

Lois asymptotiques d'estimateurs adaptatifs de sous espaces introduits dans la littérature neuronale

Jean Pierre Delmas⁽¹⁾ et Florence Alberge⁽²⁾

⁽¹⁾ Institut National des Télécommunications,
9 rue Charles Fourier, 91011 Evry Cedex, France
Email: delmas@int-evry.fr

⁽²⁾ Ecole Nationale Supérieure des Télécommunications.
46 rue Barrault, 75634 Paris Cedex, France.
Email: alberge@sig.enst.fr

RÉSUMÉ

De nombreux algorithmes adaptatifs ont été proposés dans la littérature neuronale pour estimer les structures propres de matrice de covariance. Nous montrons que décrits dans un cadre commun, la plupart de ces algorithmes présentent des similarités avec les algorithmes stochastiques de type gradient, rencontrés en traitement de signal. Grâce à un résultat d'approximation gaussienne, nous dérivons les lois asymptotiques de différents estimateurs de structures propres dans la situation de pas fixe. Des expressions analytiques des matrices de covariance des vecteurs propres estimés et/ou des matrices de projection associées estimées sont données. Ces expressions sont comparées à celles obtenues en traitement par bloc. Enfin, après avoir précisé le domaine de validité de cette analyse asymptotique, les déviations par rapport à l'orthogonalité sont comparées.

ABSTRACT

In the neural network literature, many algorithms have been proposed for estimating the eigenstructure of covariance matrices. We first show that many of these algorithms, when presented in a common framework, show great similitudes with the gradient-like stochastic algorithms usually encountered in the signal processing literature. We derive the asymptotic distribution of these different recursive subspace estimators. A closed form expression of the covariances in distribution of eigenvectors and associated projection matrix estimators are given. In particular, closed form expressions of the mean square error of these estimators are given. It is found that these covariance matrices have a structure very similar to those describing batch estimation techniques. The accuracy of our asymptotic analysis is checked by numerical simulations, and it is found to be valid not only for a "small" step size but in a very large domain. Finally, deviation from orthonormality of the different algorithms are compared.

1 Introduction

De nombreux algorithmes adaptatifs d'estimation de sous espaces de matrice de covariance ont vu le jour en traitement de signal afin de traiter des problèmes d'analyse spectrale haute résolution, d'imagerie d'antenne et plus récemment d'estimation de réponse impulsionnelle finie en égalisation aveugle. Parallèlement de nombreuses réalisations neuronales ont été proposées dans la littérature neuronale en analyse en composantes principales appliquée à la compression de données et à la reconnaissance de forme, et en ajustement optimal au sens moindres carrés totaux. Ces algorithmes ont été seulement étudiés sous deux points de vue : analyse de convergence en situation de pas décroissant grâce à l'étude de la stabilité de l'ODE associée et implantation neuronale. Après avoir donné une présentation unifiée de ces algorithmes, le but de cette article est de fournir les lois asymptotiques des estimées des vecteurs propres et/ou des matrices de projection associées obtenues dans les versions pas fixe de ces algorithmes en situation stationnaire.

2 Présentation des algorithmes

Considérons une matrice de covariance $\mathbf{R}_x = E(\mathbf{x}\mathbf{x}^T)$ d'un vecteur aléatoire centré \mathbf{x} gaussien et notons $\lambda_1 \geq \dots \geq \lambda_n$ ses valeurs propres et $\mathbf{v}_1, \dots, \mathbf{v}_n$ ses vecteurs propres associés. Nous sommes intéressés par deux types de problèmes : d'une part, par l'estimation adaptative des r vecteurs propres normalisés associés aux r plus grandes [resp. plus petites] valeurs propres distinctes $(\lambda_1, \dots, \lambda_r)$ [resp. $\lambda_{n-r+1}, \dots, \lambda_n$] de \mathbf{R}_x et d'autre part, par l'estimation d'une base (approximativement) orthonormée quelconque de l'espace dominant de dimension r de \mathbf{R}_x .

La plupart des algorithmes étudiés ont été décrits par Oja [1]. Ils peuvent tous être dérivés à partir de la méthode d'itérations simultanées d'analyse numérique [2], sous forme de l'algorithme d'approximation stochastique suivant :

$$\mathbf{W}'_{t+1} = \mathbf{W}_t + \mathbf{R}_t \mathbf{W}_t \Gamma_t \quad (1)$$

$$\mathbf{W}_{t+1} = \mathbf{W}'_{t+1} \mathbf{S}_{t+1}^{-1} \quad (2)$$

dans laquelle $\mathbf{W}_t = (\mathbf{w}_{t,1}, \dots, \mathbf{w}_{t,r}) \in \mathcal{R}^{n \times r}$ est une matrice dont les colonnes $\mathbf{w}_{t,k} \in \mathcal{R}^n$ sont orthonormées et approximent les r vecteurs propres dominants de \mathbf{R}_x .

Dans (1), la matrice Γ_t est usuellement une matrice diagonale $r \times r$ de gains. Nous supposons que $\Gamma_t = \gamma_t \mathbf{I}_r$ excepté dans un algorithme pour lequel $\Gamma_t = \gamma_t \text{Diag}(1, \alpha_2, \dots, \alpha_r)$ avec $\alpha_i > 0$ est utilisé pour assurer un meilleur compromis erreur résiduelle/vitesse de convergence. La matrice \mathbf{R}_t de (1) est une estimée de la matrice de covariance \mathbf{R}_x . Nous n'utiliserons dans ce papier, l'estimée instantanée $\mathbf{x}_t \mathbf{x}_t^T$. Dans (2), \mathbf{S}_{t+1} est une matrice dépendant de \mathbf{W}'_{t+1} , qui diagonalise les colonnes de \mathbf{W}'_{t+1} . Donc, \mathbf{W}_t a des colonnes orthonormées pour tout t . Selon la forme de la matrice \mathbf{S}_{t+1} , différents algorithmes d'approximations stochastiques peuvent être dérivés [1],[3].

Nous distinguerons parmi ces algorithmes, un algorithme convergeant vers une base orthonormée quelconque d'un sous espace propre dominant que l'on dénotera, l'algorithme *Subspace Network Learning* (SNL) :

$$\mathbf{w}_{t+1,k} = \mathbf{w}_{t,k} + \gamma_t [\mathbf{I}_n - \sum_{i=1}^r \mathbf{w}_{t,i} \mathbf{w}_{t,i}^T] \mathbf{x}_t \mathbf{x}_t^T \mathbf{w}_{t,k}, \quad (3)$$

pour $k = 1, \dots, r$.

Puis nous considérerons trois algorithmes convergeant vers les vecteurs propres d'un espace dominant respectivement appelés algorithmes *Stochastic Gradient Ascent* (SGA), *Generalized Hebbian Algorithm* (GHA) et *Weighted Subspace Algorithm* (WSA) :

$$\begin{aligned} \mathbf{w}_{t+1,k} &= \mathbf{w}_{t,k} + \alpha_k \gamma_t [\mathbf{I}_n - \mathbf{w}_{t,k} \mathbf{w}_{t,k}^T \\ &\quad - \sum_{i=1}^{k-1} (1 + \frac{\alpha_i}{\alpha_k}) \mathbf{w}_{t,i} \mathbf{w}_{t,i}^T] \mathbf{x}_t \mathbf{x}_t^T \mathbf{w}_{t,k} \end{aligned} \quad (4)$$

$$\mathbf{w}_{t+1,k} = \mathbf{w}_{t,k} + \gamma_t [\mathbf{I}_n - \sum_{i=1}^k \mathbf{w}_{t,i} \mathbf{w}_{t,i}^T] \mathbf{x}_t \mathbf{x}_t^T \mathbf{w}_{t,k} \quad (5)$$

$$\mathbf{w}_{t+1,k} = \mathbf{w}_{t,k} + \gamma_t [\mathbf{I}_n - \sum_{i=1}^r \frac{\beta_k}{\beta_i} \mathbf{w}_{t,i} \mathbf{w}_{t,i}^T] \mathbf{x}_t \mathbf{x}_t^T \mathbf{w}_{t,k} \quad (6)$$

pour $k = 1, \dots, r$ et pour ce dernier avec $0 < \beta_1 < \dots < \beta_r$.

Et enfin, nous examinerons un algorithme convergeant vers les r vecteurs propres d'un espace minorant, appelé algorithme *Optimal Fitting Analyzer* (OFA) :

$$\begin{aligned} \mathbf{w}_{t+1,k} &= \mathbf{w}_{t,k} + \gamma_t [\mathbf{I}_n - \mathbf{x}_t \mathbf{x}_t^T + \mathbf{w}_{t,k} \mathbf{w}_{t,k}^T \mathbf{x}_t \mathbf{x}_t^T \\ &\quad - \mathbf{w}_{t,k} \mathbf{w}_{t,k}^T - \beta \sum_{i=k+1}^n \mathbf{w}_{t,i} \mathbf{w}_{t,i}^T \mathbf{x}_t \mathbf{x}_t^T] \mathbf{w}_{t,k} \end{aligned} \quad (7)$$

pour $k = n - r + 1, \dots, n$.

La convergence de ces 5 algorithmes a été analysée grâce à l'examen des points stationnaires stables de leur ODE associée. Seul le cas $r = 1$, où les algorithmes SNL, SGA, GHA et WSA se confondent alors avec l'algorithme du neurone d'Oja [4], a permis de démontrer la convergence presque sûre de $\mathbf{w}_{t,1}$ vers $\pm \mathbf{v}_1$ [5].

3 Analyse de performance asymptotique

3.1 Rappel d'un résultat général d'approximation gaussienne

Nous utiliserons le résultat suivant classique de Benveniste [6] dont nous rappelons les grandes lignes : considérons

l'algorithme stochastique à pas fixe :

$$\Theta_{t+1} = \Theta_t + \gamma f(\Theta_t, \mathbf{x}_t) + \gamma^2 h_t(\Theta_t, \mathbf{x}_t) \quad (8)$$

dans lequel le paramètre Θ_t converge presque sûrement vers Θ_* pour l'algorithme à pas décroissant associé. Supposons que l'équation de Lyapunov continue :

$$\mathbf{D} \mathbf{C}_\Theta + \mathbf{C}_\Theta \mathbf{D}^T + \mathbf{G} = \mathbf{O} \quad (9)$$

dans laquelle \mathbf{D} et \mathbf{G} sont respectivement la dérivée du champ moyen et la covariance du champ de l'algorithme (8) :

$$\mathbf{D} \stackrel{\text{def}}{=} \mathbb{E} \left[\frac{\partial f}{\partial \Theta}(\Theta, \mathbf{x}_t) \right]_{\Theta=\Theta_*} \quad (10)$$

et

$$\mathbf{G} \stackrel{\text{def}}{=} \sum_{t=-\infty}^{\infty} \text{Cov}[f(\Theta_*, \mathbf{x}_t), f(\Theta_*, \mathbf{x}_0)] \quad (11)$$

vérifie que toutes les valeurs propres de \mathbf{D} soient à parties réelles strictement négatives. Alors en situation stationnaire, quand $\gamma \rightarrow 0$ et $t \rightarrow \infty$, nous avons :

$$\frac{1}{\sqrt{\gamma}} (\Theta_t - \Theta_*) \xrightarrow{\mathcal{L}} \mathcal{N}(0, \mathbf{C}_\Theta) \quad (12)$$

où \mathbf{C}_Θ est l'unique solution symétrique de l'équation de Lyapunov (9).

3.2 Distribution asymptotique d'estimateurs de vecteurs propres

Bien que nous ne sommes pas dans les strictes hypothèses de Benveniste concernant l'unicité de Θ_* et la convergence presque sûre des algorithmes à pas décroissants associés, nous appliquerons le résultat général que nous venons de rappeler. Après avoir donné les expressions analytiques des matrices \mathbf{D} et \mathbf{G} des algorithmes SGA, GHA, WSA et OFA, nous obtenons les théorèmes suivants [3] :

Théorème 3.1 — *Les valeurs propres de la matrice dérivée \mathbf{D} du champ moyen ont des valeurs propres strictement négatives pour les algorithmes SGA, GHA et OFA et à parties réelles strictement négatives pour l'algorithme WSA.*

Théorème 3.2 — *Les matrices de covariance asymptotique \mathbf{C}_W des algorithmes SGA, GHA, WSA et OFA qui apparaissent dans (12), solutions de (9), se mettent sous la forme commune :*

$$\begin{aligned} \mathbf{C}_W &= \sum_{\substack{1 \leq i \leq r \\ 1 \leq k \neq i \leq n}} b_{k,i} (\mathbf{e}_i^r \mathbf{e}_i^{rT} \otimes \mathbf{v}_k \mathbf{v}_k^T) \\ &\quad + \sum_{1 \leq i \neq j \leq r} c_{i,j} (\mathbf{e}_i^r \mathbf{e}_j^{rT} \otimes \mathbf{v}_j \mathbf{v}_j^T) \end{aligned} \quad (13)$$

dans laquelle $(\mathbf{e}_i^r)_{i=1, \dots, r}$ désigne la base canonique de \mathbb{R}^r . Pour l'algorithme OFA les sommations sont à prendre sur respectivement $n - r + 1 \leq i \leq n, 1 \leq k \neq i \leq n$ et $n - r + 1 \leq i \neq j \leq n$. Les termes $b_{i,j}$ et $c_{i,j}$ sont donnés respectivement par :

$$b_{k,i}^{SGA} = \frac{\alpha_k \lambda_i \lambda_k}{2(\lambda_k - \lambda_i)} 1_{k < i} + \frac{\alpha_i \lambda_i \lambda_k}{2(\lambda_i - \lambda_k)} 1_{k > i} \quad (14)$$

$$c_{i,j}^{SGA} = -\frac{\alpha_j \lambda_j \lambda_i}{2(\lambda_i - \lambda_j)} 1_{i < j} - \frac{\alpha_j \lambda_j \lambda_j}{2(\lambda_j - \lambda_i)} 1_{i > j} \quad (15)$$

$$b_{k,i}^{GHA} = \frac{\lambda_i^2}{2(\lambda_k - \lambda_i)} 1_{k < i} + \frac{\lambda_i \lambda_k}{2(\lambda_i - \lambda_k)} 1_{k > i} \quad (16)$$

$$c_{i,j}^{GHA} = -\frac{\lambda_j^2}{2(\lambda_i - \lambda_j)} 1_{i < j} - \frac{\lambda_i^2}{2(\lambda_j - \lambda_i)} 1_{i > j} \quad (17)$$

$$b_{k,i}^{WSA} = \lambda_i \lambda_k c^{ki} 1_{k < i} + \lambda_i \lambda_k b^{ik} 1_{i < k \leq r} + \frac{\lambda_i \lambda_k}{2(\lambda_i - \lambda_k)} 1_{k > r} \quad (18)$$

$$c_{i,j}^{WSA} = \lambda_i \lambda_j d^{ij} 1_{i < j} + \lambda_i \lambda_j d^{ji} 1_{i > j} \quad (19)$$

$$b_{k,i}^{OFA} = \frac{\lambda_i \lambda_k}{2(\lambda_k - \lambda_i)} 1_{k < i} + g^{ik} 1_{k > i} \quad (20)$$

$$c_{i,j}^{OFA} = h^{ij} 1_{i < j} + h^{ji} 1_{i > j}. \quad (21)$$

où b^{ik} , c^{ki} and d^{ij} de (18) et (19), puis g^{ik} and h^{ij} de (20) et (21) sont respectivement donnés par :

$$b^{ij} = \frac{2\lambda_i a_{i,j} d^{ij} - (1 - a_{i,j})^2}{2(\lambda_j(1 - a_{i,j}) - \lambda_i)}, \quad c^{ij} = \frac{2\lambda_j a_{i,j}^{-1} d^{ij} - (1 - a_{i,j}^{-1})^2}{2(\lambda_i(1 - a_{i,j}^{-1}) - \lambda_j)},$$

$$d^{ij} = -\frac{(1 - a_{i,j})^2 a_{i,j}^{-1}}{2(\lambda_j a_{i,j} + \lambda_i a_{i,j}^{-1})} \\ \frac{(\lambda_i^2(2a_{i,j}^{-1} - 1) + \lambda_j^2(2a_{i,j} - 1) + \lambda_i \lambda_j(4 - a_{i,j} - a_{i,j}^{-1}))}{(\lambda_i^2(a_{i,j}^{-1} - 1) + \lambda_j^2(a_{i,j} - 1) + \lambda_i \lambda_j(2 - a_{i,j} - a_{i,j}^{-1}))}$$

avec $a_{i,j} \stackrel{\text{def}}{=} \frac{\beta_i}{\beta_j} < 1$,

$$g^{ij} = -\frac{\lambda_i \lambda_j}{2(\lambda_i - (1 + \beta)\lambda_j)} \left[(1 + \beta)^2 - \frac{2\lambda_i}{\lambda_j} \left(1 + \beta + \frac{\beta \lambda_i}{2(\lambda_j - \lambda_i)} \right) \right]$$

$$h^{ij} = \frac{\lambda_i}{\beta} \left(1 + \beta - \frac{\beta \lambda_i}{2(\lambda_i - \lambda_j)} \right).$$

3.3 Distribution asymptotique d'estimateurs des matrices de projections associées

Par un théorème de continuité, (énoncé par exemple dans ([7, theorem 6.2a, p. 387])), nous pouvons déduire que les matrices de projection associées estimées $\mathbf{P}_t \stackrel{\text{def}}{=} \sum_{k=1}^r \mathbf{w}_{t,k} \mathbf{w}_{t,k}^T$ obéissent aussi à la convergence en loi (12), dont l'expression de la matrice de covariance asymptotique \mathbf{C}_p est donnée par :

$$\mathbf{C}_p = \sum_{1 \leq i \leq j \leq n} (b_{j,i} 1_{i \leq r < j} + d_{i,j} 1_{i < j \leq r}) (\mathbf{v}_i \otimes \mathbf{v}_j + \mathbf{v}_j \otimes \mathbf{v}_i) (\mathbf{v}_i \otimes \mathbf{v}_j + \mathbf{v}_j \otimes \mathbf{v}_i)^T \quad (22)$$

où $b_{j,i}$ et $d_{j,i}$ ont pour expressions respectives :

$$b_{j,i}^{SGA} = \frac{\alpha_i \lambda_i \lambda_j}{2(\lambda_i - \lambda_j)}, \quad d_{i,j}^{SGA} = 0$$

$$b_{j,i}^{GHA} = \frac{\lambda_i \lambda_j}{2(\lambda_i - \lambda_j)}, \quad d_{i,j}^{GHA} = \frac{\lambda_j}{2}$$

$$b_{j,i}^{WSA} = \frac{\lambda_i \lambda_j}{2(\lambda_i - \lambda_j)}, \quad d_{i,j}^{WSA} = \lambda_i \lambda_j (b^{ij} + c^{ij} + 2d^{ij})$$

$$b_{j,i}^{OFA} = \frac{\lambda_i \lambda_j}{2(\lambda_j - \lambda_i)}, \quad d_{i,j}^{OFA} = \frac{\lambda_i \lambda_j}{2(\lambda_i - \lambda_j)} + 2h^{ij} + g^{ij}$$

Ne convergeant pas vers les vecteurs propres dominants, l'algorithme SNL ne peut être directement étudié à partir des résultats de Benveniste. Pour résoudre cette difficulté, nous suivrons l'approche utilisée dans [8], dont nous rappelons les principales étapes. Nous associons à l'algorithme (3), l'algorithme suivi par la matrice $\mathbf{P}_t \stackrel{\text{def}}{=} \mathbf{W}_t \mathbf{W}_t^T$ avec $\mathbf{W}_t \stackrel{\text{def}}{=} (\mathbf{w}_{t,1}, \dots, \mathbf{w}_{t,r})$.

$$\mathbf{P}_{t+1} = \mathbf{P}_t + \gamma_t f(\mathbf{P}_t, \mathbf{x}_t \mathbf{x}_t^T) + \gamma_t^2 h(\mathbf{P}_t, \mathbf{x}_t \mathbf{x}_t^T) \quad (23)$$

avec

$$f(\mathbf{P}, \mathbf{M}) \stackrel{\text{def}}{=} (\mathbf{I}_n - \mathbf{P}) \mathbf{M} \mathbf{P} + \mathbf{P} \mathbf{M} (\mathbf{I}_n - \mathbf{P}) \quad (24)$$

$$h(\mathbf{P}, \mathbf{M}) \stackrel{\text{def}}{=} (\mathbf{I}_n - \mathbf{P}) \mathbf{M} \mathbf{P} \mathbf{M} (\mathbf{I}_n - \mathbf{P}) \quad (25)$$

Une deuxième difficulté se présente pour cet algorithme (23), car certaines valeurs propres de la dérivée de son champ moyen sont réelles positives. Pour appliquer les résultats de Benveniste, nous associons finalement à l'algorithme (23), l'algorithme stochastique suivi par le vecteur Θ_t représentant $\frac{\gamma}{2}(2n - r + 1)$ composantes de la matrice \mathbf{P}_t , $n \times n$ symétrique de rang r dans une base judicieuse de l'espace vectoriel des matrices symétriques réelles. Cela nous permet d'appliquer les résultats de Benveniste pour le paramètre Θ_t et par le changement de variable $\Theta_t \leftrightarrow \mathbf{P}_t$, d'en déduire l'expression analytique de la matrice de covariance asymptotique \mathbf{C}_p :

$$\mathbf{C}_p = \sum_{1 \leq i \leq r < j \leq n} \frac{\lambda_i \lambda_j}{2(\lambda_i - \lambda_j)} (\mathbf{v}_i \otimes \mathbf{v}_j + \mathbf{v}_j \otimes \mathbf{v}_i) (\mathbf{v}_i \otimes \mathbf{v}_j + \mathbf{v}_j \otimes \mathbf{v}_i)^T \quad (26)$$

3.4 Analyse des résultats

Ces résultats sont comparés en particulier à ceux obtenus en traitement par bloc. Nous avons, en désignant par \mathbf{W}_t , la matrice contenant les r vecteurs propres associés à l'EVD de la matrice de covariance empirique $\frac{1}{t} \sum_{k=1}^t \mathbf{x}_k \mathbf{x}_k^T$, la convergence en loi suivante pour $t \rightarrow \infty$ (voir par exemple ([9, theorem 13.5.1 p. 541])) :

$$\sqrt{t}(\mathbf{W}_t - \mathbf{W}_*) \xrightarrow{\mathcal{L}} \mathcal{N}(0, \mathbf{C}_W), \quad (27)$$

avec :

$$\mathbf{C}_W = \sum_{\substack{1 \leq i \leq r \\ 1 \leq k \neq i \leq n}} \frac{\lambda_i \lambda_k}{(\lambda_i - \lambda_k)^2} (\mathbf{e}_i \mathbf{e}_i^T \otimes \mathbf{v}_k \mathbf{v}_k^T) - \sum_{1 \leq i \neq j \leq r} \frac{\lambda_i \lambda_j}{(\lambda_i - \lambda_j)^2} (\mathbf{e}_i \mathbf{e}_j^T \otimes \mathbf{v}_j \mathbf{v}_i^T) \quad (28)$$

et d'après [8] :

$$\sqrt{t}(\mathbf{P}_t - \mathbf{P}_*) \xrightarrow{\mathcal{L}} \mathcal{N}(0, \mathbf{C}_p) \quad (29)$$

avec :

$$\mathbf{C}_p = \sum_{1 \leq i \leq r < j \leq n} \frac{\lambda_i \lambda_j}{(\lambda_i - \lambda_j)^2} (\mathbf{v}_i \otimes \mathbf{v}_j + \mathbf{v}_j \otimes \mathbf{v}_i) (\mathbf{v}_i \otimes \mathbf{v}_j + \mathbf{v}_j \otimes \mathbf{v}_i)^T. \quad (30)$$

Nous constatons que ces expressions sont tout à fait voisines des expressions respectives (13) et (22) (26) de \mathbf{C}_W et \mathbf{C}_p de nos algorithmes adaptatifs.

Des expressions analytiques des erreurs quadratiques moyennes sont données en conjecturant que les convergences en loi sont accompagnées de la convergence des deux premiers moments, comme c'est le cas en traitement batch : $\|E(\mathbf{W}_t) - \mathbf{W}_*\|_{\text{Fro}}^2 = o(\gamma)$ et $\text{Cov}(\text{Vec}\mathbf{W}_t) = \gamma\mathbf{C}_W + o(\gamma)$, et par développement de \mathbf{P}_t autour de \mathbf{P}_* , $\|E(\mathbf{P}_t) - \mathbf{P}_*\|_{\text{Fro}}^2 = o(\gamma)$ and $\text{Cov}(\text{Vec}\mathbf{P}_t) = \gamma\mathbf{C}_P + o(\gamma)$. On obtient alors :

$$E\|\mathbf{W}_t - \mathbf{W}_*\|_{\text{Fro}}^2 = \gamma\text{Tr}(\mathbf{C}_W) + o(\gamma) \quad (31)$$

$$E\|\mathbf{P}_t - \mathbf{P}_*\|_{\text{Fro}}^2 = \gamma\text{Tr}(\mathbf{C}_P) + o(\gamma) \quad (32)$$

dont les expressions de $\text{Tr}(\mathbf{C}_W)$ et de $\text{Tr}(\mathbf{C}_P)$ s'obtiennent immédiatement à partir de (13) et de (22) (26) [3].

4 Simulations

Enfin des simulations sont présentées afin de montrer que ces résultats analytiques asymptotiques ont un très large domaine de validité pour γ (Fig. 1). Les vitesses de convergence et les déviations à l'orthogonalité $E\|\mathbf{W}_t^T\mathbf{W}_t - \mathbf{I}_r\|_{\text{Fro}}^2$ des vecteurs propres estimés par ces algorithmes peuvent être comparées à erreur quadratique moyenne imposée [3]. On montre en particulier que cette déviation à l'orthogonalité varie en γ pour les algorithmes GHA, WSA et OFA et en γ^2 pour les algorithmes SGA et SNL (Fig. 2).

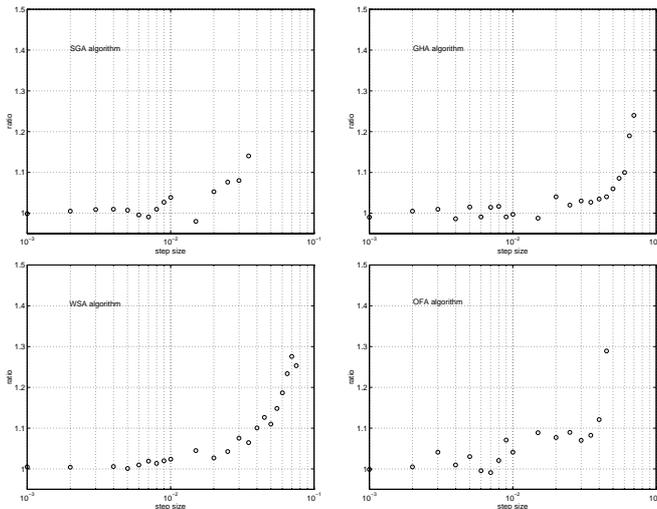


FIG. 1 — Rapport de l'erreur quadratique moyenne estimée en moyennant 100 réalisations indépendantes à l'erreur théorique asymptotique $E\|\mathbf{P}_t - \mathbf{P}_*\|_{\text{Fro}}^2 = \gamma\text{Tr}(\mathbf{C}_P)$ en fonction de γ pour les algorithmes SGA ($\alpha_2 = 2$), GHA, WSA ($\frac{\beta_2}{\beta_1} = 0.9$) et OFA ($\beta = 5$).

Références

[1] E. Oja, "Principal components, minor components and linear neural networks," *Neural Networks*, vol. 5, pp. 927-935, 1992.
 [2] H. Rutishauser, "Computational aspects of F.L.Bauer's simultaneous iteration method," *Numer. Math.* vol. 13 pp. 4-13, 1969.

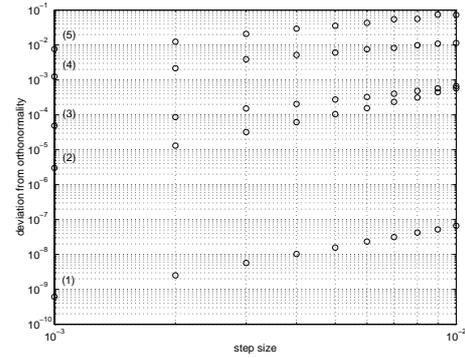


FIG. 2 — Distance à l'orthogonalité $d^2(\gamma) \stackrel{\text{def}}{=} E\|\mathbf{W}_t^T\mathbf{W}_t - \mathbf{I}_r\|_{\text{Fro}}^2$ à la "convergence" estimée en moyennant 100 réalisations indépendantes en fonction de γ en échelle log-log pour les algorithmes de Yang (1) [8], SGA (2), WSA (3), GSA (4), OFA (5).

[3] J.P. Delmas, F. Alberge, "Asymptotic Performance Analysis of Subspace Adaptive Algorithms Introduced in the Neural Network Literature," submitted to *IEEE Trans. on Signal Processing*, Feb. 1997.
 [4] E. Oja, "A simplified neuron model as a principal components analyzer," *Journal of Math. Biology*, vol. 15, pp. 267-273, 1982.
 [5] E. Oja, J. Karhunen, "On stochastic approximation of the eigenvectors and eigenvalues of the expectation of a random matrix," *Journal of math. analysis and applications*, 106, pp. 69-84, 1985.
 [6] A. Benveniste, M. Métivier, P. Priouret, *Adaptive algorithms and stochastic approximations*. Springer Verlag, 1990.
 [7] C.R. Rao, *Linear statistical inference and its applications*, Wiley and Sons, 1973.
 [8] J.P. Delmas, J.F. Cardoso "Performance analysis of an adaptative algorithm for tracking dominant subspaces," submitted to *IEEE Trans. on Signal Processing*, Dec. 1996.
 [9] T.W. Anderson, *An introduction to multivariate statistical analysis*. Second Edition, Wiley and Sons, 1984.