PERGAMON

AVAILABLE AT
www.ComputerScienceWeb.com
POWERED BY SCIENCE DIRECT®

Neural Networks 16 (2003) 801–808

Neural Networks

www.elsevier.com/locate/neunet

2003 Special issue

A network for recursive extraction of canonical coordinates

Ali Pezeshki*, Mahmood R. Azimi-Sadjadi, Louis L. Scharf

Department of Electrical and Computer Engineering, Colorado State University, Fort Collins, CO 80523-1373, USA

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.

Keywords: Canonical coordinates; Singular value decomposition; Deflation; Iterative learning

1. Introduction

Canonical correlation analysis (Hotelling, 1936; Anderson, 1958) 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. The corresponding canonical coordinates resolve the channels into coordinates that are only pairwise correlated. 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 (Scharf & Thomas, 1998; Scharf & Mullis, 2000). 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 (Scharf & Mullis, 2000).

The conventional method of finding canonical coordinates involves computation of square-root-inverses of covariance matrices followed by a singular value decomposition (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 linear and nonlinear transformations. Several neural network-based approaches have been reported for extracting principal components of a stochastic vector process directly from the input data set. Oja (1982) proved that a linear model with a single node trained with a normalized Hebbian rule can extract the dominant principal component of a stationary vector process. Sanger (1989) and Foldiak (1989), extended Oja's work to the multi-node case in order to simultaneously extract the first m principal components of a vector process. Diamantaras and Kung (1994) exploited

* Corresponding author.
E-mail address: ali@engr.colostate.edu (A. Pezeshki).

0893-6080/03/$ - see front matter © 2003 Elsevier Science Ltd. All rights reserved.
doi:10.1016/S0893-6080(03)00112-6

the idea of using lateral connections with anti-Hebbian learning to recursively extract the principal components. In a different approach, based on recursive least squares (RLS) learning, Bannour and Azimi-Sadjadi (1995) proposed another structure for recursive extraction of principal components. Readers are referred to Diamantaras and Kung (1996) for a review of other related work in this area. Kung and Diamantaras (1994) also introduced a network structure for computing the reduced-rank Wiener filter. A network-based approach has been reported in Lai and Fyfe (1999) 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 includes a set of lateral connections that whiten the output. The structure of each subnetwork is similar to the structure of the single network proposed by Diamantaras and Kung (1994). 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. In these cases a test of linear dependence or mutual information may be performed to determine whether additional canonical coordinate pairs need to be extracted. A simulation example is given to demonstrate the validity of the proposed network and the corresponding learning rules.

2. A review on canonical coordinate decomposition

Let us follow the development of Scharf and Mullis (2000) and consider the two random vectors, $\mathbf{x} \in \mathbb{R}^{m\times 1}$ and $\mathbf{y} \in \mathbb{R}^{n\times 1}$ with m being the smaller dimension ($m \leq n$). Assume that $\mathbf{x}$ and $\mathbf{y}$ have zero means and share the composite covariance matrix

$$E\left[\begin{pmatrix}\mathbf{x}\\ \mathbf{y}\end{pmatrix}(\mathbf{x}^T \quad \mathbf{y}^T)\right] = \begin{bmatrix} R_{xx} & R_{xy} \\ R_{yx} & R_{yy}\end{bmatrix} \tag{1}$$

This composite covariance matrix has the following block tridiagonal decomposition

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

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

$$\begin{aligned} &C = R_{xx}^{-1/2}R_{xy}R_{yy}^{-T/2} = FKG^T \text{ and } F^TCG = K,\\ &F^TF = I,\ \ G^TG = I,\ \ K = \operatorname{diag}[k_1, k_2, \ldots, k_m]\end{aligned} \tag{3}$$

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 $\mathbf{x}$ and $\mathbf{y}$ are defined as

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

Correspondingly, the matrices

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

map $\mathbf{x}$ and $\mathbf{y}$ to their corresponding canonical coordinates $\mathbf{u}$ and $\mathbf{v}$. Thus we may rewrite the canonical coordinate map of Eq. (4) as

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

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

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

The canonical correlation matrix K is the cross covariance matrix of $\mathbf{u}$ and $\mathbf{v}$. That is

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

The squared canonical correlations k_i^2 decompose the linear dependence H between $\mathbf{x}$ and $\mathbf{y}$, which is measured by

the Hadamard ratio

$$H = \det\{I - KK^T\} = \prod_{i=1}^{m}(1 - k_i^2) \tag{9}$$

In the case where $\mathbf{x}$ and $\mathbf{y}$ are marginally Gaussian, the squared canonical correlation decompose the mutual information $I(\mathbf{x}, \mathbf{y})$ between $\mathbf{x}$ and $\mathbf{y}$

$$I(\mathbf{x},\mathbf{y}) = -\frac{1}{2}\log\det\{I - KK^T\} = -\frac{1}{2}\sum_{i=1}^{m}\log(1 - k_i^2) \tag{10}$$

The conventional method of canonical coordinate decomposition, i.e. Eq. (4), requires the computation of the SVD of the coherence matrix $C = R_{xx}^{-1/2}R_{xy}R_{yy}^{-T/2}$ and the products $F^T R_{xx}^{-1/2}$ and $G^T R_{yy}^{-1/2}$, which involve computation of matrix square-root-inverses. In addition, the conventional method does not allow for recursive extraction of a subset of canonical coordinates at a lower computational cost. These deficiencies of the conventional method motivate our next discussion.

3. A network for recursive extraction of canonical coordinates

This section presents a network structure 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 $\mathbf{w}_i \in \mathbb{R}^{m\times 1}$ and $\mathbf{d}_i \in \mathbb{R}^{n\times 1}$ denote the ith columns of W and D. Then, the ith canonical coordinates of $\mathbf{x}$ and $\mathbf{y}$ and their corresponding canonical correlation are

$$u_i = \mathbf{w}_i^T\mathbf{x},\ v_i = \mathbf{d}_i^T\mathbf{y}, \text{ and } k_i = E\{u_i v_i\} = \mathbf{w}_i^T R_{xy}\mathbf{d}_i \tag{11}$$

From here on, we refer to $\mathbf{w}_i$ and $\mathbf{d}_i$ as the ith canonical coordinate mapping vectors. From Eq. (7), we have

$$\begin{aligned} E[u_i u_j] &= \mathbf{w}_i^T R_{xx}\mathbf{w}_j = \delta(i-j) \\ E[v_i v_j] &= \mathbf{d}_i^T R_{yy}\mathbf{d}_j = \delta(i-j) \\ E[u_i v_j] &= \mathbf{w}_i^T R_{xy}\mathbf{d}_j = k_i\delta(i-j) \end{aligned} \tag{12}$$

where $\delta(\cdot)$ is the Kronecker delta. Noting that $k_1 = \mathbf{w}_1^T R_{xy}\mathbf{d}_1$ is the largest canonical correlation, the problem of finding the first canonical coordinate mapping vectors, $\mathbf{w}_1$ and $\mathbf{d}_1$, may be formulated as the maximization problem

$$\max_{\mathbf{w}_1,\mathbf{d}_1} \mathbf{w}_1^T R_{xy}\mathbf{d}_1 \tag{13}$$

subject to the constraints

$$\mathbf{w}_1^T R_{xx}\mathbf{w}_1 = 1 \text{ and } \mathbf{d}_1^T R_{yy}\mathbf{d}_1 = 1 \tag{14}$$

Using the method of Lagrange multipliers we may rewrite the constrained optimization problem defined by Eqs. (13) and (14) as minimizing the objective function J_1 of the form

$$\begin{aligned} J_1 = &-\mathbf{w}_1^T R_{xy}\mathbf{d}_1 + (\mathbf{w}_1^T R_{xx}\mathbf{w}_1 - 1)\frac{\lambda_{1,1}}{2} \\ &+ (\mathbf{d}_1^T R_{yy}\mathbf{d}_1 - 1)\frac{\lambda_{1,2}}{2} \end{aligned} \tag{15}$$

where $\lambda_{1,1}$ and $\lambda_{1,2}$ are Lagrange multipliers that enforce the constraints in Eq. (14).

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

$$W_r = [\mathbf{w}_1, \ldots, \mathbf{w}_r] \text{ and } D_r = [\mathbf{d}_1, \ldots, \mathbf{d}_r] \tag{16}$$

The first r canonical coordinates of $\mathbf{x}$ and $\mathbf{y}$ are then given by

$$\mathbf{u}_r = [u_1, \ldots, u_r]^T = W_r^T\mathbf{x} \text{ and } \mathbf{v}_r = [v_1, \ldots, v_r]^T = D_r^T\mathbf{y} \tag{17}$$

By deflating the contribution of the first r canonical coordinates $\mathbf{u}_r$ and $\mathbf{v}_r$ from the input channels, we may formulate the problem of finding the $(r+1)$th pair of canonical coordinates as one of finding the first canonical coordinate pair of the deflated input channels. This may be done by replacing R_{xy} in Eq. (15) with its deflated version $(I - R_{xx}W_r W_r^T)R_{xy}(I - R_{yy}D_r D_r^T)^T$ (See Appendix for the proof). Thus, the $(r+1)$th pair of canonical coordinate mapping vectors $\mathbf{w}_{r+1}$ and $\mathbf{d}_{r+1}$ may be found by minimizing the objective function

$$\begin{aligned} J_{r+1} = &-\mathbf{w}_{r+1}^T(I - R_{xx}W_r W_r^T)R_{xy}(I - R_{yy}D_r D_r^T)^T\mathbf{d}_{r+1} \\ &+ (\mathbf{w}_{r+1}^T R_{xx}\mathbf{w}_{r+1} - 1)\frac{\lambda_{r+1,1}}{2} \\ &+ (\mathbf{d}_{r+1}^T R_{yy}\mathbf{d}_{r+1} - 1)\frac{\lambda_{r+1,2}}{2} \end{aligned} \tag{18}$$

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,

$$\begin{aligned} E\{u_{r+1}^2\} &= \mathbf{w}_{r+1}^T R_{xx}\mathbf{w}_{r+1} = 1 \\ E\{v_{r+1}^2\} &= \mathbf{d}_{r+1}^T R_{yy}\mathbf{d}_{r+1} = 1 \end{aligned} \tag{19}$$

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

$$\begin{aligned} \frac{\partial J_{r+1}}{\partial \mathbf{w}_{r+1}} &= -(I - R_{xx}W_r W_r^T)R_{xy}(I - R_{yy}D_r D_r^T)^T\mathbf{d}_{r+1} \\ &\quad + R_{xx}\mathbf{w}_{r+1}\lambda_{r+1,1} \\ \frac{\partial J_{r+1}}{\partial \mathbf{d}_{r+1}} &= -(I - R_{yy}D_r D_r^T)R_{yx}(I - R_{xx}W_r W_r^T)^T\mathbf{w}_{r+1} \\ &\quad + R_{yy}\mathbf{d}_{r+1}\lambda_{r+1,2} \end{aligned} \tag{20}$$

At the solution the constraints in Eq. (19) are satisfied. Moreover, $\mathbf{w}_{r+1}$ and $\mathbf{d}_{r+1}$ are, respectively, orthogonal to $R_{xx}W_r$ and $R_{yy}D_r$. That is

$$\mathbf{w}_{r+1}^T R_{xx} W_r = 0 \text{ and } \mathbf{d}_{r+1}^T R_{yy} D_r = 0 \tag{21}$$

Using Eqs. (19) and (21) the optimal values of Lagrange multipliers in Eq. (18) are found to be

$$\lambda_{r+1,1} = \lambda_{r+1,2} = \lambda_{r+1} = \mathbf{w}_{r+1}^T R_{xy} \mathbf{d}_{r+1} = k_{r+1} \tag{22}$$

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

$$u_{r+1} = \mathbf{w}_{r+1}^T \mathbf{x} \text{ and } v_{r+1} = \mathbf{d}_{r+1}^T \mathbf{y} \tag{23}$$

Using Eqs. (17) and (21) we may rewrite Eq. (23) as

$$\begin{aligned} u_{r+1} &= \mathbf{w}_{r+1}^T (I - R_{xx} W_r W_r^T)\mathbf{x} = \mathbf{w}_{r+1}^T \mathbf{x} - \mathbf{q}_r^T \mathbf{u}_r \\ v_{r+1} &= \mathbf{d}_{r+1}^T (I - R_{yy} D_r D_r^T)\mathbf{y} = \mathbf{d}_{r+1}^T \mathbf{x} - \mathbf{p}_r^T \mathbf{v}_r \end{aligned} \tag{24}$$

where

$$\mathbf{q}_r^T = \mathbf{w}_{r+1}^T R_{xx} W_r \text{ and } \mathbf{p}_r^T = \mathbf{d}_{r+1}^T R_{yy} D_r \tag{25}$$

The pair of equations in Eq. (24) 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 Eq. (24) 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. Fig. 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

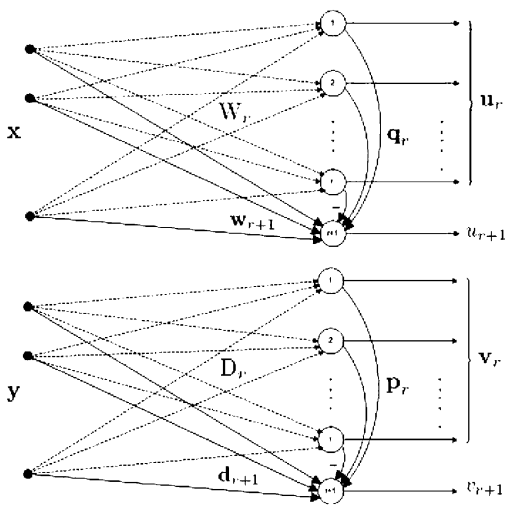

Fig. 1. The structure of the network for recursive extraction of canonical coordinates of $\mathbf{x}$ and $\mathbf{y}$.

weights, the network may be trained, by minimizing J_{r+1} in Eq. (18), to extract the $(r+1)$th canonical coordinate pair and the corresponding mapping vectors. 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 $\mathbf{v}_r$ to $\mathbf{v}_{r+1}$. 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 structure allows for adding new nodes for extracting additional canonical coordinates without the need for retraining the previous nodes.

Using the stochastic gradient descent learning algorithm, with instantaneous values of covariance matrices inserted into Eq. (20), we may derive the following updating rules for $\mathbf{w}_{r+1}$, and $\mathbf{d}_{r+1}$

$$\begin{aligned} \mathbf{w}_{r+1}(j+1) &= \mathbf{w}_{r+1}(j) + [(\mathbf{x}(j+1) - S_r(j+1)\mathbf{u}_r(j+1) \\ &\quad \cdot v_r(j+1) - \mathbf{x}(j+1)\mathbf{x}(j+1)^T \mathbf{w}_{r+1}(j)\lambda_{r+1} \\ &\quad \times (j+1)]\beta(j+1) \\ \mathbf{d}_{r+1}(j+1) &= \mathbf{d}_{r+1}(j) + [(\mathbf{y}_{(j+1)} - T_r(j+1)\mathbf{v}_r(j+1) \\ &\quad \cdot u_r(j+1) - \mathbf{y}(j+1)\mathbf{y}(j+1)^T \mathbf{d}_{r+1}(j)\lambda_{r+1} \\ &\quad \times (j+1)]\beta(j+1) \end{aligned} \tag{26}$$

where j is the index of iteration. Matrices S_r and T_r are updated to asymptotically approximate $R_{xx}W_r$ and $R_{yy}D_r$, respectively. From Eq. (22), the Lagrange multiplier $\lambda_{r+1} = \lambda_{r+1,1} = \lambda_{r+1,2}$ shall be updated to asymptotically approximate $\mathbf{w}_{r+1}^T R_{xy} \mathbf{d}_{r+1} = k_{r+1}$. Thus the updating rules for S_r, T_r, and λ_{r+1} are

$$\begin{aligned} S_r(j+1) &= \frac{j}{j+1} S_r(j) + \frac{1}{j+1} \mathbf{x}(j+1)\mathbf{u}_r^T(j+1) \\ T_r(j+1) &= \frac{j}{j+1} T_r(j) + \frac{1}{j+1} \mathbf{y}(j+1)\mathbf{v}_r^T(j+1) \\ \lambda_{r+1}(j+1) &= \frac{j}{j+1} \lambda_{r+1}(j) + \frac{1}{j+1} \mathbf{w}_{r+1}^T(j)\mathbf{x}(j+1) \\ &\quad \cdot \mathbf{y}^T(j+1)\mathbf{d}_{r+1}(j)] \end{aligned} \tag{27}$$

Finally using Eq. (25) the learning rules for the lateral weight vectors $\mathbf{q}_r$, and $\mathbf{p}_r$ may be written as

$$\begin{aligned} \mathbf{q}_r(j+1) &= S_r^T(j+1)\mathbf{w}_{r+1}(j+1) \\ \mathbf{p}_r(j+1) &= T_r^T(j+1)\mathbf{d}_{r+1}(j+1) \end{aligned} \tag{28}$$

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

vectors as

$$\mathbf{u}_r(j+1)=W_r^T\mathbf{x}(j+1) \text{ and } \mathbf{v}_r(j+1)=D_r^T\mathbf{y}(j+1)$$

$$u_{r+1}(j+1)=\mathbf{w}_{r+1}^T(j+1)\mathbf{x}(j+1)-\mathbf{q}_r^T(j)\mathbf{u}_r(j+1)$$

$$v_{r+1}(j+1)=\mathbf{d}_{r+1}^T(j+1)\mathbf{y}(j+1)-\mathbf{p}_r^T(j)\mathbf{v}_r(j+1)$$

$$\lambda_{r+1}(j+1)=\frac{1}{j+1}[j\lambda_{r+1}(j)+\mathbf{w}_{r+1}^T(j)\mathbf{x}(j+1)\mathbf{y}^T(j+1)\mathbf{d}_{r+1}(j)]$$

$$S_r(j+1)=\frac{1}{j+1}[jS_r(j)+\mathbf{x}(j+1)\mathbf{u}_r^T(j+1)]$$

$$T_r(j+1)=\frac{1}{j+1}[jT_r(j)+\mathbf{y}(j+1)\mathbf{v}_r^T(j+1)]$$

$$\mathbf{w}_{r+1}(j+1)=\mathbf{w}_{r+1}(j)+[(\mathbf{x}(j+1)-S_r(j+1)\mathbf{u}_r(j+1))\mathbf{v}_r(j+1)-\mathbf{x}(j+1)\mathbf{x}(j+1)^T\mathbf{w}_{r+1}(j)\lambda_{r+1}(j+1)]\beta(j+1)$$

$$\mathbf{d}_{r+1}(j+1)=\mathbf{d}_{r+1}(j)+[(\mathbf{y}(j+1)-T_r(j+1)\mathbf{v}_r(j+1))\mathbf{u}_r(j+1)-\mathbf{y}(j+1)\mathbf{y}(j+1)^T\mathbf{d}_{r+1}(j)\lambda_{r+1}(j+1)]\beta(j+1)$$

$$\mathbf{q}_r(j+1)=S_r^T(j+1)\mathbf{w}_{r+1}(j+1)$$

$$\mathbf{p}_r(j+1)=T_r^T(j+1)\mathbf{d}_{r+1}(j+1) \tag{29}$$

The initial values $\mathbf{w}_{r+1}(0)\in\mathbb{R}^{m\times 1}$, $\mathbf{d}_{r+1}(0)\in\mathbb{R}^{n\times 1}$, $\mathbf{q}_r(0)\in\mathbb{R}^{r\times 1}$, $\mathbf{p}_r(0)\in\mathbb{R}^{r\times 1}$, $S_r(0)\in\mathbb{R}^{m\times r}$, $T_r(0)\in\mathbb{R}^{n\times r}$, and λ_{r+1} may be chosen randomly. The learning rate β may be varied or kept fixed (Haykin, 1991). After convergence, the linear dependence captured by the first $(r+1)$ pair of canonical coordinates is

$$H_r=\prod_{i=1}^{r+1}(1-k_i^2)=\prod_{i=1}^{r+1}(1-\lambda_i^2) \tag{30}$$

and in case that $\mathbf{x}$ and $\mathbf{y}$ are marginally Gaussian, the mutual information preserved by the first $(r+1)$ pair of canonical coordinates becomes

$$I_r(\mathbf{x},\mathbf{y})=-\frac{1}{2}\sum_{i=1}^{r+1}\log(1-k_i^2)=-\frac{1}{2}\sum_{i=1}^{r+1}\log(1-\lambda_i^2) \tag{31}$$

Since in most applications, the number of canonical coordinate pairs to be extracted is not known a priori, we may run a test based on Eqs. (30) or (31) to determine if the amount of linear dependence or mutual information preserved, meets a pre-specified threshold. If the threshold is not reached, we may add another node to the network to extract the next pair of canonical coordinates.

4. 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 Eq. (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 $e_{\mathbf{w}_i}^2$ and $e_{\mathbf{d}_i}^2$ as the squared estimation error of the ith canonical coordinate mappings $\mathbf{w}_i$ and $\mathbf{d}_i$. That is

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

Also, $e_{\mathbf{k}_i}^2=(k_i-\hat{k}_i)^2$, 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 Eq. (3). From Eq. (22), 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 $\mathbf{x}\in\mathbb{R}^{4\times 1}$, $\mathbf{y}\in\mathbb{R}^{5\times 1}$ governed by the linear model

$$\mathbf{x}=H_x\eta_x \text{ and } \mathbf{y}=H_y\eta_y+H_{yx}\mathbf{x}$$

The matrices $H_x\in\mathbb{R}^{4\times 4}, H_y\in\mathbb{R}^{5\times 5}$ and $H_{yx}\in\mathbb{R}^{5\times 4}$ are used to synthesize $\mathbf{x}$ and $\mathbf{y}$ from $\eta_x\in\mathbb{R}^{4\times 1}$ and $\eta_y\in\mathbb{R}^{5\times 1}$, which are two independent white Gaussian vectors. The network is trained for 2500 epochs using the training algorithm in Eq. (29), without knowledge of the generating mechanism for $\mathbf{x}$ and $\mathbf{y}$. The learning rate is varied linearly from $\beta=5\times 10^{-3}$ to 5×10^{-6} in 2500 steps. All the initial values in Eq. (29) are randomly selected.

Fig. 2(a)–(d) show the squared estimation errors $e_{\mathbf{w}_i}^2, i\in[1,4]$ 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 (Haykin, 1996) and thus the weights of the upper subnetwork (Fig. 1) converge to the actual canonical coordinate mapping vectors that map the first data channel $\mathbf{x}$ into its canonical coordinates $\mathbf{u}$.

The plots of the squared estimation errors $e_{\mathbf{d}_i}^2, i\in[1,4]$ vs. epoch index for the 10 initializations are shown in Fig. 3. The convergence behaviors are very similar to those in Fig. 2. 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 mapping vectors that map the second data channel $\mathbf{y}$ into its canonical coordinates $\mathbf{v}$.

Fig. 4 shows the squared estimation errors $e_{\mathbf{k}_i}^2, i\in[1,4]$ 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 k_i is given by the Lagrange multiplier λ_i. These plots indicate that λ_i's converge to the actual canonical correlations k_i's.

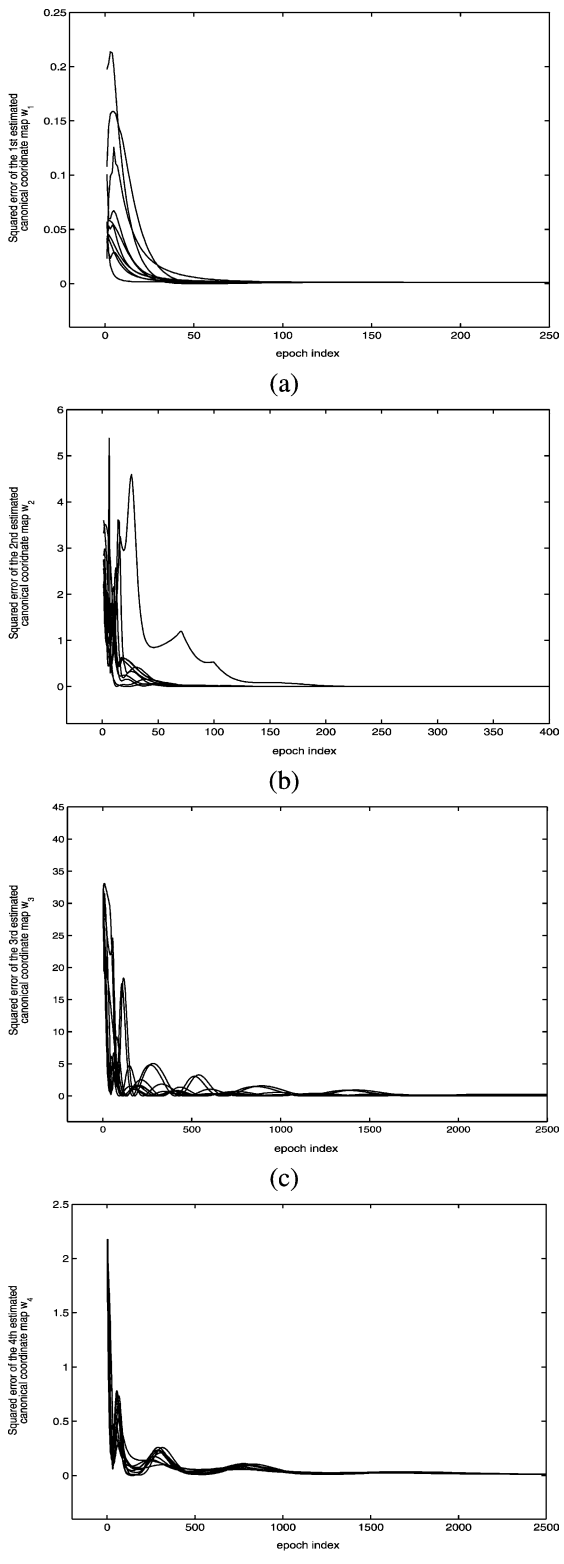

Fig. 2. The squared error for $\mathbf{w}_i$'s, $i \in [1,4]$ vs. the epoch index for 10 independent initializations of the network (a) $i = 1$; $e^2_{\mathbf{w}_1} = \|\mathbf{w}_1 - \hat{\mathbf{w}}_1\|^2$. (b) $i = 2$; $e^2_{\mathbf{w}_2} = \|\mathbf{w}_2 - \hat{\mathbf{w}}_2\|^2$. (c) $i = 3$; $e^2_{\mathbf{w}_3} = \|\mathbf{w}_3 - \hat{\mathbf{w}}_3\|^2$. (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 mapping vectors that map the first data channel $\mathbf{x}$ into its canonical coordinates $\mathbf{u}$.

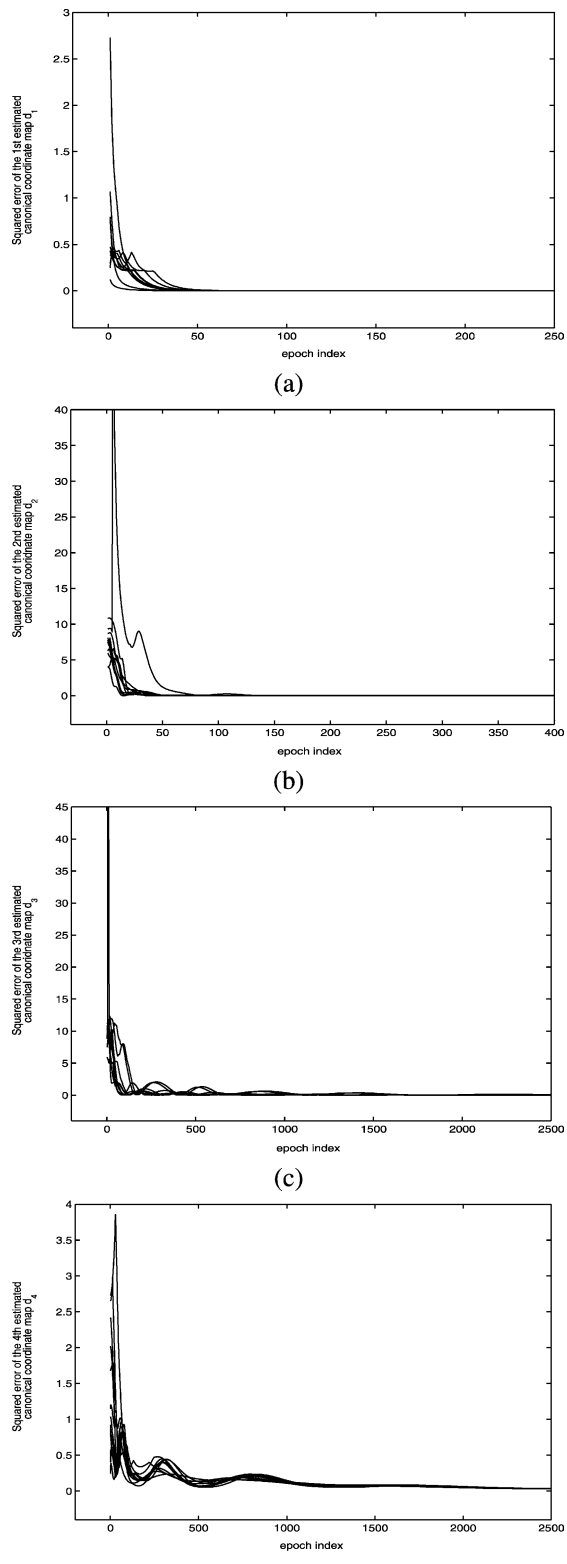

Fig. 3. The squared error for $\mathbf{d}_i$'s $i \in [1,4]$ vs. the epoch index for 10 independent initializations of the network (a) $i = 1$; $e^2_{\mathbf{d}_1} = \|\mathbf{d}_1 - \hat{\mathbf{d}}_1\|^2$. (b) $i = 2$; $e^2_{\mathbf{d}_2} = \|\mathbf{d}_2 - \hat{\mathbf{d}}_2\|^2$. (c) $i = 3$; $e^2_{\mathbf{d}_3} = \|\mathbf{d}_3 - \hat{\mathbf{d}}_3\|^2$. (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 mapping vectors that map the second data channel $\mathbf{y}$ into its canonical coordinates $\mathbf{v}$.

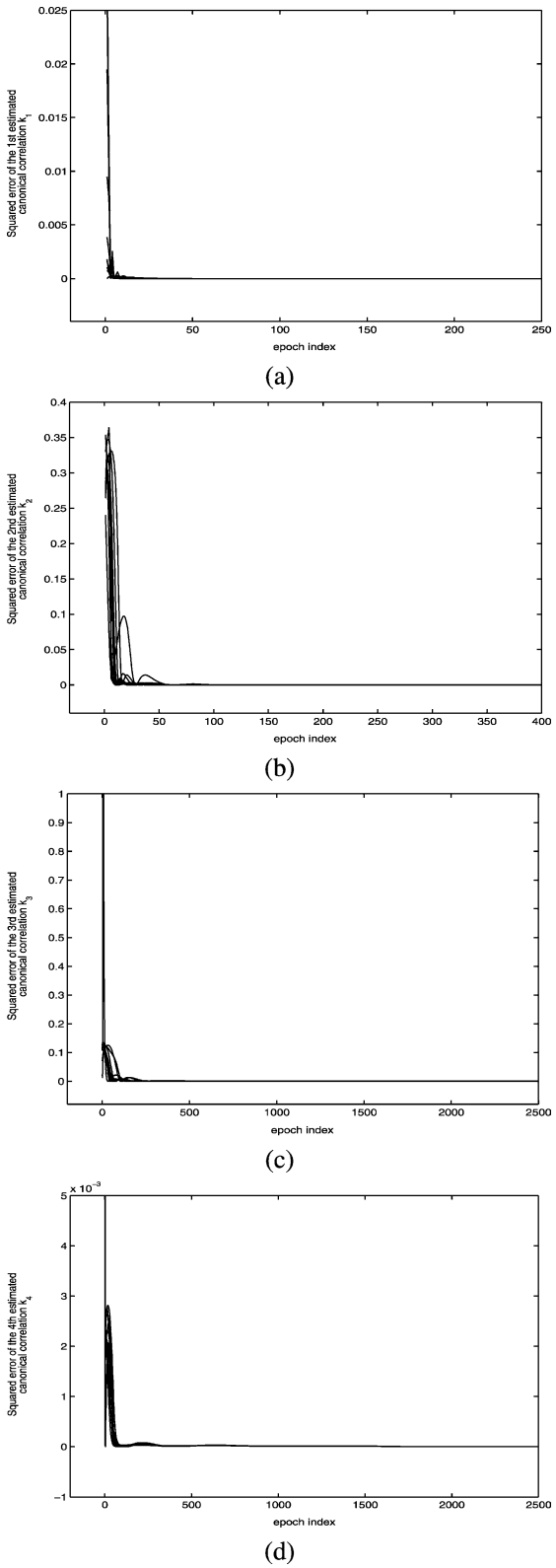

Fig. 4. The squared error for k_i's, $i \in [1,4]$ vs. the epoch index for 10 independent initializations of the network (a) $i=1$; $e^2_{k_1} = (k_1 - \hat{k}_1)^2$. (b) $i=2$; $e^2_{k_2} = (k_2 - \hat{k}_2)^2$. (c) $i=3$; $e^2_{k_3} = (k_3 - \hat{k}_3)^2$. (d) $i=4$; $e^2_{k_4} = (k_4 - \hat{k}_4)^2$. 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. Conclusion

A new network for recursive extraction of canonical coordinates 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 canonical coordinate pair without the need to retrain the previous nodes. Unlike conventional methods, no matrix inversion, matrix square root computation, or 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 indeed approximate the true ones.

Acknowledgements

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

Appendix A

Here we show that the constrained minimization problem in Eq. (18) indeed formulates the problem of finding the $(r+1)$th pair of canonical coordinate mappings $\mathbf{w}_{r+1}$ and $\mathbf{d}_{r+1}$. The assumption is that the matrices W_r and D_r, which contain the first r columns of W and D (the first r pair of canonical coordinate mappings), have already been found.

Let us start by partitioning F, G and K into

$$F = [F_r\ F_{\bar{r}}],\ \ G = [G_r\ G_{\bar{r}}],\ \ \text{and } K = \begin{bmatrix} K_r & 0 \\ 0 & K_{\bar{r}} \end{bmatrix} \tag{32}$$

where matrices F_r and G_r contain the first r and matrices $F_{\bar{r}}$ and $G_{\bar{r}}$ the last $m-r$ columns of F and G. The diagonal matrices $K_r = \text{diag}[k_1, \ldots, k_r]$ and $K_{\bar{r}} = \text{diag}[k_{r+1}, \ldots, k_m]$, respectively, contain the first r and last $m-r$ canonical correlations. Then from Eq. (3), we have

$$C = R_{xx}^{-1/2} R_{xy} R_{yy}^{-T/2} = FKG^T = F_r K_r G_r^T + F_{\bar{r}} K_{\bar{r}} G_{\bar{r}}^T \tag{33}$$

and

$$\begin{bmatrix} F_r^T F_r & F_{\bar{r}}^T F_r \\ F_r^T F_{\bar{r}} & F_{\bar{r}}^T F_{\bar{r}} \end{bmatrix} = \begin{bmatrix} I_r & 0 \\ 0 & I_{\bar{r}} \end{bmatrix}$$
$$\begin{bmatrix} G_r^T G_r & G_{\bar{r}}^T G_r \\ G_r^T G_{\bar{r}} & G_{\bar{r}}^T G_{\bar{r}} \end{bmatrix} = \begin{bmatrix} I_r & 0 \\ 0 & I_{\bar{r}} \end{bmatrix} \tag{34}$$

where I_r and $I_{\bar{r}}$ are the $r \times r$ and $(m-r) \times (m-r)$ identity matrices. The SVD in Eq. (33), may be rewritten as

$$R_{xy} = R_{xx}^{1/2} F_r K_r G_r^T R_{yy}^{T/2} + R_{xx}^{1/2} F_{\bar{r}} K_{\bar{r}} G_{\bar{r}}^T R_{yy}^{T/2} \tag{35}$$

Using Eqs. (33) and (34), it may easily be verified that the first term on the right hand side of Eq. (35) has three equivalent representations. That is

$$\begin{aligned} R_{xx}^{1/2} F_r K_r G_r^T R_{yy}^{T/2} &= R_{xx}^{1/2} F_r F_r^T R_{xx}^{-1/2} R_{xy} \\ &= R_{xy} R_{yy}^{-T/2} G_r G_r^T R_{yy}^{T/2} \\ &= R_{xx}^{1/2} F_r F_r^T R_{xx}^{-1/2} R_{xy} R_{yy}^{-T/2} G_r G_r^T R_{yy}^{T/2} \end{aligned} \tag{36}$$

Using this property, we may rewrite Eq. (35) as

$$\begin{aligned} &(I - R_{xx}^{1/2} F_r F_r^T R_{xx}^{-1/2}) R_{xy} (I - R_{yy}^{1/2} G_r G_r^T R_{yy}^{-1/2})^T \\ &\quad = R_{xx}^{1/2} F_{\bar{r}} K_{\bar{r}} G_{\bar{r}}^T R_{yy}^{T/2} \end{aligned} \tag{37}$$

We now partition W and D into $W = [W_r \; W_{\bar{r}}]$ and $D = [D_r \; D_{\bar{r}}]$, where the matrices $W_{\bar{r}} = [\mathbf{w}_{r+1}, \ldots, \mathbf{w}_m]$ and $D_{\bar{r}} = [\mathbf{d}_{r+1}, \ldots, \mathbf{d}_m]$ contain the last $m-r$ columns of W and D. Then, from Eq. (5) we have

$$\begin{aligned} &F_r = R_{xx}^{T/2} W_r, \quad F_{\bar{r}} = R_{xx}^{T/2} W_{\bar{r}}, \\ &G_r = R_{yy}^{T/2} D_r, \quad G_{\bar{r}} = R_{yy}^{T/2} D_{\bar{r}} \end{aligned} \tag{38}$$

Using Eq. (38), we may rewrite Eq. (37) as

$$(I - R_{xx} W_r W_r^T) R_{xy} (I - R_{yy} D_r D_r^T)^T = R_{xx} W_{\bar{r}} K_{\bar{r}} D_{\bar{r}}^T R_{yy} \tag{39}$$

Pre-multiplying Eq. (39) by $\mathbf{w}_{r+1}^T$, post-multiplying it by $\mathbf{d}_{r+1}$, and noting that $\mathbf{w}_{r+1}^T R_{xx} W_{\bar{r}} = [1, 0, \ldots, 0]$ and $\mathbf{d}_{r+1}^T R_{yy} D_{\bar{r}} = [1, 0, \ldots, 0]$ results in

$$\mathbf{w}_{r+1}^T (I - R_{xx} W_r W_r^T) R_{xy} (I - R_{yy} D_r D_r^T)^T \mathbf{d}_{r+1} = k_{r+1} \tag{40}$$

Considering that k_{r+1} is the largest diagonal element of $K_{\bar{r}}$ we may formulate the problem of finding $\mathbf{w}_{r+1}$, $\mathbf{d}_{r+1}$ as

$$\max_{\mathbf{w}_{r+1}, \mathbf{d}_{r+1}} \mathbf{w}_{r+1}^T (I - R_{xx} W_r W_r^T) R_{xy} (I - R_{yy} D_r D_r^T)^T \mathbf{d}_{r+1}$$

subject to the constraints

$$\mathbf{w}_{r+1}^T R_{xx} \mathbf{w}_{r+1} = 1 \text{ and } \mathbf{d}_{r+1}^T R_{yy} \mathbf{d}_{r+1} = 1.$$

References

Anderson, T. W. (1958). *An introduction to multivariate statistical analysis*. New York: Wiley.

Bannour, S., & Azimi-Sadjadi, M. R. (1995). Principal component extraction using recursive least squares learning. *IEEE Transactions on Neural Networks*, *6*, 457–469.

Diamantaras, D. I., & Kung, S. Y. (1994). Multi-layer neural networks for reduced-rank approximation. *IEEE Transactions on Neural Networks*, *5*, 684–697.

Diamantaras, K. I., & Kung, S. Y. (1996). *Principal component neural networks: theory and applications* (1st ed). New York: Wiely.

Foldiak, P. (1989). Adaptive network for optimal linear feature extraction. *Proceedings of the International Joint Conference on Neural Networks, IJCNN', Washington, DC*, *89*, I401–I406.

Haykin, S. (1991). *Neural networks: a comprehensive foundation* (2nd ed). Upper Saddle River, NJ: Printice Hall.

Haykin, S. (1996). *Adaptive filter theory* (3rd ed). Upper Saddle River, NJ: Printice Hall.

Hotelling, H. (1936). Relation between two sets of variates. *Biometrica*, *28*, 321–377.

Kung, S. Y., & Diamantaras, K. I. (1994). Adaptive principal component extraction (apex) and applications. *IEEE Transactions on Signal Processing*, *42*, 1202–1217.

Lai, P. L., & Fyfe, C. (1999). A neural network implementation of canonical correlation analysis. *Neural Networks*, *12*, 1391–1397.

Oja, E. (1982). A simplified neuron model as principal component analyzer. *Journal of Mathematical Biology*, *15*, 267–273.

Sanger, T. D. (1989). Optimal unsupervised learning in a single-layer linear feedforward neural network. *Neural Networks*, *2*, 459–473.

Scharf, L. L., & Mullis, C. T. (2000). Canonical coordinates and the geometry of inference, rate and capacity. *IEEE Transactions on Signal Processing*, *48*, 824–831.

Scharf, L. L., & Thomas, J. T. (1998). Wiener filters in canonical coordinates for transform coding, filtering, and quantizing. *IEEE Transactions on Signal Processing*, *46*, 647–654.