Reinforcement learning, particle filters and the EM algorithm

Vivek S. Borkar
Department of Electrical Engineering,
Indian Institute of Technology,
Powai, Mumbai 400076, India.
Email: borkar.vs@gmail.com

Ankushkumar Jain
Department of Electrical Engineering,
Indian Institute of Technology,
Powai, Mumbai 400076, India.
Email: jainankush@iitb.ac.in

***Abstract*—We consider a parameter estimation problem for a Hidden Markov Model in the framework of particle filters. Using constructs from reinforcement learning for variance reduction in particle filters, a simulation based scheme is developed for estimating the partially observed log-likelihood function. A Kiefer-Wolfowitz like stochastic approximation scheme maximizes this function over the unknown parameter. The two procedures are performed on two different time scales, emulating the alternating 'expectation' and 'maximization' operations of the EM algorithm. Numerical experiments are presented in support of the proposed scheme.**

I. Introduction

We consider a partially observed Markov chain on a finite state space, in other words a Hidden Markov Model (HMM for short), whose transition probabilities depend on an unknown parameter. The problem we address is to estimate this parameter given a trace of observations. The standard procedure for this is the 'Expectation-Maximization' or 'EM' algorithm of Dempster et al [10] wherein one alternates between an 'Expectation' (or 'E') step that computes the conditional expectation of the log-likelihood function given the current parameter estimate, and the 'Maximization' (or 'M') step that performs a maximization of the said expectation with respect to the unknown parameter in order to update the estimate. In our case, we are looking at a stochastic dynamical system, which requires us to develop an appropriate nonlinear filter (smoother, to be precise) for evaluating the conditional behavior of state given observations. Such filters are usually computationally cumbersome. This has prompted a large body of research on particle filters, which employ Markov Chain Monte Carlo (MCMC) for a simulation based methodology to estimate the conditional averages using the strong law of large numbers [2]. This is our starting point. In a recent work [8], we introduced ideas from reinforcement learning (see [3], Chapter 16, for background) in order to reduce the problematic high variance in particle filters. In this article, we introduce a suitable modification of the scheme proposed in [8] for estimating the partial log-likelihood function for the HMM. A Kiefer-Wolfowitz like scheme is then used in order to perform the maximization of this over the parameter space. This executes an approximate gradient ascent using a finite difference approximation of the gradient. The two procedures are supposed to alternate in the classical EM algorithm. We perform them concurrently albeit on different time scales, exploiting the theory of 'two time scale stochastic approximation' ([6], Chapter 6) to achieve the same effect.

The paper is organized as follows. The next section describes the problem set up and the algorithm. Section 3 presents supporting numerical experiments for a simple example. Section 4 concludes with a discussion of further possibilities.

We conclude this section by recalling from [8] the motivation for this approach. Particle filters inherit the usual issues concerning MCMC, such as possibly high variance. Recently reinforcement learning has been used as an alternative to pure MCMC, the idea being to combine MCMC with classical numerical schemes so as to retain some advantages of both [7]. As argued in *ibid.*, being 'incremental', it inherits from MCMC lower per iterate computation and memory requirement than the deterministic numerical schemes. From numerical schemes, it inherits a lower variance than pure MCMC. The latter, as pointed out in *ibid.*, is because the learning schemes, through 'one step' analysis, estimate conditional expectations rather than expectations, which can be thought of as a Rao-Blackwellization of the particle filter. We consider an importance sampling version thereof as in [1], where it offers the added advantage that it involves only a one step likelihood ratio per iterate rather then the full likelihood ratio, which has a much higher variance.

II. The EM particle filter

Consider the HMM given by the (state, observation) processes $(X_n, Y_n), n \geq 0$, taking values in a finite product space $\mathcal{S} \times \mathcal{O}$, such that

$$P(X_{n+1} = i, Y_{n+1} = j | X_m, Y_m, m \leq n) = p_{\theta^*}(i, j | X_n)$$

for a prescribed transition probability function $p_{\theta^*}(\cdot, \cdot|\cdot)$. The latter is known to belong to a parametrized family $\{p_\theta(\cdot, \cdot|\cdot) : \theta \in \Theta\}$ for a compact parameter set Θ that contains the true parameter θ^*. For simplicity of notation, we take Θ to be a closed bounded interval in $\mathcal{R}$, though what follows extends easily to parameter sets in $\mathcal{R}^d, d > 1$. Let $\mathcal{F}_n := \sigma(Y_i, i \leq$

$n), n \geq 0$. We pick a $\hat{\theta}_0$ as our initial guess for θ^*. The EM algorithm in our context is as follows.

1) (*E step:*) Estimate

$$\Lambda(\theta, \theta') := E_\theta[\sum_{m=0}^{N-1} \log p_{\theta'}(X_{m+1}, Y_{m+1}|X_m)|\mathcal{F}_N] \tag{1}$$

for $\theta = \hat{\theta}_n$ based on observations $Y_m, 1 \leq m \leq N$.

2) (*M step:*) Find $\hat{\theta}_{n+1} := \text{argmax}\Big(\Lambda(\hat{\theta}_n, \cdot)\Big)$,

till convergence.

We begin with a stochastic approximation scheme for the first, or 'E', step above. For this, a 'stochastic approximation' or 'reinforcement learning' based particle filter/smoother in the spirit of [8] is given as follows. We fix a realization $Y_m = y_m, 1 \leq m \leq N$, of the observation process. This features as a fixed parameter in the scheme below. Let $Q := [[q(j|i)]]_{i,j \in \mathcal{S}}$ be an irreducible stochastic matrix such that $\max_y p(j, y|i) > 0 \Longrightarrow q(j|i) > 0$. Consider a Markov chain $\{\tilde{X}_n\}$ with transition matrix Q and initial distribution π_0. The law of this chain will be our 'importance sampling' measure. Simulate independent runs $\{\tilde{X}^k_m, 0 \leq m \leq N\}, k \geq 1$, of the chain with initial distribution π_0 ('particles'). The reinforcement learning particle filter/smoother is: for $k = 1, 2, \cdots$, do

1) *STEP 1:* This step evaluates the common normalizing factor.

$$\begin{aligned} \hat{V}^{k+1}_n(i) &= \Big(1 - a(k)I\{\tilde{X}^{k+1}_n = i\}\Big)\hat{V}^k_n(i) + a(k) \\ &\quad \times I\{\tilde{X}^{k+1}_n = i\}\left(\frac{p_{\hat{\theta}_k}(\tilde{X}^{k+1}_{n+1}, y_{n+1}|i)}{q(\tilde{X}^{k+1}_{n+1}|i)}\right) \\ &\quad \times \left(\hat{V}^{k+1}_{n+1}(\tilde{X}^{k+1}_{n+1})\right), \ n < N, \end{aligned} \tag{2}$$

with terminal condition $\hat{V}^{k+1}_N(i) = 1$.

2) *STEP 2:* This step evaluates unnormalized conditional expectation of the log-likelihood at parameter $\theta =$ the current estimate of θ^*.

$$\begin{aligned} &\tilde{V}^{k+1}_n(i) \\ &= \Big(1 - a(k)I\{\tilde{X}^{k+1}_n = i\}\Big)\tilde{V}^k_n(i) + a(k) \\ &\quad \times I\{\tilde{X}^{k+1}_n = i\}\left(\frac{p_{\hat{\theta}_k}(\tilde{X}^{k+1}_{n+1}, y_{n+1}|i)}{q(\tilde{X}^{k+1}_{n+1}|i)}\right) \\ &\quad \times \Big(\log p_{\hat{\theta}_k}(\tilde{X}^{k+1}_{n+1}, y_{n+1}|i)\hat{V}^k_{n+1}(\tilde{X}^{k+1}_{n+1}) \\ &\quad + \tilde{V}^{k+1}_{n+1}(\tilde{X}^{k+1}_{n+1})\Big), \ n < N, \end{aligned} \tag{3}$$

with terminal condition $\tilde{V}^{k+1}_N(i) = 0$.

3) *STEP 3:* This step evaluates unnormalized conditional expectation of the log-likelihood at parameter $\theta =$ a perturbation of the current estimate of θ^*.

$$\begin{aligned} &\breve{V}^{k+1}_n(i) \\ &= \Big(1 - a(k)I\{\tilde{X}^{k+1}_n = i\}\Big)\breve{V}^k_n(i) + a(k) \\ &\quad \times I\{\breve{X}^{k+1}_n = i\}\left(\frac{p_{\hat{\theta}_k}(\tilde{X}^{k+1}_{n+1}, y_{n+1}|i)}{q(\tilde{X}^{k+1}_{n+1}|i)}\right) \\ &\quad \times \Big(\log p_{\hat{\theta}_k + \delta}(\tilde{X}^{k+1}_{n+1}, y_{n+1}|i)\hat{V}^k_{n+1}(\tilde{X}^{k+1}_{n+1}) \\ &\quad + \breve{V}^{k+1}_{n+1}(\tilde{X}^{k+1}_{n+1})\Big), \ n < N. \end{aligned} \tag{4}$$

with terminal condition $\breve{V}^{k+1}_N(i) = 0$.

Here $\{a(k)\}$ is a positive sequence of step-sizes satisfying the standard conditions: $\sum_k a(k) = \infty, \ \sum_k a(k)^2 < \infty$. The iteration (2) is nothing but a reinforcement learning scheme for 'policy evaluation' of the constant policy (here, uncontrolled) Markov chain $\{\tilde{X}_n\}$ with transition matrix Q and the 'finite horizon cost'

$$\prod_{m=1}^{N}\left(\frac{p_{\hat{\theta}_k}(\tilde{X}_{m+1}, y_{m+1}|\tilde{X}_m)}{q(\tilde{X}_{m+1}|\tilde{X}_m)}\right).$$

It is a stochastic approximation scheme for solving the corresponding 'dynamic programming equation' given by $V(\cdot, N) \equiv 1$ and for $m < N$,

$$\begin{aligned} \hat{V}(i, m) &= \sum_{j \in \mathcal{S}} q(j|i)\left(\frac{p_{\hat{\theta}_k}(j, y_{m+1}|i)}{q(j|i)}\right)\hat{V}(j, m+1) \\ &= \sum_{j \in \mathcal{S}} p_{\hat{\theta}_k}(j, y_{m+1}|i)\hat{V}(j, m+1). \end{aligned} \tag{5}$$

The second expression on the right of (5) shows that

$$\hat{V}(i, 0) = P(Y_m = y_m, \ 1 \leq m \leq N|X_0 = i)$$

for the Markov chain $\{X_m\}$ with associated observation process $\{Y_m\}$, governed by transition probabilities $p_{\hat{\theta}_m}(\cdot, \cdot|\cdot), 0 \leq m < N$. Note that $\sum_i \pi_0(i)\hat{V}(i, 0)$ is also the normalizing factor for the passage from normalized to unnormalized filter or smoother. On the other hand, (3) is a reinforcement learning scheme for evaluating

$$\begin{aligned} &E\Big[\sum_{m=0}^{N-1}\left(\frac{p_{\hat{\theta}_k}(\tilde{X}_{m+1}, y_{m+1}|\tilde{X}_m)}{q(\tilde{X}_{m+1}|\tilde{X}_m)}\right) \\ &\times \ \log p_{\hat{\theta}_k}(\tilde{X}_{m+1}, y_{m+1}|\tilde{X}_m)\hat{V}^k_{n+1}(\tilde{X}^{k+1}_{n+1})\Big] \end{aligned} \tag{6}$$

for a Markov chain $\{\tilde{X}_m\}$ governed by the transition matrix Q. To see this, note that (6) equals $\sum_i \pi_0(i)\tilde{V}(i, 0)$, where $\tilde{V}(i, m), i \in \mathcal{S}, 0 \leq m \leq N$, is given by the 'dynamic programming equation'

$$
\begin{aligned}
&\tilde{V}(i,m)\\
&= \sum_j q(j|i)\left(\frac{p_{\hat{\theta}_k}(j,y_{m+1}|i)}{q(j|i)}\right)\log(p_{\hat{\theta}_k}(j,y_{m+1}|i))\\
&\quad \times \hat{V}^k_{n+1}(j) + \sum_j q(j|i)\left(\frac{p_{\hat{\theta}_k}(j,y_{m+1}|i)}{q(j|i)}\right)\tilde{V}(j,m+1)\\
&= \sum_j p_{\hat{\theta}_k}(j,y_{m+1}|i)\log(p_{\hat{\theta}_k}(j,y_{m+1}|i))\hat{V}^k_{n+1}(j)\\
&\quad + \sum_j p_{\hat{\theta}_k}(j,y_{m+1}|i)\tilde{V}(j,m+1).
\end{aligned}
$$

In view of the discussion in [8], section 3, bullet 2, the second equality shows that $\sum_i \pi_0(i)\tilde{V}(i,0)$ equals the finite horizon cost

$$
\begin{aligned}
&E\Big[\left(\sum_{m=0}^{N-1}\log p_{\hat{\theta}_k}(X_{m+1},y_{m+1}|X_m)\right)\\
&\times \quad I\{Y_m = y_m \ \forall \ m \le N\}\Big], \qquad (7)
\end{aligned}
$$

for the Markov chain $\{X_m\}$ with associated observation process $\{Y_m\}$, governed by transition probabilities $p_{\hat{\theta}_k}(\cdot,\cdot|\cdot)$. Then (3) is precisely the asynchronous stochastic approximation scheme to solve (6) or *ipso facto*, evaluate (7), in other words, a reinforcement learning scheme for this purpose.

Likewise, (4) is a reinforcement learning scheme for exactly the same computation but with the 'running cost' $\log p_{\hat{\theta}_k}(X_{m+1},y_{m+1}|X_m)$ in (7) replaced by $\log p_{\hat{\theta}_k+\delta}(X_{m+1},y_{m+1}|X_m)$.

For the 'M' step, let

$$
F_k := \frac{\pi_0\tilde{V}_0}{\pi_0\hat{V}_0},\ F'_k := \frac{\pi_0\breve{V}_0}{\pi_0\hat{V}_0}.
$$

Consider the approximate gradient ascent (*Kiefer-Wolfowitz* scheme)

$$
\hat{\theta}_{k+1} = \hat{\theta}_k + b(k)\left(\frac{F'_k - F_k}{\delta}\right),
$$

with $b(k) = o(a(k))$. The latter condition ensures that this is a two time scale stochastic approximation ([6], Chapter 6) and standard analysis of this class of schemes ensures that it tracks (within a certain approximation) the gradient ascent for conditional expectation of the log-likelihood given observations, as desired by the EM methodology.

III. Example

In this section we present a simple example as 'proof of concept'. More extensive examples are being planned and will be reported elsewhere.

Consider a queueing system

$$
Q_{n+1} = \Big(Q_n - D_n I\{X_n > 0\} + \xi_{n+1}\Big)\bigwedge 100. \qquad (8)
$$

Here, given $0 < b << a < 1$,

- D_n (the *departure* process) is i.i.d., $D_n = 1$ with probability $\mu_n \in \{a,b\}, n \ge 0$, and 0 otherwise,
- ξ_n (the *arrival* process) is i.i.d., $\xi_n = 1$ with probability $\lambda, b < \lambda < a$ and 0 otherwise,
- $\{\mu_n\}$ (the *service* rate) is an $\{a,b\}$-valued Markov chain with two states: '*working*' when $\mu_n = a$ and '*faulty*' where $\mu_n = b$.

Denote by $p_2(\cdot|\cdot)$ the transition probabilities of $\{\mu_n\}$. We have limited the maximum queue size to 100, i.e., assumed a finite buffer of size 100. In case of buffer overflow, the extra packets are assumed lost.

In this example the observed process is the queue length $\{Q_n\}$ and the state process is $\{\mu_n, Q_n\}$. Comparing with our earlier notation, the correspondence is:

$$
X_n \longleftrightarrow \{\mu_n, Q_n\},\ Y_n \longleftrightarrow Q_n.
$$

Thus $X_n \in \{1,....,100\} \times \{a,b\} := \mathcal{S}$ and $Y_n \in \{1,....,100\} := \mathcal{O}$. The transition probability function is:

$$
\begin{aligned}
&p((u',i'),\tilde{i}|(i,u))\\
&:= \quad P(X_{t+1} = (i',u'), Y_{n+1} = \tilde{i}|X_n = (u,i))\\
&= \quad p_1(i'|i,u)p_2(u'|u)\delta_{i'\tilde{i}},
\end{aligned}
$$

where, for $0 < i < 100$,

$$
\begin{aligned}
p_1(i'|i,u) &= \lambda(1-u), && \text{for } i' = i+1,\\
&= (1-\lambda)u, && \text{for } i' = i-1,\\
&= u\lambda + (1-u)(1-\lambda), && \text{for } i' = i,\\
&= 0, && \text{otherwise,}
\end{aligned}
$$

for $i = 0$,

$$
\begin{aligned}
p_1(i'|i,u) &= \lambda(1-u), && \text{for } i' = i+1,\\
&= u + (1-u)(1-\lambda), && \text{for } i' = i,\\
&= 0, && \text{otherwise,}
\end{aligned}
$$

and for $i = 100$,

$$
\begin{aligned}
p_1(i'|i,u) &= (1-\lambda)u, && \text{for } i' = i-1,\\
&= u\lambda + (1-u), && \text{for } i' = i,\\
&= 0, && \text{otherwise.}
\end{aligned}
$$

The nonlinear filter then turns out to be

$$
\nu_{n+1}(j,q) = \sum_{i=a,b}\nu_n(i,Q_n)p((j,Q_{n+1}),Q_{n+1}|i,Q_n) \qquad (9)
$$

when $q = Q_{n+1}$, otherwise $\nu_{n+1}(j,q) = 0$.

The algorithm becomes: (We have suppressed the dependence on queue length of $\hat{V}^k_n, \tilde{V}^k_n, \breve{V}^k_n$, because this component of the state is observed exactly and enters the computation only parametrically.)

1) *STEP 1:* $\hat{V}_N^{k+1}(u) = 1$ and for $n < N$,

$$\begin{aligned}
&\hat{V}_n^{k+1}(u) \\
&\quad = \Big(1 - a(k) I\{\tilde{\mu}_n^{k+1} = u\}\Big)\hat{V}_n^k(u) \\
&\qquad + a(k) I\{\tilde{\mu}_n^{k+1} = u\} \\
&\qquad \times \left(\frac{p_{\hat{\theta}_k}((\tilde{\mu}_{n+1}^{k+1}, Q_{n+1}), Q_{n+1}|u, Q_n)}{q(\tilde{\mu}_{n+1}^{k+1}|u)}\right) \\
&\qquad \times \left(\hat{V}_{n+1}^{k+1}(\tilde{\mu}_{n+1}^{k+1})\right). \qquad (10)
\end{aligned}$$

2) *STEP 2:* $\tilde{V}_N^{k+1}(u) = 0$ and for $n < N$,

$$\begin{aligned}
&\tilde{V}_n^{k+1}(u) \\
&\quad = \Big(1 - a(k) I\{\tilde{\mu}_n^{k+1} = u\}\Big)\tilde{V}_n^k(u) \\
&\qquad + a(k) I\{\tilde{\mu}_n^{k+1} = u\} \times \\
&\qquad \left(\frac{p_{\hat{\theta}_k}((\tilde{\mu}_{n+1}^{k+1}, Q_{n+1}), Q_{n+1}|u, Q_n)}{q(\tilde{\mu}_{n+1}^{k+1}|u)}\right) \times \\
&\qquad \Big(\log p_{\hat{\theta}_k}((\tilde{\mu}_{n+1}^{k+1}, Q_{n+1}), Q_{n+1}|u, Q_n) \\
&\qquad \times \hat{V}_{n+1}^k(\tilde{\mu}_{n+1}^{k+1}) + \tilde{V}_{n+1}^{k+1}(\tilde{\mu}_{n+1}^{k+1})\Big). \qquad (11)
\end{aligned}$$

3) *STEP 3:* $\breve{V}_N^{k+1}(u) = 0$ and for $n < N$,

$$\begin{aligned}
&\breve{V}_n^{k+1}(u) \\
&\quad = \Big(1 - a(k) I\{\tilde{\mu}_n^{k+1} = u\}\Big)\breve{V}_n^k(u) \\
&\qquad + a(k) I\{\tilde{\mu}_n^{k+1} = u\} \times \\
&\qquad \left(\frac{p_{\hat{\theta}_k}((\tilde{\mu}_{n+1}^{k+1}, Q_{n+1}), Q_{n+1}|u, Q_n)}{q(\tilde{\mu}_{n+1}^{k+1}|u)}\right) \times \\
&\qquad \Big(\log p_{\hat{\theta}_k+\delta}((\tilde{\mu}_{n+1}^{k+1}, Q_{n+1}), Q_{n+1}|u, Q_n) \\
&\qquad \times \hat{V}_{n+1}^k(\tilde{\mu}_{n+1}^{k+1}) + \breve{V}_{n+1}^{k+1}(\tilde{\mu}_{n+1}^{k+1})\Big). \qquad (12)
\end{aligned}$$

We use the Kiefer-Wolfowitz type scheme proposed above with truncation, i.e., we truncate the approximate empirical gradient on both positive and negative sides at ±1 in order to avoid numerical instabilities.

In the following sample plots, the various parameter values are:

- The Markov chain is evaluated for N time steps and here $N = 30$
- $a = 0.8$
- $b = 0.1$
- $\lambda = 0.5$
- The actual transition probabilities of the service rate chain is
  - Probabilty to go from a to b = 0.4 and to stay in a = 0.6
  - Probabilty to go from b to a = 0.7 and to stay in b = 0.3
- The transition probabilities used in the importance sampling measure are the same as the actual transition probabilities of the service rate chain.
- $\delta = 0.01$.
- $a(n) = \frac{1}{1+\lceil \frac{n}{M} \rceil}$ for $M = 1000$.
- $b(n) = \frac{1}{2+\lceil \frac{m \log m}{N'} \rceil}$ for $n = mM', m = 0, 1, \cdots$, and 0 otherwise where $M' = 500$ and $N' = 10$.

The following figures show simulation results for $N = 30$ for three different initial guesses and two sets of observations. Figures 1-3 correspond to one set of observations and figures 4-6 to the other. As seen in these figures, the scheme showed consistent convergence, though to a value dependent on the observation trace. We also experimented with $N = 10$ and 20 (not reported here). The accuracy of the estimate tended to improve for higher N as expected.

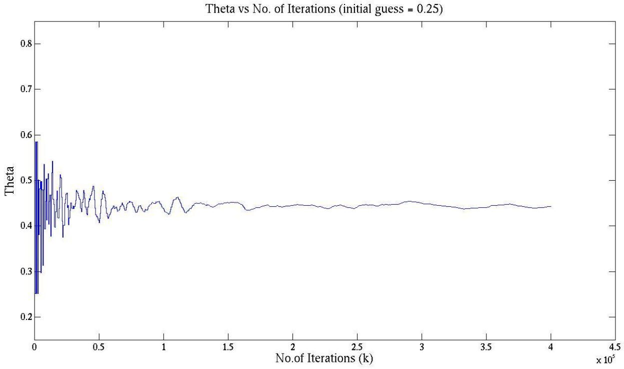

Fig. 1.

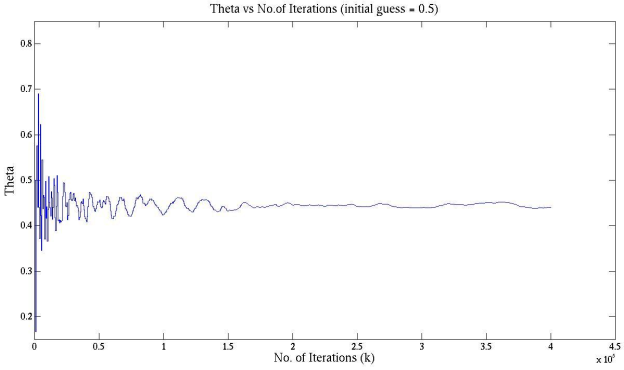

Fig. 2.

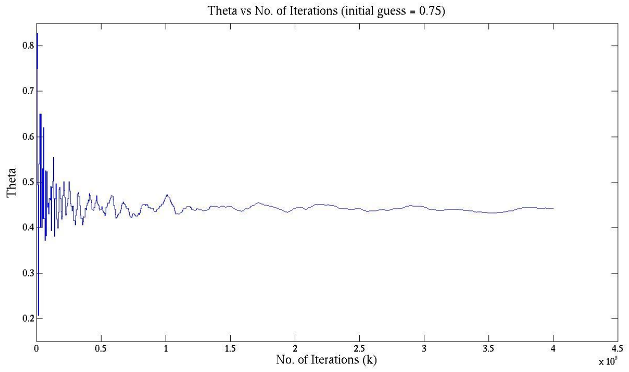

Fig. 3.

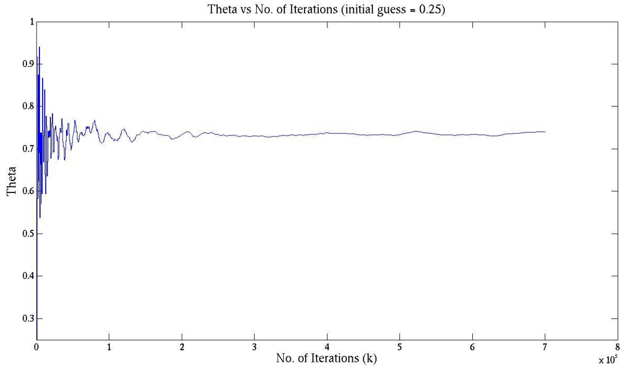

Fig. 4.

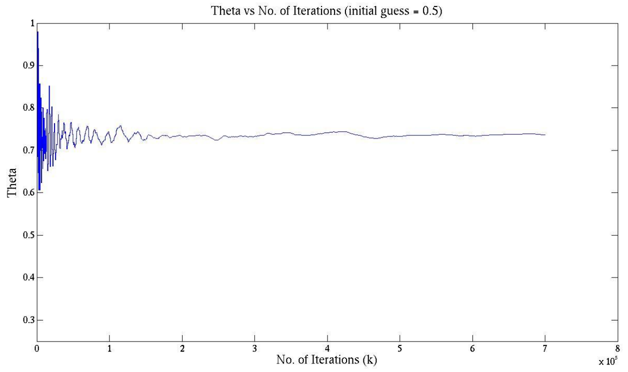

Fig. 5.

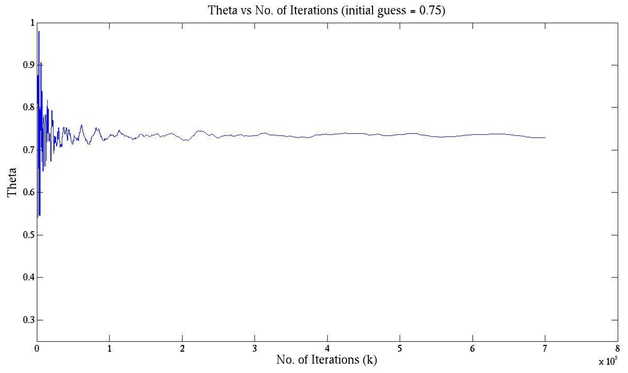

Fig. 6.

IV. Conclusions

We proposed and analyzed a simulation based EM algorithm for parameter estimation in HMMs. This employed ideas from reinforcement learning for variance reduction. The numerical experiments presented here support the theoretical claims. This opens up several possibilities for future research.

1) As it stands, the scheme is a computationally intensive, off-line scheme. A good on-line variant would call for possibly further approximations to lower the computational budget and acceleration methods to speed up convergence. Ideas from reinforcement learning that can be fruitfully imported are function approximation [3] (Chapter 16), adaptive importance sampling [1], [8], and split sampling [7].
2) More generally, this opens up the possibility of an alternative approach for reinforcement learning for Partially Observed Markov Decision Processes (POMDPs). This will be pursued in a sequel.

Acknowledgment

Research of the first author was supported in part by a J. C. Bose Fellowship, a grant from IRCC, IIT Bombay, and a grant ‘Distributed Computation for Optimization over Large Networks and High Dimensional Data Analysis’ from the Dept. of Science and Technology, Govt. of India.

References

[1] T. P. I. Ahamed, V. S. Borkar and S. Juneja, “Adaptive importance sampling technique for Markov chains using stochastic approximation”, *Operations Research* 54(3), pp. 489-504, 2006.

[2] A. Bain and D. Crisan, *Fundamentals of Stochastic Filtering*, Springer Verlag, Berlin-Heidelberg, 2009.

[3] D. P. Bertsekas, *Dynamic Programming and Optimal Control, Vol. II* (4th edition), Athena Scientific, Belmont, MA, 2012.

[4] V. S. Borkar, *Topics in Controlled Markov Chains*, Pitman Research Notes in Maths. No. 240, Longman Scientific and Technical, Harlow, UK, 1991.

[5] V. S. Borkar, “A remark on control of partially observed Markov chains”, *Annals of Operations Research* 29(1), pp. 429-438, 1991.

[6] V. S. Borkar, *Stochatic Approximation: A Dynamical Systems Viewpoint*, Hindustan Book Agency, New Delhi, India, and Cambridge Uni. Press, Cambridge, UK, 2008.

[7] V. S. Borkar, “Reinforcement learning – a bridge between numerical methods and Monte Carlo”, in ‘*Perspectives in Mathematical Sciences I: Probability and Statistics* (N. S. N. Sastry, T. S. S. R. K. Rao, M. Delampady and B. Rajeev, eds.), World Scientific, Singapore, pp. 71-91, 2009.

[8] V. S. Borkar and A. V. Jain, “A reinforcement learning approach to particle filters”, *submitted*, 2013.

[9] O. Cappé, E. Moulines and T. Rydén, *Inference in Hidden Markov Models*, Springer Verlag,New York, 2005.

[10] A. P. Dempster, N. M. Laird and D. B. Rubin, “Maximum likelihood from incomplete data via the EM algorithm”, *Journal of the Royal Statistical Society, Series B* 39(1), pp. 138, 1977.