Hidden Markov Model to Extract Leucine Zipper Motif

Yukiko Fujiwara [1] Minoru Asogawa [2] Akihiko Konagaya [2]
yukiko@csl.cl.nec.co.jp asogawa@csl.cl.nec.co.jp konagaya@csl.cl.nec.co.jp

[1] C&C Research Laboratories, NEC Corporation
4-1-1, Miyazaki, Miyamaeku, Kawasaki, Kanagawa 216, Japan
[2] Massively Parallel Systems NEC Laboratory, RWCP [3]
c/o C&C Research Laboratories, NEC Corporation
4-1-1, Miyazaki, Miyamaeku, Kawasaki, Kanagawa 216, Japan

Abstract

To represent motifs of amino acid sequences using Hidden Markov Models (HMMs) with high accuracy, the HMM topology must be specified according to the motif's characterisitcs. For this purpose, the "iterative duplication method", which learns the optimal HMM topology, was developed. In this method, a small fully-connected HMM was gradually expanded by a state splitting and a transition deleting. However, the method did not clearly determine a splitting state, because it randomly selected one of the mostly connected states. To determine a splitting state, we improve the iterative duplication method. The improved method selects the most ambiguous state for splitting. Since this ambiguity relies on the transition probabilities and observation distributions, the splitting state can be determined. Additionally, the improved method considers negligible state deletion. In an experiment, an HMM is obtained for a leucine zipper motif using this improved method. The prediction accuracy of this HMM is 96.48 percent. It is Compared with that of the HMM obtained by the previous method and the fully-connected HMM estimation method. The accuracy of the previous method was 95.85 percent and that of the fully-connected HMM was 95.22 percent.

[1]藤原 由希子： NEC C&C 研究所 コンピュータ・システム研究部，〒 216 川崎市宮前区宮崎 4 丁目 1-1
[2]麻生川 稔，小長谷 昭彦： RWCP 超並列 NEC 研究室，〒 216 川崎市宮前区宮崎 4 丁目 1-1
[3]Real World Computing Partnership

1 Introduction

Recently, HMMs have been applied to protein modeling. For example, Haussler et al. used HMMs for protein modeling and multiple alignment[1]. Baldi et al. used HMMs for globins, immunoglobins and kinases[2]. Since their goals were alignment, they used special models, one of the left-to-right models. They prepared delete states (for gaps) and insertion states. In these models, states were linearly connected with self-loop transitions; once it leaves a state, that state never revisited. Therefore, these HMMs could not represent periodic topology, such as global loops for α helices. Also, these studies did not treat the OR rule, such as different families or types in protein sequences, because the data for alignment was considered coming from one family and the alignment score must be calculated.

However, it is necessary to use direct periodic topology or the OR rule to represent motifs. The direct periodic topology is used to represent cyclic structures. For example, a leucine zipper motif constructs an α helix with the various lengths from 15 to 43. If direct periodic topology is not used, it must be consider that which turns of the long sequences correspond to the first turns of the 15 length sequences. The OR rule is used to represent different families or types in motifs. Therefore, to represent motifs, HMMs had better contain the direct periodic topology and the OR rule. Asai et al used HMMs containing direct periodic topology for secondary structure prediction[3]. They proposed a method to construct a large HMM from smaller HMMs using a grammar of protein structure. However, the grammar depends on human knowledge at the current stage. Consequently, HMM topology decision is one of the problems to use general HMMs.

In speech recognition, there were some studies for HMM topology learning. For example, Hwang and Huang studied a merging method, which learns from a large model and merges states. Takami proposed a successive state splitting (SSS) algorithm, which learns from small models and splits states[4]. However, for speech recognition, it was assumed that the distribution of the utterances was continuous and the models were left-to-right models. Amino acid sequences, on the other hand, have discrete distributions of amino acids. Furthermore, we do not want to restrict the model in left-to-right models. Thus, SSS algorithm can not be applied directly.

To represent motifs, the "*iterative duplication method*" was developed for HMM topology learning[6][7][8]. In this method, a small fully-connected HMM was gradually expanded by a state splitting and a transition deleting. For splitting-state selection, the iterative duplication method selected one of the mostly connected state. However, this did not clearly determine the splitting state, because sometimes several states had the same number of incoming and outgoing transitions. When the target motif is complex, the HMM may have many states, this maybe turn into a big problem. Therefore, we improve this splitting-state selection method. The improved selection method relies on the transition ambiguity and the observation ambiguity of states, while the previous method relied only on the transition ambiguity. Additionally, the improved method considers a state deletion as well as the transition deletion.

Using this method, we study the problem of representing amino acid sequences by a stochastic motif[5]. Motifs are the preserved sites in the evolution process and are considered to represent the functions or structure of proteins. In an experiment, an HMM for a leucine zipper motif is obtained using the improved method. The prediction accuracy of this HMM compares with that of the HMM obtained by the previous method and that of the HMM obtaind by

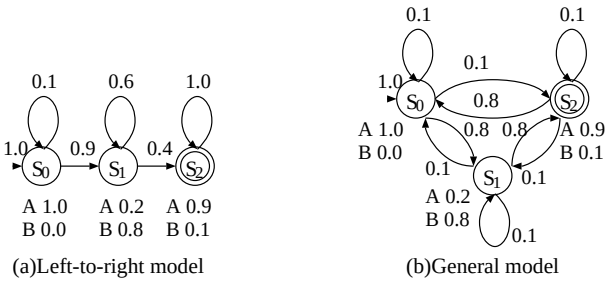

Figure 1: Two examples of HMMs

training the fully-connected HMM.

This paper is organized as follows. First, we explain the HMMs and notations used in this paper. Then, we present the iterative duplication method and its improvement. After that, we give the experimental results of a leucine zipper motif.

2 HMMs

An HMM is characterized by a network with a finite number of parameters and states[9]. Parameters represent initial probabilities, transition probabilities, and observation probabilities. At discrete instants of time, the process is assumed to be at one state. An observation (or output symbol) is generated by the observation probability corresponding to the current state. This state then changes depending upon its transition probability.

Figure 1 shows two examples of an HMM, a left-to-right model and a general model. A left-to-right model is widely used in the speech recognition. In this model, states are linearly connected with self-loop transitions. In the model described in figure 1(a), the process is in the state S_0 initially, observes A with the probability 1.0, stays at or changes the state with the probability 0.1 or 0.9 and continues. The initial state is shown as the pointing arrow with the probability 1.0. The final state is S_2 shown as a double circle. On the other hand, a general model allows any transition. It can represent periodic topology directly. For example, the model in figure 1(b) can represent a loop with period 3. Note that general HMMs include left-to-right models.

Let $V = \{v_0, v_1, \cdots, v_K\}$ be the alphabet, and let $y = y_0 y_1 \cdots y_{T-1}$ $(y_t \in V)$ be the observations, the sequence of data. In the case of amino acid sequences, the alphabet is the set of amino acids. Thus, K equals 20. Suppose that the HMM has N states $S_0, S_1, \cdots, S_{N-1}$, initial probabilities $\pi = (\pi_0, \pi_1, \cdots, \pi_{N-1})$, transition probabilities $A = [a_{ij}]$ $(0 \le i, j \le N-1)$ and observation probabilities $B = [b_{jk}]$ $(0 \le j \le N-1, \ \ 0 \le k \le K-1)$. Here, π_i is the probability of S_i at the initial time, and a_{ij} is the probability of transiting from S_i to S_j. b_{jk} is the probability of output symbol v_k at the state S_j. For the given HMM M, the probability of observing the sequence y is calculated by

$$Prob(y|M) = \sum_{i_0, i_1, \cdots, i_{T-2}} \pi_{i_0} a_{i_0 i_1} b_{i_1}(y_0) \cdots a_{i_{T-2} F} b_F(y_{T-1}),$$

where $b_i(y_t)$ denotes b_{ik} $(y_t = v_k)$.

For a motif representation, positive data which is a set of sequences annotated as a motif and negative data which is a set of other sequences are given. HMM parameters and topology are estimated to represent a part of positive data. Then, the threshold value is decided based on the probabilities of observing the rest of the positive data, which are not used for training, and the negative data on the learned HMM. After that, if a new sequence having unknown functions or structures is obtained, we can predict whether this sequence has the motif or not by comparing the probability of observing the sequence on the learned HMM with the threshold value.

3 Methods

3.1 Iterative duplication method

In the following, we present the iterative duplication method[7]. The method started from a small fully-connected HMM. It iterated a parameter estimation phase followed by a topology modification phase using a part of the data. It terminated when the accuracy of the remaining data is maximum. In the parameter estimation phase, the Baum-Welch algorithm was used on several initial HMMs with random parameters to avoid converging at a local maximum. After that, the best HMM was chosen based on the likelihood of training data. In the topology modification phase, the transitions with negligible transitional probabilities were deleted. These probabilities were less than $\epsilon = \max(\epsilon_1, r)$, where ϵ_1 was a smoothing value and r was a convergence radius[9]. The purpose of deleting the negligible transitions was to keep the topology space small and eventually to reduce the computation for parameter estimation. Next, the splitting state S_c was selected based on the number of incoming and outgoing transitions. Then a copy of it (denoted as S_{new}) was made so that S_{new} had the same transitions as S_c (see figure 2 (a)). If the state S_c had a self-loop, S_{new} had a self-loop and transitions from S_c to S_{new} and from S_{new} to S_c (see figure 2 (b)).

In each step, we want to split an overlapping state, because splitting the overlapping state grows the HMM topology after parameter estimation. The splitting state selection is based on the ambiguity in terms of transitions. However, in the case that several states have the same transition ambiguity, this criteria did not clearly determine the splitting state. Thus, the method is improved.

3.2 Improvement

The improved iterative duplication method goes the similar process to the iterative duplication method(see Figure 3). In the topology modification phase, after the negligible transition deletion, the negligible state deletion is added to the iterative duplication method. This is performed by deleting the states with negligible initial probabilities and negligible incoming transitions of which the sums of the initial probabilities and the incoming transition probabilities are less than $min(\epsilon_1, r)$. Since these states are scarcely visited, these states do not play role for computing the likelihood. This negligible state deletion has a good effect for selecting the splitting state.

The splitting state is selected based on the transition ambiguity and the observation ambiguity instead of the number of transitions. In terms of transition ambiguity, states with many

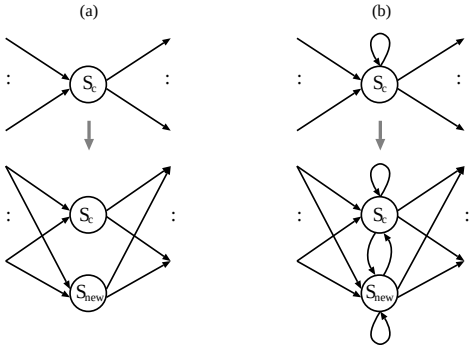

Figure 2: A state copy

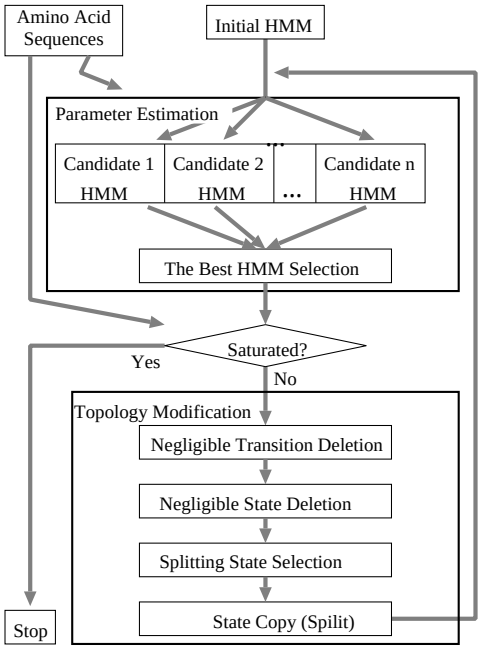

Figure 3: Improved iterative duplication method

transitions with large probabilities are desirable candidates to be split. Since these states are considered to be ambiguous in transitions before or after the states, these tend to be overlapping states. Thus, the transition ambiguity of the state S_j is defined as $in(j) + out(j)$, where, $in(j) = \pi_j + \sum_i a_{ij}$ and $out(j) = \sum_k a_{jk}/in(k)$. This definition counts the number and the probabilities of the transitions, while the previous method only counts the number of the transitions.

In terms of observation ambiguity, states with large variances in observation probabilities are good for splitting at first thought. However, in the case of motif representation, large observation variance often occurs in an unimportant state corresponding to 'X' (any amino acid). If this unimportant state S_c is split into S_c and S_{new}, these two states S_c and S_{new} tend to be estimated to have the similar transitions and the similar observations. This does not elevate the likelihood and accuracy. Thus, the observation ambiguity is defined as the estimated observation difference of two states when the state S_c is split. To estimate this difference, the observation distribution at the state S_i on the condition that the previous observation is v_k is estimated. Then, these distributions at state S_i are divided into two groups, by gathering the similar distribution repeatedly. The distribution difference of the two groups is defined by the error squared. The observation ambiguity at the state S_i of prefix is defined as the distribution difference of two groups. Similarly, the observation distribution at the state S_i on the condition that the next observation is v_k is estimated and the observation ambiguity at the state S_i of suffix is defined. The observation ambiguity is defined the sum of the distribution difference of prefix and suffix.

Then, the sum of the regularized ambiguities is calculated, and the state with the maximum value is chosen.

3.3 Prediction

Prediction is performed by comparing a likelihood, that is a probability of generating a given sequence. If a sequence achieves higher likelihood than the threshold value, then it is predicted to have the target motif, that is, the sequence has the same structure and/or function of the motif. The threshold value α for classification is defined to maximize the accuracy of the rest training data.

The positive prediction performance is measured by $1.0 - E^+/N^+$, where N^+ is the number of positive test data, and E^+ is the number of errors occurred in the positive test data, that is, the number of the positive test data categorizing as negative by the HMM. Similarly, the negative prediction performance is measured by $1.0 - E^-/N^-$, where N^- is the number of negative test data, and E^- is the number of negative errors. In order to regularize the positive and negative performance, the total prediction performance E_{ac} is measured by the following equation:

$$E_{ac} \equiv 1.0 - \frac{1}{2} \times (\frac{E^+}{N^+} + \frac{E^-}{N^-}).$$

This value represents the mean of positive and negative prediction performance.

4 Experimental Results

The improved method is applied to a leucine zipper motif. As this motif has a weak pattern, no data set is in the PROSITE database. Thus, positive data is the subsequences annotated as leucine zipper (like) in the SWISS-PROT database Release 30. Negative data are the collection of subsequences which are not annotated as leucine zipper (like). There are 183 positive data and 377429974 negative data. In order to evaluate the prediction performance, positive or negative data are divided into 3 groups. The first group is used for learning the HMM topology and parameters. The second group is used for selecting the model size and the threshold values. Thus, these two groups are closed data. The third is open data for testing, which is eliminated the sequences included in the first or the second groups.

In order to assure the results, the results are compared to the HMMs trained with fully-connected HMMs. These fully-connected HMMs are trained as follows. For each number of states, one hundred initial fully-connected HMMs with random parameters are trained to avoid converging in the local maximum. Then, the best HMM on the likelihood of training data is chosen and the negligible transitions are deleted. This method takes quite a bit training time because it must estimate the numerous free parameters (see [7]).

The improved method achieved similar likelihood to the fully-connected method or the previous method (see figure 4). In figure 4, the likelihood of the improved method is shown as thick line (Imp), the likelihood of the previous method is shown as broken line (Pre), the likelihood of the fully-connected method is shown as thin line (Ful).

Figure 5 and 6 show the accuracy results, the relationship between the number of states and the accuracy. In figure 5, t shows the accuracy of the training data and s shows the accuracy of the data used for determining the threshold value. The improved method is superior to the others for the open data at a lot of steps (the number of states). One of the reasons why the accuracy changes irregularly is that the number of positive data is small. Another reason is that the annotation errors exist even in the positive data.

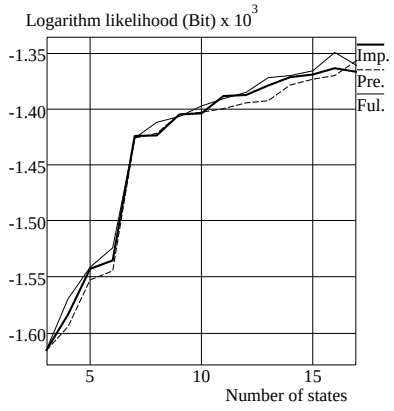

Figure 4: Likelihood

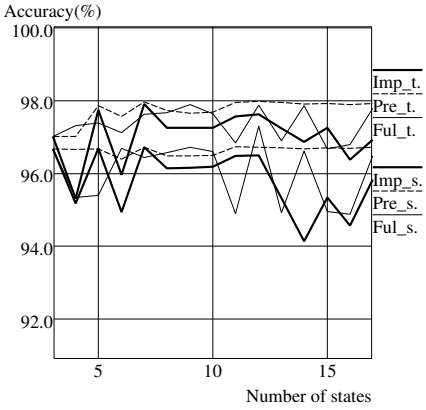

Figure 5: Accuracy (closed)

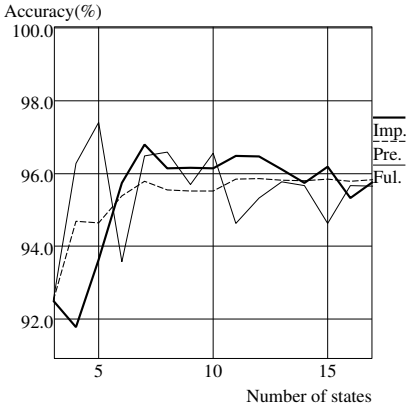

Figure 6: Accuracy (open)

Table 1: Accuracy

	Closed						Open		
	Training			Selecting			test		
	pos.	neg.	ave.	pos.	neg.	ave.	pos.	neg.	ave.
Imp.	100.00%	95.14%	97.57%	96.61%	96.36%	96.49%	97.87%	95.09%	96.48%
Pre.	100.00%	95.90%	97.95%	96.61%	96.85%	96.73%	95.74%	95.95%	95.85%
Ful.	98.46%	97.30%	97.88%	96.61%	97.99%	97.30%	93.62%	97.05%	95.22%

Table 1 shows the accuracy obtained by each method. The accuracy of the improved method is 96.48 percent for open data, while the accuracy of the previous method was 95.85 percent and the accuracy of the fully-connected HMM was 95.22 percent.

Figure 7 shows topology of HMMs. The number in quadrilateral represents the logarithm likelihood of each model. There are some differences between them. In particular, topology of (b) or (c) is very different from that of (a) in 6 states. However, they eventually become similar to one another in 7 states.

5 Conclusion

We improved the iterative duplication method, which learns the optimal HMM topology by deleting transitions and splitting states. To determine a splitting state, the improved method selects the most ambiguous state for splitting. This ambiguity relies on both the transition ambiguity and the observation ambiguity, while the previous method relied only on the transition ambiguity. Since this ambiguity relies on the transition probabilities and observation distributions, the splitting state can be determined. Additionally, the improved method considers negligible state deletion. In an experiment, we obtain an HMM for a leucine zipper motif using the improved method. Compared with the previous methods and fully-connected HMM, the improved method is superior in terms of the prediction accuracy.

References

[1] D. Haussler, A. Krogh, I. Mian and K. Sjolander, "Protein Modeling using Hidden Markov Models: Analysis of Globins", *Proceedings of Hawaii International Conference on System Science*, pp. 792-802, 1993.

[2] P. Baldi, Y. Chauvin, T. Hunkapiller and M. A. McClure, "Hidden Markov Models of Biological Primary Sequence Information", *Proceedings of the National Academy of Sciences of the USA*, pp. 1059-1063, 1994.

[3] K. Asai, S. Hayamizu and K. Onizuka, "HMM with Protein Structure Grammar", *Proceedings of Hawaii International Conference on System Science*, pp. 783-791, 1993.

[4] J.Takami and S.Sagayama, "Automatic Generation of the Hidden Markov Network by Successive State Splitting", *Proceedings of ICASSP*, 2-5-13, 1991.

[5] A. Konagaya and M. Kondo, "Stochastic Motif Extraction using a Genetic Algorithm with the MDL principle", *Proceedings of Hawaii International Conference on System Science*, pp. 746-753, 1993.

[6] Y. Fujiwara and A. Konagaya, "Protein Motif Extraction using Hidden Markov Model", *Proceedings Genome Informatics Workshop IV*, pp. . 56-64, 1993.

[7] Y. Fujiwara, M. Asogawa and A. Konagaya, "Stochastic Motif Extraction using Hidden Markov Model", *Proceedings Intelligent Systems for Molecular Biology*, pp. 121-129, 1994.

[8] M. Asogawa, Y. Fujiwara and A. Konagaya, "A Combination of a Functional Motif Model and a Structual Motif Model for a Database Validation", *Proceedings of Hawaii International Conference on System Science*, PP. 174-183, 1995.

[9] S. Nakagawa, "Speech Recognition Using Stochastic Models", *Electronic Society of Information Communication*, pp. 29–108, 1988.

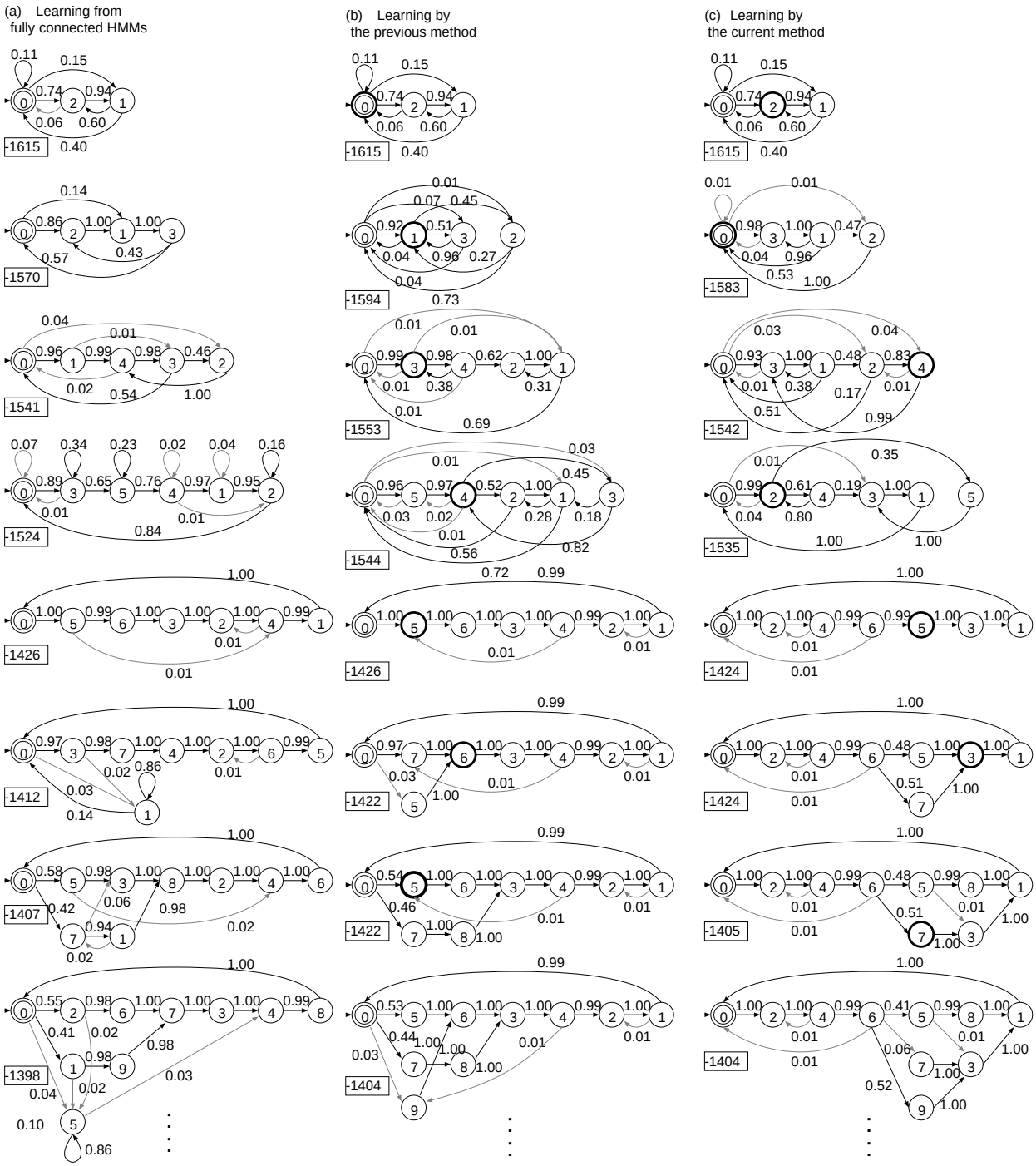

Figure 7: Topology of HMMs