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-2001-55 (R.1)
June 5th , 2001*

The paper introduces Voting EM, an online learning algorithm of Bayesian network parameters that builds on the EM(η) algorithm suggested by (Bauer et al., 1997). We prove convergence properties of the algorithm in the mean and variance, and demonstrate the algorithm's behavior on synthetic data. We show the relationship between Maximum-Likelihood (ML) counting and Voting EM. We demonstrate that Voting EM is able to adapt to changes in the modelled environment and to escape local maxima of the likelihood function. Voting EM also handles both the complete and missing data cases. We use the convergence properties to further improve Voting EM by automatically adapting the learning rate η. The resultant enhanced Voting EM algorithm converges more quickly and more closely to the true CPT parameters; further, it adapts more rapidly 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

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 online learning algorithm of Bayesian network parameters that builds on the EM(η) algorithm suggested by (Bauer et al., 1997). We prove convergence properties of the algorithm in the mean and variance, and demonstrate the algorithm's behavior on synthetic data. We show the relationship between Maximum-Likelihood (ML) counting and Voting EM. We demonstrate that Voting EM is able to adapt to changes in the modelled environment and to escape local maxima of the likelihood function. Voting EM also handles both the complete and missing data cases. We use the convergence properties to further improve Voting EM by automatically adapting the learning rate η. The resultant enhanced Voting EM algorithm converges more quickly and more closely to the true CPT parameters; further, it adapts more rapidly to changes in the modelled environment.

1 INTRODUCTION

Bayesian networks (BNs) have gained wide popularity in the Artificial Intelligence community over the past few years. BNs can be used for general-purpose classifications, monitoring of systems, prediction of events, analyses of data and more (Heckerman et al., 1995). Researchers from disparate fields have suggested possible applications of BNs ranging from illness classification in medicine (Andreassen et al., 1999) to geological analysis and prediction (Pedersen et al., 1998).

The parameters of a BN are determined by the use of expert opinion or by learning from data (Heckerman, 1995)(Pearl, 1988). 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 (Spiegelhalter & Lauritzen, 1990) and in the work of (Bauer et al., 1997). In this paper, we present a modified version of the online learning algorithm introduced in (Bauer et al., 1997), which we show can be used to learn the parameters of a BN up to a fixed and known accuracy. We call this algorithm Voting EM and prove convergence of the estimates of the network parameters both in the mean and in the variance. The advantages of Voting EM are that it adapts to changes in the modelled environment and can escape local maxima of the likelihood function. We further demonstrate that Voting EM is a simple approximation to the general incremental EM suggested by (Neal & Hinton, 1998), adapted for the BN parameter learning problem.

In the earlier referenced works on online learning, the issue of how to optimally weight the accumulating data is left open. With a fixed learning rate, we show that Voting EM converges, but with non-zero error, to the true parameters. When the learning rate is small, convergence is achieved with small error but at a slow rate. When the learning rate is large, convergence is fast, but the variance is large. We therefore propose a dynamic learning rate, exploiting these convergence properties of Voting EM, that adapts to changes in the modelled environment, avoids local maxima traps, and provides fast convergence to the true parameters with reduced error.

Similar paradigms for adapting the learning rate have been suggested in in the field of control and guidance (Bar-Shalom & Fortmann, 1988) and in the Neural network context (Barkai et al., 1995)(Murata et al., 1996). In the Neural

network references, the update of the learning rate is error driven. These algorithms, like the dynamic learning rate variance of Voting EM, 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 brief description of the EM(η) algorithm. In section 3, we describe and analyze Voting EM and compare it to online ML counting and incremental EM introduced in (Neal & Hinton, 1998). In section 4, we improve Voting EM by presenting an algorithm that automatically adapts the learning rate. Finally we summarize our contributions and discuss directions for future work Throughout the paper, we demonstrate the algorithms using a synthetic BN.

2 NOTATION AND BRIEF DESCRIPTION OF THE EM(η) ALGORITHM

A Bayesian network is a graphical model that encodes causal relationships among a set of variables, the strength of those relationships reflected in a set of probabilistic parameters. The task at hand 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. The notation we use follows that of (Bauer et al., 1997). Let Z_i be a node in the network that takes any value from the set $\{z^1, ..., z^{r_i}\}$. 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 (new or previously seen) data cases $D = \{y_1, ..., y_N\}$, 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 in-complete.

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

$$\begin{aligned} \tilde{\theta} = &\ argmax_\theta[F(\theta)] \\ = &\ argmax_\theta[\eta L_D(\theta) - d(\theta, \bar{\theta})] \end{aligned} \quad (1)$$

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 (which is an approximation of the KL divergence). Using a first order Taylor approximation for F and solving the maximization under the constraint that $\sum_k \theta_{ijk} = 1$ for $\forall i, j$, the following approximate solution is derived in (Bauer et al., 1997):

$$\tilde{\theta}_{ijk} = \bar{\theta}_{ijk} + \eta(\frac{E_{\bar{\theta}}[z_i^k, pa_i^j|D]}{\hat{P}(pa_i^j)} - \frac{E_{\bar{\theta}}[pa_i^j|D]}{\hat{P}(pa_i^j)} \cdot \bar{\theta}_{ijk}) \quad (2)$$

Where

$$E_\theta[z_i^k, pa_i^j|D] = \frac{1}{N}\sum_{l=1}^{N} P(z_i^k, pa_i^j|y_l, \theta) \quad (3)$$

and $\hat{P}(pa_i^j)$ is an estimate of $P_{\tilde{\theta}}(Pa_i = pa_i^j)$ given as:

$$\hat{P}(pa_i^j) = E_{\bar{\theta}}[pa_i^j|D] = \frac{1}{N}\sum_{l=1}^{N} P(pa_i^j|y_l, \bar{\theta})$$

This parameterized update rule, denoted EM(η) by (Bauer et al., 1997), can be used in both a batch and online learning mode. In batch mode, there are multiple data cases in D; iterating the update rule amounts to running an EM like algorithm (this reduces to EM if η is chosen to be 1). The Expectation step is computing the expectations as shown in Eq. 3, and the Maximization step is computing the update of the probability table entries as shown in Eq. 2. In the batch case, (Bauer et al., 1997) show that different values of η result in different speeds of convergence of this algorithm.

3 VOTING EM- DESCRIPTION AND ANALYSIS

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

$$\theta_{ijk}^T = \begin{cases} \theta_{ijk}^{T-1} + \eta[\frac{P(z_i^k, pa_i^j|y_T, \theta^{T-1})}{\hat{P}(pa_i^j)} - \frac{P(pa_i^j|y_T, \theta^{T-1})}{\hat{P}(pa_i^j)} \cdot \theta_{ijk}^{T-1}], \\ \quad \text{for} \quad \hat{P}(pa_i^j) \neq 0 \\ \theta_{ijk}^{T-1}, \\ \quad \text{otherwise.} \end{cases} \quad (4)$$

Where $\hat{P}(pa_i^j)$ is the estimated probability of the parents given the evidence and the previous estimated network and is given by the following:

$$\hat{P}(pa_i^j) = P(pa_i^j|y_T, \theta^{T-1}) \quad (5)$$

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. (Bauer et al., 1997) prove that the

convergence rate to a local maximum is faster than regular EM for the batch mode when η is greater than 1, and prove convergence for any η between 0 and 2. However, convergence was not proved for the online case. In fact, in the online case, for η greater then 1, probabilities can become negative. In this paper, we prove convergence of the Voting EM algorithm only when η is constrained to be less than 1.

3.1 ANALYSIS OF THE ONLINE LEARNING RULE

In the following analysis, if we assume that there is no missing data in the evidence vectors y_T, Eq. 4 reduces to:

$$\theta_{ijk}^{T} = \begin{cases} \eta + (1-\eta)\theta_{ijk}^{T-1}, \\ \quad \text{for } P(pa_i^j|y_T) = 1 \text{ and } P(z_i^k|y_T) = 1 \\ (1-\eta)\theta_{ijk}^{T-1}, \\ \quad \text{for } P(pa_i^j|y_T) = 1 \text{ and } P(z_i^k|y_T) = 0 \\ \theta_{ijk}^{T-1}, \\ \quad \text{otherwise} \end{cases} \tag{6}$$

This update rule is interpreted as follows. If the parents of Z_i are observed in their *j'th* configuration and if Z_i is equal to its *k'th* value, increase the value of θ_{ijk}. If the parents are observed in their *j'th* configuration but Z_i is not equal to its *k'th* value, decrease the current value. If the parents of node Z_i are not observed to be in their *j'th* configuration, do nothing.

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 following properties generally still apply. For the case of hidden nodes (that is nodes that are never observed), these theorems do not hold. Neal and Hinton (Neal & Hinton, 1998), however, have shown empirically that estimation of the parameters using an online EM approach with hidden variables yields good results in many cases.

We call the online update method of Eq. 4 in the missing data case, and Eq. 6 in the fully observed case, the Voting EM algorithm. The expression of Eq. 6 shows the reason for this name. Incoming data cause a change of η in the corresponding probabilities; the observation of the child and parents in a certain configuration is a 'vote of confidence' for that state of the nodes, and is rewarded by η, while the other states of the child (with the same parent configuration) are reduced in importance by the same weight.

Given the sequence of full evidence data from the network $D = \{y_1, ..., y_n, ...\}$ 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 $t = \{1, .., n, ...\}$, 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. 6 as:

$$X_t = (1-\eta)X_{t-1} + \eta \cdot I_t \tag{7}$$

where I_t is an indicator function given as:

$$I_t = \begin{cases} 1 & \text{with probability} \quad \theta_{ijk} = c^* \\ 0 & \text{with probability} \quad 1 - c^* \end{cases} \tag{8}$$

The process $\{I_t\}$ is an independent identically distributed (i.i.d) random process. For each instance t, I_t is a Bernoulli random variable equal to 1 with probability c^*, which is the true probability θ_{ijk} of the Bayesian network, that is $c^* = P(X_i = x_i^k | Pa_i = pa_i^j)$.

Theorem 1 *Given a discrete Bayesian Network S, a sequence of full observation vectors D, the update rule given in Eq. 7 and the constraint* $0 < \eta \leq 1$*, the following properties hold:*

1. *X_t is a consistent estimate of c^*, i.e. $E[X_t] = c^*$ as $t \to \infty$.*
2. *The variance of the estimate X_t is finite and has a limiting value of $Var[X_t] = \sigma^2 = \frac{\eta}{2-\eta} \cdot c^*(1-c^*)$ as $t \to \infty$*
3. *For $t \to \infty$ the following inequality holds: $P(|X_t - c^*| \geq q\sigma) \leq \frac{1}{q^2}$, where $q \geq 0$*

Proof: Taking the expectation of the recursion in Eq. 7, noting that $E[I_t] = c^*$ yields:

$$E[X_t] = (1-\eta)E[X_{t-1}] + \eta \cdot c^* \tag{9}$$

This is a regular difference equation that can be solved using the Z-transform or by using regular algebraic recursion methods. The solution to the recursion is given by:

$$E[X_t] = (1-\eta)^t X_0 + (1 - (1-\eta)^t) \cdot c^*, \quad t \geq 0 \tag{10}$$

where X_0 is the initial value of θ_{ijk}.

Obviously this converges only for $0 < \eta \leq 1$, and $lim_{t\to\infty}(E[X_t]) = c^*$, which proves property 1.

Proof of the second property uses a similar approach. The steps to derive the recursion for the variance are given in detail in (Starks & Woods, 1994). The final result is:

$$Var[X_t] = \frac{\eta}{2-\eta} \cdot c^*(1-c^*) \cdot (1 - (1-\eta)^{2t+2}) \tag{11}$$

Again, taking t to ∞ results in

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

The third property is simply an application of Chebychev inequality to X_t as t approaches ∞. □

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. 10 and 11 imply that $\eta = 1$ yields the fastest convergence, but also that there is no reference to the past estimations. When $\eta = 1$, the estimate of the probability oscillates between 0 and 1 based on whether, in the current sample, Z_i is equal to its *k'th* value. For smaller η's the convergence is slower, but change is smoother and less sensitive to the current sample. η can be understood as a 'forgetting bias' of the learning algorithm: the bigger it is, the less is remembered from past observations.

The effect of η on the variance is opposite to its effect on the convergence rate. The smaller the η is, the smaller the variance of the estimate. While $\eta = 1$ yields the fastest convergence, it yields the largest variance. Therefore a small η eventually yields a solution closer to the true CPT parameter.

It is important to note that the variance does not converge to 0; the estimated CPT entries oscillate around the true CPT's. Further, the magnitude of this variance depends on the value of the true probability entry (c^*). The variance is maximal for $c^* = 0.5$, and decreases as the probability approaches 1 or 0. Figure 1 shows the standard deviation(the square root of the variance) of the estimate as a function of c^* for different values of η. Although oscillation around the true probability seems undesirable, it has two advantages. First, a small but finite variance allows the algorithm to get out of a local maximum, given new evidence. Secondly, it allows adaptation to a changing environment, with η, the learning rate, controlling the speed of adaptation. Note that the ability to escape local maxima is not guaranteed and depends on the size of η relative to the shape of the likelihood function.

The third property of the theorem gives the confidence intervals of the estimated CPT's with respect to the variance of the estimate. This property can help in the choice of an acceptable η when using the Voting EM algorithm. We use this property in the adapting learning rate algorithm of section 4.

3.2 EXPERIMENT

To test the algorithm we create a BN consisting of one parent node and two children. We generate 2000 sets of variable values according to the probabilities shown in Figure 2. We then construct a test network with the same structure and randomly selected initial CPT values, and execute the online update algorithm, using the complete synthesized data. We also vary the learning rate η. Figure 3 illustrates the results, showing two of the estimated parameters as a function of the number of samples used. For all of the parameters, the test network moves in the direction of the true network. Each subfigure displays the results for two values of η. With the larger η the change from the initial guess to the real probability is faster than that with the smaller η. However, after converging (in the mean) to the true probability, the estimates with the larger η remain noisier than the estimates with the smaller η.

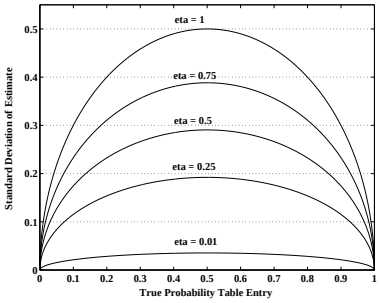


Figure 1: Effect of η and c^* on the standard deviation of the estimated probability

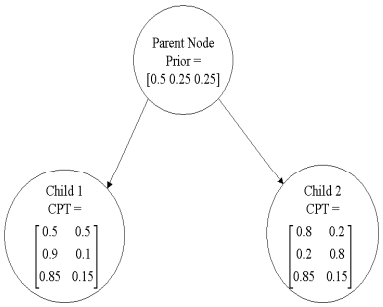


Figure 2: BN used in experiments

3.3 MAXIMUM LIKELIHOOD ESTIMATION AND VOTING EM

There is a close relation between Voting EM and the ML estimator of discrete BN parameters. The ML estimator for each of the CPT entry after seeing T samples, in the case of no missing data is simply given by:

$$\theta_{ijk}^T = \frac{N_{ijk}^T}{N_{ij}^T} \tag{12}$$

where N_{ijk}^T is the number of times the $i'th$ node was observed to be equal to its $k'th$ value and the parents equal to their $j'th$ configuration, and N_{ij}^T is the number of times the parents were equal to their $j'th$ configuration. T is the total number of observations.

The ML estimate can be computed exactly for every T in

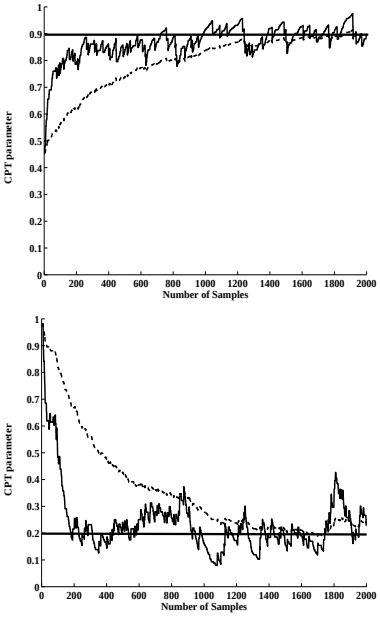


Figure 3: Results of online learning illustrated for 2 parameters. Thick straight line is the true probability, dashed and solid lines are Voting EM using $\eta = 0.01$,0.05 respectively.

an incremental way using the following update rule:

$$\theta_{ijk}^{T} = \begin{cases} \frac{1}{N_{ij}^{T}} + (1 - \frac{1}{N_{ij}^{T}})\theta_{ijk}^{T-1}, \\ \quad \text{for } P(pa_i^j|y_T) = 1 \text{ and } P(z_i^k|y_T) = 1 \\ (1 - \frac{1}{N_{ij}^{T}})\theta_{ijk}^{T-1}, \\ \quad \text{for} P(pa_i^j|y_T) = 1 \text{ and } P(z_i^k|y_T) = 0 \\ \theta_{ijk}^{T-1}, \\ \quad \text{otherwise} \end{cases} \tag{13}$$

Note that this rule has the same form as the update rule in Eq. 6, with the replacement of the fixed η by a varying form of $\frac{1}{N_{ij}^{T}}$. The effective learning rate not only varies in time, but is also specific for each row in the CPT's of the BN. Further, note that each additional observation of the parents in a particular configuration effects a progressively smaller update on the CPT entries of the child node.

Voting EM is analogous to a 'fixed memory' version of the ML estimate. In Voting EM, the fixed η can be seen as having a similar effect of fixing of N_{ij}, which amounts to looking back in time only far enough such that the count of observations is N_{ij}. Although in the static network case this simplifying assumption in Voting EM results in a noisy estimation, it is beneficial when the network parameters change over time. The ML approach adapts poorly to these changes, basically averaging the values before and after a change. Voting EM, on the other hand, with its shorter memory, adapts more quickly to the change.

In case of missing data, the ML estimator does not work in the form presented above and can be replaced by the incremental EM presented by (Neal & Hinton, 1998). The quantity $\frac{1}{N_{ij}^{T}}$ is replaced by the estimated sufficient statistics of that node. This is equivalent to the replacement of the indicator function in Eq. 8 by the inferred probability of the parents given the evidence in Eq. 4. Again, Voting EM behaves as a fixed memory version of the incremental EM.

4 ADAPTIVE LEARNING RATE FOR VOTING EM

Choosing an appropriate learning rate η for Voting EM algorithm is one of the shortcomings of the algorithm. Choosing η small yields a small variance in the estimation, but the convergence is slow. On the other hand, a large η yields fast but noisy convergence. Another shortcoming is that η is constant for all of the network parameters. A good learning rate for some parameters might prove a poor one for others. For example, when the priors of the parent node are biased towards one value, other values rarely appear. A small η results in good estimates of the CPTs for the often observed values, but the CPT's for the rarely observed events hardly move from the initial condition. Choosing a large η solves this problem for the rarely observed values, but the often observed values display large oscillations. An adaptive learning rate addresses the first shortcoming. A different learning rate per row of the CPTs, much like the ML estimation, addresses the second problem.

Adaptive learning rates have been demonstrated successfully in several works related to Neural networks (Barkai et al., 1995)(Murata et al., 1996): several options of adaptation have been used depending on the function being explored. We use a similar approach which follows from the properties of Voting EM algorithm. Intuitively, for Voting EM, 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, which happens when there is a local maximum or when the modelled environment changes. A large error can be inferred using the third property of Voting EM. To achieve a better estimation while maintaining the ability to adapt to changes in the network parameters or escape from local maximum we propose a heuristic scheme in which an initially large η is adjusted over time. Letting t denote the number of times $Pa_i = pa_i^j$ since the last time η_{ij} changed, the proposal for varying η 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 $t = 0$.

- 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. 7, where η is replaced by η^{ij}.
  2. If $|\theta_{ijk}^{t+1} - E[\theta_{ijk}^{t+1}]| > q \cdot \sigma_{ij}$ then
     increase η_{ij},
     set $t = 0$
     Else if $(1 - \eta_{ij})^t < threshold$
     decrease η_{ij}
     set $t = 0$
     Else set $t = t + 1$
  3. Read the next observation and repeat steps 1-2.

Note that $E[\theta_{ijk}^{t+1}]$ and σ_{ij} are the mean and variance of the estimated parameter. q is a positive number. q determines the confidence in the decision to increase η; from the Chebychev inequality this confidence is equal to $1 - \frac{1}{q^2}$. $threshold$ is specified by the user and reflects the acceptable convergence of the parameters. The rate at which η is increased or decreased is also specified by the user, and is discussed in the next section.

Key to this heuristic is that the learning rate both increases and decreases. From the first two properties outlined in theorem 1, the convergence of the mean and variance is a function of $(1 - \eta)^t$, where t is the number of times $Pa_i = pa_i^j$. This expression goes to 0 as t approaches ∞. Instead of waiting for infinite data, we test against the parameter $threshold$ to determine if convergence has approximated with parameterized precision. If it has been, we decrease the learning rate. As more evidence is presented, the learning rate for all the parameters becomes smaller and smaller, but remains finite. This means that the property of being able to adapt (or break out of a local maximum) is maintained, but the adaptation is slower.

If a learning rate is too slow in adapting, we should increase it. The theorem's third property implies the ability to detect changes that are faster than the current learning rate can address. By taking the absolute difference between the present estimate and its mean and comparing it to the confidence interval defined by $q \cdot \sigma_{ij}$, we can assert, with confidence $1 - \frac{1}{q^2}$, that there has been a change that warrants increasing the learning rate.

In practice the mean and variance of the parameters are approximated. 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 closed form analytical form of Eq. 11, using the 'worst case' true probability entry of 0.5 (see figure 1) which can be offset by a smaller choice of q.

4.1 EXPERIMENTAL RESULTS

We demonstrate the improved Voting EM using the same synthetic BN structure shown in Figure 2. We use three different cases: the BN parameters are static, change abruptly or slowly over time, corresponding to a static modelled environment and abrupt or slow changes in the modelled environment. Figure 4 show the adaptive Voting EM in the static case. Compared to Figure 3, the convergence is quicker and closer to the true CPT, and is comparable with the online ML estimation.

Figures 5(a)(b) show the results for the other two cases, for both the adapting Voting EM and the incremental ML estimation. As expected, the ML estimation adapts poorly to the changes in parameters. Although in the abrupt change case, it does start to adapt to the new parameters (and would adapt given infinite samples), in the case of slowly (but constantly) changing parameters, it cannot follow the changes. In contrast, the adapting Voting EM follows the abrupt change quickly and converges to a close value. Voting EM is also able to follow the slowly varying changes with good accuracy.

Figure 6 demonstrates how the learning rate η changes over time for the abrupt change case. The learning rate decreases constantly, until the change occurs (after 2000 samples). It increases rapidly shortly thereafter in response to the change, only to decrease again after no more changes are detected. Note that the change is η is not continuous in contrast to the learning rate update schedules used in (Barkai et al., 1995) and (Murata et al., 1996) which are continuous. The increase and decrease rate chosen in the experiments were exponential. Although (Barkai et al., 1995) showed that exponential was problematic for some adaptive learning rate, it works in this case because the learning rate is not updated at every step but at varying length intervals. As η_{ij} becomes smaller, the intervals between the updates become longer, therefore even an exponential update of η_{ij} is still within the limits set by (Barkai et al., 1995).

5 CONCLUSIONS AND FUTURE WORK

We have presented Voting EM, an online learning algorithm for Bayesian network parameters, based on the EM(η) algorithm suggested by (Bauer et al. 97). We have shown its convergence properties, and explained its relationship with the paradigm of Maximum-Likelihood counting. We have demonstrated the advantages of the Voting EM algorithm, namely its ability to continuously adapt to changes in the modelled environment and to escape local

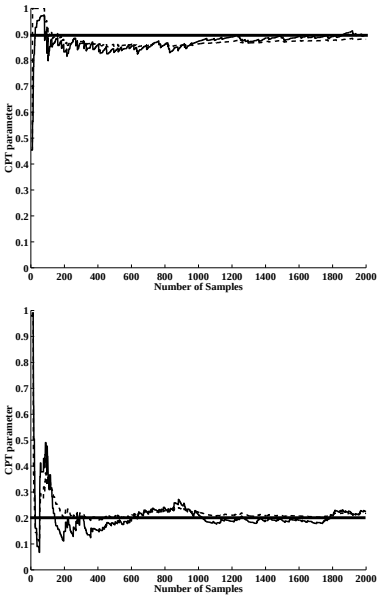


Figure 4: Adaptive learning rate for Voting EM. Thick line is the true probability. Solid line is the adaptive Voting EM, dashed line is the ML estimate

maxima of the likelihood function. In addition, by adding a mechanism to automatically adjust the learning rate η, we have obtained a new version of Voting EM that converges to the true value of the CPTs, and does so more quickly than the base version. It also adapts more quickly to changes in the modelled environment. We intend to explore further the relation between the value of the learning rate and the ability to escape local maxima of the likelihood function for both the static and the variable learning rate cases.

The synthetic data 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 (Bronstein et al., 2001). 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

Andreassen, S., Riekehr, C., Kristensen, B., Schnheyder, H., & Leibovici, L. (1999). Using probabilistic and decision-theoretic methods in treatment and prognosis modeling. *Artificial Intelligence in Medicine, 15*, 121–134.

Bar-Shalom, Y., & Fortmann, T. (1988). Tracking and data association. *Mathematics in Science and Engineering, 179*.

Barkai, N., Seung, H., & Sompolinsky, H. (1995). Local and global convergence of online learning. *Physical Review Letters, 75*, 1415–18.

Bauer, E., Koller, D., & Singer, Y. (1997). Update rules for parameter estimation in bayesian networks. *Uncertainty in Artificial Intelligence (UAI)* (pp. 3–13).

Bronstein, A., Cohen, I., Das, J., Duro, M., Kleyner, G., Mueller, M., & Singhal, S. (2001). Self-aware services: Using bayesian networks for detecting anomalies in internet-based services. *Proceedings of the IEEE/IFIP 7th International Symposium on Integrated Network Management IM-01*. IEEE Publishing.

Heckerman, D. (1995). A tutorial on learning with bayesian networks. *Report No. MSR-TR-95-06*. Microsoft Research.

Heckerman, D., Mamdani, A., & Wellman, M. (1995). Real-world applications of bayesian networks. *Communications of the ACM, 38*.

Murata, N., Muller, K., Ziehe, A., & Amari, S. (1996). Adaptive on-line learning in changing enviroments. *Advances in Neural Information Processing Systems (NIPS)* (pp. 599–605). MIT Press.

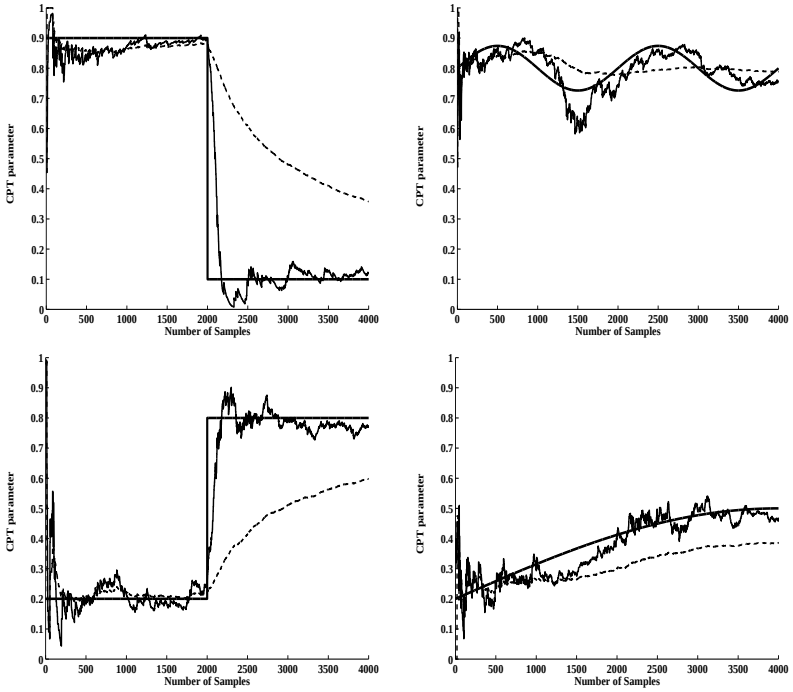
Neal, R., & Hinton, G. (1998). A view of the EM algorithm that justifies incremental, sparse and other variants. *in Learning in Graphical Models*, 355–368.

Pearl, J. (1988). *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan Kaufmann.

Pedersen, L., Apostolopoulos, D., Whittaker, W., G. Benedix, G., & Roush, T. (1998). Sensing and data classification for robotic meteorite search. *Proceedings of the SPIE - Mobile Robots XIII and Intelligent Transportation Systems*. SPIE.

Spiegelhalter, D., & Lauritzen, S. (1990). Sequential updating of conditional probabilities on directed graphical structures. *Networks, 20*, 579–605.

Starks, H., & Woods, J. (1994). *Probability, random processes, and estimation theory for engineers*. Prentice Hall.



(a) Sudden change in environment (b) Slow change in environment

Figure 5: Adaptive learning rate for Voting EM in changing environments. Thick line is the true probability. Solid line is the adaptive Voting EM, dashed line is the ML estimate.

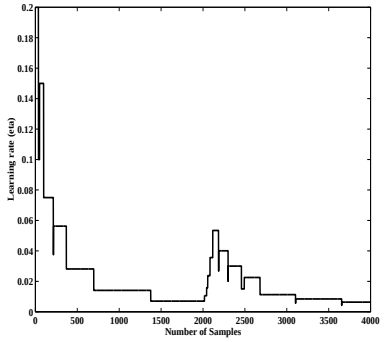


Figure 6: The learning rate η_t of one of the parameter for the sudden change in environment example 5(b).