Joint EUROGRAPHICS - IEEE TCVG Symposium on Visualization (2004)
O. Deussen, C. Hansen, D.A. Keim, D. Saupe (Editors)

Visual Component Analysis

Wolfgang Müller[1] and Marc Alexa[2]

[1] Department of Computer Science, PH Ludwigsburg, Germany
[2] Department of Computer Science, Darmstadt University of Technology, Germany

Abstract

We propose to integrate information visualization techniques with factor analysis. Specifically, a principal direction derived from a principal component analysis (PCA) of the data is displayed together with the data in a scatterplot matrix. The direction can be adjusted to coincide with visual trends in the data. Projecting the data onto the orthogonal subspace allows determining the next direction. The set of directions identified in this way forms an orthogonal space, which represents most of the variation in the data. We call this process visual component analysis (VCA). Furthermore, it is quite simple to integrate VCA with clustering. The user fits poly-lines to the displayed data, and the poly-lines implicitly define clusters. Per-cluster projection leads to the definition of per-cluster components.

Categories and Subject Descriptors (according to ACM CCS): I.3.6 [Computer Graphics]: Visual Data Mining, Information Visualization

1. Introduction

An important requirement in the exploratory analysis of unknown data is the identification of main factors accounting for variances in the data. In general, identifying these factors leads to a more compressed description of correlations in the data and, thus, for a better understanding of the underlying processes that generated the data.

In statistics and data mining, there exist a number of approaches to identify such main factors. The most important of these techniques are factor and principal component analysis (PCA [Jol86]). While factor analysis targets primarily the identification of main factors and their correlation only, principal component analysis also provides information about the kind of influence it takes on the data. Usually, this influence is described in terms of a direction in the multivariate data field.

In practice, the application of component analysis and other traditional forms of factor analysis have proven to be very difficult [Cli87]. One has to understand the underlying mathematics in detail to interpret the meaning of the principal components and directions. For instance, the principal component analysis not only provides principal component vectors but also the influence of a component in terms of so-called factor loadings. These loadings are usually taken for granted as a direct measure for the importance of principal components. However, it has been shown that the situation is not that easy and that the real influence of a principal component has to be verified on a case-by-case basis.

A fundamental problem of all automatic component analysis approaches is the consideration of all data elements and all variables. While the data might have clear trends in some of the variables others might be uncorrelated and noisy. Because automatic component analysis has to treat all variables equally, trends in few variables are not represented accurately due to possible noise in other variables. A particular incarnation of this problem in the PCA is that it sometimes produces directions orthogonal to what seems to be the dominant direction. Indeed, statistical theory advises the inspection of the data and possible rotation of principal direction if found to be appropriate.

Another problem particular to PCA is the assumption of global linearity. Oftentimes, processes generating data could be well described piecewise linear, but a global linear approximation of the trends is inadequate. For a piecewise linear description the multi-variate space should first be clustered into components that could be well-approximated using linear spaces. Note that automatic clustering without a-priori knowledge about the number of clusters is difficult. As the PCA itself sometimes generates misleading results, it is

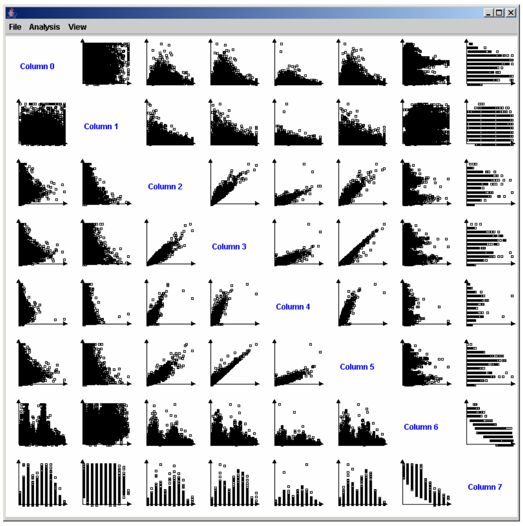

Figure 1: *A scatterplot matrix*

even more unlikely that it could be integrated with clustering techniques to yield a fully automatic data mining solution that correctly identifies trends, noise, outliers, and clusters.

In this paper, we follow the idea of Visual Data Mining [Shn02, KW02] and propose a visual verification, correction and steering of the determination of principal components and clusters. In particular, we use scatterplot matrices [WB97, Cle94] for the visualization of data, which we enhance by the presentation of principal directions. Moreover, interactive techniques to modify the principal components are provided.

We apply scatterplots in combination with line graphs and panel matrices for the visualization of data and principal components since they are well suited to display large amounts of data. Many other popular visualization techniques for multi-variate data (e.g. glyphs [Che73, PG88, Bed90, RAEM94], parallel coordinates [Ins84, ID90], pixel displays [KKA95, Kei00]) lack the possibility to display principal directions.

2. State of the Art

It is long known that defects in the principal components could be corrected by visually inspecting them [Cli87]. This applies obviously only to low-dimensional data and few components so far. We couple the idea with *interactive* information visualization techniques (for an overview on such techniques, see [CMS99]).

2.1. Scatterplot and panel matrix

A scatterplot displays bivariate data elements as markers in a two-dimensional coordinate system [CM84]. Panel matrices [Cle94, WB97] are a means to visualize multi-parameter data in terms of a set of scatterplot panels, organized in such that each combination of two parameters is visualized by an individual scatterplot graph (see Figure 1 for an example). The panel matrix has proven to be one of the most powerful techniques to analyze multi-parameter data visually. Several extensions have been proposed based on interactive techniques such as brushing and focusing & linking (a tool integrating many of these techniques is XmdvTool [War94]). None of these extensions, however, is directly targeted to support the identification of principal components in the data.

2.2. Parallel coordinates

Parallel coordinates [Ins84, ID90] are another technique which is well-suited to visualize multi-variate data. Here, each data element is depicted in as a line segment in 2d space, intersecting the different parallel data axes at the scale defined by the corresponding attribute value. Parallel coordinates work very well to identify principal components in the data based on line segment bundles in the graph. On the other hand, it is unclear how to display the principal direction, as a line in variable space completely covers all variable values on all axes. The application of parallel coordinates is typically limited to few hundred data elements. For larger data sets, the amount of space required for the display of each individual data element hinders the identification of elements (or principal components).

2.3. Icons, Glyphs, and Pixel Displays

Many visualization techniques for multi-variate data try to find representative visual elements depicting the data elements. Icons (or glyphs) could be constructed in a variety of ways (e.g. see [Che73, PG88, Bed90, RAEM94]). The general idea is to map the variables in the data to axes in the feature space of icons. Pixel displays could be interpreted as specific icons, which map each variable to a pixel using color-coding [KKA95, Kei00]. The icons or pixels could be arranged in different ways, either directly reflecting two variables or minimizing some measure of layout optimization. If the space of different icons is rich, prominent features in the data are clearly visible. As with parallel coordinates, it is unclear how to represent a line in variable space.

2.4. Reducing dimensionality

Our approach to identifying principal components step-by-step using visual feedback could be seen as a way of reducing the number of dimensions (i.e. uncorrelated variables) in the data. In general, the goal of dimension reduction is to identify projections that lead to minimal loss of information. This could be the case when data are highly correlated (e.g. there is a linear mapping from one variable to another) or highly uncorrelated (because the projection would not introduce any loss of information). Note that we are interested exclusively in finding trends.

Dimension reduction has been extensively studied for visualizing multi-variate data, as most visualization techniques have a natural limit on the number of variables they can display. Common techniques (besides factor analysis) are multidimensional scaling [KW78, CC94] and Kohononen maps [Koh95]. Integrating dimension reduction with information visualization has proven to yield to more meaningful visualizations of the projected data [YWRH03].

2.5. Visual Data Mining

While data mining (e.g. [WB98]) and visualization techniques have both been applied for the analysis of data for a while, there are still few examples for an integration from techniques of both areas. Mostly, Visual Data Mining approaches still focus on the application of analytical techniques (e.g. for clustering and segmentation), restricting the visualization as a method for verifying the results (e.g. [FGW01, SD02]). A notable approach towards the integration of information visualization and clustering is HD-Eye [HKW99]. An overview on current techniques in information visualization as well as on current approaches in Visual Data Mining can be found in [KW02].

3. Notation and PCA

Let the multivariate data be described by vectors $\mathbf{x}_i \in \mathbb{R}^d, i \in \{1,\ldots,n\}$ in a linear space of d dimensions. Then the set of data $\{\mathbf{x}_i\}$ is represented by a matrix

$$X = \begin{pmatrix} \mathbf{x}_1 \\ \ldots \\ \mathbf{x}_n \end{pmatrix}, \quad X \in \mathbb{R}^{d \times n} \tag{1}$$

The *mean* $\bar{\mathbf{x}}$ can be written in matrix notation as $\bar{\mathbf{x}} = (1/n,\ldots,1/n)X$. A matrix $\bar{X}$ consisting of mean vectors can be formed with the outer product, i.e. $\bar{X} = (1,\ldots,1)^T\bar{\mathbf{x}}$. The *deviation* of the data from the mean is thus $Y = X - \bar{X}$. Note that Y has zero mean – we have shifted the mean to the origin of the system.

The *variance* of a variable describes how close the data is to the mean in this variable. This is described as the sum of squares of differences to the mean. The *co-variance* is the sum over products of differences between two variables. Both, variances and co-variances, are represented in the *co-variance matrix*

$$T = Y^T Y \ \mathrm{diag}(1/n) = (X - \bar{X})^T (X - \bar{X}) \ \mathrm{diag}(1/n) \tag{2}$$

Given zero mean data Y, we might be interested in certain linear subspaces that represent almost all of the variation in the data. A one-dimensional subspace that represents most of the variation in the data is called *principal component*, and we denote it $\mathbf{c}_1$. Because of the goal to represent the data (and its variation), $\mathbf{c}_1$ is a linear combination of the data vectors $\{\mathbf{x}_i\}$ weighted appropriately with $\mathbf{w}_1 \in \mathbb{R}^n$, i.e. $\mathbf{c}_1 = Y\mathbf{w}_1$. The weight vector represents a line in the space of variables and is called *principal direction*. This principal direction is represented with the unit length vector $||\mathbf{w}_1|| = 1$. Given this normalization, it is feasible to maximize the variance c_1^2 of $\mathbf{c}_1$.

Note that c_1^2 has the matrix representation

$$c_1^2 = \mathbf{c}_1^T \mathbf{c}_1 = \mathbf{w}_1^T Y^T Y \mathbf{w}_1 = \mathbf{w}_1^T T \mathbf{w}_1. \tag{3}$$

To maximize c_1^2 subject to the constraint $||\mathbf{w}_1|| = 1$, we introduce a Lagrange multiplier λ_1 and equivalently maximize $\mathbf{w}_1^T \mathbf{w}_1 - \lambda \mathbf{w}_1^T \mathbf{w}_1$. The maximum can be determined by setting the gradient $T\mathbf{w}_1 - \lambda_1 \mathbf{w}_1$ to zero and solving for $\mathbf{w}_1$. Thus, $\mathbf{w}_1$ is an eigenvector of $Y^T Y$ and because we wanted to maximize, it is the eigenvector corresponding to the largest eigenvalue.

One might want to determine additional components $\mathbf{c}_2,\ldots,\mathbf{c}_z, z < d$, so that $\{\mathbf{c}_k\}$ represents most of the variation in the data. It is natural to ask that these components are mutually uncorrelated. In the linear setting with zero mean data this means the direction vectors are orthogonal. Another look at this is to find first $\mathbf{c}_1$ and then project the data onto the subspace orthogonal to $\mathbf{w}_1$, i.e.

$$Y' = Y - Y\mathbf{w}_1\mathbf{w}_1^T \tag{4}$$

The data in Y' represents Y without the variation that is described by $\mathbf{c}_1$. Computing the principal component of Y_1 yields $\mathbf{c}_2$. Note that $Y'^T Y'$ has the same eigenvectors as $Y^T Y$ except for $\mathbf{w}_1$, which corresponds to a zero eigenvalue in $Y'^T Y'$. Thus, $\mathbf{w}_2$ is an eigenvector of $Y^T Y$ and it corresponds to the second largest eigenvalue. More generally, the $\mathbf{w}_i$ are eigenvectors of $Y^T Y$ sorted according to the eigenvalues.

In practice, we compute the principal components of Y using the singular value decomposition (SVD). The SVD factors Y directly into $Y = C\Lambda^2 W$, where C represents the components $\mathbf{c}_i$ and W the directions $\mathbf{w}_j$.

4. Visual Component Analysis

Factoring given data by means of the PCA can reveal trends in the data only, if the underlying processes generating the data are free from noise. Especially outliers can have significant influence on the principle components because the analysis accounts for all data in the same way. If some knowledge about the process introducing outliers and noise is given, there are mathematical ways to adapt the PCA. In most practical cases, however, we are lacking such information.

The main point of Visual Component Analysis (VCA) is to facilitate the human visual system for identifying principal components. An ideal way of presenting the data and the principal direction visually is a scatterplot matrix (see 2.1). In addition to the data points, the user can interactively modify a two-dimensional projection of the current principal direction.

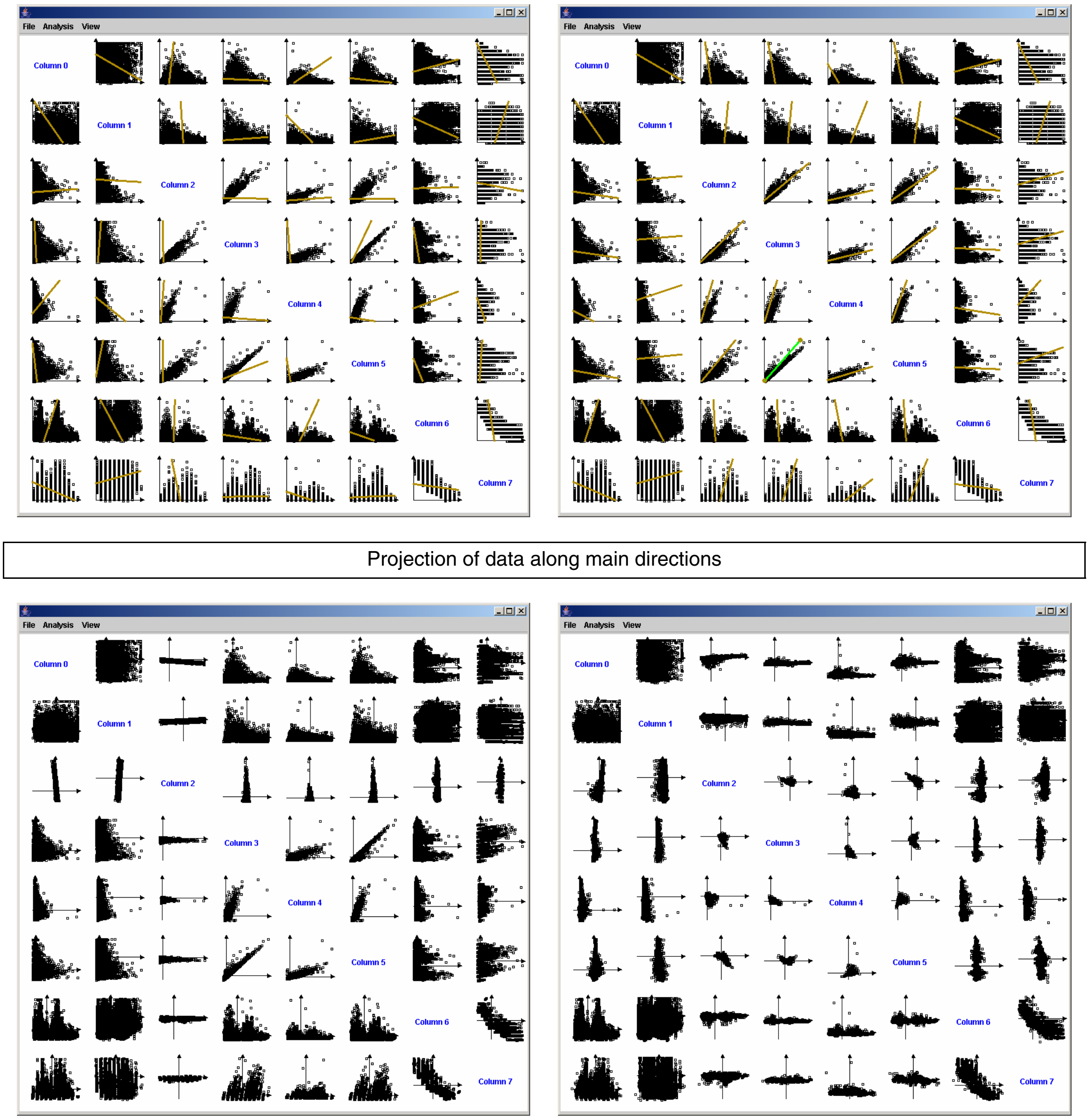

Figure 2: *Principal and visual component analysis of data in a scatterplot matrix. The left column shows the principal direction corresponding to the principal component of the data and the projection along this direction onto the orthogonal subspace. In the right column, the user has corrected the direction based on the trend apparent in the middle block of the panel matrix. In the projection (lower right), the main variation of the data in the middle part is removed.*

Note that many other visualization techniques lack the possibility to display a line, or more generally, a family of data values. In space of icons or glyphs, a line would be represented by infinitely many such objects. It is unclear how to display this family. A line through variable space in parallel coordinates would simply span the whole coordinate system.

The overall process of VCA works like this: Given data X, this data is visualized using a scatterplot matrix. Note that this data might have non-zero mean. The mean and the principal component are computed and displayed as lines in the scatterplots. The user can now move the lines in each scatterplot. Note that moving the lines modifies the mean as well as the principal direction. We will explain the mapping from user interaction to vectors later in more detail.

Once the first component $\mathbf{c}_1$ is determined, the data are projected onto the subspace orthogonal to $\mathbf{w}_1$ (cf Eq. 4). This projected data is displayed using the same scatterplot matrix as before. Again, the user can interactively specify direction and mean, however, now the direction is also projected onto the remaining subspace. This forces the specification of orthogonal directions (or uncorrelated components) as desired.

Rotating the axes according to the currently determined directions and reducing the dimensions of the matrix is usually not a good idea, because the correspondence between axes and data variables is lost.

The process is repeated until the data shows no recognizable trend. After the process, the user has extracted a set of mutually uncorrelated principal components that describe the major variation in the data accounting for noise, outliers, or other known non-linear process affecting the data. Figure 2 shows a scatterplot matrix of data in 4 variables, its main components, the scatterplots of the data after removal of the main component, and the remaining principal components. The benefit from visually correcting the main component can be clearly observed.

We will now describe the details of determining a single component $\mathbf{c}$, resp. direction $\mathbf{w}$.

4.1. Scatterplot matrix setup

For each of the d variables maximum and minimum values (i.e. min and max of columns in X) are computed to define the scales of the axes. A row in the scatterplot matrix has identical variables mapped to the horizontal axes, likewise, a column has mapped the same variables to the vertical axes. The mean $\bar{\mathbf{x}}$ and the principal component $\mathbf{c}$ (resp. the direction $\mathbf{w}$) are computed using the SVD. A line with parametric form $\bar{\mathbf{x}} + t\mathbf{w}$ shows the resulting principal direction. In addition, the mean can be displayed in the graph to provide additional hints on the structure of the data.

4.2. 2D line user interaction

The user can modify the line in each scatterplot by moving its endpoints. Each user interaction induces a change to $\bar{\mathbf{x}}$ and $\mathbf{w}$. As $\mathbf{w}$ describes a *direction* we keep it unit length, i.e. $||\mathbf{w}|| = 1$. Assume the line in scatterplot (k,l) is modified to have endpoints p_x, p_y and q_x, q_y. Then we map this change to $\bar{\mathbf{x}}$ and $\mathbf{w}$ as follows.

The mean necessarily lies somewhere on the line. We found it is best to restrict the movement of the mean due to the movement of the line to one dimension, i.e. only in the vertical direction of the scatterplot. The other dimension is accessible in scatterplot (l,k) (as this has the same axes as (k,l), however, reversed). Restricting the movement of the mean to one dimension avoids that the line representing the pair $(\bar{\mathbf{x}}, \mathbf{w})$ changes in all scatterplots. Instead, the line changes only in one column of the matrix. Specifically, we change only the k-the component of $\bar{\mathbf{x}}$ to

$$\bar{x}'_k = \frac{q_x - \bar{x}_l}{q_x - p_x}(q_y - p_y) \tag{5}$$

The direction of $\mathbf{w}$ is essentially described by the ratios of its elements. We let the user control the ratio of the length of the k-th and l-th element relative to the remaining elements. During modification of the line segment $w_k^2 + w_l^2$ (and, thus, also $||\mathbf{w}||$) is fixed. We provide a separate interaction for changing $(w_k^2 + w_l^2)$ relative to $||\mathbf{w}||$. Thus, mapping during line modification is determined by

$$w'_k = (q_x - p_x)\frac{w_k^2 + w_l^2}{(q_x - p_x)^2 + (q_y - p_y)^2} \tag{6}$$

and

$$w'_l = (q_y - p_y)\frac{w_k^2 + w_l^2}{(q_x - p_x)^2 + (q_y - p_y)^2} \tag{7}$$

Note that changing the slope of a line in one of the scatterplots introduces changes in row k and column l of the scatterplot matrix.

5. Clustering

Oftentimes, the process generating the data X is non-linear. Trying to find a linear subspace that represents the data is, therefore, infeasible. One could try, however, to model the process (and, thus, the data) in a piecewise linear fashion. This leads to the idea of clustering the data and performing a PCA on each of the clusters. Figure 3 shows data in a single scatterplot that benefits from representing it with three line segments instead of only one.

Automatic methods for the combination of clustering and PCA have been used in the area of neural computation [KL94, KL97, TB99] and have recently been introduced to the graphics community [SHHS03]. It is clear, however, that finding trends in noisy data requires a careful selection of clusters and principal components at the same time – a task

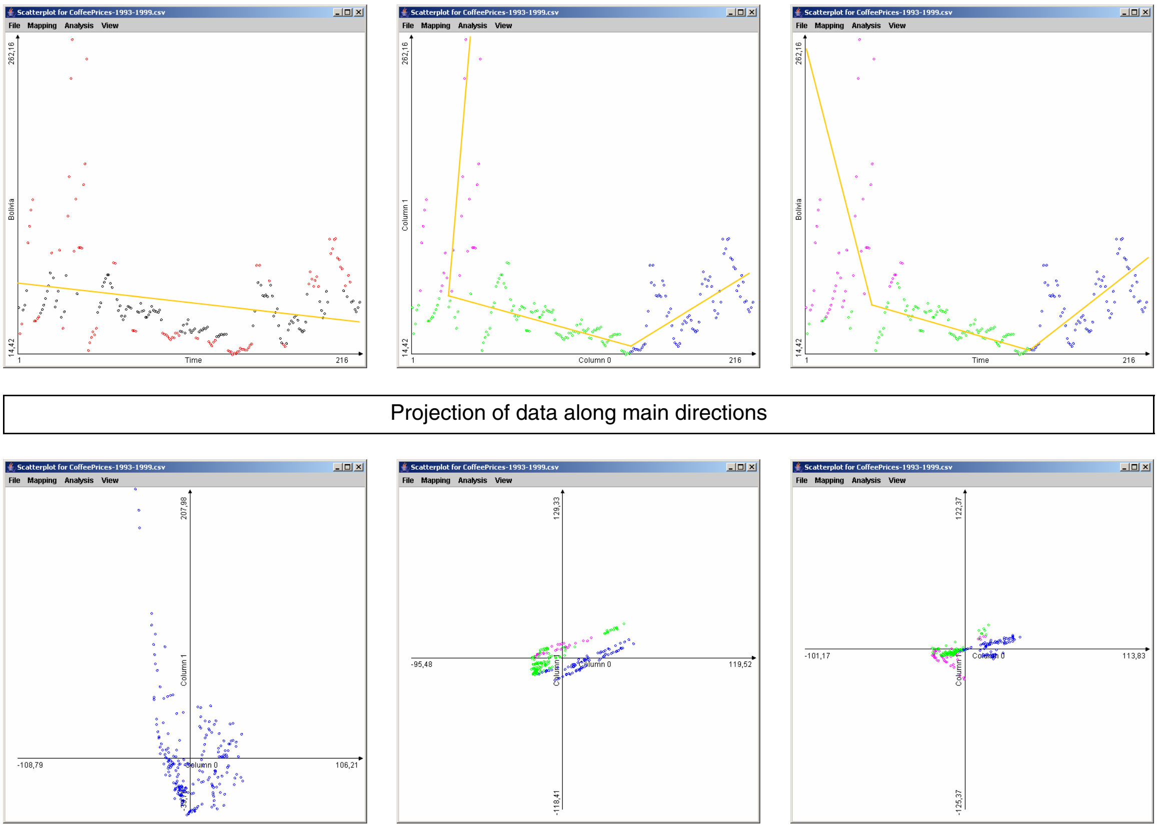

Figure 3: *Data that benefits from a piecewise linear representation. The left column shows the usual representation with a single line segment. The other two columns show different representations with three line segments. The lower row shows the data after projecting (each cluster) along the principal directions.*

that could hardly be done automatically. On the other hand, the human visual systems is very good at identifying clusters.

We combine our idea of VCA with clustering. Instead of allowing the user to place only one line segment to represent the data, a poly-line could be fitted. This poly-line represents the linear components as well as the clustering of the data.

The clustering is performed as follows: The poly-line with s segments is represented by d-dimensional points $\mathbf{p}_m \in \mathbb{R}^d, m \in \{1, \dots, s+1\}$. These control points are projected into the scatterplots and connected to form a poly-line. Each segment $(\mathbf{p}_m, \mathbf{p}_{m+1})$ defines a cluster $\mathcal{C}_m$, a per-cluster mean $\bar{\mathbf{x}}^m$, and a per-cluster principal direction $\mathbf{w}^m$, exactly as described for a single line in Section 4.2.

Clustering is performed based on Euclidean distances to the line segments. For each data vector $\mathbf{x}_i$ we compute the distance to the segments

$$d_m(\mathbf{x}) = ||\mathbf{x} - \mathbf{p}_m - t_m(\mathbf{x})(\mathbf{p}_{m+1} - \mathbf{p}_m)||, \qquad (8)$$

where

$$t_m(\mathbf{x}) = min(1, max(0, (\mathbf{x} - \mathbf{p}_m)(\mathbf{p}_{m+1} - \mathbf{p}_m))), \qquad (9)$$

and define a cluster as

$$\mathcal{C}_m = \{\mathbf{x}_i | d(\mathbf{x}_i)_m < (1 + \varepsilon_d) d(\mathbf{x}_i)_k, k \neq m\}. \qquad (10)$$

Note that for $\varepsilon_d > 0$ there is some overlap between clusters, i.e. some data vectors belong to more than one cluster. The idea is that points around the “joint” of the poly-line cannot be attributed clearly to one or the other cluster.

Using these definitions, the process of defining a poly-line proceeds analogous to to defining a single line. Once the poly-line is determined, all points need to be projected onto the appropriate orthogonal subspaces.

6. Projecting clustered data

All data vectors $\mathbf{x}_i$ that belong to only one cluster $\mathcal{C}_m$ are projected as explained before, however using mean $\bar{x}^m$ and direction $\mathbf{w}^m$. For data vectors belonging to more than one cluster we need to adapt the procedure.

Let $\mathbf{x}_i$ belong to the set of clusters $\{\mathcal{C}_m\}$. We can define a projection of $\mathbf{x}_i$ per cluster as usual:

$$\mathbf{x}_i^m = \mathbf{x}_i - \mathbf{x}_i \mathbf{w}^m \mathbf{w}^{mT} \tag{11}$$

We will represent the projection as a linear combination of per-cluster projections

$$\mathbf{x}_i' = \omega_i^m \mathbf{x}_i^m. \tag{12}$$

For determining reasonable weights ω we like to exploit the distance information, however, so that the weighting continuously varies between the clusters. We define intermediate weights

$$\theta_i^m = \frac{(1+\varepsilon_d)\left(\prod_{k\neq m} d_k(\mathbf{x}_i)\right)^{1/\#\{k\}} - d_m(\mathbf{x}_i)}{\varepsilon_d}, \tag{13}$$

which have the property that they are zero for the boundaries of the intersection of cluster areas. These weights are normalized to yield the (affine independent) weights

$$\omega_i^{m_j} = \frac{\theta_i^{m_j}}{\sum_k \theta_i^{m_k}}. \tag{14}$$

Using these weights the projection of each data vector is uniquely determined. Note that the weights are in general only C^0 functions. While we have found this to be acceptable in applications, weights with higher order of continuity could certainly be constructed.

7. Conclusions

We introduce an approach to the visual determination of principal components in unknown multi-variate data. It integrates basic techniques from the field of multivariate analysis with interactive information visualization. We call this approach *Visual Component Analysis* to stress the importance of interaction and visualization for the resulting components, which could be understand as "visually optimized" principal components.

A panel matrix is used to visualize both data and initial candidates for principal components calculated in an analysis step. Based on this visualization, it is possible to inspect and verify principal components interactively. For instance, one can evaluate the importance of principal components and rotate them in case of an orthogonal result from the analysis step. Also, the weight of outliers in the data can directly be adjusted.

Our system also allows the easy integration of clustering in the analysis step. The data are then represented piecewise linear by a poly-line. For data resulting from non-linear processes, poly-lines provide a simple and powerful primitive for sketching trends in the data. Defining a poly-line automatically defines clusters based on distances to the line segments. This form of clustering is much more intuitive during the search for principal trends in the data.

The analysis is done in a loop, where the component with main variation is removed from the data by projecting along the principal direction. This leads to an iterative process, in which the main variations (and directions) in the data are identified.

We see this work as an initial step that demonstrates the effectiveness of integrating the human visual system into well-established multivariate analysis tools. While not all visualization techniques are equally adequate, it is clear that the general idea could probably be applied to other information visualization techniques or other analysis tools. This seems to be an interesting avenue for future research.

References

[Bed90] BEDDOW J.: Shape coding of multidimensional data on a microcomputer display. In *Proceedings of Visualization 90* (1990), pp. 238–246.

[CC94] COX T. F., COX M. A. A.: *Multidimensional scaling*. Chapman & Hall, London, 1994. With 1 IBM-PC floppy disk (3.5 inch, HD).

[Che73] CHERNOFF H.: Using faces to represent points in *k*–dimensional spae graphically. *Journal of the American Statistical Association 68* (1973), 361–368.

[Cle94] CLEVELAND W. S.: *Visualizing Data*. Hobart Press, 1994.

[Cli87] CLIFF N.: *Analyzing Multivariate Data*. Harcourt Brace Jovanovich, 1987.

[CM84] CLEVELAND W. S., MCGILL R.: The many faces of a scatterplot. *Journal of the American Statistical Association 79*, 388 (1984), 807–822.

[CMS99] CARD S. K., MACKINLAY J. D., SHNEIDERMAN B. (Eds.): *Readings in information visualization: Using vision to think.* Morgan Kaufmann Publishers, San Francisco, 1999.

[FGW01] FAYYAD U. M., GRINSTEIN G. G., WIERSE A.: *Information Visualization in Data Mining and Knowledge Discovery*. Morgan Kaufmann, 2001.

[HKW99] HINNEBURG A., KEIM D. A., WAWRYNIUK M.: HD-Eye: Visual mining of high-dimensional data. *IEEE Computer Graphics and Applications 19*, 5 (Sept./Oct. 1999), 22–31.

[ID90] INSELBERG A., DIMSDALE B.: Parallel coordinates: a tool for visualizing multidimensional geometry. In *IEEE Visualization '90 Proceedings* (Oct. 1990), IEEE Computer Society, pp. 361–378.

[Ins84] INSELBERG A.: Parallel coordinates for multidimensional displays. In *Spatial Information Technologies for Remote Sensing Today and Tomorrow, The Ninth William T. Pecora Memorial Remote Sensing Symposium* (1984), IEEE Computer Society Press, pp. 312–324.

[Jol86] JOLLIFFE I. T.: *Principal Component Analysis*. Series in Statistics. Springer-Verlag, 1986.

[Kei00] KEIM D. A.: Designing pixel-oriented visualization techniques: Theory and applications. *IEEE Transactions on Visualization and Computer Graphics 6*, 1 (2000), 59–78.

[KKA95] KEIM D. A., KRIEGEL H.-P., ANKERST M.: Recursive pattern: A technique for visualizing very large amounts of data. In *Proc. Visualization'95* (1995), pp. 297–286.

[KL94] KAMBHATLA N., LEEN T. K.: Fast non-linear dimension reduction. In *Advances in Neural Information Processing Systems: Proceedings of the 1993 Conference* (San Francisco, CA, 1994), Cowan J. D., Tesauro G.,, Alspector J., (Eds.), Morgan Kaufman Publishers.

[KL97] KAMBHATLA N., LEEN T. K.: Dimension reduction by local principal component analysis. *Neural Computation 9*, 7 (1997), 1493–1516.

[Koh95] KOHONEN T.: *Self-Organizing Maps*, vol. 30 of *Springer Series in Information Sciences*. Springer, Berlin, Heidelberg, 1995. (Second Extended Edition 1997).

[KW78] KRUSKAL J. B., WISH M.: *Multidimensional Scaling*. Sage Piblications, Beverly Hills, CA, 1978.

[KW02] KEIM D., WARD M.: *Visual Data Mining Techniques*, 2 ed. 2002, pp. 403–428.

[PG88] PICKETT R., GRINSTEIN G.: Iconographics display for viusalizing multidimensional data. In *Proceedings of the IEEE Conference on Systems, Man, and Cybernetics* (Piscataway, N.J., 1988), IEEE Press, pp. 514–519.

[RAEM94] RIBARSKY W., AYERS E., EBLE J., MUKHERJEA S.: Glyphmaker: Creating customized visualizations of complex data. *Computer 27*, 7 (July 1994), 57–64.

[SD02] SOUKUP T., DAVIDSON I.: *Visual Data Mining*. John Wiley & Sons, 2002.

[SHHS03] SLOAN P.-P., HALL J., HART J., SNYDER J.: Clustered principal components for precomputed radiance transfer. *ACM Transactions on Graphics 22*, 3 (July 2003), 382–391.

[Shn02] SHNEIDERMAN B.: Inventing discovery tools: combining information visualization with data mining. *Information Visualization 1*, 1 (2002), 5–12.

[TB99] TIPPING M. E., BISHOP C. M.: Mixtures of probabilistic principal component analyzers. *Neural Computation 11*, 2 (1999), 443–482.

[War94] WARD M. O.: XmdvTool: Integrating multiple methods for visualizing multivariate data. In *Proceedings of the Conference on Visualization* (Los Alamitos, CA, USA, Oct. 1994), Bergeron R. D., Kaufman A. E., (Eds.), IEEE Computer Society Press, pp. 326–336.

[WB97] WONG P. C., BERGERON R. D.: *30 Years of Multidimensional Multivariate Visualization*. IEEE Computer Society Press, Los Alamitos, CA, 1997, pp. 3–33.

[WB98] WESTPHAL C., BLAXTON T.: *Data Mining Solutions*. John Wiley & Sons, New York, USA, 1998.

[YWRH03] YANG J., WARD M. O., RUNDENSTEINER E. A., HUANG S.: Visual hierarchical dimension reduction for exploration of high dimensional datasets. pp. 19–28.