


High-dimensional data analysis with subspace comparison using matrix visualization

Information Visualization
2019, Vol. 18(1) 94–109
© The Author(s) 2017
Article reuse guidelines:
sagepub.com/journals-permissions
DOI: 10.1177/1473871617733996
journals.sagepub.com/home/ivi


Junpeng Wang¹, Xiaotong Liu² and Han-Wei Shen¹

Abstract

Due to the intricate relationship between different dimensions of high-dimensional data, subspace analysis is often conducted to decompose dimensions and give prominence to certain subsets of dimensions, i.e. subspaces. Exploring and comparing subspaces are important to reveal the underlying features of subspaces, as well as to portray the characteristics of individual dimensions. To date, most of the existing high-dimensional data exploration and analysis approaches rely on dimensionality reduction algorithms (e.g. principal component analysis and multi-dimensional scaling) to project high-dimensional data, or their subspaces, to two-dimensional space and employ scatterplots for visualization. However, the dimensionality reduction algorithms are sometimes difficult to fine-tune and scatterplots are not effective for comparative visualization, making subspace comparison hard to perform. In this article, we aggregate high-dimensional data or their subspaces by computing pair-wise distances between all data items and showing the distances with matrix visualizations to present the original high-dimensional data or subspaces. Our approach enables effective visual comparisons among subspaces, which allows users to further investigate the characteristics of individual dimensions by studying their behaviors in similar subspaces. Through subspace comparisons, we identify dominant, similar, and conforming dimensions in different subspace contexts of synthetic and real-world high-dimensional data sets. Additionally, we present a prototype that integrates parallel coordinates plot and matrix visualization for high-dimensional data exploration and incremental dimensionality analysis, which also allows users to further validate the dimension characterization results derived from the subspace comparisons.

Keywords

High-dimensional data, matrix visualization, subspace comparison

Introduction

High-dimensional (HD) data are comprised of numerous dimensions and usually contain a large number of data items, which often overwhelm analysts and stop them from in-depth analysis. Different dimensions, as the building blocks of a HD data, demonstrate different facets of the data. Understanding them and the relationship between them will provide insight into the complex HD data.^{1,2} For example, analyzing dimensions that dominate the distribution of HD data items can help to provide a quick intuition of the overall data

trend; knowing what dimensions are similar to each other can draw inference on other dimensions when only knowing one of them; being aware of dimensions that conform to the patterns formed by other

¹The Ohio State University, Columbus, OH, USA

²IBM Research–Almaden, San Jose, CA, USA

Corresponding author:

Junpeng Wang, The Ohio State University, 395 Dreese Laboratories, 2015 Neil Avenue, Columbus, OH 43210, USA.
Email: wang.7665@osu.edu

dimensions can help to guide subspace selection in certain scenarios. Characteristics of dimensions can be reflected by data patterns,³ i.e. relationship among data items (such as cluster, distribution, and relative distance), in the space formed by the corresponding dimensions. However, precisely portraying the characteristics of individual dimensions in a HD data is notoriously difficult, as only sporadic data patterns can be found in space with higher dimensionality. This is also known as the “Curse of Dimensionality” problem,⁴ i.e. the volume of space grows exponentially while the data become relatively more sparse in the HD space, which makes data patterns harder to find and characteristics of dimensions vaguer. For example, a data cluster may only reside in a space constituted by a certain combination of dimensions (a subspace). Considering all dimensions together will dilute the contributions from these dimensions and make it more difficult to discover the cluster. Even if a prominent data pattern can be found in a HD space, it is hard to find out which dimensions have contributed to it, as dimensions are mixed together and treated equally in the HD data.

To address the aforementioned problem, researchers have tried to decompose HD data and study them in the context of different subspaces.⁵ On one hand, data patterns in subspaces are easier to be identified; on the other hand, instead of treating all dimensions equally, higher priority can be given to certain dimension or dimension combinations (subspaces) according to users’ analytic questions. In addition, we found that comparing data patterns in different subspaces can help to deduce the characteristics of individual dimensions in the subspaces. For example, considering a five-dimensional (5D) data, if the subspace formed by dimensions 1, 2, and 3 is similar to the subspace formed by dimensions 1, 2, and 4, then we are prone to believe that dimensions 3 and 4 are similar dimensions, since they play similar roles in similar subspaces. Many metrics can be used to quantify the similarity between subspaces. However, to the best of our knowledge, little work has been conducted to effectively visualize the similarity, and thus visually compare subspaces.

Most of the existing subspace analysis and visualization approaches^{6–9} resort to algorithms of dimensionality reduction to project HD data or the subspaces of them to two dimensions and visualize them with scatterplots, which are sometimes effective in revealing clusters. Typical dimensionality reduction algorithms include principal component analysis (PCA),¹⁰ multi-dimensional scaling (MDS),¹¹ and t-distributed stochastic neighbor embedding (t-SNE).¹² However, projecting HD data to two-dimensional (2D) space will not always produce stable results, making scatterplot visualization difficult to track and compare across

subspaces. For example, considering a data set with 10 dimensions, the result of projecting all 10 dimensions to a 2D space using MDS may be very different from projecting only nine dimensions. Even with similar dimensionality reduction results, one can hardly track data items (points) in different scatterplots and visually compare them. In other words, the dimensionality reduction results cannot always reflect the subspace similarity, and scatterplots are typically not very effective for comparative visualization.

In this article, we propose a dimension aggregation-based matrix visualization approach to effectively visualize and compare subspaces. Our approach presents a subspace with a matrix view, in which each matrix element represents a pair-wise distance value between two data items in the subspace. Multiple subspaces are presented with multiple small juxtaposed matrix views, and the consistent row/column order across matrices makes the comparison among them available. The *k*-means clustering algorithm has been employed to optimize the order of rows/columns in matrix visualizations to regularize and enhance data patterns in a subspace. Based on the comparison results, we group similar subspaces using hierarchical clustering. Studying on the dimension combinations in separate but similar subspaces allows us to derive dimension characteristics (like dominant, similar and conforming) with automatic algorithms. To make our approach comprehensive, we further include a dimension characteristics verification process, which is conducted through an incremental dimensionality analysis prototype. The prototype integrates parallel coordinates plot (PCP) and matrix views to present progressive dimension aggregation results with a history of dimension exploration. It allows users to iteratively explore and intuitively interact with different dimensions and dimension combinations (subspaces), which further extends users’ ability of analyzing HD data. In summary, the contributions of this article are twofold:

1. We propose a dimension aggregation and subspace visualization approach based on matrix views for HD data, which enables effective subspace comparisons.
2. We analyze dimension characteristics based on subspace similarity and present an incremental dimensionality analysis solution to verify and adjust the characterization results.

Related work

HD data analysis and visualization

In general, the existing HD data visualization approaches can be categorized into two major groups.

The first group directly visualizes all dimensions of HD data by sacrificing space, such as PCPs,^{13,14} scatterplot matrices,¹⁵ table lens,¹⁶ and radial-layout visualizations.¹⁷ Although limited by their scalability and criticized by the visual clutter problem, these approaches have been used extensively for correlation analysis on data with a small number of dimensions. Conversely, the second group relies on dimensionality reduction algorithms, such as PCA,¹⁰ MDS,¹¹ and t-SNE,¹² to project HD data to manageable lower dimensions and visualize them with scatterplots.^{8,18} The drawbacks for this group, especially for the non-linear dimensionality reduction algorithms, are the hardly controllable reduction process and the barely interpretable results. Many existing works^{7,19,20} have successfully combined these two groups of approaches to analyze HD data with multiple linked views, in which different types of interactions are enabled to facilitate users' exploration. In addition, researchers have tried to analyze the characteristics of dimensions in HD data from many different perspectives, such as uncertainty,²¹ sensitivity,²² associativity,² and interest-^{1,7} ingness. Most of these works describe the property of dimensions when considering all dimensions of the HD data. We derive the characteristics of dimensions by investigating their behavior in separate but similar subspaces. From the existing literature, we have also found many visualization works that focus on subspace analysis.^{6,8,9,23,24} Different from them, our focus in this article is to provide dimension insights for HD data using effective visual comparisons between subspaces. We target on distinguishing the dominant, similar, and conforming dimensions in different subspace contexts in this work and hope the results we presented would inspire more research on dimension characterization based on subspace analysis.

Subspace analysis and visualization

Due to the "Curse of Dimensionality" problem,⁴ analyzing HD data through subspace analysis has attracted consistently increasing attention. For HD data, the number of subspaces scales exponentially with the number of dimensions. To handle the large number of subspaces, numerous subspace clustering algorithms^{3,5} have been proposed, such as CLIQUE,²⁵ ENCLUS,²⁶ PROCLUS,²⁷ and DOC.²⁸ Features (like the number and size) of clusters in different subspaces can be used to rank them and filter out less interesting subspaces. In the context of visualization, subspace visualization is an emerging field. Ferdosi et al.⁶ ranked subspaces in astronomical data using connected morphological operators and provided linked visualizations for interactive subspace exploration. Tatu et al.⁷ applied SURFING²⁹ for subspace

search and compared subspaces based on topological/dimensional similarity to mitigate the redundancy problem in their subspace exploration framework. Yuan et al.⁸ proposed dimension projection matrix/tree to analyze HD data in a hierarchical manner, which enables simultaneous data and dimension correlations exploration. Watanabe et al.²³ conducted feature subspace mining by bi-clustering. Liu et al.²⁴ proposed an interactive framework to find meaningful low-dimensional structures in HD data by dynamic projections and view transition graphs. Zhou et al.⁹ enhanced cluster structures in subspaces by reconstructing new dimensions from projected 2D data. Our focus in this work is to provide effective visualization to visually compare subspaces and investigate dimension characteristics using subspaces' similarity.

Matrix visualization

A matrix organizes an array of elements in a rectangular layout with certain numbers of rows and columns. The matrix elements can be numerical values, color squares, or even plots (such as scatterplot matrices¹⁵). Matrix visualization with color squares as elements is more commonly known as an alternative to the traditional node-link diagram for graph representations^{30,31} due to its merits of eliminating node-overlapping and line-crossing.³²⁻³⁴ The adjacency matrix of a graph will be symmetric if the graph is undirected. Previous studies³⁵ have shown that symmetric square matrices and triangular matrices present no noticeable difference in the effectiveness of comparative visualization. Different juxtaposition layouts of multiple triangular matrices can also enhance comparative visualization.³⁵ Matrix visualizations have also been used to show pair-wise distance between data items. If the distance metric used in this case is symmetric, i.e. the distance from item *a* to item *b* is the same with the distance from *b* to *a*, the resulting matrix will be symmetric and a triangular matrix (instead of a symmetric square matrix) would be enough to present all distance information. Multiple such matrices, one for each variable, can be presented in juxtaposition to show different facets of a complex multivariate data set.^{35,36} In this work, we use one matrix to present one dimension or one aggregated subspace (multiple dimensions) of HD data sets.

Problem and approach overview

In this section, we elaborate the reasons why matrix visualization is more effective than scatterplot visualization in comparing subspaces. Following that, we provide an overview of our approach for dimension characteristics analysis by subspace comparison.

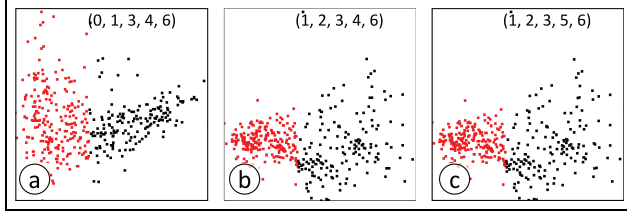


Figure 1. Subspaces in (a) and (b) are more similar when compared with the subspace in (c). Scatterplots, though present the cluster information, cannot reflect subspaces' similarity. Two colors are used for two groups of data items across subspaces.

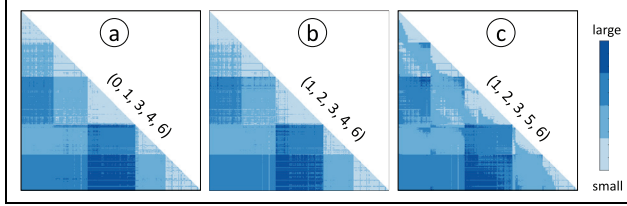


Figure 2. Matrix visualization reveals the similarity between subspaces (a) and (b), as well as the dissimilarity between (b) and (c).

Visualization problem

A subspace of a HD data set contains all data items but only a subset of the original dimensions. Considering a 5D data, one possible subspace could be the space formed by dimensions 1, 2, and 3 (denoted as (1, 2, 3)). The major problem we want to address in this article is to provide an effective visualization for intuitive visual comparison of subspaces, which can be detailed by answering two questions: (1) how to visually present a subspace (visualization question) and (2) how to precisely quantify the similarity between two subspaces (similarity measurement question). These two questions are often considered separately in the existing literature, which leads to the situation that visualizations disagree with similarity measurements. Specifically, HD subspaces are usually transferred to 2D spaces and visualized with scatterplots, but their similarities are often measured in the original HD space. For example, existing works, such as Tatu et al.,⁷ took samples from subspaces and measured subspace similarity by comparing the k -nearest neighbors of those samples in their respective subspace. However, it turns out that subspaces with similar data topology can have dissimilar scatterplot visualizations, which results in the conflict that visualizations disagree with similarity measurements.

The conflict can be demonstrated by the following example. Figure 1 shows the MDS projection results of three subspaces from a real-world data (the Car

data, details about this data set can be found in the “Case studies” section). Similar to Tatu et al.,⁷ we measured the similarity between subspaces based on subspaces' topology in this work. According to this similarity quantification (explained in the “Comparing and clustering subspaces” section), subspaces (0, 1, 3, 4, 6) and (1, 2, 3, 4, 6) (shown in Figure 1(a) and (b)) are more similar to each other when compared with subspace (1, 2, 3, 5, 6) (shown in Figure 1(c)), but the visual appearances of their scatterplot cannot reflect the similarity. Conversely, the scatterplots in Figure 1(b) and (c) look more similar, though the corresponding subspaces have dissimilar topologies (our supplementary materials contain the MATLAB code for repeating this experiment). Note that the term topology is a broad concept that has been used in different contexts to describe different things, such as the connectivity of graphs and the critical points of smooth functions. In this work, it is specifically used to indicate the spatial distribution and the structure of data items in a subspace, which is consistent with the term used by Tatu et al.⁷

To resolve the aforementioned conflict, we present subspaces with matrix visualizations, in which the consistent row/column order across matrices makes two subspaces comparable. We aggregate dimensions of HD data items and present them with lower dimensional distance values in matrix visualizations, i.e. each matrix element presents the distance between pairs of data items in the corresponding subspace. Euclidean distance is used for aggregation (values in each dimension have been normalized before aggregation) since the focus of comparison is the topology of subspaces, i.e. the distance between two data items $a = \{a_1, a_2, \dots, a_n\}$ and $b = \{b_1, b_2, \dots, b_n\}$ in an n -dimensional space is $\sqrt{\sum_{i=1}^n (a_i - b_i)^2}$. To make a matrix pattern easily recognizable, certain heuristics (e.g. clustering) can be used to optimize the orders of rows/columns, since they (the rows/columns) are order-irrelevant. Figure 2 shows the matrix visualizations of the same three subspaces in Figure 1. Since the Euclidean distance is non-directional, the matrices are symmetric and only lower left parts are shown. From the figure, we can see subspace (0, 1, 3, 4, 6) and (1, 2, 3, 4, 6), shown in Figure 2(a) and (b), are more similar to each other when compared with subspace (1, 2, 3, 5, 6) (shown in Figure 2(c)). Subspace (1, 2, 3, 5, 6) also has some similar data patterns to the other two subspaces due to its dimension overlap with them, i.e. it is not completely dissimilar to the other two.

Approach overview

With the effective subspace comparison, we propose a visual analytics approach for HD data to perform

dimension characteristics analysis. The approach has two major steps: (1) identify dimension characteristics by subspace comparison and (2) verify and further investigate dimension characteristics with incremental dimensionality analysis. For the first step, we deduce dimension characteristics by comparing the dimension combinations in similar subspaces. For example, the similarity between the two subspaces in Figure 2(a) and (b) makes us believe that dimension 0 and 2 are similar dimensions, as they are the only difference in the similar subspaces. The second step incrementally studies dimensions by decomposing a subspace into progressive dimension combinations. For example, the progressive combinations for subspace (1, 2, 3, 4, 6) are as follows: (1), (1, 2), (1, 2, 3), (1, 2, 3, 4), and (1, 2, 3, 4, 6). These five combinations (smaller subspaces) will be studied incrementally to reveal the contribution of individual dimensions to the (bigger) subspace.

Figure 3 shows the detailed pipeline of our approach. For a given HD data with a large number of possible subspaces, we first apply a subspace clustering algorithm, i.e. CLIQUE,²⁵ on it to reduce the number of subspaces (by excluding less interesting subspaces from our analysis). After the subspace filtering, a manageable number of subspaces with relatively more prominent data patterns (interesting subspaces) will be visualized with matrices in multiple small juxtaposed views for visual comparison. The hierarchical clustering algorithm, which groups similar subspaces into the same cluster, helps users to focus their comparisons on a small number of similar subspaces. Automatic algorithms can then be applied on those similar subspaces to identify dimensions with different characteristics. These characteristics of dimensions work as a priori-knowledge for further incremental dimensionality analysis, which will also verify the characterization results. The knowledge of each dimension obtained during the subspace comparison and incremental dimensionality analysis will (1) guide users to pick out more subspaces from the possible subspace set and (2) motivate users to test their hypotheses about certain dimensions' characteristics. A new exploration cycle could then start with a new set of interesting subspaces. This iterative process will consistently improve users' understanding about the HD data set.

Subspace comparison and dimension characterization

For an n -dimensional data, the total number of subspaces is $2^n - 1$. There is not a fixed threshold for the number of dimensions (i.e. the value of n), when

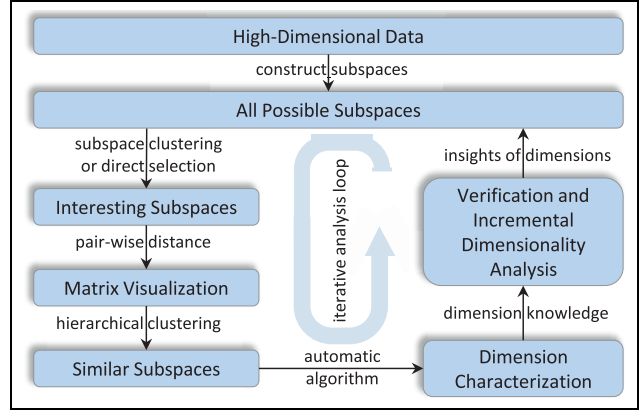


Figure 3. The iterative analysis loop for high-dimensional data.

considering whether a data set is a HD data or not. In this article, we specifically focus on HD data with dimensions up to 20. We show how our proposed approach can apply to data in this scale and discuss potential limitations when extending our approach to data sets of even higher dimensions.

In practice, it is hard to exhaust all subspaces of a HD data. Moreover, with the constraint of human cognitive load, presenting a large number of subspaces at once may overwhelm users. To address the issues, subspace clustering algorithms have been employed in earlier works to give priority to certain subspaces and filter out less interesting ones, so that the large number of subspaces can be reduced to a manageable amount. For instance, Watanabe et al.²³ conducted subspace mining using bi-clustering; Tatu et al.⁷ performed subspace filtering with SURFING.²⁹ We prioritize subspaces according to the cluster density in subspaces since our focus of comparison is the spatial distribution of data items. The CLIQUE algorithm from the ELKI platform^{37,38} is used to perform the subspace filtering. There is no specific reason why we choose CLIQUE; other density-based subspace clustering algorithms would also work.

Subspaces in matrix visualization

Figure 4 shows an overview of our proposed prototype for dimension characteristics analysis using subspace comparison. Our approach presents subspaces with triangular matrices (Figure 4(a)). The element in the i th row and j th column of a matrix represents the Euclidean distance (data values have been normalized in each dimension before the distance calculation) between data items i and j in the subspace represented by the matrix. The blue colors from light to dark encode the pair-wise distance values from small to

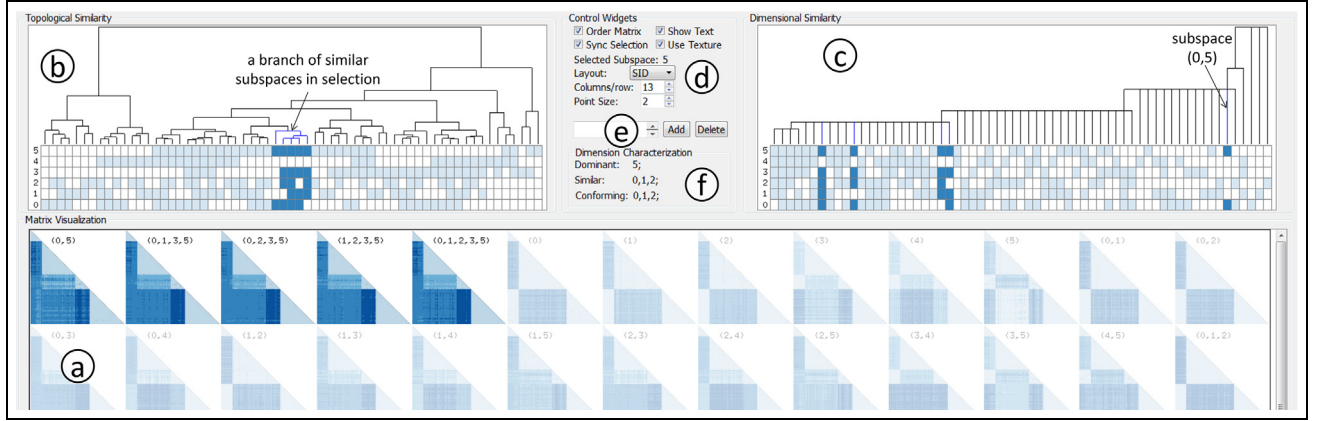


Figure 4. Dimension characteristics analysis prototype: (a) subspaces are presented with lower triangular matrices; (b, c) hierarchical cluster tree reflects topological/dimensional similarity of subspaces; (d) widgets to switch among matrix layouts; (e) subspace searching widgets; (f) widgets to list dimension characteristics. After brushing on a branch of the hierarchical tree in (b), the corresponding five subspaces are highlighted in (a); the tree in (c) also synchronizes the selection; the automatic algorithm derives the dominant, similar and conforming dimensions and they are shown in (f).

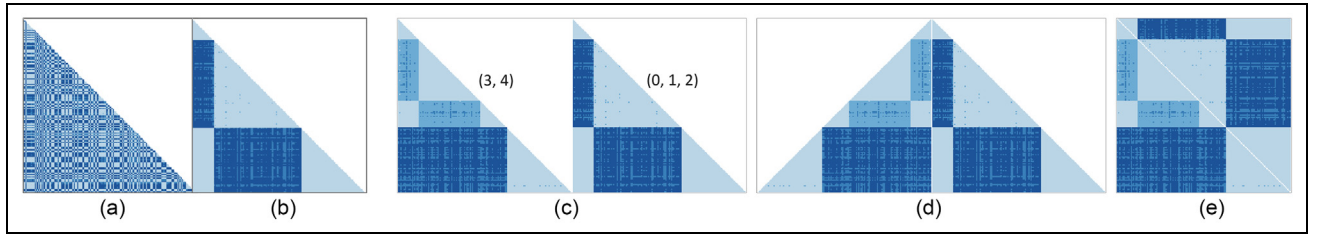


Figure 5. (a, b) Matrix pattern before and after reordering (using k -means clustering algorithm); (c, d, e) three layouts for triangular matrices: side-by-side (SID), back-to-back (BAK) and complementary (CMP).

large, and all matrices share the same color mapping. More color levels can be added on-demand to improve the sensitivity of the visual representation. Our major design rationales of using triangular matrices instead of symmetric square matrices are (1) avoiding the redundancy of symmetric square matrices and (2) allowing different layouts of the matrices to take advantage of humans' symmetric perception in visual comparison (details explained in Figure 5(c)–(e)).

Matrix pattern enhancement. Without rows/columns reordering, data patterns in matrix visualizations are hard to be recognized, as an example shows in Figure 5(a). Such a visualization cannot effectively demonstrate the unique features of a subspace, let alone the comparison between subspaces. In fact, numerous reordering methods^{30,39} have been used in earlier works to reveal regularity and discover data patterns hidden in matrices. Here, we use the k -means algorithm^{40,41} to enhance matrix pattern, as it is simple and sufficient for our work ($k = 5$ throughout the

article, but the value can be adjusted by users during exploration). The algorithm takes the symmetric matrix of a subspace as input and outputs cluster indices for rows of the matrix (rows are considered as observations in k -means). The matrix is then reordered by putting rows with the same cluster indices together. Due to symmetry, columns will also be reordered to have the same order with rows. Figure 5(b) shows the reordering result of the matrix in Figure 5(a). Additionally, different orders may be optimal in different sets of subspaces. Therefore, we allow users to change the row/column order by giving different matrices as inputs to the k -means algorithm. Once the row/column order is decided, it will be applied to all matrices. Consequently, users can visually compare different subspaces by tracking the same spatial region across matrices.

Matrix layouts. Our prototype also provides different layouts (Figure 5(c)–(e)), i.e. side-by-side (SID), back-to-back (BAK), and complementary (CMP), of

triangular matrices for better comparisons between subspaces. The control widgets shown in Figure 4(d) help users to switch among different matrix layouts (please find interaction details in our attached video). This design is inspired by the fact that humans' symmetric perception can be used to enhance comparative visualization, i.e. BAK and CMP outperform SID layout when comparing two matrices, which has been proved in previous statistical studies.³⁵

Comparing and clustering subspaces

Putting multiple views side-by-side for juxtaposed comparison is well known as *small multiples* introduced by Tufte,⁴² and he also suggested that small multiples should be within eye-span for effective comparison. Given the relatively large number of subspaces (even after filtering from CLIQUE), it is hard for users to compare all matrices at the same time. For this reason, we employ hierarchical clustering to group subspaces based on their topological or dimensional similarity. Users can interact with the dendrograms (Figure 4(b) and (c)) to focus on a branch of similar subspaces.

Clustering based on topological similarity. Matrix visualization that demonstrates all pair-wise distances between data items is a compact representation of the data distribution, as well as relative distance between data items, for the corresponding subspace. Therefore, topological similarity can be measured by comparing the matrix visualizations of subspaces. The pixel-wise mean square error (MSE) can be used to quantify the similarity between two images. For accuracy consideration, the original distance values in matrices, instead of pixel values, can also be used, as certain details may not be retained in the limited number of pixels after color mapping (the scalability issue of matrix visualization is discussed in the "Discussion and limitation" section). In this article, we always use the original distance matrix to measure the topological similarity between subspaces. Figure 4(b) shows a hierarchical cluster tree of subspaces based on their topological similarity. The vertical height of each internal node indicates the dissimilarity scale between its two children. Users can select a certain branch of the tree, and similar subspaces in the branch will be highlighted and gathered in a small view range for detailed comparison, as shown in Figure 4(a). The matrix visualization results also provide visual feedback to help users to refine their focus of comparison. For example, if the highlighted subspaces look very different, users may want to choose a child of the currently selected internal node to focus on a smaller but more similar subspace set. The heat map under the dendrogram in Figure 4(b) illustrates dimensions involved in different

subspaces. The horizontal axis of the heat map shows all subspaces; while the vertical axis lists all dimensions. One column in the heat map corresponds to one leaf node of the hierarchical tree (one subspace), and dimensions involved in the subspace are colored with blue squares.

Clustering based on dimensional similarity. Two subspaces are dimensionally similar if they share common dimensions. Similar to Tatu et al.,⁷ we use Tanimoto⁴³ distance to measure the dimensional similarity between subspaces. For example, considering two subspaces, (1, 2, 3, 4) and (1, 2, 3, 5), of a 5D data, the bitmap representations of these two subspaces are 11110 and 11101 (1 for presence and 0 for absence of a dimension). Their Tanimoto distance is the ratio between the number of non-zero bits in (11110 AND 11101) and the number of non-zero bits in (11110 OR 11101), which is 0.6. Based on the Tanimoto distance between all subspaces, we construct another hierarchical tree, as shown in Figure 4(c). Providing a dimensional hierarchical tree is to (1) allow users to efficiently select dimensionally similar subspaces and compare their topological similarity in the matrix views and (2) help users to perceive how topologically similar subspaces are dimensionally similar to each other by comparing two trees. Subspace selection will always be synchronized between the two trees. In addition, the dimensional hierarchical tree always presents regular patterns, i.e. many internal nodes have very similar vertical heights, because many pairs of subspaces have the same Tanimoto distance. The regularity helps users to quickly identify subspaces with dissimilar dimension combinations. For example, in Figure 4(c), from the height of internal nodes, it is very obvious that subspace (0, 5) is dimensionally dissimilar from the other selected subspaces.

In addition to selecting subspaces from the hierarchical trees, users can also directly interact with the triangular matrices (in Figure 4(a)) to add/remove subspaces to/from a/the selection. Additionally, all possible subspaces can be searched out from the search box in Figure 4(e) and added into selection. If the searched subspace is not in the set of interesting subspaces, i.e. it was filtered out by CLIQUE in pre-processing, we will add it back, compute its matrix representation on-the-fly, and update the hierarchical trees. That is to say, although we cannot show all possible subspaces at the beginning, users are still capable of exploring any subspaces.

Dimension characterization

Dimension characteristics can be derived by comparing dimensions involved in the selected similar

subspaces. These characteristics of dimensions are only valid in the context of the selected subspaces and the accuracy of them depends on the subspaces' similarity.

Dominant dimensions are dimensions with unique data patterns. They are essential to form the topology of a subspace. The common dimensions in similar subspaces are candidates for dominant dimensions, as subspaces' similarities are mostly resulted from their dimension overlap. In other words, the common dimensions contribute a lot to the similarity of subspaces. For example, considering one selected subspace with a dominant dimension A , if other selected subspaces want to be similar to this one, then dimension A has to present in them to include the unique features from A . These similar subspaces will make A a dominant dimension since it appears in all of them. In Figure 4(a), the selected subspaces are (0, 5), (0, 1, 3, 5), (0, 2, 3, 5), (1, 2, 3, 5), and (0, 1, 2, 3, 5), and they are similar. The only dimension overlap in these five subspaces is dimension 5, making it the dominant dimension in the context of the selected subspaces. Moreover, dominant dimensions can be easily verified with our incremental dimensionality analysis (details in the "Dimension characteristics verification" section).

Similar dimensions are dimensions that can be replaced with each other but still produce similar matrix patterns. For example, if both subspaces (1, 2, 3, 4) and (1, 2, 3, 5) are in the set of similar subspaces, we will consider dimensions 4 and 5 as similar dimensions since they have the same contribution to form the similar subspaces. Given that users are capable of exploring all subspaces, we choose not to derive any information from subspaces differing more than one dimension. Continuing our previous example, if subspaces (1, 4, 6, 8) and (1, 5, 7, 8) are also in the set of similar subspaces, we will not conclude if dimension 6 is similar to 7 or not since these two subspaces have differences in two dimensions (though we have known dimensions 4 and 5 are similar). Transitivity can be enabled by users when computing the similar dimensions. For example, if users visually confirm that subspaces (1, 2, 3, 4), (1, 2, 3, 5), (1, 4, 7), and (1, 6, 7) are similar, then dimensions 4, 5, and 6 will be similar to each other. The similarity between the first two subspaces indicates dimensions 4 and 5 are similar, whereas the similarity between the last two subspaces indicates dimensions 4 and 6 are similar. Therefore, the transitivity rule merges these two sets of similar dimensions together. In addition, it is possible to have several groups of similar dimensions. Similar dimensions inside the same group will be separated by commas; whereas, different groups will be separated by semicolons, as shown in Figure 4(f).

Conforming dimensions are dimensions that conform to data topology of the selected similar subspaces, which means the presence/absence of them will not significantly affect subspaces' topology. They can be identified if the set of dimensions involved in one subspace is a subset of dimensions involved in another subspace and these two subspaces are similar. For example, if subspaces (1, 3, 5) and (1, 2, 3, 5) are similar, then dimension 2 conforms to the topology of these two subspaces, as the presence/absence of it does not noticeably change the matrix pattern. If subspaces (1, 3, 5) and (1, 2, 3, 5, 6) are similar, then dimensions 2 and 6 together conform to the subspaces' topology. Note that this does not indicate 2 or 6 conforms to the subspaces' topology individually. They may have complementary patterns that were canceled after aggregating them together. Here, we only derive conforming dimensions if the number of dimensions involved in two similar subspaces differs by one. Other dimension characteristics might be able to derived if allowing difference in two dimensions (please find further discussions in the "Discussion and limitation" section).

According to the above descriptions, dimensions with these characteristics can be queried out by automatic algorithms. In our prototype, they are immediately identified after users update the selection of subspaces (Figure 4(f)). Users need to make sure that the selected subspaces are visually similar to each (by adding/removing subspaces to/from a/the selection) to guarantee the accuracy of the automatic algorithm. Additionally, the dimension characteristics derived from subspace comparisons are only valid within the context of selected subspaces instead of all subspaces. Completely relying on the automatic algorithm may lead to imprecise results, as different similarity thresholds will result in different subspace contexts. To make our approach more comprehensive, we feel the need of visually verifying the characterization results and the importance of revealing how individual dimensions affect data topology in a particular subspace. Therefore, we conduct incremental dimensionality analysis to further examine on different dimensions.

Dimension characteristics verification

The characteristics of dimensions in a subspace are verified and further analyzed by investigating how individual dimensions incrementally affect subspace' topological pattern. For example, considering subspace (1, 2, 3), the incremental dimensionality analysis will compare matrices of subspaces (1), (1, 2), and (1, 2, 3) progressively. The dimension order is important here as it decides what subspaces will be involved in the analysis. In our previous example, the dimension

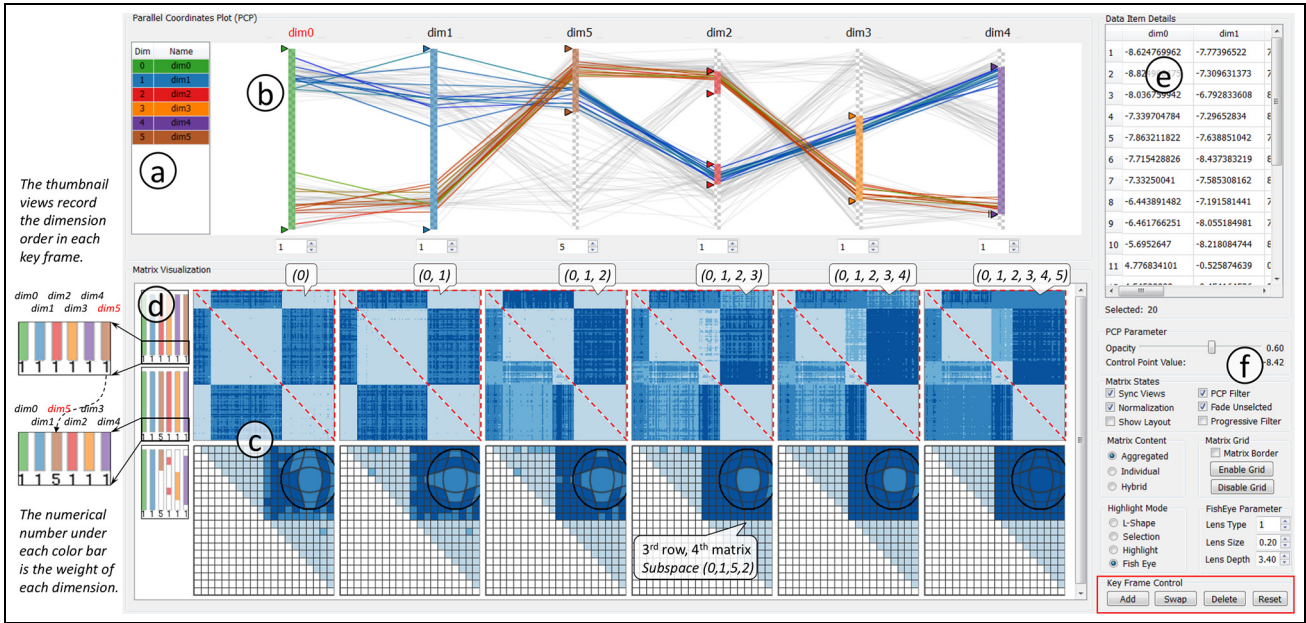


Figure 6. Incremental dimensionality analysis: (a) dimension selection view; (b) PCP; (c) three rows of triangular matrices show three key frames (the first row is in red dashed triangles, the row of lower left triangular matrices is the second; the third row contains less matrix elements due to data filtering); (d) three thumbnail views record the corresponding PCP states; (e) details of selected data; and (f) control widgets.

order is 1, 2, and 3. If we change the order to 1, 3, and 2, subspaces (1), (1, 3), and (1, 3, 2) will be analyzed (subspace (1, 2, 3) and (1, 3, 2) are the same subspace). We also allow users to adjust the weight of each dimension during data aggregation (i.e. computing pair-wise distance using weighted Euclidean distance) to give priority to a certain dimension. As shown in Figure 6, the incremental dimensionality analysis prototype combines a PCP and matrix visualizations. The PCP provides a quick overview of the selected dimensions. It also plays as the interface for users to interact with the current set of selected dimensions (i.e. changing the order, weight, and data range of dimensions). Dimensions can be loaded into PCP via the dimension selection view (Figure 6(a)) and the color of each dimension (row) in this view is the same with the color of corresponding PCP axis. In Figure 6(a), we loaded all six dimensions of our synthetic data into PCP for the purpose of demonstrating the data. However, loading a subset of dimensions based on users' focus of studies or loading dimensions involved in a particular subspace are more common cases. The PCP (Figure 6(b)) axes order is the dimension order of a subspace, which decides what subspaces will be involved in the incremental analysis. The matrix views under the PCP show both data exploration history and incremental dimension aggregation result.

Data exploration history

Users explore loaded dimensions with interactions including (1) changing dimension aggregation order by swapping PCP axes, (2) adjusting the weight of each dimension during aggregation by interacting with the spin box under each PCP axis, and (3) filtering data items on each dimension. These interactions change the states of PCP and matrix views and move the exploration process forward. Some important steps in the exploration process are called *key frames* in our work and they are recorded by rows of triangular matrices. In Figure 6(c), three rows of triangular matrices (from top to bottom) record three key temporal exploration steps, and the states of PCP in these steps are captured in thumbnail views (Figure 6(d)) on the left of each row. The color bars in the thumbnail views represent the PCP axes in the same color. Certain intervals of the color bars may become white, which indicates the value ranges have been filtered out. The numerical value under each color bar indicates the weight of each PCP axis (dimension) during data aggregation in the corresponding key frame. According to these thumbnail views, the first row of triangular matrices (top right triangles in red dashed lines) shows all data items and the dimension order is *dim0*, *dim1*, *dim2*, *dim3*, *dim4*, and *dim5*. The second row (lower left triangles) also shows all data items but

the weight of *dim5* (the gray bar) changes from 1 to 5 and the dimension order has been modified. From the second row to the third, certain data range(s) are filtered out on *dim5*, *dim2*, *dim3*, and *dim4*. The history recorded in the matrix and thumbnail views helps users to recall what aggregation orders and what value ranges have been explored, which avoids duplicating previous works. Here, we put matrices in CMP layout, i.e. the row of top right triangles records the key frame before the frame recorded in the bottom left row, to save space and enhance comparison between consecutive key frames. However, users can change the layout based on their preference or specific exploration goals.

Incremental dimension aggregation

Inside each row of the triangular matrices, dimensions are aggregated progressively, i.e. the n th triangular matrix shows the pair-wise distance between data items when considering the first n dimensions from the left. For example, the fourth triangular matrix on the third row shows the distance between data items when considering four dimensions: *dim0*, *dim1*, *dim5*, and *dim2* (Figure 6(c)). Therefore, by watching one row of matrices, we know how each dimension incrementally affects the matrix pattern. The third row shows the current exploration state and the matrices contain less elements than the matrices in previous two rows because of the data filtering shown in Figure 6(b), as well as the third thumbnail view in Figure 6(d). The data filtering allows users to focus on particular value range(s) of each dimension and compare the data pattern in matrix views. Different rows of matrices (key frames) may have different value ranges due to the filtering. So, before color mapping, a normalization process is applied inside each row to fully utilize the color span. Double click on a matrix view will trigger an execution of the k -means algorithm by taking the clicked matrix as input. The clicked matrix view, and also other matrix views, will then be reordered based on the output from the k -means. This interaction enables users to flexibly change matrices' appearance according to topology in a particular subspace. By default, the contents of triangular matrices are the incremental dimension aggregation results. They can also be the data patterns of individual dimensions (pair-wise distances between data items when considering only one dimension). The matrix contents, as well as matrix layouts, can be flexibly changed by users to assist their visual comparisons. For example, one can put the aggregated result on top right triangles and the individual dimension result on bottom left ones for complementary comparison. Figure 8 (in the "Case studies" section) shows an example of both individual and

aggregated results in two rows of matrices, with both SID (rows) and BAK (columns) layouts.

Users can interact with polylines (in PCP) or matrix elements to explore individual data items. Details of selected items will be presented in the table view shown in Figure 6(e). Under the data table, we provide multiple control widgets to help users control system states and record exploration steps (Figure 6(f)). For example, the *Key Frame Control* widgets (in the red box) allow users to add/delete key frames (rows of triangular matrices) or move two key frames closer (for better comparison) by swapping different rows. The *Fish Eye* highlighting option provides users with *focus + context* visualization and enables them to track details of certain spatial regions in different matrices, as shown in Figure 6(c). Direct interactions with the triangular matrices are also available to zoom into particular matrix regions. We demonstrate more interaction details with this prototype in the "Case studies" section with different HD data sets (details can also be found from the associated video).

Case studies

In this section, we demonstrate the effectiveness and usefulness of our subspace comparison and incremental dimensionality analysis prototypes with synthetic and real-world HD data.

Synthetic data

We generated a synthetic data set, with six dimensions and 150 data items, using Gaussian mixture model,⁴⁴ to verify the correctness of our analysis approach. Dimensions 0 (*dim0*), 1 (*dim1*), and 2 (*dim2*) have two clusters; dimensions 3 (*dim3*), 4 (*dim4*), and 5 (*dim5*) have three clusters. With different mean and variance values, we can make the cluster structure clear or vague in each dimension. The PCP in Figure 6(b) shows the data items, as well as the cluster information in each dimension. It is obvious that *dim0*, *dim1*, and *dim2* have clear and similar cluster structures (two clusters appear in all the three dimensions and the relative distance between data items inside each cluster, as well as between clusters, is similar after normalization). The similarity can also be found in Figure 7. The relative distances among three clusters in *dim3*, *dim4*, and *dim5* are set to be different on purpose, so that the matrix visualizations can present different data patterns, as shown in Figure 7.

For this synthetic data, 63 subspaces were generated. Since the number of subspaces was not very large, we did not filter them with CLIQUE. The results shown in Figure 4 are from this data. Different branches of the topological hierarchical tree (Figure

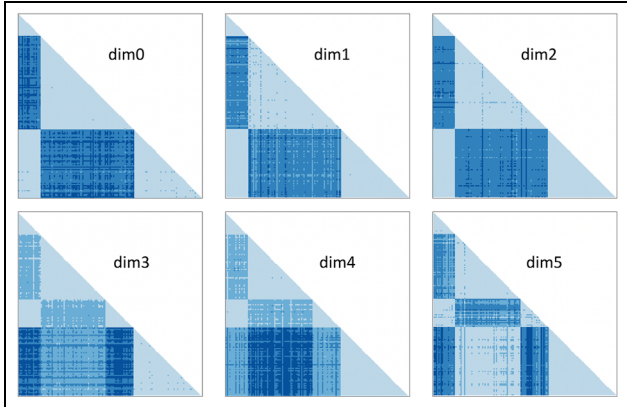


Figure 7. Data patterns in each dimension of the synthetic data.

4(b)) present subspaces' topology dominated by different dimensions. We brushed on different branches of this tree, and the automatic algorithm derived that dimensions 0, 1, and 2 were similar dimensions in most cases during our exploration. Dimensions 3, 4, and 5 were more often to be considered as dominant dimensions, as they presented unique data patterns.

We loaded all six dimensions into the incremental dimensionality analysis prototype, as shown in Figure 6. The first row of triangular matrices (highlighted in red dashed triangles) in the matrix views (Figure 6(c)) shows how individual dimensions incrementally affect subspaces' topology. From the first thumbnail view on the left (the top one), we know that the dimension order from left to right is *dim0*, *dim1*, *dim2*, *dim3*, *dim4*, and *dim5*. The first three matrices, subspaces (0), (0, 1), and (0, 1, 2) in this row are similar, but adding *dim3* changes the matrix pattern and the difference shown in the fourth matrix demonstrates how *dim3* contributes to the aggregated pattern. Involving *dim4* and *dim5* (the fifth and sixth triangular matrices) enhances the pattern in the fourth matrix. The second row of matrices (bottom left triangles) shows incremental aggregation results with a different dimension order (the order is 0, 1, 5, 2, 3, and 4) and more weights on *dim5*. Putting these two rows in CMP layout was to compare pattern changes in consecutive exploration steps. We applied data filtering on *dim5*, *dim2*, *dim3*, and *dim4* (Figure 6(b)) in the third key frame. The 20 data items that passed the filter are listed in the table in Figure 6(e), and the incremental aggregation results are shown in the third row of matrices. This row of matrices, from left to right, demonstrates how data distribution changes progressively in HD spaces. Specifically, the leftmost matrix in this row contains blue color in three scales. Progressively, from left to right, the number of matrix

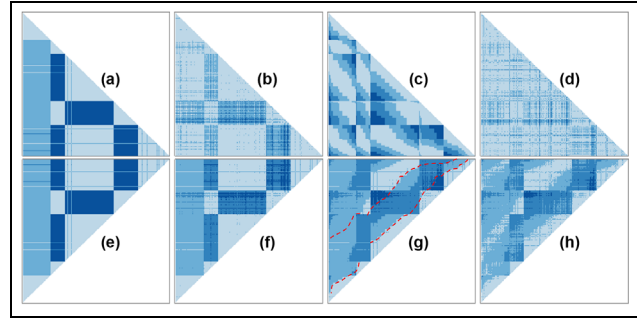


Figure 8. Incremental dimensionality analysis: the first/second row shows individual/aggregated results in SID layout; each column compares one dimension and the aggregated result in BAK layout: (a) Cylinder, (b) Horsepower, (c) Year, (d) Acceleration, (e) [Cylinder], (f) [Cylinder, Horsepower], (g) [Cylinder, Horsepower, Year] and (h) [Cylinder, Horsepower, Year, Acceleration].

elements with one blue color is decreasing and only two colors are left when reaching the rightmost matrix. The progressive results are more important than the cluster information in individual subspaces in revealing dimension characteristics, and they also expose what dimension(s) are affecting what data item(s). The fish-eye interaction shown in Figure 6(c) helps to track a particular matrix element and the elements around it.

Car data

Our second case study uses the Car data set, which contains seven dimensions and 392 data items, and it has been explored in several previous HD data visualization works.^{14,45} In total, 127 subspaces were generated from this data. We brushed on the topological hierarchical tree to select different branches and compare the selected subspaces from the matrix views. Dimension characterization results were not considered to be valid if the selected matrices looked obviously different. In this process, we found that dimension *MPG*, *Horsepower*, *Weight*, and *Acceleration* were both similar and conforming dimensions in many subspace contexts, which makes us believe that there is no very contrasting pattern in these four dimensions under the subspaces' contexts. However, dimension *Cylinder*, *Year*, and *Origin* appeared to be dominant dimensions and were not categorized as similar or conforming dimensions in our exploration, which indicates they have very distinct data patterns, respectively.

We loaded four dimensions, *Cylinder*, *Year*, *Horsepower*, and *Acceleration*, into the incremental dimensionality analysis prototype. Choosing which dimensions for further analysis usually depends on

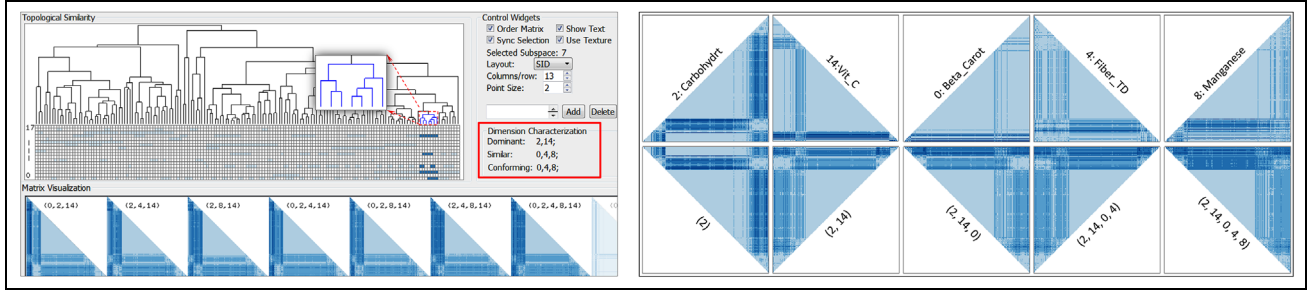


Figure 9. Exploring the USDA Food data: (left) dominant, similar and conforming dimensions in the subspace context are identified; (right) individual (top) and aggregated (bottom) results of five dimensions with BAK layout in the incremental dimensionality analysis prototype.

users' analytic goal. With the knowledge that dimension *Cylinder* and *Year* are dominant dimensions, and there is no extremely contrasting pattern in *Horsepower* and *Acceleration*, we decide to put them in this order: *Cylinder*, *Horsepower*, *Year*, and *Acceleration*. Figure 8 shows one state (key frame) of matrix views during our exploration. Two rows of matrices present both the individual (the top row) and aggregated results (the bottom row). SID layout is used to compare individual or aggregated matrices horizontally; whereas, BAK layout is used to compare individual and aggregated matrices vertically. The matrix of aggregating *Cylinder* and *Horsepower* (Figure 8(f)) does not show significant differences from the matrix that only contains *Cylinder* (Figure 8(e)), though some contrasting patterns have been diluted (some dark blue regions become lighter). The dominant dimension *Year* (Figure 8(c)) shows obvious differences from *Cylinder* (Figure 8(a)); it adds many small light blue patches along the diagonal of the aggregated matrix (some diagonal patches are highlighted in the red dashed lines in Figure 8(g)). The dimension *Acceleration* only shows sporadic patterns under the consistent row/column order (Figure 8(d)). Adding it to the aggregated matrix does not significantly change the matrix pattern (Figure 8(h)), which reveals the reason why it conforms to other matrix patterns.

USDA Food Composition data

Our third case study uses the USDA Food Composition data, which contains 18 dimensions, 722 data items, and it has been explored in several subspace visualization works.^{7-9,23} Different from the previous two studies, the CLIQUE algorithm was applied first on this data and it derived 159 interesting subspaces out of 262,143 ($2^{18} - 1$) possible subspaces.

Brushing on different branches of the topological tree helped us understand how similar subspaces look like in each branch. In Figure 9 (left), the selected

branch presents subspaces' topology dominated by dimensions 2 (*Carbohydrat*) and 14 (*Vit_C*). Dimensions 0 (*Beta_Carot*), 4 (*Fiber_TD*), and 8 (*Manganese*) are similar to each other and conform to the selected subspaces' topology (see the red box). We then loaded these five dimensions into the incremental dimensionality analysis prototype for detailed examination. Parts of the matrix views are shown in Figure 9 (right). Since we have known that dimensions 2 and 14 dominate the data patterns of the similar subspaces, we put them as the first two dimensions during aggregation. The top row of matrices in Figure 9 (right) shows the data pattern of each dimension in BAK layout. Apparently, dimensions 2 and 14 are different. Dimensions 4 and 8 have similar data pattern, which is, to some extent, similar to the data pattern in dimension 0. Our automatic algorithm considered these three dimensions as similar dimensions in the current subspaces' context. From the verification process, we understand more about them and confirm that dimensions 4 and 8 are more similar to each other when compared with dimension 0. The bottom row of matrices demonstrates the incremental dimension aggregation results. Subspaces (2, 14), (2, 14, 0), (2, 14, 0, 4), and (2, 14, 0, 4, 8) show similar data patterns, which verifies the characterization results that dimensions 0, 4, and 8 conform to the topology formed by dimensions 2 and 14.

Discussion and limitation

Scatterplot and matrix visualization

In this article, we use matrix visualizations to enable visual comparisons between subspaces and analyze dimension characteristics of HD data. Scatterplots are not as good as matrix visualizations for cross-comparison due to their instability resulted from dimensionality reduction algorithms (MDS is used throughout the article to generate scatterplots). Our

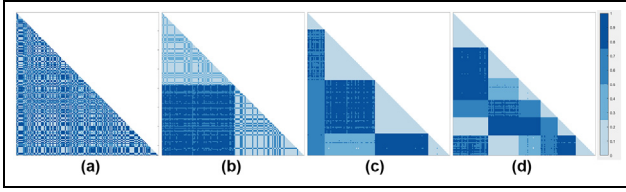


Figure 10. (a) The original distance matrix without reordering; (b, c, d) matrices reordered by the k -means algorithm with $k = 2, 4$, and 6 , respectively.

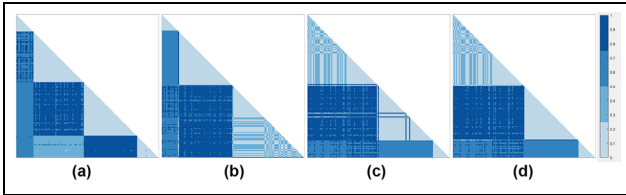


Figure 11. Different reordering algorithms: (a) k -means with $k = 4$; (b) reverse Cuthill-McKee (RCM); (c) approximate minimum degree permutation (AMD); and (d) column permutation based on non-zero count (CP).

work is by no means to claim that matrix visualizations are always better than scatterplots. Both of them have advantages and disadvantages in different scenarios. In this work, our fundamental task is to visually compare subspaces. Presenting subspaces with matrix visualizations allows us to take advantage of the consistent row/column order of matrices and conduct cross-subspace comparisons. For tasks that need to demonstrate cluster information of a subspace or interact with individual data items, we would resort to scatterplots since they are more intuitive.

Matrix pattern enhancement

In Figure 5(a) and (b), we have shown that the reordering of rows/columns of a matrix is very important to reveal the matrix pattern and assist the comparison between matrices. We use the k -means algorithm as it is simple and sufficient in this work. All experiments use $k = 5$, though users can adjust the value of k on-demand. Here, we show the effect of different k values in Figure 10. The four matrices in this figure present the same subspace, which is subspace $(0, 1, 2, 3, 4, 5)$ of the synthetic data. From the visualization results, when $k = 2$, two major clusters (light and dark blue regions) can be observed in the matrix visualization. However, the two clusters are not separated very well, as the light blue regions (two sides of the triangular matrix) are mixed with many dark blue rows/columns, and light blue rows/columns can also be found in the

dark blue region (the bottom left corner of the triangular matrix). When $k = 6$, the matrix visualization shows little mixture of colors inside different color regions but data patterns are decomposed into many small patches. A proper k values ($k = 4$ in this case) can make the matrix pattern easier to be recognized. We would like to emphasize that our goal in this work is to compare subspaces (i.e. matrix visualization results), rather than to reveal cluster information in individual subspaces. For the latter purpose, we agree that scatterplot is better than matrix visualization.

We had also considered several other matrix reordering algorithms in the design stage of the prototype. However, since our objective is not comparing different reordering algorithms, we did not explicitly discuss them in the “Subspace comparison and dimension characterization” section (a thorough review of matrix reordering algorithms can be found in the state-of-the-art report by Behrisch et al.³⁰). Here, we present the results of those reordering algorithms in comparison with k -means in Figure 11. The same matrix (the same data used in Figure 10) was reordered by four different algorithms: (1) k -means with $k = 4$, (2) reverse Cuthill-McKee (RCM),⁴⁶ (3) approximate minimum degree permutation (AMD),⁴⁷ and (4) column permutation based on non-zero count (CP).⁴⁸ The reordering results from the four algorithms all present certain visual patterns in the matrix visualizations. We cannot conclude which algorithm outperforms others, as every algorithm has its own set of parameters to play with. Although k -means has been used in this article to demonstrate our prototype, it can be easily replaced if other algorithms are preferred in certain scenarios.

Dimension characterization

We derive dimension characteristics based on data topology of subspaces in this work, thus the Euclidean distance is used. For a specific application, other distance metrics, such as cosine distance, might be considered based on concrete analytic goals. However, since our approach takes advantage of the three different layouts of triangular matrices, we restrict the distance metric used in our prototype to be symmetric. New dimension characteristics can also be derived by adapting the automatic algorithm to specific requirements. The accuracy of the automatic characterization algorithm is highly dependent on the subspace context. For example, in our exploration with the USDA Food data, although dimension *Beta_Carot* and *Filer_TD* are not very similar, they are still considered as similar dimensions in the context of selected subspaces. Refining the selection of subspaces will make the characterization results more accurate. In addition,

the definition of “dominant,” “similar,” and “conforming” may be different in different applications. Users probably need to adapt the automatic algorithm to their specific requirements. Based on different analytic questions, more characteristics of dimensions can also be derived from subspace comparisons. For example, in the conforming dimension analysis part, we can allow our algorithm to accept similar subspaces differing with two dimensions, such as (1, 2, 3) and (1, 2, 3, 4, 5). If dimensions 4 and 5 are not conforming to the subspaces’ pattern individually, they will be complementary to each other. Complementing dimensions in this subspace context can then be derived.

Scalability

We realized two issues that may affect the scalability of our approach: one is the large number of possible subspaces and the other is the large number of data items. Currently, we rely on subspace clustering algorithms (CLIQUE) to filter subspaces and start explorations with a small number of subspaces that contain relatively more prominent data patterns. With the exploration process going on, users can add/delete interesting subspaces, and the hierarchical trees will be updated dynamically. This strategy currently works well, as the number of interesting subspaces is not extremely large under the filtering criteria. However, this also indicates that our approach needs to choose proper parameters for subspace clustering algorithms based on available resources (computing power, spatial resolution, etc.). As demonstrated in our case studies, our approach can scale up to data with around 20 dimensions. For data with even more dimensions (e.g. 100 dimensions), the bottleneck of our approach is the number of interspersing subspaces filtered out by subspace clustering algorithms. Specifically, when the number of interesting subspaces is very large, we may not have enough space to demonstrate matrices for all subspaces and the performance of the automatic characterization algorithm may become very poor. In addition, matrix visualizations, with limited spatial resolutions, may not be able to show all data items. For example, if a data set contains 500 items (at least 500×500 pixels are needed) but the available space for a matrix is 100×100 pixels, then not all distance values can be reflected on the matrix visualization. We actually have encountered this problem when working with the real-world data sets in the “Case studies” section. The reordering of rows/columns of matrices, which moves similar matrix elements spatially closer to each other, not only enhances matrix patterns but also makes visible pixels representative. Although some matrix elements are not shown in matrix visualization, the visible elements around them, to some extent, represent the invisible ones due

to their similarity. It is worth mentioning that our focus in this article is to present a new perspective (i.e. subspace comparison) to analyze HD data, through which, the characteristics of different dimensions can be revealed. The studies presented in the work used real-world HD data from previous similar works, but in moderate scales. Extending the solution to large-scale HD data (e.g. more than 100 dimensions) could be an interesting direction for future work.

Conclusion and future work

In this article, we proposed a visual analytics approach to analyze dimension characteristics in HD data by effectively comparing their subspaces. By aggregating dimensions of HD data and presenting the resulted lower dimensional pair-wise distance values with matrix visualizations, we resolved the conflict between subspace similarity measurements and visualizations in dimensionality reduction-based scatterplot visualizations. Through subspace comparisons, three characteristics (dominant, similar, and conforming) of dimensions in different subspace contexts were established. An incremental dimensionality analysis prototype was also developed to further verify and incrementally investigate the behaviors of individual dimensions. Through case studies with synthetic and real-world HD data, we demonstrated the effectiveness and usefulness of the proposed approach. In the future, we would like to put more efforts on the scalability problem for data with more dimensions and design intelligent strategies to filter subspaces. Identifying more dimension characteristics based on different analysis tasks is another interesting direction for further exploration.

Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work was supported in part by NSF grants IIS-1250752, IIS-1065025, and US Department of Energy grants DE-SC0007444, DE-DC0012495, program manager Lucy Nowell.

References

1. Fernstad SJ, Shaw J and Johansson J. Quality-based guidance for exploratory dimensionality reduction. *Inform Visual* 2013; 12(1): 44–64.
2. Liu X and Shen HW. Association analysis for visual exploration of multivariate scientific data sets. *IEEE T Vis Comput Gr* 2016; 22(1): 955–964.
3. Kriegel HP, Kröger P and Zimek A. Clustering high-dimensional data: a survey on subspace clustering,

- pattern-based clustering, and correlation clustering. *ACM T Knowl Discov D* 2009; 3(1): 1.
4. Beyer K, Goldstein J, Ramakrishnan R, et al. When is “nearest neighbor” meaningful? In: *Proceedings of the international conference on database theory*, Jerusalem, Israel, 10–12 January 1999, pp. 217–235. Berlin; Heidelberg: Springer.
 5. Müller E, Günnemann S, Assent I, et al. Evaluating clustering in subspace projections of high dimensional data. *Proc VLDB Endowm* 2009; 2(1): 1270–1281.
 6. Ferdosi BJ, Buddelmeijer H, Trager S, et al. Finding and visualizing relevant subspaces for clustering high-dimensional astronomical data using connected morphological operators. In: *Proceedings of the IEEE symposium on visual analytics science and technology (VAST)*, Salt Lake City, UT, 25–26 October 2010, pp. 35–42. New York: IEEE.
 7. Tatu A, Maas F, Färber I, et al. Subspace search and visualization to make sense of alternative clusterings in high-dimensional data. In: *Proceedings of the IEEE conference on visual analytics science and technology (VAST)*, Seattle, WA, 14–19 October 2012, pp. 63–72. New York: IEEE.
 8. Yuan X, Ren D, Wang Z, et al. Dimension projection matrix/tree: interactive subspace visual exploration and analysis of high dimensional data. *IEEE T Vis Comput Gr* 2013; 19(12): 2625–2633.
 9. Zhou F, Li J, Huang W, et al. Dimension reconstruction for visual exploration of subspace clusters in high-dimensional data. In: *Proceedings of the IEEE Pacific visualization symposium (PacificVis)*, Taipei, Taiwan, 19–22 April 2016, pp. 128–135. New York: IEEE.
 10. Jolliffe I. *Principal component analysis*. Hoboken, NJ: Wiley Online Library, 2002.
 11. Cox TF and Cox MA. *Multidimensional scaling*. Boca Raton, FL: CRC Press, 2000.
 12. Van der Maaten L and Hinton G. Visualizing data using t-SNE. *J Mach Learn Res* 2008; 9: 2579–2605.
 13. Inselberg A. *Parallel coordinates*. New York: Springer, 2009.
 14. Palmas G, Bachynskyi M, Oulasvirta A, et al. An edge-bundling layout for interactive parallel coordinates. In: *Proceedings of the 2014 IEEE Pacific visualization symposium (PacificVis)*, Yokohama, Japan, 4–7 March 2014, pp. 57–64. New York: IEEE.
 15. Becker RA and Cleveland WS. Brushing scatterplots. *Technometrics* 1987; 29(2): 127–142.
 16. Rao R and Card SK. The table lens: merging graphical and symbolic representations in an interactive focus + context visualization for tabular information. In: *Proceedings of the SIGCHI conference on human factors in computing systems*, Boston, MA, 24–28 April 1994, pp. 318–322. New York: ACM.
 17. Tominski C, Abello J and Schumann H. Axes-based visualizations with radial layouts. In: *Proceedings of the 2004 ACM symposium on applied computing*, Nicosia, Cyprus, 14–17 March 2004, pp. 1242–1247. New York: ACM.
 18. Williams M and Munzner T. Steerable, progressive multidimensional scaling. In: *Proceedings of the IEEE symposium on information visualization (INFOVIS 2004)*, Austin, TX, 10–12 October 2004, pp. 57–64. New York: IEEE.
 19. Turkay C, Filzmoser P and Hauser H. Brushing dimensions—a dual visual analysis model for high-dimensional data. *IEEE T Vis Comput Gr* 2011; 17(12): 2591–2599.
 20. Krause J, Dasgupta A, Fekete JD, et al. SeekAView: an intelligent dimensionality reduction strategy for navigating high-dimensional data spaces. In: *Proceedings of the 2016 IEEE 6th symposium on large data analysis and visualization (LDAV)*, Baltimore, MD, 23–28 October 2016, pp. 11–19. New York: IEEE.
 21. Biswas A, Dutta S, Shen HW, et al. An information-aware framework for exploring multivariate data sets. *IEEE T Vis Comput Gr* 2013; 19(12): 2683–2692.
 22. Sedlmair M, Heinzl C, Bruckner S, et al. Visual parameter space analysis: a conceptual framework. *IEEE T Vis Comput Gr* 2014; 20(12): 2161–2170.
 23. Watanabe K, Wu HY, Niibe Y, et al. Biclustering multivariate data for correlated subspace mining. In: *Proceedings of the 2015 IEEE Pacific visualization symposium (PacificVis)*, Hangzhou, China, 14–17 April 2015, pp. 287–294. New York: IEEE.
 24. Liu S, Wang B, Thiagarajan JJ, et al. Visual exploration of high-dimensional data through subspace analysis and dynamic projections. *Comput Graph Forum* 2015; 34(3): 271–280.
 25. Agrawal R, Gehrke J, Gunopulos D, et al. Automatic subspace clustering of high dimensional data for data mining applications. *SIGMOD Rec* 1998; 27(2): 94–105.
 26. Cheng CH, Fu AW and Zhang Y. Entropy-based subspace clustering for mining numerical data. In: *Proceedings of the 5th ACM SIGKDD international conference on knowledge discovery and data mining*, San Diego, CA, 15–18 August 1999, pp. 84–93. New York: ACM.
 27. Aggarwal CC, Wolf JL, Yu PS, et al. Fast algorithms for projected clustering. *SIGMOD Rec* 1999; 28: 61–72.
 28. Procopiuc CM, Jones M, Agarwal PK, et al. A Monte Carlo algorithm for fast projective clustering. In: *Proceedings of the 2002 ACM SIGMOD international conference on management of data*, Madison, WI, 3–6 June 2002, pp. 418–427. New York: ACM.
 29. Baumgartner C, Plant C, Railing K, et al. Subspace selection for clustering high-dimensional data. In: *Proceedings of the 4th IEEE international conference on data mining*, Brighton, 1–4 November 2004, pp. 11–18. New York: IEEE.
 30. Behrisch M, Bach B, Henry Riche N, et al. Matrix reordering methods for table and network visualization. *Comput Graph Forum* 2016; 35(3): 693–716.
 31. Bertin J. *Semiology of graphics: diagrams, networks, maps*. Madison, WI: University of Wisconsin Press, 1983.
 32. Alper B, Bach B, Henry Riche N, et al. Weighted graph comparison techniques for brain connectivity analysis. In: *Proceedings of the SIGCHI conference on human factors*

- in computing systems (CHI'13), Paris, 27 April –2 May 2013, pp. 483–492. New York: ACM.
33. Ghoniem M, Fekete JD and Castagliola P. On the readability of graphs using node-link and matrix-based representations: a controlled experiment and statistical analysis. *Inform Visual* 2005; 4(2): 114–135.
 34. Keller R, Eckert CM and Clarkson PJ. Matrices or node-link diagrams: which visual representation is better for visualising connectivity models? *Inform Visual* 2006; 5(1): 62–76.
 35. Liu X and Shen HW. The effects of representation and juxtaposition on graphical perception of matrix visualization. In: *Proceedings of the 33rd annual ACM conference on human factors in computing systems (CHI'15)*, Seoul, Republic of Korea, 18–23 April 2015, pp. 269–278. New York: ACM.
 36. Goodwin S, Dykes J, Slingsby A, et al. Visualizing multiple variables across scale and geography. *IEEE T Vis Comput Gr* 2016; 22(1): 599–608.
 37. Achtert E, Kriegel HP and Zimek A. ELKI: a software system for evaluation of subspace clustering algorithms. In: *Proceedings of the international conference on scientific and statistical database management*, Hong Kong, China, 9–11 July 2008, pp. 580–585. Berlin; Heidelberg: Springer.
 38. Schubert E, Koos A, Emrich T, et al. A framework for clustering uncertain data. *Proc VLDB Endowm* 2015; 8(12): 1976–1979.
 39. Liiv I. Seriation and matrix reordering methods: an historical overview. *Stat Anal Data Min* 2010; 3(2): 70–91.
 40. Jain AK. Data clustering: 50 years beyond K-means. *Pattern Recogn Lett* 2010; 31(8): 651–666.
 41. Tibshirani R, Walther G and Hastie T. Estimating the number of clusters in a data set via the gap statistic. *J Roy Stat Soc B* 2001; 63(2): 411–423.
 42. Tufte ER. *Beautiful evidence*. Cheshire, CT: Graphics Press, 2006.
 43. Rogers DJ and Tanimoto TT. A computer program for classifying plants. *Science* 1960; 132(3434): 1115–1118.
 44. Bilmes JA. *A gentle tutorial of the EM algorithm and its application to parameter estimation for Gaussian mixture and hidden Markov models*, vol. 4. Berkeley, CA: International Computer Science Institute, 1998, p. 126.
 45. Peng W, Ward MO and Rundensteiner EA. Clutter reduction in multi-dimensional data visualization using dimension reordering. In: *Proceedings of the IEEE symposium on information visualization*, Austin, TX, 10–12 October 2004, pp. 89–96. New York: IEEE.
 46. George A and Liu JW. *Computer solution of large sparse positive definite*. Upper Saddle River, NJ: Prentice Hall, 1981.
 47. Amestoy PR, Davis TA and Duff IS. An approximate minimum degree ordering algorithm. *SIAM J Matrix Anal A* 1996; 17(4): 886–905.
 48. Gilbert JR, Moler C and Schreiber R. Sparse matrices in MATLAB: design and implementation. *SIAM J Matrix Anal A* 1992; 13(1): 333–356.