NOTE Communicated by Geoffrey Goodhill

Improved Multidimensional Scaling Analysis Using Neural Networks with Distance-Error Backpropagation

Lluís Garrido
Departament d'Estructura i Constituents de la Matèria/IFAE, Universitat de Barcelona, E-08028 Barcelona, Spain

Sergio Gómez
Departament d'Enginyeria Informàtica, Universitat Rovira i Virgili, E-43006 Tarragona, Spain

Jaume Roca
Departament d'Estructura i Constituents de la Matèria/IFAE, Universitat de Barcelona, E-08028 Barcelona, Spain

We show that neural networks, with a suitable error function for backpropagation, can be successfully used for metric multidimensional scaling (MDS) (i.e., dimensional reduction while trying to preserve the original distances between patterns) and are in fact able to outdo the standard algebraic approach to MDS, known as classical scaling.

1 Introduction

A standard problem in multidimensional scaling analysis is to map a collection of patterns, represented as points in an n-dimensional space

$$\{\mathbf{x}_a \in \mathbb{R}^n;\ a = 1, \ldots, p\},$$

to a lower-dimensional space in such a way that the distances between the projected points resemble as closely as possible the distances between the original ones.

More precisely, given the collection $\{\mathbf{x}_a\}$, with Euclidean distances between pairs (a, b) of patterns:

$$d_{ab}^{(n)} = \sqrt{(\mathbf{x}_a - \mathbf{x}_b)^2},$$

one has to find a map, $\varphi : \mathbb{R}^n \to \mathbb{R}^m$, with $m < n$, such that it minimizes the quadratic distance-error function

$$E_\varphi = \frac{1}{2} \sum_{a,b} \left(d_{ab}^{(n)} - d_{ab}^{(m)} \right)^2,$$

where $d_{ab}^{(m)}$ are the Euclidean distances computed in the projected space

$$d_{ab}^{(m)} = \sqrt{(\varphi(\mathbf{x}_a) - \varphi(\mathbf{x}_b))^2}\ .$$

Typically, m is chosen to be two or three in order to make available a graphical representation of the projected configuration. This can help visualize an underlying structure that might be obscured by cluttered data in the original space.

It is not known in general how to find the exact expression of the best map φ. Yet there is a standard method to approximate it, known as classical scaling (CLS), which involves the diagonalization of the symmetric matrix S of scalar products

$$S_{ab} = \mathbf{x}_a \cdot \mathbf{x}_b,$$

by means of an orthogonal matrix C. Taking $\{\mathbf{x}_a\}$ to be centered at the origin, that is, $\sum_a \mathbf{x}_a = 0$, and assuming that $p > n$, it is easy to show that S can have at most n nonzero eigenvalues. Each of these eigenvalues can be regarded as the typical scale of a principal direction. If we denote by $\Lambda_1, \ldots, \Lambda_m$ the m largest eigenvalues, the resultant mapping to $\mathbb{R}^m$ is given by

$$\varphi_{\text{CLS}}^{\alpha}(\mathbf{x}_a) = \Lambda_{\alpha}^{1/2}\, C_{a\alpha} \qquad \alpha = 1, \ldots, m.$$

(See Cox & Cox, 1994, for a detailed description of this method.)

CLS can be used in a broader context, when only a matrix of dissimilarities δ_{ab} is known, as a tool to assign coordinates to the patterns. Once coordinates are already known for patterns, as in our case, CLS reduces to principal component analysis (PCA).

2 Multidimensional Scaling with Neural Networks

In this article, we provide an alternative solution to this problem, which involves the use of neural networks. The main idea consists of building a net with n input units and a number of hidden layers, containing a bottleneck layer with only m units and an output layer with n units. A modified version of the standard backpropagation algorithm is then invoked (Rumelhart, Hinton, & Williams, 1986). In addition to the quadratic error term between input and output, it contains a new term that is introduced to minimize the difference between the distances of pairs in the input and neck layers. When enough iterations have been performed, the projected configuration is read out from the neck layer.

In order to use the net in the most efficient way, it is convenient to perform a translation and a global scaling of the initial data,

$$\mathbf{x}_a \longrightarrow \boldsymbol{\xi}_a^{\text{in}} = \lambda^{\text{in}}(\mathbf{x}_a - \mathbf{a}),$$

so as to make $\boldsymbol{\xi}_a^{\mathrm{in}} \in [0,1]^n$. Then one can use $\boldsymbol{\xi}_a^{\mathrm{in}}$ as the input to the net. The outcome of the neck layer, $\boldsymbol{\xi}_a^{\mathrm{nk}}$, lives in the region $[0,1]^m$ since we are using sigmoid activation functions. This implies that $0 \le d_{ab}^{\mathrm{nk}} \le \sqrt{m}$ while $0 \le d_{ab}^{\mathrm{in}} \le \sqrt{n}$ for any pair of input points $(\boldsymbol{\xi}_a^{\mathrm{in}}, \boldsymbol{\xi}_b^{\mathrm{in}})$, where d_{ab}^{nk} and d_{ab}^{in} stand for the distances between patterns a and b in the neck and initial layers, respectively.

The error function that we have considered in the backpropagation method is given by

$$E = \alpha\, E_1 + (1-\alpha)\, E_2,$$

where

$$E_1 = \sum_a \left(\boldsymbol{\xi}_a^{\mathrm{out}} - \boldsymbol{\xi}_a^{\mathrm{in}}\right)^2 \qquad \text{and} \qquad E_2 = \sum_{a,b} \left(\frac{d_{ab}^{\mathrm{in}}}{\sqrt{n}} - \frac{d_{ab}^{\mathrm{nk}}}{\sqrt{m}}\right)^2,$$

and $\alpha \in [0,1]$ controls the relative contribution of each part. The term E_1 favors those maps for which the representation in the bottleneck layer can be best accurately inverted to recover the original configuration. The second term, E_2, is the most important one since it forces this representation in the bottleneck to inherit, as closely as possible, the metric structure of the original configuration. The different scalings for d_{ab}^{in} and d_{ab}^{nk} in this term are introduced in order to have both numbers in the same range. In this way, we can guarantee that all possible configurations can still be covered with the use of sigmoids.

The various scalings involved in this process make the outcome of the neck layer not to be directly interpretable as the final answer; we can bring it back to the original scale by setting

$$\varphi_{\mathrm{NN}}(\mathbf{x}_a) = \lambda^{\mathrm{out}} \boldsymbol{\xi}_a^{\mathrm{nk}},$$

with $\lambda^{\mathrm{out}} = \sqrt{n/m}\,\lambda^{\mathrm{in}}$. A slightly better solution can be obtained by choosing instead

$$\lambda^{\mathrm{out}} = \frac{\sum_{a,b} d_{ab}^{(n)} d_{ab}^{\mathrm{nk}}}{\sum_{a,b} \left(d_{ab}^{\mathrm{nk}}\right)^2},$$

since this is the value of λ that minimizes the function $E(\lambda) = \frac{1}{2}\sum_{a,b}(d_{ab}^{(n)} - \lambda d_{ab}^{\mathrm{nk}})^2$ for the given neck configuration, which is what we are ultimately trying to achieve with the procedure.

In the practical use of the neural network, we have noticed that the best results are obtained by letting the parameter α fall to zero as the learning grows so that the error function E reduces to E_2 after a certain number of iterations. Actually, a nonzero value of α is useful only in the early stages of the learning, in order to speed up convergence. In this situation, with $E = E_2$, it is easy to prove analytically that the configuration minimizing E differs from the one minimizing directly $\sum(d^{\text{in}}–d^{\text{nk}})^2$ only by a global scaling $\sqrt{n/m}$ of all coordinates. Thus, the (otherwise technically convenient) scalings that we have introduced are completely harmless for the purpose of searching for the best mapped configuration.

It is commonly known that a network with just the input, output, and neck layers, with linear activation functions and subject to self-supervised backpropagation, is equivalent to PCA (Sanger, 1989). Our approach goes beyond PCA, not only because of the use of sigmoid (nonlinear) activation functions and the addition of a number of hidden layers, but essentially for the presence of this new distance-term contribution, E_2, which favors those configurations in the neck layer that approximate the original distances better.

One may wonder how our method compares to nonlinear PCA (NLPCA) (Kramer, 1991; DeMers & Cottrell, 1994; Kambhatla & Leen, 1995; Garrido, Gaitán, Serra-Ricart, & Calbet, 1995; Garrido, Gómez, Gaitán, & Serra-Ricart, 1996). Actually, NLPCA can be recovered as a particular case of our approach by setting $\alpha = 1$ in the error function (i.e., with $E = E_1$). NLPCA will generally do better than ordinary PCA in the minimization of the term E_1 because of the ability to model nonlinear configurations. However, NLPCA does not care at all about the distances between patterns in the bottleneck representation: any two neck configuration are equally good for NLPCA if both provide the same result in the output layer. Hence, the comparison of NLPCA with our approach is inappropriate because both methods are in fact designed for different purposes (minimizing E_1 and E_2, respectively). On the contrary, the projected configuration of standard PCA still retains part of the metric structure of the initial configuration since it is just a linear orthogonal projection onto the largest-variance axes, and hence it produces better results for E_2 than NLPCA. This is why we will compare the performance of our method with CLS (i.e., PCA) and not with NLPCA.

A comparative analysis of both approaches over several types of configurations shows that our method produces better results in the tougher situations, when some of the discarded directions in the CLS method still have relatively large associated eigenvalues. Finally, it is worth stressing that CLS provides only a linear orthogonal projection, whereas the neural net is able to produce more general (nonlinear) mappings.

Example. As an illustration of both procedures, we have considered a data set[1] consisting of different animal species, characterized by $n = 17$ attributes each (15 boolean + 2 numerical). The coordinates $\mathbf{x}_a$ and distances $d_{ab}^{(n)}$ have been obtained after scaling the numerical attributes to the range

[1] Extracted from the Zoo Database created by Richard S. Forsyth (1990) (ftp://ftp.ics.uci.edu:~/pub/machine-learning-databases/zoo).

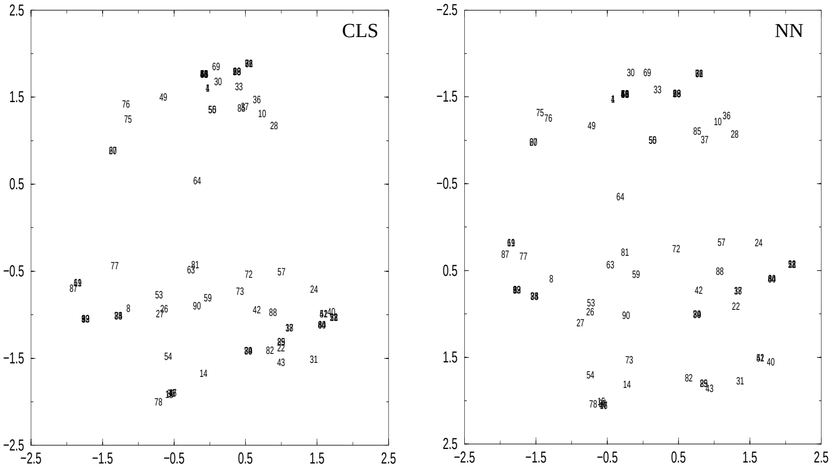

Figure 1: Two-dimensional mapped configurations obtained with classical scaling (CLS) and with a neural network (NN).

[0, 1] in order to assign an equal weight to all attributes (implying in this case that we simply have $\boldsymbol{\xi}_a^{\text{in}} = \mathbf{x}_a$).

The best scaling for the two-dimensional neck representation when using the neural net is given by $\lambda^{\text{out}} = 2.946$, which is in less than 1.1% disagreement with the expected value of $\lambda^{\text{out}} = \sqrt{17/2}$.

The projected configurations obtained with each method are drawn in Figure 1. Patterns are represented by their label. As the plot shows, both approaches produce a fairly similar configuration. However, the computation of the overall relative error,

$$\mathcal{E} = \left(\frac{\sum_{a,b} \left(d_{ab}^{(n)} - d_{ab}^{(m)} \right)^2}{\sum_{a,b} \left(d_{ab}^{(n)} \right)^2} \right)^{\frac{1}{2}},$$

shows for each method that the neural network gives a slightly better result,

$$\mathcal{E}_{\text{CLS}} = 0.2728, \qquad \mathcal{E}_{\text{NN}} = 0.2346,$$

which amounts to a 14.00% improvement over the CLS method.

Acknowledgments

This work was supported in part by CICYT contract AEN95-0590 and a URV project, URV96-GNI-13. J.R. also thanks the Ministerio de Educación y Cultura of Spain for financial support.

References

Cox, T. F., & Cox, M. A. A. (1994). *Multidimensional scaling.* London: Chapman & Hall.

DeMers, D., & Cottrell, G. (1994). Non-linear dimensionality reduction. In C. L. Giles, S. J. Hanson, & J. D. Cowan (Eds.), *Advances in neural information processing systems, 5*. San Mateo, CA: Morgan Kauffman.

Forsyth, R. S. (1990). Zoo Database. Available online at ftp: //ftp.ics.uci.edu/pub/machine-learning-databases/200.

Garrido, Ll., Gaitán, V., Serra-Ricart, M., & Calbet, X. (1995). Use of multilayer feedforward neural nets as a display method for multidimensional distributions. *Int. J. Neural Systems, 6*, 273.

Garrido, Ll., Gómez, S., Gaitán, V., & Serra-Ricart, M. (1996). A regularization term to avoid the saturation of the sigmoids in multilayer neural networks. *Int. J. Neural Systems, 7*, 257.

Kambhatla, N., & Leen, T. K. (1995). Fast non-linear dimension reduction. In J. D. Cowan, G. Tesauro, & J. Alspector (Eds.), *Advances in neural information processing systems, 6*. San Mateo, CA: Morgan Kauffman.

Kramer, M. A. (1991). Non-linear principal component analysis using autoassociative neural networks. *AICHE Journal, 37*, 233.

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by backpropagating errors. *Nature, 323*, 533.

Sanger, T. D. (1989). Optimal unsupervised learning in a single-layer linear feedforward neural network. *Neural Networks, 2*, 459.

Received October 29, 1997; accepted July 10, 1998.