Inferring Mixtures of Markov Chains

Tuğkan Batu*[1], Sudipto Guha[2], and Sampath Kannan**[2]

[1] Department of Computer Sciences, University of Texas, Austin, TX.
batu@cs.utexas.edu
[2] Department of Computer and Information Science, University of Pennsylvania, Philadelphia, PA.
{sudipto, kannan}@cis.upenn.edu

Abstract. We define the problem of inferring a "mixture of Markov chains" based on observing a stream of interleaved outputs from these chains. We show a sharp characterization of the inference process. The problems we consider also has applications such as gene finding, intrusion detection, etc., and more generally in analyzing interleaved sequences.

1 Introduction

In this paper we study the question of inferring Markov chains from a stream of interleaved behavior. We assume that the constituent Markov chains output their current state. The sequences of states thus obtained are interleaved by some switching mechanism (such as a natural mixture model). Observe that if we only observe a (probabilistic) function of the current state, the above problem already captures hidden Markov models and probabilistic automata, and is computationally intractable as shown by Abe and Warmuth [1]. Our results can therefore be interpreted as providing an analytical inference mechanism for one class of hidden Markov models. The closely related problem of learning switching distributions is studied by Freund and Ron [10].

Thiesson et al. study learning mixtures of Bayesian networks and DAG models [16, 17]. In related works, learning mixtures of Gaussian distributions are studied in [6, 3]. The hidden Markov model, pioneered in speech recognition (see [14, 4]) has been the obvious choice for modeling sequential patterns. Related Hierarchical Markov models [11] were proposed for graphical modeling. Mixture models have been studied considerably in the context of learning and even earlier in the context of pattern recognition [8]. To the best of our knowledge, mixture models of Markov chains have not been explored.

Our motivation for studying the problem is in understanding interleaved processes that can be modeled by discrete-time Markov chains. The interleaving process controls a token which it hands off to one of the component processes

* This work was supported by ARO DAAD 19-01-1047 and NSF CCR01-05337.
** This work was supported by NSF CCR98-20885 and NSF CCR01-05337.

at each time step. A component process that receives the token makes a transition, outputs its state, and returns the token. We consider several variants of the interleaving process. In the simplest, tokens are handed off to the component processes with fixed probabilities independent of history. A more general model is where these hand-off probabilities are dependent on the chain of the state that was generated last. The following are potential applications of our framework.

- The problem of *intrusion detection* is the problem of observing a stream of packets and deciding if some improper use is being made of system resources.[3] We can attempt to model the background (good) traffic and the intrusive traffic being different Markov processes. We then model the overall traffic as a random mixture of these two types of traffic. The problem of *fraud detection* arises in this context as well; see [7, 18, 12, 9] for models on intrusion and fraud detection.

- Given a genome sequence (a sequence from a chromosome) the problem is to locate the regions of this sequence (called *exons*) that collectively represent a gene. Again, precise defining characteristics are not known for exons and the regions in between them called *introns*. However, a number of papers have attempted to identify statistical differences between these two types of segments. Because the presence of a nucleotide at one position affects the distribution of nucleotides at neighboring positions one needs to model these distributions (at least) as first-order Markov chains rather than treating each position independently. In fact, fifth-order Markov chains and Generalized Hidden Markov Models (GHMMs) are used by gene finding programs such as GENSCAN [5].

- The problem of validation and mining of log-files of transactions arises in e-commerce applications [2, 15]. The user interacts with a server and the only information is available at the server end is a transcript of the interleaved interactions of multiple users . Typically searches/queries/requests are made in "sessions" by the same user; but there is no obvious way to determine if two requests correspond to the same user or different ones. Complete information is not always available (due to proxies or explicit privacy concerns) and at times unreliable. See [13] for a survey of issues in this area.

The common theme of the above problems is the analysis of a sequence that arises from a process which is not completely known. Furthermore the problem is quite simple if *exactly one* process is involved. The complexity of these problems arise from the interleaving of the two or more processes due to probabilistic linearization of parallel processes rather than due to adversarial intervention.

[3] We do not have a precise definition of what constitutes such intrusion but we expect that experts "will know it when they see it."

1.1 Our Model

Let $M^{(1)}, M^{(2)}, \ldots, M^{(k)}$ be Markov chains where Markov chain $M^{(l)}$ has state space V_l for $l = 1, 2, \ldots, k$. The inference algorithm has no *a priori* knowledge of which states belong to which Markov chains. In fact, identifying the set of states in each chain is the main challenge in the inference problem.

One might be tempted to "simplify" the picture by saying that the process generating the data is a *single* Markov chain on the cross-product state space. Note, however, that at each step we only observe one component of the state of this cross-product chain and hence with this view, we are faced with the problem of inferring a hidden Markov model. Our results can therefore be interpreted as providing an analytical inference mechanism for one class of hidden Markov models where the hiding function projects a state in a product space to an appropriate component. We consider two mixture models.

- In the simpler mixture model, we assume that there are probability values $\alpha_1, \ldots, \alpha_k$ summing to 1 such that at each time step, Markov chain $M^{(i)}$ is chosen with probability α_i. The choices at different time steps are assumed to be independent. *Note that the number k of Markov chains (and, necessarily, the mixing probabilities) are not known in advance.*

- A more sophisticated mixture model, for example, in the case of modeling exons and introns, would be to assume that at any step the current chain determines according to some probability distribution which Markov chain (including itself) will be chosen in the next step. We call this more sophisticated model the *chain-dependent mixture model*.

We assume that all Markov chains considered are *ergodic* which means that there is a k_0 such that every entry in M^k is non-zero for $k \geq k_0$. Informally, this means that there is a non-zero probability of eventually getting from any state i to any state j and that the chain is *aperiodic*. We also assume that the *cover time*[4] of each of the Markov chains is bounded by τ, a polynomial in the maximum number of states in any chain — these restrictions are necessary to estimate the edge transition probabilities of any Markov chain in polynomial time. Furthermore, since we cannot infer arbitrary real probabilities exactly based on polynomially many observations, we will assume that all probabilities involved in the problem are of the form p/q where all denominators are bounded by some bound Q. As long as we are allowed to observe a stream whose length is some suitable polynomial in Q, we will infer the Markov chains exactly with high probability.

[4] The cover time is the maximum over all vertices u of the expected number of steps required by a random walk that starts at u and ends on visiting every vertex in the graph. For a Markov chain M, if we are at vertex v we choose the next vertex to be v' with probability $M_{vv'}$.

1.2 Our Results

We first consider the version of the inference problem where the Markov chains have pairwise-disjoint state sets in the chain-dependent mixture model. In this model, the interleaving process is itself a Markov Chain whose cover time we denote by τ_1. We show the following result in Section 3.

Theorem 1. *For Markov chains over disjoint state sets and the chain-dependent mixture model, we can infer a model of the source that is observationally equivalent, to the original source, i.e., that the inferred model generates the exact same distribution as the target model. We make the assumption that α_{ii}, i.e., the probability of observing the next label from the same Markov process is non-zero. We require a stream of length $O(\tau^2\tau_1^2Q^2)$, where Q is the upper bound on the denominator of any probability represented as a fraction, and τ_1, τ are upper bounds on the cover times of the interleaving and constituent processes, respectively.*

We can easily show that our upper bound in Theorem 1 is a polynomial function of the minimum length required to estimate each of the probabilities. Next, we prove that it is necessary to restrict to disjoint-state-set Markov chains to achieve polynomial-time inference schemes.

Theorem 2. *Inferring chain dependent mixture of Markov chains is computationally intractable. In particular, we show that the inference of two state probablistic automata (with variable alphabet size) can be represented in this model.*

The question about the inference of simple probabilistic mixture of Markov chains with overlapping state spaces arises naturally as a consequence of the above two theorems. Although we do not get as general a result as Theorem 1, we show the following in Section 4, providing evidence towards a positive result.

Theorem 3. *For two Markov chains on non-disjoint state sets, we can infer the chains in the simple mixture model with a stream of length $O(\text{poly}(n))$ where n is the total number of states in both chains, provided that there is a state i_s that occurs in only one chain, say $M^{(1)}$, and satisfies the technical condition:*

$$\textit{either } M^{(1)}_{i_s j} > S_1(j) \textit{ or } M^{(1)}_{i_s j} = 0 \quad \textit{for all states } j$$

where S_1 is the stationary distribution of $M^{(1)}$.

To make sense of the technical condition above consider the special case where the Markov chain is a random walk in a graph. The condition above is satisfied if there is a state that occurs in only one graph that has a small degree. This condition sounds plausible in many applications.

2 Preliminaries and Notation

We identify the combined state space of the given Markov chains with the set $[n] \stackrel{\text{def}}{=} \{1, \ldots, n\}$. Suppose $M^{(1)}, \ldots, M^{(k)}$ are finite-state ergodic Markov chains

in discrete time with state space $V_l \subseteq [n]$ corresponding to $M^{(l)}$. We consider two possible cases—one where the state spaces of the individual Markov chains are disjoint and the other where they are allowed to overlap. Suppose each Markov chain outputs its current state after it makes a transition. The first and the simpler mixture model that we consider generates streams with the alphabet $[n]$ in the following manner. Let $\alpha_1, \ldots, \alpha_k$ be such that $\sum_l \alpha_l = 1$. Assume that initial states are chosen for each of the Markov chains arbitrarily. The stream is generated by interleaving the outputs of Markov chains $M^{(1)}, \ldots, M^{(k)}$. For each stream element, an index l is chosen according to the distribution defined by α_l's. Then, $M^{(l)}$ is allowed to make a transition from its previous state and its output is appended to the stream. Define $S_l(i)$ to be the probability of i in the stationary distribution of $M^{(l)}$.

A more general mixture model we explore is where the probability distribution for choosing the Markov chain that will make a transition next is dependent on the chain of the last output state. For $i, j \in [n]$, we use α_{ij} to denote the probability that the control is handed off to Markov chain that j belongs to when the last output was i. Note that for states i_1, i_2 in the same chain, $\alpha_{i_1 j} = \alpha_{i_2 j}$ and $\alpha_{j i_1} = \alpha_{j i_2}$ for all states $j \in [n]$. Since we use this mixture model only for Markov chains with disjoint state spaces, α_{ij}'s are well defined.

We will sometimes denote the interleaving process by $\mathcal{I}$. Then we can denote the entire interleaved Markov process by a tuple, $\langle M^{(1)}, M^{(2)}, \ldots, M^{(k)}; \mathcal{I} \rangle$.

Let $\hat{T}_i$ denote the (relative) frequency of occurrence of the state i. Given a pattern $\langle ij \rangle$ let $\hat{T}_{ij}$ be the frequency of j occurring immediately after i. Likewise define $\hat{T}_{ijs}$ to be the frequency of the pattern $\langle ijs \rangle$.

We define the problem of inferring mixtures of Markov chains as given a stream generated as described above, constructing the transition matrices for the underlying Markov chains as well as the mixing parameters. The problem reduces to identifying the partitioning of the state space—since given a partitioning we can project the data on each of the partitions and identify the transition probabilities.

It is also clear that if two Markov chain mixtures produce each finite length stream with equal probability, then they are indistinguishable by our techniques. Consequently we need a notion of *observational equivalence*.

Definition 1. *Two interleaved processes* $\mathcal{P} = \langle M^{(1)}, \ldots, M^{(k)}; \mathcal{I} \rangle$ *and* $\mathcal{P}' = \langle M'^{(1)}, \ldots, M'^{(k')}; \mathcal{I}' \rangle$ *are* observationally indistinguishable *if there is an assignment of initial state probabilities to each chain of* $\mathcal{P}'$ *for every assignment of initial states to the chains in* $\mathcal{P}$ *such that for any finite sequence in* $[n]^*$ *the probability of the sequence being produced by* $\mathcal{P}$ *is equal to the probability of the sequence being produced by* $\mathcal{P}'$.

Note that we have no hope of disambiguating between observationally equivalent processes. We provide an example of such pairs of processes:

Example. Let process $\mathcal{P} = \langle M^{(1)}, M^{(2)}; \mathcal{I} \rangle$ where $M^{(1)}$ is the trivial single-state Markov chain on state 1 and $M^{(2)}$ is the trivial single-state Markov chain on state 2. Let $\mathcal{I}$ be the process which chooses each chain with probability $\frac{1}{2}$ at each step.

Let process $\mathcal{P}' = \langle M'; \mathcal{I}' \rangle$ where $\mathcal{I}'$ trivially always chooses M' and M' is a 2-state process which has probability $\frac{1}{2}$ for all transitions. $\mathcal{P}$ and $\mathcal{P}'$ are observationally indistinguishable.

Definition 2. *A Markov chain $M^{(1)}$ is defined to be reducible to* one-step mixing *if for all $i, j \in V_1$ we have $M^{(1)}_{ij} = S_1(j)$, i.e., the next state distribution is also the stationary distribution.*

Proposition 1. *If $M^{(1)}$ is reducible to one-step mixing, where $|V_1| = z$, the interleaved process $\mathcal{P} = \langle M^{(1)}, \ldots, M^{(k)}; \mathcal{I} \rangle$ is observationally indistinguishable from $\mathcal{P}' = \langle M^{(1)}_1, M^{(1)}_2, \ldots M^{(1)}_z, M^{(2)}, \ldots, M^{(k)}; \mathcal{I}' \rangle$ for some interleaving process $\mathcal{I}'$, where $M^{(1)}_r$ indicates the Markov chain defined on the single state $r \in V_1$.*

The interleaving process $\mathcal{I}'$ is defined as follows: If in $\mathcal{I}$ the probability of transition from some chain into $M^{(1)}$ in $\mathcal{P}$ is α, in $\mathcal{I}'$ the probability of transition from the same chain to $M^{(1)}_j$ is $\alpha S_1(j)$. Transition probabilities from $M^{(1)}_j$ are the same in $\mathcal{I}'$ as the transition probabilities from $M^{(1)}$ in $\mathcal{I}$.
Remark: Note that a one-step-mixing Markov chain is a zeroth-order Markov chain and a random walk on it is akin to drawing independent samples from a distribution. Nevertheless, we use this terminology to highlight the fact that such chains are a special pathological case for our algorithms.

3 Markov Chains on Disjoint State Spaces

In this section, we consider the problem of inferring mixtures of Markov chains when state spaces are pairwise disjoint. To begin with, we will assume the simpler mixture model. In Section 3.2, we show how our techniques extend to the chain-dependent mixture model.

3.1 The Simple Mixture Model

Our algorithm will have two stages. In the first stage, our algorithm will discover the partition of the whole state space $[n]$ into sets $\hat{V}_1, \ldots, \hat{V}_m$ which are the state spaces of the component Markov chains. Then, it is easy to infer the transition probabilities between states by looking at the substream corresponding to states in each $\hat{V}_l$. Once we infer the partition of the states, the mixing parameter α_l's can be estimated accurately from the fraction of states in $\hat{V}_l$ within the stream.

The main idea behind our algorithm is that certain patterns of states occur with different probabilities depending on whether the states in the pattern come from the same chain or from different chains. We make this idea precise and describe the algorithm in what follows.

Recall that S_l is the stationary distribution vector for the Markov chain $M^{(l)}$ extended to $[n]$. It is well know that the probability that Markov chain $M^{(l)}$ visits a state i tends to $S_l(i)$ as time goes to infinity. It follows that in our mixture model, the probability that we see a state i in our stream tends to

$$S(i) \stackrel{\text{def}}{=} \alpha_l S_l(i)$$

where l is such that $i \in V_l$. Note that l is unique since the state spaces are disjoint. Hence, one can get an estimate $\hat{T}_i$ for $S(i)$ by observing the frequencies[5] of each state i in the stream. The accuracy of this estimate is characterized by the following lemma.

Lemma 1. *For all i, the estimate $\hat{S}(i)$ is within $e^{-O(t)}$ of $\hat{T}_i$ when the length of the stream is at least $\tau t/(\min_i(\alpha_i))$ where τ is maximum cover time of any chain.*

We make the following key observations.

Proposition 2. *For $i, j \in V_l$, we expect to see the pattern $\langle ij \rangle$ in the stream with the frequency $\alpha_l S(i) M^{(l)}_{ij}$.*

In particular, if states i and j belong to the same Markov chain but the transition probability from i to j is 0, the pattern $\langle ij \rangle$ will not occur in the stream.

Proposition 3. *For states i and j from separate Markov chains, we expect the frequency of the pattern $\langle ij \rangle$, $\hat{T}_{ij}$ to be equal to $\hat{T}_i \hat{T}_j$.*

There is an important caveat to the last proposition. In order to accurately measure the frequencies of patterns $\langle ij \rangle$ where i and j occur in different Markov chain, it is necessary to look at positions in the stream that are sufficiently spaced to allow mixing of the component Markov chains. Consequently, we fix *a priori*, positions in the stream which are $\Omega(\tau Q)$ apart where τ is the maximum cover time and Q is the upper bound on the denominator of any probability represented as a fraction. We then sample these positions to determine the estimate on the frequency of various patterns.
Since the values of $\hat{S}$ and $\hat{T}$ are only estimates, we will use the notation "$\approx$" when we are comparing equalities relating such values. By the argument given in Lemma 1, these estimation errors will not lead us to wrong deductions, provided that the estimates are based on a long enough stream. Using the estimates $\hat{S}(\cdot)$ and the frequency $\hat{T}_{ij}$ one can make the following deduction:

- If $\hat{T}_{ij} \not\approx \hat{T}_i \hat{T}_j$, then i, j belong to the same chain.

In the case that $i, j \in V_l$ and $\alpha_l M^{(l)}_{i,j} = S(j)$, or equivalently $M^{(l)}_{i,j} = S_l(j)$. the criterion above does not suffice to provide us with clear evidence that i and j belong to the same Markov Chain and not to different Markov Chains. The next proposition may be used to disambiguate such cases.

[5] Here and elsewhere in the paper "frequency" refers to an estimated probability, i.e., it is a ratio of the observed number of successes to the total number of trials where the definition of "success" is evident from the context

Proposition 4. *Suppose $i, j \in V_l$ such that $M^{(l)}_{ij} \neq S_l(j)$. Suppose for a state p we cannot determine if $p \in V_l$ using the test above,*[6] *then $p \in V_l$ if and only if pattern $\langle ipj \rangle$ has the frequency $S(i)S(p)S(j)$, which translates to the test $\hat{T}_{ipj} \approx \hat{T}_i \hat{T}_p \hat{T}_j$.*

Proof. If $p \in V_l$, then $\alpha_l M^{(l)}_{ip} = S(p)$ by the assumption $\hat{T}_{ip} \approx \hat{S}(i)\hat{S}(p)$. Similarly, $\alpha_l M^{(l)}_{pj} = S(j)$. Therefore, the frequency of the pattern $\langle ipj \rangle$ in the stream is expected to be $\alpha_l^2 S(i) M^{(l)}_{ip} M^{(l)}_{pj} = S(i)S(p)S(j)$. In the case $p \notin V_l$, the same frequency is expected to be $\alpha_l S(i)S(p)M^{(l)}_{ij}$. These two expectation are separated since $\alpha_l M^{(l)}_{ij} \neq S(j)$ by the assumption.

Next, we give the subroutine `Grow_Components` that constructs a partition of $[n]$ using the propositions above and the frequencies $\hat{T}$. The algorithms uses the notation $C(i)$ to denote the component to which i belongs to.

```
Grow_Components(T̂)
Initialize: ∀i ∈ [n], C(i) ← {i}
Phase 1:
   For all i, j ∈ [n]
     If T̂_ij ≉ T̂_i T̂_j then
       Union(C(i), C(j))
Phase 2:
   For all i, j, p ∈ [n] such that T̂_ij ≉ T̂_i T̂_j and T̂_ipj ≈ T̂_i T̂_p T̂_j
       Union(C(i), C(p))
Return: the partition defined by C(·)'s
```

Lemma 2 (Soundness). *At the end of* `Grow_Components`*, if $C(i) = C(j)$ for some i, j, then there exists l such that $i, j \in V_l$.*

Proof. At the start of the subroutine, every state is initialized to be a component by itself. In Phase 1, two components are merged when there is definite evidence that the components belong to the same Markov chain by Proposition 2 or Proposition 3. In Phase 2, $\hat{T}_{ij} \not\approx \hat{T}_i \hat{T}_j$ implies that i and j are in the same component and hence Proposition 4 applies and shows the correctness of the union operation performed.

Lemma 3 (Completeness). *At the end of* `Grow_Components`*, $C(i) = C(j)$ for all i, j such that $i, j \in V_l$ for some l and $M^{(l)}_{i'j'} \neq S_l(j')$ for some $i', j' \in V_l$.*

Proof. First notice that our algorithm will identify i' and j' as being in the same component in phase 1. Now if either $M^{(l)}_{i'i} \neq S_l(i)$ or $M^{(l)}_{ij'} \neq S_l(j')$ we would have identified i as belonging to the same component as i' and j' in phase 1. Otherwise, phase 2 allows us to make this determination. The same argument holds for j as well. Thus, i and j will be known to belong to the component as i' and hence to each other's component.

[6] i.e., $\hat{T}_{ip} \approx \hat{S}(i)\hat{S}(p) \approx \hat{T}_{pi}$ and $\hat{T}_{jp} \approx \hat{S}(j)\hat{S}(p) \approx \hat{T}_{pj}$.

```
Infer_Disjoint_MC_Mixtures(X)
   Compute T̂_i, T̂_ij and T̂_ipj
   Let V̂_1, ..., V̂_m be the partition Grow_Components(T̂) returns
   For each 1 ≤ l ≤ m
     Considering the substream of X formed by all i ∈ V̂_l, calculate
       estimates for the transition probabilities involving i, j ∈ V̂_l.
```

At this point, we can claim that our algorithm identifies the irreducible Markov chains $M^{(l)}$ in the mixture (and their parameters). For other chains which have not been merged, from the contrapositive of the statement of Lemma 3 it must be the case that for all $i', j' \in V_l$ we have $M^{(l)}_{i'j'} = S_l(j')$, and the chains reduce to one-step mixing processes.

Theorem 4. *The model output by the algorithm is observationally equivalent to the true model with very high probability.*

3.2 Chain-Dependent Mixture Model

We now consider the model where the mixing process chooses the next chain with probabilities that are dependent on the chain that last made a transition. As in our algorithm for the simple mixture model, we will start with each state in a set by itself, and keep growing components by merging state sets as long as we can.

Definition 3. *A triple (i, j, s) satisfying $\hat{T}_{ij}\hat{T}_{js} \not\approx \hat{T}_{ijs}\hat{T}_j$ is termed as a* revealing triple*, otherwise a triple is called* non-revealing*.*

The following lemma ensues from case analysis.

Lemma 4. *If (i, j, s) is a revealing triple then i and s belong to the same chain and j belongs to a different chain.*

The algorithm, in the first part, will keep combining the components of the first two states in revealing triples, till no further merging is possible. Since the above test is sound, we will have a partition at the end which is possibly finer than the actual partition. That is, the state set of each of the original chains is the union of some of the parts in our partition. We can show the following:

Lemma 5. *If $i, s \in M^{(l)}$, $j \in M^{(k)}, k \neq l$, $M^{(l)}_{is} \neq S_l(s)$ and $\alpha_{ij} \cdot \alpha_{js} \neq 0$ then (i, j, s) is a revealing triple.*

Proof. Given i, j, s as in the statement consider the left hand side of the inequality in Lemma 4. $\hat{T}_{ij}\hat{T}_{js} \approx \hat{T}_i\alpha_{ij}S_k(j)\hat{T}_j\alpha_{js}S_l(s)$ and the right hand side, $\hat{T}_{ijs}\hat{T}_j \approx \hat{T}_i\alpha_{ij}S_k(j)\alpha_{js}M^{(l)}_{is}\hat{T}_j$. Evidently, these two expressions are not equal whenever $M^{(l)}_{is} \neq S_l(s)$.

The contrapositive of the above Lemma shows that if the triple (i, j, s) is a non-revealing triple where i and s belong to the same chain and $\hat{T}_{ij}\hat{T}_{js} \not\approx 0$ then it must be the case that j belongs to the same chain as i and s. This suggests the following merging algorithm:

```
Grow_Components_2(T̂)
Initialize: ∀i ∈ [n], C(i) ← {i}
Phase 1:
   For all i, j, s ∈ [n]
     If T̂_ij T̂_js ≉ T̂_ijs T̂_j then
       Union(C(i), C(s))
Phase 2:
   For all i, j, s ∈ [n] such that i, s ∈ C(i) ≠ C(j)
     If T̂_ij T̂_js ≈ T̂_ijs T̂_j ≉ 0 then
     Union(C(i), C(j))
Return: the partition defined by C(·)'s
```

Thus if the condition $\alpha_{ij}\alpha_{ji} \neq 0$ is satisfied and the Markov chain of i is not united in a single component, it must be the case that the Markov chain in question is observationally reducible to one step mixing. Thus the only remaining case to consider are (irreducible) Markov chains (containing i) such that for any other chain (containing j) it must be that $\alpha_{ij}\alpha_{ji} = 0$.
To handle Markov chains $M^{(l)}$ such that for all $l' \neq l$ and $j \in M^{(l')}$, we have $\alpha_{ij}\alpha_{ji} = 0$ the algorithm, in the second part, will perform the following steps:

1. Let $F_i(j) = \hat{T}_{ij}/\hat{T}_i$, i.e., the relative frequency that the next label after an i is j.

2. For all pairs i, j such that $\hat{T}_{ij} \neq 0$, and i and j are still singleton components, start with $D_{ij} = \{i, j\}$.
   (a) If for some state p, $F_i(p) \not\approx F_j(p)$, then include p in D_{ij}.
   (b) If for some state q, $\frac{F_q(i)}{F_q(j)} \not\approx \frac{T_i}{T_j}$, then include q in D_{ij}.

3. Keep applying the above rules above using all pairs in a component so far until D_{ij} does not change any more.

4. For each starting pair i, j, a set D_{ij} of states will be obtained at the end of this phase. Let $\mathcal{D}$ be the collection of those D_{ij}'s that are minimal.

5. Merge the components corresponding to the elements belonging to $D_{ij} \in \mathcal{D}$.

Lemma 6. *For states i and j from separate Markov chains, $D_{ij} \notin \mathcal{D}$.*

Proof. For any state s in the same chain $M^{(l)}$ as i, $F_{js} = 0$, because $\alpha_{js} = 0$. Therefore, the second closure rule will eventually include all the states from $M^{(l)}$ to D_{ij}. On the other hand for states i, v such that $v \in M^{(l)}$, D_{iv} will contain states only from $M^{(l)}$. Hence, as $D_{iv} \subset D_{ij}$, D_{ij} will not be minimal.

Now we know that each set in $\mathcal{D}$ is a subset of the state space of a Markov chain. Thus, we get

Theorem 5. *Let $\langle M^{(1)}, M^{(2)}, \ldots, M^{(k)}; \mathcal{I}\rangle$ be an interleaved process with chain-dependent mixing and no one-step-mixing Markov chains. If for all $l \in [k]$, $\alpha_{ii} \neq 0$ for $i \in M^{(l)}$, then we can infer a model observationally equivalent to the true model.*

3.3 A Negative Result

Suppose H is a two state probabilistic automaton where the transition probabilities are H_{ija} where $i, j \in \{1, 2\}$. Let $\{a\} = L$ be the collection of all possible labels output.

Consider the following mixture process: We will create two Markov chains $M_1^{(a)}, M_2^{(a)}$ for each label $a \in L$. Each of the Markov chains $M_1^{(a)}, M_2^{(a)}$ is a markov chain with a single state corresponding to the label a. The transition probability from chain $M_i^{(a)}$ to $M_j^{(b)}$ is H_{ijb}.

Clearly the "states" of the Markov chains $M_1^{(a)}, M_2^{(a)}$ overlap – and it is easy to see that the probability of observing a sequence of labels as the output of H is the same as observing the sequence in the interleaved mixture of the Markov chains. Since the estimation of H is intractable [1], even for two states (but variable size alphabet), we can conclude:

Theorem 6. *Identifying interleaving Markov chains with overlapping state spaces under the chain dependent mixture model is computationally intractable.*

4 Non-Disjoint State Spaces

In the previous section we showed that in the chain dependent mixture model we have a reasonably sharp characterization. A natural question that arises from the negative result is: *can we characterize under what conditions can we infer the mixture of non-disjoint Markov chains, even for two chains ?* A first step towards the goal would be to understand the simple mixture model.

Consider the most extreme case of overlap where we have a mixture of two identical Markov chains. The frequency of states in the sequence gives an estimate of the stationary distribution S of each chain which is also the overall stationary distribution. Note that $M_{ij}^{(l)} = M_{ij}$ for all i, j.

Consider the pattern $\langle ij\rangle$. This pattern can arise because there was a transition from i to j in some chain $M^{(l)}$ or it can arise because we first observed i and control shifted to the other chain and we observed j. Let α_l be the probability that the mixing process chooses $M^{(l)}$. Then,

$$\hat{T}_{ij} \approx \sum_{c=1}^{k} \alpha_c S(i)((\alpha_c M_{ij}) + (1 - \alpha_c)S(j)).$$

Letting $w = \sum_c \alpha_c^2$ we can simplify the above equation to get: $\hat{T}_{ij} = S(i)[wM_{ij} + (1-w)S(j)] = S(i)[w(M_{ij} - S(j)) + S(j)]$. Rearranging terms we have $M_{ij} = \frac{\frac{\hat{T}_{ij}}{S_i} - S_j}{w} + S_j$. Any value of w that results in $0 \leq M_{ij} \leq 1$ for all i, j leads to an observationally equivalent process to the one actually generating the stream. The set of possible w's is not empty since, in particular, $w = 1$ leads to $M_{ij} = \frac{\hat{T}_{ij}}{S_i}$ corresponding to having just one Markov chain with these transition probabitlities.

What we see above is that the symmetries in the problem introduced by assuming that all Markov chains are identical facilitate the inference of an observationally equivalent process. The general situation is more complicated even for two Markov chains.

We consider the mixtures of two Markov chains with non-disjoint state spaces. We give an algorithm for this case under a technical condition that requires a special state. Namely, we require that there is a state i_s that is exclusively in one of the Markov chains, say $M^{(1)}$, and

$$\text{either } M^{(1)}_{i_s j} > S_1(j) \text{ or } M^{(1)}_{ij} = 0 \text{ for all } j \in V_1.$$

Let α_1, α_2 be the mixture probabilities. Then, considering the four possible ways of $\langle ij \rangle$ occurring in the stream, we get

$$\hat{T}_{ij} = \alpha_1^2 S_1(i) M^{(1)}_{ij} + \alpha_1\alpha_2 \left(S_1(i)S_2(j) + S_2(i)S_1(j)\right) + \alpha_2^2 S_2(i) M^{(2)}_{ij}.$$

Let $A_{ij} \stackrel{\text{def}}{=} \hat{T}_{ij} - (SS^T)_{ij}$ where $S = \alpha_1 S_1 + \alpha_2 S_2$ as before. Then, we can write

$$A_{ij} = \alpha_1^2 S_1(i) \left(M^{(1)}_{ij} - S_1(j)\right) + \alpha_2^2 S_2(i) \left(M^{(2)}_{ij} - S_2(j)\right).$$

Consider the state i_s required by the technical condition. For any state j such that $M^{(1)}_{i_s j} > 0$, we have $A_{i_s j} = \alpha_1^2 S_1(i_s) \left(M^{(1)}_{i_s j} - S_1(j)\right) > 0$. For any other state j with $S_1(j) > 0$, $A_{i_s j} = -\alpha_1^2 S_1(i_s) S_1(j) < 0$. Finally, $A_{i_s j} = 0$ for all the remaining states.

Since $S(i_s) = \alpha_1 S_1(i_s)$, for each $j \in [n]$, we can infer $\alpha_1 S_1(j)$ from the observations above. Hence, we can infer $\alpha_2 S_2(j)$ for each j by $S(j) = \alpha_1 S_1(j) + \alpha_2 S_2(j)$. Since we know the vectors S_1, S_2, we can now calculate $M_{ij} \stackrel{\text{def}}{=} \alpha_1 M^{(1)}_{ij} + \alpha_2 M^{(2)}_{ij}$ for all i, j pairs.

If state i or j exclusively belongs to one of the Markov chains, M_{ij} gives the product of the appropriate mixing parameter and the transition probability. In the case when both states i and j are common between the Markov chains, we will use the frequency $\hat{T}_{ii_s j}$ of pattern $\langle ii_s j \rangle$ to infer $M^{(1)}_{ij}$ and $M^{(2)}_{ij}$.

The frequency of the pattern $\langle ii_s j \rangle$ is expected to be

$$\hat{T}_{ii_s j} \approx \alpha_1^2 S_1(i) M^{(1)}_{ii_s} (\alpha_2 S_2(j) + \alpha_1 M^{(1)}_{i_s j}) + \alpha_1\alpha_2 S_2(i) S_1(i_s) (\alpha_1 M^{(1)}_{i_s j} + \alpha_2 M^{(2)}_{ij}).$$

Note that all but the last term is already inferred by the algorithm. Therefore, $\alpha_2 M_{ij}^{(2)}$, hence $\alpha_1 M_{ij}^{(1)}$, can be calculated.

Finally, using the next state distribution for the state i_s, we can calculate α_1 and α_2. This completes the description of our algorithm.

5 Conclusions and Open Problems

In this paper we have taken the first steps towards understanding the behavior of a mixture of Markov chains. We believe that there are many more problems to be explored in this area which are both mathematically challenging and practically interesting.

A natural open question is the condition $\alpha_{ii} \neq 0$, i.e., there is a non-zero probability of observing the next label from the same Markov chain. We note that Freund and Ron had made a similar assumption that α_{ii} is large, which allowed then to obtain "pure" runs from each of the chains. It is conceivable that the inference problem of disjoint state Markov chains becomes intractable after we allow $\alpha_{ii} = 0$.

Another interesting question is the optimizing the length of the observation required for inference – or if sufficient lengths are not available then compute the best partial inference possible. This is interesting even for small ~ 50 states and a possible solution may be trade off computation or storage against observation length.

References

1. Naoki Abe and Manfred Warmuth. On the computational complexity of approximating distributions by probabilistic automata. *Machine Learning*, 1992. (to appear in the special issue for COLT 1990).
2. Serge Abiteboul, Victor Vianu, Brad Fordham, and Yelena Yesha. Relational transducers for electronic commerce. pages 179–187, 1998.
3. Sanjeev Arora and Ravi Kannan. Learning mixtures of arbitrary gaussians. In *ACM Symposium on Theory of Computing*, pages 247–257, 2001.
4. Y. Bengio and P. Frasconi. Input-output HMM's for sequence processing. *IEEE Transactions on Neural Networks*, 7(5):1231–1249, September 1996.
5. C.B. Burge and S. Karlin. Finding the genes in genomic dna. *J. Mol. Bio.*, 268:78–94, 1997.
6. Sanjoy Dasgupta. Learning mixtures of gaussians. Technical Report CSD-99-1047, University of California, Berkeley, May 19, 1999.
7. Dorothy E. Denning. An intrusion-detection model. *Transactions of software engineering*, 13(2):222–232, 1987.
8. R. O. Duda and P. E. Hart. *Pattern Classification and Scene Analysis*. John Wiley and Sons, New York, 1974.
9. Tom Fawcett and Foster J. Provost. Adaptive fraud detection. *Data Mining and Knowledge Discovery*, 1(3):291–316, 1997.
10. Yoav Freund and Dana Ron. Learning to model sequences generated by switching distributions. In *Proceedings of the 8th Annual Conference on Computational Learning Theory (COLT'95)*, pages 41–50, New York, NY, USA, July 1995. ACM Press.

11. Charles Kervrann and Fabrice Heitz. A hierarchical Markov modeling approach for the segmentation and tracking of deformable shapes. *Graphical models and image processing: GMIP*, 60(3):173–195, 1998.
12. Wenke Lee, Salvatore J. Stolfo, and Kui W. Mok. A data mining framework for building intrusion detection models. In *IEEE Symposium on Security and Privacy*, pages 120–132, 1999.
13. Alon Y. Levy and Daniel S. Weld. Intelligent internet systems. *Artificial Intelligence*, 118(1-2):1–14, 2000.
14. Lawrence R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2), 1989.
15. Marc Spielmann. Verification of relational transducers for electronic commerce. In *Symposium on Principles of Database Systems*, pages 92–103, 2000.
16. B. Thiesson, C. Meek, D. Chickering, and D. Heckerman. Learning mixtures of Bayesian networks. Technical Report MSR-TR-97-30, Microsoft Research, Redmond, WA, 1997.
17. Bo Thiesson, Christopher Meek, David Maxwell Chickering, and David Heckerman. Learning mixtures of DAG models. In Gregory F. Cooper and Serafín Moral, editors, *Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence (UAI-98)*, pages 504–513, San Francisco, July 24–26 1998. Morgan Kaufmann.
18. Christina Warrender, Stephanie Forrest, and Barak A. Pearlmutter. Detecting intrusions using system calls: Alternative data models. In *IEEE Symposium on Security and Privacy*, pages 133–145, 1999.