Universiteit
Leiden
The Netherlands

Algorithmic tools for data-oriented law enforcement

Cocx, T.K.

Citation
Cocx, T. K. (2009, December 2). *Algorithmic tools for data-oriented law enforcement*. Retrieved from https://hdl.handle.net/1887/14450

Version:	Corrected Publisher's Version
License:	Licence agreement concerning inclusion of doctoral thesis in the Institutional Repository of the University of Leiden
Downloaded from:	https://hdl.handle.net/1887/14450

Note: To cite this publication please use the final published version (if applicable).

Chapter 3

Object-Centered Interactive Multi-Dimensional Scaling: Ask the Expert

Multi-Dimensional Scaling (MDS) is a widely used technique to show, relations between objects—such as humans, documents, soil samples—that are defined by a large set of features, in a low-dimensional space. Key benefit is that it enables visual inspection of object relations in an intuitive, non-technical way, very well suited for, for example, police officers without technical backgrounds. One of the limitations is that different projections exist, leading to different graphical representations and therefore different interpretations of the data. This problem is made more significant in case of noisy data or heuristic approaches to MDS. We propose Object-Centered Interactive Multi-Dimensional Scaling (OCI-MDS), a technique that allows a data expert, for example a police analyst, to try alternative positions for objects by moving them around the space in real time. The expert is helped by several types of visual feedback, such as the proportional error contribution of the expert-controlled object. Here we show that this technique has potential in a number of different domains.

3.1 Introduction

The use of computers has enabled people to create large amounts of data. This is a trend that is not restricted to a specific domain. For example, policy makers write large numbers of reports, individuals publish personal web-pages, police officers create large numbers of case data, and databases enable structured storage of large amounts of medical data. Extracting potential relations between data objects—i.e., an individual data element such as a document—is a current challenge. Many techniques have been developed for this purpose, like data clustering, graph-mining and Principal Component Analysis (PCA) [24]. In this chapter we focus on one such technique, Multi-Dimensional Scaling [13, 31].

Multi-Dimensional Scaling (MDS) is a widely used technique to show, in a low-dimensional space, relations between objects—such as human subjects, documents, soil samples—that are defined in a higher-dimensional space. If MDS is used to create a 2D visual representation of the high-dimensional dataset, a key benefit of this technique is that it enables visual inspection of object relations in an intuitive way. This is important, especially when the users of the analysis (i.e., those interpreting the final 2D projection) are not machine-learning experts. One of its limitations, however, is that different projections exist, leading to different graphical representations and therefore different interpretations of the data. This problem is especially important in case of noisy data or heuristic approaches to MDS. First, noisy (or unstructured) data introduce variation in the high-dimensional distance between objects, and as such these variations will be reflected in the 2D projection. As this noise does not convey any information regarding the relation between objects, interpretation of the 2D projection is hindered by this noise. The algorithm does not know the underlying relation between objects, and as such cannot correct for it. An expert could. Second, heuristic approaches to MDS, such as the push-pull technique [11, 20] where the projection is constructed through repeated local comparison between pairs of objects, introduce sub optimality in the 2D projection and can converge to local minima. However, heuristic approaches can have important benefits such as reduced computational cost and scalability [35] and are therefore useful for solving MDS problems.

In this chapter we propose Object-Centered Interactive Multi-Dimensional Scaling (OCI-MDS), a technique that allows a data expert to propose alternative positions for objects by moving them around the 2D space in real time. The approach is compatible with (and helps) heuristic approaches to MDS. The expert is helped by several types of visual feedback, such as the proportional error contribution of the controlled object. We use the technique in a heuristic MDS approach and show that this technique has potential in two different domains: visualization of high-dimensional computer simulation experiment configurations [6] and raw biomedical data.

Our interactive approach relates to other approaches, such as those by Stappers et al. [28]. They use interaction to enable exploration of data. Objects can be selected by the user, after which the algorithm clusters the newly selected object. Subsequently, a next object can be added (or removed). This approach is Object-Centered and allows expert-controlled visualization of object-relations, but different in the sense that objects, once positioned, are not interactively movable to (a) generate alternative hypotheses about object relations, or (b) help the MDS mechanism. Further they focus on small amounts of objects (about 10). Other approaches include non Object-Centered ones, such as those that enable experts to direct computational resources at specific parts of the space in order to reduce computational resources needed for data projection [35], and those that enable experts to interactively change algorithm parameters (like noise) and to stop the algorithm [30].

In Section 3.2 we describe some of the problems our approach addresses. In Section 3.3 we introduce Object-Centered Interactive MDS. Section 3.4 presents experimental results and contains all figures. Finally, we present our conclusion and some directions for future work in Section 3.5.

3.2 Expert Interaction Helps Heuristic Projection

A key motivation to use MDS for visualization of high-dimensional data is its ability to give an overview of a complete dataset. This is important in the exploratory phase of data analysis. For example, in the criminal investigation area, visualization of datasets supports police officials in their process of generating hypotheses about the relation between different criminal records [11]. In the computer simulation domain, such as simulation of adaptive behavior [6], scientists often repeat experiments with slightly different settings. It is important to avoid making errors in the configuration of the experiments and it is important to have a clear overview of the variations introduced in the parameters. Visualization of the relation between configurations of these experiments (not just the results) can therefore provide insight into both the completeness of a set of experiments as well as potential configuration mistakes. In the domain of biomedical data analysis, clustering, dimension reduction and visualization are used to, for example, find in a set of patients different variations of one disease, or find the most important factors underlying a certain disease.

In all those domains, MDS can be used to cluster high-dimensional data by projecting it onto a 2D space (note that data is not really clustered, as explicit groups are not made). Visualization of that space enables domain experts to get an intuitive idea of the relation between the objects in high-dimensional space. Typically, a 2D projection is constructed such that the least-square error is minimized (see Section 3.3). However, an error of 0 is usually not possible, and, if the projection technique is heuristic, minimal error cannot be guaranteed.

Another typical problem in heuristic approaches—that use incremental error minimization by inducing small changes to object locations in 2D—is that two objects that should be close to each other can be separated by a large cluster, because the large cluster pushes both objects away from each other (see Figure 2, Section 3.4). Standard incremental techniques cannot solve this problem. Thus, even though the solution is near optimal, large local errors can exist.

However, domain experts can detect such local errors by looking at the object and comparing it with its neighbors. So, from an optimality point of view the ability to move objects and clusters of objects is a useful addition to heuristic approaches to MDS. For data interpretation it is also a useful addition, as interactive real-time movement of objects enables experts to test hypotheses of relations between objects directly in the clustering result. This means that, e.g., police officials are able to test if two criminal records are related just by moving the objects close to each other and observing, e.g., the clustering result. Another advantage is the possibility to add new objects at user specified locations, and observe the trajectory of these objects in the projection as well as the influence of these objects on the projected location of other objects.

To summarize, object-based interaction with MDS is useful, provided that users get feedback information so that they can (1) select objects to move, and (2) evaluate the result of the move.

3.3 Object-Centered Interactive MDS

We propose Object-Centered Interactive MDS (OCI-MDS). This allows experts to interactively manipulate the projection result produced by a heuristic MDS algorithm. We present our algorithm and the kind of user-feedback the system gives. In the next section we show that it is a useful technique in two domains: computer simulation experiments and biomedical data analysis. Analogous to standard MDS, four steps are needed to project m-dimensional data onto a low-dimensional (2D) space. The first two are preparatory and the second two are iterative until some stop-criterion (usually reaching a minimal error, or stalled improvement for some fixed number of iterations).

First, a distance matrix is constructed that captures the distances between n individual objects in m-dimensional space, where m typically is the number of features used to describe an object. If objects do not have the same number of features (i.e., objects in one set have different dimensionality) then the distance matrix must be able to cope with this. We assume we have such an $n \times n$ distance matrix D.

Second, objects are randomly placed in a low-dimensional space, in our case a 2D space. A vector O of size n represents the coordinates of all n objects.

Third, the first iterative step selects (e.g., randomly) an object i and adds random noise to the coordinates of that object. Noise is added as follows:

$$O_i[x] \leftarrow O_i[x] + \text{rnd}() \cdot step$$

$$O_i[y] \leftarrow O_i[y] + \text{rnd}() \cdot step$$

where $O_i[x]$ and $O_i[y]$ are the coordinates of an individual object i, and rnd() a function giving a random number in $[-0.5, 0.5]$. The variable *step* is a noise size factor that is local-error, total 2D space span, and annealing dependent:

$$step = \alpha \cdot d \, \frac{n \cdot e_i}{e}$$

where d is the largest distance between objects in O, and thus equivalent to $\max(D^L)$ (see below), e_i the local error associated with object i, e the global error (see below), and α an exponentially decreasing annealing factor. The local error e_i is defined by:

$$e_i = \sum_{j=1}^{n} (D_{ij} - D^L_{ij})^2$$

where D^L is the distance matrix of objects in 2D space. The motivation for our step factor is as follows. It expresses a normalized maximum step that depends on the error contribution of an object to the global error and the size of the space covered by all objects. This has the following two benefits. First, an object with a high local error is moved through the 2D space with high speed, in order to find a better location for it. This increases the probability that wrongly placed objects eventually find a suitable location with small local error. Second, if the 2D space that is occupied by the objects is large, the objects will also move quicker. This ensures that the algorithm is not dependent on the absolute distance between objects. Further, we decrease the annealing factor α exponentially

whenever for all objects i there has been no decrease in e. So, if the algorithm approaches a minimum, smaller steps can be used to better approach that minimum.

Fourth, update the distance matrix D^L (note that we use Euclidean distances for D^L). Then evaluate the least-square error (LSE) between D and D^L:

$$e = \sum_{i=1}^{n} \sum_{j=1}^{n} (D_{ij} - D^L_{ij})^2$$

If the local noise added to object i decreases global error e, keep the new coordinates; if not, discard the change. Repeat step three and four until e is smaller than a threshold t, or until e has not decreased for a fixed number of steps s. If this criterion is met, the process is paused until the user interactively changes positions of objects.

Objects are drawn in a two-dimensional plane (e.g., Figure 3.1(a), Section 3.4). The user is able to, at any time, grab objects and place them at alternative positions. This enables the user to (1) help the heuristic MDS, and (2) experiment with potential clusters. The user is given two types of direct feedback. First, when objects are released, the projection mechanism restarts iteration of step three and four, so the user can directly observe the effect of the moved objects on the total distribution of objects. Second, objects are drawn in a color that represents the local error contribution of the object. This is non-trivial, as color changes need to be reactive enough to reflect small changes in local error but also reflect global changes in global error e. We used the following formula:

$$color_i = \log\left(1 + \frac{n \cdot e_i}{\log(1+n) \cdot e_{min}}\right)$$

where n is the number of objects, e_i the local error of object i and $e_{min} = \min(e_s)$ for $s = 0, 1, 2, \ldots$, where e_s is the global error at iteration s. The variable $color_i$ can be used to define, for example, drawing intensity or color of an object i. This color scheme was used to experiment with interactive visualization of simulation experiment configurations, as well as interactive visualization of biomedical data.

We also experimented with a different coloring scheme where objects are colored using a per object measure that is relevant to the dataset, not the algorithm. This scheme was used to experiment with biomedical data visualization. The data consisted of about 400 objects with 10 features. The objects are patients. Every feature represents the severity of a disease in a different part of the body. The color represents the total severity calculated by averaging over the different features. This average is meaningful but somewhat arbitrary, as it is not clear that the average is actually a good representation of the global severity of the disease. However, as our focus is the visualization technique, not the dataset, we do not consider this to be a problem at this moment. For the same reason we do not specify the datasets in detail in this chapter.

3.4 Experimental Results

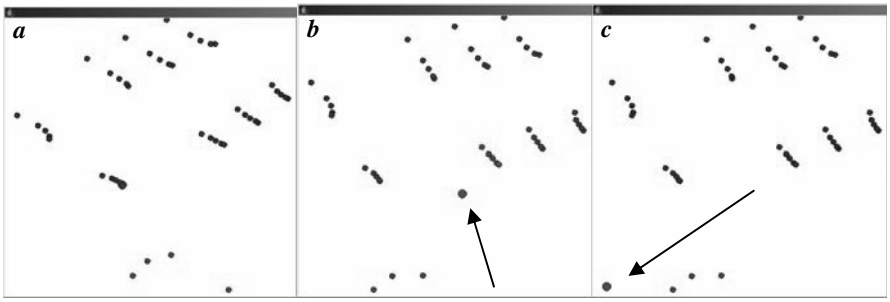
We have developed a Java application that allows us to test our approach. First we present results that investigated its use in visualizing simulation experiment configurations. The

dataset consisted of 44 experimental configurations, all of which are used in research into the influence of emotion on learning using reinforcement learning [6]. The features of the objects consisted of 40 different learning parameter settings such as learning rate, exploration-exploitation settings, etc. This can be considered structured data. We have a vector representation of these configuration documents, and defined a distance measure based on the parameter type (boolean, string, double) and the range of values found for one parameter. We do not detail the distance measure here. Based on the measure we constructed the distance matrix D.

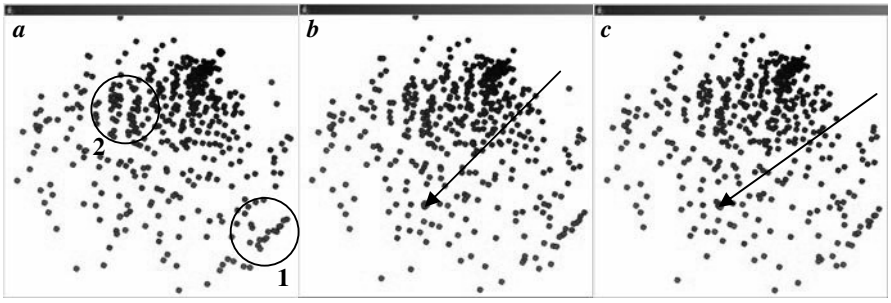
Figure 3.1(a)a shows an initial 2D projection of a set of experiment configurations. The visualization clearly shows that there are 4 experiments that are special (bottom), and several groups of other experiments. The objects at the bottom are control experiments, and are indeed the control experiments with which the others are compared. The control experiment at the right is further away from the other three (and further away from all other experiments). Did our algorithm correctly place it here? The user can grab (Figure 3.1(a)b) the object, and while moving it, the local errors start to increase (objects color red). Apparently, the object should not be moved in that direction. After letting the object go, the algorithm projects the object back to a similar (but not necessarily equal) position. The object indeed belongs there. Other object clusters show a nice regular pattern, as a result of the distance function. The four top clusters (Figure 3.1(a)c) all belong to one typical parameter value, while the middle four all belong to a different value on that same parameter. The clusters themselves are organized and correspond well to the actual configurations. This enabled us to verify that no configuration errors had been made in the course of these experiments.

Second, we experimented with the biomedical data mentioned earlier. The projection resulted in a clustering that showed a trend from high severity to low severity, even though global severity is not a feature in the data (Figure 3.1(b)a). Although the projection clearly does not give as much insight into the data as the projection of the structured experiment data shown before, several clusters appear to exist. For example, cluster 1 represents a coupling of two severely affected body parts. Cluster 2 represents a coupling of two other severely affected body parts where the two parts of cluster 1 are not affected. This might indicate correlation between certain body parts and dissociation between others. Although the heuristic technique extracts some useful information from unstructured raw biomedical data, as it stands, the technique is clearly not usable for quantitative conclusions about the dataset, but only for explorative data analysis. However, the main topic of this chapter is dataset exploration and interactivity. Interaction enabled us to relocate two items that appeared to be badly clustered due to the separation problem mentioned earlier, i.e., a cluster divides otherwise closely related objects (Figure 3.1(b)b and c). After grabbing and relocating the two severe disease cases to an arbitrary position on the other side of the cluster consisting of non-severe disease objects, they were relocated by the algorithm at positions that better matched their real-world relation, as could be observed from a comparison with the objects near to that new location. Finally, Figure 3 shows the variation in the distribution of local errors. Figure 3.1(c)a also shows one object ('x') with high local error positioned in between objects with small local errors. When grabbing and repositioning the object at a location in which it appears to have smaller

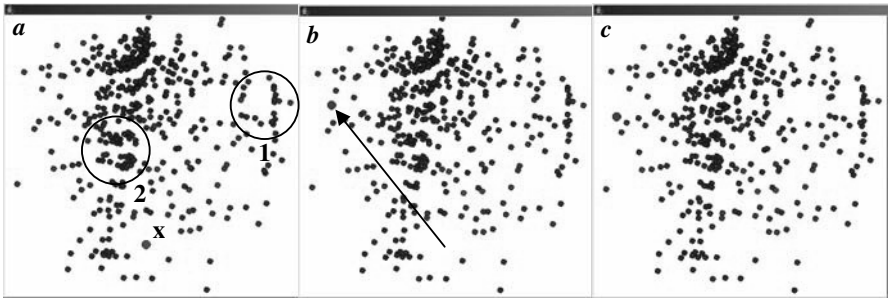
local error, we were able to relocate it at a better place. Although the exact meaning of the projection is at this point unclear (and strongly dependent on the distance measure we used), our experiment shows that Object-Centered interactivity is a useful method to explore object relations.



(a) Manipulating experiment configuration clusters (local error color)



(b) Manipulating biomedical data (severity color)



(c) Manipulating biomedical data (local error color)

Figure 3.1: Experimental Results

3.5 Conclusion and Future Directions

Our experiments show that Object-Centered Interactive MDS has potential. It can be used for direct manipulation of clustering results based on a heuristic MDS approximation. It can help in verifying the MDS result, and help to generate hypotheses about alternative object relations, that were not found, for example, because the MDS converged to a local optimum. However, currently its usefulness is somewhat limited on highly unstructured data.

Future work includes adding multiple-object drag-and-drop, extensive user testing, and scaling mechanisms like those introduced by Williams and Munzner [35]. Also, relating the re-placement of single items with a high local error (a high contribution to the global error) to the change in the global error is important for the analysis of the proposed approach. Changes in the global error can be represented by a Boolean figure (higher or lower) or be represented by a (color-)scale during the process of human intervention. Future research can strive to find a relation between the decrease in local errors and the error made in the total image. If such a positive relation exists, automating the process of relocating those items with the highest local error can be an asset worth pursuing. This global optimization can be of interest in areas where the correctness of the image as a whole is more important than the relation between a small subset of individual items, like, for example, a clustering on customer behavior.

Our main goal is to use the proposed tool as part of a larger data mining framework for law enforcement purposes as described in Chapters 4 and 5. These chapters focus on the contributions a domain expert can make to the process of analyzing criminal careers using our interactive clustering tool.