Adaptive Online Learning of Bayesian Network Parameters

Ira Cohen[1], Alexandre Bronstein, Fabio G. Cozman[2]
Internet Systems and Storage Laboratory
HP Laboratories Palo Alto
HPL-2001-156
June 20th , 2001*

Bayesian networks, machine learning, adaptive systems

The paper introduces Voting EM, an adaptive online learning algorithm of Bayesian network parameters. Voting EM is an extension of the EM (η) algorithm suggested by [1]. We show convergence properties of the Voting EM that uses a constant learning rate. We use the convergence properties to formulate an error driven scheme for adapting the learning rate. The resultant algorithm converges with the optimal rate of 1/*t* near a maximum while retaining the ability to increase the learning rate in the vicinity of a local maximum or due to changes in the modelled environment.

* Internal Accession Date Only Approved for External Publication
[1] Beckman Institute, University of Illinois at Urbana Champaign, Urbana, IL 61801
[2] Escola Politecnica, Universidade de Sao Paulo, Sao Paulo, SP - Brazil

Adaptive Online Learning of Bayesian Network Parameters

Ira Cohen
Beckman Institute
University of Illinois at Urbana Champaign
405 N. Mathews Ave Urbana, IL 61801

Alexandre Bronstein
Hewlett-Packard Laboratories
1501 Page Mill Road
Palo-Alto, CA 94304

Fabio G. Cozman
Escola Politecnica
Universidade de Sao Paulo
Av. Prof. Mello Moraes 2231 - 05508-900
Sao Paulo, SP - Brazil

Abstract

The paper introduces Voting EM, an adaptive online learning algorithm of Bayesian network parameters. Voting EM is an extension of the EM(η) algorithm suggested by [1]. We show convergence properties of the Voting EM that uses a constant learning rate. We use the convergence properties to formulate an error driven scheme for adapting the learning rate. The resultant algorithm converges with the optimal rate of $1/t$ near a maximum while retaining the ability to increase the learning rate in the vicinity of a local maximum or due to changes in the modelled environment.

1 Introduction

The parameters of a Bayesian network(BN) are determined by the use of expert opinion or by learning from data [2, 3]. The former has the benefit of the life experience of the expert, but often is either too expensive or not accurate enough to set the probabilities of the network. The latter, that is learning from data, is problematic in that data are not always available at the time the BN is constructed. This lack of data at the time of construction can be addressed either by waiting for a batch of data, and performing offline learning on the dataset, or by learning the parameters from data as they are generated and continually adapting, namely online learning. A challenge for both approaches, frequently encountered in real systems, arises when the environment being modelled by the BN changes, either slowly or abruptly, in time or in characteristic.

Online learning of BN parameters has been discussed by [4, 5, 6] and in the work of [1]. Following the general approach introduced in [1], we developed an online learning algorithm which we named Voting EM [7]. The update of the BN parameters is governed by a learning rate which can be kept constant or adaptive.

With a constant learning rate, we show that Voting EM converges, with non-zero error,

to the true parameters. We propose a dynamic learning rate, exploiting the convergence properties of the constant learning rate version of Voting EM. We show that the resultant annealing schedule is proportional to $1/t$ when the errors between the current and previous estimates are small, thus achieving the optimal convergence rate to the maximum of the likelihood function [8, 9]. Allowing the learning rate to increase based on the error allows fast adaptation to changes in the modelled environment and avoids local maxima traps. We base our arguments on known results from similar paradigms for adapting the learning rate that have been suggested in the Neural network context [10, 11]. In the Neural network references, like the dynamic learning rate Voting EM, the algorithms balance the trade off between fast, but potentially only local convergence, and accurate global convergence.

The rest of this paper is organized as follows: in Section 2, we define notations and describe Voting EM. We show the convergence properties of Voting EM using a constant learning rate. In Section 3 we present the adaptation schedule for the learning rate and prove that it both follows the optimal annealing rate $1/t$ and retains the ability to adapt quickly given new contradictory evidence to the previous estimates. We further demonstrate the algorithm in Section 4 using the ICU Alarm network. Finally we summarize our contributions and discuss directions for future work.

2 Voting EM

2.1 Description

The general task is to learn the parameters of the network from a set of data. This implementation assumes a fixed structure S of the network and that the variables are discrete valued. The learning is then the estimation of the conditional probability tables (CPT) entries of the network.

We first describe the general setting of the problem, following the notation in [1]. Let Z_i be a node in the network that takes any value from the set $\{z_i^1, ..., z_i^{r_i}\}$. Let Pa_i be the set of parents of Z_i in the network that takes one of the configurations denoted by $\{{pa_i}^1, ..., {pa_i}^{q_i}\}$. An entry in the CPT of the variable Z_i is given by $\theta_{ijk} = P(Z_i = z_i^k | Pa_i = pa_i^j)$. We are given a set of data cases $D = \{y_1, ..., y_T, ...\}$, and we have a current set of parameters, $\bar{\theta}$, that define the network. The data are either complete, that is all values of the variables are given, or incomplete.

The updating of the network parameters is achieved by the following maximization:

$$\tilde{\theta} \quad = \quad argmax_\theta[\eta L_D(\theta) - d(\theta, \bar{\theta})], \tag{1}$$

where $L_D(\theta)$ is the normalized log likelihood of the data given the network, $d(\theta, \bar{\theta})$ is a distance between the two models and η is the learning rate. The distance that we use in our implementation is the Chi squared distance. Solving the maximization under the constraint that $\sum_k \theta_{ijk} = 1$ for $\forall i, j$, Bauer et al. [1] derived an algorithm which they named EM(η).

Adapting the EM(η) algorithm to the online learning case is straightforward. The evidence becomes a single instance of the network and for each new evidence vector, the network's parameters are all updated according to the rule:

$$\theta_{ijk}^T = \begin{cases} \theta_{ijk}^{T-1} + \eta[\frac{P(z_i^k, pa_i^j | y_T, \theta^{T-1})}{P(pa_i^j | y_T, \theta^{T-1})} - \theta_{ijk}^{T-1}], & \text{for } P(pa_i^j | y_T, \theta^{T-1}) \neq 0 \\ \theta_{ijk}^{T-1}, & \text{otherwise,} \end{cases} \tag{2}$$

We call the online update method of Eq. 2 the Voting EM algorithm. Such online update rule is referred to as stochastic learning, with $\frac{P(z_i^k, pa_i^j | y_T, \theta^{T-1})}{P(pa_i^j | y_T, \theta^{T-1})}$ being the instantaneous gradient estimate of the constraint optimization problem. The learning rate η controls how

much we rely on the past. As η approaches 1, the past is weighted less, and the update of the parameters is based more on the present data. As η approaches zero, the network parameters change slowly from the previous model. The learning rate can either be constant or adaptive. In the next sections we discuss both, and present an adaptive error-driven learning rate.

2.2 Convergence properties using a constant learning rate

Given the sequence of full evidence data from the network $D = \{y_1, ..., y_T, ...\}$ and a constant learning rate η, the following theorem characterizes the asymptotic behavior of the online update rule. With no loss of generality, assume that $P(pa_i^j|y_t, \theta^t) = 1$ for all $t = \{1, .., T, ...\}$, that is the parents are always observed in their *j'th* configuration. For ease of notation, we denote θ_{ijk}^t as X_t and rewrite Eq. 2 as:

$$X_t = (1 - \eta)X_{t-1} + \eta \cdot I_t \ , \tag{3}$$

where I_t is an indicator function, making the process $\{I_t\}$ an independent identically distributed (i.i.d) Bernoulli random process, given as:

$$I_t = \begin{cases} 1 & \text{with probability} \quad \theta_{ijk} = c^* \\ 0 & \text{with probability} \quad 1 - c^* \end{cases} \tag{4}$$

where $c^* = P(X_i = x_i^k | Pa_i = pa_i^j)$ is the true CPT entry of the Bayesian network.

Theorem 1 *Given a discrete Bayesian Network S, a sequence of full observation vectors D, the update rule given in Eq. 3 and the constraint $0 < \eta \leq 1$, the following properties hold:*

1. *X_t is a consistent estimate of c^*, i.e.,*

$$E[X_t] = (1 - \eta)^t X_0 + (1 - (1 - \eta)^t) \cdot c^*, \ t \geq 0 \ \Rightarrow \ lim_{t \to \infty} E[X_t] = c^*, \tag{5}$$

   where X_0 is the initial value at $t = 0$.

2. *The variance of the estimate X_t is finite and follows:*

$$Var[X_t] = \frac{\eta \cdot c^*(1 - c^*)}{2 - \eta} \cdot (1 - (1 - \eta)^{2t+2}) \ \Rightarrow \tag{6}$$

$$lim_{t \to \infty} Var[X_t] = \frac{\eta}{2 - \eta} \cdot c^*(1 - c^*) \tag{7}$$

3. *For $t \to \infty$ the following inequality holds: $P(|X_t - c^*| \geq q\sigma) \leq \frac{1}{q^2}$, where $q > 0$*

The proof is given in [7]. From the theorem we see that in the mean, the online update rule approaches the true CPT values. The parameter η controls the rate of convergence. Eq. 5 and 6 imply that $\eta = 1$ yields the fastest convergence, but with the largest variance. For smaller η's the convergence is slower, but the variance is also small. The variance is proportional to η, and remains finite in the limit, thus the estimate will always oscillate around the true parameter. η can also be understood as a 'forgetting bias' of the learning algorithm. The system forgets the past at an exponential rate, proportional to η. This is similar to the *fading* factor introduced by Olesen et al. [6].

The third property stated in the theorem gives the confidence intervals of the estimated CPT's with respect to the variance of the estimate. We use this property in the adaptive learning rate algorithm of the next section.

When there are missing data, the updated probabilities change less than in the complete case. For sufficiently long sequences of data, missing data have diminishing influence on

the estimate, and the properties generally still apply. For the case of hidden nodes (that is nodes that are never observed), these theorems do not hold. Nevertheless, the update rule can be used with hidden variables as well. Spiegelhalter and Cowell [5] discuss various cases of online learning in BN with hidden variables with regard to the Spiegelhalter-Lauritzen(SL) algorithm [4]. They show that meaningful results are achieved if the hidden variable has descendants and parents that are observable. Similar results are expected from Voting EM as well despite the difference between the algorithms; Voting EM taking a frequentist approach and the SL using a Bayesian approach to parameter estimation.

3 Adding adaptiveness to Voting EM

Various studies discuss constant learning rates and different schedules for adapting the learning rate. As discussed in the previous section, constant learning rate converges in the mean to the global maximum, but with non-zero error. Adapting the learning rate can achieve convergence with zero-error. Several different adaptation schedules have been suggested in the stochastic learning literature, guaranteeing different types of convergence. The annealing rate $1/t$ yields the fastest asymptotic convergence to a local maxima with zero mean-square error. In the context of BN, incremental maximum-likelihood(ML) counting and the incremental EM of Neal and Hinton [12] can be formulated as such a schedule in the case of full and missing data (see [7] for a detailed discussion). It is also known that for η proportional to $1/log(t)$, escape from local maxima traps is assured, but convergence to the global maximum is very slow [13]. One disadvantage of both schedules is the dependency on an absolute time origin, making them less sensitive to changes as time passes. Other schedules suggest using a constant learning rate during the early part of training (a search phase) followed by a switch to an annealing rate of $1/t$ when in the vicinity of the function's maximum [8]. This schedule attempts to overcome local maxima traps by switching back to a constant learning rate when the trajectory of the estimates become too smooth. Error driven adaptation schedules eliminate the dependency on absolute origin of time and yield fast convergence when in the vicinity of a global maximum while retaining the ability to escape local maxima [10]. We use a similar error-driven approach which follows from the properties of Voting EM with a constant learning rate.

From the first and second properties stated in Theorem 1, the learning rate should be reduced when convergence is reached. On the other hand, the learning rate should be increased when there is a large error between the estimated parameter and its mean value. A large error can occur around a local maximum or when the modelled environment changes. The inequality stated in the third property of Theorem 1 defines what is a large error. In the following schedule we assign a different learning rate for each set of parameters in the network, that is, for each node Z_i with parents pa_i and parent's configuration pa_i^j, the learning rate is denoted as η_{ij}. Letting T denote the total number of data, t the number of times $Pa_i = pa_i^j$ and δt the number of times $Pa_i = pa_i^j$ since the last time η_{ij} changed, the proposed schedule for adapting the learning rate is as follows:

For each $Pa_i = pa_i^j$, the $j'th$ configuration of the parents of node i do the following steps:

- Initialize the following:
  - Set $P[X_i = x_i^k | Pa_i = {pa_i}^j] = \theta_{ijk}^t$ to some initial value for $k = 1, ..., r_i$
  - Set η_{ij} to some value between 0 and 1. A high value can be initially set.
  - Set $t,\ \delta t = 0$.
- Given an observation vector y_T, if $Pa_i = pa_i^j$ do the following:
  1. Estimate θ_{ijk}^{t+1} using the update rule of Eq. 3, where η is replaced by η_{ij}.
  2. If $\mid \theta_{ijk}^{t+1} - E[\theta_{ijk}^{t+1}] \mid > q \cdot \sigma_{ij}^{t+1}$ then $\quad \backslash\backslash\ q > 0$

increase η_{ij}: $\eta_{ij} \leftarrow \eta_{ij} \cdot m$, $\quad \backslash\backslash\ m > 1$
set $\delta t = 0$
Else if $(1 - \eta_{ij})^{\delta t} \leq \alpha$ $\quad \backslash\backslash\ \alpha << 1$
decrease η_{ij}: $\eta_{ij} \leftarrow \eta_{ij} \cdot m^{-1}$
set $\delta t = 0$
Else set $\delta t = \delta t + 1$

3. Get the next observation and repeat steps 1-2.

q, m and α are adjustable parameters. q determines the confidence in the decision to increase η; from the Chebychev inequality this confidence is equal to $1-\frac{1}{q^2}$. α is a threshold reflecting the acceptable convergence of the parameters. $E[\theta_{ijk}^{t+1}]$ and σ_{ij}^{t+1} are the mean and variance of the estimated parameter. The mean can be estimated by a running average (to be reset every time η_{ij} is increased). Although it is not an unbiased estimate of the mean, it is a consistent one. The variance can be estimated using the analytical closed form of Eq. 6, with t replaced by δt and $c^* = 0.5$ (which gives the worst case estimate of the variance).

We now show that the rate of decrease of η_{ij} is proportional to $1/t_n$, where t_n is the number of times $Pa_i = pa_i^j$ until the *n'th* reduction of η. This rate is consistent with the optimal annealing rate $1/t$. For purpose of the following analysis, assume that η_{ij} is not increased at any point in time. If η_{ij} is increased, t is reset to zero and the analysis still holds.

Theorem 2 *Using the reduction rule outlined above, with $\eta_{ij}(0) = \eta_0$ and for $t_n \leq t < t_{n+1}$, the learning rate $\eta_{ij}(t)$ is bounded by the following:*

$$\frac{log(\alpha^{-1})}{m-1}\frac{1}{t_n + K + log(\alpha^{-1})n} \leq \eta_{ij}(t) < \frac{log(\alpha^{-1})}{m-1}\frac{1}{t_n - n + K}, \tag{8}$$

where $K = \frac{log(\alpha^{-1})}{\eta_0(m-1)}$, $0 < \alpha < 1$, $n \in \mathbb{N}$ and $m > 1$.

Proof of the bounds follows from the recursion $\eta(t_n) = \eta_0 \cdot m^{-n}$ where $t_n = \sum_{l=1}^{n} \delta t_l$ and $\frac{log(\alpha)}{log(1-\eta(t_{l-1}))} \leq \delta t_l < \frac{log(\alpha)}{log(1-\eta(t_{l-1}))} + 1$. Using the approximation $\frac{-x}{1-x} < log(1-x) < -x$ (for $0 < x < 1$) and manipulations on t_n the bounds are derived. We do not give a full length proof for lack of space. As t_n increases, the bounds become tighter. η_{ij} is reduced at discrete steps, that increase in length as t increases. Therefore η_{ij} will have longer intervals at which it remains constant, but at the end of the interval, it reduces as $1/t$.

As long as the error between the current estimate and its mean remains small, η_{ij} reduces with the optimal schedule, and at the limit the estimated CPT's converge to the target CPT's with zero-error. However, if the error becomes large, η_{ij} increases, increasing the ability to adapt faster to changes in the modelled environment or break out of local maxima. Furthermore, the time origin is effectively shifted forward every time η_{ij} is increased, making the algorithm insensitive to an absolute time origin.

4 Experimental results

We demonstrate the adaptive learning rate Voting EM using the Alarm network for ICU ventilator management [14]. To demonstrate the ability of Voting EM to adapt to abrupt changes in the modelled environments, we draw 2000 i.i.d samples using the original CPT settings, and 2000 more samples after changing the CPTs of two nodes in the network; HISTORY and HR. In addition, we sample two test sets, one for each of the two different BN setting. We sample a second set to demonstrate the adaptability to slow changes of

environments by slowly changing some of the CPTs for each new sample. These changes are hypothetical for the ICU Alarm network case.

Figure 1(a-c) shows some of the estimated conditional probabilities as function of the number of samples for the abrupt change in environment. The estimated probabilities are shown for both the Voting EM and incremental ML.

The ML estimation converges as $1/t$ to the true CPT before the change, at a similar rate as Voting EM. However, ML adapts slower to the abrupt change and would adapt even slower had the change occurred after more samples. Voting EM adapts faster by increasing η as the change is detected, and decreasing it after reaching the new static parameter. The global effect is demonstrated through the log-likelihood(LL), computed over the test set after each new training sample. The LL plots are shown in Figure 1(d) and are compared to the baseline LL computed over the test sets using the true network parameters. For the first 2000 samples, the LL of Voting EM is only slightly higher than ML estimation. After the change of parameters, both estimates have a significant drop in the LL, but the quick adaptation of Voting EM is apparent, while ML adapts at a slower rate.

The results of two slowly varying parameters are shown in Figure 1(e-f). Voting EM is capable of tracking the changes by keeping η large enough to keep up with the rate of change, thus finding the tradeoff between memory size and smooth convergence. ML estimation adapts slowly, and as more samples are presented is less capable of tracking the changes.

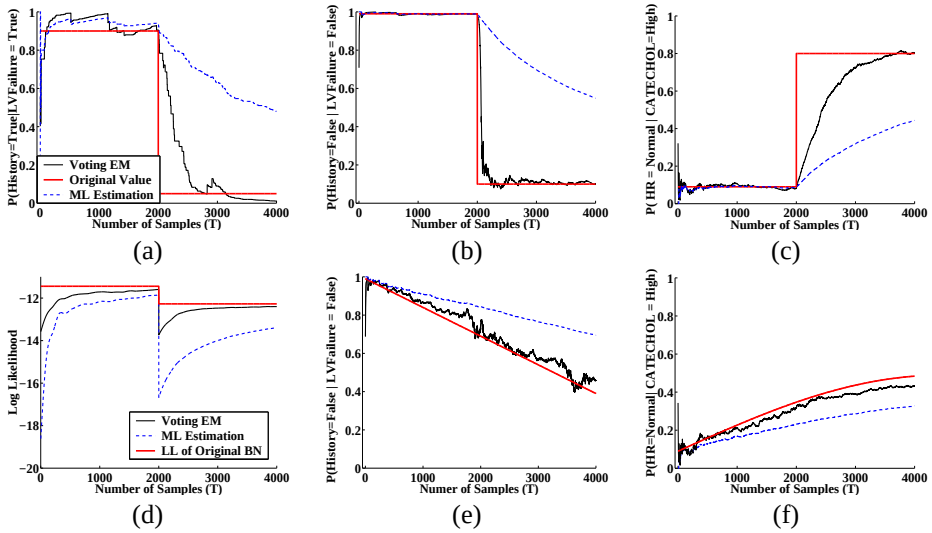

Figure 1: Adaptive learning rate for Voting EM in changing environments. (a)-(c) are examples of CPT estimates for the abrupt changes case. (d) The log-likelihood over the test data sets (e-f) CPT estimates for the slowly varying case.

5 Conclusions and Future Work

We have presented Voting EM, an adaptive online learning algorithm for Bayesian network parameters and discussed different choices for the learning rate. We have shown the convergence properties using a constant learning rate. We use the convergence properties of the constant learning rate of Voting EM to devise an error driven schedule for updating the learning rate. The update schedule reduces the learning rate with the optimal annealing rate of $1/t$ and increases it when a large error is detected between the current and past esti-

mates. This schedule provides both fast asymptotic convergence and the ability to adapt to a changing environment and break out of local maxima traps.

The experiments described in this paper show convergence and adaptation with relatively scarce sequential learning data. This suggests to us that Voting EM may be useful in real-world applications with those characteristics. We intend to evaluate Voting EM as part of a classification application, run against a corporate mail firewall, aimed at fault detection [15]. We also intend to explore further how Voting EM performs in classification situations with abundant unlabelled learning data.

Acknowledgements

We would like to thank Marsha Duro of HP Labs for her editorial assistance on this paper.

References

[1] E. Bauer, D. Koller, and Y. Singer. Update rules for parameter estimation in bayesian networks. In *Uncertainty in Artificial Intelligence (UAI)*, pages 3–13, 1997.

[2] D. Heckerman. A tutorial on learning with bayesian networks. In *Report No. MSR-TR-95-06*. Microsoft Research, 1995.

[3] J. Pearl. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan Kaufmann, 1988.

[4] D.J. Spiegelhalter and S.L. Lauritzen. Sequential updating of conditional probabilities on directed graphical structures. *Networks*, 20(5):579–605, Aug 1990.

[5] D.J. Spiegelhalter and R.G. Cowell. Learning in probabilistic expert systems. *Bayesian Statistics 4*, pages 447–466, 1992.

[6] K.G. Olesen, S.L. Lauritzen, and F.V. Jensen. ahugin: A system creating adaptive causal probabilistic networks. In *Uncertainty in Artificial Intelligence (UAI)*, pages 223–229. Morgan Kaufmann, 1992.

[7] I. Cohen, A. Bronstein, and F.G. Cozman. Online learning of bayesian network parameters. In *Report No. HPL-2001-55(R.1)*. HP Labs, June 2001.

[8] G.B. Orr and T.K. Leen. Using curvature information for fast stochastic search. In *Advances in Neural Information Processing Systems 9*. MIT Press, 1997.

[9] H.J. Kushner and G.G. Yin. *Stochastic approximation algorithms and applications*. Springer-Verlag, 1997.

[10] N. Barkai, H.S. Seung, and H. Sompolinsky. Local and global convergence of online learning. *Physical Review Letters*, 75:1415–18, 1995.

[11] N. Murata, K.R. Muller, A. Ziehe, and S.I. Amari. Adaptive on-line learning in changing enviroments. In *Advances in Neural Information Processing Systems (NIPS)*, pages 599–605. MIT Press, 1996.

[12] R.M. Neal and G.E. Hinton. A view of the EM algorithm that justifies incremental, sparse and other variants. *in Learning in Graphical Models*, pages 355–368, 1998.

[13] H.J. Kushner. Asymptotic global behavior for stochastic approximation and diffusions with slowly decreasing noise effects: Global minimization via monte carlo. *SIAM Journal on Applied Mathematics*, 47(1):169–185, 1987.

[14] L. Beinlich, H. Suermondt, and G. Cooper. The alarm monitoring system: A case study with two probabilistic inference techniques for belief networks. In *Proceedings of the Second European Conference on Artificial Intelligence in Medicine*, pages 247–256. Springer Verlag, 1989.

[15] A. Bronstein, I. Cohen, J. Das, M. Duro, G. Kleyner, M. Mueller, and S. Singhal. Self-aware services: Using bayesian networks for detecting anomalies in internet-based services. In *Proceedings of the IEEE/IFIP 7th International Symposium on Integrated Network Management IM-01*, pages 623–638. Pavlou, George, Anerousis, Nikos, and Liotta, Antonio (eds.), IEEE Publishing, 2001.