

# Nonmetric multidimensional scaling: Neural networks versus traditional techniques

M.C. van Wezel<sup>a,\*</sup> and W.A. Kusters<sup>b</sup>

<sup>a</sup>*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*

<sup>b</sup>*Leiden Institute of Advanced Computer Science (LIACS), Leiden University, P.O. Box 9512, 2300 RA Leiden, The Netherlands*

*E-mail: kusters@liacs.nl*

Received 15 July 2003

Revised 14 September 2003

Accepted 23 November 2003

**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. All methods are briefly described. We use sequential quadratic programming to estimate the weights of the neural network. An experimental comparison of the methods is given for various synthetic and real-life datasets. The monotone neural network performs comparable to the traditional methods.

Keywords: Data visualization, MDS, multidimensional scaling, neural networks

## 1. Introduction

This paper is concerned with 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 dimensionality. The embedding is often used for the purpose of data visualization and exploratory data 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.

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

---

\*Corresponding author. Tel.: +31 10 4081341; Fax: +31 10 4089167.

seems more appropriate. In fact, the cars dataset mentioned in Section 4 is an example of a nonmetric dataset because it consists of a mere ranking of similarities.

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 [14] 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 [8] 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 [15], 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 [15], 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.

Because MDS is a data visualization technique, it is a competitor for projection techniques such as a Kohonen SOM and principal component analysis. However, although MDS can be used for data projection, it can also be used for the embedding of dissimilarity data – something that can't be done with projection techniques. Dissimilarity-based data analysis has applications in a number of domains, such as marketing and image analysis, and thus it deserves attention from the AI community. We hope that this paper helps in raising some interest in the topic.

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

Roughly stated, multidimensional scaling attempts to find an embedding in a metric space and a suitable dimension for this space. Before we are able to describe the analysis problem in metric and nonmetric MDS accurately, we introduce some terminology and notation:

**Dissimilarities:** In MDS analyses, the starting point is a matrix  $\delta$  of dissimilarities, of which an element  $\delta_{ij}$  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  $\vec{x}_i$ . The dimensionality of the embedding space is denoted by  $m$ , so  $\vec{x}_i = (x_{i1}, \dots, x_{im})^T$ .

**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  $\vec{x}_i$  and  $\vec{x}_j$  of objects  $i$  and  $j$  is denoted by  $d_{ij} = \|\vec{x}_i - \vec{x}_j\|$ , where  $\|\cdot\|$  denotes the Euclidean norm. Collectively these distances are denoted by the matrix  $\mathbf{d}$ .

**Disparities:** These quantities are used in nonmetric scaling. Disparities  $\hat{\delta}$  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  $\delta$ .<sup>1</sup>

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

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

This minimization gives us the correct coordinates up to a scale factor, a translation and a rotation.<sup>2</sup>

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

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

One can look upon the  $\delta$  values as being monotonically transformed distance values:  $\delta_{ij} = f(\hat{d}_{ij})$  where  $f(\cdot)$  is an unknown strict<sup>3</sup> 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-1 (due to Kruskal [9]), in an iterative fashion:

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

By keeping the inter-pattern distances normalized ( $\sum_{ij, i \neq j} d_{ij}^2 = 1$ ), as suggested in [2], the error function reduces to (ignoring the square root)

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

which is computationally simpler. In this paper we use  $E^{nm}$  with normalized distances as the error function and refer to it as ‘normalized stress’.

The disparities  $\hat{\delta}$  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  $\hat{\delta}$  play the role of dissimilarities. In this phase, the embedding coordinates are altered as to minimize Eq. (4). We used the conjugate gradient minimization algorithm from the Numerical Recipes library ([12]) for this phase. The whole procedure is schematically depicted in Fig. 1.

<sup>1</sup>In the literature the disparities are often denoted by  $\hat{d}_{ij}$ , but we use  $\hat{\delta}_{ij}$ , following [2].

<sup>2</sup>Solutions are prone to be local minima. Much research has gone into avoiding this.

<sup>3</sup>In the literature this strict monotonicity restriction is often relaxed to a monotonicity restriction, where  $f(\cdot)$  is monotone and  $\forall i, j, k, \ell : \hat{d}_{ij} < \hat{d}_{k\ell} \Rightarrow \delta_{ij} \leq \delta_{k\ell}$ . Existence of an inverse for  $f(\cdot)$  is not guaranteed in this case.

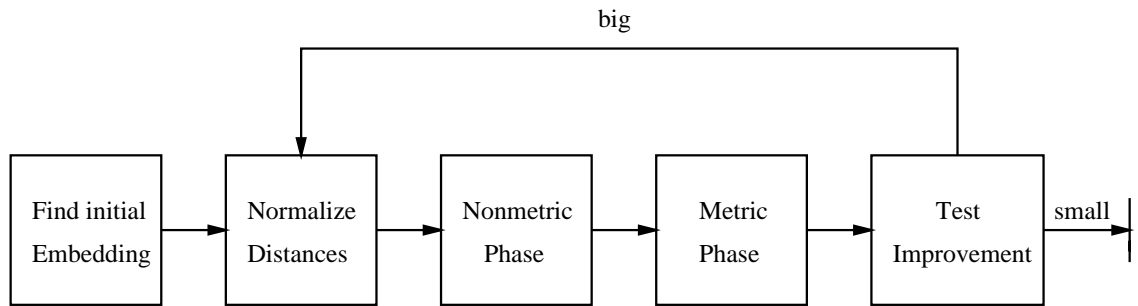


Fig. 1. Schematic representation of a nonmetric MDS algorithm.

The most suitable embedding dimensionality in MDS is usually found by considering ‘fit versus number of dimensions’. We do not consider this problem in this paper, nor do we pay attention to the metric phase of nonmetric MDS. Instead, 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:

$$\forall i, j, k, \ell : \delta_{ij} < \delta_{k\ell} \Rightarrow \hat{\delta}_{ij} \leq \hat{\delta}_{k\ell}. \quad (5)$$

This is done by performing a monotone regression (also called isotonic regression) with the current distances  $\mathbf{d}$  as targets and dissimilarities  $\delta$  as inputs. Note that Eq. (5) is a non-strict monotonicity condition. This is standard practice in MDS and, although it is inconsistent with condition Eq. (2), we follow it.

By minimizing Stress-1 (Eq. (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.

### 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. The first two methods manipulate the disparities directly in order to minimize Eq. (4), the latter two methods estimate parameters for a monotone regression model which is then used to compute the disparity values for the dissimilarities. For clarity we note that in the MDS literature the term monotone regression usually refers to Kruskal’s method.

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 in the case where the dissimilarity matrix is of substantial size. The reason for this is that traditional nonmetric phases (like Guttman’s) have a high worst case time complexity. E.g., for Guttman’s nonmetric phase it is  $\mathcal{O}(n^4(\log n^2)^2)$ .

The time complexity for the monotone regression models is likely to be linear in the number of data points  $n^2$  whenever the shape of the monotone transformation being fitted is simple. This leads to a potential benefit for large dissimilarity matrices, which is demonstrated in our experiments in Section 4. 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.

### 3.1. Kruskal’s and Guttman’s nonmetric phases

If Kruskal’s nonmetric phase [9,10] is employed, the first step is the computation of the rank order of the original dissimilarity data. Let  $r(i)$  denote the disparity associated with the  $i$ -th smallest dissimilarity (i.e., occupying the same position in the matrix). E.g., when  $\delta_{34}$  happens to be the 4-th smallest dissimilarity,  $r(4) = \hat{\delta}_{34}$ . In a given iteration of the nonmetric scaling algorithm the computations proceed as follows.

First, the disparities are set equal to the distances between objects in the current embedding (i.e.,  $\hat{\mathbf{d}} = \mathbf{d}$ ). Then, a monotonicity check follows: for each  $i = 1, \dots, n - 1$ , disparities  $r(i)$  and  $r(i + 1)$  associated with the  $i$ -th and  $i + 1$ -th smallest dissimilarities are compared. If this disparity pair does not violate the monotonicity constraint, so  $r(i) \leq r(i + 1)$ , nothing is done. If, however, this disparity pair violates the monotonicity constraint, two blocks of disparities with equal values to  $r(i)$  and  $r(i + 1)$  are identified. The first block is  $(a, i)$  with  $a \leq i$ ,  $(r(a - 1) \neq r(a) \vee a = 1)$ ,  $r(a) = r(i)$ , the second block is  $(i + 1, b)$  with  $i + 1 \leq b$ ,  $(r(b + 1) \neq r(b) \vee b = n)$ ,  $r(b) = r(i + 1)$ . So, these are blocks of equal disparities to  $r(i)$  and  $r(i + 1)$  of which the associated dissimilarities are adjoining in rank order. Subsequently, the two blocks are unified, i.e., all disparities in block  $(a, b)$  are set to the average value of the disparities in  $(a, b)$ . The pairwise monotonicity check of the disparities is continued until all disparities have been considered. If any unifications took place, another iteration follows.

Kruskal’s approach to the nonmetric phase has been proven to be least-squares optimal by de Leeuw in [3]. Unfortunately, the mapping that results often resembles a step function. Since it is reasonable to assume that the mapping we are after has at least second order continuity this can be considered to be a demerit.

It should be noted that the above version of Kruskal’s algorithm, which was taken from Davison [2], is somewhat different from the version described by Borg and Groenen [1] – the latter authors describe a version of the algorithm where a new block value is immediately compared with the adjoining block value of lower rank order, instead of waiting until the next iteration.

If Guttman’s nonmetric phase (also called Guttman’s rank image procedure [7]) is employed, the first step is the computation of the rank order of the original dissimilarity data. Let  $r(i)$  be as before. In a given iteration of the nonmetric scaling algorithm the computation proceeds as follows. First, the rank order of the current distance estimates  $\mathbf{d}$  is computed. Let  $r'(i)$  denote the value of the  $i$ -th smallest distance estimate. Then, for  $i = 1, \dots, n$ ,  $r(i) = r'(i)$ . So, the disparity associated with the  $i$ -th smallest dissimilarity is set to the  $i$ -th smallest distance estimate.

The same problems as with Kruskal’s method apply to Guttman’s method. As for the time complexity: the computation of the ranking of the distance estimates requires sorting of these estimates. The fastest available sorting algorithm, quicksort, has a  $\mathcal{O}(n \log n)$  worst case time complexity. In practice, however, Guttman’s method performs well, even though convergence of the overall algorithm cannot be guaranteed.

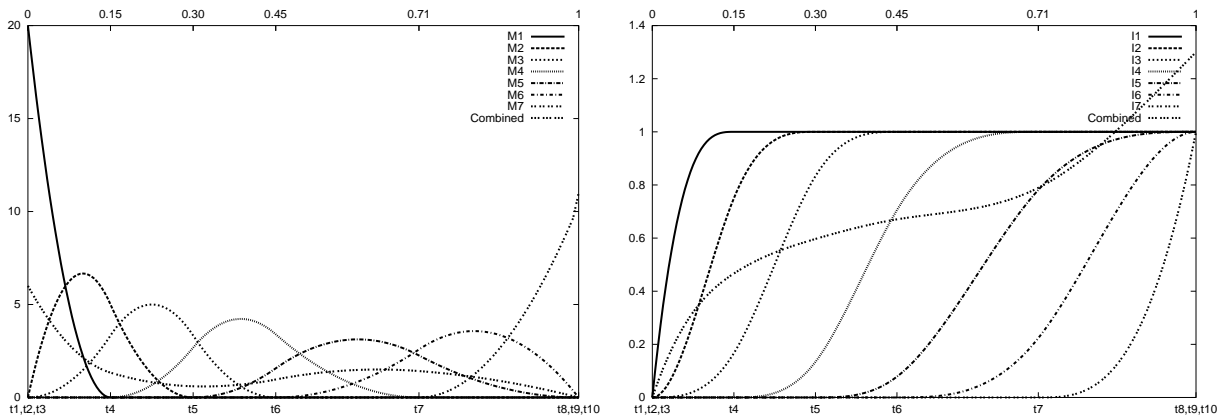


Fig. 2. Examples of spline bases with  $M$ -splines left and  $I$ -splines right. The linear combinations  $0.3M_1 + 0.2M_2 + 0.1M_3 + 0.1M_4 + 0.4M_5 + 0.2M_6$  and  $0.3I_1 + 0.2I_2 + 0.1I_3 + 0.1I_4 + 0.4I_5 + 0.2I_6$  are shown as the bold dotted lines. The interior knots are situated at 0.15, 0.3, 0.45, and 0.71. The spline order in this example is 3.

### 3.2. 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. In this subsection we briefly explain how (monotone) splines work. In Section 4 we present the results obtained with this technique.

A useful reference for monotone regression splines is the paper by Ramsay [13], where the application in nonmetric MDS is also mentioned. In the article, two types of splines are discussed: normal regression splines ( $M$ -splines) and monotone splines ( $I$ -splines). We will briefly explain both types below. For a more elaborate treatment, the reader is referred to [13], upon which the discussion below is inspired.

#### 3.2.1. $M$ -splines

Basically  $M$ -splines are piecewise polynomials with a certain degree  $k - 1$ , and thus order  $k$ , which have a nonzero value in a limited interval. Multiple  $M$ -splines are used to form a spline basis, which can be linearly combined for function fitting. Figure 2 displays a basis of 7  $M$ -splines. The first  $M$ -spline ( $M_1$ ) is nonzero in the interval  $[0;0.15]$ , the third one ( $M_3$ ) in  $[0;0.45]$ . All  $M$ -splines in the figure are of order 3.

Central to a basis of  $M$  splines is the so-called *knot sequence*  $t = [t_1, \dots, t_{n+k}]$ . Knots are placed on a mesh  $\Delta$  consisting of points  $L = \xi_1 < \xi_2 < \dots < \xi_q = U$  where  $q \leq n - k + 2$ . In the example  $\Delta = 0, 0.15, 0.3, 0.45, 0.71, 1, q = 6, k = 3, n = 7$  and  $t = [0, 0, 0, 0.15, 0.3, 0.45, 0.71, 1, 1, 1]$ . In general,  $t_1 \leq \dots \leq t_{n+k}$ , and for all  $i$  there is some  $j$  such that  $t_i = \xi_j$ .

Commonly, one knot is placed onto each interior mesh point, while  $k$  knots are always placed at  $L$  and  $U$ . As will be explained shortly, this allows for discontinuity at  $L$  and  $U$ . In general, the knot sequence has the properties  $t_1 = \dots = t_k = L$  and  $t_{n+1} = \dots = t_{n+k} = U$  and  $t_i < t_{i+k}$  for all  $i$ .

The number of knots that is placed on one mesh point determines the order of continuity with which adjoining splines ‘meet’ in this mesh point: the fewer knots, the higher the order of continuity. When  $1 \leq m \leq k$  knots are placed on a mesh point  $\xi$ , splines  $M_i$  and  $M_{i+k}$  that meet in this mesh point agree in their derivatives up to the  $(k - m - 1)$ -th order, i.e.,  $M_i^{(\ell)}(\xi) = M_{i+k}^{(\ell)}(\xi), \ell = 0, \dots, k - m - 1$ . In the example there was 1 knot placed at 0.15 so  $M_1$  and  $M_4$  have common values in the first  $3 - 1 - 1 = 1$

derivatives at 0.15, so we have second order continuity in 0.15. The boundaries  $L$  and  $U$  contain 3 knots each.

Ramsay [13] gives a recursive definition of the  $M$ -splines  $M_1, \dots, M_n$ :

$$M_i(x|1, t) = \begin{cases} \frac{1}{t_{i+1}-t_i} & \text{if } t_i \leq x < t_{i+1}, \\ 0 & \text{otherwise,} \end{cases} \tag{6}$$

$$M_i(x|k, t) = \frac{k[(x-t_i)M_i(x|k-1, t) + (t_{i+k}-x)M_{i+1}(x|k-1, t)]}{(k-1)(t_{i+k}-t_i)} \text{ for } k > 1. \tag{7}$$

A linear combination  $\sum_{i=1}^n a_i M_i$  of the  $M$ -splines can be used in nonlinear regression.

### 3.2.2. $I$ -splines

Since  $M$ -splines are essentially nonnegative polynomials of degree  $k - 1$ , integration of an  $M$ -spline yields a monotonically increasing polynomial of degree  $k$ . This polynomial is referred to as an  $I$ -spline:

$$I_i(x|k, t) = \int_L^x M_i(u|k, t) du . \tag{8}$$

Again according to Ramsay, for knot sequences with one knot at each interior mesh point, for which  $t_j \leq x < t_{j+1}$  for all  $x$ ,  $I_i$  can be written as follows:

$$I_i(x|k, t) = \begin{cases} 0 & \text{if } i > j, \\ \frac{\sum_{m=i}^j (t_{m+k+1}-t_m)M_m(x|k+1, t)}{k+1} & \text{if } j - k + 1 \leq i \leq j, \\ 1 & \text{if } i < j - k + 1. \end{cases} \tag{9}$$

A basis of  $I$ -splines can be used in monotonic regression by imposing a nonnegativity constraint on the coefficients in  $\sum_{i=1}^n a_i I_i$ . In the case of a squared error function (like Eq. (4)) this leads to a quadratic programming problem, which is easily solved by a quadratic programming solver. We used the SQP solver BPMPD by Mészáros [11].

### 3.3. Nonmetric phase using a monotone neural network

A fourth possibility for the nonmetric phase is the use of 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  $\hat{\delta}$  as inputs and generates the disparities  $\tilde{\delta}$  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 network except the biases. A network of this kind is depicted in Fig. 3.

We implemented two approaches to enforce the positivity constraint on the weights. 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, when dissimilarity  $\delta_{ij}$  is offered, would be  $in_{b|ij} = w_b^2 \delta_{ij} + \theta_b$ , where  $w_b$  and  $\theta_b$  denote unit  $b$ 's weight and bias. The learning rule has to be altered to incorporate the squared weights. Details can be found in [15]. This minimization procedure takes very long to converge and its use is therefore not recommended.

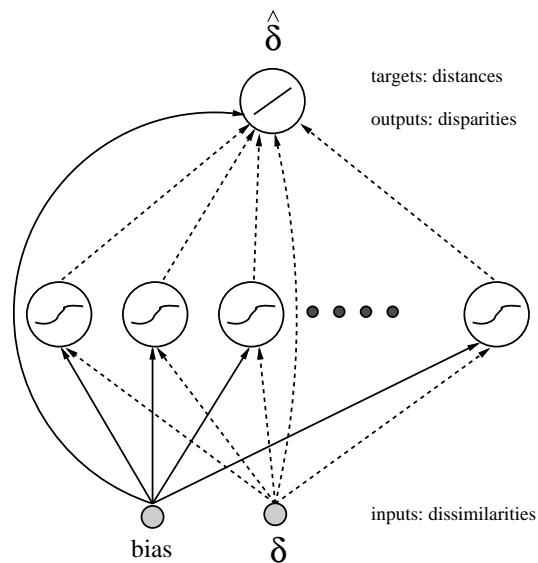


Fig. 3. Monotone neural network for nonmetric phase in MDS.

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.<sup>4</sup> 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 amount of computation required by the SQP method is acceptable.

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 following datasets:

**cars:** This dataset, shown in Table 1, comes with SPSS and contains pairwise dissimilarity judgments on 11 car models. The dataset was created by having a test person make a dissimilarity ranking of all 55 possible model pairs, which was done by splitting the original 55 dissimilarities into two sets – one set of similar pairs and one set of dissimilar pairs – and applying this procedure recursively onto both subsets. This yields rank ordered dissimilarities. The procedure is described in [5].

<sup>4</sup>Our constraints are linear while our objective function is nonlinear. Algorithms for this special case also exist (see [4]) but an implementation of such a method was not available to us.



Table 1  
The cars dataset

	FordM	Merc	Linc	FordT	FordF	Chry	Jag	AMC	Plym	Buick
Merc	8									
Linc	50	38								
FordT	31	9	11							
FordF	12	33	55	44						
Chry	48	37	1	13	54					
Jag	36	22	23	16	53	26				
AMC	2	6	46	19	30	47	29			
Plym	5	4	41	25	28	40	35	3		
Buick	39	14	17	18	45	24	34	27	20	
Chevy	10	32	52	42	7	51	49	15	21	43

Table 2  
The riasec dataset: Vocational preference dissimilarity data

	R	I	A	S	E	C
Realistic	0.0000	1.0392	1.2961	1.2570	1.1832	1.1314
Investigative	1.0392	0.0000	1.1489	1.1832	1.2961	1.2961
Artistic	1.2961	1.1489	0.0000	1.0770	1.1402	1.3342
Social	1.2570	1.1832	1.0770	0.0000	0.9592	1.1136
Enterprising	1.1832	1.2961	1.1402	0.9592	0.0000	0.8000
Conventional	1.1314	1.2961	1.3342	1.1136	0.8000	0.0000

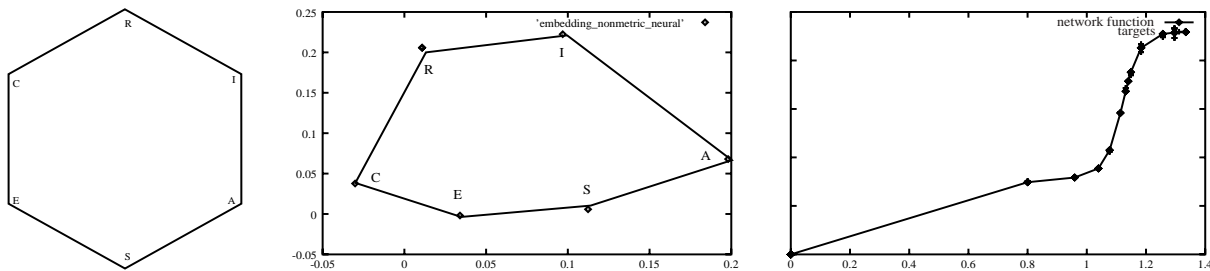


Fig. 4. Theoretical hexagon for vocational preferences (left). Embedding found by neural MDS (middle). Transformation implemented by mono-nn (right).

**square:** This dataset was artificially generated. First, we created a dataset consisting of 16 points arranged in a square in  $[0; 1]^2$ . The distances between all points were computed and transformed by a nonlinear transformation:  $0.5(\tanh(5(x - 0.5)) + 1)$ . This yielded the dissimilarities  $\delta$  that we used as a starting point for our algorithm.

**riasec:** We adopted this dataset from the MDS textbook of Davison [2] who, in turn, adopted it from psychometric literature. The dataset is shown in Table 2. It contains dissimilarities between six ‘vocational preference inventory’ scales in a sample of 1234 men. In psychological literature these six occupational types are often displayed in a hexagon (see the left side of Fig. 4). It is claimed that the vocational interests of persons in occupational types adjacent to each other in the hexagon are more similar than the interests of people with occupational types more distant from each other.

**citations:** This dataset was adopted from a paper by Groenen and Heiser [6]. It consists of citation numbers between journals in the psychometric literature. Visualization of these data gives insight into the positioning of journals relative to each other – thematically linked journals will appear in each other’s vicinity on the map. Since the raw data in this table are similarities (reference counts)

Table 3

Normalized stress ( $E^{nm}$ ) values obtained for all four datasets with metric and various types of nonmetric scaling, averaged over 50 runs

	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	0.004855	0.012246	0.000019	0.036478
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	0.006940	0.007838	0.000472	0.022375	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	0.009947	0.010895	0.000071	0.035881	0.057731	0.002347	0.053119	0.064688

rather than dissimilarities, they were transformed into dissimilarities using the transformation

$$\delta_{ij} = \frac{m_i + m_j}{n_{ij}}, \quad (10)$$

where  $n_{ij}$  is the number of citations between journals  $i$  and  $j$ , i.e., half of the the number of citations from journal  $i$  to journal  $j$  and vice versa.  $m_{i+}$  denotes the number of citations to journal  $i$  and  $m_{+j}$  is the number of citations from journal  $j$ .

Table 3 shows the resulting normalized stress values of the embeddings of these datasets. All results are averaged over 50 independent runs, to compensate for the variance in the final error values due to different initial configurations. The normalized error values for metric MDS can be computed by dividing  $E_{\text{metric\_mds}}$  by  $(\sum_{ij, i \neq j} d_{ij}^2)^2$ , which is equivalent to normalizing both dissimilarities and distances by dividing by  $\sum_{ij, i \neq j} d_{ij}^2$ . The dimension of the embedding space was 2 in all experiments.

One can see that the 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 normalized stress value, but the spline based monotone regression outperforms the neural network by a small factor. More specifically, when neural network based nonmetric scaling is applied the metric normalized stress level is reduced by a factor 4.52 on average, whereas the use of monotone splines in the nonmetric phase reduces the metric normalized stress level by a factor 5.34.

The monotone transformations yielded for the `citations` dataset by the various scaling methods are depicted in Fig. 5. It is clearly visible that the non-interpolating methods yield a non-smooth transformation, which is not very plausible. Also shown in Fig. 5 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: the original square shaped embedding is nearly perfectly reconstructed by the nonmetric embedding, whereas the metric embedding has a rounder appearance.

In order to verify the hypothesis that vocational interests of persons in occupational types adjacent to each other in the hexagon are more similar than the interests of people with occupational types more

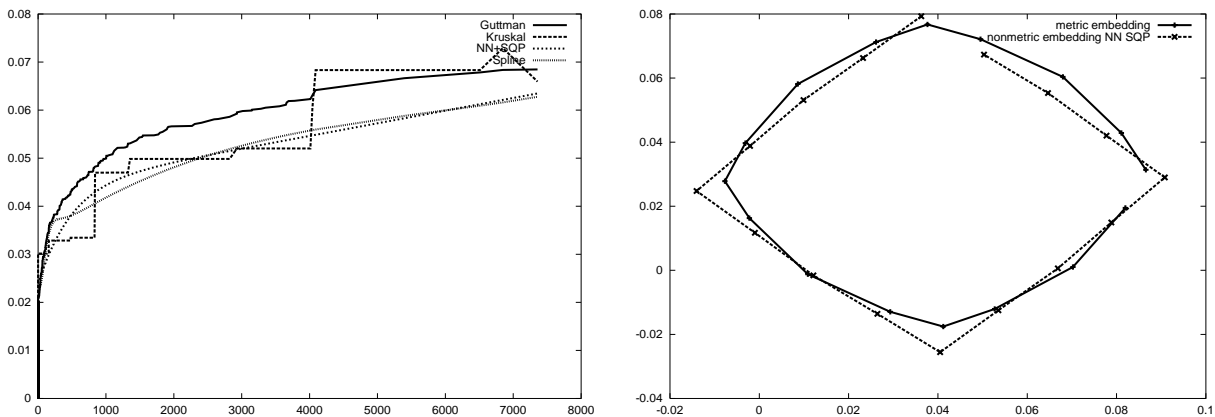


Fig. 5. Transformations yielded by the various nonmetric phases for the citations dataset (left), and embeddings yielded for the square dataset (right) by metric and nonmetric scaling.

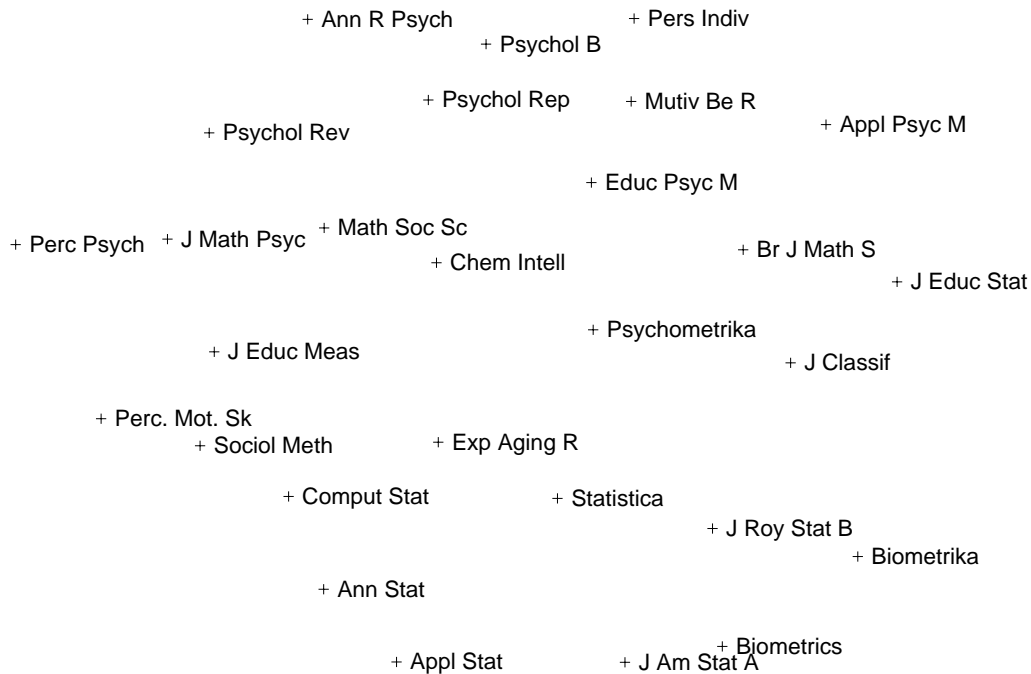


Fig. 6. Nonmetric embedding of the citations dataset.

distant from each other, we applied our neural nonmetric scaling algorithm to the riasec dataset and ended up with the embedding shown in the middle of Fig. 4. In this embedding, the data points representing the six occupational types have indeed settled themselves in a roughly hexagonal shape.

An embedding of the citations dataset yielded by nonmetric MDS using a neural network in the nonmetric phase is shown in Fig. 6. Note that the statistically oriented journals are located at the bottom of the embedding, while psychological journals are located at the top.

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,

Table 4

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

	Number of objects $n$					$t_{500}/t_{10}$
	10	50	100	300	500	
Metric	0.01 (0.019)	0.21 (0.027)	2.03 (0.027)	9.99 (0.024)	34.01 (0.020)	3401
Kruskal	0.28 (0.000)	8.81 (0.000)	28.83 (0.000)	113.87 (0.000)	193.55 (0.000)	689
Guttman	0.11 (0.000)	0.39 (0.011)	21.00 (0.000)	66.79 (0.000)	209.10 (0.000)	1900
Spline	2.33 (0.000)	1.54 (0.000)	3.86 (0.000)	35.20 (0.000)	84.11 (0.000)	36
Neural	19.36 (0.000)	79.64 (0.003)	502.38 (0.001)	5051.14 (0.001)	89566.09 (0.001)	4626
SQP Neural	2.95 (0.000)	31.73 (0.000)	75.90 (0.003)	644.76 (0.004)	615.80 (0.006)	208

calculating the distance matrix and applying a nonlinear monotone transformation ( $f(x) = \log(x + 0.01) + 5$ ) to these distances. The same metric phase based on a conjugate gradient algorithm was used for all experiments. All methods were implemented in C++ and compiled with the gnu C++ compiler gcc version 2.96 under Mandrake Linux 8.1 on an IBM thinkpad type 2655 PC with a 797 MHz Intel Celeron CPU.

Table 4 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 inherent simplicity of this method. It is clear that the monotone neural network, when trained with the modified backpropagation procedure of [15] is unacceptably slow. But when the SQP method is used, the required computation times are acceptable, whereas the results reported in Table 3 indicate that the obtained normalized stress values are usually lower. However, it should be pointed out that the monotone spline method outperforms the monotone neural network both in terms of embedding quality and computation speed.

## 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 backpropagation 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 – it was able to reduce the normalized stress value yielded by metric embedding by a factor 4.52 on average – but have the advantage of yielding smooth mappings instead of step functions, which is more plausible and makes interpolation easier. We have thus shown that the use of neural networks in the nonmetric phase of MDS is a sensible approach.

A possible benefit of the use of monotone neural networks is that they are potentially faster than traditional nonmetric procedures for large dissimilarity matrices. This advantage also holds for the monotone spline approach, which was found to outperform the monotone neural network both in terms of embedding quality and speed of the MDS process. In practice, the use of monotone splines is therefore to be preferred over the use of monotone neural networks.

## Acknowledgements

The authors thank professor J. Ramsay for making the Fortran code for  $I$ -splines and  $M$ -splines available, and the anonymous referees and professor P. Groenen for commenting on the manuscript.

## References

- [1] I. Borg and P. Groenen, *Modern Multidimensional Scaling. Theory and Applications*, Springer Series in Statistics, Springer-Verlag, New York, 1997.
- [2] M.L. Davison, *Multidimensional Scaling*, Wiley, New York, 1983.
- [3] J. de Leeuw, Differentiability of Kruskal's stress at a local minimum, *Psychometrika* **49**(1) (1984), 111–113.
- [4] R. Fletcher, *Practical Methods of Optimization* (2nd edition), Wiley, 1987.
- [5] P.E. Green, F.J. Carmone and S.M. Smith, *Multidimensional Scaling – Concepts and Applications*, Allyn and Bacon, 1989.
- [6] P.F.J. Groenen and W.J. Heiser, The tunneling method for global optimization in multidimensional scaling, *Psychometrika* **61**(3) (1996), 529–550.
- [7] L. Guttman, A general nonmetric technique for finding the smallest coordinate space for a configuration of points, *Psychometrika* **33** (1968), 469–506.
- [8] H. Klöck and J.M. Buhmann, Data visualization by multidimensional scaling: A deterministic annealing approach, *Pattern Recognition* **33** (1999), 651–669.
- [9] J.B. Kruskal, Multidimensional scaling by optimizing goodness-of-fit to a nonmetric hypothesis, *Psychometrika* **29** (1964), 1–29.
- [10] J.B. Kruskal, Non-metric multidimensional scaling: a numerical method, *Psychometrika* **29** (1964), 115–129.
- [11] Cs. Mészáros, *The BPMPD interior point solver for convex quadratic problems*, Technical Report WP 98-8, Hungarian Academy of Sciences, Budapest, 1998.
- [12] W.H. Press, S.A. Teukolsky, W.T. Vetterling and B.P. Flannery, *Numerical Recipes in C*, (2nd edition), Cambridge University Press, Cambridge, 1988.
- [13] J.O. Ramsay, Monotone regression splines in action, *Statistical Science* **3**(4) (1988), 425–441.
- [14] 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), Brussels, 1997, pp. 97–102, D factio.
- [15] 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** (2001), 145–156.