CLUSTERED MULTIDIMENSIONAL SCALING FOR EXPLORATION IN INFORMATION RETRIEVAL

Enikő Székely, Éric Bruno, Stéphane Marchand-Maillet
Viper Group, Computer Vision and Multimedia Laboratory, University of Geneva
7 Route de Drize, Geneva, Switzerland
{*eniko.szekely, eric.bruno, marchand*}*@cui.unige.ch*

Keywords: clustering, nearest neighbour, multidimensional scaling, exploration.

Abstract: The data that needs to be processed nowadays is frequently represented in high-dimensional spaces with the dimension given by the number of features selected. There is a gap between human perception of low-dimensional spaces and the behaviour of distances within high-dimensional spaces. In data analysis the phenomenon of "curse of dimensionality" has consequences on the (dis)similarity matrices because the points become equidistant. In such a situation, methods for dimensionality reduction fail to reveal in the low-dimensional projected space structures existing in the data. We therefore propose in this article a clustered multidimensional scaling method for the discovery and understanding of data structures in view of exploration. Firstly, the data is clustered in the original space based on the closest k neighbours of each point which results in a disconnected graph. Secondly, an MDS is performed on each of the graph components. And finally the clusters' representatives are projected in the reduced space by means of an MDS in order to preserve the distances between clusters from the original space.

1 Introduction

In recent years, a lot of attention has been dedicated to the "curse of dimensionality" due to its implications in many fields that process high-dimensional data. In the field of Information Retrieval the data processed (image, text, video) is represented in hundreds or thousands of dimensions: an image may be represented by its pixels (of the order of hundreds or thousands) or by its colours (of the order of hundreds, for example in the 256 RGB colour space), a document can be represented by the different terms that appear in the document (of the order of hundreds or thousands). Processing such data is under the influence of the "curse of dimensionality" effect, an effect of high-dimensional spaces that poses problems to all machine learning algorithms for clustering, discrimination, dimensionality reduction. The term was first introduced by Bellman in 1961. It refers to the fact that in a high-dimensional space the sampling of the data grows exponentially with the number of variables (dimensions). In such spaces the data is very sparse and the points become equidistant. These two behaviours lead to a degradation of the results of the algorithms in the field of data-mining.

Exploration in Information Retrieval is another field that has been given a lot of attention in the last years. Exploration is one of the last steps of the Information Retrieval process and is meant to help the user get acquainted with the content of the collection that he/she is exploring either for browsing or for query retrieval. On one hand the data is more and more represented in high-dimensional spaces and on the other hand this data has to be understood by humans, who can visualize it in hardly more than 2 or 3 dimensions. We propose in this article a clustered multidimensional scaling method for data preparation meant to diminish the gap existing between the behaviour of distances within high-dimensional spaces and human perception of low-dimensional

spaces. The purpose of our method is to enable guiding the user through the prepared data space and even hardly improve the exploration process of datasets assumed to be composed of multiple distinct classes.

In our work we are exploring the fundamentals of distances' behaviours in high-dimensional spaces in order to further develop algorithms that will improve the quality of clustering and dimensionality reduction techniques which suffer enormously because of the "curse of dimensionality". Therefore we propose here a dimensionality reduction method for cluster detection optimization meant to avoid the undesired effects of high-dimensional spaces. The paper is organized as follows: Section 2 is dedicated to an overview of classical methods for dimensionality reduction based namely on distance matrices (Multidimensional Scaling, Sammon Mapping, Locally Linear Embedding, Isomap, Curvilinear Component Analysis, Curvilinear Distance Analysis, Distributional Scaling and Relational Perspective Map). Section 3 describes our approach for processing high-dimensional data using a clustered multidimensional scaling method. Section 4 presents the database used and the experimental results of our method. The paper ends with some discussions on our method and conclusions in sections 5 and 6.

2 Methods for dimensionality reduction

Methods for dimensionality reduction are employed each time high-dimensional data has to be reduced from a high to a low-dimensional space. The principle of the mapping process for methods based on distance matrices is to find the configuration of points that best preserves the original inter-distances.

The most widely used dimensionality reduction method is Multidimensional Scaling (MDS) - a method for projecting the data from a high-dimensional into a low-dimensional space (Borg and Groenen, 2005). The attempt of MDS is to model the proximities from the original space as geometrical distances in the projected (reduced) space. The proximities in the original space may be any (dis)similarities: distances, correlations, co-occurences, etc. and the dissimilarities in the reduced space are geometrical distances, the mostly used being the Euclidean distance.

The first MDS algorithm is Classical Scaling due to Torgerson (1952) and Gower (1966) where the coordinates of the elements in the reduced space are found by an eigendecomposition of the Gram scalar product matrix derived from the dissimilarity matrix. Once the eigendecomposition is done, the coordinates are computed based on the first k eigenvalues (the equivalent of Principal Component Analysis for distance matrices). Classical Scaling is used because it gives an analytical solution and makes no iterations.

However, Classical Scaling is less used than interative MDS methods in which point coordinates in the reduced space are found by minimizing a particular nonlinear stress (error) function (maximizing the goodness-of-fit). The basic stress function is given by:

$$E = \sum_{i<j} (d_{ij} - \delta_{ij})^2$$

where δ_{ij} are the dissimilarities (distances) in the original space and d_{ij} are the geometrical distances in the projected space. A good way of chosing the best value for the dimension of the reduced space is to plot the Shepard diagram (de Leeuw, 2004) for different values of the dimension and to choose the one that best preserves the ratio between the original and the reproduced distances.

(Sammon, 1969) introduces one of the most widely used nonlinear MDS algorithms whose purpose is to emphasize the preservation of small distances. This is done by minimizing the error function:

$$E = \sum_{i<j} w_{ij}(d_{ij} - \delta_{ij})^2$$

where:

$$w_{ij}^{(g)} = \frac{1}{\sum_{k=1}^{N} \sum_{l>k} \delta_{kl}^2}$$

$$w_{ij}^{(l)} = \frac{2}{N(N-1)\delta_{ij}^2}$$

The terms w_{ij} are introduced to weight the differences between the dissimilarities in the original space and the distances in the reduced space. Either all differences are weighted equally by a constant weight factor $w_{ij}^{(g)}$, or local structure is to be preserved using a variable weighting factor $w_{ij}^{(l)}$ that gives higher importance to smaller distances. The method used for the minimization of the error is a steepest descent procedure (gradient descent).

Classical Scaling - like PCA - reveals linear structure in data by eigendecomposition solutions. PCA preserves the highest variances and classical MDS preserves the proximities. The nonlinear MDS algorithms, when they use Euclidean distance in the original space - no matter which is the error function that they try to minimize -, give good results for structures lying on linear Euclidean subspaces but fail when trying to reduce the space for structures lying on nonlinear Euclidean subspaces (nonlinear data is data that cannot be mapped onto a reduced space by a linear transformation). The difficulty of mapping nonlinear data with methods using Euclidean distance is that two points that are located far away on a nonlinear Euclidean low-dimensional manifold can still be located very close according to the Euclidean distance (eg. the well-known "Swiss Roll" configuration). A manifold is a space which locally resembles a Euclidean space (the neighbours of each point are located in a Euclidean space) but globally has a more complicated structure.

Two methods effective in detecting nonlinear Euclidean structures in data are Isomap and Locally Linear Embedding (LLE). They both attempt to find the low-dimensional manifold on which the data lies by avoiding to compute the Euclidean distances between all points. Isomap is a global method that replaces the Euclidean distances for far-away points by the geodesic distance and LLE is a local method that takes into account only the local structure (only the neighbours of each point). Isomap (Isometric Mapping), presented in (Tennenbaum et al., 2000), is a generalization of the MDS using the geodesic distance (a "geodesic" is a straight line in a curved space, measuring the shortest path between points in this curved space). The Isomap algorithm in the first step constructs the neighbourhood graph to unfold the structures, in the second step computes the geodesic distance between all points by following the edges built in the first step. And the last step consists in constructing the low-dimensional space using MDS, but with the geodesic distances, instead of the Euclidean distances.

LLE (Roweis and Saul, 2000) is based on the assumption that a point together with its first k neighbours lie on a locally linear patch of the manifold, even for nonlinear manifolds. Instead of computing the pairwise distances between points and trying to preserve them in the reduced space (like MDS and Isomap do), LLE tries to preserve only the linear reconstruction of a point from its neighbours.

$$E(W) = \sum_{i=1}^{n} \left| x_i - \sum_{j=1}^{k} w_j^{(i)} x_{N(j)}^i \right|^2$$

where $x_{N(j)}^i$ are the first k neighbours of point x_i weighted by the their contribution to the point x_i.

Both Isomap and LLE have analytical solutions and thus avoid local minima. We present now two other methods, Curvilinear Component Analysis (CCA) (Demartines and Hérault, 1997) and Curvilinear Distance Analysis (CDA) (J.A.Lee et al., 2000) which are generalizations of MDS and outperform it. Still these two last methods use gradient methods to find a solution and thus they can get stuck in local minima - in contrast with LLE and Isomap which avoid local minima.

CCA considers that it is difficult to match distances at all scales and so it introduces a weighting function $F(d_{ij}^l)$ and then tries to minimize the following error

$$E = \sum_{i<j} (d_{ij}^l - d_{ij}^L)^2 F(d_{ij}^l)$$

The differences from the other dimensionality reduction methods are the weighting function $F(d_{ij}^l)$ and the fact that the weighting is done in the *output* (reduced) space and not in the *input* (original) space. Choosing a monotonically and unbounded function like

$$F = \frac{1}{d_{ij}^l}$$

gives too much emphasis to small distances. This is the reason for which CCA selects for the weighting function a simple step function:

$$F = \begin{cases} 1, & \text{if } d^l_{ij} \leq \lambda; \\ 0, & \text{if } d^l_{ij} > \lambda. \end{cases}$$

which introduces the local linearity principle. CCA, like LLE, preserves rather local geometry.

A small change in CCA resulted in Curvilinear Distance Analysis. The difference consists in using instead of the Euclidean distance, the curvilinear (geodesic distance). The distance is computed along the structure and not through the object, like the Euclidean distance.

$$E = \sum_{i<j} (d^l_{ij} - \delta^L_{ij})^2 F(d^l_{ij})$$

In (Quist and G.Yona, 2004) the authors introduce the Distributional Scaling, an embedding method aimed at maintaining not only the individual dissimilarities but also the distribution of dissimilarities. They also present a hierarchical MDS method for embedding clustered data. The idea behind this latter method is that when the data is clustered it is normal to treat the dissimilarities between clusters differently than those within a cluster.

In (Li, 2004) the Relational Perspective Map (RPM) method is introduced which combines the traditional multidimensional scaling method for proximity preserving with topology constraints. The achievement from RPM is that it divides the data into multiple partitions and embeds them on the 2D space without overlapping.

3 Clustered Multidimensional Scaling

Multidimensional scaling is used for different purposes among which exploratory data analysis. As mentioned in section 1, exploration in the field of Information Retrieval, either for browsing or query retrieval, is a direct application of the mapping process of the data into a low-dimensional space. When data (image, text, video) is represented only by some measures of similarity/dissimilarity and it is not generated by a known model it is useful to be able to explore the data's structure. Multidimensional scaling can be used in this purpose to help the user inspect the structure of the data. We start this section by presenting the difficulties that appear when exploring high-dimensional data since the distances between all elements tend to be equal and we further present an algorithm that is meant to reveal clusters in data in order to improve the exploration.

We have reviewed in the previous section a few methods for mapping the data from the high-dimensional original space into a low-dimensional space. Dimensionality reduction methods are used to make the embedding between the original and the reduced space but difficulties appear due to the gap existing between the behaviour of distances in high-dimensional spaces and human perception of low-dimensional spaces. To ilustrate this gap we give here two typical examples. The first example is that of points uniformly sampled from within a hyper-sphere where, as we increase the dimension, the data is more and more distributed near the surface. A second example is that of points uniformly sampled from within a cube where, as we increase the dimension, the data is more and more distributed in the corners (for a 7-dimensional cube, 96% of the mass is concentrated in the $128 (= 2^7)$ "corners").

The phenomenon that is responsible for the "curse of dimensionality" is the "empty space phenomenon": in high-dimensional spaces the data is very sparse, which requires an exponentially growing sample set for data analysis. One problem that appears in high-dimensional spaces is that the ratio between the distance to the nearest and to the farthest neighbour goes to 1. Hence the distance histogram goes towards a more and more concentrated peak as illustrated in figure 1 [see section 4 for details on the dataset and the distances used in the examples]. This, together with the (wrong) choice of the distance measure, has consequences on the quality of the MDS results (figure 2 - Stress MDS or figure 3 - Sammon mapping).

In an unsupervised case (supposing we don't know *a priori* the labels/classes), the result of such an MDS (one compact cluster) would be visually almost useless to the user. We want to guide the user during the exploration of the collection and facilitate the dis-

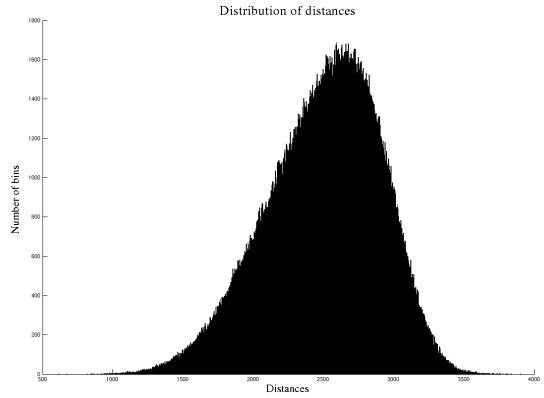

Figure 1: Histogram of Euclidean distances computed using 1000 random digits from the MNIST dataset.

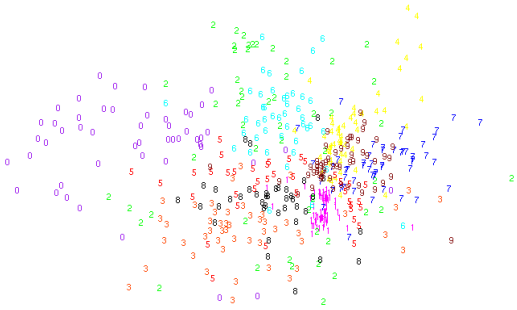

Figure 2: MDS Stress mapping done on 500 digits.

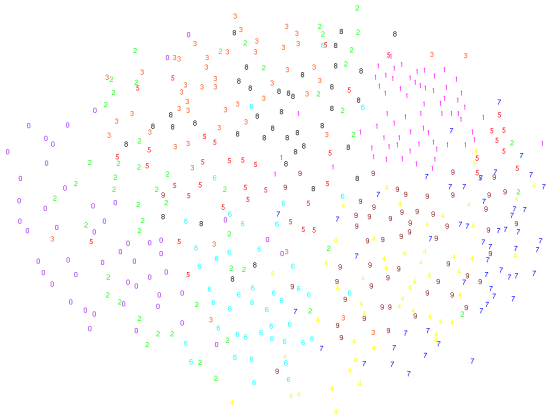

Figure 3: Sammon mapping on the same 500 digits.

covery and understanding of the class structure. In (Marchand-Maillet and Bruno, 2005) we proposed Collection Guiding as a new principle for the management of large multimedia collections. The context is that of a user which is faced with a large collection of data. The objective of the Collection Guiding is to help the user easily inspect the collection. As shown in figures 2 and 3 the inspection of such a collection without any *a priori* knowledge about its content is difficult. This leads us to propose here a new hierarchical approach based firstly on a clustering of the data in the original space and a dimension reduction applied to each cluster.

The clustering is done using the closest neighbours in the original space inspite the scepticism existing in the literature on the quality of the clustering results in high-dimensional spaces. Our assumption is that even if the distances between points tend to be equal, the rank "order" of distances is still preserved in high dimensions. Of course this order is also dependent on the similarity (distance) measure used and it can fail when using real data (handwritten "1" and "7" can be very similar when Euclidean distance is computed on their pixel values, figure 4).

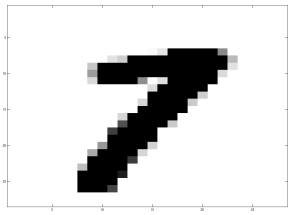

Figure 4: A '1' or a '7'?

In the following we will present the three steps of our approach:

1. cluster the data in the original space,
2. apply MDS on each cluster,
3. organise the data in the 2D space around the cluster representatives projected using an MDS.

Step 1. Clustering criteria

The distances (here Euclidean distances) between elements follow a Gaussian distribution (figure 1) with a more and more concentrated peak as the dimensionality increases. This phenomenon was first observed by Paul Lévy in the 1930s' and put forward in the 1970s' by Vitali Milman: the "concentration of measure phenomenon" (also known as the Law of Large Numbers) considered one of the blessings of dimensionality (Donoho, 2000): in high dimensions, almost all points are very

close to every subset containing at least half of the points. Our clustering method is based on the nearest neighbours taking advantage on the smallest distances and leaving apart the rest of them (figure 1). The concentration of distances around one value due to the "concentration of measure phenomenon" may influence the behaviour of the nearest neighbour which is investigated in (Beyer et al., 1999) where the authors identify cases when kNN is meaningful or not. Given the distances d^i_{min} and d^i_{max} in a L-high-dimensional space to the closest and farthest neighbour from a point i the *ratio contrast* $\frac{d^i_{min}}{d^i_{max}}$ converges to 1 as $L \to \infty$:

$$\lim_{L\to\infty} \frac{d^i_{min}}{d^i_{max}} = 1$$

Thus more the dimensionality increases, more the variance of the distance distribution (which approaches a Gaussian one) scaled by the overall magnitude of the distances converges to 0 and more the *ratio contrast* converges to 1: the distances distribution $F_{dist} \sim N(\mu, \sigma^2)$ with $\sigma \to 0$.

The clustering step is similar to the first step of Isomap: construct the neighbourhood graph. Isomap computes the first k neighbours of each point, builds the graph based on these neighbours and then computes the geodesic distance between the elements that are not directly connected in the graph. Isomap's approach is the construction of a one connected graph such that all geodesic distances can be computed (there are no individual components in the graph, which leads to a "one cluster" MDS). On the contrary, our approach is data division into clusters to avoid the "one cluster" analysis. When building the graph of the first k neighbours the choice of k is critical because it will influence the number of clusters. We favorise values of k which lead to many small "clean" over few "noisy" clusters. The motivation is to distinguish coherent data structures and process them in isolation (ie. without suffering from the influence of external data when processing data belonging to one class).

Nearest neighbours clustering.
Let $X = \{x_i, i = 1, ..., n\}$ be a set of n data points $x_i = \{x_{i1}, ..., x_{iL}\}$ in the L high-dimensional space. Let $\delta_{ij} = dist(x_i, x_j)$ be the distance between points x_i and x_j.

1. Compute the vertex degree matrix $V = \{v_{ij}, i = 1, ..., n, j = 1, ..., n\}$ with

$$v_{ij} = \begin{cases} 1, & \text{if } x_j \text{ is among the first } k \text{ nearest} \\ & \text{neighbours of point } x_i; \\ 0, & \text{otherwise.} \end{cases}$$

2. Decompose V in independent components c_p with each c_p corresponding to one cluster. Let $C = \{c_p, p = 1, ..., m\}$ be the set of clusters.

The choice of this clustering method is preferred over k-means or Expectation Maximization algorithms because it makes no assumptions on the real distribution of the data and the number of clusters does not have to be known or guessed in advance: the vertex degree matrix V displays a structure made of multiple separate independent components and m - the number of clusters - is given by the number of independent components in V.

Step 2. Multidimensional scaling for each cluster

Our clusters correspond to the individual components in our graph. These clusters are initially represented in the original high-dimensional space. We then project separately each of the clusters with an MDS. The MDS method used can be a Classical Scaling, Sammon mapping, even Isomap, CCA or CDA. What is important here is that we assume to process data that shows coherent homogeneous structure (since belonging to the same class).

Dimensionality reduction.
Let $X_{cp} = \{x^p_i \mid x^p_i \in c_p\}$ be the set of points that belong to cluster c_p. Embed each X_{cp} from the L high-dimensional space (δ_{ij}) to a lower dimensional space with dimension l (d_{ij}).

$$E = \sum_{i<j} (d_{ij} - \delta_{ij})^2 \qquad i, j = 1, ..., \mid X_{cp} \mid$$

Step 3. Data exploration

The last step is the grouping of all clusters within the same space. We perform an MDS on the cluster centers in order to preserve the original arrangements between elements in the reduced space. The cluster center is chosen among the

points that have the highest number of links to the other elements in the cluster (the highest vertex degree) and is, by construction, the closest to the gravity center of the cluster. Once the centers are projected in the reduced space, the data from step 2 is gathered around its centers without overlap between clusters. The trade-off is between the cleaness and the number of clusters. For an easy inspection we prefer cleaness.

Clusters' centers.
For each $X_{cp} = \{x_i^p \mid x_i^p \in c_p\}$ *let* x_t^p *be the center of cluster* c_p *if*

$$\sum_{k=1}^{|X_{cp}|} v_{tk} = max\left(\sum_{k=1}^{|X_{cp}|} v_{ik}, i = 1, ..., \mid X_{cp} \mid\right)$$

and

$dist(x_t^p, g_p) < dist(x_i^p, g_p), \forall i = 1, ..., \mid X_{cp} \mid, i \neq t$

where g_p *is the gravity center of cluster* c_p.

4 Experiments

We use the MNIST digit dataset for our experiments. The dataset has a training set of 60,000 samples and a testing set of 10,000 samples all being labeled. All digits are normalized to fit in a $20 * 20$ pixel box and centered in a $28 * 28$ image by computing the center of the mass of the points. This allows to use as distance measure between two digits the Euclidean distance over the 784 dimensions ($28 * 28$ pixel values p_i).

$$d_{ij} = \sqrt{\sum_{k=1}^{784} (p_i^k - p_j^k)^2}$$

After the clustering step the ideal case would be to have exactly 10 classes containing each all handwritten digits corresponding to one label digit (eg. all 1s in a cluster). The results depend, as mentioned in the previous section, on the value of k, the number of neighbours taken to build the graph and on the distance measure. The experimental results of our clustering approach reveal the possibility of having 20 clusters, with multiple clusters for each digit. However the advantage is that the clusters are clean (as we could see from our experiments, especially for small values of k) and during the inspection process the user will get a clear idea of the dataset content. In the next figures (5, 6, 7, 8) we illustrate the results of our method for 100 digits taken randomly from the database using the first nearest neighbour. In figure 7 only intra-cluster distances are considered when plotting the data. Inter-cluster distances could be further used to better position the clusters with respect to the other clusters, which would employ some rotations applied to the clusters in figure 6.

5 Discussion

The clustered multidimensional scaling introduced in this paper is intended to improve the exploration when the results of the dimensionality reduction methods are ambiguous for the user.

In this study we work with essentially two ideas in mind:

1. We should make as little as possible use of the distances within high-dimensional spaces. Hence we use the ranking rather than the actual distance values.
2. Assuming the data contains classes, we should target working within each class in isolation. Hence clustering is desirable. However, traditional clustering techniques heavily rely on the notion of distance. We simplify the clustering so as to trade between cluster quality and information preservation.

From the advantages of the clustered multidimensional scaling we state here three of them: 1) MDS convergence is better guaranteed by breaking the data into clusters and performing the MDS on each cluster; 2) a better quality of distances preservation within one cluster is guaranteed by smaller matrices processing and 3) division of data into smaller matrices represents a solution to the burden of analyzing a huge proximity matrix.

Considering the proximity matrix as the input to the algorithm, the clustering configuration depends only on one control parameter: the number of nearest neighbours needed to build the graph, an advantage but also the disadvantage of our method since the choice of the number of neighbours remains sensible and influences the clustering.

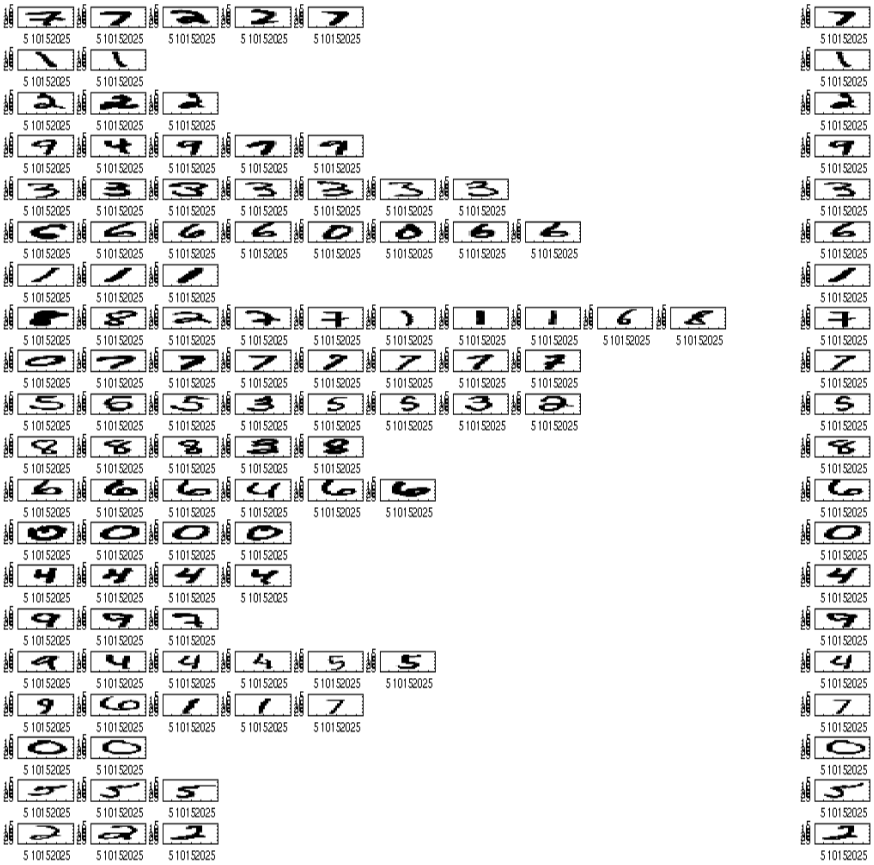

Figure 5: The results of the clustering step and on the right the centers chosen by the algorithm.

The initial results obtained with the clustered multidimensional scaling, as presented in this paper, are encouraging as they show that the distribution of points in the reduced space can be better organised in view of exploration using a clustered dimensionality reduction method. The evaluation of this method involves two aspects: objective quantitative evaluation and human-centered evaluation. Using the nearest neighbours to perform the clustering, some quantitative evaluation is already incorporated in the algorithm itself. However, further quantitative evaluation can be performed to test to what extent "invariants" (structures) present in the high-dimensional space are preserved in the low-dimensional space too. The human-centered evaluation may consist in a comparison between the reactions (time, effort with respect to some predefined tasks) of the user to our method as opposed to the baseline (the global multidimensional scaling).

6 Conclusion

In this article we have presented a new approach for the preparation of large collections of class-based data in view of exploration. Data is clustered in the original space and all clusters are then independently projected in a reduced space by dimensionality reduction methods. In order to maintain the initial proximities between clusters, an MDS is performed on the cluster centers which will allow a guided exploration of the collection of data.

ACKNOWLEDGEMENTS

This work has been partly funded by SNF fund No. 200021-109377 in parallel with the Swiss NCCR(IM)2.

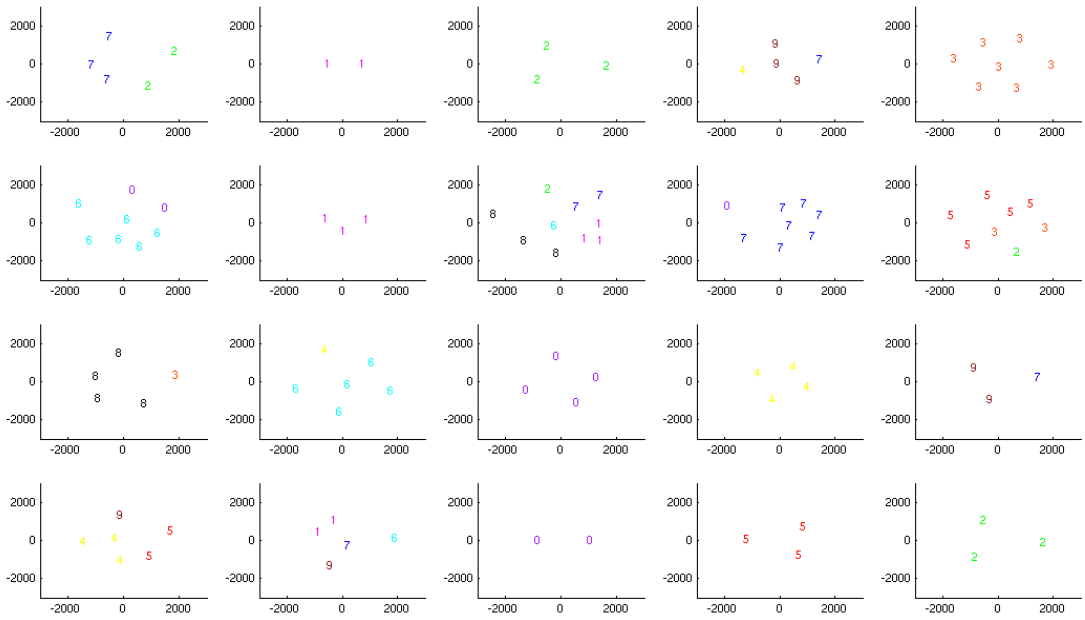

Figure 6: The results of the second step, the individual MDS performed on each cluster (note that the arrangement in each cluster is defined up to a rotation).

REFERENCES

Beyer, K., Goldstein, J., Ramakrishnan, R., and Shaft, U. (1999). When is "nearest neighbor" meaningful? In *Proceeding of the 7th International Conference on Database Theory, Springer*, volume 1540, pages 217–235.

Borg, I. and Groenen, P. (2005). Modern multidimensional scaling: Theory and applications. *Springer*.

de Leeuw, J. (2004). Shepard diagram. eScholarship Repository, University of California.

Demartines, P. and Hérault, J. (1997). Curvilinear component analysis: A self-organizig neural network for nonlinear mapping of data sets. In *IEEE Transactions on Neural Network*, volume 8.

Donoho, D. (2000). High-dimensional data analysis: The curses and blessings of dimensionality. American Mathematical Society Conference "Math Challenges of the 21st century", Los Angeles.

J.A.Lee, A.Lendasse, and Verleysen, M. (2000). A robust nonlinear projection method. In *Proceddings of ESANN'2000, Belgium*, pages 13–20.

Li, J. X. (2004). Visualization of high-dimensional data with relational perspective map. *Information Visualization*, 3:49–59.

Marchand-Maillet, S. and Bruno, E. (2005). Collection guiding: a new framework for handling large multimedia collections. In *First Workshop on Audio-visual Content And Information Visualization In Digital Libraries, AVIVDiLib05,Cortona,Italy*.

Quist, M. and G.Yona (2004). Distributional scaling: An algorithm for structure-preserving embedding of metric and nonmetric spaces. *Journal of Machine Learning Research*, 5:399–420.

Roweis, S. and Saul, L. (2000). Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290:2323–2326.

Sammon, J. W. (1969). A nonlinear mapping for data structure analysis. In *IEEE Transactions on Computers*, volume C-18.

Tennenbaum, J. B., de Silva, V., and Langford, J. C. (2000). A global geometric framework for nonlinear dimensionality reduction. *Science*, 290:2319–2323.

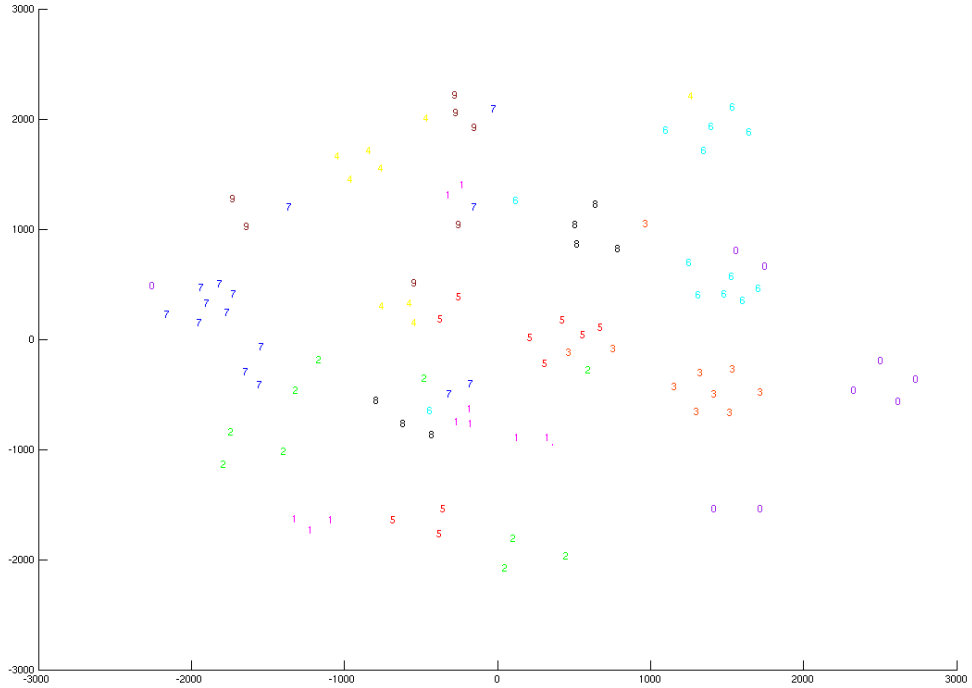

Figure 7: Cluster superposition on the centers projected with MDS in the third step (see text for a better arrangement of the data).

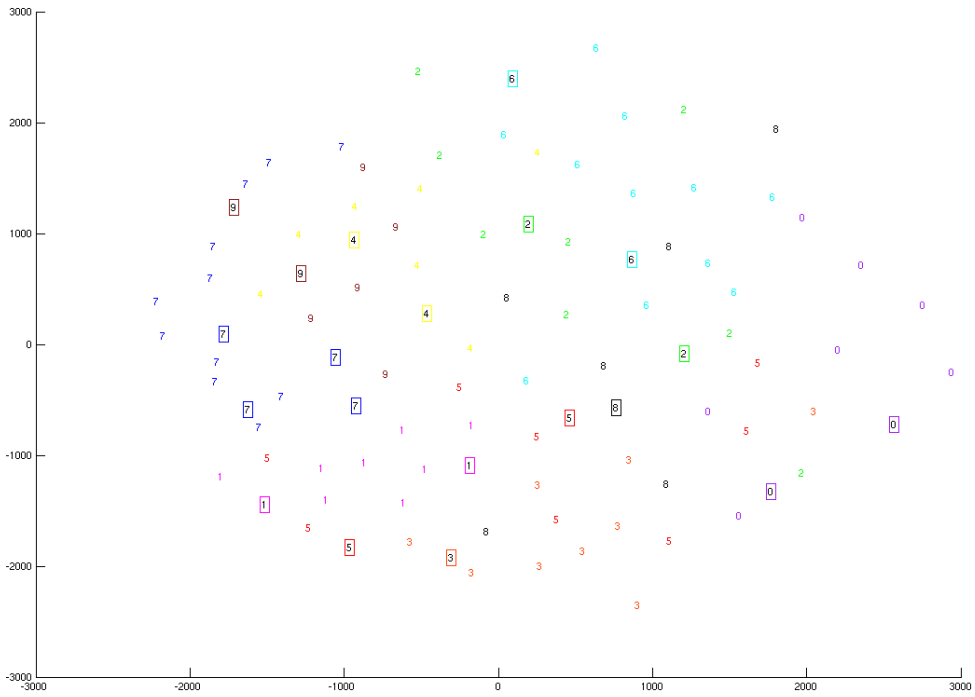

Figure 8: Emphasis on the centers in the initial MDS.