Supporting Analysis of Dimensionality Reduction Results with Contrastive Learning

Takanori Fujiwara, Oh-Hyun Kwon, and Kwan-Liu Ma

Abstract— Dimensionality reduction (DR) is frequently used for analyzing and visualizing high-dimensional data as it provides a good first glance of the data. However, to interpret the DR result for gaining useful insights from the data, it would take additional analysis effort such as identifying clusters and understanding their characteristics. While there are many automatic methods (e.g., density-based clustering methods) to identify clusters, effective methods for understanding a cluster's characteristics are still lacking. A cluster can be mostly characterized by its distribution of feature values. Reviewing the original feature values is not a straightforward task when the number of features is large. To address this challenge, we present a visual analytics method that effectively highlights the essential features of a cluster in a DR result. To extract the essential features, we introduce an enhanced usage of contrastive principal component analysis (cPCA). Our method, called ccPCA (contrasting clusters in PCA), can calculate each feature's relative contribution to the *contrast* between one cluster and other clusters. With ccPCA, we have created an interactive system including a scalable visualization of clusters' feature contributions. We demonstrate the effectiveness of our method and system with case studies using several publicly available datasets.

Index Terms—Dimensionality reduction, contrastive learning, principal component analysis, high-dimensional data, visual analytics

1 Introduction

High-dimensional data visualization is one of the major research topics in the visualization community [46, 47]. Various types of visualization methods (e.g., the parallel coordinates [33], scatterplot matrices [27], and star coordinates [38]) have been introduced to present high-dimensional information in a space [47] (typically 2D on a computer screen) that human viewers can perceive and interpret. Among these methods, dimensionality reduction (DR) methods are suitable to provide an overview of the relationships across the high-dimensional data points [47, 54, 61].

The strength of DR methods is their capability of uncovering the similarity between data points as spatial proximity [75]. In DR results, by referring to the "similarity ≈ proximity" [75] relationship, we can intuitively find useful patterns, such as clusters and outliers. Many fields of study, including biology [31], social science [68], and machine learning [58], require analyzing high-dimensional data and thus rely on DR methods.

According to the recent surveys [12, 54], analyzing a DR result involves the following tasks: (1) identifying clusters in the DR result, (2) understanding the characteristics of the clusters, and (3) comparing the clusters with predefined classes of data points [12, 54]. In the case that the DR result has interpretable axes, such as the dimensions generated by principal components analysis (PCA) [32, 37], understanding the characteristics of each axis and comparing the axis with the original dimensions (or features) are also included as part of the analysis tasks.

Among the aforementioned tasks, the main task sequence is first identifying clusters and then understanding their characteristics [12]. While many automatic methods (e.g., density-based clustering methods [8, 15, 22, 40]) have been introduced to identify clusters (the first task), methods to assist the second task have still not been well studied, especially in the case that the data has many features. Reviewing the original feature values is essential to understanding each cluster's characteristics. To support this task, many existing visual analytics systems [18, 42, 48, 56, 63] employ basic statistical plots, such as histograms and parallel coordinates, for inspecting each feature of the selected clusters. However, because these visualizations render all of the features' values, they are limited in handling a large number of features. In addition, even if we were able to show all the features, it could be very time-consuming to find the common patterns within each cluster or find the differences among the clusters by individually referring to the values for each of the many associated features.

To address these problems, we have developed an analysis method that highlights those essential features for understanding characteristics of each cluster in a DR result. For our method, we adopt contrastive learning [77], a new emerging analysis approach for high-dimensional data. Contrastive learning aims to discover "*patterns that are specific to, or enriched in, one dataset relative to another*" [4]. Among the contrastive learning methods, we specifically choose contrastive principal component analysis (cPCA) [4, 5, 25] and enhance it for visual analysis. Our usage of cPCA, which we call ccPCA (contrasting clusters in PCA), can measure each feature's relative contribution to each cluster's *contrast* to the others. By referring to these relative contributions, users can easily focus on the features they should review in detail. We describe the strengths of using ccPCA with both numerical formulas and concrete examples. In addition, because cPCA requires parameter tuning to obtain a useful result, we develop an automatic selection method that finds the best parameter value.

Moreover, we introduce a heatmap-based visualization showing all the features' contributions of each cluster. By employing hierarchical clustering and matrix reordering, our visualization helps the user find where clusters have similar features' contributions or how the features have similar contributions within or across clusters. Additionally, with these methods, we are able to provide a scalable visualization that can handle the case of analyzing many features (e.g., 100 features or more). We have built an interactive visual analytics system using ccPCA and a heatmap-based visualization. We demonstrate the effectiveness of our methods and system with case studies using several publicly available datasets.

2 Related Work

We survey the relevant works in (1) visualization for exploring DR results and (2) discriminant analysis and contrastive learning.

2.1 Visualization for Exploring DR Results

Various visualizations have been developed to assist analysis tasks for a DR result [20, 39, 43, 44, 46, 47]. Here, we focus on describing the works that supports the aforementioned main task sequence (i.e., identifying clusters and understanding clusters' characteristics). Stahnke et al. [63] developed visualizations to help understand multidimensional-scaling (MDS) [67] results. To support a feature comparison of clusters in the MDS result, their visualization allows the user to manually select clusters and then it depicts the selected clusters' density plots for each of the features. Similarly, for a cluster comparison in the DR results, other

• *Takanori Fujiwara, Oh-Hyun Kwon, and Kwan-Liu Ma are with University of California, Davis. E-mail: {tfujiwara, kw, klma}@ucdavis.edu*

works [18, 42, 48, 56] visualized statistical charts (e.g., bar charts and boxplots) of the features for each manually or automatically selected cluster. However, because the approaches in [18, 42, 48, 56, 63] depict the statistical chart for each feature, they are not scalable when there is a large number of features (e.g., 10 features). Broeksema et al. [14] took further steps to provide a summary of the DR results. They developed visualizations to help understand patterns that appeared in multiple correspondence analysis (MCA) [3], which is a similar DR method as PCA for categorical data. They visualized each data point's salient feature value extracted with MCA as a colored Voronoi cell around each projected point in the MCA result. This linking of the DR result and the salient features helps the user interpret the DR result. Similarly, Joia et al. [36] linked the DR result and the information of features into one plot. In addition to an automatic selection of clusters, they obtained representative features for each cluster by using PCA. Afterward, they visualized these features' names as a word cloud within each clustered region instead of showing the projected points. Turkay et al. [69] also used PCA to obtain the representative features in the MDS result.

Among the mentioned studies, the works by Joia et al. [36] and Turkay et al. [69] are most related to ours in terms of identifying the representative features for each cluster. To identify such features, both methods refer to each cluster's principal components (PCs) computed by PCA (and the correlation between the features and PCs). Even though they applied PCA within each cluster, the computed PCs might capture only the global tendency in the dataset. For example, all clusters may have similar or even the same PCs. Also, their methods cannot find features that highly contribute to the differentiation or *contrast* between one cluster and the others. It is important to provide features that make each cluster's characteristics unique.

2.2 Discriminant Analysis and Contrastive Learning

Discriminant analysis, including linear discriminant analysis (LDA) [34], quadratic discriminant analysis (QDA) [51], and mixture discriminant analysis (MDA) [29], is a supervised learning method used for classification and DR. Discriminant analysis methods use labeled data points as a learning set and construct a classifier to distinguish each class as much as possible [34]. For example, LDA finds new dimensions (or components) which provide good separations between each class. Note that while both PCA and LDA can be categorized as linear DR methods, PCA is an unsupervised method and finds dimensions which maximize the variance of the input data points.

As similar to PCA, we can obtain the contribution of each original dimension (or feature) to each component constructed by LDA. Therefore, for visual analytics, LDA has been utilized to inform the features which have an important role to distinguish clusters. For example, Wang et al. [72] developed linear discriminative star coordinates (LDSC). LDSC shows each feature's contribution to distinguishing a cluster from each other as a length of a corresponding axis of the star coordinates [38]. To obtain a better-clustered result, the user can use these axes as interfaces to discard the less contributed features or change the weight of the features used for clustering.

While discriminant analysis is used for discriminating the data points based on their classes, contrastive learning [77] focuses on finding patterns which contrast one dataset with another [4]. For example, contrastive PCA (cPCA) [4, 5, 25] is the extended version of PCA for contrastive learning. cPCA takes two different datasets (i.e., target and background), and then identifies the directions (or contrastive principal components) that have a higher variance in the target dataset when compared to the background dataset. Projection of the target dataset with these contrastive principal components provides the patterns which are uniquely found only in the target dataset. In addition to cPCA, several extended methods for contrastive learning have been developed (e.g., contrastive versions of latent Dirichlet allocation [77], hidden Markov models [77], regressions [25], multivariate singular spectrum analysis [19], and variational autoencoders [6]).

To the best of our knowledge, this paper is the first research using a contrastive learning method, specifically cPCA, for interactive visual analytics. We demonstrate the major advantages of using cPCA instead of PCA or LDA in Sect. 4.

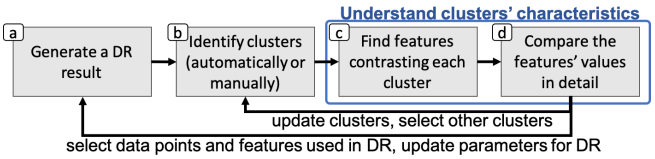


Fig. 1: The analysis workflow.

3 Workflow and an Analysis Example

We first define a workflow for analyzing high dimensional data using DR, and then provide an analysis example to motivate our work.

3.1 Analysis Workflow

Fig. 1 shows an analysis workflow using our method. It starts from (a) applying a DR method (e.g., MDS, PCA, or t-SNE [71]) on high-dimensional data. Then, the task is (b) to identify clusters in the DR result by applying a clustering method (e.g., k-means [28], DBSCAN [22], or spectral clustering [53]) or selecting clusters manually. Afterward, the task is to understand the clusters' characteristics. This task has two steps. The first step is (c) finding features (or dimensions) which have a high contribution to contrasting each cluster with the others. For this step, we utilize cPCA [4, 5, 25], as described in Sect. 4. The second step is (d) reviewing the detailed differences of values of the highly contributed features between each corresponding cluster and the other data points. We use existing methods for DR and clustering while we introduce new methods for the last two steps. With the last two steps, we can obtain an understanding of which and how features contribute to the uniqueness of each cluster. After understanding the selected clusters' characteristics, as indicated with the arrows from (d) to (a) and (b), the user can update the DR result or clusters by selecting a subset of the data points based on his/her interest, changing the parameters of the algorithms, etc.

3.2 An Analysis Example

We analyze the Wine Recognition dataset from UCI Machine Learning Repository [21] while following the workflow shown in Fig. 1. The dataset includes 178 data points (wines) with 13 features (e.g., alcohol, color intensity, and flavanoids). First, to generate a DR result, we use t-SNE [71] for all of the data points. Then, to detect clusters, we apply DBSCAN [22] to the DR result. As shown in Fig. 2a, we identify three clusters, colored with green, orange, and brown. The black data points are outliers or noise points labeled by DBSCAN. To understand the characteristics of the wines in each cluster, the system immediately applies our cPCA-based analysis method for each detected cluster. Now, we have obtained the features' contributions to contrasting each cluster. The measures of contributions are visualized with a blue-to-red divergent colormap, as indicated in Fig. 2b. As the absolute value of the measure approaches 1, the corresponding feature has a higher contribution. Finally, for each cluster, we visualize histograms of values of the three features that have the highest contributions. The results are shown in Fig. 2c. The histograms for each target cluster are colored with its respective cluster color, while the others are colored gray. The y-axis shows relative frequency and its maximum limit is set to the maximum relative frequency of each pair of the histograms.

Based on the result shown in Fig. 2, we can easily perceive each cluster's characteristics. For example, the green cluster has higher alcohol percentage ('Alc') and flavanoids when compared to the others. The orange cluster has lower magnesium, proline, and alcohol percentage. Also, the brown cluster has lower OD280/OD315 (i.e., low dilution degree), lower hue, and higher color intensity. The black outliers have higher magnesium and proanthocyanidins ('Proanth').

Even though this analysis example uses relatively a small number of features and clusters, finding these results is not a trivial task without the suggestions of highly contributed features. For example, in Fig. 3, similar to [42], we visualize each cluster's feature values with parallel coordinates [33]. Without our method, to find the same results, the user would need to review all the features of each cluster one by one. This is not only time-consuming but also introduces a possibility of overlooking important characteristics.

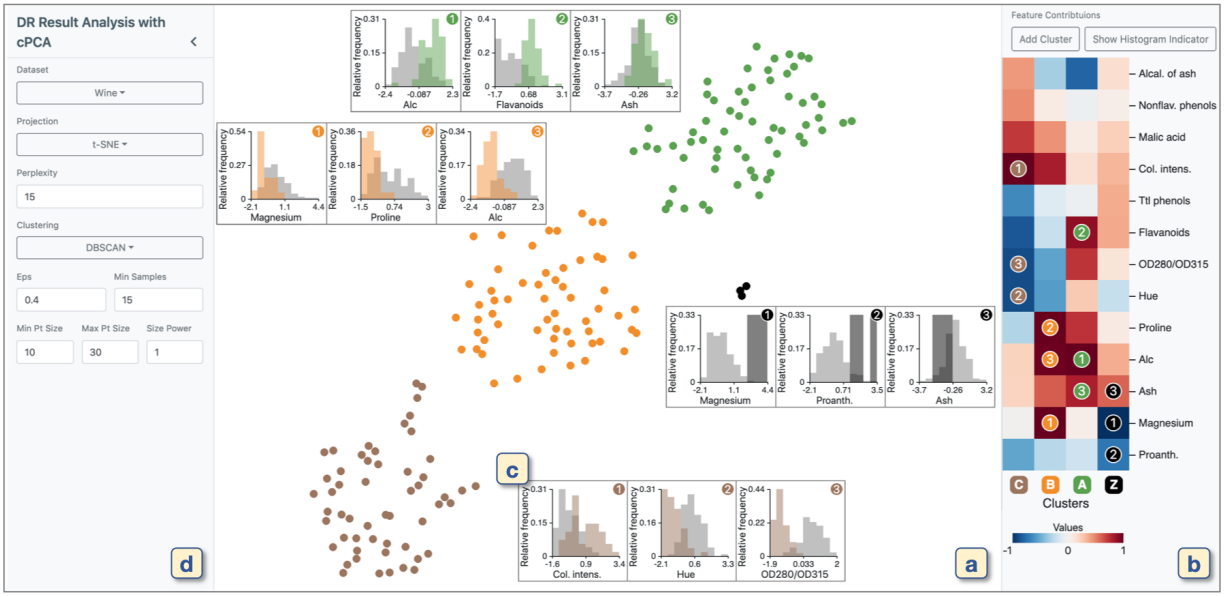


Fig. 2: A screenshot of our prototype system. The dimensionality reduction (DR) view (a) visualizes a result after DR and clustering. The feature contributions view (b) shows the measures of each feature's contribution to *contrasting* each cluster with the others. The feature values of the selected cells in (b) are visualized as histograms, as shown in (c). In (d), we can change the settings for the analysis methods and visualizations.

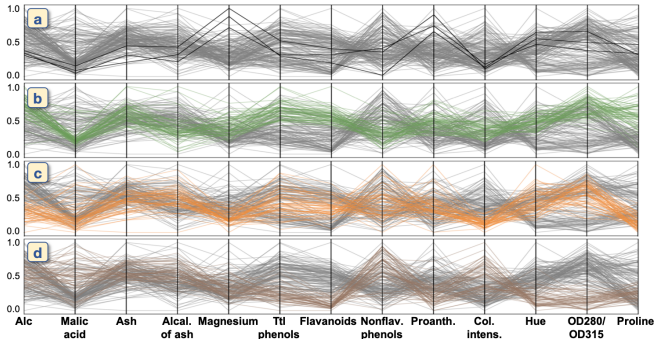


Fig. 3: Parallel coordinates showing all features in the Wine Recognition dataset. The corresponding polylines for the wines are highlighted with (a) black, (b) green, (c) orange, and (d) brown clusters. It is difficult to discern the essential features from this visualization.

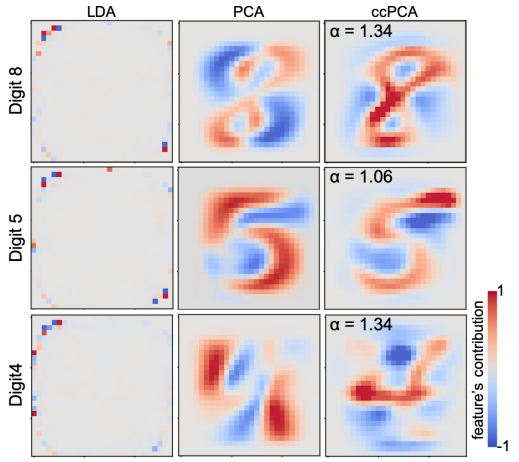


Fig. 4: Comparison of features' relative contributions of MNIST digits. We compare LDA, PCA, and ccPCA. All of these methods can calculate the features' relative contributions to the first component by respectively referring to either LDA's loadings, PC loadings, or cPC loadings described in Sect. 4.3. We scale each loading in the range from -1 to 1 by dividing the maximum absolute value of the loadings. We visualize the scaled loading for each pixel with a blue-to-red colormap. For LDA, we perform classification between the target digit and the others. The LDA results, placed on the left column, show that the outside pixels have high contributions. We can consider that LDA tries to distinguish each target digit from the others by referring to the pixels that are less frequently used in the other digits. We apply PCA to each target cluster in the same manner as [36, 69]. We can see that the PCA results show variations of the strokes when drawing each digit. The cPCA results are obtained from ccPCA with the automatic selection of α (refer to Sect. 4.2). When compared with PCA, the cPCA results clearly show the strokes contrasting the target digit with the others. For example, for Digit 5, the pixels on the upper right have high contributions, as indicated in dark red. When only drawing Digit 5, we tend to use these pixels, and thus, we can see that cPCA captures Digit 5's characteristics. Similarly, for Digit 4, we can see that there are dark red pixels around the middle left.

4 Methodology

As demonstrated in Sect. 3.2, when a dataset has many features, even only around ten, reviewing the values for each feature becomes tedious. Finding features which contrast each cluster with the other data points is the core analysis of our approach. To do this, we utilize cPCA [4,5] and its linearity to obtain the features' contributions (FCs) to the contrast.

There is a clear advantage of using cPCA over PCA [37] and LDA [34], both of which are linear DR methods. PCA has been used to find the representative features within the selected data points [36, 69]. However, as shown in the examples of Fig. 4(middle), while PCA is useful to find variations within each cluster, it cannot consider the differences between one cluster and the others. This consideration is important to find the unique characteristics in the target cluster. On the other hand, LDA focuses only on distinguishing the target cluster from the others. Thus, as shown in Fig. 4(left), LDA would judge whether a feature has a high contribution to distinguishing the target cluster even in the case where the feature has little variance in the target cluster and zero variance in the others. This could frequently happen especially when the number of features is large. Our cPCA-based method, ccPCA, finds the features which are well-balanced in terms of variety (similar to PCA) and separation (similar to LDA). Also, this balance can be controlled with the contrast parameter, as described in Sect. 4.2.3.

4.1 Contrastive PCA (cPCA)

We provide a brief introduction to cPCA [4, 5, 25], which we utilize to find features contrasting a target cluster with the other data points. cPCA is developed for "the setting where we have multiple datasets and are interested in discovering patterns that are specific to, or enriched in, one dataset relative to another" [4]. For instance, from the examples in [4], when we have a medical dataset X of diseased patients, we would want to find trends and variations of the disease's influence. If we apply the classical PCA [32, 37] to X, the first principal component

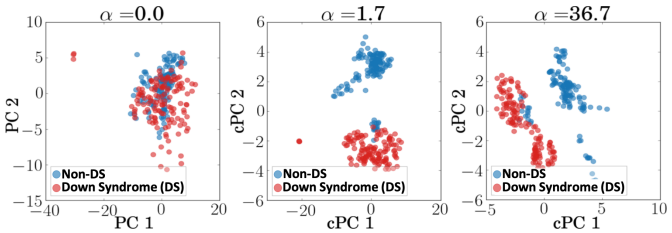
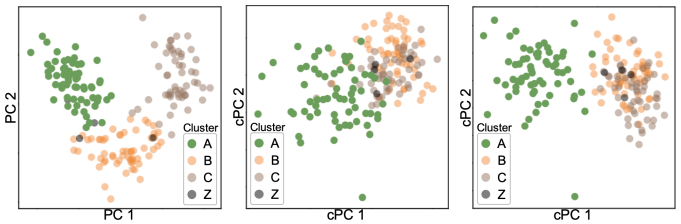


Fig. 5: cPCA results of the Mice Protein Expression dataset [30] from [4]. A different contrast parameter α value is used for each result. When $\alpha = 0$, cPCA generates the same result when applying PCA to the target dataset. In this result, we cannot see clear differences between down syndrome (DS) and non-DS mice, indicated with red and blue points, respectively. While clear differences between DS and non-DS start to appear when $\alpha = 1.7$, we can see that DS is further separated into two groups when $\alpha = 36.7$. More examples can be found in [4, 5].



(a) PCA (b) cPCA ($\alpha = 2.15$) (c) ccPCA ($\alpha = 4.38$)

Fig. 6: The DR results of the Wine Recognition dataset. The cluster labels generated in Sect. 3.2 are used. Here, we try to find the (c)PC contrasting the green cluster. In (a), we apply the classical PCA to the entire dataset. Though there is a separation of the green cluster when using the first and second PCs, there are overlaps of the green and orange clusters when only using the first PC (PC 1). In (b), the data points in the green cluster are used as the target dataset and the other data points are used as the background dataset. α value is selected from the suggestions using the semi-automatic selection in Sect. 4.1.2. We cannot see a clear separation of the green cluster from the others. In (c), we use the entire data points instead of only the green cluster as the target dataset. α value is selected with our automatic selection method in Sect. 4.2.3. We can see a better separation when compared to that of (a) and (b) even when using only the first cPC (cPC 1).

would only present the diseased patients' demographic variations [24], instead of showing the variation of the disease's effects. However, if there is another medical dataset Y of healthy patients, cPCA can utilize the fact that Y could have similar demographic variations as X, and no variations related to the disease. By taking X and Y as the target and background datasets, respectively, cPCA can find the directions (or components) in which X has high variance but Y has low variance.

4.1.1 Description of the Algorithm

Now, we describe how cPCA obtains such directions by using the target and background datasets. Let $X = \{\mathbf{x}_i\}_{i=1}^{n}$ be the target dataset and $Y = \{\mathbf{y}_i\}_{i=1}^{m}$ be the background dataset where $\mathbf{x}_i, \mathbf{y}_i \in \mathbb{R}^d$, n and m are the numbers of data points, and d is the number of dimensions (or features). Similar to the classical PCA, for the first step, cPCA applies centering to each dimension of X and Y and then obtains their corresponding empirical covariance matrices $\mathbf{C_X}$ and $\mathbf{C_Y}$. Let $\mathbf{v}$ be any unit vector of d dimensions.

Then, with a given direction $\mathbf{v}$, the variances for the target and background datasets can be written as: $\lambda_X(\mathbf{v}) \stackrel{\text{def}}{=} \mathbf{v}^\mathsf{T}\mathbf{C_X}\mathbf{v}$, $\lambda_Y(\mathbf{v}) \stackrel{\text{def}}{=} \mathbf{v}^\mathsf{T}\mathbf{C_Y}\mathbf{v}$. Now, the optimization that finds a direction $\mathbf{v}^*$ where X has high variance but Y has low variance can be written as:

$$\mathbf{v}^* = \underset{\mathbf{v}}{\operatorname{argmax}}\ \lambda_X(\mathbf{v}) - \alpha\lambda_Y(\mathbf{v}) = \underset{\mathbf{v}}{\operatorname{argmax}}\ \mathbf{v}^\mathsf{T}(\mathbf{C_X} - \alpha\mathbf{C_Y})\mathbf{v} \qquad (1)$$

where α is a contrast parameter ($0 \leq \alpha \leq \infty$). We describe the details of α in Sect. 4.1.2. From Eq. 1, we can see that $\mathbf{v}^*$ corresponds to the first eigenvector of the matrix $\mathbf{C} \stackrel{\text{def}}{=} (\mathbf{C_X} - \alpha\mathbf{C_Y})$. The eigenvectors of $\mathbf{C}$ can be calculated with eigenvalue decomposition (EVD). These computed eigenvectors are called contrastive principal components (cPCs) and are orthogonal to each other. Similar to the classical PCA, by using these cPCs (typically two cPCs), we can plot the DR result of X. An example from [4] is shown in Fig. 5.

4.1.2 The Contrast Parameter and Semi-Automatic Selection

The contrast parameter α controls the trade-off between having high target variance and low background variance. When $\alpha = 0$, cPCs will only maximize the variance of the target dataset. These cPCs are the same as the principal components (PCs) of the target dataset when computed with the classical PCA. As α increases, cPCs will become more optimal directions that reduces the variance of the background dataset. Fig. 5 shows the example from [4] with different α values.

As shown in Fig. 5, the selection of α has a strong impact on the DR result. Thus, Abid and Zhang et al. [4, 5] introduced an algorithm suggesting multiple α values. Their algorithm calculates a set of cPCs for each of the multiple values of α (with 40 values as their default), and the α values are logarithmically spaced in a certain range (the default is between 0.1 and 1000). Then, the similarity between each pair of the different cPCs, each obtained with a different α value, is measured by calculating the product of the cosine of the principal angles. Afterward, based on the user's input p (the number of values of α to suggest), the algorithm finds p clusters from the similarities with spectral clustering [53]. Finally, the algorithm returns p values of α which correspond to the medoids of the clusters. From the suggested p values, the algorithm returns a set of DR results. By referring to this set, the user can choose their preferred α value.

4.2 Finding the Direction that Contrasts a Target Cluster

As described above, cPCA discovers patterns that are specific to, or enriched in, the target dataset relative to the background dataset. In [4, 5], cPCA is designed for the situation where the patterns the user wants to identify are included within the target dataset X, while the background dataset Y contains the structure the user wants to remove from the target dataset. Therefore, in [4, 5], the provided examples for $\{X, Y\}$ are {'diseased subjects', 'control group subjects'}, {'patients after treatment', 'patients before treatment'}, {'images mixed with interests and noises', 'images only including noises'}, etc.

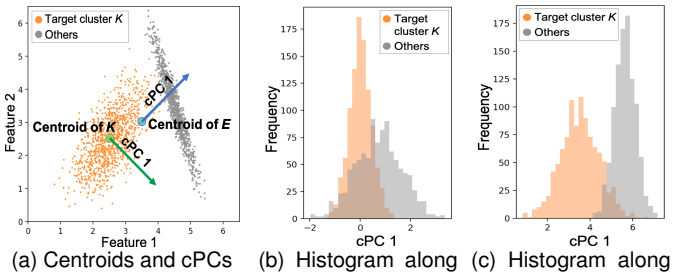
In our case, we want to find the directions (i.e., cPCs) which contrast one cluster with the other data points. If we follow the examples of X and Y as stated above, X can be the target cluster and Y can be the other data points. However, in this case, cPCA will find cPCs that only enrich the variations specific to the target cluster. For example, when the target cluster includes diseased subjects and the other data points correspond to healthy subjects, cPCA will find enriched variations within the diseased subjects (e.g., differences among multiple diseases), but will not consider the differences between diseased and healthy subjects.

To utilize cPCA for finding the directions contrasting a target cluster with the others, we introduce a novel usage of cPCA, named ccPCA. Instead of using the target cluster as the target dataset X and the other data points as the background dataset Y, we use the entire dataset as X and the data points other than the target cluster as Y. With this approach, we can find the directions that contrast the target cluster. As we describe in the following subsections, ccPCA has the strengths in regards to two aspects: (1) an implicit extension of the contrast parameter α and (2) a proper setting of the centroid. The DR results shown in Fig. 6 provide a comparison of the classical PCA, original usage of cPCA (i.e., using only the target dataset as X), and ccPCA.

Let $E = \{\mathbf{e}_i\}_{i=1}^{s}$ be the entire dataset and $K = \{\mathbf{k}_i\}_{i=1}^{t}$ be the target cluster ($K \subset E$, $\mathbf{e}_i, \mathbf{k}_i \in \mathbb{R}^d$, s and t are the numbers of data points). Then, we denote $R = \{\mathbf{r}_i\}_{i=1}^{u}$ as the difference of the two sets K and E (i.e., $R = E \setminus K$ and $u = s - t$). With these notations, we can say that ccPCA uses E and R as the target X and background Y datasets, respectively.

4.2.1 An Implicit Extension of the Contrast Parameter

To provide a simple and clear explanation, we assume the centering effects to the datasets E, K, and R are all the same (i.e., E, K, and R have the same mean value for each feature). After centering the target dataset E and the background dataset R, cPCA obtains their corresponding



(a) Centroids and cPCs (b) Histogram along the green cPC 1 (c) Histogram along the blue cPC 1

Fig. 7: A comparison of centering effects to the cPCA results. For this example, we generate two sets of data points from different 2D Gaussian distributions. In (a), the green circle and arrow show the centroid and the first cPC when using the target cluster K as the target dataset and the others as the background dataset. The blue circle and arrow are the centroid and the first cPC when using the entire dataset E as the target dataset. From the DR results, as shown in (b) and (c), ccPCA, using the entire dataset E as the target dataset, generates a better separation between the target cluster and the others.

empirical covariance matrices $\mathbf{C_E}$ and $\mathbf{C_R}$. Then, cPCA calculates cPCs by performing EVD to $\mathbf{C_E} - \alpha\mathbf{C_R}$. Let $\mathbf{C_K}$ be the empirical covariance matrix of the target cluster K after centering. Because $\mathbf{C_K} = \sum_{i=1}^{t} \mathbf{k}_i\mathbf{k}_i^\mathsf{T}/t$, $\mathbf{C_R} = \sum_{i=1}^{u} \mathbf{r}_i\mathbf{r}_i^\mathsf{T}/u$, $E = K \sqcup R$, and $s = u + t$, $\mathbf{C_E}$ can be represented as $\mathbf{C_E} = (t\mathbf{C_K} + u\mathbf{C_R})/s$. With this, $\mathbf{C_E} - \alpha\mathbf{C_R}$ can be rewritten as:

$$\mathbf{C_E} - \alpha\mathbf{C_R} = (t\mathbf{C_K} + u\mathbf{C_R})/s - \alpha\mathbf{C_R} \tag{2}$$

$$= \frac{t}{s}\left(\mathbf{C_K} - \frac{(s\alpha - u)}{t}\mathbf{C_R}\right) = \frac{t}{s}(\mathbf{C_K} - \beta\mathbf{C_R}) \tag{3}$$

where $\beta = (s\alpha - u)/t$. Because $0 \leq \alpha \leq \infty$, $-u/t \leq \beta \leq \infty$. Note that if we use K and R as the target and background datasets, respectively, cPCA performs EVD to $\mathbf{C_K} - \alpha\mathbf{C_R}$. Therefore, a fundamental difference between the cases of using E (i.e., the entire dataset) and using K (i.e., only the target cluster) as the target dataset for cPCA is the difference between α and β.

While α only takes a non-negative value, β can be a negative value. When $\beta = -u/t$, cPCA selects the directions that maximize the variance of the entire dataset E, and hence reduces to PCA applied on E. As β increases to 0, cPCA provides more weight to the target cluster K than the others R to select the directions. When $\beta = 0$, cPCA selects the directions that maximize the variance of the target cluster K, and hence reduces to PCA applied on K. Then, as β increases from 0 to ∞, the directions from cPCA will become more optimal to reduce the variance of the others R. While Eq. 3 with $\beta >= 0$ has a capability to find the same directions with $\mathbf{C_K} - \alpha\mathbf{C_R}$, ccPCA also searches the directions that considers the differences between the target cluster K and the others R by using the range $\beta < 0$.

4.2.2 The Centering of the Target Dataset

ccPCA not only implicitly extends the searching range of α of $\mathbf{C_K} - \alpha\mathbf{C_R}$, but it also uses a proper centroid of the dataset. The centering (i.e., the mean subtraction for each feature) in cPCA is used for translating the dataset to its centroid. When using K as the target dataset, the centroid is calculated from only the target cluster K. In contrast, ccPCA uses E as the target dataset, and the centroid is calculated from all the data points. Fig. 7 shows an example of the two methods of calculating the centroid and the first cPC in each case. As the same reason as the classical PCA, the centering should be applied to the entire dataset in our case. This is to ensure that the first cPC is the direction of the maximum variance, which contrasts the differences between the target cluster and the others.

4.2.3 Automatic Selection of the Best Contrast Parameter

The selection of the contrast parameter α is the remaining procedure. Even though we can use the existing semi-automatic selection of α in Sect. 4.1.2, selecting the best alpha from the multiple suggested options is tedious when analyzing multiple clusters. Thus, we introduce a method for an automatic selection of the best α for our usage. The pseudocode of this method is available in the Supplementary Materials [1]. To understand the characteristics of the cluster, we should find the first cPC which not only (1) shows a clear separation between the target cluster from the others, but also (2) maintains the variability in the target cluster well (i.e., a high variance within the target cluster). Similar to the classical PCA, the second condition tries to preserve the target clusters' original structure. Without the second condition, when using a large α, cPCA may preferentially select features where the target cluster only has subtle variability, but the other data points have no variability (i.e., zero variance). This example can be seen in the far right of Fig. 8.

Similar to the semi-automatic selection in Sect. 4.1.2, our automatic selection lists multiple candidates of α (our default is also 40 values). These candidates consist of 0 and a set of logarithmically spaced values given a certain range (our default also ranges from 0.1 to 1000). We denote these alphas as $\{\alpha_i\}_{i=1}^{q}$ (q is the number of candidate values for the best α) and assume $\{\alpha_i\}$ is sorted by ascending order (i.e., $\alpha_1 = 0$). Then our method selects a value that obtains the best separation while having enough variance in the target cluster K.

To measure the separation between the target cluster and the others along the first cPC, we use the histogram intersection (HI) [64], which can measure the overlaps of the histograms of the two sets. While there are many different (dis)similarity measures between two probability distributions, such as the Kullback-Leibler divergence [41], we chose HI for its robustness to outliers and low computational cost. Let $H_A = \{h_{A_j}\}_{j=1}^{b}$, $H_B = \{h_{B_j}\}_{j=1}^{b}$ be the histograms of two given sets of real numbers A and B where b is the number of bins, h_{A_j} and h_{B_j} are the numbers of data points in the j-th bin of A and B, respectively. Both H_A and H_B have the same bins. We decide the bin-width using Scott's normal reference rule [62] from the set of real numbers obtained by combining A and B. The HI of the two sets A and B is defined as: $I(A,B) = \sum_{j=1}^{b} \min(h_{A_j}, h_{B_j})$. Let K'_i and R'_i be the data points of 1D DR results of K and R with the first cPC corresponding to the i-th candidate α value (i.e., α_i), respectively. Then, we can calculate the measurement of separation with the inverse HI (i.e., $I(K'_i, R'_i)^{-1}$) for each α_i. We refer $I(K'_i, R'_i)^{-1}$ as the discrepancy score $D(\alpha_i)$.

For the variance of K'_i, to handle the scaling differences in each DR result, first, we apply the min-max scaling to K'_i with the minimum and maximum values of $K'_i \sqcup R'_i$. Then, we calculate the variance of the scaled K'_i. We denote this variance of K'_i as $V(\alpha_i)$.

With the measures of $D(\alpha_i)$ and $V(\alpha_i)$, our automatic selection method selects the best alpha with:

$$\underset{\alpha_i \in \{\alpha_1, \ldots, \alpha_q\}}{\text{argmax}} \quad D(\alpha_i) \;\; s.t. \;\; V(\alpha_i) \geq \gamma V(\alpha_1) \tag{4}$$

where γ ($\gamma \geq 0$) is a ratio that controls the threshold of the variance $V(\alpha_i)$. Note that $V(\alpha_1)$ is the variance of K'_1 of the cPCA result with $\alpha = 0$, which will be the same result when applying the classical PCA to the entire dataset E. While our method allows the user to select any non-negative value for γ, we set $\gamma = 0.5$ as the default to ensure that $V(\alpha_i)$ has at least a half of $V(\alpha_1)$. Fig. 8 shows the cPCA results with different α values. Our automatic α selection chooses $\alpha = 1.06$ in this case. More comprehensive experimental results with various datasets and α values can be found in the Supplementary Materials [1].

In summary, the original cPCA is enhanced as ccPCA by using Eq. 1 with $X = E$ and $Y = E \setminus K$ and by selecting α as the solution to Eq. 4.

Parallel calculation of the best contrast parameter: The original semi-automatic selection of the contrast parameter in [4,5] calculates cPCA for each $\alpha_i \in \{\alpha_i\}_{i=1}^{q}$ in serial [2] ($q = 40$ by default). Because the calculation of cPCA for each α_i is independent of each other, in order to achieve faster computation, our method uses multi-threads and calculates each cPCA result, $D(\alpha_i)$, and $V(\alpha_i)$ in parallel. The comparison of the completion time of the original cPCA and our implementation with and without parallelization is available in [1].

4.3 Features' Relative Contributions to the First cPC

By using cPCA, with our automatically selected α, we can now obtain the direction (i.e., the first cPC) that contrasts the target cluster. Next,

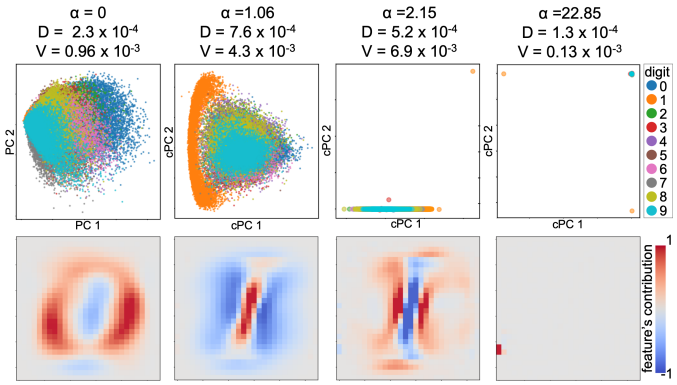


Fig. 8: The DR results with the first and second cPCs (top row) and the features' (i.e., pixels') relative contributions (bottom row) of the MNIST dataset [45] with different α values (refer to Sect. 4.3 about the features' relative contributions). Here, we try to contrast Digit 1 with the other digits. We can see that when $\alpha = 0$ (reduced to applying the classical PCA for all digits), cPCA does not separate Digit 1, and the features' contributions do not show any useful information to understand the characteristics of Digit 1. On the other hand, when $\alpha = 22.85$, while some of Digit 1 (e.g., points placed on the top left) are well separated, the variance V is small. Also, from the features' contributions, we can see that only a few pixels in the lower left have high contributions. This is expected because these pixels are rarely used when drawing Digit 1. The result with $\alpha = 1.06$ produces the best discrepancy score D and a large variance V. This α will be selected by Eq. 4. Also, we can see that cPCA highlights the pixels around the center, which are typically used for drawing Digit 1.

we determine how strongly each feature of the target cluster contributes to this direction. Similar to the classical PCA, by using the top eigenvalue λ^* and the corresponding eigenvector $\mathbf{v}^*$ (i.e., the first cPC) of the matrix $\mathbf{C_E} - \alpha\mathbf{C_R}$, the relative contributions can be calculated with: $\mathbf{w}^* = \sqrt{\lambda^*}\mathbf{v}^*$ where $\mathbf{w}^* = \{w_i^*\}_{i=1}^d$ $(-1 \leq w_i^* \leq 1)$. Analogous to the classical PCA, we call $\mathbf{w}^*$ the cPC loadings of the first cPC. As $|w_i|$ approaches 1, the i-th feature has a stronger contribution (or correlation) to the first cPC. Based on this value, we can decide which features we should review to understand the target cluster. Fig. 4 shows an example of the features' contributions and comparisons with the results from LDA and PCA. Comprehensive comparisons of LDA and PCA, using multiple datasets, can be found in the Supplementary Materials [1]. As shown in Fig. 4, signed cPC loadings can clearly differentiate features whose positive centered values contribute to the negative or positive direction of the first cPC by using blue and red, respectively. This is as opposed to taking the absolute value of the signed cPC loadings.

5 Visual Analytics System

To demonstrate our methods of analyzing real-world datasets, we develop a prototype system that supports the analysis workflow shown in Fig. 1. A major portion of the system's functionality and a video of an interaction demonstration are available in our online site [1].

5.1 Dimensionality Reduction View

The dimensionality reduction (DR) view, as shown in Fig. 2a, is used for the first two processes: generating a DR result and identifying clusters. In this view, first, the user can visualize a 2D DR result of a high-dimensional dataset. We employ t-SNE [71] (specifically, Barnes-Hut t-SNE implementation [70]) as a DR method because t-SNE can effectively depict the local structure of the dataset, and thus, it is useful to visually identify the clusters within the dataset. From the settings in Fig. 2d, the user can adjust the perplexity parameter of t-SNE, which controls a balance of the effects from local and global structures of the dataset [71]. While a larger perplexity will preserve more of the distance relationship in the global structure, a smaller perplexity will focus on more preserving the distance relationship among a small number of neighbors.

After obtaining the DR result with t-SNE, the user can identify clusters automatically or manually. As a default, the automatic clustering method will be immediately applied to the obtained DR result. As part

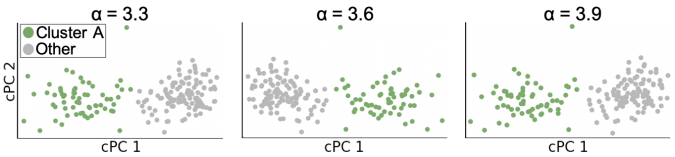


Fig. 9: Sign flipping of cPCs. We generate the cPCA results of the Wine Recognition dataset used in Sect. 3.2 with different α values. Sign flipping occurs between $\alpha = 3.3$ and $\alpha = 3.6$; $\alpha = 3.6$ and $\alpha = 3.9$.

of the automatic method, our system supports DBSCAN [22] because the density-based clustering algorithm is able to identify clusters with arbitrary shapes [57], which are often generated from DR. The user can change the parameters required for DBSCAN from the settings in Fig. 2d. The categorical color of each point in the DR result is assigned to the clustering label obtained from DBSCAN. The color black, in particular, is used to represent outliers or noise points labeled by DBSCAN. For a manual selection of a cluster, the system supports a rectangle selection. The user can select data points by drawing a rectangle with mouse dragging in the DR result. Also, the user can add additional data points or unselect data points by using different selection modes provided in the system. From these interactions, the user can create a new cluster consisting of the selected points by clicking the "Add Cluster" button placed at the top of Fig. 2b. The system also supports basic view-level interactions, such as zooming and panning.

5.2 Features' Contributions View

The two remaining processes (i.e., finding features contrasting each cluster and comparing the features' values in detail) are performed with the features' contributions (FCs) view shown in Fig. 2b. In the FCs view, the FCs contrasting each cluster described in Sect. 4.3 are visualized as a heatmap. While each row name shows the corresponding feature, each column name shows the cluster label ('Z' is used to represent the outliers, noise points, or both). Also, to indicate the corresponding cluster in the DR view, the background of each column name is colored with the corresponding color. We scale each cluster's FCs in the range from -1 to 1 by dividing each FC by the maximum absolute value of the FCs (e.g., the original range from -0.1 to 0.5 will be changed to the range from -0.2 to 1.0). Then, we encode the scaled FCs with a blue-to-red colormap. In the next subsections, we describe our algorithm organizing the heatmap.

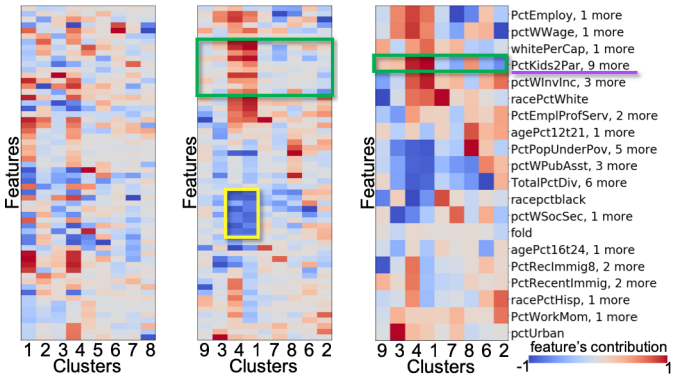
5.2.1 Optimal Sign Flipping of cPCs and FCs

Similar to the classical PCA, cPCA has the "sign ambiguity" problem [13, 23, 35]. Because of this problem, arbitrary sign flipping in each (c)PC occurs when performing EVD. An example of sign flipping in cPCA is shown in Fig. 9. Sign ambiguity affects the comparison of the FCs among the clusters. Each cluster might have the opposite direction of the first cPC only due to this sign ambiguity problem. In this case, the FCs also have opposite signs, and thus, it is difficult to judge whether these clusters have similar patterns in the FCs or not.

To solve this problem as much as possible, we introduce a method to optimally reduce unnecessary sign flipping. Let $\mathbf{v}_i^*$ and $\mathbf{v}_j^*$ be the first cPCs of i-th and j-th clusters, respectively. We can measure how the directions $\mathbf{v}_i^*$ and $\mathbf{v}_j^*$ are similar with the cosine similarity $sim(i,j) = \mathbf{v}_i^* \cdot \mathbf{v}_j^* / (\|\mathbf{v}_i^*\| \|\mathbf{v}_j^*\|)$. $\mathbf{v}_i^*$ and $\mathbf{v}_j^*$ have the same direction when $sim(i,j) = 1$, while $\mathbf{v}_i^*$ and $\mathbf{v}_j^*$ have opposite directions when $sim(i,j) = -1$. Ideally, by flipping the signs of the first cPCs of some clusters, we want to ensure that all of the clusters' first cPCs face the same side (i.e., $sim(i,j) \geq 0 \ \ \forall i,j$). However, the sign flipping to a certain cluster affects all cosine similarities related to this particular cluster. Thus, in many cases, it is theoretically impossible to obtain the result stated above. However, alternatively, we can maximize the sum of all $sim(i,j)$ with sign flipping. This optimization can be written as:

$$\underset{\varphi=\{\varphi_i,\ldots,\varphi_l\}}{\operatorname{argmax}} \sum_{i=1}^{l} \sum_{j=1, j\neq i}^{l} \frac{(\varphi_i \mathbf{v}_i^*) \cdot (\varphi_j \mathbf{v}_j^*)}{\|\mathbf{v}_i^*\| \|\mathbf{v}_j^*\|} = \sum_{i=1}^{l} \sum_{j=1, j\neq i}^{l} \varphi_i \varphi_j sim(i,j)$$
$$s.t.\ \varphi_i, \varphi_j \in \{-1, 1\} \quad (5)$$

where l is the number of clusters and φ is a set of signs.



(a) Original (b) Reordered (c) Aggregated

Fig. 10: Reordering and aggregation of the FCs. (a) shows the original FCs. There are 8 clusters and 60 features. (b) shows the reordered FCs in both rows (i.e., features) and columns (i.e., clusters). With (b), we can see a group of similar FCs (e.g., the features are indicated with a yellow rectangle). In (c), the 60 feature clusters are aggregated into 20 rows. For example, the ten features indicated with the green rectangle in (b) is aggregated into one row indicated with the green rectangle in (c).

We solve Eq. 5 with a heuristic approach. We initialize $\varphi = \{1,1,\ldots,1\}$. We can expect that there is a higher chance to obtain a better result if we start to flip the sign where i-th cluster has the largest negative value in the sum of the similarities ($\sum_{j=1}^{l} \varphi_i \varphi_j sim(i,j)$). Therefore, our approach first checks whether sign flipping to the first cPC of such a cluster provides a better result in the objective function of Eq. 5. If so, we flip its first cPC's sign. Then, we repeatedly apply this procedure until $\sum_{j=1}^{l} \varphi_i \varphi_j sim(i,j) \geq 0$ for all $i \in \{1,2,\ldots,l\}$ is satisfied or all clusters have been checked. Afterward, based on the optimized set φ, we allocate the new signs to respective cPC and FCs for each cluster.

5.2.2 Ordering of Features and Clusters

The FCs view can be used for finding not only the heatmap cells which have high FCs, but also the clusters which have similar FC patterns; the features which have similar FCs within and/or among clusters. The case when the clusters have similar FCs implies that these clusters are contrasted due to the same features, but they have different distributions in their features' values. When the features have similar FCs, by reviewing the distributions of one of these features' values, we can expect that the other features may also have similar distributions.

To help find these patterns, our system applies reordering of the features (i.e, rows) and clusters (i.e., columns) based on the FCs. Ordering choice is important since this affects how easily we can find patterns in a heatmap [10]. We use a hierarchical clustering, specifically the complete-linkage method [52], with the optimal-leaf-ordering [9]. Recent survey [10] reported that this combination tends to produce a coherent and quality result to help reveal patterns. Fig. 10a and b show the results before and after the reordering. From Fig. 10b, we can easily see a group of similar FCs.

5.2.3 Scalable Visualization

When the number of features is large (e.g., 100 or more), the heatmap-based visualization would have a scalability issue. Moreover, in this case, many features could have high FCs, and as a result, it would still be difficult to decide which features we should review in detail. To solve this issue, we introduce an aggregation method, utilizing the hierarchical clustering result obtained through the reordering method.

When the number of features is larger than threshold δ (we set $\delta = 40$ as a default), our method obtains δ clusters from the features by referring to the hierarchical clustering result. Then, our method aggregates the FCs into one representative value: the mean or the maximum absolute value. As a default, our method takes the maximum absolute value to show the most prominent feature. Fig. 10c shows an example of the aggregation. Additionally, to provide a representative name for each aggregated feature, our method chooses the name based on which FC has the maximum absolute value. With this name, our method also shows how many features are aggregated in each row, as shown with a purple underline on the right side of Fig. 10c ('PctKids2Par, *9 more*').

5.3 Interactions between Views

From DR View: When the user updates the clusters with the clustering method in the DR view, the FCs view updates the heatmap with the reordering (and aggregation) method(s). When the user adds a new cluster manually, the FCs view updates the heatmap with the new cluster.

From FCs View: The FCs view can be used as an interface to compare the details of the features' values within/across features or clusters. When the user places the mouse over a certain heatmap cell, the system shows a popup window of the histograms of feature values of the corresponding cluster and the others (e.g., Fig. 2c and Fig. 14b). We color the selected cluster's histogram with a categorical color representing its cluster label, while the gray color is used for the other data points' histogram. When hovering over a certain (representative) feature name, the system shows a value of the (representative) feature as the size of each data point in the DR view (e.g., Fig. 12a and Fig. 13a).

Moreover, when hovering over a certain cluster label, the system highlights the corresponding cluster in the DR view. In addition, with the popup window, the system visualizes the histograms of 1D DR results of the cluster and the others. From these histograms, the user can grasp how well the cluster is contrasted with the other data points. Additionally, the system shows the histograms of three (representative) feature values that have the highest absolute FCs. These histograms are useful to understand each cluster's characteristics quickly.

Also, to make the comparison within/across features or clusters easier, our system allows the user to prevent the histograms from disappearing with a mouse-click. The clicked histograms can also be moved with mouse-dragging. The corresponding heatmap cell for each histogram is annotated with a gray line and a pair of numbers shown in the heatmap cell and the histogram (e.g., Fig. 2 and Fig. 14b). The gray line can be turned on or off by clicking the "Show/Hide Histogram Indicator" placed at the top of the FCs view.

5.4 Implementation

We have developed our system as a web application. To achieve fast calculation, we have implemented our methods described in Sect. 4 with C++ and Eigen library [26] for linear algebraic calculations. We have also provided Python bindings for our C++ implementation. The source code is available in [1]. The back-end of the system uses Python with the stated bindings. The front-end visualization is implemented with a combination of Elm [17], HTML5, JavaScript, WebGL, and D3 [11]. While we use D3 for the FCs view, WebGL is used to render the data points efficiently for the DR view. We use WebSocket to communicate between the front- and back-ends.

6 Case Studies

We have shown the effectiveness of our methods with the Wine Recognition [21] and MNIST [45] datasets in the previous sections. We demonstrate three additional case studies with publicly available datasets. For each case study, we preprocess the corresponding dataset to clean up missing values in the data or extract useful information for the analysis. All the preprocessed datasets are available online [1].

6.1 Tennis Major Tournament Match Statistics

We analyze the Tennis Major Tournament Match Statistics dataset from UCI Machine Learning Repository [21]. This dataset contains the match statistics for both females and males at four major tennis tournaments in 2013. The statistics include first serve won by each player, double faults committed by each player, etc. From this dataset, we obtain female players' mean values for each statistic across all tournaments. The obtained dataset consists of 174 data points (tennis players) and 13 features (statistics).

Similar to the analysis of Sect. 3.2, we obtain the DR result with t-SNE, clusters with DBSCAN, and FCs with our methods. Then, to analyze each cluster's characteristics, we show the histograms of the top 3 contributed features. The result is shown in Fig. 11.

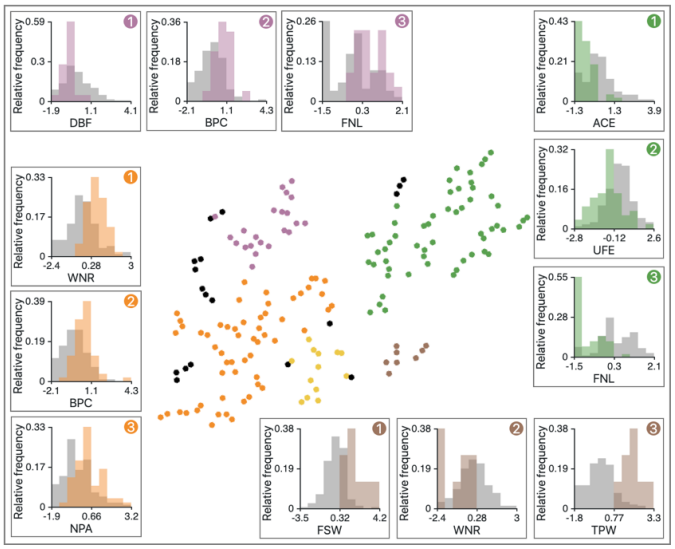


Fig. 11: An analysis result of female players from the Tennis Major Tournament Match Statistics dataset.

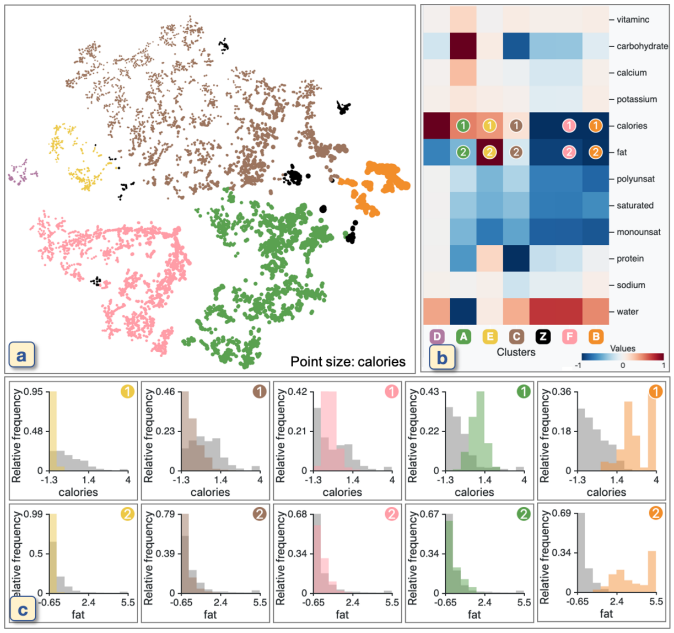


Fig. 12: A result of the Nutrient dataset. (a) shows the result after applying t-SNE and DBSCAN. A point's color and size show the clustering label and the value of 'calories', respectively. (b) shows the FCs of each cluster. (c) shows the histograms of the selected cells in (b), as indicated with the colored numbers in both (b) and (c).

From Fig. 11, we can see that each cluster has a different playing style. For example, the purple cluster tends to have low 'DBF' (double faults committed by player), high 'BPC' (break points created by player), and high 'FNL' (final number of games won by player). This indicates that these players had fewer mistakes in their serves and performed well when they were the receiver, and as a result, they won more games. Similarly, the orange cluster has high 'WNR' (winners earned by player) and 'NPA' (net points attempted by player). These statistics will tend to be higher when a player tries to obtain points aggressively during a rally. On the other hand, the brown cluster has low 'WNR' but high 'FSW' (first serve won by player) and 'TPW' (total points won by player). Therefore, we can say that these players tend to obtain more points with their serves.

6.2 Food and Nutrient

We analyze the Nutrient dataset in the USDA Food Composition Databases [66] as an analysis example with a large number of data

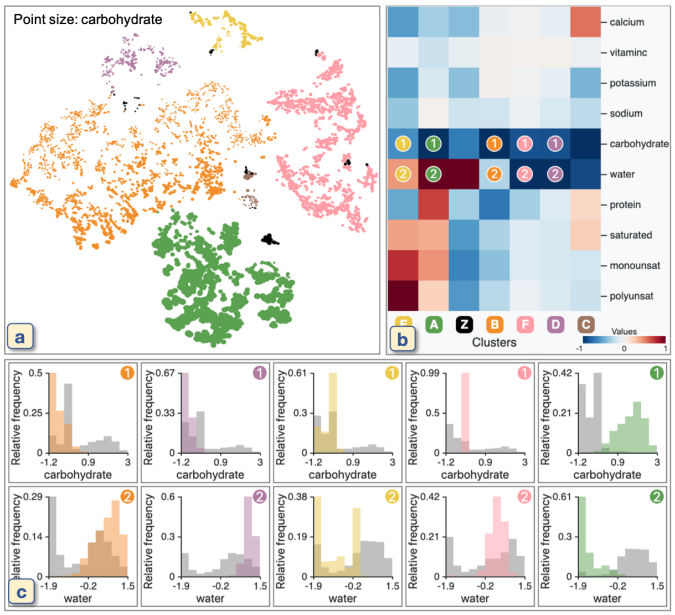


Fig. 13: The result after filtering out the 'calories' and 'fat' features from the Nutrient dataset. In (a), a point size represents the value of 'carbohydrate'. The histograms of the selected cells in (b) are shown in (c).

points. We use the version available from [16]. This dataset consists of the nutrient content for each food. The dataset has 7,637 data points (foods) and 14 features (nutrients).

This dataset has 12,507 missing values and this is 11.7% of all the values. Since this high percentage of missing values could affect an analysis result [7], we first preprocess the dataset to reduce this ratio to less than 5% [7]. We remove features where more than 40% of the values are missing. Also, we remove data points where more than 40% of the feature values are missing. Afterward, 7,499 data points and 12 features remain and there are 4,447 missing values (4.9% of all the values). We replace the missing values with the mean of each corresponding feature.

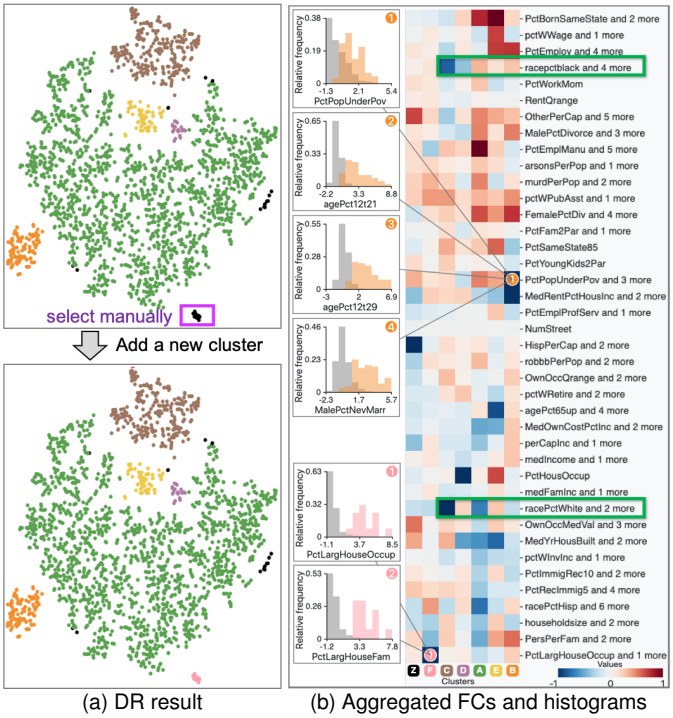
The result after using t-SNE, DBSCAN, and our methods is shown in Fig. 12. As shown in Fig. 12b, we can see that all clusters except for the brown cluster have high FCs in 'calories', 'fat', or both. When comparing the histograms of 'calories' and 'fat' for each cluster, as shown in Fig. 12c, each cluster, in fact, has different distributions in 'calories' and 'fat'. For example, while the yellow cluster tends to have low calories and fat, the orange cluster tends to have high values for both.

We have understood the main characteristics of each cluster. However, the effects of the two specific features ('calories' and 'fat') are too dominant. As a result, we cannot find any other interesting patterns. We preprocess the dataset to filter out these two features and generate a new result with new cluster labels, as shown in Fig. 13. At this time, from Fig. 13b, we can see that most of the clusters are contrasted by mainly 'water', 'carbohydrate', or both. For example, the purple and orange clusters placed in the upper left of Fig. 13a have fewer carbohydrates and more water when compared with the pink and green clusters, as shown in Fig. 13c. These two examples show that the FCs are useful to know which features have a dominant effect on cluster forming in the DR result.

6.3 Communities and Crime

As an example with a large number of features, we analyze the Communities and Crime dataset [59] from [21]. This dataset consists of both socio-economic and crime statistics (e.g., the median family income and the number of murders) for each community. The dataset contains 2,215 data points (communities) and 143 features (statistics) after excluding identifiers (e.g., county codes).

Because this dataset has many missing values (42,147 values, 13.3% of all the values), as similar to Sect. 6.2, we remove the features where more than 80% of the values are missing. The dataset now has 121



(a) DR result (b) Aggregated FCs and histograms

Fig. 14: The results for the Communities and Crime dataset. The top of (a) shows the result with t-SNE and DBSCAN. In the bottom of (a), the pink cluster which was not identified by DBSCAN is manually added. (b) shows 40 aggregated features from 121 features. Also, some of the histograms of the original features are visualized at the left of (b).

features and only 963 missing values (0.4% of all the values). We replace the missing values with the mean of each corresponding feature.

Fig. 14a (top) shows the result after DR and clustering. As indicated with the purple rectangle, we manually select an additional cluster as a pink cluster. Then, we obtain the FCs, as shown in Fig. 14b. Because there are many features, the system has aggregated them into 40 features using the aggregation method described in Sect. 5.2.3. From Fig. 14b, we can say that the small clusters (yellow, purple, brown, orange, and pink) are separated from the green cluster due to race, house size, etc.—not due to the criminal statistics. For instance, as indicated with the green rectangles, the brown cluster has high FCs in race percentages of African Americans and Caucasians ('racepctblack' and 'racePctWhite'). Also, the pink cluster has high FC in 'PctLargHouseOccup' (percentage of all occupied households that are large).

We show the histograms of the features aggregated to the 'PctLargHouseOccup and 1 more', as shown in the lower left of Fig. 14b. We can see that both 'PctLargHouseOccup' and 'PctLargHouseFam' (percentage of family households that are large) have similar distribution patterns. These patterns can be found because our aggregation method is performed after applying the optimal sign flipping and ordering described in Sect. 5.2. Our aggregation method is able to provide a scalable visualization and help the user analyze many features. Another example for 'PctPopUnderPov and 3 more' of the orange cluster is shown in the upper left of Fig. 14b. All 'PctPopUnderPov' (percentage of people under the poverty level), 'agePct12t21' (percentage of population that is 12–21), 'agePct12t29' (percentage of population that is 12–29), and 'MalePctNevMarr' (percentage of males who have never married) tend to have a higher value in comparison to that of others.

7 Discussion and Limitations

Generality of our method. We utilize cPCA [4, 5] to find features contrasting the target cluster. We discuss the reason why we use this approach instead of analyzing how the DR method generates clusters. If possible, the latter approach would be effective because the cluster formation is a result of the DR method. However, many of the nonlinear DR methods used for visualization (e.g., t-SNE [71], LargeVis [65], and UMAP [50]) generate irreversible low-dimensional projection of the original data structure. These methods do not have a parametric mapping between the original and projected dimensions; therefore, it is difficult to provide information about how these DR methods affect cluster forming. Our methods provide flexibility for analyzing results from any type of DR methods.

We introduce using cPCA to understand the characteristics of the clusters identified in the DR result. Our methods can also be used in other situations. For example, even though using DR before clustering is a common approach [74,76], our methods can support visual analytics of clusters that are obtained from the clustering methods without going through the DR step. This would be helpful to understand clusters' characteristics and to analyze the quality of the clustering methods without any effects derived from DR (e.g., distortion in the projection space). Another example is applying our methods to labeled data. Our methods can identify the essential features to contrast a labeled group from the others. Therefore, our methods would be useful to understand the characteristics of each group and could help design classifiers based on the gained knowledge. Our prototype system can support these types of analysis by changing the parts related to steps (a) and (b) in Fig. 1, such as the DR view and clustering algorithms.

Advantages of using cPCA. In Sect. 4, we have already discussed the advantages of using cPCA when compared with using PCA and LDA. It is also possible to compute the discrepancy score D introduced in Sect. 4.2.3 for each original feature without using ccPCA and then use the score as the feature contribution. However, this approach has a similar problem with LDA because the obtained score only shows the separation and does not take into account the variety (i.e., variance) for each feature.

Another potential option is using the two-group differential statistics methods [49], such as two-sample t-test, Wilcoxon signed-rank test, and Mann-Whitney U test, to find features that have differences between the target cluster and the others. Unlike LDA, PCA, or cPCA, these methods cannot produce a quantitative measure for analyzing the FCs to the contrast of the cluster. More importantly, these statistical methods are designed to test whether there is a difference in a certain statistic (typically mean) between two clusters. Therefore, these methods are not suitable for performing exploratory analysis on clusters when we do not know their characteristics beforehand.

Limitations. Since we use cPCA, we will need to address its limitations in terms of time and space complexity for a large scale problem. Similar to the classical PCA, cPCA computes the covariance matrices and then performs EVD. For a fixed α, it has the same time and space complexity with PCA, which are $O(d^2n+d^3)$ and $O(d^2)$, respectively, where n is the number of data points and d is the number of features. Thus, cPCA can achieve fast computation for a dataset which has a large n, but not for a dataset with a large d (we include the experimental results in the Supplementary Materials [1]). For PCA, incremental algorithms [55, 60, 73] have been developed to solve this issue. For example, the algorithm in [60] has the time and space complexity of $O(dm^2)$ and $O(d(k+m))$, respectively, where m is the number of data points used in each batch, and k is the number of principal components. We thus plan to develop an incremental version of cPCA next.

8 Conclusions

Dimensionality reduction is widely used to analyze high dimensional data for pattern discovery and real-world problem-solving. Our work makes a tangible contribution to interpreting and understanding DR results by introducing a visual analytics method that capitalizes on contrastive learning. Using a scalable visualization, the method directs the user to the essential features within the data. Our work, thus, further enhances the usability of DR methods.

Acknowledgments

The authors wish to thank Suyun Bae (suybae@ucdavis.edu) of VIDI Labs at the University of California, Davis, for her assistance in improving the clarity of the paper content. This research is sponsored in part by the U.S. National Science Foundation through grants IIS-1528203 and IIS-1741536.

REFERENCES

[1] The experimental results, prototype system, source code, and preprocessed datasets. https://takanori-fujiwara.github.io/s/dr-cl/.

[2] The original cPCA implementation. https://github.com/abidlabs/contrastive. Accessed: 2019-3-6.

[3] H. Abdi and D. Valentin. Multiple correspondence analysis. *Encyclopedia of Measurement and Statistics*, pp. 651–657, 2007.

[4] A. Abid, M. J. Zhang, V. K. Bagaria, and J. Zou. Contrastive principal component analysis. *arXiv preprint arXiv:1709.06716*, 2017.

[5] A. Abid, M. J. Zhang, V. K. Bagaria, and J. Zou. Exploring patterns enriched in a dataset with contrastive principal component analysis. *Nature Communications*, 9(1):2134, 2018.

[6] A. Abid and J. Zou. Contrastive variational autoencoder enhances salient features. *arXiv preprint arXiv:1902.04601*, 2019.

[7] E. Acuna and C. Rodriguez. The treatment of missing values and its effect on classifier accuracy. In *Classification, Clustering, and Data Mining Applications*, pp. 639–647. Springer, 2004.

[8] M. Ankerst, M. M. Breunig, H.-P. Kriegel, and J. Sander. OPTICS: Ordering points to identify the clustering structure. In *Proceedings of ACM SIGMOD International Conference on Management of Data*, pp. 49–60, 1999.

[9] Z. Bar-Joseph, D. K. Gifford, and T. S. Jaakkola. Fast optimal leaf ordering for hierarchical clustering. *Bioinformatics*, 17(1):S22–S29, 2001.

[10] M. Behrisch, B. Bach, N. Henry Riche, T. Schreck, and J.-D. Fekete. Matrix reordering methods for table and network visualization. *Computer Graphics Forum*, 35(3):693–716, 2016.

[11] M. Bostock, V. Ogievetsky, and J. Heer. D^3 data-driven documents. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2301–2309, 2011.

[12] M. Brehmer, M. Sedlmair, S. Ingram, and T. Munzner. Visualizing dimensionally-reduced data: Interviews with analysts and a characterization of task sequences. In *Proceedings of Workshop on Beyond Time and Errors: Novel Evaluation Methods for Visualization*, pp. 1–8, 2014.

[13] R. Bro, E. Acar, and T. G. Kolda. Resolving the sign ambiguity in the singular value decomposition. *Journal of Chemometrics*, 22(2):135–140, 2008.

[14] B. Broeksema, A. C. Telea, and T. Baudel. Visual analysis of multi-dimensional categorical data sets. *Computer Graphics Forum*, 32(8):158–169, 2013.

[15] R. J. G. B. Campello, D. Moulavi, and J. Sander. Density-based clustering based on hierarchical density estimates. In *Proceedings of Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pp. 160–172. Springer, 2013.

[16] K. Chang. Nutrient explorer. http://bl.ocks.org/syntagmatic/raw/3150059/, 2012. Accessed: 2019-3-7.

[17] E. Czaplicki. Elm. https://elm-lang.org/. Accessed: 2019-3-7.

[18] Ç. Demiralp. Clustrophile: A tool for visual clustering analysis. *arXiv preprint arXiv:1710.02173*, 2017.

[19] A.-H. Dirie, A. Abid, and J. Zou. Contrastive multivariate singular spectrum analysis. *arXiv preprint arXiv:1810.13317*, 2018.

[20] M. Dowling, J. Wenskovitch, J. Fry, L. House, and C. North. SIRIUS: Dual, symmetric, interactive dimension reductions. *IEEE Transactions on Visualization and Computer Graphics*, 25(1):172–182, 2018.

[21] D. Dua and C. Graff. UCI machine learning repository. http://archive.ics.uci.edu/ml, 2019.

[22] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of International Conference on Knowledge Discovery and Data Mining*, pp. 226–231, 1996.

[23] T. Fujiwara, J.-K. Chou, Shilpika, P. Xu, L. Ren, and K.-L. Ma. An incremental dimensionality reduction method for visualizing streaming multidimensional data. *arXiv preprint arXiv:1905.04000*, 2019.

[24] S. Garte. The role of ethnicity in cancer susceptibility gene polymorphisms: The example of CYP1A1. *Carcinogenesis*, 19(8):1329–1332, 1998.

[25] R. Ge and J. Zou. Rich component analysis. In *Proceedings of International Conference on Machine Learning*, pp. 1502–1510, 2016.

[26] G. Guennebaud, B. Jacob, et al. Eigen v3. http://eigen.tuxfamily.org, 2010. Accessed: 2019-3-6.

[27] J. A. Hartigan. Printer graphics for clustering. *Journal of Statistical Computation and Simulation*, 4(3):187–213, 1975.

[28] J. A. Hartigan and M. A. Wong. A k-means clustering algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 28(1):100–108, 1979.

[29] T. Hastie and R. Tibshirani. Discriminant analysis by gaussian mixtures. *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 155–176, 1996.

[30] C. Higuera, K. J. Gardiner, and K. J. Cios. Self-organizing feature maps identify proteins critical to learning in a mouse model of down syndrome. *PloS one*, 10(6):e0129126, 2015.

[31] T. Höllt, N. Pezzotti, V. van Unen, F. Koning, B. P. Lelieveldt, and A. Vilanova. CyteGuide: Visual guidance for hierarchical single-cell analysis. *IEEE Transactions on Visualization and Computer Graphics*, 24(1):739–748, 2018.

[32] H. Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, 24(6):417, 1933.

[33] A. Inselberg and B. Dimsdale. Parallel coordinates: a tool for visualizing multi-dimensional geometry. In *Proceedings of IEEE Conference on Visualization*, pp. 361–378, 1990.

[34] A. J. Izenman. Linear discriminant analysis. In *Modern Multivariate Statistical Techniques*, pp. 237–280. Springer, 2013.

[35] D. H. Jeong, C. Ziemkiewicz, W. Ribarsky, and R. Chang. Understanding principal component analysis using a visual analytics tool. Technical report, UNC Charlotte, 2009.

[36] P. Joia, F. Petronetto, and L. G. Nonato. Uncovering representative groups in multidimensional projections. *Computer Graphics Forum*, 34(3):281–290, 2015.

[37] I. T. Jolliffe. Principal component analysis and factor analysis. In *Principal Component Analysis*, pp. 115–128. Springer, 1986.

[38] E. Kandogan. Star coordinates: A multi-dimensional visualization technique with uniform treatment of dimensions. In *Proceedings of IEEE Information Visualization Symposium*, pp. 9–12, 2000.

[39] H. Kim, J. Choo, C. Lee, H. Lee, C. K. Reddy, and H. Park. PIVE: per-iteration visualization environment for real-time interactions with dimension reduction and clustering. In *Proceedings of AAAI Conference on Artificial Intelligence*, pp. 1001–1009, 2017.

[40] H.-P. Kriegel, P. Kröger, J. Sander, and A. Zimek. Density-based clustering. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 1(3):231–240, 2011.

[41] S. Kullback and R. A. Leibler. On information and sufficiency. *Annals of Mathematical Statistics*, 22(1):79–86, 1951.

[42] B. C. Kwon, B. Eysenbach, J. Verma, K. Ng, C. De Filippi, W. F. Stewart, and A. Perer. Clustervision: Visual supervision of unsupervised clustering. *IEEE Transactions on Visualization and Computer Graphics*, 24(1):142–151, 2018.

[43] B. C. Kwon, H. Kim, E. Wall, J. Choo, H. Park, and A. Endert. AxiSketcher: Interactive nonlinear axis mapping of visualizations through user drawings. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):221–230, 2016.

[44] C. Lai, Y. Zhao, and X. Yuan. Exploring high-dimensional data through locally enhanced projections. *Journal of Visual Languages and Computing*, 48:144–156, 2018.

[45] Y. LeCun, C. Cortes, and C. J.C. Burges. The MNIST database of handwritten digits. http://yann.lecun.com/exdb/mnist/, 1999. Accessed: 2019-3-7.

[46] S. Liu, W. Cui, Y. Wu, and M. Liu. A survey on information visualization: Recent advances and challenges. *The Visual Computer*, 30(12):1373–1393, 2014.

[47] S. Liu, D. Maljovec, B. Wang, P.-T. Bremer, and V. Pascucci. Visualizing high-dimensional data: Advances in the past decade. *IEEE Transactions on Visualization and Computer Graphics*, 23(3):1249–1268, 2017.

[48] W. E. Marcílio, D. M. Eler, and R. E. Garcia. An approach to perform local analysis on multidimensional projection. In *Proceedings of Conference on Graphics, Patterns and Images*, pp. 351–358. IEEE, 2017.

[49] M. Marusteri and V. Bacarea. Comparing groups for statistical differences: How to choose the right statistical test? *Biochemia Medica*, 20(1):15–32, 2010.

[50] L. McInnes, J. Healy, and J. Melville. UMAP: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018.

[51] S. Mika, G. Ratsch, J. Weston, B. Scholkopf, and K.-R. Mullers. Fisher discriminant analysis with kernels. In *Neural Networks for Signal Processing IX: Proceedings of IEEE Signal Processing Society Workshop*, pp. 41–48, 1999.

[52] D. Müllner. Modern hierarchical, agglomerative clustering algorithms. *arXiv preprint arXiv:1109.2378*, 2011.

[53] A. Y. Ng, M. I. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In *Proceedings of Advances in Neural Information Processing Systems*, pp. 849–856, 2002.

[54] L. G. Nonato and M. Aupetit. Multidimensional projection for visual analytics: Linking techniques with distortions, tasks, and layout enrichment. *IEEE Transactions on Visualization and Computer Graphics*, 2018.

[55] E. Oja and J. Karhunen. On stochastic approximation of the eigenvectors and eigenvalues of the expectation of a random matrix. *Journal of Mathematical Analysis and Applications*, 106(1):69–84, 1985.

[56] L. Pagliosa, P. Pagliosa, and L. G. Nonato. Understanding attribute variability in multidimensional projections. In *Proceedings of Conference on Graphics, Patterns and Images*, pp. 297–304. IEEE, 2016.

[57] M. Parimala, D. Lopez, and N. Senthilkumar. A survey on density based clustering algorithms for mining large spatial databases. *International Journal of Advanced Science and Technology*, 31(1):59–66, 2011.

[58] P. E. Rauber, S. G. Fadel, A. X. Falcao, and A. C. Telea. Visualizing the hidden activity of artificial neural networks. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):101–110, 2017.

[59] M. Redmond and A. Baveja. A data-driven software tool for enabling cooperative information sharing among police departments. *European Journal of Operational Research*, 141(3):660–678, 2002.

[60] D. A. Ross, J. Lim, R.-S. Lin, and M.-H. Yang. Incremental learning for robust visual tracking. *International Journal of Computer Vision*, 77(1-3):125–141, 2008.

[61] D. Sacha, L. Zhang, M. Sedlmair, J. A. Lee, J. Peltonen, D. Weiskopf, S. C. North, and D. A. Keim. Visual interaction with dimensionality reduction: A structured literature analysis. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):241–250, 2017.

[62] D. W. Scott. On optimal and data-based histograms. *Biometrika*, 66(3):605–610, 1979.

[63] J. Stahnke, M. Dörk, B. Müller, and A. Thom. Probing projections: Interaction techniques for interpreting arrangements and errors of dimensionality reductions. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):629–638, 2016.

[64] M. J. Swain and D. H. Ballard. Color indexing. *International Journal of Computer Vision*, 7(1):11–32, 1991.

[65] J. Tang, J. Liu, M. Zhang, and Q. Mei. Visualizing large-scale and high-dimensional data. In *Proceedings of International Conference on World Wide Web*, pp. 287–297. International World Wide Web Conferences Steering Committee, 2016.

[66] The USDA Nutrient Data Laboratory, the Food and Nutrition Information Center, and Information Systems Division of the National Agricultural Library. USDA food composition databases. `https://ndb.nal.usda.gov/ndb/`, 2018. Accessed: 2019-3-7.

[67] W. S. Torgerson. Multidimensional scaling: I. theory and method. *Psychometrika*, 17(4):401–419, 1952.

[68] F. S. Tsai. Dimensionality reduction techniques for blog visualization. *Expert Systems with Applications*, 38(3):2766–2773, 2011.

[69] C. Turkay, A. Lundervold, A. J. Lundervold, and H. Hauser. Representative factor generation for the interactive visual analysis of high-dimensional data. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2621–2630, 2012.

[70] L. van der Maaten. Accelerating t-SNE using tree-based algorithms. *Journal of Machine Learning Research*, 15(1):3221–3245, 2014.

[71] L. van der Maaten and G. Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9(Nov):2579–2605, 2008.

[72] Y. Wang, J. Li, F. Nie, H. Theisel, M. Gong, and D. J. Lehmann. Linear discriminative star coordinates for exploring class and cluster separation of high dimensional data. *Computer Graphics Forum*, 36(3):401–410, 2017.

[73] J. Weng, Y. Zhang, and W.-S. Hwang. Candid covariance-free incremental principal component analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(8):1034–1040, 2003.

[74] J. Wenskovitch, I. Crandell, N. Ramakrishnan, L. House, and C. North. Towards a systematic combination of dimension reduction and clustering in visual analytics. *IEEE Transactions on Visualization and Computer Graphics*, 24(1):131–141, 2017.

[75] J. Wenskovitch, I. Crandell, N. Ramakrishnan, L. House, and C. North. Towards a systematic combination of dimension reduction and clustering in visual analytics. *IEEE Transactions on Visualization and Computer Graphics*, 24(1):131–141, 2018.

[76] R. Xu and D. Wunsch. Survey of clustering algorithms. *IEEE Transactions on Neural Networks*, 16(3):645678, 2005.

[77] J. Y. Zou, D. J. Hsu, D. C. Parkes, and R. P. Adams. Contrastive learning using spectral methods. In *Proceedings of Advances in Neural Information Processing Systems*, pp. 2238–2246, 2013.