Principal Components in General Line Coordinates for Visualization and Machine Learning

Boris Kovalerchuk
Central Washington University
USA
borisk@cwu.edu

Brent D. Fegley
Aptima Inc.
USA
bfegley@aptima.com

***Abstract* — Understanding and visualizing principal components without losing multi-dimensional information has been a long-standing challenge until now. Here, we describe a novel lossless visual technique that overcomes such difficulties, General Line Coordinates-Principal Components Analysis (GLC-PCA)—an approach that provides much more information about individual cases than other methods, with improved structural readability and comprehensibility for PCA. Experimental case studies and comparisons with other works demonstrate this effect. A software system called MAVERICK implements it.**

Keywords — Principal Components Analysis, lossless visualization, high dimensional data, general line coordinates, explainable machine learning, shifted paired coordinates.

I. INTRODUCTION

Principal component analysis (PCA) has been an efficient tool in many domains for years [3]. In machine learning (ML), PCA is a robust **dimensionality reduction** technique for building smaller models and visualizing n-dimensional (n-D) data. Nevertheless, PCA's output suffers from a well-known problem: interpretability [4, 18-21]. For instance, the meaning of spectroscopic features identified through PCA is often **unclear** [4]. The problem is broader in ML because complex ML models built on principal components can become **unexplainable** even if all input attributes are interpretable. Consequently, data visualization methods have become essential in explainability studies of ML models [9-14].

The linear combinations of the observed attributes that PCA reveals are called **principal components** (**PCs**), **loadings,** or **eigenvectors**. For the n-D space, the number of these linear functions is also *n*, and they are mathematically orthogonal. We will denote them as PC1, PC2, …, PC*n*. The values of these linear combinations PC*i*(**x**) for each n-D point $\boldsymbol{x} = (x_1, x_2, \dots, x_n)$ are known as **scores**.

Commonly, PC1 provides the largest contribution to the **total variance** of the given dataset, with the contribution of all other PCs following in descending order. Practitioners often concentrate their analyses on the first few PCs to a threshold variance (e.g., 95%), discarding the remaining PCs. Regardless, if dimensionality reduction is the goal, including more PCs will capture more variance. For supervised ML, the value of unsupervised selection of the "best" PCs using **covariance** is limited because it ignores the class labels of the cases.

In general, PC visualizations appear in the scientific literature to (1) extract meaning and insight from their axes, (2) discover patterns in the PC space, (3) analyze trends in the observation space, and (4) analyze and order data objects there too.

Scatterplots are a common data visualization technique for representing PC pairs or triples in orthogonal coordinates. However, their use to visualize only the first two or three PCs (excluding all others) is shortsighted and not limited to PCA. For example, **multidimensional projections** (MDP) map n-D data into a 2-D space but do not show how the original attributes **influence** the projection's layout. A user can see which points are similar, but the source of that similarity is hidden [7,8]. Uncovering it can lead to data understanding and insight.

Interactive PCA, iPCA, provides a more sophisticated approach to PCA visualization [5] (Figure 1). Within iPCA's panels (A through F), a user can view (A) a scatter plot of two PC scores, (B) each original n-D point in terms of PCs in parallel coordinates, (C) each original n-D point in parallel coordinates, and (D) correlation coefficients and a scatter plot matrix for pairs of original attributes. A user can adjust the contribution of original attributes to the PC in E using multiple controls in F.

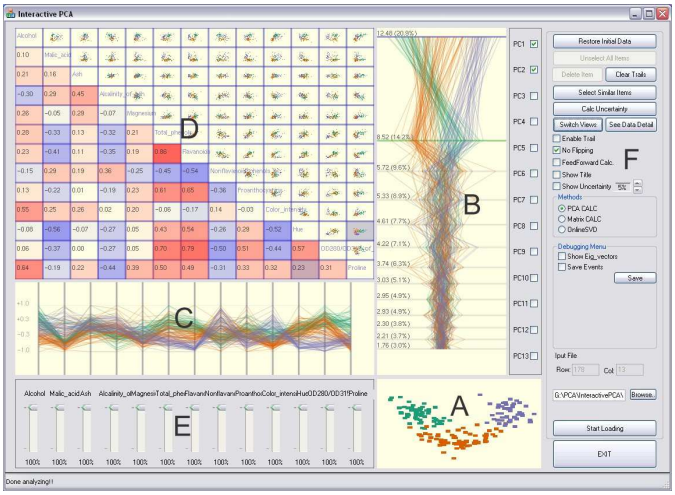

Figure 1. Example of the iPCA system applied to an E. coli dataset [5].

Despite its coordinated views, iPCA neglects to link PCs and original attributes to promote their meaningful explanation. For example, no visible connection exists between views in panels B and C, and A only shows PC pairs. iPCA does not **explain** the contributions of PCs on the individual original n-D points. Instead, it explores the opposite: the impact of adjusting the contribution of original attributes on the PCs.

Another PCA visualization is presented in [6] for demographic data. It uses **colored data bars in the data tables**, where the directions of the bars indicate the positive or negative value of the coefficients of the linear functions. Each **column** represents the contribution of original demographic attributes to the respective principal component PC*i*. Similarly, each **row** represents the contribution of principal components PC*i* to the actual demographic attributes. This visualization's advantage is its simplicity, requiring little time for the user to learn the visualization metaphor. A user can derive from this visualization the dominance of the first three PCs and the original attributes contributing most to the PCs generally but not for individual cases. Tabular visualization with data bars in tables for input data and PCs is limited by the number of cases that can fit on the screen. The topic of outliers is discussed in [5,6] by exploring links between original data and PCs.

In [22], the typical PC visualization limit of visualizing only two PCs was resolved by visualizing five PCs simultaneously. Unfortunately, the representation is lossy (it loses information) because the five PCs are condensed to a single 2-D point using the **RadViz** spring approach. In contrast, iPCA uses parallel coordinates to represent all n-D data without losing n-D information but at the cost of many overlapping (occluded) polylines (Figure 1).

A **heatmap** provides another way to highlight the key attributes that make closely projected points similar. For example, attributes can be ranked by increasing variance over each point-neighborhood with a visual encoding of areas to show the least-varying dimensions over each neighborhood [8]. Its deficiency lies in showing a single dominant attribute for each neighborhood. Often other attributes are comparably important. Moreover, it shows neighborhoods, not **individual points**.

One could implement the **heatmap** method in at least two ways for PCA. In the first, following [7], the PC with the highest variance could be used to color the vicinity of a point **p** to a single color. In the second, following [8], the (causal) source of the dominant feature (perhaps the original attribute with the highest contribution to PC*k* for point **p**) could be used to color the vicinity of **p**.

The first way shows the distribution of dominant variance among PCs in the space (PCi, PCj)—the distribution of the most critical PCs. Although it explains why two points are close, it does not explain the meaning of those PCs to the original attributes. The second way shows the distribution of dominant original attributes to PC*k*, explaining PC*k* in terms of the original attributes.

As we seek to explain PCs in terms of their original attributes, our approach herein lies closer to [8] but uses **General Line Coordinates-Principal Components Analysis (GLC-PCA)** [9] as the basis for overcoming the preceding deficiencies and limitations. GLC-PCA visualizes all contributing attributes and shows the level of their contribution. This is a crucial feature because the similarity of two points in the projection space of two PCs does not imply their similarity in the entire n-D PC space. Giving such points the same color provides no insight into their n-D similarity. Additionally, unlike iPCA which focuses exclusively on PCA, GLC-PCA aligns explicitly with **supervised machine learning**.

II. METHODOLOGY FOR PCA INTERPRETATION AND VISUAL KNOWLEDGE DISCOVERY

The interpretability of principal components (PCs) for **heterogeneous** attributes remains an unresolved decades-old challenge for PCA in all domains. For instance, what is the meaning in the demography domain of the weighted sum of, say, GDP and female life expectancy, like $0.1 * \text{GDP} + 0.3 *$ Lifeexp_F? We might establish meaning implicitly by assuming that **total variance** is **meaningful** to demography. However, such an assumption is difficult, if not impossible, to justify for **heterogeneous attributes** like GDP and life expectancy. Conversely, the correlation between GDP and life expectancy can be understood and explained.

In general, each PC is a **compressed representation** of some correlated attributes; it is a new artificial attribute without an express meaning in the application domain. However, some PCs are **interpretable**. For instance, three correlated life expectancy attributes can be converted to a new artificial attribute (their linear combination). In contrast, the meaning of an artificial attribute with literacy added to it is less evident in the demography domain, even if literacy is highly correlated with the different life expectancies in the dataset. Thus, a **deep explanation** of PCs (understanding what they represent) requires more information, deeper analysis, data visualization, and an objective: elucidating the meaning of links between attributes and PCs discovered mathematically.

We could conduct such a deeper analysis as follows. Assume we created an artificial attribute A (principal component) such that $A(\mathbf{x}) = 0.4 * \text{Literacy}(\mathbf{x}) + 0.45 * \text{Lifeexp}(\mathbf{x})$ where A for person $\mathbf{p}$ is $A(\mathbf{p}) = 7$. We can now translate this equation into meaningful text: "The literacy of **x** is between 65% and 75% and the life expectancy of **x** is between 70 and 80 years old" by methods such as those presented in [10,16].

In this way, we produce an **equivalent statement** with a clear meaning in the demography domain. Moreover, the statement can be meaningfully **verified** or falsified in this domain, unlike the abstract $A(\mathbf{x}) = 7$. This approach provides PCs a deeper **meaning** than merely visualizing how coefficients in a table relate to one another.

The visualization of the first PC1 values for different Asian countries $\mathbf{q}$ on the map in [6] is like the example $A(\mathbf{x}) = 7$. Although coloring shows the difference in $\text{PC}_1(\mathbf{q})$ values in these countries, from deep blue to intense yellow, it is not informative about their meaning. If the colored circles could be associated with a statement like the above for literacy and life expectancy, demography experts are more likely to understand and communicate their findings publicly.

The lack of deep explanations for PCA has two primary sources: (1) lack of PCA interpretability studies and (2) limited visualization methods.

The PCA visualization methods, which have existed for years, include *projection scatter plots*, *biplots, heatmaps* for correlation tables, *data bars in tables (table lens)*, *parallel coordinates*, and *spring* visualization (RadViz). Of these methods, only the biplot (introduced in 1971, [17]) can be called a **PCA-specific visualization** method integrating two aspects of PCA: scores and loadings. Until now, only parallel and radial coordinates could visualize high-dimensional data **losslessly** beyond coloring tables with heatmaps and data bars.

Today we have other data visualization methods designed for **visual knowledge discovery** through **lossless** high-dimensional data visualization. **General Line Coordinates (GLC)** and its multiple subsets, like GLC-Linear (GLC-L) and Shifted Paired Coordinates (SPC), are among those methods [9, 10, 15]. We use GLC-L and SPC in our proposed GLC-PCA system, the primary goal of which is to help users **interpret** individual n-D points in PCs—to help them see the **contribution** of each original attribute to the respective PCs relative to the contribution of other attributes in the visual form.

As presented in this paper, our implementation of GLC-PCA in the MAVERICK software system is a first step toward a new PCA-specific computational and visualization method like the biplot of decades before.

III. METHODS

A. GLC-PCA and MAVERICK visualization system

Despite PCA's popularity as an n-D data visualization and dimension reduction (DR) method, it suffers from two significant deficiencies already outlined. Recall that these are (1) difficulties in **explaining** PCs and (2) **loss** of information when only the first two PCs are used to visualize n-D data. The proposed GLC-PCA algorithm provides a solution for both challenges. It combines a **visual explanation** of each PC using the GLC-Linear (GLC-L) visualization algorithm [9], which visualizes n-D linear functions in 2-D and Shifted Paired Coordinates [9], which visualizes all *n* PCs in 2-D without loss of information. Figure 2 illustrates the results of GLC-PCA algorithm application.

Here, the *explained variance (EV) ratio* value for each principal component w_i is shown next to its name w_i (as axis annotations). All PCs are ordered according to their EV in pairs, left to right, with odd-numbered PCs on the x-axis and even-numbered ones on the y-axis. The black polyline (directed graph) shows all eight PCs of 8-D data losslessly, as a sequence of pairs in SPC:

$$(w_1, w_2) \rightarrow (w_3, w_4) \rightarrow (w_5, w_6) \rightarrow (w_7, w_8).$$

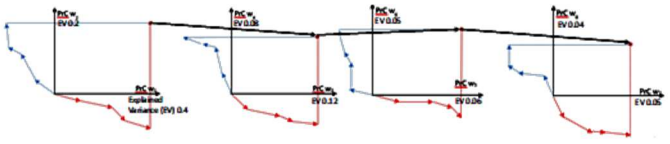

Figure 2. Illustration in which all PCA principal components $w_1 - w_8$ are visualized losslessly in SPC and GLC-L for an 8-D point $\boldsymbol{x} = (x_1, x_2, \dots, x_8)$ with all contributions of original attributes depicted for a single observation.

The red and blue polylines show, in GLC-L coordinates, how each PC, w_i, is formed as a linear combination of original attributes $x_1 - x_8$ of 8-D point **x**. The length of each red and blue segment shows the value of the respective original attribute x_j.

The angle q_{ij} of the segment to the respective coordinate represents the contribution of the original attribute x_i to the principal component w_i. The original attributes x_j with smaller angles q_{ij} contribute most to PC w_i. Figure 3 shows the details of this visualization for the first two PCs, where **contribution lines** are in blue and red.

B. Base GLC-PCA algorithm

Each PC y_i is a linear function of original attributes $x_1 - x_n$ of an n-D point $\boldsymbol{x} = (x_1, x_2, \dots, x_n)$,

$$y_i = a_{i1}x_1 + a_{i2}x_2 + \cdots + a_{in}x_n, i = 1, 2, \dots, n.$$

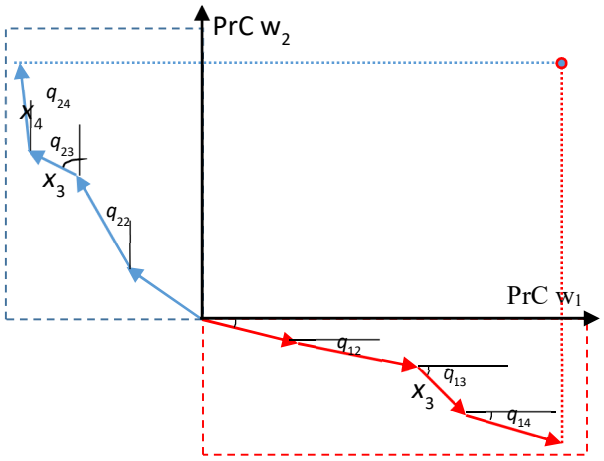

*Figure 3. First two principal components w_1 and w_2 of 8-D point **x** in GLC-PCA.*

A simplified GLC-PCA version can show each GLC-L plot with a single normalized n-D point with all $x_i = 1$ but different angles q_{ij}, instead of showing w_i for each n-D point **x**. This simplification allows the display of each x_i contribution to each PC while decreasing occlusion in the GLC-L plots. Another simplification is to show only the most important PC that covers, say, 80% of the total variance. The MAVERICK software allows us to distinguish train/test data within a single plot using a different point coloring. Thus, one can explore the differences between contribution lines for training and test data.

C. Exploration of similar pairs of cases: SIM Algorithm

The **SIM algorithm** allows us to gain additional benefits from GLC-PCA. First, find all pairs of cases from opposing classes next to each other in the first pair of PCs. Second, check whether these cases lie far away or close in the second pair of PCs. If they are far away, flag the cases with a 1, otherwise with 0. Third, count the number of pairs where flag = 1 and evaluate their importance. That is, if the count is relatively large, we have an indication of the **importance** of the second PC pair as a

discriminator of opposing classes. This importance can be **global** or **local**.

Steps of GLC-PCA algorithm

Step 1: Producing **normalized coefficients** k_{ij} instead of a_{ij} computed by the PCA algorithm $k_{ij} = \frac{a_{ij}}{m_{ai}}$, where $m_{ai} = \max_{j=1,2,\dots n}(|a_{ij}|)$. For instance, in 4-D with $a_{11} = 2, a_{12} = 3.5, a_{13} = 4.1,$ and $a_{14} = 5$, we have $m_{a1} = 5$ resulting in $k_{11} = \frac{2}{5}, k_{12} = \frac{3.5}{5}, k_{13} = \frac{4.1}{5},$ and $k_{14} = \frac{5}{5}$ which allows computing **normalized** PCs $w_1 = k_{i1}x_1 + k_{i2}x_2 + \cdots + k_{in}x_n, i = 1, 2, \dots, n.$

Step 2: Producing **SPC visualization** [9] of n-D point $\mathbf{w} = (w_1, w_2, \dots, w_n)$ as a black polyline in Figure 2.

Step 3: Getting **angles** $q_i = arccos(k_i), i = 1, 2, \dots, n.$

Step 4: Producing **GLC-L visualizations** [9] for the consecutive pairs of PCs: $(w_1, w_2), (w_3, w_4), \dots, (w_{n-1}, w_n)$. Each PC from the pair is visualized in a separate plot. Figure 2 shows these plots as the dotted blue and red rectangles. The red one is for the PC represented by a horizontal axis, like w_1, and the blue one is for the PC represented by the vertical axis, like w_2.

Step 5: Repeating steps 1-4 for other n-D points and plotting them along the point **x**.

Global importance means that the second PC pair is generally helpful for distinguishing classes in the dataset. Local importance indicates that the second PC pair has discriminatory utility for a subset of the data. Otherwise, we have evidence that the second PC pair will likely be low contributing in a discrimination model.

In the SIM algorithm, comparing PC pairs continues until a PC pair is eliminated (systematically or by user control) due to its low discriminatory utility. At this point, only the visualization of the non-eliminated PC pairs remains. Using GLC-PCA, the user can see and analyze the cases these PCs represent to gain understanding and insight into the data and underlying generative processes at play.

We present the SIM algorithm more formally in what follows. Consider an n-D point $\boldsymbol{a}_1$ and its PC representation $\boldsymbol{w}_1 = (w_{11}, w_{12})$ in the first two PCs and $\boldsymbol{h}_1 = (h_{11}, h_{12})$ in the second two PCs. Similarly, we define an n-D point $\boldsymbol{a}_2$ with its PC representation $\boldsymbol{w}_2 = (w_{21}, w_{22})$ in the first two PCs and $\boldsymbol{h}_2 = (h_{21}, h_{22})$ in the second two PCs.

Next, we compute the distance (here Euclidean) between $\boldsymbol{w}_1$ and $\boldsymbol{w}_2$: $\|\boldsymbol{w}_1, \boldsymbol{w}_2\| = \sqrt{(w_{11} - w_{21})^2 + (w_{12} - w_{22})^2}$ and establish a similarity threshold T (e.g., 0.1) representing the range of the PC's values from any given point. Application of the range requires normalized PCs, which for simplicity, we assume has occurred. Then, we compute $\boldsymbol{h}_1$ and $\boldsymbol{h}_2$: $\|\boldsymbol{h}_1, \boldsymbol{h}_2\| = \sqrt{(h_{11} - h_{21})^2 + (h_{12} - h_{22})^2}$ and test both $\|\boldsymbol{w}_1, \boldsymbol{w}_2\|$ and $\|\boldsymbol{h}_1, \boldsymbol{h}_2\|$ against the threshold: If $\|\boldsymbol{w}_1, \boldsymbol{w}_2\| \leq T$ and $\|\boldsymbol{h}_1, \boldsymbol{h}_2\| > T$ then $w2h = 1,$ else $w2h = 0.$

Similarly, we compute $\|\boldsymbol{w}_1, \boldsymbol{w}_2\| \leq T$ and $\|\boldsymbol{h}_1, \boldsymbol{h}_2\| > T$ then $h2w = 1,$ else $h2w = 0.$

Next, we sum up *w2h* and *h2w* for all pairs, $W2H = \sum_{i=1:n} w2h_i$ and $H2W = \sum_{i=1:n} h2w_i$.

W2H represents the proportion of pairwise distances that are *less than or equal to T* in (PC1, PC2) and *greater than T* in (PC3, PC4), and *H2W* represents the proportion of pairwise distances that are *greater than T* in (PC1, PC2) and *less than or equal to T* in (PC3, PC4). Scalar Euclidean distance is only one distance measure option; we could have chosen a **two-valued measure (TVM)** for computing *w2h* and *h2w*:

$$\|\boldsymbol{w}_1, \boldsymbol{w}_2\| = (|(w_{11} - w_{21}|, |w_{12} - w_{22}|)$$
$$\|\boldsymbol{h}_1, \boldsymbol{h}_2\| = (|(h_{11} - h_{21}|, |h_{12} - h_{22}|)$$

If $|\boldsymbol{w}_{11} - \boldsymbol{w}_{21}| \leq T$ and $|\boldsymbol{w}_{12} - \boldsymbol{w}_{22}| \leq T$ and $|\boldsymbol{h}_{11} - \boldsymbol{h}_{12}| > T$ and $|\boldsymbol{h}_{12} - \boldsymbol{h}_{22}| >$ T then $w2h = 1,$ else $w2h = 0.$

If $|\boldsymbol{h}_{11} - \boldsymbol{h}_{21}| \leq T$ and $|\boldsymbol{h}_{12} - \boldsymbol{h}_{22}| \leq T$ and $|\boldsymbol{w}_{11} - \boldsymbol{w}_{12}| > T$ and $|\boldsymbol{w}_{12} - \boldsymbol{w}_{22}| >$ T then $h2w = 1,$ else $h2w = 0.$

The advantage of TVM over Euclidean distance is its independence from the different scaling of each PC and visual appeal (square vs. circle).

D. k-NN generalization with GLC-PCA: Square Algorithm

In this section, we discuss the benefits of similarity analysis with *W2H* and *H2W* and the generalization of the *k*-nearest neighbors (k-NN) machine learning algorithm, which we denote as the **Square algorithm**.

Practitioners typically use *k*-NN to compute three to five nearest neighbors of a given case **a** and then classify **a** to the same class as most of those *k*-neighbors. *k*-NN commonly uses Euclidean distance, which is very **sensitive** to the scales of the underlying attributes. Consequently, the set of nearest neighbors for the same new case can vary significantly. Many other **scalar distances** have a similar scaling sensitivity issue.

If all attributes and PCs are scaled equally, their contribution to Euclidean distance and *k*-NN will be equal too. This diminishes the contribution of important PCs and increases the noise from the least important PCs to the classifier. Using a **non-scalar similarity measure**, like TVM defined above, mitigates this challenge.

Although experts often use *k*-NN to classify new cases, "nearness" is not always reducible to a scalar metric (such as Euclidean distance). Instead, humans use an informal **multi-criteria decision analysis** across attribute subsets—e.g., considering one subset of attributes, PCs, and nearest neighbors and then making another subset out of the neighbors. This dynamic process establishes a final combination of attributes,

PCs, and nearest neighbors. The GLC-PCA method allows domain experts to conduct such complex analyses in the visualization.

We can make nearest neighbors more relevant by modifying *k*-NN to treat as neighbors all points within a $T \times T$ square relative to another point, e.g., $10\% \times 10\%$ of the range around that point. Some nearest neighbors in the typical formulation of *k*-NN can be much further than the $T \times T$ square. Our modification provides an analytical advantage (explained shortly).

The general steps of the **Square algorithm** are as follows: (1) find a square S centered in the given case to be classified, (2) select all cases within S (not *k* nearest neighbors), and (3) identify the dominant class by majority vote for all cases in S. The **basic square algorithm** first checks the informativeness of squares in a pair of PCs relative to another pair of PCs (recall the SIM algorithm above). If the outcome is positive, use the informative squares to vote in modified *k*-NN for a new case.

What is the benefit of the Square algorithm relative to the standard *k*-NN algorithm? If both *W2H* and *H2W* are large, all cases in the square vote (unlike k-NN where *k* is often fixed to three to five) ensuring a robust result. If only *W2H* or *H2W* is large, the Square algorithm avoids the noise from the second PC pair. If both *W2H* and *H2W* are small, the decision refusal naturally increases the result's robustness.

We can use the basic square algorithm to predict the effectiveness of similarity analysis and avoid otherwise unnecessary computation. For example, by considering only squares with training cases, we localize global *W2H* and *H2W*. If the analysis of local values confirms the importance of the squares, we have evidence of the Square algorithm's broader applicability. This approach can work in the original attributes and PCs and expands to any number of PC pairs.

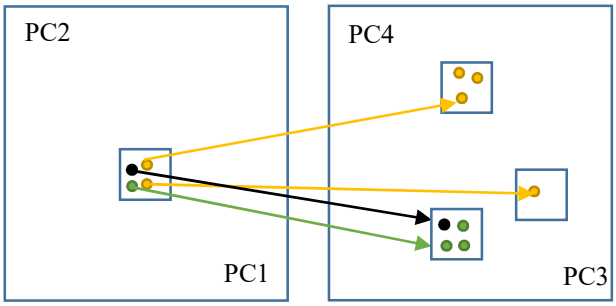

Figure 4. Illustration of the Square algorithm, a modification of the k-NN algorithm.

Figure 4 illustrates the Square algorithm. Here, we wish to classify a new case (colored black). In (PC1, PC2), the vote 2:1 (in the only square) is for the yellow class. In (PC3, PC4), the vote 3:0 (lowest square, where the unclassified case resides) is for the green class. Because green is more robust, it becomes the class to which the new case is assigned. An **interactive visual approach** that allows users to observe squares containing cases in (PC3, PC4) related to a square in (PC1, PC2) promotes critical thinking and individual decision-making instead of blind computation of *k* nearest neighbors. Running a 10-fold cross-validation to check solution accuracy (such as by **adjusting the size** of the squares or prohibiting the use of some squares) is a natural expansion of the algorithm. Generally, visualization in GLC-PCA with squares allows a user to observe visual patterns, find and mark misclassified cases, and trace them in different pairs of PCs.

E. Detecting and explaining outliers

Identifying outliers in a dataset is challenging, even with an exact distance criterion, especially without data visualization. For example, a case can be an outlier in some or all PC pairs. In lossless GLC-PCA, a user can easily see the **outliers** in each PC pair or find them interactively without formally defining them. Interacting with points also allows a user to assess the strength of an outlier in all the PCs. Crucially, GLC-PCA is not constrained to the PCs but enables the analysis of outliers in the original attributes. Thus, a user can meaningfully trace the **cause** of the outliers.

F. Generation of linguistic summary

Automatically generated linguistic (**natural language**) summaries of visible patterns in the GLC-PCA plot offer an opportunity for knowledge discovery beyond interactivity. Such summaries may lighten cognitive workloads by drawing users' attention to visual features (such as those within PC pairs or in the polyline patterns that connect those pairs but, more importantly, the contributions of original attributes to respective PCs). For instance, a linguistic summary might say that: (1) cases of class 1 concentrate on the left, (2) cases of both classes overlap in the middle, and (3) the significant contributing attribute to middle cases is attribute x_7. Linguistic summaries provide a mechanism to aid user memory and communicate discovered patterns in both a visual and linguistic form with GLC-PCA. They are a form of self-service that avoids otherwise challenging mathematical descriptions.

IV. CASE STUDIES

A. Variances

The main characteristic of a dataset that PCs capture is the variance split between PCs. Table 1 presents those variances for three datasets [1,2] centered and scaled. Only the first four PCs are depicted. (The iris data has only four PCs).

A scatterplot typically represents only two PCs, meaning that users analyze only two PCs visually in practice. Additionally, a typical **dimension reduction** (DR) practice retains only the first few PCs that cover the most explained variance. Figure 5 through 10 show why both practices need to be revised.

Table 1. Explained Variance by principal components (PCs).

PC	Iris	Wisconsin breast cancer (WBC)	California housing (CH)
1	0.7277	0.6081	0.4347
2	0.2303	0.0997	0.2136
3	0.0368	0.0762	0.1885
4	0.0052	0.0547	0.1011
Sum	**1.0000**	**0.8387**	**0.9379**

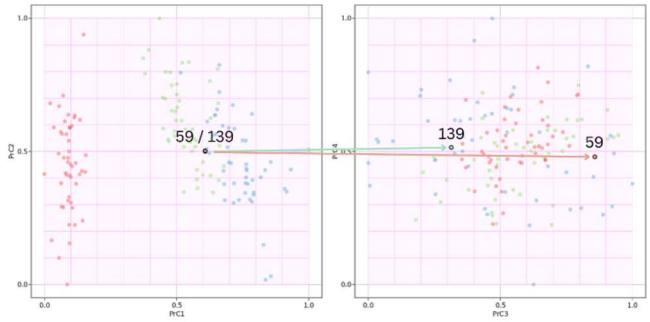

Figure 5. Two iris cases (rows 59 and 139 in the original dataset) of opposite classes indistinguishable in (PC1, PC2) but well-separated in (PC3, PC4) based on scaled and centered input.

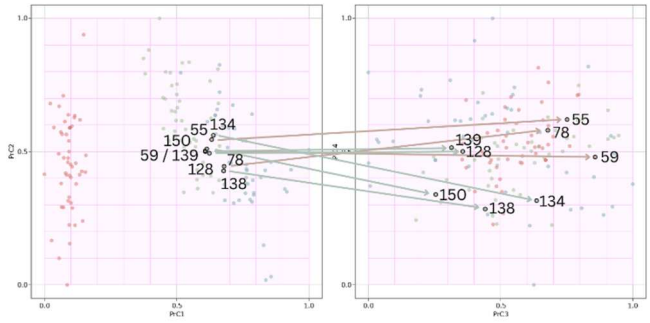

Figure 6. Top five pairs of Iris cases of opposite classes that are very close in (PC1, PC2) but well-separated in (PC3, PC4).

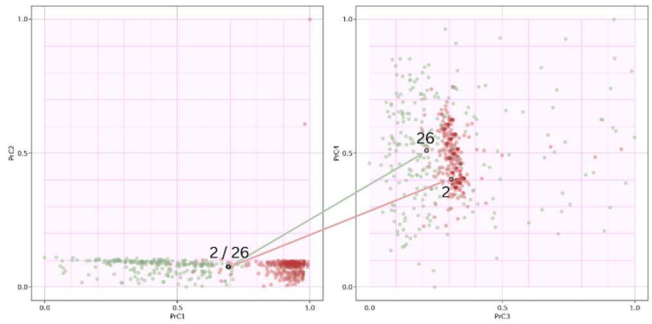

Figure 7. Two WBC cases (rows 2 and 26 in the original dataset) indistinguishable in (PC1, PC2) but well-separated in (PC3, PC4) based on scaled and centered input.

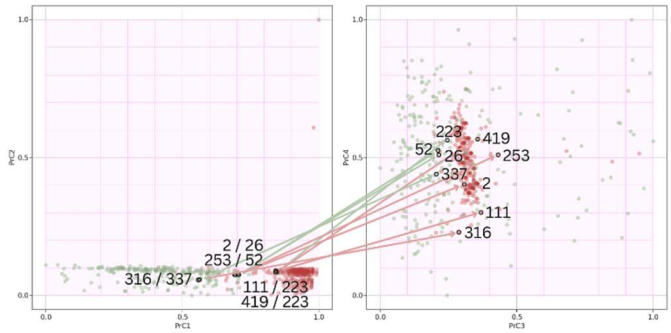

Figure 8. Top five WBC cases of opposite classes that are very close in (PC1, PC2) but well-separated in (PC3, PC4).

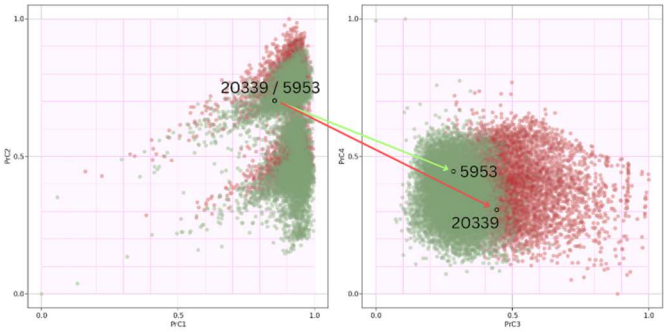

Figure 9. Two California housing cases (rows 5953 and 20339 in the original dataset) indistinguishable in (PC1, PC2) but well-separated in (PC3, PC4) based on centered and scaled input.

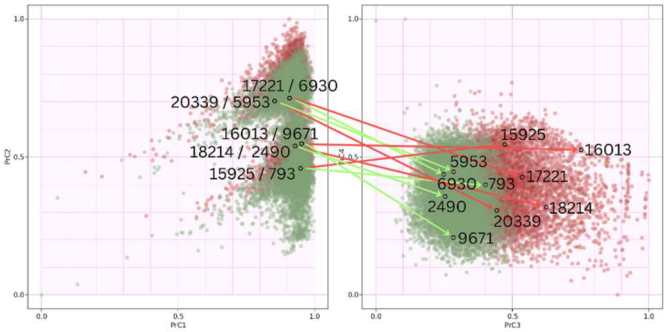

Figure 10. Top five California housing cases of opposite classes that are very close in (PC1, PC2) but well-separated in (PC3, PC4).

In these examples, the second PC pair allows us to separate pairs of cases from different classes that are otherwise indistinguishable in the first PC pair. Computing *W2H* and *H2W* values allows us to capture the situations presented in these figures and pick up the pairs of cases with the top five *W2H* and *H2W* values. The GLC-PCA visualization method allows us to see this difference and classify cases interactively.

The values *W2H* and *H2W* can be small but useful, even for low-valued cases. Although machine learning (ML) algorithms can classify most cases, the cases requiring individual attention can be handled in the proposed visually interactive way. For traditional ML algorithms, such exceptional cases typically cause **overfitting**, one of ML's significant problems. For example, decision trees whose branches have only a few cases in the terminal node are difficult to justify statistically. Decision tree pruning is not often satisfactory because shortening branches decreases the tree's classification accuracy. An **interactive approach** allows a domain expert to observe such unique n-D cases in detail in the lossless GLC-PCA and make individual decisions about difficult cases for formal algorithms.

B. GLC-PCA in the MAVERICK system

Below we demonstrate how GLC-PCA, implemented within the MAVERICK software system, can be used to analyze multidimensional data. We start with the Wisconsin Breast Cancer (WBC) data, which contain 683 9-D cases of benign and malignant classes. Figure 11 shows all WBC data in these five

pairs of PCs losslessly, where the contribution lines in five plots are quite different and more distinct than the scatterplots. Because SPC requires an even number of coordinates, we repeat the ninth PC to produce the tenth (hence the correlated scatter in the farthest right PC pair). The contribution lines can be informative for users. Figure 12 zooms into the left plot of Figure 11.

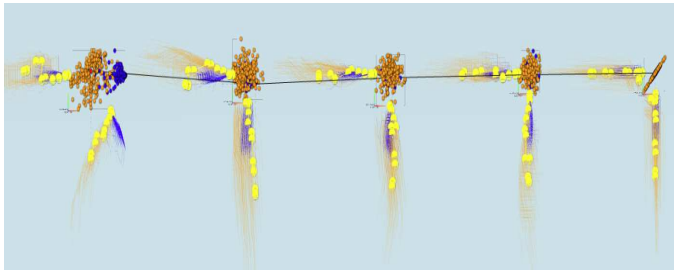

Figure 11. Overview of WBC data in GLC-PCA.

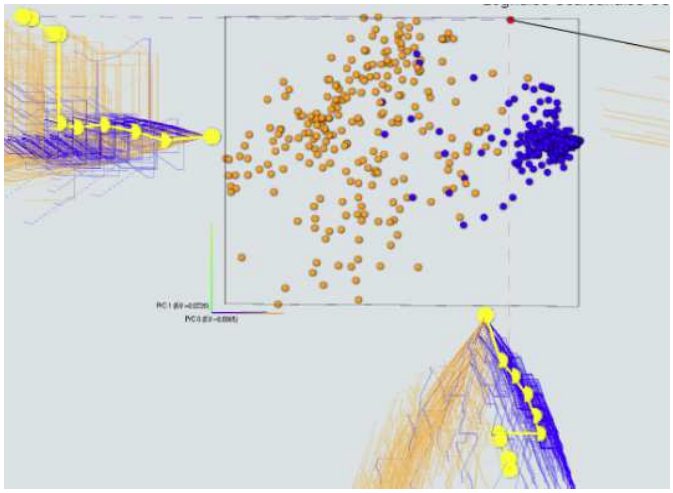

Figure 12. An outlier point (marked red) on the first PC pair.

Here we see that the contribution lines for the blue class are on the right and on the left for the brown class. The blue contribution lines are much shorter for the vertical coordinate than the brown lines. Figure 12 also shows an outlier point (red) and its contribution lines for PC1 at the bottom and PC2 on the left. It belongs to the blue class. GLC-PCA allows us to see the most important original contributing attributes easily. The bottom contribution line (highlighted in yellow) has one horizontal segment, which creates the most significant shift of the point. This is attribute x_6, which we can identify quickly by counting the number of yellow points from the beginning of the contribution line.

Similarly, for the second PC (on the left), attribute x_6 is also the most important in having the longest vertical segment. Thus, the user receives an immediate benefit from awareness of attribute contribution.

A user can analyze this case in PC pairs other than (PC1, PC2) similarly. Figure 13 shows it in (PC4, PC5), where it is not an outlier but in the middle of the scatterplot. The attributes contributing significantly to this PC pair are x_4 for PC4 and x_9 for PC5.

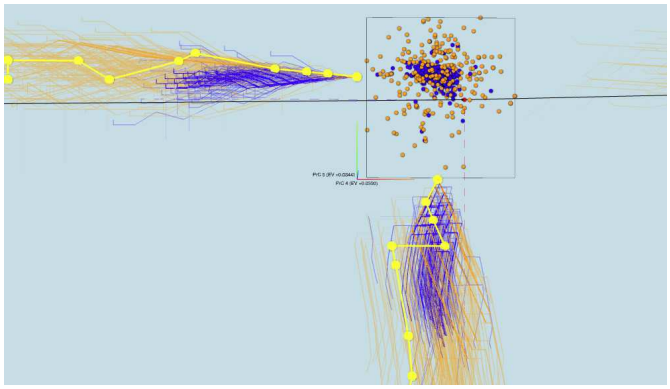

Figure 13. Selected point in (PC4, PC5).

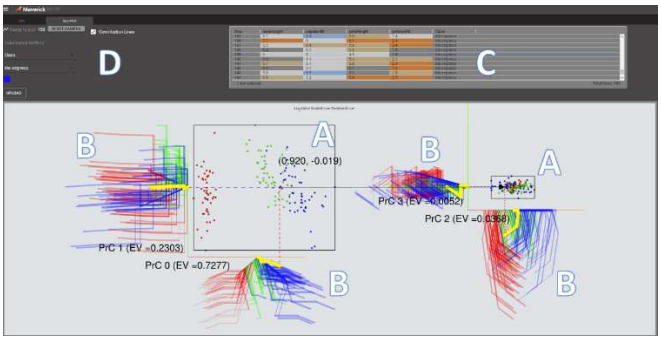

Figure 14. Iris data in GLC-PCA within the MAVERICK system.

Figure 14-16 also show Iris and Californian housing data [1, 2] in GLC-PCA within MAVERICK, where we may conduct a similar analysis. A screenshot of MAVERICK's GLC-PCA user interface appears in Figure 14, with a few features of interest labeled A through D. With this interface, a user can view, simultaneously: (A) a lossless set of *scatterplots* for all consecutive pairs of PC scores visualized using SPC, and (B) each PC in terms of all *n original attributes* visualized using GLC-L. A user can select and modify data in C and exercise data visualization controls in D. Figure 15 shows a view of original n-D points in (adjustable) parallel coordinates. Still, we can view PCs here, also. To assess GLC-PCA's effectiveness as a tool for knowledge discovery, we conducted a usability study asking participants to discover patterns like those mentioned above within the MAVERICK GLC-PCA interface.

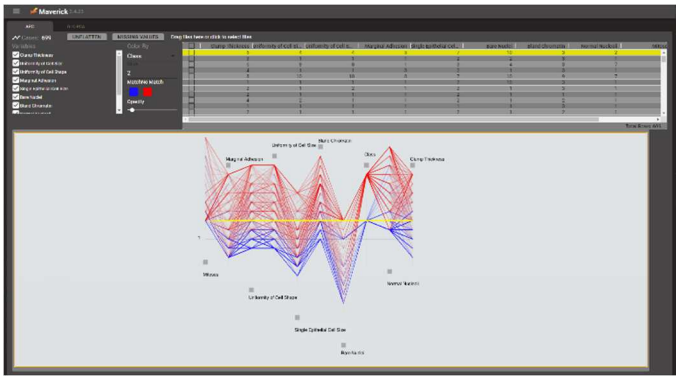

Figure 15. WBC data (original attributes) with class coloring in parallel coordinates within the MAVERICK system.

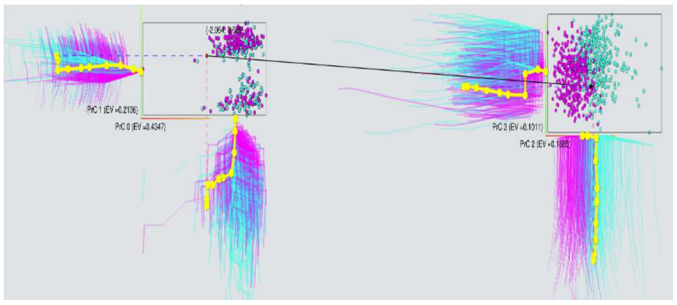

Figure 16. First four PCs of the California housing data in GLC-PCA within the MAVERICK system.

The study had four participants, all computer science undergraduate students. Before the study, participants had not heard about or seen the GLC-PCA concept or MAVERICK. All but one of the participants had taken a machine learning class with one lecture on PCA. We asked each participant 26 questions. A pilot question followed their receipt of a one-paragraph description of a GLC-PCA plot. The next questions regarded the participants' understanding of details within other plots provided. The participants were allowed to ask clarifying questions, and all did. On average, each participant took 31 minutes to answer the questions and be correct 91.4% of the time.

V. Conclusion

This paper presented a new approach to knowledge discovery, GLC-PCA, that addresses long-standing challenges in visualizing and understanding principal components (PCs): loss of n-D information in the visualization space and loss of meaning, given the typical disconnect between PCs and their original attributes.

The **advantages** of GLC-PCA over [5,6,8,22] are in providing:

(1) **lossless visualization** of each n-D point and PC.
(2) **visible connection** between (i) scatter plots of all consecutive PC pairs and (ii) representations of each PC with all original n-D attributes.
(3) **explanation and meaning** to a PC by showing the contributions of the original n-D attributes to the PC.
(4) visualizations useful for **supervised machine learning**.

For (2)(i) above, [5] allows observing the scatter plot only for the first two PCs, [6] does not generate any of the scatter plots, and [22] provides lossy visualization of all PC. For (2)(ii), [6] only allows analyzing the PCs generally but not for individual cases, [5] does not visualize them, and [8] shows only a single dominant attribute for a neighborhood, not for an individual point.

GLC-PCA has been implemented in a software system called MAVERICK, a joint effort between Aptima, Inc. and Central Washington University. Experimental case studies and comparisons with others' work show that our proposed methods deliver significantly more information with higher structural readability and comprehensibility. Our future work will expand the GLC-PCA approach to sparse and non-linear robust (weighted) PCA, independent component analysis, and Linear Discriminant Analysis for a wider variety of data.

VI. Acknowledgments

This work was partly supported by U.S. DoD/MDA Contracts HQ0147-19-C-7057 and HQ0860-20-C-7136. The authors thank Spencer Lynch for his implementation contribution.

VII. References

[1] UCI ML repository https://archive.ics.uci.edu/ml/index.php.
[2] Housing data, http://lib.stat.cmu.edu/datasets/houses.zip
[3] Gewers FL, Ferreira GR, Arruda HF, Silva FN, Comin CH, Amancio DR, Costa LD. Principal component analysis: A natural approach to data exploration. ACM Computing Surveys (CSUR). 2021 May 24;54(4):1-34. https://doi.org/10.1145/3447755
[4] Beattie JR, Esmonde-White FW. Exploration of principal component analysis: deriving principal component analysis visually using spectra. Applied Spectroscopy. 2021 Apr;75(4):361-75.
[5] Jeong DH, Ziemkiewicz C, Ribarsky W, Chang R, Center CV. Understanding principal component analysis using a visual analytics tool. Charlotte visualization center, UNC Charlotte. 2009;19.
[6] Müller W, Nocke T, Schumann H. Enhancing the visualization process with principal component analysis to support the exploration of trends. In: Proceedings of the 2006 Asia-Pacific Symposium on Information Visualisation-Volume 60 2006 Jan 1 (pp. 121-130).
[7] da Silva RR, Rauber PE, Martins RM, Minghim R, Telea AC. Attribute-based Visual Explanation of Multidimensional Projections. InEuroVA@ EuroVis 2015 (pp. 31-35).
[8] Tian Z, Zhai X, van Driel D, van Steenpaal G, Espadoto M, Telea A. Using multiple attribute-based explanations of multidimensional projections to explore high-dimensional data. Computers & Graphics. 2021 Aug 1;98:93-104.
[9] Kovalerchuk B., Visual Knowledge Discovery and Machine Learning, 2018, Springer
[10] Kovalerchuk, B., Ahmad, M.A., Teredesai A., Survey of Explainable Machine Learning with Visual and Granular Methods beyond Quasi-explanations, In: Interpretable Artificial Intelligence: A Perspective of Granular Computing, 217-267, Springer, 2022.
[11] Chatzimparmpas A, Martins RM, Jusufi I, Kucher K, Rossi F, Kerren A. The state of the art in enhancing trust in machine learning models with the use of visualizations. In: Comp. Graph. Forum 2020,. 39(3), 713-756).
[12] Yuan J, Chen C, Yang W, Liu M, Xia J, Liu S. A survey of visual analytics techniques for machine learning. Computational Visual Media. 2021,3-6.
[13] Ahn Y, Lin YR. Fairsight: Visual analytics for fairness in decision making. IEEE trans. on vis. and comp. graphics. 2019,19;2 6(1):1086-95.
[14] Tzeng FY, Ma KL. Opening the black box-data driven visualization of neural networks. in Proc. IEEE Visualization, 2005, pp.383–390
[15] Kovalerchuk, B., Nazemi, K., Andonie, R., Datia, N., Bannissi E. (Eds), Integrating Artificial Intelligence and Visualization for Visual Knowledge Discovery, Springer, 2022
[16] Huber L. Kovalerchuk B., Recaido C. Visual Knowledge Discovery with General Line Coordinates, CWU Techincal report, 2023. Arxiv
[17] Gabriel, K.R. The biplot graphic display of matrices with application to principal component analysis. Biometrika. 58 (3): 453–467, 1971.
[18] Björklund M. Be careful with your principal components. Evolution. 2019 Oct 1;73(10):2151-8.
[19] Boudt K, d'Errico M, Luu HA, Pietrelli R. Interpretability of Composite Indicators Based on Principal Components. Journal of Probability and Statistics. 2022 Sep 29;2022.
[20] Chipman HA, Gu H. Interpretable dimension reduction. Journal of applied statistics. 2005 Nov 1;32(9):969-87.
[21] Enki DG, Trendafilov NT, Jolliffe IT. A clustering approach to interpretable principal components. Journal of Applied Statistics. 2013 Mar 1;40(3):583-99.
[22] Theisel, H.; and Kreuseler, M. (1998): An Enhanced Spring Model for Information Visualization. Computer Graphics Forum, vol. 17, no. 3, Proceedings Eurographics '98.