On-Line Singular Value Decomposition of Stochastic Process Covariances

Tomas Landelius tc@isy.liu.se
Hans Knutsson knutte@isy.liu.se
Magnus Borga magnus@isy.liu.se

Computer Vision Laboratory
Department of Electrical Engineering
Linköping University, S-581 83 Linköping, Sweden

Abstract

This paper presents novel algorithms for finding the singular value decomposition (SVD) of a general covariance matrix by stochastic approximation. General in the sense that also non-square, between sets, covariance matrices are dealt with. For one of the algorithms, convergence is shown using results from stochastic approximation theory. Proofs of this sort, establishing both the point of equilibrium and its domain of attraction, have been reported very rarely for stochastic, iterative feature extraction algorithms.

1 Introduction

The ability to perform dimensionality reduction is crucial to systems exposed to high dimensional data. One way of approaching this problem is to project the data on the direction of maximal data variation, the largest principal component. There are also a number of applications in signal processing where the largest eigenvalue and the corresponding eigenvalue of input data correlation or covariance matrices play an important role. One such example is found in our own research where such techniques are used for adaption of local response models [3]. When relations between two sets of data, e.g. process input and output, are to be investigated it becomes interesting to find the directions accompanying the largest singular value of the between sets covariance matrix.

Given two sets of random vectors with zero mean, $\{\mathbf{x}_k\}$ and $\{\mathbf{y}_k\}$, the between sets covariance matrix is defined as

$$\mathbf{C}_{xy} = E_{xy}\{\mathbf{x}\mathbf{y}^T\} = \sum \lambda_i \; \hat{\mathbf{e}}_{xi}\hat{\mathbf{e}}_{yi}^T \tag{1}$$

where $\lambda, \hat{\mathbf{e}}_x$ and $\hat{\mathbf{e}}_y$ are positive singular values and vectors respectively. The expectation operator, $E_{xy}\{\cdot\}$, works over both of the sequences of vectors. The hat indicates that the vector is of unit length. It is always possible to express a covariance matrix as a unique sum of orthogonal outer products with positive coefficients λ_i.

In the following two sections both the special case when the two sets are identical, resulting in a within set covariance matrix, and the more general case where the two sets differ, will be considered.

2 Within set covariance

In this case finding the largest singular value will be the same as finding the largest eigenvalue of the usual covariance matrix. The largest eigenvalue and its corresponding eigenvector to a matrix $\mathbf{C}_{xx}$ can, according to eq. 1, be found by maximizing the *Rayleigh quotient*, ρ:

$$\lambda_1 = \hat{\mathbf{e}}_{x1}^T \mathbf{C}_{xx} \hat{\mathbf{e}}_{x1} = \max \frac{\mathbf{w}_x^T \mathbf{C}_{xx} \mathbf{w}_x}{\mathbf{w}_x^T \mathbf{w}_x} = \max \rho. \tag{2}$$

Taking the derivatives of the expression for ρ with respect to the vector $\mathbf{w}_x$ gives the condition

$$\frac{d\rho}{d\mathbf{w}} = \frac{2}{\|\mathbf{w}_x\|}(\mathbf{C}_{xx}\hat{\mathbf{w}}_x - \rho\hat{\mathbf{w}}_x) \tag{3}$$

$$\frac{d\rho}{d\mathbf{w}} = 0 \Rightarrow \mathbf{C}_{xx}\hat{\mathbf{w}}_x = \rho\hat{\mathbf{w}}_x, \tag{4}$$

which naturally yields the solution as the eigenproblem of the matrix $\mathbf{C}_{xx}$. Note that the solution corresponding to the largest singular value is given by $\rho = \lambda_1$ and $\hat{\mathbf{w}}_x = \pm\hat{\mathbf{e}}_{x1}$. To get an iterative algorithm that on the average performs a gradient search on the energy function ρ, it should be the case that, at each iteration, the estimated vector is updated by some amount in the direction of the gradient according to eq. 3.

$$E\{\Delta\mathbf{w}_x\} \propto \frac{d\rho}{d\mathbf{w}_x} \propto \mathbf{C}_{xx}\hat{\mathbf{w}}_x - \rho\hat{\mathbf{w}}_x = E_x\{\mathbf{x}\mathbf{x}^T\hat{\mathbf{w}}_x - \rho\hat{\mathbf{w}}_x\}. \tag{5}$$

Now let the length of the vector represent the estimated singular value, i.e. $\|\mathbf{w}_x\| = \rho$. This leads to a novel unsupervised Hebbian learning algorithm that finds both the direction of maximal data variation and how much the data varies along that direction. The update rule for this algorithm is given by

$$\Delta\mathbf{w}_{xk} = \gamma_k \, (\mathbf{x}_k\mathbf{x}_k^T\hat{\mathbf{w}}_{xk} - \mathbf{w}_{xk}), \tag{6}$$

where γ is the gain controlling how far, in the direction of the gradient, the vector estimate is updated at each iteration.

To verify that a learning rule results in the vector $\mathbf{w}$ arriving at the desired point of equilibrium, no matter where it starts out, is often very hard. This is equal to determining the domain of attraction of the point of equilibrium, in this case the largest singular value times its corresponding vector. However, in this case a proof of convergence based on methods from stochastic approximation theory has been established and is outlined in section 5.

When the largest singular value and its corresponding vector is found it is a simple matter to proceed and find the second largest pair, and so on. In order to do this a new signal vector, orthogonal to the ones already estimated, is constructed and used as a new input to the algorithm.

3 Between sets covariance

When the size of the two spaces differ, the largest singular value and its corresponding vectors of a non-square between sets covariance matrix $\mathbf{C}_{xy}$ is to be found. Again, by looking at eq. 1, an energy function to maximize can be found:

$$\lambda_1 = \hat{\mathbf{e}}_{x1}^T \mathbf{C}_{xy} \hat{\mathbf{e}}_{y1} = \max \frac{\mathbf{w}_x^T \mathbf{C}_{xy} \mathbf{w}_y}{\sqrt{\mathbf{w}_x^T \mathbf{w}_x \mathbf{w}_y^T \mathbf{w}_y}} = \max \rho. \tag{7}$$

Taking the derivatives of the energy function ρ with respect to the vectors $\mathbf{w}_x$ and $\mathbf{w}_y$ gives

$$\frac{d\rho}{d\mathbf{w}_x} = \frac{1}{\|\mathbf{w}_x\|}(\mathbf{C}_{xy}\hat{\mathbf{w}}_y - \rho\hat{\mathbf{w}}_x) \tag{8}$$

$$\frac{d\rho}{d\mathbf{w}_y} = \frac{1}{\|\mathbf{w}_y\|}(\mathbf{C}_{yx}\hat{\mathbf{w}}_x - \rho\hat{\mathbf{w}}_y). \tag{9}$$

Setting these expressions to zero and solving for $\mathbf{w}_x$ and $\mathbf{w}_y$ results in

$$\mathbf{C}_{xy}\mathbf{C}_{yx}\hat{\mathbf{w}}_x = \rho^2\hat{\mathbf{w}}_x \tag{10}$$

$$\mathbf{C}_{yx}\mathbf{C}_{xy}\hat{\mathbf{w}}_y = \rho^2\hat{\mathbf{w}}_y. \tag{11}$$

Hence, the problem of finding the largest singular value and the corresponding vectors can be separated into two eigenproblems, one for each vector. In this case the length of the vectors would be $\rho^2 = \lambda_1^2$. But since the singular values are positive no sign is lost in this process. This means that the algorithm for the previous case could be used also for this problem if the two new covariance matrices

$$\mathbf{C}_x = \mathbf{C}_{xy}\mathbf{C}_{yx} = E_{xy}\{\mathbf{x}\mathbf{y}^T\}E_{xy}\{\mathbf{y}\mathbf{x}^T\} \tag{12}$$

$$\mathbf{C}_y = \mathbf{C}_{yx}\mathbf{C}_{xy} = E_{xy}\{\mathbf{y}\mathbf{x}^T\}E_{xy}\{\mathbf{x}\mathbf{y}^T\}. \tag{13}$$

can be expressed as expectation sums. It is however not obvious how this should be accomplished since

$$E_{xy}\{\mathbf{x}\mathbf{y}^T\}E_{xy}\{\mathbf{y}\mathbf{x}^T\} \neq E_{xy}\{\mathbf{x}\mathbf{y}^T\mathbf{y}\mathbf{x}^T\} \tag{14}$$

$$E_{xy}\{\mathbf{y}\mathbf{x}^T\}E_{xy}\{\mathbf{x}\mathbf{y}^T\} \neq E_{xy}\{\mathbf{y}\mathbf{x}^T\mathbf{x}\mathbf{y}^T\}. \tag{15}$$

However, another learning rule could be deduced from eq. 8 and eq. 9. By the same line of reasoning that gave the update vector for the previous algorithm it should be clear that since

$$E\{\Delta\mathbf{w}_x\} \propto E\{\frac{d\rho}{d\mathbf{w}_x}\} \propto \mathbf{C}_{xy}\hat{\mathbf{w}}_y - \rho\hat{\mathbf{w}}_x \propto E_{xy}\{\mathbf{x}\mathbf{y}^T\hat{\mathbf{w}}_y - \rho\hat{\mathbf{w}}_x\} \tag{16}$$

$$E\{\Delta\mathbf{w}_y\} \propto E\{\frac{d\rho}{d\mathbf{w}_y}\} \propto \mathbf{C}_{yx}\hat{\mathbf{w}}_x - \rho\hat{\mathbf{w}}_y \propto E_{xy}\{\mathbf{y}\mathbf{x}^T\hat{\mathbf{w}}_x - \rho\hat{\mathbf{w}}_y\}, \tag{17}$$

an update rule that on the average is moving in the direction of the gradient could be constructed according to

$$\Delta\mathbf{w}_{xk} = \gamma_k\ (\mathbf{x}_k\mathbf{y}_k^T\hat{\mathbf{w}}_{yk} - \mathbf{w}_{xk}) \tag{18}$$

$$\Delta\mathbf{w}_{yk} = \gamma_k\ (\mathbf{y}_k\mathbf{x}_k^T\hat{\mathbf{w}}_{xk} - \mathbf{w}_{yk}). \tag{19}$$

This time the length of $\mathbf{w}_x$ and $\mathbf{w}_y$ should again equal ρ, the largest singular value. Note that this is a coupled learning rule in difference with the hypothetical case where the first algorithm is applied on both vectors individually. Even if it had beed possible to estimate $\mathbf{C}_x$ and $\mathbf{C}_y$ on-line, the coupled version still needs fewer scalar products to be calculated. However, for the coupled algorithm a proof of convergence, available for the previous version has not been established. It is anyway true that it on average does a steepest ascent search on the energy function defined in eq. 7. Again, a complete SVD can be found by construction of orthogonal signal vectors which are used as new inputs to the algorithm.

4 Experiments

In order to make a statement about the performance of the proposed algorithm in the more general between-sets case, the "optimal", in the sense of maximum likelihood, deterministic solution was calculated by performing a SVD decomposition on the covariance matrix of the data accumulated so far.

For use in the comparison, two random vector sequences were produced. In the shown example the dimensionality of an instance in the $\{\mathbf{x}_k\}$ and $\{\mathbf{y}_k\}$ sequences was 10 and 5 respectively. The largest singular value was equal to 10. The singular values of the covariance matrix describing the vector distribution declined as $\exp(-0.5i)$, where i is the index to the i:th largest singular value.

Magnitude and angular errors were calculated as $|1 - \|\mathbf{w}\|/\|\mathbf{w}_c\||$ and $\arccos(\hat{\mathbf{w}}^T\hat{\mathbf{w}}_c)$, where $\mathbf{w}_c$ is the correct singular vector. The error estimates were averaged over 50 runs, each consisting of 5000 instances from the vector distributions. These averaged measurements are shown on the top and bottom row of figure 1. The right and left column of the figure corresponds to errors in $\{\mathbf{w}_x\}$ and $\{\mathbf{w}_y\}$ respectively. For the proposed algorithm the gain sequence $\gamma_k = (1+\beta)/(k+\beta)$, with $\beta = 1.25$, was used.

No time was spent on finding and trimming an intelligent gain sequence, instead this comparison is only intended to show that the algorithm is robust and works. Even though the difference in computational is $\mathcal{O}(d^3)$ to $\mathcal{O}(d)$ in favour to the proposed algorithm, its behavior is still quite similar to that of optimal brute force SVD.

5 Proof outline

Verifying that the vector enters, infinitely often, the domain of attraction for a given asymptotically stable equilibria is usually extremely hard. However, this is possible for Oja's one-unit algorithm, leading to a global analysis of the asymptotic solutions [4]. According to the survey on convergence analysis of local feature extraction algorithms by Hornik and Kuan [1], Oja's one-unit algorithm is the only feature extraction algorithm of their knowledge for which a global analysis has been presented. This no longer holds true since recently an analysis of the domain of attraction for a class of principal component algorithms was presented by Plumbley [6]. This class does however not include algorithms where

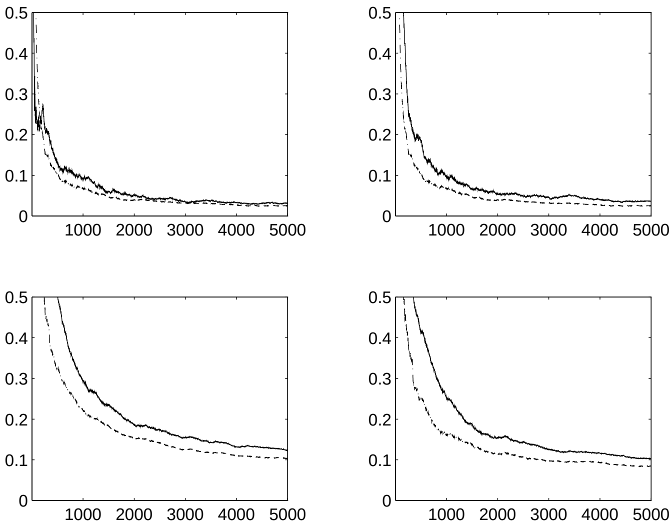

Figure 1: Averaged errors in $\mathbf{w}_x$ (left) and $\mathbf{w}_y$ (right). Top row shows magnitude errors and bottom row angular errors. Estimates are shown with a solid line for the proposed algorithm and with a dashed line for the optimal approach.

normalizations occur, which is the case for the algorithm proposed in this paper.

The proof outlined below is similar to that presented by Oja and Karhunen for their one-unit algorithm [5]. It relies on the results by Kushner and Clark [2] concerning the almost sure convergence of stochastic approximation algorithms where the update rule for $\{\mathbf{w}_k\}$ is given by

$$\Delta \mathbf{w}_k = \gamma_k \; h(\mathbf{w}_k). \tag{20}$$

Under assumptions A.1 to A.5, presented below, $\mathbf{w}$ is bounded in the limit and satisfies the *ordinary differential equation* (ODE)

$$\frac{d\mathbf{w}}{dt} = h(\mathbf{w}), \tag{21}$$

which can be seen as the continuous counterpart of eq. 20. Let $\mathbf{w}_a$ be a locally asymptotically stable solution to eq. 21, with domain of attraction $\mathcal{D}(\mathbf{w}_a)$. If there is a compact subset $\mathcal{A} \subset \mathcal{D}(\mathbf{w}_a)$ such that $\mathbf{w}_k \in A$ infinitely often, then $\mathbf{w}_k \to \mathbf{w}_a$, when $k \to \infty$.

The following five assumptions must be valid in order for the above theorem to hold.

A.1 Each $\mathbf{x}$ is almost surely bounded, mutually independent and with $E_x\{\mathbf{x}\mathbf{x}^T\} = \mathbf{C}$.

A.2 The largest eigenvalue of $\mathbf{C}$ has unit multiplicity.

A.3 Each $\mathbf{x}_k\mathbf{x}_k^T$ has a probability density that is bounded away from zero uniformly in k in some neighbourhood of $\mathbf{C}$ in $\mathbb{R}^{d\times d}$.

A.4 The $\mathbb{R}^d$ function $h(\cdot)$ is continuous on $\mathbb{R}^d$.

A.5 The sequence γ_k of positive real numbers is bounded s.t. $\gamma_k \to 0$, $\sum_k \gamma_k = \infty$.

This theorem will be used to prove the convergence of the algorithm in eq. 6. The ODE for this algorithm is given by

$$\frac{d\mathbf{w}}{dt} = h(\mathbf{w}) = \lim_{t\to\infty} E_{\mathbf{x}}\{h(\mathbf{w},\mathbf{x})\} = \mathbf{C}\hat{\mathbf{w}} - \mathbf{w}. \tag{22}$$

It can now be shown that the locally asymptotically stable solution to the ODE in eq. 22 is given by $\pm\lambda_1\hat{\mathbf{e}}_1$ where $\pm\hat{\mathbf{e}}_1$ are the unit eigenvectors corresponding to the largest eigenvalue λ_1 of the covariance matrix $\mathbf{C} = E_x\{\mathbf{x}\mathbf{x}^T\}$. The domain of attraction only exclude points with zero projection on $\hat{\mathbf{e}}_1$, i.e. $\mathcal{D}(\pm\lambda_1\hat{\mathbf{e}}_1) = \{\mathbf{x}_0 \in \mathbb{R}^d \mid \mathbf{x}_0^T\hat{\mathbf{e}}_1 \neq 0\}$.

Expressing $\mathbf{w}$ in the orthonormal set of eigenvectors of $\mathbf{C}$,

$$\mathbf{w} = \sum \alpha_i \hat{\mathbf{e}}_i, \tag{23}$$

gives an ODE for the coefficients α_i in the above expansion

$$d\alpha_i/dt = \lambda_i\alpha_i/\|\mathbf{w}\| - \alpha_i. \tag{24}$$

With $\theta_i = \alpha_i/\alpha_1$, assuming $\alpha_1 \neq 0$ or $\mathbf{x}_0^T\hat{\mathbf{e}}_1 \neq 0$, the ODE becomes

$$d\theta_i/dt = \theta_i(\lambda_i - \lambda_1)/\|\mathbf{w}\|. \tag{25}$$

But $\|\mathbf{w}\|$ may be expressed in terms of α_1 using eq. 24 and inserted in eq. 25 to give a solution for α_i

$$\alpha_i = \alpha_1^{\lambda_i/\lambda_1} \;\exp t(\lambda_i/\lambda_1 - 1). \tag{26}$$

Since $\lambda_i < \lambda_1$ for $i > 1$ this gives

$$\lim_{t\to\infty} \alpha_i = 0 \;,\; i > 1. \tag{27}$$

Using this result together with eq. 24 yields an expression in the limit $t > T$ also for $i = 1$, which concludes the proof:

$$\lim_{t\to\infty} \alpha_1 = \begin{cases} +\lambda_1 & , \alpha_1(T) > 0 \\ -\lambda_1 & , \alpha_1(T) < 0. \end{cases} \tag{28}$$

References

[1] K. Hornik and C.-M. Kuan. Convergence analysis of local feature extraction algorithms. *Neural Networks*, 5:229–240, 1992.

[2] H. J. Kushner and D. S. Clark. *Stochastic Approximation Methods for Constrained and Unconstrained Systems.* Springer, New York, 1978.

[3] T. Landelius. Behavior representation by growing a learning tree, September 1993. Thesis No. 397, ISBN 91–7871–166–5.

[4] E. Oja. A simplified neuron model as a principal component analyzer. *J. Math. Biology*, 15:267–273, 1982.

[5] E. Oja and J. Karhunen. On stochastic approximation of the eigenvectors and eigenvalues of the expectation of a random matrix. *Journal of Mathematical Analysis and Applications*, 106:69–84, 1985.

[6] Mark D. Plumbley. Lyapunov functions for convergence of principal component algorithms. *Neural Networks*, 8(1):11–23, 1995.