Manuscript submitted to
AIMS' Journals
Volume **X**, Number **0X**, XX **200X**

Website: http://AIMsciences.org

pp. **X–XX**

BAYESIAN ONLINE ALGORITHMS FOR LEARNING IN DISCRETE HIDDEN MARKOV MODELS

Roberto C. Alamino

Neural Computing Research Group, Aston University
Main Building, Birmingham, B7 4ET, United Kingdom

Nestor Caticha

Instituto de Física, Universidade de São Paulo
CP 66318, São Paulo, SP, CEP 05389-970, Brazil

(Communicated by the associate editor name)

Abstract. We propose and analyze two different Bayesian online algorithms for learning in discrete Hidden Markov Models and compare their performance with the already known Baldi-Chauvin Algorithm. Using the Kullback-Leibler divergence as a measure of generalization we draw learning curves in simplified situations for these algorithms and compare their performances.

1. **Introduction.** The unifying perspective of the Bayesian approach to machine learning allows the construction of efficient algorithms and sheds light on the characteristics they should have in order to attain such efficiency. In this paper we construct and characterize the performance of mean field online algorithms for discrete *Hidden Markov Models* (HMM) [5, 9] derived from approximations to a fully Bayesian algorithm.

HMMs form a class of graphical models used to model the behavior of time series. They have a wide range of applications which includes speech recognition [9], DNA and protein analysis [3, 4] and econometrics [10].

Discrete HMMs are defined by an underlying Markov chain with hidden states q_t, $t = 1, ..., T$, in the set $S = \{s_1, s_2, ..., s_n\}$, initial probability vector $\pi_i = \mathcal{P}(q_1 = s_i)$ and transition matrix $A_{ij}(t) = \mathcal{P}(q_{t+1} = s_j | q_t = s_i)$, with $i, j = 1, .., n$. At each time step, the state q_t emits a state y_t in the set $O = \{o_1, ..., o_m\}$ with a probability given by $B_{i\alpha}(t) = \mathcal{P}(y_t = o_\alpha | q_t = s_i)$, with $i = 1, ..., n$ and $\alpha = 1, ..., m$. When A and B do not depend on time the HMM is said *homogeneous*, this is a simplifying assumption, which is not needed for online learning.

The *observed states* y_t of the HMM represent the observations of the time series, i.e., a time series from time $t = 1$ to $t = T$ is represented by the *observed sequence* $y_1^T = \{y_1, y_2, ..., y_T\}$. The unavailable or *hidden states* q_t form the so called *hidden sequence* $q_1^T = \{q_1, q_2, ..., q_T\}$.

2000 *Mathematics Subject Classification.* Primary: 68T05; Secondary: 60J20, 62F15.

Key words and phrases. HMM, online algorithm, generalization error, Bayesian algorithm.

This work was supported mainly by FAPESP and partially by EVERGROW, IP no. 1935 in FP6 of the EU.

The parameters that define the HMM are π, A and B which we denote compactly as $\omega = (\pi, A, B)$. The probability of observing a particular sequence y_1^T given parameters ω is

$$\begin{aligned}\mathcal{P}(y_1^T|\omega) &= \sum_{q_1^T} \mathcal{P}(y_1^T, q_1^T|\omega) \\ &= \sum_{q_1^T} \mathcal{P}(y_1)\mathcal{P}(y_1|q_1) \prod_{t=2}^{T} \mathcal{P}(q_{t+1}|q_t)\mathcal{P}(y_t|q_t).\end{aligned} \tag{1}$$

Most applications consist in adapting the parameters of the HMM in order to produce sequences which mimic the behavior of some given time series. This is the *learning* process on the HMM. Depending on how the data is presented it can range from *offline* to *online*. In the former, the whole dataset, usually given by just one observed sequence y_1^T, is presented to the algorithm and the optimal parameters are calculated all at once. In the later, the dataset, usually composed of a certain number P of independently generated observed sequences, is presented only a part at a time after which, a partial calculation of the parameters is made. We will concentrate our analysis in the last case.

Several merit figures can be used to measure the performance of a learning algorithm. For the sake of simplicity we will study a scenario where the data set has been generated by a HMM of unknown parameters. This is an extension of the student-teacher scenario where the statistical mechanical properties of neural networks have been extensively studied. The performance of the learning process, as a function of the number of observations P, is given by how *far*, as measured by some suitable defined criterion, lies the student HMM from the teacher HMM. In this paper we only concentrate in the naturally arising *Kullback-Leibler divergence* (KL-divergence), which although not immediately usable in practice, since it needs knowledge from the unavailable teacher, is a simple extension of the idea of generalization error and therefore can be very informative.

We propose two algorithms for learning on HMMs and compare their performances with the online *Baldi-Chauvin Algorithm* (BC) [2] where a softmax representation of the HMM parameters is used to calculate a maximum likelihood estimate. Starting from a Bayesian formulation of the learning problem we introduce a *Bayesian Online Algorithm* (BOnA) for HMMs. This can be simplified, without noticeable deterioration of the performance, leading to a *Mean Posterior Algorithm* (MPA). Using the KL-divergence, we draw learning curves for these algorithms in simplified situations and compare with the BC algorithm.

These methods, inspired by the works of Opper [7] and Amari [1] are essentially mean field methods [8]. The Bayesian prescription for the posterior usually leads to a distribution that cannot be handled in practice. The idea is to introduce a manifold of tractable distributions which will be used as priors. The information of the new datum leads, through Bayes theorem, to a non-tractable posterior. The basic inference step is to consider as the new prior to be used with the next datum, not the posterior itself, but one of the tractable distributions. Which one? The closest, in some sense. The choice of distance defines the algorithm.

The rest of the paper is organized as follows: in section 2 we derive the Bayesian online algorithm for HMMs and propose the MPA approximation for it. We present conclusions and perspectives for future works in section 3.

2. **The Bayesian Online Algorithm.** The Bayesian Online Algorithm (BOnA), which can be found in [7], is a method for estimating parameters using the power of Bayesian inference.

Our task here is to adjust the HMM parameters ω using the information contained in a dataset given by

$$D_P = \{y^1, ..., y^P\}. \tag{2}$$

To accomplish it in a Bayesian way, we need to estimate the parameters using a probability distribution $\mathcal{P}(\omega|D_P)$ which encodes the information about the probability of the parameters after the observation of D_P. Assuming that the observations in D_P are independent, at each new sequence we can update our previous distribution in a new posterior using Bayes' theorem.

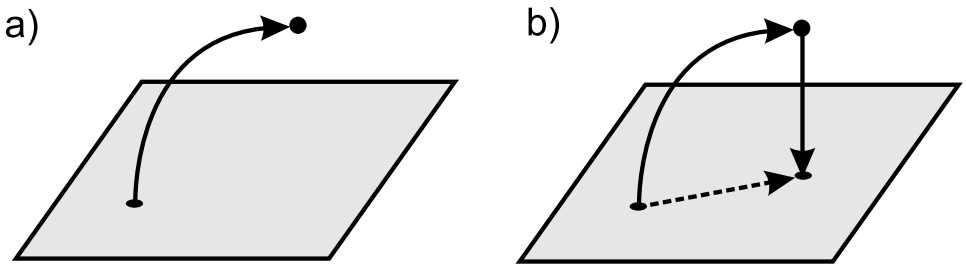

FIGURE 1. In gray we show the manifold of the tractable distributions. The two-step algorithm is represented by the arrows: a) 1st. step: a new observation modifies the former distribution taking it away from the parametric family manifold. b) 2nd. step: the modified distribution is projected back into the original manifold by minimization of the KL-divergence.

If we choose a prior distribution belonging to a parametric family with parameters λ, after one update the posterior will no longer be in this family. The method consists in projecting this posterior back into the parametric manifold by minimizing the KL-divergence

$$D_{KL}(\mathcal{P}(\omega|\lambda^P, y^{P+1})||\mathcal{P}(\omega|\lambda^{P+1})) = \int \mathcal{P}(\omega|\lambda^P, y^{P+1}) \ln \frac{\mathcal{P}(\omega|\lambda^P, y^{P+1})}{\mathcal{P}(\omega|\lambda^{P+1})} d\omega, \tag{3}$$

what is represented schematically in figure 1.

At each point in the learning process, the estimative for ω that we will choose will be its mean in the final distribution.

The choice of the parametric family of distributions will be dictated by the particular problem we are dealing with. When the parametric family is of the form

$$P(x) = \frac{1}{Z} e^{-\sum_i \lambda_i f_i(x)}, \tag{4}$$

which is the case when the distribution is obtained by the maximum entropy method using as constraints the average value of n arbitrary functions $f_i(x)$

$$\langle f_i(x) \rangle_P \equiv \int dx f_i(x) P(x), \qquad i = 1, ..., n, \tag{5}$$

minimizing the KL-divergence turns out to be equivalent to equate the average of the n constraint functions

$$\langle f_j \rangle_P = \langle f_j \rangle_Q \,, \tag{6}$$

where the indices indicate in which distribution we need to take the averages and $Q(x)$ is the still unprojected posterior.

2.1. **Bayesian Online Algorithm for HMMs.** In order to apply the BOnA to the case of HMMs, we need to choose an appropriate parametric family. Making the simplifying assumption that matrices π, A and B are independent, we can write a factorized distribution

$$\mathcal{P}(\omega|u) \equiv \mathcal{P}(\pi|\rho)\mathcal{P}(A|a)\mathcal{P}(B|b), \tag{7}$$

where $u = (\rho, a, b)$ are parameters of the distributions. The vector π and each row of the matrices A and B are discrete probability distributions for the initial states, the transition of a given hidden state and the emission of an observed state by a given hidden one respectively. As they are different distributions, we will assume that they can be treated independently and factorize the distribution (7) once again using

$$\mathcal{P}(A|a) = \prod_{i=1}^{n} \mathcal{P}(A^i|a^i), \qquad \mathcal{P}(B|b) = \prod_{i=1}^{n} \mathcal{P}(B^i|b^i), \tag{8}$$

where

$$A^i \equiv (A_{i1}, ..., A_{in}), \qquad B^i \equiv (B_{i1}, ..., B_{im}), \tag{9}$$

which gives for (7)

$$\mathcal{P}(\omega|u) \equiv \mathcal{P}(\pi|\rho) \prod_{i=1}^{n} \mathcal{P}(A^i|a^i)\mathcal{P}(B^i|b^i). \tag{10}$$

where each factor is a density of probability that a probability has a given value. A natural choice is the family of Dirichlet distributions

$$\mathcal{D}(x|u) = \frac{\Gamma(u_0)}{\prod_{i=1}^{N} \Gamma(u_i)} \prod_{i=1}^{N} x_i^{u_i - 1}, \tag{11}$$

where $x = (x_1, ..., x_N)$ is a N-dimensional real vector and $u = (u_1, ..., u_N)$ subject to the constraints

$$0 \leq x_i \leq 1, \qquad \sum_{i=1}^{N} x_i = 1, \qquad u_i > 0, \tag{12}$$

and $u_0 = \sum_i u_i$.

As (10) is a factorized distribution, the minimization of the KL-divergence can be made separately for each factor distribution. The Dirichlet distribution can be obtained from the maximum entropy method using as constraints the values of the averages of the logarithms of the variables [11]

$$0 \leq x_i \leq 1, \sum_i x_i = 1, \tag{13}$$

and

$$\int d\mu \, \mathcal{D}(x) = 1, \int d\mu \, \mathcal{D}(x) \ln x_i = \alpha_i, \tag{14}$$

where $d\mu$ is defined as

$$d\mu \equiv \delta\left(\sum_i x_i - 1\right)\prod_i \theta(x_i)dx_i. \tag{15}$$

Extremizing the entropy subject to the constraints is equivalent to extremizing

$$\mathcal{L} = \int d\mu\, \mathcal{D}\ln\mathcal{D} + \lambda\left(\int d\mu\, \mathcal{D} - 1\right) + \sum_i \lambda_i\left(\int d\mu\, \mathcal{D}\ln x_i - \alpha_i\right), \tag{16}$$

and applying $\delta\mathcal{L}/\delta\mathcal{D} = 0$ we have

$$\delta\left(\sum_i x_i - 1\right)\left[\prod_i \theta(x_i)\right]\left(1 + \ln\mathcal{D} + \lambda + \sum_i \lambda_i \ln x_i\right) = 0, \tag{17}$$

with solution

$$\mathcal{D}(x) = \exp(-\lambda - 1)\prod_i x_i^{-\lambda_i}, \tag{18}$$

which is the Dirichlet distribution for $Z = \exp(\lambda + 1)$ and $u_i = 1 - \lambda_i$.

So, the minimization of the KL-divergence corresponds to equating the average of the logarithms in the original and projected distributions. The final result is

$$\begin{aligned}
\psi(\rho_i) - \psi\left(\sum_j \rho_j\right) &= \langle \ln \pi_i \rangle_Q \equiv \mu_i(\rho), \\
\psi(a_{ij}) - \psi\left(\sum_k a_{ik}\right) &= \langle \ln A_{ij} \rangle_Q \equiv \mu_{ij}(a), \\
\psi(b_{i\alpha}) - \psi\left(\sum_\beta b_{i\beta}\right) &= \langle \ln B_{i\alpha} \rangle_Q \equiv \mu_{i\alpha}(b),
\end{aligned} \tag{19}$$

where Q is the posterior distribution and ψ is the digamma function, the logarithm derivative of the gamma function

$$\psi(x) = \frac{d}{dx}\ln\Gamma(x) = \frac{\Gamma'(x)}{\Gamma(x)}. \tag{20}$$

We will call a set of N equations of the form

$$\psi(x_i) - \psi\left(\sum_j x_j\right) = \mu_i, \tag{21}$$

with $i = 1, ...N$ a *digamma system* in the variables x_i with coefficients μ_i.

To solve the problem we need two more steps: calculate the coefficients μ and solve the digamma system (19).

Let us call $P^p(\omega)$ the projected distribution after observation of the sequence y^p, and $Q^{p+1}(\omega)$ the posterior distribution (not projected yet) after the observation of the sequence y^{p+1}. Then, by Bayes' theorem,

$$Q^{p+1}(\omega) = \frac{1}{Z_Q}P^p(\omega)\sum_{q^{p+1}} \mathcal{P}(y^{p+1}, q^{p+1}|\omega), \tag{22}$$

$$Z_Q = \sum_{q^{p+1}}\int d\omega\, P^p(\omega)\mathcal{P}(y^{p+1}, q^{p+1}|\omega). \tag{23}$$

Due to the summation over hidden states, the posterior distribution is a mixture of Dirichlets. If we had access to these states, the posterior would be a simple Dirichlet distribution and we would not need the projection step. The projected distribution P^p is, by construction, the product distribution

$$P^p(\omega) = \mathcal{P}(\pi|\rho^p)\mathcal{P}(A|a^p)\mathcal{P}(B|b^p), \tag{24}$$

where ρ^p, a^p and b^p are the values of the parameters u after observation of the sequence y^p. From here on, let us suppress the indices relative to the sequences by adopting the convention that primed variables correspond to the upper index $p+1$ and unprimed to p. Then, equation (22) can be written as

$$Q'(\omega) = \frac{1}{Z_{Q'}}\mathcal{P}(\pi|\rho)\mathcal{P}(A|a)\mathcal{P}(B|b)\sum_{q'}\mathcal{P}(y',q'|\pi,A,B), \tag{25}$$

$$Z_{Q'} = \sum_{q'}\int d\pi\, dA\, dB\, \mathcal{P}(\pi|\rho)\mathcal{P}(A|a)\mathcal{P}(B|b)\mathcal{P}(y',q'|\pi,A,B). \tag{26}$$

Using the notation

$$\begin{aligned} q'_t = s_i &\Rightarrow \pi(q'_t) \equiv \pi_i, \\ q'_t = s_i, q'_{t+1} = s_j &\Rightarrow A(q'_t, q'_{t+1}) \equiv A_{ij}, \\ q'_t = s_i, y'_t = o_\alpha &\Rightarrow B(q'_t, y'_t) \equiv B_{i\alpha}, \end{aligned} \tag{27}$$

the normalization of the posterior can be factored as

$$\begin{aligned} Z_{Q'} = \sum_{q'}\int d\pi\, \mathcal{P}(\pi|\rho)\pi(q'_1)\cdot\int dA\, \mathcal{P}(A|a)\prod_{t=1}^{T'-1}A(q'_t,q'_{t+1}) \\ \times\int dB\, \mathcal{P}(B|b)\prod_{t=1}^{T'}B(q'_t,y'_t), \end{aligned} \tag{28}$$

where T' is the length of the sequence $y' \equiv y^{p+1}$. The integrals on A and B can be factored once more in the rows of these matrices:

$$\begin{aligned} Z_{Q'} = \sum_{q'}\int d\pi\, \mathcal{P}(\pi|\rho)\pi(q'_1)\cdot\prod_i\int dA^i\, \mathcal{P}(A^i|a^i)\prod_{q'_t=s_i}A(q'_t,q'_{t+1}) \\ \times\prod_i\int dB^i\, \mathcal{P}(B^i|b^i)\prod_{q'_t=s_i}B(q'_t,y'_t). \end{aligned} \tag{29}$$

The calculation of the μ coefficients in (19) now involves expectations over Dirichlet distributions of the form

$$\mu_i = \left\langle\left[\prod_j x_j^{r_j}\right]\ln x_i\right\rangle. \tag{30}$$

Taking the average over the Dirichlet distribution $\mathcal{D}(x|u)$, we have

$$\mu_i = \left\langle\prod_j x_j^{r_j}\right\rangle[\psi(u_i+r_i)-\psi(u_0+r_0)], \tag{31}$$

with

$$\left\langle\prod_j x_j^{r_j}\right\rangle = \frac{\Gamma(u_0)}{\prod_j\Gamma(u_j)}\frac{\prod_j\Gamma(u_j+r_j)}{\Gamma(u_0+r_0)}, \tag{32}$$

solving the problem of calculating the coefficients of the digamma systems.

2.1.1. *Digamma Systems.* The next step is to solve the digamma systems themselves. In order to find a solution for the general form

$$\psi(x_i) - \psi\left(\sum_j x_j\right) = \mu_i, \qquad i = 1, ..., d, \tag{33}$$

we devised the simple method of transforming these equations in a one-dimensional map and find numerically the fixed point by iterating from an arbitrary initial point. What we need to do is to solve for x_i

$$x_i = \psi^{-1}\left[\mu_i + \psi\left(\sum_j x_j\right)\right]. \tag{34}$$

Summation over i using the definition $x_0 \equiv \sum_i x_i$ gives

$$x_0 = \sum_i \psi^{-1}[\mu_i + \psi(x_0)]. \tag{35}$$

We solve this equation by iteration of the map

$$x_0^{(n+1)} = \sum_i \psi^{-1}[\mu_i + \psi(x_0^{(n)})]. \tag{36}$$

This completes the algorithm. BOnA suffers from a common problem of Bayesian algorithms: we have to sum over the hidden variables making the complexity scales as n^T, i.e., exponential in the length of the observed sequences. In the BOnA case we have a worse situation due to the great quantity of digamma functions needed to be calculated numerically. Everything summed up makes the algorithm very time consuming. In the next section, we will develop an approximation that runs faster, although still have exponential complexity in T. This is however *not* a problem since T can be kept fixed and small without depending on the total number of examples, this dependence must be stressed is polynomial.

2.2. **Mean Posterior Approximation.** The algorithm we present now will be called the Mean Posterior Approximation (MPA) and it is a simplification of the BOnA that is itself inspired in the results of its application to Gaussian distributions.

In the BOnA, we projected the posterior distribution into a parametric distribution by minimizing the KL-divergence. For Gaussian distributions this is equivalent to matching the first and second moments of the posterior and projected distributions. Observing this fact, we decided instead of minimizing the KL-divergence matching the mean and one of the variances of the posterior with those of the projected distributions.

With a hat over variables indicating the new estimated values, the matching of the means is given by

$$\hat{\pi}_i = \langle \pi_i \rangle_{Q'} \Rightarrow \hat{\rho}_i = \langle \pi_i \rangle_{Q'} \sum_j \hat{\rho}_j, \tag{37}$$

$$\hat{A}_{ij} = \langle A_{ij} \rangle_{Q'} \Rightarrow \hat{a}_{ij} = \langle A_{ij} \rangle_{Q'} \sum_k \hat{a}_{ik},$$

$$\hat{B}_{i\alpha} = \langle B_{i\alpha} \rangle_{Q'} \Rightarrow \hat{b}_{i\alpha} = \langle B_{i\alpha} \rangle_{Q'} \sum_\beta \hat{b}_{i\beta}.$$

We still have some freedom that can be fixed by matching the first variance of each Dirichlet distribution. For the initial probabilities π, defining $\rho_0 \equiv \sum_i \rho_i$, this will be given by

$$\left\langle \pi_1^2 \right\rangle_{Q'} - \langle \pi_1 \rangle_{Q'}^2 = \frac{\hat{\rho}_1(\hat{\rho}_0 - \hat{\rho}_1)}{\hat{\rho}_0^2(\hat{\rho}_0 + 1)}. \tag{38}$$

After some algebra we find that the final update equations for the parameters are

$$\begin{aligned} \hat{\rho}_i &= \langle \pi_i \rangle_{Q'} \frac{\langle \pi_1 \rangle_{Q'} - \left\langle \pi_1^2 \right\rangle_{Q'}}{\left\langle \pi_1^2 \right\rangle_{Q'} - \langle \pi_1 \rangle_{Q'}^2}, \\ \hat{a}_{ij} &= \langle a_{ij} \rangle_{Q'} \frac{\langle a_{i1} \rangle_{Q'} - \left\langle a_{i1}^2 \right\rangle_{Q'}}{\left\langle a_{i1}^2 \right\rangle_{Q'} - \langle a_{i1} \rangle_{Q'}^2}, \\ \hat{b}_{i\alpha} &= \langle b_{i\alpha} \rangle_{Q'} \frac{\langle b_{i1} \rangle_{Q'} - \left\langle b_{i1}^2 \right\rangle_{Q'}}{\langle b_{i1}^2 \rangle - \langle b_{i1} \rangle^2}. \end{aligned} \tag{39}$$

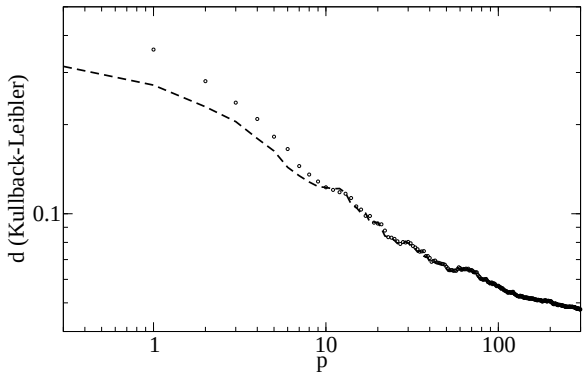

FIGURE 2. Comparison between the learning curves of MPA (dashed line) and BOnA (circles). Logarithmic scale.

These formulas set the values of $\hat{u}$ in the projected distribution. Figure 2 shows that the performance of MPA and BOnA are different in the beginning of the learning process, but as more sequences are processed, the loss of information becomes smaller and both algorithms come closer relatively fast. It is interesting to observe that MPA is always best than BOnA in the simulation, an effect that needs more studies to be fully explained. For this simulation we used $n = 2$, $m = 3$ and $T = 2$ and made the average over 150 random teachers. The initial student is symmetric.

The real computational time for BOnA was 340 minutes, while for MPA was 5 seconds in a 1G

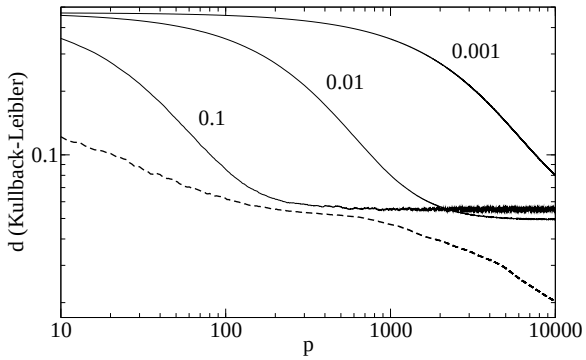

FIGURE 3. Comparison between MPA (dashed line) and Baldi-Chauvin (continuous lines). The values next to the curves indicate λ for the corresponding simulation. The learning rate was fixed to $\eta_{BC} = 0.5$. Logarithmic scale.

Figure 3 compares the performance of MPA to Baldi-Chauvin showing that the generalization ability of MPA is superior. The parameters of the simulations are $n = 2$, $m = 3$ and $T = 2$. The curves are the result of the average over 500 random teachers with symmetric initial students.

3. **Conclusions.** We proposed and analyzed two Bayesian algorithms for online learning in discrete HMMs: the full Bayesian Online Algorithm (BOnA) and the Mean Posterior Approximation (MPA) to the BOnA.

After introducing the BOnA, we developed a simplification that runs faster which we called MPA. The MPA was then compared with the Baldi-Chauvin algorithm and shown to perform better with respect to the generalization ability. Although MPA scales exponentially with the size of the sequence like BOnA, it is not a problem for we can fix this size to some small value and still have a good generalization.

We have applied these algorithms to learning real data time series with good preliminary results. We have also studied the effects of drifting rules and of sharp changes in the series. These results will be published elsewhere.

Acknowledgements. In addition to the funding cited in the beginning of this paper, we would like to thank also Evaldo Oliveira, Manfred Opper and Lehel Csato. This work was made mostly in the University of São Paulo, Brazil and part in Aston University, UK. We would also like to thank Prof. Ruedi Stoop for the kind invitation for contributing with this work.

REFERENCES

[1] S. Amari, *Neural learning in structured parameter spaces - natural Riemannian gradient* NIPS'96 **9**, MIT Press (1996).

[2] P. Baldi and Y. Chauvin *Smooth on-line learning algorithms for hidden Markov models* Neural Computation **6** (1994), 307–318.

[3] P. Baldi and S. Brunak "Bioinformatics: The Machine Learning Approach" (Second Edition) MIT Press, 2001.

[4] R. Durbin, S. Eddy, A. Krogh and Mitchison, G. "Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids" Cambridge University Press, Cambridge, 1998.
[5] Y. Ephraim and N. Merhav *Hidden Markov processes* IEEE Trans. Inf. Theory **48** (2002), 1518–1569.
[6] T. Heskes and W. Wiegerinck *On-line learning with time-correlated examples* in "On-line Learning in Neural Networks" (Ed. D. Saad), Cambridge University Press, Cambridge, (1998), 251–278.
[7] M. Opper *A Bayesian approach to on-line learning* in "On-line Learning in Neural Networks" (Ed. D. Saad), Publications of the Newton Institute, Cambridge Press, Cambridge, (1998), 363–378.
[8] M. Opper and D. Saad "Advanced Mean Field Methods: Theory and Practice" The MIT Press, 2001.
[9] L. R. Rabiner *A tutorial on hidden Markov models and selected applications in speech recognition* Proc. IEEE **77** (1989), 257–286.
[10] T. Rydén, T. Terasvirta and S. Asbrink *Stylized facts of daily return series and the hidden Markov model* J. Applied Econometrics **13** (1998), 217-244.
[11] M. O. Vlad, M. Tsuchiya, P. Oefner and J. Ross *Bayesian analysis of systems with random chemical composition: renormalization-group approach to Dirichlet distributions and the statistical theory of dilution* Phys. Rev. E **65** (2001), 011112(1)–01112(8).

Received September 2006; revised February 2007.

E-mail address: alaminrc@aston.ac.uk
E-mail address: nestor@if.usp.br