A Canonical Coordinate Decomposition Network

Ali Pezeshki, Mahmood R. Azimi-Sadjadi, and Louis L. Scharf
Department of Electrical and Computer Engineering
Colorado State University
Fort Collins, Colorado 80523–1373
Email: ali@engr.colostate.edu

Abstract—**A network structure for canonical coordinate decomposition is presented. The network consists of two single-layer linear subnetworks that together extract the canonical coordinates of two data channels. The connection weights of the networks are trained by a stochastic gradient descent learning algorithm. Each subnetwork features a hierarchical set of lateral connections among its outputs. The lateral connections perform a deflation process that subtracts the contributions of the already extracted coordinates from the input data subspace. This structure allows for adding new nodes for extracting additional canonical coordinates without the need for retraining the previous nodes. The performance of the network is evaluated on a synthesized data set.**

I. Introduction

Canonical correlation analysis [1]-[4] provides a minimal description of the correlation between two data channels by concentrating the linear dependence of the channels into a small set of canonical variables. Canonical correlations are maximal invariants to uncoupled linear transformations of two-channel data [3],[4]. The corresponding canonical coordinates resolve the channels into coordinates that are only pairwise correlated [3],[4]. Canonical coordinates have been used to decompose Wiener filters and Gaussian communication channels into their canonical modes, where each mode corresponds to a scalar Gaussian channel or Wiener filter [3],[4]. They provide an elegant framework for analyzing linear dependence and mutual information between two data channels. In this coordinate system, the linear dependence and mutual information between the original channels are decomposed into those of canonical coordinates of the channels, which are determined by the corresponding canonical correlations. The canonical correlation associated with each pair of canonical coordinates determines the contribution of that pair to the linear dependence and mutual information between the channels [3],[4].

The conventional method of finding canonical coordinates [3],[4] involves computation of square-root-inverses of covariance matrices followed by an SVD of a coherence matrix. These operations become computationally intractable and inefficient especially for large dimensional data. In addition all the singular values and singular vectors of the coherence matrix have to be evaluated even though only the most significant singular values and their associated singular vectors are used in most applications. These deficiencies make the conventional scheme inefficient for real-time applications. Consequently, to perform the canonical coordinate decomposition efficiently, a method is required to extract the most significant canonical coordinate pairs and the corresponding canonical correlations recursively, and without any matrix inversion, matrix square root computation or direct SVD operations.

Neural networks have been proven to be powerful tools for performing complex transformations. Several neural network-based approaches haven been reported for extracting principal components of a stochastic vector process directly from the input data set [5]-[10]. In [11], a neural network structure has been proposed for computing the reduced-rank Wiener filter [4],[12]. A neural network-based approach has been reported in [13] for performing canonical correlation analysis. However, this network only finds the most significant canonical coordinate pair and the corresponding canonical correlation.

In this paper a network and a set of updating rules for performing canonical coordinate decomposition is presented. First, the problem of finding the first canonical coordinate pair is formulated as a constrained minimization problem. Then, given the first r canonical coordinate pairs, the problem of finding the $(r+1)$th pair is formulated as one of finding the first canonical coordinate pair after the contributions of the first r pairs are deflated from the input data subspace. This formulation is used to propose a network structure that consists of two single-layer linear subnetworks. The weights of the subnetworks are trained using a stochastic gradient descent learning algorithm. Each subnetwork consists of a set of lateral connections that whiten the output. The idea of using lateral connections among the outputs was first exploited in [7] for recursive extraction of principal components. In fact, the structure of each subnetwork is similar to the structure of the network reported in [7]. The lateral connections are trained to deflate the contributions of the already extracted canonical coordinates from the input data subspace. This structure allows for adding new nodes for extracting a new canonical coordinate without the need for retraining the previous nodes. This is very useful since in most cases the number of canonical coordinates or canonical correlations required is not known *a priori*. It is also useful where the statistics of the data channels are slowly varying with time. A simulation example is given to demonstrate the validity of the proposed network and the learning rules.

II. Canonical Coordinate Decomposition: A Review

Consider the two random vectors, $\mathbf{x} \in R^{m\times 1}$ and $\mathbf{y} \in R^{n\times 1}$ with m being the smaller dimension $(m \leq n)$. Assume that $\mathbf{x}$ and $\mathbf{y}$ have zero means and share the composite covariance

0-7803-7898-9/03/$17.00 ©2003 IEEE

matrix

$$E\left[\begin{pmatrix}\mathbf{x}\\ \mathbf{y}\end{pmatrix}\begin{pmatrix}\mathbf{x}^T & \mathbf{y}^T\end{pmatrix}\right]=\begin{bmatrix}\mathrm{R}_{xx} & \mathrm{R}_{xy}\\ \mathrm{R}_{yx} & \mathrm{R}_{yy}\end{bmatrix} \tag{1}$$

This composite covariance matrix has the following block tridiagonal decomposition [3],[4].

$$\begin{bmatrix}\mathrm{F}^T & 0\\ 0 & \mathrm{G}^T\end{bmatrix}\begin{bmatrix}\mathrm{R}_{xx}^{-1/2} & 0\\ 0 & \mathrm{R}_{yy}^{-1/2}\end{bmatrix}\begin{bmatrix}\mathrm{R}_{xx} & \mathrm{R}_{xy}\\ \mathrm{R}_{yx} & \mathrm{R}_{yy}\end{bmatrix}\cdot \\ \cdot\begin{bmatrix}\mathrm{R}_{xx}^{-T/2} & 0\\ 0 & \mathrm{R}_{yy}^{-T/2}\end{bmatrix}\begin{bmatrix}\mathrm{F} & 0\\ 0 & \mathrm{G}\end{bmatrix}=\begin{bmatrix}\mathrm{I} & \mathrm{K}\\ \mathrm{K} & \mathrm{I}\end{bmatrix} \tag{2}$$

where $\mathrm{R}_{xx}^{-1/2}\mathrm{R}_{xx}\mathrm{R}_{xx}^{-T/2}=\mathrm{I}$, $\mathrm{R}_{xx}^{1/2}\mathrm{R}_{xx}^{T/2}=\mathrm{R}_{xx}$, and F, G and K are chosen to be the economical SVD of the coherence matrix $\mathrm{C}=\mathrm{R}_{xx}^{-1/2}\mathrm{R}_{xy}\mathrm{R}_{yy}^{-T/2}$. That is,

$$\mathrm{C}=\mathrm{R}_{xx}^{-1/2}\mathrm{R}_{xy}\mathrm{R}_{yy}^{-T/2}=\mathrm{FKG}^T \quad \text{and} \quad \mathrm{F}^T\mathrm{CG}=\mathrm{K}, \\ \mathrm{F}^T\mathrm{F}=\mathrm{I},\ \ \mathrm{G}^T\mathrm{G}=\mathrm{I},\quad \mathrm{K}=\mathrm{diag}[k_1,k_2,\ldots,k_m]; \tag{3}$$

The diagonal matrix K is the canonical correlation matrix of canonical correlations k_i. The canonical correlations are arranged in descending order $(1\geq k_1\geq\ldots\geq k_m>0)$.

The canonical coordinates of **x** and **y** are defined as

$$\begin{bmatrix}\mathbf{u}\\ \mathbf{v}\end{bmatrix}=\begin{bmatrix}\mathrm{F}^T & 0\\ 0 & \mathrm{G}^T\end{bmatrix}\begin{bmatrix}\mathrm{R}_{xx}^{-1/2} & 0\\ 0 & \mathrm{R}_{yy}^{-1/2}\end{bmatrix}\begin{bmatrix}\mathbf{x}\\ \mathbf{y}\end{bmatrix} \tag{4}$$

where the elements of **u** are the canonical coordinates of **x** and the elements of **v** are the canonical coordinates of **y**. Correspondingly, the matrices

$$\mathrm{W}^T=\mathrm{F}^T\mathrm{R}_{xx}^{-1/2} \quad \text{and} \quad \mathrm{D}^T=\mathrm{G}^T\mathrm{R}_{yy}^{-1/2} \tag{5}$$

map **x** and **y** to their corresponding canonical coordinates **u** and **v**. Thus we may rewrite the canonical coordinate map of (4) as

$$\begin{bmatrix}\mathbf{u}\\ \mathbf{v}\end{bmatrix}=\begin{bmatrix}\mathrm{W}^T & 0\\ 0 & \mathrm{D}^T\end{bmatrix}\begin{bmatrix}\mathbf{x}\\ \mathbf{y}\end{bmatrix}. \tag{6}$$

The composite vector of canonical coordinates, $[\mathbf{u}^T\mathbf{v}^T]^T$ has covariance matrix

$$E\left[\begin{pmatrix}\mathbf{u}\\ \mathbf{v}\end{pmatrix}\begin{pmatrix}\mathbf{u}^T & \mathbf{v}^T\end{pmatrix}\right]=\begin{bmatrix}\mathrm{R}_{uu} & \mathrm{R}_{uv}\\ \mathrm{R}_{vu} & \mathrm{R}_{vv}\end{bmatrix} \\ =\begin{bmatrix}\mathrm{W}^T\mathrm{R}_{xx}\mathrm{W} & \mathrm{W}^T\mathrm{R}_{xy}\mathrm{D}\\ \mathrm{D}^T\mathrm{R}_{yx}\mathrm{W} & \mathrm{D}^T\mathrm{R}_{yy}\mathrm{D}\end{bmatrix}=\begin{bmatrix}\mathrm{I} & \mathrm{K}\\ \mathrm{K} & \mathrm{I}\end{bmatrix} \tag{7}$$

which indicates that the canonical coordinates **u** and **v** are individually white but diagonally cross-correlated. The canonical correlation matrix is the cross covariance matrix of **u** and **v** and is given by

$$E[\mathbf{u}\mathbf{v}^T]=\mathrm{K}=\mathrm{W}^T\mathrm{R}_{xy}\mathrm{D}=\mathrm{F}^T\mathrm{R}_{xx}^{-1/2}\mathrm{R}_{xy}\mathrm{R}_{yy}^{-T/2}\mathrm{G} \tag{8}$$

The diagonal elements of this matrix are arranged in descending order. Thus the first canonical coordinate pair $\{u_1,v_1\}$ has the largest correlation.

The conventional method of canonical coordinate decomposition, i.e. (4), requires the computation of the SVD of the coherence matrix $\mathrm{C}=\mathrm{R}_{xx}^{-1/2}\mathrm{R}_{xy}\mathrm{R}_{yy}^{-T/2}$ and the products $\mathrm{F}^T\mathrm{R}_{xx}^{-1/2}$ and $\mathrm{G}^T\mathrm{R}_{yy}^{-1/2}$. However, the major computational burden is not just the SVD. The procedure before and after the SVD involves computation of square-root-inverses that requires more than $O(\min(m^2n,n^2m))$ flops. These operations become computationally intractable and inefficient for large dimension data. This motivates our next discussion.

III. Canonical Coordinate Decomposition Network

This section presents a network and a set of updating rules to recursively extract the canonical coordinates of two data channels. The updating rules are derived so that no matrix inversion or square root computation is required. The network may be trained in either batch or sequential mode and thus may be used for online applications as well.

Let $\{u_i,v_i\}$ denote the ith pair of canonical coordinates of **x** and **y**. The canonical correlation associated with this pair is k_i, which is the ith diagonal element of the canonical correlation matrix K. Correspondingly the ith columns of matrices W and D, denoted by $\mathbf{w}_i\in R^{m\times 1}$ and $\mathbf{d}_i\in R^{n\times 1}$, are respectively, the mapping vectors that map **x** and **y** to their ith canonical coordinates u_i and v_i. The canonical correlation k_i is then $k_i=E[\mathbf{w}_i^T\mathbf{x}\mathbf{y}^T\mathbf{d}_i]=\mathbf{w}_i^T\mathrm{R}_{xy}\mathbf{d}_i$. We further refer to $\mathbf{w}_i$ and $\mathbf{d}_i$ as the ith canonical coordinate mappings.

Noting that $k_1=\mathbf{w}_1^T\mathrm{R}_{xy}\mathbf{d}_1$ is the largest canonical correlation, the problem of finding the first canonical coordinate mappings, $\mathbf{w}_1$ and $\mathbf{d}_1$, may be formulated as the maximization problem

$$\max_{\mathbf{w}_1,\mathbf{d}_1}\ \mathbf{w}_1^T\mathrm{R}_{xy}\mathbf{d}_1 \tag{9}$$

subject to the constraints

$$\mathbf{w}_1^T\mathrm{R}_{xx}\mathbf{w}_1=1 \quad \text{and} \quad \mathbf{d}_1^T\mathrm{R}_{yy}\mathbf{d}_1=1. \tag{10}$$

Using the method of Lagrange multipliers we may rewrite the constrained optimization problem defined by (9) and (10) as minimizing the objective function J_1 of the form

$$J_1=-\mathbf{w}_1^T\mathrm{R}_{xy}\mathbf{d}_1+(\mathbf{w}_1^T\mathrm{R}_{xx}\mathbf{w}_1-1)\tfrac{\lambda_{1,1}}{2} \\ +(\mathbf{d}_1^T\mathrm{R}_{yy}\mathbf{d}_1-1)\tfrac{\lambda_{1,2}}{2}, \tag{11}$$

where $\lambda_{1,1}$ and $\lambda_{1,2}$ are Lagrange multipliers that enforce the constraints in (10).

Now, assume that the first $r<m$ columns of W and D have already been found. Let $\mathrm{W}_r\in R^{m\times r}$ and $\mathrm{D}_r\in R^{n\times r}$ be the matrices that, respectively contain the first r columns of W and D. That is

$$\mathrm{W}_r=[\mathbf{w}_1,\ldots,\mathbf{w}_r] \quad \text{and} \quad \mathrm{D}_r=[\mathbf{d}_1,\ldots,\mathbf{d}_r]. \tag{12}$$

The first r canonical coordinates of **x** and **y** are then given by

$$\mathbf{u}_r=[u_1,\ldots,u_r]^T=\mathrm{W}_r^T\mathbf{x} \\ \mathbf{v}_r=[v_1,\ldots,v_r]^T=\mathrm{D}_r^T\mathbf{y} \tag{13}$$

By introducing deflation we may minimize the deflated version of J_1 to find the next canonical coordinate mappings. It can be shown [14] that the (r+1)th canonical coordinate pair of **x** and **y**, $\{u_{r+1},v_{r+1}\}$, is the first canonical coordinate pair of $(\mathrm{I}-\mathrm{R}_{xx}\mathrm{W}_r\mathrm{W}_r^T)\mathbf{x}$ and $(\mathrm{I}-\mathrm{R}_{yy}\mathrm{D}_r\mathrm{D}_r^T)\mathbf{y}$. Thus, the problem of finding the (r+1)th canonical coordinate mappings $\mathbf{w}_{r+1}$ and

$\mathbf{d}_{r+1}$, may be formulated in the context of (11) with $\mathbf{x}$ and $\mathbf{y}$ being replaced by their deflated versions $(\mathrm{I}-\mathrm{R}_{xx}\mathrm{W}_r\mathrm{W}_r^T)\mathbf{x}$ and $(\mathrm{I}-\mathrm{R}_{yy}\mathrm{D}_r\mathrm{D}_r^T)\mathbf{y}$. We may now write the problem of finding the canonical coordinate mappings $\mathbf{w}_{r+1}$ and $\mathbf{d}_{r+1}$ as minimizing the objective function

$$\begin{aligned} J_{r+1} = &-\mathbf{w}_{r+1}^T(\mathrm{I}-\mathrm{R}_{xx}\mathrm{W}_r\mathrm{W}_r^T)\mathrm{R}_{xy}(\mathrm{I}-\mathrm{R}_{yy}\mathrm{D}_r\mathrm{D}_r^T)\mathbf{d}_{r+1} \\ &+(\mathbf{w}_{r+1}^T\mathrm{R}_{xx}\mathbf{w}_{r+1}-1)\tfrac{\lambda_{r+1,1}}{2}+(\mathbf{d}_{r+1}^T\mathrm{R}_{yy}\mathbf{d}_{r+1}-1)\tfrac{\lambda_{r+1,2}}{2} \end{aligned} \tag{14}$$

where $\lambda_{r+1,1}$ and $\lambda_{r+1,2}$ are Lagrange multipliers that guarantee the unit variance property of the new pair of coordinates;

$$\mathbf{w}_{r+1}^T\mathrm{R}_{xx}\mathbf{w}_{r+1}=1 \quad \text{and} \quad \mathbf{d}_{r+1}^T\mathrm{R}_{yy}\mathbf{d}_{r+1}=1. \tag{15}$$

Taking the partial derivatives of J_c with respect to $\mathbf{w}_{r+1}$ and $\mathbf{d}_{r+1}$ yield

$$\begin{aligned} \tfrac{\partial J_{r+1}}{\partial \mathbf{W}_{r+1}} &= -(\mathrm{I}-\mathrm{R}_{xx}\mathrm{W}_r\mathrm{W}_r^T)\mathrm{R}_{xy}(\mathrm{I}-\mathrm{R}_{yy}\mathrm{D}_r\mathrm{D}_r^T)^T\mathbf{d}_{r+1} \\ &\qquad +\mathrm{R}_{xx}\mathbf{w}_{r+1}\lambda_{r+1,1} \\ \tfrac{\partial J_{r+1}}{\partial \mathbf{d}_{r+1}} &= -(\mathrm{I}-\mathrm{R}_{yy}\mathrm{D}_r\mathrm{D}_r^T)\mathrm{R}_{yx}(\mathrm{I}-\mathrm{R}_{xx}\mathrm{W}_r\mathrm{W}_r^T)^T\mathbf{w}_{r+1} \\ &\qquad +\mathrm{R}_{yy}\mathbf{d}_{r+1}\lambda_{r+1,2} \end{aligned} \tag{16}$$

At the solution the constraints in (15) are satisfied. Moreover, due to deflation, $\mathbf{w}_{r+1}$ and $\mathbf{d}_{r+1}$ are respectively, orthogonal to $\mathrm{R}_{xx}\mathrm{W}_r$ and $\mathrm{R}_{yy}\mathrm{D}_r$. That is

$$\mathbf{w}_{r+1}^T\mathrm{R}_{xx}\mathrm{W}_r=\mathbf{0} \quad \text{and} \quad \mathbf{d}_{r+1}^T\mathrm{R}_{yy}\mathrm{D}_r=\mathbf{0}. \tag{17}$$

Using (15) and (17) the optimal values of Lagrange multipliers in (14) are found to be

$$\lambda_{r+1}=\lambda_{r+1,1}=\lambda_{r+1,2}=\mathbf{w}_{r+1}^T\mathrm{R}_{xy}\mathbf{d}_{r+1} \tag{18}$$

From (6), the $(r+1)$th canonical coordinate pair of $\mathbf{x}$ and $\mathbf{y}$ is given by

$$\begin{aligned} u_{r+1} &= \mathbf{w}_{r+1}^T\mathbf{x} \\ v_{r+1} &= \mathbf{d}_{r+1}^T\mathbf{y} \end{aligned} \tag{19}$$

Using (13) and (17) we may rewrite (19) as

$$\begin{aligned} u_{r+1} &= \mathbf{w}_{r+1}^T(\mathrm{I}-\mathrm{R}_{xx}\mathrm{W}_r\mathrm{W}_r^T)\mathbf{x}=\mathbf{w}_{r+1}^T\mathbf{x}-\mathbf{q}_r^T\mathbf{u}_r \\ v_{r+1} &= \mathbf{d}_{r+1}^T(\mathrm{I}-\mathrm{R}_{yy}\mathrm{D}_r\mathrm{D}_r^T)\mathbf{y}=\mathbf{d}_{r+1}^T\mathbf{x}-\mathbf{p}_r^T\mathbf{v}_r \end{aligned} \tag{20}$$

where

$$\begin{aligned} \mathbf{q}_r^T &= \mathbf{w}_{r+1}^T\mathrm{R}_{xx}\mathrm{W}_r \\ \mathbf{p}_r^T &= \mathbf{d}_{r+1}^T\mathrm{R}_{yy}\mathrm{D}_r \end{aligned} \tag{21}$$

The pair of equations in (20) may be used to define a network structure for extracting the $(r+1)$th pair of canonical coordinates, given the first r pairs. Each equation in (20) defines a single layer subnetwork that features a feedforward set of weights from the input to the output and a set of lateral connections that connects the first r nodes to the $(r+1)$th node. Figure 1 shows the structure of this network. In this structure, W_r and D_r are the weight matrices that map $\mathbf{x}$ and $\mathbf{y}$ to their first r canonical coordinates $\mathbf{u}_r$ and $\mathbf{v}_r$. Given these weights, the network may be trained, by minimizing J_{r+1} in (14), to extract the $(r+1)$th canonical coordinate pair and the corresponding mappings. The weight vectors $\mathbf{w}_{r+1}$ and $\mathbf{d}_{r+1}$ are trained to maximize the correlation between the outputs u_{r+1} and v_{r+1} and make them unity variance. The lateral weight vector $\mathbf{q}_r$ is trained to orthogonalize $\mathbf{u}_r$ (the first r canonical coordinates of $\mathbf{x}$) to u_{r+1} (the $(r+1)$th canonical coordinate of $\mathbf{x}$.) Similarly, the lateral weight vector $\mathbf{p}_r$ is trained to orthogonalize $\mathbf{v}_r$ (the first r canonical coordinates of $\mathbf{y}$) to v_{r+1} (the $(r+1)$th canonical coordinate of $\mathbf{y}$.) The lateral connections perform a deflation process that subtracts the contributions of the already extracted coordinates from the linear subspaces of $\mathbf{x}$ and $\mathbf{y}$. This structures allows for adding new nodes for extracting additional canonical coordinates without the need for retraining the previous nodes.

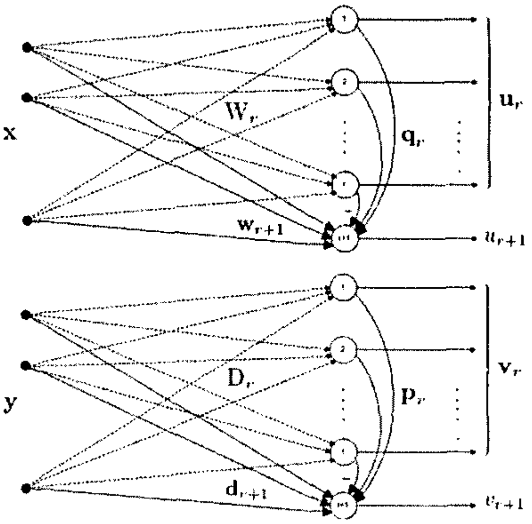


Fig. 1. The structure of the network for recursive extraction of canonical coordinates of **x** and **y**.

Using the stochastic gradient descent learning algorithm, with instantaneous values of covariance matrices inserted into (16), we may derive the following updating rules for $\mathbf{w}_{r+1}$, and $\mathbf{d}_{r+1}$:

$$\begin{aligned} \mathbf{w}_{r+1}(j+1) = \mathbf{w}_{r+1}(j) + [(&\mathbf{x}(j+1)-\mathrm{S}_r(j+1)\mathbf{u}_r(j+1))\cdot \\ &\cdot v_r(j+1)-\mathbf{x}(j+1)\mathbf{x}(j+1)^T\cdot \\ &\cdot\mathbf{w}_{r+1}(j)\lambda_{r+1}(j+1)]\beta(j+1) \\ \mathbf{d}_{r+1}(j+1) = \mathbf{d}_{r+1}(j) + [(&\mathbf{y}(j+1)-\mathrm{T}_r(j+1)\mathbf{v}_r(j+1))\cdot \\ &\cdot u_r(j+1)-\mathbf{y}(j+1)\mathbf{y}(j+1)^T\cdot \\ &\cdot\mathbf{d}_{r+1}(j)\lambda_{r+1}(j+1)]\beta(j+1) \end{aligned} \tag{22}$$

where j is the index of iteration. Matrices S_r and T_r are updated to asymptotically approximate $\mathrm{R}_{xx}\mathrm{W}_r$ and $\mathrm{R}_{yy}\mathrm{D}_r$ respectively. From (18), the Lagrange multiplier $\lambda_{r+1} = \lambda_{r+1,1} = \lambda_{r+1,2}$ shall be updated to asymptotically approximate $\mathbf{w}_{r+1}^T\mathrm{R}_{xy}\mathbf{d}_{r+1} = k_{r+1}$. Thus the updating rules for S_r, T_r and λ_{r+1} are

$$\begin{aligned} \mathrm{S}_r(j+1) &= \tfrac{j}{j+1}\mathrm{S}_r(j)+\tfrac{1}{j+1}\mathbf{x}(j+1)\mathbf{u}_r^T(j+1) \\ \mathrm{T}_r(j+1) &= \tfrac{j}{j+1}\mathrm{T}_r(j)+\tfrac{1}{j+1}\mathbf{y}(j+1)\mathbf{v}_r^T(j+1) \\ \lambda_{r+1}(j+1) &= \tfrac{j}{j+1}\lambda_{r+1}(j)+ \\ &\qquad \tfrac{1}{j+1}\mathbf{w}_{r+1}^T(j)\mathbf{x}(j+1)\mathbf{y}^T(j+1)\mathbf{d}_{r+1}(j) \end{aligned} \tag{23}$$

Finally using (21) the learning rules for the lateral weight vectors $\mathbf{q}_r$, and $\mathbf{p}_r$ may be written as

$$\begin{aligned}\mathbf{q}_r(j+1) &= \mathrm{S}_r^T(j+1)\mathbf{w}_{r+1}(j+1)\\ \mathbf{p}_r(j+1) &= \mathrm{T}_r^T(j+1)\mathbf{d}_{r+1}(j+1)\end{aligned} \tag{24}$$

Thus, we may summarize the training algorithm for extracting the (r+1)th canonical coordinate pair for $r = 0, 1, \ldots, m-1$ and the corresponding mapping vectors as

$$\begin{aligned}
\mathbf{u}_r(j+1) &= \mathrm{W}_r^T\mathbf{x}(j+1)\\
\mathbf{v}_r(j+1) &= \mathrm{D}_r^T\mathbf{y}(j+1)\\
u_{r+1}(j+1) &= \mathbf{w}_{r+1}^T(j+1)\mathbf{x}(j+1) - \mathbf{q}_r^T(j)\mathbf{u}_r(j+1)\\
v_{r+1}(j+1) &= \mathbf{d}_{r+1}^T(j+1)\mathbf{y}(j+1) - \mathbf{p}_r^T(j)\mathbf{v}_r(j+1)\\
\lambda_{r+1}(j+1) &= \tfrac{j}{j+1}\lambda_{r+1}(j) + \tfrac{1}{j+1}\mathbf{w}_{r+1}^T(j)\mathbf{x}(j+1)\mathbf{y}^T(j+1)\mathbf{d}_{r+1}(j)\\
\mathrm{S}_r(j+1) &= \tfrac{j}{j+1}\mathrm{S}_r(j) + \tfrac{1}{j+1}\mathbf{x}(j+1)\mathbf{u}_r^T(j+1)\\
\mathrm{T}_r(j+1) &= \tfrac{j}{j+1}\mathrm{T}_r(j) + \tfrac{1}{j+1}\mathbf{y}(j+1)\mathbf{v}_r^T(j+1)\\
\mathbf{w}_{r+1}(j+1) &= \mathbf{w}_{r+1}(j) + [(\mathbf{x}(j+1) - \mathrm{S}_r(j+1)\mathbf{u}_r(j+1))v_r(j+1)\\
&\qquad -\mathbf{x}(j+1)\mathbf{x}(j+1)^T\mathbf{w}_{r+1}(j)\lambda_{r+1}(j+1)]\beta(j+1)\\
\mathbf{d}_{r+1}(j+1) &= \mathbf{d}_{r+1}(j) + [(\mathbf{y}(j+1) - \mathrm{T}_r(j+1)\mathbf{v}_r(j+1))u_r(j+1)\\
&\qquad -\mathbf{y}(j+1)\mathbf{y}(j+1)^T\mathbf{d}_{r+1}(j)\lambda_{r+1}(j+1)]\beta(j+1)\\
\mathbf{q}_r(j+1) &= \mathrm{S}_r^T(j+1)\mathbf{w}_{r+1}(j+1)\\
\mathbf{p}_r(j+1) &= \mathrm{T}_r^T(j+1)\mathbf{d}_{r+1}(j+1)
\end{aligned} \tag{25}$$

The initial values $\mathbf{w}_{r+1}(0) \in R^{m\times 1}$, $\mathbf{d}_{r+1}(0) \in R^{n\times 1}$, $\mathbf{q}_r(0) \in R^{r\times 1}$, $\mathbf{p}_r(0) \in R^{r\times 1}$, $\mathrm{S}_r(0) \in R^{m\times r}$, $\mathrm{T}_r(0) \in R^{n\times r}$, and λ_{r+1} may be chosen randomly. The learning rate β may be varied or kept fixed [15]. It is important to note that, owing to the deflation performed by the lateral connections, the outputs within each subnetwork are decoupled. Thus during extraction of the (r+1)th pair of canonical coordinates there is no need to retrain the previous nodes and the weight matrices W_r and D_r are not changed.

IV. Simulation Results

In this section, the proposed network is used to recursively extract the canonical coordinate mappings for a synthesized data set. The performance of the network is demonstrated by presenting the plots of squared error between the actual canonical coordinate mappings, computed using the direct method in (5), and the ones estimated by the network, along with the plots of squared error for canonical correlations. Let $\hat{\mathbf{w}}_i$ and $\hat{\mathbf{d}}_i$, respectively, denote the estimate of the ith pair of the actual canonical coordinate mappings $\mathbf{w}_i$ and $\mathbf{d}_i$. We define $e^2_{\mathbf{w}_i}$ and $e^2_{\mathbf{d}_i}$ as the squared estimation error of the ith canonical coordinate mappings $\mathbf{w}_i$ and $\mathbf{d}_i$. That is,

$$e^2_{\mathbf{w}_i} = \|\mathbf{w}_i - \hat{\mathbf{w}}_i\|^2 \quad \text{and} \quad e^2_{\mathbf{d}_i} = \|\mathbf{d}_i - \hat{\mathbf{d}}_i\|^2$$

Also, $e^2_{\mathbf{k}_i} = (k_i - \hat{k}_i)^2$, is defined as the squared estimation error of the ith canonical correlation k_i. The actual canonical correlation k_i is found from the SVD in (3). From (18), it is seen that the ith canonical correlation k_i is estimated by the Lagrange multiplier λ_i. The data set is formed from 500 samples of two data channels governed by the linear model

$$\begin{aligned}\mathbf{x} &= H_x\eta_x\\ \mathbf{y} &= H_y\eta_y + H_{yx}\mathbf{x}\end{aligned}$$

where $\mathbf{x} \in R^{4\times 1}$, $\mathbf{y} \in R^{5\times 1}$. The matrices $H_x \in R^{4\times 4}$, $H_y \in R^{5\times 5}$ and $H_{yx} \in R^{5\times 4}$ are known matrices, and $\eta_x \in R^{4\times 1}$ and $\eta_y \in R^{5\times 1}$ are two independent white Gaussian vectors. The network is trained for 2500 epochs using the training algorithm in (25). The learning rate is varied linearly from $\beta = 5\times 10^{-3}$ to $\beta = 5\times 10^{-6}$ in 2500 steps. All the initial values in (25) are randomly selected.

Figure 2 shows the squared estimation errors $e^2_{\mathbf{w}_i}$, $i \in [1,4]$ vs. the epoch index for 10 independent initializations of the network. It is seen that in all the cases the squared error approaches zero within a misadjustment error and thus the weights of the upper subnetwork (Fig. 1) converge to the actual canonical coordinate mappings that map the first data channel $\mathbf{x}$ into its canonical coordinates $\mathbf{u}$.

The plots of the squared estimation errors $e^2_{\mathbf{d}_i}$, $i \in [1,4]$ vs. epoch index for the 10 initializations are shown in Fig. 3. The convergence behaviors are very similar to those in Fig. 2. It is seen that in all the cases the squared error approaches zero within a misadjustment error and thus the weights of the lower subnetwork (Fig. 1) converge to the actual canonical coordinate mappings that map the second data channel $\mathbf{y}$ into its canonical coordinates $\mathbf{v}$.

Figure 4 shows the squared estimation errors $e^2_{\mathbf{k}_i}$, $i \in [1,4]$ vs. the epoch index for the 10 initializations. The plots show that the squared error decays to zero in all the cases. The estimate of the ith canonical correlation is given by the Lagrange multiplier λ_i. These plots indicate that λ_i's converge to the actual canonical correlations k_i's in all the cases.

V. Conclusion

A new network for recursive extraction of canonical coordinates/correlations of two data channels is introduced. The network is based on a constrained minimization problem that exploits a deflation process. The deflation process is performed by incorporating lateral connections into the subnetworks. The learning rules are derived using a stochastic gradient descent algorithm. The structure of the network along with the learning rules allow for adding a new node to the network in order to extract a new pair of canonical coordinates without the need to retrain the previous nodes. Unlike conventional methods, no matrix inversion, matrix square root computation, direct SVD is required during the training. A simulation example demonstrates the validity of the proposed network and learning rules. The results confirm that the extracted canonical coordinate mappings approximate the true ones.

Acknowledgment

This work is supported by the Office of Naval Research (ONR) under the contract number N00014-02-1-0006.

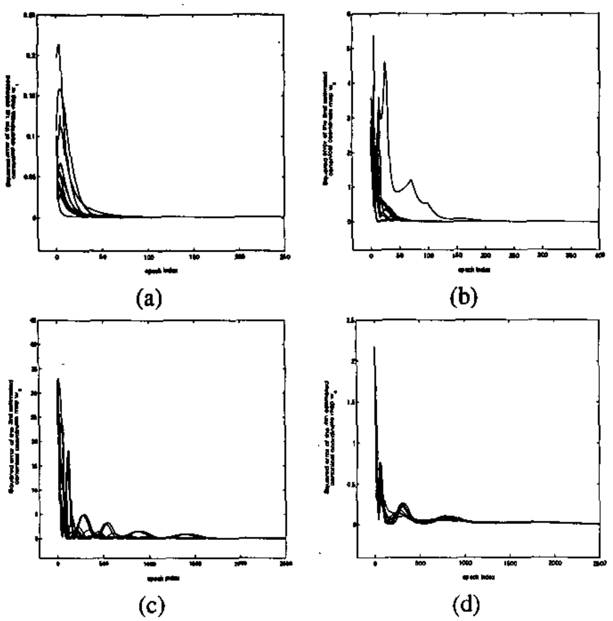
References

[1] H. Hotelling, "Relation between two sets of variates," *Biometrica*, vol. 28, pp. 321-377, 1936.

[2] T. W. Anderson, *An Introduction to Multivariate Statistical Analysis*, New York: Wiley, 1958.

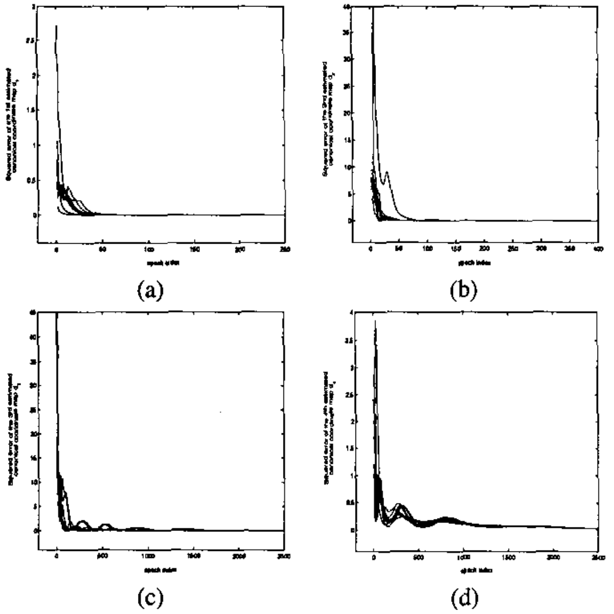
[3] L. L. Scharf and C.T. Mullis, "Canonical coordinates and the geometry of inference, rate and capacity," *IEEE Trans. on Signal Processing*, vol. 48, pp. 824-831, March 2000.

[4] L. L. Scharf and J. T. Thomas, "Wiener filters in canonical coordinates for transform coding, filtering, and quantizing," *IEEE Trans. on Signal Processing*, vol. 46, pp. 647-654, March 1998.



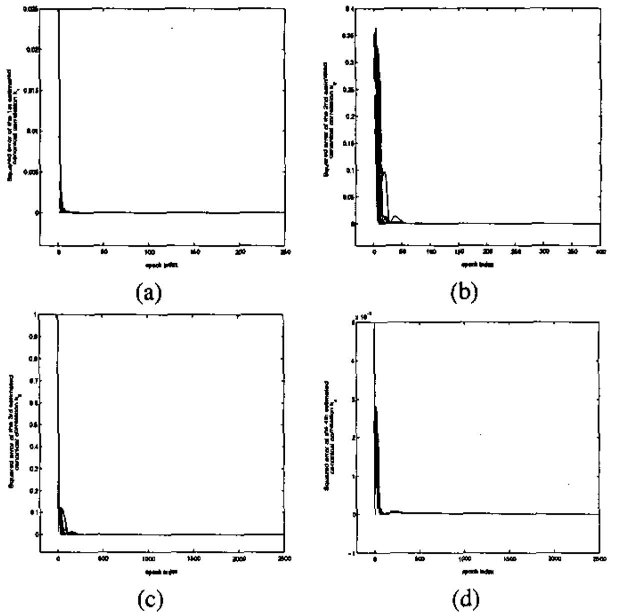
(a) (b) (c) (d)

Fig. 2. The squared error for $\mathbf{w}_i$'s, $i \in [1,4]$ vs. the epoch index for 10 independent initializations of the network (a) $i = 1$; $e^2_{\mathbf{w}_1} = \|\mathbf{w}_1 - \hat{\mathbf{w}}_1\|^2$. (b) $i = 2$; $e^2_{\mathbf{w}_2} = \|\mathbf{w}_2 - \hat{\mathbf{w}}_2\|^2$. (c) $i = 3$; $e^2_{\mathbf{w}_3} = \|\mathbf{w}_3 - \hat{\mathbf{w}}_3\|^2$. (d) $i = 4$; $e^2_{\mathbf{w}_4} = \|\mathbf{w}_4 - \hat{\mathbf{w}}_4\|^2$. In all the cases the squared error approaches zero and the weights of the upper subnetwork (Fig. 1) converge to the actual canonical coordinate mappings that map the first data channel $\mathbf{x}$ into its canonical coordinates $\mathbf{u}$.



(a) (b) (c) (d)

Fig. 3. The squared error for $\mathbf{d}_i$'s, $i \in [1,4]$ vs. the epoch index for 10 independent initializations of the network (a) $i = 1$; $e^2_{\mathbf{d}_1} = \|\mathbf{d}_1 - \hat{\mathbf{d}}_1\|^2$. (b) $i = 2$; $e^2_{\mathbf{d}_2} = \|\mathbf{d}_2 - \hat{\mathbf{d}}_2\|^2$. (c) $i = 3$; $e^2_{\mathbf{d}_3} = \|\mathbf{d}_3 - \hat{\mathbf{d}}_3\|^2$. (d) $i = 4$; $e^2_{\mathbf{d}_4} = \|\mathbf{d}_4 - \hat{\mathbf{d}}_4\|^2$. In all the cases the squared error approaches zero and the weights of the lower subnetwork (Fig. 1) converge to the actual canonical coordinate mappings that map the second data channel $\mathbf{y}$ into its canonical coordinates $\mathbf{v}$.



(a) (b) (c) (d)

Fig. 4. The squared error for k_i's, $i \in [1,4]$ vs. the epoch index for 10 independent initializations of the network (a) $i = 1$; $e^2_{k_1} = (k_1 - \hat{k}_1)^2$. (b) $i = 2$; $e^2_{k_2} = (k_2 - \hat{k}_2)^2$. (c) $i = 3$; $e^2_{k_3} = (k_3 - \hat{k}_3)^2$. (d) $i = 4$; $e^2_{k_4} = (k_4 - \hat{k}_4)^2$. The estimate of k_i is given by the Lagrange multiplier λ_i. The plots show that λ_i converges to the actual canonical correlation k_i in all the cases.

[5] K. I. Diamantaras and S. Y. Kung, *Principal Component Neural Networks: Theory and Applications*, 1st ed. John Wiely & Sons Inc. 1996.

[6] S. Bannour and M. R. Azimi-Sadjadi, "Principal component extraction using recursive least squares learning," *IEEE Trans. on Neural Networks*, vol. 6, pp. 457-469, March 1995.

[7] S. Y. Kung and K. I. Diamantaras, "Adaptive principal component extraction (APEX) and applications," *IEEE Trans. on Signal Processing*, vol. 42, pp. 1202-1217, May 1994.

[8] P. Baldi, and K. Hornik, "Neural networks and principal component analysis: Learning from examples without local minima," *Neural Networks*, vol. 2, pp. 53-58, 1989.

[9] T. D. Sanger, "Optimal unsupervised learning in a single-layer linear feedforward neural network," *Neural Networks*, vol. 2, pp. 459-473, 1989.

[10] E. Oja, "A simplified neuron model as principal component analyzer," *J. Math. Biology*, vol. 15, pp. 267-273, 1982.

[11] K. I. Diamantaras and S. Y. Kung, "Multi-layer neural networks for reduced-rank approximation," *IEEE Trans. Neural Networks*, vol. 5, pp. 684-697, Sept. 1994.

[12] L. L. Scharf, *Statistical Signal Processing*. Reading, MA: Addison-Wesley, 1991, pp. 330-331.

[13] P. L. Lai and C. Fyfe, "A neural network implementation of canonical correlation analysis," *Neural Networks*, vol. 12, pp. 1391-1397, 1999.

[14] A. Pezeshki, M. R. Azimi-Sadjadi, and L. L. Scharf, "A network for recusive extraction of canonical coordinates," *To appear in the Special Issue of Journal Neural Networks*, 2003.

[15] S. Haykin, *Neural Networks: A Comprehensive Foundation*. Upper Saddle River, NJ: Printice Hall, second edition, 1991.