Nonmetric Multidimensional Scaling: Neural Networks versus Traditional Techniques

[a]M.C. van Wezel [b]W.A. Kosters

[a] Faculty of Economical Sciences, Department of Computer Science, Erasmus University
P.O. Box 1738, 3000 DR, Rotterdam, The Netherlands.
E-mail: mvanwezel@few.eur.nl

[b] LIACS, Leiden University
P.O. Box 9512, 2300 RA Leiden, The Netherlands.
E-mail: kosters@liacs.nl

Abstract

In this paper we consider various methods for nonmetric multidimensional scaling. We focus on the nonmetric phase, for which we consider various alternatives: Kruskal's nonmetric phase, Guttman's nonmetric phase, monotone regression by monotone splines, and monotone regression by a monotone neural network. We use sequential quadratic programming to estimate the weights of the neural network. All methods are briefly described. An experimental comparison of the methods is given for various synthetic and real-life datasets.

1 Introduction

This paper concerns visualization of multidimensional data using nonmetric multidimensional scaling. Generally stated, multidimensional scaling (abbreviated MDS) is a collection of techniques for embedding dissimilarity data, given in the form of a dissimilarity matrix, in a space with a chosen dimension. The embedding is often used for the purpose of data visualization and exploratory data analysis. In this sense, MDS is a competitor for other data visualization techniques, such as a Kohonen SOM and principal component analysis. Traditional MDS techniques are subdivided into metric MDS, where the dissimilarities between objects are assumed to be proportional to Euclidean distances, and nonmetric MDS, where the dissimilarities are only assumed to be related to Euclidean distances by some unknown monotone transformation. Nonmetric MDS nevertheless attempts to find a good embedding in Euclidean space by finding the inverse of the transformation. We describe the MDS problem in Section 2. In the case where the dissimilarities represent, e.g., distances between the capitals of the European countries, the Euclidean distance assumption of metric MDS is realistic. However, in the case of dissimilarities between soda brands that have been reported by a panel of test persons, nonmetric MDS seems more appropriate[1].

[1]Merely the data collection method employed in such situations may lead to distortions.

Although traditional MDS is a well established pattern recognition technique (see, e.g., [1, 2]), to our knowledge this subject has not received a lot of attention from researchers in the field of neural networks (or, for that matter, artificial intelligence in general). In particular, all publications concerning neural networks for MDS that are known to us concern metric MDS. In [9] a simple neural network is given for metric MDS. This neural network merely performs a gradient descent on the cost function, which carries the risk of getting stuck in local minima in the error function. To prevent this, in [5] Klöck and Buhmann apply annealing methods from statistical mechanics to the metric MDS problem.

No neural algorithms have been applied to nonmetric MDS so far, besides in [10], where a multilayer perceptron with a special architecture was used to perform the monotone transformation that forms an essential part of all nonmetric MDS algorithms. The aim of the current paper is to extend the method proposed in [10], and to place it in a context by a comparison with other techniques. In particular, in this paper we use the minimization technique sequential quadratic programming for the estimation of the network weights and compare the performance with monotone splines and Kruskal's and Guttman's methods.

This paper is organized as follows. First, an exact problem statement is presented in Section 2. Next, the various methods for the nonmetric phase of MDS are described in Section 3. After that, the results of some experiments are given in Section 4 and finally Section 5 gives conclusions. This paper does not give an in-depth treatment of MDS in general. The reader is referred to [1, 2] for more information on the subject.

2 Multidimensional scaling: problem statement

Before we are able to describe the analysis problem in metric and nonmetric MDS we introduce some terminology and notation:

Dissimilarities: In MDS analyses, the starting point is a matrix $\boldsymbol{\delta}$ of dissimilarities, of which an element δ_{ij} denotes the dissimilarity between two objects i and j. The number of objects is denoted by n.

Embedding: An embedding of the objects in Euclidean space. The coordinates of an object i in this embedding are denoted by $\mathbf{x}_i$. The dimension of the embedding space is denoted by m, so $\mathbf{x}_i = (x_{i1}, \ldots, x_{im})^T$.

Distances: The (real) Euclidean distance between objects i and j is denoted by $\hat{d}_{ij}$. So, $\hat{d}_{ij}$ denotes the distances in the 'true' embedding in m-space. It is this embedding that we attempt to reconstruct. The distance between the estimates for the spatial representations $\mathbf{x}_i$ and $\mathbf{x}_j$ of objects i and j is denoted by $d_{ij} = \| \mathbf{x}_i - \mathbf{x}_j \|$, where $\| \, . \, \|$ denotes the Euclidean norm. Collectively the distances are denoted by the matrix $\mathbf{d}$.

Disparities: These quantities are used in nonmetric scaling. Disparities $\hat{\boldsymbol{\delta}}$ are as close as possible to distances between the corresponding coordinate estimates $\mathbf{d}$ but with the restriction that they are monotonically related to the original dissimilarity data $\boldsymbol{\delta}$.

Using the above concepts, the MDS problem can be accurately described. We differentiate between metric and nonmetric MDS. In *metric* MDS, it is assumed that the dissimilarities are *proportional* to Euclidean distances: $\delta_{ij} = c\hat{d}_{ij}$. One way to obtain a spatial representation of dissimilarity data under the above assumption is to minimize the following error function

$$E_{\mathrm{metric\_mds}} = \sum_{ij, i \neq j} (\delta_{ij} - d_{ij})^2. \tag{1}$$

This minimization gives us the correct coordinates up to a scale factor, a translation and a rotation [2].

In *nonmetric MDS* the 'distance assumption' $\delta_{ij} = c\hat{d}_{ij}$ is relaxed to a 'monotonicity assumption'. It is assumed that the dissimilarities $\boldsymbol{\delta}$ are *monotonically related* to Euclidean distances:

$$\forall i, j, k, \ell : \hat{d}_{ij} < \hat{d}_{k\ell} \Rightarrow \delta_{ij} < \delta_{k\ell} \ . \tag{2}$$

One can look upon the $\boldsymbol{\delta}$ values as being monotonically transformed distance values: $\delta_{ij} = f(\hat{d}_{ij})$ where $f(.)$ is an unknown strict monotonically increasing function. Examples of such functions include some linear, power and logarithmic functions. Nonmetric MDS algorithms estimate a spatial representation for a given dissimilarity matrix in which the rank order of the distances between the embedded objects agrees with the rank order of the dissimilarities as much as possible.

Traditionally, in nonmetric MDS one attempts to minimize the following cost function, often called stress (due to Kruskal [6]), in an iterative fashion:

$$E_{\mathrm{nonmetric\_mds}} = \sqrt{\sum_{ij, i \neq j} (\hat{\delta}_{ij} - d_{ij})^2 / \sum_{ij, i \neq j} d_{ij}^2} \tag{3}$$

Often, the inter-pattern distances are kept normalized ($\sum_{ij, i \neq j} d_{ij}^2 = 1$), in which case the error function reduces to (ignoring the square root)

$$E^{nm} = \sum_{ij, i \neq j} (\hat{\delta}_{ij} - d_{ij})^2, \tag{4}$$

which is computationally simpler. The disparities $\hat{\boldsymbol{\delta}}$ must be re-estimated in each iteration in a so-called nonmetric phase. This nonmetric phase is alternated with a *metric phase* in which a metric MDS problem is solved where the current disparities $\hat{\boldsymbol{\delta}}$ play the role of dissimilarities. The procedure is depicted in Figure 1.

In this paper we do not pay attention to the metric phase of nonmetric MDS — we focus on various possibilities for the nonmetric phase. In the nonmetric phase, the disparities are chosen to resemble the distances $\mathbf{d}$ of the current embedding as close as possible subject to a monotonicity constraint: $\forall i, j, k, \ell : \delta_{ij} < \delta_{k\ell} \Rightarrow \hat{\delta}_{ij} < \hat{\delta}_{k\ell}$. This is done by performing a monotone regression with the current distances $\mathbf{d}$ as targets and dissimilarities $\boldsymbol{\delta}$ as inputs.

By minimizing stress (Equation 3), nonmetric MDS finds a spatial representation in which the dissimilarities are transformed by the monotone transformation in a way that inverts the monotone transformation that distorted the true distances between the objects in m-space, and thus improves the final fit to the data.

[2] Solutions are prone to be local minima. Much research has gone into avoiding this.

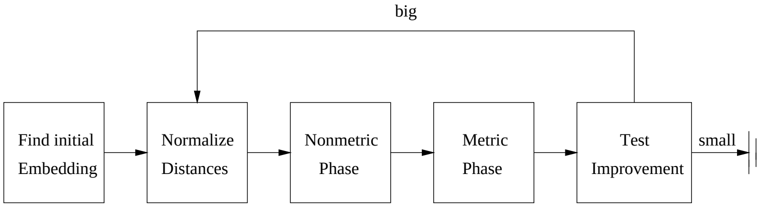

Figure 1: Schematic representation of nonmetric mds algorithm

3 Approaches to the nonmetric phase

Various methods have been proposed to perform the monotone regression in the nonmetric phase. We consider four methods here: Kruskal's nonmetric phase, Guttman's nonmetric phase, monotone regression by monotone splines, and monotone regression by a monotone neural network. Due to space limitations, the presentation here is very brief — for a more elaborate presentation the reader is referred to a forthcoming technical report at `http://www.erim.nl`, or to the references cited in this section. The first two methods manipulate the disparities directly in order to minimize Equation 4, the latter two methods estimate parameters for a monotone regression model which is then used to compute the disparity values for the dissimilarities.

The reason for this explicit modeling using a monotone regression model is threefold. First, explicit modeling gives the user the opportunity to visualize and examine the total monotone transformation, because the regression models perform interpolation. Second, the use of interpolation yields a smooth, continuous mapping from dissimilarities to disparities, which is intuitively more plausible than a step function, which results from Kruskal's and Guttman's nonmetric phases. Third, the use of interpolation in the nonmetric phase can potentially speed up the nonmetric phase considerably in the case where the dissimilarity matrix is of substantial size. The reason for this is that traditional nonmetric phases (like Kruskal's and Guttman's) have a high worst case time complexity. E.g., for Guttman's nonmetric phase it is $\mathcal{O}(n^4)$, whereas for Kruskal's method it is $\mathcal{O}(n^2 \log n^2)$. The time complexity for the monotone regression models is much more favorable. E.g., in the case of the monotone regression splines it is $\mathcal{O}(n^2)$. This leads to a potential benefit for large dissimilarity matrices. Furthermore, the explicit regression models can be 'trained' using only a subset of all available data, and can be used as an interpolator for the remaining data, leading to a further performance gain, but this is not exploited in this paper.

Kruskal's and Guttman's nonmetric phases. In Kruskal's nonmetric phase, disparities are grouped in blocks in which all disparities have equal values. Initially, one block is associated with each disparity in $\hat{\boldsymbol{\delta}}$. The disparity blocks are then compared pairwise in rank order of the associated dissimilarity (i.e., the disparity associated with the i-th largest dissimilarity is the i-th disparity considered), and when a violation of the monotonicity assumption is found, the violating blocks are unified by setting all disparities they contain to the mean of disparities in the two blocks. This is iterated until no violations occur.

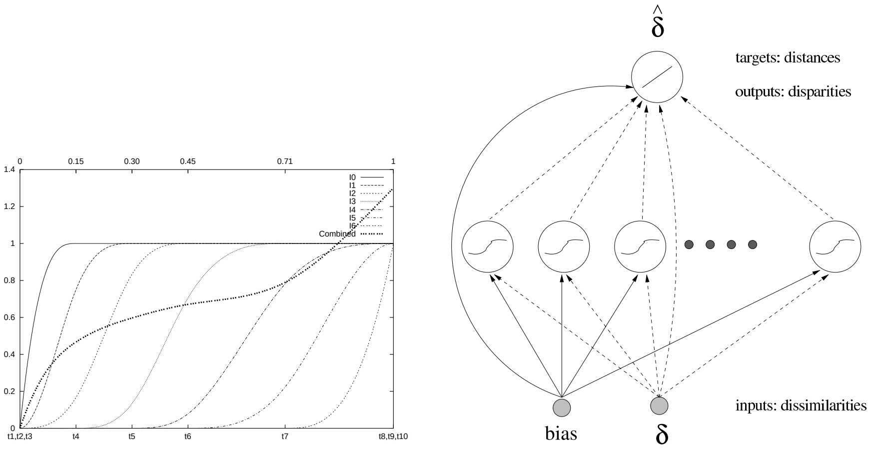

Figure 2: Left: Example of an I-spline basis of order 3 with interior knots situated at $0.15, 0.3, 0.45$, and 0.71, and the linear combination $0.3I_0 + 0.2I_1 + 0.1I_2 + 0.1I_3 + 0.4I_4 + 0.2I_5$. Right: Monotone neural network for nonmetric phase in MDS.

In Guttman's nonmetric phase, the disparity associated with the i-th largest dissimilarity is set to the i-th largest element of $\mathbf{d}$ (i.e., the i-th largest distance estimate). Both methods yield a step function as the result and suffer from a high worst case time complexity. In practice the time complexity is hardly a disadvantage because of the small number of objects that is usually embedded.

Nonmetric phase using monotone regression splines. As an alternative to manipulating the disparities directly, one can also construct an explicit model for the relation between dissimilarities and disparities. The first of these models we will consider is monotone regression based on monotone splines, introduced by Jim Ramsay in [8], where the application to nonmetric MDS is also mentioned.

Briefly stated, a monotonically increasing nonlinear function can be approximated by forming a linear combination of a number of *monotone splines* or I-splines, collectively referred to as the spline basis, where the coefficients of the combination are constrained to be positive. An example of a basis of I-splines and a linear combination is given in Figure 2. The number and form of the I-splines in a basis can be controlled by means of some parameters. In our case, estimation of the optimal coefficients boils down to a quadratic programming problem, for which we used the QP solver `BPMPD` by Csaba Meszaros [7].

Nonmetric phase using a monotone neural network. A fourth possibility for the nonmetric phase, is the use of a a multilayer perceptron neural network that is only capable of modeling monotone transformations. We will refer to this network type as a *mono-nn* in the remainder. It takes the dissimilarities $\boldsymbol{\delta}$ as inputs and generates the disparities $\hat{\boldsymbol{\delta}}$ as outputs. It uses the distances $\mathbf{d}$ as targets, and has one hidden layer with non-linear (hyperbolic tangent) transfer functions. The output unit uses the identity as a transfer function. Since the individual transfer functions are monotonically increasing, the monotonicity constraint is always satisfied if we impose a positivity constraint on all weights of the neural

name	description	# objects	reference
`cars`	Pairwise dissimilarity judgments on 11 car models.	11	SPSS
`square`	Nonlinearly transformed distances between points arranged in a square.	16	[10]
`riasec`	Dissimilarities between six vocational preference inventory scales.	6	[2]
`citations`	Transformed inter-journal citation numbers in psychometric literature.	28	[4]

Table 1: Description of used datasets.

network except the biases. A network of this kind is depicted in Figure 2.

We implemented two approaches to enforce the positivity constraint on the weights. Details on these approaches can be found in the forthcoming technical report mentioned earlier. In the first approach, we *squared* all the weight values prior to their use. So, the input to a unit b in the hidden layer would be $in_{b|ij} = w_b^2\delta_{ij} + \theta_b$, where w_b and θ_b denote unit b's weight and bias. The learning rule has to be altered to incorporate the squared weights. Details can be found in [10] and in the forthcoming technical report.

In the second approach, we used the minimization technique *sequential quadratic programming* (SQP) for estimation of the weights. SQP is a technique for minimization of a general (possibly nonlinear) function under general (possibly nonlinear) equality- and inequality-constraints [3]. It obtains a solution by replacing the original objective function by a succession of quadratic approximations, so that each iteration involves solving a quadratic programming problem. We used the SQP code `CFSQP` in our experiments. `CFSQP` is available for free to the academic research community (see `http://www.aemdesign.com`).

The estimation of the parameters for the mono-nn is accelerated by using a good initial embedding. In the experiments reported in the next section, we used the solution obtained by metric MDS as the initial embedding.

4 Experiments and results

To assess the quality of the final embedding obtained with all nonmetric scaling methods, we used the datasets mentioned in Table 1. Table 2 shows the results of the experiments with these datasets. All results are averaged over 50 independent runs, to compensate for the variance in the final stress-values due to different initial configurations.

One can see that all nonmetric MDS methods clearly improve the data fit compared to the metric MDS method. In general, the 'non-interpolating' nonmetric MDS methods (Kruskal and Guttman) give a better fit than the interpolating MDS methods. The remaining nonmetric MDS methods, the monotone neural network and the monotone spline, are close to one-another with respect to the normalised stress value.

The monotone transformations yielded for the `citations` dataset by the various scaling methods are depicted in Figure 3. It is clearly visible that the non-interpolating methods yield a non-smooth transformation, which is not very plau-

[3] Our constraints are linear while our objective function is nonlinear. Algorithms for this special case also exist (see [3]) but an implementation of such a method was not available to us.

	avg	stdv	min	max	avg	stdv	min	max
	cars				**square**			
Metric	0.009911	0.000213	0.009845	0.010640	0.017536	0.013033	0.010766	0.043688
Kruskal	0.001215	0.000000	0.001214	0.001215	0.001463	0.007313	0.000000	0.037328
Guttman	0.002609	0.000004	0.002604	0.002624	0.000000	0.000000	0.000000	0.000000
Spline	0.003606	0.000001	0.003606	0.003607	0.023922	0.068711	0.010766	0.500000
Neural	0.007260	0.000201	0.007094	0.008635	0.012375	0.014460	0.002173	0.035308
SQP neural	0.006416	0.001096	0.002187	0.006780	0.006325	0.010753	0.000070	0.035434
	riasec				**citations**			
Metric	0.023089	0.006914	0.016013	0.036849	0.661513	0.003918	0.658505	0.687850
Kruskal	0.004635	0.006764	0.000000	0.017992	0.057253	0.000293	0.057174	0.059301
Guttman	0.000000	0.000000	0.000000	0.000000	0.072910	0.000031	0.072838	0.072994
Spline	0.170304	0.225297	0.000472	0.500000	0.056720	0.000013	0.056691	0.056744
Neural	0.010163	0.009910	0.000150	0.023258	0.059584	0.001811	0.056677	0.064066
SQP Neural	0.014982	0.010172	0.000072	0.035448	0.057731	0.002347	0.053119	0.064688

Table 2: Normalised stress (E^{nm}) values obtained for all four datasets with metric and various types of nonmetric scaling, averaged over 50 runs.

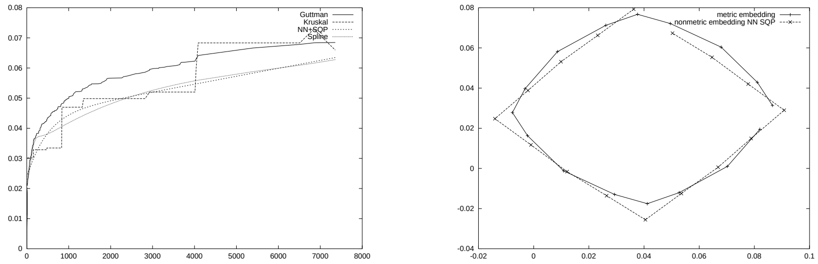

Figure 3: Transformations yielded by the various nonmetric phases for the `citations` dataset (left), and embeddings yielded for the `square` dataset (right) by metric and nonmetric scaling.

sible. Also shown are the embeddings yielded by metric scaling and by nonmetric scaling using a neural network (trained with SQP) in the nonmetric phase for the `square` dataset. It is clear that nonmetric scaling improves the embedding yielded by metric scaling.

To examine the computational demands of the various methods we created artificial dissimilarity matrices of various sizes by randomly generating a number of points n in a 2-dimensional space, calculating the distance matrix and applying a nonlinear monotone transformation ($f(x) = \log(x + 0.01) + 5$) to these distances. Table 3 shows the wall clock times in seconds needed for the nonmetric embedding of these artificial datasets for various n. For large n, the use of a spline method in the nonmetric phase is relatively fast, due to the inherit simplicity of this method. It is clear that the monotone neural network, when trained with the modified back-propagation procedure of [10] is unacceptably slow, but when the SQP method is used, the required computation times are acceptable, whereas the results reported in Table 2 indicate that the obtained stress values are usually lower.

5 Summary and conclusions

We have applied neural networks in the nonmetric phase in nonmetric multidimensional scaling. The weights of the neural network were estimated by an altered

	Number of objects n					
	10	50	100	300	500	t_{500}/t_{10}
Metric	0.01 (.019)	0.21 (.027)	2.03 (.027)	9.99 (.024)	34.01 (.020)	3401
Kruskal	0.28 (.000)	8.81 (.000)	28.83 (.000)	113.87 (.000)	193.55 (.000)	689
Guttman	0.11 (.000)	0.39 (.011)	21.00 (.000)	66.79 (.000)	209.10 (.000)	1900
Spline	2.33 (.000	1.54 (.000)	3.86 (.000)	35.20 (.000)	84.11 (.000)	36
Neural	19.36 (.000)	79.64 (.003)	502.38 (.001)	5051.14 (.001)	89566.09 (.001)	4626
SQP Neural	2.95 (.000)	31.73 (.000)	75.90 (.003)	644.76 (.004)	615.80 (.006)	208

Table 3: Wall clock computation times required by the various nonmetric MDS methods. Shown between brackets are the resulting stress values. The final column shows the quotient of the required time with 500 (t_{500}) and 10 (t_{10}) objects.

training procedure and by sequential quadratic programming. The approach was compared with other approaches (Kruskal's and Guttman's methods and monotone splines) in a series of experiments. The neural network based methods perform comparable to the existing methods, but have the advantage of yielding smooth mappings instead of step functions, which is more plausible and makes interpolation easier.

References

[1] I. Borg and P. Groenen. *Modern multidimensional scaling. Theory and applications.* Springer-Verlag, New York, 1997.

[2] M. L. Davison. *Multidimensional Scaling.* Wiley, New York, 1983.

[3] R. Fletcher. *Practical methods of optimization.* Wiley, 2nd edition, 2000.

[4] P.F.J. Groenen and W.J. Heiser. The tunneling method for global optimization in multidimensional scaling. *Psychometrika*, 61(3):529–550, 1996.

[5] H. Klöck and J.M. Buhmann. Data visualization by multidimensional scaling: A deterministic annealing approach. *Pattern Recognition*, 33:651–669, 1999.

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

[7] Cs. Mészáros. The BPMPD interior point solver for convex quadratic problems. Technical Report WP 98-8, Hungarian Academy of Sciences, Budapest, 1998.

[8] J.O. Ramsay. Monotone regression splines in action. *Statistical science*, 3(4):425–441, 1988.

[9] M. C. van Wezel, J. N. Kok, and W. A. Kosters. Two neural network methods for multidimensional scaling. In *European Symposium on Artificial Neural Networks (ESANN'97)*, pages 97–102, Brussels, 1997. D facto.

[10] M.C. van Wezel, W.A. Kosters, P. van der Putten, and J.N. Kok. Nonmetric multidimensional scaling with neural networks. *Lecture Notes in Computer Science*, 2189:145–156, 2001.