

# Unsupervised Identification of Sequential Patterns under a Markov Assumption

Darya Chudova, Padhraic Smyth  
Information and Computer Science  
University of California, Irvine  
CA 92697-3425  
{dchudova,smyth}@ics.uci.edu

## Abstract

In this paper we investigate a sequential pattern discovery problem: unsupervised identification of recurrent sequential patterns embedded in a background process, under a Markov assumption for both pattern and background. Motif finding in DNA sequences provides a good example of a real-world problem of this nature. The problem can be formulated as an unsupervised hidden Markov learning problem (background state versus pattern state) and the EM algorithm can be used for pattern discovery. This problem appears deceptively simple, yet results from the motif-finding literature (e.g., Pevzner and Sze (2000) and Buhler and Tompa (2001)) indicate that certain versions of the problem are very challenging from a learning viewpoint. In this paper we investigate the fundamental aspects of the data-generating process and the learning framework that can make discovery “easy” or “hard.” We present a general framework for characterizing learning in this context based on the Markov-Bayes error rate. We characterize a resulting “spectrum” of difficulty in sequential pattern discovery problems and present a number of empirical results that complement the theoretical framework.

## 1 Introduction

In the sequential pattern-finding problem considered in this paper we have an observed categorical sequence defined over an alphabet of  $m$  symbols. The sequence can contain multiple occurrences of an unknown pattern subsequence embedded in a background process. For example, the pattern sequence ADDABB might be embedded in a background process of uniformly distributed and independent occurrences of the symbols A, B, C, D:

BACADBCDBBC [ADDABB] BACCDCAABDBA [ADDABB] DAACCCBAD . . .

The occurrences of the patterns in this sample sequence are enclosed in brackets.

From a practical viewpoint it is useful to allow the pattern to be non-deterministic, i.e., to have some variation. For example in motif-finding in DNA sequences symbol substitution

is common, i.e., at each position in the subsequence pattern there may be a dominant symbol (e.g., A) but other symbols may also occur in that position with some noise probability. In the results here we follow the convention of Pevzner and Sze (2000) and Buhler and Tompa (2001) for motif-finding and assume that the length  $W$  of the pattern is known a priori (although it is straightforward to generalize the results to variable-length patterns) and a noise level is specific in terms of expected number of “symbol substitution errors.” This provides a simple mechanism for controlling pattern variability in the simulation experiments described later in the paper.

In what follows we assume that the number of patterns, their locations within the sequence, and their form, are all unknown. However, we will allow specification of prior knowledge (if available) in the form of prior distributions on the expected frequency of the pattern, pattern variability and so forth. Our goal is to identify the location and nature of the patterns in an unsupervised manner, given an observed sequence and some limited prior knowledge. The computational biology motif-finding problem provides a significant example of a real-world problem that fits nicely within this general pattern-discovery class of problems; however, we are also interested in analyzing the problem in a general sense and potential applications in other areas involving event streams over time such as telecommunications and network alarm streams and customer transaction data.

## 2 A Simple Probabilistic Model for Generating Patterns

We adopt a simple generative model for this problem and assume that the state transitions within the pattern and the transitions from background to pattern and back are all first-order Markov in nature. Specifically we assume that the observed data are being generated by a  $(W + 1)$ -state hidden Markov process, where  $S_0$  is the background state (generating symbols according to some specific multinomial distribution, e.g., uniform), and  $S_1$  to  $S_W$  are the pattern states, with each state having an output multinomial distribution that is typically “tuned” to a specific symbol for that position (e.g.,  $p(A|S_1) = 0.95$ ). In effect we can view the model as a simple 2-state hidden Markov model (HMM), the first state being the background  $S_0$  and the second state being the disjunction of pattern states  $S_1, \dots, S_W$ .

The transition probability from background to pattern governs the frequency with which patterns are observed. The transitions within the pattern states influence the variability of the patterns. The simplest case is a model where each state  $S_j$  has outgoing transition probability 1 to the next state  $S_{j+1}$  and state  $S_W$  has transition probability 1 to the background. This imposes a deterministic fixed-length state sequence, with a specific state for each position in the pattern. For each pattern occurrence any variability in the pattern is captured by noise in the output multinomial distributions for each state (this is the model we will investigate in this paper). More generally the pattern can be viewed as a hidden stochastic finite state machine, i.e., pattern states could have arbitrary transition probabilities with “exit” transitions to the background state allowing for the generation of variable length patterns. Even more generally the pattern and background could be higher-order Markov, or entirely non-Markov (e.g., stochastic context-free grammars). In this paper, however, we

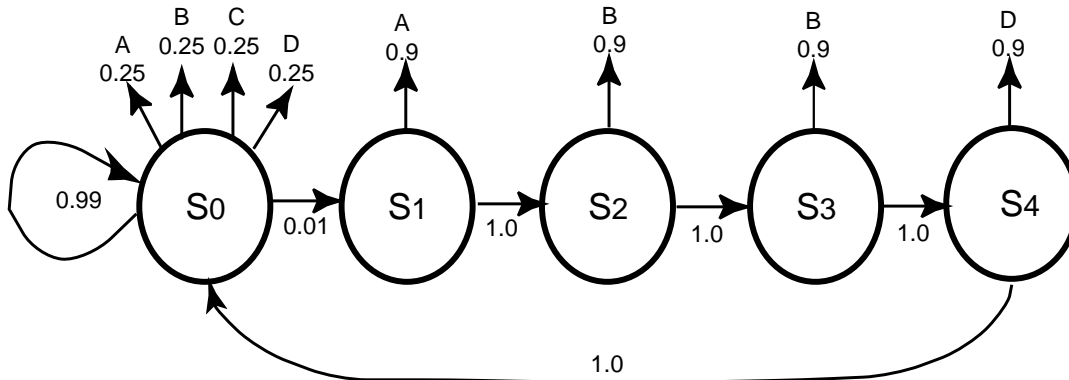


Figure 1: Example of an HMM representing a pattern whose most likely instantiation is *ABBD* in a uniform background sequence.

focus attention on the simple first-order hidden Markov model since it can be viewed as a baseline for sequential pattern learning.

Figure 1 shows a model for generating patterns of length 4 (the pattern *ABBD*) with a 5-state HMM. The background state (marked  $S_0$ ) is characterized by a high entropy distribution for the background frequencies of symbols in the sequences. Emissions in each of the  $W$  pattern states have low entropy with the probability mass concentrated on the consensus symbol in the corresponding position of the pattern. There is a high probability of 0.99 of self-transition to the background state and a small chance (0.01) of entering the first state of the pattern. The transitions between states within the pattern are modeled using the first-order Markov assumption, and the corresponding transition matrix is rather sparse, allowing only transitions to the next pattern state.

Under the above model assumptions, the pattern finding problem is reduced to that of estimating the parameters of the corresponding HMM. Even though there exist well-known techniques for solving this problem (such as the expectation-maximization (EM) procedure), relatively little appears to be known specifically about the success rate of these learning procedures on pattern-finding tasks of the type described above, in terms of the quality of the learned models, susceptibility to local maxima of the likelihood surface, and so forth.

Our primary goal in this paper is to investigate how easy or hard it is to learn such patterns in cases where the general Markov form and some aspects of structure (e.g., pattern length) are assumed known. In what follows we investigate the factors that influence learnability of any given Markov pattern problem, and characterize a spectrum of “simple” and “hard” problems. In a sense, the results provide a gauge for how difficult it may be to identify sequential patterns in real-world applications, in situations where the Markov model may not represent the true data-generating process. For example the distribution of distances between patterns in the sequence might not follow the geometric distribution dictated by a first-order Markov model, but instead might require a more specific distribution (resulting in a semi-Markov model). If learning is hard in the case even when we know the correct functional form of the model, and where we have significant prior knowledge, the

implication may be that unsupervised learning in real-world situations may be quite difficult (at least from a standard HMM learning viewpoint).

## 3 What Makes the Learning Problem Hard?

### 3.1 Factors that Influence Learnability

The complexity of learning a particular pattern can be measured along multiple dimensions. For example, consider the following factors that independently influence the difficulty of the problem:

- the expected length of the pattern (the longer the pattern is, the easier it should be to separate it from the background);
- the degree of determinism within the pattern (the more variation it exhibits, the harder it should be to find it);
- the frequency of occurrence of the pattern (the more often it occurs, the easier it should be to learn it);
- the distinction between the pattern and the background (the closer the pattern is to the background, the harder it should be to learn it)

Rather than characterizing learnability along each of these dimensions, we instead look at a single characteristic, the Bayes error rate, that provides a more fundamental indication of how hard it is to learn a pattern. The Bayes error rate can be used to place different problems on the same scale in terms of their complexity, or equivalently, to predict the difficulty of learning any specific pattern given knowledge of the Bayes error rate.

### 3.2 The Bayes Error Rate for Pattern Finding

The Bayes error rate is a well-known concept in classification: it is the minimum error rate for a given problem, achieved by using the true model to make the optimal Bayes decision for the class variable given an observed feature vector. It is a function of the overlap of the class-conditional density functions. If there is significant overlap, there can be significant ambiguity about the class identity of a particular measurement, and the Bayes error rate may be quite high. The important point is that the Bayes error rate is an inherent theoretical property of a classification problem, independent of any actual trained classification model or any actual training data set, providing an absolute lower-bound on achievable error rate for *any* classifier.

We can extend the notion of the Bayes error rate to the hidden Markov model described in the last section. In effect, our HMM is a two-class classification problem (where the background and pattern states are the two classes) with Markov dependence among the classes. Intuitively, the Bayes error rate in this case corresponds to the average number of symbols that would be misclassified in the limit using an optimal Bayes classifier to classify the states using observed data from an infinitely long realization of the chain. Technically

we would need a more precise definition, such as requiring that the Markov chain is recurrent (i.e., that no states are absorbing), but for the purposes of this paper the intuitive “limiting” definition is sufficient. The optimal classifier in this case (on a per-symbol basis) is of course to classify the hidden state at position  $i$  as being “background” or “pattern” based on which of the two probabilities  $p(\text{label } i = \text{background}|\mathbf{o})$  or  $p(\text{label } i = \text{pattern}|\mathbf{o})$  is greater, where  $\mathbf{o} = \{o_1, \dots, o_N\}$  is the observed symbol sequence. These probabilities are calculated using the well-known forward-backward algorithm for HMMs, and the parameters of the data-generating process are assumed known (i.e., we are using the true model for classification).

This “Markov-Bayes error rate” can be thought of as the average number of mistakes that a Bayes-optimal decision rule makes. Intuitively we will make classification mistakes whenever a background symbol looks more similar to a pattern state than the background state, given the context, or vice-versa. The Bayes error rate in principle indicates how difficult it is even in the “perfect knowledge” case to isolate the occurrences of the pattern from the sea of background, and thus characterizes the difficulty of the original unsupervised learning problem.

It appears to be an open problem as to whether the Bayes error rate can be calculated in closed form as a function of the model parameters (the transition probabilities and multinomial output densities) even for the simplest two-class problems (see Lee (1974) and Chu (1970) for some bounds on the Bayes error for *sub-optimal* decision rules). For the results described below we empirically estimate the Bayes error on long sequences, by the average number of per-symbol classification errors made by the forward-backward algorithm using the true parameters, relative to the simulated true state sequence.

Figure 2 illustrates that the Bayes error varies appropriately with the “learning complexity factors” discussed earlier in Section 3.1. Notice, that the Bayes error rate is bounded from above by the baseline error (the probability of seeing a pattern letter) which itself varies depending on pattern length and the probability of entering the pattern. To bring the Bayes error rate onto the same scale for comparisons across different values of these parameters, we look at the ratio of the Bayes error rate and the baseline error, in effect a normalized Bayes error rate. This normalized Bayes error rate has a simple interpretation: if all pattern symbols are optimally classified as “pattern” and all background as “background” we get a zero normalized rate. If all pattern symbols are optimally classified as background (and all background symbols as background), we get a normalized error rate of 1. If exactly half of the pattern symbols are optimally classified as background (and all background symbols as background) we get a normalized error rate of 0.5.

The first plot shows how this normalized Bayes error rate increases with an increasing number of corrupted symbols per pattern: as expected, error increases as the variability in patterns increases (there is more potential for confusion between background and patterns since the pattern class is essentially increasing in size). The second plot shows the decrease in relative error as the probability of entering the pattern increases, i.e., as patterns become more frequent the normalized Bayes error rate is reduced. The third plot shows an abrupt decrease in error as the pattern length increases, for the case of deterministic patterns (no symbol errors) with lengths varying from 4 to 13. Patterns of size 4 are hard to identify (even deterministic ones), while patterns of length 7 or longer are quite easy to find. Note that all of these results are model-specific in that the actual values of the normalized Bayes error rate (the y-axis values) depend on other parameters (the transition probabilities, pattern

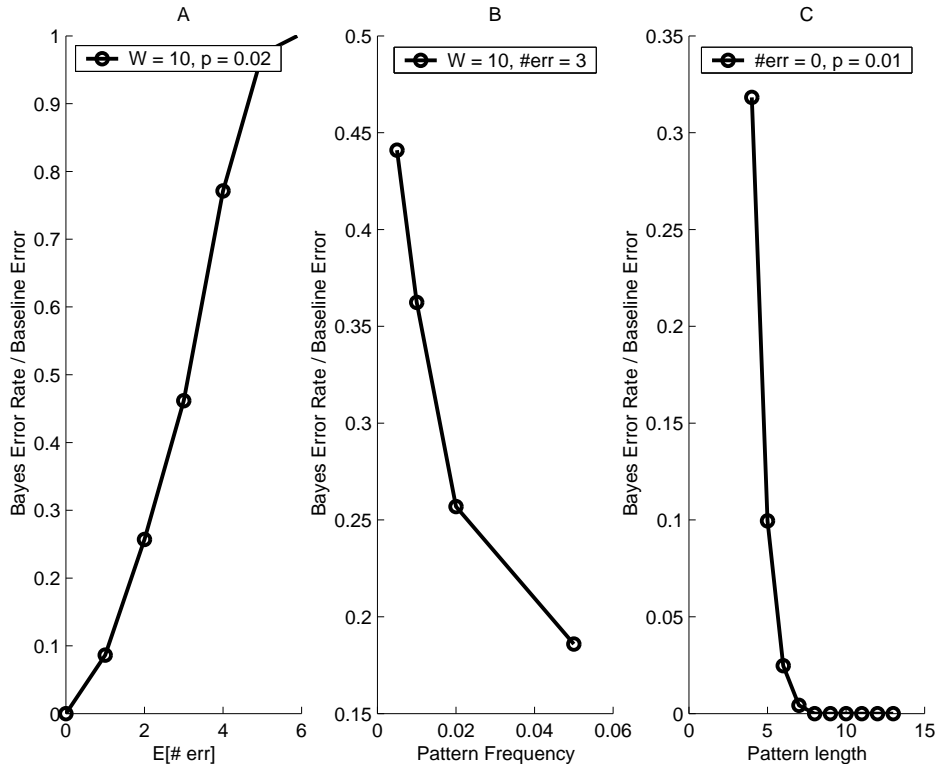


Figure 2: Influence of pattern variability (A), pattern length (B) and pattern frequency (C) on the normalized Bayes error rate.

frequency, and pattern symbol noise). Nonetheless, the general trends are quite clear and although interactions between the different parameters are somewhat hidden, the Bayes error rate appears to be quite useful if we seek a single general number that characterizes learning difficulty. We expand on this theme further in the next section.

## 4 Experimental Results

We investigate three specific questions in the experiments below:

1. How well does the proposed problem complexity measure (the Bayes error rate) correlate with the actual success of learning the model from data?
2. Can prior knowledge help in learning?
3. For the cases where EM has a difficult time finding the patterns can we aid EM in finding the right region of parameter space?

### 4.1 Experimental Setup

We performed the experiments on several simulated problems. In general the data was simulated from an HMM with uniformly distributed emissions in the background, and a pattern structure similar to that shown in Figure 1. Each problem is characterized by:

- The expected pattern length (in our experiments,  $W = 8, 9, 10$ ),
- The expected number of corrupted letters in each occurrence of the pattern ( $E[Err] = 0, \dots, 4$ ),
- The probability of entering a pattern from the background ( $p \in \{0.005, 0.01, 0.02, 0.05\}$  (this determines the relative frequency at which patterns occur),
- The size of the observable alphabet ( $m = 4$ ).

These simulated problems are a probabilistic formulation of the challenge problems proposed by Pevzner and Sze (2000) to provide a common benchmark for testing and comparison of pattern-finding algorithms. In each of the standard challenge problems the number of corrupted symbols per pattern is fixed, while in our probabilistic framework we only specify the expected number of erroneous symbols.

## 4.2 Correlation of Bayes Error Rate and Learnability

Other authors have tested their pattern-finding algorithms on the Pevzner and Sze (2000) challenge problems. For example, Buhler and Tompa (2001) used a random projections approach and found some of the problems to have enough ambiguity that the patterns are in effect "non-learnable." In these ambiguous problems the amount of variation in each pattern makes them look as if they were produced by the background by pure chance. Table 1 aggregates results from both Buhler and Tompa (2001) and our work on probabilistic versions of the same problems. Column 1 contains the length of pattern. Column 2 contains the total number of errors in each pattern (deterministic for Buhler and Tompa (2001), and an expected value for our simulations). Column 3 indicates whether the deterministic version of the problem is learnable (results from Buhler and Tompa (2001)). The fourth column contains the empirically-estimated normalized Bayes error rate that characterizes the difficulty of each problem. The higher this number is, the closer the optimal classification strategy is to random guessing. There is clearly a high correlation between Bayes error rate and learnability as determined in the experiments of Buhler and Tompa (2001), i.e., they were able to learn all patterns with normalized Bayes error below 0.5, but unable to learn any patterns with normalized Bayes error above 0.6.

## 4.3 Evaluating the Quality of a Learned Model

Evaluation of the learned models is not a simple task even when that the true data-generating model is known. The difficulty stems from the fact that several different models can explain the data equally well (for example, we could obtain an equivalent model by arbitrarily permuting the states of the learned model). So, comparing the models directly in parameter space is not adequate in this case. To deal with these difficulties we use out-of-sample negative per-symbol log likelihood score (or, simply, "l-score"). We use this score to estimate how well the learned model predicts or explains unseen sequences generated by the same true pattern model. Note that we omit the background symbols from these test sequences so that

Table 1: Properties of simulated data: challenge problems and their probabilistic analogs.

Pattern length	E[# errors]	Learnable	Bayes Error Rate / Baseline
9	2	No	0.85
10	2	Yes	0.36
11	2	Yes	0.18
11	3	No	0.73
12	3	Yes	0.42
13	3	Yes	0.55
13	4	No	0.91
14	4	Yes	0.53
15	4	Yes	0.43
15	5	No	0.55
16	5	Yes	0.30
17	5	Yes	0.30

they do not dominate the score, and use only the patterns for scoring (in effect our metric is measuring the quality of how well the patterns are learned, not the background).

When evaluating the quality of the model with an l-score, it is useful to relate the score to some natural upper and lower bounds so that the quality of the estimated model can be easier to discern. A lower bound is given by the l-score of the true model while a possible upper bound is given by the *l - score* of the background model (or a uniform distribution). It is useful to bring the l-score to an understandable scale by working with its exponent. The exponent of the l-score (also known as the *perplexity*) can be interpreted as the effective output alphabet size (Jelinek, 1997), and is the level of compression (in nats or bits) that can be achieved by using the model for data transmission. Using the exponent of the l-score, a simple upper bound is the number of distinct output symbols (4 in our experiments).

Figure 3 shows the out-of-sample perplexity (smaller is better, less uncertainty in the predictions) as a function of the Bayes error rate, across a wide variety of simulated pattern problems. It is again clear from this figure that as the Bayes error rate increases it becomes increasingly difficult for the EM-based HMM learning algorithm to discover the correct patterns.

For these experiments we used both maximum likelihood and maximum a posteriori (MAP) estimates of parameters obtained with standard EM algorithm. The number of iterations was bounded from above by 500, and the algorithm was allowed to run while the relative increase in the likelihood was greater than  $10^{-5}$ . In the simplest case, we used multiple (10) random restarts to overcome the problem of local maxima. We investigate how the choice of initial conditions can influence the quality of obtained solution later in section 4.5. We discuss MAP estimation and the selection of priors in section 4.4.

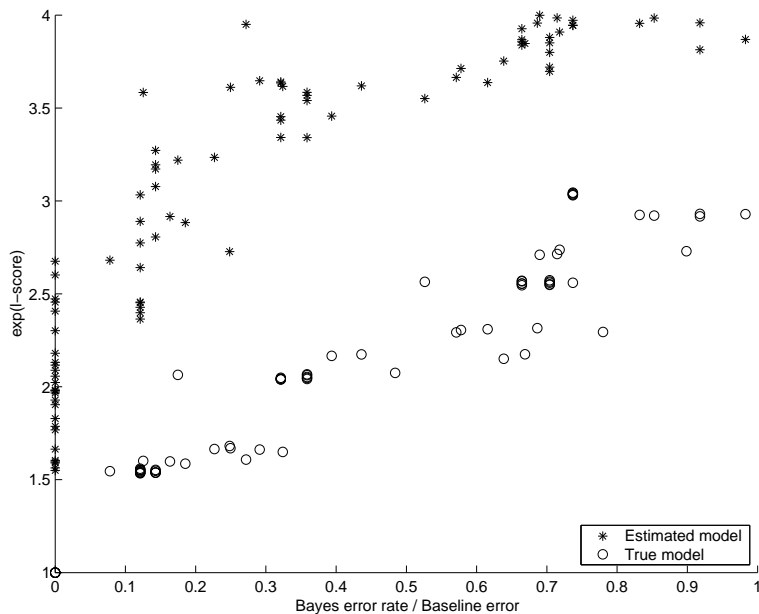


Figure 3: Quality of the estimated models in terms of out-of-sample perplexity as a function of the Bayes error rate associated with original problem.

#### 4.4 The Value of Prior Knowledge

Whenever prior knowledge is available about the model, incorporating it into the MAP estimation procedure can simplify the search space for EM and, thus, lead to better solutions. For the pattern finding problem, we identified three different aspects of prior knowledge:

- State transition probabilities have a dominant linear structure (background  $\rightarrow$  first pattern letter  $\rightarrow$  second pattern letter  $\rightarrow \dots \rightarrow$  last pattern letter  $\rightarrow$  background). This can be ensured by specifying a Dirichlet prior on each of the transition distributions, where the priors are biased towards a dominant successor state for each state.
- Emissions from the background state are proportional to the overall empirical frequency of each letter in the training data. This, again, can be conveyed to the algorithm by a strong Dirichlet prior on the corresponding distribution (in effect this is an empirical Bayes approach, i.e., using a prior that is itself estimated from the data).
- Emissions from the pattern states are not known a-priori, but these distributions should be nearly deterministic, i.e., have low entropy. This type of prior knowledge can be enforced by the use of a so-called entropic prior (Brand, 1999) that encourages low-entropy solutions.

Incorporating prior information into the models and employing MAP estimation during EM training generally tended to lead to more accurate models and acceleration of convergence. This is illustrated in Figure 4. The first scatter plot shows out-of-sample perplexity

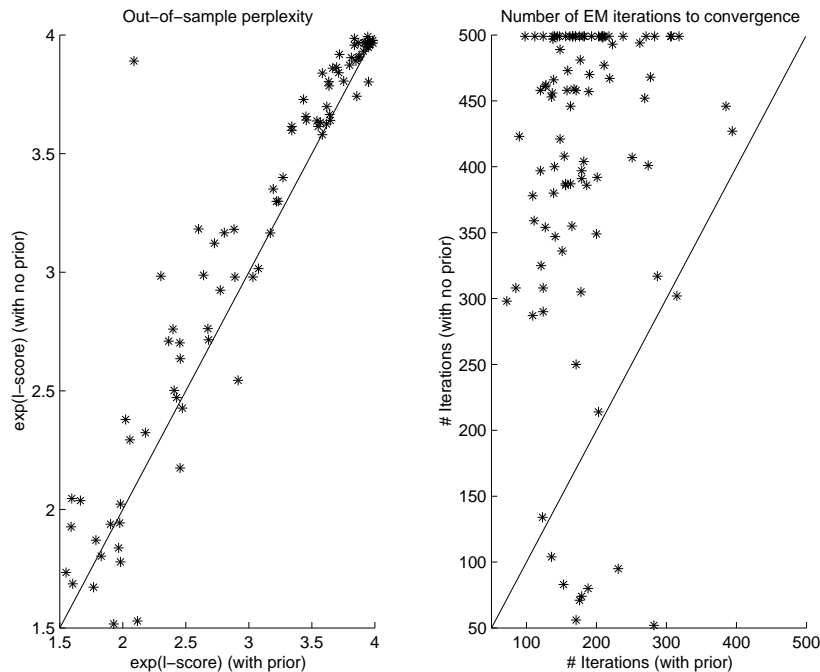


Figure 4: Value of prior knowledge: more accurate models and faster convergence.

for the models learned without any prior knowledge versus the perplexity of the models learned with a prior on the same input data. It can be seen that simple models (the ones with the low values of l-score) can be learned successfully with or without prior. However, for the more complex models, the introduction of prior knowledge leads to systematic improvements in predictive accuracy (i.e., the points are generally located above the diagonal on the plot). In fact, in some cases EM completely fails to recover the pattern if used in the maximum likelihood setting, while MAP estimates with informative priors succeed at the same task. The second scatter plot in Figure 4 shows the number of iterations required for EM to converge under ML and MAP scenarios (the number of EM iterations was upper bounded by 500 in this experiment). Again, the points are significantly above the diagonal, indicating a noticeable speed-up in convergence.

## 4.5 Initialization and Local Maxima in EM

A common problem we found with HMM training is that of encountering a bad local maximum of the likelihood/posterior surface, especially when the probability of a patterns is low or the amount of the training data is insufficient. We followed the convention of performing multiple (10) restarts of EM from different random initial conditions. When looking at the models generated by as many as 400 consecutive iterations of EM, we noticed that sometimes the HMM would capture only some part of the pattern (left/ right/middle) correctly, but not the whole pattern.

For example, it might find the beginning of the pattern, but place the first letter into the third pattern state, so that it does not have “enough” successor states to represent the whole pattern. In such cases, allowing the EM algorithm to run for more iterations or changing

the prior to be less restrictive doesn't lead to significant improvements in the likelihood: the EM algorithm is trapped in a local maximum. The resulting solution has a low score and fails to identify the whole pattern, even though what appears to be the hardest part of the job is done - the HMM generally knows where the patterns are located throughout the training sequence (although it can have a significant number of false alarms since it finds shorter patterns by random chance). However, EM fails to recover the complete pattern and thus produces low overall scores on unseen patterns. Other papers in the HMM literature have also reported the existence of multiple suboptimal local maxima in HMM learning from finite data, even for learning relatively simple models (e.g., see Bengio and Frasconi, 1996).

We were able to remedy this somewhat by taking the solution obtained from running EM with random initial conditions, shifting the block of pattern states one position at a time to the right or to the left, and using these transitions/emissions as initial conditions for another run of EM. (This is somewhat similar to a search heuristic used in the MEME algorithm of Bailey and Elkan (1995) for motif-finding). In a sense, we just help the EM to jump over to another potential peak in parameter space by choosing this specific set of initial conditions. A "mountain-climbing" analogy is as follows: from the base of the mountains, mountain A might appear highest to a mountaineer. Upon climbing mountain A the mountaineer now can clearly see from his or her new perspective that peaks B and C are in fact taller. The "shifting" heuristic could be viewed as a way to fly our "EM-mountaineer" over to a potential more promising peak.

Table 2: Improving the quality of extracted patterns by initializing EM with perturbed solutions from previous restarts.

Model type	E[# errors]	Trn size	Pattern	NIter(EM)	exp (l-score)
True Model	1	2000	ADBBDDCC	-	1.53837
Restarts w/shifting	-	-	ADBBDDCC	169	2.78136
Random restarts	-	-	DBBDDCC?A	274	3.17138
True Model	1	4000	BCCDBBBDD	-	1.5533
Restarts w/shifting	-	-	BCCDBBBDD	49	2.21253
Random restarts	-	-	CDBBBDDBC?	170	2.53697
True Model	2	2000	CDBDDCABAC	-	2.03842
Restarts w/shifting	-	-	CDBDDCABAC	116	3.02485
Random restarts	-	-	BDDCABACD?	387	3.34601
True Model	2	4000	ADAADBBBAC	-	2.07573
Restarts w/shifting	-	-	ADAADBBBAC	128	2.94405
Random restarts	-	-	AD?BBACCCD	308	3.43739

In Table 2 we show several examples of these suboptimal solutions and how they can be improved by allowing EM to search for the optimal alignment of states to pattern positions. The table contains a total of 4 patterns, each separated by a horizontal line. The first column indicates the type of model used to produce the pattern: either the true model, or the best one obtained by 10 random restarts (Random restarts), or the one where shifted

patterns were used as the initial conditions (Restarts w/shifting) in addition to 5 random restarts. The second column contains the expected number of errors in each pattern, the third column indicates the size of the training data. All experiments were conducted with a pattern frequency equal to 0.01. The fourth column contains the most likely emissions in each state if they exceed a threshold value (between 0.5 and 0.65 depending on the maximum emission in the true model) and a question mark otherwise. The fifth column indicates the number of iterations required for EM to converge (note that starting from the shifted pattern leads to quicker convergence). The last column contains the out-of-sample exponentiated l-score (ranging from 1 to 4, the smaller the better). Allowing the algorithm to search for the optimal alignment of pattern positions to states leads to a significant drop in this score. The out-of-sample predictive score obtained with shifting is often well outside the range of scores obtained from any of the random initial conditions.

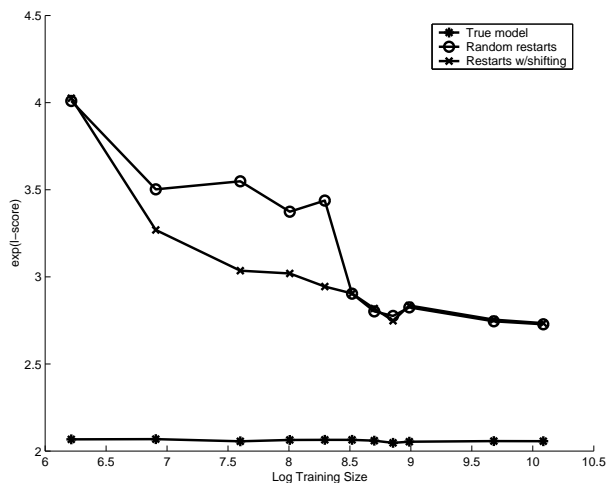


Figure 5: Increase in out-of-sample accuracy due to EM initialization strategy for various sizes of training data.

The idea of aiding EM in exploring the parameter space by shifting the solutions is especially promising when little data is available for training. Our experiments show that as the amount of training data increases, the pure random restart strategy for initializing the EM works just as well as the shifting method. However, for the more interesting cases of smaller data sizes, where the algorithm has a difficult time detecting the pattern, the distinction between these two strategies is critical. Fig. 5 shows how the perplexity of the models learned with different initialization algorithms varies as a function of the training data size. For this experiment, we used the true model that generates patterns of length 10 with 2 errors on average. The probability of pattern occurrence was set to 0.01. The amount of training data varied from 500 to 24000 symbols, and different models had similar out-of-sample scores starting from about 8000 training symbols. For the smallest data set (500 symbols), none of the models could in fact capture the pattern, yielding maximum possible perplexity. However, employing pattern shifting heuristic for the initialization of EM allowed to bring the perplexity of the models down significantly in all data sets between 1000 and 5000 symbols.

## 5 Discussion and Conclusions

In this paper we described preliminary results on the learnability of patterns embedded in a background process, under a general Markov assumption. We saw that the Bayes error rate plays a critical role in learnability. Of course in practice the Bayes error rate is unknown (it is a function of the true model), so these results are primarily of theoretical interest. Nonetheless, they suggest that discovery of subtle patterns may require a combination of prior knowledge (it helps to have some idea of what one is looking for), clever search algorithms (hill-climbing with random restarts may not be sufficiently accurate on its own), and significant amounts of data (if the patterns are relatively rare we need sufficient occurrences of them to be able to reliably identify them). Future work on this topic includes the derivation of closed-form analytical expressions (or bounds) for the Bayes error rate as a function of problem parameters, improved search techniques to overcome local maxima given limited training data, and the development of application-specific priors and search heuristics for specific problems.

## Acknowledgements

The work described in this paper was supported by the National Science Foundation under Grant IRI-9703120.

## References

- Bailey, T. and Elkan C. (1995) Unsupervised learning of Multiple Motifs in Biopolymers Using Expectation Maximization. *Machine Learning Journal*, 21, pp. 51-83.
- Bengio and Frasconi (1995) Diffusion of Context and Credit Information in Markovian Models. *Journal of Artificial Intelligence Research* 3, p. 249-270.
- Brand, M. (1999) Structure Learning in Conditional Probability Models via an Entropic Prior and Parameter Extinction. "Neural Computation", V. 11, 5, pp. 1155-1182
- Buhler, J., and Tompa, M. (2001), *Proceedings of the Fifth Annual International Conference on Computational Molecular Biology (RECOMB01)*, Montreal, Canada, 2001.
- Chu, J.T., (1974) Error Bounds for a Contextual Recognition Procedure. *IEEE Trans. Elect. Comput.*, Vol. C-20, No. 10
- Jelinek, F. (1997) Statistical methods for speech recognition. MIT Press, Cambridge, Massachusetts 1997
- Lee, E.T. (1974) Bounds and approximations for error probabilities in character recognition. *Proceedings of the International conference on Cybernetics and Society*, pp. 324 - 329.
- Pevzner, P. A., and Sze, S.-H. (2000) Combinatorial approaches to finding subtle signals in DNA sequences. *Proceedings of the 8th International Conference on Intelligent Systems for Molecular Biology (ISMB'2000)*, pp. 269-278.