Self-Similar Layered Hidden Markov Models

Jafar Adibi and Wei-Min Shen

Information Sciences Institute, Computer Science Department
University of Southern California
4676 Admiralty Way, Marina del Ray, CA 90292
{adibi,shen}@isi.edu

Abstract. Hidden Markov Models (HMM) have proven to be useful in a variety of real world applications where considerations for uncertainty are crucial. Such an advantage can be more leveraged if HMM can be scaled up to deal with complex problems. In this paper, we introduce, analyze and demonstrate Self-Similar Layered HMM (SSLHMM), for a certain group of complex problems which show self-similar property, and exploit this property to reduce the complexity of model construction. We show how the embedded knowledge of self-similar structure can be used to reduce the complexity of learning and increase the accuracy of the learned model. Moreover, we introduce three different types of self-similarity in SSLHMM, and investigate their performance in the context of synthetic data and real-world network databases. We show that SSLHMM has several advantages comparing to conventional HMM techniques and it is more efficient and accurate than one-step, flat method for model construction.

1 Introduction

There is a vast amount of natural structures and physical systems which contain self-similar structures that are made through recurrent processes. To name a few: ocean flows, changes in the yearly flood levels of rivers, voltages across nerve membranes, musical melodies, human brains, economic markets, Internet web logs and network data create enormously complex self-similar data [21]. While there have been much effort on observing self-similar structures in scientific databases and natural structures, there are few works on using self-similar structure and fractal dimension for the purpose of data mining and predictive modeling. Among these works, using fractal dimension and self-similarity to reduce the dimensionally curse [21], learning association rules [2] and applications in spatial joint selectivity in databases [9] are considerable. In this paper we introduce a novel technique which uses the self-similar structure for predictive modeling using a Self-Similar Layered Hidden Markov Model (SSLHMM).

Despite the broad range of application areas shown for classic HMMs, they do have limitations and do not easily handle problems with certain characteristics. For instance, classic HMM has difficulties to model complex problems with large states spaces. Among the recognized limitations, we only focus on complexity of HMM for a certain category of problems with the following characteristics: 1) The uncertainty and complexity embedded in these applications make it difficult and impractical to construct the model in one step. 2) Systems are self-similar, contain self-similar struc

L. De Raedt and A. Siebes (Eds.): PKDD 2001, LNAI 2168, pp. 1-15, 2001.

tures and have been generated through recurrent processes. For instance, analysis of traffic data from networks and services such as ISDN traffic, Ethernet LAN's, Common Channel Signaling Network (CCNS) and Variable Bit Rate (VBR) video have all convincingly demonstrated the presence of features such as self-similarity, long range dependence, slowly decaying variances, heavy-tailed distributions and fractal dimensions [24].

In a companion paper, Adibi and Shen introduced a domain independent novel technique to mine sequential databases through Mining by Layered Phases (MLP) in both discrete and continuous domains [1]. In this paper we introduce a special form of MLP as Self-Similar Layered HMM (SSLHMM) for self-similar structures. We show how SSLHMM uses the information embedded in a self-similar structure to reduce the complexity of the problem and learn a more accurate model than a general HMM. Our result is encouraging and show a significant improvement when a self-similar data are modeled through SSLHMM in comparison with HMM.

The rest of this paper is organized as follows. In section 2 we review the related work to this paper. In section 3, we introduce SSLHMM, its definition and properties. We explain major components of the system and we drive the sequence likelihood for a 2-layers SSLHMM. Section 4 shows the current result with an experimental finding in Network data along with discussion and interpretation followed by the future work and conclusions in section 5.

2 Related Work

HMMs proven tremendously useful as models of stochastic planning and decision problems. However, the computational difficulty of applying classic dynamic and limitation of conventional HMM to realistic problems has spurred much research into techniques to deal with the large states and complex problems. These approaches includes function approximation, ratability consideration, aggregation techniques and extension to HMM. In the following we refer to those works which are related to our approach in general or in specific. We categorize these woks as extension to HMM, aggregation techniques and segmentation.

Regular HMMs are capable of modeling only one process over time. To overcome such limitation there are several works to extend HMMs. There are three major extension which are close to our method. The first method introduced by Gharamani and Jordan as Factorial Hidden Markov Model (FHMM)[12]. This models generalize the HMM in which a state is factored into multiple state variables and therefore represented in a distributed manner. FHMM combines the output of the N HMMs in a single output signal, such that the output probabilities depend on the N dimensional meta-state. As the exact algorithm for this method is intractable they provide approximate inference using Gibbs sampling or variational methods. Williams and Hinton also formulated the problem of learning in HMMs with distributed state representations[23], which is a particular class of probabilistic graphical model by Perl [16]. The second method known as Coupled Hidden Markov Model (CHMM) consists of modeling the N process in N HMMs, whose state probabilities influence one another and whose outputs are separate signals. Brand, Oliver and Pentland described polynomial time training methods and demonstrate advantage of CHMM over HMM [5].

The last extension to HMM related to our approach introduced by Voglar and Metaxas as Parallel Hidden Markov Models (PHMM) which model the parallel process independently and can be trained independently [22]. In addition, the notion of hierarchical HMM has been introduced in [11] in which they extend the conventional Baum-Welch method for hierarchical HMM. Their major application is on text recognition in which the segmentation techniques benefits of the nature of handwriting. The major difference of SSLHMM with most of the above mentioned approaches is that they do not consider self-similarity for data. SSLHMM uses a recursive learning procedure to find the optimal solution and make it possible to use an exact solution rather approximation. In addition, SSLHMM as a specific case of MLP use the notion of phase in which learner consider laziness for the systems which is along with long range dependence and slowly decaying variances. For a detail description of MLP please refer to [1]. In addition, FHMM does not provide a hierarchical structure and its model is not interpretable while SSLHMM is designed toward interpretability. Also, HHMM does not provide the notion of self similarity.

In sequential planning, HMM-in general and Partial Observable Markov Decision Process models (POMDP) specifically have proven to be useful in a variety of real world applications [18]. The computational difficulty of applying dynamic programming techniques to realistic problems has spurred much research into techniques to deal with the large state and action spaces. These include function approximation [3] and state aggregation techniques [4, 8]. One general method for tackling large MDPs is decomposition of a large state model to smaller models [8, 17]. Dean and Lin [8], Berteskas and Tsikits [3] also showed some Markov Decision Process are loosely coupled and hence enable to get treated by divide-and-conquer algorithms. The evolution of the model over time also has been modeled as a semi-Markov Decision Process (SMDP) [18]. Sutton1[20] proposed temporal abstraction, which concatenate sequences of state transition together to permit reasoning about temporarily extended events, and form a behavioral hierarchy as in [17]. Most of the work in this direction split a well-defined problem space to smaller spaces and they come up with subspaces and intra actions. In contrast SSLHMM attempt to build a model out of a given data through a top down fashion. The use of hierarchical HMMs mostly has been employed to divide a huge state space to smaller space or to aggregate actions and decisions. MLP in general and SSLHMM in specific are orthogonal to state decomposition approaches.

Complexity reduction also has been investigated through segmentation specially in Speech Recognition literature. Most of the work is based on probabilistic network, Viterbi search for all possible segmentation and using of domain knowledge as hypothesized segment start and end times [6, 7, 15]. Segmental HMMs also has been investigated in [13]. Even though the approach fits in speech recognition applications, but it decompose a waveform to local segments each present a “shape” with additive noise. A limitation of these approaches in general is that they do not provide a coherent language for expressing prior knowledge, or integrating shape cues at both the local and global level. SSLHMM integrates the prior knowledge in the infrastructure of model and as part of knowledge discovery process.

Based on our knowledge, the notion of Self-Similar Layered HMM has not been introduced yet. In addition, the notion of locality and boundary in phases make this work distinguish with similar approaches.

3 Self-Similar Layered Hidden Markov Model (SSLHMM)

Conventional HMMs are enable to model only one process at the time which represent by transition among the states. Fig. 1(a) shows a HMM with 9 states. A HMM λ for discrete symbol observation characterized by the following set of definitions: *state transition matrix: S, observation distribution matrix: B,* a set of *observations M*, a set of *states: n* and *initial distribution* π [19]. Having a set of observation O and a model λ, the old well-known problem is to adjust model parameters to maximize $P(O \mid \lambda)$.

In the modeling of complex processes, when the number of states goes high, the maximization process gets more difficult. A solution provided in other literature is to use of a Layered HMM instead [1, 12]. Layered HMM has the capability to model more than one process. Hence, it provides an easier platform for modeling complex processes. Layered HMM is a combination of two or more HMM processes in a hierarchy. Fig. 1(b) shows a Layered HMM with 9 states and 3 super-states, or macro-states (big circles with shade), which we refer to them as p*hases*. As we can see, each phase is a collection of states bounded to each other. The real model transition happens among the states. However, there is another transition process in upper layer among phases. The comprehensive transition model is a function of transition among states and transition among phases. Layered HMM similar to conventional HMM characterized by the following set of definitions: *a set of observation: M and* a set of *states: n*, a set of *phases: N, state transition matrix: S*, *phase transition matrix: R, observation distribution in each state: B and observation distribution in each phase :C* and *initial condition for each layer:* π.Learning and modeling follows the well-known Baum-Welch algorithm with some modification in forward and backward algorithm.

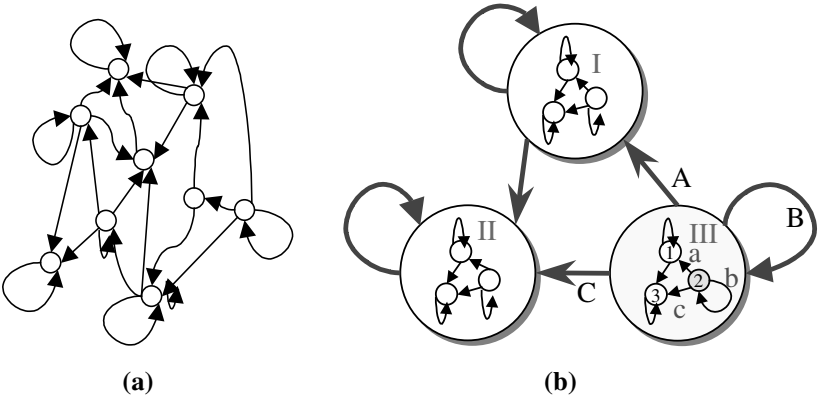


Fig. 1. (a) A normal Hidden Markov Model with 9 states, **(b)** Self-Similar Layered Hidden Markov Model with 9 states and 3 phases. As it shows each *phase* contains similar structure

A macro point of view suggests that the overall system behavior is more a trajectory among phases. In particular, system may go from one phase to another and stays in each phase for a certain amount of time. From a modeling point of view, *phase* is a set of properties, which remain homogenous through a set of states of the system and during a period of time. *phase* may be considered as a collection of locally connected sets, groups, levels, categories, objects, states or behaviors. The notion of *Phase* comes with the idea of granularity, organization and hierarchy. An observed sequence of a system might be considered as a collection of a behaviors among phases (rather than a big collection of states in a flat structure), and it may provide enough information for reasoning or be guidance for further details. Hence, a sequence with such property could be modeled through a layered structure. For example in network application domain a phase could define as "congestion" or "stable". A micro point of view shows that the overall system behavior is a transition among the states.

SSLHMM is a special form of Layered HMM in which there are some constraints on state layer transition, phase layer transition, and observation distribution. A closer look at Fig. 1(b) shows that this particular Layered HMM structure indeed is a *self-similar* structure. As it shows, there is a copy of the super model (model consists of *phases* and transition among them) inside of each phase. For instance the probability *A* of going form *phase III phase I* is equal to the probability *a* of transition from *state 3* to *state 1* in *phase III* (and in other phases as well).

The advantage of such structure is that like any other self-similar model it is possible to learn the whole model having any part of the model. Although there are a couple of assumptions to hold such properties but fortunately for a large group of systems in nature self-similarity is one of their characteristics. In the following, we introduce a self-similar Markovian structure in which the model shows similar structure across all or at least a range of structure scale.

3.1 Notation

In the following we describe our notation for the rest of this paper along with assumptions and definitions. We follow and modify Rabiner [19] notation for discrete HMM. A SSLHMM for discreet observation is characterized by the Table 1.

For the simplicity we use $\lambda = (S, B, \pi)$ for the state layer and $\Lambda = (R, C, \pi)$ for a given phase layer structure. In addition we use $\Theta = (\lambda, \Lambda, Z)$ for the whole structure of SSLHMM in which *Z* holds the hierarchical information including leaf structure and layer structure. Even though the states are hidden but in real world application there is a lot of information about physical problems, which points out some characteristics of state or phase.

Table 1. Self-Similar Layered Hidden Markov Model parameters and definition

Parameter	Definition
N	The number of *Phases* in the model We label individual phases as $\{1,2,\cdots,N\}$ and denote the phase at time t as Q_t.
n	The number of states. We label individual states as $\{1,2,\cdots,n\}$ and denote the state at time t as q_t.
M	The number of distinct observations
$R=\{r_{IJ}\}$	Phase layer transition probability, where $r_{IJ}=P(Q_{t+1}=J \mid Q_t=I)$ and $1\le I,J\le N$
$S=\{s_{ij}\}$	State layer transition probability: where $s_{ij}=P(q_{t+1}=j \mid q_t=i)$ and $1\le i,j\le n$
$C=\{c_J^t(k)\}$	The observation probability for phase layer in which $c_J^t(k)=P[o_t=v_k \mid Q_t=J]$
$B=\{b_j^t(k)\}$	The observation probability for state layer in which $b_j^t(k)=P[o_t=v_k \mid q_t=j]$
$O=\{o_1,o_2,\cdots,o_T\}$	The observation series
$\pi_{iI}=P[q_t=i\wedge Q_t=I]$	The initial state distribution in which $1\le i\le n$ and $1\le I\le N$

3.2 Parameter Estimation

All equations of Layered HMM can be derived similar to conventional HMM. However without losing generality we only derive the forward algorithm for a two layer HMM as we apply such algorithm to calculate likelihood in next section. In addition, we assume a one-to-one relation among states and phases for hidden self-similarity. Similar to HMM, we consider the forward variable $\alpha_t(I,i)$ defined as

$$\alpha_t(I,i)=P(o_1,o_2,\cdots o_t,Q_t=I\wedge q_t=i\mid\Theta) \tag{1}$$

which is the probability of the partial observation sequence, $o_1,o_2,\cdots o_t$ at time t at state i and phase I, given the model Θ. Following the *Baum-Welch* forward procedure algorithm we can solve for $\alpha_t(I,i)$ inductively as follows:

Initialization:

$$\alpha_1^{\Theta}(J,j)=\pi_{(J,j)}P(o_1\mid Q_1=J\wedge q_1=j) \tag{2}$$

Induction:

$$\alpha_{t+1}^{\Theta}(J,j) = \left[\sum_{I=1}^{N}\sum_{i=1}^{n} \alpha_t^{\Theta}(I,i)\cdot W_{(I,i)(J,j)}\right] * P(o_{t+1} \mid Q_{t+1} = J \wedge q_{t+1} = j) \tag{3}$$

in which $W_{(I,i)(J,j)}$ is the transition matrix form state *i* and phase *I* to state *j* in phase *J*. We will show how we calculate this transition matrix in a simple way.

Termination:

$$P(O \mid \Theta) = \sum_{I=1}^{N}\sum_{i=1}^{n} \alpha_t^{\Theta}(I,i) \tag{4}$$

3.3 Self-Similarity Definition and Conditions

In geometry, self-similarity comes with the term *fractal*. *Fractals* have two interesting features. First they are self-similar on multiple scales. Second, *fractals* have a fractional dimension, as opposed to an integer dimension that idealized objects or structures have. To address self-similarity in Layered HMMs, we define three major types of Markovian Self-Similar structures: s*tructural self-similarity*, *hidden self-similarity* and *strong self-similarity.*

Structural Self-Similarity: The structural self-similarity refers to similarity in structure in different layers. In our example if phase structure transition be equivalent to the state structure transition, we consider model Θ as a self-similar HMM. In this case, we will have $r_{IJ} = s_{ij}$ if *i=I*, *J=j* and *n=N*2*. This type of self-similarity refers to the structure of the layers. The scale of self-similarity can goes further depends on the nature of the problem. It is important to mention that in general, in modeling via HMM the number of states preferably keep low to reduce the complexity and to increase accuracy. One of the main advantage of SSLHMM as it was described is that it reduces the number of states dramatically.

Hidden Self-Similarity: The Hidden self-similarity refers to similarity in observation distribution in different layers. We define Hidden self-similarity as the following. There is a permutation of I, i, $I = \Psi(i)$ in which $P(o_t \mid \Psi(i)) = P(o_t \mid (\Psi(i), i))$ in which

$$P(o_t \mid (\Psi(i), i)) = \sum_{j=1}^{n} P(o_t \mid (\Psi(i), j)) \cdot P(j \mid \Psi(i)) = \sum_{j=1}^{n} P(o_t \mid j) \cdot P(j \mid \Psi(i)) \tag{5}$$

in our example if we assume $\Psi(1) = I, \Psi(2) = II$ and $\Psi(3) = III$, the above mention property for *state* 1 and *phase I* will be as the following:

$$P(o_t \mid (1,I)) = P(o_t \mid (I,1))P(1 \mid I) + P(o_t \mid (I,2))P(2 \mid I) + P(o_t \mid (I,3))P(3 \mid I) \quad (6)$$

We refer to this type of self-similarity as *hidden* because it is not intuitive and it is very hard to recognize.

Strong Self-Similarity*:* A SSLHMM $\Theta = (\lambda, \Lambda)$ is *strong self-similar* if the model satisfies requirements of *structural self-similarity* and *hidden self-similarity.*

3.4 Assumptions

In the following we describe our major assumptions, definitions and lemmas to re-write the sequence likelihood.

Decomposability: we assume layers in a Layered HMM model are decomposable. The probability of occupancy of a given state in a given layer is:

$$P[Q_{t+1} = J \wedge q_{t+1} = j] = P[Q_{t+1} = J] * P[q_{t+1} = j \mid Q_{t+1} = J] \quad (7)$$

Decomposability property assumes that system transition matrix is decomposable to phase transition matrix and state transition matrix. Considering such assumption, the overall transition probability for a given state to another state is a Tensor product of phase transition and state transition. For a multi-layered HMM the over all transition probability would be equal to *Tensor products* of HMM transition models. Without loosing generality we only explain the detail of a 2-layer SSLHMM. The transition probability among states and phases will be as following:

$$P[Q_{t+1} = J \wedge q_{t+1} = j \mid Q_t = I \wedge q_t = i] = r_{IJ} \times s_{ij} \quad (8)$$

We show the tensor product with *W*.

$$W = S \otimes R \text{ and } w_{(I,i)(J,j)} = r_{IJ} \times s_{ij} \quad (9)$$

Example: If we consider the transition probability for state layer and phase layer as

$$S = \begin{pmatrix} .2 & .3 & .5 \\ .7 & .1 & .2 \\ .3 & .1 & .6 \end{pmatrix} \text{ and } R = \begin{pmatrix} .2 & .3 & .5 \\ .7 & .1 & .2 \\ .3 & .1 & .6 \end{pmatrix} \text{, we will have : } W = \begin{pmatrix} .04 & .06 & .1 & .06 & .09 & .15 & .1 & .15 & .25 \\ .14 & .02 & .04 & .21 & .03 & .06 & .35 & .05 & .1 \\ .06 & .02 & .12 & .09 & .03 & .18 & .15 & .05 & .3 \\ .14 & .21 & .35 & .02 & .03 & .05 & .04 & .06 & .01 \\ .49 & .07 & .14 & .07 & .01 & .02 & .14 & .02 & .04 \\ .21 & .07 & .42 & .03 & .01 & .06 & .06 & .02 & .12 \\ .06 & .09 & .15 & .02 & .03 & .05 & .12 & .18 & .3 \\ .21 & .03 & .06 & .07 & .01 & .02 & .42 & .06 & .12 \\ .09 & .03 & .18 & .03 & .01 & .06 & .18 & .06 & .36 \end{pmatrix}$$

Lemma 1: Tensor Product of HMMs: Considering a HMM Model as $\lambda = (W, B, \pi)$, it is possible to decompose λ to smaller models if $\exists W_1, W_2$ of order $|W_1|$ and $|W_2|$ such that $|W| = |W_1| \times |W_2|$ and $W = W_1 \otimes W_2$.

Note: Not all HMMs are decomposable to a Tensor product of smaller models.

Lemma 2: Markov Property of HMMs Tensor Products: If S and R are Markovian transition matrix then $W = R \otimes S$ is Tensor Markov.

$$\sum_{J=1}^{N} R_{IJ} = 1 \text{ for all } I \in R \text{ and } \sum_{j=1}^{n} S_{ij} = 1 \text{ for all } i \in S \tag{10}$$

$$\sum_{J=1}^{N}\sum_{j=1}^{n} W_{(I,i)(J,j)} = \sum_{J=1}^{N}\sum_{j=1}^{n} r_{IJ}\, s_{ij} = \sum_{J=1}^{N} r_{IJ} \sum_{j=1}^{n} s_{ij} = 1 \tag{11}$$

Any Tensor Markov Model $|W_1| \times |W_2|$ is isomorphic by a Markov Model to order of $|W| = |W_1| \times |W_2|$.

3.5 Re-writing Sequence Likelihood

By using above mentioned assumptions we can re-write the sequence likelihood for a strong self-similar (one-to-one) HMM as following. *Hidden self-similarity* implies:

$$P(o_{t+1} \mid Q_{t+1} = J \wedge q_{t+1} = j) = P(o_{t+1} \mid q_{t+1} = j) \text{ if } J = j \tag{12}$$

Decomposability assumption along with *structural self-similarity* make it possible to calculate *W*. Hence equation (3) becomes as:

$$\alpha_{t+1}^{\Theta}(J, j) = \left[\sum_{I=1}^{N}\sum_{i=1}^{n} \alpha_t^{\Theta}(I, i) \cdot W_{(I,i)(J,j)}\right] * P(o_{t+1} \mid q_{t+1} = j) \text{ if } i = j \tag{13}$$

$$\alpha_{t+1}^{\Theta}(J, j) = \left[\sum_{I=1}^{N}\sum_{i=1}^{n} \alpha_t^{\Theta}(I, i) \cdot W_{(I,i)(J,j)}\right] * \sum_{j=1}^{n} P(o_t \mid j) \cdot P(j \mid \Psi(i)) \text{ if } i \neq j$$

3.6 The Learning Process

The learning procedure for SSLHMM is similar to traditional HMM via the expectation maximization (EM) [19] except the calculation of $\alpha_t(i, I)$ and $\beta_t(i, I)$ as above. We can choose $\Theta = (\lambda, \Lambda, Z)$ such that its likelihood $P(O \mid \Theta)$ is locally maximized using an iterative procedure such as *Baum-Welch* method. This procedure iterates

between E step which fixes the current parameters and computes posterior probabilities over the hidden states and M step which uses these probabilities to maximize the expected log likelihood of the observation. We derived forward variable α in last section, and deriving B, the backward parameter is similar to forward parameter.

4 Result

We have applied SSLHMM approach to synthetic data and a Network domain database. Our implementation is in *MATLAB* programming language and has been tested on Pentium III 450 MHz processor with 384 MB RAM.

4.1 Experiment 1: Synthetic Data

To compare SSLHMM with HMM, we employed a SSLHMM simulator with the capability of simulation of discrete and continuous data. In our simulator, a user has the capability to define the number of sequence in experimental pool, length of each sequence, number of layers, number of states in each phase, number of phases and observation set for discrete environment or a range for continuous observation. We verified that the synthetic data is indeed self-similar with H=.6. In this paper we only report the comparison of *Baum-Welch forward algorithm* for HMM with n_{HMM} states and a 2-layer *strong* SSLHMM with *N* phases and *n* states. The main purpose of this experiment is built on the following chain of principles:

Assume there is a sequence of observation *O* generated by a self-similar structure.

- We would like to estimate HMM parameter for such data (*n* assume to be known in advance)
- We would like to adjust model parameters $\lambda = (S, B, \pi)$ to maximize $P(O \mid \lambda)$.
- Model could be either a flat HMM or a SSLHMM
- We illustrate that for $O = \{o_1, o_2, \cdots, o_T\}$, $P(O \mid SSLHMM)$ is higher than $P(O \mid HMM)$, the probability of the observation given each model.
- We also observed that if O is not generated by a SSLHMM but by a HMM $P(O \mid SSLHMM) \approx P(O \mid HMM)$. However due to space limitation we do not show the result.

We ran a series of test for a problem consists of pre-selected number of states, up to 15 perceptions and 100 sequence of observation for each run. We assume the number of states and phases are known so *Baum-Welch* algorithm uses n_{HMM} to build the model and SSLHMM use N and *n* (number of phases and number of states). The assumption of *strong* self-similarity implies that $n = N^2$, as we have a copy of phase structure inside of each phase to present state layer. We repeat the whole experience with a random distribution for each phase but in a self-similar fashion and for a variety of different *n* and *N*. First we trained on the 50% of the data and find *P(Model | train)* for both HMM and SSLHMM. In second step we calculate *P(Model | test)* where "*test*" is the remaining 50% of the data. Fig. 2 shows *-log(likelihood)* of different experiments. A smaller number of *-log(likelihood)* indicate a higher probability. We ran HMM with prior number of states equal to 9, 16 and 64, and SSLHMM with

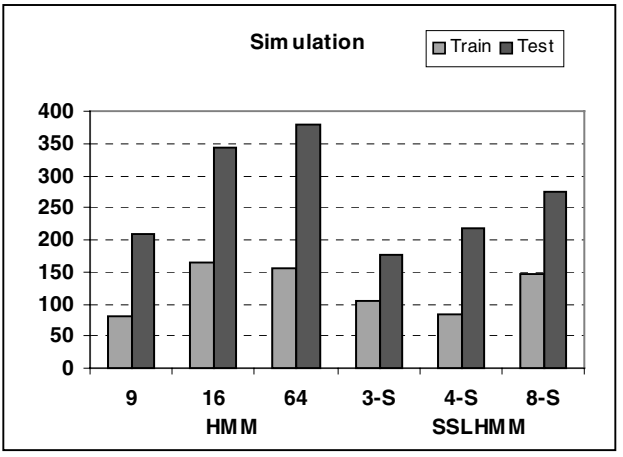


Fig. 2. Negative log likelihood for synthetic data. "x-*s*" indicates a 2 layers SSLHMM with x as number of state in each layer

number of phases equal to 3, 4 and 8 (shown as 3-s, 4-s and 8-s in the Fig. 4. As we may see the best model of SSLHMM outperforms the best model of HMM. In addition, the average *-log(likelihood)* of modeling through SSLHMM in all experiences is lower than modeling through HMM by 39%.

4.2 Experiment 2: Network Data

Understanding the nature of traffic in high-speed, high-bandwidth communications is essential for engineering and performance evaluation. It is important to know the traffic behavior of some of the expected major contributors to future high-speed network traffic. There have been a handful research and development in this area to analyze LAN traffic data. Analyses of traffic data from networks and services such as ISDN traffic and Ethernet LAN's have all convincingly demonstrated the presence of features such as self-similarity, long term dependence, slowly decaying variance and fractal dimensions.[10, 14].

In this experiment we applied the same principle similar to synthetic data experiment. A sample of network data is logged by the Spectrum NMP. There are 16 ports p_n on the routers that connect to 16 links, which in turn connect to 16 Ethernet subnets (S_n). Note that traffic has to flow through the router ports in order to reach the 16 subnets. Thus, we can observe the traffic that flows through the ports. There are three independent variables:

- *Load*: a measure of the percentage of bandwidth utilization of a port during a 10 minute period.
- *Packet Rate:* a measure of the rate at which packets are moving through a port per minute.
- *Collision Rate:* a measure of the number of packets during a 10 minute period that have been sent through a port over the link but have collided with other packets.

Data has collected for 18 weeks, from '94 to '95. There are 16,849 entries, representing measurements roughly every 10 minutes for 18 weeks. Fig. 3 illustrates an example of collected data for port #8.

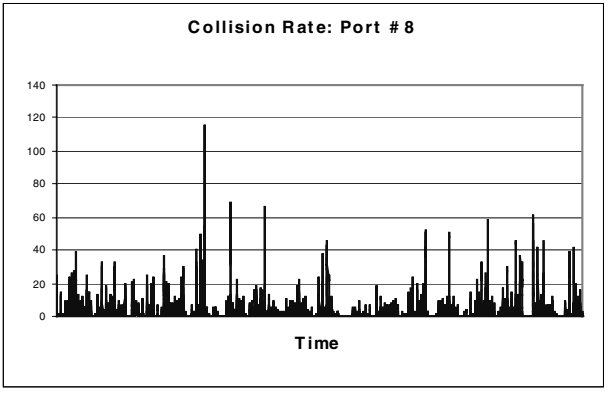
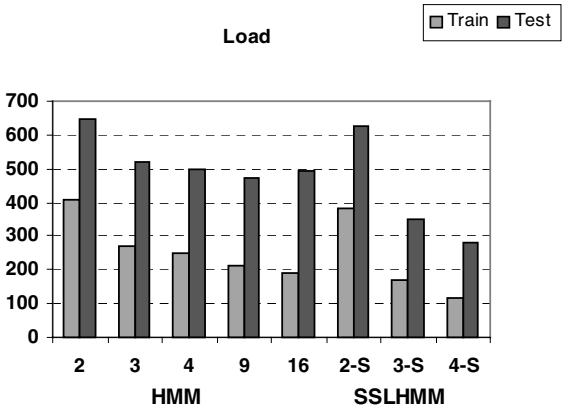


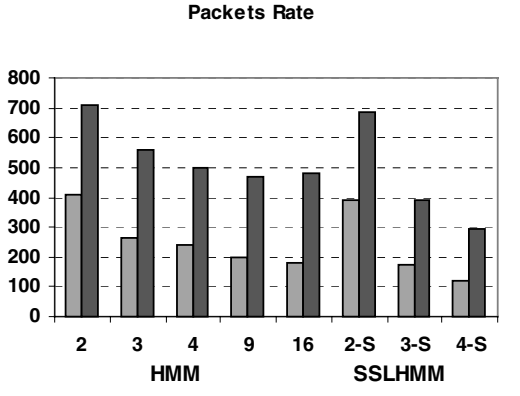
Fig. 3. The number of collisions of port #8. Data show self-similarity over different scales

We applied the HMM and SSLHMM to a given port of database with the purpose of modeling the Network data. We did test our technique through cross validation and in each round we trained the data with a random half of the data and test over the rest. We repeat the procedure for Load, Packet Rate and Collision Rate on all 16 ports. Fig. 4 illustrates the comparison of HMM and SSLHMM for Load, Packet Rate and Collision Rate. Respectively, we ran HMM with prior number of states equal to 2, 3, 4, 9 and 16, and SSLHMM with number of phases equal to 2, 3 and 4 (shown as 2-s, 3-s and 4-s in the Fig. 4). As it shows in Fig. 4 the SSLHMM model with $N=4$ outperforms other competitors in all series of experiments. Our experiment showed - *log(likelihood)* increases dramatically for models with number of sates grater than 16 as it over fits the data. The best SSLHMM performance beats the best HMM by 23%, 41% and 38% for Collision Rate, Load and Packets Rate respectively.

Our experiments show SSLHMM approach behave properly and does not perform worse than HMM even when the data is not self similar or when we do not have enough information. The SSLHMM provides a more satisfactory model of the network data from three point of views. First, the time complexity is such that it is possible to consider model with a large number of states in a hierarchy. Second, these larger number of states do not require excessively large numbers of parameters relative to the number of states. Learning a certain part of the whole structure is enough to extend to the rest of the structure. Finally SSLHMM resulted in significantly better predictors; the test set likelihood for the best SSLHMM was 100 percent better than the best HMM



.



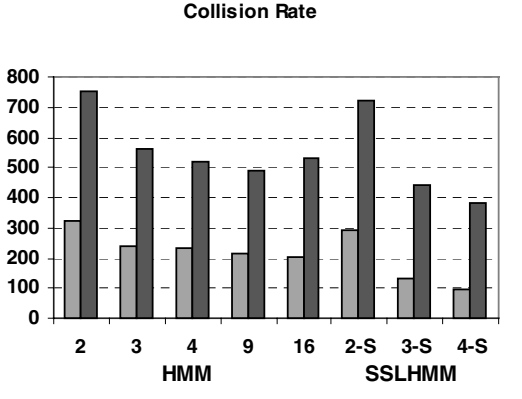


Fig. 4. The comparison of negative log likelihood for Network data for *Load, Packets Rate* and *Collision Rate*. SSLHMM outperform HMM in all three experiments

While the SSLHMM is clearly better predictor than HMM, it is easily interpretable than an HMM as well. The notion of phase may be considered as a collection of locally connected sets, groups, levels, categories, objects, states or behaviors as a collection of certain behavior and it comes with the idea of granularity, organization and hierarchy. As it mentioned before in Network application domain a phase could define as "congestion" or "stable". This characteristics is the main advantage of SSLHMM over other approaches such as FHMM [12]. SSLHMM is designed toward better interpretation as one the main goal of data mining approaches in general.

5 Conclusion and Future Work

Despite the relatively broad range of application areas, a general HMM, could not easily scale up to handle larger number of states. The error of predictive modeling will increased dramatically when the number of sates goes up. In this paper we proposed SSLHMM and illustrate it is a better estimation than flat HMM when data shows self-similar property. Moreover, we introduced three different types of self-similarity along with some result on synthetic data and experiments on Network data. Since SSLHMM has hierarchical structures and abstract states into phases, it overcomes, to a certain extent, the difficulty of dealing with larger number of states at the same layer, thus making the learning process move efficient and effective.

As future work we would like to extend this research to leverage the MLP power for precise prediction in both long term and short term. In addition we would like to extend this work when the model shows self-similar structure only at a limited range of structure scale. Currently we are in process of incorporation of self-similar property for Partially Observable Markov Decision Process (POMDP) along with generalization of SSLHMM.

Acknowledgement. This work was partially supported by the National Science Foundation under grant: NSF-IDM 9529615.

References

1. Adibi, J., Shen, W-M. *General structure of mining through layered phases*. submitted to *International Conference on Data Engineering*. (2002). San Jose, California, USA: IEEE.
2. Barbara, D. *Chaotic Mining: Knowledge discovery using the fractal dimension*. in *ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery (DMKD)*. (1999). Philadelphia, USA,.
3. Bertsekas, D.C., Tsitsiklis, N. J., *Parallel and distributed computation: Numerical Methods*. (1989), Englewood Clifffs, New Jersey: Printice-Hall.
4. Boutilier, R.I., Brafman, I., Geib, C. *Prioritized goal decomposition of Markov Decision Processes: Toward a synthesis of classical and decision theoretic planning*. in *IJCAI-97*. (1997). Nagoya, Japan.
5. Brand, N., Oliver, N., and Pentland, A.,. *Coupled Hidden Markov Models for complex action recognition*. in *CVPR*. (1997).
6. Chang, J.W., Glass, J, *Segmentation and modeling in segment-based recognition.* (1997).

7. Cohen, J., *Segmentation speech using dynamic programming.* ASA, (1981). **69**(5): p. 1430-1438.
8. Dean, T.L., S.H. *Decomposition techniques for planning in stochastic domains*. in *IJCAI*. (1995). Montreal, CA.
9. Faloutsos, C., Seeger, B., Traina, A. and Traina Jr., C. *Spatial Join selectivity using power law*. in *SIGMOD*. (2000). Dallas, TX.
10. Feldmann, A., Gilbert, A. C., Willinger, W., Kurtz, T.G., *The changing nature of Network traffic: Scaling phenomena.* ACM Computer Communication Review,, (1998). **28**(April): p. 5-29.
11. Fine, S., Singer Y, Tishby N, *The hierarchical Hidden Markov Model: Analysis and applications.* Machine Learning, (1998). **32**(1): p. 41-62.
12. Ghahramani, Z., Jordan, M., *Factorial Hidden Markov Models.* Machine Learning, (1997). **2**: p. 1-31.
13. Holmes, W.J., Russell, M. L., *Probablistic-trajectory segmental HMMs.* Computer Speech and Languages, (1999). **13**: p. 3-37.
14. Leland, W., Taqqu, M., Willinger, W., Wilson, D. *On the self-similar nature of Ethernet traffic*. in *ACM SIGComm*. (1993). San Francisco, CA.
15. Oates, T. *Identifying distinctive subsequences in multivariate time series by clustering*. in *KDD-99*. (1999). San Diego, CA: ACM.
16. Pearl, J., *Probabilistic reasoning in intelligence: Network of plausible inference*. (1988), San Mateo, CA: Morgan Kaufmann.
17. Precup, D., Sutton, R. S., *Multi-time models for temporally abstract planning*, in *NIPS-11*, M. Moser, Jordan, M.Petsche, T., Editor. 1998, MIT Press: Cambridge.
18. Puterman, M.L., *Markov Decision Process: discrete stochastic dynamic programming*. (1994), New Yourk: Wiley.
19. Rabiner, L.R., *A tutorial on Hidden Markov Models and selected applications in speech recognition.* IEEE, (1989). **7**(2): p. 257-286.
20. Sutton, R., Barto, A., *Reinforcement learning: An Introduction*. (1998), Cambridge: MIT Press.
21. Traina, C., Traina, A., Wu, L., and Faloutsos, C. *Fast feature selection using the fractal dimension*. in *XV Brazilian Symposium on Databases (SBBD)*. (2000). Paraiba, Brazil.
22. Vogler, C., Metaxas, D. *Parallel Hidden Markov Models for American Sign Language recognition*. in *International Conference on Computer Vision*. (1999). Kerkyra, Greece.
23. Williams, C., Hinton, G.E., ed. *Mean field networks that leat discriminate temporally distorted strings*. , ed. D. Touretzkey, Elman, J., Sejnowski, T. and Hinton G. (1991).
24. Willinger, W., Taqqu M. S., Erramilli, A., ed. *A bibliographical guide to self-similar trace and performance modeling for modern high-speed networks*. Stochastic Networks: Theory and Applications, ed. F.P. Kelly, Zachary, S. and Ziedins, I. (1996), Clarendon Press, Oxford University Press: Oxford. 339-366.