Kernel Maximum Likelihood Hebbian Learning

Jos Koetsier[1], Emilio Corchado[2], Donald MacDonald[1],
Juan Corchado[3], Colin Fyfe[1].

[1] Applied Computation Intelligence Research Unit, University of Paisley, Scotland.
{koet-ci0,macd-ci0, fyfe-ci0}@paisley.ac.uk
[2]Departamento de Ingenieria Civil. Universidad de Burgos. Spain.
escorchado@ubu.es
[3]Departamento de Informática y Automática. Universidad de Salamanca. Spain.
corchado@usal.es

Abstract. We present a novel method based on a recently proposed extension to a negative feedback network which uses simple Hebbian learning to self-organise called Maximum Likelihood Hebbian learning [2]. We use the kernel version of the ML algorithm on data from a spectroscopic analysis of a stained glass rose window in a Spanish cathedral. It is hoped that in classifying the origin and date of each segment it will help in the restoration of this and other historical stain glass windows.

1 Introduction

One problem with the analysis of high dimensional data is identifying structure or patterns which exist across dimensional boundaries. By projecting the data onto a different basis of the space, these patterns may become visible. This presents a problem - how does one decide which basis is optimal for the visualisation of the patterns, without foreknowledge of the patterns in the data.

One solution is Principal Component Analysis (PCA), which is a statistical technique aimed at finding the orthogonal basis that maximises the variance of the projection for a given dimensionality of basis. This involves finding the direction which accounts for most of the data's variance, the first principal component; this variance is then filtered out. The next component is the direction of maximum variance from the remaining data and orthogonal to the 1st PCA basis vector.

We [3, 4] have over the last few years investigated a negative feedback implementation of PCA defined by (1- 3). Let us have an N-dimensional input vector, $\mathbf{x}$, and an M-dimensional output vector, $\mathbf{y}$, with W_{ij} being the weight linking the jth input to the ith output. The learning rate, η, is a small value which will be annealed to zero over the course of training the network. The activation passing from input to output through the weights is described by (1). The activation is then fed back though the weights from the outputs and the error, $\mathbf{e,}$ calculated for each input dimension. Finally the weights are updated using simple Hebbian learning.

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

$$e_j = x_j - \sum_{i=1}^{M} W_{ij} y_i, \ \forall j \tag{2}$$

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

We have subsequently modified this network to perform clustering with topology preservation [5], to perform Factor Analysis [8, 1] and to perform Exploratory Projection Pursuit (EPP) [7, 6].

2 Maximum Likelihood Hebbian Learning

This paper deals with a recently developed variation of the basic network which also performs Exploratory Projection Pursuit. We show the validity of our method by applying it to data from a spectroscopic analysis of the stained glass rose window in a Spanish cathedral. It is hoped that in classifying the origin and date of each segment it will help in the restoration of this and other historical stain glass windows.
Let us now consider the residual after the feedback to have probability density function.

$$p(\mathbf{e}) = \frac{1}{Z} \exp(-|\mathbf{e}|^p) \cdot \tag{4}$$

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

$$J = -\log p(\mathbf{e}) = |\mathbf{e}|^p + K \tag{5}$$

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

$$\Delta W \propto -\frac{\partial J}{\partial W} = -\frac{\partial J}{\partial \mathbf{e}} \frac{\partial \mathbf{e}}{\partial W} \approx y(p|\mathbf{e}|^{p-1} \ sign(\mathbf{e}))^T \tag{6}$$

We would expect that for leptokurtotic residuals (more kurtotic than a Gaussian distribution), values of $p < 2$ would be appropriate, while platykurtotic residuals (less kurtotic than a Gaussian), values of $p > 2$ would be appropriate.

3 Kernel Maximum Likelihood

The first step in our kernel version of the ML algorithm can be performed by projecting our data onto a set of eigenvectors in feature space and thus we obtain sphered data in feature space. We can subsequently reduce the dimensionality as a further pre-processing step to the ML method. As the transformed data are actually points in feature space, we can simply apply the maximum likelihood method on the transformed data, as though it would be in data space.

Let $\mathbf{z}_k$ be the datapoint $\mathbf{x}_k$ transformed into feature space and projected onto the principal components in feature space. The Kernel Maximum Likelihood (KML) learning rules then become

Feedforward:
$$y_i = \sum_{j=1}^{N} W_{ij} z_{kj}, \forall_i \qquad (7)$$

Feedback:
$$e_j = z_{kj} - \sum_{i=1}^{M} W_{ij} y_i \qquad (8)$$

Weightchange:
$$\Delta W_{ij} = \eta y_i sign(e_j) \|e_j\|^{p-1} \qquad (9)$$
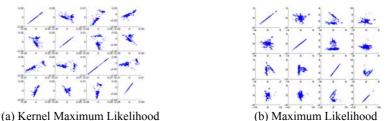
4 Experiments

The data used to illustrate our method is composed of samples from 76 different sections of the stained glass rose window in a Spanish cathedral. The data contained 450 data vectors obtained from 90 samples each having been analysed 5 times. The data is 1020 dimensions, which after normalisation was reduced to 390 dimensions.

The data was analysed by [9] and during their analysis of the data they found that it contained three different clusters. These clusters were identified to be separate by their chemical composition, one belonging to the 16th century, and the other two classes which were from the 13th century. These clusters were classified by the proportion of sodium and potassium in the glass.

We visualise the data using our KML method on this data so we can look for clusters by eye. As we are interested in projection exhibiting clusters, we wish to extract subgaussian signals and therefore we use a value for $p = 0$. We compare our results to those obtained by normal ML.

4.1 Results

In Figure 1, we show the projections obtained by ML (Right) and KML (Left). The first row of the figures show scatterplots of projections onto the first direction set out against projections onto first until the fourth directions. The second row shows projections onto the second direction set out against projections onto first until the fourth directions and so forth. In both figures we can identify clusters, but the projections obtained by the KML method generally result in a better separation of the clusters. The non-linear extension to the standard Maximum Likelihood method has given the method greater flexibility which allows us to get a different view on the data that certainly in this case yields much better clustering.

(a) Kernel Maximum Likelihood   (b) Maximum Likelihood

Figure. 1. Projection of the glass data using the KML (left) and the ML methods (right). The figures show the result of projection onto four directions found. It can be seen that the Kernel Maximum Likelihood method results in greater separation of the clusters.

5 Conclusion

In this paper we have introduced a novel extension to the Maximum Likelihood Hebbian learning algorithm that allows the method to use non linear projections. The data is actively being used in research projects to help in future restoration of stain glass windows. Using our new clustering methods we visualise the data to identify relationships between the different chemical properties of the glass samples.

We have shown that our new non linear extension of Maximum Likelihood has found clusters in this data set that could not be identified by ML. This new method therefore allows us to better analyse the data and give us different visualisations of our data set.

References

1. Charles, D. and Fyfe, C. Modelling Multiple Cause Structure using Rectification conStraints. Network: Computation in Neural Systems, 9:167-182, May 1998.
2. Corchado, E. and Fyfe, C. Maximum Likelihood Hebbian Rules. In Tenth European Symposium on Artificial Neural Networks, ESANN2002, pages 143-148, 2002.
3. Fyfe, C. PCA Properties of Interneurons. In From Neurobiology to Real World Computing, ICANN 93, pages 183-188, 1993.
4. Fyfe, C. Introducing Asymmetry into Interneuron learning. Neural Computation, 7(6):1167-1181, 1995.
5. Fyfe, C. Radial Feature Mapping. In International Conference on Artificial Neural Networks, ICANN95, Oct. 1995.
6. Fyfe, C. A Comparative Study of Two Neural Methods of Exploratory Projection Pursuit. Neural Networks, 10(2):257-262, 1997.
7. Fyfe, C. and Baddeley, R. Non-linear Data Dtructure Extraction using Simple Hebbian Networks. Biological Cybernetics, 72(6):533{541, 1995.
8. Fyfe, C. and Charles, D. Using Noise to Form a Minimal Overcomplete Basis. In Seventh International Conference on Artificial Neural Networks, ICANN99, 1999.
9. Lopez-Gejo, J. Colina, A. Lopez-Palacios, J. and Bravo, P. Principal Components Analysis in the Classification of Medieval Glasses by Scanning Electron Microscopy Coupled with Energy Dispersive X-ray Analysis. (submitted), 2003.