Dimension Projection Matrix/Tree: Interactive Subspace Visual Exploration and Analysis of High Dimensional Data

Xiaoru Yuan, *Member, IEEE*, Donghao Ren, Zuchao Wang, and Cong Guo

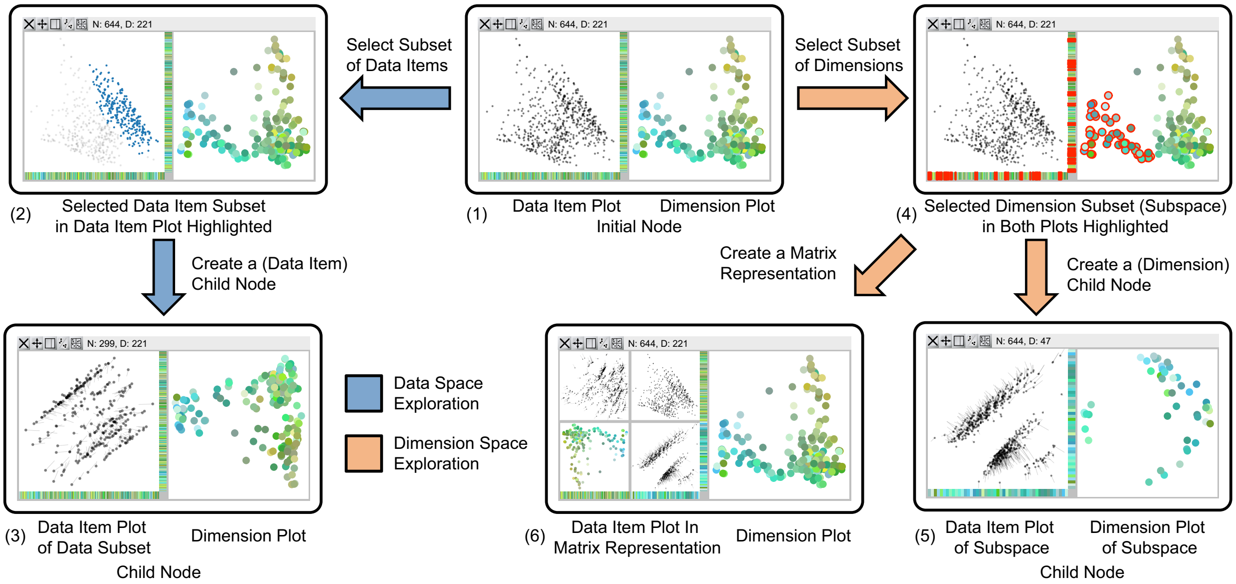

N: 644, D: 221
Select Subset of Data Items
Select Subset of Dimensions
(2) Selected Data Item Subset in Data Item Plot Highlighted
(1) Data Item Plot
Dimension Plot
Initial Node
(4) Selected Dimension Subset (Subspace) in Both Plots Highlighted
Create a (Data Item) Child Node
Create a Matrix Representation
Create a (Dimension) Child Node
N: 299, D: 221
N: 644, D: 47
Data Space Exploration
Dimension Space Exploration
(3) Data Item Plot of Data Subset
Dimension Plot
Child Node
(6) Data Item Plot In Matrix Representation
Dimension Plot
(5) Data Item Plot of Subspace
Dimension Plot of Subspace
Child Node

Fig. 1. An illustration of exploration enabled in our proposed Dimension Projection Tree/Matrix visualization of high dimensional data. Each node is consist of one data item plot and one dimension plot. Starting from an initial node (1), users can explore either in data item space (2, 3) or dimension space (4, 5, 6), by first selecting a subset of data items or dimensions, and then creating child nodes (3, 4, 5) or converting one single dimension project plot into a matrix representation (6).

Abstract— For high-dimensional data, this work proposes two novel visual exploration methods to gain insights into the data aspect and the dimension aspect of the data. The first is a Dimension Projection Matrix, as an extension of a scatterplot matrix. In the matrix, each row or column represents a group of dimensions, and each cell shows a dimension projection (such as MDS) of the data with the corresponding dimensions. The second is a Dimension Projection Tree, where every node is either a dimension projection plot or a Dimension Projection Matrix. Nodes are connected with links and each child node in the tree covers a subset of the parent node's dimensions or a subset of the parent node's data items. While the tree nodes visualize the subspaces of dimensions or subsets of the data items under exploration, the matrix nodes enable cross-comparison between different combinations of subspaces. Both Dimension Projection Matrix and Dimension Project Tree can be constructed algorithmically through automation, or manually through user interaction. Our implementation enables interactions such as drilling down to explore different levels of the data, merging or splitting the subspaces to adjust the matrix, and applying brushing to select data clusters. Our method enables simultaneously exploring data correlation and dimension correlation for data with high dimensions.

Index Terms—High Dimensional Data, Hierarchical Visualization, Sub-dimensional Space, User Interaction, Subspace, Tree, Matrix.

1 Introduction

High-dimensional data occurs frequently in science, engineering, and daily life. For example, DNA microarray technology can produce vast amounts of measurement data with millions of micrometer-scale probes. When analyzing text documents, the number of dimensions can equate to the size of a dictionary if a word-frequency vector is employed. As data are being accumulated at unprecedented speed, handling such data efficiently to provide insights to the users is critical for effective data analysis. Visualizing and understanding multi-dimensional data that is large in both size and dimensionality, is a major challenge in the research community.

Currently, one major category of techniques for high-dimensional data visualization uses dimension reduction. By converting the data to lower dimensions, which are easier to visualize, dimension reduction aids comprehensive and focused analysis. One problem with the current dimension reduction technology is that users have little control over the process. In addition, after the dimension reduction, the original dimensionality information is lost. The intrinsic information on the relationship between the dimensions is no longer accessible to the end user. User interaction is not usually provided in such cases. Another group of visualization techniques, including parallel coordinates [15, 34], scatterplot matrix [7], and table lens [29], avoids dimension reduction and visualizes the high-dimensional data at the expense of spatial resolution. Such techniques mostly deal with data with no

- *Xiaoru Yuan, Donghao Ren, Zuchao Wang and Cong Guo are with the Key Laboratory of Machine Perception (Ministry of Education) and School of EECS, Peking University, Beijing, P.R. China. E-mail: {xiaoru.yuan, donghao.ren, zuchao.wang, cong.guo}@pku.edu.cn*

Manuscript received 31 March 2013; accepted 1 August 2013; posted online 13 October 2013; mailed on 4 October 2013.
For information on obtaining reprints of this article, please send e-mail to: tvcg@computer.org.

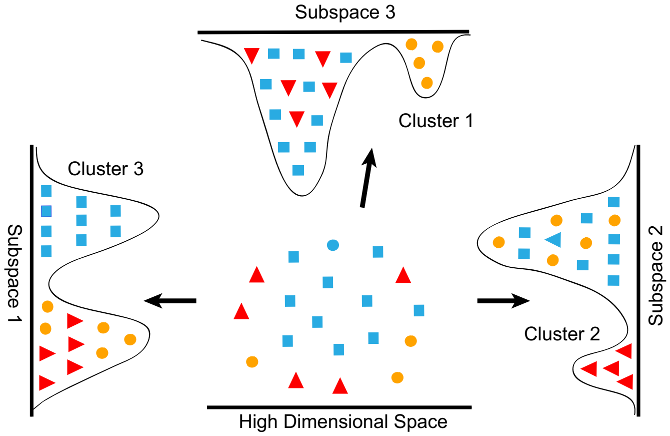


Fig. 2. Illustration of clustering in subspaces. Separation of clusters in appropriate selection of dimension subspaces can be much easier than that in the original high dimensional space.

more than twenty dimensions.

Most of the current work on high-dimensional data visualization has focused on large datasets and reducing data clutter during visualization. Relatively few visualization systems can handle datasets with hundreds of dimensions, although these datasets are becoming increasingly common in many application areas, such as bio-informatics. A scalable visualization tool that allows the user to interactively visualize data with high dimensionality is critical for understanding the data.

Furthermore, the above mentioned techniques do not capitalize on the differences between dimensions. Indeed, not all dimensions are relevant for analysis in high-dimensional data. Irrelevant dimensions can make the discovery of interesting features, such as clusters, much more difficult by hiding them in noisy data. Even worse, in data with very high dimensions, data objects are nearly equidistant. In data mining, instead of examining the dataset as a whole, more recent research has employed subspace clustering algorithms to localize, search and uncover clusters that exist in multiple, possibly overlapping subspaces as illustrated in Figure 2. In large datasets, on the other hand, interesting subspaces may only be discovered when appropriate set of data items are selected, as other data items may act as noise.

We suggest to support subspace exploration in a divide and conquer manner. The datasets can be divided or reduced into subsets of dimensions and/or data items, and those subsets can be organized in a hierarchical manner (tree). In this work, we develop a set of visual exploration methods and tools called Dimension Projection Matrix/Tree (see Figure 1), to visualize high-dimensional data and help users gain insights from both the data aspect and dimension aspect. In our work, a Dimension Projection Matrix (Figure 1 (6)) can be considered as an extension of a scatterplot matrix, where each row or column of the matrix represents a group of dimensions, and each cell illustrates a dimension projection (such as MDS) of the data with the corresponding dimensions. In the Dimension Projection Tree, every node is either a dimension projection plot or a Dimension Projection Matrix. Nodes are connected with curved links and each child node in the tree covers a subset of the parent node's dimensions or a subset of the parent node's data items. While the nodes of the Dimension Projection Tree visualize the subspaces of dimensions (Figure 1 (5, 6)) or subsets of the data (Figure 1 (3)) under exploration, the matrix nodes further provide cross-comparison between different combinations of subspaces. Both Dimension Projection Matrix and Dimension Project Tree can be constructed algorithmically through automation, or manually invoked through user interaction.

The user interface for Dimension Projection Matrix/Tree is designed to enable interactive operation and exploration in the dimension and data item hierarchy, such as drilling down to explore different levels of the data, merging or splitting the subspaces to adjust the matrix, and applying brushing to select data clusters (Figure 1 (2, 4)). Our methods enable simultaneously exploring data correlation and dimension correlation for data with high dimensions. The interaction provided by our proposed methods allows users to effectively and efficiently explore large datasets containing hundreds of dimensions. Furthermore, our visualization methods enable simultaneous exploration on the data correlation and dimension correlation for data with high dimensionality.

To summarize, the proposed Dimension Projection Matrix/Tree has the following features:

- Improved scalability in terms of number of dimensions. The hierarchy enables users to explore high-dimensional data sets at different levels from the aspect of both data items and dimensions.

- Convenient means of exploration and manipulation of dimension subspaces. Users can operate on each plot of the Dimension Projection Matrix/Tree to explore corresponding dimension subspaces.

- Simultaneous exploration and manipulation of both data items and dimensions. By providing a visualization of both the data projection and the sub-dimensional space projection, users can interact directly with both data items and dimensions.

The reminder of the paper is organized as the follows. We start by summarizing related works. Then we present the design of the Dimension Projection Matrix/Tree, show details of the interactions, offer guidelines and then show some notable implementation details. After presenting our case studies and discussions, we conclude the paper.

2 Related Work

In this section, we first show the limitations of general multidimensional data visualizations, when applied to data with large scale and high dimensionality. We then discuss the hierarchical (divide and conquer) visualization strategy. Finally we discuss a few existing techniques on subspace clustering analysis comparable to our work.

2.1 Multi-dimensional Data Visualization

A few visualization techniques can simultaneously represent all variables of a multi-dimensional dataset.

The scatterplot matrix [7] visualizes the data projection in 2D subspaces spanned by all combinations with two dimensions. The parallel coordinates [15, 16] show data simultaneously on multiple dimensions via polyline metaphor. These two methods work well for correlation detection and data filtering, but suffer from the cluttering problem when handling large data or high dimensionality as one high dimensional data point has to be presented as many points or a polyline. Although researchers have tried to cope with the cluttering problem by navigation [10] and feature extraction [30, 36, 9], generally such methods do not scale well. Tablelens [29, 25] and Pixel-based visualization [20] are also useful techniques for multi-dimensional data visualization. They are scalable, however usually not good at showing data correlation and dimension correlation. One exception is the Value and Relation (VaR) display [39], which can visualize dimension correlation. However it is not able to show data correlation at the same time.

Due to the complexity caused by the number of dimensions, high-dimensional datasets are usually visualized after dimension reduction. Statistical methods to reduce the dimensionality can be categorized as linear projection methods (such as PCA [18]), and non-linear methods (such as MDS [38], SOM [21]). Multi-dimensional Scaling [38] (MDS) projects high dimensional data points into a low dimensional space. An MDS algorithm starts with a matrix of item-to-item similarities and the output dimensionality (usually lower than the input). The location of each item in the output is assigned according to the similarities. In many visualization applications, a dimension projection method, such as MDS, projects data items into a two-dimensional plane, which can be displayed as a scatterplot.

In general, the dimension projection approaches are computationally expensive, despite of the techniques [5, 37, 14, 43] that reduce the computational load. These projection methods are usually scalable to data scale and dimensionality. The major limitation of them is that the individual dimensional information is lost, therefore the results are hard to explain. We use PCA and MDS projection in our implementation due to their good scalability. By allowing user exploration, the results are easier to explain.

Most dimensionality reduction methods focus predominantly on preserving one or a few significant structures in data. Often, the question of which structure to preserve is uncertain and task-dependent. To deal with this problem, grand tour [4, 35, 8] examines structure of high-dimensional data from all possible angles. Projection pursuit [8, 12] only shows the important aspects of high-dimensional space. Johansson et al.'s system [17] selects dimensions by quality metrics. Although many techniques above allow human intervention to select dimensions, the interventions are rather restricted. Our technique allows free dimension selection.

2.2 Hierarchical Visualization of Multi-dimensional Data

To handle large multi-dimensional datasets, the major challenge is to solve the clutter problem in terms of both dimensionality and amount of data items. A visualization that employs a hierarchical data structure together with a level of detail approach is promising. Such a hierarchy can be built in the data item space. Hierarchical parallel coordinates [13] have been proposed as a multi-resolution view of large multi-dimensional data and are based on hierarchical clustering. Long and Linsen [23] developed MultiClusterTree to interactively explore hierarchical clusters in multi-dimensional multi-variate data. Sifer [31] designed a variety of user interfaces based on parallel coordinate trees to support the exploration of hierarchical multi-dimensional data. Slingsby et al. [32] created the data item hierarchy by gradually partitioning the dataset dimension by dimension. They then explored the effects of a modified treemap layout to show the hierarchy. Piringer et al.'s hierarchical difference scatterplot [28] explores the possible sub-dimension spaces, and explicitly visualizes differences between them. More recently, Elmqvist and Fekete [11] systematically studied the hierarchical aggregation for visualization which can be applied to visualize multi-dimensional data.

There are also hierarchies in dimension space. Yang et al. [41] proposed a radial, space-filling hierarchy visualization tool called InterRing for visually navigating and manipulating hierarchical structures. The work is then integrated in Visual Hierarchical Dimension Reduction (VHDR) [42] to explore high-dimensional datasets. Dimensions are first grouped into a hierarchy, and lower dimensional spaces are then constructed based on the clustering in the hierarchy, either manually or automatically, with the assistance of InterRing.

Our technique includes both data item hierarchy and dimension hierarchy. It therefore supports scalable exploration in both data space and dimension space.

2.3 Sub-dimensional Space Analysis

High-dimensional data faces the "Curse of Dimensionality", which means that the data items tend to be equidistant. This adds significant difficulty to pattern detection. Furthermore, for most high-dimensional data, each pattern is prominent only in a few dimensions, while the other dimensions hide more features than they reveal. Recent research has introduced analysis in sub-dimension spaces.

Subspace clustering aims to detect clusters in subspaces. For each cluster, the data items as well as the relevant dimensions are calculated at the end of the algorithms. Müller et al. [26] have classified subspace clustering algorithms as cell-based approaches (e.g. CLIQUE [3]), density-based approaches (e.g. SUBCLU [19]) and clustering oriented approaches (e.g. PROCLUS [2]). They have evaluated a number of algorithms and provided an open-source interactive framework for the analysis. Parsons et al. [27] conducted a review of common algorithms. Lex et al.'s method [22] can be used to compare clustering results across subspaces. Tatu et al. [33] proposed an interactive visualization designed for users to navigate through the subspaces, using the SURFING [6] algorithm for subspace search.

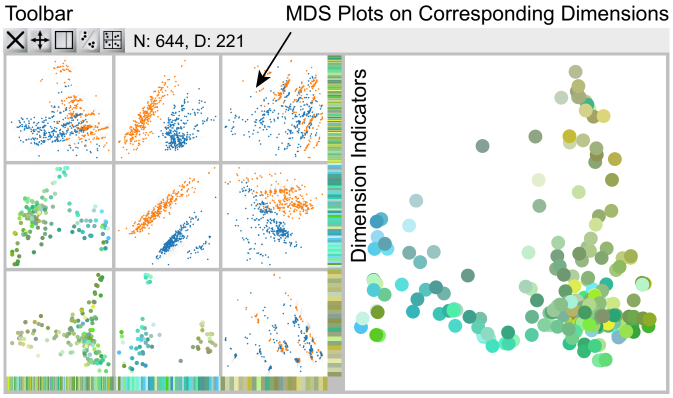
Other measures have aimed to find the most important dimensions, and are used to study the dataset in the right subspace. For example, Yang et al. [40] proposed DOSFA, in which they filtered out unimportant dimensions or dimensions similar to others. The above-mentioned work from Johansson and Johansson [17] also falls into this category.

All works above allow rather limited user intervention in the subspace analysis. In contrast, our technique supports free exploration of the subspaces. In this way, human knowledge can be better integrated into the exploration process.

3 Design of Dimension Projection Matrix/Tree

Our proposed methods aim to discover clusters and interesting subspaces from high-dimensional data. High-dimensional data can be very complex; some interesting clusters can only be identified by selecting the correct subspace and some interesting subspaces can only be found by choosing the right subset of data items.

Our methods help users analyze high-dimensional data by creating, visualizing and exploring a hierarchy of subsets. In particular, we support simultaneous exploration of subspaces and subsets of data items with a hierarchy of Dimension Projection nodes. Each Dimension Projection node corresponds to a selection of the dataset, which is determined by a set of dimensions (subspace) and a set of data items. The nodes are organized in a hierarchical structure, whereby the dimensions and data items in the child nodes are subsets of the ones in their parents. The root node of the tree represents the whole dataset, with all dimensions and data items selected. During the process of investigation, a divide and conquer strategy is employed. Users always select a subset of either dimensions or data items, e.g. effectively reduce the either the number of dimensions or the size of the data. Users can create, modify and delete child nodes with a rich set of interactions. The interactions are illustrated in Figure 4, and will be detailed in this section.



(1) 3x3 Dimension Projection Matrix (2) Dimension Plot

Fig. 3. Illustration of a Dimension Projection Matrix node. Each node represents a portion of the high-dimensional dataset defined by a set of dimensions and a set of data items. The data item plot is shown as an Dimension Projection Matrix of dimension groups. The upper right cells of the matrix show Dimension Projection of data items on the corresponding dimensions, while the lower left cells of the matrix show Dimension Projection of the corresponding dimensions. The dimensions for each cell are indicated by the dimension indicators. The dimension plot is an Dimension Projection of all dimensions in the node.

3.1 Dimension Projection Matrix Nodes

Each of our Dimension Projection Matrix node has three major components: one dimension plot, one data item plot and one toolbar.

3.1.1 Dimension Plot

The dimension plot, which is located on the right part of the Dimension Projection Matrix node (Figure 3), shows the correlations be-

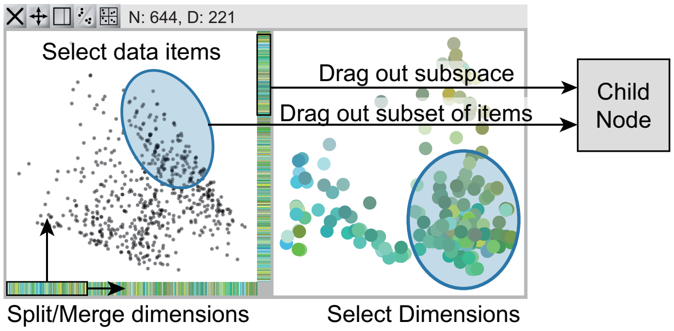


Fig. 4. Illustration of the interactions of a Dimension Projection Matrix node. Users can select data items and dimensions, and drag them out as new nodes, or create a Dimension Projection Matrix by splitting the dimensions. Linked brushing is also supported among all the nodes.

tween dimensions. Each point in the dimension plot represents one dimension, whose position is given by a method of dimension projection.

In the plot, point colors can show the dimension-wise similarity. The color is decided by the following algorithm. After the mapping of points representing the dimension onto a 3D space with dimension projection, we assign a color for each dimension point by linearly transforming its three dimensional projected coordinates to the lab color space.

After this step, however, dimensions close to each other have very similar colors, which makes it very difficult to distinguish between different dimensions when a subset of similar dimensions are grouped together. To solve this problem, we reassigned the colors with the following method[1]. First a set of colors in the lab color space are sampled uniformly, before a K-means algorithm is performed with the original dimension colors as initial centers. Finally, the dimensions are assigned with colors from the new centers when the K-means algorithm converges. The result of this process is a set of distinguishable colors that corresponds to the proximity of dimensions.

3.1.2 Data Item Plot

The data item plot, located on the left part of each Dimension Projection Matrix node (Figure 3), is a matrix of Dimension Projection plots. The design can be viewed as an adapted version of the scatterplot matrix [7], except that the rows and columns represent multiple dimensions and the scatterplots are based on Dimension Projection. The dimensions in this node can be split into k mutually excluding groups, G_1 to G_k. Each dimension in this node is represented by one small color strip on dimension indicator, at the right and bottom boundaries. Position of a dimension on the dimension indicator shows its group belonging. Each cell of the matrix corresponds to a set of dimensions. Specifically, cell at i-th column and j-th row corresponds to the dimensions given by $G_i \cup G_j$. Upper right cells display Dimension Projection of data items on the corresponding dimensions, while lower left cells display the Dimension Projection of the corresponding dimensions. Initially when a node is created, its data item plot is a 1×1 matrix. The user may later split the dimensions into k groups, resulting in $k \times k$ subplots. Each node in the upper right cell is a data item. The color of it is given by users' brushing.

3.1.3 Toolbar

The toolbar, which is located on the top of each node, contains several buttons, including "close", "move", "toggle dimension plot", "cluster data items", and "cluster dimensions". The latter two buttons are used to invoke automatic clustering algorithms to split the data items or dimensions and then create child nodes for each cluster. Next to the toolbar, N indicates the data item number in the data set, while D indicates the number of dimensions. In Figure 3, the dataset under exploration has 644 data items and 221 dimensions.

The following sections provide details about the interactions on nodes.

3.2 Dimension Space Exploration

A suite of interactions is provided for dimension space exploration, including splitting/merging, zooming in/out and dragging. Together, these interactions allow users to efficiently explore the subspaces and find the ones with the most salient clusters.

3.2.1 Splitting and Merging

Users are initially presented with only one node that covers all the dimensions in the dataset. This can be seen as a 1×1 Dimension Projection Matrix. However, interesting structures are usually hidden in the subspaces of the dataset. Users can split the dimensions manually or use some clustering algorithms. For manual splitting, users can select interested dimensions by either selecting them in the dimension indicators, or selecting them from the dimension plot with a lasso. After selecting the dimensions, users can drag them into the node to create a new dimension group, or into existing groups to merge them together. Users can also click the button on the toolbar, and run the clustering algorithms, which automatically splits the dimensions and the split new nodes are laid out automatically. By this way, a tree can be automatically constructed based with the predefined clustering methods.

Figure 5 is an example of splitting and merging dimensions.

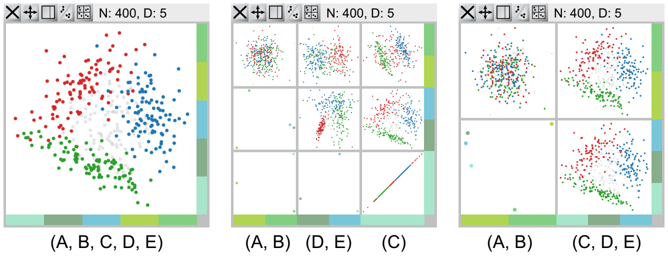


Fig. 5. Subspace exploration via splitting and merging dimension groups. Dimensions A to E in the left node are split into 3 groups (center), and then two groups are merged (right).

3.2.2 Zooming

Users seldom handle all the k^2 Dimension Projection plots simultaneously. Our methods allow users to zoom the cells in the matrix in order to change the span of the dimension groups. The size of the whole matrix is maintained so that the widening of one group would make the other groups smaller. This policy aims to make the most interesting dimension group the largest one, while keeping other interesting groups visible. Figure 6 illustrates the effects of zooming.

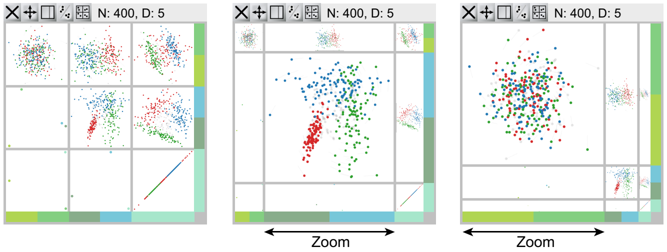


Fig. 6. Focus+Context zoom on plot size. Left: default plot sizes; Center: the second dimension group is zoomed in; Right: the first group is zoomed in.

[1] The idea of K-means clustering in the lab colorspace is inspired from Mathieu Jacomy's "I want hue" project: http://tools.medialab.sciences-po.fr/iwanthue/theory.php

3.2.3 Dragging Out

Although zooming can provide a greater amount of detail about one subspace, other surrounding plots may still distract the users if they are not necessary. Therefore we allow the users to drag the dimensions to create a child node (Figure 7). The child node is linked to its parent with several lines that correspond to the dimensions. With a hierarchical structure, the exploration history is recorded and visualized and users can easily trace the dimensions in a leaf node along its ancestors and see the effects of subspace selections; this helps users make sense of the whole analysis process.

3.2.4 Guidelines

Here we present some general guidelines for subspace exploration using our tool. The dimension plot is the main tool with which to gains insights into the subspaces, the most natural way to utilize it is to group the dimensions into several clusters, each of which represents a subspace in which the dimensions are correlated. It is a good idea to first split these clusters into a Dimension Projection matrix in order to gets insights into each cluster. Another potentially useful way is to select a few dimensions from each cluster to form a subspace, which helps reduce the number of dimensions.

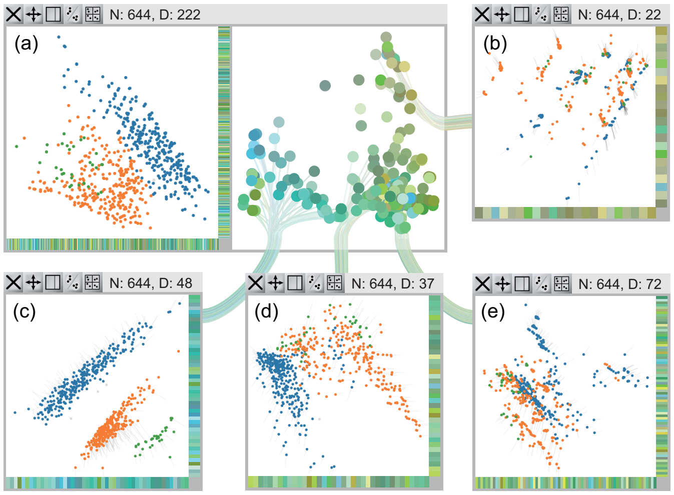


Fig. 7. Subspaces can be dragged out for further exploration. In this example, subspaces from (a) are dragged out as (b), (c), (d), (e), while the dimensions in each child node are indicated by links to the dimension plot of the parent node.

3.3 Data Item Space Exploration

For data item space exploration, users try to understand the data distribution, especially the potential clustering in each subspace. They can create child nodes for interesting subsets of data items, or label the data items via brushing.

To create child nodes, users select a set of data items within one node by using mouse to specify the desirable region. The selected region is then computed with a highlighted bounding curve filled with grey background color. Users can then directly drag the selected data items out to create a new child node through mouse movement (see Figure 8). The child node will contain these data items and all the dimensions of its parent node. Selecting a subset of data items will change the correlation of the dimensions, which means that some interesting dimension correlations can be found in the dimension plot of the child node. The user could then drag some dimensions out, and see the altered data item distribution.

Like traditional high-dimensional analysis tools, our approach support linked brushing. Users can choose a brush color and then select some points with a lasso tool; the points in other plots will also be highlighted, while points that are not selected will be faded during brushing. In addition, some datasets may come with data item labels, and the methods also support brushing points according to their labels.

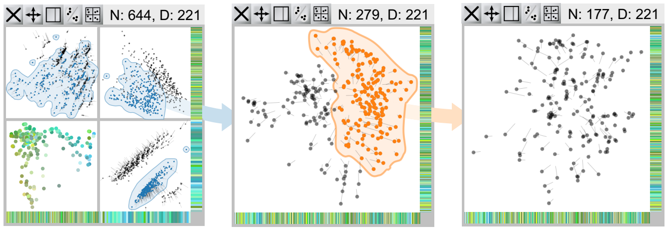


Fig. 8. Child nodes with subset of data items. A region surrounding the selected data items is drawn behind the plots, indicating the data item selection of each child node.

3.4 Node Layout

An automatic method is designed for node layout. As the nodes are organized in a hierarchical manner, traditional tree layout algorithms can be applied. However, our deisgn contains two different kinds of child nodes: one that selects data items, the other that selects dimensions. The algorithm works in a recursive manner. First we put the root node on the top left corner, then place its child nodes of data items below it and the child nodes of dimensions on its right-hand side. Manual layout is also supported and users can move the nodes if needed.

4 Implementation Details

Our interactive Dimension Projection Tree/Matrix can be implemented in many different ways. Our current implementation has a dataset server program and a visualization client program. The server part is written in Python and is responsible for loading the datasets, keeping track of data item and dimension selection, computing the Dimension Projection plots, and performing automatic clustering algorithms. The client program, which is written in C++ and Cocoa Frameworks, presents the visualization and handles user interactions. Communication between the server and the client is achieved through a custom protocol over TCP. Separating the computationally intensive tasks can minimize the CPU and memory usage of the client-side computer. It is also possible to implement the server on a cluster, making it possible to deal with large datasets.

Our method uses classical Multidimensional Scaling (CMDS) to calculate the dimension projection. For dimensions, we set the distance function as one minus Pearson correlation coefficient. For data items, we set the distance function as Euclidean distance. As CMDS is equivalent to PCA when Euclidean distances are used (in terms of dimensionality reduction, PCA can be seen as a particular case of MDS) and, in our cases, the number of dimensions are generally less than the number of points. Therefore, in terms of implementation, PCA is used instead of CMDS. Our method does not limit the dimension projection algorithms; any algorithm that can reduce high-dimensional data to two-dimensional data, such as several kinds of MDS, Kernel PCA, and Isomap, can be used. In practice, the analyst can choose appropriate algorithms for the datasets under analysis.

We used spectrum clustering with kNN graph to perform automatic clustering of data items and dimensions. The distance function is identical to that used in the CMDS. Details about the spectrum clustering algorithm can be found in Luxburg's tutorial [24].

5 Case Study

In this section, we demonstrate the effectiveness of the proposed methods with two real-world datasets. The case studies are done by the paper authors.

5.1 USDA Food Data

First we present a case-study on the USDA food composition dataset. This dataset is used in [33] for their case study, here we demonstrate our prototype system with this dataset, and compare our results with theirs. The dataset is a collection of foods, each dimension represents a certain type of nutrient. After preprocessing, it contains 722 data items and 18 dimensions.

We loaded the preprocessed data into our system. In the main data item plot, we immediately see three clusters, and several groups in the dimension plot (see Figure 9 (a)). We brushed the three clusters with different colors to track them during exploration. In Figure 9 (b), following the guidelines we discussed, we selected two clusters of dimensions and created child nodes for them. We found that the clustering feature changed. For example, the red cluster becomes concentrated under the left node. The blue and green clusters merged into one in the right node. In (c) we demonstrate the matrix representation. The dimensions are divided into four mutually exclusive groups according to the clustering feature of the dimension plot in (a). While observing the same information as in (b), we can also see the Dimension Projection plots of the combined subspaces. In (d) we selected the lower cluster and dragged it out as a new node, we found out that the dimensions Energy, Lipid and Water, became closer to each other in the dimension plot. Thus we grouped them together and separated them with the the remaining dimensions. In the resulting matrix, we found two different ways to cluster the selected set of data items. However, as shown in (e), in the original dataset, if we select the dimensions Energy, Lipid and Water, it is hardly possible to tell the three clusters directly, because other data items changed the clustering feature.

5.2 CADASTER Challege Data

We also conducted a case study on a real world dataset from the CADASTER challenge [1]. The dataset contains the structural information of molecules. Each item in the dataset is a molecule, and each dimension is a SimulationsPlus descriptor, such as the number of atoms, the formal electric charge, or the largest principal moment of inertia. The challenge itself aims to predict the environmental toxicity ($\log(\text{IGC50}^{-1})$) of molecules from these features. In our case study, we wanted to see how environmental toxicity is related to the dimensions and find a model of the dataset through visual analytics. However, the environmental toxicity is a continuous value, which means that the original problem of the challenge is a regression problem.

To make it simpler, we reduced the regression problem to a classification problem by trying only to identify whether the environmental toxicity is a positive number. There are 644 data items in the dataset. Excluding the environmental toxicity, there are 221 numeric dimensions. Figure 10 (a) shows the original Dimension Projection plot of both data items and dimensions.

Because we already know the two classes (positive/non-positive environmental toxicity), we started by coloring the data items with two distinctive colors (see Figure 10 (a)). From the original Dimension Projection plot, the two classes are mixed together, which makes it very difficult to discriminate them directly.

The dimensional space can be clustered into three general clusters; we selected the dimensions in the left part and created a new node for them (see Figure 10 (c)). It then became very clear from the data item plot, that the dataset can be grouped into three clusters, in this subspace. Having identified that most of the blue points are in the first cluster, we dragged this cluster out (see Figure 10 (d)). We then examined the node in (d), split the dimensions into two groups using the clustering feature in the dimension plot. As shown in the Dimension Projection matrix, the bottom-right group appeared to be good for classification, so we dragged its dimensions out to create node (e). We repeated the process in node (e) and found a smaller subspace to discriminate the two classes. This subspace can be used to construct a classifier for the two classes. However, it is not sufficient to only take the dimensions; because we selected a subset of data items before we found the subspace, we must be able to make sure that a new point is in the subset before classifying it with the subspace. Accordingly, we went back to node (c) and tried to identify a smaller subspace that preserved the three clusters. The dimension plot gave us a strong hint to separate the dimensions; we created a child node (i) from the few remaining dimensions and found that it preserved the three clusters. After this step, we were able to perform the classification by first using the subspace in (i) to determine whether the point is in the first cluster, then using the subspace in (f) to separate the two classes.

The classifier we identified in this example is not very accurate, because we only did a rough exploration of the subspaces; for example, the second cluster in (c) also contains two classes and we have not explored yet in this case. However, the classifier we created through visual analytics can be a good starting point for automatic optimization algorithms.

6 Discussion

This section summarizes our design and presents several limitations and possible extensions.

Our proposed visual analytic exploration tool called Dimension Projection Matrix/Tree helps users explore subspaces and subsets of high-dimensional data in a hierarchical way. The tool facilitates the investigation process by providing data item plots and dimension plots for investigating the subspaces and subsets, and interactive ways to construct the hierarchy, both manually and automatically. While the leaf nodes show the structure of a subset of the data, the whole tree depicts the global structure of the investigation process, which reflects the dataset.

There are very few existing works which support subspace exploration for high dimensional data. We compare our approach with the semi-automatical approach proposed by Tatu et. al [33] in 2012. In their approach, SURFING method [6] is used to find out possible subspaces for a given dataset. For example, in the USDA food data, 216 subspaces are found. In their approach, all subspaces are arranged according to clustering or ranking order. Therefore it is very difficult for the user to understand the relationship between different subspaces. In our approach the subspaces are organized as a tree.That is why the relationship between the subspaces can be clearly depicted by the tree structure, which is extremely helpful for the user to navigate and explore. The contexts of parent and child nodes can help the users to create a mental image to position the targeting subspaces or subsets in the overall high-dimensional data space.

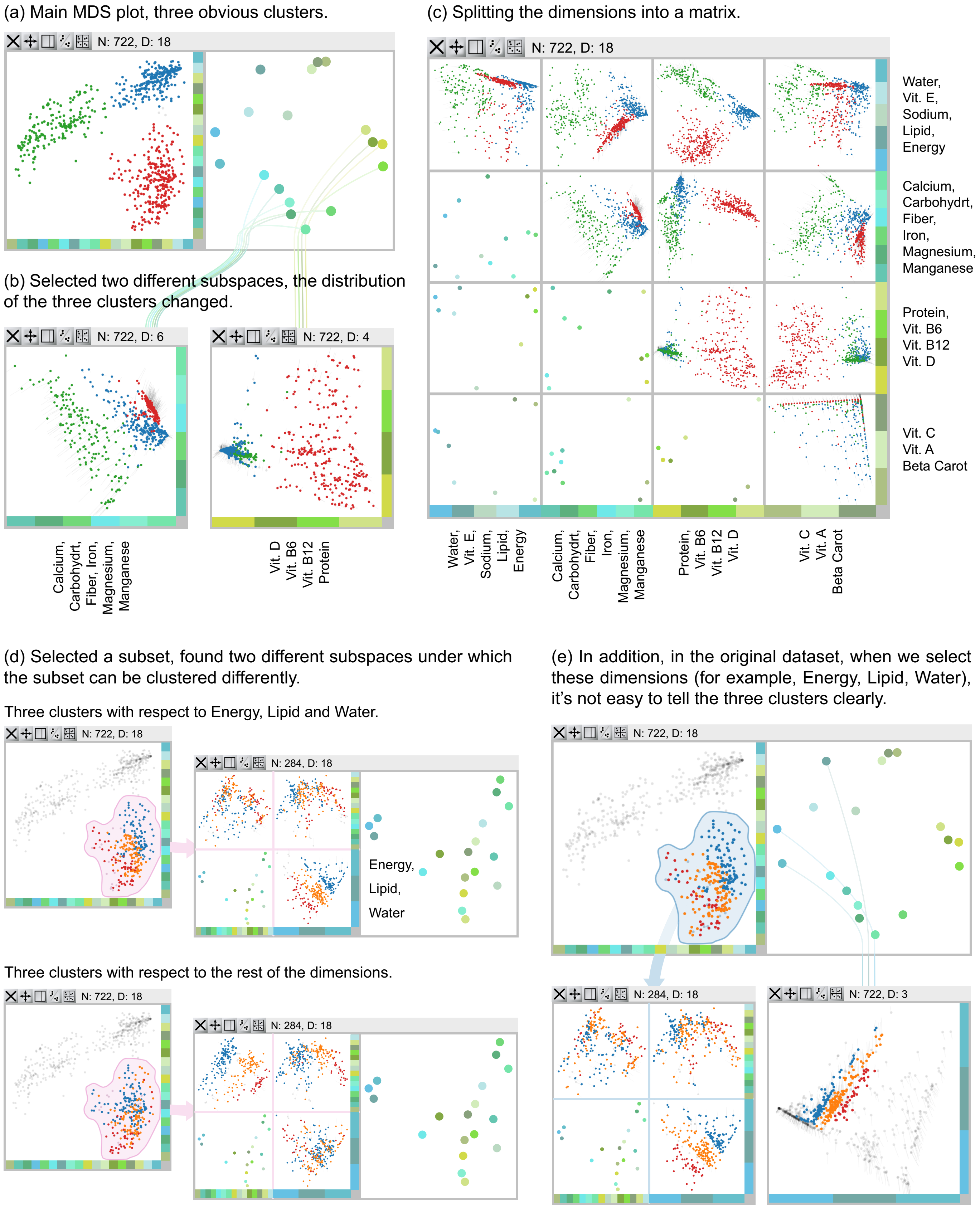
Furthermore, our approach can provide the data subset exploration simultaneously, which is not supported by other methods. Of course, our methods can be further enhanced by the existing semi-automatical approach. Such an algorithm could be used to provide traces and hints for the users to select the next level of subspaces to explore.

Moreover, our design closely integrates the user into the exploration loop. When domain experts are the users, they could directly harness their expertise and domain knowledge to guide their exploration. We plan to conduct a user study with domain experts in the future to verify such a statement and seek a possibly better design of the knowledge integration.

There are several limitations in our current prototype implementation. Below we discuss them and provide possible ways to improve.

Correlation Representation. We used the most basic Pearson correlation coefficient to create the dimension plots. However, numerous interesting correlations are nonlinear, which means they cannot be effectively measured by the Pearson correlation coefficient. In addition, we did not cover correlations between multiple dimensions. From our case studies, we found that the main way to identify the interesting subspaces is to group correlated dimensions together, inspect the effect of grouping with the Dimension Projection Matrix, and then create child nodes for interesting groups. However, the fact that the dimensions are correlated to each other does not imply that the subspace is useful for a certain task, such as classification or clustering. To deal with such tasks, the dimension plot can be enhanced by providing more task-related information about each dimension; for example, the sizes of the points can be useful for representing the dimensions' relevance for classification, which can be measured by mutual information or other metrics. In this way, several variants of our design can be created to support different tasks.

Dimension Projection Algorithms and Performance. As our case studies have shown, we have applied our method on one dataset with 644 items and 221 dimensions. Currently, the data item plots are computed by PCA and the dimension plots are computed by MDS. There



(a) Main MDS plot, three obvious clusters.

(b) Selected two different subspaces, the distribution of the three clusters changed.

(c) Splitting the dimensions into a matrix.

(d) Selected a subset, found two different subspaces under which the subset can be clustered differently.

Three clusters with respect to Energy, Lipid and Water.

Three clusters with respect to the rest of the dimensions.

(e) In addition, in the original dataset, when we select these dimensions (for example, Energy, Lipid, Water), it's not easy to tell the three clusters clearly.

N: 722, D: 18
N: 722, D: 6
N: 722, D: 4
Calcium, Carbohydrt, Fiber, Iron, Magnesium, Manganese
Vit. D Vit. B6 Vit. B12 Protein
N: 722, D: 18
Water, Vit. E, Sodium, Lipid, Energy
Calcium, Carbohydrt, Fiber, Iron, Magnesium, Manganese
Protein, Vit. B6 Vit. B12 Vit. D
Vit. C Vit. A Beta Carot
Water, Vit. E, Sodium, Lipid, Energy
Calcium, Carbohydrt, Fiber, Iron, Magnesium, Manganese
Protein, Vit. B6 Vit. B12 Vit. D
Vit. C Vit. A Beta Carot
N: 722, D: 18
N: 284, D: 18
Energy,
Lipid,
Water
N: 722, D: 18
N: 284, D: 18
N: 722, D: 18
N: 284, D: 18
N: 722, D: 3

Fig. 9. Experiments on the USDA food composition dataset. See section 5.1 for details about the exploration process.

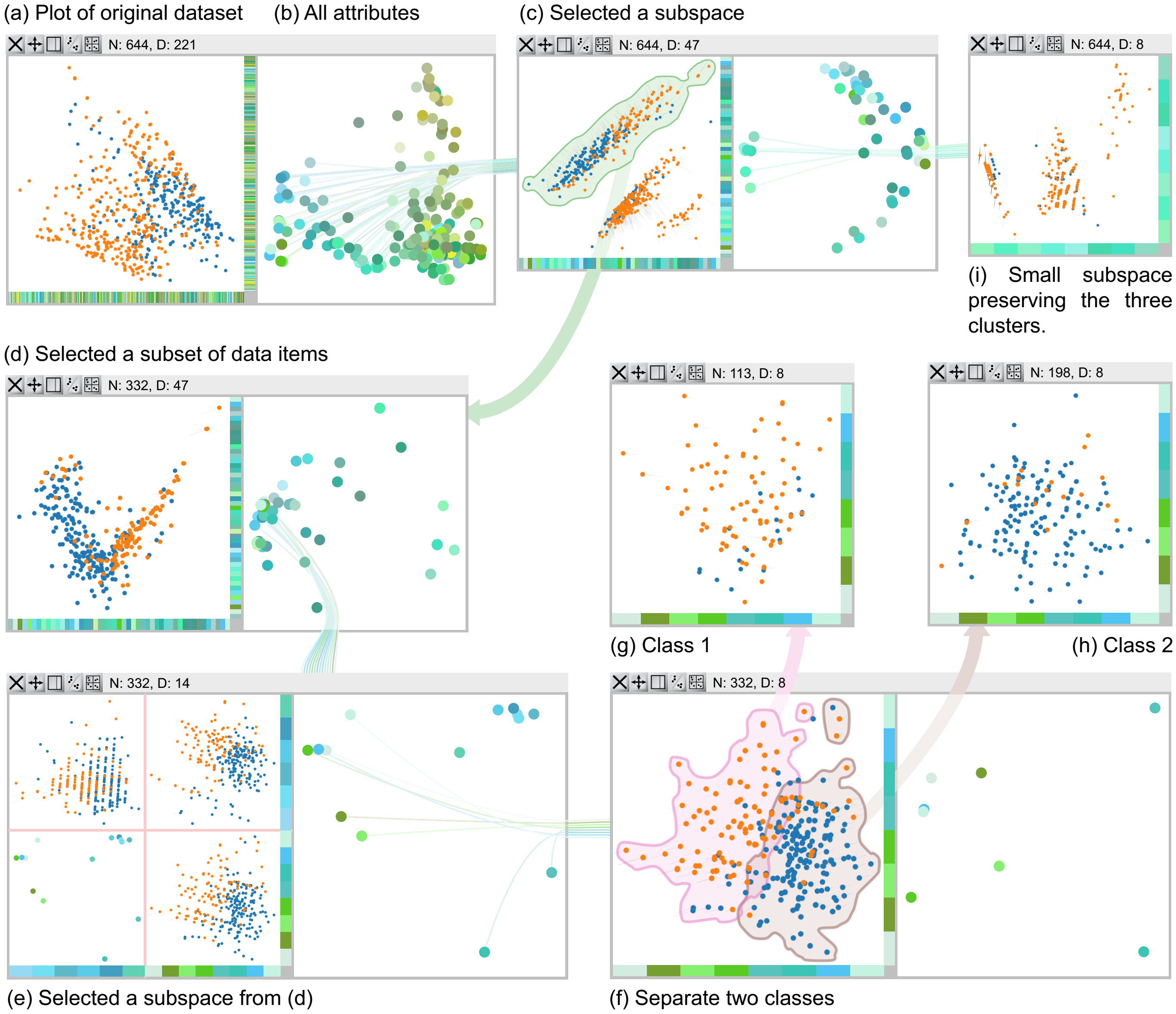

(a) Plot of original dataset
(b) All attributes
(c) Selected a subspace
(i) Small subspace preserving the three clusters.
(d) Selected a subset of data items
(g) Class 1
(h) Class 2
(e) Selected a subspace from (d)
(f) Separate two classes

Fig. 10. Experiments on the CADASTER dataset. (a) shows the plot of the original dataset and the two classes. The tree is constructed in an order of (a), (c), (d), (e), (f), (g), (h) and (i). See section 5.2 for details about the exploration process.

is no limit on which algorithm to use, so the main difficulty is to find suitable methods with which to compute the plots of data items and dimensions. There are many existing algorithms for doing this (see the related work section), each of which has different advantages and disadvantages. Therefore, analysts should find the best algorithm that can output meaningful plots of the dataset being analyzed, as well as keeping the response time acceptable. The performance and scalability is closely related to the dimension projection algorithm we choose. However, when there are numerous data items or dimensions, the rendering itself can take a long time; this issue can be solved by computing a density map with the appropriate resolution to replace the current scatterplot representation.

7 Conclusion

This paper has presented approaches called Dimension Projection Matrix/Tree for visualizing high-dimensional datasets. Our approach constructed a tree of Dimension Projection nodes in which each node corresponds to a selection of the dataset. The selections are represented as a Dimension Projection Matrix with MDS projections of the data items on split dimensions. We have designed a flexible user interface for user interaction and exploration. Our tool allows users to explore complex datasets with both data item aspect and dimension aspect.

Our methods share the same aspect of grouping dimensions into a hierarchy as Yang et al.'s work on Visual Hierarchical Dimension Reduction (VHDR) [42] for exploration of high-dimensional datasets. However, our design emphasizes investigating the data from the perspective of both data items and dimensions. Our approach is complementary to other high-dimensional data visualization methods.

In the future, we plan to integrate other techniques in high-dimensional data visualization into our methods and may include advanced data analysis methods in statistics. Further investigation on the effectiveness of our approach also requires a formal user study.

Acknowledgments

The authors wish to thank Andrada Tatu and Daniel Keim for providing their preprocessed USDA Food Dataset. We are thankful to the anonymous reviewers for their helpful comments. This work was supported by NSFC No. 61170204 and NSFC Key Project No. 61232012.

REFERENCES

[1] Cadaster challenge dataset (simulationplus, training). http://www.cadaster.eu/node/65.

[2] C. C. Aggarwal, J. L. Wolf, P. S. Yu, C. Procopiuc, and J. S. Park. Fast algorithms for projected clustering. *SIGMOD Rec.*, 28:61–72, 1999.

[3] R. Agrawal, J. Gehrke, D. Gunopulos, and P. Raghavan. Automatic subspace clustering of high dimensional data for data mining applications. *SIGMOD Rec.*, 27:94–105, 1998.

[4] D. Asimov. The grand tour: a tool for viewing multidimensional data. *SIAM J. Sci. Stat. Comput.*, 6(1):128–143, 1985.

[5] W. Basalaj. Incremental multidimensional scaling method for database visualization. In *Proceedings of SPIM Visual Data Exploration and Analysis VI*, pages 149–158, 1999.

[6] C. Baumgartner, C. Plant, K. Railing, H.-P. Kriegel, and P. Kroger. Subspace selection for clustering high-dimensional data. In *Data Mining, 2004. ICDM'04. Fourth IEEE International Conference on*, pages 11–18. IEEE, 2004.

[7] R. A. Becker and W. S. Cleveland. Brushing scatterplots. *Technometrics*, 29(2):127–142, 1987.

[8] S. L. Crawford and T. C. Fall. Projection pursuit techniques for the visualization of high dimensional datasets. *Visualization in Scientific Computing*, pages 94–108, 1990.

[9] A. Dasgupta and R. Kosara. Pargnostics: Screen-space metrics for parallel coordinates. *IEEE Trans. Vis. Comput. Graph.*, 16(6):1017–1026, 2010.

[10] N. Elmqvist, P. Dragicevic, and J.-D. Fekete. Rolling the dice: Multidimensional visual exploration using scatterplot matrix navigation. *IEEE Trans. Vis. Comput. Graph.*, 14(6):1141–1148, 2008.

[11] N. Elmqvist and J.-D. Fekete. Hierarchical aggregation for information visualization: Overview, techniques, and design guidelines. *IEEE Trans. Vis. Comput. Graph.*, 16(3):439–454, 2010.

[12] J. Faith. Targeted projection pursuit for interactive exploration of high-dimensional data sets. In *Proceedings of the 11th International Conference Information Visualization*, pages 286–292, 2007.

[13] Y.-H. Fua, M. Ward, and E. Rundensteiner. Hierarchical parallel coordinates for exploration of large datasets. In *Proceedings of the IEEE Visualization'99*, pages 43 –50, 1999.

[14] S. Ingram, T. Munzner, and M. Olano. Glimmer: Multilevel mds on the gpu. *IEEE Trans. Vis. Comput. Graph.*, 15(2):249–261, 2009.

[15] A. Inselberg. The plane with parallel coordinates. *The Visual Computer*, 1(2):69–91, 1985.

[16] A. Inselberg and B. Dimsdale. Parallel coordinates: a tool for visualizing multi-dimensional geometry. In *Proceedings of the IEEE Visualization'90*, pages 361–378, 1990.

[17] S. Johansson and J. Johansson. Interactive dimensionality reduction through user-defined combinations of quality metrics. *IEEE Trans. Vis. Comput. Graph.*, 15(6):993–1000, 2009.

[18] J. Jolliffe. *Principal Componenet Analysis*. Springer Verlag, 1986.

[19] K. Kailing, H. peter Kriegel, and P. Kröger. Density-connected subspace clustering for high-dimensional data. In *Proc. SDM.*, pages 246–257, 2004.

[20] D. Keim, H.-P. Kriegel, and M. Ankerst. Recursive pattern: a technique for visualizing very large amounts of data. In *Proceedings of the IEEE Visualization'95*, pages 279 –286, 463, 1995.

[21] T. Kohonen. The self-organizing map. *Proceedings of the IEEE*, 78(9):1464 –1480, 1990.

[22] A. Lex, M. Streit, C. Partl, and D. Schmalstieg. Comparative analysis of multidimensional, quantitative data. *IEEE Trans. Vis. Comput. Graph.*, 16(6):1027–1035, 2010.

[23] T. V. Long and L. Linsen. Multiclustertree: Interactive visual exploration of hierarchical clusters in multidimensional multivariate data. *Comput. Graph. Forum*, 28(3):823–830, 2009.

[24] U. Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416, 2007.

[25] P. McLachlan, T. Munzner, E. Koutsofios, and S. North. Liverac: interactive visual exploration of system management time-series data. In *Proceedings of the ACM CHI'08*, pages 1483–1492, 2008.

[26] E. Müller, S. Günnemann, I. Assent, and T. Seidl. Evaluating clustering in subspace projections of high dimensional data. *Proc. VLDB Endow.*, 2:1270–1281, 2009.

[27] L. Parsons, E. Haque, and H. Liu. Subspace clustering for high dimensional data: a review. *SIGKDD Explor. Newsl.*, 6(1):90–105, 2004.

[28] H. Piringer, M. Buchetics, H. Hauser, and E. Gröller. Hierarchical difference scatterplots interactive visual analysis of data cubes. In *Proceedings of the ACM SIGKDD Workshop on VAKD'09*, pages 56–65, 2009.

[29] R. Rao and S. K. Card. The table lens: merging graphical and symbolic representations in an interactive focus + context visualization for tabular information. In *Proceedings of the ACM CHI'94*, pages 318–322, 1994.

[30] J. Seo and B. Shneiderman. A rank-by-feature framework for unsupervised multidimensional data exploration using low dimensional projections. In *Proceedings of the IEEE InfoVis'04*, pages 65–72, 2004.

[31] M. Sifer. User interfaces for the exploration of hierarchical multi-dimensional data. In *Proceedings of the IEEE VAST'06*, pages 175 –182, 2006.

[32] A. Slingsby, J. Dykes, and J. Wood. Configuring hierarchical layouts to address research questions. *IEEE Trans. Vis. Comput. Graph.*, 15(6):977–984, 2009.

[33] A. Tatu, F. Maas, I. Farber, E. Bertini, T. Schreck, T. Seidl, and D. Keim. Subspace search and visualization to make sense of alternative clusterings in high-dimensional data. In *Visual Analytics Science and Technology (VAST), 2012 IEEE Conference on*, pages 63–72. IEEE, 2012.

[34] E. J. Wegman. Hyperdimensional data analysis using parallel coordinates. *the American Statistical Association*, 411(85):664–675, 1990.

[35] E. J. Wegman and Q. Luo. High dimensional clustering using parallel coordinates and the grand tour. *Computing Science and Statistics*, 28:352–360, 1997.

[36] L. Wilkinson, A. Anand, and R. Grossman. Graph-theoretic scagnostics. In *Proceedings of the IEEE InfoVis'05*, pages 157 – 164, 2005.

[37] M. Williams and T. Munzner. Steerable, progressive multidimensional scaling. In *Proceedings of the IEEE InfoVis'04*, pages 57–64, 2004.

[38] P. C. Wong and R. D. Bergeron. Multivariate visualization using metric scaling. In *Proceedings of the IEEE Visualization'97*, pages 111–118, 1997.

[39] J. Yang, A. Patro, S. Huang, N. Mehta, M. Ward, and E. Rundensteiner. Value and relation display for interactive exploration of high dimensional datasets. In *Proceedings of the IEEE InfoVis'04*, pages 73–80, 2004.

[40] J. Yang, W. Peng, M. O. Ward, and E. A. Rundensteiner. Interactive hierarchical dimension ordering, spacing and filtering for exploration of high dimensional datasets. In *Proceedings of the IEEE InfoVis'03*, pages 105–112, 2003.

[41] J. Yang, M. O. Ward, and E. A. Rundensteiner. Interring: An interactive tool for visually navigating and manipulating hierarchical structures. In *Proceedings of the IEEE InfoVis'02*, pages 77 – 84, 2002.

[42] J. Yang, M. O. Ward, E. A. Rundensteiner, and S. Huang. Visual hierarchical dimension reduction for exploration of high dimensional datasets. In *VisSym '03: Proceedings of the symposium on Data visualisation 2003*, pages 19–28, 2003.

[43] X. Yuan, P. Guo, H. Xiao, H. Zhou, and H. Qu. Scattering points in parallel coordinates. *IEEE Trans. Vis. Comput. Graph.*, 15(6):1001–1008, 2009.