Visual Data Mining and Machine Learning

Fabrice Rossi

Projet AxIS, INRIA, Domaine de Voluceau, Rocquencourt, B.P. 105
78153 Le Chesnay Cedex - France

Abstract. Information visualization and visual data mining leverage the human visual system to provide insight and understanding of unorganized data. In order to scale to massive sets of high dimensional data, simplification methods are needed, so as to select important dimensions and objects. Some machine learning algorithms try to solve those problems. We give in this paper an overview of information visualization and survey the links between this field and machine learning.

1 Introduction

The rational of Information Visualization (infovis [14]) and of Visual Data Mining (VDM [44, 19]) is to leverage the very high processing capabilities of the human visual system to allow interactive exploration and analysis of massive data sets. It has been demonstrated that the low level visual system has preattentive processing capabilities (see e.g. [32]) that enables humans to detect and recognize some features without effort and extremely rapidly, generally in less than 200 ms, even in a large image. As shown in [24], relying on this type of features (e.g. the color of items, their orientation, etc.) enables to display up to one million of items without overloading the human visual system.

Information visualization faces however two major limitations of human vision. Its main limitation is its physical restriction to three dimensional (3D) displays. Moreover, while stereovision hardware (for instance based on shutter glasses) is now affordable, it is still quite uncommon: for the vast majority of users, visualization methods must rely on two dimensional displays (2D). 3D is also intrinsically limited by many problems such as occlusions, disorientation, two dimensional interaction devices, etc.

The second major limitation of human vision is pointed out in [32]: preattentive features cannot be combined freely. If more than two or three such features are used in the same image, they can interfere and reduce greatly the processing rate of the visual system. Therefore, if objects of a data set are described by more than a few attributes, their visualization becomes difficult: a trade-off between completeness of the representation and processing rate has to be made. Complete representations have therefore two drawbacks: they must rely on complex layout methods to transform high dimensional objects into 2D images and they imply a tedious browsing of the full image to obtain a complete understanding of the data set. Simplification methods are therefore needed for visual mining of massive data sets.

Machine learning (in a broad sense) and visual data mining are therefore strongly connected. Machine learning algorithms benefit from expert knowledge:

clustering is easier when the number of clusters is known *a priori*, recognition rates are higher if the training set is free of outliers and if useless variables have been removed, etc. Many of these tasks (outlier detection, number of cluster evaluation, etc.) can be performed by users via a visual inspection of the considered data set: this is exactly the purpose of visual data mining. However, as explained before, information visualization is efficient for simplified data sets: images are easier to read if they represent a small number of objects described by a small number of attributes. Machine learning algorithms can provide the simplifying methods that make visual data mining efficient: dimension reduction can be used to select important attributes, clustering allows one to replace homogeneous groups of objects by some representative examples, etc. Interactive methods can mix visualization and model construction: the user guide the modeling process via the display of results obtained so far (see e.g. [12, 35, 63, 76]).

We survey in this paper the links between visual data mining and machine learning. In section 2 we give a short introduction to information visualization and to its limits. In section 3 we survey dimension reduction techniques that can provide dimension scalability to 2D displays. In section 4 we briefly outline the links between clustering and infovis. We conclude in section 5 with a short overview of two major models coming from machine learning and extremely useful for infovis: the Self Organizing Map and the Generative Topographic Mapping.

2 Information visualization

According to [14] information visualization is "the use of computer-supported interactive, visual representation of abstract data to amplify cognition". In this survey, we focus on a special type of abstract data: each object is a vector from $\mathbb{R}^p$, described by p real values. The descriptors are called variables, features or attributes. This model is frequently named the "table data model" in the information visualization community [19, 36].

2.1 Taxonomy of information visualization methods

Several attempts have been made to classify infovis methods in order to get a clear overview of the field (see e.g. [19]). Daniel Keim proposes in [43, 44] to analyze visualization methods according to three "orthogonal" axes: the **visualization technique** itself, the **interaction technique** and the **data type**. Keim's analysis is based on the fact that interaction methods (such as zooming [7], linking and brushing [6], dynamic distortion [54], etc.) can be freely combined with visualization techniques (see section 2.3) and applied to different types of data (vectors, trees, graphs [33], text [30], etc.).

Card et al. propose in [14] a different taxonomy based on the nature of the information to be visualized. The part of this taxonomy that gathers visual data mining methods is further subdivided according to the data type, in a way quite similar to Keim's approach.

Unfortunately, those taxonomies don't help in identifying how machine learning methods can be used to improve visualization. For instance, dimension reduction methods are buried in the class of "Geometrically-transformed displays" in [44] or considered as preprocessing steps in [19].

2.2 A formal model

Further high level understanding of visualization methods can be obtained with the help of the formal model of Chi and Riedl [18]. In this model, raw data go through four stages via three processing steps. Each step is implemented by an operator that map the representation of the data in one stage to another representation in the next stage (the structure of the data representation is modified). A variation of this model appears in [14].

In the **data transformation** step, raw data are mapped to a mathematical representation (for instance texts are parsed into a vector model of word occurrences); this representation is called the "analytical abstraction". In the **visualization transformation** step, the analytical abstraction is transformed into another representation adapted to visualization (for instance a graph is transformed into a tree by a traversal algorithm in order to use a tree visualization method); this new representation is called the "visualization abstraction". The **visual mapping** step translates the visualization abstraction into a view/image (for instance a tree is visualized with the TreeMap method [71]). Additionally, operators can be used to modify the data representation within one stage: for instance, interaction methods can be considered as operators that modify the view or that have impact on the visual mapping step.

Keim's visualization technique axis corresponds roughly to the visualization transformation step, whereas his data axis gathers the early steps (data transformation and visualization transformation steps). The formal model gives a better understanding of the reasons why Keim's axes are more or less independent ("orthogonal"). In [17], Chi leverages his formal model to produce a taxonomy of 36 visualization methods. He shows how methods are constructed based on standard operators (data extraction, clustering, projection, etc.).

Machine learning methods fit nicely in this formal model and correspond to some operators. For instance, dimension reduction methods are generally operators from the visualization abstraction step: they produce new coordinates from the original mathematical representation.

2.3 Some visualization methods for high dimensional data

Keim's visualization technique axis is further subdivided into broad classes of methods. We briefly survey methods adapted to high dimensional data, using Keim's taxonomy. We refer to [44, 19, 14] for comprehensive presentation. Keim identifies four classes of methods described in the following sections.

2.3.1 *Geometrically-transformed displays*

Some methods use a layout algorithm to transform some high dimensional data into low (2 or 3) dimensional data that are displayed by standard methods.

The main standard tool is the scatter plot, i.e. the standard 2D image in which each object described by two attributes is represented by a point whose coordinates are given by the values of the attributes. For p-dimensional data, a $p \times p$ scatter plot matrix is obtained by arranging in a matrix all the possible scatter plots. Each attribute appears both as a line and as a column: the image at position (i, j) is the scatter plot for attribute i and attribute j. The diagonal can be used to display some standard graphical representation of each attribute, for instance a histogram. Scatter plot matrices don't scale to a large number of attributes, because the number of scatter plots grows quadratically with the number of variables. Moreover, scatter plots themselves suffer from the superpositions of objects with similar (or close) attribute values. Linking and brushing techniques [6] help nevertheless the user to understand large scatter plot matrices: the user can select a region in one scatter plot and observe the results of this selection in all the plots. Dimension reduction methods (see section 3) can be used to avoid displaying the full scatter plot matrix.

Another standard 2D display is the functional plot in which $y = f(x)$ is represented by a smooth line obtained by a high frequency sampling of the x-axis. Data in $\mathbb{R}^p$ can be transformed into functions, for instance with the method proposed by Andrews in [2]: the coordinates of each object are used as the coefficients of a Fourier series to define a function. The display is obtained by plotting together all the functions on the $[-\pi, \pi]$ interval (see [29] for a recent application of Andrews' curves).

Another use of functional like plot is the parallel coordinates technique [38, 39]. It consists simply in using as many vertical axes as there are attributes to represent. An object is then displayed as a polygonal line that links the values of its attributes on the corresponding axis. Parallel coordinates don't scale to a large number of objects mainly because of overlapping (see [3] for an example of a rendering method that limits this overlapping problem).

2.3.2 *Iconic displays*

Iconic methods represent each object with a complex icon or glyph [86]. Famous examples of such glyph are Chernoff's faces [16]: each object is represented by a small face where different data dimensions are mapped to different facial characteristics (face width, radius of the eyes, etc.). Other examples include the star glyph [72] and the stick-figure icon [59]. Those methods have scaling issues because representing many characteristics implies to use complex icons that use a lot of space on the screen and limit strongly the number of objects that can be visualized at once. Moreover, comparison of two objects is difficult is the corresponding glyphs are far away from each other: the problem of optimal glyph positioning is therefore quite accurate [86].

2.3.3 Dense pixel displays

In pixel oriented methods [45, 43], each attribute of each object is represented by an unique pixel, via its color. This family of methods scales to large number of objects and/or attributes. It has however ordering problems, as the insights on the data it provides depend strongly on the quality of pixel arrangements: pixels corresponding to related objects and/or related attributes should be close in the image. As pointed out in [43], dimension/attribute ordering is in fact an NP-complete problem and only sub-optimal solutions can be obtained in reasonable time.

2.3.4 Stacked displays

Stacked displays correspond to methods in which the image is partitioned recursively in such a way that each level of the hierarchy represent one or several attributes of the data (see [51] and [23] for instance). As for many other layout methods, the quality of the visualization strongly depends on the ordering of the dimensions.

2.4 Visual data mining tasks

The main goal of VDM is to enable users to explore massive data sets and to search for interesting information. As pointed out in [34], visualization allows to find patterns in the data by proximity and similarity reasoning: plots of the characteristics of objects might reveal dependency between variables as well as clusters of objects. More generally, as shown by P. Hoffman in [36], visual data mining is efficient for many classical tasks, such as: cluster detection, outlier detection, feature importance assessment, feature correlation, prior classification analysis, etc.

Another goal of VDM is to display the results of mining algorithms [47], for instance association rules or frequent patterns extracted from a database, clusters (either extracted by the algorithm or pre-specified in case of supervised learning), etc.

While the bulk of VDM methods is dedicated to unsupervised problems, there is also an important need of visual methods for supervised problems such as classification. Due to size constraints, we won't however cover this important field in this paper (an example of the visualization of classification is given in this volume by [40]).

2.5 Links with machine learning

While integrating machine learning and information visualization appears clearly as potentially rewarding (see e.g. the paper [15] by Chen, editor-in-chief of *Information Visualization*), such integration is still rare, with some remarkable exceptions such as multidimensional scaling (MDS, see section 3.1) which is a standard method in VDM and the Self-Organizing Map (SOM, see section

5.1) whose visualization capabilities have been widely recognized by the infovis community.

It is obvious that most of the methods designed to display high dimensional data have scaling problems. In order to display voluminous data sets in which objects are described by numerous variables, they must rely on simplification methods that reduce either the number of variables or the number of objects (or both). Numerous machine learning methods have been designed to tackle those problems. However, infovis methods tend to favor user intervention over automatic methods. For instance, in order to avoid saturation of the human vision system, information visualization uses frequently the concept of "focus+context" (see chapter 4 of [14]): the general idea is to provide a detailed view of a part of the data while retaining as much context information as possible. Distortion techniques have been used to implement this idea (see [54] for a survey). User intervention consists in choosing the interesting part of the data by browsing the summarized version (see [62, 50] for well known examples).

A possible way for building more links between machine learning methods and visualization methods would be to favor user intervention and control. Ward describes for instance in [87] how visual clues allow user to monitor dimension reduction and clustering techniques in order to check whether important information might have been removed. Visual representations of the quality of dimension reduction methods have been produced (see e.g. [4, 5, 9, 76]). Their generalization to e.g. manifold learning methods (see section 3.2), might be a first step toward the integration of those methods in infovis algorithms.

3 Dimensionality reduction

While dimension reduction [25, 13, 68] is generally considered as an important pre-processing task in infovis (see e.g. [19]), it seems that only standard methods, such as principal component analysis (PCA) and multidimensional scaling (MDS), are commonly used in visual data mining.

Among classical methods, and apart from PCA, projection pursuit [26] is quite popular in infovis. As Independent Component Analysis (ICA) [41, 37] (but for different reasons), projection pursuit is looking for linear combination of the original features that are non Gaussian. Generalization of PCA such as Hastie's principal curves and surfaces [31], neural network based non linear PCA [21] or Kernel Principal Component Analysis [69], are not widely used in VDM.

3.1 Multidimensional scaling

In fact, the most popular non linear projection method in infovis is multidimensional scaling (MDS) framework. The main idea of MDS is to compare distances between the objects in the low dimensional space to the corresponding dissimilarities in the original space. Different *stress functions* (i.e. measure of the distortion of the distances) lead to different algorithms. Torgerson's original MDS (classical metric MDS [79]) works for original data in an Euclidean space and tries to preserve inner products. It also corresponds to finding the linear projection of the data that preserves best the square euclidean distances between the original observations. It can be shown to be equivalent to PCA.

Commonly used MDS are based on Kruskal's version [48, 49]. Let us denote $\delta_{i,j}$ the dissimilarity between objects i and j in the original space and $d_{i,j}$ the Euclidean distance between the low dimensional representations of i and j. A generic stress function is given by $\frac{\sum_{i,j} w_{i,j}(f(\delta_{i,j})-d_{i,j})^2}{\sum_{i,j}(d_{i,j})^2}$, where f is a transformation of the original dissimilarities and the $w_{i,j}$ are weighting coefficients. Variations on the normalization method, the coefficients and on the transformation lead to Sammon's non linear mapping (NLM [66], see also [55] for a smooth version based on a MLP), to Curvilinear Component Analysis (CCA [20]) and to other variants such as the non metric scaling (in which f is monotone) that tries to preserve ranking between dissimilarities rather than their actual values.

It should be noted that MDS methods are quite computationally intensive. As a consequence a lot of work has been done in order to reduce their actual cost (see [73] in this volume). Fast Map for instance [22], a well known dimension reduction algorithm in the visualization community, is in fact an approximate realization of MDS, as shown in [60].

3.2 Manifold learning

While MDS methods overcome some limitations of PCA and related methods, they still have an important shortcoming: if the data happen to belong to a subspace of the original space, MDS methods might fail to discover this fact when the shape of the subspace is complex. The class of manifold learning techniques try to overcome this limitation for instance by feeding a MDS like method with "smart" dissimilarities. One popular idea of those methods it to work at the local level: the structure of the manifold around a data point is described by the k nearest neighbors (k-nn) of this point in the original space.

The field of manifold learning is evolving very quickly and numerous methods have been proposed to address the problem. Two recent surveys [13, 68] present current tendencies of the field. Among popular methods, we can mention Isomap [75], Curvilinear Distances Analysis (CDA [52]), Locally Linear Embedding (LLE [65, 67]) and Laplacian Eigenmaps [8].

3.3 Latent variable models

In latent variable models, the high dimensional observed data $t_1, \dots, t_n$ in $\mathbb{R}^p$ are supposed to be generated from corresponding low dimensional unobserved (or latent) data $x_1, \dots, x_n$ in $\mathbb{R}^q$ with $q < p$ via the general formulation $t = y(x; W)+\epsilon$, where y is a function of the latent variables x and of some parameters W, and where ϵ represents some noise.

The simplest and oldest latent variable model is the one of factor analysis (see e.g. [13]). In this model, y is a linear function of both x and W: $t = Wx + \mu + \epsilon$ (where μ is the expectation of t). Moreover, we assume that x has Gaussian distribution (with identity covariance matrix) and ϵ is also Gaussian with covariance matrix Ψ. The parameters of the model can be estimated via an

EM algorithm. In practice however, additional constraints are used to simplify the estimation, for instance by assuming that Ψ is known. Additional (strong) assumptions allows one to show that PCA is a special case of factor analysis.

Factor analysis is revisited in [78] to produce a probabilistic PCA (PPCA) model for which a closed-form solution is given. The main advantage of this model over PCA is its tolerance to missing data, but it can also be extended to a mixture of PPCA models [77]: data are assumed to be generated by a mixture of local factor models. This provides multiple local linear projections of the same data set. A related hierarchical latent model has also been proposed [12]. Further development of similar models can be found in [85] (see also section 5.1.1).

3.4 Limitations of dimension reduction for infovis

An important limitation of advanced dimension reduction methods, apart from their computational cost, is that many of them don't always produce interesting visualization. As pointed out in e.g. [13], Kernel PCA for instance, is more adapted to feature extraction than to dimension reduction as PCA is conducted in the high dimensional kernel induced feature space. Manifold learning is also intrinsically limited by the fact that low dimensional manifolds cannot in general be projected to two dimensions without introducing distortions. Earth maps are well known examples of this problem. A possible solution can be found in a recent tearing method proposed in [53].

Finally, the main problem is to evaluate the impact of the dimension reduction on the ability to conduct data mining tasks: if the neighborhood relationship between objects are not correctly preserved by the reduction methods, for instance, the corresponding visualization can lead to false conclusion because close objects can be mapped to distant points and *vice versa* and can therefore produce visualizations that lead to false conclusion. Linear projection methods always reduce distances between points and can therefore project outliers close to the bulk of the other data, for instance. Non linear methods introduce more complex distortions which mix compression and stretching. Moreover, as pointed out in [70], some manifold learning methods, such as LLE, don't even explicitly preserve distances.

The global distortion of a projection method can be assessed with neighborhood preservation measures [82, 42]. These measures allow the user to decide whether she should trust the projection or not. Detailed and local analysis of the distortion can be done with the visualization methods proposed in [4, 5].

4 Reducing the number of objects

Limiting the number of objects to display is extremely important for most of the visualization methods. Clustering algorithms have been used for this task in order to provide scalability to some visualization methods, especially those, such as parallel coordinates, that are impaired by superposition. Objects are then replaced by a representative object (a prototype) chosen in the cluster to which they belong (see [87]).

It should be noted that this type of automatic data reduction techniques have been used in infovis only quite recently, e.g. in [63, 27]. However, rather than simply relying on prototypes produced by clustering algorithms, summarized displays include visual representation of the clusters themselves. The method proposed in [27] (and latter generalized in [28]) displays both prototypes and the variability in clusters, using some color coding. Moreover, as the simplification is based on hierarchical clustering, the user can interactively choose the amount of simplification.

5 Specialized models

5.1 Self-Organizing Map

The great success of the Self-Organizing Map (SOM [46]) as a visualization tool might be a consequence of the simultaneous simplifications of a data set that it operates: it acts both as a clustering algorithm and as a non linear projection method. Moreover, the grid structure solves nicely the superposition problems associated to algorithms surveyed in section 3: complex representation of the prototypes can be used without superposition. The basic SOM has been completed by many visualization enhancements ranging from component planes to the U-matrix [81]. Surveys of visualization methods based on SOM can be found in [83, 84, 34].

Extensions to the SOM have been designed based on some important discoveries of the infovis community (see in particular [34]). The "focus+context" concept for instance has motivated the introduction of hyperbolic SOM [64] and more recently of hierarchically growing hyperbolic SOM [57] (see also [1] for a combination of neural gas [56] with the "focus+context" principle).

Other recent works on visualization methods for the SOM include graph based approaches [61], P-Matrix [80], connectivity matrix [74], etc.

5.1.1 Generative Topographic Mapping

In addition to the latent models surveyed in 3.3, which operate only a (local) dimension reduction, a very interesting non linear model, the Generative Topographic Mapping (GTM) has been proposed [11]. As the SOM, this model can be seen as doing both a dimension reduction and some form of clustering. The model is based as the SOM on a grid of points $(x_i)_{1\leq i\leq k}$ chosen in the low dimensional latent space. A set of m non linear functions $(\phi_j)_{1\leq j\leq m}$ is used to map the latent space to $\mathbb{R}^m$ (the mapping is denoted Φ). The observed data t are assumed to be distributed as a mixture of k Gaussians with a common covariance matrix $\beta^{-1}\mathbf{1}$ and with centers given by $(W\Phi(x_i))_{1\leq i\leq k}$, where W is a $p\times m$ parameter matrix. An EM algorithm is used to fit W and β to the data.

The application of GTM to visualization is based on the fact that each observation t_i induces a posterior distribution in the latent space. This distribution can be summarized by its mean or by its mode, and provides this way a non linear projection of each observation to the latent space. An important difference with the SOM is that when the mean is used to represent an observation, its position is not constrained in the grid of points and the projection is therefore smooth. If the mode is considered, then the visualization is quite similar to what can be obtained with the SOM.

The initial GTM model has been modified and adapted in many ways (see [10]). For the visualization aspect, interesting developments include the visualization of the distortion (magnification factors) [9] and a hierarchical GTM model proposed in [76], which is an extension of the hierarchical local linear projection developed in [12]. Variations of the GTM can be used to identify outliers [58].

6 Conclusion

Integrating machine learning and information visualization is potentially rewarding, as demonstrated by successful visual data mining tools such as the Self-Organizing Map and the Generative Topographic Mapping. A lot of integration work remains however to be done in order to benefit from advanced results of both domains. User control and interaction, aesthetic layout, linking-and-brushing remain for instance quite rare in machine learning oriented programs, whereas advanced dimension reduction and clustering methods are seldom use in infovis. As pointed out in [15], tightening the bounds between machine learning and information visualization is one of the challenge in which both community might find very rewarding results.

References

[1] S. Al Shehabi and J.-C. Lamirel. A new hyperbolic visualization method for displaying the results of a neural gas model: application to webometrics. In *Proceedings of XIVth European Symposium on Artificial Neural Networks (ESANN 2006)*, Bruges (Belgium), April 2006. In this volume.

[2] D. F. Andrews. Plots of high dimensional data. *Biometrics*, 28:125–136, 1972.

[3] A. O. Artero, M. C. F. de Olivera, and H. Levkowitz. Uncovering clusters in crowded parallel coordinates visualizations. In *Proceedings of the IEEE Symposium on Information Visualization 2004*, pages 81–88, Austin, Texas, USA, October 2004.

[4] M. Aupetit. Visualizing distortion in continuous projection techniques. In *Proceedings of XIIth European Symposium on Artificial Neural Networks (ESANN 2004)*, pages 465–470, Bruges (Belgium), April 2004.

[5] M. Aupetit. Visualizing the trustworthiness of a projection. In *Proceedings of XIVth European Symposium on Artificial Neural Networks (ESANN 2006)*, Bruges (Belgium), April 2006. In this volume.

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

[7] B. B. Bederson and J. D. Hollan. Pad++: a zooming graphical interface for exploring alternate interface physics. In *UIST '94: Proceedings of the 7th annual ACM symposium on User interface software and technology*, pages 17–26, New York, NY, USA, 1994. ACM Press.

[8] M. Belkin and P. Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, 15(6):1373–1396, 2003.

[9] C. M. Bishop, M. Svensén, and C. K. I. Williams. Magnification factors for the som and the gtm algorithms. In *Proceedings of the Workshop on Self-Organizing Maps (WSOM 97)*, pages 333–338, Helsinki, Finland, 1997.

[10] C. M. Bishop, M. Svensén, and C. K. I. Williams. Developments of the generative topographic mapping. *Neurocomputing*, 21:203–224, 1998.

[11] C. M. Bishop, M. Svensén, and C. K. I. Williams. GTM: The generative topographic mapping. *Neural Computation*, 10(1):215–234, 1998.

[12] C. M. Bishop and M. E. Tipping. A hierarchical latent variable model for data visualization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(3):281–293, 1998.

[13] C. J. C. Burges. Geometric methods for feature extraction and dimensional reduction. In L. Rokach and O. Maimon, editors, *Data Mining and Knowledge Discovery Handbook: A Complete Guide for Practitioners and Researchers.* Kluwer Academic Publishers, 2005.

[14] S. K. Card, J. D. Mackinlay, and B. Shneiderman, editors. *Readings in Information Visualization: Using Vision to Think.* Morgan Kaufmann, San Francisco, 1999.

[15] C. Chen. Top 10 unsolved information visualization problems. *IEEE Computer Graphics and Applications*, 25(4):12–16, July/August 2005.

[16] H. Chernoff. The use of faces to represent points in k-dimensional space graphically. *Journal of the American Statistical Association*, 68:361–368, 1973.

[17] E. H. Chi. A taxonomy of visualization techniques using the data state reference model. In *Proceedings of the IEEE Symposium on Information Visualization 2000 (InfoVis'00)*, 2000.

[18] E. H. Chi and J. T. Riedl. An operator interaction framework for visualization systems. In *Proceedings of the IEEE Symposium on Information Visualization (InfoVis'98)*, pages 63–70, Research Triangle Park, North Carolina, USA, October 1998.

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

[20] P. Demartines and J. Hérault. Curvilinear component analysis: a self-organizing neural network for nonlinear mapping of data sets. *IEEE Transactions on Neural Networks*, 8(1):148–154, 1997.

[21] K. Diamantaras and S.-Y. Kung. *Principal Component Neural Networks: Theory and Applications.* Adaptive and Learning System. John Wiley & Sons, 1996.

[22] C. Faloutsos and K.-I. Lin. Fastmap: A fast algorithm for indexing, data-mining and visualization of traditional and multimedia datasets. In *Proceedings ACM SIGMOD International Conference Management of Data (ACM SIGMOD '95)*, pages 163–174, 1995.

[23] S. Feiner and C. Beshers. Worlds within worlds – metaphors for exploring n-dimensional virtual worlds. In *Proceedings of the ACM Symposium on User Interface Software and Technology (UIST'90)*, pages 76–83, Snowbird, USA, October 1990.

[24] J.-D. Fekete and C. Plaisant. Interactive information visualization of a million items. In *Proceedings of IEEE Symposium on Information Visualization 2002 (InfoVis 2002)*, pages 117–124, Boston (USA), October 2002. IEEE Press.

[25] I. K. Fodor. A survey of dimension reduction techniques. Technical Report UCRL–ID–148494, Lawrence Livermore National Laboratory, June 2002.

[26] J. H. Friedman and J. W. Tukey. A projection pursuit algorithm for exploratory data analysis. *IEEE Transactions on Computers*, C-23(9):881–890, 1974.

[27] Y. Fua, M. Ward, and E. Rundensteiner. Hierarchical parallel coordinates for exploration of large datasets. In *Proceedings of IEEE conference Visualization'99*, pages 43–50, October 1999.

[28] Y. Fua, M. Ward, and E. Rundensteiner. Structure-based brushes: A mechanism for navigating hierarchically organized data and information spaces. *IEEE Transactions Visu alization and Computer Graphics*, 6(2):150–159, April–June 2000.

[29] C. García-Osorio, J. Maudes, and C. Fyfe. Using Andrews curves for clustering and sub-clustering self-organizing maps. In *Proceedings of XIIth European Symposium on Artificial Neural Networks (ESANN 2004)*, pages 477–482, Bruges (Belgium), April 2004.

[30] M. Hascoët and M. Beaudouin-Lafon. Visualisation interactive d'information. *Revue I3*, 1(1):77–108, 2001.

[31] T. Hastie. *Principal Curves and Surfaces*. PhD thesis, Stanford University, November 1984.

[32] C. G. Healey, K. S. Booth, and J. T. Enns. Visualizing real-time multivariate data using preattentive processing. *ACM Transactions on Modeling and Computer Simulation*, 5(3):190–221, July 1995.

[33] I. Herman, G. Melançon, and M. Scott Marshall. Graph visualization and navigation in information visualisation. *IEEE Transactions on Visualization and Computer Graphics*, 6(1):24–43, 2000.

[34] J. Himberg. *From Insights to Innovations: Data Mining, Visualization, and User Interfaces*. PhD thesis, Helsinki University of Technology, Espoo (Finland), November 2004.

[35] A. Hinneburg, D. Keim, and M. Wawryniuk. Hd-eye: Visual mining of high-dimensional data,. *IEEE Computer Graphics and Applications*, 19(5):22–31, September/October 1999.

[36] P. E. Hoffman. *Table Visualizations: A Formal Model and Its Applications*. PhD thesis, University of Massachusetts at Lowell, 1999.

[37] A. Hyvärinen. Survey on independent component analysis. *Neural Computing Surveys*, 2:94–128, 1999.

[38] A. Inselberg. The plane with parallel coordinates. *The Visual Computer*, 1(4):69–91, December 1985.

[39] A. Inselberg and B. Dimsdale. Parallel coordinates: a tool for visualizing multi-dimensional geometry. In *Proceedings of the First IEEE Conference on Visualization*, pages 361–378, San Francisco (USA), October 1990.

[40] T. Iwata, K. Saito, and N. Ueda. Visual nonlinear discriminant analysis for classifier design. In *Proceedings of XIVth European Symposium on Artificial Neural Networks (ESANN 2006)*, Bruges (Belgium), April 2006. In this volume.

[41] C. Jutten and J. Hérault. Blind separation of sources, part I: An adaptive algorithm based on neuromimetic architecture. *Signal Processing*, 24:1–10, 1991.

[42] S. Kaski, J. Nikkila, M. Oja, J. Venna, P. Toronen, and E. Castren. Trustworthiness and metrics in visualizing similarity of gene expression. *BMC Bioinformatics*, 4, 2003.

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

[44] D. A. Keim. Information visualization and visual data mining. *IEEE Transactions on Visualization and Computer Graphics*, 7(1):100–107, January–March 2002.

[45] D. A. Keim and H.-P. Kriegel. VisDB: database exploration using multidimensional visualization. *Computer Graphics and Applications*, 14(5):40–49, September 1994.

[46] T. Kohonen. *Self-Organizing Maps*, volume 30 of *Springer Series in Information Sciences*. Springer, third edition, 1995. Last edition published in 2001.

[47] I. Kopanakis and B. Theodoulidis. Visual data mining modeling techniques for the visualization of mining outcomes. *Journal of Visual Languages and Computing*, 14:543–589, 2003.

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

[49] J. B. Kruskal. Nonmetric multidimensional scaling: a numerical method. *Psychometrika*, 29:115–129, 1964.

[50] J. Lamping, R. Rao, and P. Pirolli. A focus+context technique based on hyperbolic geometry for visualizing large hierarchies. In *Proc. ACM Conf. Human Factors in Computing Systems, CHI*, pages 401–408. ACM, 1995.

[51] J. LeBlanc, M. O. Ward, and N. Wittels. Exploring n-dimensional databases. In *Proceedings of the First IEEE Conference on Visualization*, pages 230–237, San Francisco (USA), October 1990.

[52] J. A. Lee, A. Lendasse, and M. Verleysen. Nonlinear projection with curvilinear distances: Isomap versus curvilinear distance analysis. *Neurocomputing*, 57:49–76, March 2004.

[53] J. A. Lee and M. Verleysen. Nonlinear dimensionality reduction of data manifolds with essential loops. *Neurocomputing*, 67:29–53, August 2005. Special issue on Geometrical Methods in Neural Networks and Learning.

[54] Y. K. Leung and M. D. Apperley. A review and taxonomy of distortion-oriented presentation techniques. *ACM Trans. Comput.-Hum. Interact.*, 1(2):126–160, 1994.

[55] J. Mao and A. K. Jain. Artificial neural networks for feature extraction and multivariate data projection. *IEEE Transactions on Neural Networks*, 6(2):296–317, March 1995.

[56] T. M. Martinetz, S. G. Berkovich, and K. J. Schulten. "neural-gas" network for vector quantization and its application to time-series prediction. *IEEE Transactions on Neural Networks*, 4(4):558–569, 1993.

[57] J. Ontrup and H. Ritter. A hierarchically growing hyperbolic self-organizing map for rapid structuring of large data sets. In *Proceedings of the 5th Workshop on Self-Organizing Maps (WSOM 05)*, Paris (France), September 2005.

[58] M. Peña and C. Fyfe. Outlier identification with the harmonic topographic mapping. In *Proceedings of XIVth European Symposium on Artificial Neural Networks (ESANN 2006)*, Bruges (Belgium), April 2006. In this volume.

[59] R. M. Pickett and G. G. Grinstein. Iconographic displays for visualizing multidimensional data. In *Proceedings of the 1988 IEEE International Conference on Systems, Man, and Cybernetics*, volume 1, pages 514–519, August 1988.

[60] J. C. Platt. FastMap, MetricMap, and Landmark MDS are all Nyström algorithms. In *Procedings of the 10th International Workshop on Artificial Intelligence and Statistics*, pages 261–268, 2005.

[61] G. Pölzlbauer, A. Rauber, and M. Dittenbach. Graph projection techniques for self-organizing maps. In *Proceedings of XIIIth European Symposium on Artificial Neural Networks (ESANN 2005)*, pages 533–538, Bruges (Belgium), April 2005.

[62] R. Rao and S. K. Card. The table lens: Merging graphical and symbolic representations in an interactive focus+context visualization for tabular information. In *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems*, pages 318–322, Boston, MA, April 1994.

[63] W. Ribarsky, J. Katz, F. Jiang, and A. Holland. Discovery visualization using fast clustering. *IEEE Computer Graphics and Applications*, 19(5):32–39, 1999.

[64] H. Ritter. Self-organizing maps in non-euclidean spaces. In E. Oja and S. Kaski, editors, *Kohonen Maps*, pages 97–108. Elsevier, Amsterdam, 1999.

[65] S. T. Roweis and L. K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(22):2323–2326, December 2000.

[66] J. W. Sammon. A nonlinear mapping for data structure analysis. *IEEE Transactions on Computer*, C–18(5):401–409, May 1969.

[67] L. K. Saul and S. T. Roweis. Think globally, fit locally: Unsupervised learning of low dimensional manifolds. *Journal of Machine Learning Research*, 4:119–155, 2003.

[68] L. K. Saul, K. Q. Weinberger, F. Sha, J. Ham, and D. D. Lee. Spectral methods for dimensionality reduction. In B. Schölkopf, O. Chapelle, and A. Zien, editors, *Semisupervised Learning*. MIT Press, Cambridge, MA, 2006. In press.

[69] B. Schölkopf, A. J. Smola, and K.-R. Müller. Kernel principal component analysis. In B. Schölkopf, C. J. C. Burges, and A. J. Smola, editors, *Advances in Kernel Methods – Support Vector Learning*, pages 327–352. MIT Press, Cambridge, MA, USA, 1999.

[70] F. Sha and L. K. Saul. Analysis and extension of spectral methods for nonlinear dimensionality reduction. In *Proceedings of the Twenty Second International Conference on Machine Learning (ICML-05)*, pages 785–792, Bonn, Germany., 2005.

[71] B. Shneiderman. Tree visualization with tree-maps: 2-d space-filling approach. *ACM Trans. Graph.*, 11(1):92–99, 1992.

[72] J. H. Siegel, E. J. Farrell, R. Goldwyn, and H. Friedman. The surgical implication of physiologic patterns in myocardial infarction shock. *Surgery*, 72:126–141, 1972.

[73] M. Strickert, N. Sreenivasulu, and U. Seiffert. Sanger-driven mdslocalize for genomic data – a comparative study. In *Proceedings of XIVth European Symposium on Artificial Neural Networks (ESANN 2006)*, Bruges (Belgium), April 2006. In this volume.

[74] K. Taşdemir and E. Merényi. Data topology visualization for the self-organizing map. In *Proceedings of XIVth European Symposium on Artificial Neural Networks (ESANN 2006)*, Bruges (Belgium), April 2006. In this volume.

[75] J. B. Tenenbaum, V. de Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(22):2319–2323, December 2000.

[76] P. Tino and I. Nabney. Hierarchical gtm: Constructing localized non-linear projection manifolds in a principled way. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(5):639–656, 2002.

[77] M. E. Tipping and C. M. Bishop. Mixtures of probabilistic principal component analyzers. *Neural Computation*, 11(2):443–482, 1999.

[78] M. E. Tipping and C. M. Bishop. Probabilistic principal component analysis. *Journal of the Royal Statistical Society, Series B*, 61(3):611–622, 1999.

[79] W. S. Torgerson. Multidimensional scaling: I. theory and method. *Psychometrika*, 17:401–419, 1952.

[80] A. Ultsch. Maps for the visualization of high-dimensional data spaces. In *Proceedings of the 4th Workshop on Self-Organizing Maps (WSOM 03)*, Kyushu, Japan), 2003.

[81] A. Ultsch and H. P. Siemon. Kohonen's self organizing feature maps for ex ploratory data analysis. In *Proceedings of International Neural Network Conference (INNC'90)*, pages 305–308, 1990.

[82] J. Venna and S. Kaski. Neighborhood preservation in nonlinear projection methods: An experimental study. In G. Dorffner, H. Bischof, and K. Hornik, editors, *Proceedings of the International Conference on Artificial Neural Networks (ICANN 2001)*, pages 485–491, Berlin, 2001. Springer.

[83] J. Vesanto. Som-based data visualization methods. *Intelligent Data Analysis*, 3(2):111–126, 1999.

[84] J. Vesanto. *Data Exploration Process Based on the Self–Organizing Map.* PhD thesis, Helsinki University of Technology, Espoo (Finland), May 2002. Acta Polytechnica Scandinavica, Mathematics and Computing Series No. 115.

[85] Y. Wang, L. Luo, M. T. Freedman, and S.-Y. Kung. Probabilistic principal component subspaces: A hierarchical finite mixture model for data visualization. *IEEE Transactions on Neural Networks*, 11(3):625–636, May 2000.

[86] M. O. Ward. A taxonomy of glyph placement strategies for multidimensional data visualization. *Information Visualization*, 1(3–4):194–210, December 2002.

[87] M. O. Ward. Finding needles in large-scale multivariate data haystacks. *IEEE Computer Graphics and Applications*, pages 16–19, September/October 2004.