

# Version récursive de l’algorithme EM pour l’estimation en ligne des paramètres de modèles de Markov cachés

Olivier CAPPÉ

LTCI, Télécom ParisTech & CNRS,  
46, rue Barrault, 75634 Paris cédex 13, France  
cappe à telecom-paristech.fr

**Résumé** – Cette contribution est consacrée à l’estimation en ligne (ou récursive) de paramètres de modèles de Markov cachés. L’algorithme proposé constitue une version récursive de l’algorithme EM (Expectation-Maximization) qui généralise la construction de Mongillo & Denève (2008) au cas où les observations ne prennent plus un nombre fini de valeurs. On montre également comment cet algorithme peut être relié à l’existence d’un algorithme EM limite ainsi qu’à l’implémentation récursive du lissage de fonctionnelles additives de l’état pour les modèles de Markov cachés.

**Abstract** – This paper describes an algorithm for online (or recursive) estimation of parameters in Hidden Markov Models (HMMs). The algorithm is based on the familiar EM (Expectation-Maximization) principle and generalizes the proposal of Mongillo & Denève (2008) to the case of continuous observations. The proposed algorithm is related to the existence of a limiting EM recursion and to the availability of a general recursive implementation of smoothing for sum functionals of the state in HMMs.

## 1 Introduction

Depuis leur introduction à la fin des années 1960, le succès des modèles de Markov cachés ne s’est pas démenti et ils restent un des modèles les plus utilisés en traitement du signal et en statistique appliquée. L’estimation des paramètres de tels modèles nécessite cependant le recours à l’optimisation numérique du fait de la forme relativement complexe de la vraisemblance. L’approche la plus commune dans le cas hors ligne (ou en bloc), c’est-à-dire le cas où toutes les données sont disponibles, consiste à utiliser l’algorithme EM (Expectation-Maximization) proposé dès les premiers papiers de Baum et ses collègues [1]. Dans le contexte “en ligne” où les données ne peuvent être stockées et où le volume de données peut être très important, la situation est nettement moins claire. Les algorithmes connus pour l’estimation en ligne reposent soit sur une forme d’approximation des relations de lissage [6] ou du critère de vraisemblance lui-même [10], soit sur l’utilisation d’algorithmes de type gradient stochastique [7] qui ne s’inspirent pas directement du principe de l’algorithme EM et rencontrent des problèmes de calibration des pas de l’algorithme dans le cas de paramètres multiples.

Récemment, l’algorithme proposé par [8] a ouvert une piste nouvelle dans la mesure où il repose véritablement sur une écriture récursive des quantités utilisées dans l’algorithme EM traditionnel. L’algorithme est toutefois décrit dans le cas d’observations discrètes uniquement et ne semble pas, a priori, se prêter à une généralisation. Le but de cette contribution est de montrer que cet algorithme a en fait une portée plus générale en le reliant, d’une part, au cadre introduit dans [3] pour l’estimation dans les mélanges (indépendants) et, d’autre part, aux schémas de calcul récursif de fonctionnelles additives dans les modèles de Markov cachés [11, 5, 4].

## 2 Principe de l’algorithme EM en ligne

On considère tout d’abord le cas plus simple d’observations indépendantes et identiquement distribuées en notant  $Y_t$  les observations,  $X_t$  les données latentes non observées et  $p_\theta$  leur loi jointe. On se focalise sur le cas où  $p_\theta$  appartient à une famille exponentielle :

$$p_\theta(x_t, y_t) = c(x_t, y_t) \exp \{ -A(\theta) + \langle s(x_t, y_t), \phi(\theta) \rangle \},$$

où  $s(x_k, y_k)$  est la statistique exhaustive associée aux données “complètes” (observées et non observées). Dans ce cadre, si on dispose d’observations  $Y_1, \dots, Y_n$ , la  $k$ ième itération de l’algorithme EM prend la forme

**Étape E** Calcul de  $E_{\theta_k} [s(X_t, Y_t) | Y_t]$ , pour  $t = 1, \dots, n$ .

**Étape M** Mise à jour du paramètre par

$$\theta_{k+1} = \bar{\theta} \left( \frac{1}{n} \sum_{t=1}^n E_{\theta_k} [s(X_t, Y_t) | Y_t] \right),$$

où  $\bar{\theta}$  est la fonction qui à  $S$  associe le maximum de  $-A(\theta) + \langle S, \phi(\theta) \rangle$ .

Une première constatation intéressante est que la loi des grand nombre (uniforme) implique que lorsque  $n$  augmente, chaque étape de l’algorithme EM se rapproche d’une mise à jour déterministe appelée “algorithme EM limite” dans [3]. L’algorithme EM limite consiste en la mise à jour

$$\theta_{k+1} = \bar{\theta} (E_{\theta_*} \{ E_{\theta_k} [s(X_t, Y_t) | Y_t] \}), \quad (1)$$

dans laquelle la moyenne empirique sur les observations a été remplacée par l’espérance par rapport à la loi des observations pour la vraie valeur (inconnue) du paramètre. Cette observation permet de comprendre pourquoi les performances de l’algorithme EM en bloc deviennent décevantes lorsque  $n$  augmente puisque le coût de calcul augmente proportionnellement à  $n$  tandis que l’effet de chaque itération tend à se stabiliser [9].

L'algorithme étudié dans [3] pour approcher (1) empiriquement est un algorithme classique d'optimisation stochastique (ou de Robbins-Monro) de la forme

$$\hat{S}_{n+1} = \gamma_{n+1} \mathbb{E}_{\bar{\theta}(\hat{S}_n)} [s(X_{n+1}, Y_{n+1}) | Y_{n+1}] + (1 - \gamma_{n+1}) \hat{S}_n, \quad (2)$$

où  $\gamma_n \in ]0, 1[$  est un pas décroissant; l'estimation correspondante du paramètre étant donnée par  $\hat{\theta}_{n+1} = \bar{\theta}(\hat{S}_{n+1})$ . L'intuition principale consiste ici à *reparamétriser le modèle dans le domaine des statistiques exhaustives* en utilisant l'algorithme d'approximation stochastique sur  $\hat{S}_n$  plutôt que sur  $\hat{\theta}_n$  directement.

Un raisonnement analogue dans les cas des modèles de Markov cachés montre que, sous des hypothèses de mélange suffisantes [2], l'algorithme EM limite doit avoir la forme

$$\theta_{k+1} = \bar{\theta}(\mathbb{E}_{\theta_*} \{ \mathbb{E}_{\theta_k} [s(X_{t-1}, X_t, Y_t) | Y_{-\infty:\infty}] \}), \quad (3)$$

où la statistique  $s$  est maintenant associée à la représentation sous forme exponentielle de la loi  $p_\theta(x_t, y_t | x_{t-1})$ . La difficulté dans le cas des modèles de Markov cachés vient du fait que l'approximation la plus naturelle du second membre de (3),  $\frac{1}{n} \mathbb{E}_\theta [\sum_{t=1}^n s(X_{t-1}, X_t, Y_t) | Y_{0:n}]$ , ne peut plus être mise à jour récursivement, à la différence du cas d'observations indépendantes. La contribution principale de [8] a été de proposer une récursion permettant de mettre à jour récursivement (en  $n$ ) une quantité liée à  $\frac{1}{n} \mathbb{E}_\theta [\sum_{t=1}^n s(X_{t-1}, X_t, Y_t) | Y_{0:n}]$  dans le cas des modèles de Markov cachés à observations discrètes.

Pour généraliser ce résultat, le premier élément important est de reconnaître dans la récursion de [8] une version particulière d'une approche de portée très générale permettant de calculer de façon récursive des fonctionnelles additives d'une chaîne de Markov cachée [11, 5, 4]. Plus précisément, si on note  $q_\theta(x_t, x_{t+1})$  la matrice de transition de la chaîne  $X_t$ ,  $\nu(x_0)$  sa probabilité initiale,  $g_\theta(x_t, y_t)$  la densité de transition de l'observation (loi de  $Y_t$  sachant  $X_t$ )<sup>1</sup> et en définissant les deux quantités

$$\phi_{n,\theta}(x) = P_\theta(X_n = x | Y_{0:n}), \quad (4)$$

$$\rho_{n,\theta}(x) = \frac{1}{n} \mathbb{E}_\theta \left[ \sum_{t=1}^n s(X_{t-1}, X_t, Y_t) \middle| Y_{0:n}, X_n = x \right], \quad (5)$$

il devient possible de mettre à jour la seconde quantité de façon récursive sous la forme

$$\rho_{n+1,\theta}(x) = \sum_{x' \in \mathcal{X}} \left\{ \frac{1}{n+1} s(x', x, Y_{n+1}) + \left( 1 - \frac{1}{n+1} \right) \times \rho_{n,\theta}(x') \right\} \frac{\phi_{n,\theta}(x') q_\theta(x', x)}{\sum_{x'' \in \mathcal{X}} \phi_{n,\theta}(x'') q_\theta(x'', x)}, \quad (6)$$

où  $\mathcal{X}$  désigne l'ensemble des valeurs prise par  $X_t$ . L'argument principal qui justifie cette récursion (cf. chapitre 4 de [4]) consiste à noter que

$$P_\theta(x_t, x_{t+1} | x_{n+1}, Y_{0:n+1}) = \sum_{x_n \in \mathcal{X}} P_\theta(x_t, x_{t+1} | x_n, Y_{0:n}) P_\theta(x_n | x_{n+1}, Y_{0:n}),$$

<sup>1</sup>  $s(x_{t-1}, x_t, y_t)$  est donc la statistique exhaustive associée à la représentation sous forme exponentielle du produit  $q_\theta(x_t, x_{t+1}) g_\theta(x_t, y_t)$ .

le terme de droite de l'équation (6) étant précisément égal à  $P_\theta(X_n = x' | X_{n+1} = x, Y_{0:n})$ . Par la suite, la quantité d'intérêt peut être retrouvée à partir de  $\rho_{n,\theta}$  grâce à la relation

$$\sum_{x \in \mathcal{X}} \rho_{n,\theta}(x) \phi_{n,\theta}(x) = \frac{1}{n} \mathbb{E}_\theta \left[ \sum_{t=1}^n s(X_{t-1}, X_t, Y_t) \middle| Y_{0:n} \right].$$

C'est de ce principe dont s'inspire implicitement la contribution de [8] qui n'est donc absolument pas limitée au cas où les observations sont discrètes. De façon générale, et par analogie avec (2), on peut donc proposer un algorithme EM d'estimation en ligne des paramètres d'un modèle de Markov caché qui a la forme suivante,

**Algorithme 1** *Etant donné une séquence de poids  $(\gamma_n)_{n \geq 1} \in ]0, 1[$  vérifiant  $\sum_{n \geq 1} \gamma_n = \infty$  et  $\sum_{n \geq 1} \gamma_n^2 < \infty$ ,*

**Initialisation** *Calculer, pour  $x \in \mathcal{X}$ ,*

$$\hat{\phi}_0(x) = \frac{\nu(x) g_{\hat{\theta}_0}(x, Y_0)}{\sum_{x' \in \mathcal{X}} \nu(x') g_{\hat{\theta}_0}(x', Y_0)},$$

$$\hat{\tau}_0(x) = 0,$$

*où  $\hat{\theta}_0$  est arbitraire.*

**Récursion** *Pour  $n \geq 0$ ,*

*Calculer, pour  $x \in \mathcal{X}$ ,*

$$\hat{\phi}_{n+1}(x) = \frac{\sum_{x' \in \mathcal{X}} \hat{\phi}_n(x') q_{\hat{\theta}_n}(x', x) g_{\hat{\theta}_n}(x, Y_{n+1})}{\sum_{x', x'' \in \mathcal{X}} \hat{\phi}_n(x') q_{\hat{\theta}_n}(x', x'') g_{\hat{\theta}_n}(x'', Y_{n+1})}, \quad (7)$$

$$\hat{\rho}_{n+1}(x) = \sum_{x' \in \mathcal{X}} \left\{ \gamma_{n+1} s(x', x, Y_{n+1}) + (1 - \gamma_{n+1}) \times \hat{\rho}_n(x') \right\} \frac{\hat{\phi}_n(x') q_{\hat{\theta}_n}(x', x)}{\sum_{x'' \in \mathcal{X}} \hat{\phi}_n(x'') q_{\hat{\theta}_n}(x'', x)}. \quad (8)$$

*Mettre à jour le paramètre avec*

$$\hat{\theta}_{n+1} = \bar{\theta} \left( \sum_{x \in \mathcal{X}} \hat{\rho}_{n+1}(x) \hat{\phi}_{n+1}(x) \right).$$

Pour l'instant, les propriétés de convergence de l'algorithme décrit ci-dessus n'ont pas été établies et elles ne déduisent pas directement des résultats de [3] du fait de la présence du noyau rétrograde  $P_\theta(X_n = x' | X_{n+1} = x, Y_{0:n})$  dans (8) ainsi que de l'utilisation d'un filtre approximatif  $\hat{\phi}_n$ ; approximatif puisqu'on ne reprend pas les récursions de filtrage depuis le début des temps alors que l'estimation  $\hat{\theta}_n$  du paramètre est continuellement remise à jour. Les résultats de simulation présentés dans [2] montrent toutefois que l'approche est robuste et permet, dans certains cas, de retrouver le comportement établi dans [3], à savoir que  $\hat{\theta}_n$  est asymptotiquement efficace.

### 3 Application : modèle de convolution

Pour rendre l'algorithme 1 plus concret et illustrer l'intérêt qu'il présente par rapport à d'autres alternatives dans le cas de modèles typiquement utilisés en traitement de signal, on considère ci-dessous un exemple (simpliste) de modèle de convolution d'une source discrète observée dans un bruit additif gaussien (voir [2] pour un autre exemple). Soit le modèle suivant

$$Y_n = \sum_{k=0}^d h_k U_{n-k} + \sqrt{v} V_n,$$

où  $U_n$  est une source indépendante de loi marginale Bernoulli de paramètre  $p$  et  $V_n$  est un bruit blanc gaussien de variance 1. Les paramètres du modèle sont le filtre  $h = (h_0, \dots, h_d)'$ , la probabilité de source  $p$  ainsi que la variance  $v$  du bruit. Pour simplifier la présentation, on considère dans la suite le cas où  $d = 1$ .

Pour représenter ce modèle sous forme d'un modèle de Markov caché, on introduit un état caché  $X_n \in \{1, \dots, 4\}$  correspondant aux quatre valeurs possibles de  $(U_n, U_{n-1})'$  :

$$\xi_1 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \quad \xi_2 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \quad \xi_3 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad \xi_4 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

avec une matrice de transition associée

$$q_p = \begin{pmatrix} (1-p) & 0 & p & 0 \\ (1-p) & 0 & p & 0 \\ 0 & (1-p) & 0 & p \\ 0 & (1-p) & 0 & p \end{pmatrix},$$

et la représentation  $Y_n = h' \xi_{X_n} + \sqrt{v} V_n$ .

L'algorithme EM usuel (en bloc) appliqué à ce modèle conduit au calcul des statistiques suivantes au cours de l'étape E :

$$S_n^p(i, j) = \frac{1}{n} E_\theta \left[ \sum_{t=1}^n \mathbb{1}\{X_{t-1} = i, X_t = j\} \middle| Y_{0:n} \right],$$

$$S_{n,1}^h = \frac{1}{n} E_\theta \left[ \sum_{t=1}^n Y_t \xi_{X_t} \middle| Y_{0:n} \right],$$

$$S_{n,2}^h = \frac{1}{n} E_\theta \left[ \sum_{t=1}^n \xi_{X_t} \xi_{X_t}' \middle| Y_{0:n} \right],$$

$$S_{n,1}^v = \frac{1}{n} \sum_{t=1}^n 1 = 1,$$

$$S_{n,2}^v = \frac{1}{n} \sum_{t=1}^n Y_t^2,$$

$S_{n,1}^v$  et  $S_{n,2}^v$  étant particulièrement simples car elles ne dépendent que des observations. Lors de l'étape M, la mise à jour des paramètres se fait selon les équations

$$\begin{aligned} p &= (S_n^p(1, 3) + S_n^p(2, 3) + S_n^p(2, 4) + S_n^p(4, 4)) \\ &\quad / (S_n^p(1, 3) + S_n^p(2, 3) + S_n^p(2, 4) + S_n^p(4, 4) \\ &\quad + S_n^p(1, 1) + S_n^p(2, 1) + S_n^p(3, 2) + S_n^p(4, 2)), \\ h &= (S_{n,2}^h)^{-1} S_{n,1}^h, \\ v &= (S_{n,2}^v - h' S_{n,1}^h) / S_{n,1}^v, \end{aligned} \quad (9)$$

qui définissent la fonction  $\bar{\theta} : (S_n^p, S_n^h, S_n^v) \mapsto (p, h, v)$ .

Dans le cas de ce modèle, l'itération principale de l'algorithme 1 (on omet l'initialisation par manque de place) devient :

**Algorithme 2** Pour  $n \geq 0$ , calculer, pour  $i, j, k \in \{1, \dots, 4\}$ ,

$$\hat{\phi}_{n+1}(k) = \frac{\sum_{k'=1}^4 \hat{\phi}_n(k') q_{\hat{p}_n}(k', k) g_{\hat{h}_n, \hat{v}_n}(k, Y_{n+1})}{\sum_{k', k''=1}^4 \hat{\phi}_n(k') q_{\hat{p}_n}(k', k'') g_{\hat{h}_n, \hat{v}_n}(k'', Y_{n+1})}, \quad (10)$$

où  $g_{h,v}(k, y) = \exp[-(y - h' \xi_k)^2 / (2v)]$ ;

$$\begin{aligned} \hat{\rho}_{n+1}^p(i, j, k) &= \gamma_{n+1} \delta(j - k) r_{n+1}(i, j) \\ &\quad + (1 - \gamma_{n+1}) \sum_{k'=1}^m \hat{\rho}_n^p(i, j, k') r_{n+1}(k', k), \end{aligned}$$

$$\begin{aligned} \hat{\rho}_{n+1,1}^h(k) &= \gamma_{n+1} Y_{n+1} \xi_k \\ &\quad + (1 - \gamma_{n+1}) \sum_{k'=1}^m \hat{\rho}_{n,1}^h(k') r_{n+1}(k', k), \end{aligned}$$

$$\begin{aligned} \hat{\rho}_{n+1,2}^h(k) &= \gamma_{n+1} \xi_k \xi_k' \\ &\quad + (1 - \gamma_{n+1}) \sum_{k'=1}^m \hat{\rho}_{n,1}^h(k') r_{n+1}(k', k), \end{aligned}$$

où

$$r_{n+1}(i, j) = \frac{\hat{\phi}_n(i) q_{\hat{p}_n}(i, j)}{\sum_{k'=1}^m \hat{\phi}_n(k') q_{\hat{p}_n}(k', j)};$$

$$\hat{S}_{n+1}^p(i, j) = \sum_{k'=1}^4 \hat{\rho}_{n+1}^p(i, j, k') \hat{\phi}_{n+1}(k'),$$

$$\hat{S}_{n+1,1}^h = \sum_{k'=1}^4 \hat{\rho}_{n+1,1}^h(k') \hat{\phi}_{n+1}(k'),$$

$$\hat{S}_{n+1,2}^h = \sum_{k'=1}^4 \hat{\rho}_{n+1,2}^h(k') \hat{\phi}_{n+1}(k'),$$

$$\hat{S}_{n+1,1}^v = \gamma_{n+1} + (1 - \gamma_{n+1}) \hat{S}_{n,1}^v,$$

$$\hat{S}_{n+1,2}^v = \gamma_{n+1} Y_{n+1}^2 + (1 - \gamma_{n+1}) \hat{S}_{n,2}^v.$$

Mettre à jour le paramètre avec

$$(\hat{p}_{n+1}, \hat{h}_{n+1}, \hat{v}_{n+1}) = \bar{\theta} \left( \hat{S}_{n+1}^p, \hat{S}_{n+1}^h, \hat{S}_{n+1}^v \right), \quad (11)$$

où  $\bar{\theta}$  est défini par (9).

Dans l'algorithme ci-dessus, on reconnaît l'étape de filtrage approximatif en (10), l'approximation stochastique appliquée à l'étape E dans les équations qui suivent et enfin, l'étape M en (11). On constate que l'utilisation d'une étape d'approximation stochastique qui porte sur les statistiques et non sur les paramètres permet effectivement de traiter de façon transparente les contraintes qui portent sur les paramètres (ici  $p \in [0, 1]$  et  $v \geq 0$ ). Il est aisé de vérifier que les matrices  $\hat{S}_{n+1}^p$  et  $r_{n+1}$  sont des matrices creuses qui ont la même structure que  $q_p$ . La complexité numérique par observation de cet algorithme est donc de l'ordre de deux fois le nombre de valeurs prises par  $X_n$  (ici  $4 = 2^{d+1}$ ), multiplié par la dimension des statistiques nécessaires lors de l'étape M, c'est à dire ici, principalement,  $(d+1)^2$  pour  $\hat{S}_{n,2}^h$ .

Pour illustrer le fonctionnement de l'algorithme, on considère des trajectoires simulées avec les paramètres  $p = 0.05$ ,  $h = (0.9, -0.3)'$  et  $v = 0.1$ , ce qui correspond à un rapport signal à bruit de -3dB au niveau des observations  $Y_n$  (voir la figure 1 pour une trajectoire typique). Le comportement de l'algorithme est illustré par la figure 2 pour une valeur du pas de  $\gamma_n = n^{-0.6}$ , les trois trajectoires superposées sur la figure 2 donnant une idée de la variabilité. On note que l'algorithme fournit une estimation raisonnable des paramètres pour un nombre d'observations de l'ordre de quelques milliers. Ce comportement est assez typique de celui observé en général : l'algorithme en

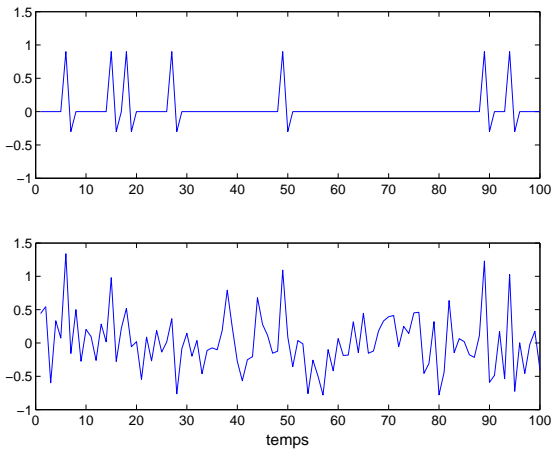


FIG. 1 – Trajectoire typique de  $U_n$  (en haut) et de  $Y_n$  (en bas).

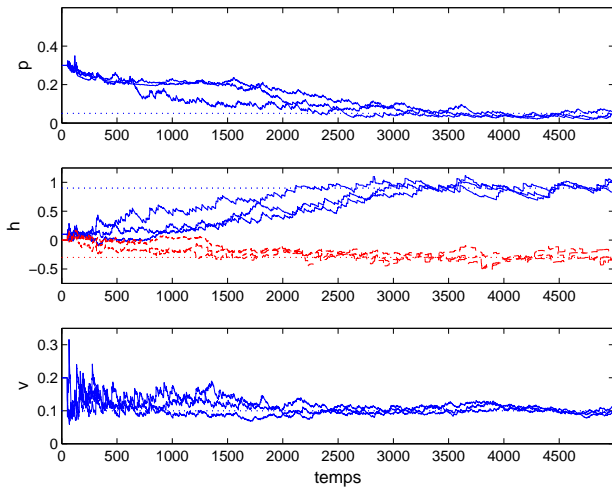


FIG. 2 – Convergence des paramètres estimés avec l’algorithme en ligne pour  $\gamma_n = n^{-0.6}$  (trois trajectoires indépendantes superposées). Les vraies valeurs des paramètres sont indiquées en pointillés.

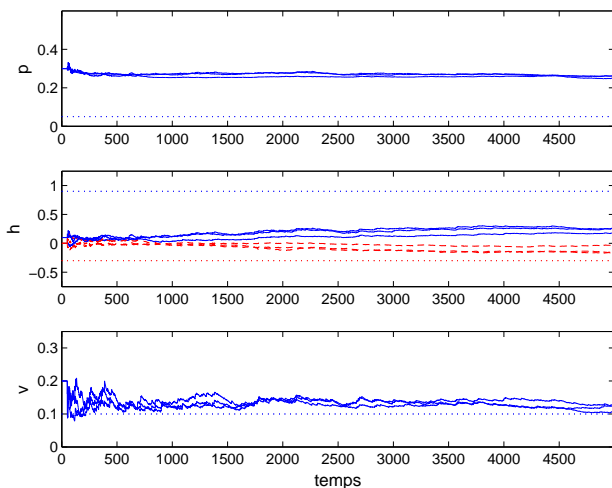


FIG. 3 – Même figure que ci-dessus avec  $\gamma_n = n^{-0.75}$ .

ligne est peu précis lorsque le nombre d’observations est faible (de l’ordre de quelques centaines) mais devient très compétitif lorsque le nombre d’observations dépasse plusieurs milliers. On note sur la figure 2 que la variance de l’estimateur en ligne ne décroît que lentement du fait du choix d’un pas qui lui-même décroît assez lentement ( $\gamma_n = n^{-0.6}$ ). La solution recommandée dans [3] consiste à utiliser la “moyennisation” dite de Polyak-Ruppert qui reviendrait ici à remplacer les estimés obtenus au delà de la trois millième observation (par exemple) par leur moyenne. De façon générale, cette façon de procéder est plus recommandable que le fait d’utiliser des pas décroissants rapidement mais cela semble particulièrement vrai dans cette application (cf. figure 3), probablement du fait de l’information assez faible apportée par les observations individuelles (penser en particulier à la façon dont  $p$  est estimé) mais aussi du temps nécessaire à l’oubli des conditions initiales par le filtre approximatif. Il est intéressant en particulier de noter que le choix de  $\gamma_n = 1/n$  qui conduit pourtant à une forte analogie signalée ci-dessus avec l’algorithme EM en bloc est inutilisable dans le contexte d’un algorithme en ligne.

## Références

- [1] L. E. Baum, T. P. Petrie, G. Soules, and N. Weiss. A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains. *Ann. Math. Statist.*, 41(1) :164–171, 1970.
- [2] O. Cappé. Online EM algorithm for hidden Markov models. preprint, 2009.
- [3] O. Cappé and E. Moulines. On-line expectation-maximization algorithm for latent data models. *J. Roy. Statist. Soc. B*, 71(3) :593–613, 2009.
- [4] O. Cappé, E. Moulines, and T. Rydén. *Inference in Hidden Markov Models*. Springer, 2005.
- [5] R. J. Elliott, L. Aggoun, and J. B. Moore. *Hidden Markov Models : Estimation and Control*. Springer, New York, 1995.
- [6] V. Krishnamurthy and J. B. Moore. On-line estimation of hidden Markov model parameters based on the kullback-leibler information measure. *IEEE Trans. Signal Process.*, 41(8) :2557–2573, 1993.
- [7] F. Le Gland and L. Mevel. Recursive estimation in HMMs. In *Proc. IEEE Conf. Decis. Control*, pages 3468–3473, 1997.
- [8] G. Mongillo and S. Denève. Online learning with hidden Markov models. *Neural Computation*, 20(7) :1706–1716, 2008.
- [9] R. M. Neal and G. E. Hinton. A view of the EM algorithm that justifies incremental, sparse, and other variants. In M. I. Jordan, editor, *Learning in graphical models*, pages 355–368. MIT Press, Cambridge, MA, USA, 1999.
- [10] T. Rydén. On recursive estimation for hidden Markov models. *Stochastic Process. Appl.*, 66(1) :79–96, 1997.
- [11] O. Zeitouni and A. Dembo. Exact filters for the estimation of the number of transitions of finite-state continuous-time Markov processes. *IEEE Trans. Inform. Theory*, 34(4), July 1988.