ELSEVIER

Pattern Recognition Letters 19 (1998) 711–720

Pattern Recognition Letters

Non-linear dimensionality reduction techniques for unsupervised feature extraction [1]

S. De Backer, A. Naud, P. Scheunders *

Vision Lab, Department of Physics, University of Antwerp, Groenenborgerlaan 171, 2020 Antwerpen, Belgium

Received 17 July 1997; revised 30 January 1998

Abstract

Dimensionality reduction techniques have been regularly used for visualization of high-dimensional data sets. In this paper, reduction to $d \geqslant 2$ is studied, with the purpose of feature extraction. Four different non-linear techniques are studied: multidimensional scaling, Sammon's mapping, self-organizing maps and auto-associative feedforward networks. All four techniques will be presented in the same framework of optimization. A comparison with respect to feature extraction is made by evaluating the reduced feature sets ability to perform classification tasks. The experiments involve an artificial data set and grey-level and color texture data sets. We demonstrate the usefulness of non-linear techniques compared to linear feature extraction.

Keywords: Dimensionality reduction; Feature extraction; Self-organizing maps; Multidimensional scaling; Sammon's mapping; Auto-associative feedforward neural networks; Texture and color classification; Data projection

1. Introduction

In many classification problems, high-dimensional data are involved, because large feature vectors are generated to be able to describe complex objects and to distinguish between them. On the other hand, the amount of available data points is limited in many practical situations. For a classifier the estimation of the class probability distributions in these sparsely sampled high-dimensional data spaces is troublesome and generally affecting the liability of the obtained classification results.

To avoid these problems, the dimension of the feature space is reduced. This can be done in several ways. The easiest way is to select a limited set of features out of the total set (Devijver and Kittler, 1982). The classification performance serves as a measure for selecting the features. Some well-known feature selection techniques are forward selection and branch and bound techniques. Another way is feature extraction. Here, features are extracted as functions (linear or non-linear) of the original set of features. Unsupervised linear feature extraction techniques more or less all rely on Principal Component Analysis (PCA), which rotates the original feature space, before projecting the feature vectors onto a limited amount of axes. Supervised feature extraction techniques usually relate to the discriminant analysis

* Corresponding author.

[1] Electronic Annexes available. See http://www.elsevier.nl/locate/patrec.

0167-8655/98/$19.00
PII: S0167-8655(98)00049-X

technique (Fukunaga, 1990) which uses the within and between-class scatter matrices.

Already in the early days of pattern recognition several non-linear mapping techniques were developed. For example multidimensional scaling (Shepard, 1962; Kruskal, 1964) and Sammon's mapping (Sammon, 1969) are such techniques which, according to some predefined error criterion, try to map the original data space into a lower-dimensional space, hereby preserving as much as possible the local structure of the original space.

With the development of neural networks, new possibilities for non-linear mapping were created. Amongst them, Self-Organizing Maps are probably the most well known (Kohonen, 1995). Other ways include auto-associative feedforward networks (Baldi and Hornik, 1989) and a neural network version of Sammon's mapping (Mao and Jain, 1995). Although the mentioned techniques are theoretically capable of generating a non-linear mapping into a space with arbitrary dimension, most applications were mappings to $d = 2$, with the purpose of visualizing the data (some recent applications are found in (Kraaijveld et al., 1995; Mao and Jain, 1995)). Recently a supervised neural network approach was presented for feature extraction for classification purposes, and compared to PCA (Lee and Landgrebe, 1997).

In this paper, a study is performed on unsupervised non-linear dimensionality reduction. Four techniques are evaluated: a multidimensional scaling algorithm, Sammon's mapping technique, Kohonen's self-organizing map and an auto-associative feedforward neural network. First we will present these four algorithms within the same formalism, based on a minimization of an error function, which will be performed by gradient-descent or higher order optimization techniques. Secondly the performance of the techniques for feature extraction is evaluated and compared to linear mapping (PCA). The performance is evaluated by examining the mapped feature space's ability to perform supervised classification tasks. Apart from an artificial data set, real world applications are studied in the field of texture analysis where high-dimensional wavelet-based feature sets from grey-level and color texture images are used. We will show that non-linear feature extraction leads to feature sets which improve classification performance compared to linear mappings.

The outline of the paper is the following. In Section 2 the four non-linear techniques are presented, and an efficient optimization scheme is presented for each of them. The experiments are conducted and a discussion on the results is given in Section 3.

2. Mapping algorithms

Let us first fix the notations used in this section:

N: number of objects in the feature space,
D: dimension of the feature space (called input space),
$\boldsymbol{y}_i$: D-dimensional feature vector representing point i in the input space,
D_{ij}: Euclidean distance between points i and j in the input space,
d: dimension of the space of extracted features (called output space),
$\boldsymbol{x}_i$: d-dimensional vector representing point i in the output space,
d_{ij}: Euclidean distance between points i and j in the output space,
$\boldsymbol{X}$: vector in $(N \times d)$-dimensional space of the coordinates x_i^r:

$$(\boldsymbol{X})_k = x_i^r, \quad k = (i-1)d + r, \quad k = 1, \ldots, Nd,$$

$\boldsymbol{\nabla} f$: gradient vector of function $f(\boldsymbol{X})$, evaluated at $\boldsymbol{X}$:

$$(\boldsymbol{\nabla} f)_k = \frac{\partial f(\boldsymbol{X})}{\partial x_i^r}, \quad k = (i-1)d + r,$$

$$k = 1, \ldots, Nd,$$

H_f: Hessian matrix of function $f(\boldsymbol{X})$, evaluated at $\boldsymbol{X}$:

$$(H_f)_{kl} = \frac{\partial^2 f(\boldsymbol{X})}{\partial x_i^r \partial x_j^s}, \quad k = (i-1)d + r,$$

$$l = (j-1)d + s, \quad k, l = 1, \ldots, Nd.$$

The different methods presented below are based on the minimization of a multivariate function $f(\boldsymbol{X})$. This can be done iteratively by *descent methods*. The most common technique is gradient descent, in which the search is conducted towards the opposite of the gradient of $f(X)$. Refinements can be obtained by higher order approximations which involve the Hes-

sian of $f(X)$. For each of the four mapping techniques, one such refinement will be required.

2.1. Multidimensional scaling (MDS)

Multidimensional scaling covers a variety of multivariate data analysis techniques originally developed in mathematical psychology (Shepard, 1962; Kruskal, 1964). Nowadays, the term MDS is used in a broader sense for any method searching for a low (in particular 2) dimensional representation of objects given their high-dimensional representation (Cox and Cox, 1994; Ripley, 1996). The search is conducted such that the distances $\{d_{ij}\}$ between the representative points match as well as possible some given dissimilarities between the points in the original space. If the dissimilarities are proportional to distances, the ensuing method is called *metric* MDS. If the dissimilarities are assumed to be merely ordinal, it is *nonmetric* MDS. In the latter case the rank order of the distances $\{d_{ij}\}$ has to be as close as possible to the rank order of the dissimilarities.

In the following we will elaborate nonmetric MDS, in which the distances in the input space D_{ij} will serve as dissimilarities. A loss function is defined and minimized through a gradient descent procedure. A loss function similar to the stress S proposed by Kruskal (1964), is defined as

$$E_{\mathrm{MDS}} = \frac{\sum_{i<j}^{N}\left(d_{ij} - \hat{d}_{ij}\right)^2}{\sum_{i<j}^{N} \hat{d}_{ij}^2} \tag{1}$$

and expresses how well the order of the interpoint distances in the output space fits the order of the dissimilarities. The $\hat{d}_{ij}$ are pseudo-distances (target distances) derived from the d_{ij} with Kruskal's *monotone regression* procedure (Kruskal, 1964; Barlow et al., 1972). The $\hat{d}_{ij}$ are calculated in such a way that their rank order matches perfectly the rank order of the D_{ij} and they are as close as possible to the d_{ij}.

An outline of the nonmetric MDS algorithm is given below.

Nonmetric MDS algorithm.

1. Define an initial configuration $\boldsymbol{X}^{(0)}$. At iteration t:
2. compute the distances d_{ij} for the current configuration $\boldsymbol{X}^{(t)}$,
3. compute the target distances $\hat{d}_{ij}$,
4. compute $\boldsymbol{\nabla} E_{\mathrm{MDS}}^{(t)}$,
5. compute the learning rate $\alpha(t)$,
6. $\boldsymbol{X}^{(t+1)}$ is obtained by $\boldsymbol{X}^{(t+1)} = \boldsymbol{X}^{(t)} - \alpha(t)\nabla E_{\mathrm{MDS}}^{(t)}$,
7. if E_{MDS} has converged, then STOP else GOTO 2.

In the original algorithm, α is obtained from an ad hoc rule which works fairly well for small values of d (Kruskal, 1977). Since the algorithm becomes very sensitive to the value of α for larger values of d, we prefer to compute α optimally by a second-order approximation:

$$\alpha(t) = \frac{\|\boldsymbol{\nabla} E_{\mathrm{MDS}}^{(t)}\|^2}{\left(\boldsymbol{\nabla} E_{\mathrm{MDS}}^{(t)}\right)^{\mathrm{T}} \cdot H_{E_{\mathrm{MDS}}}^{(t)} \cdot \boldsymbol{\nabla} E_{\mathrm{MDS}}^{(t)}}$$

(Duda and Hart, 1973). $H_{E_{\mathrm{MDS}}}$ is symmetric, so that the computation of the denominator can be done efficiently. As

$$\binom{N}{2} = \frac{N(N-1)}{2}$$

distances d_{ij} have to be computed at each iteration, the complexity of step 2 scales with N^2. Computation of the learning rate α in step 5 scales with d^2N^2. Remark that the algorithm's complexity does not depend on D.

2.2. Sammon's mapping (SAM)

Sammon introduced a non-linear mapping technique that has been more widely known and applied in the pattern recognition society than the above MDS techniques (Sammon, 1969). Sammon's algorithm permits a mapping to a d-dimensional space through a minimization of the following error function:

$$E_{\mathrm{SAM}} = \frac{1}{\sum_{i<j}^{N} D_{ij}} \sum_{i<j}^{N} \frac{\left(d_{ij} - D_{ij}\right)^2}{D_{ij}}, \tag{2}$$

which expresses how well the distances in the output space fit the distances in the input space, giving more weight to the small distances. Minimization of E_{SAM} is performed by using gradient descent. In this case, we observed an even higher sensitivity to the

learning rate than in the case of MDS. Therefore, we apply exactly the same minimization strategy as is done in MDS (steps 4–6). Here again the algorithm scales with d^2N^2. It has been pointed out that minimizing (2) is strongly related to the metric MDS procedure (Kruskal, 1971).

2.3. Self-Organizing Map (SOM)

The SOM learning rule was proposed by Kohonen (1990, 1995) to build topology preserving mappings. The output space (i.e., the mapped space) is defined as a regular d-dimensional lattice. Each lattice point represents a neuron. For each neuron k, a D-dimensional weight vector $\boldsymbol{W}_k$ is defined. The weights represent the neurons position in the input space. Mapping is performed by assigning each data point $\boldsymbol{y}_i$ in the input space to one of these neurons, namely the one whose weight vector is closest to the point. The position vector $\boldsymbol{x}_i$ in the output space is then given by the lattice position of this neuron.

The error function looks as follows:

$$E_{\mathrm{SOM}} = \sum_k \sum_{i \in S_k} (\boldsymbol{W}_k - \boldsymbol{y}_i)^2, \tag{3}$$

with S_k being the set of data points which have neuron k as closest neuron. E_{SOM} expresses the average squared distance from a point to its representative. Minimization of E_{SOM} is now performed with respect to the weight vectors $\boldsymbol{W}_k$. The gradient descent approach leads to the following updating rule:

$$\boldsymbol{W}_k^{(t+1)} = \boldsymbol{W}_k^{(t)} - \alpha^{(t)} \cdot \left(\boldsymbol{\nabla} E_{\mathrm{SOM}}^{(t)}\right)_k. \tag{4}$$

This algorithm clusters the input space, but is not useful for mapping since the learning rate does not depend on the output space. For this reason the learning rate is replaced by a neighborhood function h_c which explicitly depends on the mapped space,

$$h_c^{(t)} = \begin{cases} \alpha^{(t)} & \text{if } k \in N_c^{(t)}, \\ 0 & \text{if } k \notin N_c^{(t)}, \end{cases} \tag{5}$$

where $N_c^{(t)}$ is the set of all neurons within a certain range of the winning neuron c (i.e., the nearest element to the presented data point) in the output space. During training this range and $\alpha^{(t)}$ are decreased monotonically. Since neighbouring neurons in the output space will be neighbours in the input space, the mapping preserves topology.

In contrast to the previous algorithms, SOM scales linearly with N and with D. SOM also scales linearly with W, the total number of neurons. When maintaining an equal sampling of the output space when d increases, W scales $\mathrm{O}(\exp d)$ so that the total complexity of SOM becomes $\mathrm{O}(DN \exp d)$.

2.4. Auto-associative Feedforward Neural Networks (AFN)

In an AFN the goal is to reproduce the feature space at the output layer while obtaining a reduced representation at the hidden layer (Baldi and Hornik, 1989; Kramer, 1991). The error function is given by

$$E_{\mathrm{AFN}} = \sum_i \left(\boldsymbol{y}_i - g(f(\boldsymbol{y}_i))\right)^2, \tag{6}$$

where f and g are the outputs of layer 2 and 3, respectively, and are generally non-linear functions. Since our aim is dimensionality reduction, the hidden layer will be representing the mapped space, the number of hidden neurons being d, and $f(\boldsymbol{y}_i) = \boldsymbol{x}_i$, being the mapping. For obtaining good non-linear representations, a 5-layer network is constructed, 2 layers to generate f, and 2 for g.

The gradient descent learning rule to minimize E_{AFN} is given by

$$\boldsymbol{W}^{(t+1)} = \boldsymbol{W}^{(t)} - \alpha \boldsymbol{\nabla} E_{\mathrm{AFN}}^{(t)}, \tag{7}$$

where $\boldsymbol{W}$ is the vector of the weights of all neurons. The gradient can be found efficiently by the well-known backpropagation rule.

Again, Eq. (7) is found to be very sensitive to the value of α. A second-order expression of α as in the case of MDS and SAM is difficult to obtain because of the backpropagation steps involved. Therefore a quasi-Newton technique is used instead. Here,

$$\boldsymbol{W}^{(t+1)} = \boldsymbol{W}^{(t)} - \alpha^{(t)} \left(H_{E_{\mathrm{AFN}}}^{(t)}\right)^{-1} \boldsymbol{\nabla} E_{\mathrm{AFN}}^{(t)}. \tag{8}$$

The inverse Hessian is estimated at each iteration by a limited memory version of the *Broyden–Fletcher–Goldfarb–Shanno* (BFGS) procedure (Liu and Nocedal, 1989). The learning rate $\alpha^{(t)}$ is updated to satisfy some optimality constraints.

Since the total number of weights W is linear in D and d, the complexity of AFN scales linearly with d. The total complexity is O(DdN).

3. Experiments and discussion

In the previous section, the four non-linear mapping techniques were described as minimizations of an error function. All four error functions aim at preserving topology of the original space into the map in one way or another. Instead of trying to analyse the behaviour of the mappings directly from the form of the error function, we chose to use the maps for a specific task, and compare the behaviour of the mappings by the performances of the task. The topology preserving properties of the corresponding error function is then reflected by the performance of a technique for that specific task. Since feature extraction is an important motivation for this study, the performance of a mapping technique will be measured by the quality of the obtained reduced feature set. That quality will be evaluated by performing classification tasks.

The experiments involve supervised classification, in which a D-dimensional data space is subdivided into a given number of classes, given a training set of labeled feature vectors. The classifier is a k-NN classifier ($k = 5$), applied together with the leave-one-out technique. The original feature space is first mapped onto a d-dimensional space ($d < D$), after which classification is performed on the mapped space. The classification performances for different values of d and for different mappings are compared.

An important property of a data set will be its intrinsic dimensionality. We define the intrinsic dimensionality as the minimum number of features needed to obtain a similar classification performance as by using the total number of features. Of course this number will depend on the feature extraction technique used. We have estimated the intrinsic dimensionality of a data set in two independent ways. First a suboptimal supervised feature selection is used (Pudil et al., 1994). Here, an optimal set of features is selected from the complete set, where the classification performance of the set is the optimality criterion. The second technique is the calculation of the fractal dimension of the complete data set (Sarkal and Chaudhuri, 1992).

The following three data sets are used:

1. An artificial data set, generated by defining $D + 1$ D-dimensional data points in a simplex configuration (all interpoint distances are equal to 1). We have chosen $D = 10$. Around these points, 10-dimensional multinormal distributions are generated, using diagonal covariance matrices, with variance = 0.0625. For each distribution, 100 points are generated ($N = 1100$). Each distribution forms a class (total of 11 classes). The variance is chosen so that the Bayes error-rate (i.e., the inter-class overlap) is about 20%. For this data set all features are equally important. The intrinsic dimensionality of this data set is 10 by construction. The k-NN classifier on the complete space leads to a classification performance of 81%.
2. A grey-level texture data base of $N = 245$ images containing 5 different Brodatz textures (5 classes). A continuous wavelet transform is performed and rotation-invariant features are extracted (Vautrot et al., 1997). A 39-dimensional data space is obtained. The intrinsic dimensionality of this data set is estimated to be about 3. Classification performance on this space is 90%.
3. A colour texture data base of $N = 1024$ images containing 16 different textures (16 classes). The feature vectors are generated by employing a discrete wavelet transform on the R, G and B plane separately and calculating the energy features of the detail images (Van de Wouwer et al., 1997). A 48-dimensional data space is obtained, with $N = 1024$. The intrinsic dimensionality is estimated to be about 5. The classification performance is 93.5%.

In the conducted experiments, the 4 mapping techniques are compared to linear dimensionality reduction. For this purpose, a Principal Component Analysis (PCA) is performed by computing eigenvectors of the tridiagonalized correlation matrix (Press et al., 1992). The 4 proposed non-linear algorithms are written in C and implemented on HP/9000 series workstations. The BFGS optimization algorithm is taken from http://www.netlib.org/opt/lbfgs_um.shar.

In the case of MDS and SAM, the initial configuration of data points in the mapped space is obtained by the results of PCA. In both cases, the number of iterations was set to 100. In the case of SOM and AFN, the initial configuration of weights is randomly distributed.

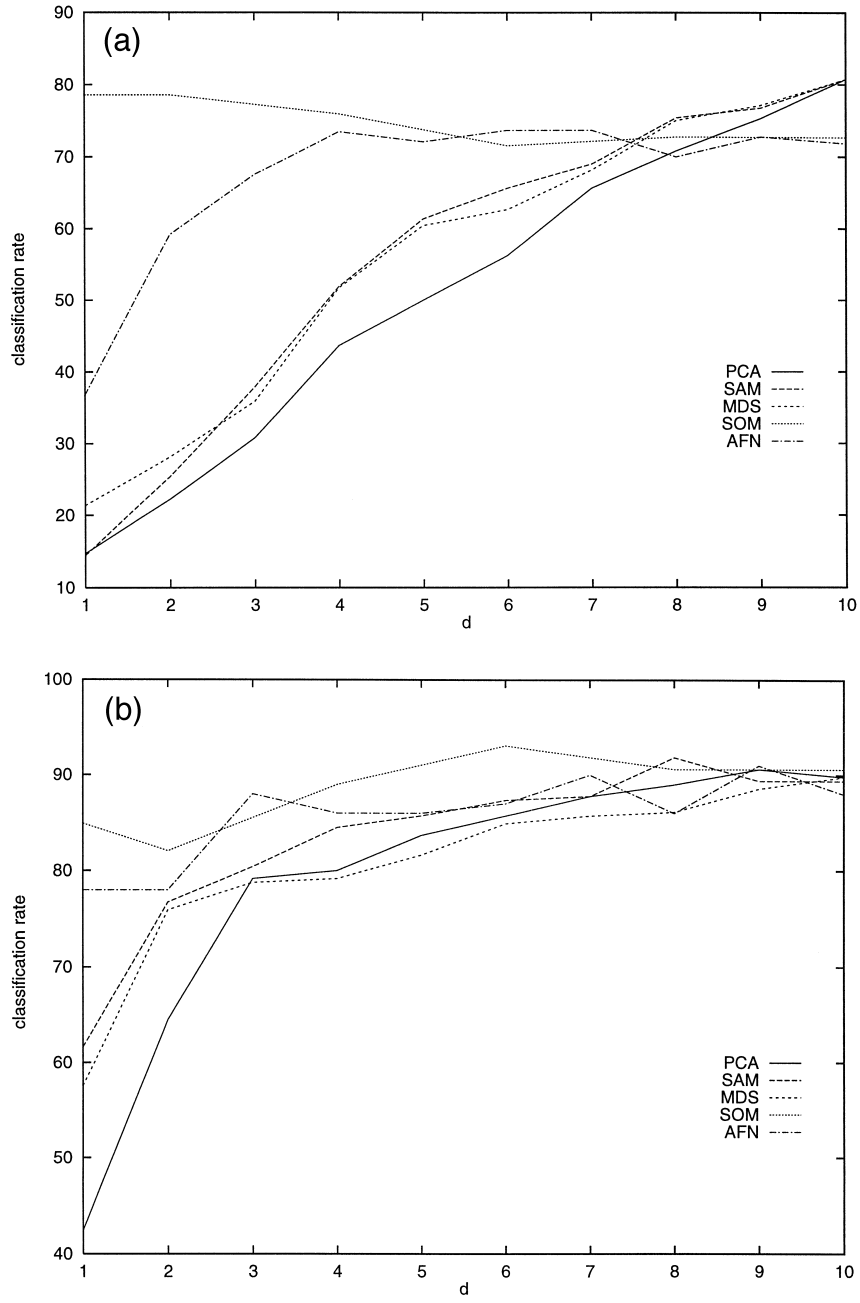

Fig. 1. Classification performance in function of d (in percentage) for PCA, MDS, SAM, SOM and AFN; (a) for data set 1; (b) for data set 2; (c) for data set 3.

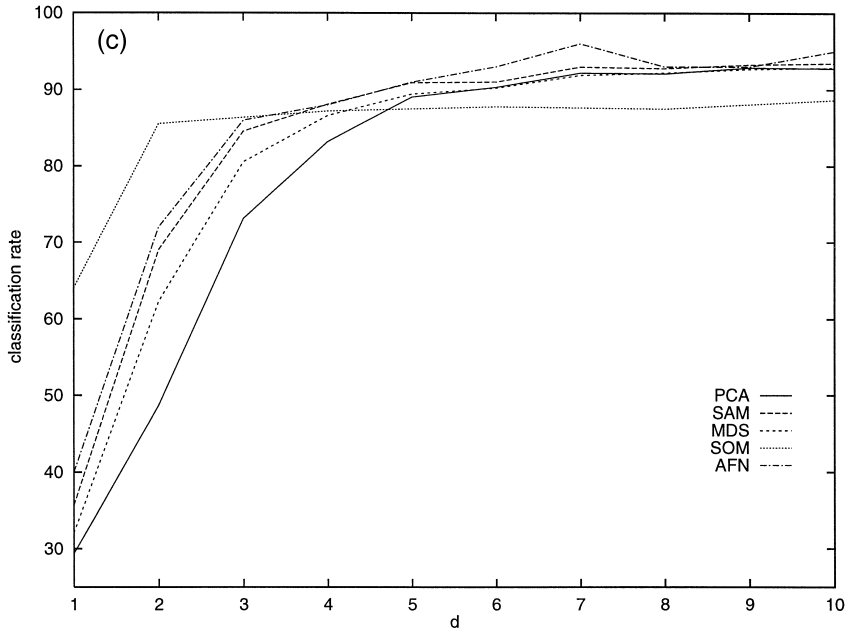

Fig. 1 (continued).

The SOM algorithm is very time consuming for large values of d, and becomes impractical when $d > 4$. Therefore the following approximative strategy is applied. The feature set is subdivided in $d/2$ smaller sets. Each set forms a lower-dimensional feature space, which is then reduced to a 2-dimensional space using SOM. The total reduced space is then obtained by gathering the $d/2$ 2-dimensional reduced spaces. The number of iterations is chosen to be $100N$ for each data set. The number of neurons per reduced dimension is 10 for the first and second data sets and 50 for the third one. Remark that the total number of neurons is higher than the number of data points. This means that after mapping not all neurons will be occupied by a data point, and that it is unlikely that two data points are mapped onto the same neuron. When applying the 5-NN classifier, the 5 nearest non-empty neurons are investigated.

In AFN the total number of weights is of importance. The number of neurons is D in the input and output layer, and d in the middle hidden layer. In the other 2 hidden layers it is chosen to be $x = 20$ for data set 1, $x = 30$ for data set 2 and $x = 40$ for data set 3. The total number of neurons then becomes $2D + d + 2x$. Weights are defined between subsequent layers only. An extra neuron is defined between all input neurons and all neurons of the middle layer and also between all neurons of the middle layer and all output neurons. The total number of weights $W = 2Dx + 2dx + 2Dd$. The algorithm is observed to converge slowly, the total number of iterations being of the order of $1000N$.

Fig. 1(a) (respectively (b) and (c)) displays the classification performances in function of d for data set 1 (respectively 2 and 3). From these curves we can deduct the following:

- Classification performance using PCA increases almost linearly with d until the intrinsic dimensionality is obtained. From then on it increases slowly to an optimal value.
- Classification performance of all 4 non-linear techniques starts off from a higher value at $d = 1$, increases faster than PCA for d smaller than the intrinsic dimensionality and reaches slowly its optimal value thereafter. Both MDS and SAM perform equally well, SAM yielding slightly better results than MDS. AFN performs even better and SOM performs the best, especially for the

first data set, where optimal results are already obtained from $d = 1$.

It is clear from these figures that improvement can be obtained by applying non-linear instead of linear feature extraction. Especially for small values of d a substantial gain in the classification performance is obtained.

In Table 1, the classification rates are given in detail for all techniques for $d \leqslant 4$. Remark that this time SOM is applied without approximation. To keep the total number of neurons constant, the number of neurons per reduced dimension is chosen to be 2500 in $d = 1$, 50 in $d = 2$, 15 in $d = 3$ and 7 in $d = 4$. From this table, the superior behaviour of SOM becomes even more clear.

The linear behaviour of the classification performance of PCA can intuitively be related to the linear character of the technique. It is clear that by performing non-linear feature extraction, a better characterization of the problem may be enclosed in a smaller number of features. From these experiments, one can observe that this is indeed the case. The different behaviour of the four techniques can be explained by the four different objective functions to be optimized. In the case of MDS and SAM, the objective is to preserve interpoint distances during mapping.

Table 1
Classification performance of PCA, MDS, SAM, SOM and AFN for $d = 1,2,3,4$

d	PCA	MDS	SAM	SOM	AFN
Data set 1					
1	15	21	18	82	37
2	22	28	24	80	60
3	30	36	39	79	68
4	43	52	53	76	73
Data set 2					
1	42	60	58	85	78
2	62	75	78	88	78
3	83	82	81	88	88
4	89	83	84	87	86
Data set 3					
1	29	32	36	74	40
2	50	62	69	88	72
3	72	81	85	91	86
4	83	87	88	90	88

Table 2
The CPU-times in minutes on an HP9000/712 workstation for the experiments of Table 1

d	PCA	MDS	SAM	SOM	AFN
Data set 1					
1	–	14	5.4	17	75
2	–	19	15	13	120
3	–	25	26	15	135
4	–	30	34	10	150
Data set 2					
1	–	0.5	0.1	13	76
2	–	0.7	0.5	9	100
3	–	0.9	0.7	10	120
4	–	1.1	1	6	130
Data set 3					
1	–	7.8	1.3	67	611
2	–	11	7.5	33	790
3	–	13	15	31	840
4	–	21	27	18	893

This means that short-range as well as long-range distances will be preserved. Optimal classification is obtained by mostly preserving the first ones. The fact that SAM is slightly better than MDS can be explained by the extra weighting of smaller distances in its error function. The objective function of SOM is more appropriate for the problem of classification. In this case, distances between a point and its local environment are preserved. This cluster property makes that SOM is more suited for classification purposes, and can explain the excellent performance of the technique for very low dimensions. Finally, by construction, AFN also tends to preserve distances of neighbouring points, which makes also this technique to be a good candidate for classification.

To give an idea about the time complexity of the techniques in real problems, the CPU-times are given in Table 2 (in minutes on an HP9000/712 workstation) for the runs of Table 1. No results are given for PCA, because run times were much smaller than one minute. When comparing these values to the complexities described in the previous section, the dependence on N can be discussed by looking at results on data set 2, compared to data sets 1 and 3 (4 times smaller N). The dependence on d can be discussed for small d only, while the dependence on D can be discussed by comparing sets 2 and 3 to set

1 (factor of 4 and 5 larger D). The following conclusions can be drawn:

- MDS scales with N^2 as expected, but with d rather than d^2 for small values of d. For higher values of d (see the experiment of Fig. 1) we observed it to scale with d^2. No dependence on D can be observed.
- The same holds for SAM.
- SOM scales linearly with N. Since the total number of neurons is kept more or less constant for different values of d, CPU times should be constant. This cannot clearly be observed from the table, probably due to the dependence of N_c on d. The complexity seems to grow less than linearly with D.
- AFN can be observed to scale linearly with D, d and N. One iteration takes about the same time as an iteration of SOM. Due to the number of iterations (10 times larger than in the case of SOM) before convergence, the CPU-times are excessively high.

It follows that large scale classification tasks cannot be performed using any of the proposed techniques. While SOM can only be applied efficiently on problems with low values of d (for higher values of d the approximative strategy can be applied), MDS and SAM are only practically useful for problems with low values of N. AFN is linear in d and N but slow convergence seems to be the problem there.

4. Conclusion

In this paper, a study is performed on unsupervised non-linear feature extraction. Four techniques were studied: a multidimensional scaling approach (MDS), Sammon's mapping (SAM), Kohonen's self-organizing map (SOM) and an auto-associative feedforward neural network (AFN). All four yield better classification results than the optimal linear approach (PCA), and therefore can be utilized as a feature extraction step in a design for classification schemes. Because of the nature of the techniques, SOM and AFN perform better for very low dimensions. Because of the complexity of the techniques, MDS and SAM are most suited for high-dimensional data sets with a limited number of data points, while SOM and AFN are more appropriate for low-dimensional problems with a large number of data points.

Acknowledgements

The first author is granted by the Flemish Institute for the promotion of Scientific and Technological Research for the Industry (IWT).

References

Baldi, B., Hornik, K., 1989. Neural networks and principal component analysis: Learning from examples without local minima. Neural Networks 2, 53–58.

Barlow, R.E., Bartholomew, D.J., Bremer, J.M., 1972. Statistical Inference under Order Restrictions: The Theory and Application of Isotonic Regression. Wiley, London.

Cox, T.F., Cox, M.A.A., 1994. Multidimensional Scaling. Chapman and Hall, London.

Devijver, P.J., Kittler, J., 1982. Pattern Recognition: A Statistical Approach. Prentice-Hall, Englewood Cliffs, NJ.

Duda, R.O., Hart, P.E., 1973. Pattern Classification and Scene Analysis. Wiley, New York.

Fukunaga, K., 1990. Introduction to Statistical Pattern Recognition, 2nd Edition. Academic Press, London.

Kohonen, T., 1990. The Self-Organizing Map. Proc. IEEE 78, 1464–1480.

Kohonen, T., 1995. Self-Organizing Maps. Springer, Berlin.

Kraaijveld, M.A., Mao, J., Jain, A.K., 1995. A nonlinear projection method based on Kohonen's topology preserving maps. IEEE Trans. Neural Networks 6 (3), 548–559.

Kramer, M.A., 1991. Nonlinear principal component analysis using auto-associative neural networks. AIChE J. 37 (2), 233–243.

Kruskal, J.B., 1964. Nonmetric multidimensional scaling: A numerical method. Psychometrika 29, 115–129.

Kruskal, J.B., 1971. Comments on ''a nonlinear mapping for data structure analysis''. IEEE Trans. Comput., p. 1614.

Kruskal, J.B., 1977. Multidimensional scaling and other methods for discovery structure. In: Enslein, K., Ralston, A., Wilf, H. (Eds.), Statistical Methods for Digital Computers, Vol. III. Wiley, New York, pp. 296–339.

Lee, C., Landgrebe, D., 1997. Decision boundary feature extraction for neural networks. IEEE Trans. Neural Networks 8 (1), 75–83.

Liu, D.C., Nocedal, J., 1989. On the limited memory BFGS method for large scale optimization. Math. Programming 45, 503–528.

Mao, J., Jain, A.K., 1995. Artificial neural networks for feature extraction and multivariate data projection. IEEE Trans. Neural Networks 6 (2), 296–317.

Press, W.H., Teukolsky, S.A., Vetterling, W.T., Flannery, B.P., 1992. Numerical Recipes in C. Cambridge University Press, Cambridge, MA, Section 11.2.

Pudil, P., Novovičová, J., Kittler, J., 1994. Floating search meth-

ods in feature selection. Pattern Recognition Letters 15, 1119–1125.

Ripley, B.D., 1996. Pattern Recognition and Neural Networks. Cambridge University Press, Cambridge, MA, Section 9.2.

Sammon, J.W., 1969. A nonlinear mapping for data analysis. IEEE Trans. Comput. 18, 401–409.

Sarkal, N., Chaudhuri, B., 1992. An efficient approach to estimate fractal dimension of textural images. Pattern Recognition 25, 1035–1041.

Shepard, R.N., 1962. The analysis of proximities: Multidimensional scaling with an unknown distance function. Psychometrika 27 (2), 125–140.

Van de Wouwer, G., Scheunders, P., Livens, S., Van Dyck, D., 1997. Color texture classification by wavelet energy-correlation signatures. Pattern Recognition, to appear.

Vautrot, P., Van de Wouwer, G., Scheunders, P., Livens, S., Van Dyck, D., 1997. Continuous wavelets for rotation-invariant texture classification and segmentation. Unpublished.