Importance sampling for online variational learning in state-space models

Mathis Chagneux [1], Pierre Gloaguen [2], Sylvain Le Corff [3], Jimmy Olsson [4]

[1] *Institut Polytechnique de Paris, France, mathis.chagneux@telecom-paris.fr*
[2] *Université Bretagne Sud, France, pierre.gloaguen@univ-ubs.fr*
[3] *Sorbonne Université, France, sylvain.le_ corff@sorbonne-universite.fr*
[4] *KTH Royal Institute of Technology, Sweden, jimmyol@kth.se*

Résumé. Dans ce travail, nous considérons le problème de l'apprentissage en ligne dans les modèles espace d'états, c'est à dire un cadre où les observations dépendent d'états cachés, eux mêmes supposés issus un processus de Markov. Notre objectif est d'apprendre la distribution de lissage, *i.e.* la distribution a posteriori de états cachés conditionnellement aux observations. Nous nous intéressons au cadre de l'apprentissage en ligne, c'est à dire où l'actualisation de la loi se fait à l'arrivée de chaque nouvelle observation, et chaque observation n'est vue qu'une seule fois. Nous proposons un nouvel algorithme pour estimer en ligne la distribution de lissage dans un cadre variationel. Cet algorithme repose sur une estimation en ligne efficace de la fonction de coût classique en inférence variationelle, l'*evidence lower bound* (ELBO), ainsi que de son gradient. Nous mimons ensuite les idées du maximum de vraisemblance récursif pour l'apprentissage en ligne. Nous montrons comment on peut exploiter i) la structure de la vraie loi a posteriori ciblée ii) les idées des approches de Monte Carlo séquentiel et iii) les astuces de paramétrisation des approches variationnelles récentes pour apprendre efficacement une loi a posteriori en grande dimension.

Mots-clés. Apprentissage en ligne ; Inférence variationnelle séquentielle ; Modèles à espace d'état ; Apprentissage pour les séries temporelles

Abstract. In this work, we consider the problem of online learning in state-space models, *i.e.* when the observations are time-series depending on hidden states, themselves assumed to arise from a Markov process. Our objective is to learn the smoothing distribution, *i.e.* the a posteriori distribution of hidden states conditional on the observations. We are interested in the online learning framework, *i.e.* where the updating of the distribution takes place upon the arrival of each new observation, and each observation is seen only once. We propose a new algorithm for online estimation of the smoothing distribution in a variational framework. This algorithm is based on an efficient online estimation of the classical cost function in variational inference, the *evidence lower bound* (ELBO), as well as its gradient. We then mimic the ideas of recursive maximum likelihood for online learning. We show how we can exploit i) the structure of the true targeted a posteriori law ii) the ideas of sequential Monte Carlo approaches and iii) the parameterization tricks of recent variational approaches to efficiently learn a high-dimensional a posteriori law.

Keywords. Online learning ; Sequential variational inference ; State-space models ; Time-series learning

1 Modèle et objectif d'inférence

Soit un modèle de Markov caché (HMM) où le processus caché, à valeurs dans $\mathbb{R}^{d_x}$, est noté $(X_t)t \geqslant 0$. On suppose que la distribution de X_0 admet une densité χ par rapport à la mesure de Lebesgue μ et pour tout $t \geqslant 0$, la distribution conditionnelle de X_{t+1} conditionnellement à $X_{0:t}$ admet une densité $m_t(X_t, \cdot)$. Dans un HMM, on suppose que cet état est partiellement observé à travers un processus d'observation $(Y_t)t \geqslant 0$ prenant des valeurs dans $\mathbb{R}^{d_y}$. Les observations $Y_{0:t}$ sont supposées indépendantes conditionnellement à $X_{0:t}$ et, pour tout $t \geqslant 0$, la distribution de Y_t sachant $X_{0:t}$ dépend de X_t uniquement. On suppose que cette distribution admet une densité $g_t(X_t, \cdot)$ par rapport à la mesure de Lebesgue. L'ensemble du modèle est alors défini par la distribution conjointe des états cachés et des observations :

$$p_{0:t}(x_{0:t}, y_{0:t}) = \prod_{s=0}^{t} \ell_s(x_{s-1}, x_s, y_s),$$

où

$$\begin{aligned} \ell_0(x_{-1}, x_0, y_0) &:= \chi(x_0) g_0(x_0, y_0) \\ \ell_s(x_{s-1}, x_s, y_s) &:= m_s(x_{s-1}, x_s) g_s(x_{s-1}, y_s) \qquad \text{for } s \geqslant 1 \,. \end{aligned}$$

1.1 Lissage dans les HMMs

Un des objectifs d'inférence classique dans les HMM est d'estimer la valeur prise par les états cachés, conditionnellement aux observations. Formellement, cela consiste à estimer la distribution de lissage, c'est-à-dire la loi conditionnelle de $X_{0:t}$ sachant $Y_{0:t}$. Dans notre cadre (où tout est défini via des densités), cette distribution admet également une densité donnée par

$$\phi_{0:t}(x_{0:t}) \propto p_{0:t}(x_{0:t}, y_{0:t}) \,.$$

La marginale au temps t de cette distribution conjointe est appelée *distribution de filtrage* au temps t, et sa densité par rapport à la mesure de Lebesgue s'écrit ϕ_t. On montre facilement que cette densité de lissage peut se factoriser de la manière suivant :

$$\phi_{0:t}(x_{0:t}) = \phi_t(x_t) \prod_{s=1}^{t} b_{s-1|s}(x_s, x_{s-1}) \,. \tag{1}$$

où, pour $1 \leqslant s \leqslant t$, les noyaux

$$b_{s-1|s}(x_s, x_{s-1}) = \frac{m_s(x_{s-1}, x_s)\phi_{s-1}(x_{s-1})}{\int m_s(x_{s-1}, x_s)\phi_{s-1}(x_{s-1}) \,\mathrm{d}x_{s-1}}, \tag{2}$$

sont appelés noyaux *backward*. À x_s fixé, ce noyau donne la densité conditionnelle de X_{s-1} étant donnés $(X_s = x_s, Y_{0:s-1} = y_{0:s-1})$. Une remarque importante est que la factorisation donnée dans l'équation (2) met en lumière le caractère markovien de la distribution de lissage. La loi de lissage n'a généralement pas d'expression explicite en raison de l'impossibilité de calculer les lois de filtrage. La section suivante décrit comment on estime la loi de lissage dans un cadre variationel.

1.2 Inférence variationelle séquentielle *backward*

Dans les approches variationnelles, la distribution de lissage $\phi_{0:t}$ est approximée en choisissant un candidat dans une famille paramétrique $\{q^\lambda_{0:t}\}_{\lambda\in\Lambda}$, appelée famille variationnelle, où Λ est un ensemble de paramètres. Un point critique réside donc dans le choix de la forme de la famille variationnelle.

Les approches variationelle classiques [Blei et al., 2017] se basent sur la famille dite de champ moyen, où l'on suppose que :

$$q^\lambda_{0:t}(x_{0:t}) = \prod_{s=0}^{t} q^\lambda_s(x_s, x_{s-1}) \ .$$

Cette famille où tous les états cachés sont donc indépendants a posteriori ne respecte pas la structure markovienne de la distribution de lissage. Par conséquent, la plupart des travaux d'inférence variationelle séquentielle imposent une structure à la famille variationnelle via une décomposition factorisée de $q^\lambda_{0:t}$ sur $x_{0:t}$. Une contrepartie variationnelle de (1), introduite dans les travaux de [Campbell et al., 2021], consiste à définir

$$q^\lambda_{0:t}(x_{0:t}) = q^\lambda_t(x_t) \prod_{s=1}^{t} q^\lambda_{s-1|s}(x_s, x_{s-1}) \ , \tag{3}$$

où les q^λ_t (resp. $q^\lambda_{s-1|s}(x_s, \cdot)$) sont des densités de probabilités, que l'on choisit, qui dépendront de $Y_{0:t}$ (resp. $Y_{0:s-1}$). L'un des principaux avantages de cette factorisation est qu'elle respecte les véritables dépendances induites dans (2). En outre, récemment, [Chagneux et al., 2024a] ont établi une borne supérieure sur l'erreur lorsque l'on approche des espérances par rapport à la distribution de lissage, par des espérances par rapport aux distributions variationnelles satisfaisant cette factorisation.

Au sein de cette famille variationnelle, on choisira le meilleur λ au sens de l'*evidence lower bound* (ELBO), à savoir le λ qui maximise :

$$\mathcal{L}^\lambda_t = \mathbb{E}_{q^\lambda_{0:t}}\left[\log \frac{p_{0:t}(X_{0:t}, Y_{0:t})}{q^\lambda_{0:t}(X_{0:t})}\right] \ . \tag{4}$$

Cette maximisation se fait généralement via des algorithmes de gradient qui nécessitent donc de calculer le gradient de l'ELBO. Les sections suivantes décrivent un nouvel algorithme permettant d'approximer ce gradient de manière récursive.

2 Recursions pour le calcul du gradient de l'ELBO

On définit [1]

$$\tilde{f}^\lambda_t(x_{t-1}, x_t) = \begin{cases} \log \ell_0(x_{-1}, x_0, y_0) & \text{if } t = 0, \\ \log \frac{\ell_t(x_{t-1}, x_t, y_t)}{q^\lambda_{t-1|t}(x_t, x_{t-1})} & \text{if } t > 0 \end{cases} \tag{5}$$

et $f^\lambda_{0:t}(x_{0:t}) = \sum_{s=0}^{t} \tilde{f}^\lambda_s(x_{s-1}, x_s)$. On remarque directement qu'avec ces notations

$$\mathcal{L}^\lambda_t = \mathbb{E}_{q^\lambda_{0:t}}\left[f^\lambda_{0:t}(X_{0:t}) - \log q^\lambda_t(X_t)\right] .$$

Notre méthode d'estimation de λ se base sur les résultats suivant [2] :

1. La dépendance de chaque terme $\tilde{f}^\lambda_t$ en y_t is ommise pour alléger les notations.
2. Dans la suite, tous les gradients sont calculés par rapport à λ

Proposition 1. *Pour l'ELBO et son gradient, on a :*

$$\mathcal{L}_t^\lambda = \mathbb{E}_{q_t^\lambda}\left[H_t^\lambda(X_t)\right] - \mathbb{E}_{q_t^\lambda}\left[\log q_t^\lambda(X_t)\right] \tag{6}$$

$$\nabla\mathcal{L}_t^\lambda = \mathbb{E}_{q_t^\lambda}\left[\{\nabla \log q_t^\lambda \cdot H_t^\lambda\}(X_t) + G_t^\lambda(X_t)\right], \tag{7}$$

où $H_t^\lambda(x_t)$ est une fonction de $\mathbb{R}^{d_x}$ dans $\mathbb{R}$ satisfaisant la recursion

$$H_t^\lambda(x_t) = \mathbb{E}_{q^\lambda_{t-1|t}}\left[H_{t-1}^\lambda(X_{t-1}) + \tilde{f}_t^\lambda(X_{t-1}, x_t)\right] , \tag{8}$$

avec $H_0^\lambda(x_0) = \tilde{f}_0^\lambda(x_{-1}, x_0)$, et où $G_t^\lambda(x_t)$ est une fonction de $\mathbb{R}^{d_x}$ dans Λ satisfaisant $G_0^\lambda(x_0) = 0$ et

$$G_t^\lambda(x_t) = \mathbb{E}_{q^\lambda_{t-1|t}}\left[G_{t-1}^\lambda\left(X_{t-1}\right) + \nabla \log q^\lambda_{t-1|t}(x_t, X_{t-1})\left(H_{t-1}^\lambda(x_{t-1}) + \tilde{f}_t^\lambda(x_{t-1}, x_t)\right)\right] . \tag{9}$$

Démonstration. Voir l'Annexe A. □

Intérêt de la Proposition 1 Pour une séquence fixe de longueur t, le calcul récursif de $\nabla\mathcal{L}_t^\lambda$ consiste donc à (i) calculer de manière récursive $\nabla H_t^\lambda(x_t)$ de 0 à t en utilisant les récursions (8) et (9) et (ii) calculer l'espérance finale grâce à l'équation (7). Cette espérance est prise relativement à la distribution variationnelle choisie q_t^λ, selon laquelle on sait simuler facilement. Ainsi, on pourra utiliser une approximation Monte Carlo pour l'estimer. Il reste à approximer récursivement les fonctions intermédiaires $H_t^\lambda(x_t)$ et $G_t^\lambda(x_t)$, impliquant des espérances conditionnelles par rapport aux noyaux $q^\lambda_{t-1|t}$. Notez que ces noyaux visent à approximer les vrais noyaux *backward* $b_{t-1|t}$, qui dépendent uniquement des observations $Y_{0:t-1}$, c'est-à-dire seulement du passé. Les récursions proposées sont donc adaptées à l'apprentissage en ligne.

3 Apprentissage de λ en ligne

Dans l'apprentissage en ligne, nous visons, à chaque paas de temps t, à actualiser notre estimation $\hat{\lambda}_t$ de λ (obtenue avec les observations $y_{0:t-1}$) à l'aide l'observation y_t.

Initialisation. À partir d'une estimation initiale $\hat{\lambda}_0$, on simule un N-échantillon

$$\{\xi_0^j\}_{1\leq j\leq N} \overset{i.i.d}{\sim} q_0^{\hat{\lambda}_0} ,$$

et on pose

$$\hat{H}_0^{\hat{\lambda}_0,j} = H_0^{\hat{\lambda}_0}(\xi_0^j),$$
$$\hat{G}_0^{\lambda,j} = G_0^{\hat{\lambda}_0}(\xi_0^j)$$

Il est important de remarquer ici que :

— L'observation y_0 intervient dans i) la simulation du N-échantillon (la dépendance de q_0^λ en y_0 étant implicite par souci de notation et ii) dans le calcul de $H_0^{\hat{\lambda}_0}(\xi_0^j)$.

— Les fonctions $H_0^{\hat{\lambda}_0}$ et $G_0^{\hat{\lambda}_0}$ sont simplement évaluées [3] sur un support fini. C'est ce support fini qui permettra la propagation récursive de ces statistiques.

Le premier gradient est donc approché par :

$$\widehat{\nabla}\mathcal{L}_0^{\hat{\lambda}_0} = \frac{1}{N}\sum_{i=1}^{N} \nabla \log q_0^{\hat{\lambda}_0}(\xi_0^i) \cdot \hat{H}_0^{\hat{\lambda}_0,i} + \hat{G}_0^{\hat{\lambda}_0,i} \ .$$

Notre estimation de λ est donc mise à jour en utilisant ce gradient, typiquement en posant :

$$\hat{\lambda}_1 = \hat{\lambda}_0 + \gamma_0 \widehat{\nabla}\mathcal{L}_0^{\hat{\lambda}_0} \ ,$$

pour un certain pas de gradient γ_0.

Approximation récursive de $H_t^{\hat{\lambda}_t}$ and $G_t^{\hat{\lambda}_t}$. Au temps t, on simule un N-échantillon

$$\{\xi_t^i\}_{1\leq i\leq N} \overset{i.i.d}{\sim} q_t^{\hat{\lambda}_t} \ .$$

$H_t^{\hat{\lambda}_t}(\xi_t^i)$ and $G_t^{\hat{\lambda}_t}(\xi_t^i)$ sont estimés respectivement par

$$\hat{H}_t^{\hat{\lambda}_t,i} = \sum_{j=1}^{N} \bar{w}_{t-1|t}^{\hat{\lambda}_t,i,j} \left(\hat{H}_{t-1}^{\hat{\lambda}_{t-1},j} + \tilde{f}_t^{\hat{\lambda}_t}(\xi_{t-1}^j, \xi_t^i) \right) \ , \tag{10}$$

$$\begin{aligned}\hat{G}_t^{\hat{\lambda}_t,i} = \sum_{i=1}^{N} \bar{w}_{t-1|t}^{\hat{\lambda}_t,i,j} \Big\{ \hat{G}_{t-1}^{\hat{\lambda}_{t-1},j} + \nabla \log q_{t-1|t}^{\lambda_t}(\xi_t^i, \xi_{t-1}^j) \times \\ \left(\hat{H}_{t-1}^{\lambda_{t-1},j} + \tilde{f}_t^{\hat{\lambda}_t}(\xi_{t-1}^j, \xi_t^i) \right) \Big\} \ ,\end{aligned} \tag{11}$$

où

$$\bar{w}_{t-1|t}^{\hat{\lambda}_t,i,j} = \frac{q_{t-1|t}^{\hat{\lambda}_t}(\xi_t^i, \xi_{t-1}^j) / q_{t-1}^{\hat{\lambda}_{t-1}}(\xi_{t-1}^j)}{\sum_{k=1}^{N} q_{t-1|t}^{\hat{\lambda}_t}(\xi_t^i, \xi_{t-1}^k) / q_{t-1}^{\hat{\lambda}_{t-1}}(\xi_{t-1}^k)} \ . \tag{12}$$

L'approximation du gradient est alors donnée par l'équation

$$\widehat{\nabla}\mathcal{L}_t^{\hat{\lambda}_t} = \frac{1}{N}\sum_{i=1}^{N} \nabla \log q_t^{\hat{\lambda}_t}(\xi_t^i) \cdot \hat{H}_t^{\hat{\lambda}_t,i} + \hat{G}_t^{\hat{\lambda}_t,i} \ ,$$

qui peut être utilisée pour obtenir $\hat{\lambda}_{t+1}$.

Remarque sur l'algorithme Les équations (10) et (11) sont des estimateurs des equations (8) and (9) par *self-normalized importance sampling* (SNIS), dont l'expression des poids auto-normalisés (partagés par les deux estimateurs) est donnée par (12). On notera ici que l'on ne peut pas obtenir des approximations Monte Carlo directes de (8)-(9) en simulant des échantillons selon $q_{t-1|t}^{\lambda}(\xi_t^i, \cdot)$, car les fonctions $H_{t-1}^{\hat{\lambda}_{t-1}}$ and $G_{t-1}^{\hat{\lambda}_{t-1}}$ n'auraient pas été approchées sur ces échantillons. L'*importance sampling* est donc indispensable pour la mise à jour récursive de l'approximation du gradient. On sait cependant que les performances d'un tel estimateur dépendent fortement du lien entre la loi de proposition

3. Á l'étape $t = 0$, il s'agit d'une évaluation exacte, il s'agira ensuite d'approximations

et la loi cible. La Section 4 propose une mise en oeuvre efficace pour relier la la loi de proposition $q_{t-1}^{\hat{\lambda}_{t-1}}$ à la loi cible $q_{t-1|t}^{\hat{\lambda}_t}$. L'auto-normalisation dans (12) est motivée par des considérations pratiques, car elle réduit la variance de l'estimateur dans nos simulations. Cette réduction de la variance vient cependant au prix de l'ajout d'un biais.

Une autre source de biais dans notre approximation est la présence dans les termes de droite de (10)-(11) de quantités ayant été calculées sous le paramètre $\hat{\lambda}_{t-1}$ pour approcher des quantités sous le paramètre $\hat{\lambda}_t$. Ces approximations, couramment effectuées dans des contextes de maximum de vraisemblance récursif (voir [Tadić and Doucet, 2020] par exemple), sont indispensables pour la mise en place pratique de l'apprentissage en ligne. Nos expériences numériques montrent que ces approximations, qui rendraient l'étude théorique plus complexe, conduisent toujours à de bons résultats dans la pratique.

Le lecteur familier des méthodes de Monte Carlo séquentiel pourra voir des similarités avec l'algorithme PaRIS proposé dans [Olsson et al., 2017] et étendu dans [Gloaguen et al., 2022]. Bien qu'inspiré de ces travaux, notre algorithme présente la différence majeure de ne fonctionner qu'avec des échantillons i.i.d., et selon une loi variationelle que l'on choisit (les tirages sont donc directs). Cet aspect évite le problème de la dégénérescence des poids des méthodes de Monte Carlo séquentiel. Dans la pratique, nous nous servons d'astuces de rééchantillonnage des deux articles cités pour éviter d'avoir à calculer la constante de normalisation dans (12), ce qui entrainerait une complexité en N^2. Les détails sont disponibles dans [Chagneux et al., 2024b].

4 Détails d'implémentation

Définitions de noyaux *backward* à partir de noyaux *forward* L'équation (12) suggère que la performance de l'algorithme proposé dépend fortement de la définition des distributions variationnelles et du lien entre $q_{t-1}^{\hat{\lambda}_{t-1}}$ et $q_{t-1|t}^{\hat{\lambda}_t}$. Nous introduisons donc une structure supplémentaire dans la famille variationnelle donnée par (3), en utilisant des fonctions de potentiel $\psi_t^\lambda : \mathbb{R}^{d_x} \times \mathbb{R}^{d_x} \to \mathbb{R}$ pour relier ces distributions. Les fonctions de potentiel $(q_t^\lambda)_{t\geq 0}$ relient explicitement les distributions $(q_t^\lambda)_{t\geq 0}$ aux noyaux rétrospectifs $(q_{t-1|t}^\lambda)_{t\geq 0}$. Plus précisément, nous imposons que, pour tout $t \geq 1$,

$$q_{t-1|t}^{\hat{\lambda}_t}(x_t, x_{t-1}) \propto q_{t-1}^{\hat{\lambda}_{t-1}}(x_{t-1})\psi_t^{\hat{\lambda}_t}(x_{t-1}, x_t) \; . \tag{13}$$

Les fonctions ψ_t^λ peuvent être rendues arbitrairement complexes, de sorte que, pour tout t, le noyau variationnel *backward* $q_{t-1|t}^{\lambda_t}$ a une dépendance en x_t arbitrairement complexe.

Paramétrisation des distributions variationelles En pratique, l'utilisation de la décomposition (13) dans le cadre en ligne nécessite la paramétrisation explicite, pour tout $t \geq 0$, d'une distribution q_t^λ et d'un potentiel ψ_t^λ (pas nécessairement normalisé) qui dépendent tous deux des observations jusqu'à l'instant t (au maximum). Dans un souci de temps de calcul, chaque q_t^λ est choisi comme une densité paramétrique au sein de la famille exponentielle. Cette famille est notée $\mathsf{P} = \{P_\eta\}_{\eta\in\mathcal{E}}$ où η est le paramètre naturel correspondant et $\mathcal{E}$, l'espace des paramètres de cette famille (typiquement la famille des distributions gaussiennes définies sur $\mathbb{R}^{d_x}$). On note η_t^λ le paramètre de q_t^λ. Pour garantir

que les distributions $q^\lambda_{t-1|t}$ appartiennent à la même famille, nous imposons que

$$\psi^\lambda_t(x_{t-1}, x_t) = \exp\left(\tilde{\eta}^\lambda_t(x_t) \cdot T(x_{t-1})\right) ,$$

où $\tilde{\eta}^\lambda_t(x_t) = \mathsf{MLP}^\lambda(x_t)$ [4] et $T(x_{t-1})$ sont le paramètre naturel et la statistique suffisante pour la famille P. Ainsi, grâce à (13), $q^\lambda_{t-1|t}(x_t, \cdot)$ sera une densité dans P avec pour paramètre naturel $\eta^\lambda_{t-1|t} = \eta^\lambda_{t-1} + \tilde{\eta}^\lambda_t$. Dans ce cadre pratique, les noyaux $q^\lambda_{t-1|t}$ peuvent avoir des dépendances arbitrairement complexes sur x_t tandis que leur densité est obtenue explicitement à partir des potentiels. Cela élimine la nécessité de calculer des constantes de normalisation (requises, par exemple, lors du calcul de (11)), tout en évitant la réduction de ces noyaux à des familles trop simples (par exemple, les noyaux linéaires gaussiens). Pour les paramètres de q^λ_t, il existe deux approches principales :

- *Approches amorties* où les paramètres de q^λ_t sont mis à jour à l'aide d'une fonction paramétrée à chaque instant t. Cela peut se faire à l'aide de quantités intermédiaires $a_t \in \mathsf{A}$ (où A est un espace défini par l'utilisateur), telles que $a_t = \mathsf{MLP}^\lambda(a_{t-1}, y_t)$, et $\eta^\lambda_t = \mathsf{MLP}^\lambda(a_t)$. L'initialisation est effectuée à l'aide d'un paramètre aléatoire a_{-1}, qui peut être fixe ou appris. Les schémas amortis sont efficaces d'un point de vue numérique (puisque les connaissances des prédictions précédentes sont utilisées pour produire les paramètres actuels), mais nécessitent la définition manuelle de fonctions complexes. Les récursions peuvent être analytiques (et ne pas reposer sur un MLP), par exemple lorsque $q^\lambda_{0:t}$ est la distribution de lissage d'une gaussienne linéaire, ou lorsque la conjugaison est davantage exploitée pour mettre à jour les paramètres $(\eta^\lambda_t)_{t\geq 0}$ (voir l'annexe **??**). Quoi qu'il en soit, le nombre de paramètres devient indépendant de t, cependant la rétropropagation des gradients va mécaniquement croître linéairement avec t. Pour éviter cela, une solution consiste à tronquer la rétropropagation, c'est-à-dire à supposer que $(a^\lambda_s)_{s\leq t-\Delta}$ est indépendant de λ pour un certain Δ.
- *Approches non amorties* où chaque q^λ_t et ψ^λ_t ont leurs propres paramètres η_t et $\tilde{\eta}_t$, sans rapport avec ceux du temps $t-1$. Dans ce cas, le vecteur optimisé λ contient les paramètres $(\eta_t)_{t\geq 0}$, et le nombre de paramètres croît alors linéairement avec t. Ce schéma modifie l'équation (9) (voir [Chagneux et al., 2024b] pour plus de détails).

Réduction de variance Les équations (7) et (9) nécessite l'approximation d'espérance selon une fonction de score, i.e. des espérances de la forme $\mathbb{E}_{q^\lambda}\left[\nabla \log q^\lambda(X) \cdot f(X)\right]$, pour une certaine distribution $q^\lambda(X)$. Comme discuté dans [Mohamed et al., 2020], l'estimateur Monte Carlo direct est suvent sujet à une grande variance, et il est souhaitable, pour un bonne performance pratique, d'essayer de réduire la variance en introduisant une variable de contrôle. En remarquant que $\mathbb{E}_{q^\lambda}\left[\nabla \log q^\lambda(X)\right] = 0$, notre espérance cible est donc égale à $\mathbb{E}_{q^\lambda}[\nabla \log q^\lambda(X)(f(X) - \mathbb{E}_{q^\lambda}[f(X)])]$. Or, des estimations Monte Carlo de $\mathbb{E}_{q^\lambda_t}[H^\lambda_t]$ et $\mathbb{E}_{q^\lambda_{t-1|t}}[H^\lambda_{t-1}(X_{t-1}) + \tilde{f}^\lambda_t(X_{t-1}, x_t)]$ sont calculés dans l'algorithme de la Section 3, à savoir $N^{-1}\sum_{i=1}^N \hat{H}^{\lambda,i}_t$ et $\{\hat{H}^{\lambda,i}_t\}_{1\leq i\leq N}$. Notre méthodologie amène donc directement à une réduction de variance efficace.

4. MLP est une notation pour désigner un *multi-layer perceptron*

Références

[Blei et al., 2017] Blei, D. M., Kucukelbir, A., and McAuliffe, J. D. (2017). Variational inference : A review for statisticians. *Journal of the American Statistical Association*, 112(518) :859–877.

[Campbell et al., 2021] Campbell, A., Shi, Y., Rainforth, T., and Doucet, A. (2021). Online variational filtering and parameter learning. *Advances in Neural Information Processing Systems*, 34.

[Chagneux et al., 2024a] Chagneux, M., Gassiat, É., Gloaguen, P., and Le Corff, S. (2024a). Additive smoothing error in backward variational inference for general state-space models. *Journal of Machine Learning Research*.

[Chagneux et al., 2024b] Chagneux, M., Gloaguen, P., Corff, S. L., and Olsson, J. (2024b). Importance sampling for online variational learning.

[Gloaguen et al., 2022] Gloaguen, P., Corff, S. L., and Olsson, J. (2022). A pseudo-marginal sequential Monte Carlo online smoothing algorithm. *Bernoulli*, 28(4) :2606 – 2633.

[Mohamed et al., 2020] Mohamed, S., Rosca, M., Figurnov, M., and Mnih, A. (2020). Monte carlo gradient estimation in machine learning. *The Journal of Machine Learning Research*, 21(1) :5183–5244.

[Olsson et al., 2017] Olsson, J., Westerborn, J., et al. (2017). Efficient particle-based online smoothing in general hidden markov models : the PaRIS algorithm. *Bernoulli*, 23(3) :1951–1996.

[Tadić and Doucet, 2020] Tadić, V. Z. and Doucet, A. (2020). Asymptotic properties of recursive particle maximum likelihood estimation. *IEEE Transactions on Information Theory*, 67(3) :1825–1848.

A Preuve de la Proposition 1

En partant de la définition de l'ELBO, on a que :

$$\begin{aligned}
\mathcal{L}_t^\lambda &= \mathbb{E}_{q_{0:t}^\lambda}\left[\log \frac{p_{0:t}(X_{0:t}, Y_{0:t})}{q_{0:t}^\lambda(X_{0:t})}\right] \\
&= \mathbb{E}_{q_{0:t}^\lambda}\left[\sum_{s=0}^{t} \tilde{f}_s^\lambda(X_{s-1}, X_s)\right] - \mathbb{E}_{q_t^\lambda}\left[\log q_t^\lambda(X_t)\right] \\
&= \int \left\{\sum_{s=0}^{t} \tilde{f}_s^\lambda(x_{s-1}, x_s)\right\} q_t^\lambda(x_t) \prod_{s=1}^{t} q_{s-1|s}^\lambda(x_s, x_{s-1}) \mathrm{d}x_{1:t} - \mathbb{E}_{q_t^\lambda}\left[\log q_t^\lambda(X_t)\right] \\
&= \mathbb{E}_{q_t^\lambda}\left[H_t^\lambda(X_t)\right] - \mathbb{E}_{q_t^\lambda}\left[\log q_t^\lambda(X_t)\right] \ ,
\end{aligned}$$

où

$$H_t^\lambda(x_t) := \int \left\{\sum_{s=0}^{t} \tilde{f}_s^\lambda(x_{s-1}, x_s)\right\} \prod_{s=1}^{t} q_{s-1|s}^\lambda(x_s, x_{s-1}) \mathrm{d}x_{1:t-1} = \mathbb{E}_{q_{0:(t-1)|t}^\lambda}\left[f_{0:t}^\lambda(X_{0:t-1}, x_t)\right] \ ,$$

où

$$q_{0:(t-1)|t}^\lambda(x_{0:t-1}, x_t) = \prod_{s=1}^{t} q_{s-1|s}^\lambda(x_s, x_{s-1})$$

est la densité de $X_{0:t-1}$ conditonnellement à $X_t = x_t$. On remarque ensuite que

$$\begin{aligned}
H_t^\lambda(x_t) &= \int \left(\int \left\{ \sum_{s=0}^{t} \tilde{f}_s^\lambda(x_{s-1}, x_s) \right\} \prod_{s=1}^{t-1} q_{s-1|s}^\lambda(x_s, x_{s-1}) \mathrm{d}x_{1:t-2} \right) q_{t-1|t}^\lambda(x_t, x_{t-1}) \mathrm{d}x_{t-1} \\
&= \int \left(H_{t-1}^\lambda(x_{t-1}) + \tilde{f}_t^\lambda(x_{t-1}, x_t) \right) q_{t-1|t}^\lambda(x_t, x_{t-1}) \mathrm{d}x_{t-1} \\
&= \mathbb{E}_{q_{t-1|t}^\lambda} \left[H_{t-1}^\lambda(X_{t-1}) + \tilde{f}_t^\lambda(X_{t-1}, X_t) \right] .
\end{aligned}$$

Les égalités (6) et (8) sont donc établies. Considérons désormais le gradient :

$$\begin{aligned}
\nabla \mathcal{L}_t^\lambda =& \nabla \mathbb{E}_{q_t^\lambda} \left[H_t^\lambda(X_t) \right] - \nabla \mathbb{E}_{q_t^\lambda} \left[\log q_t^\lambda(X_t) \right] \\
=& \mathbb{E}_{q_t^\lambda} \left[\nabla H_t^\lambda(X_t) \right] - \mathbb{E}_{q_t^\lambda} \left[\nabla \underbrace{\log q_t^\lambda(X_t)}_{=0} \right] + \\
& \int \left(H_t^\lambda(x_t) - \log q_t^\lambda(x_t) \right) \nabla \left(\log q_t^\lambda(x_t) \right) q_t^\lambda(x_t) \mathrm{d}x_t \\
=& \mathbb{E}_{q_t^\lambda} \left[\nabla H_t^\lambda(X_t) + \left(H_t^\lambda(x_t) - \log q_t^\lambda(x_t) \right) \times \nabla \log q_t^\lambda(x_t) \right] ,
\end{aligned}$$

où $\mathbb{E}_{q_t^\lambda} \left[\nabla \log q_t^\lambda(X_t) \right] = 0$ car $\mathbb{E}_{q_t^\lambda} \left[\nabla \log q_t^\lambda(X_t) \right] = \int \nabla q_t^\lambda(x_t) \mathrm{d}x_t = 0$. En notant $G_t^\lambda(x_t) = \nabla H_t^\lambda(x_t)$, on retrouve don l'égalité (7). On a de plus que :

$$\begin{aligned}
G_t^\lambda(x_t) &= \nabla \mathbb{E}_{q_{0:(t-1)|t}^\lambda} \left[f_{0:t}^\lambda(X_{0:t-1}, x_t) \right] \\
&= \mathbb{E}_{q_{0:(t-1)|t}^\lambda} \left[\left\{ \nabla \log q_{0:(t-1)|t}^\lambda \times f_{0:t}^\lambda \right\} (X_{0:t-1}, x_t) \right] + \mathbb{E}_{q_{0:(t-1)|t}^\lambda} \left[\nabla f_{0:t}^\lambda(X_{0:t-1}, x_t) \right] .
\end{aligned}$$

Or ici, on remarque [5] que, par definition de $f_{0:t}^\lambda$:

$$\nabla f_{0:t}^\lambda(X_{0:t-1}, x_t) = -\nabla \log q_{0:(t-1)|t}^\lambda(X_{0:t-1}, x_t),$$

donc, l'espérance de ce terme est égale à 0 et :

$$G_t^\lambda(x_t) = \mathbb{E}_{q_{0:(t-1)|t}^\lambda} \left[\left\{ \nabla \log q_{0:(t-1)|t}^\lambda \times f_{0:t}^\lambda \right\} (X_{0:t-1}, x_t) \right] .$$

En développant la fonction dans l'espérance, on aboutit à :

$$\begin{aligned}
G_t^\lambda(x_t) =& \mathbb{E}_{q_{t-1|t}^\lambda} \left[G_{t-1}^\lambda(X_{t-1}) \right] \\
& + \mathbb{E}_{q_{t-1|t}^\lambda} \left[\nabla \log q_{t-1|t}^\lambda(X_{t-1}, x_t) \left(\mathbb{E}_{q_{0:(t-2)|t-1}^\lambda} \left[f_{0:t-1}^\lambda(X_{0:t-1}) \right] + \tilde{f}_t^\lambda(X_{t-1}, x_t) \right) \right] \\
& + \mathbb{E}_{q_{t-1|t}^\lambda} \left[\tilde{f}_t^\lambda(X_{t-1}, x_t) \times \mathbb{E}_{q_{0:(t-2)|t-1}^\lambda} \left[\nabla \log q_{0:(t-2)|t-1}^\lambda(X_{0:t-1}) \right] \right] .
\end{aligned}$$

Dans l'espérance de la deuxième ligne, on reconnait H_{t-1}^λ et la troisème ligne est encore égale à 0. Ce qui montre (9).

5. Ceci est spécifique à l'ELBO, car le numérateur de dépend pas de λ