Generalised Canonical Correlation Analysis

Zhenkun Gou and Colin Fyfe
Applied Computational Intelligence Research Unit,
The University of Paisley,
Scotland.
email:zhenkun.gou,colin.fyfe@paisley.ac.uk

Abstract. Canonical Correlation Analysis [3] is used when we have two data sets which we believe have some underlying correlation. In this paper, we derive a new family of neural methods for finding the canonical correlation directions by solving a generalized eigenvalue problem. Based on the differential equation for the generalized eigenvalue problem, a family of CCA learning algorithms can be obtained. We compare our family of methods with a previously derived [2] CCA learning algorithm. Our results show that all the new learning algorithms of this family have the same order of convergence speed and in particular are much faster than existing algorithms; they are also shown to be able to find greater nonlinear correlations. They are also much more robust with respect to parameter selection.

1 Canonical Correlation Analysis

Canonical Correlation Analysis is a statistical technique used when we have two data sets which we believe have some underlying correlation. Consider two sets of input data; \( x_1 \) and \( x_2 \). Then in classical CCA, we attempt to find the linear combination of the variables which give us maximum correlation between the combinations. Let

\[
y_1 = w_1 x_1 = \sum_j w_{1j} x_{1j}
\]

\[
y_2 = w_2 x_2 = \sum_j w_{2j} x_{2j}
\]

where we have used \( x_{ij} \) as the \( j \)th element of \( x_1 \). Then we wish to find those values of \( w_1 \) and \( w_2 \) which maximise the correlation between \( y_1 \) and \( y_2 \). Then the standard statistical method (see [3]) lies in defining:

\[
\Sigma_{11} = E\{(x_1 - \mu_1)(x_1 - \mu_1)^T\}
\]

\[
\Sigma_{22} = E\{(x_2 - \mu_2)(x_2 - \mu_2)^T\}
\]

\[
\Sigma_{12} = E\{(x_1 - \mu_1)(x_2 - \mu_2)^T\}
\]

and

\[
K = \Sigma_{11}^{-\frac{1}{2}} \Sigma_{12} \Sigma_{22}^{-\frac{1}{2}}
\]

(1)

© Springer-Verlag Berlin Heidelberg 2000
where $T$ denotes the transpose of a vector and $E\{\}$ denotes the expectation operator. We then perform a Singular Value Decomposition of $K$ to get

$$K = (\alpha_1, \alpha_2, ..., \alpha_k)D(\beta_1, \beta_2, ..., \beta_k)^T \tag{2}$$

where $\alpha_i$ and $\beta_i$ are the standardised eigenvectors of $KK^T$ and $K^T K$ respectively and $D$ is the diagonal matrix of eigenvalues. Then the first canonical correlation vectors (those which give greatest correlation) are given by

$$w_1 = \Sigma_{11}^{-\frac{1}{2}} \alpha_1 \tag{3}$$

$$w_2 = \Sigma_{22}^{-\frac{1}{2}} \beta_1 \tag{4}$$

with subsequent canonical correlation vectors defined in terms of the subsequent eigenvectors, $\alpha_i$ and $\beta_i$.

2 A neural implementation

A previous ‘neural implementation’ [2] of CCA was derived by phrasing the problem as that of maximising

$$J = E\{(y_1y_2) + \frac{1}{2}\lambda_1(1 - y_1^2) + \frac{1}{2}\lambda_2(1 - y_2^2)\}$$

where the $\lambda_i$ were motivated by the method of Lagrange multipliers to constrain the weights to finite values. By taking the derivative of this function with respect to both the weights, $w_1$ and $w_2$, and the Lagrange multipliers, $\lambda_1$ and $\lambda_2$ we derive learning rules for both:

$$\Delta w_{1j} = \eta x_{1j}(y_2 - \lambda_1 y_1)$$

$$\Delta \lambda_1 = \eta_0 (1 - y_1^2)$$

$$\Delta w_{2j} = \eta x_{2j}(y_1 - \lambda_2 y_2)$$

$$\Delta \lambda_2 = \eta_0 (1 - y_2^2) \tag{5}$$

where $w_{1j}$ is the $j^{th}$ element of weight vector, $w_1$ etc. If we consider the general problem of maximising correlations between two data sets which may be have an underlying nonlinear relationship, we can use some nonlinear function, for example tanh(), to train the output neurons. So, the outputs $y_1$ and $y_4$ can be calculated from:

$$y_3 = \sum_j w_{3j} \tanh(v_{3j}x_{1j}) = w_3g_3 \tag{6}$$

$$y_4 = \sum_j w_{4j} \tanh(v_{4j}x_{2j}) = w_4g_4 \tag{7}$$

The weights $v_3$ and $v_4$ are used to optimise the nonlinearity, which gives us extra flexibility in maximising correlations. The maximum correlation between $y_3$ and $y_4$ was found by maximising the function:

$$J = E\{(y_3y_4) + \frac{1}{2}\lambda_3(1 - y_3^2) + \frac{1}{2}\lambda_4(1 - y_4^2)\}$$