SYNTHESIS OF HIDDEN MARKOV MODELS BASED ON FINITE SAMPLE PATHS AND APPLICATIONS TO COMPUTATIONAL BIOLOGY

M. Vidyasagar

Advanced Technology Centre
Tata Consultancy Services
6th Floor, Khan Lateefkhan Building
Hyderabad 500 001, INDIA
sagar@atc.tcs.co.in

Keywords: Computational biology, hidden Markov models, coding regions, protein classification

Abstract

In this paper, we study the problem of modelling a given stationary stochastic process using a hidden Markov model (HMM). In particular, we show how to construct a HMM for an arbitrary stochastic process so as to match perfectly its statistics up to a prespecified order, and to match optimally its statistics of higher order. This approach is applied to two problems in computational biology, namely: distinguishing between the coding and non-coding regions of a Prokaryote genome, and classifying a protein into a small family of proteins.

1 Introduction

In the past, hidden Markov models (HMM's) have been successfully applied to the problem of classifying a new protein into one of several previously determined groups of proteins; see for example [5]. However, this approach leads to HMM's with very large-sized state spaces. Specifically, in order to apply this method one first has to carry out a so-called optimal gapped alignment of the protein family. The complexity of optimal gapped alignment is *exponential* in the number of strings being aligned, so usually this step is carried out "by hand."[1] If the length of the optimally gap-aligned sequences is N, then the method of [5] results in a HMM with a state space of size $3N + 2$. A part of the reason for the large-sized state spaces is that the approach in some sense attempts to realize a non-stationary process using a stationary HMM. As a result, in this approach one attempts to estimate something like $72N$ parameters based on a comparable number of data points. Hence it is desirable to have an alternate approach that results in a smaller-sized state space.

In this paper, we take an entirely different approach to the problem of synthesizing a HMM to model a given stationary stochastic process. In [1], a procedure is presented for synthesizing a HMM for a stochastic process assuming values in a finite output space, provided the *entire* statistics of the stochastic process are known, and *assuming* that the process is generated by a HMM. Thus the results of [1] do not answer the fundamental question, namely: When does a given stationary stochastic process have a HMM realization? The results of [1] are akin to a *procedure*, rather than a solution to the realization problem.

The solution to the full realization problem is not presented here. Rather, we study the problem of *partial realization* of a stochastic process using a HMM. That is, we present a procedure for matching the k-tuple frequencies of the output process, where k is some prespecified integer. Since in practice the only thing available is a *finite-length* sample path of a stochastic process, such an approach more closely mirrors the actual modelling problems in realistic situations. In case the quantities being matched are the *exact* frequencies of various k-tuples, then the problem being solved here can be thought of as the "partial realization" problem for the given stochastic process. On the other hand, if (as often happens in practice), the quantities being matched are the *empirical* frequencies of k-tuples based on a finite-length sample path of the given stochastic process, then it does not make any sense to attempt matching these empirical frequencies beyond a certain length k. Moreover, even for the specified integer k, there is no sense in matching these empirical frequencies *exactly*. Rather, one should explore ways of matching these frequencies only *approximately*, in the process greatly reducing the size of the state space. Such a procedure is presented here.

The method presented here will be applied to two problems in computational biology, namely: distinguishing between coding and non-coding regions of a Prokaryote genome, and classifying a protein. The results will be presented in the conference.

2 Problem Formulation

Suppose $\mathcal{M} := \{1, \ldots, m\}$ is a finite set, and that $\{\mathcal{Y}_t\}_{t \geq 0}$ is a stationary stochastic process assuming values in $\mathcal{M}$. It is desired to construct a hidden Markov model (HMM) for the stochastic process. Thus, it is desired to find an integer n, a state space $X := \{1, \ldots, n\}$, a state transition matrix $A \in [0,1]^{n \times n}$, and an output transition matrix $B \in [0,1]^{n \times m}$ such

[1] See a companion paper in this conference for background material on computational biology.

that the following statements are true:

1. A is nonnegative and a column-stochastic matrix. In other words, if $\mathbf{e}_n$ denotes the column vector of all 1's with n rows, then $A\mathbf{e}_n = \mathbf{e}_n$. In such a case, it is known (see e.g., [3]) that there exists at least one stationary probability vector π. Thus $\pi \in [0,1]^n$, $\sum_{j=1}^n \pi_j = 1$, and $\pi A = \pi$.

2. Suppose $\{\mathcal{X}_t\}$ is a stationary Markov process evolving on the finite set $X = \{1, \ldots, n\}$, with transition probability matrix A and initial distribution π. Thus $\{\mathcal{X}_t\}$ satisfies the following properties:

   - $\Pr\{\mathcal{X}_t | \mathcal{X}_{t-1}, \ldots, \mathcal{X}_0\} = \Pr\{\mathcal{X}_t | \mathcal{X}_{t-1}\},\ \forall t \geq 0$. (Markov property)
   - $\Pr\{\mathcal{X}_t | \mathcal{X}_{t-1}\} = \Pr\{\mathcal{X}_1 | \mathcal{X}_0\},\ \forall t \geq 0$. (Stationarity of the Markov chain)
   - $\Pr\{\mathcal{X}_1 = j | \mathcal{X}_0 = i\} = a_{ij}$. ($A$ is the state transition matrix)

3. $\Pr\{\mathcal{Y}_t | \mathcal{X}_i, i \leq t, \mathcal{Y}_j, j < t\} = \Pr\{\mathcal{Y}_t | \mathcal{X}_t\},\ \forall t \geq 0$.

4. $\Pr\{\mathcal{Y}_t = j | \mathcal{X}_t = i\} = b_{ij},\ \forall i, j$.

Up to now we have just defined a HMM with state transition matrix A and output transition matrix B; we have not said what it means for this HMM to "model" the given stochastic process. For this purpose, suppose the Markov process $\{\mathcal{X}_t\}$ is started off with the initial distribution π. Thus $\Pr\{\mathcal{X}_0 = i\} = \pi_i,\ \forall i$. The stationarity of π implies that $\Pr\{\mathcal{X}_t = i\} = \pi_i,\ \forall i,\ \forall t$. Now let $\mathcal{M} := \{1, \ldots, m\}$, and define the matrices $M^{(l)}, l \in \mathcal{M}$, as follows:

$$M_{ij}^{(l)} = \Pr\{\mathcal{X}_1 = j, \mathcal{Y}_1 = l | \mathcal{X}_0 = i\}.$$

It is obvious that $m_{ij}^{(l)} = a_{ij} b_{jl}$. Therefore we have that

$$m_{ij}^{(l)} \geq 0\ \forall i, j, l, \text{ and } \sum_{l=1}^{m} M^{(l)} = A.$$

Now we can define what it means for the stochastic process $\{\mathcal{Y}_t\}$ to be modelled by the HMM. As is customary, let $\mathcal{M}^*$ denote the set of all strings (including the empty string) over the finite alphabet $\mathcal{M}$. Suppose $\mathbf{u} \in \mathcal{M}^*$, and to be specific, suppose that $\mathbf{u} = u_1 u_2 \ldots u_s$. Then the frequency of the string $\mathbf{u}$ occuring in any sample path of the stochastic process is denoted by $f_{\mathbf{u}}$. Now the stochastic process $\{\mathcal{Y}_t\}$ is said to be modelled by the HMM if

$$f_{\mathbf{u}} = \pi M^{(u_1)} M^{(u_2)} \ldots M^{(u_s)} \mathbf{e}_n,\ \forall \mathbf{u} \in \mathcal{M}^*, \qquad (2.1)$$

where $\mathbf{e}_n$ denotes the column vector consisting of n one's. In other words, the HMM models $\{\mathcal{Y}_t\}$ if it faithfully reproduces the frequency of *all* strings. Thus the HMM realization problem is simply this: Given the stationary stochastic process $\{\mathcal{Y}_t\}$, find conditions under which there exists a HMM that models it.

The preceding can be thought of as the problem of "perfectly" realizing the stochastic process $\{\mathcal{Y}_t\}$ using a HMM. The "partial realization" problem can be defined as follows: Suppose that an integer k is specified, and it is desired to construct a HMM for the given stochastic process $\{\mathcal{Y}_t\}$ such that the frequencies of k-tuples of the HMM matches those of the stochastic process. Note that there is no requirement that the frequencies of strings longer than k match in the two cases. Is it possible to find such a HMM, and if so, how would one go about it? Before proceeding further, it should be noted that if frequencies of k-tuples in both cases match, then so do frequencies of s-tuples for all $s \leq k$.

3 Motivation

There are two motivations for studying this problem. First, suppose that one is in possession of the frequencies of *all* strings of arbitrary length; that is, one has perfect knowledge of the statistics of a stochastic process. By opting to match only the first so many frequencies, it might be possible to obtain HMM's of *greatly reduced order*, i.e., state spaces with a much smaller number of states than is produced by a HMM that perfectly reproduces all the statistics. This problem is referred to here as the "partial realization problem" for HMM's. Second, and more important in practice, suppose that the so-called frequencies of various strings are computed empirically on the basis of a sample path of *finite length*, say T. Then (assuming that $m \ll T$), the empirical frequency of an s-tuple is computed on the basis of approximately T samples. Eventually, as k becomes larger and larger, the quantity m^k becomes comparable to T. When this happens, empirical frequencies of k-tuples will cease to be meaningful, and it makes no sense to try and match them. Thus it makes sense to attempt a match of only the k-tuple frequencies such that $m^k \ll T$. But there is an additional twist here. In the partial realization problem, the underlying assumption is that the statistics of the underlying stochastic process are known perfectly, and we are opting to reproduce them perfectly only up to order k. However, in the present problem, the empirical frequencies need not be the same as the true frequencies, and it is therefore not necessary to match them *exactly*. By matching the empirical frequencies only *approximately*, it may be possible to reduce the dimension of the state space still further.

A very specific practical motivation for studying the above problem is that of protein classification. Hidden Markov models have been applied with some success to the problem of classifying a new protein into one of several families, each of which consists of several similar proteins. See [5] for details. In this application, one begins with r distinct families of proteins. Now a new protein is specified in terms of its amino acid sequence (i.e., its primary structure), and it is desired to classify it as belonging to one of these r families. To solve this problem, the following approach is adopted in [5]. For each of the r families, a corresponding HMM is set up. Now for the new protein to be classified, the likelihood that each of the r HMM's could have produced this particular amino acid se-

quence is computed. The protein is assigned to the family for which the likelihood is the highest.

To synthesize the HMM, let N denote the length of the gap-aligned sequences within a particular family. Then the HMM constructed in [5] is shown below.

Figure 1: Hidden Markov Model of [5]

Now let us look at the kinds of numbers involved. Actually, each family of proteins $\mathcal{S}_i$ consists of "optimally gap-aligned" versions of all the amino acid sequences of proteins belonging to $\mathcal{S}_i$. Since there are 20 amino acid symbols and one gap symbol, the output space in this instance has 21 symbols, i.e., $m = 21$. Typically there might be 500 to 1,000 proteins in a given family, with the gap-aligned length being of the order of 200. In such a case, the HMM of [5] would consist of 602 states.

There are several other noteworthy features of the above HMM. In most papers on HMM's, it is assumed that the underlying Markov process is irreducible. In contrast, in the present instance the Markov process is most definitely *reducible*, since there is no possibility of a transition from any of the three states in position i to any states in positions prior to i. This structure is adopted because the likelihoods of insertion, deletion and mutation are in general different at different locations along the amino acid chain. Thus it appears that the approach of [5] actually attempts to model a *nonstationary* stochastic process using a stationary HMM but with a very large-sized state space.

The second thing to notice about the HMM is that it has a *huge* number of parameters to be estimated. Since the number of states is $3N + 2$, and there are three successor states at each time (except for the end state), the total number of transition probabilities to be estimated is $9N + 3$. Similarly, the number of output probability vectors to be estimated is $63N$. Hence the total number of constants to be estimated is about $72N$. On the other hand, if there are k proteins within a given family, then the total number of data points is about kN.

Let us take some typical values. If $N \approx 1,000$ which is the typical length of a multiple gapped alignment, then the total number of quantities to be estimated is about $72N$, or about 72,000. On the other hand, the number of proteins within a given family could be as small as 500, meaning that the number of sample points is just about 500,000. It is not always desirable to try and estimate so many probabilities on the basis of so few data points.

4 The Complete Realization Problem

Given integers k, l, let us define the matrix $F_{k,l} \in [0,1]^{m^k \times m^l}$ as follows. Suppose $\mathbf{i} \in \mathcal{M}^k, \mathbf{j} \in \mathcal{M}^l$. Then the $(\mathbf{i}, \mathbf{j})$-th element of $F_{k,l}$ is the frequency of the $(k+l)$-tuple $\mathbf{ij}$. The convention is that the rows of $F_{k,l}$ are numbered in lexicographical order starting from the *beginning*, whereas the columns of $F_{k,l}$ are numbered in lexicographical ordering starting from the *end*. If either k or l equals zero, the corresponding set $\mathcal{M}^k$ or $\mathcal{M}^l$ is deemed to consist of the empty string.

To illustrate this definition, suppose $m = 2$. Then

$$F_{1,2} = \begin{bmatrix} f_{111} & f_{112} & f_{121} & f_{122} \\ f_{211} & f_{212} & f_{221} & f_{222} \end{bmatrix}.$$

Similarly,

$$F_{0,3} = [f_{111} \;\; f_{112} \;\; f_{121} \;\; f_{122} \;\; f_{211} \;\; f_{212} \;\; f_{221} \;\; f_{222}].$$

Note that both $F_{1,2}$ and $F_{0,3}$ consist of triplet frequencies, since $1 + 2 = 0 + 3$. However, the arrangement of the entries is different.

Next, we introduce the matrix

$$H_{k,l} := \begin{bmatrix} F_{0,0} & F_{0,1} & \dots & F_{0,l} \\ \vdots & \vdots & \vdots & \vdots \\ F_{k,0} & F_{k,1} & \dots & F_{k,l} \end{bmatrix}.$$

Note that $F_{0,0}$ is taken as the number 1. The matrix $H_{k,l}$ has $1 + m + \ldots + m^k$ rows and $1 + m + \ldots + m^l$ columns. If we do not put any bounds on k, l and simply form the above matrix for arbitrarily large values of k, l, we will get an infinite matrix, which we denote by H. If $\mathbf{i} \in \mathcal{M}^k$ for some k, then there is a unique row of H whose left-most element is the frequency of the string $\mathbf{i}$. We call this the $\mathbf{i}$-th row of H. Similarly, we can also speak of the $\mathbf{j}$-th column of H for each $\mathbf{j} \in \mathcal{M}^l$ for each l.

Theorem 4.1 *For each k, we have that* $\mathrm{Rank}(H_{k,k}) = \mathrm{Rank}(F_{k,k})$.

Proof: Fix $\mathbf{i} \in \mathcal{M}^r$, and observe that, for each $\mathbf{j} \in \mathcal{M}^s$, we have

$$f_{\mathbf{ij}} = \sum_{l=1}^{m} f_{l\mathbf{ij}}.$$

Therefore the $\mathbf{i}$-th row of H is the sum of rows $1\mathbf{i}, 2\mathbf{i}, \ldots, m\mathbf{i}$. This argument shows that, for each k, we have

$$\begin{aligned} \mathrm{Rank}(H_{k,k}) &= \mathrm{Rank} \begin{bmatrix} F_{0,0} & F_{0,1} & \dots & F_{0,k} \\ \vdots & \vdots & \vdots & \vdots \\ F_{k,0} & F_{k,1} & \dots & F_{k,k} \end{bmatrix} \\ &= \mathrm{Rank}[F_{k,0} \;\; F_{k,1} \;\; \dots \;\; F_{k,k}]. \end{aligned}$$

Now the above argument can be applied columnwise. For $\mathbf{i} \in \mathcal{M}^r, \mathbf{j} \in \mathcal{M}^s$ we have

$$f_{\mathbf{ij}} = \sum_{l=1}^{m} f_{\mathbf{ij}l}.$$

Therefore column $\mathbf{j}$ of $F_{k,s}$ is the sum of columns $\mathbf{j}1, \mathbf{j}2, \ldots, \mathbf{j}m$ of $F_{k,s+1}$. This shows that

$$\mathrm{Rank}[F_{k,0} \;\; F_{k,1} \;\; \ldots \;\; F_{k,k}] = \mathrm{Rank}(F_{k,k})$$

and leads to the desired conclusion that $\mathrm{Rank}(H_{k,k}) = \mathrm{Rank}(F_{k,k})$. ■

Corollary 4.1 *Suppose H has finite rank, say r. Then there exists a smallest integer k such that*

$$\mathrm{Rank}(F_{k,k}) = \mathrm{Rank}(F_{k+l,k+l}) = r, \; \forall l > 0. \tag{4.1}$$

Proof: This result is a direct corollary of the theorem. Let us examine the sequence of integers $\mathrm{Rank}(H_{k,k})$ as k increases. Since $H_{k,k}$ is a submatrix of $H_{k+1,k+1}$, this sequence is nondecreasing. Now suppose H has finite rank, say r. Then we can have $H_{k+1,k+1} > \mathrm{Rank}(H_{k,k})$ only finitely many times. This shows that there exists a smallest k such that

$$\mathrm{Rank}(H_{k+l,k+l}) = r, \; \forall l > 0.$$

Now Theorem 4.1 shows that $\mathrm{Rank}(H_{k+l,k+l}) = \mathrm{Rank}(F_{k+l,k+l})$. This leads to the desired conclusion. ■

Theorem 4.2 *Suppose H has finite rank, say r, and choose the smallest integer k such that $\mathrm{Rank}(F_{k,k}) = r$. Then there exists a matrix $C \in \mathbb{R}^{m^k \times m^{k+1}}$ such that*

$$F_{k,k+1} = F_{k,k}C. \tag{4.2}$$

Proof: Define the matrix $\bar{H}_{k,l}$ as follows.

$$\bar{H}_{k,l} := \begin{bmatrix} F_{k,l} & F_{k,l+1} & \cdots \\ F_{k+1,l} & F_{k+1,l+1} & \cdots \\ \vdots & \vdots & \vdots \end{bmatrix}.$$

Since $\bar{H}_{k,k}$ is a submatrix of H, it follows that $\mathrm{Rank}(\bar{H}_{k,k}) \leq r$. On the other hand, since $F_{k,k}$ is a submatrix of $\bar{H}_{k,k}$, it is clear that equality must hold, i.e., $\mathrm{Rank}(\bar{H}_{k,k}) = r$. In particular, it follows that

$$r = \mathrm{Rank}([F_{k,k} \;\; F_{k,k+1}]) = \mathrm{Rank}(F_{k,k}).$$

This is another way of stating the conclusion, i.e., that there exists a matrix $C \in \mathbb{R}^{m^k \times m^{k+1}}$ such that (4.2) holds. ■

Note that if $F_{k,k}$ has full rank, then C is unique, but otherwise there might exist more than one C such that the above holds. The following results hold whether or not C is unique.

Theorem 4.3 *Suppose H has finite rank, say r, and choose k as in Theorem 4.2. Choose C such that (4.2) holds, and partition C as*

$$C \in \mathbb{R}^{m^k \times m^{k+1}} = [C^1 \ldots C^m], \; C^j \in \mathbb{R}^{m^k \times m^k} \; \forall j. \tag{4.3}$$

Then

$$F_{k,k+2} = F_{k,k}[C^1C^1 \;\; C^1C^2 \ldots C^mC^{m-1} \;\; C^mC^m],$$

and in general, for each $l > 0$, we have

$$F_{k,k+l} = F_{k,k}[C^1C^1 \ldots C^1 \quad \ldots \quad C^mC^m \ldots C^m], \tag{4.4}$$

where the matrices are l-fold products arranged in lexicographic order with respect to the last component.

Theorem 4.4 *Suppose H has finite rank, and choose C such that (4.2) holds. Let $\mathbf{u} = u_1 \ldots u_l \in \mathcal{M}^l$. Then*

$$f_{\mathbf{u}} = F_{0,k}C^{u_1} \ldots C^{u_l}\mathbf{e}_{m^k}, \tag{4.5}$$

where $\mathbf{e}_{m^k}$ denotes the $m^k \times 1$ column vector consisting of all one's.

To state the next theorem we introduce the notion of mixing.

Definition 4.1 *The stochastic process $\{\mathcal{Y}_t\}$ is said to be* **mixing** *if*

$$\max_{\mathbf{u},\mathbf{v} \in \mathcal{M}^k} \left| f_{\mathbf{u}} \cdot f_{\mathbf{v}} - \sum_{\mathbf{w} \in \mathcal{M}^l} f_{\mathbf{uwv}} \right| \to 0 \text{ as } l \to \infty. \tag{4.6}$$

The above definition can be interpreted as follows: Clearly the summation is the frequency of a string of length $2k + l$ beginning with $\mathbf{u}$ and ending with $\mathbf{v}$ (and we are indifferent as to what is in-between). The condition (4.6) states that asymptotically this frequency approaches the product of $f_{\mathbf{u}}$ and $f_{\mathbf{v}}$. Thus, asymptotically, the beginning and the end of a string become independent.

Theorem 4.5 *Suppose the matrix H has finite rank, and in addition, the stochastic process $\{\mathcal{Y}_t\}$ is mixing in the sense of Definition 4.1. Choose C to be the minimum-norm solution of the equation $F_{k,k+1} = F_{k,k}C$. Partition C as above and define $S = \sum_{i=1}^{m} C^i$. Then*

1. *The spectral radius of S equals 1.*
2. *The m^k-dimensional row vector $F_{0,k}$ is a row eigenvector of S corresponding to the eigenvalue $\lambda = 1$.*
3. *The matrix S has only one eigenvalue of magnitude 1, and that is a simple eigenvalue at $\lambda = 1$.*

Thus the matrix S behaves almost like the state transition matrix of a Markov chain, except that it is not necessarily a nonnegative matrix. Compare the expressions (2.1) and (4.5). If the matrices C^i were all to be nonnegative and if their sum S were to be column-stochastic, then in fact we would have solved the HMM realization problem. For this reason, we refer to the set of matrices $C_1, \ldots, C_m$ as a "proto-realization" of the process $\{\mathcal{Y}_t\}$.

5 The Partial Realization Problem for HMM's

5.1 Partial Realization with Perfect Matching

The preceding developments assume that the complete statistics of the stochastic process $\{\mathcal{Y}_t\}$ are known, that the matrix H has finite rank, and that it is desired to construct a HMM for the stochastic process. Now let us focus on the so-called 'partial realization' problem. Suppose that the frequencies of all k-tuples $f_{\mathbf{ij}}, \mathbf{i}, \mathbf{j} \in \mathcal{M}^k$ are given.[2] The objective is to construct a HMM that perfectly reproduces these frequencies. It turns out that the choice of such an HMM is not unique. So while we are at it, in case the frequencies of $(k+1)$-tuples are also specified, it is possible to choose the HMM so as to approximate these frequencies optimally.

Recall that we are specified the frequencies of all k-tuples. Equivalently, we are given the m^k-dimensional row vector $F_{0,k}$. Thus, if we choose nonnegative matrices $C^1, \ldots, C^m$ such that (i) the sum $S := \sum_{i=1}^m C^i$ is column-stochastic, and (ii) S has $F_{0,k}$ as a row eigenvector corresponding to the eigenvalue $\lambda = 1$, then we have a solution to the partial realization problem. We can turn this around and do the following: Choose S to be an *arbitrary* nonnegative and column-stochastic matrix of order $m^k times m^k$, which has $F_{0,k}$ as a row eigenvector corresponding to the eigenvalue $\lambda = 1$. Choose $C^1, \ldots, C^m$ to be any nonnegative $m^k \times m^k$ matrices that add up to S. Then these m matrices constitute a solution to the HMM realization problem.

Note that there are infinitely many choices for S, leading in turn to infinitely many HMM realizations, all with a state space of dimension m^k. Let $\{\mathbf{Z}_t\}$ denote the output process of such a HMM. Then the statistics of $\{\mathbf{Z}_t\}$ match those of the original process $\{\mathcal{Y}_t\}$ only up to order k, but not necessarily beyond k. One of these HMM's will correspond to the rather uninteresting case where the output process $\{\mathbf{Z}_t\}$ is k-dependent, that is, where $\mathbf{Z}_{k+1}$ is independent of $\mathbf{Z}_0$. But in general the output of the model will not be k-dependent.

We can actually use the freedom to choose the matrices C_i so that we not only match the k-tuple frequencies *perfectly*, but also match the $(k+1)$-tuple frequencies *optimally*. Suppose we are also given the vector $F_{0,k+1}$. Then we can solve the following quadratic programming problem:

$$\min_C \parallel F_{0,k+1} - F_{0,k}C \parallel^2,$$

subject to the following constraints:

$$C^l_{i,j} \geq 0 \; \forall l \in \mathcal{M}, i, j \in \mathcal{M}^k,$$

$$F_{0,k}\left[\sum_{l=1}^m C^l\right] = F_{0,k},$$

$$\sum_{l=1}^m \sum_{j \in \mathcal{M}^k} C^l_{i,j} = 1, \; \forall i \in \mathcal{M}^k.$$

[2] Note that the frequencies of the k-tuples uniquely determine the frequencies of all tuples of shorter length.

It is also possible to match higher-order statistics optimally, but the resulting optimization problem would no longer be a quadratic programming problem.

5.2 Partial Realization with Imperfect Matching

The partial realization procedure described above always results in a HMM with a state space of size m^k. With the preceding background, we can see how it is possible to reduce the size of the state space further. Specifically, suppose the frequency vector $F_{0,k}$ is given. Then we can set some threshold, and simply throw away all components that are smaller than this threshold, and assign those weights to the remaining components. In this way, we would obtain a nonnegative vector π with n rows where n is less than m^k. Of course, the smaller we make n, the greater the mismatch between the approximate vector π and the true frequency vector $F_{0,k}$. Now we simply find a column-stochastic matrix S of dimension $n \times n$ that has π as a row eigenvector corresponding to the eigenvalue $\lambda = 1$ and proceed as above. Note that the above procedure can be incorporated into the quadratic programming approach (and thereby match the frequencies of $(k+1)$-tuples) by constraining some components of C to be zero.

6 Discussion

In this paper, we have studied the problem of constructing hidden Markov models (HMM's) for a stochastic process taking values over a finite alphabet. We have shown how to construct HMM's that match the observed frequencies of a single sample path, either perfectly or imperfectly. This approach can be applied to the problems of protein classification and to distinguishing between coding and non-coding regions of a Prokaryote genome. Actual results will be presented in the conference.

In the final conference version of the paper, complete results will be presented.

References

[1] B. D. O. Anderson, "The realization problem for hidden Markov models," *Math. Control, Sig., Sys.*, 12(1), 80-120, 1999.

[2] P. Baldi and S. Brunak, *Bioinformatics: A Machine Learning Approach*, MIT Press, Cambridge, MA, 2001.

[3] A. Berman and R. J. Plemmons, *Nonnegative Matrices*, Academic Press, New York, 1979; reprinted in the series *Classics in Applied Mathematics*, No. , SIAM Publications, Philadelphia, PA, .

[4] W. J. Ewens and G. R. Grant, *Statistical Methods in Bioinformatics*, Springer-Verlag, New York, 2001.

[5] A. Krogh, M. Brown, I. S. Mian, K. Sjölander and D. Haussler, "Hidden Markov models in computational biology: Applications to protein modeling," *J. Mol. Biol.*, 235, 1501-1531, 1994.