Eurographics Conference on Visualization (EuroVis) (2014)
N. Elmqvist, M. Hlawitschka, and J. Kennedy (Editors)

Interactive Design of Multidimensional Data Projection Layout

Vladimir Molchanov and Lars Linsen

Jacobs University, Bremen, Germany

Abstract

Projection methods support effective visualizations of multidimensional data. Linear projections are an important subclass, as they allow for interactive visual exploration of the data space and feature sensitivity analysis. The user interaction is usually based on an iterative modification of the projection matrix elements, for example, by the use of a star coordinate widget. However, such interaction mechanisms become inefficient with increasing number of dimensions. We propose to adapt the projection matrix by allowing the user to directly operate on the projection domain. The desired configuration of the projection layout is obtained by adjusting the positions of (freely chosen) control points. The update of the projection matrix is performed according to the interactive modifications by computing a least-square solution of a linear equation system. Changes can be tracked easily using animation. We apply our method to classified multidimensional data and demonstrate that our approach allows for an intuitive and effective design of projections with desired properties like improved class segregation or reduced clutter.

Categories and Subject Descriptors (according to ACM CCS): I.4.7 [Computer Graphics]: Feature Measurement—Projections

1. Introduction

High-dimensional data visualization techniques have been investigated for decades. In particular, many dimensionality reduction methods (prominent examples being principal component analysis (PCA) [Jol86] or multidimensional scaling (MDS) [BG10]) exist, which map the high-dimensional attribute space to a low-dimensional visual domain. Each method can be characterized according to its characteristics such as computational complexity, preservation of distances, cluster segregation, or clutter avoidance, which determine the choice of method for concrete application tasks.

Linear projections (such as PCA) have a number of nice properties such as algorithmic simplicity, low computation costs, and easy interpretation. They do not depend on the data in the sense that dynamically added points do not require recomputation of the projection. Most importantly though, any linear projection operator has a compact matrix representation, which can be analyzed, re-used for direct application to new datasets, visualized, and interactively modified. The possibility to interactively modify a projection allows for adapting the data layout on the visual domain and performing feature sensitivity analysis. Star-coordinates widgets [Kan00, Kan01, TM03, BCL*06, MFL13a, MFL13b, LT13]) have proven to be effective in this regard due to their intuitiveness: Changing an axis' position results in changing values in a single column of the projection matrix, thus, the user controls the contribution of the selected attribute.

Most projective methods produce a non-editable resulting placement of data objects aiming at optimization of some criterion. However, in some situations, a user-defined layout is desired, for instance, when separation of a particular cluster is much more important than separation of other groups or when placement of certain data objects at a preferable position on the projection view is required. Teoh and Ma [TM03] proposed interactive classification with a star-coordinates representation of the projection matrix. However, finding the desired layout can be very tedious, as the user may have to iteratively change the positions of all widget axes and remember the effect of these interactions. Thus, it becomes impractical for datasets with tens of dimensions, which are not rare in modern simulations and measurements.

We propose a novel technique that overcomes this drawback by steering the data distribution in the projected view by interactively changing control point positions. Our algorithm then updates the projection matrix in order to optimally satisfy the user-defined configuration. The update of

the projection matrix is preformed iteratively by computing a least-square (LS) solution of an overdetermined system of linear equations. The user can easily track and explore the modification by navigating over intermediate results of the iterations (including an undo-operation). We use coordinated views of the projection result and a star-coordinates widget, which allows for the immediate investigation of which coordinate axes changed their lengths and orientations. The simultaneous update of projection and star-coordinates views allows for an analysis on which attributes are important and how their weighting should be changed in order to approximate the desired layout.

2. Related Work

Star coordinates were first introduced by Kandogan [Kan00, Kan01]. The proposed interactive visualization technique allowed for gaining insight into hierarchically clustered datasets, analyzing correlation of multiple dimensions, finding trends and outliers, etc. Teoh and Ma [TM03] used the star-coordinates method to construct a decision tree. The user interacts with the star coordinates producing a set of projections, each separating a certain class of data objects. Molchanov et al. [MFL13a] proposed a technique to compute continuous representation of star-coordinates projection for volumetric multi-attribute datasets. Recently, Lehmann and Theisel [LT13] restricted star coordinates to orthographic projections, which guarantee that proximity of points in the high-dimensional space is conserved in the projection. In particular, this prevents significant deformation of cluster shapes when mapping to the visual domain. We do not limit our method to orthographic projection, as it would contradict to the main goal of user-defined layout design of projections. Wang et al. [WRM13] proposed a N-D touchpad polygon to control the parameters of non-axis aligned scatterplots. The projection plane axes are represented there as points using barycentric coordinates.

3. Background: Star-coordinates Widget

Any linear projection of a d-dimensional attribute space onto a 2-dimensional visual space can be represented as a $2 \times d$ matrix L. The columns of L consist of the coordinates of the images of the attribute space basis vectors in the visual domain. Since the image of the origin is always the origin, one can interpret the columns of L consisting of two entries as coordinates of the star-coordinates axes. This representation is unique and invertible, i.e., any configuration of basis vectors in the star-coordinates widget corresponds to a projection matrix.

The star-coordinates widget can be used for interactive analysis of multidimensional data: When the user changes the position of a certain axis of the widget, the column of the projection matrix corresponding to the selected axis gets assigned new entries. The original data are then re-projected to the linked visual domain using the updated matrix L. In particular, if the end point of an axis is put to the origin of the star coordinates, the contribution of the feature vanishes. For example, a standard 2D scatterplot can be obtained by selecting only two attributes with non-vanishing contributions and setting their axes to be orthonormal. Interaction with the star-coordinates widget tells the user, which attributes of the data are correlated, are negligible, or have the most descriptive power. If a classification of the data is known, it is possible to find projections, which perform best in terms of class separation in the visual domain, and determine a subset of attributes for better characterization of classes.

4. Approach

The star-coordinates widget allows for direct manipulation of the projection matrix. If the goal is to achieve an optimal configuration, which, for example, separates some class best (like in [TM03]) or places classes in a desired order (like in [MFL13a]), it may take a significant amount of time to define a proper projection, especially if the number of attributes is large. We propose to invert the process by allowing the user to specify the desired configuration in the projected view and let the system calculate the best-fitting projection matrix. The user input is provided in the form of control point placement (drag-and-drop using a mouse). The projection matrix recalculation is based on an LS solution of an overdetermined system of linear equations.

A control point is an image of a particular data sample in the projection space. This sample can be arbitrarily chosen by the user. If a classification of data is given, we propose the medians of classes to be the selected samples. Their images are highlighted for the user interaction.

Since the data attributes are often very different in their nature, range, and distribution, it is common practice to normalize data in each dimension before applying a projection. In our tests, we map each feature value to the interval $[0, 1]$ based on its minimal and maximal values in a preprocessing step. Also, we prefer to initialize the star-coordinates widget layout to correspond to the PCA result of the normalized data. This configuration is a more natural choice for the default state than a uniform distribution of axes along the unit circle as proposed in [MFL13a].

5. Iterative LS Solution

It is a common demand to provide a good separation of clusters or reduce the classes' overlap in the projection. If a data classification is known a priori, we define images $\mathbf{p}_i$ of the clusters medians $\mathbf{m}_i$, $i = 1, \ldots, k$, as control points, i.e., $L\,\mathbf{m}_i = \mathbf{p}_i$. The user can then interact with any of them or may also pick other/additional control points. If the user changes the position of a control point corresponding to a median of a class (or cluster), we interpret this input as the following: The class, whose median's image was moved, should be

shifted to the specified location, while other classes should ideally keep their position. Since it is, in general, not possible to satisfy all these conditions precisely, we derive an LS solution of the problem.

Without loss of generality, we assume that the control point $\mathbf{p}_1$ corresponding to the median $\mathbf{m}_1$ changed its position to $\mathbf{q}$. We form the system of linear equations

$$L'(\mathbf{m}_1, \mathbf{m}_2, \ldots, \mathbf{m}_k) = (\mathbf{q}, \mathbf{p}_2, \ldots, \mathbf{p}_k),$$

or for short $L'M = P$, where matrix L' is unknown. Here, we assumed that the system of linear equations is not underdetermined, i.e., $k \geq d$. If k is not large enough, we add random samples to the set of classes' medians. The LS solution to the problem above is given by $L' = PM^T (MM^T)^{-1}$, where $(.)^T$ denotes matrix transposition and $(.)^{-1}$ matrix inversion, which can be computed via Cholesky decomposition.

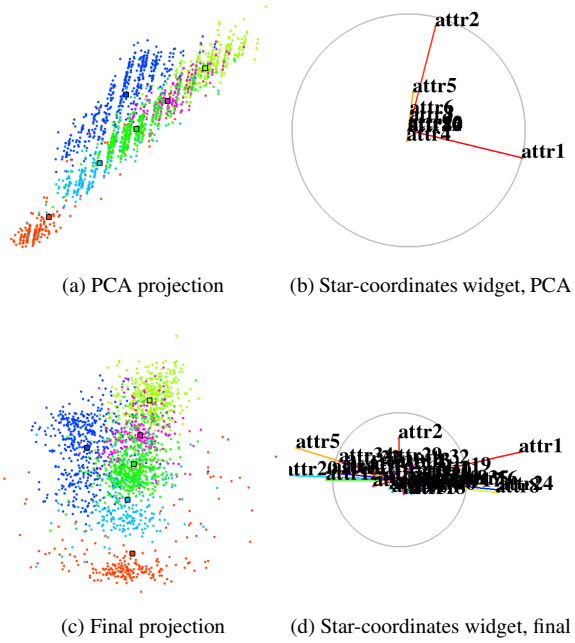
For large values of k, the number of equations which are satisfied with $L' \equiv L$ is large. Thus, the solution L' is close to initial projection matrix L. For small k, the matrix MM^T may become ill-conditioned. Adding more control points (in our tests, we picked $k \in [1.5d, 2d]$) can, in general, resolve the issue.

Since the updated projection matrix L' may result in a significant change of the projection when compared to the original distribution of mapped samples, it may be hard for the user to understand and evaluate the executed transformation. Therefore, we propose to approach the solution L' iteratively by computing a sequence of matrices starting from L and converging to L'. The user specifies the number of iterations n and can navigate over the sequence of computed matrices using a slider. This leads to a smooth transition of the projection from its original state towards the final configuration. The user may terminate at any intermediate step and may even return to the initial projection corresponding to L. The iterative process is defined by

$$L_{i+1} M = L_i M + \mathrm{diag}(\mathbf{w}_i)\ (P - L_i M), \qquad i = 0, \ldots, n-1,$$

where $L_0 \equiv L$, $L' = L_n$, $\mathbf{w}_i \in \mathbb{R}^k$, and all entries of the diagonal relaxation matrix $\mathrm{diag}(\mathbf{w}_i)$ belong to the range $[0, 1]$. (Note that the first elements of vectors $\mathbf{w}_i$ correspond to the modified control point.) In our tests, we used $\mathbf{w}_i = \left(\frac{i+1}{n}, C, \ldots, C\right)$ with a constant $C \in [0, 1]$. Large C penalize change of position of static control points. Low values of the first element prevent large distortions and allow the user to preserve the mental map during navigation.

When navigating over computed projections L_i, the user can evaluate the change of the projection quality both visually and numerically. E.g., relevant quality measures (for instance, silhouette coefficient [TSK05], stress measure [BG10], or correlation coefficient [GZZ05]) can be computed and shown for each intermediate L_i. Then, the user may stop navigation at the projection corresponding to the optimal numerical value and continue interactions.



(a) PCA projection (b) Star-coordinates widget, PCA

(c) Final projection (d) Star-coordinates widget, final

Figure 1: PCA projection of "Statlog" dataset (a) and corresponding configuration of the star-coordinates widget (b). Samples are colored according to given classification. Control points are shown as black squares. Final projection layout (c) and configuration of star-coordinates widget (d).

6. Results

In the first experiment, we use the "Statlog" dataset [BL13], which contains multi-spectral values of pixels in 3×3 neighborhoods in a satellite image. There are 2.000 samples with 36 attributes and 6 identified classes. We start with the PCA layout (Figure 1a) and the respective star-coordinates widget (Figure 1b). The unit circle is shown in gray and colors of samples correspond to different classes. The task is to reduce clutter (occlusion of samples) via more efficient distribution of classes in the visual domain. The classes' ordering and separation should not be significantly affected.

The high number of attributes does not allow for an efficient use of the star-coordinates widget to steer the projection. In contrast, since the number of classes is low, the task can be completed very quickly when interacting with the medians of classes as control points (black squares in figures). The results are shown in Figure 1c and the respective configuration of the star-coordinates widget in Figure 1d. Our interactive technique can best be observed in the accompanying video, of which the paper only shows some keyframes.

In the second test, we demonstrate how our approach helps to identify essential features for given classes. We applied our technique to the "Ecoli" dataset [BL13] containing

protein localization sites. There are 336 samples with 7 attributes forming 8 classes. Starting from PCA (Figures 2a and 2b), the user changes positions of medians of some classes (shown as black arrows). The resulting changes of the projection matrix can be seen in the star-coordinates widget. In particular, moving the green class down affects the axes corresponding to attributes **alm1** and **alm2**. The position of the magenta class is fully defined by attribute **chg**, which does not affect positions of images of other classes. Finally, horizontal shifting of the class depicted in light-green affects the position of all axes of the star-coordinates widget. Since the number of attributes is not that high for this example, one could discover these findings also by the star-coordinates widget manipulations, but at the cost of longer interaction. The goal-oriented re-computation of the projection matrix was found to be more intuitive and efficient.

7. Discussion and Conclusion

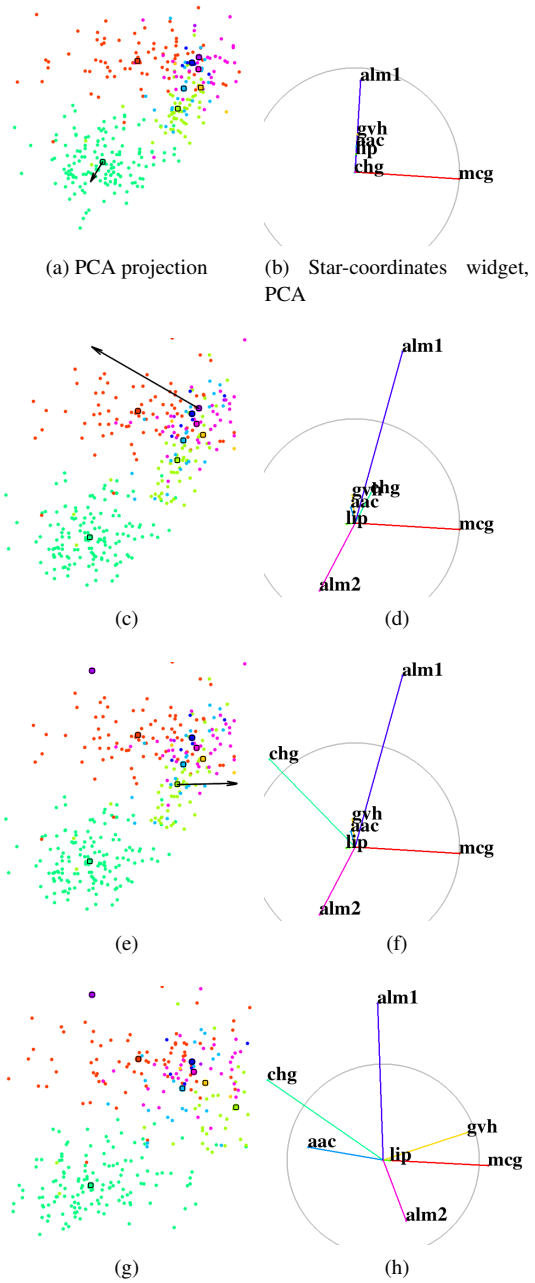
A technique for steering the projection layout by the user has been proposed. The approach is currently applied and tested for a space of linear projections. It is efficient for any number of data attributes and, therefore, overcomes the limitation of interactions based on the star-coordinates widget (which is still supported). A star-coordinates display is used for analyzing dependencies of classes and attributes.

An update of the projection matrix is performed according to the user interaction with a control point in the projected view. We iteratively find an approximation to the desired projection matrix by solving an overdetermined system of linear equations in an LS sense. The number of equations shall be not less than the number of data attributes. If necessary, additional control points can be chosen. Relaxation vectors $\mathbf{w}_i$ are optional and have a limited influence on the dynamic of the projection matrix change. Simplicity of computations allows for using the proposed algorithm in real-time interactive applications. The user interactions are quite intuitive and may be reverted.

An alternative approach to our method would be to directly model the change of control point $\mathbf{p} = (p_1, p_2)$ with $L\,\mathbf{a} = \mathbf{p}$ to a new position $\mathbf{q} = (q_1, q_2)$ by $L'\,\mathbf{a} = \mathbf{q}$. The problem has an infinite number of solutions L'. Assuming that it is the user's wish to map neighbors of $\mathbf{a} = (a_1, \ldots, a_d)$ close to position $\mathbf{q}$, we define $\triangle L = L' - L$ and set

$$\triangle L = \frac{1}{(\mathbf{a}, \mathbf{a})} \begin{pmatrix} a_1 & \ldots & a_d \\ a_1 & \ldots & a_d \end{pmatrix} \begin{pmatrix} q_1 - p_1 & 0 \\ 0 & q_2 - p_2 \end{pmatrix},$$

where $(\cdot,\cdot)$ stands for inner product. Then, $\triangle L\ \mathbf{a} = \mathbf{q} - \mathbf{p}$, i.e., the control point is mapped exactly to the position specified by the user. This alternative solution is unconditionally stable and parameter-free. However, the method has limited efficiency for cluster separation tasks. We concluded that the proposed LS approach is, in general, preferable.



(a) PCA projection (b) Star-coordinates widget, PCA

(c) (d)

(e) (f)

(g) (h)

Figure 2: Projections of "Ecoli" dataset. Samples are colored according to given classification. Control points are shown as black squares. User interactions are shown as black arrows. Resulting changes in star-coordinates display allow to identify main features responsible for projection of selected class.

Acknowledgments. This research was funded by DFG via grant LI 1530/6-2. We thank Petar Dobrev and Fernando Paulovich for discussions.

References

[BCL*06] BORDIGNON A., CASTRO R., LOPES H., LEWINER T., TAVARES G.: Exploratory visualization based on multidimensional transfer functions and star coordinates. In *Sibgrapi 2006 (XIX Brazilian Symposium on Computer Graphics and Image Processing)* (Manaus, AM, oct 2006), IEEE, pp. 273–280. doi:10.1109/SIBGRAPI.2006.17. 1

[BG10] BORG I., GROENEN P. J. F.: *Modern Multidimensional Scaling Theory and Applications*, 2nd edition ed. Springer Series in Statistics. Springer, 2010. 1, 3

[BL13] BACHE K., LICHMAN M.: UCI machine learning repository, 2013. URL: http://archive.ics.uci.edu/ml. 3

[GZZ05] GENG X., ZHAN D.-C., ZHOU Z.-H.: Supervised nonlinear dimensionality reduction for visualization and classification. *IEEE Transactions on Systems, Man, and Cybernetics, Part B 35*, 6 (2005), 1098–1107. 3

[Jol86] JOLLIFFE I. T.: *Pincipal Component Analysis*. Springer-Verlag, 1986. 1

[Kan00] KANDOGAN E.: Star coordinates: A multi-dimensional visualization technique with uniform treatment of dimensions. In *Proceedings of IEEE Information Visualization Symposium (Hot Topics)* (2000), pp. 4–8. 1, 2

[Kan01] KANDOGAN E.: Visualizing multi-dimensional clusters, trends, and outliers using star coordinates. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining* (New York, NY, USA, 2001), KDD '01, ACM, pp. 107–116. URL: http://doi.acm.org/10.1145/502512.502530, doi:10.1145/502512.502530. 1, 2

[LT13] LEHMANN D. J., THEISEL H.: Orthographic star coordinates. *IEEE Trans. Vis. Comput. Graph. 19*, 12 (2013), 2615–2624. 1, 2

[MFL13a] MOLCHANOV V., FOFONOV A., LINSEN L.: Continuous representation of projected attribute spaces of multifields over any spatial sampling. *Computer Graphics Forum 33*, 3 (Jun 2013), 301–310. doi:10.1111/cgf.12117. 1, 2

[MFL13b] MOLCHANOV V., FOFONOV A., LINSEN L.: Frequency-based progressive rendering of continuous scatterplots. *Journal of WSCG 21*, 1 (July 2013), 49–58. 1

[TM03] TEOH S. T., MA K.-L.: Starclass: Interactive visual classification using star coordinates. In *SDM* (2003), Barbará D., Kamath C., (Eds.), SIAM. 1, 2

[TSK05] TAN P.-N., STEINBACH M., KUMAR V.: *Introduction to Data Mining, (First Edition)*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2005. 3

[WRM13] WANG B., RUCHIKACHORN P., MUELLER K.: SketchPadN-D: WYDIWYG sculpting and editing in high-dimensional space. *IEEE Transactions on Visualization and Computer Graphics 19*, 12 (2013), 2060–2069. doi:http://doi.ieeecomputersociety.org/10.1109/TVCG.2013.190. 2