



On the Initialisation of Sammon's Nonlinear Mapping

B. Lerner¹, H. Guterman², M. Aladjem², I. Dinstein²

¹University of Cambridge Computer Laboratory, Cambridge, UK; ²Department of Electrical and Computer Engineering, Ben-Gurion University, Beer-Sheva, Israel

Abstract: The initialisation of a neural network implementation of Sammon's mapping, either randomly or based on the principal components (PCs) of the sample covariance matrix, is experimentally investigated. When PCs are employed, fewer experiments are needed and the network configuration can be set precisely without trial-and-error experimentation. Tested on five real-world databases, it is shown that very few PCs are required to achieve a shorter training period, lower mapping error and higher classification accuracy, compared with those based on random initialisation.

Keywords: Classification; Data projection; Initialisation; Neural networks; Principal component analysis (PCA); Sammon's mapping

1. INTRODUCTION

Sammon's nonlinear mapping [1] is a projection method for analysing multivariate data. The method attempts to preserve the inherent structure of the data when the patterns are projected from a higher-dimensional space to a lower-dimensional space by maintaining the distances between patterns under projection. Denote the distances between pattern X_i and pattern X_j in the input space and their projections Y_i and Y_j in the projected space as d_{ij}^* and d_{ij} , respectively. Employing Euclidean metric to measure distances, Sammon's mapping minimises the mapping error:

$$E = \frac{1}{\sum_{i < j} d_{ij}^*} \sum_{i < j} \frac{[d_{ij}^* - d_{ij}]^2}{d_{ij}^*} \quad (1)$$

where n is the number of patterns. The mapping attempts to fit n points in the lower-space, such that their interpoint distances approximate the corresponding distances in the higher-space.

Sammon's mapping has been designed and usually used to project high-dimensional data onto one to three dimen-

sions in order to analyse the data structure [1–4], or for classification based on two projections [3]. However, there is no obstacle that prevents extracting more than two or three projections, and hence the application of the mapping to feature extraction and classification. Recently, it has been suggested [5,6] to extract an *arbitrary* number of projections of Sammon's mapping and thereby exploit the 'classification potential' of the mapping. It was found that the classification accuracy based on Sammon's projections is comparable with, and in some cases even superior, to that based on other feature extractors [5–7]. This classification capability is utilised here to study initialisation aspects of the mapping. Usually, random initialisation is used [2], although initialisations based on PCs of the sample covariance matrix have also been suggested [1,3,5]. Nevertheless, to the best of our knowledge, these initialisations have never been extensively compared. Therefore, in the present paper, we compare experimentally random and PC-based initialisations of Sammon's mapping using two evaluation criteria: Sammon's mapping error and the classification accuracy based on an arbitrary number of projections. Section 2 introduces Sammon's algorithm and an implementation of the mapping. Section 3 describes experiments with five real-world databases to evaluate random and PC-based initialisations using the mapping error and classification accuracy based on the mapping. Finally, Section 4 concludes the study.

2. SAMMON'S NON-LINEAR MAPPING

2.1. Sammon's Algorithm

A general mapping f transforms a pattern X of a d -dimensional input space to a pattern Y of an m -dimensional projected space, $m < d$, i.e. $Y = f(X)$, such that a criterion J is optimised. The mapping f is determined from among all the transformations g , as the one that satisfies $J\{f(X)\} = \min_g J\{g(X)\}$. The mappings vary by the functional forms of f and the criteria they have to optimise. Although providing a very well established criterion (Eq. (1)), Sammon's algorithm does not provide an explicit mapping function, f ; hence, the projection of a new pattern requires re-execution of the algorithm to the 'new' data set.

Besides this lack of generalisation capability, Sammon's mapping has two other main drawbacks. The first drawback is the computational load of the mapping, which is $O(n^2)$. This means that $n(n-1)/2$ distances (as well as the error derivatives) must be calculated. The second drawback is that since the mapping employs steepest descent procedure to minimise the error, it is prone to be trapped in local minima, and hence a large number of simulations with random initialisations is required to yield satisfactory results. Several methods aim to overcome this problem by making use of some knowledge of the data. For example, using the first and second norms (the lengths in l_1 and l_2) of the patterns to initialise a mapping into two projections can be ten times faster than randomly initialised mapping [2]. Another initialisation that is not limited to two projections is based on mapping the data onto the space spanned by the PCs [1,3,5]. Nevertheless, in all the research [1-7], the choice of the initialisation method is arbitrarily made, and the mapping results are not compared thoroughly with those based on other initialisation methods. Such an evaluation however, is the subject of this paper.

2.2. A Neural Network Implementation of Sammon's Mapping

Mao and Jain [3] have suggested a neural network (NN) implementation of Sammon's mapping. The architecture they use (Fig. 1) is a two-layer perceptron where the number of input units is set to be the input space dimension, d , and the number of output units is specified as the projected space dimension, m . They derive a weight updating rule for the multilayer network that minimises the mapping error (Eq. (1)) based on gradient descent similar to the backpropagation (BP) learning rule. The general updating rule for all the hidden layers, $l = 1, L-1$ and the output layer ($l = L$) is

$$\Delta \omega_{jk}^{(l)} = -\eta \frac{\partial E}{\partial \omega_{jk}^{(l)}} = -\eta (\Delta_{jk}^{(l)}(X_s) y_j^{(l-1)}(X_s) - \Delta_{jk}^{(l)}(X_t) y_j^{(l-1)}(X_t)) \quad (2)$$

where $\omega_{jk}^{(l)}$ is the weight between unit j in layer $l-1$ and unit k in layer l , η is the learning rate, $y_j^{(l-1)}$ is the output of the j th unit in layer $l-1$ and X_s and X_t are a pair of

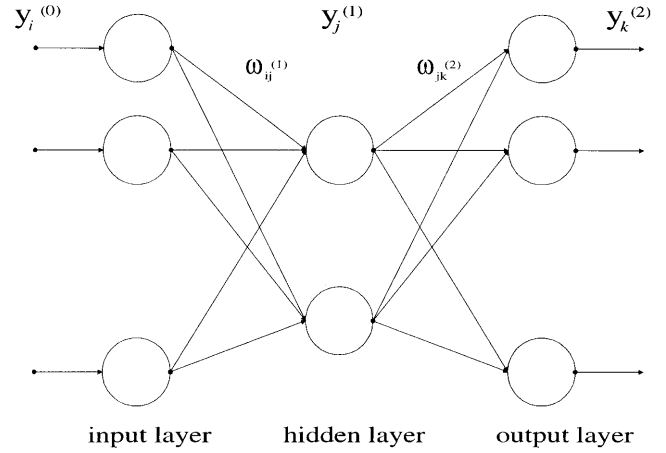


Fig. 1. A two-layer perceptron NN for the implementation of Sammon's mapping.

patterns. Both the errors $\Delta_{jk}^{(l)}$ and the network outputs $y_j^{(l-1)}$ are functions of the input patterns X_s and X_t . The errors $\Delta_{jk}^{(l)}$ are accumulated in each layer and backpropagated to a preceding layer, similar to the BP algorithm. Since the network error is a function of distances between projected patterns, the learning algorithm does not depend on category information (as in the BP algorithm), and thus can be considered as an extension of the BP learning rule to an unsupervised one [3]. The weights are updated in a BP manner following the presentation of each pair of randomly selected patterns, and similar to the BP algorithm, a momentum constant is frequently added to accelerate the convergence.

In Mao and Jain's NN implementation, the network is able to project new patterns after training, a property Sammon's algorithm does not have. They employ a two-stage training phase using the same configuration. In the first stage, they perform an initial mapping by employing the standard BP algorithm and the results of principal component analysis (PCA) to approximate PCA by the network. In the second stage, they use the trained network and their unsupervised BP algorithm to refine the first stage mapping. The initialisation and training of our NN-based Sammon's mapping implementation are different and simpler than those of Mao and Jain's implementation. For the initialisation, we use the eigenvectors of the sample covariance matrix and not the PCA projected patterns. The eigenvectors are exploited to establish the columns of the initial input-hidden weight matrix (ω) of the implementation, i.e. $\omega = [\varphi_1, \varphi_2, \dots, \varphi_m]$, where φ_i , $i = 1, m$ are the eigenvectors corresponding to the m largest eigenvalues. The initial hidden-output weight vectors are selected randomly. The network, initialised by these matrices, implements Sammon's mapping by performing a one-stage training phase using Mao and Jain's unsupervised BP algorithm.

There are several advantages for our implementation compared with that of Mao and Jain's. First, the long training of the first stage of Mao and Jain's implementation [3,4] is avoided as our implementation performs only a one-stage training phase similar to Mao and Jain's second training

stage. Secondly, although the initialisation is based on the PCs, and therefore exploits the advantages of PCA, the initial mapping is neither identical to that of PCA nor is it restricted to a linear projection. When patterns are projected using Mao and Jain's implementation, the maps and mapping errors are found similar to those of PCA [3,4]. These findings also suggest that after an initialisation using their first stage, the role of Mao and Jain's second stage (Sammon's mapping itself) is secondary. Moreover, the implementation of a linear mapping (such as the PCA) by a nonlinear model, as in the first stage of Mao and Jain's implementation, seems inefficient. Third, the configuration of our network can be determined before initialising the mapping, during the PCA. A requirement to preserve a desired variance of the data prescribes a desired number of (eigenvalues and thus) eigenvectors, and hence specifies the network configuration and the initial input-hidden weight matrix. This avoids the conventional trial-and-error experimentation to find an optimal configuration [3,4,6]. Finally, since the initial input-hidden weight matrix in our implementation is based on the eigenvectors, the number of hidden units is restricted to be lower than, or equal to, the number of input units. Usually, when applied to feature extraction of real-world applications, this poses no limitations, but it is a flaw of the method.

3. METHODOLOGY AND RESULTS

3.1. The Data Sets

To evaluate the two initialisations, we employ data sets extracted from five databases. The first two data sets were derived from chromosome images. In the first set, chromosome patterns are represented by 64 density profile (d.p.) features (integral intensities along sections perpendicular to the medial axis of the chromosome) [5]. In the second data set, patterns are represented by four geometrical features, i.e. length, perimeter, area and the centromeric index (the ratio of the short arm length to the total length) of the chromosome [7]. The third data set is extracted from a satellite image database [8]. Each pattern in the database corresponds to intensities measured in four spectral bands of a 3×3 neighbourhood of pixels of a sub-scene image, hence, consists of 36 features. The fourth data set is based on a Research Assessment Exercise (RAE) database of 72 subject areas in all higher education institutions in the UK. Variables such as the number of active researchers, postgraduate students and number of publications formed a 79-dimensional database that is used to assess research, on a scale of 1 to 5, in each subject area at each institute [9]. The last data set is the much analysed iris data [10], where the patterns are represented by four attributes (sepal and petal lengths and widths).

Since in three of the databases (chromosome (d.p.), chromosome (geometrical) and RAE) there are (around) 100 patterns in each class, we also extract for comparison one hundred patterns per class from the satellite data and use all the 50 patterns *per* class which are available in the iris

data. Moreover, since each of the chromosome databases and the iris data are of three classes, we use for a comparison, three classes of the satellite and RAE databases. These classes are chromosome types '13', '19' and 'x' in the first two data sets, three soil types in the third set, the subjects Physics, Chemistry and Biology in the fourth set, and three iris types in the last data set.

3.2. The Experiments

We have compared the random and PC-based initialisations by evaluating the extracted patterns using Sammon's mapping error and the two-layer perceptron probability of correct classification. The configuration of the NN implementation consists of 64, 4, 36, 79 or 4 input units for the five data sets, respectively. The numbers of output and hidden units are changed according to the experiment. In the first experiment, the network employs two hidden units to evaluate the mapping error when two and ten projections (outputs) are extracted. Two projections are appropriate for exploratory data projection and classification based on a very low-dimensional feature space, whereas ten projections are an example for classification based on a higher-dimensional feature space. In the second experiment, the same number of hidden units is used and the classification accuracy is measured based on one to ten projections for the chromosome (d.p.), satellite and RAE data sets and one to three projections for the chromosome (geometrical) and iris data sets. These ranges of projections are representative for a wide extent of requirements of classification performances and compression ratios. The number of hidden units in the first two experiments is set at two to avoid overtraining. In the third experiment, the network output is set at two and the dimension of the hidden layer is changed in the range 1–10 for the chromosome (d.p.), satellite and RAE data sets and in the range 1–3 for the chromosome (geometrical) and iris data sets.

Using the PC-based initialisation, eigenvectors corresponding to the largest eigenvalues, instead of random vectors, define the initial input-hidden weight matrix. The initial hidden-output weight matrix is selected randomly. Each simulation is repeated using four random initial hidden-output weight matrices for both the initialisations. In the case of the random initialisation, four random initial input-hidden weight matrices are used. The classification accuracy and mapping error are averaged over these simulations (four or sixteen, respectively). The mapping parameters are set according to Lerner et al. [5] to be a learning rate of 0.9 and a momentum constant of 0.5. Since data is limited, we average the results over 20 replications of random selections of the training set (using 90% of the patterns) and test set (10%). Therefore, the mapping error and classification accuracy reported here are averages over all possible combinations of 20 randomly chosen data sets, ten random classifiers (in the classification experiments) and sixteen or four random initialisations using the random or PC-based initialisations, respectively.

3.3. The Classifier

A two-layer perceptron trained by the standard BP learning algorithm is used as a classifier in the last two experiments. The number of input units of the classifier, m , is set by the projected space dimension, the number of output units is determined by the number of classes (three in all cases) and the number of hidden units is two. The classifier parameters are adopted from a previous investigation [5]: a learning rate of 0.1, a momentum constant of 0.95 and a training period of 500 epochs. Both the configuration and the parameters are checked to provide sufficient accuracy without overtraining. The calculation of the probability of correct classification is based on the maximum network output, which approximates the maximum *a posteriori* probability decision rule. This probability is averaged over ten simulations with randomly chosen initial weight matrices.

3.4. Results

Experiment 1: The Mapping Error. Figure 2 shows the evolution of Sammon's mapping error based on both initialisations for two and ten projections and the five data sets. An epoch represents presentation in random order of all the $n(n-1)/2$ possible pairs of patterns to the network once. The results demonstrate that the PC-based initialisation yields a lower ultimate error and a shorter training period (for a specific error) than the random initialisation, regardless of the projected space dimension or the database*. A comparable error to that achieved using the random initialisation after 70 epochs is reached using the PC-based initialisation (e.g. for $m = 2$) after only 66 (chromosome (d.p.)), 62 (RAE), 49 (satellite), 56 (chromosome (geometrical)) or 61 (iris) epochs. These training sessions are 5.7–30% shorter than those required by the random initialisation, the exact amount depending on the data. The standard deviation of the mapping error is between 0.001 and 0.01 for the different databases.

Experiment 2: Classification Accuracy for Various Numbers of Projections. The two-layer perceptron probabilities of correct classification of the test sets are measured for one to ten (chromosome (d.p.), satellite and RAE) and one to three (chromosome (geometrical) and iris) projections of Sammon's mapping. Figure 3 shows the superiority of the PC-based initialisation over the random one for every number of projections and each of the databases. Usually, and especially for the PC-based initialisation, a few projections are required by the classifier to achieve almost the maximum classification accuracy. The standard deviation of the classification accuracy in this experiment is between 2% and 4%, depending on the data.

Experiment 3: Classification Accuracy for Various Numbers of Hidden Units. Figure 4 depicts the probabilities of correct classification for two projections ($m = 2$) and various numbers of input-hidden weight vectors. The standard deviation of the classification accuracy is between 3.9% and 5.9%. Figure 4 reveals that only a few eigenvectors are needed to accomplish the highest classification accuracy based on the PC-based initialisation. As the number of hidden units increases, however, the benefit of using the PC-based initialisation is reduced until the two initialisations yield similar results. This is because using eigenvectors that relate to smaller eigenvalues adds only a small amount of uncorrelated information, and at the same time increases the problem complexity thus reducing the accuracy. However, projection of the input space using increasing numbers of random weight vectors extracts (not necessarily monotonically) increasing amounts of information thus improving the accuracy.

4. DISCUSSION

We have compared random and PC-based initialisations of an NN implementation of Sammon's mapping. Understanding the initialisation benefits is essential in designing applications that involve Sammon's mappings. The PC-based initialisation we suggest exploits the benefits of PCA while mapping the input space linearly on the hidden layer, but employing a random hidden-output weight matrix, it also extends the network initialisation to perform a nonlinear mapping.

We use Sammon's mapping error (a data projection 'tool') and the NN classification accuracy (a classification 'tool') to evaluate the initialisations. The PC-based initialisation has several advantages compared to the random initialisation: (a) fewer experiments are necessary. (b) The network configuration is set before the implementation; hence, there is no need for preliminary experimentation. Additional advantages of the PC-based initialisation that are found when experimenting with a few eigenvectors and five real-world databases are: (c) a shorter training period, (d) a lower mapping error and (e) a higher classification accuracy. These advantages are the result of extracting useful information about the input space using a few principal axes, which cannot be extracted using a similar number of random axes.

In summary, the NN implementation of Sammon's mapping that provides a generalisation capability, and the PC-based initialisation of the mapping that provides computational and performance advantages are attractive for both exploratory data visualisation and classification in real-world applications. Finally, it would be of interest to extend the study to applications of more than three classes and to other domains, and to compare theoretically the two initialisations.

Acknowledgement

Supported in part by the Paul Ivanier Center for Robotics and Production Management, Ben-Gurion University, Israel.

* The slightly different results on the RAE data are related to the fact that we obtained and experimented with only one replica of training and test sets for this data compared to 20 replicas for the other databases.

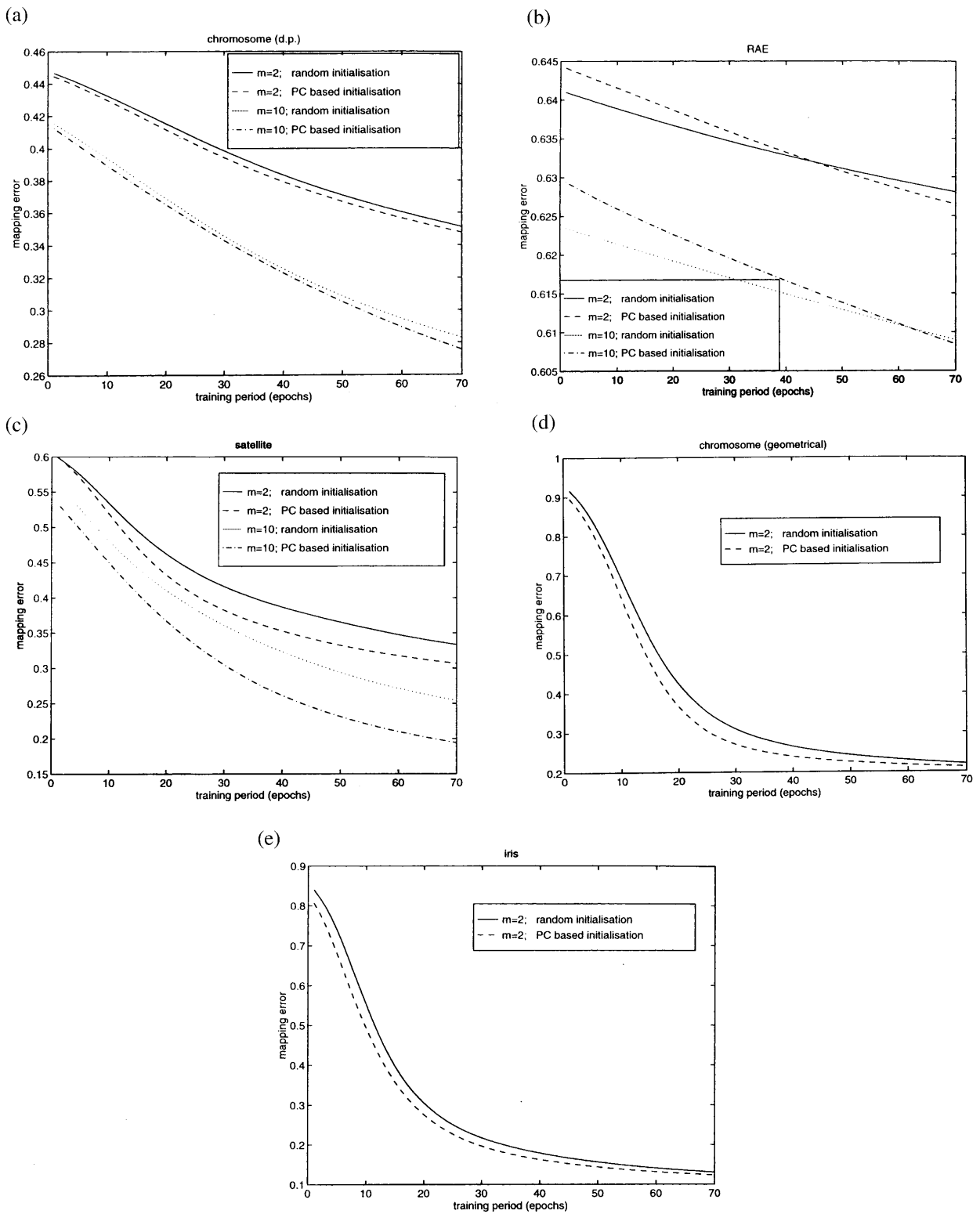


Fig. 2. The mapping error of Sammon's mapping based on two initialisation methods for increasing training periods. The error is plotted for the (a) chromosome (d.p.), (b) RAE, (c) satellite, (d) chromosome (geometrical) and (e) iris data sets and for 2 and 10 projections (a, b and c) or 2 projections (d and e).

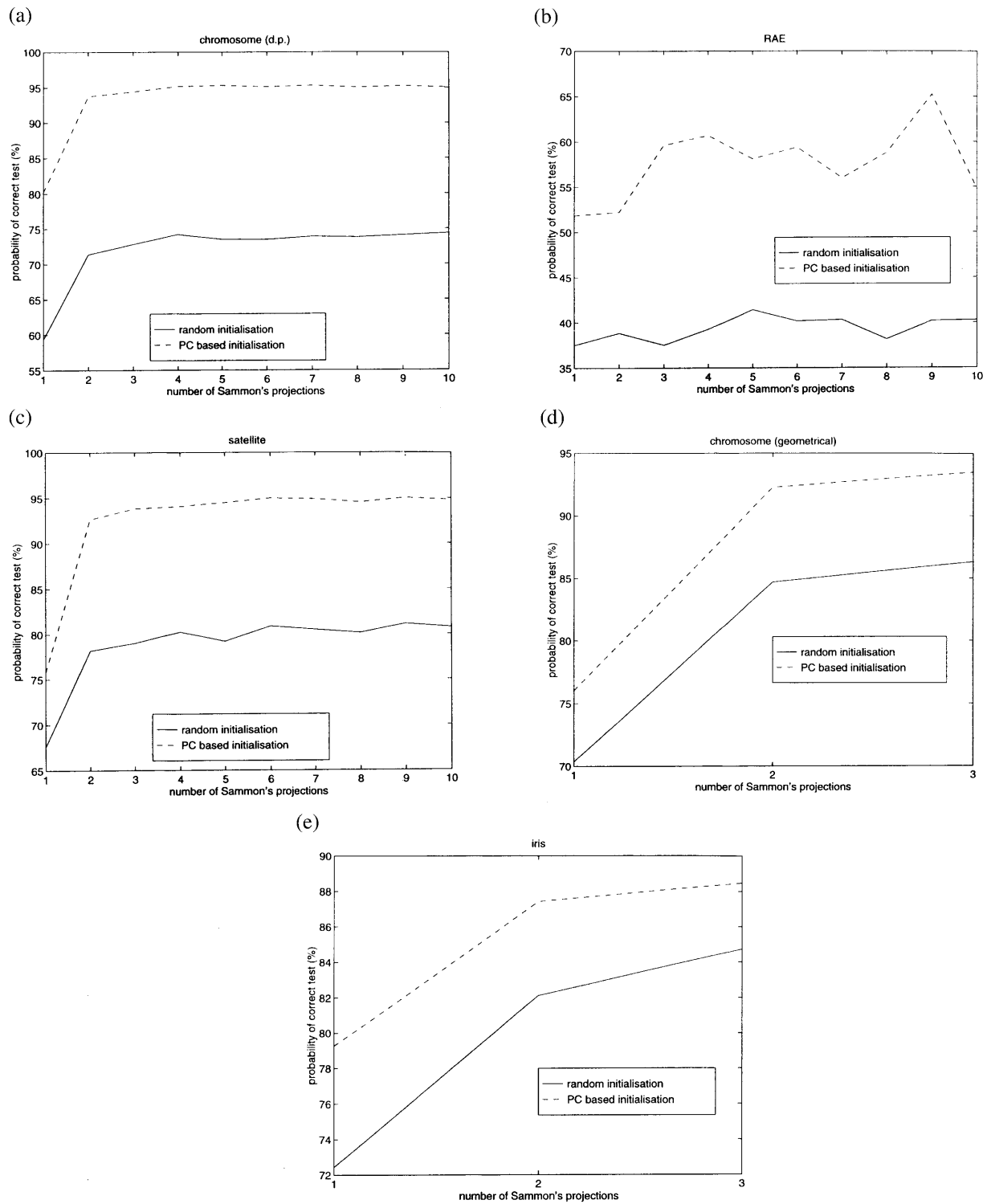


Fig. 3. The probability of correct classification of the test set based on Sammon's mapping for increasing numbers of projections and two initialisation methods for the (a) chromosome (d.p.), (b) RAE, (c) satellite, (d) chromosome (geometrical) and (e) iris data sets.

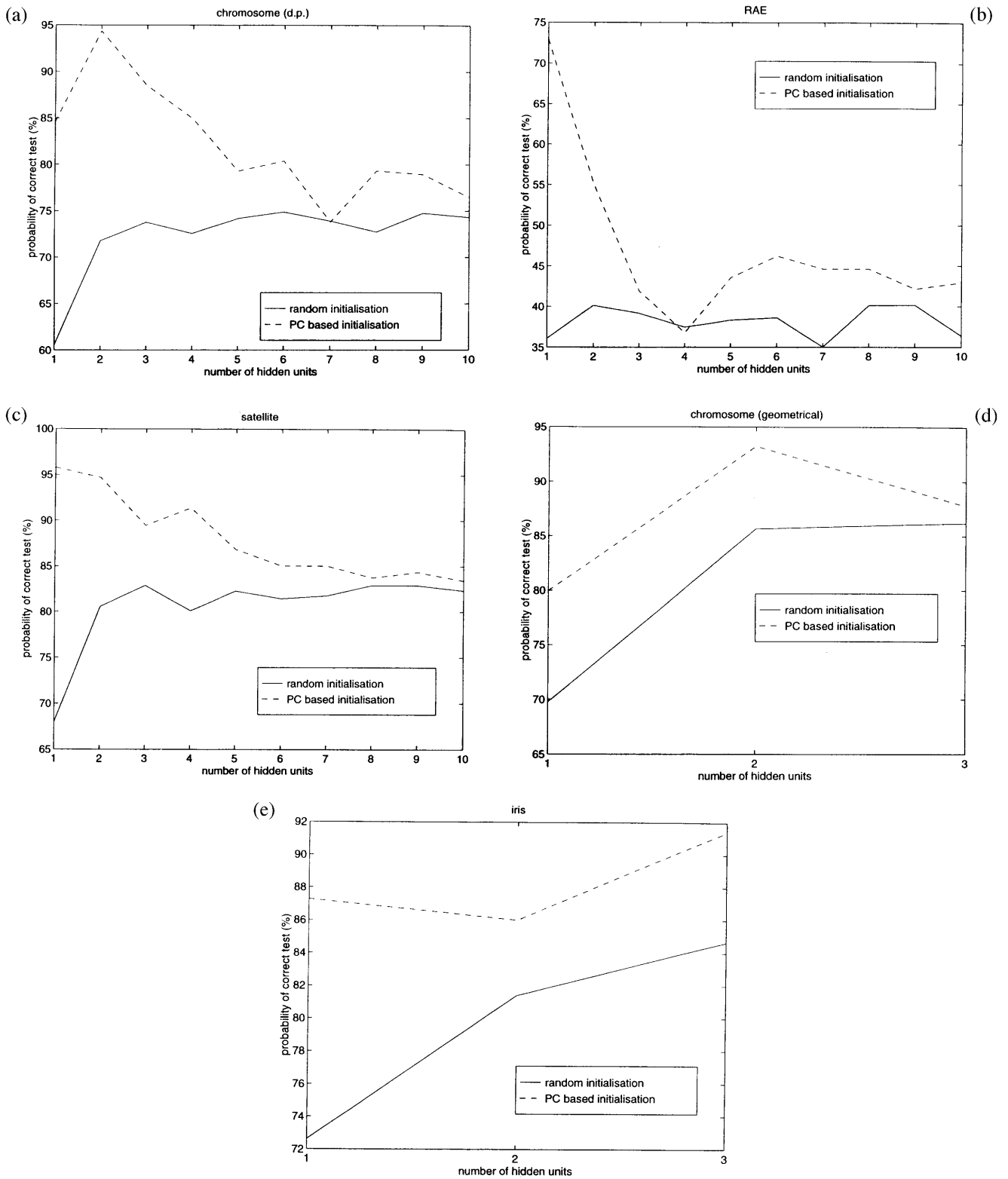


Fig. 4. The probability of correct classification of the test set based on Sammon's mapping for two projections ($m = 2$), two initialisation methods and different network configurations for the (a) chromosome (d.p.), (b) RAE, (c) satellite, (d) chromosome (geometrical) and (e) iris data sets.

References

1. Sammon JW Jr. A nonlinear mapping for data structure analysis. *IEEE Transactions on Computers* 1969; 18:401–409
2. Chien Y. *Interactive pattern recognition*. Marcel Dekker, New York, 1978
3. Mao J, Jain AK. Artificial neural networks for feature extraction and multivariate data projection. *IEEE Transactions on Neural Networks* 1995; 6:296–317
4. De Ridder D, Duin RPW. Sammon's mapping using neural networks : A comparison. *Pattern Recognition Letters* 1997; 18: 1307–1316
5. Lerner B, Guterman H, Aladjem M, Dinstein I, Romem Y. On pattern classification with Sammon's nonlinear mapping – an experimental study. *Pattern Recognition* 1998; 31:371–381
6. De Backer S, Naud A, Scheunders P. Non-linear dimensionality reduction techniques for unsupervised feature extraction. *Pattern Recognition Letters* 1998; 19:711–720
7. Lerner B, Guterman H, Aladjem M, Dinstein I. A comparative study of neural network based feature extraction paradigms. *Pattern Recognition Letters* 1999; 20:7–14
8. Michie D, Spiegelhalter DJ, Taylor CC. *Machine Learning, Neural and Statistical Classification*. Ellis Horwood, New York, 1994
9. Lowe D, Tipping ME. Feed-forward neural networks and topographic mappings for exploratory data analysis. *Neural Computing and Applications* 1996; 4:83–95
10. Merz CJ, Murphy PM. *UCI repository of machine learning databases*. Irvine, CA: University of California, Department of Information and Computer Science, 1998

Boaz Lerner is a Research Associate with the University of Cambridge Computer Laboratory, Cambridge, UK. Before joining the Computer Laboratory, he was a Research Fellow with the Neural Computing Research Group at Aston University, UK. He received his BA from the Hebrew University, Israel, in 1982, MSc from

Tel-Aviv University, Israel, in 1988 and PhD from Ben-Gurion University, Israel, in 1996. His research interests include neural networks and statistical pattern recognition for image classification and segmentation, Bayesian networks for image interpretation and learning probabilistic graphical models.

Hugo Guterman received a PhD degree in electrical and computer engineering from the Ben-Gurion University of the Negev, Israel in 1988. During the years 1988–90 he was a visiting researcher at the Center of Biotechnology at the MIT. He is currently a senior lecturer at the Department of Electrical and Computer Engineering at the Ben Gurion University of the Negev. His research interests include control theory, signal and image processing, biotechnology, neural networks and fuzzy logic.

Mayer Aladjem received an MSc and a PhD degree in electrical engineering from the Technical University, Sofia, Bulgaria in 1975 and 1980, respectively. He was an Associate Professor of Biomedical Cybernetics in the Central Laboratory on Bioinstrumentation and Automation of the Bulgarian Academy of Sciences (1988–1990). He is currently a senior lecturer in the Department of Electrical and Computer Engineering at Ben-Gurion University of the Negev, Israel. His research interest is in feature extraction and selection methods, interactive pattern recognition, systems for data analysis and their applications in biology and medicine.

Its'hak Dinstein received his PhD in electrical engineering from the University of Kansas, Lawrence Kansas, in 1974. He was with COMSAT Laboratories, Gaithersburg Maryland, from 1974 to 1977, when he returned to Israel and joined Ben-Gurion University, in Beer Sheva. He is now a full professor at the Electrical and Computer Engineering Department. During 1982–1984 he was a visiting scientist at IBM Research Laboratory, San Jose, California. During 1988–1990 he was a visiting associate professor at Polytechnic University, Brooklyn, New York. His research interests include image processing and computer vision.

Correspondence and offprint requests to: B. Lerner, University of Cambridge Computer Laboratory, New Museums Site, Cambridge CB2 3QG, UK.