Nonlinear Generalized Canonical Correlation Analysis by Neural Network Models[1]

Yoshio Takane and Yuriko Oshima-Takane

McGill University
1205 Dr.Penfield Avenue, Montreal, Quebec, Canada H3A 1B1

Summary: A method of K-set canonical correlation analysis capable of joint multivariate nonlinear transformations of data was proposed. The method consists of K nonlinear data transformation modules, each of which is a multi-layered feed-forward network, and one integrator module which combines information from the K transformation modules. The proposed method is useful for integrating information from K concurrent sources.

1. Introduction

We propose a method of nonlinear generalized (K-set) canonical correlation analysis (NGCANO), where $K \geq 2$. Generalized CANO (Carroll, 1967; Horst, 1961; Meredith, 1964) is an interesting technique because it subsumes a number of existing techniques for multivariate data analysis as special cases. It specializes into the usual (2-set) CANO when $K = 2$. It reduces to principal component analysis (PCA) when each of the K sets consists of a single (usually) continuous variable, and to multiple correspondence analysis (MCA) when each set consists of a matrix of dummy variables representing responses to a multiple-choice questionnaire item. Generalized CANO has been extended to allow variable-wise nonlinear transformations of input variables (Gifi, 1990), called OVERALS. In this paper we further extend it to allow for joint multivariate nonlinear transformations of input variables by artificial neural networks.

There are a number of situations in which NGCANO could be useful. We may, for example, have a problem of integrating information from different sensors, from different modalities, from different measurement tools, and so on. Several different cues are available for depth perception, e.g., binocular disparities, motion parallax, shading, textures, occluding contours, etc. Different cues are processed more or less independently up to certain levels by separate brain modules, which should be integrated into coherent images (Marr, 1982). NGCANO approximates such information integration mechanisms. Becker and Hinton (1992) developed a similar procedure for $K = 2$ based on a somewhat different fitting criterion, which they successfully applied to identify surface structures in random dot stereograms. Asoh and Takechi (1994) devised an approximate method for Becker and Hinton's

[1]In Nishisato, S. et al. (Eds.), *Measurement and Multivariate Analysis* (pp. 183-190). Tokyo: Springer Verlag

method. NGCANO extends their methods to $K \geq 2$.

2. The method

Figure 1 displays the basic construction of NGCANO for $K = 3$. The three modules are enclosed by large squares. Each module (corresponding to a set of input variables) consists of a multi-layered feed-forward (MLFF) network. It accepts inputs, forms linear combinations of the inputs and transforms them by sigmoid transformations to obtain hidden-layer activations, which capture nonlinear and interaction effects among the input variables. It then forms linear combinations of hidden-layer activations as outputs from the network. NGCANO attempts to make the outputs from different modules as homogeneous as possible. This information integration part is depicted inside the octagon in the figure. The outputs from all the modules are made to approximate a single common set of quantities (called common canonical variates) as closely as possible.

2.1 Optimization criterion

Let O_k denote the matrix of outputs from module k, and let $F = [f_1, f_2]$ denote the matrix of canonical variates. Define

$$g = \sum_{k=1}^{K} g_k \qquad \text{with} \qquad g_k = \|F - O_k\|^2, \tag{1}$$

where $O_k = H_k W_k$, and $H_k = \sigma(X_k V_k)$ with σ being the sigmoid transformation. Here, X_k is the matrix of inputs, and V_k and W_k are matrices of the first and the second layer weights, respectively. We minimize g in (1) with respect to V_k, W_k $(k = 1, \ldots, K)$, and F subject to:

$$F'F = I \qquad \text{and} \qquad F'1_N = 0, \tag{2}$$

where 1_N is an N-element vector of ones (where N is the number of cases in the training sample, and 0 is a zero vector of appropriate size.

Output O_k from each module should approximate F as much as possible. Each O_k, in turn, is obtained by linear combinations of the matrix of hidden layer activations (H_k), which, in turn, are obtained by sigmoid transformations of some linear combinations of the input matrix (X_k). Constraints (2) state that F is column-wise centered and orthogonal, which are required for identification purposes.

2.2 Algorithms

There are three sets of parameters, $\{V_k, W_k, F\}$. We propose two algorithms to minimize (1). Algorithm I splits the parameter set into $\{V_k, W_k\}$ and $\{F\}$, whereas Algorithm II into $\{V_k\}$ and $\{W_k, F\}$.

Algorithm I: We alternate the following two steps until convergence is reached.

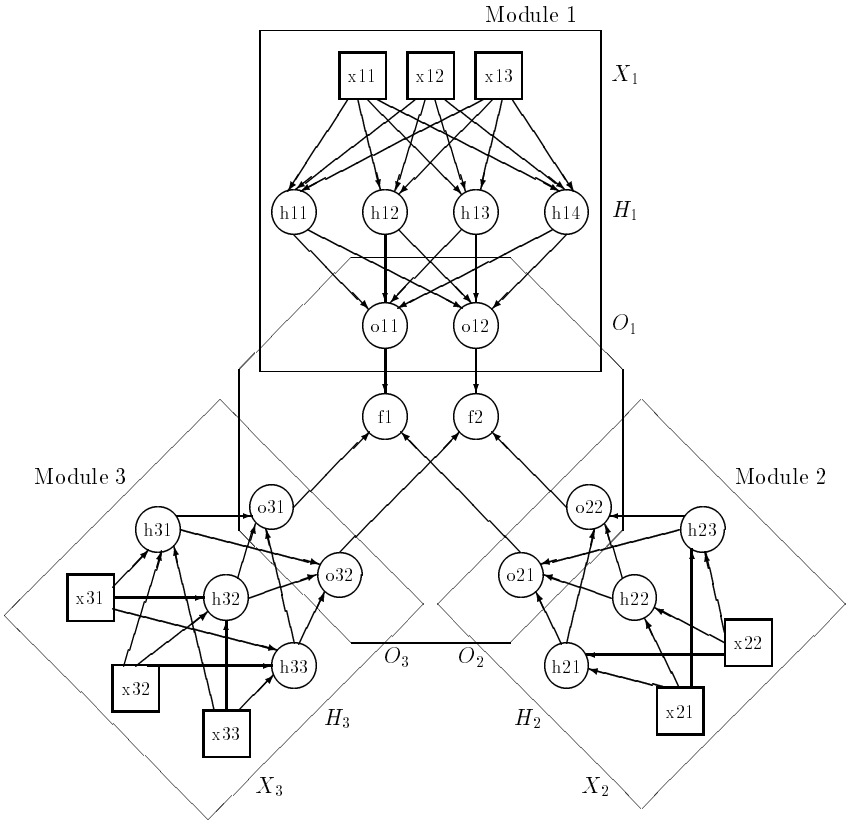

Figure 1: The basic construction of NGCANO for $K = 3$ modules with a single hidden layer in each module.

1. For $k = 1, ..., K$, minimize g_k with respect to V_k and W_k for fixed F by the Levenberg-Marquardt (LM) method.

2. Minimize g with respect to F subject to (2) for fixed V_k and W_k. This is done by first defining $F^* = (I_N - 1_N 1_N'/N) \sum_{k=1}^{K} O_k/K)$, where I_N is the identity matrix of order N, and then by applying the Gram-Schmidt orthogonalization method to F^* to obtain F. (Matrix $I_N - 1_N 1_N'/N$ has the effect of column-wise centering the matrix that follows.)

Step 1 in the above algorithm works just like separate MLFF networks with F as target outputs. A ready-made algorithm for MLFF networks (like the one in the Neural Network Tool Box in MATLAB) can be used for optimization in this phase.

Algorithm II: In the above algorithm, the most time consuming part is

Step 1 which involves an iterative optimization method. It is best to minimize the number of parameters in this step. In Algorithm II, we alternate the following two steps.

1. For $k = 1, ..., K$, minimize g_k with respect to V_k for fixed F and W_k by the LM method.

2. Minimize g with respect to F and W_k for fixed V_k. This is done as follows: Define $A = (I_N - 1_N 1_N'/N)HD^{-1/2}$ where $H = [H_1, H_2, \cdots, H_K]$ and D is a block diagonal matrix with $H_k'H_k (k = 1, ..., K)$ as diagonal blocks. We compute the generalized singular decomposition of A with column metric matrix D to obtain F. We then calculate W_k by $W_k = (H_k'H_k)^{-1}H_k'F$.

Both algorithms are monotonically convergent. Note also that in both algorithms Step 1 can be carried out for each module separately, which significantly reduces the number of parameters updated simultaneously in each optimization problem.

3. An illustrative example

The data used to demonstrate the feasibility of NGCANO is part of large scale survey data collected at the Institute of Statistical Mathematics in Tokyo. The survey questions asked about traditional versus modern views on Japanese society and culture. We used six items from the survey, five of which (items 1, 2, 3, 5 and 6) had three response categories and one (item 4) had two response categories. There were 1864 subjects responding to the survey questionnaire. This is the kind of data set to which multiple correspondence analysis (MCA) is typically applied. An analytic solution exists, so we know what NGCANO is supposed to obtain.

We used the so-called analog coding instead of dummy coding (as typically done in MCA); we arbitrarily assigned numbers to the response categories and treated them as values on an input variable in each module. Each module consisted of a single (continuous) input variable, so that the situation has direct analogy to nonlinear PCA. The assigned numbers could be any distinct numbers, although the first 2 or 3 consecutive integers were used in the present example. Which integers are used to code which response categories is also essentially arbitrary. NGCANO is supposed to find the best nonlinear transformations of these prescribed numbers.

We used one less hidden unit than the number of response categories in each module and obtained a solution with two canonical variates. The derived solution was virtually indistinguishable from that obtained by MCA. Figure 2 depicts the hidden unit activations and the output activations for two output units as functions of the single input variable for items (which coincide with modules) 2, 3 and 4. The hidden unit activations were obtained by sigmoid transformations of the input variable times the weights. The activation functions are bound to be monotonic (either increasing or decreasing). The output activations were obtained by linear combinations of the hidden unit activations, and may no longer be monotonic. An important thing is

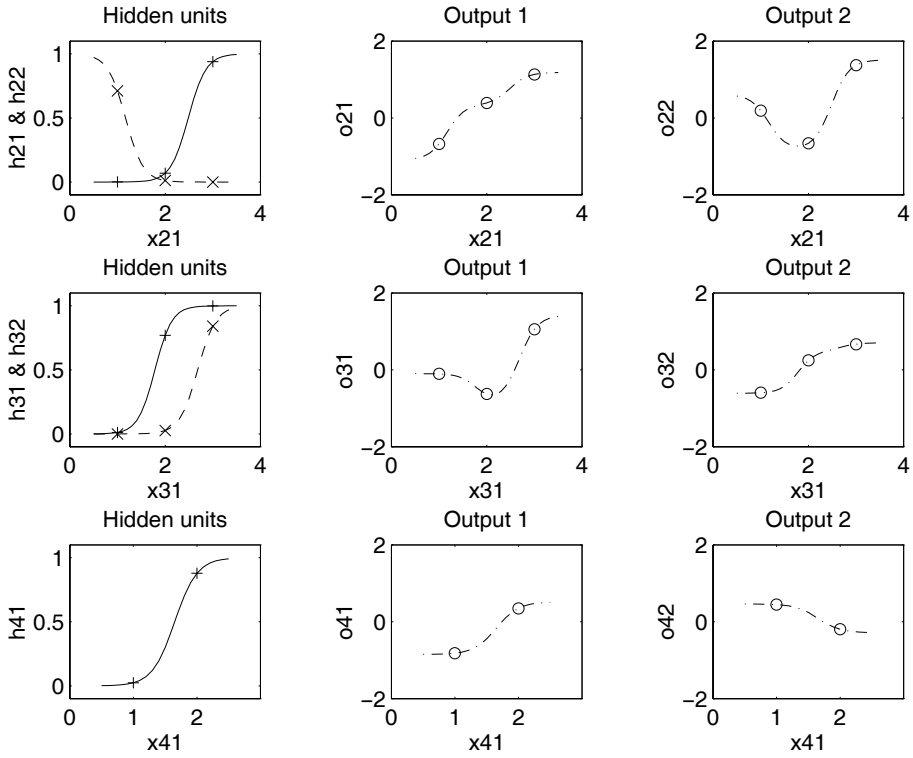

Figure 2: Hidden unit activations and best nonlinear transformations of input variables for module 2 (top), module 3 (middle), and module 4 (bottom). In each row the leftmost figure gives hidden unit activations (marked by "+" for h_1 and by "×" for h_2), and the center and the rightmost figures give output activations as functions of the input variable in the module.

that whereas in MCA we only get values at the response categories (marked by small circles), NGCANO fits a continuous nonlinear function that passes through these points. This does not mean that we can freely interpolate values between the response categories, but that in principle the input variables can take infinitely many values (in fact, a continuum of values) between prescribed values of the response categories. The output activation functions are generally not unique, but always pass through the points marked by small circles representing the response categories of the item in the module. There are infinitely many functions that pass through the three points, and different functions are typically obtained from different initial estimates of the weights in the network.

The second analysis investigates what happens if we include more than one item in one module. We included the first two items in module 1, and left the rest of the items as in the previous analysis. Again analog coding was used, and a solution with two canonical variates were obtained. There were

thus nine possible input patterns (3 by 3) in module 1, which were coded as 1 1, 1 2, 1 3, 2 1, 2 2, 2 3, 3 1, 3 2, and 3 3 on the two input variables in this module. We used eight hidden units in module 1 to obtain the solution. The derived solution was essentially the same as the one obtained by the so-called interactive coding of the first two items in MCA. In the interactive coding, we create an item with nine categories by factorial combinations of the three response categories in each of the two items.

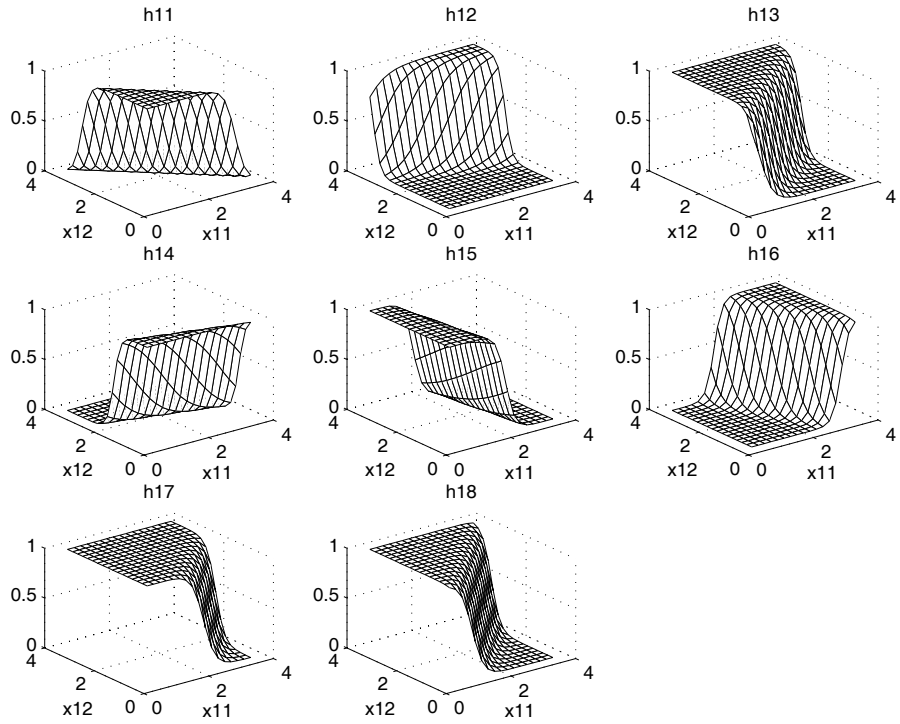

Figure 3: Hidden unit activations for the eight hidden units in module 1 as functions of the two input variables in the module, obtained by sigmoid transformations of some linear combinations of the two input variables.

Figure 3 depicts hidden unit activations as functions of the two input variables for the eight hidden units in module 1. All of them are monotonic in a particular direction on the $x_1 - x_2$ plane. These hidden unit activations were linearly combined to obtain output activations which may no longer be monotonic in any direction on the plane. Figure 4 depicts output activations for output 1 in module 1 corresponding to the first canonical variate, which is a nonlinear transformation of the two input variables in module 1. Small

circles indicate the function values at the prescribed values of the response categories. Some degree of interaction effects are observed, although there does not seem to be much interactions, as indicated by near parallel lines connecting points within particular response categories in each of the two items. A similar output activation function was obtained for output 2.

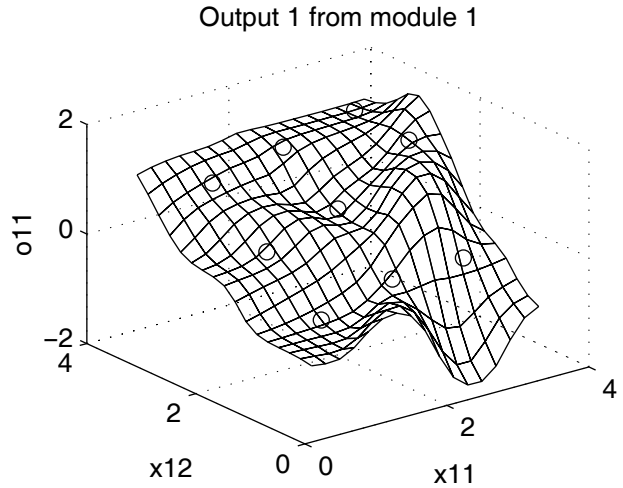

Figure 4: Output activations as functions of the two input variables in module 1 obtained by a linear combination of the hidden unit activations depicted in Figure 3.

4. Discussion and future prospects

The proposed method works in the way it is supposed to in the analysis of multiple-choice categorical data. Although results are not presented here because of space limitation, it has also been demonstrated that NGCANO works in situations where there are many more response categories in each item (to the extent that each category receives only one response). This is the situation in which nonlinear PCA is typically called for, which is another special case of NGCANO. These are very encouraging, although admittedly they are still preliminary.

The basic framework of the method presented above can be extended in a variety of ways. Below is a list of possible extensions, some of which have already been implemented.

1) Differential weighting of modules. Information processed by some modules is more important than others. Information from contradicting modules may duly be ignored completely. Different modules may therefore be differentially weighted to reflect their importance. The weights may be cho-

sen *a priori* or algorithmically according to how well each module fits to the criterion values, as in robust regression analysis by iterative reweighting.

2) Different numbers of hidden layers and different types of transfer functions across modules. In the model depicted in Figure 1, all modules had only one hidden layer, and the same sigmoid transfer functions were used for all the hidden units. Information processed by different modules may be of different types and of different complexity. In such cases it would be useful to allow different numbers of hidden layers and different types of transfer functions across different modules. In the example above, one of the items had only two response categories. Since two points can always be perfectly fitted by a linear function, no nonlinear transformation was necessary for this item, while the other items required multiple nonlinear transformations. This attests the necessity of differentiating the size and the complexity of the networks for data transformation purposes.

3) Differential weighting of training patterns. Some training patterns are more important and/or more reliable than others. In a manner similar to 1) above, a differential weighting scheme may be introduced to different training patterns. This may also be useful in dealing with missing data. We give the weight of one to observed data, while that of zero to missing observations.

4) Regularizations. The problem each module is solving may be extremely complicated, sometimes "ill-posed" in the sense that no unique solutions can be obtained due to the lack of key ingredients as inputs, as in the problem of recovering three-dimensional depth structures based on two-dimensional retinal images. This is called an inverse problem (Marr, 1982). In such cases it is essential that the information integrator is equipped with regularization terms representing prior knowledge (also known as smoothing terms, penalty terms, constraint terms, shrinkage terms, etc.) about the problems to be solved (Poggio, Torre, and Koch, 1985). There may be as many such terms as necessary for each module. This can be implemented by redefining g_k in (1) as

$$g_k = \|F - O_k\|^2 + \sum_i^{n_k} \rho_{ki} \|q_{ki}(V_k)\|^2, \tag{3}$$

where the ρ_{ki} are the penalty parameters, the q_{ki} are some functions of V_k, and n_k is the number of regularization terms for module k.

Acknowledgments

The work reported in this paper has been supported in part by NSERC individual operating grants to the authors, and in part by a team grant from Fonds pour la Formation de Chercheurs et l'Aide a la Recherche. We would like to extend our appreciation to Marina Takane who prepared Figure 1.

References:

Asoh, H., and Takechi, O. (1994). An approximation of nonlinear canonical correlation analysis by multilayer perceptrons. *ICANN 94 Proceedings of the International Conference on Artificial Neural Networks.* New York: Springer, 713–716.

Becker, S., and Hinton, G.E. (1992). Self-organizing neural network that discovers surfaces in random-dot stereograms. *Nature*, **355**, 161–163.

Carroll, J.D. (1968). A generalization of canonical correlation analysis to three or more sets of variables. *Proceedings of the 76th Annual Convention of the American Psychological Association*, 227–228.

Gifi, A. (1990). *Nonlinear multivariate analysis.* Chichester: Wiley.

Horst, P. (1961). Relations among m sets of measures. *Psychometrika*, **26**, 129–149.

Marr, D. (1982). *Vision: A computational investigation into the human representation and processing of visual information.* San Francisco: Freeman.

Meredith, W. (1964). Rotation to achieve factorial invariance. *Psychometrika*, **29**, 187–206.

Poggio, M.J., Torre, V., and Koch, C. (1985). Computational vision and regularization theory. *Nature*, **317**, 314–319.