ICANN'2002 proceedings - International Conference on Artificial Neural Networks
Madrid (Spain), 28-30 August 2002, Springer, Lecture Notes in Computer Science 2415,
J.R. Dorronsoro ed., ISBN 3-540-44074-7, pp. 933-938

Nonlinear Projection with the Isotop Method

John A. Lee[*] and Michel Verleysen[**]

Université catholique de Louvain
Laboratoire de Microélectronique
Place du Levant, 3, B-1348 Louvain-la-Neuve
verleysen@dice.ucl.ac.be

Abstract. Isotop is a new neural method for nonlinear projection of high-dimensional data. Isotop builds the mapping between the data space and a projection space by means of topology preservation. Actually, the topology of the data to be projected is approximated by the use of neighborhoods between the neural units. Isotop is provided with a piecewise linear interpolator for the projection of generalization data after learning. Experiments on artificial and real data sets show the advantages of Isotop.

1 Introduction

Often the analysis of numerical data raises some difficulties because of their high dimensionality. This problem can be attenuated by projection techniques such as the well-known Principal Component Analysis (PCA, [6]). However, PCA is a strictly linear method that is unable to detect nonlinear dependencies between variables. Numerous nonlinear projection methods have been created to address this issue. For example, the nonmetric Multidimensional Scaling (MDS, [12]) and Sammon's nonlinear mapping (NLM [11]) are based on the preservation of either pairwise dissimilarities or Euclidean distances. Neural versions of the NLM, like Curvilinear Component Analysis (CCA, [3, 4]), generally show better performance, particularly when they do not use the traditional Euclidean metrics [9, 13]. Finally, nonlinear projection can be achieved by the Self-Organizing Map (SOM, [8, 14, 10]), that works with true topology preservation rather than the more constraining distance reproduction. In this framework, Isotop is a new nonlinear projection algorithm combining the advantages of the SOM and the distance preserving algorithms like Sammon's NLM.

The following of this paper describes how Isotop works (Sect. 2) and shows some results of experiments (Sect. 3). Finally, Sect. 4 draws the conclusions and sketches some perspectives for future developments.

[*] This work was realized with the support of the 'Ministère de la Région wallonne', under the 'Programme de Formation et d'Impulsion à la Recherche Scientifique et Technologique'.

[**] M.V. works as a senior research associate of the Belgian FNRS.

2 Description of Isotop

Isotop proceeds in three stages: vector quantization of the raw data, linking of the neighboring prototypes and mapping to the projection space.

Assuming that N data patterns are stored in matrix X (one row x_i per pattern), Isotop first proceeds with a vector quantization (VQ) step. For example, the well-known Competitive Learning (CL, [1]) may be used. This neural algorithm transforms the raw data into a set of n representative units called prototypes. These are stored as the rows p_j of matrix P and may be seen as neurons. Formally, P is initialized with randomly selected rows in the data set and is then modified adaptively in several epochs (sweeps of the data set). For each data row x_i the closest prototype p_* (best matching unit) is modified according to the rule:

$$p_* \leftarrow \alpha^t (x_i - p_*) \tag{1}$$

where α_t is a learning rate with values between 0 and 1, decreasing as epochs go by.

The second step of Isotop consists in defining neighborhood relations between the prototypes. Actually, this task is realized by linking prototypes that are close to each other. For example, each prototype can be linked with the k closest ones, with k being a predetermined constant. Another possible method links each prototypes with the ones lying closer than a fixed radius ϵ. In both cases, the result is a connected structure, where each link can be characterized by its Euclidean length.

The third and last step of Isotop builds the mapping from the d-dimensional data space to the p-dimensional projection space. It uses only the neighborhoods defined by the links. The link lengths define distances $\delta_{j,k}$ between direct neighbors j and k. These distances can be extended to any pair (k, l) of prototypes by summing the lengths associated with the shortest path [5] walking from k to l. Such distances help to build matrix M whose rows m_j correspond to those of P and contain the coordinates of the neurons in the projection space. Matrix M is initialized randomly around zero. Next, the twisted structure of links has to be unfolded in the projection space, in order to retrieve the same neighborhoods as in the data space. This goal is reached by randomly stimulating the mapped prototypes. At time t, stimulus $g(t)$ is drawn from a zero-mean, unit-variance, p-dimensional Gaussian distribution. Defining the best matching unit (BMU) as the closest mapped prototype m_* from the stimulus, all prototypes m_j are then moved towards the stimulus. The movement of each mapped prototype becomes smaller and smaller as its neighborhood distance from the BMU grows. Formally, adjustments are made according to:

$$m_j \leftarrow \alpha^t \nu_j^t (g^t - m_j) \tag{2}$$

where α^t is a learning rate with time-decreasing values between 0 and 1. The neighborhood factor ν_j is defined as:

$$\nu_j^t = \exp^{\frac{1}{2}\left(\frac{\delta_{*,j}}{\lambda^t}\right)^2} \tag{3}$$

where $\delta_{*,j}$ is the neighborhood distance from the BMU p_* to the prototype p_j and where λ^t is a time-decreasing neighborhood width.

The choice of a Gaussian distribution for the network stimulation is arbitrary in the absence of a priori information about what would be the best distribution of the mapped prototypes. The Gaussian distribution is just an ‘average’ choice as is the choice of a uniformly distributed rectangular grid when using a SOM. An advantage of the Gaussian pdf is its smoothness, by comparison to the sharp edges of a SOM grid.

Once the three learning stages are completed, Isotop has build a mapping between the data space and the projection space, resulting from the correspondence between the rows of P and the rows of M. Starting from this discrete representation of the mapping, Isotop can work in conjunction with a piecewise linear interpolator. Such device generalizes the mapping and projects new data.

3 Experiments

Experiments on artificial and real data sets have been conducted in order to compare Isotop with the SOM, which is the most used neighborhood preserving mapping algorithm for nonlinear projection.

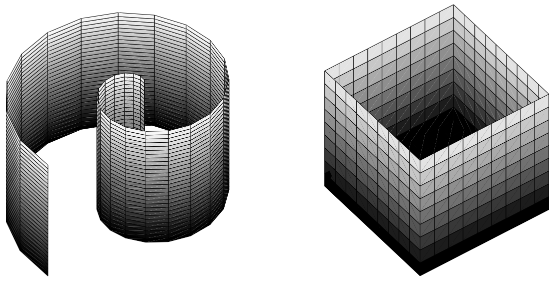

Fig. 1. Artificial data sets: Swiss roll (left) and open box (right)

The implemented SOM algorithm uses rectangularly shaped grids, with a hexagonal neighborhood structure. The neighborhood factor is an exponentially decreasing function of the grid distance from the best matching unit, as described in [7]. Isotop uses Competitive Learning for the VQ step.

As an illustrative example, the ‘Swiss roll’ data set (see left of Fig. 1) contains 10000 samples. It can be unfolded by Isotop (300 prototypes) and by the SOM (30 × 10 prototypes). The projections are shown in Fig. 2. Despite of a careful parameterization, a grid shaped SOM poorly unfolds the data because its learning process occurs in the data space, whose dimensionality (3D) is higher than the one of the grid (2D). The result is a map that jumps from one spire to the following one in the Swiss roll, as shown in Fig. 2. Isotop does not suffer

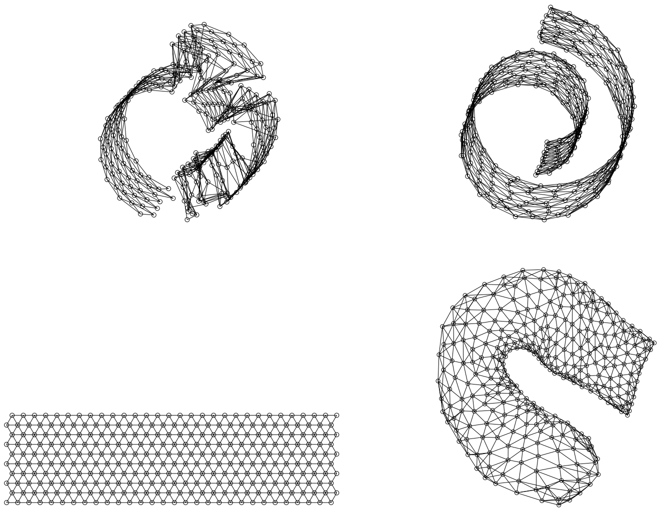

Fig. 2. Swiss roll unfolded by a SOM (left) and by Isotop (right), both shown in the 3D data space (top) and the 2D projection shown (bottom)

from this shortcoming since it builds its mapping in the projection space, whose dimensionality is ideally equal to the one of the linked structure (2D).

Another artificial example is the ‘open box’ shown in the right part of Fig. 1 (20000 samples). A 30 × 10 SOM converges easily. But this time, the problem comes from the rectangular shape of the grid that difficultly fits to the topology of the box. Indeed, Fig. 4 shows that some neighborhoods are not preserved: faces of the box are cut out into two parts that are not directly contiguous on the grid. With the help of its data-driven linking step, Isotop (300 prototypes) works with no difficulties and perfectly reproduces the neighborhoods.

Finally, Isotop and the SOM have been applied to a real database, namely the ‘Abalone’ set from the UCI machine learning repository [2]. This set gathers various attributes from 4177 abalone shells in order to determine their age. Among the nine given attributes, the sex and the age (given by the number of rings in the shell) are eliminated because they are respectively nominal and integer-valued. The seven kept attributes are real values related to the size and weight of the shells. Each attribute is normalized to have zero mean and unit variance. Next, the dimensionality is reduced from 7 to 2 by a SOM and Isotop. After some preliminary analysis, it appears that the data cloud is quite elongated. Therefore, the above-mentioned SOM algorithm give good results when used with 20 × 10 prototypes. In the same way, the stimulation distribution of Isotop is modified in order to have standard deviations equal to 4 and 1. Like the SOM, Isotop works with 200 prototypes. The results are shown in Fig. 4, where the gray level of each prototype is proportional to the mean age of the shells it

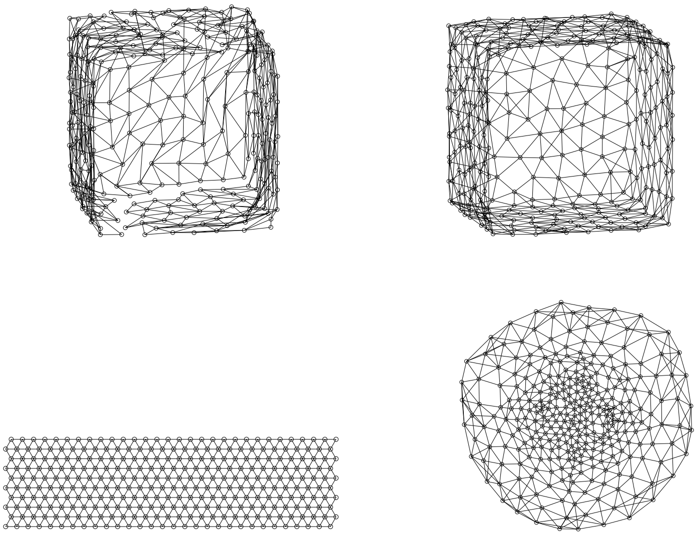

Fig. 3. Open box unfolded by a SOM (left) and by Isotop (right), both shown in the 3D data space (top) and the 2D projection shown (bottom)

represents. Both projection method converges well. At first sight, the regular grid of the SOM is visually pleasant, compared to the irregular cloud given by Isotop. However, a careful examination shows that Isotop preserves more information about the data than the SOM: the global shape of the data cloud is well reproduced and the outliers are still visible.

4 Conclusion

Isotop has been shown as an effective nonlinear projection method. Isotop combines advantages of different projection methods. Indeed, in the same way as many distance preserving algorithms do (e.g. Sammon's mapping), Isotop builds the mapping by working mainly in the low-dimensional projection space; by comparison, a SOM learns exclusively in the high-dimensional data space, what often leads to undesired twists and folds. Like the SOM, Isotop also uses neighborhood preservation instead of distance preservation, which is more constraining. Moreover, Isotop builds data-driven neighborhood structures, while a SOM suffers from its predetermined shape.

Perspectives for future work relates to the choice of the probability distribution for stimulating Isotop. Ideally, the stimulation pdf should be automatically chosen by the algorithm, according to the data distribution.

ICANN'2002 proceedings - International Conference on Artificial Neural Networks
Madrid (Spain), 28-30 August 2002, Springer, Lecture Notes in Computer Science 2415,
J.R. Dorronsoro ed., ISBN 3-540-44074-7, pp. 933-938

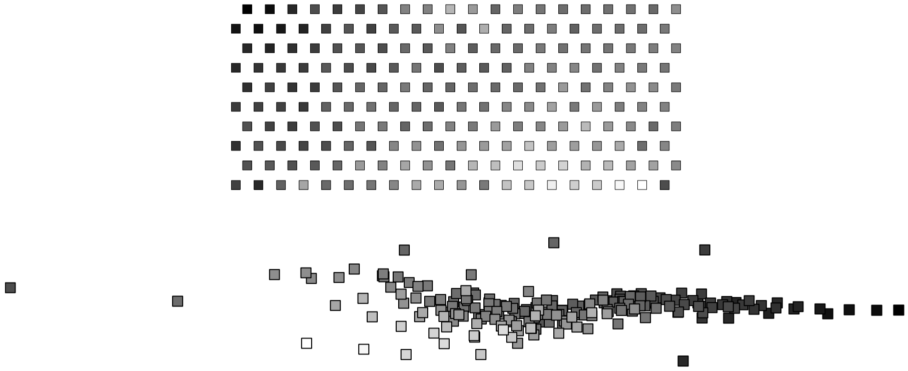

Fig. 4. Abalone dataset: nonlinear projection from 7 to 2 dimensions by a SOM (above) and by Isotop (bottom); the gray level of each prototype is proportional to the mean age of the shells it represents (black is young and white is old)

References

1. A. Ahalt, A. K. Krishnamurthy, P. Chen, and D. E. Melton. Competitive learning algorithms for vector quantization. *Neural Networks*, 3:277–290, 1990.
2. C. L. Blake and C. J. Merz. UCI repository of machine learning databases, 1998.
3. P. Demartines and J. Hérault. Vector Quantization and Projection Neural Network. In A. Prieto, J. Mira, and J. Cabestany, editors, *Lecture Notes in Computer Science*, volume 686, pages 328–333. Springer-Verlag, 1993.
4. P. Demartines and J. Hérault. Curvilinear Component Analysis: A self-organizing neural network for nonlinear mapping of data sets. *IEEE Transaction on Neural Networks*, 8(1):148–154, January 1997.
5. E. W. Dijkstra. A note on two problems in connection with graphs. *Numerical Mathematics*, (1):269–271, 1959.
6. I. T. Jollife. *Principal Component Analysis.* Springer-Verlag, New York, 1986.
7. T. Kohonen. Self-organization of topologically correct feature maps. *Biological Cybernetics*, 43:59–69, 1982.
8. T. Kohonen. *Self-Organizing Maps.* Springer, Heidelberg, 2nd edition, 1995.
9. J. A. Lee, A. Lendasse, N. Donckers, and M. Verleysen. A Robust Nonlinear Projection Method. In M. Verleysen, editor, *Proceedings of ESANN'2000, 8th European Symposium on Artificial Neural Networks*, pages 13–20. D-Facto public., Bruges (Belgium), April 2000.
10. H. Ritter, T. Martinetz, and K. Schulten. *Neural Computation and Self-Organizing Maps.* Addison-Wesley, 1992.
11. W. Sammon, J. A nonlinear mapping algorithm for data structure analysis. *IEEE Transactions on Computers*, CC-18(5):401–409, 1969.
12. R. N. Shepard. The analysis of proximities: Multidimensional Scaling with an unknown distance function. *Psychometrika*, 27:125–140, 1962.
13. J. B. Tenenbaum, V. de Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, 2000.
14. C. Von Der Malsburg. Self-organization of orientation sensitive cells in the striate cortex. *Kybernetik*, 14:85–100, 1973.