A new method of generalizing Sammon mapping with application to algorithm speed-up

Elżbieta Pękalska[*], Dick de Ridder[*], Robert P.W. Duin[*], Martin A. Kraaijveld[†]

[*]Pattern Recognition Group, Department of Applied Physics, Faculty of Applied Sciences,
Delft University of Technology, Lorentzweg 1, 2628 CJ Delft, The Netherlands

[†]Shell International E&P, PO Box 60, 2280 AB Rijswijk, The Netherlands

email: {ela,dick}@ph.tn.tudelft.nl

Keywords: Sammon mapping, multidimensional scaling, triangulation, neural networks, distance mapping

Abstract

Sammon mapping comes from the area of multidimensional scaling and is an important pattern recognition tool. It is a nonlinear projection method, which reveals the structure present in data. Sammon mapping has two disadvantages. Firstly, it lacks generalization, which means that new points cannot be added to the obtained map without recalculating it. Secondly, because it operates on all interpoint distances, the complexity of finding the mapping is very high. The solution to the first problem is used in improving the speed of the algorithm.

To save computation time without losing the mapping quality, we investigate three possible speed-ups. They are hybrid methods being a combination of Sammon mapping, based on a subset of points, and respectively: triangulation, neural network and our proposal, distance mapping. These approaches are verified by some experiments, showing that distance mapping performs the best.

1 Introduction

Understanding data is often a difficult task, especially when it refers to a complex phenomenon that is described by many variables. If the data consists of one or two variables, many simple methods are available, showing or emphasizing some of the properties or relations between objects. But, when multivariate data is examined, it is nearly impossible to comprehend its structure. This presents a need for more sophisticated techniques.

The early stage of data analysis is to visualize data on a plane or in a $3D$ space. By this, a researcher hopes to gain some intuition about the data and to understand the relations between objects, see the intrinsic structure or possible cluster tendencies, etc. Techniques allowing to visualize high-dimensional data in a low-dimensional space are called *projection* methods. In this area, nonlinear projection techniques play an important role. They look for a nonlinear manifold imposing e.g. the preservation of all (or some) interpoint distances in the mapping.

Such techniques are powerful tools for data visualization and exploration, but as iterative processes, they are time consuming. In this paper we consider one type of such projections: Sammon mapping [2, 8], which is a technique of high complexity. We are interested in a significant speed-up of this method. We investigate three possible approaches: triangulation, artificial neural network (ANN) and distance mapping in order to save computation time.

The paper is organized as follows. Section 2 describes the idea of Sammon mapping and explains its complexity. Section 3 presents the triangulation and ANN methods and introduces distance mapping, all used for improving the speed of Sammon projection. Section 4 describes datasets and performed experiments and Section 5 shows results. In Section 6 conclusions are drawn.

2 Sammon mapping

A researcher is often interested in mappings that reveal the inherent structure in order to explore the data, to find possible clusters, correlations or underlying distributions. Methods from the area of multidimensional scaling (MDS) [2] meet these needs. During the mapping, the MDS projections aim to preserve all interpoint distances and by this they reveal the structure present in the data. One of such algorithms is Sammon mapping [2, 8].

Suppose that we consider a set of n objects. Each object is represented by one point in an m-dimensional (high-dimensional) space. The aim of Sammon mapping is to find n points in a d-dimensional space (with $d < m$), in such a way that the corresponding distances approximate the original ones as well as possible. We denote:

- $d_{ij}, \forall\, i,j = 1,\dots,n$ - the distance between two points in an m-dimensional space.
- $\delta_{ij}, \forall\, i,j = 1,\dots,n$ - the distance between two points in a d-dimensional space.

Without loss of generality, only projections onto a 2-dimensional space are studied ($d = 2$), since our interest is in data visualization.

There is a need for a criterion to decide whether one configuration is better than another. For that purpose, the error (stress) function E is considered, which measures the difference between the present configuration of n points in the d-dimensional space and the configuration of n points in the original m-dimensional space. The stress is given by the following formula:

$$E = \frac{1}{\sum_{i=1}^{n-1}\sum_{j=i+1}^{n}\delta_{ij}} \sum_{i=1}^{n-1}\sum_{j=i+1}^{n} \frac{(\delta_{ij}-d_{ij})^2}{\delta_{ij}} \qquad (1)$$

and yields in fact a badness-of-fit measure for the entire representation. The stress range is $[0,1]$ with 0 indicating a lossless mapping.

2.1 A minimization problem

The problem of finding the right configuration in a low-dimensional space is an optimization problem: we are interested in obtaining such a configuration that the stress function yields minimum. In general, this optimization problem is difficult because of the very high dimensionality of the parameter space. The stress function is optimal when all the original distances δ_{ij} are equal to the distances of the projected points d_{ij}. However, this is not likely to happen exactly. Therefore, the found distances will be distorted representations of the relations within the data. The larger the stress, the greater the distortion.

To find a projected map, we start from the initial configuration of points (e.g. randomly chosen), and then the stress is calculated by using equation (1). Next, the configuration is improved by shifting around all points in small steps to approximate better and better the original distances (thus decreasing the stress). This process is reiterated, until the map corresponding to a (local) minimum of the stress is found.

2.2 Algorithm complexity

The complexity of Sammon mapping is high, because the stress function is based on $\mathcal{O}(n^2)$ distances. In order to speed up this mapping without losing the projection quality, two possible approaches can be investigated. The first improvement relates to a choice of suitable minimization algorithm. Examples of such algorithms are e.g. Pseudo-Newton, Conjugate Gradients or Scaled Conjugate Gradients (SCG) [7]. In the earlier experiments, we concluded that although the SCG technique tends to be the fastest in many cases, the Pseudo-Newton works well in general. Therefore, in our study we limit ourselves to this algorithm.

The second improvement concerns the idea of performing Sammon mapping only on a subset of all points. The remaining points are then added to the obtained map in a way that avoids stress distortion. Our problem is that there is no generalization in Sammon mapping. To solve this, three possible approaches will be discussed next.

3 Fast variants of Sammon mapping

In this paper possible ways of significantly speeding up the Sammon algorithm are studied. The general idea is to use a subset taken from the original points and project it by Sammon mapping. The remaining points can then be added to the existing map. To reduce the problem complexity, we are interested in a way of adding points, which requires relatively little computation.

Unfortunately, Sammon mapping does not give any mathematical rule or algorithmic procedure for mapping previously unseen points. It means that when a new point is to be projected, the entire map should be recalculated. To solve this, some alternative ways of projecting new points can be considered. Three possible methods are investigated and compared: the use of triangulation [1, 5], an ANN [3] and distance mapping (to be introduced).

3.1 Triangulation

The triangulation method [1, 5] is based on the observation that when a new object is projected, its distances to two points (for 2D-mappings) previously mapped can be preserved exactly. If there are n points in a m-dimensional space, then they can be mapped in sequence on a plane in such a way that $(2n-3)$ of the $n(n-1)/2$ distances are exactly preserved. This is a fast algorithm, since it is non-iterative and preserves only some distances. Figure 1 explains the triangulation algorithm. To find the mapping X' of a point X, its distances to points A and B in the original space, d_{AX} and

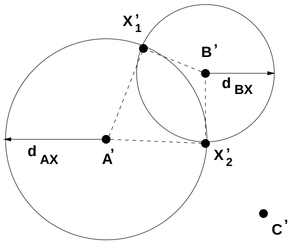


Figure 1: An explanation of triangulation method.

d_{BX}, have to be preserved. It allows to find a desired value on a plane, for the distances between X' and the mappings A' and B'. The requirements can result in no candidate points (if the circles do not overlap), one candidate point (if the circles touch) or two candidate points. In the first case a simple linear interpolation weighed with d_{AX} and d_{BX} is used; in the last case a third point C decides which candidate is chosen (see Figure 1). The point X'_2 is taken, as the distance between X'_2 and C' approximates better the distance between X and C.

This hybrid method first creates a base from the data by projecting p of the all n points $(p < n)$ on the plane, using the Sammon iterative technique. The p projected patterns are then fixed and the remaining $(n - p)$ points are projected sequentially, using the triangulation method.

3.2 Artificial neural networks

Besides triangulation, one can use an ANN to approximate a Sammon mapping. The training set consists of the original, high-dimensional input samples. The target outputs are the Sammon map coordinates. Although Mao and Jain [6] developed a special-purpose training rule (SAMANN) for such a network, it was shown [3] that normal networks trained on a Sammon mapping generally outperform this SAMANN both in terms of stress and computational cost.

A problem with using ANNs for Sammon mapping is that, in principle, the output of a Sammon mapping is unbounded, whereas many ANNs use sigmoid transfer functions which can only give output in the range $[0, 1]$.

One possible solution is to rescale both the dataset and the map in such a way that the target outputs lie in the range of the output units. However, it is not known beforehand whether the Sammon map used to train the ANN accurately represents the possible ranges; i.e., the part of the dataset yet to be mapped may give large Sammon map coordinate values.

A second solution is to use linear transfer function output units. Although ANNs containing these units can be harder to train (since the error gradient is not bounded as it is when using sigmoid transfer functions), they can represent any $\mathcal{R}^m \rightarrow \mathcal{R}^d$ mapping. We chose this solution in our experiments.

3.3 Distance mapping

Another possibility to save both memory and computation time is distance mapping, which is our proposal. It looks for a linear transformation of the distance matrix in the high-dimensional space, so that the low-dimensional Sammon configuration is obtained. After determining that operator, it is applied to the distance matrix computed between new points and old points (being in the Sammon map).

Let us denote the base of p points, chosen for Sammon mapping, as $\boldsymbol{X_{base}} \in \mathcal{R}^{p \times m} (p < n)$. The corresponding distance matrix is $\boldsymbol{D_{base}}$. The d-dimensional Sammon result is denoted by the matrix $\boldsymbol{Y_{base}} \in \mathcal{R}^{p \times d}$. The matrix $\boldsymbol{D_{base}}$ as a distance matrix is full rank, thus, by applying a linear transformation $\boldsymbol{V} \in \mathcal{R}^{p \times d}$ to this, the matrix $\boldsymbol{Y_{base}}$ can be found. The matrix $\boldsymbol{V}$ is defined as:

$$\text{define } \boldsymbol{V} : \quad \boldsymbol{D_{base}} \boldsymbol{V} = \boldsymbol{Y_{base}} \qquad (2)$$

Having found the linear operator $\boldsymbol{V}$, the new (remaining) points can be added to the low-dimensional map (denoted by $\boldsymbol{Y_{base}}$) as follows:

$$\text{find } \boldsymbol{Y_{new}} : \quad \boldsymbol{Y_{new}} = \boldsymbol{D_{new-to-base}} \boldsymbol{V}, \qquad (3)$$

where the matrix $\boldsymbol{D_{new-to-base}} \in \mathcal{R}^{(n-p) \times p}$ is the distance matrix consisting of all distances between points from the sets: $\boldsymbol{X_{base}}$ and $\boldsymbol{X_{new}}$.

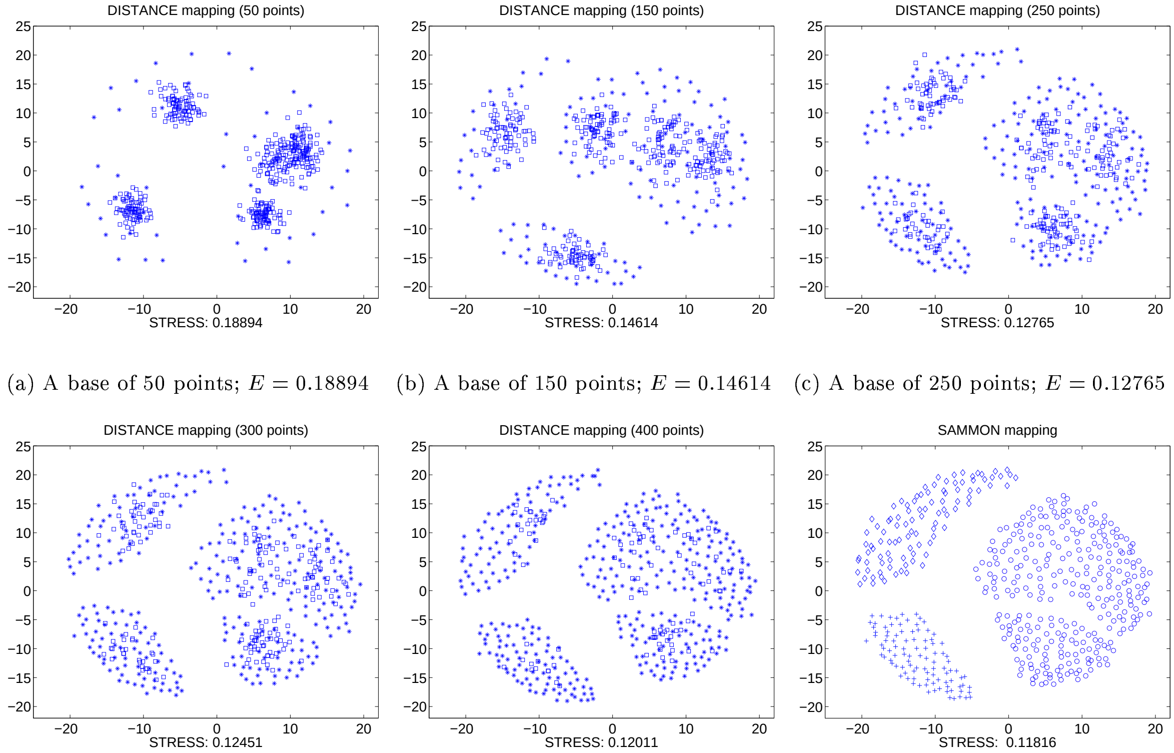
4 Experiments

Considering the hybrid methods, there are two important questions. The first one concerns the necessary number of points used to build a base for Sammon mapping. The second question is how well these techniques perform. We will study their performance by determining the number of floating point operations (FLOPS) needed for the whole mapping and the values of the stress measure.

4.1 Optimal size of a base

The optimal size of a base seems to be a crucial question. We want to avoid large stress distortion, but also allow a significant speed improvement of the hybrid methods. To find a good trade-off between these two aspects, different base sizes were investigated. All bases were chosen randomly.

Figure 2 presents results of distance mapping with different base sizes for pump vibration dataset (see Section 4.2). It is not surprising that the stress measure after distance mapping is larger than for Sammon mapping performed for all points. However, there is a general tendency of the stress to decrease when the number of points included in a base



(a) A base of 50 points; $E = 0.18894$ (b) A base of 150 points; $E = 0.14614$ (c) A base of 250 points; $E = 0.12765$

(d) A base of 300 points; $E = 0.12451$ (e) A base of 400 points; $E = 0.12011$ (f) Sammon mapping; $E = 0.11816$

Figure 2: Distance mapping results for pump vibration data with bases (marked by stars) of different sizes.

is increasing. For a base of 250 points, the stress (0.12763) seems to be close enough to the original one (0.11816). Visual judgment confirms that the resulting map (see Figure 2(d)) resembles the Sammon map (see Figure 2(f)) quite well.

To make a proper choice, the curves of averaged (over all distances) Sammon stress for TRAINING set (points chosen for a base for Sammon projection), TESTING set (points added by distance mapping) and ALL (all points) were studied. These curves can be observed in Figure 3. We want to take this base size, for which all curve are enough close to each other, which suggests that a good generalization is reached. For $250 - 300$ points, being $50\% - 60\%$ of all points, the stress for all curves decreases significantly and takes more similar values. Keeping in mind that the complexity of our problem decreases quadratically with the number of points used for Sammon mapping, a choice of 50% of all points to build a base, seems to be a reasonable one. Therefore, as a rule of thumb, we will consider a base of that size in our experiments.

4.2 Datasets

The following five datasets were investigated:

- *Circle dataset*: An artificial dataset consisting of 200 points lying on 2 parallel circles in 3D.
- *Iris dataset*: A real dataset characterizing 3 species of iris flowers. In total, 150 observations are available. The iris flowers are described by 4 attributes.
- *Pump vibration dataset*: Pump vibration was measured with 3 accelerometers mounted on a submersible pump which was operated in 3 states: normal, presence of imbalance and presence of bearing failure. The data consists of 500 observations with 256 spectral features of the acceleration spectrum (see [9]).
- *Handwritten digit dataset*: The digits were taken from NIST special database 3 [4]. The dataset consists of 2000 points equally distributed over ten possible classes. Each point corresponds to a grey value image with 16×16 pixels. Thus the data dimensionality is 256.
- *Borehole dataset*: The borehole dataset is composed of samples taken from one hydrocarbon well. It consists of 11 petrophysical logs, 6 image derive features and 2 core measurements. This dataset includes 1001 points. It contains information on 6 different facies (types of rock).

These datasets cover a wide dimensionality range, as there are examples of low, average and high di-

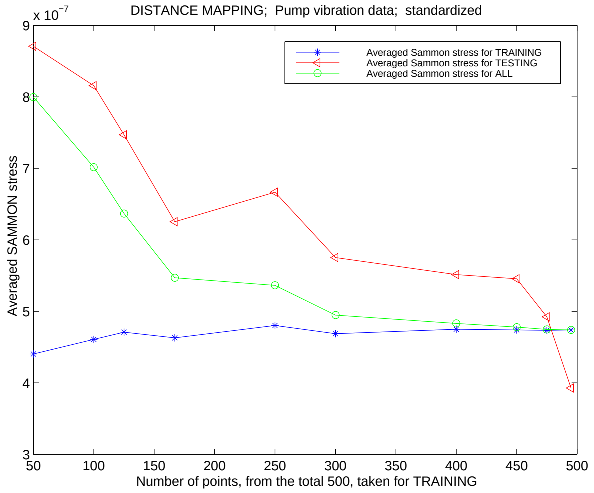


Figure 3: The averaged stress for distance mapping.

mensionality. They also contain different numbers of points.

4.3 Experimental setup

All experiments concerning the hybrid methods of Sammon and respectively: triangulation, an ANN or distance mapping were performed for all datasets. For each dataset, 10 random subsets (bases), consisting of 50% of all original points, were randomly chosen. Then, Sammon mappings were done for each base. The points excluded from bases were later added to the obtained configurations. All routines were implemented in MATLAB.

4.4 Triangulation

The triangulation method does not need any training. Points are added sequentially to the map using the nearest neighbourhood relations in the original space. Therefore, the time required for adding points is only the time needed to apply this mapping.

4.5 ANN architecture

Neural networks were trained on the 10 chosen subsets (bases) of data. To get an idea of the variability in network performance, for each of these subsets 10 networks were trained. The used network architecture consisted of m input units (where m is the dimensionality of the original dataset), 2 hidden layers of 25 units with sigmoid transfer functions each and an output layer containing 2 linear transfer function units. The networks were trained in MATLAB's Neural Network Toolbox, using the SCG training algorithm [7]. Training was stopped when the MSE reached $1e{-}06$ or after 10,000 steps, whichever came first.

A problem arose with the *digit* dataset. The target outputs were very high, in the order of $1e03$. The ANNs are usually initialized with small weights and may take quite a long time to reach these output values. To facilitate training, target values for this dataset were divided by $1e03$.

Note that some ANNs did not converge at all (3% of the ANNs trained on the *iris* dataset). Results for these ANNs were not taken into account, since they are not representative.

One could argue that the stopping criteria for network training are a bit arbitrary. If training were stopped when the gradient reached say $1e{-}04$, training times would be shorter. However, as ANNs are trained using the MSE between target and actual outputs rather than the stress, it is hard to predict what the resulting stress would be. In practice, one trains ANNs so that the MSE is as low as possible. Calculating the stress during training could give more insight into the relation between the MSE and the stress, but also adds computational complexity.

4.6 Distance mapping

Training distance mapping can be understood as determining the linear operator $\boldsymbol{V}$ (see expression 2). This requires some computations, as it means calculating the distance matrix and then solving a linear, well-defined equation of order m. Having found the matrix $\boldsymbol{V}$, a mapping of new points is then easily and cheaply performed.

5 Results

As discussed in Section 4.1, for our experiments, a base consisting of 50% of all points were randomly chosen. The performance of our methods is shown in Table 1 and Table 2. Table 1 presents an averaged number of floating point operations (FLOPS) with their standard deviations for all considered datasets. Some observations are:

- Both ANN and distance mapping require some time for training. However, for ANNs the averaged number of FLOPS is quite large and surpasses not only the number of FLOPS for distance mapping, but also for Sammon projection. In the aspect of adding new points, the ANN and distance mapping perform similarly.
- Concerning the total amount of FLOPS needed for all hybrid methods, one can easily observe that the number of FLOPS performed for triangulation and distance mapping is relatively the same, and an order of magnitude less than for Sammon mapping performed for all points. The total number of FLOPS needed for the ANN, because of its costly training, is at least one order of magnitude higher than in case of Sammon mapping for all points.

Table 1. Averaged number of FLOPS for mappings.

Dataset	Method	Number of FLOPS			
		Sammon (base)	Training mapping	Applying mapping	Total
Circle	Sammon	2.92e08	——	——	2.92e08
	Triang.	2.19e07 ± 5.74e06	——	6.27e05 ± 2.33e04	2.25e07 ± 5.73e06
	ANN	2.19e07 ± 5.74e06	3.20e09 ± 1.68e09	1.69e05	3.22e09 ± 1.69e09
	Distance	2.19e07 ± 5.74e06	8.69e05 ± 7.87e02	1.68e05	2.29e07 ± 5.74e06
Iris	Sammon	2.74e07	——	——	2.74e07
	Triang.	7.89e06 ± 1.78e06	——	4.76e05 ± 3.76e03	8.36e06 ± 1.78e06
	ANN	7.89e06 ± 1.78e06	7.73e09 ± 4.34e06	1.30e05	7.74e09 ± 6.12e06
	Distance	7.89e06 ± 1.78e06	4.21e05 ± 3.54e02	1.07e05	8.42e06 ± 1.78e06
Pump	Sammon	3.36e09	——	——	3.36e09
	Triang.	8.33e08 ± 1.21e08	——	1.32e08 ± 1.38e04	9.65e08 ± 1.21e08
	ANN	8.33e08 ± 1.21e08	1.05e10 ± 1.14e10	3.59e06	1.13e10 ± 1.15e10
	Distance	8.33e08 ± 1.21e08	4.38e07 ± 4.78e03	3.30e07	9.10e08 ± 1.21e08
Digits	Sammon	4.39e10	——	——	4.39e10
	Triang.	1.02e10 ± 2.13e09	——	2.08e09 ± 1.57e05	1.23e10 ± 2.13e09
	ANN	1.02e10 ± 2.13e09	3.97e11 ± 1.65e11	1.43e07	4.07e11 ± 1.67e11
	Distance	1.02e10 ± 2.13e09	1.18e09 ± 7.38e04	5.19e08	1.19e10 ± 2.13e09
Borehole	Sammon	1.22e10	——	——	1.22e10
	Triang.	2.61e09 ± 7.44e08	——	5.05e07 ± 1.88e05	2.66e09 ± 7.44e08
	ANN	2.61e09 ± 7.44e08	6.56e10 ± 2.52e06	1.25e06	6.81e10 ± 7.46e08
	Distance	2.61e09 ± 7.44e08	9.76e07 ± 1.55e04	1.23e07	2.72e09 ± 7.44e08

Table 2. Sammon stress for mappings.

Dataset	Original Sammon	Triangulation	ANN mapping	Distance mapping
Circle	0.04329	0.04605 ± 0.0024	0.04540 ± 0.0016	0.04526 ± 0.0015
Iris	0.00396	0.01197 ± 0.0013	0.00777 ± 0.0045	0.00603 ± 0.0007
Pump	0.11816	0.30136 ± 0.0145	0.12200 ± 0.0011	0.12761 ± 0.0020
Digits	0.12348	0.19844 ± 0.0033	0.13045 ± 0.0020	0.12846 ± 0.0004
Borehole	0.07212	0.10918 ± 0.0046	0.07993 ± 0.0018	0.07636 ± 0.0014

Table 2 presents an averaged stress calculated for Sammon and other hybrid methods for all datasets. Comparing the stress values between Sammon and its variants, a few conclusions can be drawn:

- Firstly, for the *circle* dataset, all stress measures are similar to each other. This is an artificial, non-complex dataset and locally linear, therefore all mappings perform well.
- Secondly, in case of real datasets, more differences can be observed. The ANN and distance mapping give similar stress measures, which are close to the original one. Therefore not much map distortion is caused either by the ANN or distance mappings. However, ANN mapping needs a large amount of time for training the network, which exceeds the time needed for the original Sammon mapping, so there is no speed-up to be gained by using an ANN.
- Finally, the triangulation method definitely gives the worst stress values.

Figure 4 presents some examples of the original Sammon and the hybrid methods. Points marked by stars belong to a base (and were projected by Sammon mapping), while points marked by squares correspond to new points, added to the existing map. Visual judgment confirms also that the maps obtained for ANN and distance mappings resemble the original ones quite well.

6 Conclusions and discussion

In order to speed up Sammon mapping without losing map quality, we considered and compared a combination of Sammon mapping, based on a subset of points, with three other methods: triangulation, an ANN and distance mapping. The last approach was introduced in this paper.

As a rule of thumb (as discussed in Section 4.1), for Sammon mapping, we chose a random base consisting of 50% of all points.

We investigated the performance of the methods by the experiments (described in Section 4) performed on five different datasets, which leads us to the following conclusions:

1. Both triangulation and distance mapping require similar amount of time for adding points, but triangulation gives larger map distortion (confirmed by both the larger stress values and visual judgment).
2. Although the networks approximate the Sammon mapping quite well, the computational cost required for training them is very high.
3. Therefore, distance mapping is the only recommended method in order to save both computation time and preserve the mapping quality.

One can also think of fine-tuning the final configuration by applying a few more iterations of Sammon mapping. However, in our experience the stress and the obtained configuration change only very little.

Sammon mapping itself does not allow for adding new points to the existing configuration. When new points appear in the original data, the whole map should be found again, which is a big disadvantage of this projection, as it requires many computations. A smart way of adding new points, which is proposed by us, seems to minimize this inconvenience. Distance mapping does not only allow for generalization, but it is the reasonable solution of improving the speed of Sammon mapping. Therefore, we solved two problems.

6.1 Further research

There are two issues for further study. Firstly, distance mapping might be improved further by reducing the considered distance matrix $\boldsymbol{D_{base}} \in \mathcal{R}^{p \times p}$, needed for determining the linear operator $\boldsymbol{V}$ (see equation 2), to a matrix $\boldsymbol{D'_{base}} \in \mathcal{R}^{p \times k}$ $(k < p)$. This would make the matrix $\boldsymbol{V}$ be a solution in the least square sense, which might generalize better.

Secondly, Sammon mapping is described by minimization of the stress function given by equation 1. However, this is a special case of more general stress function expressed as:

$$E_S = \frac{1}{\sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \delta_{ij}^{p+2}} \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \delta_{ij}^{p} \left(\delta_{ij} - d_{ij}\right)^2$$

with $p = ... - 2, -1, 0, 1, 2, ...$
Our results suggest that distance mapping can be also considered as a reasonable way both to generalize and to speed up any projection defined by the E_S function.

Acknowledgments

This research was performed as part of a project for Shell (SIEP), under project no. TN-97-036. It was partly supported by the Foundation for Computer Science in the Netherlands (SION) and the Dutch Organization for Scientific Research (NWO).

References

[1] G. Biswas, A.K. Jain, and R.C. Dubes. Evaluation of projection algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 3(6):701–708, 1981.

[2] I. Borg and P. Groenen. *Modern Multidimensional Scaling.* Springer-Verlag, New York, 1997.

[3] D. de Ridder and R.P.W. Duin. Sammon's mapping using neural networks: a comparison. *Pattern Recognition Letters*, 18(11-13):1307–1316, 1997.

[4] D. de Ridder, A. Hoekstra, and R.P.W. Duin. Feature extraction in shared weights neural networks. In E.J.H. Kerckhoffs, P.M.A. Sloot, J.F.M. Tonino, and A.M. Vossepoel, editors, *Proceedings of the* 2^{nd} *annual conference of the Advanced School for Computing and Imaging*, pages 289–294. ASCI, ASCI, 1996.

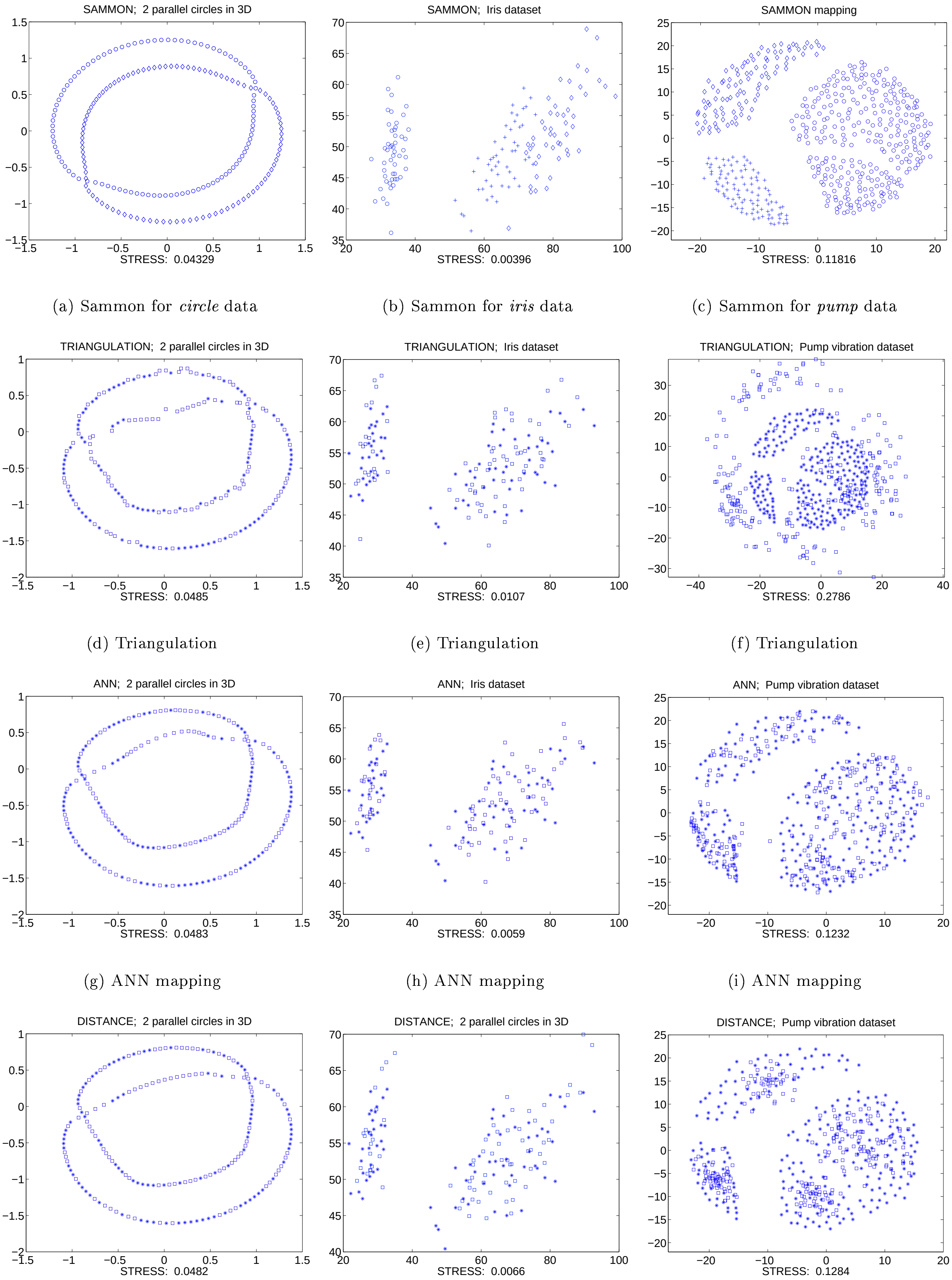
[5] R.C.T. Lee, J.R. Slagle, and H. Blum. A triangulation method for the dequential mapping of points from n-space to two-space. *IEEE Transactions on Computers*, C-26:288–292, 1977.

[6] J. Mao and A.K. Jain. Artificial neural networks for feature extraction and multivariate data projection. *IEEE Transactions on Neural Networks*, 6:296–317, 1995.

[7] M.F. Møller. *Efficient Training of Feed-Forward Neural Networks.* PhD thesis, Computer Science Department, Aarhus University, 1993.

[8] J.W. Sammon Jr. A nonlinear mapping for data structure analysis. *IEEE Transactions on Computers*, C-18:401–409, 1969.

[9] A. Ypma, E. Pekalska, and R.P.W. Duin. Domain approximation for condition monitoring. In B.M. ter Haar Romeny, D.H.J. Epema, J.F.M. Tonino, and A.A. Wolters, editors, *Proceedings of the* 4^{th} *annual conference of the Advanced School for Computing and Imaging*, pages 257–263. ASCI, ASCI, 1998.



(a) Sammon for *circle* data (b) Sammon for *iris* data (c) Sammon for *pump* data

(d) Triangulation (e) Triangulation (f) Triangulation

(g) ANN mapping (h) ANN mapping (i) ANN mapping

(j) Distance mapping (k) Distance mapping (l) Distance mapping

Figure 4: Examples of Sammon projected datasets and the mappings obtained by using hybrid methods.