Available online at www.sciencedirect.com

ScienceDirect

Neural Networks 20 (2007) 139–152

Neural Networks

www.elsevier.com/locate/neunet

A learning algorithm for adaptive canonical correlation analysis of several data sets☆

Javier Vía*, Ignacio Santamaría, Jesús Pérez

Department of Communications Engineering, University of Cantabria, 39005 Santander, Cantabria, Spain

Received 4 January 2006; accepted 7 September 2006

Abstract

Canonical correlation analysis (CCA) is a classical tool in statistical analysis to find the projections that maximize the correlation between two data sets. In this work we propose a generalization of CCA to several data sets, which is shown to be equivalent to the classical maximum variance (MAXVAR) generalization proposed by Kettenring. The reformulation of this generalization as a set of coupled least squares regression problems is exploited to develop a neural structure for CCA. In particular, the proposed CCA model is a two layer feedforward neural network with lateral connections in the output layer to achieve the simultaneous extraction of all the CCA eigenvectors through deflation. The CCA neural model is trained using a recursive least squares (RLS) algorithm. Finally, the convergence of the proposed learning rule is proved by means of stochastic approximation techniques and their performance is analyzed through simulations.

Keywords: Canonical correlation analysis (CCA); Recursive least squares (RLS); Principal component analysis (PCA); Adaptive algorithms; Hebbian learning

1. Introduction

Canonical correlation analysis (CCA) is a well-known technique in multivariate statistical analysis to find maximally correlated projections between two data sets, which has been widely used in economics, meteorology and in many modern information processing fields, such as communication theory (Dogandzic & Nehorai, 2002), statistical signal processing (Dogandzic & Nehorai, 2003), independent component analysis (Bach & Jordan, 2002) and blind source separation (Friman, Borga, Lundberg, & Knutsson, 2003). CCA was developed by Hotelling (1936) as a way of measuring the linear relationship between two multidimensional sets of variables and was later extended to several data sets by Kettenring (1971). Typically, CCA is formulated as a generalized eigenvalue (GEV) problem; however, a direct application of eigendecomposition techniques is often unsuitable for high dimensional data sets as well as for adaptive environments due to their high computational cost.

Recently, several neural networks for CCA have been proposed: for instance, in Lai and Fyfe (1999), the authors present a linear feedforward network that maximizes the correlation between its two outputs. This structure was later improved (Gou & Fyfe, 2004) and generalized to nonlinear and kernel CCA (Lai & Fyfe, 2000). Other approaches from a neural network point of view to find nonlinear correlations between two data sets have been proposed by Hsieh (2000) and Hardoon, Szedmak, and Shawe-Taylor (2004).

Although CCA of several data sets has received increasing interest (Hardoon et al., 2004), adaptive CCA algorithms have been mainly proposed for the case of $M = 2$ data sets (Pezeshki, Azimi-Sadjadi, & Scharf, 2003; Pezeshki, Scharf, Azimi-Sadjadi, & Hua, 2005; Vía, Santamaría, & Pérez, 2005a), with the exception of the three data sets example presented by Lai and Fyfe (1999). Specifically, Pezeshki et al. (2003) present a network structure for CCA of $M = 2$ data sets, which is trained by means of a stochastic gradient descent algorithm. A similar architecture was presented in Pezeshki et al. (2005), but the training was carried out by means of power methods. The heuristic generalization to three data sets proposed by Lai and Fyfe (1999) is also based on

☆ This work was supported by MEC (Ministerio de Educación y Ciencia) under grant TEC2004-06451-C05-02 and FPU grant AP-2004-5127.

* Corresponding author. Tel.: +34 942201392 13; fax: +34 942201488.
E-mail addresses: jvia@gtas.dicom.unican.es (J. Vía), nacho@gtas.dicom.unican.es (I. Santamaría), jperez@gtas.dicom.unican.es (J. Pérez).

0893-6080/$ - see front matter © 2006 Elsevier Ltd. All rights reserved.
doi:10.1016/j.neunet.2006.09.011

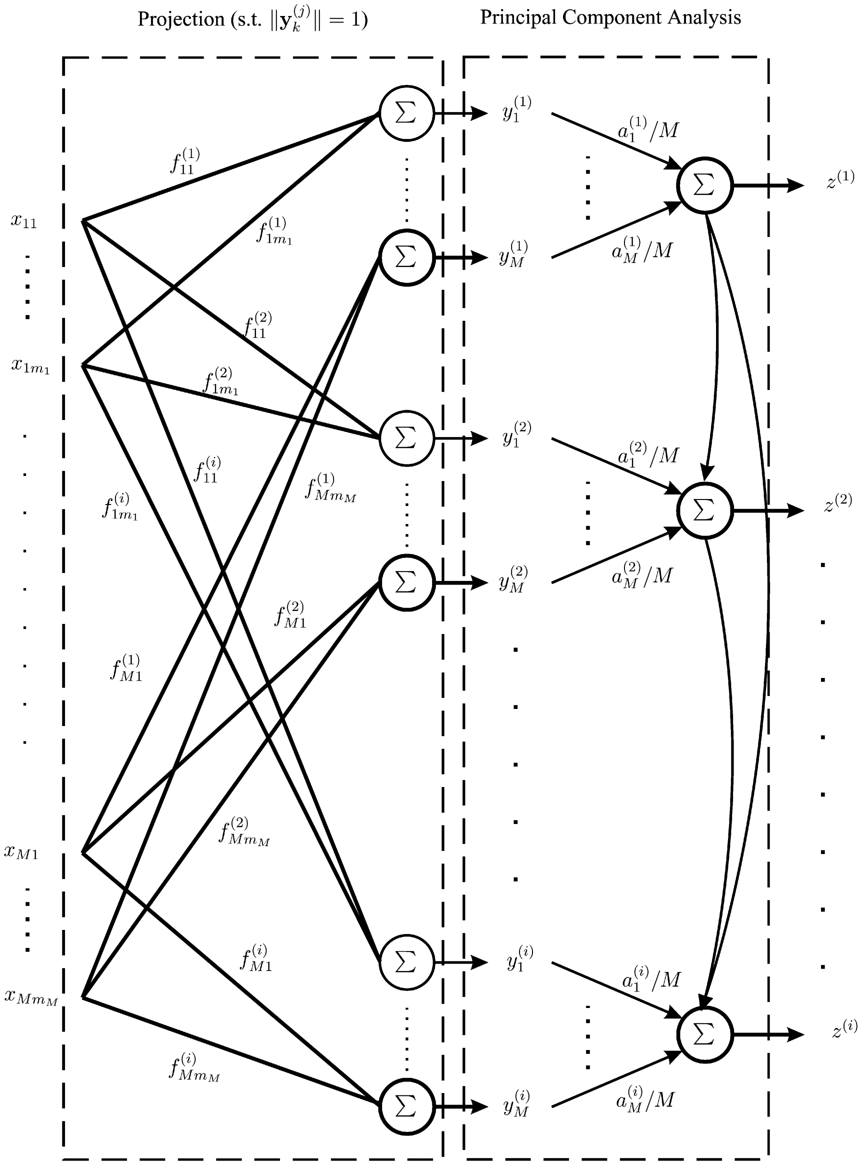

Fig. 1. Architecture of the CCA network. Classical PCA-based formulation.

a gradient descent algorithm and it seems to be unconnected with the classical generalizations to several data sets proposed by Kettenring (1971). In Vía et al. (2005a), we have exploited the reformulation of CCA of two data sets as a pair of LS regression problems to derive a recursive least squares (RLS) algorithm for adaptive CCA. We have also extended this algorithm to the case of several data sets in Vía, Santamaría, and Pérez (2005b).

In this paper we extend the work in Vía et al. (2005a, 2005b), propose a new adaptive learning algorithm, and prove its convergence by means of stochastic approximation techniques. Although derived in a different way, it can be proved that the CCA generalization considered in this paper is equivalent to the maximum variance (MAXVAR) CCA generalization proposed by Kettenring (1971). Interestingly, this CCA generalization is closely related to the principal component analysis (PCA) method and, in fact, it becomes PCA when the data sets are one-dimensional.

The classical formulation proposed by Kettenring (1971) suggests to implement CCA–MAXVAR through a two-layer network (see Fig. 1) where the first layer performs a constrained projection of the input data and the second layer is similar to the architecture used in the adaptive principal component extraction (APEX) learning network (Diamantaras & Kung, 1996): a feedforward network with lateral connections for adaptive deflation. Unfortunately, since the projection performed by the fist layer must be optimal in the sense that it admits the best PCA representation, the training of this network is not easy. On the other hand, the proposed CCA generalization admits a similar neural model (see Fig. 2), but now the second layer has fixed weights, which is a key difference to develop CCA–MAXVAR adaptive learning algorithms.

The reformulation of CCA as a set of coupled LS regression problems in the case of several data sets is exploited to propose an iterative LS procedure (batch) for the training of the network connection weights. Furthermore, this LS regression framework

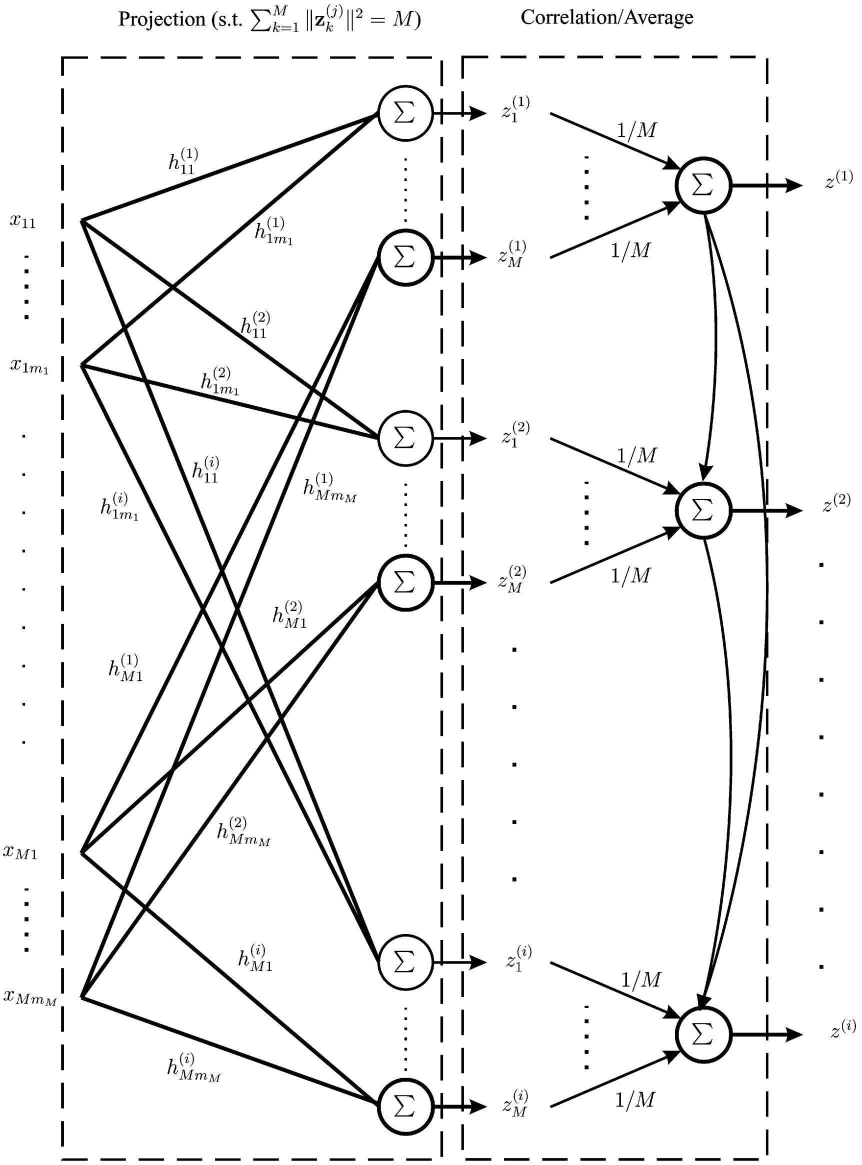

Fig. 2. Architecture of the CCA Network. Proposed formulation.

allows us to develop an adaptive RLS-based CCA algorithm. The main advantage of the new adaptive algorithm presented in this work in comparison to the power method procedure proposed by Pezeshki et al. (2005), is that the RLS-based algorithm avoids the previous estimation of the $M(M-1)/2$ cross-correlation matrices between each pair of data sets, where M is the total number of data sets. For high dimensional data or when the number of data sets is large, this can be a great advantage in terms of computational cost.

The paper is structured as follows: Sections 2 and 3 review, respectively, the CCA problem for $M = 2$ data sets and the MAXVAR generalization to several data sets. In Section 4 the proposed CCA generalization to $M > 2$ data sets is presented, and we prove that it is equivalent to the PCA-based MAXVAR generalization proposed by Kettenring. Sections 5 and 6 present, respectively, the new batch and adaptive learning algorithms for CCA. The convergence properties of the adaptive algorithm are analyzed in Section 7, and a brief comparison with other adaptive CCA techniques is presented in Section 8. Finally, the simulation results are presented in Section 9, and the main conclusions are summarized in Section 10.

Throughout the paper, the following notations are adopted:

$(\cdot)^{\mathrm{T}}$	Transpose
$(\cdot)^{\mathrm{H}}$	Conjugate transpose
$(\cdot)^*$	Complex conjugate
$(\cdot)^+$	Pseudoinverse
$\|\cdot\|$	Frobenius norm
$\mathbf{I}$	Identity matrix
$\mathbf{0}$	All-zero matrix
$\mathrm{Tr}(\cdot)$	Matrix trace
$E[\cdot]$	Statistical expectation

2. Review of canonical correlation analysis of two data sets

In this section, the classical two data set formulation of CCA is presented. The associated structure is a linear feedforward network with lateral connections for deflation, and it can be adaptively trained by means of the algorithms proposed in Gou and Fyfe (2004), Lai and Fyfe (1999) and Pezeshki et al. (2003, 2005) or with the new technique presented in Section 6 (see also Vía et al. (2005a)).

2.1. Main CCA solution

Let $\mathbf{X}_1 \in \mathbb{R}^{N \times m_1}$ and $\mathbf{X}_2 \in \mathbb{R}^{N \times m_2}$ be two known full-rank data matrices. Canonical correlation analysis (CCA) can be defined as the problem of finding two canonical vectors: $\mathbf{h}_1$ of size $m_1 \times 1$ and $\mathbf{h}_2$ of size $m_2 \times 1$, such that the canonical variates $\mathbf{z}_1 = \mathbf{X}_1\mathbf{h}_1$ and $\mathbf{z}_2 = \mathbf{X}_2\mathbf{h}_2$ are maximally correlated, i.e.,

$$\operatorname*{argmax}_{\mathbf{h}_1,\mathbf{h}_2} \rho = \frac{\mathbf{z}_1^{\mathrm{T}}\mathbf{z}_2}{\|\mathbf{z}_1\|\|\mathbf{z}_2\|} = \frac{\mathbf{h}_1^{\mathrm{T}}\mathbf{R}_{12}\mathbf{h}_2}{\sqrt{\mathbf{h}_1^{\mathrm{T}}\mathbf{R}_{11}\mathbf{h}_1\mathbf{h}_2^{\mathrm{T}}\mathbf{R}_{22}\mathbf{h}_2}}, \tag{1}$$

where $\mathbf{R}_{kl} = \mathbf{X}_k^{\mathrm{T}}\mathbf{X}_l$ is an estimate of the cross-correlation matrix. Problem (1) is equivalent to the maximization of

$$\rho = \mathbf{h}_1^{\mathrm{T}}\mathbf{R}_{12}\mathbf{h}_2,$$

subject to the constraints

$$\mathbf{h}_1^{\mathrm{T}}\mathbf{R}_{11}\mathbf{h}_1 = \mathbf{h}_2^{\mathrm{T}}\mathbf{R}_{22}\mathbf{h}_2 = 1. \tag{2}$$

The solution to this problem is given by the eigenvector corresponding to the largest eigenvalue of the following generalized eigenvalue problem (GEV) (Borga, 1998)

$$\begin{bmatrix} \mathbf{0} & \mathbf{R}_{12} \\ \mathbf{R}_{21} & \mathbf{0} \end{bmatrix} \mathbf{h} = \rho \begin{bmatrix} \mathbf{R}_{11} & \mathbf{0} \\ \mathbf{0} & \mathbf{R}_{22} \end{bmatrix} \mathbf{h},$$

where ρ is the canonical correlation and $\mathbf{h} = [\mathbf{h}_1^{\mathrm{T}}, \mathbf{h}_2^{\mathrm{T}}]^{\mathrm{T}}$ is the eigenvector.

2.2. Remaining CCA solutions

In order to determine additional CCA solutions a series of optimization problems are solved successively. Denoting the i-th canonical vectors, variables, and correlations as $\mathbf{h}_1^{(i)}, \mathbf{h}_2^{(i)}, \mathbf{z}_1^{(i)} = \mathbf{X}_1\mathbf{h}_1^{(i)}, \mathbf{z}_2^{(i)} = \mathbf{X}_2\mathbf{h}_2^{(i)}$ and $\rho^{(i)}$, respectively; and defining $\mathbf{z}^{(i)} = \frac{1}{2}(\mathbf{z}_1^{(i)} + \mathbf{z}_2^{(i)})$ the following orthogonality constraint is imposed for $i \neq j$

$$\mathbf{z}^{(i)\mathrm{T}}\mathbf{z}^{(j)} = 0,$$

which, for the two data set case also implies

$$\mathbf{z}_k^{(i)\mathrm{T}}\mathbf{z}_l^{(j)} = 0, \quad k, l = 1, 2. \tag{3}$$

For each new CCA solution the following GEV problem is obtained

$$\begin{bmatrix} \mathbf{0} & \mathbf{R}_{12} \\ \mathbf{R}_{21} & \mathbf{0} \end{bmatrix} \mathbf{h}^{(i)} = \rho^{(i)} \begin{bmatrix} \mathbf{R}_{11} & \mathbf{0} \\ \mathbf{0} & \mathbf{R}_{22} \end{bmatrix} \mathbf{h}^{(i)}, \tag{4}$$

where $\rho^{(i)}$ is the i-th canonical correlation and $\mathbf{h}^{(i)} = [\mathbf{h}_1^{(i)\mathrm{T}}, \mathbf{h}_2^{(i)\mathrm{T}}]^{\mathrm{T}}$ is the associated eigenvector.

3. MAXVAR generalization of CCA to several data sets

In this section the classical maximum variance (MAXVAR) CCA generalization proposed by Kettenring (1971) is summarized. The network structure associated to this generalization is a two-layer feedforward network where the first layer performs a constrained projection of the input data and the second layer is a PCA network (see Fig. 1). The lateral connections of the PCA layer impose the orthogonality constraints among the output variables, similarly to the APEX network (Diamantaras & Kung, 1996). Here we must point out that, excluding the case of $M = 2$ data sets,[1] it is not evident how to train this neural model in an adaptive fashion, since the projection performed by the first layer must be optimal in the sense that it admits the best PCA representation. Therefore both layers must be trained simultaneously.

3.1. Main CCA–MAXVAR solution

Given M data sets $\mathbf{X}_k \in \mathbb{R}^{N \times m_k}$, $k = 1, \ldots, M$, the MAXVAR CCA generalization can be stated as the problem of finding a set of M vectors $\mathbf{f}_k$ and the corresponding projections $\mathbf{y}_k = \mathbf{X}_k\mathbf{f}_k$, which admit the best possible one-dimensional PCA representation $\mathbf{z}$ and subject to the constraints $\|\mathbf{y}_k\| = 1$ for $k = 1, \ldots, M$. The cost function to be minimized with respect to $\mathbf{f} = [\mathbf{f}_1^{\mathrm{T}}, \ldots, \mathbf{f}_M^{\mathrm{T}}]^{\mathrm{T}}$ is

$$J_{\mathrm{PCA}}(\mathbf{f}) = \min_{\mathbf{z},\mathbf{a}} \frac{1}{M} \sum_{k=1}^{M} \|\mathbf{z} - a_k\mathbf{y}_k\|^2, \tag{5}$$

where $\mathbf{a} = [a_1, \ldots, a_M]^{\mathrm{T}}$ is the vector containing the weights for the best combination of the outputs and $\mathbf{f}$ is the vector containing the projectors. In order to avoid the trivial solution ($\mathbf{a} = \mathbf{0}, \mathbf{z} = \mathbf{0}$), the energy of $\mathbf{z}$ or $\mathbf{a}$ has to be constrained to some fixed value. For reasons which will become clear later, we select $\|\mathbf{a}\|^2 = M$, although any other restriction (for instance $\|\mathbf{a}\| = 1$ or $\|\mathbf{z}\| = 1$) will provide the same solutions for $\mathbf{f}$ (since $\|\mathbf{y}_k\| = 1$) and a scaled version of $\mathbf{a}$ and $\mathbf{z}$.

Taking the derivative of (5) with respect to $\mathbf{z}$ and equating to zero we get

$$\mathbf{z} = \frac{1}{M}\mathbf{Y}\mathbf{a}, \tag{6}$$

where $\mathbf{Y} = [\mathbf{y}_1 \cdots \mathbf{y}_M]$ is a matrix containing the projections of the M data sets. Now, substituting (6) into (5), the cost function becomes

$$J_{\mathrm{PCA}}(\mathbf{f}) = \min_{\mathbf{a}} \left(1 - \frac{\mathbf{a}^{\mathrm{T}}\mathbf{Y}^{\mathrm{T}}\mathbf{Y}\mathbf{a}}{M^2}\right) = 1 - \beta,$$

where β is the largest eigenvalue of $\mathbf{Y}^{\mathrm{T}}\mathbf{Y}/M$ (which depends on $\mathbf{f}$) and $\mathbf{a}$ is the associated eigenvector scaled to $\|\mathbf{a}\|^2 = M$.

[1] In the two data set case the PCA weights are always 1/2, and then the second layer does not require training.

3.2. Remaining CCA–MAXVAR solutions

In order to determine additional CCA solutions the optimization problem is solved including orthogonality constraints in the PCA approximations $\mathbf{z}^{(i)}$ (other alternative orthogonality constraints can be found in Kettenring (1971)). Therefore, defining the vectors associated to the i-th CCA solution as $\mathbf{f}_k^{(i)}$, the CCA–MAXVAR problem can be stated as the problem of successively finding the set of M vectors $\mathbf{f}_k^{(i)}$ and the corresponding projections $\mathbf{y}_k^{(i)} = \mathbf{X}_k\mathbf{f}_k^{(i)}$, which admit the best possible one-dimensional PCA representation $\mathbf{z}^{(i)}$ and subject to the constraints $\|\mathbf{y}_k^{(i)}\| = 1$ and $\mathbf{z}^{(i)\mathrm{T}}\mathbf{z}^{(j)} = 0$ for $j = 1, \ldots, i-1$.

Analogously to the procedure for the main CCA solution, the cost function to be minimized with respect to $\mathbf{f}^{(i)}$, and subject to $\|\mathbf{a}^{(i)}\|^2 = M$, is

$$J_{\mathrm{PCA}}(\mathbf{f}^{(i)}) = \min_{\mathbf{z}^{(i)},\mathbf{a}^{(i)}} \frac{1}{M}\sum_{k=1}^{M} \|\mathbf{z}^{(i)} - a_k^{(i)}\mathbf{y}_k^{(i)}\|^2,$$

where $\mathbf{a}^{(i)} = [a_1^{(i)}, \ldots, a_M^{(i)}]^{\mathrm{T}}$ groups the weights for the best combination of the outputs and $\mathbf{f}^{(i)} = [\mathbf{f}_1^{(i)\mathrm{T}}, \ldots, \mathbf{f}_M^{(i)\mathrm{T}}]^{\mathrm{T}}$ stacks the projectors for the i-th CCA–MAXVAR solution. The minimization of this cost function is obtained for

$$\mathbf{z}^{(i)} = \frac{1}{M}\mathbf{Y}^{(i)}\mathbf{a}^{(i)},$$

where $\mathbf{Y}^{(i)} = [\mathbf{y}_1^{(i)} \cdots \mathbf{y}_M^{(i)}]$, which implies

$$J_{\mathrm{PCA}}(\mathbf{f}^{(i)}) = \min_{\mathbf{a}^{(i)}}\left(1 - \frac{\mathbf{a}^{(i)\mathrm{T}}\mathbf{Y}^{(i)\mathrm{T}}\mathbf{Y}^{(i)}\mathbf{a}^{(i)}}{M^2}\right) = 1 - \beta^{(i)},$$

where $\mathbf{a}^{(i)}$ is the eigenvector (scaled to $\|\mathbf{a}^{(i)}\|^2 = M$) associated to the largest possible eigenvalue $\beta^{(i)}$ of $\mathbf{Y}^{(i)\mathrm{T}}\mathbf{Y}^{(i)}/M$ (which depends on $\mathbf{f}^{(i)}$) satisfying the orthogonality restrictions $\mathbf{z}^{(i)\mathrm{T}}\mathbf{z}^{(j)} = 0$, for $j = 1, \ldots, i-1$.

3.3. CCA–MAXVAR solution based on the SVD

In Kettenring (1971), the solutions $\mathbf{f}^{(i)}, \mathbf{a}^{(i)}, \mathbf{z}^{(i)}$ of the CCA–MAXVAR generalization were obtained using the singular value decomposition (SVD) of $\mathbf{X}_k = \mathbf{U}_k\boldsymbol{\Sigma}_k\mathbf{V}_k^{\mathrm{T}}$, where $\mathbf{U}_k^{\mathrm{T}}\mathbf{U}_k = \mathbf{I}$, $\mathbf{V}_k^{\mathrm{T}}\mathbf{V}_k = \mathbf{I}$, and $\boldsymbol{\Sigma}_k$ is a diagonal matrix with the singular values of $\mathbf{X}_k$. This implies

$$\mathbf{y}_k^{(i)} = \mathbf{X}_k\mathbf{f}_k^{(i)} = \mathbf{U}_k\mathbf{g}_k^{(i)},$$

and taking into account the constraint $\|\mathbf{y}_k^{(i)}\| = 1$ and the property $\mathbf{U}_k^{\mathrm{T}}\mathbf{U}_k = \mathbf{I}$, we can write

$$\begin{aligned}\|\mathbf{y}_k^{(i)}\|^2 &= \mathbf{y}_k^{(i)\mathrm{H}}\mathbf{y}_k^{(i)} = \mathbf{g}_k^{(i)\mathrm{H}}\mathbf{U}_k^{\mathrm{H}}\mathbf{U}_k\mathbf{g}_k^{(i)} = \mathbf{g}_k^{(i)\mathrm{H}}\mathbf{g}_k^{(i)} \\ &= \|\mathbf{g}_k^{(i)}\|^2 = 1,\end{aligned}$$

i.e., $\mathbf{g}_k^{(i)} = \boldsymbol{\Sigma}_k\mathbf{V}_k^{\mathrm{T}}\mathbf{f}_k^{(i)}$ is a unit norm vector. Defining $\mathbf{U} = [\mathbf{U}_1 \cdots \mathbf{U}_M]$, $\beta^{(i)}$ can be rewritten as

$$\beta^{(i)} = \frac{1}{M^2}\mathbf{b}^{(i)\mathrm{T}}\mathbf{U}^{\mathrm{T}}\mathbf{U}\mathbf{b}^{(i)},$$

where $\mathbf{b}^{(i)} = [\mathbf{b}_1^{(i)\mathrm{T}}, \ldots, \mathbf{b}_M^{(i)\mathrm{T}}]^{\mathrm{T}}$, with $\mathbf{b}_k^{(i)} = a_k^{(i)}\mathbf{g}_k^{(i)}$, consequently $\|\mathbf{b}^{(i)}\|^2 = \|\mathbf{a}^{(i)}\|^2 = M$.

After the SVD, the solution $\mathbf{b}^{(i)}$ satisfying the orthogonality restrictions $\mathbf{z}^{(i)\mathrm{T}}\mathbf{z}^{(j)} = 0$ $(j = 1, \ldots, i-1)$ is the eigenvector of $\mathbf{U}^{\mathrm{T}}\mathbf{U}/M$ associated to its i-th largest eigenvalue $\beta^{(i)}$, and the vectors $\mathbf{g}_k^{(i)}, \mathbf{f}_k^{(i)}$ and $\mathbf{a}^{(i)}$ can be obtained from $\mathbf{b}^{(i)}$ in a straightforward manner. Furthermore, denoting $\mathbf{b}^{(i)} = \mathbf{G}^{(i)}\mathbf{a}^{(i)}$, where

$$\mathbf{G}^{(i)} = \begin{bmatrix} \mathbf{g}_1^{(i)} & \cdots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \cdots & \mathbf{g}_M^{(i)} \end{bmatrix},$$

is a matrix satisfying $\mathbf{G}^{(i)\mathrm{T}}\mathbf{G}^{(i)} = \mathbf{I}$, we can write

$$\frac{1}{M}\mathbf{Y}^{(i)\mathrm{T}}\mathbf{Y}^{(i)}\mathbf{a}^{(i)} = \frac{1}{M}\mathbf{G}^{(i)\mathrm{T}}\mathbf{U}^{\mathrm{T}}\mathbf{U}\mathbf{b}^{(i)} = \beta^{(i)}\mathbf{a}^{(i)},$$

which proves that $\mathbf{a}^{(i)}$ is some eigenvector (not necessarily the first) of $\mathbf{Y}^{(i)\mathrm{T}}\mathbf{Y}^{(i)}/M$ with eigenvalue $\beta^{(i)}$.

Finally, we must point out that, in the particular case of $M = 2$ data sets, the solution of the CCA–MAXVAR problem is the same as the classical formulation of CCA (Kettenring, 1971). This equivalence will become clear in the next section.

4. LS generalization of CCA to several data sets

In this section, a generalization of CCA to several data sets is proposed. This generalization is developed within a least squares (LS) regression framework and we prove that it is equivalent to the PCA-based CCA–MAXVAR generalization. However, unlike the CCA–MAXVAR generalization, the proposed approach does not require a prewhitening (SVD) step and it is able to find the canonical vectors and variates directly from the data sets. Additionally, the proposed generalization admits the neural model represented in Fig. 2. Again, it is a two layer model, but now the second layer has fixed weights, and therefore it does not require training. Obviously, in this new neural model the PCA coefficients $a_k^{(i)}$ of the previous model have been incorporated into the first projection layer. Although this may seem a minor modification, it is, as we will show later, a key ingredient to develop adaptive learning algorithms for this structure.

4.1. CCA generalization based on distances

Let $\mathbf{X}_k \in \mathbb{C}^{N\times m_k}$ for $k = 1, \ldots, M$ be full-rank matrices. If we denote the successive canonical vectors and variables as $\mathbf{h}_k^{(i)}$ and $\mathbf{z}_k^{(i)} = \mathbf{X}_k\mathbf{h}_k^{(i)}$, respectively; and the estimated cross-correlation matrices as $\mathbf{R}_{kl} = \mathbf{X}_k^{\mathrm{H}}\mathbf{X}_l$, then our CCA generalization can be formulated as the problem of sequentially maximizing the generalized canonical correlation

$$\rho^{(i)} = \frac{1}{M}\sum_{k=1}^{M}\rho_k^{(i)},$$

where $\rho_{kl}^{(i)} = \mathbf{h}_k^{(i)\mathrm{H}}\mathbf{R}_{kl}\mathbf{h}_l^{(i)}$ and

$$\rho_k^{(i)} = \frac{1}{M-1}\sum_{\substack{l=1\\ l\neq k}}^{M}\rho_{kl}^{(i)}.$$

In this case, the energy constraint to avoid trivial solutions is

$$\frac{1}{M}\sum_{k=1}^{M}\mathbf{h}_k^{(i)\mathrm{H}}\mathbf{R}_{kk}\mathbf{h}_k^{(i)} = 1, \tag{7}$$

and defining $\mathbf{z}^{(i)} = \frac{1}{M}\sum_{k=1}^{M}\mathbf{z}_k^{(i)}$, the orthogonality constraints are, for $i \neq j$

$$\mathbf{z}^{(i)\mathrm{H}}\mathbf{z}^{(j)} = 0. \tag{8}$$

Unlike the two data set case, here it is interesting to point out that, for $M > 2$, the energy constraint (7) is not equivalent to $\mathbf{h}_k^{(i)\mathrm{H}}\mathbf{R}_{kk}\mathbf{h}_k^{(i)} = 1$ for all k. However, as we will see later, the solutions of the proposed generalization with $M = 2$ data sets are the same of the conventional formulation of the CCA problem and then they satisfy $\mathbf{z}_k^{(i)\mathrm{H}}\mathbf{z}_l^{(j)} = 0$ and $\mathbf{h}_k^{(i)\mathrm{H}}\mathbf{R}_{kk}\mathbf{h}_k^{(i)} = 1$ for $k, l = 1, 2$ and $i \neq j$.

The proposed CCA generalization (referred to as CCA–LS) can be rewritten as a function of distances. Specifically, to extract the i-th CCA eigenvector, the generalized CCA problem consists on minimizing, with respect to the M canonical vectors $\mathbf{h}_k^{(i)}$, the following cost function

$$J^{(i)} = \frac{1}{2M(M-1)}\sum_{k,l=1}^{M}\left\|\mathbf{X}_k\mathbf{h}_k^{(i)} - \mathbf{X}_l\mathbf{h}_l^{(i)}\right\|^2 = \frac{1}{M}\sum_{k=1}^{M}\|\mathbf{z}_k^{(i)}\|^2 - \rho^{(i)},$$

subject to (7) and (8), which implies $J^{(i)} = 1 - \rho^{(i)}$.

The solutions of this generalized CCA problem can be obtained by the method of Lagrange multipliers (see the Appendix), whose solutions are determined by the following GEV problem

$$\frac{1}{M-1}(\mathbf{R}-\mathbf{D})\mathbf{h}^{(i)} = \rho^{(i)}\mathbf{D}\mathbf{h}^{(i)}, \tag{9}$$

where $\mathbf{h}^{(i)} = [\mathbf{h}_1^{(i)\mathrm{T}}, \ldots, \mathbf{h}_M^{(i)\mathrm{T}}]^{\mathrm{T}}$ stacks the canonical vectors,

$$\mathbf{R} = \begin{bmatrix}\mathbf{R}_{11} & \cdots & \mathbf{R}_{1M}\\ \vdots & \ddots & \vdots\\ \mathbf{R}_{M1} & \cdots & \mathbf{R}_{MM}\end{bmatrix}, \qquad \mathbf{D} = \begin{bmatrix}\mathbf{R}_{11} & \cdots & \mathbf{0}\\ \vdots & \ddots & \vdots\\ \mathbf{0} & \cdots & \mathbf{R}_{MM}\end{bmatrix}, \tag{10}$$

and $\rho^{(i)}$ is a generalized eigenvalue. Then, the CCA–LS solutions are obtained as the eigenvectors associated with the largest eigenvalues of (9). Here, we must note that, in the case of $M = 2$ data sets, the GEV problem (9) is reduced to (4), which proves that, for $M = 2$, the energy and orthogonality relaxed restrictions (7) and (8) are equivalent to the strict restrictions (2) and (3). Furthermore, Bach and Jordan (2002) prove that the eigenvalues of (9) can be used as a measure of the dependency (or mutual information) between several Gaussian data sets.

4.2. Equivalence to the CCA–MAXVAR approach

Here we show that the CCA–LS generalization in terms of distances given by (9) is equivalent to the MAXVAR generalization proposed by Kettenring in terms of PCA projections. Let us start by rewriting (9) as

$$\frac{1}{M}\mathbf{R}\mathbf{h}^{(i)} = \beta^{(i)}\mathbf{D}\mathbf{h}^{(i)}, \tag{11}$$

where

$$\beta^{(i)} = \frac{1 + (M-1)\rho^{(i)}}{M}.$$

Taking into account that replacing the transpose by the conjugate transpose operation, the CCA–MAXVAR method can be extended to complex numbers, the equivalence between the proposed and the MAXVAR generalization of CCA is stated in the following theorem

Theorem 1. *The classical PCA-based solutions $\mathbf{z}^{(i)}$, $\beta^{(i)}$ of the MAXVAR generalization of CCA coincide with those of the CCA–LS formulation, and the canonical vectors and variables are related by $\mathbf{h}_k^{(i)} = a_k^{(i)}\mathbf{f}_k^{(i)}$ and $\mathbf{z}_k^{(i)} = a_k^{(i)}\mathbf{y}_k^{(i)}$.*

Proof. In order to obtain the CCA–MAXVAR solution directly from $\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^{\mathrm{H}}$ we write

$$\frac{1}{M}\mathbf{U}^{\mathrm{H}}\mathbf{U}\mathbf{b}^{(i)} = \frac{1}{M}\mathbf{\Sigma}^{-1}\mathbf{V}^{\mathrm{H}}\mathbf{X}^{\mathrm{H}}\mathbf{X}\mathbf{V}\mathbf{\Sigma}^{-1}\mathbf{b}^{(i)} = \beta^{(i)}\mathbf{b}^{(i)}, \tag{12}$$

where $\mathbf{\Sigma}$ and $\mathbf{V}$ are block-diagonal matrices with block elements $\mathbf{\Sigma}_k$ and $\mathbf{V}_k$ ($k = 1, \ldots, M$), respectively, and $\mathbf{X} = [\mathbf{X}_1 \cdots \mathbf{X}_M]$.

Left-multiplying (12) by $\mathbf{V}\mathbf{\Sigma}^{-1}$ we have

$$\frac{1}{M}\mathbf{V}\mathbf{\Sigma}^{-2}\mathbf{V}^{\mathrm{H}}\mathbf{X}^{\mathrm{H}}\mathbf{X}\mathbf{V}\mathbf{\Sigma}^{-1}\mathbf{b}^{(i)} = \beta^{(i)}\mathbf{V}\mathbf{\Sigma}^{-1}\mathbf{b}^{(i)},$$

and taking into account that $\mathbf{R} = \mathbf{X}^{\mathrm{H}}\mathbf{X}$ and $\mathbf{D} = \mathbf{V}\mathbf{\Sigma}^2\mathbf{V}^{\mathrm{H}}$ are the matrices defined in (10), the GEV problem (11) is obtained, where $\mathbf{h}^{(i)} = \mathbf{V}\mathbf{\Sigma}^{-1}\mathbf{b}^{(i)}$. Obviously, (11) has the same solutions (eigenvectors) as the proposed generalized CCA problem in (9). Noting that $a_k^{(i)}\mathbf{f}_k^{(i)} = a_k^{(i)}\mathbf{V}_k\mathbf{\Sigma}_k^{-1}\mathbf{g}_k^{(i)} = \mathbf{V}_k\mathbf{\Sigma}_k^{-1}\mathbf{b}_k^{(i)}$ we also find that $\mathbf{h}_k^{(i)} = a_k^{(i)}\mathbf{f}_k^{(i)}$ and $\mathbf{z}_k^{(i)} = a_k^{(i)}\mathbf{y}_k^{(i)}$.

Now, it is easy to realize that $\|\mathbf{y}_k^{(i)}\| = 1$ and $\|\mathbf{a}^{(i)}\|^2 = M$ implies (7) (which justify our election $\|\mathbf{a}^{(i)}\|^2 = M$), and finally, the PCA approximation is given by

$$\mathbf{z}^{(i)} = \frac{1}{M}\mathbf{Y}^{(i)}\mathbf{a}^{(i)} = \frac{1}{M}\mathbf{U}\mathbf{b}^{(i)} = \frac{1}{M}\mathbf{X}\mathbf{h}^{(i)} = \frac{1}{M}\sum_{k=1}^{M}\mathbf{z}_k^{(i)},$$

which concludes the proof. □

5. Iterative algorithm for CCA–LS

In this section, the least squares framework is exploited to introduce a batch algorithm based on iterative least squares regression. This algorithm constitutes the framework used in the next section to develop an adaptive algorithm based on the RLS.

5.1. Main CCA–LS solution

By noting that $\mathbf{R}_{kk}^{-1}\mathbf{R}_{kl} = \mathbf{X}_k^{+}\mathbf{X}_l$, where $\mathbf{X}_k^{+} = (\mathbf{X}_k^{\mathrm{H}}\mathbf{X}_k)^{-1}\mathbf{X}_k^{\mathrm{H}}$ is the pseudoinverse of $\mathbf{X}_k$, the GEV problem (11) can be viewed as M coupled LS regression problems

$$\beta^{(i)}\mathbf{h}_k^{(i)} = \mathbf{X}_k^{+}\mathbf{z}^{(i)}, \quad k = 1, \ldots, M,$$

where $\mathbf{z}^{(i)} = \frac{1}{M}\sum_{k=1}^{M}\mathbf{z}_k^{(i)}$ and $\mathbf{z}_k^{(i)} = \mathbf{X}_k\mathbf{h}_k^{(i)}$. The key idea of the batch algorithm for the extraction of the main CCA–LS solution is to solve these regression problems iteratively: at each iteration t we form M LS regression problems using $\hat{\mathbf{z}}^{(1)}(t)$ as desired output, and a new solution is thus found by solving

$$\hat{\beta}^{(1)}(t)\hat{\mathbf{h}}_k^{(1)}(t) = \mathbf{X}_k^{+}\hat{\mathbf{z}}^{(1)}(t), \quad k = 1, \ldots, M,$$

where

$$\hat{\mathbf{z}}^{(i)}(t) = \frac{1}{M}\sum_{k=1}^{M}\hat{\mathbf{z}}_k^{(i)}(t),$$

$$\hat{\mathbf{z}}_k^{(i)}(t) = \mathbf{X}_k\hat{\mathbf{h}}_k^{(i)}(t-1).$$

Finally, $\hat{\beta}^{(i)}(t)$ and $\hat{\mathbf{h}}^{(i)}(t)$ can be obtained by scaling $\hat{\mathbf{h}}^{(i)}(t)$ to strictly satisfy (7). Nevertheless, this normalization step can be further simplified by imposing $\|\hat{\mathbf{h}}^{(i)}(t)\| = 1$, which only introduces a constant scale factor

$$c = \sqrt{\frac{\hat{\mathbf{h}}^{(i)\mathrm{H}}(t)\mathbf{R}\hat{\mathbf{h}}^{(i)}(t)}{M}},$$

in the CCA solutions $\hat{\mathbf{h}}_k^{(i)}(t), \hat{\mathbf{z}}_k^{(i)}(t)$, and reduces the obtention of the PCA eigenvalue to $\hat{\beta}^{(i)}(t) = \|\hat{\beta}^{(i)}(t)\hat{\mathbf{h}}^{(i)}(t)\|$.

5.2. Remaining CCA–LS solutions

The subsequent CCA eigenvectors can be obtained by means of a deflation technique that imposes the following orthogonality constraints

$$\mathbf{h}^{(i)\mathrm{H}}\mathbf{D}\mathbf{h}^{(j)} = 0, \quad i \neq j,$$

which are a direct consequence of the GEV problem (11) and the orthogonality among the subsequent solutions $\mathbf{z}^{(i)}$.

Defining now $\hat{\mathbf{H}}^{(i)}(t) = [\hat{\mathbf{h}}^{(1)}(t) \cdots \hat{\mathbf{h}}^{(i-1)}(t)]$ and $\hat{\mathbf{U}}^{(i)}(t) = [\hat{\mathbf{u}}^{(1)}(t) \cdots \hat{\mathbf{u}}^{(i-1)}(t)] = \mathbf{D}\hat{\mathbf{H}}^{(i)}(t-1)$, the update equation for the i-th CCA solution is

$$\tilde{\beta}^{(i)}(t)\tilde{\mathbf{h}}_k^{(i)}(t) = \mathbf{X}_k^{+}\hat{\mathbf{z}}^{(i)}(t), \quad k = 1, \ldots, M,$$

$$\hat{\beta}^{(i)}(t)\hat{\mathbf{h}}^{(i)}(t) = \hat{\mathbf{P}}_{\mathbf{U}}^{(i)}(t)\tilde{\beta}^{(i)}(t)\tilde{\mathbf{h}}^{(i)}(t),$$

where $\tilde{\mathbf{h}}^{(i)}(t) = [\tilde{\mathbf{h}}_1^{(i)\mathrm{T}}(t), \ldots, \tilde{\mathbf{h}}_M^{(i)\mathrm{T}}(t)]^{\mathrm{T}}$, $\hat{\mathbf{P}}_{\mathbf{U}}^{(i)}(t)$ denotes the projection matrix onto the complementary subspace to $\hat{\mathbf{U}}^{(i)}(t)$,

$$\begin{aligned}\hat{\mathbf{P}}_{\mathbf{U}}^{(i)}(t) &= \mathbf{I} - \hat{\mathbf{U}}^{(i)}(t)\left(\hat{\mathbf{U}}^{(i)\mathrm{H}}(t)\hat{\mathbf{U}}^{(i)}(t)\right)^{-1}\hat{\mathbf{U}}^{(i)\mathrm{H}}(t)\\ &= \mathbf{I} - \hat{\mathbf{V}}^{(i)}(t)\hat{\mathbf{V}}^{(i)\mathrm{H}}(t),\end{aligned}$$

and $\hat{\mathbf{V}}^{(i)}(t) = [\hat{\mathbf{v}}^{(1)}(t) \cdots \hat{\mathbf{v}}^{(i-1)}(t)]$ is an orthonormal basis of $\hat{\mathbf{U}}^{(i)}(t)$, which can be easily obtained by means of the Gram–Schmidt orthogonalization procedure

$$\hat{\mathbf{v}}^{(i)}(t) = \frac{\hat{\mathbf{P}}_{\mathbf{U}}^{(i)}(t)\hat{\mathbf{u}}^{(i)}(t)}{\|\hat{\mathbf{P}}_{\mathbf{U}}^{(i)}(t)\hat{\mathbf{u}}^{(i)}(t)\|}.$$

Finally, the recursive rule for the projection matrix is

$$\hat{\mathbf{P}}_{\mathbf{U}}^{(i+1)}(t) = \hat{\mathbf{P}}_{\mathbf{U}}^{(i)}(t) - \hat{\mathbf{v}}^{(i)}(t)\hat{\mathbf{v}}^{(i)\mathrm{H}}(t),$$

with $\hat{\mathbf{P}}_{\mathbf{U}}^{(1)}(t) = \mathbf{I}$.

6. Learning algorithm for CCA–LS

In the previous section we have shown that the reformulation of the CCA–LS generalization as a set of coupled LS regression problems yields in a natural way an iterative algorithm for batch training. In this section we go one step further and derive an adaptive learning algorithm for CCA based on the well-known RLS algorithm.

Unlike other recently proposed GEV adaptive algorithms (Rao, Principe, & Wong, 2004), the learning rule presented in this paper is a true RLS algorithm that uses a reference signal specifically constructed for CCA and derived from the LS regression framework. This reference signal opens the possibility of new improvements of CCA algorithms: for instance, it can be used to develop robust versions of the algorithm (Vía et al., 2005a), or to construct a soft decision signal useful for blind equalization problems (Vía & Santamaría, 2005).

6.1. Main CCA–LS solution

To obtain an on-line algorithm, the LS regression problems are now rewritten as the following exponentially weighted cost functions

$$J_k^{(i)}(n) = \sum_{l=1}^{n}\lambda^{n-l}\left|\hat{z}^{(i)}(l) - \hat{\beta}^{(i)}(n)\mathbf{x}_k^{\mathrm{H}}(l)\hat{\mathbf{h}}_k^{(i)}(n)\right|^2.$$

Denoting the n-th row of the k-th data set as $\mathbf{x}_k^{\mathrm{H}}(n)$, and writing the associated Kalman gain vector as $\mathbf{k}_{\mathbf{x}_k}(n)$, the update RLS equations are

$$\mathbf{k}_{\mathbf{x}_k}(n) = \frac{\mathbf{P}_{\mathbf{x}_k}(n-1)\mathbf{x}_k(n)}{\lambda + \mathbf{x}_k^{\mathrm{H}}(n)\mathbf{P}_{\mathbf{x}_k}(n-1)\mathbf{x}_k(n)}, \tag{13}$$

$$\mathbf{P}_{\mathbf{x}_k}(n) = \lambda^{-1}(\mathbf{I} - \mathbf{k}_{\mathbf{x}_k}(n)\mathbf{x}_k^{\mathrm{H}}(n))\mathbf{P}_{\mathbf{x}_k}(n-1), \tag{14}$$

where $0 < \lambda \leq 1$ is the forgetting factor and $\mathbf{P}_{\mathbf{x}_k}(n) = \mathbf{\Phi}_{\mathbf{x}_k}^{-1}(n)$ is the inverse of the estimated correlation matrix $\mathbf{\Phi}_{\mathbf{x}_k}(n) = \sum_{l=1}^{n}\lambda^{n-l}\mathbf{x}_k(l)\mathbf{x}_k^{\mathrm{H}}(l)$.

Using these update equations, a direct application of the RLS algorithm yields, for $k = 1, \ldots, M$

$$\hat{\beta}^{(1)}(n)\hat{\mathbf{h}}_k^{(1)}(n) = \hat{\beta}^{(1)}(n-1)\hat{\mathbf{h}}_k^{(1)}(n-1) + \mathbf{k}_{\mathbf{x}_k}(n)e_k^{(1)}(n),$$

where

$$e_k^{(i)}(n) = \hat{z}^{(i)}(n) - \hat{\beta}^{(i)}(n-1)\mathbf{x}_k^{\mathrm{H}}(n)\hat{\mathbf{h}}_k^{(i)}(n-1), \tag{15}$$

```
Initialize $\mathbf{P}_{\mathbf{x}_k}(0) = \delta^{-1}\mathbf{I}$, with $\delta \ll 1$ for $k = 1, \dots, M$.
Initialize $\hat{\mathbf{h}}^{(i)}(0) \neq \mathbf{0}$, $\hat{\mathbf{u}}^{(i)}(0) = \mathbf{0}$ and $\hat{\beta}^{(i)}(0) = 0$ for $i = 1, \dots, p$.
for $n = 1, 2, \dots$ do
  Update $\mathbf{k}_{\mathbf{x}_k}(n)$ and $\mathbf{P}_{\mathbf{x}_k}(n)$ with (13) and (14) for $k = 1, \dots, M$.
  for $i = 1, \dots, p$ do
    Obtain $\hat{z}^{(i)}(n)$, and $\mathbf{e}^{(i)}(n)$ with (15).
    Obtain $\hat{\beta}^{(i)}(n)\hat{\mathbf{h}}^{(i)}(n)$ using (16), (17) and (18).
    Estimate $\hat{\beta}^{(i)}(n)$, $\hat{\rho}^{(i)}(n)$ and $\hat{\mathbf{h}}^{(i)}(n)$ considering $\|\hat{\mathbf{h}}^{(i)}(n)\| = 1$.
  end for
end for
```

Algorithm 1: Summary of the proposed CCA–RLS adaptive algorithm.

is the *a priori* error for the k-th data set, and the reference signal is obtained as

$$\hat{z}^{(i)}(n) = \frac{1}{M}\sum_{k=1}^{M}\hat{z}_k^{(i)}(n),$$

where $\hat{z}_k^{(i)}(n) = \mathbf{x}_k^{\mathrm{H}}(n)\hat{\mathbf{h}}_k^{(i)}(n-1)$.

By grouping now the *a priori* errors into the vector $\mathbf{e}^{(i)}(n) = [e_1^{(i)}(n), \dots, e_M^{(i)}(n)]^{\mathrm{T}}$, we can write the overall algorithm (see Algorithm 1) in matrix form as

$$\tilde{\beta}^{(i)}(n)\tilde{\mathbf{h}}^{(i)}(n) = \hat{\beta}^{(i)}(n-1)\hat{\mathbf{h}}^{(i)}(n-1) + \mathbf{K}(n)\mathbf{e}^{(i)}(n), \qquad (16)$$

where $\hat{\beta}^{(1)}(n) = \tilde{\beta}^{(1)}(n)$, $\hat{\mathbf{h}}^{(1)}(n) = \tilde{\mathbf{h}}^{(1)}(n)$, and

$$\mathbf{K}(n) = \begin{bmatrix} \mathbf{k}_{\mathbf{x}_1}(n) & \cdots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \dots & \mathbf{k}_{\mathbf{x}_M}(n) \end{bmatrix}.$$

6.2. *Remaining CCA–LS solutions*

Analogously to the batch algorithm, the extraction of the subsequent CCA solutions is based on a deflation technique. In Vía et al. (2005b) we have proposed an RLS-based method, which resembles the APEX algorithm (Diamantaras & Kung, 1996); here we propose an alternative technique based on the vectors

$$\hat{\mathbf{u}}_k^{(i)}(n) = \sum_{l=1}^{n}\lambda^{n-l}\mathbf{x}_k(l)\mathbf{x}_k^{\mathrm{H}}(l)\hat{\mathbf{h}}_k^{(i)}(l-1),$$

which can be updated as

$$\hat{\mathbf{u}}_k^{(i)}(n) = \lambda\hat{\mathbf{u}}_k^{(i)}(n-1) + \mathbf{x}_k(n)\hat{z}_k^{(i)}(n). \qquad (17)$$

The deflation technique consists of

$$\hat{\beta}^{(i)}(n)\hat{\mathbf{h}}^{(i)}(n) = \hat{\mathbf{P}}_{\mathbf{U}}^{(i)}(n)\tilde{\beta}^{(i)}(n)\tilde{\mathbf{h}}^{(i)}(n), \qquad (18)$$

where $\hat{\mathbf{u}}^{(i)}(n) = [\hat{\mathbf{u}}_1^{(i)\mathrm{T}}(n), \dots, \hat{\mathbf{u}}_M^{(i)\mathrm{T}}(n)]^{\mathrm{T}}$, $\hat{\mathbf{P}}_{\mathbf{U}}^{(i)}(n)$ is the projection matrix onto the complementary subspace to

$$\hat{\mathbf{U}}^{(i)}(n) = [\hat{\mathbf{u}}^{(1)}(n) \cdots \hat{\mathbf{u}}^{(i-1)}(n)],$$

$$\begin{aligned}\hat{\mathbf{P}}_{\mathbf{U}}^{(i)}(n) &= \mathbf{I} - \hat{\mathbf{U}}^{(i)}(n)(\hat{\mathbf{U}}^{(i)\mathrm{H}}(n)\hat{\mathbf{U}}^{(i)}(n))^{-1}\hat{\mathbf{U}}^{(i)\mathrm{H}}(n) \\ &= \mathbf{I} - \hat{\mathbf{V}}^{(i)}(n)\hat{\mathbf{V}}^{(i)\mathrm{H}}(n),\end{aligned}$$

and $\hat{\mathbf{V}}^{(i)}(n) = [\hat{\mathbf{v}}^{(1)}(n) \cdots \hat{\mathbf{v}}^{(i-1)}(n)]$ is an orthonormal basis of $\hat{\mathbf{U}}^{(i)}(n)$, which is recursively obtained by means of the Gram–Schmidt orthogonalization procedure

$$\hat{\mathbf{v}}^{(i)}(n) = \frac{\hat{\mathbf{P}}_{\mathbf{U}}^{(i)}(n)\hat{\mathbf{u}}^{(i)}(n)}{\|\hat{\mathbf{P}}_{\mathbf{U}}^{(i)}(n)\hat{\mathbf{u}}^{(i)}(n)\|}.$$

Finally, the recursive rule for the projection matrix is

$$\hat{\mathbf{P}}_{\mathbf{U}}^{(i+1)}(n) = \hat{\mathbf{P}}_{\mathbf{U}}^{(i)}(n) - \hat{\mathbf{v}}^{(i)}(n)\hat{\mathbf{v}}^{(i)\mathrm{H}}(n),$$

with $\hat{\mathbf{P}}_{\mathbf{U}}^{(1)}(n) = \mathbf{I}$.

7. Convergence analysis

In this section the convergence of the proposed CCA–RLS algorithm is analyzed following the convergence proof outlined by Rao et al. (2004), wherein the stochastic approximation tools have been employed to prove the convergence of a GEV algorithm. Let us start by defining the data vector $\mathbf{x}(n) = \left[\mathbf{x}_1^{\mathrm{T}}(n), \dots, \mathbf{x}_M^{\mathrm{T}}(n)\right]^{\mathrm{T}}$ and the matrices

$$\mathbf{X}(n) = \begin{bmatrix} \mathbf{x}_1(n) & \cdots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \cdots & \mathbf{x}_M(n) \end{bmatrix},$$

$$\mathbf{D}(n) = \begin{bmatrix} \mathbf{\Phi}_{\mathbf{x}_1}(n) & \cdots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \cdots & \mathbf{\Phi}_{\mathbf{x}_M}(n) \end{bmatrix}.$$

Taking into account that $\mathbf{k}_{\mathbf{x}_k}(n) = \mathbf{P}_{\mathbf{x}_k}(n)\mathbf{x}_k(n)$, we can write

$$\mathbf{K}(n) = \mathbf{D}^{-1}(n)\mathbf{X}(n),$$

and rewriting $\hat{z}^{(i)}(n) = \frac{1}{M}\mathbf{x}^{\mathrm{H}}(n)\hat{\mathbf{h}}^{(i)}(n-1)$, Eqs. (15), (16) and (18) yield

$$\begin{aligned}\hat{\beta}^{(i)}(n)\hat{\mathbf{h}}^{(i)}(n) = \hat{\mathbf{P}}_{\mathbf{U}}^{(i)}(n)\mathbf{D}^{-1}(n)\Big[&\lambda\mathbf{D}(n-1)\hat{\beta}^{(i)}(n-1) \\ &+ \frac{1}{M}\mathbf{x}(n)\mathbf{x}^{\mathrm{H}}(n)\Big]\hat{\mathbf{h}}^{(i)}(n-1),\end{aligned} \qquad (19)$$

or, equivalently,

$$\begin{aligned}\hat{\beta}^{(i)}(n)\hat{\mathbf{h}}^{(i)}(n) = \lambda^n &\left[\prod_{k=1}^{n}(\hat{\mathbf{P}}_{\mathbf{U}}^{(i)}(k)\mathbf{D}^{-1}(k)\mathbf{D}(k-1))\right] \\ &\times \hat{\beta}^{(i)}(0)\hat{\mathbf{h}}^{(i)}(0) \\ &+ \frac{1}{M}\sum_{k=1}^{n}\lambda^{n-k}\hat{\mathbf{P}}_{\mathbf{U}}^{(i)}(k)\mathbf{D}^{-1}(k) \\ &\times \mathbf{x}(k)\mathbf{x}^{\mathrm{H}}(k)\hat{\mathbf{h}}^{(i)}(k-1),\end{aligned}$$

and initializing $\hat{\beta}^{(i)}(0) = 0$ we obtain

$$\hat{\beta}^{(i)}(n)\hat{\mathbf{h}}^{(i)}(n) = \frac{1}{M}\sum_{k=1}^{n}\lambda^{n-k}\hat{\mathbf{P}}_{\mathbf{U}}^{(i)}(k)\mathbf{D}^{-1}(k)\mathbf{x}(k)\mathbf{x}^{\mathrm{H}}(k) \times \hat{\mathbf{h}}^{(i)}(k-1). \tag{20}$$

The convergence analysis is based on the stochastic approximation techniques proposed by Benveniste, Métivier, and Priouret (1990). Eq. (20) can be seen as a special case of the generic stochastic approximation algorithm

$$\hat{\beta}^{(i)}(n)\hat{\mathbf{h}}^{(i)}(n) = \hat{\beta}^{(i)}(n-1)\hat{\mathbf{h}}^{(i)}(n-1) + f^{(i)}(\hat{\beta}^{(i)}(n-1)\hat{\mathbf{h}}^{(i)}(n-1), \mathbf{x}(n)),$$

which belongs to the constant gain type algorithms. The central idea of the convergence study of this class of algorithms is to associate the discrete-time update equation to an ordinary differential equation (ODE) and then link the convergence of the ODE to that of the discrete-time equation. We will make the following mild assumptions:

A.1. The inputs $\mathbf{x}_k(n)$ are at least wide sense stationary (WSS) with positive definite autocorrelation matrices $\mathbf{R}_{kk}$.

A.2. The sequence $\hat{\beta}^{(i)}(n)\hat{\mathbf{h}}^{(i)}(n)$ is bounded with probability 1, which is ensured by A.1. and the normalization step.

A.3. The update function $f^{(i)}(\hat{\beta}^{(i)}(n-1)\hat{\mathbf{h}}^{(i)}(n-1), \mathbf{x}(n))$ is continuously differentiable with respect to $\hat{\beta}^{(i)}(n-1)\hat{\mathbf{h}}^{(i)}(n-1)$ and $\mathbf{x}(n)$ and its derivatives are bounded in time.

A.4. Even if the update function has some discontinuities, a mean vector field

$$\bar{f}^{(i)}(\hat{\beta}^{(i)}\hat{\mathbf{h}}^{(i)}, \mathbf{x}) = \lim_{n\to\infty} E[f^{(i)}(\hat{\beta}^{(i)}(n-1) \times \hat{\mathbf{h}}^{(i)}(n-1), \mathbf{x}(n))],$$

exists and is regular. This fact can be easily proved based on A.1 and A.2.

A.5. The initial estimates $\hat{\mathbf{h}}^{(i)}(0)$ are chosen such that $\mathbf{h}^{(i)\mathrm{H}}\hat{\mathbf{h}}^{(i)}(0) \neq 0$, where $\mathbf{h}^{(i)}$ is the i-th eigenvector of (11).

A.6. The previously estimated canonical vectors have converged to the correct solutions $\hat{\mathbf{h}}^{(i)} = \mathbf{h}^{(i)}$, which will be proved by induction.

Based on these assumptions, the ODE update function is given by

$$\begin{aligned}\bar{f}^{(i)}(\hat{\beta}^{(i)}\hat{\mathbf{h}}^{(i)}, \mathbf{x}) &= \lim_{n\to\infty} E[\hat{\beta}^{(i)}(n)\hat{\mathbf{h}}^{(i)}(n) - \hat{\beta}^{(i)}(n-1)\hat{\mathbf{h}}^{(i)}(n-1)] \\ &= \frac{\mathrm{d}(\hat{\beta}^{(i)}(t)\hat{\mathbf{h}}^{(i)}(t))}{\mathrm{d}t} \\ &= \frac{1}{M}\hat{\mathbf{P}}_{\mathbf{U}}^{(i)}\mathbf{D}^{-1}\mathbf{R}\hat{\mathbf{h}}^{(i)}(t) - \hat{\beta}^{(i)}(t)\hat{\mathbf{h}}^{(i)}(t),\end{aligned} \tag{21}$$

and in order to find the stationary points of this ODE we can write

$$\hat{\mathbf{h}}^{(i)}(t) = \sum_{j=1}^{m}\hat{\theta}_j^{(i)}(t)\mathbf{h}^{(j)}, \tag{22}$$

where $m = \sum_{k=1}^{M} m_k$ is the number of eigenvectors of the GEV problem and $\hat{\theta}_j^{(i)}(t)$ is a time-varying projection. Considering the energy ($\mathbf{h}^{(j)H}\mathbf{D}\mathbf{h}^{(j)}/M = 1$) and orthogonality ($\mathbf{h}^{(j)H}\mathbf{D}\mathbf{h}^{(i)} = 0, i \neq j$) constraints, left-multiplying (19) and (22) by $\mathbf{h}^{(j)H}\mathbf{D}$ (with $j < i$), and taking A.6 into account, it is easy to realize that $\mathbf{h}^{(j)H}\mathbf{D}\hat{\mathbf{h}}^{(i)}(t) = M\hat{\theta}_j^{(i)}(t) = 0$, and then

$$\hat{\mathbf{h}}^{(i)}(t) = \sum_{j=i}^{m}\hat{\theta}_j^{(i)}(t)\mathbf{h}^{(j)}. \tag{23}$$

Hence, using (23) and taking into account the orthogonality constraints, (21) can be rewritten as

$$\frac{\mathrm{d}(\hat{\beta}^{(i)}(t)\hat{\theta}_j^{(i)}(t))}{\mathrm{d}t} = (\beta^{(j)} - \hat{\beta}^{(i)}(t))\hat{\theta}_j^{(i)}(t), \quad j = i, \dots, m,$$

and defining

$$\hat{\alpha}_j^{(i)}(t) = \frac{\hat{\theta}_j^{(i)}(t)}{\hat{\theta}_i^{(i)}(t)}, \quad j = i, \dots, m,$$

we obtain

$$\frac{\mathrm{d}(\hat{\alpha}_j^{(i)}(t))}{\mathrm{d}t} = -\hat{\alpha}_j^{(i)}(t)\frac{\beta^{(i)} - \beta^{(j)}}{\hat{\beta}^{(i)}(t)}, \quad j = i, \dots, m.$$

Noting that $\hat{\beta}^{(i)}(t) > 0$ and $\beta^{(i)} > \beta^{(j)}$, it can be easily shown that the time varying projections associated with all the eigenvectors except the i-th decay to zero asymptotically, and hence, as $t \to \infty$, $\hat{\mathbf{h}}^{(i)}(t) = c\mathbf{h}^{(i)}$ and $\hat{\beta}^{(i)}(t) = \beta^{(i)}$, where c is an arbitrary constant. Then, we conclude that the generalized eigenvector $\mathbf{h}^{(i)}$ is the stable stationary point of the ODE.

In order to prove that the remaining $m-1$ eigenvectors are saddle points, we linearize the ODE in (21) around the vicinity of a stationary point. The linearization matrix $\mathbf{A}_j^{(i)}$ can be computed as

$$\begin{aligned}\mathbf{A}_j^{(i)} &= \left.\frac{\mathrm{d}(\bar{f}^{(i)}(\hat{\beta}^{(i)}(t)\hat{\mathbf{h}}^{(i)}(t)))}{\mathrm{d}(\hat{\beta}^{(i)}(t)\hat{\mathbf{h}}^{(i)}(t))}\right|_{\hat{\beta}^{(i)}(t)\hat{\mathbf{h}}^{(i)}(t)=\beta^{(j)}\mathbf{h}^{(j)}} \\ &= \frac{1}{\beta^{(j)}}\hat{\mathbf{P}}_{\mathbf{U}}^{(i)}\mathbf{D}^{-1}\mathbf{R} - \mathbf{I},\end{aligned}$$

and, taking A.6 into account, the eigenvalues of $\mathbf{A}_j^{(i)}$ are given by

$$\beta^{(k)}_{\mathbf{A}_j^{(i)}} = \begin{cases} -1 & k < i, \\ \dfrac{\beta^{(k)}}{\beta^{(j)}} - 1 & k \geq i. \end{cases}$$

Here, it is easy to realize that only for $j = i$, all the eigenvalues which are analogous to the s-poles are within the LHP except the i-th which is at zero. All other stationary points have one or more poles in the RHP and hence they are saddle points, which means that near convergence, if the estimated canonical vector $\hat{\mathbf{h}}^{(i)}(t)$ reaches any of the $m-1$ saddle points, it will diverge from that point and converge only to the stable stationary point which is $\mathbf{h}^{(i)}$ (see Benveniste et al. (1990), Rao and Principe (2002) and Rao et al. (2004) for more details). Furthermore, we must note that for $i = 1$, we have $\hat{\mathbf{P}}_{\mathbf{U}}^{(i)} = \mathbf{I}$ and the main CCA solution is obtained, which validates A.6.

8. Comparison with other techniques

In this section, the CCA–RLS adaptive algorithm is compared with other learning rules for CCA of two or several data sets. The comparison is made in terms of performance, applicability to several data sets, speed of convergence and computational complexity.

8.1. Proposed algorithm (CCA–RLS)

The proposed adaptive algorithm obtains the solution of the classical CCA–MAXVAR problem by means of M coupled RLS algorithms (the reference signal $\mathbf{z}^{(i)}$ is constructed from the M previous estimates). The advantages of the RLS includes the presence of only one parameter (the forgetting factor λ) and its faster convergence in comparison with gradient-based algorithms, especially for colored data sets. Taking into account that the computational complexity of the RLS algorithm is of order $\mathcal{O}(q^2)$, where q is the length of the estimated vector, the cost of the proposed algorithm is

$$\mathcal{O}\left(\sum_{k=1}^{M} m_k^2\right),$$

per iteration and eigenvector.

8.2. Gradient descent methods (Fyfe and Fyfe-NL)

In Gou and Fyfe (2004) and Lai and Fyfe (1999) the authors propose two different adaptive algorithms for CCA of two data sets, both of them are based on gradient descent and they differ in the update of the Lagrange multipliers (or canonical correlations), which is made by means of gradient ascent (Lai & Fyfe, 1999) (Fyfe) or a nonlinear function (Gou & Fyfe, 2004) (Fyfe-NL). The learning rules of the algorithms are equivalent to two coupled LMS algorithms, which implies a low computational cost. Although Lai and Fyfe (1999) propose an extension of the algorithm to three data sets, this generalization is not one of the classical generalizations proposed by Kettenring (1971). Furthermore, the generalization is based on the maximization of the canonical correlations between pairs of canonical variates, which implies the definition of three different Lagrange multipliers or canonical correlations. The computational cost of the extended algorithm is

$$\mathcal{O}\left(\sum_{k=1}^{M} m_k\right),$$

and the main drawbacks are its slow convergence and, in the case of Lai and Fyfe (1999), the existence of two different learning rates for the canonical vectors and correlations.

Table 1
Comparison of several adaptive CCA algorithms

Algorithm	MAXVAR	Cost	Speed
CCA–RLS	Yes	$\sum_{k=1}^{M} m_k^2$	Fast
Fyfe, Fyfe-NL	No	$\sum_{k=1}^{M} m_k$	Slow
GD	No	$\sum_{k=1}^{M} m_k$	Slow
PM	Yes	$\sum_{k=1}^{M} \sum_{l=k}^{M} m_k m_l$	Fast

8.3. Gradient descent (GD)

Pezeshki et al. (2003) present an adaptive algorithm for CCA of two data sets which is based on a gradient descent technique. The algorithm is very similar to Lai and Fyfe (1999), but the estimate of the canonical correlation is obtained as the averaged correlation between the complete estimated canonical variates, which constitutes a drawback in the tracking ability of the algorithm. The computational cost of the algorithm is summarized in Table 1.

8.4. Power method (PM)

In Pezeshki et al. (2005) a method for CCA of two data sets based on an iterative power method is proposed. The main advantage of this method is its fast convergence in comparison with gradient-based techniques. Its computational cost can be estimated as

$$\mathcal{O}\left(\sum_{k=1}^{2} \sum_{l=k}^{2} m_k m_l\right),$$

which is due to the rank one update of the correlation matrices $\mathbf{R}_{kl}$. Although the algorithm could be easily extended to perform CCA–MAXVAR, the estimation of all the cross-correlation matrices $\mathbf{R}_{kl}$ implies a high increase in computational cost, mainly for high dimensional data matrices or a large number of data sets.

9. Simulation results

In this section the performance of the proposed algorithm is analyzed by means of some simulation examples. For all the examples, the convergence curves are based on the averaged results of 300 independent realizations. For each example we obtain the estimated canonical correlation $\hat{\rho}^{(i)}$ and the mean squared error (MSE) of the estimated eigenvectors $\hat{\mathbf{h}}^{(i)}$ or PCA approximations $\hat{\mathbf{z}}^{(i)}$. The parameters of the algorithm are initialized as follows:

- $\mathbf{P}_{\mathbf{x}_k}(0) = 10^5\mathbf{I}$, for $k = 1, \ldots, M$ (M is the number of data sets).
- $\mathbf{h}^{(i)}(0)$ is initialized with random values, for $i = 1, \ldots, p$, where p is the number of canonical vectors of interest.
- $\beta^{(i)}(0) = 0$ for $i = 1, \ldots, p$.

In the first example, a two data set ($M = 2$) CCA problem with canonical correlations $\rho^{(1)} = 0.9$ and $\rho^{(2)} = 0$ is generated. We compare the performance of the proposed

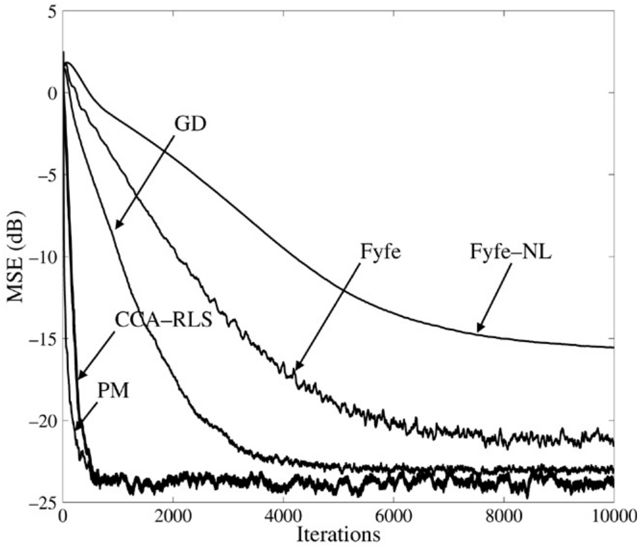

Fig. 3. Performance of the proposed algorithm, and the methods proposed by Lai and Fyfe (1999) (Fyfe), Gou and Fyfe (2004) (Fyfe-NL), Pezeshki et al. (2003) (GD) and Pezeshki et al. (2005) (PM). Two data sets, $\rho^{(1)} = 0.9$, $\rho^{(2)} = 0$.

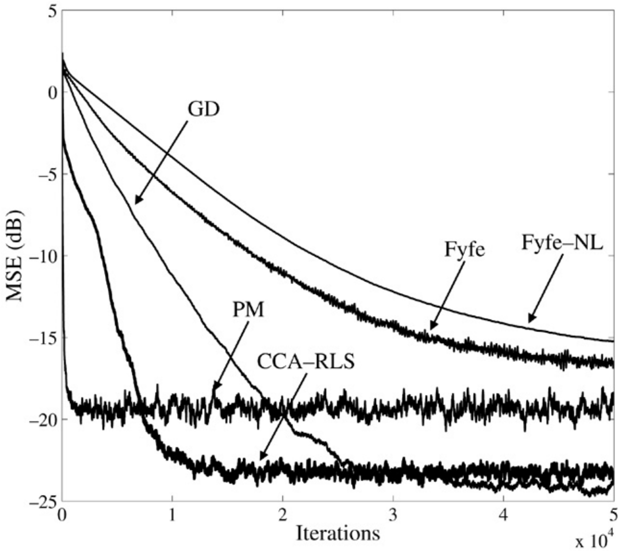

Fig. 4. Performance of the proposed algorithm, and the methods proposed by Lai and Fyfe (1999) (Fyfe), Gou and Fyfe (2004) (Fyfe-NL), Pezeshki et al. (2003) (GD) and Pezeshki et al. (2005) (PM). Two data sets, $\rho^{(1)} = 0.9$, $\rho^{(2)} = 0.8$.

adaptive CCA–RLS algorithm, the methods proposed in Lai and Fyfe (1999) (Fyfe in the figures) and Gou and Fyfe (2004) (Fyfe-NL), the CCA gradient-descent (GD) technique proposed by Pezeshki et al. (2003) and the CCA power method (PM in the figures) proposed by Pezeshki et al. (2005).

The dimensions of the data sets are $m_1 = 6$ and $m_2 = 4$, the forgetting factor for the CCA–RLS and PM algorithms is $\lambda = 0.995$, and the step sizes for the GD, Fyfe and Fyfe-NL algorithms are $\mu = 0.05, \mu = 0.025$ and $\mu = 0.01$, respectively. The results for the extraction of the first CCA canonical vectors are shown in Fig. 3. As expected, the gradient-based algorithms are the slowest in terms of convergence speed, and the PM algorithm is slightly faster than the CCA–RLS.

The same experiment has been repeated in the second example, but now the canonical correlations are $\rho^{(1)} = 0.9$ and $\rho^{(2)} = 0.8$, and the step sizes for the GD and Fyfe algorithms have been reduced to $\mu = 0.03, \mu = 0.02$, respectively, to ensure convergence. The results for the extraction of the first CCA solution are shown in Fig. 4, where we can see that the performance of the PM algorithm is degraded in the presence of close canonical correlations, whereas the CCA–RLS algorithm conserves its MSE performance at the expense of a convergence speed reduction.

The third example compares the performance of the CCA–RLS algorithm with the generalization to three data sets proposed by Lai and Fyfe (1999) of the algorithms Fyfe and Fyfe-NL. The CCA problem is similar to the example presented in Lai and Fyfe (1999), where three artificial data sets of dimensions $m_1 = m_2 = m_3 = 3$ have been generated as

$$\mathbf{X}_k = (\mathbf{S}_k + [\ \mathbf{s} \quad \mathbf{0} \quad \mathbf{0}\ ])\mathbf{C}_k, \quad k = 1, 2, 3,$$

where $\mathbf{S}_k$ is a $N \times 3$ matrix with elements drawn from $\mathcal{N}(0, 1)$, $\mathbf{s}$ is a $N \times 1$ vector drawn from $\mathcal{N}(0, \sigma)$, and $\mathbf{C}_k$ is a 3×3 mixing matrix. As pointed out in Lai and Fyfe (1999), the Fyfe and Fyfe-NL algorithms are based on the maximization of three separated constrained objective functions. The three algorithms have been compared in terms of the extraction of the common signal $\mathbf{s}$ with two different values of σ and two different selections of the mixing matrices. The estimated common signal is obtained in the Fyfe and Fyfe-NL algorithms by means of a PCA approximation of the three estimated canonical variates (two stage algorithms), whereas the CCA–RLS obtains the PCA estimate $\hat{\mathbf{z}}$ in one single step. The forgetting factor for the CCA–RLS algorithm is $\lambda = 0.99$ and the step size for the Fyfe and Fyfe-NL algorithms is $\mu = 0.005$. Fig. 5 shows the results in the case of $\mathbf{C}_1 = \mathbf{C}_2 = \mathbf{C}_3 = \mathbf{I}$, and Fig. 6 shows the results for

$$\mathbf{C}_1 = \begin{bmatrix} 0.0413 & 2.9003 & 1.6701 \\ -0.8912 & 5.3215 & 2.6235 \\ 0.2363 & -0.9105 & -0.8995 \end{bmatrix},$$

$$\mathbf{C}_2 = \begin{bmatrix} 0.1464 & 0.9850 & -1.2749 \\ 0.3724 & 0.0055 & 0.4732 \\ -0.3791 & -0.3388 & 0.7100 \end{bmatrix},$$

$$\mathbf{C}_3 = \begin{bmatrix} -0.2779 & -0.7311 & -0.0784 \\ 0.4944 & -0.0415 & 0.5029 \\ -0.5102 & 0.8189 & 0.5401 \end{bmatrix},$$

whose condition numbers are 18.12, 7.10 and 2.41 respectively. As can be seen, the CCA–RLS is much faster than the Fyfe and Fyfe-NL algorithms, specially for colored signals, which is a direct consequence of the convergence properties of the RLS algorithm.

The fourth example shows the convergence properties of the CCA–RLS algorithm, with four complex data sets of dimensions $m_1 = 40$, $m_2 = 30, m_3 = 20$ and $m_4 = 10$. The first four generalized canonical correlations are $\rho^{(1)} = 0.9, \rho^{(2)} = 0.8, \rho^{(3)} = 0.7$ and $\rho^{(4)} = 0.6$, and the forgetting factor has been selected as $\lambda = 0.99$. Fig. 7 shows the estimated

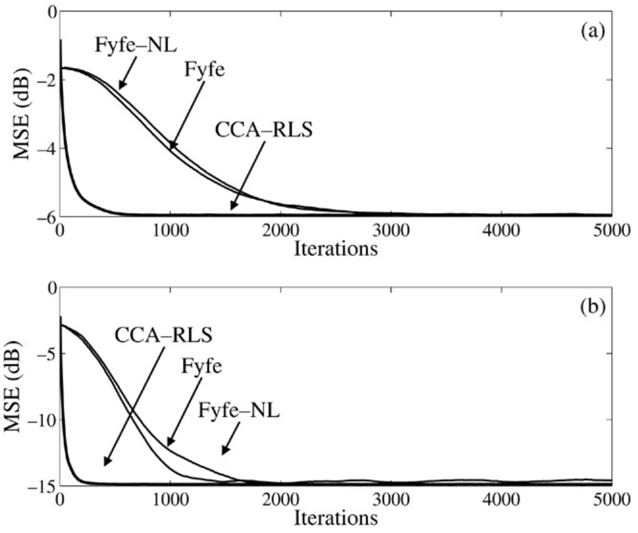

Fig. 5. Comparison of CCA–RLS and Fyfe algorithms. Three data sets. (a) $\sigma^2 = 1$, (b) $\sigma^2 = 10$. Mixing matrices $\mathbf{C}_1 = \mathbf{C}_2 = \mathbf{C}_3 = \mathbf{I}$.

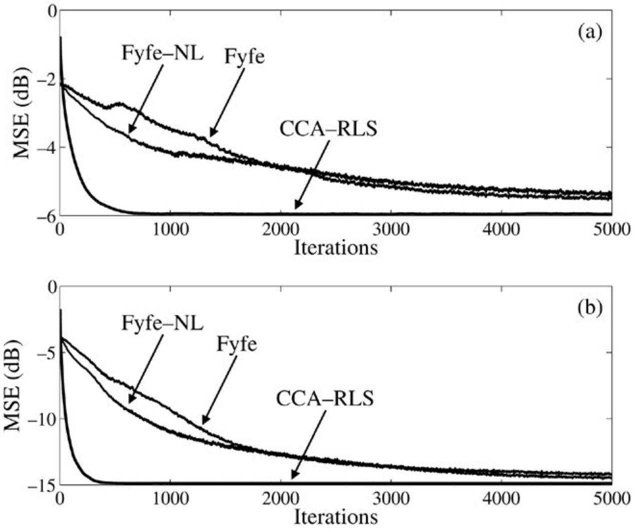

Fig. 6. Comparison of CCA–RLS and Fyfe algorithms. Three data sets. (a) $\sigma^2 = 1$, (b) $\sigma^2 = 10$. Mixing matrices with $\text{cond}(\mathbf{C}_1) = 18.12$, $\text{cond}(\mathbf{C}_2) = 7.10$, $\text{cond}(\mathbf{C}_3) = 2.41$.

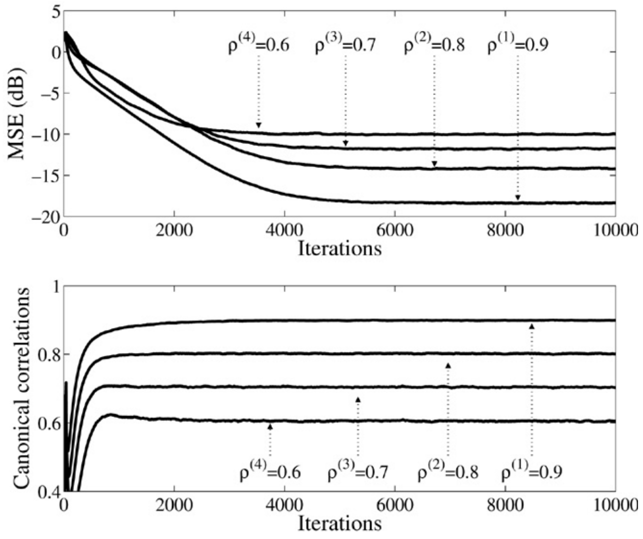

Fig. 7. Convergence properties of the CCA–RLS algorithm. Four data sets.

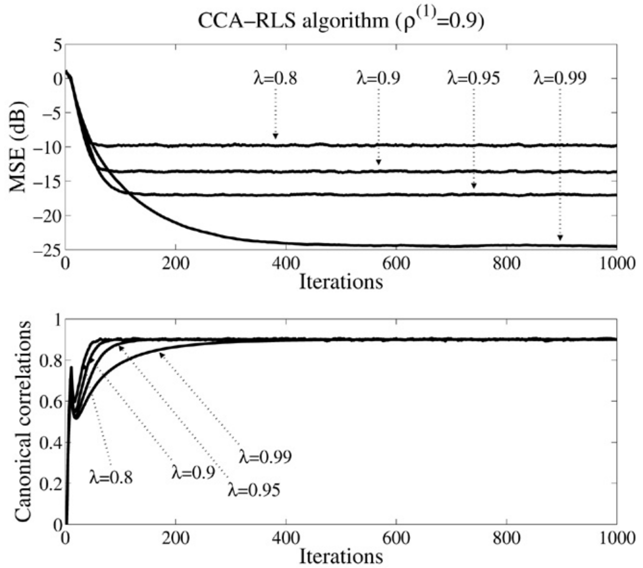

Fig. 8. Effect of the forgetting factor λ. Four data sets. $\rho^{(1)} = 0.9$.

canonical correlations and the MSE in the extraction of the canonical vectors, where we can see that they converge very fast to their theoretical values.

In the fifth example we have analyzed the effect of the canonical correlation and the forgetting factor λ on the performance of the CCA–RLS algorithm. We have simulated two CCA problems with four data sets, whose dimensions have been selected as $m_1 = 10$, $m_2 = 8$, $m_3 = 6$ and $m_4 = 4$. In the first problem, the main generalized canonical correlation was $\rho^{(1)} = 0.9$; and in the second one it was $\rho^{(1)} = 0.7$. The simulation results are shown in Figs. 8 and 9, where we can see that, after a transitory interval, the proposed algorithm converges to the theoretical solutions providing accurate estimates. Furthermore, Figs. 8 and 9 illustrate the effect of the canonical correlations on the performance of the proposed algorithm, as well as the trade off between the speed of convergence and the final MSE, which depends on λ.

Finally, the proposed algorithm has been tested in the Boston housing data set (http://www.ics.uci.edu/~mlearn/MLRepository.html), which gives housing values in the suburbs of Boston. The data set contains $N = 506$ 14-dimensional instances, which have been preprocessed to have zero mean and unit variance, and have been divided into $M = 3$ different data sets with dimensions $m_1 = 5$, $m_2 = 6$ and $m_3 = 3$:

- *First data set* (ZN, AGE, TAX, RM, MEDV): Variables directly related with the housing market.
- *Second data set* (CRIM, INDUS, NOX, PTRATIO, B, LSTAT): Variables indirectly related with the housing market.
- *Third data set* (CHAS, DIS, RAD): Geographical variables.

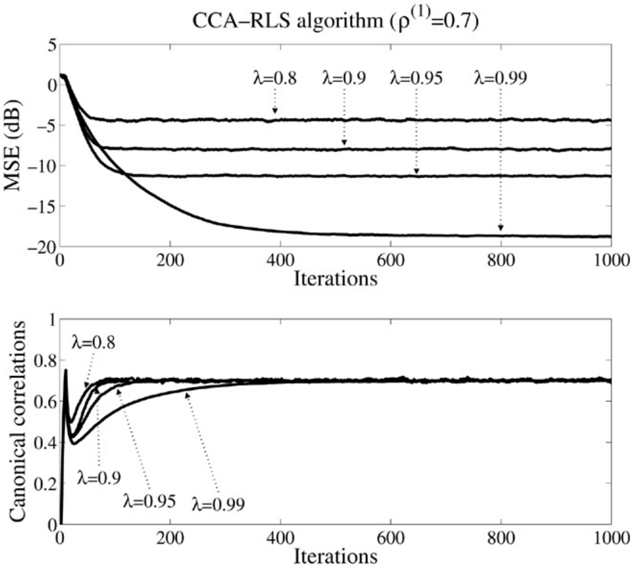

Fig. 9. Effect of the forgetting factor λ. Four data sets. $\rho^{(1)} = 0.7$.

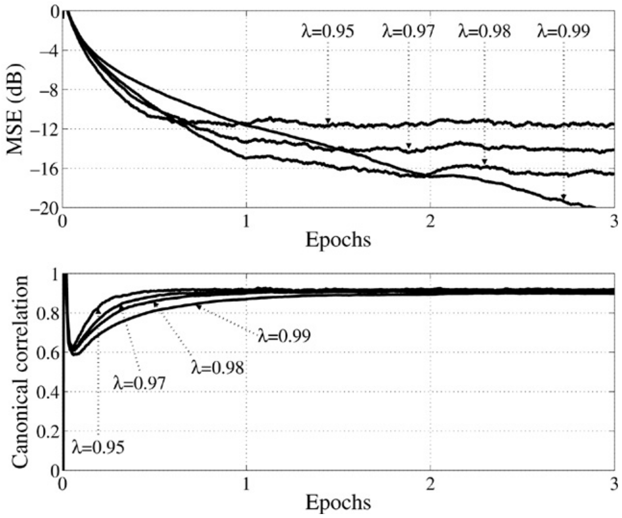

Fig. 10. Performance of the CCA–RLS algorithm in the Boston housing example. $N = 506$, three epochs. Four different forgetting factors λ.

Using the estimated correlation matrices we have found that the first canonical correlation is $\hat{\rho}^{(1)} = 0.9313$ and the canonical vectors are:

$$\mathbf{h}_1^{(1)} = [-0.1700, 0.2115, 0.5365, -0.0086, -0.0065]^{\mathrm{T}},$$
$$\mathbf{h}_2^{(1)} = [0.1716, 0.2126, 0.3806, 0.1731, -0.0473, 0.0057]^{\mathrm{T}},$$
$$\mathbf{h}_3^{(1)} = [-0.0245, -0.3920, 0.4821]^{\mathrm{T}}.$$

Fig. 10 shows the results of the application of the proposed CCA–RLS algorithm over 3 epochs of the available data ($3N$ RLS iterations). Obviously, the trade off between the forgetting factor λ and the final MSE is still present, but as can be seen, the proposed algorithm is able to obtain good estimates of the canonical vectors and correlations even in only one epoch.

10. Conclusions

In this work we have proposed a neural model and the corresponding learning rules for the generalization of canonical correlation analysis (CCA) to several data sets. We have proved that the proposed CCA generalization, which is based on the maximization of correlations (or minimization of distances), is equivalent to the maximum variance (MAXVAR) PCA-based classical generalization proposed by Kettenring in the early seventies. The main advantage of this reformulation of CCA as a set of coupled least squares (LS) regression problems is that it allows us to develop batch and adaptive learning algorithms for CCA in a natural way.

The convergence of the proposed adaptive algorithm has been proved by means of stochastic approximation techniques and its performance has been analyzed by means of simulations, showing a fast convergence and even outperforming other adaptive CCA algorithms specifically designed for the case of two data sets. Further lines in this work include the extension of the proposed algorithms to kernel CCA and their application to blind equalization of multiple-input multiple-output (MIMO) systems and blind source separation (BSS) of convolutive mixtures.

Appendix. Solution of the CCA–LS generalization

In this appendix we show that the CCA–LS generalization is equivalent to the GEV problem defined in (9). Using the definitions of $\mathbf{R}$ and $\mathbf{D}$ in (10) we can use the matrix formulation and state the CCA problem as the problem of maximizing

$$\rho^{(i)} = \frac{1}{M(M-1)} \mathbf{h}^{(i)\mathrm{H}} (\mathbf{R} - \mathbf{D}) \mathbf{h}^{(i)},$$

subject to the following restrictions

$$\frac{1}{M} \mathbf{h}^{(i)\mathrm{H}} \mathbf{D} \mathbf{h}^{(i)} = 1,$$

$$\mathbf{z}^{(i)\mathrm{H}} \mathbf{z}^{(j)} = \frac{1}{M^2} \mathbf{h}^{(i)\mathrm{H}} \mathbf{R} \mathbf{h}^{(j)} = 0 \quad j = 1, \ldots, i-1.$$

Applying the method of Lagrange multipliers, the cost function to be maximized on $\mathbf{h}^{(i)}$, and minimized on the Lagrange multipliers $\lambda^{(i)} \in \mathbb{R}$ and $\gamma^{(ij)} \in \mathbb{C}$ is

$$J^{(i)} = \frac{1}{M(M-1)} \mathbf{h}^{(i)\mathrm{H}} (\mathbf{R} - \mathbf{D}) \mathbf{h}^{(i)} + \lambda^{(i)} \left(1 - \frac{1}{M} \mathbf{h}^{(i)\mathrm{H}} \mathbf{D} \mathbf{h}^{(i)} \right) - \frac{1}{M^2} \sum_{j=1}^{i-1} \gamma^{(ij)} \mathbf{h}^{(i)\mathrm{H}} \mathbf{R} \mathbf{h}^{(j)},$$

and equating to zero the gradient vector $\nabla_{\mathbf{h}^{(i)*}}(J^{(i)})$ we can write

$$\frac{1}{(M-1)} (\mathbf{R} - \mathbf{D}) \mathbf{h}^{(i)} = \lambda^{(i)} \mathbf{D} \mathbf{h}^{(i)} + \frac{1}{M} \sum_{j=1}^{i-1} \gamma^{(ij)} \mathbf{R} \mathbf{h}^{(j)}. \quad \text{(A.1)}$$

In order to obtain the Lagrange multipliers, (A.1) can be left-multiplied by $\mathbf{h}^{(i)\mathrm{H}}$, which, applying the restrictions, implies $\lambda^{(i)} = \rho^{(i)}$. Analogously, left-multiplying (A.1) by $\mathbf{h}^{(j)H}$ ($j =$

$1, \ldots, i-1$) and taking into account the restrictions we can write

$$-\left(\frac{1}{(M-1)} + \lambda^{(i)}\right)\mathbf{h}^{(j)H}\mathbf{D}\mathbf{h}^{(i)} = \frac{1}{M}\gamma^{(ij)}\mathbf{h}^{(j)H}\mathbf{R}\mathbf{h}^{(j)}.$$

Now, assuming that $\frac{1}{(M-1)}(\mathbf{R}-\mathbf{D})\,\mathbf{h}^{(j)} = \lambda^{(j)}\mathbf{D}\mathbf{h}^{(j)}$ (which, for $j = 1$ is a direct consequence of (A.1)), it is straightforward to prove that $\mathbf{h}^{(j)H}\mathbf{R}\mathbf{h}^{(i)} = 0$ implies $\mathbf{h}^{(j)H}\mathbf{D}\mathbf{h}^{(i)} = 0$, and then

$$\gamma^{(ij)}\mathbf{h}^{(j)H}\mathbf{R}\mathbf{h}^{(j)} = 0.$$

Finally, taking into account that $\mathbf{R}$ is semi-positive definite we have $\gamma^{(ij)}\mathbf{R}\mathbf{h}^{(j)} = \mathbf{0}$, which implies that (A.1) can be rewritten as

$$\frac{1}{(M-1)}(\mathbf{R}-\mathbf{D})\,\mathbf{h}^{(i)} = \lambda^{(i)}\mathbf{D}\mathbf{h}^{(i)},$$

where $\lambda^{(i)} = \rho^{(i)}$. This validates our assumption and concludes the proof. □

References

Bach, F. R., & Jordan, M. I. (2002). Kernel independent component analysis. *Journal of Machine Learning Research*, *3*, 1–48.

Benveniste, A., Métivier, M., & Priouret, P. (1990). *Applications of mathematics*: *Vol. 22*. *Adaptive algorithms and stochastic approximations*. Berlin: Springer Verlag.

Borga, M. (1998). Learning multidimensional signal processing, Ph.D. thesis. Linköping, Sweden: Linköping University.

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

Dogandzic, A., & Nehorai, A. (2002). Finite-length MIMO equalization using canonical correlation analysis. *IEEE Transactions on Signal Processing*, *SP-50*, 984–989.

Dogandzic, A., & Nehorai, A. (2003). Generalized multivariate analysis of variance; A unified framework for signal processing in correlated noise. *IEEE Signal Processing Magazine*, *20*, 39–54.

Friman, O., Borga, M., Lundberg, P., & Knutsson, H. (2003). Adaptive analysis of fMRI data. *Neuroimage*, *19*(3), 837–845.

Gou, Z., & Fyfe, C. (2004). A canonical correlation neural network for multicollinearity and functional data. *Neural Networks*, *17*(2), 285–293.

Hardoon, D. R., Szedmak, S., & Shawe-Taylor, J. (2004). Canonical correlation analysis; An overview with application to learning methods. *Neural Computation*, *16*(12), 2639–2664.

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

Hsieh, W. W. (2000). Nonlinear canonical correlation analysis by neural networks. *Neural Networks*, *13*(10), 1095–1105.

Kettenring, J. R. (1971). Canonical analysis of several sets of variables. *Biometrika*, *58*, 433–451.

Lai, P. L., & Fyfe, C. (1999). A neural implementation of canonical correlation analysis. *Neural Networks*, *12*(10), 1391–1397.

Lai, P. L., & Fyfe, C. (2000). Kernel and nonlinear canonical correlation analysis. *International Journal of Neural Systems*, *10*(5), 365–377.

Pezeshki, A., Azimi-Sadjadi, M. R., & Scharf, L. L. (2003). A network for recursive extraction of canonical coordinates. *Neural Networks*, *16*(5–6), 801–808.

Pezeshki, A., Scharf, L. L., Azimi-Sadjadi, M. R., & Hua, Y. (2005). Two-channel constrained least squares problems: Solutions using power methods and connections with canonical coordinates. *IEEE Transactions on Signal Processing*, *53*, 121–135.

Rao, Y. N., & Principe, J. C. (2002). Robust on-line principal component analysis based on a fixed-point approach. In *Proc. of 2002 IEEE international conference on acoustics, speech and signal processing*: *Vol. 1* (pp. 981–984).

Rao, Y. N., Principe, J. C., & Wong, T. F. (2004). Fast RLS-like algorithm for generalized eigendecomposition and its applications. *Journal of VLSI Signal Processing-Systems for Signal, Image, and Video Technology*, *37*(2), 333–344.

Vía, J., & Santamaría, I. (2005). Adaptive blind equalization of SIMO systems based on canonical correlation Analysis. In: *Proc. of IEEE workshop on signal processing advances in wireless communications*. New York, USA.

Vía, J., Santamaría, I., & Pérez, J. (2005a). A robust RLS algorithm for adaptive canonical correlation analysis. In: *Proc. of 2005 IEEE international conference on acoustics, speech and signal processing*. Philadelphia, USA.

Vía, J., Santamaría, I., & Pérez, J. (2005b). Canonical correlation analysis (CCA) algorithms for multiple data sets: Application to blind SIMO equalization. *European signal processing conference, EUSIPCO*. Antalya, Turkey.