Remarks on filtering of semi-markov data

Bernard Delyon

INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Remarks on filtering of semi-markov data

Bernard Delyon

N° 1986

Septembre 1993

PROGRAMME 5

Traitement du signal,
automatique
et productique

INRIA
RENNES

Remarks on filtering of semi-markov data

Bernard Delyon*

Programme 5 — Traitement du signal, automatique et productique
Projet AS

Rapport de recherche n°1986 — Septembre 1993 — 7 pages

Abstract: This paper tries to give some insight about relationships between Viterbi and Forward-backward algorithm (used in the context of Hidden Markov Models) on one hand and Kalman filtering and Rauch-Tung-Striebel smoothing on the other. We give an unifying view which shows how those algorithms are related and give an example of an hybrid system which can be filtered through a mixed algorithm.

Key-words: filtering, smoothing, Viterbi, Kalman

(Résumé : tsvp)

*delyon@irisa.fr

Unité de recherche INRIA Rennes
IRISA, Campus universitaire de Beaulieu, 35042 RENNES Cedex (France)
Téléphone : (33) 99 84 71 00 – Télécopie : (33) 99 38 38 32

Remarques sur le filtrage de données semi-markoviennes

Résumé : Le but de cette note est la mise au clair des relations existant entre d'une part les algorithmes de Viterbi et Baum-Welch (utilisés dans le cadre des modèles à source de markovienne cachée) et d'autre part ceux de Kalman (filtrage) et de Rauch-Tung-Striebel (lissage). On donne à la fin une classe de processus semi-markoviens avec un espace d'etats hybride où vivent à la fois des variables discrètes et continues; un algorithme de filtrage et de lissage optimal est donné. On résoud également de manière exacte les équations de filtrage et lissage pour une certaine classe de problèmes non-linéaires.

Mots-clé : filtrage, lissage, Viterbi, Kalman

1 Introduction

In this paper, we will consider estimation of the state of semi-Markov processes. These processes arise in two quite different fields: Hidden Markov Models (widely used for speech recognition ([6])) and Kalman-Bucy filtering; in the first case, the state-space is discrete (generally finite) while in the second one it is the Euclidian space. However, algorithms which are used have considerable similarities. Inspection of these similarities will lead us first to a generalization of Kalman-Bucy filtering in a particular extention to non-linear systems and secondly to extend this model to a state-space which is mixed continuous-discrete.

Semi-Markov processes have the state-space representation (X_n, Y_n) (in some measurable space $\mathcal{X} \times \mathcal{Y}$) where Y is the observation and X is the hidden state; they are determined by two functions Π and Ψ:

- X_n is a Markov chain with transition probability $\Pi(x, x')$ (or $\Pi(x, x')dx'$ for continuous state space); Π may depend on n.
- Y_n a random variable whose distribution depends only on X_n; that is:

  $$P(Y_n = y | X_0^N, Y_1^{n-1}, Y_{n+1}^N) = P(Y_n = y | X_n) = \Psi(X_n, y)$$

  in discrete case, or

  $$P(Y_n \in dy | X_0^N, Y_1^{n-1}, Y_{n+1}^N) = P(Y_n \in dy | X_n) = \Psi(X_n, y)dy$$

  in continuous case

(We put $X_p^q = (X_p, X_{p+1}, ..., X_q)$ and $Y^n = Y_1^n$). An initial distribution is also given for X_0. The distribution of Y_n may depend on (X_{n-1}, X_n) (i.e. on the transition) without serious change in the theory.

Two problems are traditionnally addressed:

- Smoothing: what can be said of the sequence $X_0, X_2, ..., X_N$ once we are given an observation set $Y_1, Y_2, ..., Y_N$?
- Filtering: how to estimate recursively X_q from the observation of Y_q?

In any case, the problem is strongly connected to the maximization over $X_0, X_2, ..., X_n$ of the log-likelihood functional with has the form

$$\mathcal{L}(X_0, X_1, ..., X_n) = f(X_0, X_1, Y_1) + f(X_1, X_2, Y_2) + ... + f(X_{n-1}, X_n, Y_n) \tag{1}$$

Usual filtering algorithms (Kalman-Bucy, Rauch-Tung-Striebel, Viterbi...) consists only in fast exact maximization of such a functionnal by taking maximum advantage of this particular form. For more general case (non-linear...), one could as well be interested on approximate solution of this equation by fast numerical methods (which are frequently very efficient); this last aspect does not seem to have been really explored. In this paper, we will try to extend as far as possible exact maximization to some non-linear models.

Kalman-Bucy filtering is used when the model is linear gaussian, typically

$$\begin{aligned} X_n &= FX_{n-1} + w_{n-1}\ , \qquad w_n \simeq \mathcal{N}(0, Q) \\ Y_n &= HX_n + v_n\ , \qquad v_n \simeq \mathcal{N}(0, R) \end{aligned} \tag{2}$$

(for sake of simplicity, we assume that matrices F, H, Q, R are not time-dependent and v and w are two independent sequences of independent variables). We have

$$\begin{aligned}\Pi(x,x') &= \exp(-(x'-Fx)^T Q^{-1}(x'-Fx)/2)\frac{1}{det(2\pi Q)^{1/2}}\\ \Psi(x,y) &= \exp(-(y-Hx)^T R^{-1}(y-Hx)/2)\frac{1}{det(2\pi R)^{1/2}}\end{aligned}$$

In the case of non-linear filtering with discrete state space (Hidden Markov Models), transition probabilities are given by a matrix, and observation probabilities constitute a set of probability measures indexed by the states:

$$\begin{aligned}\Pi(X_n = j | X_{n-1} = i) &= \Pi_{ij}\\ \Psi(X_n, y) &= \Psi_j(y) \qquad \text{if } X_n = j\end{aligned}$$

We will show how the solutions for those two cases are related and, in last section, we give an example where the state space is a mixed discrete and continuous space.

2 Forward-backward and Viterbi algorithms

We compare here two algorithms: the first estimates the present state X_n by maximzing it probability conditionally to the observations, while the second maximizes the probability of the whole trajectory $X_0, X_1, ..X_n$ conditionally to the observations. In the case of filtering, the observations considered are $Y_1, Y_2, ..Y_n$ whilst in the case of smoothing it is $Y_1, Y_1, ..Y_N$, $N > n$.

2.1 Forward-backward algorithm

This algorithm is designed for recursive estimation of the probability for the current state X_n to be x once we have observed $Y_1, ..Y_n$; that is $P(X_n = x|Y^n)$; it is actually simpler to calculate the unnormalized probability $\alpha_n(x) = P(X_n = x, Y^n)$; using Markov property and Bayes formula we obtain

$$\begin{aligned}\alpha_n(x) &= P(Y_n|X_n = x, Y^{n-1})P(X_n = x, Y^{n-1})\\ &= \Psi(x, Y_n)\sum_u P(X_n = x|X_{n-1} = u, Y^{n-1})P(X_{n-1} = u, Y^{n-1})\\ &= \Psi(x, Y_n)\sum_u \alpha_{n-1}(u)\Pi(u, x). \qquad (3)\end{aligned}$$

Using this formula we can estimate recursively at each time n $\alpha_n(x)$ for all values of x.

In the same way, in continuous state space, we obtain for the unnormalized probability density the equation

$$\alpha_n(x) = \Psi(x, Y_n)\int \alpha_{n-1}(u)\Pi(u, x)du.$$

In the case of Kalman-Bucy filtering, $\alpha_n(x)$ is a Gaussian density and last formula leads directly to Kalman-Bucy filter equation, expressing the reestimation of mean and variance. In the same way, after time N, we can compute recursively the backward variables $\beta_n(x) = P(Y_{n+1}^N|X_n = x)$ with

$$\beta_n(x) = \Psi(x, Y_n)\sum_u \beta_{n+1}(u)\Pi(x, u).$$

The estimated filtered state at time n will be

$$\hat{x}_n = arg \max \alpha_n(x).$$

Markov property (independence of the past and future conditionally to the present) implies that $P(X_n = x, Y^N) = \alpha_n(x)\beta_n(x)$ and the estimated smoothed state at time n will be

$$x_n^{\#} = arg \max \alpha_n(x)\beta_n(x).$$

2.2 Viterbi algorithm

Viterbi algorithm is designed for recursive estimation, for each state x_n, of the most likely path ending at this state, say $\mathcal{C}(x_n) = (x_0(x_n), ...x_{n-1}(x_n), x_n)$. Like before we have

$$\begin{aligned}
\phi_n(x_n) &\stackrel{\Delta}{=} P(\mathcal{C}(x_n), Y^n) \\
&= \sup_{x_0,..x_{n-1}} P(x_0, ..x_n, Y^n) \\
&= \sup_{x_0,..x_{n-1}} P(Y_n, x_n | x_0, ..x_{n-1}, Y^{n-1}) P(x_0, ..x_{n-1}, Y^{n-1}) \\
&= \sup_{x_0,..x_{n-1}} P(Y_n, x_n | x_{n-1}) P(x_0, ..x_{n-1}, Y^{n-1}) \\
&= \sup_{x_{n-1}} P(Y_n, x_n | x_{n-1}) \phi_{n-1}(x_{n-1}). \qquad (4)
\end{aligned}$$

And we have

$$P(Y_n, x_n | x_{n-1}) = \Pi(x_{n-1}, x_n)\Psi(x_n, Y_n).$$

At the same time we memorize the function

$$\xi_n(x_{n+1}) = arg \sup_{x_n} \Pi(x_n, x_{n+1})\phi_n(x_n).$$

When the state space is discrete, this function is a state pointer; is the case of Gaussian linear smoothing it will be a linear function (see below). The filtered estimate at time n knowing Y^n is

$$\check{x}_n = arg \max_x \phi_n(x).$$

Smoothed estimates over the interval $[0, N]$ are given by the equations

$$x_N^* = \check{x}_N \qquad (5)$$

$$x_{n-1}^* = \xi_{n-1}(x_n^*) \qquad (6)$$

Those equation are well known in dynamic programming (cf [1]).

2.3 Comments

Forward-filtered estimate $\hat{x}_n$ maximizes the a posteriori probability $P(X_n = x|Y^n)$ with respect to x, while Viterbi-filtered estimate $\check{x}_n$ maximizes $\phi_n(x)$ which is the a posteriori probability of the whole path; those two estimates are generally different. However, as we shall see in next section, they are identical in the case of Gaussian linear filtering. Next sections we will consider smoothing only under Viterbi aspects for the following reason: nothing guarantees that the sequence $x_n^{\#} = arg \max_x P(X_n = x|Y^N)$, $n = 0, ..N$, has a non-zero probability for the Markov chain; this sequence fondamentally different from (x_n^*). However, in the case of Gaussian linear smoothing, those two sequences are identical.

3 Fast filtering in continuous state space

In the case of continuous state spaces, the problem in the application of previous formulas is that we have to memorize functions ϕ_n and ξ_n, which is impossible unless those functions are parametrized by a vector, say $\theta \in \mathbf{R}^d$. This is what happens in Kalman-Bucy (filtering) and Rauch-Tung-Striebel (smoothing) algorithms where the these functions are Gaussian densities.

We explore here a more general setting where ϕ_n and ξ_n can still be parametrized. We study only the stationary case (i.e. corresponding in the Kalman-Bucy context to the case where the variance of the first state is such that no matrix has to be reestimated during the algorithm). The assumption on the model is constituted by equation (7):

Theorem 1 *We assume that the transition and observation probability may be expressed as*

$$\log P(X_n = x', Y_n = y' | X_{n-1} = x) = -U(x - Ax') - V(x') + V(x) + \theta(y')^T x' - Z(y') \quad (7)$$

where U and V are convex fonctions and θ and Z are arbitrary fonctions. We assume that the a priori probability for X_0 is proportional to $\exp(-V(x_0) + \theta_0^T x_0)$; in that case, functions ϕ_n and ξ_n may be parametrized with a sequence θ_n and filtering and smoothing equations are

$$\theta_n = A^T \theta_{n-1} + \theta(Y_n) \quad (8)$$

$$\xi_n(x) = Ax + \nabla g(\theta_n) \quad (9)$$

$$\check{x}_n = \nabla h(\theta_n) \quad (10)$$

where g and h are Legendre transforms of fonctions U et V:

$$g(\theta) = \sup_x \theta^T x - U(x) \quad (11)$$

$$h(\theta) = \sup_x \theta^T x - V(x). \quad (12)$$

and equations (5) are used for smoothing.

Proof The proof is elementary if one uses equation (4); it consists in verifying by induction that

$$\log(\phi_n(x)) = \theta_n^T x - V(x) + \sum_{i=1}^{n} g(\theta_{i-1}) - Z(Y_i)$$

Equations (10) and (9) come from the fact that the x which realizes the supremum of equation (11) (resp. (12)) is $\nabla g(\theta)$ (resp. $\nabla h(\theta)$). ∎

The correspondence with Kalman-Bucy filtering and Rauch-Tung-Striebel smoothing (with the notations of equations (3)) is

$$\begin{aligned}
A &= P_+ F^T P_-^{-1} \\
U(x) &= x^T (F^T Q^{-1} F + P_+^{-1}) x/2 \\
V(x) &= x^T P_+^{-1} x/2 \\
\theta(y) &= H^T R^{-1} y \\
Z(y) &= y^T R^{-1} y/2 + const \\
\theta_n &= P_+^{-1} \hat{x}_n^+ \\
A^T \theta_n &= P_-^{-1} \hat{x}_{n+1}^-
\end{aligned}$$

where P_+ and P_- are variances of prediction errors of $\hat{x}_n^+ = E[x_n|Y^n] = \check{x}_n$ and $\hat{x}_n^- = E[x_n|Y^{n-1}] = F\check{x}_{n-1}$ ($P_- = FP_+F^T + Q$ and $P_+^{-1} = P_-^{-1} + H^T R^{-1} H$, cf [3] table 4.2.1 and 5.2.2). In order to verify this, observe that the dynamic of the linear model is

$$\begin{aligned}\log P(x', y'|x) &= -(x' - Fx)^T Q^{-1}(x' - Fx)/2 - (y' - Hx')^T R^{-1}(y' - Hx')/2 + const \\ &= -x^T F^T Q^{-1} Fx/2 + x^T F^T Q^{-1} x' - x'^T (Q^{-1} + H^T R^{-1} H) x'/2 \\ &\quad -y'^T R^{-1} y'/2 - y'^T R^{-1} H x' + const\end{aligned}$$

and the choice of (U, V, A, Z) above leads to

$$\begin{aligned}-U(x - Ax') - V(x') &+ V(x) + \theta(y')^T x' - Z(y') \\ &= -(x - P_+ F^T P_-^{-1} x')^T (F^T Q^{-1} F + P_+^{-1})(x - P_+ F^T P_-^{-1} x')/2 \\ &\quad -x'^T P_+^{-1} x'/2 + x^T P_+^{-1} x/2 + x'^T H^T R^{-1} y' - y'^T R^{-1} y'/2 + const \\ &= -x^T F^T Q^{-1} Fx/2 + x^T (F^T Q^{-1} F + P_+^{-1}) P_+ F^T P_-^{-1} x' \\ &\quad -x'^T (P_-^{-1} F P_+ F^T Q^{-1} F P_+ F^T P_-^{-1} + P_-^{-1} F P_+ F^T P_-^{-1} + P_+^{-1}) x'/2 \\ &\quad +x'^T H^T R^{-1} y' - y'^T R^{-1} y'/2 + const \\ &= -x^T F^T Q^{-1} Fx/2 + x^T (F^T Q^{-1} (P_- - Q) P_-^{-1} + F^T P_-^{-1}) x' \\ &\quad -x'^T (P_-^{-1} (P_- - Q) Q^{-1} (P_- - Q) P_-^{-1} + P_-^{-1} (P_- - Q) P_-^{-1} + P_+^{-1}) x'/2 \\ &\quad +x'^T H^T R^{-1} y' - y'^T R^{-1} y'/2 + const\end{aligned}$$

which is the same.

When dealing with non-linear cases, the most interesting situation seems to be the following: We are given (A, U, θ, Z), that is the probability given the past and the future

$$\log P(x', y'|x, x'') \simeq -U(x - Ax') - U(x' - Ax'') + \theta(y')^T x' - Z(y')$$

(the Markov chain is considered as a reciprocal process) and V has to be computed; for smoothing purpose, an approximation will be enough because this function is utilized only for the obtention of the filtered estimate of the state at last time N. In other words, a replacement of V by $\tilde{V}$ has the same effect as remplacing the likelihood of the sequence $(x_0, ...x_N)$ by

$$P(x_0, ..x_N) e^{V(x_N) - \tilde{V}(x_N)}$$

which has a small influence on the estimates except close to the end.

4 An hybrid model

The state is now represented by a vector and an integer (x, e). As we shall see below, the model considered here is interesting if we are given the distribution of the observation and the state at time n conditionally to the state at times $n-1$ and $n+1$ (the Markov chain is considered as a reciprocal process (cf [4],[2])), and if we want to smooth data.

Theorem 2 *We assume that the transition and observation probability may be expressed as*

$$\log P(x', e', y'|x, e) = -U_{ee'}(x - Ax') - V_{e'}(x') + V_e(x) + \theta(y')^T x' - Z_{e'}(y') \tag{13}$$

where the functions $U_{ee'}$ and V_e are convex and the functions θ_e are arbitrary. We assume that the a priori probability of (x_0, e_0) is proportional to $p_0(e_0)\exp(-V(x_0)+\theta_0^T x_0)$; filtering and smoothing may be performed by estimating the continuous and discrete states through the equations

$$\begin{aligned}
\eta_0 &= \log(p_0(e)) \\
\theta_n &= A^T\theta_{n-1} + \theta(Y_n) \\
\eta_n(e) &= \eta_{n-1}(\epsilon_{n-1}(e)) + g_{\epsilon_{n-1}(e)e}(\theta_{n-1}) - Z_e(Y_n) \\
\epsilon_n(e) &= arg\max_{e_n} \eta_n(e_n) + g_{e_n,e}(\theta_n) \\
\xi_n(x,e) &= Ax + \nabla g_{\epsilon_n(e)e}(\theta_n) \\
\check{e}_n &= arg\max\{\eta_n(e) + h_e(\theta_n)\} \\
\check{x}_n &= \nabla h_{\check{e}_n}(\theta_n)
\end{aligned} \tag{14}$$

where $g_{ee'}$ and h_e are Legendre transforms of functions $U_{ee'}$ and V_e. $\epsilon_n(e)$ the most likely discrete state at time n knowing $e_{n+1} = e$ (it is independent of x_{n+1}).

Proof By using formula (4) one shows straightforwardly by induction that

$$\begin{aligned}
\log\phi_N(x_N, e_N) &= \max_{e_0,..e_{N-1}} \log(p_0(e_0)) + \sum_{n=1}^{N} g_{e_{n-1}e_n}(\theta_{n-1}) - Z_{e_n}(Y_n) \\
&\quad +\theta_N^T x_N - V_{e_N}(x_N) \\
&= \eta_N(e_N) + \theta_N^T x_N - V_{e_N}(x_N)
\end{aligned}$$

where η_n is given in the statement. Taking the supremum over x_N, we obtain $x_N^* = \check{x}_N = \nabla h_{e_N}(\theta_N)$ and the probability of the best path arriving at e_N at time N is

$$\log\phi_N(e_N) = \eta_N(e_N) + h_{e_N}(\theta_N)$$

The optimal sequence (e_n^*) which will maximize this quantity corresponds to the equations given in the statement. ∎

Comments:

- Storage requirements of this algorithm are still reduced: for smoothing it will be N state pointers (as in Viterbi algorithm) and N vectors.
- One has
$$\theta_n = \nabla U_{e_{n-1}^* e_n^*}(x_n^* - Ax_{n+1}^*)$$
where (x_n^*, e_n^*) is the optimal sequence. This can be checked instantaneously by differentiation of the global likelihood of the sequence (expressed in the form (1)) with respect to x_n: the three terms which appear lead to the relation (14).
- We can model a signal whose law is a mixture of linear models: each element of the mixture will be indexed by a pair (e, e') and $U_{ee'}(x)$ will be (with some abuse of notation)
$$U_{ee'}(x) = -\log(p(e,e')) + x^T U_{ee'} x$$

where $p(e,e')$ is the probability of the transition from e to e'. As before, functions $V_e(x)$ will be quadratic forms $x^T V_e x$ and $\theta(y)$ is a matrix product Θy. If we drop indices in the (e,e')-dependent linear model $(F_{ee'}, Q_{ee'}, H_{ee'}, R_{ee'})$, we have to identify the likelihood (13) to

$$-(x'-Fx)^T Q^{-1}(x'-Fx)/2 - (y'-Hx')^T R^{-1}(y'-Hx')/2$$

and this leads to

$$\begin{aligned} H^T R^{-1} H + Q^{-1} &= 2A^T U_{ee'} A + 2V_{e'} \\ F^T Q^{-1} F &= 2U_{ee'} - 2V_e \\ F^T Q^{-1} + Q^{-1} F &= U_{ee'} A + A^T U_{ee'} \\ R^{-1} H &= \Theta. \end{aligned}$$

With those notations the process may be described this way: starting with a discrete state (e,x), the process jumps to another one e' with probability $p(e,e')$, and a new state x' is chosen with the dynamics of $(F_{ee'}, Q_{ee'})$ and an observation y' is then produced with x' and $(H_{ee'}, R_{ee'})$ (cf eqs (2)).

Setting $e = e'$ in the equations above (steady state), we see the principal restriction of this model: the matrices $R^{-1}H$ $(= \Theta)$ and $P_+ F^T P_-^{-1}$ $(= A)$ are independent of e.

- Functions V_e have weak importance for smoothing (with an infinite number of observations, the smoothed estimate of X_n is independent of functions V_e).
- Reciprocal point of view: We are given $(A, U_{ee'}, \theta, Z_e)$, that is the probability given the past and the future; it will have the form

  $$\log P(e',x',y'|e,x,e'',x'') \propto -U_{ee'}(x - Ax') - U_{e'e''}(x' - Ax'') + \theta(y')^T x' + Z_{e'}(y')$$

  (the Markov chain is considered as a reciprocal process) V_e has to be computed in such a way that the formula for $P(e',x',y'|e,x)$ given in the theorem is a probability (i.e. $\exp(-V)$ should be an eigenvector of the operator $f \to \sum_{e'} \int f(e',x') \exp\{-U_{ee'}(x - Ax') + \theta(y')^T x' - Z_{e'}(y')\} dx'dy'$). As we said before, for smoothing purposes, a reasonable approximation is enough.

References

[1] G.D.Forney, *The Viterbi Algorithm,* Proc. IEEE, vol 61, No 3, March 1973.

[2] C.D.Greene,B.C.Levy *Some new smoother implimentation for discrete-time gaussian reciprocal processes*, Int. J. Control, 1991, vol 54, No 5, pp 1223-1247.

[3] A.Gelb, *Applied Optimal Estimation*, M.I.T. press, 1989.

[4] B.Jamison, *Reciprocal processes,* Zeit. Wahrsch., vol 30, pp 65-86, 1974.

[5] T.Kailath, *A View of Three Decades of Linear Filtering Theory*, IEEE-IT, vol 20, No 2, March 1974.

[6] L.R.Rabiner, B.H.Juang *Introduction to Hidden Markov Models,* IEEE-ASSP Magazine, January 1986.

Unité de recherche INRIA Lorraine, Technôpole de Nancy-Brabois, Campus scientifique,
615 rue de Jardin Botanique, BP 101, 54600 VILLERS LÈS NANCY
Unité de recherche INRIA Rennes, IRISA, Campus universitaire de Beaulieu, 35042 RENNES Cedex
Unité de recherche INRIA Rhône-Alpes, 46 avenue Félix Viallet, 38031 GRENOBLE Cedex 1
Unité de recherche INRIA Rocquencourt, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex
Unité de recherche INRIA Sophia-Antipolis, 2004 route des Lucioles, BP 93, 06902 SOPHIA-ANTIPOLIS Cedex

Éditeur
INRIA, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex (France)
ISSN 0249-6399