Multilayer Development and Explanation of Machine Learning Models with Visual Knowledge Discovery

Lincoln Huber
Department of Computer Science
Central Washington University
Ellensburg, WA, USA
Lincoln.Huber@cwu.edu

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

***Abstract*—Explainable machine learning is important for increasing the confidence of domain experts in each model. However, simply using explainable methods is not enough. Explainable methods such as Decision Trees and Rule-Based models can be insufficient for certain datasets to provide satisfactory models. To solve this problem, Visual Knowledge Discovery (VKD) offers a solution through explainable, interactive, reversible lossless *n*-D visualizations using General Line Coordinates (GLC). Nevertheless, VKD and other visual methods suffer from occlusion when visualizing large amounts of data. This paper presents methods for both removing visual occlusion and limiting rule complexity and overfitting with hyperblocks at several levels. Hyperblocks create intrinsically explainable rules using meaningful numeric attributes. They allow finding and generalizing areas where data are similar, decreasing the occlusion in each hyperblock while creating less complex and more general rules. Major benefits from higher level hyperblocks include increased generalization, less complex rules, and increased user confidence in each classification.**

Keywords*—*Visual Knowledge Discovery, Explainable Machine Learning, General Line Coordinates, Machine Learning, Visualization, Classification

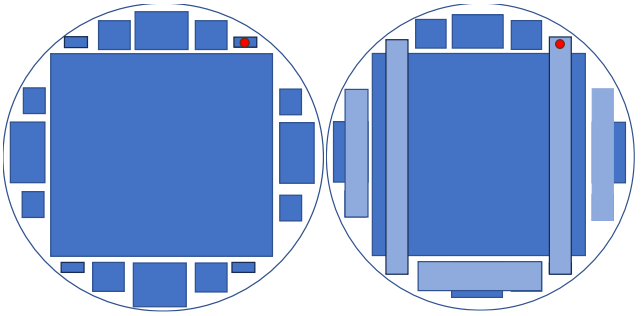
I. Introduction

The motivation to develop hyperblocks (HBs) at several levels is explained below and demonstrated in Figure 1 for 2-D data. The training data of a given class are distributed as it is shown in Figure 1a, where they are evenly distributed in each rectangle. Intuitively it is clear that the data form a circle or a hypersphere in n-D space. However, while it has a simple mathematical description, the hypersphere has no interpretation for heterogeneous attributes [1]. In contrast, rectangles, which are equivalent to simple logical formulas of a conjunction of intervals on individual attributes, are explainable in the respective domains for heterogeneous data [1].

Therefore, *rectangular envelopes,* which will be called **blocks,** are generated around those data to get an explainable generalization of these data. Similarly, n-D rectangles are called **hyperblocks** with several algorithms developed to construct them [3,4]. In Figure 1a, there are 17 blocks (16 small blocks and one large block). It is an obvious deficiency of the explainable HB approach when instead of a single circle there are 17 rectangles.

The problem is not only in the large number of blocks but also in the **low confidence** of prediction with **small blocks**. Consider the tiny block on the top right in Fig 1a with the new red case that needs to be predicted. This block has only a few training cases from the class, which does not lead to high confidence in prediction of this class.

This is a typical example of data overfitting. In contrast, Figures 1b and 1c show this case belonging to a much larger block, and finally Figure 1d shows it belonging to a very large block outlined with a dotted line. These larger blocks create much higher confidence in the case's classification when communicated to the domain expert/user.



(a) Level 1: 17 blocks (16 small blocks and one large block)

(b) Level 2: 12 blocks (9 small blocks and 3 larger blocks)

(c) Level 3: 8 blocks (2 small blocks and 6 larger blocks)

(d) Level 4: 6 large blocks

Fig. 1. Multilevel generalization of blocks.

Figure 1 demonstrates the process of generalization of blocks from level 1 to level 4, where 17 blocks of level 1 are generalized to 12 blocks of level 2, which are generalized to 8 blocks of level 3, which are finally generalized to 6 large blocks of level 4. While this visualization allows us to see and conduct naturally the **hyperblock generalization process** in 2D space, for n-D data in n-D space it cannot be done in the same way. More sophisticated visualization methods to losslessly **visualize** n-D data are needed. General Line Coordinates (GLC) [1] allow such visualizations without loss of multidimensional information for the HB generalization process, which is described in the next section.

Figure 2 outlines the proposed explainable machine learning model development process. It contains 4 stages: (I) select a task in model development, (II) discover issues, (III) generate multilayer concepts, and (IV) apply algorithm to resolve the issues. The key concepts in this approach are HBs and GLC-L at several levels to build interpretable models. The listed four stages outline the process of discovering issues with these concepts and resolving them. Below Figure 2 is commented with an example. Consider some model. While exploring this model it is discovered to have many low confidence HBs. Then, coinciding multilayer HB and GLC-L concepts are generated to resolve the discovered issues like HBs of level 2.

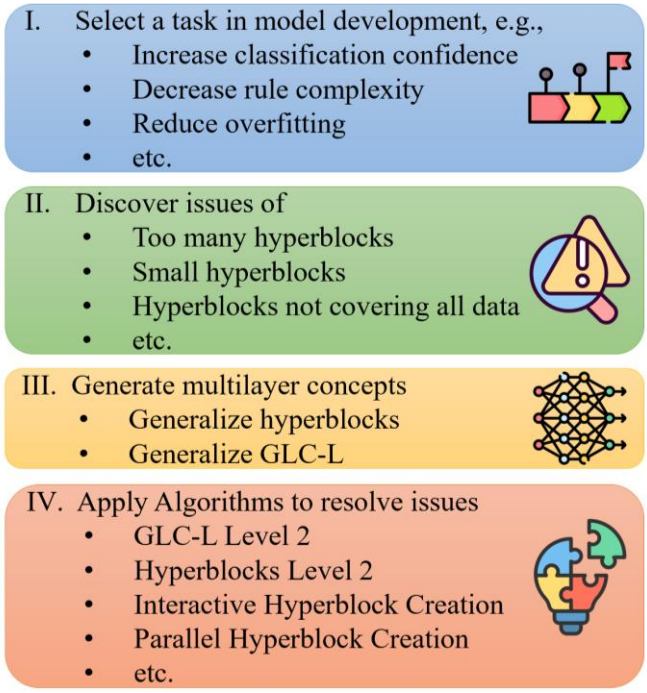


Fig. 2. Outline of concept and user approach.

By following this process, highly explainable *human-centered* machine learning models can be developed. These models will increase user confidence in each prediction and maximize the potential for interactive Visual Knowledge Discovery.

This paper is organized as follows. First, the definitions and algorithms used to create and visualize HBs are discussed in Section II. Then, the results from several case studies are shown in Section III. Next, current and upcoming features are discussed in Section IV for the DV2 software tool. Last, the pros and cons for each algorithm are discussed and all future work is mapped out in Section V.

II. Definitions and Algorithms

A. Definitions of GLC-Linear Level 1 and 2

GLC-Linear (GLC-L) is a type of GLC capable of discovering linear models and solving supervised learning classification tasks in 2-D [1,2]. To produce a GLC-L visualization, coefficients c_i of $F(\mathbf{x})$ need to be normalized to produce a new linear function $G(\mathbf{x}) = k_1x_1 + k_2x_2 + \ldots + k_nx_n$ as described below. Let $K = (k_1, k_2, \ldots, k_n)$ and $k_i = c_i / |c_{max}|$, where $|c_{max}| = \max_{i=1:n}|(c_i)|$. Here all k_i are normalized to the interval [-1, 1]. The following property is true for F and G:

$$F(\mathbf{x}) < T \text{ if and only if } G(\mathbf{x}) < T / |c_{max}|.$$

Figure 3 shows a visualization of GLC-L level 1.

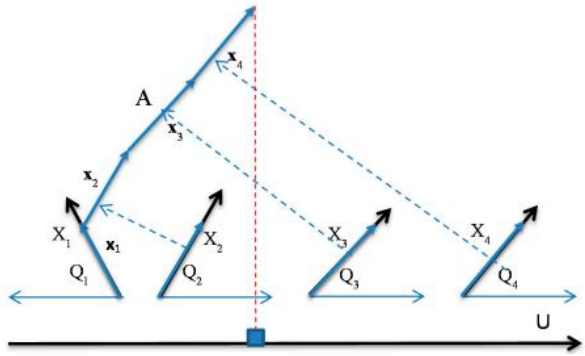


Fig. 3. Example of GLC-L level 1. 4-D point A = (1, 1, 1, 1) in GLC-L coordinates X_1 – X_4 with angles (Q_1, Q_2, Q_3, Q_4). Vectors $\mathbf{x}_i$ are shifted to connect one after another and the end of the last vector is projected to the black line [1,2].

Figure 4 shows the idea of producing a GLC-L level 2 visualization. In this figure lines L_{21} and L_{22} represent GLC-L level 2 with the blue lines forming L_{21} and the green lines forming L_{22}. Mathematically L_{21} is a sum of blue vectors and L_{22} is a sum of green vectors: $L_{21}=\mathbf{x}_1+\mathbf{x}_2+\mathbf{x}_3+\mathbf{x}_4$, $L_{22}=\mathbf{x}_5+\mathbf{x}_6+\mathbf{x}_7+\mathbf{x}_8+\mathbf{x}_9$, which will be called **vector summation**. The advantage of GLC-L level 2 is in the generalization of vectors $\mathbf{x}_i$ of GLC-L level 1. It decreases details and simplifies polylines, making patterns more visible. It is especially beneficial when multiple cases are visualized with fewer crossings of their polylines. There are several ways to produce GLC-L level 2, below are some options. The simplest way is *summing up* two adjacent level 1 vectors: $L_{21}=\mathbf{x}_1+\mathbf{x}_2$, $L_{22}=\mathbf{x}_3+\mathbf{x}_4$, $L_{23}=\mathbf{x}_5+\mathbf{x}_6$ and so on. Another method is *interactive* where a user visually groups vectors $\mathbf{x}_i$ like it is shown in Figure 4. The advantage of an interactive method is that a user, as a domain expert, can do this in a semantically meaningful way for the domain. The next method first orders all vectors $\mathbf{x}_i$ according to their coefficients in a Linear Discriminant Function (LDF) from the smallest one to the largest one. Typically attributes with the smallest coefficients are least contributing to the total value of a LDF and they can be combined into L_{21} with a relatively significant contribution. Other vectors $\mathbf{x}_i$ with large contributions can be kept at level 2 without grouping them: $L_{2k}=\mathbf{x}_i$.

Another benefit of level 2 is for situations when some coefficients are negative. Here vectors with negative

coefficients are grouped with vectors with positive coefficients leading to the sum level 2 line with a positive coefficient. Line L_{22} in Figure 4 shows it where $\mathbf{x}_6$ and $\mathbf{x}_5$ have negative coefficients but together with $\mathbf{x}_7$-$\mathbf{x}_9$ they form L_{22} with a positive coefficient. It also simplifies patterns and analysis of them. Additionally, level 2 can be generalized to **level 3** and **higher levels**. For example, in Figure 4 vectors L_{21} and L_{22} can be summed up to form a level 3 vector: $L_{31}=L_{21}+L_{22}$.

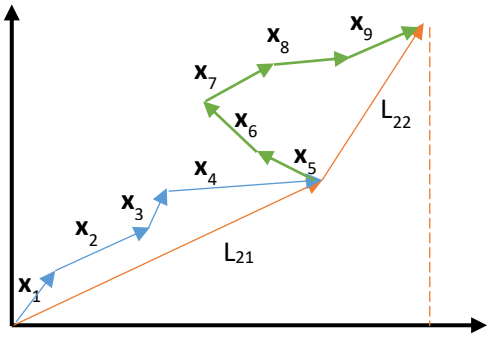
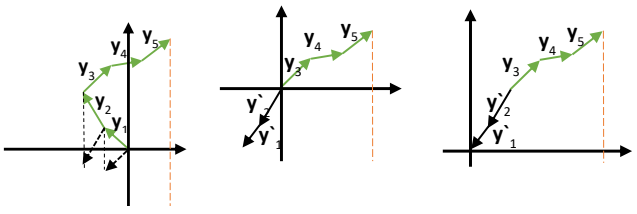


Fig. 4. Example of GLC-L level 2.

B. Algorithm for Producing Simplified GLC-L

When all coefficients of a LDF are positive it simplifies the GLC-L visualization making it easier for the user to analyze it. Below, methods to modify LDFs with some negative coefficients to make all of them positive are discussed. Generation of GLC-L level 2 resolves this issue indirectly because the coefficients of level 1 are still negative. Below are explicit methods where LDFs are modified at the same level to make all coefficients positive.

In GLC-L all attributes are normalized to be within interval [0, 1], this interval can be changed to [-1, 0] by multiplying all values of the attribute x_i with a negative coefficient k_i. This will change it to $-k_i$, which is positive. This will reverse the direction of the vector x_i in the GLC-L visualization. See Figure 4. Here the vectors were negated to $\mathbf{y}`_5=-\mathbf{y}_5$, $\mathbf{y}`_6=-\mathbf{y}_6$: $\mathbf{y}_5+\mathbf{y}_6+\mathbf{y}_7+\mathbf{y}_8+\mathbf{y}_9=-\mathbf{y}`_5-\mathbf{y}`_6+\mathbf{y}_7+\mathbf{y}_8+\mathbf{y}_9$. Figure 5b shows two versions of locating the origin for GLC-L coordinates. The top one separates positive and negative vectors, and the bottom one puts all vectors to the positive quadrant. The advantage of the last one is in allowing using the 4th quadrant to show cases of the opposite class as it is done in the GLC-L implementation in the DV2 program [5]. Technically conversion from Figure 5a to Figure 5b is done by producing negated vectors $\mathbf{y}`$ with mirroring $\mathbf{y}$ vectors, reversing their directions and putting one after another one. This is shown with dotted lines in Figure 5a.



(a) GLC-L visualization with negative coefficients for $\mathbf{x}_5$, $\mathbf{x}_6$.

(b) Two GLC-L visualizations with positive coefficients for all $\mathbf{x}_i$.

Fig. 5. Converting GLC-L for LDF with negative coefficients (a) to GLC- with positive coefficients (b).

C. Definition of Hyperblock

A hyperblock (n-orthotope) [10] is a set of n-D points $\{\mathbf{x}=(x_1,x_2,\ldots,x_n)\}$ with **center** in n-D point $\mathbf{c}=(c_1,c_2,\ldots,c_n)$ and **side lengths** $\boldsymbol{L}=(L_1, L_2,\ldots, L_n)$ such that

$$\forall i \;\; \| x_i-c_i \| \leq L_i/2 \qquad (1)$$

Figure 10 illustrates this concept. It shows 26 HBs for the Wisconsin Breast Cancer dataset.

D. Algorithms for Producing Hyperblocks of Level n

To create HBs of level *n*, the same HB creation algorithms are used as for level 1. For this paper, the HB creation algorithm used was the General Line Coordinate Hyperblock Rules Linear (GLC-HBRL) [4]. The difference between each HB level comes from the data being used in the creation of the HBs. For HBs of level 1, all cases are used. For HBs of level *n*, **representative cases** for each HB of level *n*–1 are used. Any algorithm capable of gathering these representative cases can be used alongside the Level *n* Hyperblock Creation Algorithm to create higher level HBs. Algorithm L*n*HC shows this process.

Level *n* Hyperblock Creation (L*n*HC) Algorithm

1. Create HBs of level *n*.
2. For each HB, select an algorithm to Create a Representative Dataset (CRD algorithm).
3. Create the representative dataset by using CRD algorithm.
4. Use the representative dataset of each HB to create level *n*+1 HB with the same HB creation algorithm used in Step 1.
5. Repeat steps 2 and 3 until desired level is reached.

To get a representative dataset for the creation of higher level HBs, many algorithms can be used. For a simple representation, one can find cases corresponding to the envelope of each HB. This is done in Algorithm CRD1, the steps of which are below. Other methods include finding the most average case shown in Algorithm CRD2 and splitting each HB into thirds and finding the most average case for each HB third shown in Algorithm CRD3.

For this paper all higher level HBs are created with this algorithm. This is because using an envelope representation tended to perform better than the other methods that were tested. However, that is not to say these other data representations are ineffective. For instance, if *explainable* methods will be used instead of *unexplainable* methods when finding the most average cases in Algorithms CRD2 or CRD3, then it will increase the interpretability of HBs created through such cases.

Finding the most average cases is based on a selected similarity method of cases. While the Euclidean distance is very common, it is **not explainable** for heterogeneous data [8], which is typical in machine learning. Data attributes in case studies that are presented in this paper are heterogeneous like the angles, distances, and roots of the moments of the Magic

Gamma Telescope (MGT) data. Conversely, one explainable method is the **explainable threshold similarity (ETS) method.** It uses a threshold T or a set of thresholds (T_1, T_2, …, T_n) for two n-D points **x** and **y** such that

$$|x_1 - y_1| < T_1 \,\&\, |x_2 - y_2| < T_2 \,\&\, \ldots \,\&\, |x_n - y_n| < T_n. \quad (2)$$

It compares only values of each attribute without summing up and squaring values of different attributes. The Euclidean distance also can be called as a **lossy similarity method**. It sums up and squares the similarity values of individual attributes. As a result, the individual differences $|x_1 - y_1|$ are lost and not restorable from the Euclidean distance. Many other similarity methods are lossy distance techniques, which may unexplainably compress multidimensional heterogeneous data into a single value [4]. In contrast ETS keeps this information when thresholds are tight and can be used with Algorithms CRD2 or CRD3 presented below to produce more explainable results.

Algorithm CRD1: Envelope Case Finder

1. Get all cases in each HB.
2. For each attribute of each case, check if the value is equal to the minimum or maximum value of the HB.
   a. If yes, add given case to the set of envelope cases for the HB and stop looking for cases equal to the current minimum or maximum value.
   b. If no, then continue searching.
3. Return envelope case set for the given HB.

Algorithm CRD2: Average Case Finder

1. For a given HB, create an artificial case using the average value for each attribute.
2. Using some distance technique, find the closest case to the artificial case created in Step 1.
3. Return the case found in Step 2 as a representative set.

Algorithm CRD3: Average Case Finder with Strips

1. Split a given HB into thirds on the range of each attribute.
2. For each HB third, create an artificial case using the average value for each attribute.
3. Using some distance technique, find the closest case to each artificial case.
4. Return the cases found in Step 3 as a representative set.

Additionally, optimizing our HB creation algorithms can also improve results. One way of doing this is by basing the construction of HBs on Decision Trees. Decision Trees are some of the most widely used explainable supervised learning classification techniques. Their construction through recursive partitioning to create the most homogeneous subsets of data can be mirrored in the HB creation process. Algorithm DT-HB shows this process.

Decision Tree Hyperblocks (DT-HB)

1. Create HBs of level *n*.
2. Find the largest HB created and store it.
3. Get all data not in the largest HB.
4. Use the data from Step 3 to create more HBs.
5. Repeat Steps 2-4 until all data is classified.

E. Interactive Algorithm for the Creation of Hyperblocks

Figure 6 shows the overlap area of the Wisconsin Breast Cancer (WBC) dataset [6] in GLC-L visualized in the DV2 program. This is the area between the leftmost and rightmost misclassified points within the visualization [4]. The green dotted vertical line indicates the discrimination line of two classes. The yellow vertical lines on the right and the left side indicate limits of the overlap area. Figure 6 shows *pure subareas* colored with their respective classes in the LDF overlap area. These areas can be interactively captured by a user using the GLC Interactive Rules Linear (GLC-IRL) algorithm in the DV2 program [4]. Next a user can analyze these pure areas. If the areas are too narrow from the user's viewpoint, the user can skip them to avoid overfitting. For instance, a user can decide that two magenta areas are too narrow. A user also can relax the requirement of fully pure areas but can search for areas with a purity over some threshold, say, 90%. These methods allow users to interactively and interpretably create higher level HBs. The areas that the user consider as pure enough will be classified by the **constrained LDF** in the form:

If $E_{11} \le F(\mathbf{x}) \le E_{12}$ or $E_{21} \le F(\mathbf{x}) \le E_2$ or …or $E_{k1} \le F(\mathbf{x}) \le E_{k2}$

then $\mathbf{x} \in$ Class 1,

where E_{i1} is the start and E_{i1} is the end of the interval.

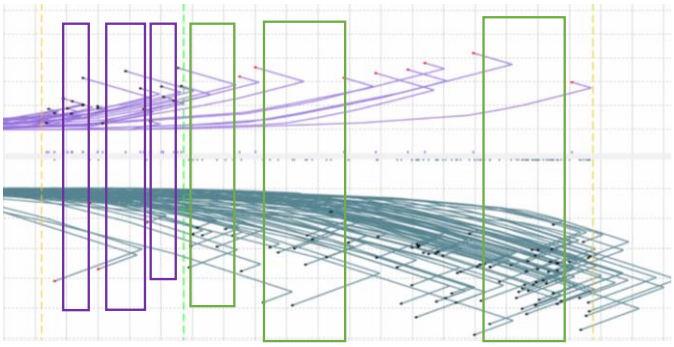


Fig. 6. LDF overlap area with pure subareas colored with respective classes.

F. Paralelization Algoritrhm

It is important to reduce the time spent generating HBs for large datasets. One way to do this is with the parallelization algorithm presented below, where each step can be run in parallel. Figure 7 shows a diagram of this process.

III. Cases Studies

Below are several case studies that illustrate the HB and GLC-L algorithms.

A. Case study 1: Level 2 GLC-L with MNIST data

This case study analyzes the MNIST dataset digits 0 and 1 with 784 features and 12665 instances of the two classes [9]. Figure 8a shows a GLC-L visualization of these data with 99.48% accuracy. Table 1 shows a confusion matrix summarizing the performance of this visualization. Here the visualization is weakened by the density of cases, and it is difficult to analyze any specific case. To improve this, the visualization can be focused to only the overlap area. This is the

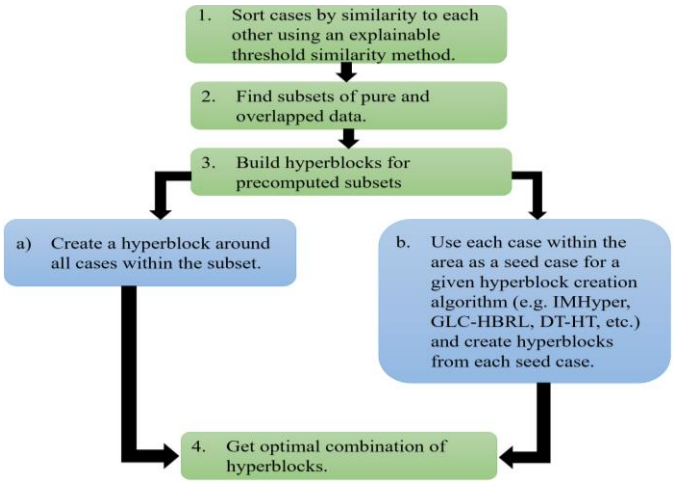
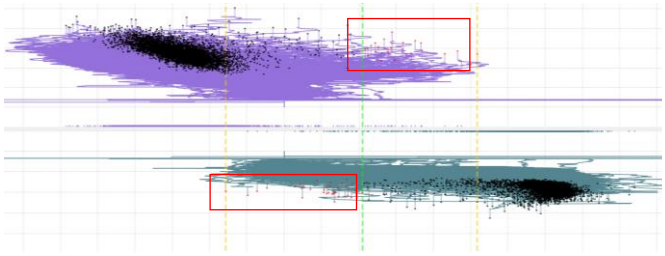
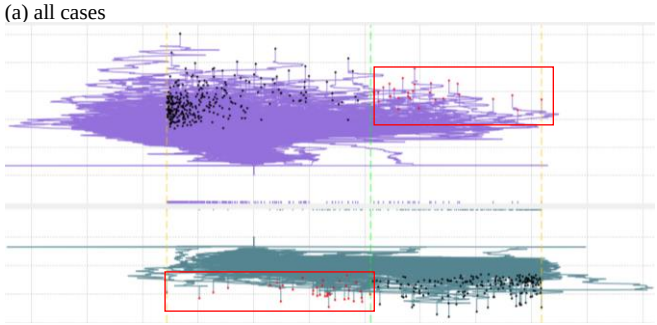


Fig. 7. Parallel Hyperblock Creation Algorithm. Green blocks can be done in parallel.

area between the leftmost and rightmost misclassified points within the visualization [4] and contains only 559 cases. This removes most confidently classified data, which require less analysis. Figure 8b shows the same GLC-L visualization but with only the overlap area.



(a) all cases

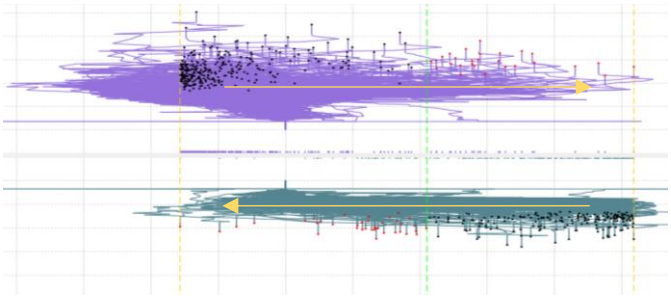


(b) overlap area cases

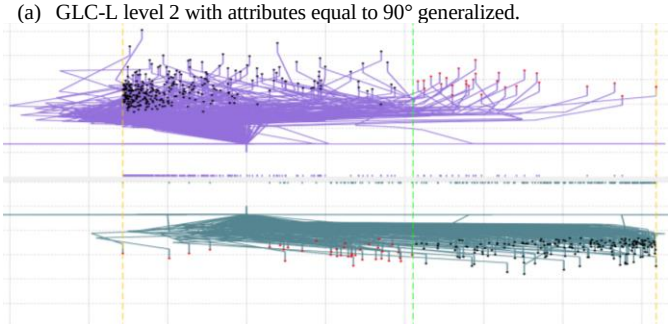
Fig. 8. MNIST dataset digits 0 and 1 visualized in GLC-L. Green line – class discrimination line. Yellow lines – overlap area bounds.

Table 1. MNIST digits 0 and 1 confusion matrix for GLC-L.			
	Predicted Condition		
Actual Condition	Class	0	1
	0	5895	28
	1	38	6704

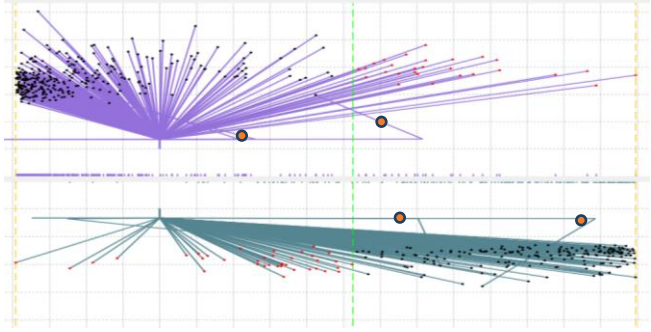
By visualizing only the overlap area the visualized data can be reduced from 12665 cases to 559 cases (4.41%). This allows the structure and patterns within the data to become more apparent. For instance, it is visible that the green blob is shorter than the purple blob. By moving from the 1-D projection used by the LDF to 2-D rectangle properties, the accuracy of classification can be improved. This can be done using the GLC-IRL algorithm mentioned in section IID. In Figures 8a and 8b this is captured by red rectangles. It should be noted that this property is less apparent in Figure 8a with all data visualized. Then, to further simply the visualization GLC-L level 2 can be applied to the overlap data. Figure 9 shows various examples with GLC-L level 2 with vector summations of least contributing attributes as defined in section IIA. In each visualization the threshold for being a least contributing attribute is loosened to create a more *generalized visualization*.



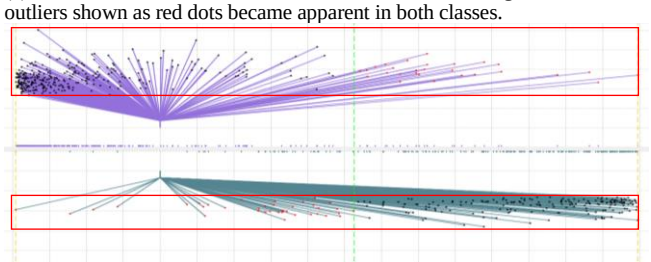
(a) GLC-L level 2 with attributes equal to 90° generalized.



(b) GLC-L level 2 with attributes between 89° and 91°generalized.



(c) GLC-L level 2 with attributes between 88° and 92° generalized. The outliers shown as red dots became apparent in both classes.



(d) GLC-L level 2 with attributes between 87° and 93°generalized.

Fig. 9. MNIST dataset digits 0 and 1 overlap area visualized with GLC-L Level 2.

Figure 8a shows all data in GLC-L and Figure 8b shows only data in the overlap area. With these visualizations, it is difficult to distinguish between individual cases. In Figure 9a, by generalizing all attributes equal to 90°, individual cases

become easier to distinguish. This remains true for each successive visualization in Figure 9. As more attributes get generalized, the easier it becomes to distinguish between individual cases.

Note, that the data can be overgeneralized, e.g., in Figure 9c, many outliers are visible and are marked with orange circles. These outliers are lost and are not visible in Figure 9d, where data are overgeneralized. However, this overgeneralization is useful too, making the difference between the height of the cases clearer as the red rectangles show.

Additionally, both classes show that negative and positive attributes are cut out or became less dense and positive. This shows simplification in GLC-L level 2 in Figure 9 that can be useful for the user.

In Figure 9b, the most significant cut out was for the purple class on the left relative to Figure 9a. It allows us to analyze the impact of a positive and negative contributions. In Figure 9b, the yellow arrows show that classes are going in opposite directions. Thus, visual exploration helps to find the impact of individual and groups of attributes.

B. Case study 2: Level 2 hyperblocks for WBC data

Figure 10 shows 26 level 1 HBs created for the WBC data using the GLC-HBRL HB creation algorithm. These HBs cover all 683 cases with 99.9% accuracy. However, 13 HBs of these 26 HBs contain a single case, this means these HBs are overfit.

In contrast, Figure 11 shows the second level of the same HBs created using data from algorithm CRD1: Envelope Case Finder. In the second level, there are 13 HBs instead of 26 with only one HB containing a single case. The accuracy remains similar at 99%. This shows the potential to minimize overfitting with level 2 HBs. Additionally, for the user, analyzing 13 HBs is much easier than 26.

However, of the 683 cases in the WBC dataset, 92 were not picked up by the level 2 HBs. This means only 86.5% of the WBC data is being represented by the level 2 HBs. A simple solution to this is shown in Figure 12 where the 92 missing cases are used to separately create two more level 2 HBs. In this way the 26 level 1 HBs to can be reduced to 15 level 2 HBs.

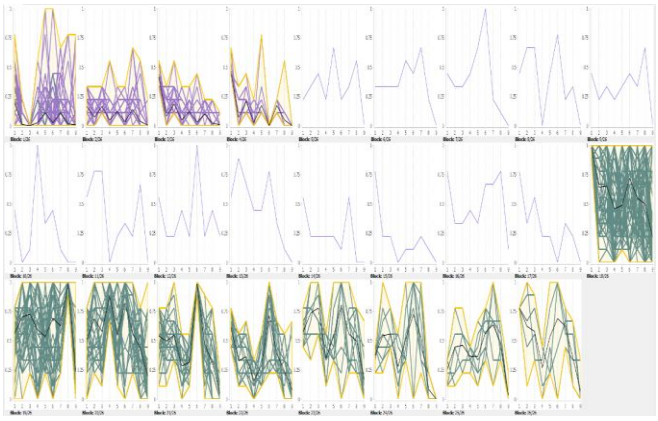


Fig. 10. 26 level 1 hyperblocks for the WBC data.

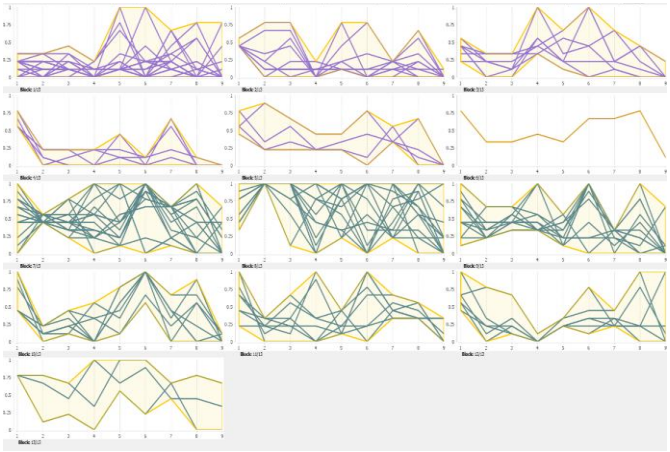


Fig. 11. 13 level 2 hyperblocks for the WBC data.

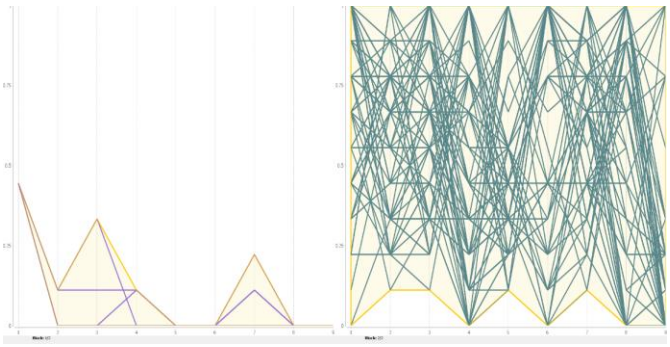


Fig. 12. Two hyperblocks created separately the using remaining 92 cases that are not in HBs from Fig. 11.

C. Case study 3: Level 3 hyperblocks for MGT data

This case study analyzes Magic Gamma Telescope (MGT) Dataset [7], with 10 features and 19020 instances of 2 classes. The features are listed in Table 2.

Table 2. Features of Magic Gamma Telescope (MGT) Dataset.

```
1. fLength:  continuous  # major axis of ellipse [mm]
2. fWidth:   continuous  # minor axis of ellipse [mm]
3. fSize:    continuous  # 10-log of sum of content of all pixels [in #phot]
4. fConc:    continuous  # ratio of sum of two highest pixels over fSize  [ratio]
5. fConc1:   continuous  # ratio of highest pixel over fSize  [ratio]
6. fAsym:    continuous  # distance from highest pixel to center, projected onto major axis [mm]
7. fM3Long:  continuous  # 3rd root of third moment along major axis  [mm]
8. fM3Trans: continuous  # 3rd root of third moment along minor axis  [mm]
9. fAlpha:   continuous  # angle of major axis with vector to origin [deg]
10. fDist:   continuous  # distance from origin to center of ellipse [mm]
11. class:   g,h         # gamma (signal), hadron (background)

g = gamma (signal):    12332
h = hadron (background): 6688
```

Figure 13 visualizes these data in GLC-L using the DV2 program. Using the GLC-HBRL algorithm to create HBs for the LDF, an explainable set of rules that match the LDF can be created. This is shown in Figure 13. On its own, LDFs are not explainable for heterogeneous data [4,8]. This is like the Euclidean distance mentioned above. The advantage of this approach is the potential for explainable HB rules with higher accuracy than the LDF. A disadvantage is that attempting to modify HBs to match a LDF can result in many HBs containing only a single case [4]. In experiments with the MGT data, 3787

HBs were generated with an accuracy 99.2%. These HBs can be seen in Figure 14. Of these HBs, the largest one contains only 269 cases while 3253 of these HBs contain only a single case. Like Case Study 2, it is obvious that the level 1 HBs are overfitting the data to explain the given LDF.

By creating level 2 HBs, the number of HBs can be reduced from 3787 to 437. These HBs can be seen in Figure 15. This is an 88.5% reduction in the number of HBs. Of these level 2 HBs, the largest contains 8449 cases and none contain only a single case. However, the accuracy of these level 2 HBs drops to 83.3%. Nonetheless, this shows that the creation of level 2 HBs can be used to drastically reduce overfitting at the cost of some accuracy. Moreover, it is significantly easier to analyze 437 sets of HB rules in comparison to 3787.

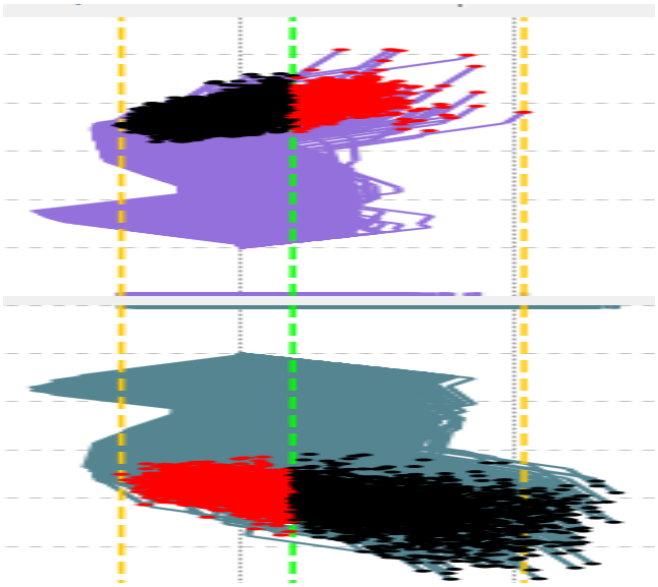


Fig. 13. MGT dataset visualized in GLC-L with 79.83% accuracy.

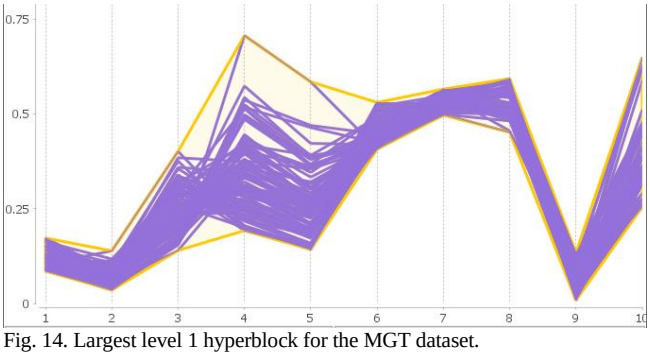


Fig. 14. Largest level 1 hyperblock for the MGT dataset.

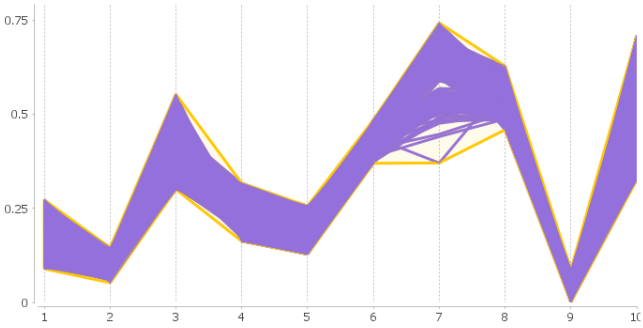


Fig. 15. Largest level 2 hyperblock for the MGT dataset.

D. *Case study 4: DT-HBs for WBC data*

This case study shows the potential for alternative HB creation algorithms. In the previous case studies, HBs were generated using the GLC-HBRL algorithm. This algorithm builds HBs to explain an LDF. Alternative HB creation algorithms can provide an explainable model directly without referencing unexplained LDF. For instance, the Interval Merger (IMHyper) [4] HB creation algorithm does not use LDFs. For the WBC dataset, using IMHyper produces 18 HBs with 100% accuracy. These HBs are shown in Figure 16.

Moreover, by using IMHyper with the DT-HB HB creation algorithm the number of HBs can be reduced from 18 to 13 at the cost of 2.5% accuracy. These HBs are shown in Figure 17. Comparing these 13 HBs with the 15 level 2 HBs created in Case Study 2 show how changes made to the HB creation algorithm can lead to results equal to or greater than that of level *n* HB creation algorithms. This shows the multitude of ways in which HB generalization can be developed.

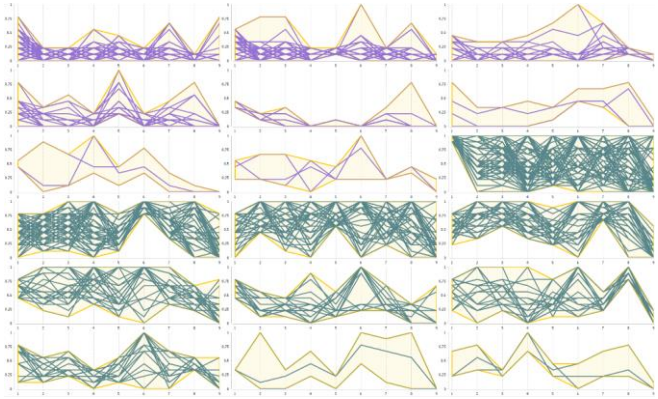


Fig. 16. WBC dataset visualized with 18 hyperblocks with 100% accuracy.

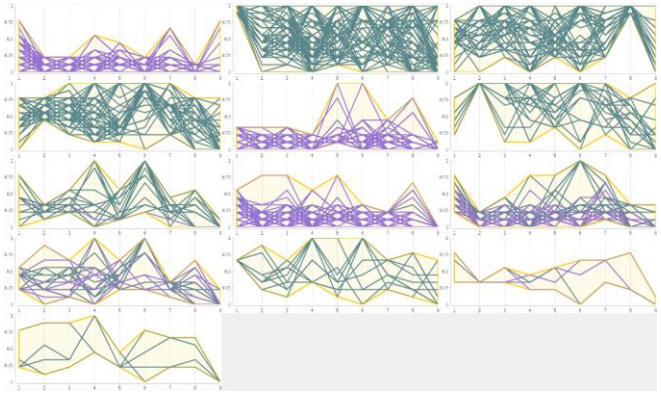


Fig. 17. WBC dataset visualized with 13 DT hyperblocks with 97.5% accuracy.

IV. Features of Software Tool

A. *Implemented Features*

The DV2 program [5] was developed and used for all experiments shown in this paper. Figure 18 shows an example of this program visualizing the WBC dataset in GLC-L. A

confusion matrix shows the results of this visual model in the bottom left corner.

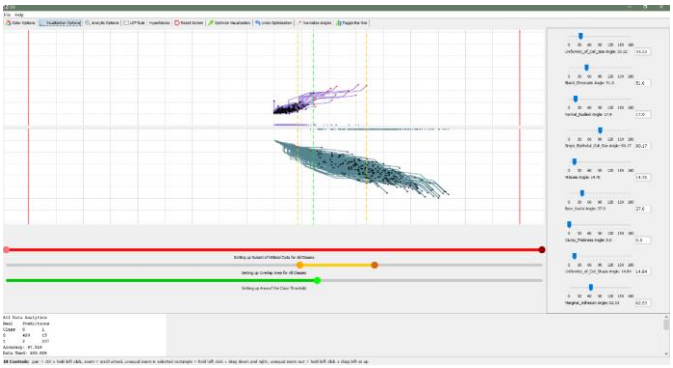


Fig. 18. WBC dataset visualized with 98.1% accuracy with GLC-L in DV2.

The option to create HBs is in the toolbar at the top of the window. By selecting this option HBs will be created and a various HB options will appear. Figure 19 shows this HB window. Options include the ability to increase the HB level and visualize in Parallel Coordinates (PC), GLC-L, and Shifted Paired Coordinate (SPC). All these lossless visualization methods for multidimensional data are defined in [1]. Options to visualize HBs together or separately, and the option to visualize overlap data within each HB separately, together, or not at all are also available. Additionally, the outlines and data within each HB can be hidden or visualized.

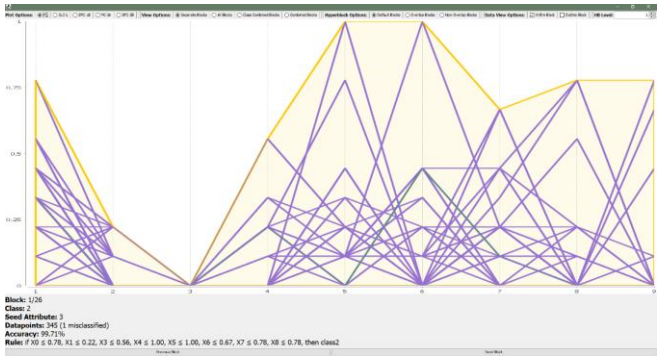


Fig. 19. Hyperblock window of DV2. First hyperblock of WBC data visualized.

B. Future features

In the future the parallel HB creation algorithm noted in Section 2 will be implemented. Scalability to larger datasets is important to the future of DV2 and other parallel and performance enhancing features will be added.

V. Conclusion and Future Work

Explainability is important for a domain expert to have confidence and trust in a machine learning model. This is why optimizing the number and generality of HBs is an important task. HBs, despite being intrinsically explainable, can still provide poor user confidence due to multiplicity of HBs and overfitting. In this paper, Visual Knowledge Discovery methodologies along with level 2 HBs were used to provide this optimization and increase the explainability, useability, and confidence of each HB.

With the multilayer approach, the ability to generalize data, creating a smaller number of HBs, has been shown. This is especially true in the case of overfit HBs containing only a single case. With both the WBC dataset and the MGT dataset, the number of HBs were reduced by 50% or greater. This increased the generality of each HB, created a smaller number of rules, and increased the user confidence in each classification. This shows that level 2 HBs work with both small and larger datasets to simplify and increase the explainability of those HBs. Next, parallelization will make this process more efficiency.

Similarly, by using GLC-L of level 2 abilities to reduce the visual occlusion and increase the explainability of each GLC-L visualization have been shown. These benefits can be further improved in the case of negative coefficients by creating simplified GLC-L where negative coefficients are visualized reversed. This was illustrated with MNIST data.

Additionally, interactive HB creation algorithms can be used to create interactive and explainable higher level HBs by combining GLC-L visualizations and HB creation. Unlike other HB creation methods, interactive HB creation allows domain experts direct access to the model discovery process. This gives domain experts the opportunity to inject deep domain knowledge into a model, enhancing it in ways that may not be possible otherwise. A user can also interactively improve accuracy of linear GLC-L models with 2-D HB properties.

Finally, instead of removing only HBs containing an individual case, it would be better removing HBs under a user specified size. In the future it would be beneficial to further optimize our HBs to be more general while still preserving as much accuracy as possible.

VI. References

[1] Kovalerchuk, B., Visual Knowledge Discovery and Machine Learning, 1st ed., Springer International Publishing, 2018. ISBN: 9783319730400.

[2] Kovalerchuk, B., Dovhalets, D. (2017). Constructing Interactive Visual classification, clustering, and dimension reduction models for N-D Data. *Informatics*, *4*(3), 1–27.

[3] Recaido, C., Kovalerchuk, B., Interpretable Machine Learning for Self-Service High-Risk Decision-Making, in Proc. of the 26th International Conference on Information Visualisation (IV), IEEE, 2022.

[4] Huber L, Kovalerchuk B, Recaido C. Visual knowledge discovery with general line coordinates. In: Artificial Intelligence and Visualization: Advancing Visual Knowledge Discovery 2024, pp. 159-202, Springer, arXiv preprint arXiv:2305.18429. 2023

[5] GitHub: https://github.com/CWU-VKD-LAB, DV2.0.

[6] Wolberg,William, Mangasarian,Olvi, Street,Nick, and Street,W.. (1995). Breast Cancer Wisconsin (Diagnostic). UCI Machine Learning Repository. https://doi.org/10.24432/C5DW2B.

[7] Bock, R.. (2007). MAGIC Gamma Telescope. UCI Machine Learning Repository. https://doi.org/10.24432/C52C8B.

[8] 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. https://arxiv.org/pdf/2009.10221

[9] MNIST, GTDLBench. https://git-disl.github.io/GTDLBench/datasets/mnist_data

[10] Kovalerchuk B, Hayes D. Discovering interpretable machine learning models in parallel coordinates. In2021 25th International Conference Information Visualisation (IV) 2021 Jul 5 (pp. 181-188). IEEE.