False neighbourhoods and tears are the main mapping defaults. How to avoid it? How to exhibit remaining ones?

Sylvain Lespinats, Michaël Aupetit

CEA, LIST, Multisensor Intelligence and Machine Learning Laboratory. F-91191 Gif-sur-Yvette, France (sylvain.lespinats@cea.fr and michael.aupetit@cea.fr).

Abstract. Tears and false neighborhoods are the defaults that may occur when a mapping is set up. Three recent articles discuss about these risks and proposed various means to detect and avoid such penalizing situations. In the present paper we link these methods and suggest a new strategy to visualize tears and false neighborhoods on a mapping by adapting well-tried tools.

Keywords: Exploratory data analysis; MultiDimensional Scaling; High-dimensional data; Error visualization; False neighborhoods and tears.

1 Introduction

Since W.S. Torgerson and his famous "embedding theorem" [34] the distance preservation is the objective of most of mapping methods (indeed, Torgerson demonstrates that Principal Component Analysis (PCA) [30, 15] objective is equivalent to look for the data projection that preserves distances "as much as possible"). In that framework, many following methods proposed to especially account for small distances, which lead to non-linear mappings. There is a very high number of methods belonging to this category (known as Non-Linear MultiDimensional Scaling or NL-MDS) and we will only cite here the most known among them. For example, Sammon's mapping [26] and Curvilinear Component Analysis (CCA) [8] interactively minimize the weighted difference between distances in the input and output space. ISOMAP [33] computes the geodesic distance [10, 32] (which can be seen as a "non-linear distance" that follows the data manifold) before linearly embedding data according to the Torgerson's method [34]. Locally Linear Embedding (LLE), [31] and related methods [3, 12] account for nearest neighbour distances in a sparse matrix and find data position in the output space according to a spectral method. Generative Topographic Mapping (GTM) [4] approaches a lower dimensional manifold to data. Gaussian Process Latent Variable Model (GP-LVM) [20] results from a probabilistic interpretation of PCA. Many others methods start from various other paradigms such as Self-Organizing Map (SOM) [16, 17] that visualizes data on a discrete grid, Non-Metric MultiDimensional Scaling (NM-MDS)

P. Lenca and S. Lallich (Eds.): QIMIE/PAKDD 2009

[18, 19] and RankVisu [22] that preserve the ranking of distances, Kernel Principal Component Analysis (KPCA) [27] that searches for non-linear relationships between variables according to the "kernel trick" [28, 29].

Several recent papers [36, 2, 21], highlight two risks while a mapping is generated from distances. These risks are named here "false neighbourhood" and "tear" according to the terminology in [21] (that corresponds to "gluing" area and "tearing" area respectively for [2] and area with low "trustworthiness" and low "continuity" for [36]). A "false neighbourhood" occurs when a large distance in the original space is associated with a small distance in the output space (the corresponding data points seems neighbours whereas they are not). Respectively, a "tear" occurs when a small distance in the original space is associated with a large distance in the output space (true neighbours are mapped apart). Please report to section 6 for some intuitive examples of "false neighbourhoods" and "tears". Obviously, "false neighbourhoods" and "tears" are expected to be avoided in mapping framework. Subsequently, [36] and [21] proposed two mapping methods designed to avoid "as much as possible" false neighbourhoods and tears.

Moreover, when such penalizing situations occur anyway, exhibiting impacted areas should be a main concern as claimed in [2]. This article proposes then a sharp mean to visualize on the mapping the true neighbourhood of a given data point. However, a local index showing the risk level for each data point would be a critical improvement.

The present article connects dimensionality reductions methods that optimize item positions by minimizing a criterion based on distances preservation. Such approach highlights pros and cons of each method and leads to considerations on detection of mappings defaults. In particular, we set up here a couple of criteria that allow visualizing and characterizing local mapping defaults. Indeed, there are actually few tools available in order to locally analyze a mapping quality.

The present paper is organized as follows. Section 2 is dedicated to Sammon's mapping and Curvilinear Component Analysis, and how either false neighbourhoods or tears are penalized in such methods. Section 3 presents and compares two recent mapping methods accounting for both defaults: Data-Driven High Dimensional Scaling and Local MultiDimensional Scaling. Section 4 refers to several usual techniques to find out remaining defaults. Section 5 describes a method allowing characterising defaults as false neighbourhood or tear. An example on an intuitive dataset is subsequently presented in section 6.

2 Mapping within avoiding false neighbourhoods or tears

Two mapping methods are particularly interesting while considering mapping from the risk of false neighbourhood and tear point of view: the Sammon's mapping [26] and the Curvilinear Component Analysis (CCA) [8].

2.1 Sammon's mapping

Historically, Sammon's mapping [26] is one of the first Non-Linear MultiDimensional Scaling methods. Its purpose is to minimize the following function:

$$E_{Sammon} = C \times \sum_{i,j} \left(\left| d_{ij} - d_{ij}^{*} \right|^{k} \times F\left(d_{ij} \right) \right) \quad (1)$$

where d_{ij} and d_{ij}^{*} represent the distances between data points *i* and *j* in the input space and output space, respectively; *F* is the so called weighting function. *F* is designed to emphasize small distances. As a consequence, function *F*: $\Re_{+} \to \Re_{+}$ has to decrease. In case of Sammon's mapping, the traditional choice is $F(x) = 1/x$, $k = 2$ and $C = \sum_{i,j} d_{ij}$ (please note that *C* is a constant and has no impact on the resulting mapping).

In case where data points lie on a low-dimensional non-linear manifold, emphasizing small distance is expected to allow "unrolling" the dataset.

Although this idea is obviously powerful (after 40 years, the Sammon's mapping is still used and inspired many subsequent methods), it shows a major drawback while considering the risk of false neighbourhood and tears:

Sammon's mapping fairly penalizes tears, but it lightly penalizes false neighbourhoods. Indeed, let us suppose that there is a large distance between data points *i* and *j* in original space (d_{ij}), but, by misfortune, the corresponding distance in output space (d_{ij}^{*}) is small. $F\left(d_{ij} \right)$ is low and the difference between d_{ij} and d_{ij}^{*} does not much weight on E_{Sammon}. Such situation (d_{ij} high and d_{ij}^{*} low) corresponds to a false neighbourhood and could easily occur with Sammon's mapping.

2.2 Curvilinear Component Analysis (CCA)

In such framework, CCA [8] offers interesting behaviour. Indeed CCA is close to Sammon's mapping, but its weighting function relies on distance in the output space rather than in original space:

$$E_{CCA} = C \times \sum_{i,j} \left(\left| d_{ij} - d_{ij}^{*} \right|^{k} \times F\left(d_{ij}^{*} \right) \right) \quad (2)$$

where $k = 2$, $C = 1/2$ and various functions have been proposed for *F*.

The drawback highlighted in case of Sammon's mapping is fixed. However, even if false neighbourhoods are now fairly penalized, tears can easily occur: If d_{ij} is low and d_{ij}^{*} is high, $F\left(d_{ij}^{*} \right)$ is low. This case (which corresponds to a tear) is then lightly penalized.

3 Accounting for both false neighbourhoods and tears

Confronting to some datasets, false neighbourhood and tears cannot be avoided. For example, everyone knows that a perfect planisphere is unreachable: there is no mean to spread out a sphere onto a plan without tearing. However, lightly penalising one of false neighbourhood or tear can cause unnecessary defaults. Penalizing the both risks is then critical.

3.1 Data-Driven High dimensional Scaling (DD-HDS)

Related to both Sammon's mapping and CCA, DD-HDS [21] is designed to cumulate advantage of these methods while avoiding the previously presented drawbacks.

The minimized function is

$$E_{DD-HDS} = C \times \sum_{i,j} \left(\left| d_{ij} - d_{ij}^{*} \right|^{k} \times F\left(\min\left(d_{ij}, d_{ij}^{*} \right) \right) \right) \tag{3}$$

where $C = 1$ and $k = 1$ (note the choice of *k* differ for DD-HDS, mostly to be consistent with the used optimization process).

Thus, if a distance (in the original or in the output space) is low, the weighting function is high in order to account for a possible penalizing situation.

Moreover, DD-HDS is yet the only one mapping method that takes account for the "concentration of measure phenomenon" (one of the most important phenomena belonging to the famous "curse of dimensionality") [11, 1] through the weighting function: *F* is proposed to be a sigmoid function adjusted on the original distance distribution. DD-HDS is then an efficient method for mapping (especially in case of high dimensional data). The main drawback is (so far) the relatively high computational time.

3.2 Local MultiDimensional Scaling (Local MDS)

The Local MDS [36], is also closely related to Sammon's mapping and CCA. It offers to the user a trade-off tuning between penalization of false neighbourhoods and tears through a parameter (noted λ in the following).

$$E_{LMDS} = C \times \sum_{i,j} \left(\left| d_{ij} - d_{ij}^{*} \right|^{k} \times \left(\lambda \times F\left(d_{ij} \right) + \left(1 - \lambda \right) \times F\left(d_{ij}^{*} \right) \right) \right) \tag{4}$$

where $k = 2$, $C = 1/2$ and $F(x) = 1$ if *x* is lower than a chosen σ and $F(x) = 0$ else. For $\lambda = 0$ the Local MDS corresponds to CCA and for $\lambda = 1$ the Local MDS corresponds to Sammon's mapping (except regarding the function *F*).

3.3 DD-HDS versus Local MDS

As shown above, solutions proposed by Local MDS and DD-HDS derived from a similar analysis. Although, they are somewhat close, advantage and drawback of each one should be highlighted.

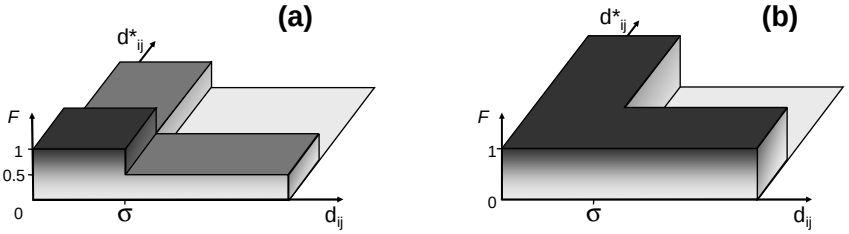
Local MDS proposes a trade-off between risks of false neighbourhood and tear. A user control (λ) allows balancing these risks. Moreover, placing mappings on a plot that quantify mapping defaults in terms of "trustworthiness" and "continuity" with varying the λ value displays a curve which can be related to a ROC curve (Receiver Operating Characteristic). ROC curves are known to be very practical in order to select an optimal decision. To the opposite, DD-HDS does not allow such control.

Because DD-HDS equally penalizes false neighbourhoods and tears, it should be compared to Local MDS with $\lambda = 0.5$.

Note that, because *F* is a decreasing function,

$$F\left(\min\left(d_{ij}, d_{ij}^{*}\right)\right) = \max\left(F\left(d_{ij}\right), F\left(d_{ij}^{*}\right)\right) \quad (5)$$

Comparison between techniques accounting for false neighbourhoods and tears relies to comparison of weights: $\max\left(F\left(d_{ij}\right), F\left(d_{ij}^{*}\right)\right)$ in case of DD-HDS and $\left(F\left(d_{ij}\right) + F\left(d_{ij}^{*}\right)\right)/2$ in case of Local MDS. For sake of simplicity, function *F* is chosen as $F(x) = 1$ if $x < \sigma$ and $F(x) = 0$ else (as proposed in Local MDS).



(a) Local MDS (b) DD-HDS

Fig. 1. Weighting functions *F* (vertical axis), according to distance in original space (x-axis) and distance in output space (y-axis).

Let us consider these functions according to distances (Fig. 1).

We can observe that when one distance is small and the corresponding one is large (this corresponds to a false neighbourhood or a tear), the Local MDS weighting function is lower than when both distances are small together. To the opposite, the DD-HDS weighting function is maximal on the whole area where a distance is small. In other words, the use of maximum corresponds to the "OR" logical operation: the weighting function is at its maximum if a false neighbourhood OR a tear is found. As a consequence, a maximum effort is given in order to avoid simultaneously both false neighbourhoods AND tears.

4 Visualizing defaults in mappings.

Even if many solutions to visualize defaults in SOM have been developed, less solution exists in case of distances preserving methods.

Of course, each proposed stress corresponds to an index that quantifies the global mapping quality. The parting in two index called "trustworthiness" and "continuity" proposed in [35] and [36] allows for placing mappings on a plot according to its ability to avoid false neighbourhoods and tears.

The Shepard Diagram [18, 19, 7, 8] plots distances in original space versus distances in output space. It allows a totally model-free observation of the distance preservation according to distances. However, it is also a global analysis: the location of errors cannot be deduced from such diagram.

Few methods have been proposed in order to locally observing mapping defaults. In this framework, [6] proposes to analyse the stability of areas on mappings initialised randomly; and [37] compares surfaces of triangles resulting from a Delaunay triangulation in the output space to surfaces of corresponding triangles in original space.

For a consistent review, please report to [2].

4.1 Distortion visualization on Voronoï cells

A local comparison of a given data point neighbourhoods is made possible by [2]. In this method, the Delaunay triangulation is computed in the output space. Each Voronoï cell is then coloured according to the proximity of the related data point in the output space (light colours for cells related to close data points). If the chosen data point is fairly mapped, every light cell will lie together, apart from dark cells. Else, if some dark cells are embedded close to the chosen data point, it corresponds to a false neighbourhood; if some light cells are apart, it corresponds to a tear.

This method has the great advantage that the true neighbourhood of a given data point is immediately assessable in the mapping through an intuitive picture. However, the necessity of choosing a point of view makes it use somewhat irksome when we do not know a priori where to looking for a default.

4.2 Pressure defined in DD-HDS algorithm

The DD-HDS mapping is achieved by optimizing a stress (eq. 3) thanks to a Force Directed Placement (FDP) [13, 14, 9] algorithm. The FDP algorithm simulates a spring system: each data point corresponds to a mass and springs rely each couple of masses. Spring lengths at rest equal expected distances (i.e. original distances), in order to make original distances resemble to output ones after relaxation of the system. The spring stiffness allows for accounting for the weighting function F. The relaxation of the system is supposed to minimise the stress. Such algorithm is known to be robust to local minima and is somewhat few time consuming [24, 25]. The popularity of such algorithm increases in mapping community [5, 24, 23, 25, 21, 22].

An other advantage is that FDP allow defining the "pressure" that locally quantifies the stress level: the pressure on a data point is the sum of strength of forces applied on the corresponding mass [21]. Truly, this does not strictly correspond to the academic definition of a pressure, but this term allows an intuitive understanding of this index: the "pressure" is the quantity of forces applied on a data point.

Nevertheless, the pressure concept does not need an FDP optimisation, and could be defined from any mapping. Given a data point *i*, the pressure would be:

$$P_{DD-HDS}(i) = C \times \sum_j \left(\left| d_{ij} - d_{ij}^* \right|^k \times F(.) \right) \quad (6)$$

Moreover, because the weighting function *F* can easily be customized, Sammon's mapping, CCA, Local MDS or DD-HDS model (and so on) could be considered as well according to the pressure concept. The combination of the mapping and the pressure of data points through a greyscale allows for locally observing the stress level.

5 How to exhibit and characterize defaults?

Plotting DD-HDS pressure allows displaying area where the mapping shows defaults, but it is not informative about the nature of the default.

Furthermore, even if Sammon's and CCA weighting function have shown there limitation in order to drive the mapping [36, 21], they can be useful together to evaluate a mapping. Indeed, pressures related to Sammon's mapping and CCA can easily be defined:

$$P_{Sammon}(i) = C \times \sum_j \left(\left| d_{ij} - d_{ij}^* \right|^k \times F\left(d_{ij}\right) \right) \quad (7)$$

$$P_{CCA}(i) = C \times \sum_j \left(\left| d_{ij} - d_{ij}^* \right|^k \times F\left(d_{ij}^*\right) \right) \quad (8)$$

Due to the high similitude between formulas (7) and (8), values for P_{Sammon} and P_{CCA} can be compared if the choice of C, k and F are shared by two pressures.

For reason discussed in section 2.1 and 2.2, P_{CCA} is an efficient mean to detect false neighbourhoods and P_{Sammon} catches tears.

Note: DD-HDS pressure equals to

$$P_{DD-HDS}(i) = C \times \sum_j \left(\left| d_{ij} - d_{ij}^* \right|^k \times F\left(\min\left(d_{ij}, d_{ij}^*\right)\right) \right) \quad (9)$$

which corresponds to

$$P_{DD-HDS}(i) = \max\left(P_{Sammon}(i), P_{CCA}(i)\right) \quad (10)$$

6 Example

A classical test for mapping methods is the openbox dataset. Data points lie on sides of a 3-dimensional cube without upper side (Fig. 2, right insert with grey background) and are embedded in 2-dimensional spaces. These mappings are presented in Fig. 2 according to E_{Sammon} and E_{CCA} stresses (curves and points) and P_{Sammon} and P_{CCA} pressures (small inserts).

For a fair comparison, every methods used the same values for C and k ($C=1, k=2$) and function F is chosen to be a rectangular function as proposed in Local MDS.

Situating mapping in a plot according to E_{Sammon} and E_{CCA} is in the spirit of Venna and Kaski "trustworthiness" and "continuity" visualisation [35, 36]. Note that, as expected, Sammon's mapping avoids tears to the cost of possible false neighbourhoods and CCA avoids false neighbourhoods to the cost of possible tears (see section 2). DD-HDS and Local MDS account for both defaults (section 3.1 and 3.2).

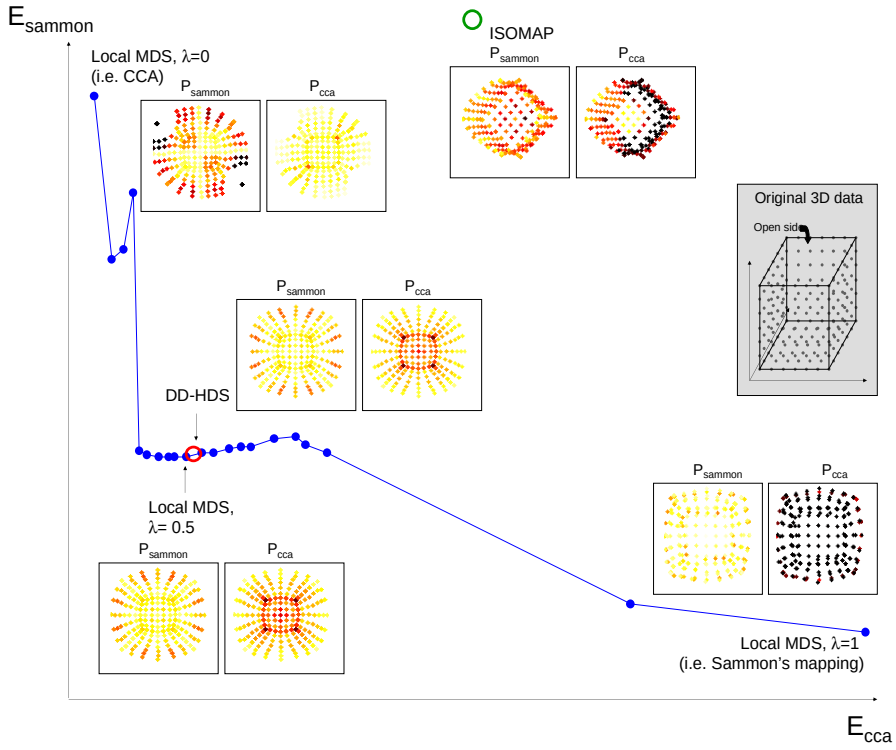


Fig. 2. 2D Mapping of a 3D open box (right insert with grey background) resulting from various methods as a function of Sammon's mapping and Curvilinear

Components Analysis stresses. ISOMAP and DD-HDS rely to empty circles, Local MDS for a λ varying from 0 to 1 (with 0.05 as step) rely to full circles linked by lines. CCA corresponds to Local MDS with $\lambda = 0$ and Sammon's mapping corresponds to Local MDS with $\lambda = 1$. The most on the left the mapping, the most false neighbourhoods are avoided; the lowest the mapping, the most tears are avoided.

Mapping reached by five methods (ISOMAP, DD-HDS, CCA, Samon's mapping and Local MDS with $\lambda = 0.5$), are provided in five couples of inserts. Positions of items are similar for each couple, but colours are different: in left inserts, darker the data points, higher the pressure related to Sammon's stress (eq. 7); in right inserts, data darker the data points, higher the pressure related to CCA stress (eq. 8). Colorscales are similar for every mapping.

For five mappings (Sammon's mapping, CCA, Local MDS with $\lambda = 0.5$, DD-HDS and ISOMAP), pressures related to Sammon's and CCA stresses are presented in small inserts. It allows visualizing tears (left inserts, which corresponds to P_{Sammon}) and false neighbourhoods (right inserts, which corresponds to P_{CCA}) in the mappings (dark areas correspond to defaults). In case of CCA, two tears can easily be observed and correspond to black areas in left insert; no false neighbourhood appears (right insert). Sammon's mapping has smashed sides on the bottom face. It results many false neighbourhoods (highlighted by dark data points in left inserts) but few tears: P_{Sammon}, is low everywhere in the mapping (right insert). Two sides have been projected on the bottom by ISOMAP, creating false neighbourhoods. DD-HDS and Local MDS with $\lambda = 0.5$ reach close mappings. Light stretch can be observed at the top of the box (darker areas in left inserts) and some compressions in the bottom side (especially on the corners, right inserts).

7 Conclusion

The objective of the present paper is not to grade mapping methods. Indeed, it is obvious that: 1) the presented methods are closely related and will often reach close results (just as Local MDS and DD-HDS with the openbox dataset); 2) there respective originalities provide to each method its own advantage that should be exploited according to the situation.

On the one hand, a good point for DD-HDS on Local MDS when there is no reason for a priori favour false neighbourhood or tear is its capacity to accounting for both false neighbourhoods and tears within one single parameter. This advantage should not be neglected; indeed it permits to spare the balancing parameter. On the other hand, the Venna and Kaski's parameter give to Local MDS a unique chance to visually appreciate the trade-off between false neighbourhood and tears. In that framework, because the choice of λ from resulting mapping falls under the data expert responsibility, tools presented here could be a useful supplementation.

The pressures proposed in section 5 are contenting themselves to finding out and characterizing mapping defaults. Such procedure should be combined with

visualization of neighbourhoods technique proposed in [2]. Indeed, P_{Sammon} and P_{CCA} can guide the user to the items for which such analysis is the most relevant.

References

1. Aggarwal C.C., Hinneburg A. and Keim D.A., On the surprising behavior of distance metrics in high dimensional space, in J.V. Bussche and V. Vianu Eds. Lecture Notes In Computer Science, ser. 1973, (Berlin, Germany, Springer-Verlag, 2001), 420–434.
2. Aupetit M., Visualizing distortions and recovering topology in continuous projection techniques" Neurocomputing, vol. 10, no. 7-9 pp. 1304–1330, 2007.
3. Belkin M. and Niyogi P., "Laplacian eigenmaps for dimensionality reduction and data representation, Neural Comput., vol. 15, pp. 1373–1396, 2003.
4. Bishop C.M., Svensén M., and Williams C.K.I, GTM: The generative topographic mapping," Neural Comput., vol. 10, pp. 215–234, 1998.
5. Chalmers M., A linear iteration time layout algorithm for visualizing high-dimensional data, in Proc. 7th Conf. Visualization, R. Yagel and G. M. Nielson, Eds., San Francisco/Los Alamitos, CA, 1996, pp. 127–132.
6. Davidson G.S., Wylie B.N., Boyack K.W., Cluster stability and the use of noise in interpretation of clustering, in: Proceedings of IEEE Information Visualization (InfoVis'Ol), 2001, pp. 23–30.
7. Demartines P., Mesures d'organisation du réseau de Kohonen. presented at Congrès Satellite du Congrès Européen de Mathématiques: Aspects Théoriques des Réseaux de Neurones, Paris, France, 1992.
8. Demartines P. and Hérault J., Curvilinear component analysis: A selforganizing neural network for nonlinear mapping of data sets, IEEE Trans. Neural Netw., vol. 8, no. 1, pp. 148–154, Jan. 1997.
9. Di Battista G.D., Eades P., Tamassia R., and Tollis I.G., Graph Drawing: Algorithms for the Visualization of Graphs. Englewood Cliffs, NJ: Prentice-Hall, 1999.
10. Dijkstra E.W., A note on two problems in connection with graphs, Numerisch Mathematik, vol. 1, pp. 269–271, 1959.
11. Donoho D.L., High-dimensional data analysis: The curses and blessings of dimensionality, Los Angeles, CA, 2000, Amer. Math. Soc. Lecture: "Math challenges of the 21st century" [Online]. Available: http://www-stat.stanford.edu/~donoho/
12. Donoho D.L. and Grimes C., Hessian eigenmaps: Locally linear embedding techniques for high-dimensional data, Proc. Nat. Acad. Sci., vol. 100, pp. 5591–5596, 2003.
13. Eades P., A heuristic for graph drawing, in Proc. 13th Manitoba Conf. Numer. Math. Comput. (Congressus Numerantium), D. S. Meek and G.H. J.V. Rees, Eds., Winnipeg, MB, Canada, 1984, vol. 42, pp. 149–160.
14. Fruchterman T. and Reingold E., Graph drawing by force-directed placement, Software-Practice Exp., vol. 21, pp. 1129–1164, 1991.
15. Jolliffe I., Principal Component Analysis, Springer-Verlag, New York, 2002.
16. Kohonen T., Self-organized formation of topologically correct feature maps, Biol. Cybern., vol. 43, pp. 59–69, 1982.
17. Kohonen T., Self-Organizing Maps., H.K.V. Lotsch, Ed. Heidelberg, Germany: Springer-Verlag, 1997.
18. Kruskal J.B., Non-metric multidimensional scaling: A numerical method, Psychometrika, vol. 29, pp. 115–129, 1964.
19. Kruskal J.B., Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis, Psychometrika, vol. 29, pp. 1–27, 1964.

20. Lawrence N.D., Probabilistic non-linear principal component analysis with Gaussian process latent variable models, J. Mach. Learn.Res., vol. 6, pp. 1783–1816, 2005.
21. Lespinats S., Verleysen M., Giron A. and Fertil B., DD-HDS: a tool for visualization and exploration of highdimensional data, IEEE Trans. Neural Netw., vol. 18, no. 5, pp. 1265-1279, 2007.
22. Lespinats S., Fertil B., Villemain P. and Herault J., Rankvisu: Mapping from the neighbourhood network. Submitted to Neurocomputing.
23. Li J.X., Visualization of high-dimensional data with relational perspective map, Inf. Visualization, vol. 3, pp. 49–59, 2004.
24. Morrison A., Ross G., and Chalmers M., Fast Multidimensional Scaling through Sampling, Springs and Interpolation, Inf. Visualization, vol. 2, pp. 68–77, 2003.
25. Paulovich F.V., Oliveira M.C.F. and Minghim R., The projection explorer: A flexible tool for projection-based multidimensional visualization, In Proceedings of XX Brazilian Symposium on Computer Graphics and Image Processing – SIBIGRAPI 2007, Belo Horizonte, Brazil, 2007. IEEE Computer Society Press. pp. 27-36.
26. Sammon J.W., A nonlinear mapping for data structure analysis, IEEE Trans. Comput., vol. C-18, no. 5, pp. 401–409, May 1969.
27. Schölkopf B., Smola A.J., and Müller K.R., Nonlinear component analysis as a kernel eigenvalue problem, Neural Comput., vol. 10, pp. 1299–1319, 1998.
28. Schölkopf B. and Smola A.J., Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond. Cambridge, MA: MIT Press, 2002.
29. Shawe-Taylor J. and Cristianini N., Kernel Methods for Pattern Analysis. Cambridge, U.K.: Cambridge Univ. Press, 2004.
30. Pearson K., On lines and planes of closest fit to systems of points in space, Philosophical Magazine n°2, p. 559-572, 1901.
31. Roweis S.T. and Saul L.K., Nonlinear dimensionality reduction by locally linear embedding, Science, vol. 290, pp. 2323–2326, 2000.
32. Skiena S., Implementing Discrete Mathematics: Combinatorics and Graph Theory with Mathematica. Reading, MA: Addison-Wesley, 1990.
33. Tenenbaum J.B., de Silva V., and Langford J.C., A global geometric framework for nonlinear dimensionality reduction, Science, vol. 290, pp. 2319–2323, 2000.
34. Torgerson W.S., “Multidimensional scaling: 1. Theory and method,” Psychometrika, vol. 17, pp. 401–419, 1952.
35. Venna J. and Kaski S., Neighborhood preservation in nonlinear projection methods: An experimental study, In G. Dor_ner, H. Bischof, and K. Hornik, editors, Proceedings of ICANN 2001, International Conference on Artificial Neural Networks, pp. 485-491, Berlin, 2001. Springer.
36. Venna J. and Kaski S., Local multidimensional scaling, Neural Networks, vol. 19, no. 6-7, pp. 889-899, 2006.
37. Warnking J., Guerin-Duguet A., Chehikian A., Olympieff S., Dojat M. and Segebarth C., Retinotopical mapping of visual areas using fMRI and a fast cortical flattening algorithm, Neuroimage vol. 11, no. 5, S646, 2000.