Visualization of Subspace Clusterings for High Dimensional Data

Sebastian Fichtner*

Seminar Visual Analysis of Large Datasets
Konstanz University, Summer 2010
Prof. Dr. Daniel Keim
Advisor: Andrada Tatu

Abstract. This seminar paper introduces subcluster visualization as an increasingly important topic in knowledge discovery research. Two recent approaches are discussed in detail: The Heidi Matrix presented by [3] and the Morpheus System as introduced in [1], [2]. In the end, the capabilities and limitations of these methods are revealed and general difficulties of visualizing subclusterings for highdimensional data are concluded.

1 Introduction

Nowadays, we must always assume that data for knowledge discovery might be high dimensional:

- Abstract data has lots of potential features (native and derived).
- Since memory is cheap, saving all possible features doesn't hurt.
- Feature selection by some predefined relevance criteria would **decrease** the probability of discovering unexpected patterns.

Clustering and cluster visualization are general knowledge discovery techniques that may be applied to any kind of data. Therefore they must comprehend the phenomena of high dimensional data. The challenge of clustering such data is to detect clusters in all subspaces. Before analyzing the data, we must assume that any pair of features is linear independent and that a data space of d features really is a d-dimensional space which has 2^d different subspaces. Like the degree of dependence between variables cannot be derived from the marginal distributions alone, the degree to what data points are clustered cannot be detected from single dimension projections. Clustering the data space with all dimensions also might fail to detect all clusters especially with high dimensional data:

- Distances become larger and increasingly similar.
- Different subspaces may have different densities.
- Subspace clusters are masked by irrelevant dimensions.
- The set of irrelevant dimensions is different for each cluster (local feature relevance problem).

* sebastian.fichtner@uni-konstanz.de

Clustering techniques that take these characteristics into account and somehow detect clusters in different subspaces result in a "subspace clustering" which can be considered as a generalization of the term "clustering". A subspace clustering consists of "subclusters". In general different subclusters can overlap in their data objects as well as subspace dimensions. Because of the number of possible subspaces, the visualization of subspace clusterings becomes a challenge of extreme complexity. Not only can subclusters exist in any subspace. Relations between subclusters like similarity, distance, data- and dimension overlap might as well depend on the subspace. In addition, the task of analyzing a single subspace must be simplified. The outcome of pattern detection algorithms depends on the basis vectors (the coordinate system) to which data vectors (data points or -objects) relate. Theoretically there is an infinite number of base vector combinations for each (sub)space. Most clustering algorithms for high dimensional data only consider bases that are a subset of the original attributes from the data set, so that the coordinate systems of all subspaces of interest are axis-parallel. Therefore the terms "feature", "attribute" and "dimension" are treated like synonyms in this work and related literature. It must be noted, that no subspace clustering algorithm can detect all subclusters, because such a simplification and other compromises must be applied. The complexity of both- the high dimensional data and the outcome of subspace clustering algorithms makes their visualization even more important to the purpose of knowledge discovery. Still, only few attempts have been made to tackle this problem. Two recent ones will be discussed in this work.

2 Heidi Matrix

The so called "Heidi Matrix" introduced by [3] is a compact pixel oriented visualization of a given clustering with emphasis on

- overall clustering structure
- structure of single clusters
- closeness of single data points within a cluster
- spatial overlap of clusters in different subspaces

A quadratic pixel matrix of size $n \times n$ displays certain relations in the data set of n objects by color. Columns and rows have the same order which is determined in ways that promote the recognition of patterns. Figure 1 shows a simple example data set and a corresponding visualization. The following sections will explain what exactly the colors mean, -how objects can be ordered, -what patterns occur and how they can be interpreted.

2.1 Color Mapping

Pixel (i, j) is colored if and only if object j is close to object i. Otherwise pixel (i, j) is white. That means: Objects are either close or distant. A core concept of the Heidi Matrix is that the color of close objects reflects the **set of subspaces** in

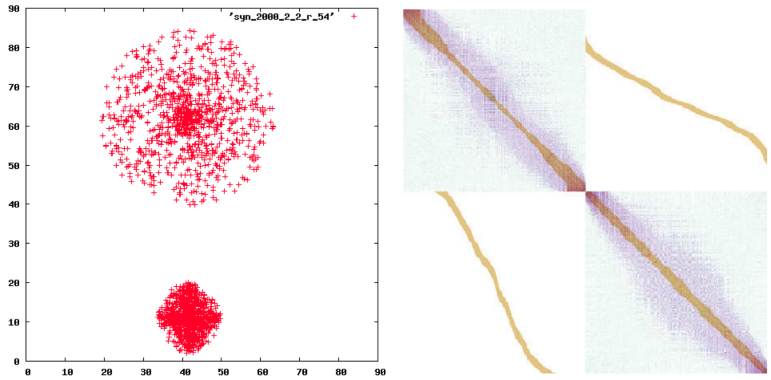

Fig. 1: *2-dimensional data set and a corresponding Heidi Matrix with knn-order*

which they are close. Note that this closeness relation isn't necessarily symmetric. In fact, object j is close to object i in subspace S if j is one of the k nearest neighbors of i in S. Of course there is no upper bound for the distance of "close" objects, but the nearest neighbor relation has proven to be appropriate for high dimensional spaces, because it adapts to varying densities (between subspaces) and high distances. The underlying measure is the euclidean distance.

Since there are $2^d - 1$ possible subspaces, pixels are represented by bit vectors of the same length - one bit for each subspace. One object has k nearest neighbors in every subspace, so the sum of bits set to 1 over all bit vectors in a column or row of the matrix is $k(2^d - 1)$. What is important is, that in general there are $2^{(2^d-1)}$ possible bit vectors (sets of subspaces), so only the m most frequent combinations are colored at all. Less frequent closeness relations are not displayed to avoid clutter. The m chosen subspace sets and their colors are depicted in a color legend like the one shown in Figure 2.

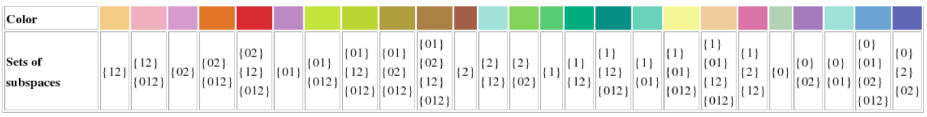

Fig. 2: *Color legend for a Heidi Matrix of 3-dimensional data*

When interpreting the colors of a Heidi Matrix, several issues and limitations should be noted:

– With high numbers of objects the effect of color blending during the process of perception or downsampling has to be considered when interpreting colors.

- It is not displayed how close points have a lot of common subspaces in which they are close, because the similarity of bit vectors does not correspond to similarity of mapped colors.
- Color mapping doesn't consider that points being close in a certain subspace are probably also close in its subsets and thus in all combinations of these subsets (see Figure 2 in the paper).

2.2 Object Order

Without ordering data objects, the image wouldn't show any patterns. So first the objects are separated so that no object of another cluster is between two objects of the same cluster. If noise and outliers of the given clustering are known, they can be treated as one cluster so they wouldn't disturb the visualization. After grouping the objects by clusters, each group is sorted in a way that roughly reflects object closeness within that cluster.

The Matrix in Figure 1 seems to consist of four blocks. In other words, the object order (along rows and columns) separates both clusters, so that the matrix shows one block for each combination of two clusters the same way it shows one pixel for each combination of two objects. Blocks on the diagonal of the matrix visualize how objects are positioned within each cluster whereas the other blocks show how objects of different clusters relate to each other. In Figure 1 the blocks in the top right- and bottom left corners contain brown stripes indicating an overlap of both clusters in this synthetic 2-dimensional data. There is no color legend given for most matrices in the paper, but important aspects of the examples are clear.

To be correct: The brown color of those stripes shows in which set of subspaces both clusters spatially overlap. In this case that set contains just one subspace and this subspace contains just one dimension. If we call the x- and y-axis of the scatterplot in Figure 1 dimensions 0 and 1, than the brown color stands for $\{\{0\}\}$.

It can be seen that the wider cluster on top is the the first cluster in the order of the Heidi Matrix: The stripe covers only a subset (middle part) of its objects, while it covers all objects of the second cluster. Now, if we imagine a projection of all points in the scatterplot onto the "0-axis", we would have lots of points from the first cluster that are distant to points from the second, while all points from the second cluster would be close to some point of the first one. Note that the matrix might be "little" asymmetric in the details, but there is no useful information in that asymmetry. In fact, the overall symmetry helps recognizing patterns.

Ordering can be used to create different kinds of Heidi Matrices for the same data which result in different images. These are the major modifications:

- Two different ordering algorithms were proposed: kNN order and spiral order.
- Ordering can be done in a user defined subspace to display more information about that particular subspace.

– The advantages of different subspace specific images can be integrated to build one composite matrix.

With spiral order the objects are sorted by their distance to the cluster center. kNN order reflects a recursive depth first traversal of the cluster objects using the k nearest neighbors of each object and starting at the object closest to the origin of the data space. Figure 3 demonstrates the visual difference between knn- and spiral order.

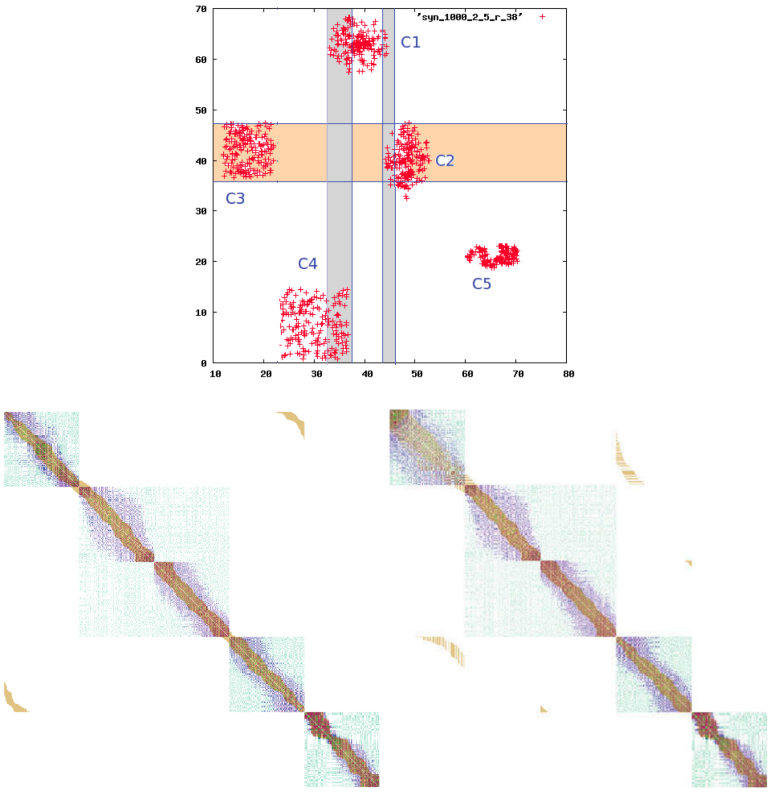

Fig. 3: *Synthetic data containing 5 clusters and corresponding Heidi Matrices with knn- and spiral order applied on subspace* $\{0\}$

2.3 Composite Matrix

As mentioned above, it is useful to order the objects in a certain subspace rather than over all dimensions to achieve clearer patterns for a subspace of special interest. In the data set shown in Figure 3 cluster overlaps $(1, 4)$ and $(2, 3)$

occur in different subspaces $\{0\}$ and $\{1\}$. Because the Heidi Images in the figure are ordered in subspace $\{0\}$, overlap $(1, 4)$ results in a clear stripe pattern, while overlap $(2, 3)$ is just a blurred colored block. If the ordering was done in subspace $\{1\}$ instead, overlap $(2, 3)$ gets clearer while $(1, 4)$ is now blurred as Figure 4 demonstrates. The authors suggest to use those "single dimension images" which

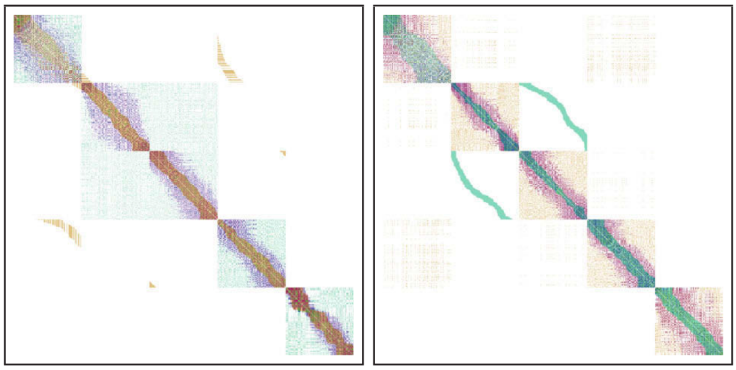

Fig. 4: *Spiral ordered Heidi Matrices over subspace $\{0\}$ and subspace $\{1\}$*

are ordered in only one dimension to create a composite matrix and integrate many distinct patterns into one image. The composite matrix of both matrices in Figure 4 can be seen in Figure 5. The resulting matrix obviously contains more clear patterns. The authors don't explicitly tell how the images are combined,

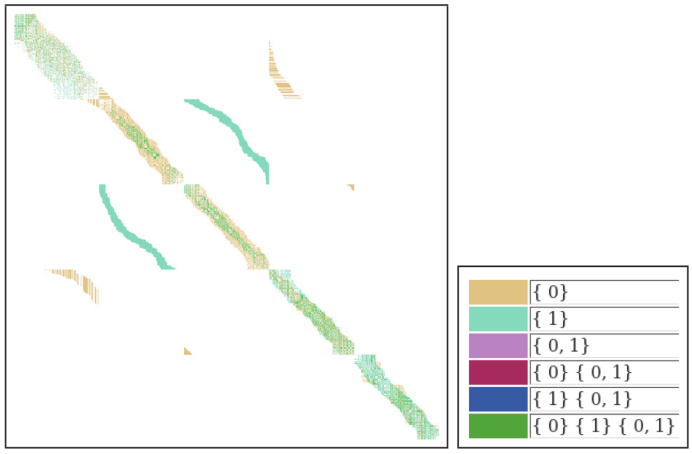

Fig. 5: *Composite Heidi Matrix*

but it would make sense to apply bitwise AND on the bit vectors. The example images support that assumption as well.

2.4 Problems

Because the Heidi Matrix is a compact non-interactive visualizations for a very complex data type, a lot of simplifications and decisions had to be made. It is a quite specific solution that hardly relies on well known visualizations, so the interpretation of a Heidi Matrix is not intuitive and demands a lot of knowledge about its creation. The user must understand how the ordering works to get all the information the image carries. For high numbers of data objects, -clusters or -dimensions Heidi Matrices become completely useless. The number of objects and clusters is limited by screen space resolution. Even more critical is the fact that the number of sets of subspaces is exponential in the number of dimensions and still subspace sets are mapped to color, while the human eye cannot distinguish more than about 20 colors. It is also hard to read how clusters and data objects are related to each other, because only some nearest neighbor relations are displayed. In depth examination of cluster overlaps in certain subspaces demands to create extra images for the subspaces of interest, so the user needs prior knowledge to choose these subspaces. Also the input clusterings are limited to clusters that don't overlap in their data. Since the whole concept relies on the euclidean distance, it is impossible to generate a Heidi Matrix for pure categorical data.

Even when a well defined subspace clustering is already given, the complexity of generating a Heidi Matrix is $O(2^d n^2)$. The authors claim to have created images for 50-dimensional data in about 15 minutes. It remains doubtful how this is possible. However, the algorithm does obviously not scale well with the number of dimensions and is therefore not practical for processing high dimensional data in general.

3 Morpheus System

A system for intuitive visualization and interactive exploration of subspace clusterings is called "Morpheus" and was presented by [1] and [2]. These are the major goals of that system:

- overview- and in detail presentation
- interactive exploration and parametrization
- dimensionality unbiased subspace clustering
- comparison of clusters from different subspaces
- framework for visualization of different clustering algorithms and educational use

The following sections will discuss the Morpheus System and its characteristics in detail.

3.1 Overview Presentation

The Morpheus System provides views for different steps of the whole discovery process. We will first take a look at the overview presentation of the clustering as displayed in Figure 6.

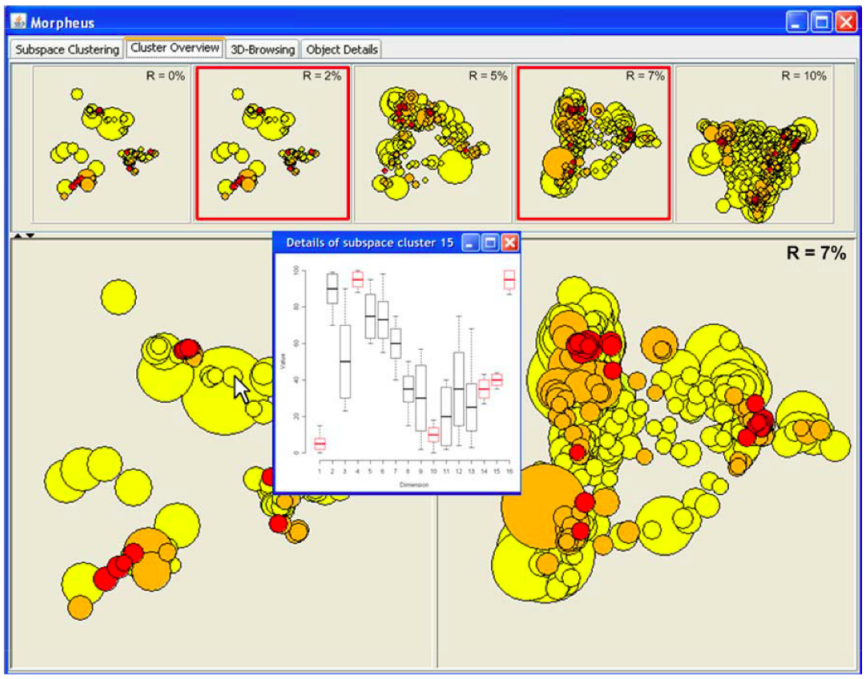

Fig. 6: *Overview presentation in the Morpheus software with bracketing for redundancy parameter*

The colored spheres represent the most important subclusters. To interpret the image correctly we must know what is mapped to the visual variables:

- The closeness (position) of subclusters in screen space reflects their similarity.
- The size of a subcluster sphere corresponds to the number of data objects contained in that subcluster.
- The color scale gives a hint about the number of dimensions of the subcluster subspace. Red means high dimensionality.
- An interestingness measure for subclusters is double coded in the intensity and saturation of the color.
- According to the given screenshots the z-order of subcluster spheres is probably derived from the subclusters dimensionality, because high dimensional subclusters are displayed in front.

The colors in the screenshots are somehow strange compared to how color mapping is described in the paper, because there are apparently only three distinguishable colors. Possibly there was some binning applied but this is not made clear by the authors. To fully understand these mappings we must look at some underlying measures that were developed for VISA - a visual subspace cluster analyses technique presented in [1] on which the Morpheus System partly relies.

3.2 Basic Measures

DUSC (dimensionality unbiased subspace clustering) is actually a density based subspace clustering algorithm. We will not discuss this algorithm but, with the help of its measures we can understand how the distance between subclusters is calculated by VISA and the Morpheus System. DUSC takes different densities of subspaces into account by simply calculating the average **local** density of data objects in each subspace. Therefor the neighborhood of an object (in a certain subspace) is defined as the set of other objects that lie within a predefined radius. The object's density (in that subspace) is the sum of influences of all neighbors. With higher dimensionality and greater distances the number of neighbors and the influences become smaller. Finally, the overall average density in a subspace is called its expected density. To make object densities comparable even between subspaces of different dimensionality, object densities are normalized (divided) by expected densities. An object lies in a dense region iff its normalized density is significantly higher than 1.

3.3 Subcluster Properties

In [1] the authors proposed 3 visual analysis criteria as requirements for subcluster visualizations. According to these criteria, the following subcluster properties should be visualized:

- subspace overlap (number of common dimensions)
- data overlap (number of common objects)
- interestingness (how much average density exeeds expected density of subspace)

It should be noted that these arbitrary criteria are not further justified and cannot be taken as general guidelines for subcluster visualizations. One major problem comes with the local densities. Because densities are only calculated where objects exist, the distribution of data over the whole data space is not taken into account. A prominent cluster may not be declared interesting. Also the distance used to define the object neighborhood does not adapt to different densities of different subspaces. However, the visual analysis criteria reflect the capabilities of the Morpheus System. Its subcluster distance function is exclusively based on dimension and data overlap:

$$\beta \left(1 - \frac{|S_i \cap S_j|}{|S_i \cup S_j|}\right) + (1 - \beta) \left(1 - \frac{|C_i \cap C_j|}{min(|C_i|, |C_j|)}\right)$$

This is basically a weighted sum of inverted dimension- and cluster overlap. β is the weighting factor. S_i and S_j are the subspaces of both clusters as sets of dimensions. C_i and C_j are both clusters as sets of objects. The distance returns a value between 0 and 1. Subclusters are positioned in screen space by applying multi dimensional scaling on their distances. It is also possible to project positions onto 3d space using the same method. This reduces the distortion of distances but possibly complicates navigation. Figure 7 gives an impression of how a 3d projection looks like.

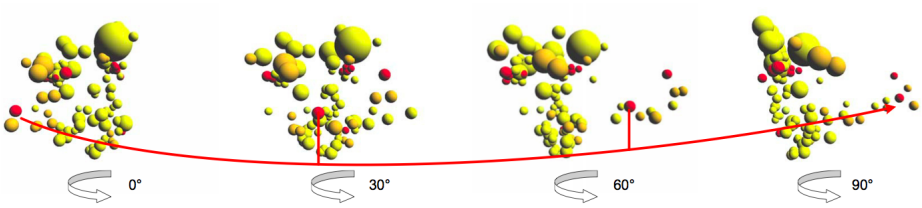

Fig. 7: *Subcluster overview projected to 3-dimensional space*

From the overview presentation a user can see how the data is clustered and identify interesting subclusters. More details about a particular subcluster are displayed by rightclicking it. The next section will describe how information about a subcluster is presented in the detail view.

3.4 Detail Presentation

To provide an overview over the actual data characteristic of a selected subcluster the Morpheus System integrates a view that shows a box plot for every dimension in the data. Figure 6 contains such a view for a 6-dimensional subcluster in a 16-dimensional data set. The dimensions of the subcluster are marked red. It can be seen that the objects are very similar in those dimensions.

A more in depth view on the dimensionality of subclusters and subcluster groups is visualized by a pixel matrix as shown in Figure 8. There is one row for every object and one column for every dimension. Although, objects can belong to several subclusters, the authors don't explain to which subcluster an object is assigned. It would make sense to choose the one of highest interestingness. The color (hue) of one cell reflects the objects value in that dimension (attribute). Saturation and brightness show the interestingness of the corresponding subcluster like in the overview presentation. Because objects can belong to several subclusters we must assume that the most interesting one is taken.

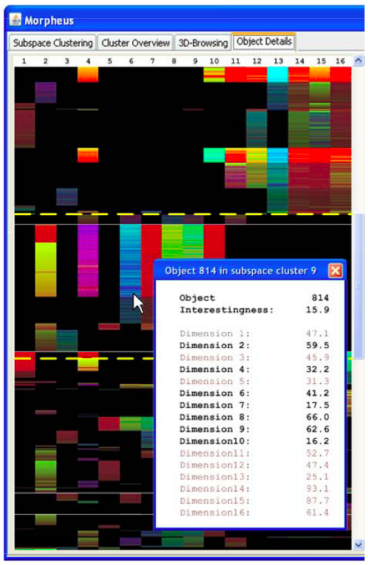

Fig. 8: *Detail view on grouped subclusters*

The main feature of this pixel matrix is that objects (rows) are grouped. A certain number of the most interesting subclusters is called anchors. Subclusters are grouped by their nearest anchors. Groups are then ordered by the interestingness of their anchors. Subclusters within a group are ordered by interestingness as well. Even objects within subclusters are ordered by interestingness, although no interestingness measure for single objects is defined in the paper. Groups of interest can be zoomed out in this view.

The advantage of this pixel matrix is that the grouping and color mapping allows the user to quickly detect the most interesting groups of similar subclusters while having an overview on their actual feature values.

Rightclicking an object reveals the interestingness of the corresponding subcluster, the exact values of each dimension and the dimensions belonging to the corresponding subcluster.

3.5 Problems

The Morpheus System is not one consistent visualization, but a software tool that involves different types of visualizations for the same data. It is not absolutely clear how these different views complement one another to support one knowledge discovery process. While the overview is suitable for high numbers of clusters and dimensions, the pixel matrix suffers from screen space limitations. Some marginal weaknesses may further be identified:

– The user has to decide on lots of parameters.

- The inner structure of single clusters is only roughly displayed in the pixel matrix view.
- The overview doesn't reflect the whole data set but only the most prominent clusters.
- MDS distorts distances and can lead to false interpretations.

4 Conclusion

Both presented methods for visualizing subclusterings are capable of giving a very general overview on the clustering structure, but are very different in what specific information is additionally presented and how this is done. The number and size of subclusters are legibly readable. The degree of spatial overlap between subclusters (block coloring vs. MDS distance) and their dimensionality (color legend vs. detail matrix) are somehow reflected in both visualizations. The general advantages of the Heidi Matrix are its compactness and how well it visualizes the dimensionality of spatial overlaps. The advantages of the Morpheus System are as follows:

- similarity of cluster subspaces as number of overlapping dimensions (MDS distance)
- dimensionality of subclusters (detail matrix)
- real subspace clustering as input (objects can belong to several subclusters)

Both presented techniques have some limits that may be general problems visualizations of subclusterings have to deal with:

- It is impossible to visualize very high numbers of subclusters, -dimensions or -objects.
- Hierarchical subclusterings are not defined or detected.
- The complexity of analyzing all subspaces is exponential in the number of dimensions.

References

1. Assent, I., Krieger, R., Müller, E., Seidl, T.: Visa: visual subspace clustering analysis. SIGKDD Explor. Newsl. 9(2), 5–12 (2007)
2. Müller, E., Assent, I., Krieger, R., Jansen, T., Seidl, T.: Morpheus: interactive exploration of subspace clustering. In: KDD '08: Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 1089–1092. ACM, New York, NY, USA (2008)
3. Vadapalli, S., Karlapalem, K.: Heidi matrix: nearest neighbor driven high dimensional data visualization. In: VAKD '09: Proceedings of the ACM SIGKDD Workshop on Visual Analytics and Knowledge Discovery. pp. 83–92. ACM, New York, NY, USA (2009)