ESANN 2014 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 23-25 April 2014, i6doc.com publ., ISBN 978-287419095-7.
Available from http://www.i6doc.com/fr/livre/?GCOI=28001100432440.

Interactive Dimensionality Reduction for Visual Analytics

Ignacio Díaz[1], Abel A. Cuadrado[1], Daniel Pérez[1],
Francisco J. García[1] and Michel Verleysen[2] *

1- Electrical Engineering Dept. University of Oviedo
Edif. Dept. 2, campus de Viesques s/n 33204, Gijón, SPAIN
2- Univ. Catholique de Louvain - Machine Learning Group
ICTEAM/ELEN - Place du Levant, 3 1348 Louvain-la-Neuve, Belgium

Abstract. In this work, we present a novel approach for data visualization based on interactive dimensionality reduction (iDR). The main idea of the paper relies on considering for visualization the intermediate results of non-convex DR algorithms under changes on the metric of the input data space driven by the user. With an appropriate visualization interface, our approach allows the user to focus on the relationships among dynamically selected groups of variables, as well as to assess the impact of a single variable or groups of variables in the structure of the data.

1 Introduction

Many problems today involve the analysis of large datasets, which also contain a very large number of variables from which the user should be able to find meaningful relationships to acquire knowledge. The mere fact of obeying laws, rules or restrictions arising from the problem domain, leads to dependencies that make the intrinsic dimensionality of the data to be much smaller. Dimensionality reduction (DR) algorithms –see [1] for a review– are able to find low dimensional latent structures hidden in high dimensional data and produce a mapping on a low dimensional space that preserves the underlying structure of data. They are extremely useful tools in the field of visual analytics, since they provide an advanced way for *spatialization* of data, allowing to create visual representations where spatial proximity between two items $\mathbf{y}_i$ and $\mathbf{y}_j$ in the visualization represents similarity between $\mathbf{x}_i$ and $\mathbf{x}_j$ in a high dimensional space.

Another key ingredient in visual analytics is *interaction.* Interaction techniques –zoom, pan, brushing, etc.– allow the user to reconfigure the visualization to focus on the interesting aspects of data or to discard information that is irrelevant to the available knowledge of the user. In this paper we present a novel approach for data visualization that suggests a low level integration of user interaction into the DR computation and visualization process, by means of the so-called interactive dimensionality reduction (iDR). In section 2 we describe the iDR approach as a user-driven visualization of intermediate results of DR algorithms, highlighting some of its potential applications, such as the analysis of time-varying datasets or sensitivity analysis of data dependencies. In section

*The authors would like to thank financial support from the Spanish Ministry of Economy (MINECO) and FEDER funds from the EU.

3 we describe an application demo of the iDR idea for the visual analysis of fault states in a rotating machine. Finally, section 4 concludes the paper.

2 The interactive Dimensionality Reduction approach

In the typical procedure to use DR algorithms for visual analytics, interaction is often done after DR computation on the input dataset. The user sets up an initial configuration for the DR algorithm, runs it until convergence and, after N iterations, the output results are used to produce a visualization. The user may later use interaction techniques to reconfigure this visualization or even decide to run the DR algorithm again using another parameterization, starting the cycle again –see for instance [2]. This approach can be thought of as a *batch mode interaction* scheme for DR visual analytics.

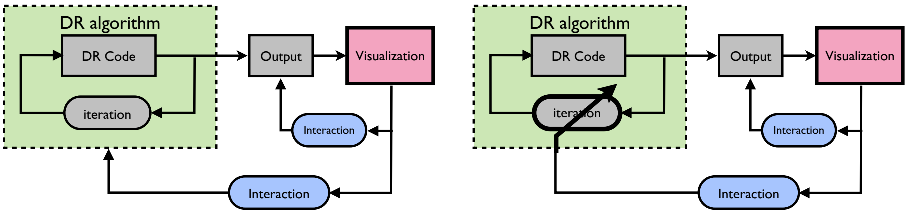

Fig. 1: Batch mode interaction scheme (left) vs. the iDR approach (right)

However, interaction can go far beyond this approach, allowing the user to take full control of the DR behavior by means of *iterative reconfiguration of computational algorithms* [3]. The right picture in Fig. 1, shows the main idea of this approach for DR analysis, where the intermediate results are used to produce a visualization at *each* iteration. The result is a dynamically changing visualization that allows the user to track changes in the resulting projection under changes in the problem formulation, such as, for instance, user-driven changes of the metric in the input space (e.g. by modifying the weights of the input variables), or under time-varying input data (e.g. in dynamic processes where the elements of the input dataset change with time). Despite this approach is still rather unexplored, a few related works can be found, as an interactive version of PCA [4] and an interactive learning of distance functions [5].

2.1 Applications of the iDR approach

To allow interaction, we shall consider iterative algorithms, such as Stochastic Neighbor Embedding (SNE) [6] or the Neigborhood Retrieval Visualizer (NeRV) [7]. For simplicity, let's consider a block diagram of the SNE algorithm for the k-th iteration –see Fig. 2. Some of the inputs –data or parameters– to the algorithm can change or be changed by the user at each iteration.

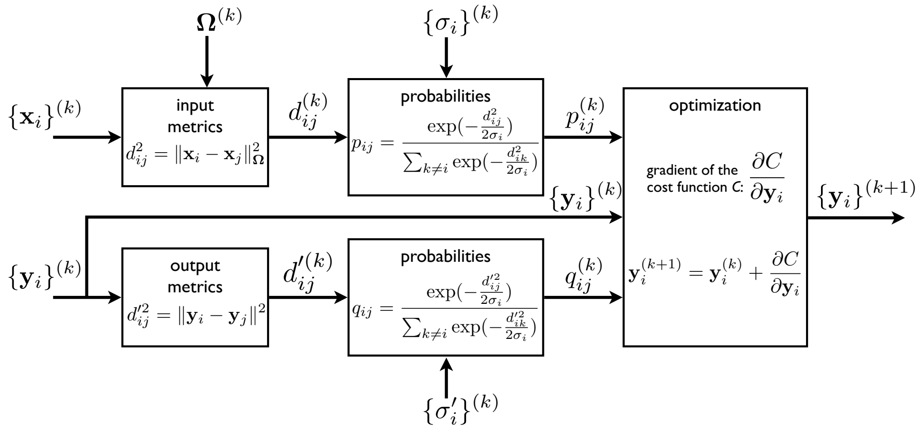
$\mathbf{\Omega}^{(k)}$
$\{\sigma_i\}^{(k)}$
$\{\mathbf{x}_i\}^{(k)}$
input
metrics
$d_{ij}^2 = \|\mathbf{x}_i - \mathbf{x}_j\|_{\mathbf{\Omega}}^2$
$d_{ij}^{(k)}$
probabilities
$p_{ij} = \frac{\exp(-\frac{d_{ij}^2}{2\sigma_i})}{\sum_{k\neq i}\exp(-\frac{d_{ik}^2}{2\sigma_i})}$
$p_{ij}^{(k)}$
optimization
gradient of the cost function C: $\frac{\partial C}{\partial \mathbf{y}_i}$
$\{\mathbf{y}_i\}^{(k)}$
$\{\mathbf{y}_i\}^{(k+1)}$
$\{\mathbf{y}_i\}^{(k)}$
output
metrics
$d_{ij}'^2 = \|\mathbf{y}_i - \mathbf{y}_j\|^2$
$d_{ij}'^{(k)}$
probabilities
$q_{ij} = \frac{\exp(-\frac{d_{ij}'^2}{2\sigma_i})}{\sum_{k\neq i}\exp(-\frac{d_{ik}'^2}{2\sigma_i})}$
$q_{ij}^{(k)}$
$\mathbf{y}_i^{(k+1)} = \mathbf{y}_i^{(k)} + \frac{\partial C}{\partial \mathbf{y}_i}$
$\{\sigma_i'\}^{(k)}$

Fig. 2: Block diagram of the SNE algorithm at the k-th iteration

2.1.1 *Time-varying input datasets*

Let the input dataset be $\{\mathbf{x}_i\}$, where $\mathbf{x}_i$ is a vector with n features $x_{i1}, x_{i2}, \ldots, x_{in}$. The Q elements of the input dataset may change over time, resulting in a time-varying dataset $\{\mathbf{x}_i\}^{(k)}$ at time k. Using the DR algorithm to visualize time-varying datasets allows the user to understand not only the main relationships and structure of data but also how these relationships evolve along time.

2.1.2 *Changes in the metric of the input space*

A simple but powerful interaction feature can stem from user-driven change in the input space metric $\mathbf{\Omega}$. Let's consider the following weighted norm in the input data space

$$\|\mathbf{x}\|_{\mathbf{\Omega}}^2 := \sum_r \sum_s x_r \omega_{rs} x_s. \tag{1}$$

Using the metric induced by the previously defined weighted norm, the input distances between input points $\mathbf{x}_i$ and $\mathbf{x}_j$ would be $d_{ij} = \|\mathbf{x}_i - \mathbf{x}_j\|_{\mathbf{\Omega}}$. Let's consider the special case where the weight matrix $\mathbf{\Omega}$ is diagonal $\mathbf{\Omega} = \mathrm{diag}(\omega_1, \omega_2, \ldots, \omega_n)$, where we have dropped the repeated index in ω_{qq} and used ω_q instead, to simplify notation. With this choice and an appropriate visual interface, the user can vary the contribution of any variable to the DR projection by changing the values of the weights ω_q. Any variable q for which $\omega_q = 0$ would not contribute to the DR projection (resulting in a pseudonorm). If a new weight matrix $\mathbf{\Omega}^{(k)}$ is used by the DR algorithm at every iteration k, the DR algorithm will result in a smooth transition (depending on the learning rate) to a new projection that considers the relationships outlined by the new weight matrix. This mechanism allows the user to quickly explore dependencies among the variables by selecting subgroups in the interface. Moreover, since under changes in the metric $\mathbf{\Omega}^{(k)}$ the algorithm converges smoothly to a new stable state –that is, the DR projection under the new metric–, changes can be tracked by the user, allowing to establish links and find differences between the new projection and the former one.

Interacting with the weights ω_q, the user can explore several kinds of nonlinear correlations between the variables. Suppose that the user has chosen a set of K nonzero weights $\{\omega_{q_1}, \omega_{q_2}, \ldots, \omega_{q_K}\}$. If a 1-dimensional structure –i.e. a snake-shaped figure– emerges after convergence in the projection, it reveals a mutual nonlinear dependency on an independent parameter t of the type $x_{q_1} = f_{q_1}(t)$, $x_{q_2} = f_{q_2}(t)$, $\ldots$, $x_{q_K} = f_{q_K}(t)$. Note that this information is much more general than the one provided by a linear correlation coefficient or the more general nonlinear correlations observable in scatter plots, which can only be visualized for two variables in a single scatter plot.

A further collateral benefit of this kind of DR interaction is *sensitivity analysis*. Whenever the user modifies a single weight ω_q, the input distance pattern d_{ij} becomes more sensitive to variable q. This will be reflected as large displacements in the projections of all elements that have significant differences in variable q with respect to the other ones. This sensitivity analysis is not restricted to a single variable. Eventually, if the interface allows it, the user could change the weights $\{\omega_{q_1}, \omega_{q_2}, \ldots, \omega_{q_M}\}$ of a group of M variables at the same time to discover elements that differ significantly in any of the variables $x_{q_1}, x_{q_2}, \ldots, x_{q_M}$. Moreover, the displacement trajectories should be different for elements with different patterns of variation within the group.

2.1.3 Interactive feature space transformations

Feature space transformations [8] allow improving the quality of an existing embedding in terms of both structural preservation and class separation. One simple feature extension scheme, for instance, is to augment each element $\mathbf{x}$ with an extended feature set $\bar{\mathbf{x}}_{c(\mathbf{x})}$ equal to the centroid of the class $c(\mathbf{x})$ it belongs to, thus forming an extended vector $\mathbf{x}_e = [\mathbf{x}, \bar{\mathbf{x}}_{c(\mathbf{x})}]$. The DR projection of $\mathbf{x}_e$, therefore contains class information, resulting in a more meaningful projection. A user-driven variant of this approach, suitable for interaction, could involve a weight factor λ

$$\mathbf{x}_e(\lambda) = [(1-\lambda)\mathbf{x},\, \lambda\bar{\mathbf{x}}_{c(\mathbf{x})}]$$

letting the user modify $\lambda^{(k)}$ and projecting $\mathbf{x}_e(\lambda^{(k)})$ at iteration level, the user can control the balance between class separation and structural preservation. As a result the user can set the optimum point or even move it to gain insight and find connections between data structure and class knowledge.

3 Application demo: fault analysis of AC motor

A javascript application using the iDR approach was developed using *processing.js* (`http://processingjs.org`), for the analysis of vibration data in a 4kW, 2 pole-pair asynchronous motor, where three vibration signals –measured in the three axes $a_x(t)$, $a_y(t)$, $a_z(t)$– and two phase currents $i_R(t), i_S(t)$ were recorded at a 5000 Hz sample rate. Two kinds of asymmetries were tested: 1) vibrations at the rotation frequency produced by a mechanical eccentricity, resulting in a main vibration component near 25 Hz; and 2) an electrical imbalance caused by a variable electrical load –impedances in the range $(0\Omega, \infty\Omega)$– in one of the phases, causing vibration mainly at twice the 50 Hz line frequency, that is, 100 Hz. Both normal and fault data, under combinations of these faults, were recorded. Thus, energies in bands of 25Hz and 100Hz were computed using the FFT algorithm for the five variables, leading to a 10-dim feature vector that describes the fault state of the machine.

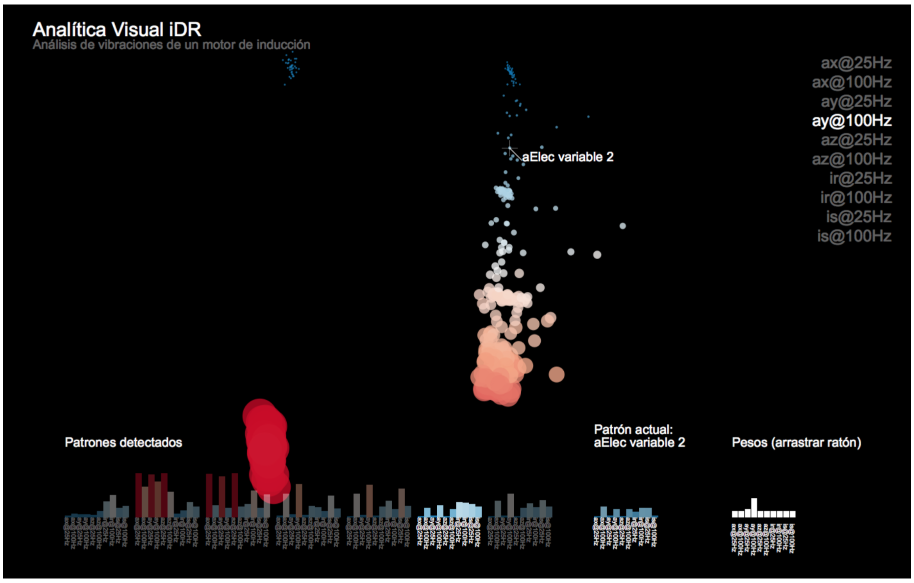

Fig. 3: Screenshot of the iDR interface

The application demo interface[1] –see Fig. 3– iteratively computes and visualizes the SNE algorithm under tunable weights for the input variables. It shows a dynamic “fault map” consisting of an animated scatterplot of the feature vectors, using a uniform color scale –blue=low, red=high– and size to describe the values of the feature highlighted in the right list. When the user modifies the weight of any variable –by dragging the bars of a small barchart of weights on the bottom right corner–, the fault map gets automatically “reordered”, grouping the states according to the weight-dependent similarity metric configured by the user. Starting from zero weights for all variables, if the user increases the weight of a variable that characterizes a given fault –e.g. 100 Hz of $a_x(t)$, for the electrical imbalance– its states gradually emerge from the remaining states, producing a new separated cluster. The same occurs with mechanical imbalance states if a 25 Hz vibration band is modified –e.g. $a_x(t), a_y(t)$ or $a_z(t)$. Further

[1] Demo available at `http://isa.uniovi.es/~idiaz/demos/iDR-vibracionesMotor/`

changes in other variables lead to subtle variations on the cluster structure. In addition, by hovering on a point, the user gets contextual information, including the label of the corresponding element, a small bar chart with the actual feature vector and another bar chart with the best matching pattern from a set of characteristic patterns obtained using the neural gas algorithm.

4 Conclusions

In this work we have presented a novel approach for data visualization based on the so-called interactive dimensionality reduction (iDR). The main role of interaction is to involve the user in the analysis loop by allowing her tuning the visual representation to focus in the interesting parts of the data. However, the main idea presented here relies on taking interaction beyond, by changing the conditions and/or input information to a running DR visualization. We have outlined here three potential applications of this idea: tracking time-varying input datasets, interactively exploring nonlinear correlations and combining the DR visualization with class information, including a demo application to give a gist of the core idea. While user studies should still be done to assess its effectiveness, it opens a wide field of exploration, lying in the intersection of data visualization and machine learning.

References

[1] J.A. Lee and M. Verleysen. *Nonlinear dimensionality reduction.* Springer Verlag, 2007.

[2] Stephen Ingram, Tamara Munzner, Veronika Irvine, Melanie Tory, Steven Bergner, and T Möller. Dimstiller: Workflows for dimensional analysis and reduction. In *Visual Analytics Science and Technology (VAST), 2010 IEEE Symposium on*, pages 3–10. IEEE, 2010.

[3] Jaegul Choo and Haesun Park. Customizing computational methods for visual analytics with big data. *Computer Graphics and Applications, IEEE*, 33(4):22–28, 2013.

[4] Dong Hyun Jeong, Caroline Ziemkiewicz, Brian Fisher, William Ribarsky, and Remco Chang. ipca: An interactive system for pca-based visual analytics. *Computer Graphics Forum*, 28(3):767–774, 2009.

[5] Eli T Brown, Jingjing Liu, Carla E Brodley, and Remco Chang. Dis-function: Learning distance functions interactively. In *Visual Analytics Science and Technology (VAST), 2012 IEEE Conference on*, pages 83–92. IEEE, 2012.

[6] Geoffrey E Hinton and Sam T Roweis. Stochastic neighbor embedding. In *Advances in neural information processing systems*, pages 833–840, 2002.

[7] Jarkko Venna, Jaakko Peltonen, Kristian Nybo, Helena Aidos, and Samuel Kaski. Information retrieval perspective to nonlinear dimensionality reduction for data visualization. *The Journal of Machine Learning Research*, 11:451–490, 2010.

[8] Matthias Schaefer, Leishi Zhang, Tobias Schreck, Andrada Tatu, John A Lee, Michel Verleysen, and Daniel A Keim. Improving projection-based data analysis by feature space transformations. In *IS&T/SPIE Electronic Imaging*, pages 86540H–86540H. International Society for Optics and Photonics, 2013.