# Interactive Subspace Cluster Analysis Guided by Semantic Attribute Associations

Salman Mahmood and Klaus Mueller, *Senior Member, IEEE*

**Abstract**—Multivariate datasets with many variables are increasingly common in many application areas. Most methods approach multivariate data from a singular perspective. Subspace analysis techniques, on the other hand. provide the user a set of subspaces which can be used to view the data from multiple perspectives. However, many subspace analysis methods produce a huge amount of subspaces, a number of which are usually redundant. The enormity of the number of subspaces can be overwhelming to analysts, making it difficult for them to find informative patterns in the data. In this paper, we propose a new paradigm that constructs *semantically consistent* subspaces. These subspaces can then be expanded into more general subspaces by ways of conventional techniques. Our framework uses the labels/meta-data of a dataset to learn the semantic meanings and associations of the attributes. We employ a neural network to learn a semantic word embedding of the attributes and then divide this attribute space into semantically consistent subspaces. The user is provided with a visual analytics interface that guides the analysis process. We show via various examples that these *semantic subspaces* can help organize the data and guide the user in finding interesting patterns in the dataset.

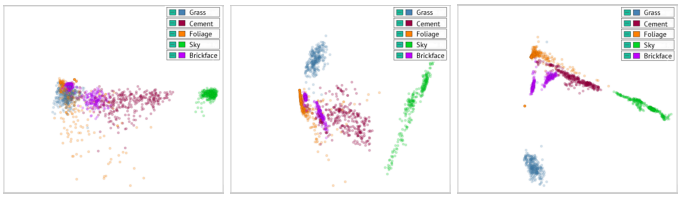
**Index Terms**—High-dimensional data, multivariate data, subspace clustering, subspace analysis, cluster analysis.

## 1 Introduction

A consequence of the age of big data is the rapid increase in complexity of multivariate datasets and the corresponding need for appropriate data analysis and interpretation tools. An important aspect of understanding multivariate data is to identify and interpret relevant patterns – data items that are associated with one another in terms of some similarity metric. For a multivariate dataset, this task can be challenging due to the curse of dimensionality. It refers to the fact that as the dimensionality of a dataset grows, the volume of the space increases so fast that the data inside that space become very sparse. As a result, all points appear to be equally far and dissimilar. This makes it difficult to locate and identify patterns in these datasets.

To ease this task a plethora of dimension reduction techniques have been developed. Commonly, these methods approach the data in their entirety – without subdividing the space. However, viewing multivariate data from a singular perspective is often not sufficient since useful information can be hidden in some subset of the attributes. Assume, for instance, a dataset on housing which may contain a group of attributes that pertain to the structure of the house (number of bedrooms, floors etc); another group of attributes might relate to the neighborhood of the house (crime, schools, etc), and so on. Interesting patterns may be embedded in each of these attribute subsets of the data, yet these patterns might not be discernible in a subsequent 2D projection with a standard dimension reduction technique, such as PCA [1], MDS [2] or t-SNE [3], where all dimensions are maintained.

Subspace analysis is a means to overcome this problem. A subspace is a subset of the data dimensions into which the data can be projected. It is the aim of subspace analysis to identify the specific subset of dimensions that can capture a given pattern, and which, when projected into 2D, will not diffuse the pattern and allow users to view its structures clearly without occlusion. An example is given in Figure 1 which shows several PCA-projections of the 16-D Image Segmentation dataset [1]. We observe that the clusters arising from the dataset's five image classes are fairly intermixed in the projection generated from the full set of features (a), but they are well differentiated in the subspace-based projections (b, c) (for more detail see the supplementary material). The capability of subspaces to disambiguate projections for display also applies to datasets without predefined classes. Our paper mainly addresses this more exploratory scenario.




(a) Full space (b) Subspace (c) Extended subspace

Fig. 1: PCA projections of the Image Segmentation dataset generated from (a) the full 16D dataspace comprised of all feature dimensions, (b) the 3D *Raw Color* semantic subspace and (c) the 5D extended *Raw Color* semantic subspace. The points are colored by their image class (legend: top right).

In general, subspace analysis identifies multiple perspectives, one per pattern, from which users can view the data, and so it can provide a narrative and guidance by which a complex data space can be effectively explored. However, subspace analysis is not trivial. A dataset with $d$ attributes contains $O(2^d)$ subspaces. This means that for large values of $d$ the search space is prohibitively vast,

• *Salman Mahmood and Klaus Mueller are with the Computer Science Department, Stony Brook University, Stony Brook, NY 11794 E-mail: {samahmood, mueller}@cs.stonybrook.edu*

*Manuscript received August 4, 2022; revised ...*

1. http://archive.ics.uci.edu/ml/datasets/image+segmentation

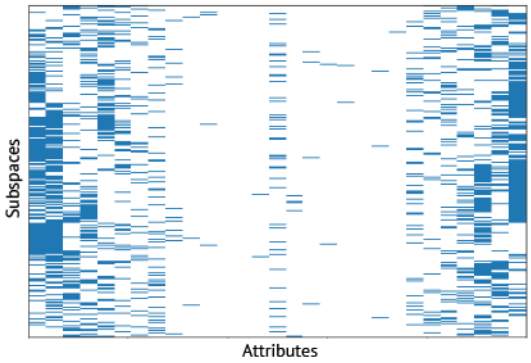



Fig. 2: Heatmap of a subset of the 2,011 subspaces created for the Filipino Family Income and Expenditure dataset using the SURFING algorithm. The columns represent the attributes and the rows represent the subspaces. The subspaces are ordered using the Jaccard similarity. Note that in this figure we have only included the subspaces with three or more attributes – about 10% of the generated subspaces.

making it impossible to explore the entire search space exhaustively. While various heuristics as well as interactive human-in-the-loop systems have been devised to cope with the exponential search space, the prime problem with subspace analysis in general remains the excessive amount of subspaces produced by the underlying algorithms.

We introduce a new paradigm to generate subspaces, yielding what we call *semantic subspaces*. Rather than using similarity metrics derived from the data values, it uses similarity metrics derived from the semantics of the data attributes to form an initial set of subspaces which then can be further explored via automated subspace expansion and user-driven cluster analysis. Our method assumes that each attribute comes with a meaningful label that is part of a natural language dictionary. It then learns the *semantic distance* between attributes from these labels and other metadata that are optionally provided.

Attributes with a small semantic distance are considered to be part of a *concept* [4], defined as an "abstract idea or general notion that occurs in the mind, in speech, or in thought" [2]. An example of a concept is the notion of "house"; it associates with various superordinates, like "neighborhood" and subordinates, like "bedroom". This paradigm, in conjunction with a set of analytical tools we devised, opens a new workflow along which the massive set of subspaces can be explored. A user starts from the detected, and possibly further refined multivariate concepts and makes discoveries there, then projects these discoveries into other concepts, and then gradually expands these concepts aided by conventional subspace search.

Our paper is organized as follows. Section 2 presents an illustrative example. Section 3 discusses related work. Section 4 presents a system overview and theory. Section 5 describes our visual analytics tool. Section 6 narrates two usage scenarios. Section 7 offers a discussion. Section 8 ends with conclusions.

2. https://en.wikipedia.org/wiki/Concept

## 2 Illustrative Example of the Problem

To gauge the enormity of the problem, Fig. 2 visualizes a small subset (about 10%) of the subspaces generated by the SURFING algorithm [5] for the 60-dimensional Filipino Family Income and Expenditure dataset [3]; each row is a subspace and each small bar denotes an attribute's subspace membership. The algorithm generates a total of 2,011 subspaces. We observe that some of the attributes are part of many more subspaces than others. This can create bias in the analysis process, as well as redundancy, since some of the attributes are not featured as often as others.

The problems arise from the fact that even though a number of the subspaces produced are either very similar or redundant, it remains difficult to determine the minimal set of representative subspaces. All this makes it very challenging for analysts to find interesting patterns in the subspaces.

## 3 Related Work

Given the ubiquity of multivariate data many papers have been written on the subject in many scientific disciplines. In visualization there are essentially four basic paradigms: visualizing the data as a set or matrix of bivariate scatterplots, called SPLOM [6], as a linear [1] or non-linear 2D embedding [2] [3], as a set of polylines across parallel axes [7], or as abstractions derived from the data, such as Scagnostics [8]. Many variants of these basic paradigms have been described or integrated into more elaborate systems.

Since the projection of multivariate data onto a 2D canvas is ill-posed in all but the most trivial cases, each of these paradigms has strengths and weaknesses. Subspace analysis addresses a shared weakness, namely the problem arising from projecting thematically unrelated, yet possibly overlapping patterns in the data into a common visualization, leading to visual interference. Yet, subspace analysis is also not without challenges, as noted in the introduction.

Subspace analysis is in some respect related to cluster analysis, but the latter is more concerned with detecting patterns in the data and less with determining the dimensions that define them. A simple cluster analysis technique is k-means but it tends to produce spherical clusters which can break up non-spherical structures in high-D space. More sophisticated techniques like DBSCAN [9] use density and connectedness measures that do a better job in these cases, but these methods often fail when there is noise in the data which can lead the structure tracking astray. An inherent problem with these techniques is that the notion of density is difficult to define numerically as the level of sparseness grows with increasing dimensionality; relating the density (and thus sparseness) measures to the dimensionality of the subspace can mitigate this problem to some extent [10]. Conversely, a human-in-the-loop approach such as ClusterSculptor [11] can help by allowing human analysts to guide the cluster analysis but it requires a high amount of effort.

More related to subspace analysis are interactive techniques that allow users to identify sets of dimensions they deem important to emphasize certain aspects of the data. Some allow users to craft novel dimensions or projections

3. https://www.kaggle.com/grosvenpaul/family-income-and-expenditure

that can differentiate data items along user-defined aspects [12] [13] [14]. Others combine data and dimension selection into a single dual-domain interaction, by ways of interactive data and dimension brushing [15] [16], biclustering [17], or dimension ordering in a 2D attribute correlation plot [18]. Our method also allows users to select dimensions via a plot in which dimensions are represented as nodes, but we use the semantic associations of the data attributes for the placement of the nodes as opposed to a statistical analysis of the data values. This adds a semantic flavor to the standard brushing and reprojection workflow which is a novel contribution of our work.

### 3.1 Subspace Clustering and Visualization

Subspace clustering is the activity of identifying subsets of dimensions where groups of closely clustered data points can be found. The most prominent automated subspace clustering schemes are PROCLUS [19], CLIQUE [20], and SURFING [5]. PROCLUS performs iterative refinement of subspaces based on a subset of the points. CLIQUE uses an apriori method where a grid is used to partition the data into equal-sized units and only units with a density beyond a threshold are kept. SURFING appears to be the most popular algorithm in the visual analytics literature. It uses a bottom-up strategy for searching subspaces for clusters by increasing dimensionality. The bottom-up heuristic is based on the idea that new subspaces should be generated using subspaces already known to be interesting. The subspaces are rated according to how interesting a subspace is, and a quality metric is used to prune the search results and determine the direction of the heuristic search.

As mentioned, a downside of automatic subspace clustering is that it can generate an abundant amount of subspaces [21], [22], [23]. A large number of these subspaces are in fact redundant, but only few are highly redundant and among the others it is difficult to determine which to keep and which to discard. Visual analytics can empower humans to make this call, based on preferences and goals.

Tatu et al. [21] allow users to group subspaces identified by subspace clustering via a customized similarity metric based on dimension overlap and KNN neighborhoods. They visualize a subspace by way of a glyph that shows a dimension bitmap and an MDS projected scatterplot. Interestingly, their introduction section shows a cartoon of two semantic subspaces (one on health and one on traveling) of the type our method exploits and facilitates, but their work does not focus on or enable this.

The ClustNails system [22] contributes a radial spike representation where the dimensions of a subspace are equally spaced in angle and the length of a spike is given by the dimension's importance in the subspace; the less spread out the points are along a dimension, the more important it is. We also use a spike representation but we use a PCA-generated biplot [24] which can also visualize the correlations of the dimensions and the spread of the data.

Other work includes VISA [25] which uses an MDS plot to visualize the similarity of subspaces; the diameter of each node depicts the number of objects in the subspace cluster while the color encodes the dimensionality. The display can get very cluttered with an increasing number of subspaces and it also does not show directly which dimensions participate in them; they offer a bar plot for this information.

Interesting is also the work by Wang et al. [23] who show that MDS plots cannot communicate the similarity of subspaces while an ordered similarity matrix computed from the data points can. We do not use MDS plots; rather we use PCA-generated biplots that allow an appreciation of how the data points relate to the dimensions in a subspace.

Lastly, there are also methods that produce animations or transitions between subspaces to show their interrelations. Wang et al. [26] decompose the high dimensional space into a continuum of generalized 3D spaces. A trackball interface is used to transition between adjacent subspaces. Liu et al [27] use animated transitions across subspaces to facilitate the exploration process. Nam et al. [28] propose a tourism metaphor to allow users to travel among subspaces in high dimensional space. Pattern Trails [29] visualizes the pattern transitions across subspaces arranged in a cube. Lines are used for linking patterns in adjacent subspaces, hence introducing pattern transitions.

All of these methods rely solely on numeric data analysis and are unified by the problem that the large number of possible subspaces can make it difficult to navigate to subspaces that offer unique information. We propose an approach that starts off with a small number of familiar subspaces, i.e. thematic groupings of dimensions, which enable analysts to identify interesting and easy-to-grasp relations quickly and then expand from these. While the dimension clustering is automated, yet user-modifiable, the clustering of the data points is mostly manual and predominantly under user control. As such our method falls into the category of interactive cluster analysis methods but adds the element of automated thematic subspace identification.

The methods discussed thus far have defined a subspace as a subset of dimensions. There are also methods that define subspaces as sets of closely associated points. These techniques typically aim to construct a sparse affinity matrix from all data points which can then be used within a spectral clustering framework to break the point set into separate groups, the subspace clusters. A prominent scheme has been Sparse Subspace Clustering (SSC) [30] which solves a convex optimization problem to find, for each data point, the sparsest combination of other data points to express it. These relations can then be used to fill the affinity matrix. Recent efforts that use deep neural networks to first obtain a latent representation of the high-D data followed by traditional clustering (such as k-means) have also embraced SSC and devised Deep Subspace Clustering [31]. The use of deep neural networks in clustering tasks can be advantageous when the high-dimensional data reside on the nonlinear manifold which is often the case in computer vision applications [32]. Our approach has entirely different goals than SSC. It considers a subspace as an extensible sparse set of semantically coherent dimensions into which all data points can be projected to reveal meaningful information.

### 3.2 Word Embeddings

We make use of *word embeddings* [33] to discover the ordinates (terms) associated with a concept from the data attributes and any additional information available on them.

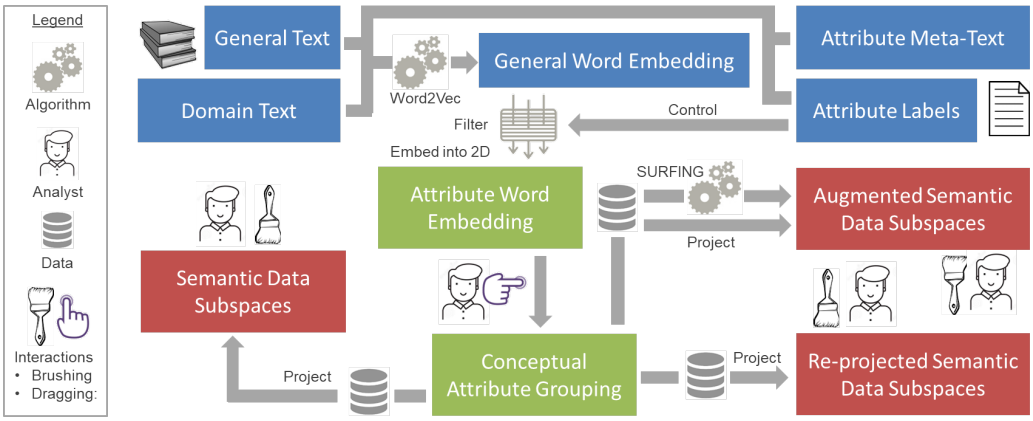



Fig. 3: An overview of our workflow and system. The blue boxes form the NLP-suite of our tool, the green boxes form the **Semantic Space View** comprised of the interactive word embedding visual interface, and the red boxes are the interactive subspace projection visualizations. Algorithms, data and the human analyst feed or manipulate the individual components.

A word embedding is a representation of words in a high-dimensional vector space where words that belong to certain concepts typically locate in a common region. The word embeddings can be learned by training a neural network on a large text corpus which can be as general as the full collection of wikipedia pages or as specific as a collection of domain-centric documents, or both. The latter can represent domain-specific terms and in conjunction with the former their relations to more general words can be uncovered.

The simplest method for learning the context of a word is to use the word's raw co-occurrence counts with context items. However, using a raw co-occurrence matrix is prohibitively expensive in terms of both space and computational effort [34]. The solution is to use neural networks to map concepts to continuous space [35], [36]. Neural networks have been proven to be much more efficient at learning language models. A breakthrough in this regard was the work by Mikolov et al. [33] who proposed skip-gram models for learning word embeddings and demonstrated that these models have the capacity to learn linguistic patterns as linear relationships between vectors [37]. This algorithm has become widely popular as *Word2Vec* and we will show that it can be effectively used in our proposed "semantic" approach to streamline subspace narration and analysis.

## 4 System Overview & Theoretical Aspects

Fig. 3 illustrates a system overview. The blue boxes form the NLP-suite of our tool, the green boxes are the interactive word embedding visualizations, and the red boxes are the interactive brushable subspace visualizations.

The input to our system is a multivariate dataset. Each attribute in the dataset is assumed to have a textual label (the *Attribute Labels* in Fig. 3) and optionally additional textual meta-data (the *Attribute Meta-Text*) that describe the attribute. These meta-data can help to: (1) disambiguate attributes that have more than one meaning, for example the word "jaguar" which can be a car or a wild cat, and (2) assign semantics to attributes that have no lexical meaning at all, like the name of a gene in a bio-informatics dataset.

Another input is a large and general corpus of text, such as the set of English Wikipedia pages (the *General Text*) and an optional set of documents specific to the data domain (the *Domain Text*). Training a word embedding with all of these textual data embodies what we call the *General Word Embedding (GWE)*. In our experiments thus far we only used Word2Vec pre-trained with the English Wikipedia corpus. Training with a domain-specific corpus is fairly straightforward and can give better vector-space representations when the application domain is highly specialized in its terminology, such as radiology [38] or patent law [39].

The dimensionality of the GWE is commonly 128 or more. A standard procedure is to use MDS to produce a 2D embedding of this space for visualization. To not overwhelm the user with visualizing the entire English Wikipedia corpus, we introduce an intermediate filtering step that only allows the attribute labels to pass through to the MDS plot, while preserving the original relational mapping of these words. The result of this process is presented to the human analyst in the *Attribute Word Embedding (AWE)* Display.

It is often the case that the initially produced AWE does not constitute a perfect semantic grouping. This can be due to the aforementioned word ambiguity (aka *conflation of word sense*), or perhaps the analyst has a different grouping in mind. To correct these word placements the user can freely modify the position of the words by simple mouse drag interactions in the AWE Display. This is similar to the interface used in our prior work reported in [40].

The outcome of these interactions is what we call the *Conceptual Attribute Grouping (CAG)*. Each such group of words defines a *semantic subspace* into which the data can be projected and visualized in a biplot. Users can edit these visualizations by brushing to define sub-clusters, de-activate points, and so on. They can then reproject these point sets into different semantic subspaces to observe their behavior.

Next, users can run SURFING to augment the subspaces with additional dimensions. This is a purely data-driven process, but taking into account the semantic subspaces which the user may also further refine. Augmenting (expanding) a subspace allows the discovery of relationships that may exist outside a given semantic subspace, broadening a concept toward its superordinates and acknowledging the fact that concepts overlap and may impact one another.

### 4.1 NLP Aspects

We use the skip-gram model [37] to learn the word embedding, more specifically the gensim implementation [4] of the skip-gram. We trained our network using the English Wikipedia corpus as a general text source. None of the datasets we used in the case studies required specific domain text. The context window size was set to 5 and the size of the word embedding space dimensionality $d$ was kept at 128. We use a minimum word count of 100 – any word with a frequency of less than 100 is removed from the vocabulary.

To gauge the semantic distance among two attribute labels, say, "Crop" and "Rice", we could use the Euclidean or Cosine distance in the high-dimensional word embedding space. Typically these terms are contained in the large corpus the model has been trained on. However, as mentioned in the overview, more accurate distance assessments with more contextual awareness can be achieved when some descriptive short *meta-text* is provided for each attribute that gives the proper context (words that only infrequently occur in the training corpus can also benefit from this) [41]. For example, instead of just using the label "Crop" one might associate more descriptive meta text with it, like "A cultivated plant that is grown as food, especially a grain, fruit, or vegetable" which can be easily obtained from web sources or the domain literature. The various key words mentioned in this short text can help the Word2Vec algorithm in the correct placement of "Crop" in the overall embedding. This can be particularly useful to resolve ambiguities or specific domain meanings. For example, even a word as simple as "Crop" can have many contexts, such as "Crop of Students", "Crop of Products" and the like.

However, meta text cannot be handled with a simple distance metric such as Euclidean, and the same also applies to multi-word attributes. There are many different types of techniques for finding the similarity of two short texts. Lexical matching methods determine whether the words in two short texts look alike, for example, edit distance, lexical overlap or largest common substring. Lexical matching may work for trivial cases but these methods are not very robust. We found that that the Short Text Similarity (STS) [42] and Word Movers Distance (WMD) [43] do a better job in estimating the distance between two texts. STS works better for texts that are only a few sentences, while WMD is used for longer texts. Both methods operate on the high-D word embeddings learned by the skip-gram neural network. The estimation yields a scalar (similarity) value which is then placed into the distance matrix used for the MDS embedding (see the supplement for more detail).

### 4.2 Semantic Subspace Generation

To recap our terminology, the input to our system is a rectangular data matrix where each row is a *data item* and each column is a *data dimension* augmented by an *attribute label*. A *subspace* is a subset of these data dimensions, while a *semantic subspace* is a subspace where the attribute labels are part of a shared concept. Likewise, a *cluster* is a general group of data items, while a *subspace cluster* can be the same group of data items but projected into a specific subspace.

4. gensim 4.3.0: https://pypi.org/project/gensim/

As explained in the introduction, using only a subset of the available dimensions in a projection algorithm can expose patterns that would otherwise not be observable. Finally, apart from the concept-based grouping of attributes, users can also extend the native semantic subspaces by additional attributes to bridge among concepts or expand them.

We note that our semantic approach to subspace analysis is different from conventional subspace clustering where the data items are clustered by progressively adding dimensions until certain cluster properties, such as density, are no longer fulfilled. The semantic subspace clusters we initially construct do not necessarily contain dense clusters, but they typically contain structured data patterns since they derive from a set of thematically-related dimensions that follow some inherent non-trivial data generation process.

This dramatically reduces the initial number of subspaces and allows users to begin the data exploration from concepts they are familiar with and then expand out to examine possible connections and interactions among them.

## 5 Our Visual Analytics Tool

The objective of our visual analytics tool is to help the user partition the multivariate data into subspaces that are semantically consistent and allow the user to identify patterns inside the data. Our visual analytics tool is embodied by an interlinked dashboard shown in Fig. 4. It is composed of five distinct components whose functionality is summarized in the figure caption. In the following we explain the design rationale and the function of each of these components.

### 5.1 Control Panel

The Control Panel (Fig. 4(a)) is used to change the various settings of the visual analytics tool. In the left-most panel the user can (1) select the text similarity algorithm used for the word embedding and (2) turn on/off the biplot axes in the subspace cluster displayed in the Subspace View (see Fig. 5(c) where the biplot axes are superimposed on the subspace cluster's PCA plot as red lines, for more detail see Section 5.4). The window on the bottom gives tool-tip like system feedback.

The other panels are used to generate and modify subspace clusters. The user begins by specifying an initial set of semantic subspaces using the 'Subspaces' slider. In Fig. 4(a) this number is set to 5, prompting the system to cluster the attribute words into 5 groups. The clustering occurs after the filtering but before the MDS mapping into the Attribute Word Embedding (AWE) Display, using the high-D vector representation of the attribute words. We use spectral clustering since it better preserves high-D structures.

The PCA projections of the associated subspace clusters are shown in the Subspace Organizer. Selecting one of these subspaces displays it in the Subspace View and its associated dimensions are listed in the Attribute List, the right-most panel. The list can be manually expanded by clicking the '+' button and selecting an attribute from the Semantic Space View. Likewise, an attribute can be removed from the list via the (-) button next to the attribute name.

The Attribute List can also be extended automatically by selecting the 'Extend' button in the panel labeled 'Subspace'

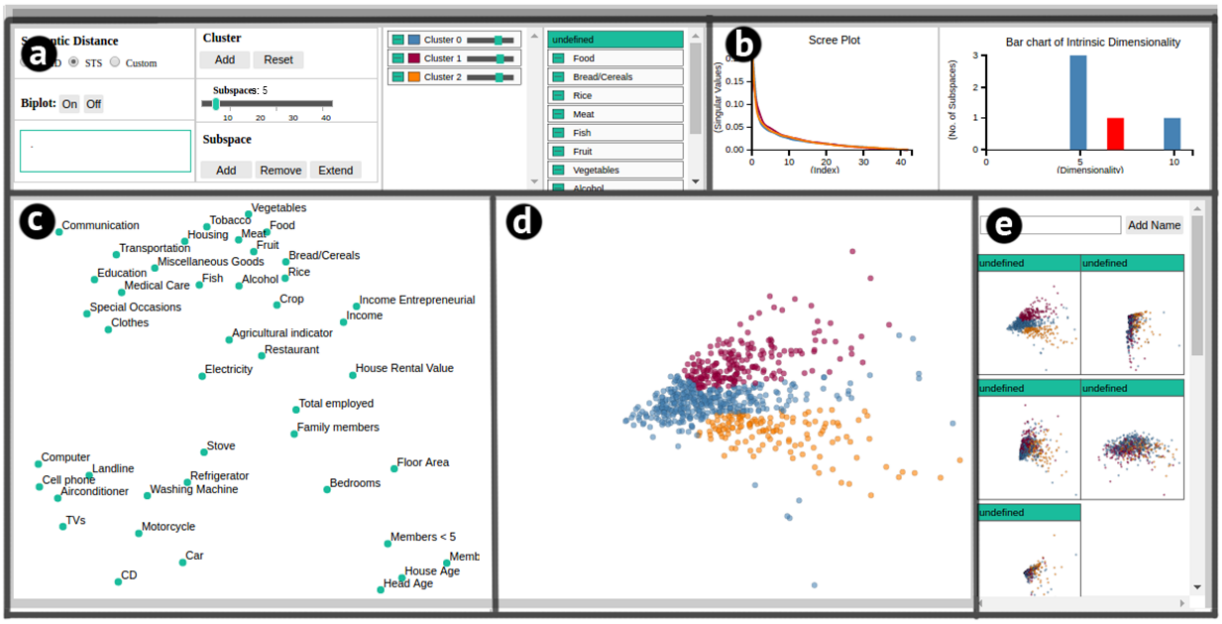



Fig. 4: An overview of the system visualizing the Filipino Family Income and Expenditure dataset. The user interface consists of five coordinated views. (a) **Control Panel**, used to change the various settings of the visual analytics tool. (b) **Dimensionality View**, provides diagnostics about the fidelity of the subspace visualizations. (c) **Semantic Space View**, visualizes the semantic space of the data using a scatter plot of attribute labels. (d) **Subspace View**, shows a user-selected subspace in more detail. (e) **Subspace Organizer**, shows an overview of all of the subspaces generated by the algorithm.

which runs the SURFING subspace clustering algorithm. This is a step-wise process – the attribute with the highest increase in subspace quality is added first, and so on. The 'Add' button allows the user to add a subspace followed by a selection of its attributes as described above. Finally, a subspace can be removed by clicking the 'Remove' button and selecting the target subspace in the Subspace Organizer.

The remaining functions serve the refinement of a subspace cluster into sub-clusters via brushing in the Subspace View. Once a sub-cluster is brushed its points are automatically tagged by a palette-defined color. The user then clicks the Cluster 'Add' button which adds a widget to the subspace's sub-cluster list in the middle panel. Users can delete the sub-cluster by the (-) button or de-emphasize its points by lowering their opacity via the slider in the same widget. We found that it can be helpful to continually interact with this slider to bring different clusters into the foreground.

### 5.2 Semantic Space View

Upon loading the data, the Semantic Space View (Fig. 4(c)) visualizes the Attribute Word Embedding (AWE) generated by the system as the initial Conceptual Attribute Grouping (CAG). Attribute words that are close in this plot tend to have a small semantic distance and strong similarity, and so are likely part of a common concept. As discussed in Section 4, we generate this plot using metric MDS. The MDS optimization $E = \sum_{i<j}^{N} (D_{ij} - d_{ij})^2$ aims to maintain the distances $D_{ij}$ of word pairs $i, j$ in the high-D word embedding and their respective distances $d_{ij}$ in the Semantic Space View by minimizing the stress $E$, allowing users to easily appreciate their neighborhood relations.

Once the Semantic Space View has been generated, a typical next step is to use the Control Panel to form the initial set of semantic subspaces which generates the associated subspace clusters. The user then has the option to alter the semantic subspaces to produce an alternative CAG. The user can (1) adjust the positions of individual attributes via mouse interactions as described in the overview, or (2) modify the initial semantic subspaces by utilizing the corresponding facilities in the Control Panel. All of these operations occur in the 2D Semantic View and do not affect the high-D word embedding.

We found that labeling the relatively crowded MDS scatter plots often results in label overlaps which makes it difficult to read individual labels. To remove the label overlap we adapted an algorithm originally designed for reducing the overlap of nodes in graphs [44]. This algorithm seeks to remove overlap while preserving as much of the initial layout as possible. It does so by creating a proximity graph using the Delaunay triangulation [45] and moving the points along the edges of the proximity graph by small amounts. It iteratively continues the process until overlap has been removed or the maximum number of iterations is reached. This is an automatic process at the moment; more control could be afforded by adding a slider by which users can control the amount of de-cluttering, as described in [46].

### 5.3 Subspace Organizer

The Subspace Organizer (Fig. 4(e)) organizes and provides overviews of the generated semantic subspaces. Each subspace is visualized using a scatter plot. We use PCA to reduce the dimensionality of subspaces with dimensionality $> 2$ and project the points into the top two PCA vectors.

When the user hovers the mouse over a subspace, the attributes that make up the subspace are highlighted in the Semantic Space View. Since the subspaces are semantically

consistent the user can name the individual subspaces by typing an appropriate text string into the "Add Name" text box on top of the panel. This helps in organizing the subspaces and aids recall when the user seeks to examine the subspace again. When a subspace in the Subspace Organizer is selected, details related to the view appear in the Subspace View, the Dimensionality View, and the Control Panel.

### 5.4 Subspace View

The Subspace View (Fig. 4(d)) presents a detailed view of a subspace selected in the Subspace Organizer. A brushing facility allows users to select data points with properties of interest. These can be saved as user-generated (sub-)clusters and can be projected into other semantic subspaces. The Subspace View is the main exploratory view of the interface.

As mentioned, the Control Panel (Fig. 4(a)) has a button by which users can turn on the projection of the dimension (attribute) axes to form a biplot [24]. A biplot (see Fig. 5(f) for an example) is a PCA plot with the attribute axes also projected into the PCA basis. The magnitude of a given vector signifies how strongly the associated attribute contributes to the visualization and the direction of the vector points toward the direction of the contribution. The biplot vectors can be useful in explaining trends and attribute preferences in the data and how the different attributes in the subspace interact with each other. Non-linear space embeddings such as MDS or t-SNE cannot support this as they loose the attribute mapping in the process. We note that this is effective for subspaces with reasonably low dimensionality.

### 5.5 Dimensionality View

The Dimensionality View (Fig. 4(b)) has two plots: the Scree Plot (left) and the Intrinsic Dimensionality View (right). Both refer to the subspace currently displayed in the Subspace View. The scree plot visualizes the sorted eigenvalues of the cluster's PCA analysis. Each eigenvalue is associated with a certain PCA axis (or *principal component*). The higher such a value the more important the principal component is in explaining some of the cluster's total variance. Summing the values from left to right indicates how much of the variance the summed principal components can explain.

A key feature is the point at which the curve flattens, called the *elbow* or *knee*. It can be used to determine how many principal components are needed to faithfully represent the data (see [1], chapter 6), which is a quantity commonly referred to as the *intrinsic dimensionality* of the data. There are various schemes by which the intrinsic dimensionality as manifested by the elbow can be detected; we use a variant of the Kneedle algorithm [47]. The intrinsic dimensionality is important since any principal component beyond it does not capture much of the cluster's variance. The closer it is to the $2^{nd}$ eigenvalue the more faithful the cluster's biplot visualization is since the biplot is a projection of the cluster into the two axes with the largest eigenvalues. Any higher-level variance is not visualized and can lead to projection ambiguities. As such the scree plot is an important diagnostic tool. Note also that the scree plot in Fig. 3(b) displays three closely matching curves. They are due to the three brushed sub-clusters in the Subspace View, with line and point colors matching.

The Intrinsic Dimensionality View is a bar chart that visualizes the histogram of the intrinsic dimensionality of all subspaces, with the currently displayed subspace highlighted in red. The graph is updated when the user makes any changes to the attributes in the subspace. Essentially, each bar in the Intrinsic Dimensionality View summarizes a subspace cluster's scree plot as a single number. It is thus an important subspace diagnostic because when its intrinsic dimensionality is high the user needs to be careful when interpreting the biplot as there can be inaccuracies (ambiguities) in the projected point locations.

## 6 Usage Scenarios

This section demonstrates how semantic subspaces can facilitate the exploration of multivariate data by ways of two usage scenarios. To identify these scenarios we recruited a small cohort of mostly graduate students from our university. None of these individuals had prior knowledge about our system, but all were familiar with fundamental concepts of statistics, such as mean, median, variance, distribution, cluster, correlation, regression, etc. as well as fundamentals of visualization, such as bar charts, pie charts, scatterplots, node-link diagrams, etc. Some were not familiar with principal components, intrinsic dimensionality, and biplots. We explained these concepts to them at the extent needed, referring to observable artifacts such as knee, height, and trend. We then tested whether our explanations were understood.

For this study we collected a few datasets beforehand from which our participants could choose. Here we preferred datasets that embraced multiple concepts, as opposed to just a single concept like the properties of a plant, car, or wine. In our study we did not fully demonstrate the software with an example as a first step; rather we briefly explained the various interface elements and functionalities after the first selected dataset was loaded and then only answered questions. While there were some initial questions, we did not detect persistent usability problems for any of these participants. On the contrary, we were able to gather quite a few interesting discoveries our participants made, and we distilled these into two usage scenarios (another is in the supplementary material) with fictitious data analysts, as presented in the following.

### 6.1 Use Case: Filipino Family Income and Expenditure

For our first scenario we follow Ken, a data analyst at the World Bank, who uses the Filipino Family Income and Expenditure dataset for a report that seeks to study the various aspects of Filipino life and how these relate to one another. The dataset's attributes are family income and expenditure, including, among others, levels of consumption by item of expenditure, sources of income in cash, and related information affecting income and expenditure levels and patterns. There are 60 attributes, but since string-valued attributes cannot be used in the program, Ken removes these (44 attributes remain). The dataset also contains some meta-data containing descriptions of the different attributes to sharpen their semantic focus (see Table 3 in the supplement).

Ken uploads both dataset and meta-data into the visual analytics tool and selects the STS metric to construct the

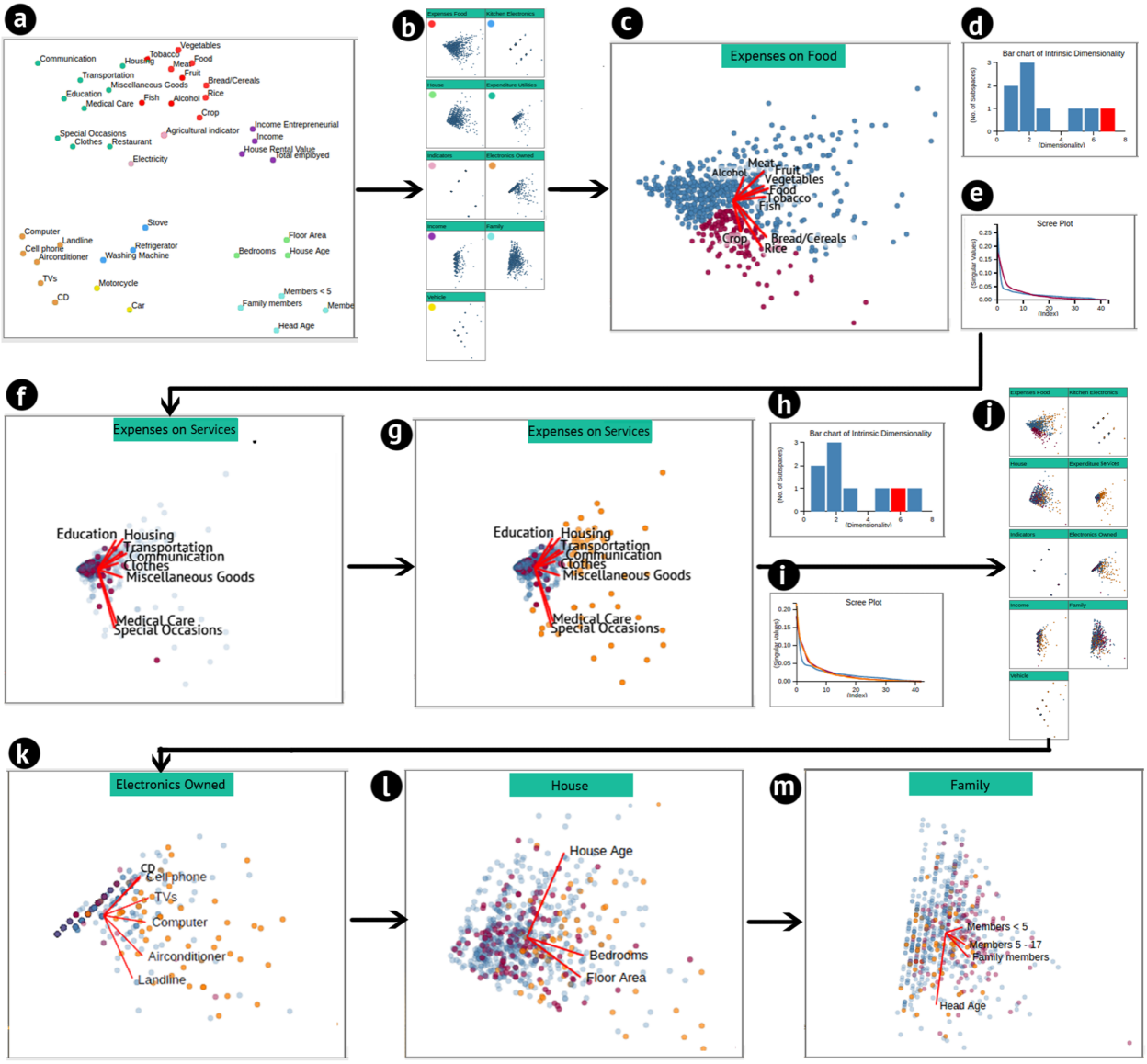



Fig. 5: Workflow for the Filipino Family Income and Expenditure dataset. Significant views of the visual analysis tool are shown as we analyze this aspect of the dataset. See the main text for a narration and the supplement for larger images.

semantic space since the meta-data consist of small descriptions of each attribute. A visualization of the semantic space is shown in Fig. 4(c). He observes that inside the semantic space the attributes related to expenditures such as "Vegetables", "Housing" etc. form one large cluster. Inside the cluster he observes that attributes related to expenses on food related items ("Vegetables", "Rice", "Alcohol") are on the right and attributes related to expenses on services ("Housing", "Transport", "Medical Care") are on the left. Attributes related to ownership such as "Car", "Computer" etc. form another cluster. Inside this cluster, he observes subcategorizations; note that "Car" and "Motorcar" are separated as are kitchen-related electronics ("Stove", "Washing Machine", "Refrigerator"). Other clusters related to income, family, house etc. are also observed in the semantic space.

However, Ken notices some discrepancies in the construction of the semantic space. For example, "Restaurant" is closer to "House Rental Value" even though the semantics are very different. This miscalculation is likely due to the similarity in meaning of house, restaurant, and hotel that are used in the description. Similarly "House Age" is closer to "Head Age" and other age related attributes. Ken uses the mouse in the Semantic Space View and drags the misaligned attributes into the correct position. Fig. 5(a) shows the semantic space's final structure (disregard the coloring).

Looking at the Semantic Space View there appear to be nine conceptual attribute groupings. Ken runs the automated clustering algorithm but observes some deficiencies. For example, "Fish" is in the same cluster as "Housing" and "Medical Care". He uses the Attribute Panel to correct this. The final grouping is shown in Fig. 5(a) encoded in different colors. Finally, Ken names the different subspaces, now visualized in the Subspace Organizer (see Fig. 5(b)). Table 4 in the supplement lists the subspace IDs.

Fig. 5(c) shows the "Expenses on Food" subspace. The biplot reveals how the different attributes contribute to the subspace. Ken finds it interesting that the grain staple attributes "Crop", "Bread/Cereal" and "Rice" have a different

direction than the other food-related expenses. He uses the brush feature to select households with relatively higher values in these features and creates a new (sub)cluster with these attributes (colored red, the other points remain blue).

Maintaining this coloring, Ken moves to the "Expenditure on Services" subspace (Fig. 5(f)) to assess the red (grain staple) households in terms of their services expenditures. The opacity of the blue cluster is reduced to prevent occlusion. He observes that this cluster stays closer to the center, indicating that, in general, these households, while spending well on grain, do not spend much on services such as education, housing, transportation, etc. They appear to lead a simple life.

Next, Ken selects the households that do show higher expenses on services and tags them yellow (Fig. 5(g)). Thus, the red cluster represents the households that spend more on grain staples, while the yellow cluster represents households that have higher expenses on services. Ken is curious to see how these populations behave in other aspects.

Ken knows it is a good habit to assess the scree plot (Fig. 5(i)) before engaging into an analysis. He observes a similar shape for the two clusters, confirming that they have similar dimensionality. The scree plot can be useful when a cluster shows very different shape. In that event it may be worth investigating the reason for the irregular dimensionality of the cluster (e.g. there might not be enough data points or too many dimensions). Finally, Ken browses the Subspace Organizer (Fig. 5(j)) which overviews the colored subspaces.

Looking at the clusters in different subspaces gives Ken some interesting insights. Fig. 5(k, l, m) shows the Electronics Owned, House, and Family subspaces, respectively. The Electronics Owned subspace shows that households in the yellow cluster own more electronics. The House subspace shows that households in the yellow cluster live in larger houses. In the Family subspace, however, the households in the red cluster seem to be evenly distributed, indicating that they occupy a specific social group of the general Filipino society. Conversely, the households in the yellow cluster seem to increase toward the direction of the Head Age attribute vector, which suggests that older families tend to accumulate more electronics than younger families (recall from the Electronics Owned subspace that the yellow cluster owns more electronics). This confirms Ken's prior beliefs.

At this stage, Ken has identified two major social groups and teased out their priorities. The yellow cluster are the households with more economic resources and a higher lifestyle. Its separation from the red cluster in many subspaces suggests that in Filipino society Rice, Bread/Cereal and Crops make up a major portion of the food consumption in the households with fewer economic resources.

The process of repeated brushing, labeling, and reprojection constructs an implicit concept hierarchy. In the example above, Ken started out with the general "All households" concept and used the "Expenses on Food" subspace to identify the "Basic staple households" concept. Then he further branched to the "Services used by basic staple households" concept. The below interactions can extend this hierarchy.

#### 6.1.1 Broadening a Subspace Beyond its Semantic Theme

Ken now moves on to a deeper analysis. He uses the Extend Subspace facility to widen a subspace beyond the confines of a specific context (see Fig. 6). This essentially allows him to discover relationships external to the theme of the given semantic subspace. This newly generated semantic subspace is then automatically added to the Subspace Organizer.

Ken decides to select households with high Expenditure on Services (Fig. 6(a)) and then uses the Extend Subspace facility to add an attribute to the subspace that can improve the quality of the subspace. To find such an attribute the program uses the SURFING algorithm's KNN based quality metric. The Extend Subspace facility automatically adds the "Electricity" attribute to the subspace, and Ken observes the emergence of a new cluster on the lower part of the plot (Fig. 6(b)). This population was previously hidden in the subspace's biplot but it is now exposed as a distinct cluster.

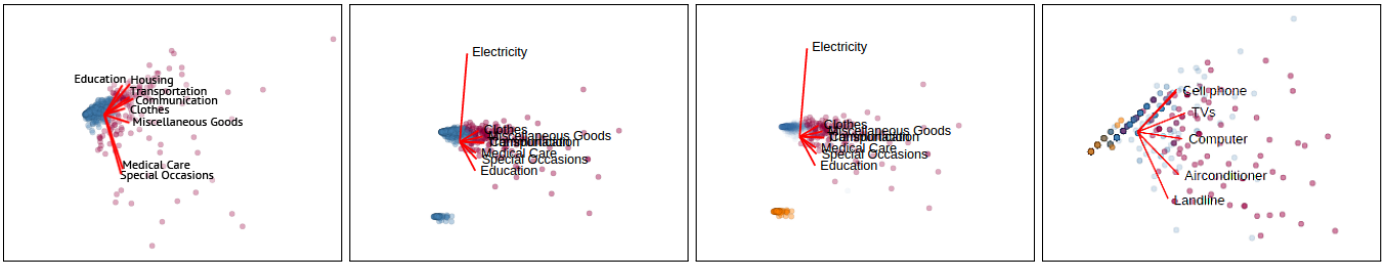
There are now two main clusters – households that have electricity and households that do not (the new cluster on the bottom). Ken further notices that all households with electricity also have high expenditure on services, while households without electricity have very low expenses on services. To explore this further, Ken tags the electricity-less households of the new cluster in yellow (Fig. 6(c)).

Moving this newly gained labeling to the Electronics Owned subspace (Fig. 6(d)) Ken observes that households with high expenses on services own more electronics; their points are well spread along all positive Electronics axes in the biplot. Ken expected this since all of these households consume electricity, albeit the Basic staple households (blue) less so than the General households (red). However, Ken finds it interesting to see that also at least some of those households without electricity (the yellow points) own cell phones even though they own no other electronics; their points are mainly spread along the positive Cellphone axis. This is a rewarding take-away for Ken – it appears that cell phones represent a ubiquitous commodity, and not a luxury. Cell phones are a must-have device to survive!

#### 6.1.2 Constructing Novel Subspaces

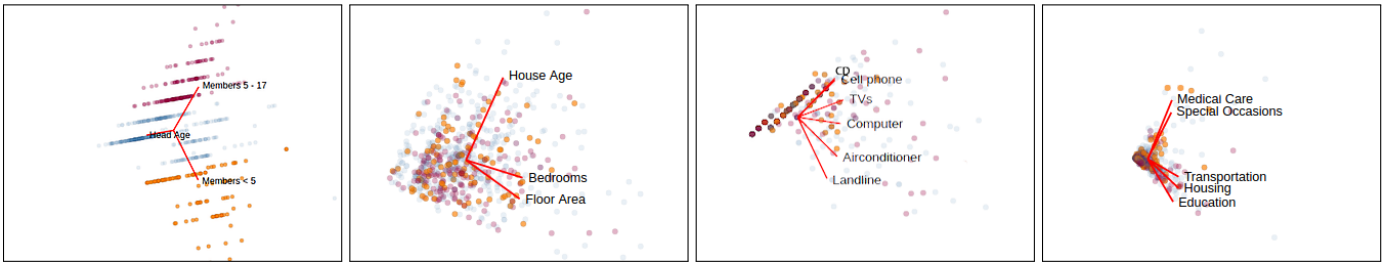
Ken is particularly interested in studying the contrasts that exist among households with different age groups. As this is not one of the initial semantic subspaces the program has identified, Ken adds and configures a new subspace in the Semantic Space View. Fig. 7(a) shows the new subspace Ken has defined, spanned by the age attributes "Members $<$ 5", "Members 5-17", and "Head Age" (shown in the lower right in Fig. 5(a)). Ken selects two clusters, one contains households with children below 5 (yellow) and the other contains households with children 5-17 (red). He finds that most of the subspaces show the same distribution for both clusters. Fig. 7(b) shows the House subspace as an example. The Electronics Owned subspace, however, shows a bias in the distribution of the households (Fig. 7(c)). Households with older children seem to own more electronics items – their red points spread further along the positive biplot axes.

The Expenditure on Services subspace (not shown) also shows some bias. However, according to its scree plot this subspace has an intrinsic dimensionality of 6, possibly introducing projection ambiguities into the biplot. A remedy is to create a lower-dimensional subspace from it to reduce these potential ambiguities. To achieve this, Ken makes use of the vectors displayed in the biplot (compare Fig. 6(a)) and iteratively selects only those attributes in the Attribute Panel




(a) Expenses on Services subspace (b) Extend with Electricity attribute (c) Yellow-tag isolated new cluster (d) The Electronics Owned subspace

Fig. 6: Broadening a subspace beyond its semantic theme using conventional subspace clustering (SURFING). The figures show the results of user interactions that lead to new insights on the Filipino society. (a) The Expenses on Services subspace; high expenses are colored in red. (b) The Extend Subspace facility has added the Electricity attribute. (c) We observe an isolated new cluster and tag its points in yellow. (d) The tagged data are projected into the Electronics Owned subspace.




(a) Household Age subspace (HS) (b) Proj. HS into House subspace (c) Proj. HS into Electron. Owned (d) Proj. HS into Expenses Services

Fig. 7: Constructing novel subspaces. This scenario compares Filipino households with members of different age groups. (a) The subspace created from the three household age attributes. (b) The Household Age subspace is projected into the House subspace. (c) The Household Age subspace is projected into the Electronics Owned subspace. (d) The Household Age subspace is projected into the Expenses on Services subspace.

| Subspace Name | Attributes |
|---|---|
| Age | built, renovated |
| Value | price, grade, condition |
| Facilities | school, hospital, police, transportation |
| Room | bathroom, bedroom, basement, attic |
| Entertainment | landmark, restaurant, park |

TABLE 1: The attributes comprising each of the five semantic subspaces we identified for the King County Dataset.

that contribute to the separation of the two clusters. These are the two sets of vectors that are most orthogonal to one another in Fig. 6(a): "Medical Care" and "Special Occasions" on one end, and "Education", "Housing", and "Transportation" on the other. The resulting subspace is shown in (Fig. 7(d)). Ken quickly realizes that households with children under 5 spend more on medical care and special occasions. whereas households with children between 5 and 17 spend more on education, transportation, and housing. These are valuable insights for Ken's report which will likely be read by marketers, policymakers, city planners, and others.

### 6.2 Use Case: King County

Here we follow Zoe, a data analyst at a mortgage company. Zoe is tasked to study the housing situation in King County. She has a dataset[5] on the available housing with attributes related to these houses (another usage scenario is given in the supplement). The dataset has 16 attributes. [6] There is no meta-data for this dataset and our tool estimates the semantic distance using only the label text of the attributes. A closer examination of the Semantic Space View in Figure 8 confirms that the word embedding procedure was able to estimate the meanings of the attributes quite accurately.

Zoe uses our tool's clustering facility to divide the attributes into five semantic groups based on their word embedding and labels them manually (see Table 1). Figure 9a shows a biplot of the Value subspace. Zoe observes a strong correlation between house "Price" and "Grade", while house "Condition" is independent of the two. Next, Zoe colors the points according to the "Grade" attribute.

Figure 9b shows the Entertainment subspace. Zoe notices that the clusters are evenly spread, suggesting that house grade is independent of a house's proximity to sources of entertainment, such as parks, restaurants, and landmarks. The Rooms subspace (Figure 9c) is more interesting in this regard. Zoe observes that the clusters are significantly more separated, which indicates that the number of rooms in a house has a fairly strong effect on the house's grade. This is an important finding for mortgage predictions. We note that these are just a few of many findings Zoe is able to make.

5. https://www.kaggle.com/harlfoxem/housesalesprediction

6. The attributes of the King County dataset are: price, bedrooms, bathrooms, condition, grade, attic, basement, built, renovated, transportation, landmark, restaurant, hospital, police, parks, school.

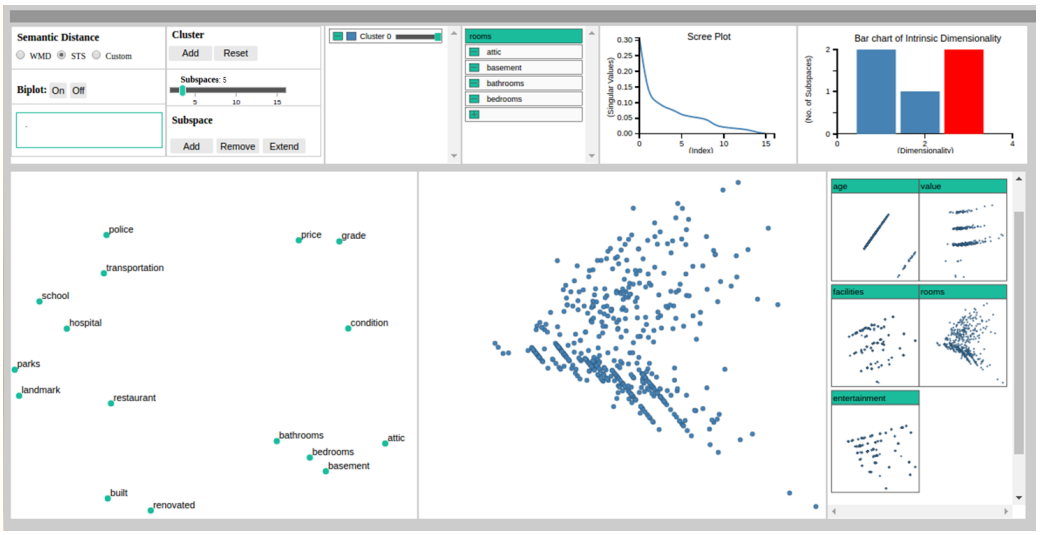



Fig. 8: Our visual analytics tool using the King County Dataset. The tool generated five semantic subspaces. The Subspace Organizer on the lower right shows PCA projections of these, with the corresponding labels printed in the top green box.


(a) Project data into Value subspace (b) Proj. in Entertainment subspace (c) Project data into Rooms subspace

Fig. 9: Projections into three of the five semantic subspaces we identified for the King County dataset (see Table 1). In all of the scatterplots the points are colored by the "Grade" attribute.

## 7 Discussion

We use this section to offer some thoughts on the design choices we made, further experiments we conducted, and the current limitations of our tool and studies, as we see them. We begin by providing a few more general notes and then delve into specific aspects of our tool.

**A Note on Cluster Analysis:** Gauging the effectiveness of the many available clustering algorithms and methods remains difficult since there is no concrete definition of what a cluster actually is. In his position paper, Estivill-Castro writes that "clustering is in the eye of the beholder" [48]. One might agree that a cluster represents a group of data items that are in some sense similar, but the notion of similarity itself can vary widely and can be difficult to capture by a formal metric, such as density, distance from a centroid, or connectivity. All of these work well in some cases, but fail in others. Hence, involving the human "eye" into the process via visual interaction, assisted by machine learning can be a good compromise and this has been demonstrated in the many visual analytics papers written on the subject (see Section 3). The methodology we propose complements these visual techniques by adding, for the first time, the element of semantic meaning of the data, as derived via NLP from the data attributes and optionally provided meta text.

**A Note on Subspace Cluster Analysis:** Similar to general cluster analysis. the construction of a meaningful and manageable set of semantic subspaces is also subject to an analyst's goals, preferences, and domain knowledge. It cannot be automated. The tool we devised can aid analysts in the construction of a set of initial familiar subspaces which can provide insights on their own and then serve as launch pads for a deeper expanded subspace exploration, aided by general subspace clustering techniques. We also encourage the reader to review the Image dataset case study in the supplemental material where we show that clustering in coherent subspaces, such as those generated by our method, can bring a much better differentiation of related data items than when clustering is performed in the entire dataspace.

**Generating Subspaces:** The NLP-constructed Semantic Space View our tool provides allows the user to recognize groupings of semantic themes. There might, however, be settings where our model is unable to find good semantic groupings or only partial groupings. The latter could be a hint that the respective concept spaces are not adequately captured. The user then has the fallback resource to either manually construct semantic subspaces or use the Extend Subspace facility to explore automatically generated subspaces that go beyond the confines of semantic themes. In fact, we found that a workflow that begins with known semantic subspaces and then extends them with conventional

subspace clustering methods gains the best of both worlds. It effectively allows users to opportunistically generate subspaces that extend a semantic theme by variables that have high potential for adding new variations in the projections which could be semantically interesting.

**Augmenting Semantics with Data Metrics:** To see whether conventional subspace algorithms could be enhanced by semantic analysis and so produce fewer subspace clusters, we experimented with merging the semantic space decomposition with a decomposition based on data-centric metrics, call it the *data space*. The data space defines how the different attributes are related to each other in a numerical sense, and the distances of pairs of points are calculated using distance metrics such as cosine or correlation. Akin to the semantic space the data space is represented using a distance matrix $M$, which has $n$ x $n$ dimensions where $n$ is the number of variables in the dataset. A cell in the matrix $c_{i,j}$ represents the distance between variable $i$ and $j$. Similar to our previous work reported in [40] the data space and the semantic space can be merged by taking a weighted average of the two spaces. The sum of weighted distances for each pair of points is computed and from it a fused distance matrix is constructed. However, after testing this approach we did not observe that the added layer of complexity helped much.

**Limitations:** An inherent limitation of our method is that the attribute labels, and any meta-text, must describe the attribute in a meaningful way. The supplemental material gives an example where we added meta-text to "explain" domain-specific attribute labels with natural language text. While this worked quite well, more studies are needed to fully explore this mechanism. For example, for an image dataset, each dimension is a pixel location which does not provide enough information about the attribute, unless it is part of a labeled region or the image has a descriptive caption possibly coupled with automated object recognition.

Another limitation is that word embeddings constructed with Word2Vec suffer from the problem of conflation of word sense, where *word sense* is a meaning of the word. In the embedding each word is represented using only one vector. If a word has many meanings then the vector representation of the word will be the union over the different meanings of the word. We saw an example of this in the Semantic Space of Figure 8 where the 'condition' attribute was halfway between the Room and the Value concept. More advanced NLP methods such as BERT [49] overcome this problem. They operate on a wider, sentence-level context and can so produce multiple mappings for a given word based on its semantics.

Further, it is also possible that there are not enough samples of the word in the text corpus to learn the embedding of the word properly. Likewise, there can be scenarios where one or more of the attribute labels are not represented in the corpus at all. Consequently, the semantic space will be inaccurate. As a remedy, we provide an interactive visual analytics interface that allows users to manually alter the position of the attribute words to mitigate this problem.

Also, while biplots are intuitive to visualize the data in the context of the attributes of the subspace – biplots are widely used in the statistics community – they are nevertheless linear projections and as such can present ambiguities when the number of major principal components of a subspace is significantly greater than two. Therefore, we advise the user to be cautious when interpreting biplots and point to the Intrinsic Dimensionality Plot as a visual aid to assess the validity and trustworthiness of a given biplot. Fortunately, a well defined subspace tends to have an intrinsic dimensionality far less than that of a full data space, often no more than 3-4 dimensions [50], and therefore a subspace biplot is a fairly reliable visual evaluation tool.

Finally, our case studies have used datasets with a fairly modest number of dimensions (less than 100). While these types of datasets are fairly common in real life applications, datasets with substantially higher dimensionality are also frequently encountered. Common remedies here are dimension reduction and level of detail management. The former could use correlation analysis and synonym detection to cull redundant dimensions from the data. The latter could take advantage of taxonomies defined on the domain. A taxonomy is a hierarchy of hypernyms, such as veal - meat. Word embeddings are a popular method for discovering the hierarchical structure of concepts but they are not perfect. Our prior work on Taxonomizer [40] demonstrates a system that inserts the user into the loop to aid in the construction of fully labeled hierarchies from data with many dimensions.

## 8 Conclusions

We presented a new paradigm for subspace cluster analysis, addressing the need for better tools to deal with the massive number of possible informative subspaces that can be found in multivariate datasets. A subspace decomposition of a data space is attractive since these subspaces are usually of much lower intrinsic dimensionality and therefore easier to understand, explore, and visualize. Our novel approach is rooted in the idea of using 2D embeddings of the data attributes constructed from text related to the attributes to create a set of semantic subspaces which have a higher likelihood to bear useful information for analysts.

We believe that our method offers a different way of looking at subspaces, one that can reveal insights into the data that might be more difficult to obtain using views derived from numerical properties of the data only. In comparison to conventional data-driven subspace analysis methods, our technique leverages the user's understanding of the semantics to organize the data in a more meaningful and domain-oriented way, and then use it as a starting point for a more conventional exploratory analysis aided by the various facilities we provide. Future work will apply our tool in active applications and refine its functionalities.

## Acknowledgments

This research was partially supported by NSF grant IIS 1941613 and IIS 1527200.

## References

[1] I. Jolliffe, *Principal Component Analysis*. Wiley, 2002.
[2] J. B. Kruskal, "Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis," *Psychometrika*, vol. 29, no. 1, pp. 1–27, 1964.
[3] L. Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of Machine Learning Research*, vol. 9, no. 11, pp. 2579–2605, 2008.

[4] M. W. Eysenck and M. Brysbaert, *Fundamentals of Cognition*, 2018.

[5] C. Baumgartner, C. Plant, K. Railing, H.-P. Kriegel, and P. Kroger, "Subspace selection for clustering high-dimensional data," in *IEEE ICDM*, 2004, pp. 11–18.

[6] J. A. Hartigan, "Printer graphics for clustering," *Journal of Statistical Computation and Simulation*, vol. 4, no. 3, pp. 187–213, 1975.

[7] A. Inselberg and B. Dimsdale, "Parallel coordinates: a tool for visualizing multi-dimensional geometry," in *IEEE Visualization*, 1990, pp. 361–378.

[8] T. Dang and L. Wilkinson, "Scagexplorer: Exploring scatterplots by their scagnostics," in *IEEE PacificVis*, 2014, pp. 73–80.

[9] M. Ester *et al.*, "A density-based algorithm for discovering clusters in large spatial databases with noise." in *ACM KDD*, vol. 96, no. 34, 1996, pp. 226–231.

[10] I. Assent, R. Krieger, E. Müller, and T. Seidl, "Dusc: Dimensionality unbiased subspace clustering," in *IEEE ICDM*, 2007, pp. 409–414.

[11] E. J. Nam, Y. Han, K. Mueller, A. Zelenyuk, and D. Imre, "Clustersculptor: A visual analytics tool for high-dimensional data," in *IEEE VAST*, 2007, pp. 75–82.

[12] M. Gleicher, "Explainers: Expert explorations with crafted projections," *IEEE Transactions on Visualization and Computer Graphics*, vol. 19, no. 12, pp. 2042–2051, 2013.

[13] H. Kim, J. Choo, H. Park, and A. Endert, "Interaxis: Steering scatterplot axes via observation-level interaction," *IEEE Transactions on Visualization and Computer Graphics*, vol. 22, no. 1, pp. 131–140, 2015.

[14] F. Zhou *et al.*, "Dimension reconstruction for visual exploration of subspace clusters in high-dimensional data," in *IEEE PacificVis*, 2016, pp. 128–135.

[15] X. Yuan, D. Ren, Z. Wang, and C. Guo, "Dimension projection matrix/tree: Interactive subspace visual exploration and analysis of high dimensional data," *IEEE Transactions on Visualization and Computer Graphics*, vol. 19, no. 12, pp. 2625–2633, 2013.

[16] 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*, vol. 17, no. 12, pp. 2591–2599, 2011.

[17] K. Watanabe, H.-Y. Wu, Y. Niibe, S. Takahashi, and I. Fujishiro, "Biclustering multivariate data for correlated subspace mining," in *IEEE PacificVis,*, 2015, pp. 287–294.

[18] Z. Zhang, K. T. McDonnell, and K. Mueller, "A network-based interface for the exploration of high-dimensional data spaces," in *IEEE PacificVis*, 2012, pp. 17–24.

[19] C. Aggarwal, J. Wolf, P. Yu, C. Procopiuc, and J. Park, "Fast algorithms for projected clustering," *ACM SIGMOD*, vol. 28, no. 2, pp. 61–72, 1999.

[20] C.-H. Cheng, A. Fu, and Y. Zhang, "Entropy-based subspace clustering for mining numerical data," in *ACM SIGKDD*, 1999, pp. 84–93.

[21] A. Tatu *et al.*, "Subspace search and visualization to make sense of alternative clusterings in high-dimensional data," in *IEEE VAST*, 2012, pp. 63–72.

[22] ——, "Clustnails: Visual analysis of subspace clusters," *Tsinghua Science and Technology*, vol. 17, no. 4, pp. 419–428, 2012.

[23] J. Wang, X. Liu, and H.-W. Shen, "High-dimensional data analysis with subspace comparison using matrix visualization," *Information Visualization*, vol. 18, no. 1, pp. 94–109, 2019.

[24] M. J. Greenacre, *Biplots in Practice*. Fundacion BBVA, 2010.

[25] I. Assent, R. Krieger, E. Müller, and T. Seidl, "Visa: visual subspace clustering analysis," *ACM SIGKDD Explorations Newsletter*, vol. 9, no. 2, pp. 5–12, 2007.

[26] B. Wang and K. Mueller, "The subspace voyager: Exploring high-dimensional data along a continuum of salient 3d subspaces," *IEEE Transactions on Visualization and Computer Graphics*, vol. 24, no. 2, pp. 1204–1222, 2018.

[27] S. Liu, B. Wang, J. J. Thiagarajan, P.-T. Bremer, and V. Pascucci, "Visual exploration of high-dimensional data through subspace analysis and dynamic projections," *Computer Graphics Forum*, vol. 34, no. 3, pp. 271–280, 2015.

[28] J. E. Nam and K. Mueller, "Tripadvisor^{ND}: A tourism-inspired high-dimensional space exploration framework with overview and detail," *IEEE Transactions on Visualization and Computer Graphics*, vol. 19, no. 2, pp. 291–305, 2013.

[29] D. Jäckle, M. Hund, M. Behrisch, D. Keim, and T. Schreck, "Pattern trails: visual analysis of pattern transitions in subspaces," in *IEEE VAST*, 2017, pp. 1–12.

[30] E. Elhamifar and R. Vidal, "Sparse subspace clustering: Algorithm, theory, and applications," *IEEE Transactions on Pattern Analysis and Machine intelligence*, vol. 35, no. 11, pp. 2765–2781, 2013.

[31] X. Peng, J. Feng, J. T. Zhou, Y. Lei, and S. Yan, "Deep subspace clustering," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 12, pp. 5509–5521, 2020.

[32] Y. Ren *et al.*, "Deep clustering: A comprehensive survey," *arXiv preprint arXiv:2210.04142*, 2022.

[33] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[34] K. Erk, "Representing words as regions in vector space," in *CoNLL*, 2009, pp. 57–65.

[35] G. Hinton, "Learning distributed representations of concepts," in *CogSci*, vol. 1, 1986, p. 12.

[36] D. Rumelhart, G. Hinton, and R. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, 1986.

[37] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *NIPS*, 2013, pp. 3111–3119.

[38] T. Chen *et al.*, "Domain specific word embeddings for natural language processing in radiology," *Journal of Biomedical Informatics*, vol. 113, p. 103665, 2021.

[39] J. Risch and R. Krestel, "Domain-specific word embeddings for patent classification," *Data Technologies and Applications*, 2019.

[40] S. Mahmood and K. Mueller, "Taxonomizer: Interactive construction of fully labeled hierarchical groupings from attributes of multivariate data," *IEEE Transactions on Visualization and Computer Graphics*, vol. 26, no. 9, pp. 2875–2890, 2019.

[41] T. Schick and H. Schütze, "Attentive mimicking: Better word embeddings by attending to informative contexts," *arXiv preprint arXiv:1904.01617*, 2019.

[42] T. Kenter and M. de Rijke, "Short text similarity with word embeddings," in *ACM CIKM*, 2015, pp. 1411–1420.

[43] M. Kusner, Y. Sun, N. Kolkin, and K. Weinberger, "From word embeddings to document distances," in *ICML*, 2015, pp. 957–966.

[44] E. Gansner and Y. Hu, "Efficient node overlap removal using a proximity stress model," in *Graph Drawing*, 2008, pp. 206–217.

[45] D.-T. Lee and B. J. Schachter, "Two algorithms for constructing a delaunay triangulation," *International Journal of Computer & Information Sciences*, vol. 9, no. 3, pp. 219–242, 1980.

[46] J. H. Lee, D. Coelho, and K. Mueller, "Cluster appearance glyphs: A methodology for illustrating high-dimensional data patterns in 2-d data layouts," *Information*, vol. 13, no. 1, p. 3, 2021.

[47] V. Satopaa, J. Albrecht, D. Irwin, and B. Raghavan, "Finding a" kneedle" in a haystack: Detecting knee points in system behavior," in *IEEE ICDCS Workshops*, 2011, pp. 166–171.

[48] V. Estivill-Castro, "Why so many clustering algorithms: a position paper," *ACM SIGKDD Explorations Newsletter*, vol. 4, no. 1, pp. 65–75, 2002.

[49] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[50] B. Wang and K. Mueller, "Does 3d really make sense for visual cluster analysis? yes!" in *IEEE VIS 3DVis Workshop*, 2014.


**Salman Mahmood** received his BSc. from Lahore University of Management Sciences (LUMS). He completed his PhD from Stony Brook University in 2018 under the guidance of Prof. Klaus Mueller. His research interests include information visualization data analytics and machine learning. He received the Stony Brook Computer Science Fellowship. He is now a software engineer at Google.

**Klaus Mueller** is currently a professor of computer science at Stony Brook University and a senior scientist at Brookhaven National Lab. His research interests include explainable AI, visual analytics, data science, and medical imaging. He won the US NSF Early CAREER Award, the SUNY Chancellor's Award for Excellence in Scholarship and Creative Activity, and the IEEE CS Meritorious Service Certificate. To date, his 300+ papers have been cited over 12,500 times.