HDDVis: An Interactive Tool for High Dimensional Data Visualization

Mingyue Tan

Department of Computer Science
University of British Columbia
mtan@cs.ubc.ca

ABSTRACT

Current high dimensional data visualization tools are not effective in helping users gain insights of the data for many domains and tasks. This paper presents HDDVis, a very accessible High Dimensional Data Visualization tool that allows user to interactively explore the datasets from both low-dimensional projections and parallel coordinates.

The dimensionality reduction methods used by HDDVis include well-studied Principal Components Analysis (PCA) and a family of linear transformations that are recently proposed by Koren and Carmel [3]. They enrich the performance of PCA by adding a tunable pairwise weights, so that the new methods can simultaneously account for many properties of the data such as coordinates, pairwise similarities, pairwise dissimilarities, and clustering decomposition. Besides projective low-dimensional representations, HDDVis also allows user to interactively explore the data by brushing and linking non-projective parallel coordinates and two 2D scatter plots. Experiments are conducted on both synthetic and real data ranging from 2 to 320 dimensions, and the results show the correctness, usefulness, and effectiveness of HDDvis.

Keywords: dimensionality reduction, principal components analysis, brush and link, parallel coordinates

1 INTRODUCTION

Dimensionality reduction techniques aim to present high dimensional data in a low dimensional space, in a way that optimally preserves the structure of the data. Dimensionality reduction can be used to support knowledge workers in early stage of their data-understanding tasks. For example, we can use dimensionality reduction as a preliminary transformation applied to the data prior to the use of other analysis tools like clustering and classification.

Dimensionality reduction techniques can be classified into two categories: linear methods versus nonlinear ones [4]. Linear dimensionality reduction methods are those for which the mapping of high dimensional data to lower dimensional ones can be written as a linear transformation. Any other methods are considered as non-linear method. Despite being more limited than their nonlinear counterparts, linear techniques have distinct advantages including: 1) It is meaningful for the low dimensional axes to be a linear combination of the original axes. For example, a linear combination of monthly incomes is an interpretation of annual income. 2) Computational complexity of linear methods is generally lower than the nonlinear ones both in time and in space.

This paper considers linear methods, and the goal is to visualize labeled data. Labeled data is a collection of data elements that are partitioned into disjoint clusters by some external source, such as a domain specific knowledge. Visualizing high dimensional labeled data is quite challenging, as we would also like to reflect the cluster structures besides the desire to convey the overall structure. Specifically, we would like the low-dimensional representations revealing interesting structures like:

- Which clusters are well separated and which are similar?
- Which clusters are dense and which are heterogeneous?
- What is the shape of the clusters (elongated or spherical)?
- Which data coordinates account for the decomposition to clusters?
- Which data points are outliers?

Besides dimensionality reduction, parallel coordinates is another important multidimensional visualization technique that induces a non-projective mapping between N-Dimensional and 2-Dimensional sets [9]. Parallel coordinates lay out coordinates in parallel, and each data element is represented as a line passing though the coordinates values at the values of that dimension. Parallel coordinates is a very powerful technique for modeling relations, except that it requires user expertise in the relative mathematics.

This paper presents HDDVis, a High Dimensional Data Visualization tool that implements a total of seven linear methods and allows user to choose the one that is best suitable for their task. For beginner users who are not familiar with the linear methods, HDDVis allows them to choose a task to perform and then directs them to a suitable method. Many interaction techniques are used in HDDVis, such as selection (focusing), and brush & link.

The rest of this paper is as follows: Section 2 describes the related work in this area. Section 3 explains the seven linear methods and their suitability for a particular task. Section 4 describes the key features of HDDvis. Following this, in section 5, one sample scenario is presented to illustrate how HDDVis could be used. Section 6 explains the high-level implementation of the system and the lessons learned over the course of the project. Section 7 shows the experimental results. Section 8 presents an evaluation of the strengths and weakness of HDDVis.

2 RELATED WORK

2.1 Methods of Dimensionality Reduction

Besides linear versus nonlinear, we can also categorize dimensionality reduction methods into coordinates-based methods versus pairwise-weights-based ones [3]. Multivariate data are usually supplied in one of two basic forms: either each data element is a vector of variables, or some numeric value is provided to describe the relationship between two data elements. In the first case, the term *coordinates* is used to denote the different entries of the data elements, and the dimensionality reduction methods that deal such data are called coordinates-based methods. In the second case, the term *weights* is used to represent the pair-wise relationships between the data elements, and those dimensionality reduction methods that can deal with such data are called weighted-based methods. Distances, similarities, and dissimilarities are commonly used weights. For example, multidimensional scaling, or MDS, is a technique that attempts to preserve the similarities in its lower embeddings.

The dimensionality reduction technique that Koren and Carmal [3] proposed spoils the dichotomy coordinates/weights by allowing for the merge of both forms in a single framework. One way to look at their methods is as coordinate based methods, which are capable of taking into consideration pair-wise weights if these are available. Details of their methods are presented in section 3.

2.2 Visualization Tools of Dimensionality Reduction

Current visualization tools are not effective in helping user understand the structure of the data because they are either lack of interaction or of accessibility.

Isomap [6] and LLE [10] are both very advanced techniques of dimensionality reduction; however, their Matlab implementations only display the low-dimensional embeddings and do not allow any interaction.

Xgvis [12] has a better GUI and allows user's interaction, but its features are too limited. FSMvis [7] has the best interface; however, it is not quite accessible. It takes time to figure out its features and functionalities.

3 LINEAR TRANSFORMATIONS

This section presents the linear methods used in HDDvis. Except the naïve PCA, all the other methods are taken from [3]. The claims taken from the paper are not proved. Interesting readers can find full proofs in the original paper.

PCA is a widely used projection scheme that projects (possibly correlated) variables into a (possibly uncorrelated) lower number of variables called principal components. The principal components are ordered such that the first principal component accounts for the most variability of the original data. By using the first few principal components, PCA make it possible to reduce the number of significant dimensions while maintaining the maximum possible variance thereof.

3.1 Naïve PCA

A common explanation for PCA is as the best variance preserving projection. Koren and Carmel [3] derived PCA in a different, yet related motivation. They proved that PCA maximizez the sum of projected pairwise squared distance:

$$\sum_{i<j} \left(\text{dist}_{ij}^{p}\right)^2 .$$

where dist_{ij}^{p} is the Euclidean distances between the element *i* and element *j*.

3.2 Weighted PCA

PCA can be generalized by introducing nonnegative pair-wise weights, and seek a projection that maximizes the weighted sum.

$$\sum_{i<j} d_{ij} \left(\text{dist}_{ij}^{p}\right)^2 .$$

As W_{ij} becomes larger, it is more important to place points *i* and *j* further apart. Thus, we control the pairs through which we want to scatter the data.

3.2.1 Weight Specified PCA

If the user has some external knowledge about the dissimilarities, they can include the dissimilarity matrix in the data file, and HDDVis will compute the projection using the dissimilarities.

3.2.2 Normalized PCA (To cope with outliers)

Naïve PCA strives to maximize the sum of squared distances, which emphasizes the contribution of the points with large distances. In the case where outliers (noise) present, this behavior will impair the results of PCA. Normalized PCA chooses

$$d_{ij} = \frac{1}{\text{dist}_{ij}} .$$

as the weight to down-weight the contribution of the large distances to the summation. The inverse squared distances can also be used as the weight.

3.3 Supervised PCA

So far we have not taken into consideration of the cluster labels. To see an embedding that separates clusters, we may artificially underweight the dissimilarities between intra-cluster pairs of data elements. This can be done by multiplying the intra-cluster dissimilarities by some decay factor $0 = t = 1$, obtaining the weight as

$$d_{ij}^{\text{labeled}} = \begin{cases} t \cdot d_{ij} & i \text{ and } j \text{ have the same label} \\ d_{ij} & \text{otherwise.} \end{cases}$$

3.4 Optimized PCA

When both inter- and intra-cluster scatters are maximized along the same directions, supervised PCA fails to separate clusters. Now we have two demands: maximizing inter-cluster scatter and minimizing intra-cluster scatter. Three methods are implemented for optimized PCA to meet the two demands. Complicated formulas are omitted in this section.

3.4.1 *Maximization of weighted pairwise dissimilarities (inter-cluster repulsion)*

This method strives to maximize the weighted pairwise dissimilarities. With intra-cluster dissimilarities set to 0, and inter-cluster dissimilarities set to 1, we are seeking a projection that separates the clusters apart.

3.4.2 *Minimization of weighted pairwise similarities (intra-cluster attraction)*

The similarity-based approach can also be used for labeled data. Here, we have to decay all the similarities between elements from different clusters, using some decay factor 0 =t =1,

$$s_{ij}^{\text{labeled}} = \begin{cases} s_{ij} & i \text{ and } j \text{ have the same label} \\ t \cdot s_{ij} & \text{otherwise} \end{cases}$$

Typically, we set $t = 0$, which means that we do not want the low dimensional embedding to reflect any proximity relations between elements from different clusters.

3.4.3 *Ratio Optimized (both inter-cluster repulsion and intra-cluster attraction)*

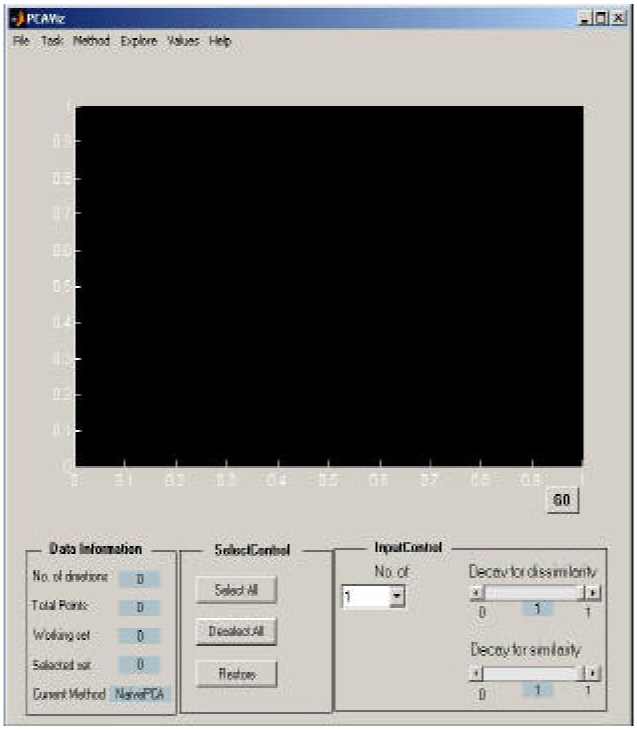
If we are given both pairwise similarities s_{ij} and pairwise dissimilarities d_{ij}, we may decay inter-cluster similarities and intra-cluster dissimilarities and to seek an projection that maximizes

$$\frac{\sum_{i<j} d_{ij} \left(\text{dist}_{ij}^{p}\right)^{2}}{\sum_{i<j} s_{ij} \left(\text{dist}_{ij}^{p}\right)^{2}}$$

In summary, the family of linear methods is robust and flexible, and is particularly suitable for labeled data.

4 HDDVis

HDDVis provides several graphical representations of the high dimensional data and allows user interactively explore them. Figure1 shows the interface of the main window of the system. The rest of this section describes the functionalities provided by the system and the various interactions supported to best aim user to comprehend the data.



No. of directions 0
Total Points 0
Working set 0
Selected set 0
Current Method NaivePCA
No. of 1
Decay for dissimilarity 0 1 1
Decay for similarity 0 1 1

Figure 1

4.1 Data Format

To use the system, the user only needs to prepare a data file containing the dataset they want to visualize. The data file must be in the .mat format, and contains 1) a N x D matrix of the dataset, where N is the number of points and D is the number of dimension, 2) a N x 1 vector of the labels of each data point. Once the data file is handy, the user can visualize the data using HDDVis.

4.2 File

This menu item provides three options, *load*, *save* and *exit*. *Load* allows user to load the data file into the workspace. *Save* allows users to save current figure and currently selected data into files so that they can retrieve and use them in their further analysis. *Exit* simply closes the window.

4.3 Task & Transformations

Once a dataset is loaded in, the user can select the dimensionality reduction methods from either the task menu item or the transformation menu item.

4.3.1 Task

Task is designed for beginner users who are not familiar with the dimensionality reduction algorithms we implemented, but definitely are clear about what tasks they want to perform. We list three common tasks (general view, outlier detection, and cluster structure view), and for each task, we list the method(s) that we think are best suitable for that particular task. For example, if the user wants to detect and remove outlying (noisy) points of the raw dataset before further analysis, then we direct him/her to naïve PCA method. In this case, we make use of the primal limitation, non-robustness towards outliers, of naïve PCA and help user perform the task.

4.3.2 Transformation

Transformations lists the linear transformations used to do dimensionality reduction. This allows expert users to quickly choose the method they want to use, and also enables beginner users to play with the various algorithms and gain some understanding about how these methods work.

4.4 Input Control Panel

The bottom right input control panel allows the user to control the display of the scatter plots by 1) specifying the number of dimensions they want their data to project into using the method they choose in 4.3. The pop-up menu provides three options, 1D, 2D or 3D 2) specifying the values of decays of similarities or dissimilarities if they use supervised PCA or optimized PCA method. By moving the sliders, users change the decays of similarities and dissimilarities.

4.5 Display & Color

Once method and parameters are specified, user can see the scatter plots of the lower dimensional embeddings by pressing the Go button. Default values of method and parameters will be used if the user press *Go* before specifying them. Different clusters are plotted with different colors, which makes it easy to see the structure of the clusters and to identify the outliers as discussed in next section.

4.6 Selection, Highlighting, & Color

Users can choose a subset of data points to visualize by selecting all points within a rectangle area. Data points are then painted as red to make them distinguishable. Subsequent actions, such as dimensionality reduction, exploration, and view (context will become clear in subsequent subsections), will apply to the selected data. *Select all* button allows users to select all the data points displayed in the window. *De-select* undos a selection and make all data points on the screen unselected. After a sequence of selections, users may want to go back and visualize the original data that were loaded from the file. *Restore* can do this job for them. By clicking restore, the scatter plots of the original data will be displayed according to current parameter settings.

4.7 Rotation

If the data is projected into 3D, then users can rotate the scatter plots using mouse.

4.8 Data Information

The left bottom panel contains the brief description of the data points by showing the number of dimensions and the number of data points. The latter may change as we select / deselect subset of data. Total data refers to the original data we loaded from a file; working data refers to the data points that are currently displayed on the screen. Selected data refers to the data points that are within the rectangle and highlighted as red.

This information provides user a context of the data points that they currently see on the screen so that they won't get lost after a series of consecutive transformations.

4.9 Explore and Brush & Link

Explore will trigger a new pop up window whose interface is shown in Figure2. This tool is useful for interactively understanding the correlations of any four dimensions of the data, either original data or transformed data. At the top of the tool is a parallel coordinate plot representing the specified data. The pop-up menus in the middle allow users to specify which subspace they are interested in visualizing. The bottom two scatter plots also visualize the selected data. Clicking on the scatter plots will create a brush which can be moved about. Any "tuples" within the brush will be highlighted across all three plots via linking.

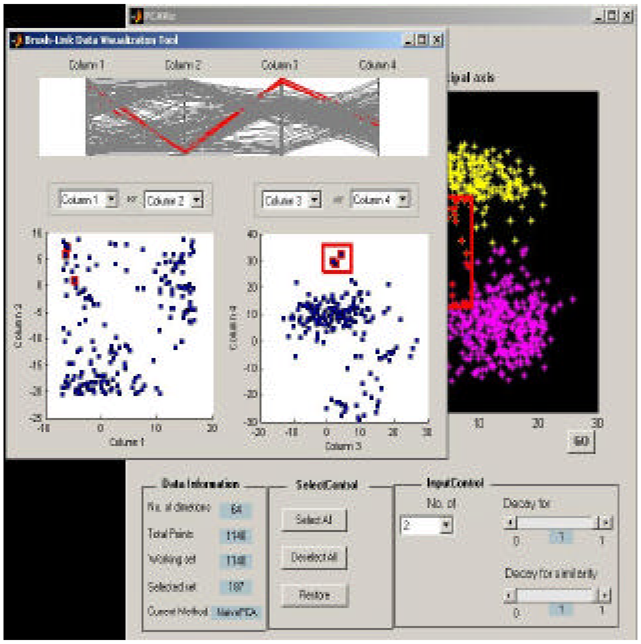
4.10 Values

The menu item *values* also triggers a new pop up window except that a table rather than plots will be shown on the new window. The table containing the values of the data, either original or transformed depending on which one the user has chosen. figure 3 shows such a table.

4.11 Help

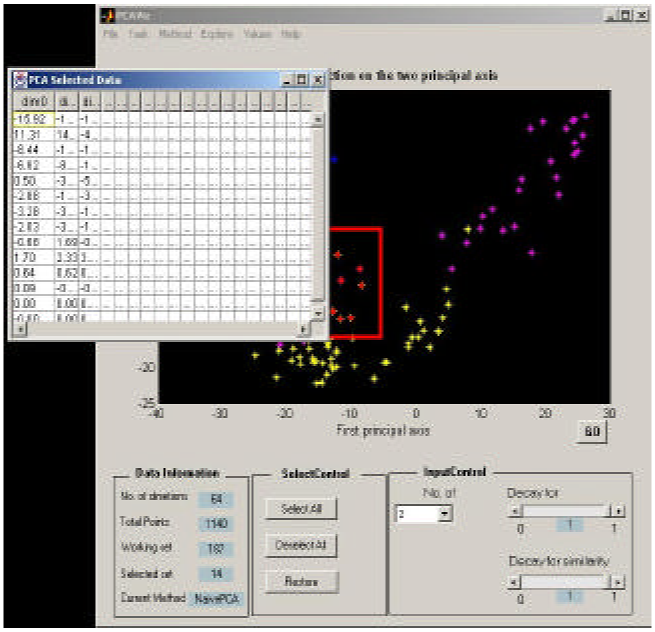
Help provides to user a detailed manual on the usage of the system.

In summary, the visualization techniques that system uses include linear dimensionality reductions, parallel coordinates, coloring, selecting (focusing), and many other interactions.



No. of dimensions 64
Total Points 1140
Working set 1140
Selected set 187
Current Method NaivePCA

Figure 2



dim0	di...	di...
-15.92	-1...	-1...
11.31	14...	-4...
-8.44	-1...	-1...
-6.12	-8...	-1...
0.50	-3...	-5...
-2.18	-1...	-3...
-3.28	-3...	-1...
-2.13	-3...	-1...
-0.86	1.69	-0...
1.70	3.33	3...
0.64	0.62	0...
0.09	-0...	-0...
0.00	0.00	0...

No. of dimensions 64
Total Points 1140
Working set 187
Selected set 14
Current Method NaivePCA

Figure 3

5 SCENARIO OF USE

The effectiveness of the dimensionality reduction methods largely depends on the specific data sets used and the particular tasks to perform. A method that is superb for one task may fail the other. Previous section has shown a detailed scenario on how to use the system to see a projection, to explore plots, and to view data values. In this section we present how outlier detection can be done using HDDVis. Conventionally, a data point is an outlier if it is far from the majority of points. In the case where data are labeled, a point is an outlier if it's far from the majority points belonging to the same cluster. Outliers are very common in real datasets. The identification of outliers can lead to the discovery of some interesting unexpected knowledge in areas such as E-Commerce and credit card fraud.

In this section, we show two ways to detection outliers using HDDVis.

5.1 The Use of Naïve PCA to Detect Outliers

As discussed above, naïve PCA is sensitive to outliers, so it can be used to detect outliers in some cases. To do this, the user can go to task menu and specify their task, HDDVis will then direct to Naïve PCA algorithm.

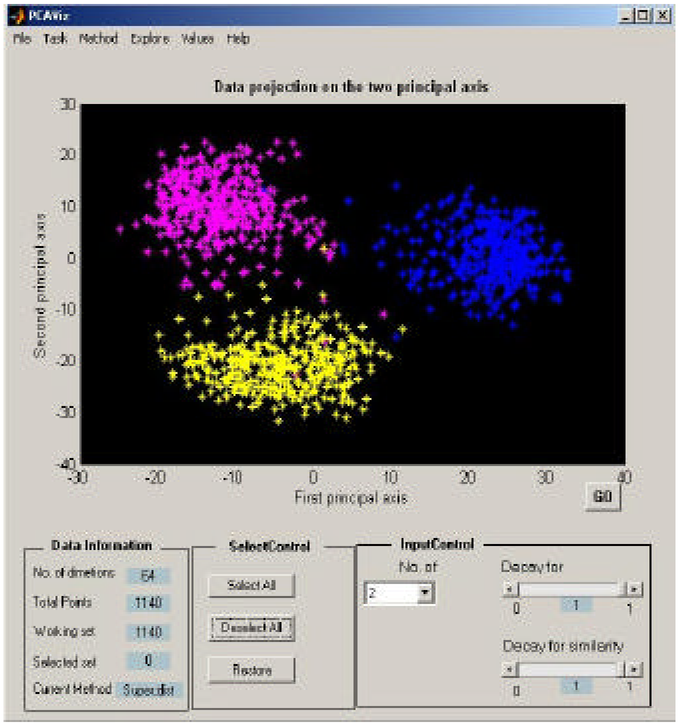
5. 2 The Use of Color & Selection to Detect Outliers

An example of using HDDVis to do outlier detection is shown in Figure 4, which shows a two-dimensional projections of a portion of Alpadin's handwritten digits dataset. The dataset, developed by Alpaydin and Kaynak [1] and publicly available in [2] . In this example, we choose three digits 0, 4, and 6.

The two-dimensional projections of these data are almost the same, but we can use color & selection to detection outliers. Figure4(a) shows a scatter plot obtained by supervised PCA with normalized weights. From this plot, we see that a pink point lies in the center of the yellow class. This pink point is definitely an outlier. To take a close look of the data, we can select a small portion of the data including the outlying point, as Figure4 (b) shows. We can then press *Go* and do the dimensionality reduction on those selected data. Now the outlier is separated apart from other points as Figure(c) shows. We can now view its values from the *values* menu. Further analysis can be performed on this outlier. For example, if the user finds it's actually a typo and mislabeled, then he/she can correct it.

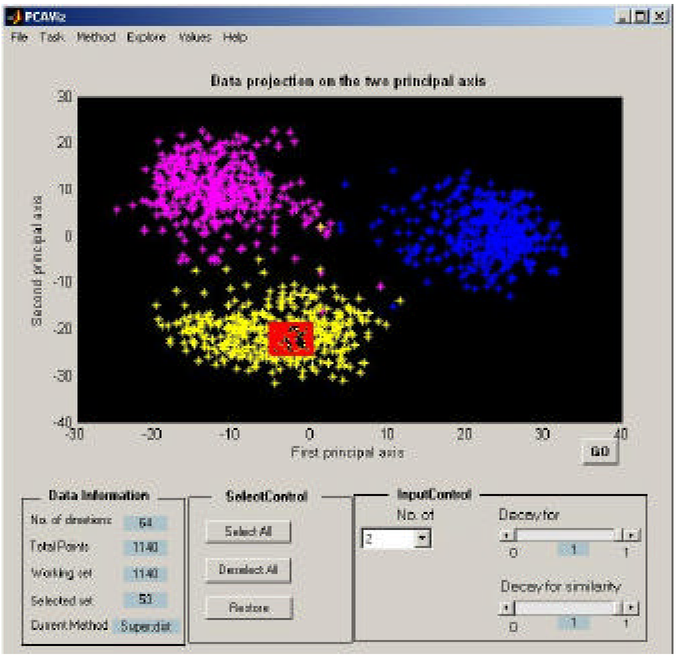
The next question one may come up with is that in which case, naïve PCA is powerful for identifying outliers. The answer to this question can be derived from the nature of PCA. Intuitively, naïve PCA is good at separating outliers from the main bulk if the direction that maximizes the distance between the outlier and the remaining points is same as the direction that maximizes the variance.

In the case where naïve PCA is not appropriate for detecting outliers, our second way, selecting and navigating a subset, can serve as a backup.



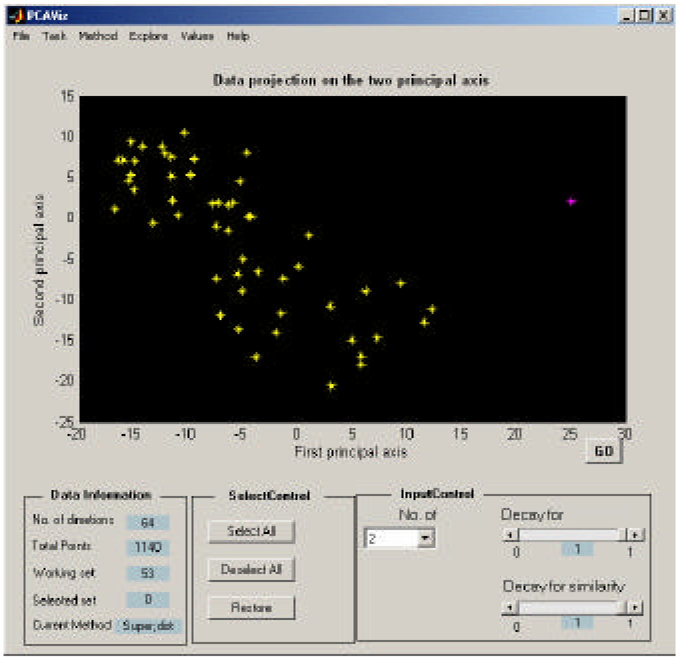
No. of dimetions 64
Total Points 1140
Working set 1140
Selected set 0
Current Method Superdist
No. of 2
Decay for 1
Decay for similarity 1

Figure 4 (a)



No. of dimetions 64
Total Points 1140
Working set 1140
Selected set 53
Current Method Superdist
No. of 2
Decay for 1
Decay for similarity 1

Figure4(b)



No. of directions 64
Total Points 1140
Working set 53
Selected set 0
Current Method Super,dist
No. of 2
Decay for 1
Decay for similarity 1

Figure4(c)

6 INPLEMENTATION

HDDVis is implemented using Matlab 6.5 and Java 2, and consists of several modules. The GUI is built using GUIDE, an interactive GUI design Environment. Each linear transformation method is implemented in a separate function M-file. The tool used for exploring parallel coordinates and scatter plots are taken from an open source called Brush Link Visualization Tool at MathWorks [5]. The pop up window containing the data tables is implemented as a java class, which is called by a Matlab function when a user chooses to view data values.

HDDVis is designed to be as extensible as possible, while maintaining the scope of visualizing high dimensional data. Other dimensionality reduction methods can be easily added in by implementing the method as a function M-file and make it as an option in the GUI transformation menu item.

The reasons why I use Matlab to implement HDDVis is that Matlab is a high-performance environment that includes tools for mathematical computation and visualization. Almost always, dimensionality reduction involves many sophistical mathematical computations. As we are developing a visualization tool, making use of Matlab's nice plots is a good choice. Though Java also provides nice visualization, its ability of matrix computing is not comparable with Matlab.

6.1 Lessons Learned

Over the course of this project, I have learned several things:

- The most important thing that I have learned is how to use Matlab to create GUI.
- Choosing an effective color scheme is difficult once we take the needs of color-blinded people into consideration. Furthermore, since HDDVis uses different colors for different clusters, as the number of clusters increase, we have less choices of effective colors that distinguish clusters well.
- The linear transformations from [3] work well on real data, but fail on the data that are randomly chosen from a subspace of the original data. This might because the topology of the original high dimensional data is deformed as when we remove some dimensions.

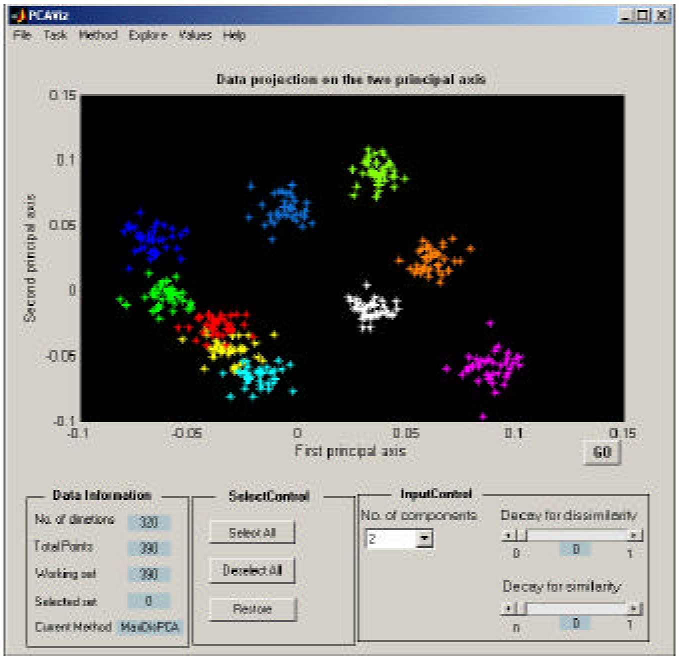
7 RESULTS

7.1 Correctness

HDDVis is tested with five synthetic and three real datasets of differing cardinality and dimensionality. Figure 4(a) and Figure 5 show two real datasets which were used in paper [4], and HDDVis generates exactly the same projections as in [4]. This demonstrates that HDDVis correctly implements the linear transformations algorithms.

7.2 Time

As discussed in section 1, linear transformations are usually much efficient than non-linear approaches. Except the optimized PCAs, all other algorithms take very little time to run. HDDVis displays the projections immediately after *Go* button is pressed.



No. of dimensions 320
Total Points 390
Working set 390
Selected set 0
Current Method MaxDifPCA
No. of components 2
Decay for dissimilarity 0
Decay for similarity 0

Figure 5

Hand-written digits dataset containing 390 samples in 320 dimensions. The two-dimensional embedding of the method described in 3.4.1, exhibiting good separation of clusters.

8 EVALUATION

The current implementation of HDDVis has many strengths and weaknesses. Several of these weaknesses are not due to technic al limitations, but time limitations.

8.1 Strength

HDDVis provides multiple graphical representations of the data and allows many interactions. The linear methods provide a total of nine ways to project the data, thus there is a great chance that users can always find one method suitable for their task. The interactive interface allows user to control the way a graph to display and gives them immediate feedback. This process will help user better understand the data.

8.2 Weakness

There are some us eful features that HDDVis missed. For example, it does not allow user to dynamically add or remove data from the plots or tables. Another feature HDDVis missed is range selection. Current implementation only allows user to select subset of points in 2D projection. It would be useful if user could select data value ranges on one or more axes and mark the corresponding points in the visualization by selected color. It should also allow user to choose a subspace to work on as some dimensions might be more important than others.

Matlab is not a very efficient programming language compared with C. From my experience, the efficiency will be dramatically improved if we write the core functions in C that can be called by Matlab.

One can conduct future work by implementing other dimensionality reduction techniques and integrated with current system so that user can compare the performance of the each algorithms and may even come up with new algorithms.

References

[1] E. Alpaydin & C. Kaynak, “Cascading Classifiers”, Kybernetika 34 (1998) 369-374

[2] C. Blake & C. Merz (1998), UCI repository of machine learning databases [http://www.ics.uci.edu/~mlearn/MLRepository.html]. Irevine, CA: University of California, Department of Information and Computer Science

[3] L. Carmel & Y. Koren, “Visualization of Labeled Data Using Linear Transformations”, Proc. IEEE Information Visualization (InfoVis’03), IEEE, pp.121-128, 2003.

[4] L. Carmel & Y. Koren, “Robust Linear Dimensionality Reduction”, IEEE Transactions on Visualization and Computer Graphics

[5] J. conti, Brush and Link Visualization Tool http://www.mathworks.com/matlabcentral/fileexchange/

[6] J. Langford, V. de Silva, & J. Tenenbaum, “A Global Geometric Framework for Nonlinear Dimensionality Reduction”, Science 290 (5500): 2319-2323, 22 December 2000.

[7]Morrison & Chalmers, FSMVis system (http://www.dcs.gla.ac.uk/~morrisaj)

[8]E. Kandogan “Visualization Multi-dimensional Clusters, Trends, and

Outliers Using Star Coordinates", Proc. KDD 2001

[9] A. Inselberg & B. Dimsdale, "Parallel Coordinates: A Tool for Visualizing Multi-Dimensional Geometry". IEEE Visualization '90, 1990

[10] S. Roweis & L. Saul, "Nonlinear Dimensionality Reduction by locally linear embedding", Science, v.290 no.5500, Dec.22,2000. pp.2323-2326.

[11] J. Wise, J. Thomas, K. Pennock, D. Lantrip, M. Pottier, and A. Schur. Visualizing the non visual: Spatial analysis and interaction with information from text documents, In Proceedings of IEEE Information Visualization, pages 51-58,

[12]Xgvis http://www.research.att.com/areas/stat/xgobi/#xgvis