SeekAView: An Intelligent Dimensionality Reduction Strategy for Navigating High-Dimensional Data Spaces

Josua Krause, Aritra Dasgupta, Jean-Daniel Fekete, Enrico Bertini

SeekAView: An Intelligent Dimensionality Reduction Strategy for Navigating High-Dimensional Data Spaces

Josua Krause, Aritra Dasgupta, Jean-Daniel Fekete, and Enrico Bertini

Abstract— Dealing with the curse of dimensionality is a key challenge in high-dimensional data visualization. We present *SeekAView* to address three main gaps in the existing research literature. First, automated methods like dimensionality reduction or clustering suffer from a lack of transparency in letting analysts interact with their outputs in real-time to suit their exploration strategies. The results often suffer from a lack of interpretability, especially for domain experts not trained in statistics and machine learning. Second, exploratory visualization techniques like scatter plots or parallel coordinates suffer from a lack of visual scalability: it is difficult to present a coherent overview of interesting combinations of dimensions. Third, the existing techniques do not provide a flexible workflow that allows for multiple perspectives into the analysis process by automatically detecting and suggesting potentially interesting subspaces. In *SeekAView* we address these issues using suggestion based visual exploration of interesting patterns for building and refining multidimensional subspaces. Compared to the state-of-the-art in subspace search and visualization methods, we achieve higher transparency in showing not only the results of the algorithms, but also interesting dimensions calibrated against different metrics. We integrate a visually scalable design space with an iterative workflow guiding the analysts by choosing the starting points and letting them slice and dice through the data to find interesting subspaces and detect correlations, clusters, and outliers. We present two usage scenarios for demonstrating how *SeekAView* can be applied in real-world data analysis scenarios.

Index Terms—High-Dimensional Data, Subspace Exploration, Guided Visualization

1 Introduction

Let's start with an example scenario where a health-care analyst working with a care provider company is confronted with patient data, containing millions of hospital visit records, and hundreds of dimensions describing conditions and demographic information of patients. Her goal is to find patient groups with similar disease conditions or similar outcomes, to treat them in a similar way; either this group is "well managed", or that one "is suffering from lack of resources in the hospitals", or another one is "having an abnormal mortality rate". These outcomes are quite different in the dimensions and values that reveal them. Therefore, the analyst will need to gather different set of dimensions, filter over different values or categories, to find *subspaces* exhibiting interesting properties for her questions. Looking at all the dimensions for all the records at the same time is not only inefficient, but also unlikely to answer her questions.

For our work, we are interested in exploring datasets containing numerical or categorical dimensions, with *n dimensions* $D = \{d_1, \cdots, d_n\}$ (or *features*) and *m items* (also called *records*, or *data points*) $R = \{r_1, \cdots, r_m\}$. The basic premise of our work is to let analysts build interesting k dimensional *subspaces* (kD) out of the n-dimensional data space (nD), where $k \ll n$. For our purpose, k should be less than 20 to be visualized using either parallel coordinates or a scatter plot matrix; n is typically in the hundreds. Each subspace is usually only interesting for a filtered selection i over the m possible records, so analysts need to slice and dice through the dimensions and items to build interesting subspaces, where each subspace is a couple (K, I) where $K_{j \in [1,\ldots,n]} = \{d_j\}$ and $I_{l \in [1,\ldots,m]} = \{r_l\}$.

While using automated approaches such as subspace clustering [17] is an option, these automatic methods give no control over the analytical process to the analyst but only provide a list of subspaces that may contain interesting combinations of dimensions and values. Besides, subspaces clustering will only find one kind of pattern in multidimensional data: dense areas, while experts are interested in many more, such as correlated dimensions, or uncorrelated ones, dimensions following visible patterns, or exhibiting outliers. For addressing these problems, researchers have argued for a human-in-the-loop approach towards subspace exploration leading to more intelligent dimensionality reduction strategies [22].

To this end, we propose *SeekAView*, a visual analytics system that leverages a transparent and flexible process-oriented approach for letting analysts build subspaces out of a high-dimensional data set. *SeekAView* is able to show hundreds of dimensions without reduction and provide suggestions to the analysts for letting them reconfigure different views, build interesting subspaces according to multiple kinds of patterns, and iteratively refine them based on their evolving analysis questions.

In this paper, we identify three main research gaps we aim to fill through *SeekAView*:

1. The lack of *scalability* of existing visualizations in the number of dimensions. By focusing on 1D plots as starting point and guiding the user in the construction and exploration of nD subspaces, we enable the analysis of large sets of combinations without overwhelming the user. Even simple visual representations like scatter plot matrices grow quadratically with the number of dimensions and can quickly become unmanageable. We support the analysts in tasks such as: *"I want to look at an overview of all dimensions and find combinations revealing interesting patterns to explore in multidimensional views"*.
2. The lack of *transparency* [6] in methods that rely exclusively on dimensionally reduced dimensions/axes, such as PCA, t-SNE [32], or MDS [18]. By making 1D plots of the original dimensions the main driver behind all visualizations and interactions and by letting the analyst interact with the output of the automated methods we increase transparency throughout the analysis process. This means the analyst can always ask questions like: *"Which of the original dimensions are involved in this pattern?"* and *"What is it going to happen if I remove/add this group of dimensions from the visualization?"*
3. Provide a *flexible* analytical workflow, that allows to enter the visual analytics loop through multiple starting points of analysis. By getting suggestions on the potential next steps of their

- *Josua Krause is with NYU. E-mail: josua.krause@nyu.edu.*
- *Aritra Dasgupta is with Pacific Northwest National Laboratory. E-mail: aritra.dasgupta@pnnl.gov*
- *Jean-Daniel Fekete is with INRIA. E-mail: Jean-Daniel.Fekete@inria.fr.*
- *Enrico Bertini is with NYU. E-mail: enrico.bertini@nyu.edu.*

Manuscript received 31 March 2007; accepted 1 August 2007; posted online 2 November 2007.
For information on obtaining reprints of this article, please send e-mail to: tvcg@computer.org.

analysis at every stage of their interaction analysts can have have different starting points: *"I want to focus on this category and reconfigure different views for finding clusters"* or *"Suggest interesting subspaces to me and I refine them by using my domain knowledge"*.

Specifically, our contributions in this paper are threefold: i) we propose a framework for allowing analysts to build subspaces in a human-guided fashion with machine-generated suggestions—faithful to the visual analytics process; the analysis can start either from a set of suggestions from algorithms, or from a view selected by the analyst, and enter into the visual analysis loop then; ii) we focus on enabling key analysis tasks through transparent operations that analysts can understand and steer at any time; and iii) we provide a system that can effectively deal with high-dimensional spaces around hundred dimensions without information loss or dimensionality reduction.

2 Related Work

High-dimensional data analysis has a long history where different approaches have been pursued to handle the *curse of dimensionality* problem [5, 10, 36]. In our work, we focus exclusively on the premise of generating interesting multidimensional views out of high-dimensional data, commonly known as *subspaces* and letting analysts refine and reconfigure those subspaces. There are three main threads of research relevant to the work reported here, which are as follows:

2.1 Subspace Clustering and Search

Automated subspace search methods being heavily parameterized [17], visual analytics techniques have been proposed for handling this problem. Some of these methods focused on helping analysts select interesting feature subsets [9,21] by interacting with the algorithm outputs. In these cases, the user is in control of the modification of the generated feature subset by being able to reconfigure them through addition or removal of features. These techniques are focused on displaying the outputs of the algorithms and not on the process. Users can modify the output, but there is no guidance from the system in being able to select more features for building different views. Moreover, they also do not allow users to manipulate the item space that would trigger a recomputation of the algorithms in the dimensions space. Another approach to this problem has been to devise intelligent subspace search methods [25], where subspace clusters are shown to the user. This approach has high visual scalability in being able to provide an overview of the subspaces to the user. However, this approach also suffers from a lack of flexibility in letting the user modify the subspaces by either getting help from the system or by incorporating her domain knowledge. Our framework also displays outputs of subspace clustering algorithms, but lets the user reconfigure the output views through different suggestions. An exploratory approach towards generating interesting subspaces have been proposed by Wang et al. by clustering dimensions [33] or showing values and relations at once in projected data spaces [37] have also been proposed. However, the method is limited to a small number of dimensions.

2.2 Quality Metric Aided Visual Exploration

The problem with traditional techniques, such as scatter plot matrices [8], parallel coordinates [12,34], or radviz [27], is that they mostly do not scale beyond 10 to 20 dimensions. While there has been recent approaches to make scatter plot matrices more scalable [19], they can be visually complex for analysts to understand and navigate without any guidance. Moreover, most visual representations for high-dimensional data exclusively focus on numerical attributes. For alleviating the scalability problem, researchers have proposed pixel-based displays [16], which are able to preserve the fidelity of the data. However in such a high-density display, showing relationships and guiding user interactions to selectively build views can be challenging.

Researchers have also looked into augmenting high-dimensional data exploration with visual feedback [25]. Approaches such as using feature statistics for producing ranked views have been proposed [24],

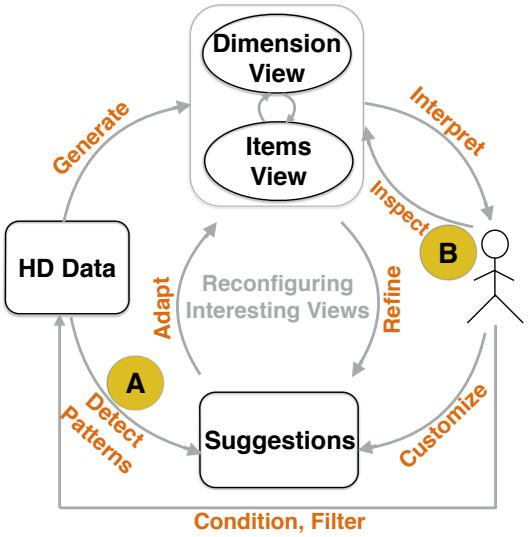

Fig. 1. **Framework for guided reconfiguration of interesting views out of high-dimensional data**: A tight integration of system generated suggestions with analysts' actions leads to flexibility in having multiple starting points of analyses (A and B), and transparency in being able to refine the suggestions and adapt both dimension and item space interactively.

however they do not focus on guiding the user to build multivariate views. Analysts often need guidance on finding the salient hidden relationships in large high-dimensional data spaces. Quality metrics [7,23,25,35] have been proposed to deal with this issue by quantifying various data-space and screen-space relationships. Most of them however focus on two-dimensional relationships and do not let refinement of the metrics with respect to conditioning of the data.

2.3 Dimensionality Reduction Strategies

For handling the curse of dimensionality problem, one solution is to use dimensionality reduction methods like PCA, t-SNE [32], or MDS [18]. However, with domain experts, reduction is often not an option as analysts need to use their domain knowledge to understand which combinations of dimensions make sense. They need a faithful representation of the data space with minimal abstraction and information loss.

Moreover, dimensionality reduction supports mainly one visualization task, that is, finding uniform groups of clustered objects, which covers only one of the possible visualization tasks with high-dimensional data visualization. Furthermore, plots obtained through linear or nonlinear transformations are hard to interpret as dimensionality reduction works in a black box fashion. In the not too rare case in which dimensionality reduction returns results that are hard to interpret (e.g., a uniform cloud of points with no clear groupings), it is not clear whether the results reflect a real characteristic of the data or rather the method itself fails to represent the underlying trends correctly (that is, it's a false negative) [4]. Even more problematic, there is a lack of techniques able to guide the user through steps that can help disambiguate this situation with only a few exceptions [11].

For a visual approach to dimensionality reduction, researchers have proposed techniques like hierarchical clustering and interactive reduction of the search space by using information loss metrics [13]. Similar to the drawbacks of subspace search approaches, these exploratory methods are also focused on the results of the metrics and less focused on the process. A notable exception is the dual analysis model proposed by Turkay et al. [29], who allow a tight interaction between the item space and the dimension space. They are however, not focused on producing subspaces of dimensionality more than 2 and also do not follow a transparent approach of showing the properties of the dimensions themselves.

3 Conceptual Framework for Building Subspaces

The focus of our framework is not only on the output of *kD* subspaces, but also on the transparent analytical process that analysts can follow to produce that output. As shown in Figure 1, we adapted the visual analytics schema proposed by Keim et al. [15] in the special case of subspace visualization. The framework aims to synergize user interactions with system generated suggestions for discovering and reconfiguring interesting multidimensional views. The reconfiguration process involves manipulating the dimension space and the item space for building subspaces with guidance from the system, using different strategies and flexible starting points. Additionally, we incorporate the dual analysis model proposed by Turkay et al. [29] with the suggestion system and allow for multiple paths through the schema.

Formally, our framework generates subspaces (K, I, C, T) that extend the standard subspaces (K, I) described in Section 1. C is a partition over the filtered selection I, and corresponds to color associated with bins of values. T is a textual description provided by a user to document what is interesting in the subspace or any kind of annotation. Our framework is meant to produce colored bins of items with a set of dimensions and a textual representation. Dimensions are typically numerical or categorical and then usually mapped to a numerical value. We call *conditioning* as the transformations of the values of a dimension; it can be a *normalization* that will map a given range of values in the $[0,1]$ range, or a scaling that will transform values by applying e.g. a log or exponential function to them, or any combination thereof. In the rest of this paper, we assume that, when mentioned, conditioning is applied both for visualization and computation. For example, refer to the view in Figure 2. It contains of a set of nine dimensions, a coloring of four colors representing clusters in the item space, and an annotation describing the view.

As seen in Figure 1, there can be different starting points of analyses (A and B). Consider for instance a nutritionist wants to study the correlation patterns of nutrients among food items such as dairy products, meat products, etc.. She would like to know: *which food products are similar in containing certain types of nutrients?* ($Q1$), *which dimensions are redundant?* ($Q2$), *which combination of nutrients are highly related to high Vitamin D content?* ($Q3$), *which combination of nutrients separate food groups the most?* ($Q4$) etc.. Using our framework, analysts can either start from system generated suggestions (scenario A) or start by interacting through the visualization and then get help from the system on further analysis and exploration steps (scenario B). For supporting such flexible analysis, the *suggestion* component adapts to the different starting points and help analysts *refine* the subspaces by *navigating* and *filtering* both the dimension and item spaces.

Suggestions: The suggestions encapsulate a set of metrics for guiding analysts towards finding interesting groups of dimensions. They also adapt to different starting points (subspaces and selections) an analyst can pursue. In case of the questions $Q1$ and $Q2$, the nutritionist does not have a starting point of the analysis and is just interested in finding interesting combinations of dimensions. In such cases, the nutritionist needs the system to guide her to potentially interesting patterns. This refers to the starting point A in Figure 1. In that case the system helps the analysts detect patterns by suggesting different subspaces which can be further iterated upon by the analysts using conditioning of the data and refinement of the generated subspaces (Figure 3). In our current implementation, the subspaces are generated by using either *subspace clustering* ($Q1$), generating groups of *correlated dimensions* ($Q2$), or manually selecting dimensions of interest.

In case of the questions $Q3$ and $Q4$, the nutritionist knows her starting point of analysis and she would focus on a given dimension in food products for visually inspecting relevant nutrients. Pure visual inspection is difficult though, as the larger the dimensionality (nutrients) and large data size (the number of food items), the larger the search space becomes. This refers to the starting point B in the figure. The analyst need guidance in reducing the search space for selecting interesting dimensions (Figure 5). This is supported by computing metrics for quantifying relationships between distributions i.e. *distribution anomalies* ($Q3$) and computing metrics for finding subspaces which *separate classes* well ($Q4$). We refer to this method as *dimension filters*.

Refinement: Once subspaces are generated by automated methods, or interesting views are suggested to the analyst, our framework allows refinement of the produced output in different ways. For example in the scenarios mentioned above, the nutritionist can either try to incorporate her domain knowledge for refining the subspaces, e.g. removing uninteresting dimensions, or get help from the system through suggestions on how to refine the subspaces. The motivation of this task is similar to that in the dual analysis model proposed by Turkay et al. [29]. But in our case we provide higher transparency in letting users manipulate the data space and directly observe how the operations affect the relationship among dimensions and items. To ease visual inspection and steer analyses, analysts can *adapt the scales* of the different dimensions based on visual inspection of the distributions to either logarithmic scale or linear scale. One of the main refinement strategies is *adding or removing dimensions* to the subspace of interest based on the system suggestions. For example, by finding dimensions with our dimension filters, analysts can choose to add those dimensions to a subspace to investigate further their influence. Similarly they can remove dimensions which are of little or no interest to their analysis. Another refinement strategy is to condition the item space based on automatic or visual detection and *removal of outliers*. After modifying the item space, recomputations of suggestions can be triggered and different subspaces or dimensions are again suggested to the analyst, steered from to their interactive selections.

4 *SeekAView*: Instantiating the Subspace Exploration Framework

In this section we describe *SeekAView*, our implementation of the framework we described above. We first describe the visualization design and user interface and then ways of building views and the mechanisms used for suggesting and refining them. In this we cover four main data analysis tasks, namely: subspace clustering, correlation, anomaly detection, and class separation.

4.1 Visual Representations and User Interface

SeekAView is split into different panels (see Figure 2). On the left is a simple list showing all dimensions by name while the center shows all dimensions as small multiple frequency plots. On the right a magnified frequency plot shows the currently selected dimension. This target plot can be used to define targets by brushing. Targets are couples (I, C) of an item space I with a partition C of its items. This partition is called coloring of the items and is limited to up to 6 buckets i.e. colors (5 brush colors and the remaining non-brushed items). At the bottom a view generated via suggestions and refinements, as explained in Section 4.3, is shown in a number of multidimensional visualizations, including a PCA scatter plot, parallel coordinates, and a scatter plot matrix. Those views can be saved and annotated in the view manager at the bottom right.

The design space of our framework facilitates progression from the selection of dimensions to defining targets and building multivariate views of the data. Three high-level goals motivate our key design choices: i) *scalability* with respect to a high number of dimensions (in the order of tens and hundreds), ii) the *flexibility* to represent mixed attribute types, like numerical, ordinal, categorical, or binary data, and iii) *transparency* with respect to user interaction for showing interesting subspaces and targets and letting analysts build multivariate views. In this section, we describe how our design choices satisfy these three criteria.

Small Multiple Frequency Plots: Unlike conventional high-dimensional data representations like scatter plot matrices and parallel coordinates which focus on showing pairwise relationships between features, our design is motivated by the fact that the starting points of an analysis is often exploring the 1D distributions. To this effect, we display small multiples [28] of each dimension. Within each small multiple, as shown in Figure 4, we use a beanplot-like layout [14], which is a frequency-based representation of the attributes. Therefore, we name the plots as small multiple frequency plots. For numerical variables, the y-axis of each plot represents the frequency of the data

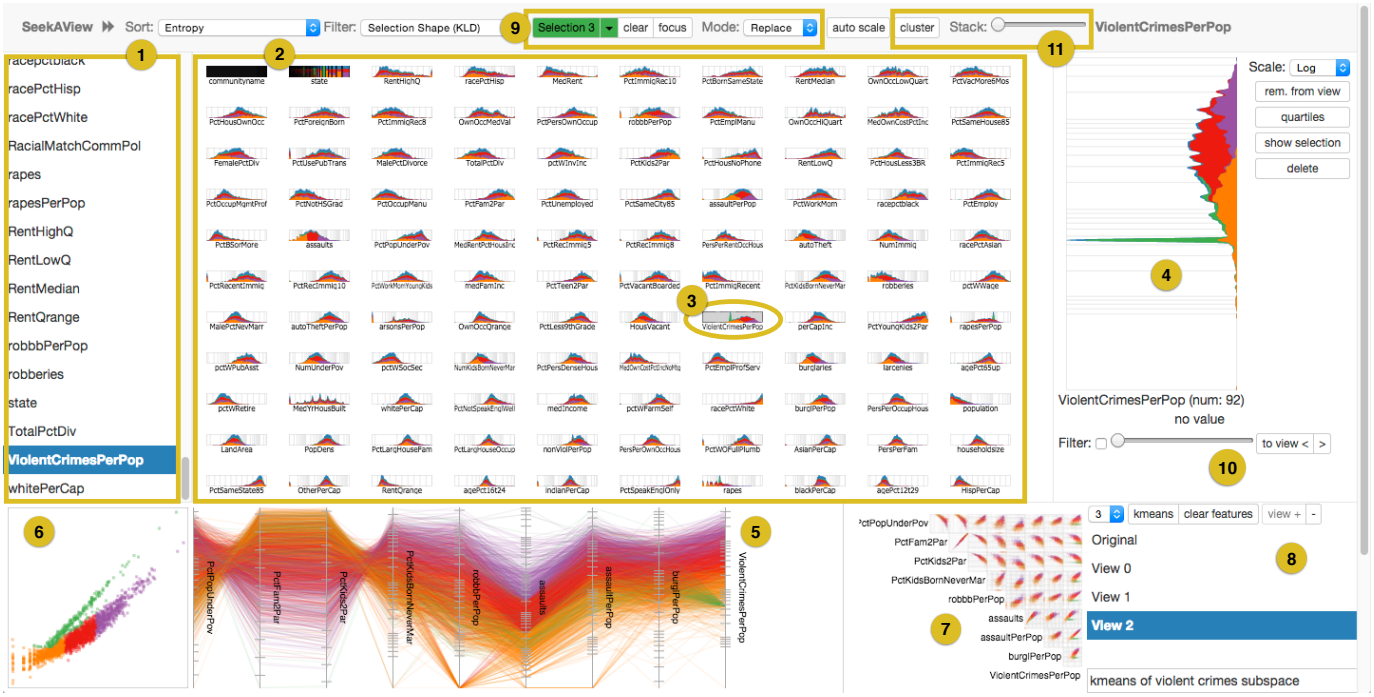

Fig. 2. ***SeekAView*** The user interface of *SeekAView*. (1) is the list of all dimensions while (2) shows them as small multiple frequency plots. Selected dimensions are high-lighted (3) and shown in the target plot (4). The current multivariate view can be seen in the multidimensional view panel consisting of parallel coordinates (5), a PCA scatter plot (6), and a scatter plot matrix (7). Views can be selected, created, removed, and annotated in the lower right (8). Manual brushing can be configured at the top (9). Subspace suggestions can be created with a one-to-many filter (10) or by computing subspace ("cluster") or dimension similarity ("stack") clusters (11). The current state is the end of the Communities and Crime case study (see Section 5.1) finding dimensions related to "Violent Crimes per Pop." and further clustering the item space.

values. A frequency-based representation provides the flexibility to meaningfully represent multiple attribute types. For ordinal or binary variables, we associate numerical values to each distinct value and for categorical variables, we divide the space into rectangles. The number of rectangles is equal to the number of categories of the feature and the area of each rectangle is scaled by the proportion of data points belonging to that category. Since there is no inherent order in the categories we order the categories by their frequency so that the most common category, i.e. with the most data points, is on top of the plot. The main motivation for showing the frequency was two-fold: i) each small multiple would have a very small resolution, therefore showing data items would only produce visual clutter, and ii) from a visual saliency point-of-view, detecting and comparing continuous shapes is relatively easier as deduced from the Gestalt principle of continuity [2]. This and the fact that we cannot assume that dimensions are normal distributed is the reason why our representation is superior to e.g. box-plots. In a small multiple setting they would not produce salient distinguishable features and would not be visually scalable. Compared to, for example, scatter plot matrices, which grow quadratically, 1D representations grow linearly with the number of features.

Target Plot: The small multiple plots are adaptive to user selections and let analysts select target dimensions for further exploration. These target dimensions then provide a focal point of analysis, while the small multiples serve as the context. Interaction between the two provides a focus-and-context analysis paradigm to the analysts in the high-dimensional data space, where they can explore multivariate relationships in a highly scalable manner. The target plot is a magnified frequency plot of the dimension. As small multiple frequency plots are displayed too small to add axes descriptions the user can hover over data points to see which value it corresponds to.

Multivariate Views Panel: In the multivariate views panel a number of multidimensional visualizations show the current view created via suggestions and refinements (Section 4.3). The visualizations include a scatter plot showing the first two principal components of the dimension subspace, parallel coordinates, and a scatter plot matrix showing all pair-wise relations between dimensions of the subspace. The usage of those visualizations is feasible since the number of dimensions here is small compared to the overall number of dimensions. The scatter plot matrix helps users detect correlations and redundant dimensions which then can be removed if desired. As the screen space allotted for the scatter plot matrix is comparatively small the user can double click on the matrix to get a full page view of it. Analysts can brush on the parallel coordinates and scatter plots much like in the target plot and select dimensions like on the small multiples panel.

View Manager: The multivariate views can be saved and annotated in the view manager on the bottom right of the user interface. A text box allows to write comments about a view for future reference. The analyst can switch between created views also to go back steps or revisit findings that were made earlier.

The views can be created by the analyst using two strategies. Manual curation of subspaces by adding or removing single dimensions or a suggestion & refinement based approach.

4.2 Manual Curation of Subspace Views

Analysts can manually curate views with user interaction. We provide some statistical measures to ease this task and also the means of defining targets and filtering the dimension space as well as the item space.

Sorting Small Multiples: Uniform and non-uniform distributions are of key interest to analysts. Uniform distributions usually denote high uncertainty or high entropy while non-uniform distributions usually denote low entropy. We compute the entropy of the dimensions using a pixel binning strategy. The number of bins is equal to the number of pixels in the magnified small multiple frequency plot, where the frequency variation can be seen at a higher resolution. In case of non-uniform distributions, it is useful for analysts to know whether high or low values dominate, i.e., whether there is a positive or negative skew. Skewed distributions are also easy to spot even in very low resolution. Therefore we use skewness as an interestingness measure for quantifying distributions. We provide both entropy and skewedness as a means to sort small multiple frequency plots. This points users to interesting dimensions for further inspection.

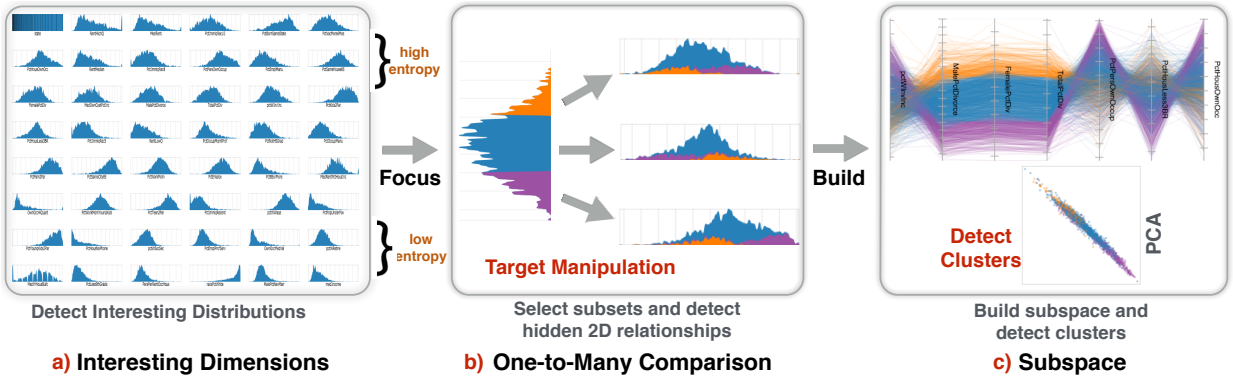

Fig. 3. **Workflow: Building through dimensions of interest.** With the help of the small multiple frequency plots of all dimensions analysts can detect dimensions with interesting distributions (a). Those dimensions can then be used to define targets that can be used as input for automatic suggestion mechanisms (b). This results in an automatically generated view illustrating interesting patterns (c).

Scaling Dimensions: A further form of dimension statistic mechanism is automatically deciding which scale (i.e. linear or logarithmic) to use for the representation of the data values. This conditioning reveals different insights about a dimension. The system provides the functionality to automatically use the representation with the higher entropy, however, analysts can manually change the scale if it is desired. After interacting with the data by e.g. removing outliers from the data set this operation can be performed again to update the optimal mapping for each dimension. Changing the scaling alters the outcome of subspace generating algorithms, the sorting of small multiples, and also helps an analyst get a clearer picture of the content of a dimension.

Brushing: Linking among the different panels and visualizations is done via brushes using differently colored data items (similar to the high-dimensional brushing [20]). Users have the flexibility to brush in the target frequency plot, the parallel coordinates, and scatter plots. Complex brushes are enabled by providing different operations, like replacing, joining, intersecting, or subtracting of brush operations. We provide the functionality to brush five quartiles of a dimension as automatic way to segment the data. For categorical attributes, we impose an order for the purpose of computing quartiles by sorting the categories by their frequency. This leads to the most common categories being in the highest "quartile", the second most common in the second "quartile", etc. (Figure 4). We also provide item space clustering (using *k-means*) w.r.t. to the current multivariate view which also provides an automatic brushing. This can be useful to look how item space clusters show up in other dimensions that are not part of the multivariate view.

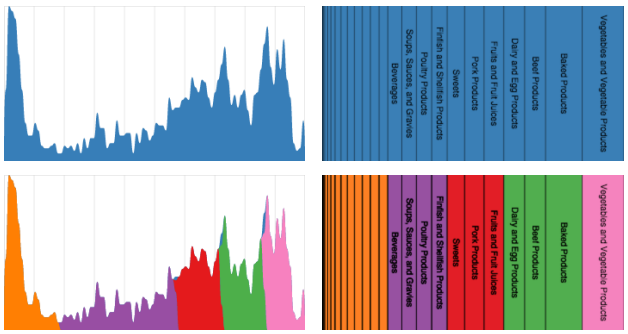

Fig. 4. **Small multiple frequency plots.** On the left is the numerical dimension "Water" showing the distribution of percentages of water in various food products. On the right is the categorical dimension "FdGrp_Desc" describing the food group of food products. The width of the rectangles representing distinct values is dependent on the number of data items having the respective value. The plots on the bottom show the quantiles of their respective data item distribution. For categorical values the most common categories are grouped together.

On a technical note, drawing parallel coordinates and scatter plots with multiple colors due to the brushes introduces a bias towards the last drawn color. In order to avoid this problem we randomize the data points before drawing scatter plots and parallel coordinates leading to a color distribution that reflects the actual distribution of the different brushes.

Brushes have multiple uses in our tool. They can be used to specify targets that are then used to suggest multivariate views. For class definitions different brush colors represent different classes in the item space. Brushing can also be used to define which items to retain when removing outliers or focus one's attention only on the defined classes. Lastly, brushes can also be used to inspect how a set of data items behaves in different dimensions.

Given brushes have multiple unrelated applications, users need to frequently reset and reassign the brushes for different tasks. For example, the user wants to remove some outliers so she brushes the data points that shall remain and removes the rest. After that, in order to define a target for further analysis the brushing has to be reset and newly assigned.

4.3 Suggestions & Refinements

Suggestions allow analysts to automatically create views which then can be refined either via manual curation or further automated suggestions. They guide analysts in performing four main data analysis tasks. Subspace clustering for finding dimensions with dense items, correlation to find redundant dimensions, anomaly detection to find outlier dimensions, and class separation for finding dimensions suitable for distinguishing labeled data.

Subspace Clusters: The goal of subspace clustering is to find dimensions and items that are compact. Under the assumption that this typically occurs as hypercubes (i.e. ranges of items in dimensions) algorithms like CLIQUE (implemented in ELKI [1]) provide good results (see Figure 6). The resulting subspaces are shown in the small multiples plot as groups. This inevitably leads to dimensions showing up at multiple locations. Due to the use of the density plots same dimensions can easily be detected due to similar shapes of the distribution. Furthermore, next to the name of a dimension the number of times this dimension is shown in the small multiple plot is indicated. Also, selecting a dimension with multiple occurrences shows all occurrences of this dimension (e.g. "robberies" in Figure 6). The resulting subspace clusters also include associated brushes which are the actual dense items in the cluster. For this reason we create a view for every subspace cluster in the view management list to include the associated brushes.

Correlated Dimensions: Finding highly correlated dimensions can be of interest when exploring relationships between dimensions. We use hierarchical clustering (DB-SCAN with single linkage) on the correlation between dimensions in order to find groups of similar dimensions. The correlation is computed using the Pearson coefficient converted into a metric by transforming the coefficient in a way such that highly (positive and negative) correlated features result in values close

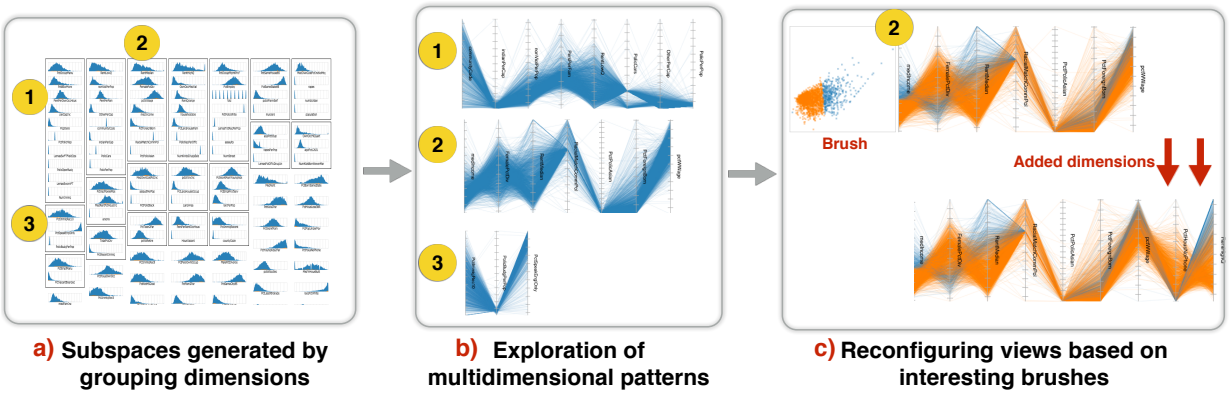

Fig. 5. **Workflow: Building and refining system-suggested subspaces.** Subspace or correlation clustering groups dimensions together in the small multiples panel (a). The analyst can then explore those suggested subspaces (b) and decide on which one to focus. In this example, the analyst chose subspace (2) as view for further exploration. Via target definition and detection of further interesting dimensions this suggested view is refined by the analyst (c).

to 0 while features that are not correlated result in values close to 1. The user can set the threshold for the clusters using a slider which are then used to group dimensions in the small multiples plot (see Figure 7). The resulting groups of dimensions are shown in the small multiples plot. DB-SCAN produces a partition of the available dimensions so no special care for dealing with dimensions showing up at multiple positions has to be done.

Anomaly detection: A way to detect outlier dimensions, i.e. dimensions with interesting properties according to a specified target, is by filtering for distributions of the specified items that are anomalous compared to the overall distribution of items in dimensions. To answer this we compute metrics comparing the distributions of colored items in a dimension to the overall distribution of items in the dimension. This provides an insight on how anomalous a given set of data items is w.r.t. a dimension and compares how the *shapes* of both data point distributions correlate. We compute those metrics by comparing the histogram of the selected data items to the histogram of all items in a given dimension. For the comparison we provide two alternatives: the Pearson coefficient applied to the distributions and the KL-divergence applied to the distributions. Both techniques give a slightly different result but both of them promote dimensions where the coloring has a different distribution than the rest of the items. Thus, the distribution of the colored items can be considered anomal for a dimension.

Class Separating Dimensions: Finding what dimensions separate and thus predict classes defined by analysts is an important task. Suggestions to answer this question are based on both selections of a dimension as well as a target. We compute how well a given dimension is able to predict certain characteristics of a target with selected dimension. Those characteristics are formulated by defining targets that reflect for example different categories of a selected dimension. A metric that describes this "predictiveness" very well is *Information Gain*. *Information Gain* is the change of entropy if a dimension is included as source of information. The larger the change the more information is gained by including this dimension. We compute the *Information Gain* for all dimensions with respect to the defined target and the selected dimension. This dimension filter is useful for finding subspaces that separate classes well.

The unique feature of *SeekAView* is that our system allows analysts to manipulate the results of view construction and influence their computation. This leads to a high-degree of transparency in the interpretation of the metrics. This can be done by changing the set of dimensions as well as the set of data items.

5 Usage Scenarios

In order to show the usefulness of *SeekAView* we demonstrate two usage scenarios. One scenario is exploring the unnormalized Communities and Crime data set. The other is based on the US Department of Agriculture (USDA) National Nutrient Database.

5.1 Communities and Crime

The Communities and Crime data set constructed from the 1990 US Census and the 1995 US FBI Crime report [30] contains 147 different dimensions and contains over 2000 items describing various crime and non-crime related attributes of communities in the United States of America. The data is provided by the University of California, Irvine (UCI) machine learning archive. Our hypothetical analyst wants to explore the data in two different ways, finding interesting subspaces and finding dimensions related to "Violent Crimes per Pop.".

As first step, the analyst clicks on "auto scale" which selects a good conditioning for each dimension automatically. This makes the distribution of items within the dimensions much clearer for her. Then, she initiates subspace clustering which results in clusters mostly related to "robberies" (see Figure 6 and 6a). Inspecting the largest cluster she sees that "NumStreet" (number of homeless people) is highly correlated with the number of "robberies". However, the smaller the number of homeless people gets the larger is the variance of "robberies". In order to improve on her results she uses the quantiles selection on "robberies", brushes the middle three quartiles (see Figure 6b) and clicks on "show selection" in order to remove the outer quartiles. After removing those outliers she then re-runs the subspace clustering algorithm. In the largest cluster she sees a thin line that occupies most of the items. She uses brushing to verify that this line is consistently compact throughout the cluster (see Figure 6c). By reintroducing the earlier removed items she sees that this pattern is still present in the original data.

Next, our analyst wants to explore which factors play into the number of "Violent Crimes per Pop." so she selects the dimension and brushes for high values. She then uses the KL-Divergence selection shape similarity metric to find dimensions where this brushing is also of interest. After deciding on a threshold for the filter using the slider she constructs a multivariate view of those dimensions. Since "community name" is one of the available dimensions in the data set but has distinct values for every item, the KL-Divergence filter will always include this dimension since every selection of items has an entirely different shape for this feature. The analyst manually removes this dimension. The remaining dimensions in the view, upon inspection, are dimensions related to violent crimes, like "Assaults", "Burglaries per Pop.", "Robberies per Pop.", etc., and, surprisingly, dimensions related to the family situation of children, like "Pct. Family with 2 Parents", "Pct. Kids with 2 Parents", "Pct. Kids Born Never Married" (i.e. children born to parents that never got married), etc.. K-means clustering further reveals three clusters that connect high violent crimes with bad family situations and vice versa. The analyst notices an odd shape in the PCA projection and decides to brush this occurrence as fourth cluster (green brush in Figure 2). This brush corresponds to a peak in "Violent Crimes per Pop." with a relatively low value. However, this peak doesn't show up in any other dimension of this subspace. The large amount of communities in this range suggest that there is no

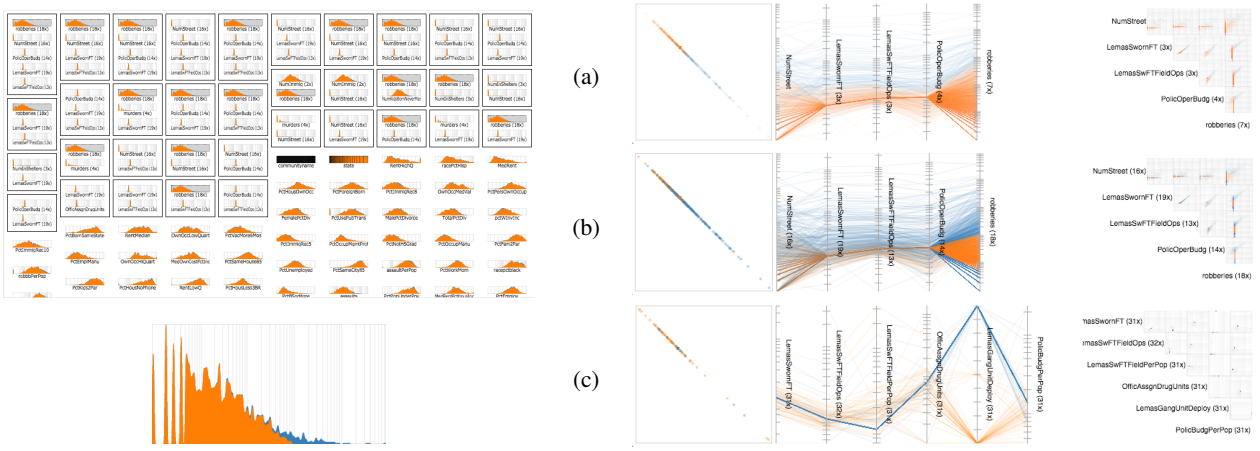

Fig. 6. **Refining subspace clusters.** Example of system generated subspace clusters as described in the Communities and Crimes case study in Section 5.1. On the left, the results of the clustering in the Small Multiple Frequency Plot Widget can be seen. The selected dimension, "robberies", shows up in multiple clusters. In (a) the largest subspace cluster can be seen. Following the case study, in (b) only data items from the inner quartiles of "robberies" are brushed which is used to remove the non-brushed outliers. After that the subspace clustering is rerun resulting in (c) as the largest subspace cluster.

correlation between violent crimes and family situation for most communities. For those, the norm also is a moderate amount of violent crimes. However, for communities with an exceptional amount of violent crimes there is a strong correlation between the number of crimes and the family situation. The analyst can write down her findings in the textual description associated with the view and save the view.

5.2 USDA National Nutrient Data

The US Department of Agriculture (USDA) National Nutrient Database [31] contains information of about 50 different nutrients of over 8000 food products as well as their name and food group categorization. For the scenario we removed all food products that had incomplete records resulting in over 1800 remaining food products.

Our hypothetical analyst is interested in correlations of nutrients. After clicking on "auto scale" to choose optimal mapping functions for the dimensions she moves the "Stack" slider to group dimensions that are highly correlated. She inspects the biggest cluster which contains seven dimensions. One of the dimensions is "NDB Number" which identifies food items. She removes this dimension. Then, she clusters the remaining subspace with k-means clustering (see Figure 7). Noticing, that the computed clusters roughly correspond to different food groups, our analyst wants to find out if she can find dimensions that separate food groups very well. She first saves this view with a textual description of her finding and next goal.

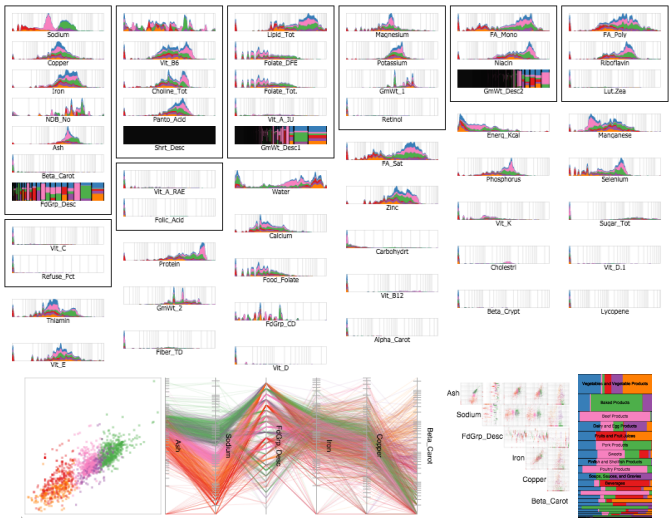

Fig. 7. **Grouping of similar dimensions.** A subspace of similar dimensions from the USDA nutrition data case study (see Section 5.2). The largest group is shown in the multidimensional view panel. The current target is computed using k-means item space clustering. The selected dimension "Food Group Desc." shows that those item space clusters roughly correspond to some food categories. This leads to a further analysis trying to better separate those categories (see Figure 8).

In order to do find dimensions that better separate the clusters, our analyst uses different brushes to define some food groups, such as "meats" ("beef", "pork", and "poultry"), "plant products" ("vegetables" and "fruits"), and "fish and shellfish". She then uses the "Information Gain" filter to find a subspace that predicts those categories well. Initially, her subspace has too many dimensions to be easily interpretable (see Figure 8a). Therefore, she uses the correlation similarity grouping, as well as the scatter plot matrix, to identify highly correlated i.e. redundant dimensions and filter them out. Furthermore, as she sees in the PCA scatter plot there are many items that are not in categories of her interest scattered around a central cluster. She uses brushing on the PCA view to filter-out those outliers. Without those outliers she finds and removes more dimensions that are not beneficial for the separation of her food classes (see Figure 8b). Also, she adds another category "sweets" to her target. Finally, she wants to see how well her classes separate without interference of unclassified items. In order to do this she clicks "show selection" to remove non-selected items (see Figure 8c). Looking at her result she sees that she can identify "meats" for example by having a relatively high "Vitamin B12" and "Iron", and low "Retinol" content. As nutritionist, she knows that "Potassium" would also be a good indicator but she removed this dimension earlier since it only provided redundant information which was also provided by the "Iron" content. "plant products" can be separated by having low "Vitamin B12", more "Carbohydrats" than "meats", and less "Lipids" than "meats". "fish and shellfish" have high "Retinol" and "Vitamin D1" contents and "sweets" are mostly identifiable by their high "Carbohydrats" contents.

6 Discussion

Through the two usage scenarios, we demonstrated how *SeekAView* helps analysts in configuring different views and building subspaces with the help of suggestions from the system. In this section we discuss our three claimed contributions (i, ii, and iii), and the benefits and limitations of the system with regard to the state-of-the-art.

Regarding claim i), *SeekAView* can effectively support the analyst in a human-guided fashion with machine-generated suggestions. At any time, the analyst can trigger a recomputation of the suggestions from

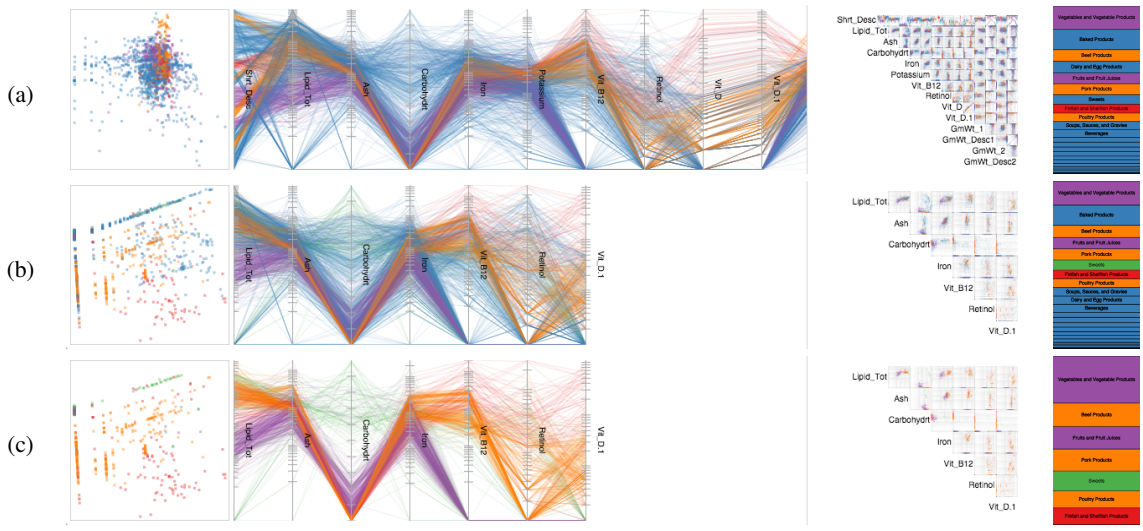

Fig. 8. **Class separating subspace.** The process of finding a subspace separating user defined food categories in the USDA nutrition data case study (see Section 5.2). Using the Information Gain based filter to create the initial view (a). The target is specified using manual brushing in the "Food Group Desc." dimension (top right). Removing outliers seen in the PCA scatter plot and removing redundant or noisy dimensions results in the multivariate view seen in (b). Showing only the manually defined categories (c) by removing non-brushed data items from the view highlights the achieved class separation.

a selected subspace, or can manually fix an automatically computed subspace using her domain expertise. Being able to enter the visual analytics loop by an automated analysis or by a human focus is essential to support important analytical tasks. Other systems such as [26] start from a subspace clustering and facilitate its exploration but never interact with the algorithm or its results.

Compared to the existing techniques for quality metric aided visual exploration, our approach is more scalable with respect to very high dimensionality (> 100), and flexible in allowing one-to-many comparison across all dimensions. The output of the suggestions, which encapsulate different metrics, are always groups of dimensions, thus focusing exclusively on visualizing multivariate patterns instead of just 1D or 2D patterns. Real-time guidance is provided on construction of different selections, the goal being similar to recommendation-based visual analytics [3].

Regarding claim ii) regarding transparency, *SeekAView* relies on visualizations that show the data items with understandable transformations, except for the PCA scatter plot that is used as a guide in addition to the other multivariate views. All the item filtering, binning, and dimension selections are always visible and represented as in the data table to enable analysts to relate the visualizations to the raw data, even if conditioning is applied. We consider conditioning operations as standard enough to be understandable by any analysts, unlike MDS for example.

Regarding claim iii) on the flexibility of *SeekAView* with regards to the multiple starting points of analyses, we have shown in the usage scenarios that handling more than 100 dimensions was manageable using multiple methods, either through guidance by starting with 1D distributions, through subspace clustering, or through elimination of redundant dimensions by grouping and filtering correlated dimensions. Compared to clustering dimensions [33, 37], we provide a more flexible visual interface with which analysts can proactively choose dimensions of interest by looking at 100 dimensions at once, and interactively build multivariate plots from target features of different attribute types. Our visual design adapts to these different attribute types and addresses the challenge of visual representation of multiple attribute types at once.

Still, *SeekAView* can be improved in two directions: scalability on the number of items, and providing more automated algorithms to suggest interesting subspaces.

In terms of scalability, *SeekAView* is implemented in JavaScript and runs in a browser; it is therefore limited in the computational power of the browser that will undoubtedly improve but remains limited compared to a standard desktop application. Providing *SeekAView* as a web-based application facilitates its deployment while limiting its scalability; this is our selected trade-off but we are confident that the same system programmed as an application could handle ten to hundred thousands of items.

7 Conclusion and Future Work

We proposed *SeekAView*, a visual analytics system for letting analysts build and refine subspaces interactively with the help of suggestions from the system. We proposed and implemented an iterative workflow catered towards reconfiguration of interesting views for navigating multiple multidimensional data spaces with respect to both the dimensions and the items. Compared to the state-of-the-art in subspace visualization, we are able to achieve a much higher degree of flexibility in adapting to the different starting points of analyses. By providing suggestions on not only the quality of the current selections but also on next steps of analysis, there is a seamless interaction between analysts' mental models and the functionalities of our system.

In the future, we will pursue research directions on further augmenting the guidance mechanism with more metrics and algorithms such as SVMs for providing more informed choices of subspaces to the analyst. Also, we plan to add functionality to restrict the number of subspaces relative to the display size. Furthermore, there is currently no in-system way of dealing with missing labels and necessary steps have to be performed in preprocessing steps which might be slow. We are planning to add a way of dealing with missing values in an interactive fashion. While we only have demonstrated the application of our approach using some usage scenarios, we are also in the process of deploying our tool in the fields of health-care and finance. We believe our tool will not only help domain experts in flexible multivariate analyses but also getting feedback from them will help us improve our framework and integrate more intelligent guidance strategies.

8 Acknowledgment

The research described in this paper is part of the Analysis in Motion Initiative at Pacific Northwest National Laboratory (PNNL). This work was funded by the Laboratory Directed Research and Development Program (LDRD) at PNNL. Battelle operates PNNL for the U.S. Department of Energy (DOE) under contract DE-AC05-76RLO01830.

References

[1] E. Achtert, H. Kriegel, and A. Zimek. ELKI: A software system for evaluation of subspace clustering algorithms. In B. Ludäscher and N. Mamoulis, editors, *Scientific and Statistical Database Management, 20th International Conference, SSDBM 2008, Hong Kong, China, July*

9-11, 2008, Proceedings, volume 5069 of *Lecture Notes in Computer Science*, pages 580–585. Springer, 2008. 4.3

[2] S. Bileschi and L. Wolf. Image representations beyond histograms of gradients: The role of gestalt descriptors. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2007. 4.1

[3] L. M. Blaha, D. L. Arendt, and F. Mohd-Zaid. More bang for your research buck: toward recommender systems for visual analytics. In *Proceedings of the Fifth Workshop on Beyond Time and Errors: Novel Evaluation Methods for Visualization*, pages 126–133. ACM, 2014. 6

[4] M. Brehmer, M. Sedlmair, S. Ingram, and T. Munzner. Visualizing dimensionally-reduced data: Interviews with analysts and a characterization of task sequences. In *Proceedings of the Fifth Workshop on Beyond Time and Errors: Novel Evaluation Methods for Visualization*, BELIV '14, pages 1–8, 2014. 2.3

[5] A. Buja, D. Cook, and D. F. Swayne. Interactive high-dimensional data visualization. *Journal of Computational and Graphical Statistics*, 5(1):78–99, 1996. 2

[6] J. Chuang, D. Ramage, C. Manning, and J. Heer. Interpretation and trust: Designing model-driven visualizations for text analysis. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 443–452. ACM, 2012. 2

[7] A. Dasgupta and R. Kosara. Pargnostics: Screen-space metrics for parallel coordinates. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):1017–1026, 2010. 2.2

[8] N. Elmqvist, P. Dragicevic, and J.-D. Fekete. Rolling the dice: Multidimensional visual exploration using scatterplot matrix navigation. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1539–1148, 2008. 2.2

[9] D. Guo. Coordinating computational and visual approaches for interactive feature selection and multivariate clustering. *Information Visualization*, 2(4):232–246, 2003. 2.1

[10] P. E. Hoffman and G. G. Grinstein. A survey of visualizations for high-dimensional data mining. *Information visualization in data mining and knowledge discovery*, pages 47–82, 2002. 2

[11] S. Ingram, T. Munzner, V. Irvine, M. Tory, S. Bergner, and T. Moller. Dimstiller: Workflows for dimensional analysis and reduction. In *Visual Analytics Science and Technology (VAST), 2010 IEEE Symposium on*, pages 3–10. IEEE, 2010. 2.3

[12] A. Inselberg and B. Dimsdale. Parallel coordinates. In *Human-Machine Interactive Systems*, pages 199–233. Springer, 1991. 2.2

[13] S. Johansson and J. Johansson. Interactive dimensionality reduction through user-defined combinations of quality metrics. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):993–1000, 2009. 2.3

[14] P. Kampstra. Beanplot: A boxplot alternative for visual comparison of distributions. *Journal of Statistical Software, Code Snippet*, 28, 2008. 4.1

[15] D. Keim, G. Andrienko, J.-D. Fekete, C. Görg, J. Kohlhammer, and G. Melançon. *Visual analytics: Definition, process, and challenges*. Springer, 2008. 3

[16] D. A. Keim. Designing pixel-oriented visualization techniques: Theory and applications. *IEEE Transactions on Visualization and Computer Graphics,*, 6(1):59–78, 2000. 2.2

[17] H.-P. Kriegel, P. Kröger, and A. Zimek. Clustering high-dimensional data: A survey on subspace clustering, pattern-based clustering, and correlation clustering. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 3(1):1, 2009. (document), 2.1

[18] J. B. Kruskal and M. Wish. *Multidimensional scaling*, vol. 11. Sage, 1978. 2, 2.3

[19] D. J. Lehmann, G. Albuquerque, M. Eisemann, M. Magnor, and H. Theisel. Selecting coherent and relevant plots in large scatterplot matrices. *Computer Graphics Forum*, 31(6):1895–1908, 2012. 2.2

[20] A. R. Martin and M. O. Ward. High dimensional brushing for interactive exploration of multivariate data. In *Proceedings of the 6th Conference on Visualization'95*, page 271. IEEE Computer Society, 1995. 4.2

[21] T. May, A. Bannach, J. Davey, T. Ruppert, and J. Kohlhammer. Guiding feature subset selection with an interactive visualization. In *Visual Analytics Science and Technology (VAST), 2011 IEEE Conference on*, pages 111–120. IEEE, 2011. 2.1

[22] Y. Pang and L. Shao. Special issue on dimensionality reduction for visual big data. *Neurocomputing*, 173:125–126, 2016. (document)

[23] J. Schneidewind, M. Sips, and D. A. Keim. Pixnostics: Towards measuring the value of visualization. In *IEEE Symposium On Visual Analytics Science And Technology*, pages 199–206. IEEE, 2006. 2.2

[24] J. Seo and B. Shneiderman. A rank-by-feature framework for interactive exploration of multidimensional data. *Information Visualization*, 4(2):96–113, 2005. 2.2

[25] A. Tatu, G. Albuquerque, M. Eisemann, J. Schneidewind, H. Theisel, M. Magnor, and D. Keim. Combining automated analysis and visualization techniques for effective exploration of high-dimensional data. In *IEEE Symposium on Visual Analytics Science and Technology*, pages 59–66. IEEE, 2009. 2.1, 2.2

[26] 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. 6

[27] C. Tominski, J. Abello, and H. Schumann. Axes-based visualizations with radial layouts. In *Proceedings of the 2004 ACM symposium on Applied computing*, pages 1242–1247. ACM, 2004. 2.2

[28] E. R. Tufte. *The Visual Display of Quantitative Information*. Graphics Press, Cheshire, CT, USA, 1986. 4.1

[29] C. Turkay, P. Filzmoser, and H. Hauser. Brushing dimensions-a dual visual analysis model for high-dimensional data. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2591–99, 2011. 2.3, 3, 3

[30] UCI. Communities and Crime Unnormalized Data Set. `http://archive.ics.uci.edu/ml/datasets/Communities+and+Crime+Unnormalized`. 5.1

[31] USDA. USDA National Nutrient Database for Standard Reference. `http://ndb.nal.usda.gov/`. 5.2

[32] L. van der Maaten and G. E. Hinton. Visualizing high-dimensional data using t-sne. *Journal of Machine Learning Research*, 9:2579–2605, 2008. 2, 2.3

[33] J. Wang, W. Peng, M. O. Ward, and E. A. Rundensteiner. Interactive hierarchical dimension ordering, spacing and filtering for exploration of high dimensional datasets. In *IEEE Symposium on Information Visualization*, pages 105–112. IEEE, 2003. 2.1, 6

[34] E. J. Wegman. Hyperdimensional data analysis using parallel coordinates. *Journal of the American Statistical Association*, 85(411):664–675, 1990. 2.2

[35] L. Wilkinson, A. Anand, and R. L. Grossman. Graph-theoretic scagnostics. In *INFOVIS*, volume 5, page 21, 2005. 2.2

[36] P. C. Wong and R. D. Bergeron. 30 years of multidimensional multivariate visualization. In *Scientific Visualization*, pages 3–33, 1994. 2

[37] J. Yang, A. Patro, H. Shiping, N. Mehta, M. O. Ward, and E. A. Rundensteiner. Value and relation display for interactive exploration of high dimensional datasets. In *IEEE Symposium on Information Visualization*, pages 73–80. IEEE, 2004. 2.1, 6