CLASSIFICATION AND ICA USING MAXIMUM LIKELIHOOD HEBBIAN LEARNING

Emilio Corchado[1], Jos Koetsier[1] ,Donald MacDonald[1] and Colin Fyfe[1]

[1] Applied Computational Intelligence Research Unit
The University of Paisley
Scotland
emilio.corchado, jos.koetsier, donald.mcdonald, colin.fyfe@paisley.ac.uk

Abstract. In this paper we investigate an extension of Hebbian learning in a Principal Component Analysis network which has been derived to be optimal for a specific probability density function(pdf). We note that this probability density function is one of a family of pdfs and investigate the learning rules formed in order to be optimal for several members of this family. We show that, whereas previous authors [6] have viewed the single member of the family as an extension of PCA, it is more appropriate to view the whole family of learning rules as methods of performing Exploratory Projection Pursuit(EPP). We explore the performance of our method first in response to an artificial data type, then to a real data set.

INTRODUCTION

Principal Component Analysis (PCA) is a standard statistical technique for compressing data; it can be shown to give the best linear compression of the data in terms of least mean square error. There are several artificial neural networks which have been shown to perform PCA e.g. [11, 12]. We shall be most interested in a negative feedback implementation [4].

The basic PCA network [4] is described by equations (1)-(3). Let us have an N-dimensional input vector at time t, x(t), and an M-dimensional output vector, y, with W_{ij} being the weight linking input *j* to output *i*. η is a learning rate. Then the activation passing and learning is described by

(1) $y_i = \sum_{j=1}^{N} W_{ij} x_j \; , \forall i$

(2) $e_j = x_j - \sum_{i=1}^{M} W_{ij} y_i$

0-7803-7616-1/02/$17.00 (c) 2002 IEEE

(3) $\Delta W_{ij} = \eta e_j y_i$

The weights converge to the Principal Component directions.

Exploratory Projection Pursuit (EPP) is a more recent statistical method aimed at solving the difficult problem of identifying structure in high dimensional data. It does this by projecting the data onto a low dimensional subspace in which we search for its structure by eye. However not all projections will reveal the data's structure equally well. We therefore define an index that measures how "interesting" a given projection is, and then represent the data in terms of projections that maximise that index. Now "interesting" structure is usually defined with respect to the fact that most projections of high-dimensional data onto arbitrary lines through most multi-dimensional data give almost Gaussian distributions [3]. Therefore if we wish to identify "interesting" features in data, we should look for those directions onto which the data-projections are as far from the Gaussian as possible.

We have previously implemented EPP using an artificial neural network [5]; the method is essentially a non-linear modification of the negative feedback network. The network can be described by the following set of equations

(4)
$$s_i = \sum_{j=1}^{N} W_{ij} x_j$$
$$e_j = x_j - \sum_{k=1}^{M} W_{kj} s_k$$
$$r_i = f(s_i)$$
$$\Delta W_{ij} = \eta r_i e_j$$

where x_j is the sphered activation of the j^th^ input neuron, s_i is the activation of the i^th^ output neuron , W_{ij} is the weight between these two and r_i is the value of the function f() on the i^th^ output neuron.

It was shown in [8] that the use of a (non-linear) function f() in equation (7) creates an algorithm to find those values of W which maximise that function whose derivative is f() under the constraint that W is an orthonormal matrix. This was applied in [5] to the above network in the context of the network performing an Exploratory Projection Pursuit. Thus if we wish to find a direction which maximises the kurtosis of the distribution which is measured by s^4, we will use a function $f(s) \approx s^3$ in the algorithm. If we wish to find that direction with maximum skewness, we use a function $f(s) \approx s^2$ in the algorithm.

In this paper, we derive a new neural method of performing Exploratory Projection Pursuit from a probabilistic perspective.

It has been shown [15] that the learning rule

$$(5)\quad \Delta W_{ij} = \eta\left(x_j y_i - y_i \sum_k W_{kj} y_k \right)$$

can be derived as an approximation to the best linear compression of the data. Thus we may start with a cost function

$$(6)\quad J(W) = 1^T E\left\{(\mathbf{x} - W\mathbf{y})^2\right\}$$

which we minimise to get the rule(5). [6] used the residual in (6) to define a cost function of the residual

$$(7)\quad J = f_1(\mathbf{e}) = f_1(\mathbf{x} - W\mathbf{y})$$

where $f_1 = \|.\|^2$ is the (squared) Euclidean norm in the standard PCA rule.

We may show [2] that the minimization of J is equivalent to minimizing the negative log probability of the residual, $\mathbf{e}$, if **e** is Gaussian. Let:

$$(8)\quad p(\mathbf{e}) = \frac{1}{Z}\exp(-\mathbf{e}^2)$$

Then we can denote a general cost function associated with this network as

$$(9)\quad J = -\log p(\mathbf{e}) = (\mathbf{e})^2 + K$$

where K is a constant. Therefore performing gradient descent on J we have

$$(10)\quad \Delta W \propto -\frac{\partial J}{\partial W} = -\frac{\partial J}{\partial \mathbf{e}}\frac{\partial \mathbf{e}}{\partial W} \approx \mathbf{y}(2\mathbf{e})^T$$

where we have discarded a less important term (see [8] for details).In general [13], the minimisation of such a cost function may be thought to make the probability of the residuals greater dependent on the pdf of the residuals. Thus if the probability density function of the residuals is known, this knowledge can be used to determine the optimal cost function which in turn gives an optimal learning rule. This suggests a family of learning rules which are derived from the family of exponential distributions. Let the residual after feedback have probability density function

$$(11)\quad p(\mathbf{e}) = \frac{1}{Z}\exp(-|\mathbf{e}|^p)\cdot$$

Then we can denote a general cost function associated with this network as

$$(12)\quad J = -\log p(\mathbf{e}) =| \mathbf{e}|^p + K$$

where K is a constant. Therefore performing gradient descent on J we have

$$(13)\quad \Delta W \propto -\frac{\partial J}{\partial W} = -\frac{\partial J}{\partial \mathbf{e}}\frac{\partial \mathbf{e}}{\partial W} \approx y(p\,|\mathbf{e}|^{p-1}\,sign(\mathbf{e}))^T\cdot$$

where T denotes the transpose of a vector. We would expect that for leptokurtotic residuals (more kurtotic than a Gaussian distribution), values of p<2 would be appropriate, while for platykurtotic residuals (less kurtotic than a Gaussian), values of p>2 would be appropriate. It has been shown in the ICA community [7] that it is less important to get exactly the correct distribution when searching for a specific source than it is to get an approximately correct distribution i.e. all supergaussian signals can be retrieved using a generic lepkurtotic distribution and all subgaussian signals can be retrieved using a generic platykurtotic distribution. Our experiments will tend to support this belief to some extent but we often find accuracy and speed of convergence are improved when we are accurate in our choice of p.

Therefore the network operation is as before except:

Weight change:

(14) $\Delta W_{ij} = \eta . y_i . sign(e_j) | e_j |^{p-1}$

By maximising the likelihood of the residual with respect to the actual distribution, we are matching the leaming rule to the pdf of the residual. We may thus link the method to the standard statistical method of Exploratory Projection Pursuit. Now the nature and quantification of the interestingness is in terms of how likely the residuals are under a particular model of the pdf of the residuals.

RESULTS USING ARTIFICIAL DATA SETS

We follow [5] in creating artificial data sets, each of 10 dimensions. All results reported are based on a set of 10 simulations each with different initial conditions. It is our general finding that sphering is necessary to get the most accurate results presented below.

In the first data set, we have 9 leptokurtotic dimensions and one gaussian dimension; this is almost the opposite of the standard EPP data sets described in [5] and is rather far from being a typical data set in that most of the directions in terms of its natural basis are non-Gaussian. However, since we wish to investigate our new models, it is a good test set since we can easily see the results of our method. We wish to identify the single Gaussian dimension and ignore the leptokurtotic dimensions. The leptokurtotic dimensions may be characterised as having long tails; if a residual can be created by removing the Gaussian direction from the data set, the residual will automatically be leptokurtotic. Thus we consider maximising the likelihood of the residual using the model

(15) $p(\mathbf{e}) = \frac{1}{Z}\exp(-|\mathbf{e}|^{p})$ with $p<2$;

We have experimented with a number of values of p and report on simulations with p=1.5. A typical result is shown in Fig. 1; the Gaussian direction is clearly identified.

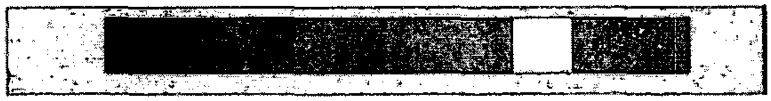

Fig. 1. The Gaussian direction was the third among 9 leptokurtotic dimensions. It has clearly been identified in this Hinton map of the weights.

We have similar results with a data set containing 9 platykurtotic dimensions and one Gaussian dimension. We use the same learning rules as before but with a value of p=3. If our data set consists of 9 Gaussian dimensions and 1 leptokurtotic dimension, we can identify the leptokurtotic dimension with a rule using p>2. This is really saying that all residuals will be unlikely using this model but that the leptokurtotic dimension is more wrong under the platykurtotic model than the Gaussian dimensions and should be removed from the residual. In the next section, we derive an alternative method for this data set.

COMPARING AND MIXING THE TWO EPP METHODS

We now compare the effectiveness of the two algorithms on artificial data sets. The artificial data is used to compare the speed of convergence of the algorithms in identifying interest in a data set since we know, in advance, exactly what sort of interesting structure is in the data set and can measure the progress of the algorithm towards identifying the structure. We will call the original algorithm the Higher Moments Algorithm.

In this section, we create a 10 dimensional data set in which 9 dimensions are drawn from a Gaussian distribution and one dimension from a uniform distribution. The uniform distribution is platykurtotic (has less kurtosis than the Gaussians) and so the higher moments algorithm can use y^3 or more stably tanh(); the maximum likelihood method will use p<2. The rate of convergence of the algorithms is shown in Figure 2: the left figure shows the dot product of the weights with the ideal solution when the higher moments EPP algorithm with a tanh() nonlinearity is used while the right shows the convergence of the Maximum Likelihood EPP algorithm with p=1. We see that the latter has extremely fast convergence but does not achieve an accuracy of more than 0.9 while the former, though it takes a little longer to get to the optimum, is much more accurate. This suggests that an algorithm which uses both rules might gain by having the best attributes of both and this is in fact the case.

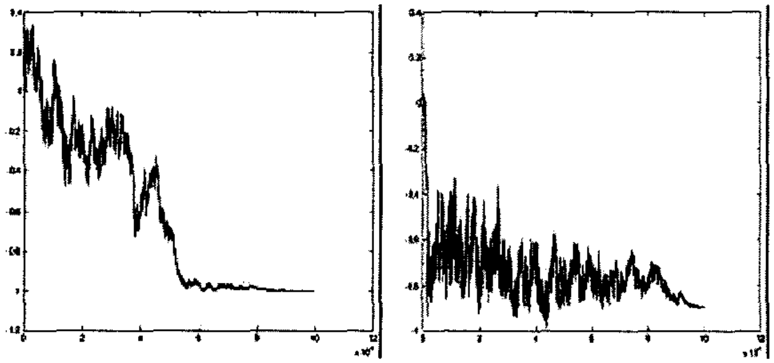

Fig 2. The left figure shows the convergence of the higher moments EPP algorithm while the right one shows the convergence of the Maximum Likelihood EPP algorithm in terms of the dot product to the ideal solution.

Figure 3 shows the convergence of an algorithm which uses a combination of these two rules i.e.

Weight change: $\Delta W_{ij} = \eta . f(y_i).sign(e_j) | e_j |^p$ where f() is the tanh() function in the experiment the results of which are shown in Figure 3. We seem to be getting the best of both worlds with this combined method though it must be conceded that the combination is somewhat ad hoc. It is for this reason that we have not included results from this method elsewhere.

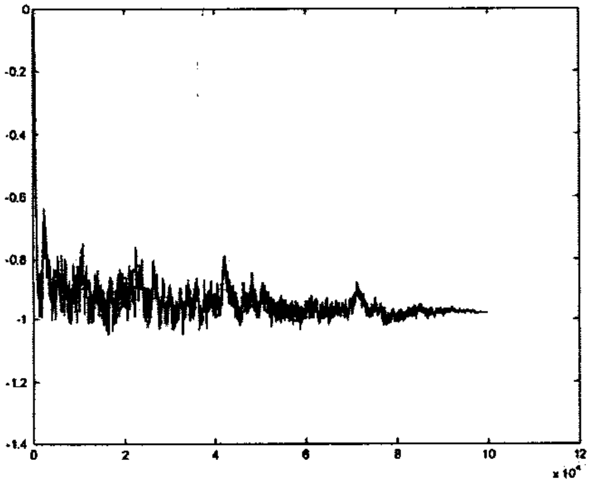

Fig.3. Convergence of the algorithm using the combined learning rule.

EXPERIMENTS USING AN ASTRONOMICAL DATA SET

The data consists of 65 colour spectra of 115 asteroids used by [10]. We have previously compared the performance of a variety of artificial neural networks on this data set [9].

The data set is composed of a mixture of the 52-colour survey by [1] together with the 8-colour survey conducted by [16] providing a set of asteroid spectra spanning 0.3-2.5μm. When this extended data set was compared by [10] to the results in Tholen [14] it was found that the additional refinement to the spectra lead to more classes in the taxonomy produced by Tholen. We have tested the networks on this data set looking at the differences in classification accuracy between clustering and projection networks. Standard PCA (p=2) separates out the classes A and (some of) B but leaves most of the others in a single group (Fig. 4)

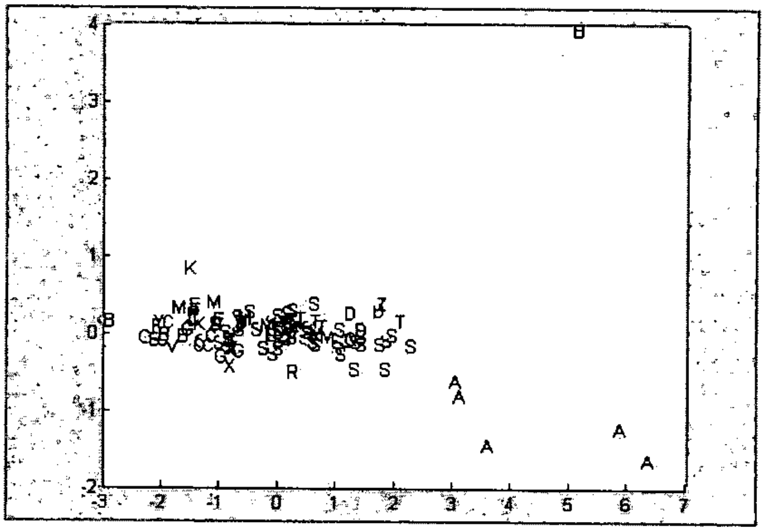

Fig. 4. Projection of the asteroid data set onto the first two Principal Components. Maximum Likelihood Learning with $p<2$ however shows a much greater separation of this central cluster (Figure 5 was from a simulation with $p=0.5$).

If we compare Figure 4 and 5, we see that both find the classes A and (some of) B easy to separate but Maximum Likelihood learning with $p<2$ does spread the data out somewhat better.

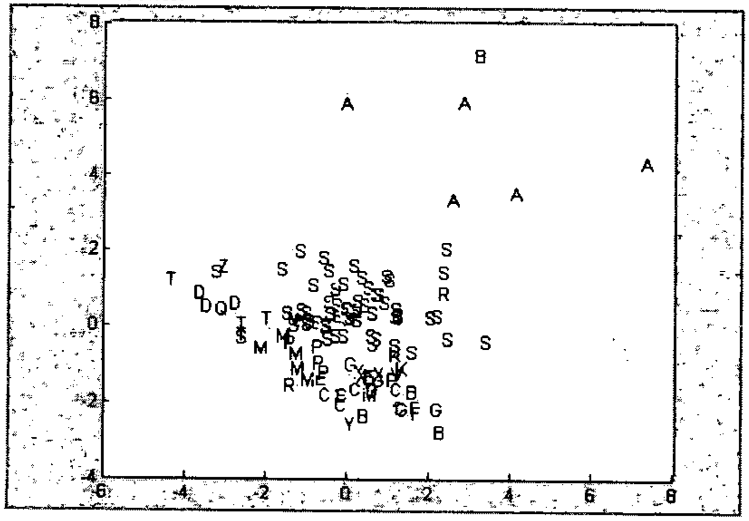

Fig. 5. Projection of the asteroid data onto the filters found by Maximum Likelihood Hebbian Learning with $p=0.5$

INDEPENDENT COMPONENT ANALYSIS

Independent Component Analysis (ICA) networks are often derived as extensions of PCA networks. ICA and the related problem of blind source separation have recently gained much interest due to many applications.We may describe the problem as follows. Let there be N independent non-Gaussian signals $\mathbf{s}_1,\mathbf{s}_2,\ldots,\mathbf{s}_N$ which are mixed using a (square) mixing matrix A to get N vectors, $\mathbf{x}_i$ each of which is an unknown mixture of the independent signals, $\mathbf{x} = A\,\mathbf{s}$.

Then the aim is to use an artificial neural network to retrieve the original input signals when the only information presented to the network is the unknown mixture of the signals. The neural network's output will be **y** where $\mathbf{y}= W\,\mathbf{x}$. Note that the outputs, **y** are to be the elements of the original signal in some order i.e. we are not insisting that the first output of our neural network is equal to the first signal, the second equal to the second signal and so on. We merely insist that neuron i's output is one of the N original signals not mixed any of the other signals. There is also an amplitude ambiguity which cannot be resolved.

Now the Central Limit Theorem states that mixtures of signals are liable to be more gaussian than the individual signals. This suggests using the Maximum Likelihood rules in quite a different way from that proposed previously. If we have a mixture of n kurtotic signals, completely removing one of these signals gives a residual which is more kurtotic than removing a little of each of the signals.

Extraction Of A Signal From A Mixture

We begin with three mixed speech signals which we have previously used to demonstrate neural methods of performing ICA. We linearly mixed these three signals using a random mixing matrix.

Each of the signals comprised 40000 samples of a speaker stating "Perhaps the most frequent use of ICA is in the extraction of independent voices from a mixture". The kurtosis of the individual signals is 6.8444, 7.7582 and 3.6833 respectively. We use a learning rate of 0.0001 and 100000 iterations (randomly sampling with replacement from the 40000 samples) to extract each signal in a deflationary manner - we use maximum likelihood learning with a value of p=1 (a Laplacian distribution) which extracts one signal completely (see below) and then repeat the experiment with the residual mixture of the other two signals.

As a measure of success, we use W*V*A i.e. the weights learned by the method times the sphering matrix times the mixture matrix:

W*V* A=

0.0170	-0.0050	**-1.0000**
1.0008	-0.0093	0.0162
0.0122	**0.9995**	-0.0021

Table 1. We see that the product matrix is very close to a permutation matrix showing that the signals have been extracted correctly.

Similarly we have experimented with 5 subgaussian artificially generated signals randomly mixed. Their kurtosis values were -0.9845, -0.9638, -0.9769, -0.9795 and -0.9673 respectively. Notice that all sample kurtosis values are approximately equal, something which causes other methods some difficulty. Again we used 40,000 samples, a learning rate of 0.0001 and no annealing of the learning rate, with p=4. This somewhat more difficult problem required 500 000 iterations (but see below) for each signal and the product matrix WVA is

WVA =

-0.0330	0.0219	**1.0011**	-0.0458	0.0134
-0.0211	-0.0351	0.0552	**0.9977**	-0.0288
0.0386	0.0228	-0.0010	-0.0122	**-0.9713**
-0.0062	**-0.9966**	0.0275	-0.0399	-0.0999
-0.9986	0.0125	-0.0363	-0.0041	-0.0214

Table 2. Again we have almost a permutation matrix indicating that the sources have been recovered. Now the reason we needed to use 500000 iterations is that the last signal is much the most difficult to extract (and we see that in the last column its accuracy is much worse). The reason for that lies in the fact that there is only one signal left in the "mixture" at this time: the network is attempting to structure the residuals to model the probability density function but if it is successful, there will be no residuals to model. This is somewhat of a conundrum. Of course, we can obviate this conundrum by simply noting that the fourth residual contains only the last signal but this is somewhat unsatisfactory since we do not (in a truly blind problem) know {\em a priori} how many signals are in the mixture.

CONCLUSIONS

We have derived a family of learning rules based on the probability density function of the residuals. The real power of these learning rules is in the context of exploratory data analysis This family of rules may be called Hebbian in that all use a simple multiplication of the output of the neural network with some function of the residuals after feedback. The power of the method comes from the choice of an appropriate function. We showed how to choose a function to maximise the likelihood of the residuals under particular models of probability density functions. We now see that both the original PCA rule and the ɛinsensitive rule [5] are merely particular cases of this class of rules. We have also shown that the rules are more akin to Exploratory Projection Pursuit and prefer to call them Maximum Likelihood Hebbian learning, believing that 'e-insensitive PCA' does not do justice to the power of the method. We have also shown how powerful Minimum Likelihood Hebbian learning is and indeed that this is, in some sense, even more closely related to EPP. These are powerful new tools for the data mining community and should take their place along with existing exploratory methods.

REFERENCES

[1].Bell, J. F., Owensby, P. D., Hawke, B. R. and Gaffey, M. J.. "The 52 colour asteroid survey: Final Results and interpretation, (abstract) Lunar Planet Sci. Conf., XIX, 57, (1988)

[2]. Bishop, C.M, Neural Networks for Pattern Recognition, Oxford, (1995).

[3]. Diaconis, P. and Freedman D., Asymptotics of Graphical Projections. The Annals of Statistics. 12(3), (1984) pp.793-815.

[4]. Fyfe, C., "PCA Properties of Interneurons", From Neurobiology to Real World Computing, Proceedings of International Conference on Artificial on Artificial Neural Networks, ICAAN 93, (1993) pp.183-188.

[5]. Fyfe, C. and Baddeley, R. Non-linear Data Structure Extraction using Simple Hebbian Learning, Biological Cybernetics,72(6), (1995) pp. 533-541,.

[6]. Fyfe, C. and MacDonald, D., e-Insensitive Hebbian learning, Neurocomputing, 2001.

[7].Hyverinnen, A. Complexity Pursuit: Separating interesting components from time series. Neural Computation, 13: (2001) pp.883-898.

[8]. Karhunen, J. and Joutsensalo, J., Representation and Separation of Signals Using Nonlinear PCA Type Learning, Neural Networks, 7 (1994) pp.113-127,.

[9]. MacDonald, D., McGlinchey S., Kawala , J. and Fyfe, C.. "Comparison of Kohonen, scale-invariant and GTM self-organising maps for interpretation of spectral data" European Symposium on Artificial Neural Networks (ESANN '99), (1999) pp.117-122

[10]. Merenyi, E. Self-organising ANNs for Planetary Surface Composition Research. Journal of Geophysical Research, 99:E5 (1994) pp. 10847-10865.

[11]. Oja, E., Neural Networks, Principal Components and Subspaces, International Journal of Neural Systems, 1 (1989) pp.61-68.

[12].Oja, E., Ogawa, H., Wangviwattana, J., Principal Components Analysis by Homogeneous Neural Networks, part 1, The Weighted Subspace Criterion, IEICE Transaction on Information and Systems, E75D (1992) pp. 366-375.

[13]. Simola, A.J. and. Scholkopf, B. A Tutorial on Support Vector Regression. Technical Report NC2-TR-1998-030, NeuroCOLT2 Technical Report Series, Oct.1998.

[14]. Tholen, D. 1994. "Asteroid taxonomy from cluster analysis of photometry", Ph.D. dissertation, University of Arizona.

[15]. Xu L., Least Mean Square Error Reconstruction for Self-Organizing Nets", Neural Networks, Vol. 6, (1993)pp. 627-648.

[16]. Zellner, B., Tholen, D. J. and Tedesco, E. F. 1985. "The eight-colour asteroid survey : Results from 589 minor planets", Icarus, pp. 355-416.