphs. Regardless of angle size, the bird’s angles are visually emphasized when wing length is longer.

In the case studies that follow, we compute the angle of the second bird segment dynamically. The first will always be static by default due to the lack of a preceding SF segment. We choose to encode the second angle statically, as static angles create greater visual distinction between bird “wings”. This is demonstrated by the bird glyphs in Fig. 5. We normalize the angle values to a range of [0, 90] degrees. This range allows for variation in shape between birds of different classes while maintaining a standard glyph shape for all data points. These parameters may be interactively adjusted in the SPC-SF data visualization program.

The remaining normalized data attributes are directly encoded to the SPC values. Two attributes encode each SPC pair, resulting in six attributes total encoded. One member of each pair is used for the corresponding coordinate system’s x-shift value. The other member is used for that coordinate system’s y-shift value. The initial positions used for the shifts are the vertices of the SF. Each SPC pair is linked to a single vertex. The result is three pairs of SPC axes arrayed around the vertices SF, as is illustrated in Figs. 4 and 5.

It is also possible to visualize data of *less than 10 dimensions* using this method. This is done through **duplication** of dimensions as well as **simplification** of the visualization. The bird metaphor can be maintained as the core feature of the visualization, however, the SPC n as **crosses**) may be removed. When there are less than 10 attributes, we may duplicate attributes to get 10 attributes. Although any attribute may be duplicated, the most significant should be considered for duplication first. Smaller dimensions of data also present more opportunities for simplification of the glyph. For example, if we have 5 attributes, we may duplicate each attribute just after it appears. This results in each pair of SPC coordinates having the same *x* and *y* values. It follows that each coordinate pair will be shifted symmetrically from the vertices of the bird. Such symmetricity will ease the user’s understanding of the pattern. Analysis will not require comparison of the vertical and horizontal shifts relative to the origin of the SPC pairs.

Even with more than five attributes, the **glyph can be simplified** through removal of the crosses. With six attributes, we can use the first two attributes to place a shifted pair of coordinates as the location of the center of the bird, its head. This pair locates the bird, and the remaining four attributes will represent the lengths and angles of the wings. If we have eight attributes, then the same method can be used with the addition of one SPC cross shifted from one vertex of the bird. If we have only four attributes, then we can put all birds centered in each cell. Two attributes can be encoded to the angle of each wing and two attributes can be encoded to the length of each wing. In this case we will have a simple bird without any crosses.

If we have only three attributes, then we can encode X_1 and X_2 to the lengths of the wings. The third attribute we can encode to the angle between the wings. In the situation with only two attributes, we can again encode them to the lengths of the wings. In this scenario, for all birds, the angles will be the same. For 2-3 attributes the standard scatter plots can visualize all data together and the birds aren’t required. However, if birds will have distinct shapes, they can be remembered by the users serving as memory tips.

D. Hypercube interpretable grouping algorithm

To identify general patterns for the shapes of glyphs within a given dataset we construct data hypercubes, which use an **interpretable similarity measure** defined below. Traditional grouping algorithms typically use distances like Euclidian distance. The problem with this traditional approach is that for heterogeneous data like blood pressure and temperature, which are common in machine learning, the distances with them *together* have no empirical meaning [9]. In contrast many arithmetic operations *within each* heterogeneous attribute have clear meaning, like difference between temperature: 75º and 80º is less than 10º. This consideration is used to define similarity based on the concept of the hypercube.

If for a given n-D point **x**=(x,x_2,…,x_n) another n-D **y**=(y_1,y_2,…,y_n) differs from **x** in each coordinate not greater than the allowed threshold value, T, $|x_i - y_i| \leq T$, then **y** belongs to the **Hypercube** (**HC**) centered in **x**. If threshold T is small, then **x** and **y** are viewed as similar n-D points.

We introduce a **difference measure** $D(\boldsymbol{x},\boldsymbol{y})$:

$D(\boldsymbol{x},\boldsymbol{y})$=0 if for all $|x_i - y_i| \leq T$, else $D(\mathbf{x},\mathbf{y})=\max_{i=1:n} |x_i - y_i|$

Thus, the difference between n-D point within a hypercube is defined as zero and outside of the hypercube it is defined as max of all coordinate distances. This difference measure is not a classical metric because it does not satisfy the triangular inequality. The example below proves this for the one-dimensional case by contradiction.

Consider 1-D points, $\mathbf{b} = 0$, $\mathbf{c} = 1$, and $\mathbf{a} = 2$. We will use threshold value $T = 1$. This gives us values $D(\mathbf{a},\mathbf{b}) = 2$, $D(\mathbf{b},\mathbf{c}) = 0$, and $D(\mathbf{a},\mathbf{c}) = 0$. Assume that the triangular inequality is satisfied for **a**, **b**, and **c**:

$$D(\mathbf{a},\mathbf{b}) \leq D(\mathbf{a},\mathbf{c}) + D(\mathbf{b},\mathbf{c})$$

This lead us to the contradiction $2 \leq 0$.

Hypercube quality can be measured in different ways. For classification tasks, *purity* is most important. Other important quality characteristics are large *number of cases*, *stability* and *minimal dependence* on the seed point of an individual HC and a *small number* of such HCs covering all data. Thus, we want to satisfy several **contradictory criteria.** Due to the inability to simultaneously satisfy them, we here focus on purity and number of cases in the HC. A future version of the algorithm will involve an efficient tradeoff between these four criteria.

Below we present steps of the base **Hypercube Grouping** (**HyG**) algorithm used to construct hypercubes:

1. Select an allowed *threshold* value, T, for admission to a hypercube. The smaller the threshold chosen, the more selective admission to each hypercube will be.
2. Select an n-D point **x**=(x_1,x_2,…,x_n) as a *seed* point to be the **center point** of the new hypercube.
3. Place the selected datapoint **x** into a new hypercube, HC_i.
4. Iterate over the remaining n-D points **y** not included yet to HCs. For each such n-D point **y**:
   - 4.1 Iterate over each attribute in **y** and compute $|x_i - y_i|$.
   - 4.2. If any difference $|x_i - y_i|$ is greater than the allowed threshold T, then do not include **y** to the hypercube.
   - 4.3. Otherwise, add **y** to the hypercube HC_i and remove **y** from the set of remaining n-D points.
5. Repeat steps 2, 3, and 4 for remaining n-D points. The result is a set of hypercubes.

The resulting HCs depend on the seed points and the threshold T. This threshold is the same for all attributes. In contrast, the algorithm in [9] uses **hyperblocks**, i.e., different thresholds T_i for different attributes. The algorithm in [9] controls **purity** of the hyperblocks. Each hyperblock should contain cases of a single class and can allow a controlled number of cases of other classes. Similarly, we transformed the base Hypercube Construction algorithm HyG to a purity-controlled algorithm **HyGC** making it a **classification** algorithm, where step 4.3 is modified to:

4.3 Otherwise, add **y** to the hypercube HC_i and remove **y** from the set of remaining n-D points, if adding **y** does not exceed an allowed HC purity threshold P.

III. RESULTS

A. Seed Dataset Glyphs

We first explore the usefulness of our glyphs in ***a low-risk classification scenario***. This case study uses 7-*D seed data* from the UCI Machine Learning Repository. These data contain three classes, each being a different species of seed. Our goal is for glyphs derived from the same class to display **common visual patterns**. These should be distinct from the patterns of glyphs derived from other classes.

Fig. 6 provides a comparison of the three classes of glyphs from the seed data set. We randomly selected nine datapoints from each class of seed three times, each with a different random selection of datapoints. This is to ensure class patterns are replicable. To pre-process the data, we removed two redundant attributes (seed length and seed width, which are captured by seed area), resulting in 5-dimensional data. As previously described, 10-D is well suited to the bird glyph visualization method. It will take time for the user to become familiar with the glyph metaphor.

As a result, changing the metaphor will extend the time taken to discover patterns using the glyphs. Therefore, for every data set with 10 or less attributes we use the same metaphor of the bird and three crosses. For some datasets the number of crosses can be decreased, as previously described. However, the use of the bird structure is based fundamentally on three pairs of coordinates. By *duplicating* each attribute of the seed dataset to transform each 5-D datapoint into a 10-D datapoint, we can make use of the full glyphs here.

The birds in Fig. 6 demonstrate a direct encoding of the data to glyph form, after normalization. Data attributes are randomly encoded to visual features of the glyph. As show, **dominant visual patterns** between classes are clear in the glyphs. These patterns may be represented linguistically as follows:

- Class 1 (rows 1, 4, 7): Very *small* bird. Bird is positioned to the *left* of the second and third crosses. Glyph is in the *center-left* of the cell.

- Class 2 (rows 2, 5, 8): Very *large* bird, with *acute* middle angle. Crosses are *wider* spread than in the other classes.
- Class 3 (rows 3, 6, 9): *Small* bird. Bird is positioned to the *right* of the first cross. Glyphs are in the center of the cell.

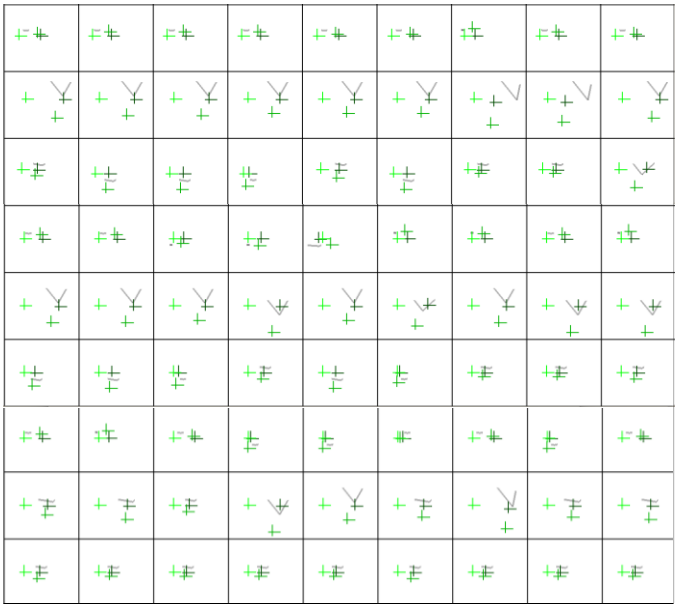


Fig. 6. Three sets of 27 glyphs from the seed dataset visualized with bird glyphs. Each set of 27 datapoints contains 9 glyphs from each class in the dataset, randomly chosen (3 classes total). Although there is some variation in pattern, for the most part class visual patterns are noticeable and distinct.

This case study illustrates how data classes can be distinctly represented in a glyph form. This will allow a domain expert to **accurately classify and gain insight** from visual analysis of glyphs. This tool can be used alongside analytical methods for increased accuracy. Specific data attributes may be purposefully encoded to specific visual features of the glyph. This allows for further customization of the visualization method by the end user. For example, with this seed dataset a user could choose to encode the seed length and width attributes to the lengths of the two wings of the bird. This would allow for accurate analysis of these attribute's values through visual analysis of bird wing length.

B. Student Success Dataset Glyths with Hypercubes

Next, we apply the bird glyphs to a knowledge domain within our field, student success in a mathematics course. The dataset used is 33-D *student success data* from the UCI Machine Learning Repository. We selected 10 numerical attributes from the 33 total dimensions for the bird visualization. Binary attributes, while not selected here, could also be encoded to the bird glyph. One strategy would be to combine multiple related binary attributes into a single numerical attribute. Another strategy would be to directly encode the binary attributes to a visual feature. This second strategy will result in only two possibilities for the shape of the glyph aspect used to encode this attribute. Such a large distinction between the shape of a bird feature may be useful, provided the binary attribute is highly representative of a datapoint's class.

Since we wish to **maintain the bird metaphor** for usability purposes, we reduce the number of attributes from 33 to 10. From this large number of attributes, we select the most informative ones. In this case, most informative means those attributes which will allow a user to formulate a qualitative rule. This rule need not be perfectly accurate but should capture the most important properties of the dataset. Thus, for this task we place ourselves as a surrogate expert. We select those attributes which we consider most correlated to student success. We then determine if the selected subset of "most informative" attributes reveal distinct visual patterns in the glyphs.

The 10 attributes selected from the dataset for this visualization are as follows: student age, parental education levels, travel time, study time, family relations, free time, alcohol consumption, health, and number of absences. Students are classified as either passing (Class P) or failing (Class F) the mathematics course based on their final grade on a scale of 0-20. Grades of 0-10 are considered Class F, while grades of 11-20 are considered Class P. Fig. 7 presents the set of representative glyphs of all HBs computed from the data using the HyGC algorithm with a threshold value $T = 5$.

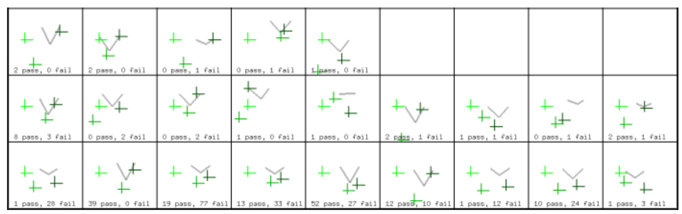


Fig. 7. All hypercubes from student success dataset constructed with HyG algorithm with a threshold of T=5. Representative glyph shown for each HC is the average point of the dominant class in each HC. Class datapoint counts for each HC are shown at the bottom of each cell.

Each cell represents an HC. The bird glyph shown represents the average case of the dominant class from that HC. The dominating class is determined from the class counts, as displayed at the bottom of each HC cell. The class totals reported for each HC demonstrate their relatively high overall purity at 76%. Dominant patterns for the Passing and Failing classes can be identified from the representative birds:

- Class F: Birds have two short wings, and a more obtuse middle angle.
- Class P: Birds have two longer wings, and a more acute middle angle.

These **clear patterns** can be used to *distinguish between instances of the two classes when they are covering many cases.* Ideally, such simple patterns will be confirmed for all HCs, while it cannot be expected that such short visual patterns will be universal for all HCs and cases. However, a user can remember the **major typical situations** with glyphs. For these data such situations can cover 60-70% of all cases, given that the analytical accuracy results of an SVM model trained with these data is in this range. Therefore, an educator could use these glyphs as an indicator of a student's future success and to provide help to those at-risk students accordingly.

C. *Cancer Dataset Glyphs with hypercubes*

This case study uses the 9-D *Wisconsin breast cancer (WBC)* data from the UCI ML Repository. We demonstrate that bird glyphs can represent this data with a **high level of visual distinction between classes**. Fig. 8 presents a representative glyph for each HC computed from the WBC dataset. The algorithm produced 61 HCs for 683 datapoints. *Ideal glyphs* are crafted for each class, by averaging values of each attribute from the datapoints of a given class. *Representative glyphs* are made for each HC using the same procedure on all datapoints contained within a given HC. Thus, all displayed glyphs are created by averaging the attributes of various subsets of the dataset. In this way they represent the overall class patterns and the dominant visual patterns within each HC.

HCs were computed using the algorithm with a threshold *T*=4. The experiments had shown that *T*=4 balanced well (1) the variance of the cases within the HC and (2) the number of HCs for WBC data. The larger *T* decreases (2) but increases (1). A threshold *T*=4 resulted in 61 HCs for the WBC dataset. Nine of these are majority benign HCs, containing 13 'misclassified' malignant points, i.e., points from the minority class of an impure HC. These 52 majority malignant HCs contain 12 misclassified benign cases total.

Malignant datapoints in the WBC dataset display much higher variation in value than the benign points. This results in many pure malignant HCs containing only a few datapoints. The implication is that any glyph displaying such large variation in shape as these malignant HCs should be considered malignant. For benign points, a more discriminatory pattern can be formed, as described below.

Out of 683 datapoints used, the data contain 444 benign points and 239 malignant points. Thus, with 25 data points misclassified by the algorithm, the purity of these HCs is 96.33%. The next task is to determine if the representative glyphs produced for these HCs can be used to classify unseen datapoints. A cursory visual analysis of the glyphs in Fig. 8 reveals distinct visual patterns between the two classes.

From the bird patterns in Fig. 8 we can formulate a **simple common pattern** for benign tumors. With 402 benign points, the first HC contains more than 90% of benign points in the data. This pattern is replicated in other benign dominant HCs.

Thus, with 90% of the benign data points, it is sufficient to consider this HC's pattern as **representative** of the benign class. As before, it is useful to define this pattern linguistically. Each benign bird is *a small bird with 3 crosses densely located*. All other birds are not benign, or at least suspicious to be malignant, so we do not need to memorize all variability of those bird shapes. The important rule is that if they do not belong to this simple category, they are malignant. The most significant four attributes of the dataset (those with the greatest average distance between classes) are encoded to the angles and lengths of the bird itself, with the remaining five less significant attributes used to encode the shifts of the SPC axes. Significant attributes are defined as those with the highest variation in value within the dataset. We converted 9-D data to 10-D data requited for SPC by duplications a most significant attribute.

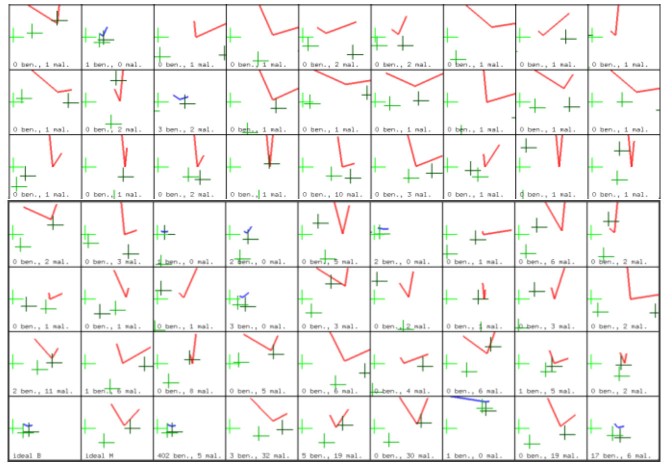
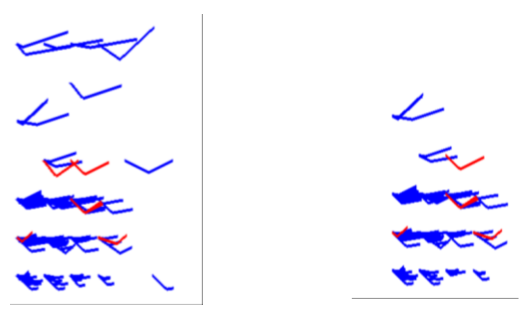


Figure 8. All glyph hypercubes from the WBC dataset constructed using HyG algorithm. Ideal glyphs for each class are displayed in the lower-left corner of the grid with labels "ideal B" and "ideal M". Remaining glyphs are representative glyphs for each HC. Color of representative glyph is determined by the dominant class of each HC (blue is benign, red is malignant). HC class counts are displayed at the bottom of each cell.

Next, we examine the patterns presented by the birds **within a hypercube**. This is demonstrated in the left cell of Fig. 9. A threshold $T = 5$ for the algorithm results in high variability within the HC. Benign glyphs further from the center of their HC than the misclassified malignant glyphs are included in the HC. This demonstrates the need for a relatively strict threshold value with the algorithm. The effectiveness of a lower threshold value can be seen in the right cell of Fig. 9. This cell displays the flock of birds computed for the first HC with a threshold $T = 4$. In this HC, most of the divergent benign glyphs have been removed. This reduces the likelihood of misclassification when using this HC.



(a) Hypercube with threshold T=5 with high variability of cases. (b) Hypercube with threshold T=4 with less variability of cases.

Fig. 9. All birds from the first hypercube containing benign (blue) and malignant (red) datapoints.

The next analysis, shown in Fig. 10, was conducted with a subset of these glyphs from the first HC. It contains all malignant datapoints in the HC, and the set of benign datapoints which meet the following criteria:

- An attribute of the datapoint differs from an attribute of the center point of the hypercube by a threshold T ≤ 2.
- This holds true for at least 2 attributes of the datapoint.

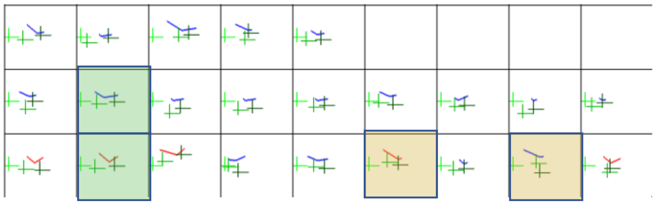


Fig. 10. Subset of glyphs from first hypercube calculated using threshold T = 4. Glyphs with high potential for misclassification highlighted in orange and green.

These glyphs represent the datapoints which are at highest risk for misclassification. As previously reported, the overall purity of all HCs is 96.33%. The purity of this first HC, which Fig. 10 displays a subset of, is 98.77% pure. Thus, the **likelihood of misclassification** based on HC is relatively low. Regardless, in a high-risk classification scenario, this small potential is significant. Both the visual patterns of the bird itself and the SPC crosses arrayed around the bird must be considered. One aspect of the glyph will not be enough to discriminate between classes in some cases, as demonstrated by the 'birds' of the cells highlighted in orange in Fig. 10. The consideration of all visual features of the bird glyph decreases the risk for misclassification. The risk does remain, however, as demonstrated by the two glyphs highlighted in green in Fig. 10.

D. Cancer dataset Glyphs with Hyperblocks

The HCs in Fig. 8 were constructed using the hypercube construction algorithm. Fig. 11 displays the *largest benign hyperblock* produced by a **more flexible** hyperblock algorithm, the Interval Merger Hyper Algorithm [9]. Potentially it can find purer HBs and a smaller number of HBs.

As shown, the **dominant visual pattern** for benign cases in this hyperblock is the pattern in the first hypercube in Fig. 8. With a purity of 99.71%, all glyphs in HB can be considered representative of the benign class. However, out of 105 glyphs, there are three cases which stand out as visually distinct due to their left wings being nearly twice as long as the typical case.

E. Glyph approach usability study with Cancer Dataset

To determine if visual patterns within the bird glyphs can be recognized by general users, we performed a usability classification study. 23 undergraduate computer science students were the test users. Subjects were asked to classify the grey glyphs reproduced in Fig. 12 based on the patterns provided by the ideal class glyphs. These ideal glyphs are shown in the same table at the left bottom corner colored by their classes.

Results of the survey demonstrate that users can **distinguish** between data classes using the visual patterns of glyphs. This classification can be done with high accuracy, even if it is a user's first exposure to the bird metaphor. In our survey of 23 students, we found that users were able to classify glyphs with an accuracy of 79% visually, without having numeric values of any attribute. This is in comparison to an accuracy of 87% from a kNN model ($k = 1$) using the two ideal glyphs as training data with numeric values of all attributes available for kNN. This shows the feasibility of the bird glyph approach with SPC.

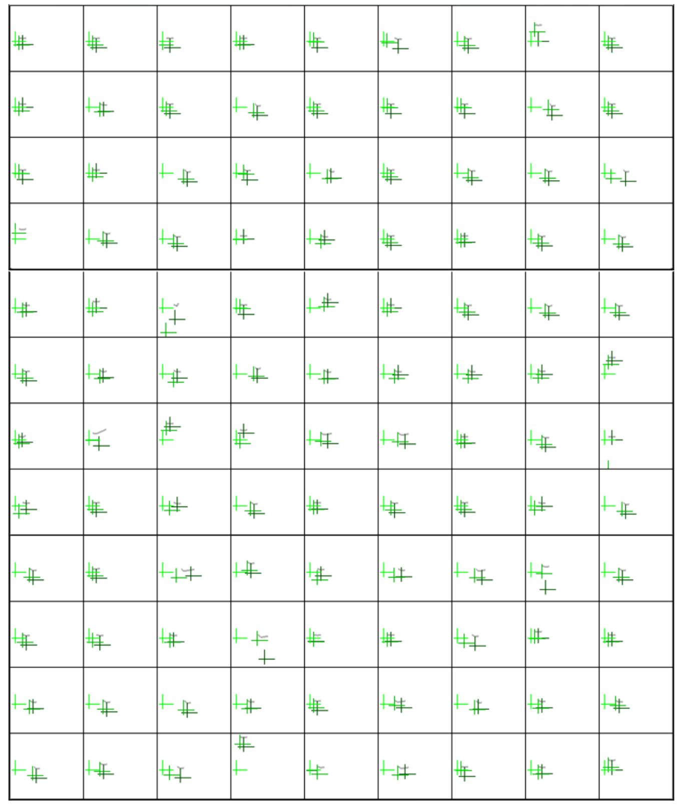


Fig. 11. The first 108 glyphs contained in the first hyperblock produced by the Interval Merger Hyper algorithm. This hyperblock contains most benign points in the dataset. Visual similarities between points within the hyperblock are clear.

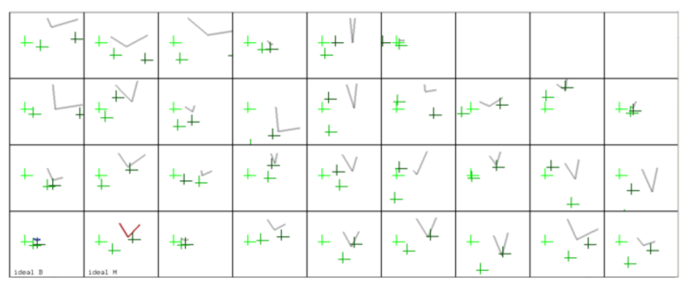


Fig. 12. Classification task given to students. Representative glyphs for each class shown in red and blue. Grey glyphs must be classified by the user.

I. GENERALIZATION

Below we represent more complex SF than birds in GLC. The GLC: (1) places *n* coordinates X_i in 2-D, (2) places attributes of the n-D point $\mathbf{x}=(x_1, x_2,\ldots,x_{1n})$ in these coordinates with each x_i mapped to X_i, (3) constructs graphs using (1)-(2). In [1] several such processes have been proposed.

The algorithm **Stick** to represent a SF in GLC is presented below and illustrated in Fig. 14.

Step 1. Number all nodes of the SF using the numbering algorithm Num described later.

Step 2. Put the origin of the first pair of orthogonal coordinates (X,Y) to node N_1 defined by the Num algorithm for the SF.

Step 3. Map attribute X_1 to X and attribute X_2 to Y coordinates.
Step 4. Place pair (x_1,x_2) from n-D point $\mathbf{x}$=(x_1,x_2,…,x_n) in (X_1,X_2).
Step 5. Draw a line from the origin to point (x_1,x_2) in (X_1,X_2).
Step 6. Conduct steps 2-5 for each node N_i of the SF instead of only N_1 in step 2, which contain edges not included in the cartesian coordinates by prior nodes.

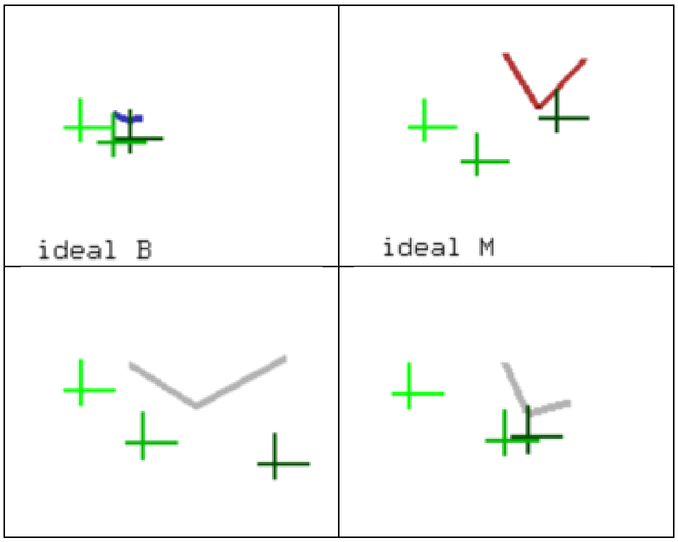


Fig. 13. Zoomed selected cases from Fig. 13.

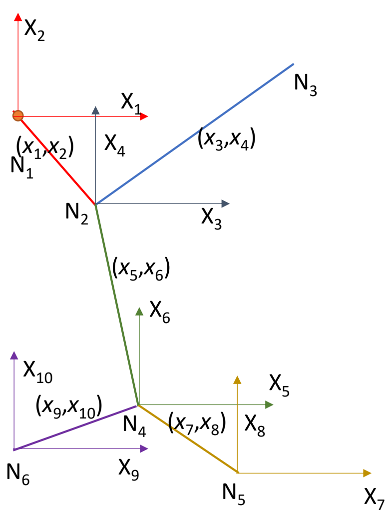


Fig. 14 Stick figure in GLC with a set of Shifted Paired Coordinates. The colored coordinates show the respective pair of coordinates, and the colored lines indicate the edges built using this pair of cardinalates.

The stick figure node numbering algorithm **Num** is as follows:
Step 1. Select the top left node of the SF, mark it by a circle, and assign it an index 1 and denote N_1.
Step 2. Find the edge from node N_1 that has the smallest angle with Y coordinate on the right from Y coordinate (clockwise move from the North (12 o'clock). Assign the second node of this edge index 2 and denote it N_2.
Step 3. Conduct Step 2 for all edges from node N_1 with increasing angle to Y coordinate. Assign these nodes consecutive indexes i and denote N_i.
Step 4. Conduct Step 3 for all edges for each node N_i. Stop when all nodes of the stick figure are processed.
Note that the visualization with SPC in Fig. 14 can be equivalently presented in the Polar coordinates.

II. Conclusion

We have demonstrated that bird glyphs can be effectively employed in classification tasks. These glyphs are **simple to use** and **easy to understand**. Domain experts may easily begin working with the bird metaphor, with little training required. The compact, lossless glyphs allow visualizing them side-by-side like in Fig. 8 without the overlap.

Further work with these hybrid glyphs will involve extending the bird metaphor to accommodate higher than 10-D data. Segments can be added to the Stick Figure, resulting in a "snake" metaphor. Multiple segments can be added to both ends of the initial segment to result in a "person" metaphor. These larger glyphs can complement the bird glyphs and used together to represent data of larger dimensions.

In lower risk classification scenarios, the glyphs may be quickly employed. These glyphs may be used as tools in high-risk classification scenarios too. In this case, more care may be taken in presenting the glyphs. Interactive methods of increasing the visual distinctions between classes will be further explored. More accurate classification rules can be formed through deeper exploration hypercubes and hyperblocks.

References

[1] Fuchs J, Isenberg P, Bezerianos A, Keim D. A systematic review of experimental studies on data glyphs. IEEE transactions on visualization and computer graphics. 2016 Mar 31;23(7):1863-79.
[2] Koc K, McGough AS, Johansson Fernstad S. PeaGlyph: Glyph design for investigation of balanced data structures. Information Visualization. 2022 Jan;21(1):74-92.
[3] Jiang L, Liu S, Chen C. Recent research advances on interactive machine learning. Journal of Visualization. 2019 Apr 9;22:401-17.
[4] Chernoff, H. (1973). The Use of Faces to Represent Points in K-Dimensional Space Graphically. *Journal of the American Statistical Association*, *68*(342), 361–368. https://doi.org/10.2307/2284077
[5] R. M. Pickett and G. G. Grinstein, "Iconographic Displays of or Visualizing Multidimensional Data," *Proceedings of the 1988 IEEE International Conference on Systems, Man, and Cybernetics*, Beijing, China, 1988, pp. 514-519, doi: 10.1109/ICSMC.1988.754351.
[6] Kovalerchuk B, Visual Knowledge discovery and Machine Learning, Springer, 2018
[7] Kovalerchuk, B., & Grishin, V. (2019). Adjustable general line coordinates for visual knowledge discovery in n-D data. *Information Visualization*, *18*(1), 3–32. https://doi.org/10.1177/1473871617715860
[8] Kovalerchuk, B., Nazemi, K., Andonie, R., Datia, N., Bannissi E. (Eds), Integrating Artificial Intelligence and Visualization for Visual Knowledge Discovery, Springer, 2022
[9] Kovalerchuk B, Ahmad MA, Teredesai A. Survey of explainable machine learning with visual and granular methods beyond quasi-explanations. Interpretable artificial intelligence: A perspective of granular computing. 2021:217-67.