ESANN'2002 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 24-26 April 2002, d-side publi., ISBN 2-930307-02-1, pp. 483-488

Exploratory Correlation Analysis

Jos Koetsier, Donald MacDonald, Darryl Charles and Colin Fyfe
Applied Computational Intelligence Research Unit,

University of Paisley,
Scotland
(koet-ci0, macd-ci0, char-ci0 , fyfe-ci0 @paisley.ac.uk)

Abstract. We present a novel unsupervised artificial neural network for the extraction of common features in multiple data sources. This algorithm, which we name Exploratory Correlation Analysis (ECA), is a multi-stream extension of a neural implementation of Exploratory Projection Pursuit (EPP) and has a close relationship with Canonical Correlation Analysis (CCA). Whereas EPP identifies "interesting" statistical directions in a single stream of data, ECA develops a joint coding of the common underlying statistical features across a number of data streams.

1 Introduction

In many real world situations, information is not available in a direct and clear way due to corruption of the signals. One approach to uncovering the inherent structure from these signals is to perform several measurements, possibly using different sensing techniques. By working on the principle that all of the signals share the same fundamental information, we may process the data in multiple streams in such a way that we identify significant features within streams that are also common between streams.

In this paper we present a neural method capable of extracting features from different data sources and combining these to form a jointly sparse coding. Other statistically based dual stream neural architectures have been proposed [1] [2] [3] but these tend to be based on second-order canonical correlation analysis. The method that we propose is capable of searching for higher order shared structure between data streams. Information theoretic based approaches have also been proposed which concentrate on the stereo disparity problem [4] or on contextual guidance [5].

2 Exploratory projection pursuit

Before we outline the Exploratory Correlation Analysis (ECA) algorithm, it is useful to explain the method on which it is based - Exploratory Projection

ESANN'2002 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 24-26 April 2002, d-side publi., ISBN 2-930307-02-1, pp. 483-488

Pursuit (EPP). EPP is a statistical technique that is used to visualise structure in high dimensional data. We project the data to a lower dimensional space which enables us to look for interesting structure by eye. The projection should capture all of the aspects that we wish to visualise, which is done by maximising an index that defines a degree of 'interest' of the output distribution [6].

One such measure is based on an argument that states that random projections tend to result in Gaussian distributions [7]. Therefore, we can define an interesting projection as one that maximises the non Gaussianity of the output distributions. Several measures of non Gaussianity currently exist. In this paper we will concentrate on measures that are based on kurtosis and skewness.

2.1 Neural EPP

Our ECA network is most strongly related to the single stream, neural EPP algorithm based on the negative feedback framework [6]. The operation of the EPP network is outlined by (1) to (3). (1) describes the feed-forward step, in which the input values, $\mathbf{x}$, are multiplied by the weights, W, and summed to activate the output. This is followed by a feedback phase (2) in which the output values, $\mathbf{y}$, are fed back through the weights and subtracted from the input to form a residual, $\mathbf{r}$. This residual is then used in the weight update rule (3), where η is a learning parameter.

$$\mathbf{y} = W\mathbf{x} \quad (1)$$

$$\mathbf{r} = \mathbf{x} - W^T\mathbf{y} \quad (2)$$

$$\Delta W = \eta \mathbf{f}(\mathbf{y})\mathbf{r}^T \quad (3)$$

The function $\mathbf{f}(\mathbf{y})$ in (3) causes the weight vectors to converge to directions that maximise a function,$\mathbf{g}(\mathbf{y})$, whose derivative is $\mathbf{f}(\mathbf{y})$.

For reasons of stability, the output functions are replaced by functions that have the same truncated Taylor Expansion. Instead of using $\mathbf{f}(\mathbf{y}) = \mathbf{y}^3$ the function $\mathbf{f}(\mathbf{y}) = -\tanh(\mathbf{y}) = -\mathbf{y} + \frac{1}{3}\mathbf{y}^3 - \frac{2}{15}\mathbf{y}^5 + \ldots$ may be used.

3 Exploratory Correlation Analysis

We have extended the Neural EPP algorithm to allow for multiple input streams. Both streams are assumed to have a set of common underlying factors. Mathematically we can write this as

$$\mathbf{y}_1 = W\mathbf{x}_1$$
$$\mathbf{y}_2 = V\mathbf{x}_2$$

The input streams are denoted by $\mathbf{x}_1$ and $\mathbf{x}_2$, the projected data by $\mathbf{y}_1$ and $\mathbf{y}_2$ and the basis vectors are the rows of the matrices W and V. Each input stream can be analysed separately by performing EPP and finding common statistical features that have maximum non Gaussianity. However, if we know

ESANN'2002 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 24-26 April 2002, d-side publi., ISBN 2-930307-02-1, pp. 483-488

that the features we are looking for have the same statistical structure, we can add another constraint which maximises the dependence between the outputs. This is depicted schematically in Figure 1.

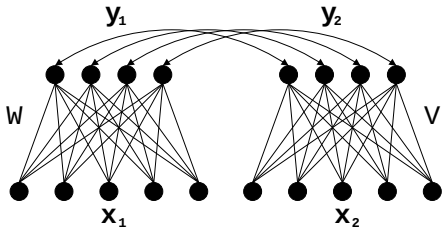

Figure 1: Diagram of the ECA network

The simplest way to express this formally is by maximising $E(\mathbf{g}(\mathbf{y}_1)^T\mathbf{g}(\mathbf{y}_2))$. We also need to ensure the weights do not grow without bound, which we can achieve by adding weight constraints $W^TW = A$ and $V^TV = B$. Writing this as an energy function with Lagrange parameters $\lambda_{i,j}$ and $\mu_{i,j}$ we obtain (4).

$$J(W,V) = E(\mathbf{g}(W\mathbf{x}_1)^T\mathbf{g}(V\mathbf{x}_2)) + \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\lambda_{i,j}(\mathbf{w}_i^T\mathbf{w}_j - a_{i,j}) + \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\mu_{i,j}(\mathbf{v}_i^T\mathbf{v}_j - b_{i,j})$$

Using a derivation similar to [8] and [6], we can optimise this function by stochastic gradient ascent. For $A = I$ and $B = I$, we obtain the following rules, where the $\otimes$ operator is defined as the element-wise multiplication of two vectors.

$$\Delta W = \eta[(\mathbf{g}(\mathbf{y}_2) \otimes \mathbf{g}'(\mathbf{y}_1))(\mathbf{x}_1^T - \mathbf{y}_1^T W)] \quad (4)$$

$$\Delta V = \eta[(\mathbf{g}(\mathbf{y}_1) \otimes \mathbf{g}'(\mathbf{y}_2))(\mathbf{x}_2^T - \mathbf{y}_2^T V)] \quad (5)$$

As with the neural EPP algorithm, we need to replace the output functions with stable versions for the ECA algorithm. In contrast to the neural EPP algorithm, we not only require the derivative of the function to be maximised, but also the function itself. We therefore need an additional stable function, whose truncated Taylor expansion is $\mathbf{g}(\mathbf{y}) = \mathbf{y}^4$. The function we chose for the experiments in this paper is $\mathbf{g}(\mathbf{y}) = 1 - \exp(-\mathbf{y}^4)$.

3.1 Connection to CCA

The linear one unit exploratory correlation analysis network is closely related to classical CCA. When the network is fully converged, the expected change in

weights will be zero [9].

$$\begin{aligned} E(\delta\mathbf{w}) &= E(\eta y_2(\mathbf{x}_1 - y_1\mathbf{w})) &&= \eta E((\mathbf{v}^T\mathbf{x}_2)\mathbf{x}_1 - (\mathbf{w}^T\mathbf{x}_1\mathbf{x}_2^T\mathbf{v})\mathbf{w}) = 0 \\ E(\delta\mathbf{v}) &= E(\eta y_1(\mathbf{x}_2 - y_2\mathbf{v})) &&= \eta E((\mathbf{w}^T\mathbf{x}_1)\mathbf{x}_2 - (\mathbf{v}^T\mathbf{x}_2\mathbf{x}_1^T\mathbf{w})\mathbf{v}) = 0 \end{aligned}$$

Writing the term $\mathbf{w}^T\mathbf{x}_1\mathbf{x}_2^T\mathbf{v}$ as λ_1, $\mathbf{v}^T\mathbf{x}_2\mathbf{x}_1^T\mathbf{w}$ as λ_2, $E(\mathbf{x}_1\mathbf{x}_2^T)$ as $C_{1,2}$ and $E(\mathbf{x}_2\mathbf{x}_1^T)$ as $C_{2,1}$ we obtain:

$$\begin{aligned} \mathbf{v}^T C_{2,1} &= \lambda_1 \mathbf{w}^T \\ \mathbf{w}^T C_{1,2} &= \lambda_2 \mathbf{v}^T \end{aligned}$$

and

$$\begin{aligned} C_{1,2}C_{2,1}\mathbf{w} &= \lambda_1\lambda_2\mathbf{w} \\ C_{2,1}C_{1,2}\mathbf{v} &= \lambda_1\lambda_2\mathbf{v} \end{aligned}$$

When the network is stable, the weight vectors will therefore be eigenvectors of $C_{1,2}C_{2,1}$ and $C_{2,1}C_{1,2}$. Classical CCA however, requires the solutions to be eigenvectors of $C_{1,1}^{-1}C_{1,2}C_{2,2}^{-1}C_{2,1}$ and $C_{2,2}^{-1}C_{2,1}C_{1,1}^{-1}C_{1,2}$. The ECA network is capable of performing CCA, if the data-sources $\mathbf{x}_1$ and $\mathbf{x}_2$ are sphered prior to training the network by pre-multiplying them by $C_{1,1}^{-1/2}$ and $C_{2,2}^{-1/2}$, which causes $C_{1,1}$ and $C_{2,2}$, and therefore their inverses to become identity matrices. The resulting CCA weight vectors will be $C_{1,1}^{-1/2}\mathbf{w}$ and $C_{2,2}^{-1/2}\ \mathbf{v}$.

4 Experiments

A simple experiment was performed to test the network. We used an artificial data-set, generated from a kurtotic and a normal data source. The inputs to the network are two three-dimensional input vectors as shown in Table 1. We used three types of data source, each with a different kurtosis value. Input S_1 and S_2 were generated by taking a value from a normal distribution and raising it to the power of 5. Input S_3 was generated from a normal distribution raised to the power of 3. The common data source S_3 is therefore less kurtotic than input S_1 or S_2. The last data source we used is S_4, which was taken from a normal distribution. In order to show the robustness of the network we added zero mean Gaussian noise with variance 0.2 to each of the inputs independently.

$x_{1,1} = S_1 + N(0, 0.2)$	$x_{2,1} = S_2 + N(0, 0.2)$
$x_{1,2} = S_3 + N(0, 0.2)$	$x_{2,2} = S_3 + N(0, 0.2)$
$x_{1,3} = S_4 + N(0, 0.2)$	$x_{2,3} = S_4 + N(0, 0.2)$

Table 1: Artificial data set. S_1 and S_2 are more kurtotic than the common source S_3. S_4 is a normal data source.

After training the network for 50000 iterations with a learning rate of 0.003, the weights converged to the values shown in Table 2 The network has clearly

identified the common kurtotic data source and has ignored the common normal input and the independent input sources S_1 and S_2, although they are more kurtotic than S_3.

w	0.0029	1.0000	0.0028
v	0.0043	1.0000	-0.0182

Table 2: Weightvectors after training the ECA network on artificial data.

4.1 Dual Stream Blind Source Separation

In this section we describe an experiment which is an adaption of the blind source separation problem [10]. In blind source separation we assume that we can model a set of observed data vectors $\mathbf{x}$ as a mixture of unknown sources $\mathbf{s}$. These sources are mixed by an unknown mixing matrix A so that $\mathbf{x} = A\mathbf{s}$. The goal is to find the unmixing matrix W, so that we can recover the unknown sources, where $\mathbf{s} = W\mathbf{x}$. The matrix W can be estimated by finding directions that are statistically independent, which amounts to maximising the non-gaussianity of the estimated source signals. It is therefore possible to use EPP to estimate a solution of the blind source separation problem.

Our adaption of the original blind source seperation problem uses two sets of inputs instead of one, which are both different linear mixtures of the same source signals. We used mixtures of three source signals, which were created artificially by randomly taking samples from a normal distribution and raising them to the power of 3, to give kurtotic signals.

The mixing matrices, A and B, are shown below.

$$A = \begin{pmatrix} 2 & 5 & 1 \\ 5 & 2 & 9 \\ 9 & 2 & 3 \end{pmatrix} \qquad B = \begin{pmatrix} 3 & 6 & 1 \\ 9 & 4 & 7 \\ 1 & 5 & 3 \end{pmatrix}$$

To show the unmixing properties of the network, we examine $WC_{1,1}^{-1/2}A$ and $VC_{2,2}^{-1/2}B$.

$$WC_{1,1}^{-1/2}A = \begin{pmatrix} -0.0013 & -1.0006 & -0.0006 \\ 1.0002 & 0.0311 & -0.0038 \\ -0.0021 & -0.0036 & 1.0000 \end{pmatrix}$$

$$VC_{2,2}^{-1/2}B = \begin{pmatrix} -0.0021 & -1.0006 & -0.0017 \\ 1.0002 & 0.0313 & -0.0031 \\ -0.0021 & -0.0036 & 1.0000 \end{pmatrix}$$

These matrices show that combining the mixing, sphering and unmixing operations result in matrices that contain a one or minus one in each row. This indicates that the ECA algorithm has successfully unmixed the sources and has identified the common sources.

5 Conclusion

We have presented a neural network based algorithm, based on the EPP network that can uncover joint structure in data streams. The learnt features represent a joint coding of the common underlying statistical features across the data streams. We have shown a close relation between the linear version of the ECA network and standard statistical CCA.

In the future we intend to explore other algorithms within this framework and investigate their application to natural image coding. Furthermore, the application of the network to areas of remote sensing may prove fruitful.

References

[1] P. L. Lai and C. Fyfe, "A neural network implementation of canonical correlation analysis," *Neural Networks*, vol. 12, no. 10, pp. 1391–1397, Dec. 1999.

[2] Z. Gou and C. Fyfe, "A family of networks which perform canonical correlation analysis," *International Journal of Knowledge-based Intelligent Engineering Systems*, April 2001.

[3] T. Landelius M. Borga, H. Knutsson, "Learning canonical correlations," *SCIA*, 1997.

[4] S.Becker, *An Information-theoretic Unsupervised Learning Algorithm for Neural Networks*, Ph.D. thesis, Gradute Department of Computer Science, University of Toronto, 1992.

[5] J. Kay and W.A. Phillips, "Activation functions, computational goals and learning rules for local processors with contextual guidance," Tech. Rep. CCCN-15, Centre for Cognitive and Computational Neuroscience, University of Stirling, April 1994.

[6] C. Fyfe, "A general exploratory projection pursuit network," *Neural Processing Letters*, vol. 2, no. 3, pp. 17–19, May 1995.

[7] P. Diaconis and D. Freedman, "Asymptotics of graphical projections," *The Annals of Statistics*, vol. 12, no. 3, pp. 793–815, 1984.

[8] J. Karhunen and J. Joutsensalo, "Representation and separation of signals using nonlinear pca type learning," *Neural Networks*, vol. 7, no. 1, pp. 113–127, 1994.

[9] A. Krogh J. Hertz and R. G. Palmer, *Introduction to the Theory of Neural Computation*, Addison-Wesley Publishing, 1992.

[10] C. Jutten and J. Herault, "Blind separation of sources, part 1: An adaptive algorithm based on neuromimetic architecture," *Signal Processing*, vol. 24, pp. 1–10, 1991.