Erschienen in: Tsinghua Science and Technology ; 17 (2012), 4. - S. 419-428

ClustNails: Visual Analysis of Subspace Clusters*

Andrada Tatu**, Leishi Zhang, Enrico Bertini, Tobias Schreck, Daniel Keim,
Sebastian Bremm†, Tatiana von Landesberger†

Department of Computer and Information Science, University of Konstanz, Konstanz 78457, Germany;
† Department of Computer Science, Technische Universität Darmstadt, Darmstadt 64283, Germany

Abstract: Subspace clustering addresses an important problem in clustering multi-dimensional data. In sparse multi-dimensional data, many dimensions are irrelevant and obscure the cluster boundaries. Subspace clustering helps by mining the clusters present in only locally relevant subsets of dimensions. However, understanding the result of subspace clustering by analysts is not trivial. In addition to the grouping information, relevant sets of dimensions and overlaps between groups, both in terms of dimensions and records, need to be analyzed. We introduce a visual subspace cluster analysis system called ClustNails. It integrates several novel visualization techniques with various user interaction facilities to support navigating and interpreting the result of subspace clustering. We demonstrate the effectiveness of the proposed system by applying it to the analysis of real world data and comparing it with existing visual subspace cluster analysis systems.

Key words: subspace cluster analysis; visualization; data exploration; pixel-based techniques

Introduction

Clustering is one of the most prominent techniques used to analyze large and complex data sets, and visualization is often helpful in understanding the output of a given clustering method. A clustering algorithm assesses the relationships among objects of a data set by organizing objects into clusters, such that objects within a cluster are similar to each other but dissimilar from objects in other clusters. Clustering has a wide range of application in areas such as business intelligence, pattern recognition, image or document analysis, and bioinformatics. With the fast development of modern technologies, vast amounts of high-dimensional data are generated. This poses new challenges for clustering that require specialized solutions. In multi-dimensional spaces it is likely that given any pair of points, there exist at least a few dimensions on which the points are far apart. Traditional clustering methods tend to break down because of this inherent sparsity of the points. To gain the full potential from high-dimensional data, many approaches have been proposed in the past to tackle the high-dimensionality problem. Among those approaches subspace clustering is one of the most actively researched areas, with many algorithms being proposed[1].

In multi-dimensional data, clusters exist often only in a subset of the dimensions. Figure 1 illustrates the concept of a subspace cluster - given three dimensions X, Y, and Z, clusters may exist in different subspaces. For example, the three cuboids highlight the region of three clusters, each of which exists in a different set of dimensions: X and Y, X and Z, and Y and Z. Subspace clustering techniques aim to find these clusters which might otherwise remain

*Supported by the German Research Foundation, by receiving funding from the DFG-664/11 Project

**To whom correspondence should be addressed.
E-mail: tatu@dbvis.inf.uni-konstanz.de; Tel: 49-7531-884364

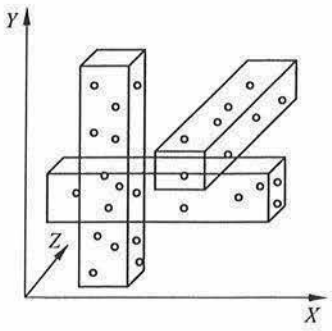


Fig. 1 Data projected in several subspaces.

hidden if a traditional clustering algorithm was applied. Subspace clustering gives for each cluster (1) the objects belonging to the cluster, and (2) the subset of dimensions which constitute the cluster. Based on the type of subspace clustering method, there exist two forms of output: a partitioning of the data into separate clusters and clusters allowing for overlapping elements. Overlap may also exist between the sets of dimensions constituting the clusters.

Designing effective visualizations to help analyze the clustering result is not trivial. In addition to the cluster membership information, the relevant sets of dimensions and the overlaps of memberships and dimensions need to be considered. Although a number of techniques (e.g., Parallel Coordinates[2,3], Scatterplot Matrices[4], HeatMaps[5], exist for visualizing traditional clustering results, little research has been carried out for visualizing subspace clustering results. There is a need for effective systems which allow the comparison and analysis of clusters in arbitrary subspace projections, supporting overview and in-depth study of the subspace clustering results.

We present ClustNails, a novel visualization system for mining subspace clusters and analyzing the results. The system takes multi-dimensional data as input, and applies a user-selectable subspace clustering algorithm from a set of algorithms, to group the objects into clusters. The system displays the subspace cluster results using two appropriately designed visual representations - Spikes and HeatNails. These representations support the interpretation the result of subspace clustering algorithms by visualizing characteristics of the clustering results from different perspectives. Appropriate ordering techniques are integrated with the visualization to help extracting meaningful patterns from the clustering results.

The main contributions of this paper are (1) an integrated data analysis and visualization tool for mining patterns in multi-dimensional data using subspace clustering algorithms, (2) a characterization of subspace cluster analysis tasks and the resulting design space, (3) two novel visualization techniques, Spike and HeatNail, for analyzing subspace clustering results, and (4) an appropriate ordering techniques for pattern extraction.

1 Subspace Clustering

1.1 Subspace clustering algorithms

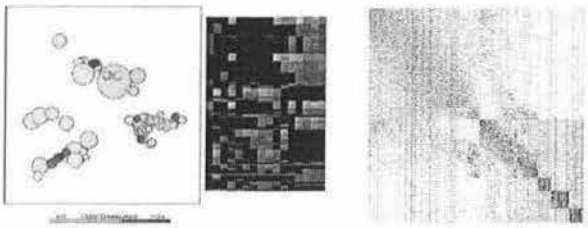
Given a set of data points in some multi-dimensional space, a subspace clustering algorithm aims to find a subset C of data points together with a subset D of dimensions such that the points in C are closely clustered in the subspace of dimension D.

The most critical part of subspace clustering is the subspace generation. Given a d-dimensional space, there are 2^d possible subsets of dimensions. It is computationally infeasible to examine each possible subset to find subspaces that have a higher density than a given threshold. A number of subspace clustering algorithms with strategies for narrowing down the search space have been proposed[1,6] and can be categorized as being bottom-up or top-down approaches, depending on their mode of operation. Proclus[7], for example, takes a top-down approach and extends the traditional k-medoid clustering algorithm. The k-medoid algorithm starts with an initial partition and then iteratively assigns objects to medoids, computes the quality of clustering, and improves the partition and medoid. Proclus extends k-medoid by associating medoids with subspaces and improves both partitions and subspaces iteratively. Proclus takes two input parameters: the number of clusters k and the average number of dimensions l.

1.2 Subspace cluster visualization

While a rich body of research has been carried out in designing subspace clustering algorithms, surprisingly little attention has been paid to developing visualization tools to help analyze the clustering result. To our knowledge, only three subspace cluster visualization systems exist: VISA[8], Heidi Matrix[9], and Ferdosi's astronomical data subspace clustering system[10].

The VISA system implements both a global view and an in-depth view (see Fig. 2a) to help interpret the subspace clustering result. In the global view, the



(a) (b)

Fig. 2 (a) VISA system[8]. Left: MDS projection for the global view of clusters. Right: Matrix of subspace clusters for in-depth view. (b) Heidi Matrix[9] over a subspace.

subspace clusters are projected onto a 2-D display using a Multi-Dimensional Scaling (MDS) projection. The MDS projection in VISA provides a good overview of the clustering results. However, using circles of different sizes in the MDS projection in VISA can be problematic; the distance between two clusters can be obscured by the radius of the circles, and the overlap between clusters often causes a cluttered display.

Heidi Matrix uses a complex arrangement of subspaces in a matrix representation. This matrix is based on the computation of the k-Nearest Neighbors (kNN) in each subspace (see Fig. 2b). Rows and columns represent the data items, and each entry (i, j) in the matrix represents the number of subspaces in which i and j are neighbors. A categorical coloring scheme is used to color the cells according to the particular combination of subspaces in which two data items are neighbors. In addition, rows and columns are ordered according to the output generated by a clustering algorithm. The biggest advantage of Heidi Matrix is that it displays the full information of the data and the subspace clustering result. However, the rather abstract visual mapping scheme makes interpretation of the results difficult.

Ferdosi et al. [10] proposed an algorithm for finding interesting subspaces in astronomical data as well as a visual system for displaying the results. The algorithm identifies candidate subspaces from data and ranks those by a quality metric based on density estimation and morphological operators. The result subspaces are visualized in different forms: line graphs for 1-dimensional subspaces, 2-D scatter plots for 2-dimensional subspaces, and Principle Component Analysis (PCA) projections for subspaces with higher dimensionalities (see Fig. 3). Ferdosi's work provides some interesting insight into subsets of dimensions in astronomical data with a high density of data objects.

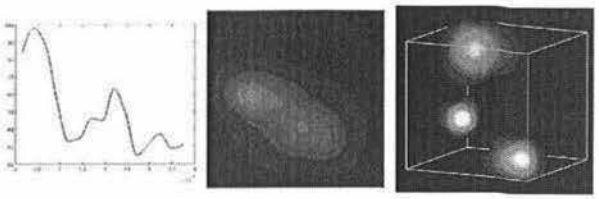


Fig. 3 Visualization techniques applied in Ferdosi's work[10]. Left: 1-D subspace. Middle: 2-D subspace. Right: Subspace with 3 or more dimensions.

However, the algorithm does not assign objects to subspaces. Hence, the subspace clustering information is partially missing from both the data mining and the visualization compared to VISA and Heidi Matrix and there is no direct way of comparing subspaces.

In all of the above mentioned visualization systems, the visualization of overlapping dimensions and overlapping clusters is lacking. It is difficult to see and compare such overlapping information in the visual representations.

1.3 Task definition for visual subspace cluster analysis

Subspace cluster visualization remains a challenging task due to the multiple types of information contained in subspace clustering results such as subspaces, cluster membership of objects, and overlap between subspaces and clusters. To develop effective visualization systems for subspace cluster analysis, it is necessary to take into consideration the different tasks that are involved in the data analysis and use it as a base for exploring the design space. We identify the following main tasks an appropriate subspace cluster visualization technique needs to address.

Reveal properties of individual clusters How many records does the cluster contain? How many dimensions are involved, and at which weights? And how are the data values distributed in each of the contained dimensions?

Cluster comparison How do clusters differ with respect to contained records and involved dimensions? Is there overlap between records and dimensions, or are they distinct?

Quality of the generated cluster output How good is the clustering quality produced by a given algorithm? How sensitive is the output with respect to parameter variations?

We take these task considerations as a baseline for developing the ClustNails system in the next section.

While we have not formally evaluated the degree to which ClustNails fulfills each of these criteria, we find they are at the core of the functionality that ClustNails offers.

2 The ClustNails System

ClustNails is designed as an interactive visualization tool for subspace clustering analysis. It integrates a number of subspace clustering algorithms with novel visual representations and ordering techniques to help analysts generate subspace clusters from multi-dimensional data and identify interesting patterns from the visualization models. We next provide an overview of the design and main functionalities of the system, as well as a detailed description of the visualization and ordering techniques applied.

2.1 Overview

ClustNail integrates the OpenSubspace library of Weka[11] which contains a range of subspace clustering algorithms including Clique, Doc, Fires, Proclus, MineClus, INSCY, P3c, Schism, Statpc, and Subclu. The system takes multi-dimensional data as input, clusters the objects using a user-selected subspace clustering algorithm, and displays the clustering result in a multi-view user interface. A number of ordering functions allow the analyst to examine the results and compare clusters from different perspectives. Various user interactions are added to allow the user to select clustering algorithms, parameters, and the order of the clustering results in the visualization panels. A linking-and-brushing function is implemented such that dimensions/clusters of interest can be highlighted in different views. By placing the mouse cursor over an item (record, dimension, or cluster) in the visualization panel, the analyst can see detailed information of the item in a tooltip.

Figure 4 illustrates the workflow supported by our tool. Figure 4 (left) shows that the system loads a d-dimensional data set as input and a user-defined clustering algorithm finds subspace clusters, where each cluster may exist in its own subset of dimensions. Figure 4 (middle) shows that each cluster is measured in terms of the number of instances and associated number of dimensions; this information, together with the output of the subspace clustering algorithm is visualized in a multiple view visualization panel which include a Spikes view for cluster-centric analysis (top) and a HeatNails view for record-centric analysis (bottom). Figure 4 (right) shows that the order of clusters, dimensions, and records can be rearranged in each view for easy comparison between clusters. Next, we describe the different views and supported ordering strategies.

2.2 Visualization components

2.2.1 Visualization of clusters: Spikes view

The Spikes view is a cluster-oriented view and provides a matrix of thumbnails, each representing a subspace cluster. Each cluster is visualized in a circular area which contains radial spikes. The spikes represent the individual dimensions (the subspace) which define the given cluster, and spike length is scaled according to the weight (importance) of a dimension for the cluster (see below for the definition). The radial dimension sequence is identical for each spike. The number of records in the cluster is represented by the area size of the inner circle.

Subspace clustering algorithms provide as output a subset of dimensions D_k for each cluster SC_k, as well as the set of instances (records) of this cluster X_k. Given a dimension m within the set of dimensions D_k in a subspace cluster SC_k, we define the weight of that dimension in that cluster as

$$w_k^m = \frac{\sum_{x_i^m \in X_k^m} |x_i^m - c_k^m|}{|X_k|} \quad (1)$$

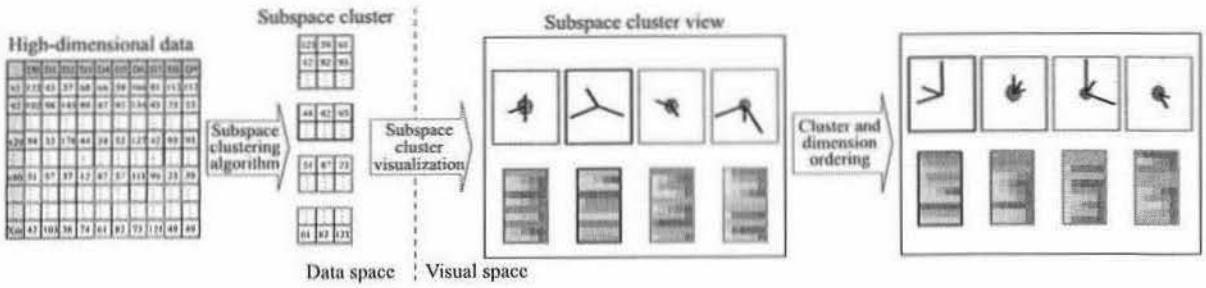


Fig. 4 Workflow of subspace cluster analysis using the ClustNails system.

where c_k^m is the center of the points in SC_k along the dimension m, x_i^m the value in dimension m of the point x_i of this cluster and $|X_k|$, the number of elements in SC_k. The smaller w_k^m is, the more compact are the points around the center in dimension m. This implies that dimensions with smaller weights have better clustered points and are defined as more important for a cluster. We normalize the weights w_k^m for all dimensions of all clusters to the interval [0, 1] and map the corresponding values inversely to the length of the spike. The lower w_k^m (the more important the dimension), the longer the corresponding spike. Note that owing to our definition of w_k^m, the relationship between weights and importance is inverse, and we reflect this by an inverse mapping between weights and size of the visual attribute (the spikes). Also note that in case the given subspace cluster algorithm natively outputs weights for each dimension, those weights can also be mapped to the spike length.

The visual representation for each subspace cluster is a circle in the Spikes view. Each spike in a circle represents a dimension contained in that subspace. The length of the spike represents the weight of the dimension for that particular cluster (the longer, the more important). The order of the dimensions is identical for each cluster. The inner circles indicate the number of records within each cluster. Figure 5 illustrates the Spikes view.

The resulting Spikes view allows users to quickly recognize overlapping dimensions by comparing the spike patterns of the different clusters. To support this comparison, a background is divided into pies and colored alternatively with two colors (gray and light red). This supports the comparison of the spike angles in two different clusters.

2.2.2 Visualization of records: HeatNails view

The HeatNails view is an extended heat map displaying the data values and dimensions. Rows represent dimensions, and columns represent data items (records). Each HeatNail cell represents a data value of a record in one dimension. Data items are grouped by clusters. These clusters are aligned next to each other and separated by black lines. Data values are normalized globally and mapped to an appropriate color scale. A yellow-to-green color scale is used for dimensions which are members of the given cluster, while a gray scale is used for the remaining dimensions per cluster (see Fig. 6 (bottom)). This allows for an effective visual perception of the distribution of values across dimensions, and the relation between dimensions and clusters with respect to their inclusion in the cluster definition.

We also give a summary representation of the values of the dimensions occurring in the clusters. The distribution of dimension values of each cluster is discretized into a histogram and visualized by color (for dimensions included) and gray scales (for dimensions not included). This allows for easy comparison between clusters with respect to data values. Figure 6 (top) shows these histogram views. Finally, depending on the clustering algorithm, it is possible that records are members in multiple clusters. We illustrate this by marking the cluster IDs of multi-cluster members at the bottom of the display. In addition to the Spikes view, the ClustNails view also allows the quick recognition of overlapping dimensions across the clusters by means of the given color and grey-scale patterns. Both Spikes and ClustNails views incorporate linking and-brushing functionality. Clicking on any set of dimensions/clusters of interest in one view highlights

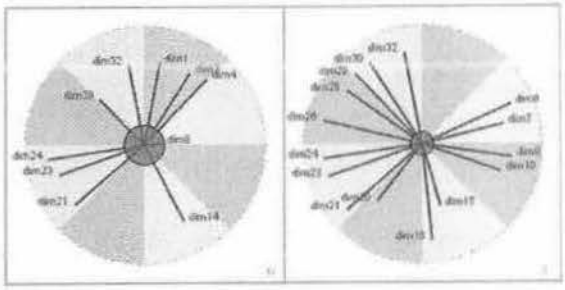


Fig. 5 Two subspace clusters visualized as spikes. The clusters share common dimensions but the importance of the dimensions for the clusters are different. Dimensions #29 and #32 in the left cluster show smaller spikes than in the right cluster, as they are considered less important for the definition of that cluster according to our measure w_k^m. Furthermore, the left cluster has fewer dimensions and more objects than the right cluster.

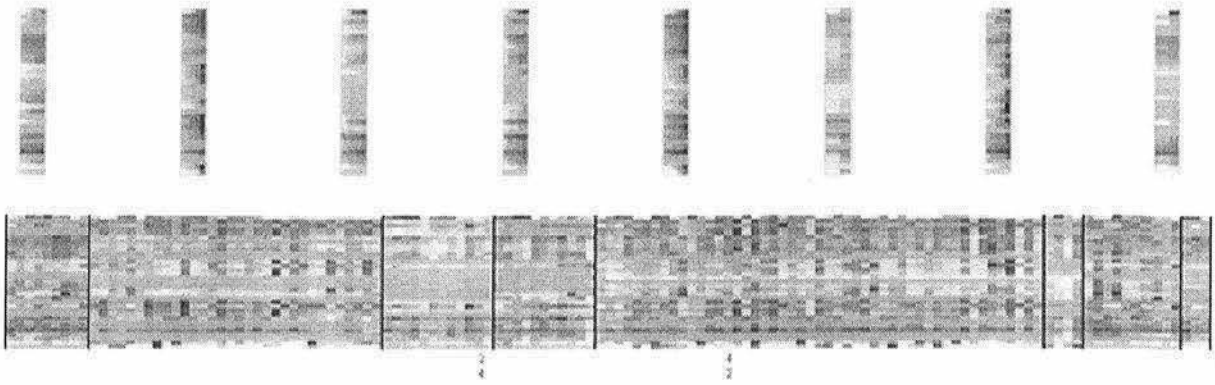


Fig. 6 HeatNails visualization. Bottom: showing the distribution of dimension values for all dimensions (rows) and records (columns). Top: showing histograms for the values of all dimensions per cluster for comparison purposes.

the same dimensions/clusters in all other views.

2.3 Ordering heuristics

Ordering is implemented to support perception of structural similarities in the definition of clusters with respect to dimensions and value distributions. As ordering problems for clusters, dimensions, and records are typically complex NP-complete combinatorial optimization problems[12], we rely on heuristics to order dimensions, records, clusters, and values in the various displays. Our essential idea is to place similar or closely related objects together to help the analyst find interesting patterns.

Dimension ordering We compute a frequency value for each dimension, denoting the number of subspace clusters that are using this dimension. We order the list of dimensions by this frequency value. The dimension ordering can be applied to both the Spikes view and HeatNails view.

Subspace cluster ordering A useful visual representation of subspace clustering results should arrange similar subspaces next to each other to reduce visual search time by the user. We propose an ordering strategy that is formalized in the following. Using the dimension weights defined in Eq. (1), we propose a measure for the global interestingness I^g_{Dm} of a cluster S_k: $I^g_{S_k} = \frac{\sum_m w^m_k}{|D_k|}$, where w^k_m is the weight of dimension $m \in D_k$ of SC_k, and $|D_k|$ is the number of dimensions in this subcluster. We define the global interestingness of a cluster k as the average of the weights of the dimensions contained in this subcluster. This measure is used to determine the first cluster in the ordering. We then use the subspace cluster distance employed in the VISA system[8] to find the most similar cluster, which is placed next to the initial cluster. This distance function is a convex sum of subspace distance and object distance. We continue this placement until all clusters are placed.

Record ordering Two different types of record ordering strategies are implemented in HeatNails. One strategy is to order the records according from min to max with respect to their values in the dimension which has the biggest variance, among all dimensions. A second strategy is to order the records according to the Euclidian distance across the contained dimensions of the given subspace, based on a selected starting record. The starting record, in turn, may either be user selected, or selected automatically as the record which shows the largest variance over all dimensions.

Value ordering A value ordering facility is implemented in the HeatMap view and visible in the top summary row of the HeatNails view. There, in each row the distribution of values in a given dimension is shown. To that end, we sort the values from min to max, and bin them into a user-selectable number of bins. In this view the distribution of values per dimension and cluster is indicated in the form of a color-coded histogram. The histograms help in understanding the distribution of data values within each dimension, and may support finding out why a particular dimension was selected or not by the clustering algorithm.

2.4 Summary and discussion of ClustNails system design

ClustNails is an integrated system for visual subspace cluster analysis. Its design features (1) a number of subspace clustering algorithms from which the user can chose and (2) a design of different visual

representations for the most important aspects of the output of automatic subspace cluster analysis.

Regarding (1), we provide access to a number of state-of-the-art algorithms as contained in the OpenSubspace library[11]. The list of algorithms is extensive.

Regarding (2), we composed a visual display of three aspects. The Spikes view is inspired by radial parallel coordinates (or star glyph) plots and distinguishes clusters from each other, in terms of included dimensions. The radial basis shape in the Spikes view is visually dominant and allows fast perception of cluster properties. Sorting of the cluster glyphs by similarity offloads users (at least partially) from sequential visual search. The Spikes view is complemented by the HeatNails view which is a dimension-oriented detail view that we provide in a coordinated view, below the cluster glyphs. The HeatNails view is based on the ideas of heat maps and the pixel-paradigm for showing the maximum possible information, allocating eventually only one pixel per record dimension (bottom view) or histogram bin per cluster dimension (top view). The overall layout of the three views follows an overview-first approach, from the most aggregate view at the top (the Spikes view of clusters) to the most detailed view (the HeatNails record view) on bottom. The histogram view showing the distribution of dimensions per cluster is located in the middle.

We designed this integrated layout having the different subspace clustering output parameters in mind, and arranged them according to the level of detail provided. While we believe our system design is justified from these considerations, we recognize that other multi-dimensional visualization techniques do exist which could be alternative views in our visualization layout. Parallel coordinates in conjunction with color-coding could be an option. A dedicated user study, as part of future work, could explore design alternatives and compare them with each other.

3 Case Studies and Comparison

We apply the ClustNails system to a real world data set, demonstrating its applicability and illustrating different types of analysis one can perform with it. Then we compare it with the state-of-the-art system VISA[8] to validate the effectiveness of the system and its design.

3.1 Case study: USDA food composition data set

We analyzed the USDA food composition data set (http://www.ars.usda.gov/), which contains a full collection of raw and processed foods characterized by their composition in terms of nutrients. The data comprises more than 7000 records and 44 dimensions. We selected Proclus for the clustering task. We set the number of clusters to 15, and the average number of dimensions to 8. Figure 7 shows the result generated by the system.

From Fig. 7 we can see that clusters C11, C12, C13, and C14 (highlighted red) all share the same two dimensions water and calories, although the sizes of the clusters vary from 4 to 24 records. All the records

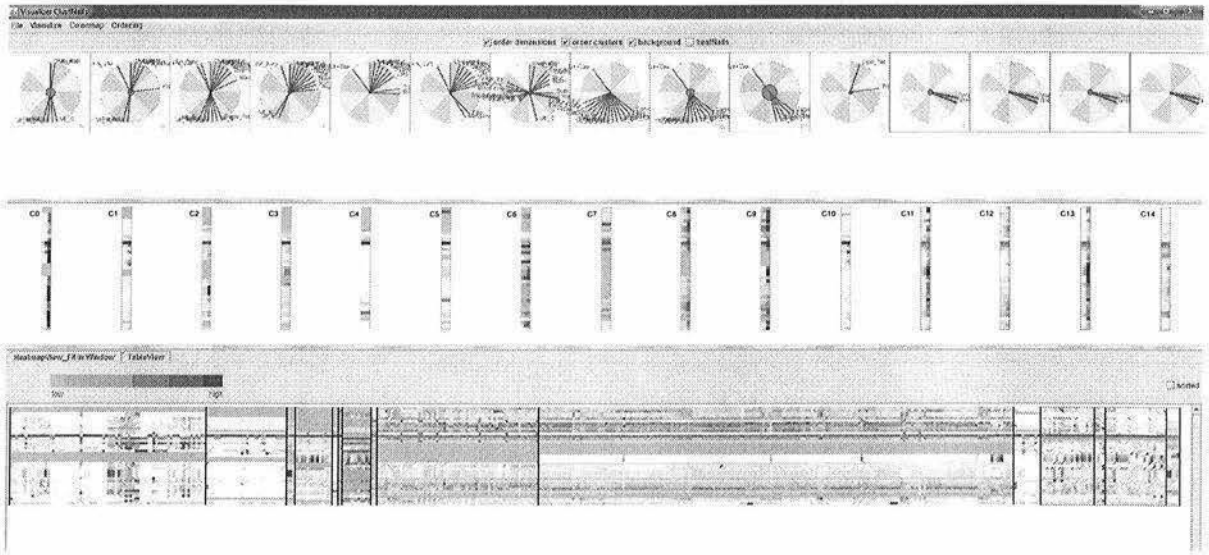


Fig. 7 Visualization of the subspace clusters in the USDA food composition data set generated by Proclus.

share some common features - high water containment and low calories. To gain more understanding of the clustering result, one can drill down to each record by checking the data table or detail-on-demand information displayed in tooltips upon mouse-over actions. It is not difficult to find out that these groups mostly consist of foods which are commonly regarded as "healthy". Foods of similar nature, e.g., lima and mango beans, various types of low-fat dairy products, and soups are placed in the same groups, which means the clustering makes good sense.

Using the value ordering function in the HeatNails, we can further explore the distribution of data values inside each cluster and look for interesting patterns (see Fig. 8). We note that most of the data values in the dimensions not selected by Proclus have relatively large variance. This is not surprising as subspace clustering algorithms are typically designed to reduce the sparsity of data by discarding dimensions that have big variances.

Taking a look at how the same two dimensions are distributed along the other clusters in the sorted view, it is not difficult to identify clusters, like C10, which have similar trends over the two dimensions but have stronger patterns in other dimensions (exceptionally low values for both total lipids and proteins, discussed later), thus the two dimensions are not selected to characterize the cluster. These types of information are not only useful in helping to understand the cluster analysis result, but also add more transparency to the data mining algorithms which are usually hidden from the user in black boxes. From a closer inspection we can identify a cluster which also shares the two dimensions, but with an inverse trend, that is, low water containment and high calories (C6). The detailed information reveals that this cluster represents a whole set of different candies (probably not the most recommendable food for a diet).

Another interesting cluster is C10 which is characterized by an exceptionally low value for both total lipids and proteins. All the other records, excluding the ones in C1, have either consistently high values or higher variances in one of these two dimensions. They represent various kinds of beverages such as alcoholic beverages, teas, and fruit-based toppings. C1 is characterized by the same trend but it forms a different cluster with exceptionally low values for other nutrients like various kinds of fats and vitamin B12. All the foods in C1 are again beverages.

Comparing C10 to C1, one can notice that C10 has, in fact, a very similar distribution of values in the dimensions that are included in C2. This is a clear example in which the output of the algorithm is not optimal and a merge of these two would make sense.

3.2 Comparison with VISA

Figure 9 shows the visualization of the same subspace clusters (same data set, same clustering result) as used

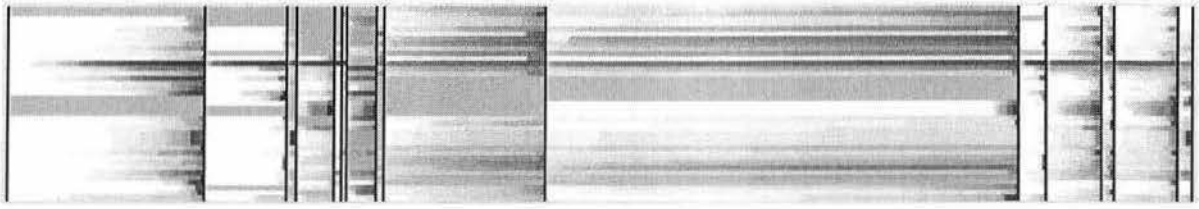


Fig. 8 Sorted view (Value ordering function applied).

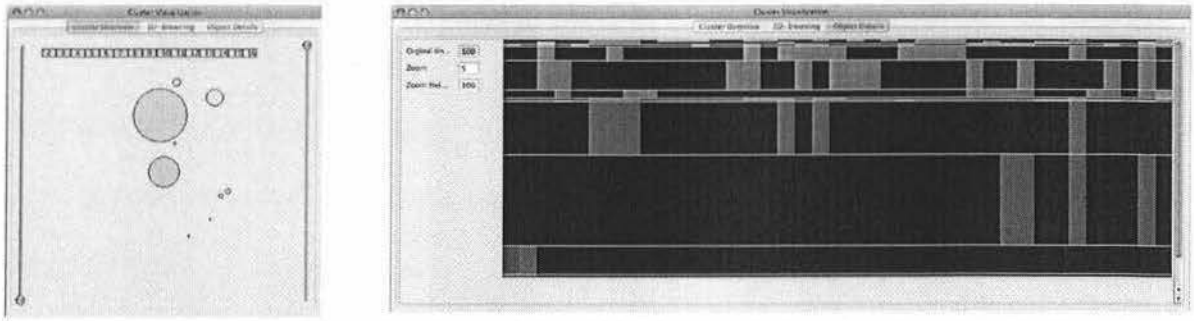


Fig. 9 Visualization of the subspace clusters discussed in the VISA[8] framework discussed in Subsection 3.2. Cluster view (left), record view (right).

for our above case study in VISA[8]. As we can see, the 15 clusters are projected to the 2-D space as MDS-based scatter plots in the cluster view (left screenshot in Fig. 9). Each cluster is represented by a circle scaled according to cluster size. The record-centric view shows the result as a heatmap (right screenshot in Fig. 9), where rows represent records and columns represent dimensions. Different color codes are used in the heatmap: black for unselected dimensions, brightness for interestingness, and hue for data values. We recognize the following benefits in the ClustNails design regarding VISA.

Overlap Circles of different sizes in the VISA MDS projection can cause occlusion problems and end up with over-cluttered displays. For example, only 9 out of 15 clusters are visible in the cluster view in Fig. 9. The Spikes and HeatNails views avoid overlap. One may argue that scatter plots scales better, but in practice the number of clusters in a result is usually small, because a large number of clusters implies, in many cases, a poor performance of the clustering algorithm[1]. Scatter plot visualization, on the other hand, suffers from occlusion problems regardless of the number of clusters. Also, the ClustNails glyphs provide a richer source of information.

Richer information VISA shows only the number of records and dimensions of each cluster and maps the similarities between clusters to distances. The Spikes view in ClustNails extends this basic encoding by including additional information about each cluster, permitting a user to (1) draw richer information from the result and (2) detect and understand the similarities between clusters more easily. Specifically, the spikes permit one to see the detailed dimensions in each subspace and thus to relate one cluster to another. The linking-and-brushing technique implemented in the Spikes view helps in highlighting the shared dimensions among clusters.

Ordering supports comparison The ClustNails ordering techniques place similar clusters, dimensions, and records close to each other. These techniques permit one to detect similarities and dissimilarities between the clusters more easily. No ordering technique is implemented in the current version of VISA.

Scalability The heatmap solution implemented in VISA is initially designed to display a limited number of records that belong to a small subset of clusters. The compression techniques we propose for the thumbnails view of HeatNails can scale up to a much larger number of records and thus is not limited to representing only a subset of data. Subspace clustering algorithms can produce hundreds of subspace clusters in minutes. Our histogram views can be used to visualize this output, they can also be ordered linearly into more rows, or even a two-dimensional ordering heuristic can be developed to make the technique scale.

Non-member dimensions In VISA all data values in the unselected dimensions are colored in black; hence the information in these segments is missing from the visualization. This may be detrimental to data understanding as the information contained in those segments provides evidence of why the clustering algorithm did not select a given dimension to characterize the cluster. The algorithm choice can be justified if the visualization shows extreme values or has large variances in the unselected dimensions.

4 Conclusions and Future Work

Subspace clustering addresses an important problem in clustering multi-dimensional data. The algorithms successfully reduce the noise in multi-dimensional data by showing clusters which exist only in subsets of dimensions in the data. Visualization of subspace clustering results is challenging. In addition to the information contained in traditional clustering results, subsets of dimensions that define clusters, and overlap between dimensions and records needs to be represented in an understandable and uncluttered way. ClustNails was presented as an interactive data analysis and visualization tool for subspace clustering analysis. It provides several novel visualization and ordering techniques to help analysts extract subspace clusters from data and then analyze the results. The system implements linked and ordered cluster-centric (Spikes) and a record-centric (HeatNails) views. We demonstrated the effectiveness of our system design in the analysis of real world data and a comparison with existing visual subspace cluster analysis systems.

In future work we should extend our system to support parameter selection, which is a difficult problem given that each algorithm has its own parameters and different settings may generate very different results. We plan to develop a so called "agreement matrix" among a set of results which shows those parts that most results agree on. The agreement matrix could then be used to evaluate the quality of individual outputs and to help the analyst to

understand the consensus made by different algorithms and parameter settings. Another future direction is to improve the scalability of the ClustNails system. While we have not done a formal evaluation, we assume scalability is restricted to dozens of clusters and dimensions, depending on the resolution of the given display. Some results may contain hundreds of clusters and thousands of dimensions, for which scalable solutions are needed.

References

[1] Kriegel H P, Kröger P, Zimek A. Clustering high-dimensional data: A survey on subspace clustering, pattern-based clustering, and correlation clustering. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 2009, **3**(1): 1-58.

[2] Fua Y H, Ward M, Rundensteiner E. Hierarchical parallel coordinates for exploration of large data sets. In: Proceedings of the Conference on Visualization. IEEE CS Press, 1999: 43-50.

[3] Inselberg A, Dimsdale B. Parallel coordinates: A tool for visualizing multi-dimensional geometry. In: IEEE Visualization. IEEE CS Press, 1990: 361-378.

[4] Becker R, Cleveland W. Brushing scatterplots. *Technometrics*, 1987, **29**: 127-142.

[5] Eisen M, Spellman P, Brown P, et al. Cluster analysis and display of genome-wide expression patterns. *Proceedings of the National Academy of Sciences*, 1998, **95**(25): 14863-14868.

[6] Parsons L, Haque E, Liu H. Subspace clustering for high dimensional data: A review. *ACM SIGKDD Explorations Newsletter*, 2004, **6**(1): 90-105.

[7] Müller E, Günnemann S, Assent I, et al. Evaluating clustering in subspace projections of high dimensional data. *Proceedings of the VLDB Endowment*, 2009, **2**(1): 1270-1281.

[8] Assent I, Krieger R, Müller E, et al. Visa: Visual subspace clustering analysis. *ACM SIGKDD Explorations Newsletter*, 2007, **9**(2): 5-12.

[9] Vadapalli S, Karlapalem K. Heidi matrix: Nearest neighbor driven high dimensional data visualization. In: Proceedings of the ACM SIGKDD Workshop on Visual Analytics and Knowledge Discovery. ACM, 2009: 83-92.

[10] Ferdosi B, Buddelmeijer H, Trager S, et al. Finding and visualizing relevant subspaces for clustering high-dimensional astronomical data using connected morphological operators. In: Proceedings of IEEE Symposium on Visual Analytics Science and Technology (VAST). IEEE CS Press, 2010: 35-42.

[11] Müller E, Assent I, Günnemann S, et al. OpenSubspace: An open source framework for evaluation and exploration of subspace clustering algorithms in WEKA. In: Proc. 1st Open Source in Data Mining Workshop (OSDM 2009) in Conjunction with 13th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD 2009), 2009: 2-13.

[12] Ankerst M, Berchtold S, Keim D. Similarity clustering of dimensions for an enhanced visualization of multidimensional data. In: Proceedings of the IEEE Symposium on Information Visualization. IEEE CS Press, 1998: 52.