FAST SUBSPACE TRACKING AND NEURAL NETWORK LEARNING BY A NOVEL INFORMATION CRITERION

Yongfeng Miao and Yingbo Hua

Department of Electrical and Electronic Engineering
The University of Melbourne
Parkville, Victoria 3052, AUSTRALIA

ABSTRACT

We introduce a novel information criterion (NIC) for searching for the optimum weights of a two-layer linear neural network (NN). The NIC exhibits a single global maximum attained if and only if the weights span the (desired) principal subspace of a covariance matrix. The other stationary points of the NIC are (unstable) saddle points. We develop an adaptive algorithm based on the NIC for estimating and tracking the principal subspace of a vector sequence. The NIC algorithm provides a fast on-line learning of the optimum weights for the two-layer linear NN. The NIC algorithm has several key advantages such as faster convergence which is illustrated through analysis and simulation.

1. INTRODUCTION

It is known that there is a close relationship between a class of linear neural networks (NNs) and the concept of principal subspace [1]. The concept of principal subspace is often referred to as principal subspace analysis (PSA) in the context of statistical analysis. When one is interested in the orthonormal eigenvectors spanning a principal subspace, the so-called principal component analysis (PCA) is then referred to. Both PSA and PCA also represent the desired function of a class of linear NNs when the weights of the NNs span a principal subspace. The process of finding the proper weights is called "learning".

The well-known Oja's algorithm [2] was developed based on some heuristic reasoning. The global convergence property of Oja's algorithm remained as a mystery for some time until the analyses done in [3, 4]. An improved version of Oja's algorithm, called the LMSER algorithm, was developed in [5] where the well-known concept of gradient searching was applied to minimize a mean squared error (MSE). Unlike Oja's algorithm, the LMSER algorithm could be claimed to be globally convergent since the global minimum of the MSE is only achieved by the principal subspace and all the other stationary points of the MSE are saddle points. The MSE has also led to many other algorithms which include the projection approximation subspace tracking (PAST) algorithm [6]. It is clear that a properly chosen criterion is a very important part in developing any learning algorithm.

In this paper, we introduce a novel information criterion (NIC) for searching for the optimum weights of a two-layers linear NN. The NIC exhibits a single global maximum attained if and only if the weights span the (desired) principal subspace of a covariance matrix. The other stationary points of the NIC are (unstable) saddle points. Unlike the MSE, the NIC is non-quadratic and has a steep peak around the global maximum. Applying gradient ascent searching to the NIC yields the NIC algorithm which is globally convergent and fast.

In Section 2, we propose the NIC formulation for PSA and depicts its landscape picture. The NIC algorithm is derived in Section 3 with comparisons to a number of existing PCA/PSA algorithms. Section 4 deals with the global convergence analysis of the NIC algorithm using the Lyapunov function approach. In Section 5, the performance of the NIC algorithm is evaluated through simulation examples. Conclusions are drawn in Section 6.

2. NOVEL INFORMATION CRITERION FORMULATION FOR PSA

Given $\mathbf{W}$ in the domain $D = \{\mathbf{W}|\mathbf{W}^T\mathbf{R}\mathbf{W} > \mathbf{0}\}$, we propose the following criterion for PSA

$$J_{NIC}(\mathbf{W}) = \frac{1}{2}tr\{log(\mathbf{W}^T\mathbf{R}\mathbf{W}) - (\mathbf{W}^T\mathbf{W})\} \qquad (1)$$

where $\mathbf{R}$ denotes a covariance matrix. The landscape of NIC is depicted by the following two theorems.

Theorem 2.1 [8] $\mathbf{W}$ is a stationary point of $J_{NIC}(\mathbf{W})$ in the domain D if and only if $\mathbf{W} = \mathbf{U}_r\mathbf{Q}$, where $\mathbf{U}_r \in$

$\Re^{n\times r}$ contains any r distinct orthonormal eigenvectors of $\mathbf{R}$ and $\mathbf{Q}$ is an arbitrary orthogonal matrix.

Theorem 2.2 [8] In the domain D, $J_{NIC}(\mathbf{W})$ has a global maximum which is attained when and only when $\mathbf{W} = \mathbf{U}_1\mathbf{P}_r\mathbf{Q}$ with $\mathbf{U}_1$ composed of the first r principal orthonormal eigenvectors of $\mathbf{R}$, $\mathbf{P}_r$ an $r \times r$ permutation matrix and $\mathbf{Q}$ an arbitrary orthogonal matrix. All other stationary points are saddle points of $J_{NIC}(\mathbf{W})$.

Theorem 2.1 establishes a property of all the stationary points of $J_{NIC}(\mathbf{W})$. Theorem 2.2 further distinguishes the global maximum attained by $\mathbf{W}$ spanning the principal subspace from all other stationary points which are saddle points. From these two theorems, we note the NIC has the following attractive properties:

- The NIC has a global maximum at the principal subspace while all other stationary points are saddle points. Therefore, the gradient ascent searching is guaranteed to converge to the desired principal subspace for proper initializations of $\mathbf{W}$.
- $\mathbf{W}$ is orthonormal at the maximum, and hence no explicit orthonormality constraint is needed. In fact, it is shown in [8] that the rate at which $\mathbf{W}$ orthonormalizes itself by the NIC algorithm is a constant independent of the eigenvalues of $\mathbf{R}$.
- Compared with the quadratic MSE [5], the NIC is non-quadratic and has a steeper landscape around the principal subspace. Therefore, the gradient searching of the NIC is expected to converge faster than that of the MSE.
- Under certain conditions, the principal subspace maximizes the MIC [9]

$$J_{MIC}(\mathbf{W}) = \frac{1}{2}tr\{log(\mathbf{W}^T\mathbf{R}\mathbf{W}) - log(\mathbf{W}^T\mathbf{W})\} \quad (2)$$

which appears almost the same as $J_{NIC}(\mathbf{W})$ except for the logarithm in the second term. However, we note that unconstrained $J_{MIC}(\mathbf{W})$ does not yield an effective PSA criterion because the maximum value of the MIC may not be achieved only by the orthonormal principal subspace [8].

3. THE NIC LEARNING ALGORITHM

The NIC algorithm admits both the batch-mode matrix and the data-driven recursive least-squares (RLS) implementations, depending on whether or not the covariance matrix is directly involved in the computations. The RLS implementation provides an on-line learning algorithm for the two-layer linear NN with n inputs, r hidden neurons and n outputs.

3.1. Batch Implementation

From the gradient of $J_{NIC}(\mathbf{W})$ with respect to $\mathbf{W}$, we have the following gradient ascent rule for updating $\mathbf{W}_k$

$$\mathbf{W}_k = (1-\eta)\mathbf{W}_{k-1} + \eta\hat{\mathbf{R}}_k\mathbf{W}_{k-1}(\mathbf{W}_{k-1}^T\hat{\mathbf{R}}_k\mathbf{W}_{k-1})^{-1} \quad (3)$$

where $0 < \eta < 1$ denotes the learning step size, and $\hat{\mathbf{R}}_k$ denotes the estimate of the covariance matrix for k available samples. It can be obtained by the rank-1 update as

$$\hat{\mathbf{R}}_k = \alpha(k-1)\hat{\mathbf{R}}_{k-1}/k + \mathbf{x}_k\mathbf{x}_k^T/k \quad (4)$$

where $0 < \alpha \leq 1$ denotes the forgetting factor which is chosen in the range $(0, 1)$ to implement an effective window of size $1/(1-\alpha)$ for subspace tracking of non-stationary process, whereas $\alpha = 1$ is chosen for the neural network learning of stationary process.

Equations (3) and (4) represent the batch implementation of the NIC algorithm where the subspace is updated after each rank-1 update of the covariance matrix. It requires $O(M^2r)$ flops per update which is in contrast to $O(M^3)$ for the direct eigenvalue decomposition (EVD). The computations involved are simple matrix additions, multiplications, and inversions, which are ready for parallel implementations.

The Oja's algorithm [2], which is an approximate gradient rule to minimize the MSE, has the following update equation

$$\mathbf{W}_k = \mathbf{W}_{k-1} + \eta(\mathbf{I} - \mathbf{W}_{k-1}\mathbf{W}_{k-1}^T)\hat{\mathbf{R}}_k\mathbf{W}_{k-1} \quad (5)$$

where η is the learning step size which is dependent on both the initial choice of $\mathbf{W}_k$ and the eigenvalues of $\hat{\mathbf{R}}_k$. Comparing (3) and (5), we can see that the NIC algorithm actually extends Oja's algorithm by introducing a mechanism to adaptively adjust the step size at each step. The adaptive step size provides the advantage of fast convergence as will be shown by simulations.

3.2. RLS Implementation

By applying matrix inversion lemma to (3) and making the same projection approximation as in [6], the RLS implementation of the NIC algorithm follows:

$$\mathbf{y}_k = \mathbf{W}_{k-1}^T\mathbf{x}_k \quad (6)$$

$$\mathbf{g}_k = \frac{\alpha^{-1}\mathbf{P}_{k-1}\mathbf{y}_k}{1 + \alpha^{-1}\mathbf{y}_k^T\mathbf{P}_{k-1}\mathbf{y}_k} \quad (7)$$

$$\mathbf{P}_k = \alpha^{-1}\mathbf{P}_{k-1} - \alpha^{-1}\mathbf{g}_k\mathbf{y}_k^T\mathbf{P}_{k-1} \quad (8)$$

$$\tilde{\mathbf{W}}_k = \tilde{\mathbf{W}}_{k-1} + (\mathbf{x}_k - \tilde{\mathbf{W}}_{k-1}\mathbf{y}_k)\mathbf{g}_k^T \quad (9)$$

$$\mathbf{W}_k = (1-\eta)\mathbf{W}_{k-1} + \eta\tilde{\mathbf{W}}_k \quad (10)$$

The initialization of this data-driven NIC algorithm is similar to that of the standard RLS algorithm. The initial settings for convergence can be: $\mathbf{P}_0 = \delta \mathbf{I}_r$, where δ is a small positive number, $\tilde{\mathbf{W}}_0 = \mathbf{0}$, and $\mathbf{W}_0 =$ a random $M \times r$ matrix. From the above equations, it is easy to note that the computational complexity of (6-10) is $O(Mr)$ flops per update, resulting in a very cheap subspace tracking algorithm.

For the learning of two-layer NNs, $\tilde{\mathbf{W}}_k$ is first adjusted according to (9) and then $\mathbf{W}_k$ by (10). Equations (7-8) calculate $\mathbf{g}_k$ which adaptively adjusts the step size at time k. Equation (6) and the calculation of $\tilde{\mathbf{x}}_k = \tilde{\mathbf{W}}_{k-1}\mathbf{y}_k$ simply form the forward feeding path of the network. Figure 1 shows the block diagram of this learning process. It should be noted from the analysis in Section 4 that as $\mathbf{W}_k$ tends to $\mathbf{W}$, so does $\tilde{\mathbf{W}}_k$.

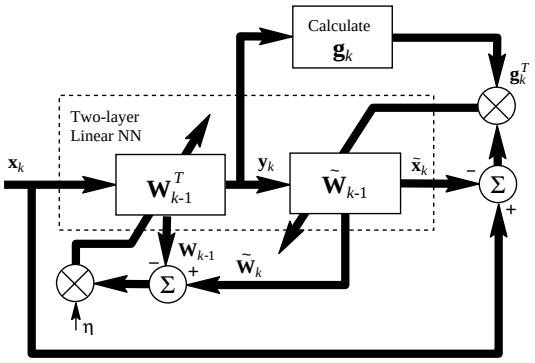

Figure 1: Block diagram of the two-layer linear NN learning using the NIC algorithm

Compared with other PSA/PCA algorithms, the RLS implementation of the NIC algorithm has the following properties:

- Unlike Oja's algorithm which uses a constant step size for learning, the NIC algorithm uses an adaptive step size $\mathbf{g}_k$ which is updated by equations (7) and (8).
- The NIC algorithm utilizes a sample covariance matrix which is updated by (4) while the the LMSER [5] and Oja's algorithm [2] only use the stochastic estimate $\hat{\mathbf{R}}_k = \mathbf{x}_k \mathbf{x}_k^T$.
- In the special case when $\eta \rightarrow 1$, (6-10) yield the set of update equations for the PAST algorithm [6]. For the NIC, η is in principle allowed to be any value within the interval $(0, 1)$. It is shown in [8] that the NIC essentially represents a robust improvement of the PAST.
- The NIC formulation and algorithm can be easily adapted to extract the individual eigenvalues and eigenvectors of the signal subspace when used in conjunction with the deflation technique as used by the APEX algorithm [7]. The NIC therefore provides some potential improvements over existing PCA algorithms based on Oja's algorithm [2].

4. GLOBAL CONVERGENCE ANALYSIS

Under the condition that $\mathbf{x}_k$ is from a stationary process and the step size η is small enough, the discrete-time difference equation (3) approximates the following continuous-time ordinary differential equation (ODE)

$$\frac{d\mathbf{W}(t)}{dt} = \mathbf{R}\mathbf{W}(t)[\mathbf{W}^T(t)\mathbf{R}\mathbf{W}(t)]^{-1} - \mathbf{W}(t) \qquad (11)$$

where $t = \eta k$. We will establish the global convergence properties of this ODE by the Lyapunov function approach [4].

We note $L'(\mathbf{W}) = -J_{NIC}(\mathbf{W})$ is a Lyapunov function for (11) with a stable equilibrium $\mathbf{W} = \mathbf{U}_1\mathbf{P}_r\mathbf{Q}$ in the region $D = \{\mathbf{W}|L'(\mathbf{W}) < \infty\} = \{\mathbf{W}|\mathbf{W}^T\mathbf{R}\mathbf{W} > \mathbf{0}\}$. To further establish a strong convergence property at $\mathbf{W} = \mathbf{U}_1\mathbf{P}_r\mathbf{Q}$, we construct the following function

$$L(\mathbf{W}) = \frac{1}{2}\{tr(\mathbf{W}^T\mathbf{W}) - tr[log(\mathbf{W}^T\mathbf{R}_r\mathbf{W})]\} \qquad (12)$$

where $\mathbf{R}_r = \mathbf{U}_1\mathbf{\Lambda}_1\mathbf{U}_1^T$ is the best (least-squares) rank-r approximation of $\mathbf{R}$. The following two lemmas can be shown [8].

Lemma 4.1 Let $\mathbf{W}(t)$ be the solution of the ODE (11) and $\mathbf{W}(0) \in D$. Then for all $t \in [0, \infty)$, we have

$$\|\mathbf{W}^T(t)\mathbf{W}(t) - \mathbf{I}_r\|_F = e^{-2t}\|\mathbf{W}^T(0)\mathbf{W}(0) - \mathbf{I}_r\|_F \qquad (13)$$

Lemma 4.2 Let $\mathbf{W}(t)$ be the solution of the ODE (11) and $\mathbf{W}^T(0)\mathbf{R}_r\mathbf{W}(0) > \mathbf{0}$. Then for all $t \in [0, \infty)$, $\mathbf{W}^T(t)\mathbf{R}_r\mathbf{W}(t) > \mathbf{0}$.

Lemma 4.1 establishes the constant convergence rate at which $\mathbf{W}(t)$ orthonormalizes itself for any initial $\mathbf{W}(0) \in D$. Lemma 4.2 implies that for any proper $\mathbf{W}(0)$, the solution of $\mathbf{W}(t)$ along the trajectory of (11) will never evolve into any of the saddle point of $J_{NIC}(\mathbf{W})$. The above mentioned global convergence is established by the following theorem (see [8] for a proof).

Theorem 4.1 $L(\mathbf{W})$ is a Lyapunov function for the ODE (11), whose domain of attraction is

$$\Omega = \{\mathbf{W}|\mathbf{W}^T\mathbf{R}_r\mathbf{W} > \mathbf{0}\} \qquad (14)$$

i.e., for any $\mathbf{W}(0) \in \Omega$, $\mathbf{W}(t)$ globally converges along the trajectory of (11) to an arbitrary orthonormal basis of the principal subspace.

Note that (14) identifies the largest domain of attraction for the ODE (11) to converge to the principal subspace solution. Because a randomly selected $\mathbf{W}(0)$ satisfies (14) almost surely, we can initialize $\mathbf{W}_k$ by a random matrix for the NIC algorithm.

5. SIMULATIONS

We present two examples to test the performance of the NIC algorithm. The first example is for the two-layer linear NN learning using the data-driven NIC algorithm (6-10). The learning curves of subspace distance are plotted in Figure 2 for the NIC, the LMSER and Oja's algorithm. It is observed that the NIC algorithm outperforms the other two in both the convergence speed and the subspace estimation accuracy.

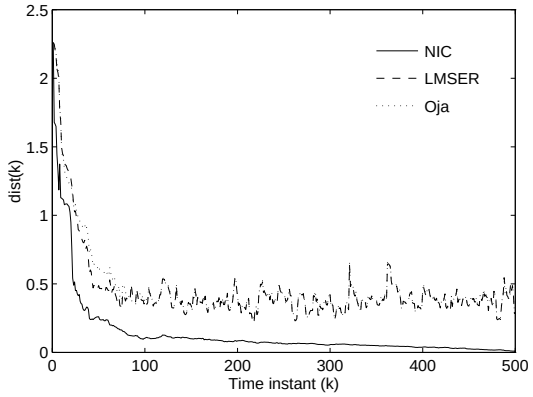

Figure 2: Learning curves for subspace distance of the NIC, the LMSER and Oja's algorithm.

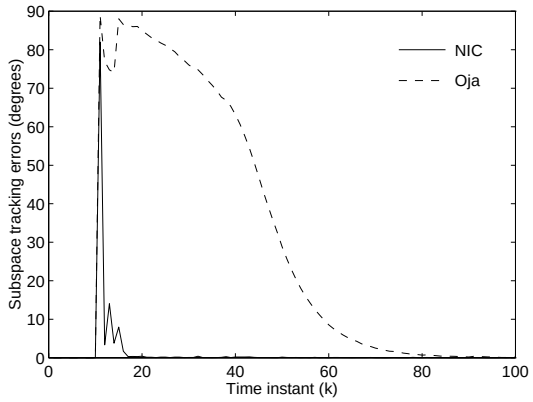

Figure 3: Subspace tracking error of the NIC and Oja's algorithm.

The second example is from [9] which tests the tracking capability of any subspace tracking algorithm by enforcing an abrupt 90° rotation of the principal subspace at time $k = 10$. The subspace tracking error of an algorithm is measured using the largest principal angle between the subspace spanned by columns of $\mathbf{W}_k$ and that obtained by EVD applied directly to (4) at each time step k. Figure 3 shows the subspace tracking error of the batch-mode NIC (3) and that of of Oja's algorithm (5). It is noted that the NIC has a good tracking capability in contrast to the poor tracking capability of Oja's algorithm.

6. CONCLUSIONS

The NIC maximization is a novel non-quadratic formulation of the PSA, and has some significant advantages over the conventional formulation. The NIC algorithm is fast and globally convergent for almost any weight initializations. Both the batch and the RLS implementations of the NIC demonstrate good subspace tracking and convergence capabilities. The NIC algorithm is clearly useful in linear NN learning and real-time signal processing applications where fast adaptive subspace estimation is required. Issues such as the explicit convergence rate of the NIC algorithm, the connections between the iterative equation (3) and the orthogonal iteration technique, and the effect of step size are currently under further investigation.

7. REFERENCES

[1] P. Baldi and K. Hornik, "Learning in linear neural networks: A survey," *IEEE Trans. Neural Networks*, vol. 6, pp. 837-858, July 1995.

[2] E. Oja, "Neural networks, principal components, and subspaces," *Intl. J. Neural Syst.*, vol. 1, pp. 61-68, 1989.

[3] W.-Y. Yan, U. Helmke, and J. B. Moore, "Global analysis of Oja's flow for neural networks," *IEEE Trans. Neural Networks*, vol. 5, pp. 674-683, 1994.

[4] M. Plumbley, "Lyapunov function for convergence of principal component algorithms," *Neural Networks*, vol. 8, pp. 11-23, 1995.

[5] L. Xu, "Least mean square error reconstruction principle for self-organizing neural nets," *Neural Networks*, vol. 6, pp. 627-648, 1993.

[6] B. Yang, "Projection approximation subspace tracking," *IEEE Trans. Signal Processing*, vol. 43, pp. 95-107, Jan. 1995.

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

[8] Y. Miao and Y. Hua, "Fast subspace tracking and neural network learning by a nocel information criterion," *IEEE Trans. Signal Processing*, revised in August 1997.

[9] P. Comon and G. H. Golub, "Tracking a few extreme singular values and vectors in signal processing," *Proc. IEEE*, vol. 78, pp. 1327-1343, Aug. 1990.