

# An Efficient PCA-type Learning Based on Scaled Conjugate Gradient Algorithm for Fast Signal Subspace Decomposition<sup>1</sup>

Bai-ling ZHANG, Irwin K. KING and Lei XU

Department of Computer Science, The Chinese University of Hong Kong, Shatin, N.T., Hong Kong

**Abstract** Nonlinear PCA type learning has been recently suggested for signal subspace decomposition and sinusoidal frequencies tracking, which outperformed the linear PCA based methods and traditional least squares algorithms. Currently, nonlinear PCA algorithms are directly generalized from linear ones that based on gradient descent (GD) technique. The convergence behavior of gradient descent is dependent upon the eigenvalue spread of the data correlation matrix and generally sensitive to the choice of the learning step size. In this paper, we proposed an efficient nonlinear PCA-type learning algorithm by using Scaled Conjugate Gradient (SCG) method proposed by Moller (1993) for signal subspace decomposition. SCG requires much less data samples and especially suitable for the large scale problems. The efficiency is demonstrated by simulations.

## 1 Introduction

Signal subspace decomposition has received considerable attentions as it has quite broad applications in various fields. One important kind of unsupervised learning that can extract the principal component subspace of input data has attracted considerable interests. The principal component subspace is defined as the space spanned by the principal eigenvectors of the data autocorrelation (or covariance) matrix. Generally they are named as PCA type learning because of its close relation with the principal component analysis (PCA).

Linear PCA or PSA (principal subspace analysis) including their linear generalizations has a number of limitations that narrows their further developments. Generally, so-called "non-linear PCA" describe what is learned by a kind of nonlinear generalization of Oja neurons<sup>[1]</sup> and other PCA networks<sup>[2]-[4],[7]-[9]</sup>. Nonlinear PCA type learning may have many favorable properties<sup>[2],[7]</sup>. Currently, nonlinear PCA algorithms are directly generalized from linear ones that based on the gradient descent (GD) technique. Recently, one of the present author<sup>[4]</sup> suggested that for the cost function based PCA type learning like the LMSER<sup>[3]</sup>, several other fast optimization methods, such as Newton, conjugate gradient method, Kalman filter approaches, can be used to speed up learning. In this paper, we will apply a more sophisticated conjugate gradient algorithm called Scaled Conjugate Gradient (SCG)<sup>[11]</sup> which can not only speed up learning, but also eliminate the disadvantages that GD algorithms usually have poor convergence rate and depend on externally specified initial weight values and step size. A number of computer simulations demonstrate that nonlinear PCA type learning based on the Scaled Conjugate Gradient technique has quite remarking advantages.

## 2 Nonlinear PCA Type Learning via SCG Algorithm

Nonlinear generalization of PCA may have many forms, mainly depending on the network structure and what kind of activation function is used. A main motivation of applying nonlinear function  $f(t)$  to output neurons in PCA type networks is that high-order statistics can be introduced into computation. For a single layer feedforward network, one nonlinear PCA learning is proposed as follows:

$$\mathbf{W}_{k+1} = \mathbf{W}_k + \mu_k [\mathbf{x}_k f(\mathbf{x}_k^T \mathbf{W}_k) - \mathbf{W}_k f(\mathbf{W}_k^T \mathbf{x}_k) f(\mathbf{x}_k^T \mathbf{W}_k)] \quad (1)$$

here the  $L \times M$  weight matrix  $\mathbf{W}_k$  has the same meaning as in [6], with function  $f(t)$  applied to each component of the argument vector.  $f(t)$  is usually taken as a monotonic odd function.

As shown in [3], Eq.(1) is actually a simplified version of stochastic gradient descent learning for minimizing the Least Mean Square Error Reconstruction (LMSER) proposed in [3]

$$J(\mathbf{W}) = E(\|\mathbf{x} - \mathbf{W}f(\mathbf{W}^T \mathbf{x})\|^2) \quad (2)$$

which is implemented by the gradient descent algorithm:

$$\mathbf{W}_{k+1} = \mathbf{W}_k - \mu_k \nabla J(\mathbf{W}_k) |_{\mathbf{x}=\mathbf{x}_k} \quad (3)$$

---

<sup>1</sup>This work was supported by the Hong Kong Research Grant Council, NO: 220500720

with

$$\nabla J(\mathbf{W}_k) |_{\mathbf{x}=\mathbf{x}_k} = -\frac{\partial J(\mathbf{W})}{\partial \mathbf{W}} |_{\mathbf{x}=\mathbf{x}_k} = -(\mathbf{x}_k \mathbf{e}_k^T \mathbf{W}_k S' + \mathbf{e}_k S)$$

where  $S = f(\mathbf{x}_k^T \mathbf{W}_k)$ ,  $S' = f'(\mathbf{x}_k^T \mathbf{W}_k)$ ,  $\mathbf{e}_k = \mathbf{x} - \mathbf{W}_k f(\mathbf{W}_k^T \mathbf{x})$  is the reconstruction error vector. Generally, nonlinear PCA algorithm Eq.(1) or (4) deviates from PCA to the independent component analysis (INCA)<sup>[10]</sup>, though the later is an underdetermined problem and Eq.(1) is far from achieving at total success for solving it. The experiments in [6] and ours in the following section indicate that learning with Eq.(1) or Eq.(4) do evolve toward INCA, if an appropriate nonlinear function  $f$  is used. This means that nonlinear algorithms like Eq.(1) and Eq.(4) can provide promising candidates for separating component signals from their mixture, which is not possible with linear PCA or PSA estimation algorithm.

Recently, one of the present author<sup>[4]</sup> suggested that for the cost function based PCA type learning like the LMSE<sup>[3]</sup>, several other fast optimization methods can be used to speed up learning. The conjugate gradient algorithms take advantages of the second order informatoin, but does not require the explicit storage and update of the inverse Hessian matrix. In [11], Moller introduced a new variation of the conjugate gradient method (Scaled Conjugate Gradient, SCG) and it has been shown that SCG is about an order of magnitude faster than the standard BP when tested on the parity problem. Generally, for neural learning problem, SCG algorithm is better, because the common CG algorithms only works for functions with positive definite Hessian matrices, and that the quadratic approximation on which the algorithm works can be very poor when the current point is far from the desired minimum. For the gradient descent based algorithm like Eq.(4), how to appropriately choosing the step size  $\mu_k$  in order for the algorithm to converge and for the weight to be stable is problem dependent. In linear case like Oja's Subspace Learning, the step size must satisfy the condition  $0 \leq \mu_k \leq 2 \|\mathbf{x}_k\|^{-2}$  and the convergence behavior generally has a relationship with the eigenvalue spread of the data autocorrelation matrix. In nonlinear situation, there is even no such theoretical guideline. If  $\mu_k$  is chosen too small, the convergence process (if there is) may be painfully slow. On the other hand, if  $\mu_k$  is too large, instability will occur. The problem has been solved in SCG by introducing a scale parameter to regulate the indefiniteness of Hessian matrix and scale the step size in the combination of conjugate directions. Therefore, SCG is fully-automated, including no critical user-dependent parameters and avoiding a time consuming line search. For more details about SCG, refer ! to [11].

### 3 Simulations

In the following, we will discuss the performance of SCG-based nonlinear PCA type learning and explain by experiment that the SCG is efficient for signal subspace decomposition problems.

The first experiment is a simple one, adopting from [6]. The input vectors  $\mathbf{x}_k$  are generated by  $\mathbf{x}_k = \alpha_1 \mathbf{i}_1 + \alpha_2 \mathbf{i}_2$ , where  $\alpha_1$  and  $\alpha_2$  are independent random numbers distributed uniformly over the interval  $(-0.7, 0.7)$ . The vectors  $\mathbf{i}_1 = [\frac{-1}{\sqrt{5}}, \frac{2}{\sqrt{5}}]^T$  and  $\mathbf{i}_2 = [1, 0]^T$  are the basis vectors of INCA which are different from the PCA basis vectors  $\mathbf{e}_1 = [\frac{-3}{\sqrt{13}}, \frac{-2}{\sqrt{13}}]^T$  and  $\mathbf{e}_2 = [\frac{2}{\sqrt{13}}, \frac{3}{\sqrt{13}}]^T$ .

For comparison purposes, we provide the learning curve of the gradient descent algorithm, which is a plot of the ensemble average of the squared instantaneous error  $e^2(i)$  versus  $i$ , for nonlinear function  $f(t) = \text{sign}(t) \ln(1 + \alpha|t|)$ ,  $\alpha = 20$ . The ensemble averaging was carried out over 100 independent trials of the experiment. In each experiment, 200 randomly generated data was used. For different learning rate parameter  $\mu$ , Fig.1 compared the convergence performance of learning algorithm Eq.(4), which confirmed that the learning parameter is a very important factor in gradient descent based algorithm. Generally, parameter  $\mu$  should be slowly decreasing to zero as the learning progresses. This is not a mild condition in many situations, because it confines the GD-based algorithms to time-invariant process. As  $\mu$  approaches zero, the network learning capability will be gradually diminishing.

Fig.2 demonstrate the learning curves for the Scaled Conjugate Gradient learning for  $\alpha = 10$ . As before, the ensemble averaging for the learning curve was carried out over 100 independent trials of the experiment. It is clear that the conjugate gradient algorithm offers a considerable improvement over the convergence compared to the gradient descent algorithm.

Both the gradient descent and the Scaled Conjugate Gradient method has a batch version (off-line) and adaptive version (on-line). The former one updates the weight after a batch of data  $\mathbf{x}_1, \dots, \mathbf{x}_m$  has been collected and  $\hat{J} = \frac{1}{m} \sum_{i=1}^m \|\mathbf{x}_i - \mathbf{W}f(\mathbf{W}^T \mathbf{x}_i)\|^2$  is used as an estimation of  $J$  in Eq.(3). The adaptive version is also called stochastic approximation in which data sample  $\mathbf{x}$  comes

randomly and an instantaneous reconstruction error replace the criterion  $J$  in Eq.(3) for updating weight as the above simulations practice. In Fig.3(a) we compare the performance of SCG and GD in batch version, in which a cycle represents a time for a same data set (500 samples) being processed by SCG and GD algorithm respectively, while in Fig.3(b) a cycle represents a time for a different data set (contain 500 samples) being used in the simulation. The results demonstrate that both performance of SCG and GD is more stable in batch version than in adaptive version and the SCG is much fast in its convergence.

In second experiment, we study the advantages of SCG based nonlinear PCA type learning for estimation of sinusoids from their noisy mixture, which has been used as testing experiment in [6] for gradient descent nonlinear PCA learning. Conjugate gradient algorithm can yields a set of  $\mathbf{w}_1(k), \dots, \mathbf{w}_M(k)$  of estimated basis vectors that can be used in various signal subspace type methods for extracting information on sinusoidal frequencies. We considered the well-known MUSIC estimator, which is usually defined in terms of the principal (or nonprincipal) eigenvectors. In the general case of complex data, MUSIC frequency estimator is defined in the form

$$P(f) = \frac{1}{L - \sum_{i=1}^M |\mathbf{e}_f^H \mathbf{w}(i)|^2} \quad (4)$$

here  $L$  is the length of data,  $H$  denote the conjugate transpose,  $\mathbf{e}_f$  represents the signal subspace spanned by pure sinusoidal vectors:  $\mathbf{e}_f = [1, e^{j2\pi f}, \dots, e^{j2\pi f(L+1)}]$  at correct normalized frequencies  $f \in [-0.5, 0.5]$ .  $M$  principal eigenvectors of signal autocorrelation matrix span the same signal subspace as the signal vectors  $\mathbf{e}_1, \dots, \mathbf{e}_M$ . In real-valued sinusoids situation,  $M = 2J$ ,  $J$  being the number of sinusoids in the data. The estimated sinusoidal frequencies are obtained as peak locations of Eq.5.

Our experiment data consists of  $N$  samples  $x[1], \dots, x[N]$  of  $J$  real sinusoids in additive noise  $x[k] = \sum_{m=1}^J A_m \cos(2\pi f_m k + \theta_m) + \epsilon[k]$ . The amplitude  $A_m$ , frequencies  $f_m$  and phases  $\theta_m$  of the sinusoids are constants unknown to the learning algorithm. The data vectors  $\mathbf{x}_k = (x[k], x[k+1], \dots, x[k+L-1])^T$  are collected from  $L$  successive samples. The nonlinear function is sigmoid as  $f(t) = \tanh(3t)$ .

In the case of white Gaussian noise, SCG algorithm can yield quite good MUSIC estimator. The training data consisted of three sinusoids at normalized frequencies  $f_1 = 0.11$ ,  $f_2 = 0.20$  and  $f_3 = 0.30$  in white noise (SNR=5). The amplitudes of the sinusoids were  $A_1 = 0.8$ ,  $A_2 = 1.2$  and  $A_3 = 0.8$ . The data vectors had  $L = 15$  components. In Fig.4, we compared the difference in spectrum estimation performance caused by data sample reduction for SCG and GD algorithm, respectively. The experiment was proceeded with two closely spaced sinusoidal frequencies. In Fig.4(a),(c), 85 data vectors were taken into computation for SCG and GD based MUSIC estimaiors, respectively, while in Fig.4(b),(d), only 40 data vectors were collected, again for SCG and GD. We can find that the reduced data still yields a good frequencies estimation from SCG. On the other hand, the performance of GD based method becomes poor when less data is available.

#### 4 Conclusions

In this paper, we have studied the Scaled Conjugate Gradient algorithm for nonlinear PCA type learning and signal subspace decomposition. With the guideline of the LMSER principle, an appropriately chosen nonlinearity makes the PCA type learning more robust again noise and capable to separate component signals from their mixture. As the pure gradient descent methods are usually very inefficient, the SCG algorithm has been proven to be an promising alternative. Without critical user dependent parameters and line search or inverting Hessian matrix, the SCG based learning is easy to be implemented in practice though a slightly more computational cost has to be paid.

#### References

- [1]. E. Oja, *Int. Journ. of Neural Systems*,, Vol.1, pp. 61-68, 1989. [2]. L. Xu, *Proc. of 1994 IEEE ICNN*, Florida, Vol.II, pp.1252-1257, 1994. [3]. L. Xu, *Neural Networks*, Vol.6, pp. 627-648, 1993. [4]. L. Xu, "Advances on Three Streams of PCA Studies," to appear on Proc. of ICANN'95, Paris, Oct. 9-13, 1995. [5]. G. Xu, etc., *IEEE Trans. Signal Processing*, Vol.42, no.6, pp. 1453-1461, 1994. [6]. J. Karhunen etc., *Neural Networks*, Vol.7, no.1, pp. 113-127, 1994. [7]. L. Xu, *Proc. of 1994 ICONIP. Seoul*, pp. 943-949, 1994. [8]. T. D.Sanger, *Neural Networks*, Vol.2, pp. 459-473, 1989. [9]. P.Foldiak, *Proc. of 1989 IEEE ICNN, Washington D.C.*, Vol.I, pp.401-405, June 1989. [10]. P. Comon, *Higher Order Statistics, J.L.Lacoume (Editor)*, Elsevier Science Pub., pp.29-38, 1992. [11]. M. F. Moller, *Neural Networks*, Vol.6, pp. 525-533, 1993.