High Dimensional Data Visualization

Fabian Keller
uni@fabian-keller.de

Advanced Seminar:
Large-Scale Visualization

Advisor:
Steffen Koch

Stuttgart
August 15, 2015

1 Introduction

The data gathered in various scientific domains and industrial applications is steadily growing in size. What today seems to be large-scale may become small-scale in five to ten years. The size of the data increases in both, data set size and the number of measured or simulated variables of a single datum. However, the increase in the size of the data leads to an increased complexity when dealing with such data sets. On the one hand the large-scale data needs to be processed within reasonable amounts of time while on the other hand, the perception of the human being analyzing data can only deal with a certain complexity limited by perception. For the latter, it is inevitable to reduce the complexity of the data. A commonly chosen method in applications such as compression, classification or visualization is to reduce the number of dimensions of the data [10]. Dimension reduction techniques aim to compute a data set with fewer dimensions based on the original data, that still represents patterns and characteristics of the original data.

The dimension reduction techniques can generally be classified into linear and non-linear techniques, as seen in Figure 1. Next to that, the different algorithms either aim to preserve global or local properties of the high dimensional space in the low dimensional counterpart. As research specifically focuses on the development of non-linear techniques, literature reviews have often summarized and compared existing techniques against each other [5, 10, 11, 33]. For visualization purposes the approaches commonly project the high-dimensional space to a two or three dimensional space which can then be visualized using 2D or 3D scatter plots. There are also visualization techniques that are able to visualize more than 3 dimensions, for example parallel coordinate plots or glyphs, which can also be used to visualize the reduced data [8, 15, 18].

This chapter aims to introduce basic methodologies to cope with high dimensional data. That includes approaches to reduce the dimensionality of data and also approaches to visualize the low-dimensional representation of the high-dimensional data. The focus hereby is on introducing general ideas, mathematical concepts and considerations to make when dealing with the different approaches and high-dimensional data. In addition to that, non-exhaustive references to more specific literature is provided.

The remaining sections of this chapter are organized as follows. Section 2 introduces general concepts of dimension reduction techniques used for visualization purposes. Different visualization techniques for high dimensional data are then introduced in section 3. A comparison of different algorithms and their characteristics is then presented in section 4 leading to a conclusion and future work for high dimensional data visualization in section 5.

2 Dimension Reduction Techniques Overview

Data gathererd from the real-world, such as fMRI scans or speech signals, is often high dimensional with thousands of dimensions. Though the actual data points x_i with $i \in \{1, .., N\}$ are represented in the high dimensional space with D dimensions, the data does not fill the whole high dimensional space, but instead lies on or close to a manifold within the high dimensional space. The smallest dimension required to represent this manifold is called the intrinsic dimension $d < D$ of the data. Dimension reduction techniques aim to find a representation y_i with $i \in \{1, .., N\}$ of the high dimensional data x_i in a space with the intrinsic dimension d of the data. However, techniques

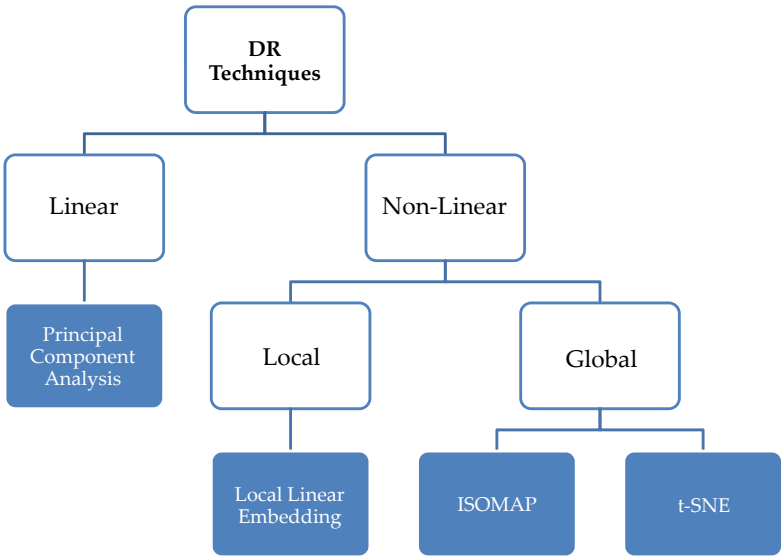


Figure 1: Simplified taxonomy of the different dimensionality reduction techniques, as introduced by van der Maaten et al. [33].

used for visualization purposes need to reduce the high-dimensional data to ≤ 3 dimensions for simple visualization techniques, or up to two-digit dimensions for enhanced techniques such that the visualizations can still be presented and perceived with reasonable effort.

Generally, dimension reduction techniques can be classified into linear and non-linear techniques. Linear techniques assume that the manifold in the high dimensional space is linear, while non-linear techniques are not based on this assumptions. Typically, dimension reduction techniques do not know the manifold nor the intrinsic dimension of the data. However, it is of high interest to obtain the lower-dimensional representation y_i before further processing the data to avoid the curse of dimensionality.

This section further presents the key ideas of selected linear and non-linear dimension reduction techniques. As it is one of the most commonly used linear techniques PCA is representatively explained. For non-linear techniques Local Linear Embedding, ISOMAP and t-SNE are explained as they are also among the commonly used techniques and they representatively showcase different approaches to reduce the dimensionality of the data.

2.1 Linear techniques

Principal Component Analysis

Principal Component Analysis (PCA) [1, 5, 12, 35] aims to project the given data $x_i \in \mathbb{R}^d, i = 1, ..., n$ to a lower-dimensional representation by iteratively projecting all data points to multiple lines $\mathcal{L}$ embedded in $\mathbb{R}^d$, which are also referred to as *principal components*. Each line can be expressed as

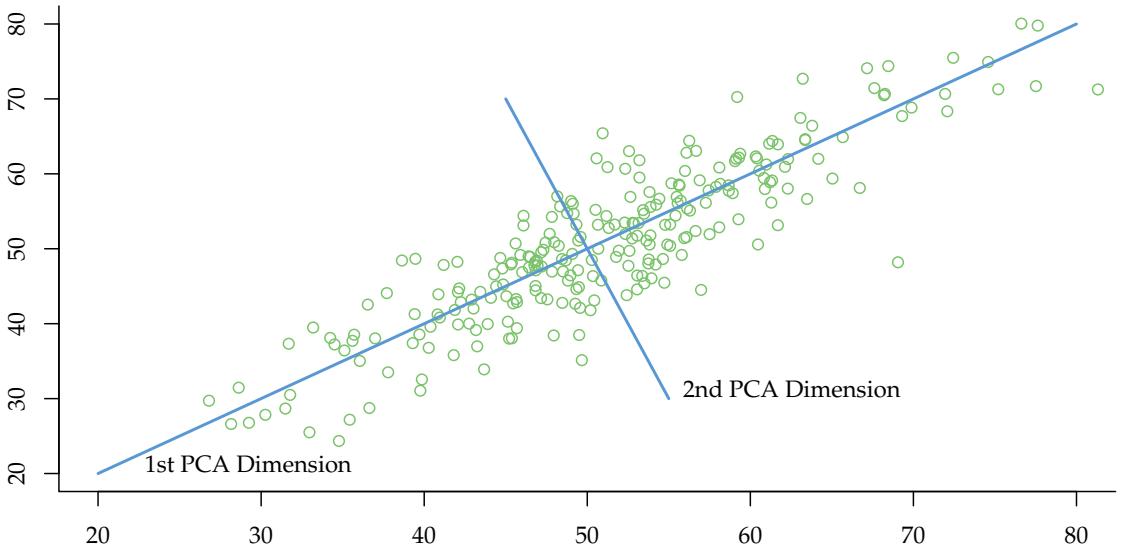


Figure 2: Example of the principal components PCA identifies for some normal distributed data.

$\mathcal{L} = \mu + \theta_i n$, where μ is the mean value of the dataset x_i and $\theta_i \in \mathbb{R}, \sum_i \theta_i = 0$ and an arbitrary $n \in \mathbb{R}^d$ which spans the line. The ideal line minimizes the total distance of all data points x_i to the projected line $\mathcal{L}$, which is equivalent to maximizing the variance of the projected points on the line using the formula:

$$v_n = \frac{1}{m} \sum_{i=1}^{m} ((x_i - \mu) \cdot n)^2 = \frac{1}{m} \sum_{i=1}^{n} \theta_i^2 \tag{1}$$

If the data points were perfectly linear, then the first computed line $\mathcal{L}$ would contain all projected points with an error of 0 and the points projected on all lines orthogonal to $\mathcal{L}$ would then show a variance of 0. However, if the data is not perfectly linear, the goal is to project the data points to further lines maximizing the variance, that additionally are orthogonal to all other existing lines. Each line is then referred to as *principal component* of the data, which are sorted by their variance in descending order. The dimension of the data can be reduced by using the first $k < d$ principal components as an approximation of the whole data set. As the remaining $k - d$ principal components show a lower variance than the first k components, the information added by the remaining principal components (and which is thus lost by the projection) is negligibly low.

Figure 2 shows some data which has been analyzed using PCA. The first line maximizing the variance of the projected points is represented as "1st PCA Dimension". The second line represented by "2nd PCA Dimension" is orthogonal to the first line and again maximizes the variance of the projected points in the remaining dimension. Each data point can then be represented by a linear combination of vectors on the two lines.

2.2 Non-linear techniques

Local Linear Embedding

Local Linear Embedding (LLE) [5, 12, 24, 25] reduces the data to a lower-dimensional representation $y_i \in \mathbb{R}^{l<d}$ by computing neighborhood preserving embeddings in the high dimensional data $x_i \in \mathbb{R}^d, i = 1..n$, that are then mapped to the low-dimensional counterpart. LLE assumes that the neighborhoods lie on or near a locally linear patch of the non-linear manifold. The locally linear patches are then mapped to global coordinates on the manifold.

The first step of the algorithm finds the nearest neighbors of each data point x_i, represented as the function $N_i(j)$ which returns the index of the jth neighbor of the ith data point. The neighbors can either be chosen by finding the k nearest neighbors or by finding all k neighbors within the fixed radius ϵ. Then the second step finds a representation of each data point x_i using its surrounding neighbors. The data point is described by a linear combination of the weights w_{ij} of the surrounding neighbors such that the reconstruction error is minimized:

$$\min_w \sum_{i=1}^{n} \left\| x_i - \sum_{j=1}^{k} w_{ij} \cdot x_{N_i(j)} \right\|^2 \tag{2}$$

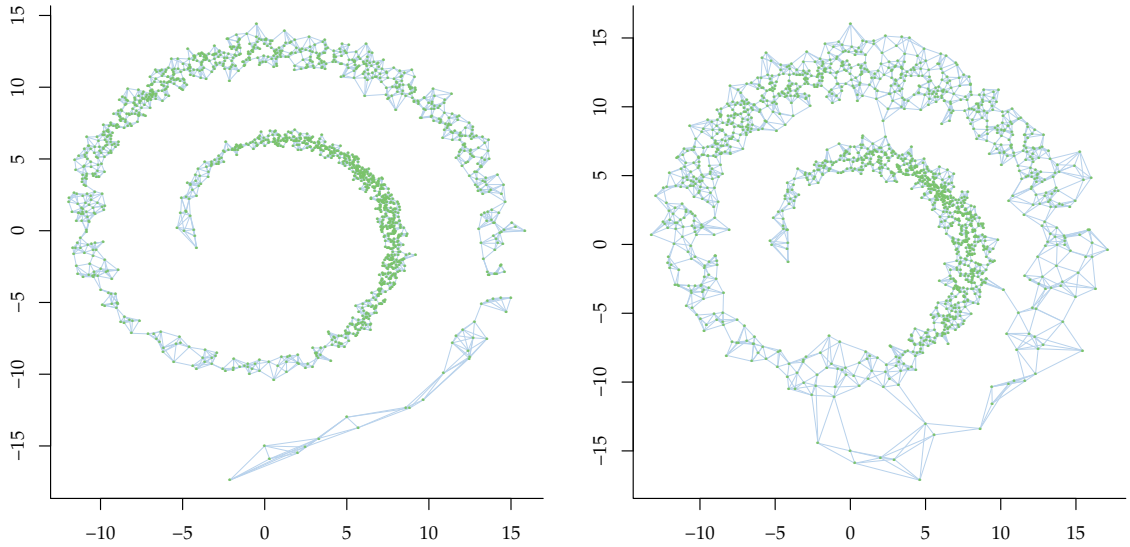
As a last step, the weights are used to find the low dimensional representation y_i of each data point by, again minimizing the reconstruction error using the surrounding neighbors, but this time in the low-dimensional representation:

$$\min_Y \sum_{i=1}^{n} \left\| y_i - \sum_{j=1}^{k} w_{ij} \cdot y_{N_i(j)} \right\|^2 \tag{3}$$

ISOMAP

ISOMAP [3, 30] reduces the dimensionality by leveraging the geodesic manifold distances between all the data points. The algorithm takes three steps to reduce the dimensionality. The first step computes an undirected nearest neighbor graph G such that all edges $\{x_i, x_j\}$ of the graph only exist iff x_i and x_j are nearest neighbors. The nearest neighbors of x_i can be found by choosing the k nearest neighbors or by finding all l neighbors within the fixed radius ϵ. The weight of the edges is the distance $d(x_i, x_j)$ between the two data points in the input space using a valid distance metric. The second step of the algorithm then estimates the geodesic distances between all pairs of points by computing the shortest path between the two points $d_G(x_i, x_j)$ on the neighborhood graph G using a standard shortest path algorithm. Those distances are then used in the final step to create a mapping in the lower-dimensional representation that preserves the distances between the mapped points. To obtain the low-dimensional mapping, a technique called multidimensional scaling (MDS) [22] is used, which is not further elaborated within the scope of this work.

Figure 3 shows two SwissRoll datasets generated by applying Gaussian noise to a perfect SwissRoll. In Figure 3a the nearest neighbor graph perfectly models the embedding. However, on the right



(a) The neighborhood relations stay within their intended location on the manifold. Unrolling the non-linear manifold is perfectly possible.

(b) Adding noise to the dataset changes the neighborhood relations such that relations between different parts of the manifold are connected. Unrolling will not yield the intended result.

Figure 3: 5-nearest neighborhood graph construction on a non-linear manifold using ISOMAP. The blue lines represent the neighborhoods relations.

side of the roll there is a gap and hence, the neighborhood graph is bipartite leading to a lot of infinite shortest paths during the execution of ISOMAP. When increasing the noise of the data as seen in Figure 3b, the nearest neighbor graph starts to establish relations between points that are not actual neighbors in the manifold of the input space. However, the connected points between the graphs have a tremendous impact on the shortest path calculations, as they can be used as shortcuts between the spirals. As adding a little noise to the input can potentially yield a completely different result, the ISOMAP algorithm is not a robust algorithm by itself. Approaches exist to make ISOMAP more robust on noisy data [7, 27, 28].

t-SNE

Van der Maaten and Hinton [32] introduced t-SNE, which is a variation of Stochastic Neighbor Embedding (SNE) [17]. SNE creates a low-dimensional representation of the high-dimensional data in two steps. First SNE assigns a similarity measure to all data point pairs (x_i, x_j) by applying a Gaussian distribution between them which expresses the probability with which point x_i would choose x_j as neighbor, as follows:

$$p_{j|i} = \frac{\exp(-||x_i - x_j||^2/2\sigma_i^2)}{\sum_{k \neq i} \exp(-||x_i - x_k||^2/2\sigma_i^2)} \tag{4}$$

with σ_i being the variance of the Gaussian centered on point x_i. Hence, the closer two points are the higher the probability x_i would choose x_j as neighbor.

SNE then tries to preserve those probabilities in a low dimensional representation, where the similarities between the mapped points y_i and y_j are calculated in a similar way. The key difference is that the variances σ_i are fixed to $\frac{1}{\sqrt{2}}$ in the low-dimensional representation. Thus the conditional probability of two mapped points y_i and y_j in the low-dimensional representation can be calculated as follows:

$$q_{j|i} = \frac{\exp(-||y_i - y_j||^2)}{\sum_{k \neq i} \exp(-||y_i - y_k||^2)} \tag{5}$$

If a perfect mapping was possible, then $p_{j|i}$ is equal to $q_{j|i}$. Thus, the optimization objective of SNE is minimizing the differences between the conditional probabilities $p_{j|i}$ and $q_{j|i}$ of the respective data points in the high- and the low-dimensional space.

t-SNE modifies the SNE approach in two points. First, t-SNE uses a different cost function for the optimization objective having the advantage that it has simpler gradients. Second, t-SNE employs a Student-t distribution instead of a Gaussian distribution in the low-dimensional space. With this distribution, t-SNE is easier to optimize and reduces the crowding problem - the problem when a lot of data points with equal distances in a higher dimensional space need to be represented in a two-dimensional space.

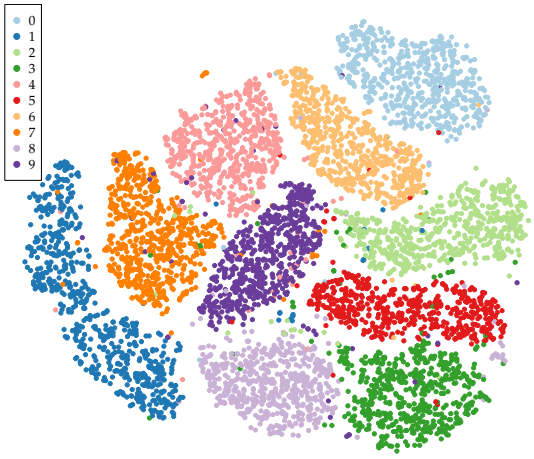
3 Visualization Techniques

Having reduced the dimensionality of the data as described in section 2 the lower-dimensional data can now be visualized leveraging various different visualization techniques. This section introduces 2D scatter plots, 2D scatter plot matrices, interactive 3D scatter plots and parallel coordinates.

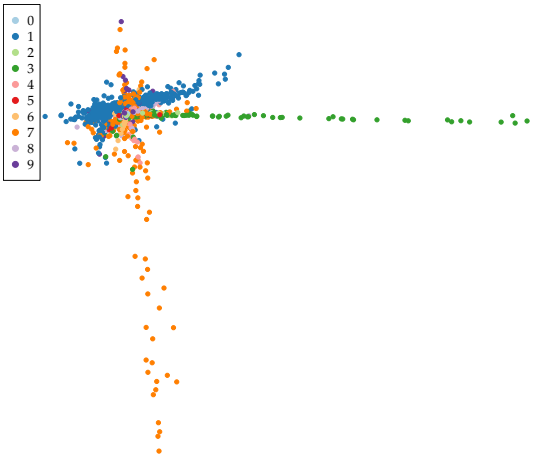
3.1 Scatter Plots

As most physical visual output devices have a two-dimensional output, it is convenient to use 2D plots for visualization purposes. Thus, the data has to be reduced to two dimensions and a commonly used visualization technique are scatter plots to display each single data point. These static scatter plots are especially useful, as according to Sedlmair et al. [26] they do not require time-consuming interaction overhead, as all data is immediately available in one region. Further there is no need to relate the visualized data to any other visualization, such that a single 2D scatter plot does not require a thorough understanding of relationships between different visualizations.

Figure 4 shows two scatter plots produced from the same data set - namely from 6000 handwritten digits represented by $28 \cdot 28 = 784$ pixels. As each pixel represents a dimension of the high dimensional space, two different dimension reduction techniques were applied to obtain the two figures 4a and 4b. Each color represents one of the 10 digits, thus there are 10 different colors used within the plot. From a perception perspective, it is easier to spot the relationships in the data in Figure 4a than in Figure 4b. This is due to the fact that the data in Figure 4a is more aligned to clusters by the dimension reduction technique than the other method. This means that while scatter plots can reveal useful information and structures, the result heavily depends on the input data.



(a) Dimension reduction with t-SNE

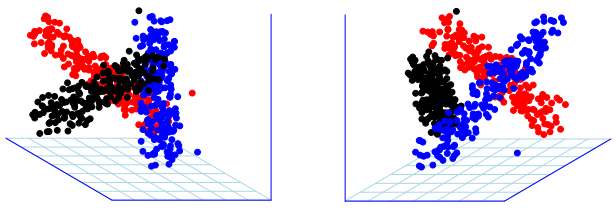


(b) Dimension reduction with Local Linear Embedding.

Figure 4: 2D scatter plot representations of 6000 hand written digits from the MNIST data set. Each digit image is represented by $28 \cdot 28 = 784$ pixels, hence dimensions. PCA was used to reduce the data set to 30 dimensions before applying the respective dimension reduction techniques to further reduce the dimensionality to 2 dimensions.

3D scatter plots: Compared to 2D scatter plots, 3D scatter plots visualize a third dimension of the low-dimensional representation of the data. The third dimension adds additional information to the visualization that cannot be visualized with a single 2D scatter plot. However, the technique requires interaction to fully yield its potential, as the perception strongly depends on the perspective from which the user looks at the data.

Figure 5 shows the same dataset visualized as 3D scatter plot from different perspectives. By only looking at a single perspective it is not possible to correlate or compare the different data classes. Even when considering the second perspective it is not clear how the different colored data classes relate to each other. Hence, 3D scatter plots are only meaningful when viewed in an interactive environment, where the user may turn, pan or zoom the visualization. However, Sedlmair et al.



(a) (b)

Figure 5: A 3D scatter plot from different perspectives. Taken from [26].

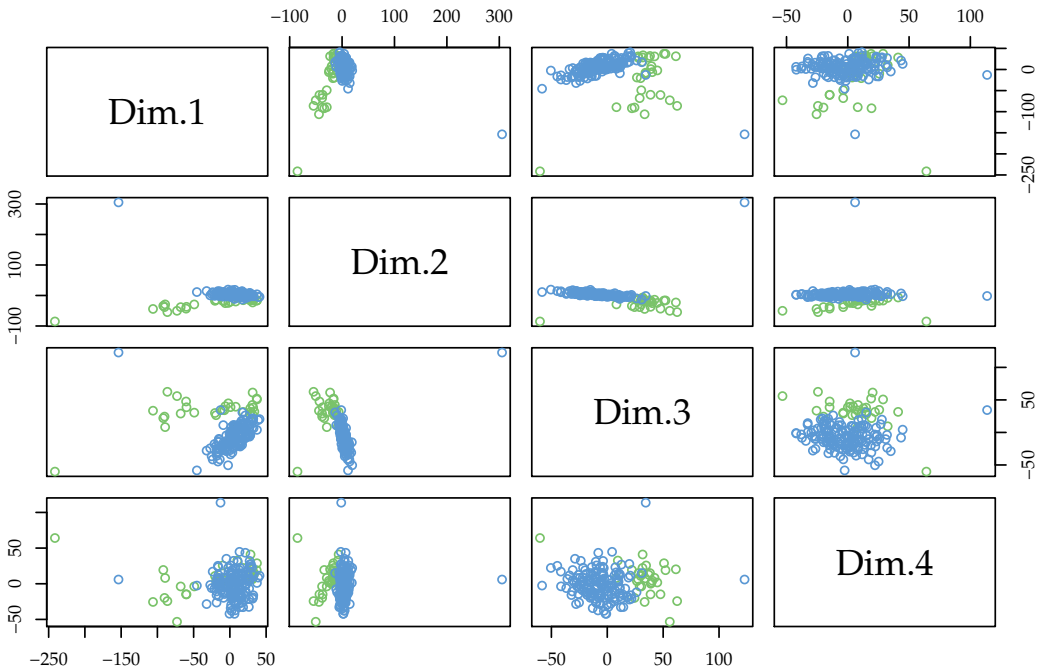


Figure 6: Principal component analysis performed on a lung cancer data set by Gordon et al. [13] using data of 181 samples. For each sample, 12533 dimensions were measured. The four principal components with the highest variance are visualized in a scatter plot.

[26] have shown in a user study that the 3D scatter plots do not help users to analyze the data, as the 3D scatter plots suffer from various limitations such as the perspective or the occlusion of data points by other data points.

3.2 2D Scatter Plot Matrices

2D scatter plot matrices can basically visualize an arbitrary number of dimensions using 2D scatter plots [6]. The technique produces a single scatter plot for all combinations of two different dimensions. As shown in Figure 6 the scatter plots are arranged in a matrix-like layout. The scatter plot in the i-th row plots the i-th dimension on the vertical axis and the j-th column plots the j-th dimension on the horizontal axis. This means that the plots in the upper right triangle can be transformed by linear transformations to the respective plots in the lower left triangle, such that only a triangular matrix would suffice to plot all available information. However, it is common practice to plot the full matrix. The diagonal of the matrix cannot contain any plots as plotting the same dimension

against itself would not yield any usable results and thus is a perfect place to contain the label of each dimension. In total, when visualizing p dimensions, a total of

$$n = \sum_{i=1}^{p-1} i = \frac{1}{2}(p-1) \cdot p \tag{6}$$

scatter plots are required in each triangle half of the matrix [6]. Thus, the number of required scatter plots increases quadratically with the number of dimensions, which does not scale from a human perception perspective.

In general, the various scatter plots reveal the correlation between the different dimensions. However, only the correlation between two dimensions is perceivable. In addition to that, when dealing with discrete dimensions scatter plots may trick the viewer as the data points seem to be aligned in rows or columns compared to a continuous data set.

The scatter plot matrix can also be used to easily spot outliers between the different dimensions, although it is impossible to compare outliers in multiple scatter plots without interaction techniques. Brushing scatter plots [4] is an interaction technique that can help identifying outliers and analyzing clusters. When brushing, certain data points in a single scatter plot of the matrix are selected and then colored differently in all scatter plots in the matrix. Thus, an analysis of the behavior of a subset of the data points for different dimensions is possible.

Elmqvist et al. [9] describe a navigation technique for scatter plot matrices. In fact, the approach uses the scatter plot matrix as navigational element where each scatter plot serves as "thumbnail" for the respective dimensions. The user can then intuitively navigate through the different scatter plots to compare and correlate the different dimensions.

3.3 Parallel Coordinate Plots

Parallel coordinate plots [19, 20] are capable of visualizing an arbitrary number of dimensions. Compared to a scatter plot, the axes of the dimensions are not aligned in an orthogonal layout, but rather parallel to each other and evenly spaced. Hence, the visualization scales linearly with the number of dimensions to visualize. There are no assumptions to the ordering of the coordinates within the visualization. Ideally, when exploring the data the plot is interactive and allows to rearrange the coordinates to experimentally find meaningful visualizations [21]. In a parallel coordinate plot data points are represented by polygon lines passing through the respective values on each coordinate. The geometry and the alignment of the polygon lines can represent the correlation between two adjacent coordinates. There are specific visual patterns for various correlations and manifolds that can be used to identify patterns within the data [34].

Parallel coordinate plots can be enriched with various visualization details. For example, each coordinate can visualize box plots, histograms or violin plots to better visualize the quantity of the data for the coordinate [31]. In addition to that, the polygon lines may be curved [14] (instead of straight lines) or brushed [16, 29] to better distinguish single data points from each other. Interaction techniques in parallel coordinate plots support the perception of the correlations of the underlying data Artero et al. [2], Panagiotidis et al. [23].

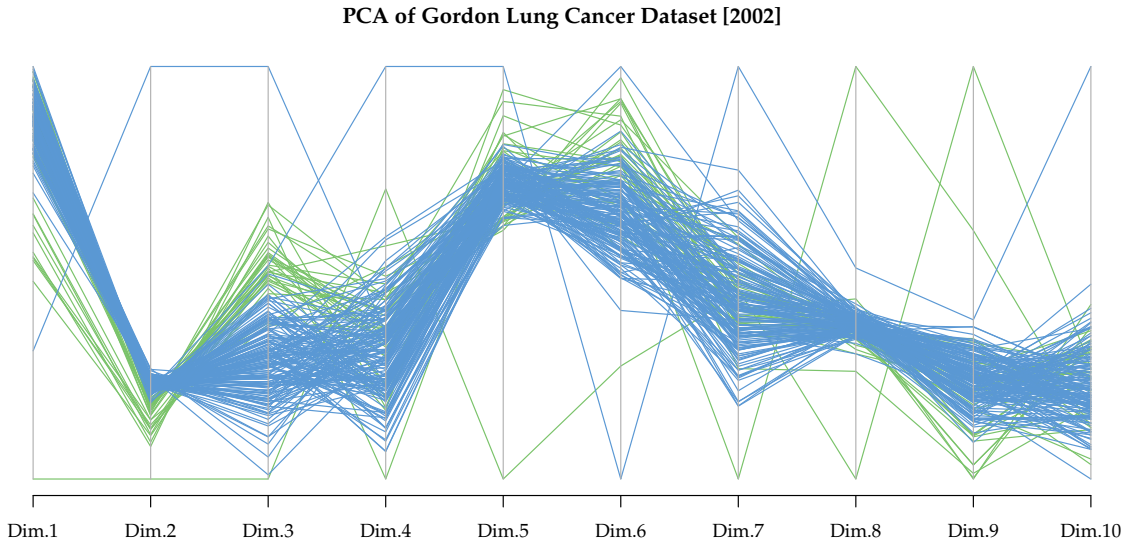


Figure 7: Principal component analysis performed on a lung cancer data set by Gordon et al. [13] using data of 181 samples. For each sample, 12533 dimensions were measured. The ten principal components with the highest variance are visualized in a parallel coordinate plot. The first four principal components are also visualized as a scatter plot in Figure 6.

4 Evaluation of Existing Techniques

4.1 Dimension Reduction Techniques

In Section 2 four dimension reduction techniques with different characteristics were introduced. While the introduced techniques are among the commonly used techniques, a sheer amount of other dimension techniques exists. Currently, there are 34 different dimensionality reduction techniques implemented in the "Matlab Toolbox for Dimensionality Reduction"[1]. The dimensionality reduction techniques were developed for different purposes and different data in mind. When using a dimension reduction method one should consider the different drawbacks of the various methods available and choose a method that suits the data being analyzed. Most techniques also have parameters that can be optimized. van der Maaten et al. [33] provides an extensive comparison of existing techniques, their parameters, their performance and their error on different data sets.

In addition to that, techniques can be combined in a pipeline to further leverage the strengths of each individual technique. For example, it is a common practice to estimate the intrinsic dimensionality d of the given data set and then use a fast dimension reduction technique to reduce the dataset to its intrinsic dimension. PCA is a candidate for reducing the data to its intrinsic dimension, as the first d principal components are kept and the remaining principal components only contain very little additional information. As such, the information loss is minimal and can be estimated by

[1] http://lvdmaaten.github.io/drtoolbox/

summing the variances of the dropped principal components. After having a data set in its intrinsic dimension, expensive and powerful techniques take less computational time to further reduce the data to its intended dimension.

4.2 Visualization Techniques

In Section 3 three visualization techniques were introduced. This section compares the techniques in terms of ease of perception, enrichment through interaction and scalability of dimensions.

Scatter plots have an easy perception, as axes and positioning of data points are clear. Interaction techniques are not required, but can support the user with brushing, for example. However, scatter plots can only display two dimensions at any time which imposes a strong limitation on the design of experiments and data analysis.

It takes effort and time to perceive scatter plot matrices, as there are many scatter plots to analyze. In addition to that, the axes and correlations need to be closely examined. Interaction techniques are not necessarily required, but suggested, as the amount of data displayed can be overwhelming and not answer specific questions. The scatter plot matrix scales quadratically with the number of displayed dimensions, which is a strong limitation when displaying more dimensions.

Parallel Coordinates have a complex perception, as the correlations between the different dimensions are more difficult to interpret. The ordering of the columns also greatly influences the perception. Interaction techniques are recommended for parallel coordinate plots, as reordering of the columns or brushing a subset of the data can reveal important information. Despite that, the parallel coordinate plot scales linearly with the number of displayed dimensions.

All in all, each visualization technique has its strengths and drawbacks which have to be considered when choosing the technique to use. It is also possible to combine the techniques by starting with techniques that are easily perceivable and changing to enhanced techniques when detailed analyses are required.

5 Conclusions and Future Directions

In general, it is difficult to deal with high dimensional data as the human brain cannot perceive more than three dimensions. However, real world data often is high dimensional and analysis of the data is required. Research has since developed various methods to reduce the dimensions and visualize the data. While visualization is only one application of dimension reduction techniques, there are several other areas where an initial dimension reduction step greatly simplifies the task at hand.

While a lot of techniques have proven to be useful for visualization purposes it is still difficult to choose the right method and the right parameters for the method. While there are quality metrics for high dimensional data visualization, most metrics are not centered on the user, but rather on the technical side. However, the user is the most important factor in reducing and visualizing high dimensional data, as he needs to understand and analyze the result. Thus, future research should focus on simplifying the visualization of high dimensional data for users.

References

[1] Abdi, H. and Williams, L. J. (2010). Principal component analysis. *Wiley Interdisciplinary Reviews: Computational Statistics*, 2(4):433–459.

[2] Artero, A. O., de Oliveira, M. C. F., and Levkowitz, H. (2004). Uncovering clusters in crowded parallel coordinates visualizations. In *Information Visualization, 2004. INFOVIS 2004. IEEE Symposium on*, pages 81–88. IEEE.

[3] Balasubramanian, M. and Schwartz, E. L. (2002). The isomap algorithm and topological stability. *Science*, 295(5552):7–7.

[4] Becker, R. A. and Cleveland, W. S. (1987). Brushing scatterplots. *Technometrics*, 29(2):127–142.

[5] Burges, C. J. (2009). Dimension reduction: A guided tour. *Machine Learning*, 2(4):275–365.

[6] Carr, D. B., Littlefield, R. J., Nicholson, W., and Littlefield, J. (1987). Scatterplot matrix techniques for large n. *Journal of the American Statistical Association*, 82(398):424–436.

[7] Choi, H. and Choi, S. (2007). Robust kernel isomap. *Pattern Recognition*, 40(3):853–862.

[8] De Oliveira, M. C. F. and Levkowitz, H. (2003). From visual data exploration to visual data mining: a survey. *Visualization and Computer Graphics, IEEE Transactions on*, 9(3):378–394.

[9] Elmqvist, N., Dragicevic, P., and Fekete, J.-D. (2008). Rolling the dice: Multidimensional visual exploration using scatterplot matrix navigation. *Visualization and Computer Graphics, IEEE Transactions on*, 14(6):1539–1148.

[10] Engel, D., Hüttenberger, L., and Hamann, B. (2011). A survey of dimension reduction methods for high-dimensional data analysis and visualization. In *Visualization of Large and Unstructured Data Sets: Applications in Geospatial Planning, Modeling and Engineering-Proceedings of IRTG 1131 Workshop 2011*, volume 27, pages 135–149. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

[11] Fodor, I. K. (2002). A survey of dimension reduction techniques.

[12] Ghodsi, A. (2006). Dimensionality reduction a short tutorial. *Department of Statistics and Actuarial Science, Univ. of Waterloo, Ontario, Canada*.

[13] Gordon, G. J. G., Jensen, R. V. R., Hsiao, L.-L. L., Gullans, S. R. S., Blumenstock, J. E. J., Ramaswamy, S. S., Richards, W. G. W., Sugarbaker, D. J. D., and Bueno, R. R. (2002). Translation of Microarray Data into Clinically Relevant Cancer Diagnostic Tests Using Gene Expression Ratios in Lung Cancer and Mesothelioma. *Cancer Research*, 62(17):4963–4967.

[14] Graham, M. and Kennedy, J. (2003). Using curves to enhance parallel coordinate visualisations. In *Information Visualization, 2003. IV 2003. Proceedings. Seventh International Conference on*, pages 10–16. IEEE.

[15] Grinstein, G., Trutschl, M., and Cvek, U. (2001). High-dimensional visualizations. In *Proceedings of the Visual Data Mining Workshop, KDD*.

[16] Hauser, H., Ledermann, F., and Doleisch, H. (2002). Angular brushing of extended parallel coordinates. In *Information Visualization, 2002. INFOVIS 2002. IEEE Symposium on*, pages 127–130. IEEE.

[17] Hinton, G. E. and Roweis, S. T. (2002). Stochastic neighbor embedding. In *Advances in neural information processing systems*, pages 833–840.

[18] Hoffman, P. E. and Grinstein, G. G. (2002). A survey of visualizations for high-dimensional data mining. *Information visualization in data mining and knowledge discovery*, pages 47–82.

[19] Inselberg, A. (1985). The plane with parallel coordinates. *The Visual Computer*, 1(2):69–91.

[20] Inselberg, A. and Dimsdale, B. (1991). Parallel coordinates. In *Human-Machine Interactive Systems*, pages 199–233. Springer.

[21] Kosara, R., Bendix, F., and Hauser, H. (2006). Parallel sets: Interactive exploration and visual analysis of categorical data. *Visualization and Computer Graphics, IEEE Transactions on*, 12(4):558–568.

[22] Kruskal, J. B. (1964). Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika*, 29(1):1–27.

[23] Panagiotidis, A., Bosch, H., Koch, S., and Ertl, T. (2011). Edgeanalyzer: Exploratory analysis through advanced edge interaction. In *System Sciences (HICSS), 2011 44th Hawaii International Conference on System Sciences*, pages 1–10. IEEE.

[24] Roweis, S. T. and Saul, L. K. (2000). Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326.

[25] Saul, L. K. and Roweis, S. T. (2000). An introduction to locally linear embedding. *unpublished. Available at: http://www. cs. toronto. edu/˜ roweis/lle/publications. html.*

[26] Sedlmair, M., Munzner, T., and Tory, M. (2013). Empirical guidance on scatterplot and dimension reduction technique choices. *Visualization and Computer Graphics, IEEE Transactions on*, 19(12):2634–2643.

[27] Shang, F., Jiao, L., Shi, J., and Chai, J. (2011). Robust positive semidefinite l-isomap ensemble. *Pattern Recognition Letters*, 32(4):640–649.

[28] Shi, L.-k., He, P.-L., Liu, B., Fu, K., and Wu, Q. (2005). A robust generalization of isomap for new data. In *Machine Learning and Cybernetics, 2005. Proceedings of 2005 International Conference on*, volume 3, pages 1707–1712. IEEE.

[29] Siirtola, H. and Räihä, K.-J. (2006). Interacting with parallel coordinates. *Interacting with Computers*, 18(6):1278–1309.

[30] Tenenbaum, J. B., De Silva, V., and Langford, J. C. (2000). A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323.

[31] Theus, M. (2003). Interactive data visualization using mondrian. *Journal of Statistical Software*, 7(11):1–9.

[32] Van der Maaten, L. and Hinton, G. (2008). Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(2579-2605):85.

[33] van der Maaten, L. J., Postma, E. O., and van den Herik, H. J. (2009). Dimensionality reduction: A comparative review. *Journal of Machine Learning Research*, 10(1-41):66–71.

[34] Wegman, E. J. (1990). Hyperdimensional data analysis using parallel coordinates. *Journal of the American Statistical Association*, 85(411):664–675.

[35] Wold, S., Esbensen, K., and Geladi, P. (1987). Principal component analysis. *Chemometrics and intelligent laboratory systems*, 2(1):37–52.