Forward-Backward Activation Algorithm for Hierarchical Hidden Markov Models

Authored by:

Kei Wakabayashi
Takao Miura

Abstract

Hierarchical Hidden Markov Models (HHMMs) are sophisticated stochastic models that enable us to capture a hierarchical context characterization of sequence data. However, existing HHMM parameter estimation methods require large computations of time complexity O(TN{2D}) at least for model inference, where D is the depth of the hierarchy, N is the number of states in each level, and T is the sequence length. In this paper, we propose a new inference method of HHMMs for which the time complexity is O(TN{D+1}). A key idea of our algorithm is application of the forward-backward algorithm to "state activation probabilities". The notion of a state activation, which offers a simple formalization of the hierarchical transition behavior of HHMMs, enables us to conduct model inference efficiently. We present some experiments to demonstrate that our proposed method works more efficiently to estimate HHMM parameters than do some existing methods such as the flattening method and Gibbs sampling method.

1 Paper Body

Latent structure analysis of sequence data is an important technique for many applications such as speech recognition, bioinformatics, and natural language processing. Hidden Markov Models (HMMs) play a key role in solving these problems. HMMs assume a single Markov chain of hidden states as the latent structure of sequence data. Because of this simple assumption, HMMs tend to capture only local context patterns of sequence data. Hierarchical Hidden Markov Models (HHMMs) are stochastic models which assume hierarchical Markov chains of hidden states as the latent structure of sequence data [3]. HHMMs have a hierarchical state transition mechanism that yields the capability of capturing global and local sequence patterns in various granularities. By their nature, HHMMs are applicable to problems of many kinds including handwritten letter recognition [3], information extraction from documents [11], musical pitch structure modeling [12], video structure modeling [13], and human activity modeling [8, 6]. For conventional HMMs, we can conduct unsupervised learning efficiently using the forwardbackward algorithm, which is a kind of dynamic programming [9]. In situations where few or no supervised data are available, the existence of the efficient unsupervised learning algorithm is a salient advantage of using HMMs. The unsupervised learning of HHMMs is an important technique, as it is for HMMs. In this paper, we discuss unsupervised learning techniques for HHMMs. We introduce a key notion, activation probability, to formalize the hierarchical transition mechanism naturally. Using this notion, we propose a new exact inference algorithm which has less time complexity than existing methods have. The remainder of the paper is organized as follows. In section 2, we overview HHMMs. In section 3, we survey HHMM parameter estimation techniques proposed to date. In section 4, we introduce our parameter estimation algorithm. Section 5 presents experiments to show the effectiveness of our algorithm. We conclude our discussion in section 6. 1

Figure 1: (left) Dynamic Bayesian network of the HHMM. (top-right) Tree representation of the HHMM state space. (bottom-right) State identification by the absolute path of the tree.

2

Hierarchical Hidden Markov Models

Let O = {O1 , ..., Ot , ..., OT } be a sequence of observations in which subscript t denotes the time in the sequence. We designate time as an integer index of observation numbered from the beginning of the sequence. HHMMs define Qdt for 1 ? t ? T, 1 ? d ? D as a hidden state at time t and level d, where d = 1 represents the top level and d = D represents the bottom level. HHMMs also define binary variables Ftd , called termination indicators. If Ftd = 1, then it is indicated that the Markov chain of level d terminates at time t. In HHMMs, a state transition at level d is permitted d+1 only when the Markov chain of level d + 1 terminates, i.e. Qdt = Qdt?1 if Ft?1 = 0. A terminated Markov chain is initialized again at the next time. Figure 1 (left) presents a Dynamic Bayesian Network (DBN) expression for an HHMM of hierarchical depth D = 3. The conditional probability distribution of Q, F and O is defined as follows [7]. ? (if b = 0) ? ?(i, j) d+1 d d A (i, j) (if b = 1, f = 0) = f, Q1:d?1 = b, Ft?1 = k) = p(Qdt = j—Qdt?1 = i, Ft?1 t ? ? dk(j) (if b = 1, f = 1) k { 0 (if b = 0) p(Ftd = 1—Qdt = i, Q1:d?1 = k, Ftd+1 = b) = t Adk (i, end) (if b = 1) p(Ot = v—Q1:D = k) = Bk (v) t }. Probabilities of the initializaWe use a notation Q1:d?1 as a combination of states {Q1t , ..., Qd?1 t t tion and the state transition of Markov chains at level d depend on all higher states Q1:d?1 . Adk (i, j) is a model parameter of the transition probability at level d from state i to j when Q1:d?1 = k. t Adk (i, end) denotes a termination probability that state i terminates the Markov chain at level d when Q1:d?1 = k. ?kd (j) is an initial state probability of state j at level d when Q1:d?1 = k. Bk (v) t t is an output probability of observation v when Q1:D = k. t A state space of HHMM is expressed as a tree structure [3]. Figure 1 (top-right) presents a tree expression of state space of an HHMM for which the depth D = 3 and the number of states in each level N = 3. The level of the tree corresponds to the level of HHMM states. Each node at level d corresponds to a combination of states Q1:d . Each node has N children because there are N possible states for each level. The rectangles in the figure denote local HMMs in which nodes can mutually transit directly using the transition probability A. For the analysis described herein, we assume the balanced N-ary tree to simplify discussions of computational complexity. However, arbitrary state space trees do not change the substance of what follows. The behavior of Markov chain at level d depends on the combination of all higher-up states Q1:d?1 , not only on the individual Qd . In the tree structure, the absolute path which corresponds to Q1:d is meaningful, rather than the relative path which corresponds to Qd . We refer to Q1:d as Z d and call it absolute path state. Figure 1 (bottom-right) presents an absolute path state identification. The set of values taken by an absolute path state at level d, denoted by ?d , contains N d elements in the balanced N-ary tree state space. We define a function to obtain the parent absolute path state of Z d as parent(Z d). Similarly, we define a function to obtain the set of child absolute path states of Z d as child(Z d), and a function to obtain the set of siblings of Z d as sib(Z d) = child(parent(Z d)). 2

Table 1: Notation for HHMMs. D N ?d Ztd ? ?d Ftd ? {0, 1} Ot ? {1, ..., V } Adij AdiEnd ?di Biv

Depth of hierarchy Number of states in each level Set of values taken by absolute path state at level d Absolute path state at time t and level d Termination indicator at time t and level d Observation at time t d State transition probability from state Ztd = i to state Zt+1 = j at level d Termination probability of Markov chain at level d from state Ztd = i Initial state probability of state Z d = i at level d Output probability of observation v with Z D = i

Table 1 presents the notation used for the HHMM description. We use the notation of the absolute path state Z d rather than Qd throughout the paper. Therefore, we define compatible notations for the model parameters. Whereas the conventional notation ?kd (j) denotes the initial state probability of Qd = j when Q1:d?1 = k, we aggregate Qd and Q1:d?1 into Q1:d = Z d and define ?di as the initial state probability ? of Z d = i. Similarly, we ?define Adij as the state transition probability from d Z = i to j. Note that i0 ?sib(i) ?di0 = 1 and j 0 ?{sib(i)?End} Adij 0 = 1.

3

Existing Parameter Estimation Methods for HHMMs

The first work for HHMMs [3] proposed the generalized Baum-Welch algorithm. This algorithm is based on an inside-outside algorithm used for inference of probabilistic context free grammars. This method takes O(T 3) time complexity, which is not practical for long sequence data. A more efficient approach is the flattening method [7]. The hierarchical state sequence can be reduced to a single sequence of the bottom level absolute path states {Z1D , ..., ZTD }. If we regard Z D as a flat HMM state, then we can conduct the inference by using the forward-backward algorithm with O(T N 2D) time complexity since —?D — = N D . Notice that the flat state Z D can transit to any other flat state, and we cannot apply efficient algorithms for HMMs of sparse transition matrix. In the flattening method, we must make a weak constraint on the HHMM parameters, say minimally self-referential (MinSR) [12], which restricts the self-transition at higher levels i.e. Adii = 0 for 1 ? d ? D ? 1. The MinSR constraint enables us to identify the path connecting two flat states uniquely. This property is necessary for estimating HHMM parameters by using the flattening method. We also discuss a sampling approach as an alternative parameter estimation technique. The Gibbs sampling is often used for parameter estimation of probabilistic models including latent variables [4]. We can estimate HMM parameters using a Gibbs sampler, which sample each hidden states iteratively. This method is applicable to inference of HHMMs in a straightforward manner on the flat HMM. This straightforward approach, called the Direct Gibbs Sampler (DGS), takes the O(T N D) time complexity for a single iteration. The convergence of a posterior distribution by the DGS method is said to be extremely slow for HMMs [10] because the DGS ignores long time dependencies. Chib [2] introduced an alternative method, called the Forward-Backward Gibbs Sampler (FBS), which calculates forward probabilities in advance. FBS samples hidden states from the end of the sequence regarding the forward probabilities. FBS method requires larger computations for a single iteration than DGS does, but it can bring a posterior of hidden states to its stationary distribution with fewer iterations [10]. Heller [5] proposed Infinite Hierarchical Hidden Markov Models (IHHMMs) which can have an infinitely large depth by weakening the dependency between the states at different levels. They proposed the inference method for IHHMMs based on a blocked Gibbs sampler of which the sampling unit is a state sequence from t = 1 to T at a single level. This inference takes only O(T D) time for a single iteration. In HHMMs, the states in each level are strongly dependent, so resampling a state at an intermediate level causes all lower states to alter into a state which has a completely different behavior. Therefore, it is not practical to apply this Gibbs sampler to HHMMs in terms of the convergence speed. 3

4

Forward-Backward Activation Algorithm

In this section, we introduce a new parameter estimation algorithm for HHMMs, which theoretically has O(T N D+1) time complexity. The basic idea of our algorithm is a decomposition of the flat D transition probability distribution p(Zt+1 —ZtD), which the flattening method calculates directly for all pairs of the flat states. We can rewrite the flat transition probability distribution into a sum of two cases that the Markov chain at level D terminates or not, as follows. D D p(Zt+1 —ZtD) = p(Zt+1 —ZtD , FtD = 0)p(FtD = 0—ZtD) + D?1 D?1 D p(Zt+1 —Zt+1 , FtD = 1)p(Zt+1 —ZtD?1 , FtD = 1)p(FtD = 1—ZtD) The first term corresponds to the direct transition without the Markov chain termination. The actual computational complexity for calculating this term is O(N D+1) because the direct transition is permitted only between the sibling states, i.e. ADij = 0 if j ? / sib(i). The second term, corresponding to the case in which the Markov chain terminates at level D, contains two factors: The D?1 upper level transition probability p(Zt+1 —ZtD?1 , FtD = 1) and the state initialization probability D?1 D for the terminated Markov chain p(Zt+1 —Zt+1 , FtD = 1). We attempt to compute these probability distributions efficiently in a dynamic programming manner. d The transition probability at level d has the form p(Zt+1 —Ztd , Ftd+1 = 1). We define ending activad tion et , as the condition of the transition probability from Ztd , formally: ? ? p(Ztd = i, Ftd+1 = 1) (if i 6= null and d ¡ D) d p(et = i) = p(Ztd = i) (if i 6= null and d = D) ? p(Ftd+1 = 0) (if i = null)

The null value in edt indicates that the Markov chain at level d + 1 does not terminate at time t. d+1 = 1). We define The state initialization probability for level d + 1 has the form p(Ztd+1 —Ztd , Ft?1 d beginning activation bt , as the condition of the state initialization probability from Ztd , formally, as ? d+1 = 1) (if i 6= null and d ¡ D and t ¿ 1) ? p(Ztd = i, Ft?1 d d p(bt = i) = p(Zt = i) (if i 6= null and (d = D or t = 1)) ? d+1 p(Ft?1 = 0) (if i = null) The null value in bdt indicates that the Markov chain at level d + 1 does not terminate at time t ? 1. Using these notations, we can represent the flat transition with propagations of activation probabilD D —ZtD) = p(bD ities as shown in figure 2 (left) because p(Zt+1 t+1 —et). This representation naturally describes the decomposition of the flat transition probability distribution discussed above, and it enables us to apply the decomposition recursively for all levels. We can derive the conditional probability distributions of edt and bdt+1 as { ? d+1 = c)A(d+1)cEnd (if i 6= null) c?child(i) p(et ? p(edt = i—ed+1) = t d+1 d+1 = c)(1?A(d+1)cEnd)+p(et = null) (if i = null) c??d+1 p(et { ? d?1 p(bt+1 = parent(i))?di + j?sib(i) p(edt = j)Adji (if i 6= null) d d d?1 p(bt+1 = i—et , bt+1) = p(edt = null) (if i = null) In the following subsections, we show the efficient inference algorithm and the parameter estimation algorithm using the activation probabilities. 4.1

Inference using Forward and Backward Activation Probabilities

We can translate the DBN of HHMMs in figure 1 (left) equivalently into simpler DBN using activation probabilities. The translated DBN is portrayed in figure 2 (right). The inference algorithm proposed herein is based on a forward-backward calculation over this DBN. We define forward activation probability ? and backward activation probability ? as follows. ?edt (i) = p(edt = i, O1:t) ?bdt (i) = p(bdt = i, O1:t?1) ?edt (i) = p(Ot+1:T , FT1 = 1—edt = i) ?bdt (i) = p(Ot:T , FT1 = 1—bdt = i) 4

Figure 2: (left) Propagation of activation probabilities for calculating the flat transition probability from time t to t + 1. (right) Equivalent DBN of the HHMM using activation probabilities. Algorithm 1 Calculate forward activation probabilities 1: for t = 1 to T do 2: if t = 1 then 3: ?b11 (i ? ?1) = ?1i 4: for d = 2 to D do 5: ?bd1 (i ? ?d) = ?bd?1 (parent(i))?di 1 6: end for 7: else ? 8: ?b1t (i ? ?1) = j?sib(i) ?e1t?1 (j)A1ji 9: for d = 2 to D do ? 10: ?bdt (i ? ?d) = ?bd?1 (parent(i))?di + j?sib(i) ?edt?1 (j)Adji t 11: end for 12: end if 13: ?eD (i ? ?D) = ?bD (i)BiOt t t 14: for d = D ? 1 to 1?do 15: ?edt (i ? ?d) = c?child(i) ?ed+1 (c)A(d+1)cEnd t 16: end for 17: end for

These probabilities are efficiently calculable in a dynamic programming manner. Algorithm 1 presents the pseudocodes to calculate whole ?. ?bdt are derived downward from ?b1t to ?bD by t summing up to the initialization probability from the parent and the transition probabilities from the siblings (Line 8 to 11). ?edt are propagated upward from ?eD to ?e1t by summing up to the probabilt ities of the child Markov chain termination (Line 13 to 16). This algorithm includes the calculation of —?d — = N d quantities involving the summation of

—sib(i)— = N terms for d = 1 to D and for ?D t = 1 to T . Therefore, the time complexity of algorithm 1 is O(T d=1 N d+1) = O(T N D+1). Algorithm 2 propagates the backward activation probabilities similarly in backward order. We can derive the conditional independence of O1:t and {Ot+1:T , FT1 = 1} given edt 6= null or bdt+1 6= null, because both of edt 6= null and bdt+1 6= null indicates that the Markov chains at level d + 1, ..., D terminates at time t. On the basis of this conditional independence, the exact inference of a posterior of activation probabilities can be obtained using ? and ? as presented below. p(edt = i—O1:T , FT1 = 1) ? p(edt = i, O1:t)p(Ot+1:T , FT1 = 1—edt = i) = ?edt (i)?edt (i) p(bdt = i—O1:T , FT1 = 1) ? p(bdt = i, O1:t?1)p(Ot:T , FT1 = 1—bdt = i) = ?bdt (i)?bdt (i) The inference of the flat state p(ZtD —O1:T , FT1 = 1) is identical to of the bottom level activation 1 probability p(eD t —O1:T , FT = 1). We can calculate the likelihood of the whole observation as follows. ? ? p(O1:T , FT1 = 1) = p(e1T = i, O1:T)p(FT1 = 1—e1T = i) = ?e1T (i)?e1T (i) i??1

i??1

5

Algorithm 2 Calculate backward activation probabilities 1: for t = T to 1 do 2: if t = T then 3: ?e1T (i ? ?1) = A1iEnd 4: for d = 2 to D do 5: ?ed (i ? ?d) = ?ed?1 (parent(i))AdiEnd T T 6: end for 7: else ? 8: ?e1t (i ? ?1) = j?sib(i) ?b1t+1 (j)A1ij 9: for d = 2 to D do ? 10: ?edt (i ? ?d) = ?ed?1 (parent(i))AdiEnd + j?sib(i) ?bdt+1 (j)Adij t 11: end for 12: end if 13: ?bD (i ? ?D) = ?eD (i)BiOt t t 14: for d = D ? 1 to 1?do 15: ?bdt (i ? ?d) = c?child(i) ?bd+1 (c)?(d+1)c t 16: end for 17: end for 4.2

Updating Parameters

Using the forward and backward activation probabilities, we can estimate HHMM parameters effi? is defined, where ? is a ciently in the EM framework. In the EM algorithm, the function Q(?, ?) ? parameter set before updating and ? is a parameter set after updating, as described below. ? ? = Q(?, ?) p? (Y —X) log p??(X, Y) Y

In that equation, X represents a set of observed variables, and Y is a set of latent variables. The dif? ference of log likelihood between the models of ? and ?? is known to be greater than Q(?, ?)?Q(?, ?) [1]. For this reason, we can increase the likelihood monotonically by selecting a new parameter ?? to maximize the function Q. For HHMMs, the set of parameters is ? = {A, ?, B}. The set of observed 1:D 1:D variables is X = {O1:T , FT1 = 1}. The set of latent variables is Y = {Z1:T , F1:T ?1 }. Therefore, the function Q can be represented as shown below. ? 1:D 1:D 1 1:D 1:D ? ? Q(?, ?) p? (O1:T , FT1 = 1, Z1:T , F1:T (1) ?1) log p??(O1:T , FT = 1, Z1:T , F1:T ?1) 1:D ,F 1:D Z1:T 1:T ?1

The joint probability of observed variables and latent variables is given below. 1:D 1:D , F1:T p? (O1:T , FT1 = 1, Z1:T ?1)

=

D ?

?dZ1d

d=1

T? ?1 ? D

Fd

D T ? ? F d+1 (1?Ftd) Ftd ?dZ d) AdZ d End BZtD Ot T t t+1 t+1 t=1 d=1

(AdZt d End AdZt d Z d t

t=1 d=1

We substitute this equation for the joint probability in equation (1). We integrate out irrelevant variables and organize around each parameter. Thereby, we obtain the following. ? ? Q(?, ?)

D ? ?

g?di log ? ?di +

D ? ?

?

?iv gBiv log B

i??D v=1

d=1 i??d j?{sib(i)?End}

d=1 i??d

V ? ?

gAdij log A?dij +

Therein, g?di , gAdij , gBiv are shown by equation (2)(3)(4)(5). They are calculable using forward and backward activation probabilities. g?di

= ?bd1 (i)?bd1 (i) +

T ?1 ?

?bd?1 (parent(i))?di ?bdt+1 (i)

(2)

t+1

t=1

gAdiEnd

=

T ?1 ?

?edt (i)AdiEnd ?ed?1 (parent(i)) + ?ed (i)?ed (i) t

t=1

6

T

T

(3)

Table 2: Log-likelihood achieved at each iteration. Iteration FBA w/o MinSR FBA with MinSR FFB

1 -773.47 -773.89 -773.89

gAdij

=

2 -672.44 -672.47 -672.47

T ?1 ?

3 -668.50 -670.40 -670.40

4 -631.30 -643.62 -643.62

5 -610.63 -614.98 -614.98

10 -577.33 -573.84 -573.84

50 -457.66 -453.09 -453.09

?edt (i)Adij ?bdt+1 (j)

100 -447.90 -448.52 -448.52

(4)

t=1

gBiv

?

=

?eD (i)?eD (i) t t

(5)

t:Ot =v

? B, ? which maximize the function Q Using Lagrange multipliers, we can obtain ? , A, ? ? parameters ? ? ? under the constraint i0 ?sib(i) ? ?di0 = 1, j 0 ?{sib(i)?End} A?dij 0 = 1, v B iv = 1 as shown below. ? ?di = ?

g?di 0 i0 ?sib(i) g?di

gAdij

, A?dij = ?

0 j 0 ?{sib(i)?End} gAdij

?iv = ?gBiv ,B v gBiv

Consequently, we can calculate the update parameters using ? and ?. The time complexity for computing a single EM iteration is O(T N D+1), which is identical to the calculation of forward and backward activation probabilities.

5

Experiments

Firstly, we experimentally confirm that the forward-backward activation algorithm yields exactly identical parameter estimation to the flattening method does. Remind that we must make the MinSR constraint on the HHMM parameter set in the flattening method (see section 3). We compare three parameter estimation algorithms: our forward-backward activation algorithm for a MinSR HHMM (FBA with MinSR), for a HHMM without MinSR (FBA w/o MinSR), and the flattening method(FFB). The dataset to learn includes 5 sequences of 10 length, which are artificially generated by a MinSR HHMM of biased parameter set. We execute three algorithms and examine the log-likelihood achieved at each iteration. Table 2 presents the result. The FBA with MinSR and the FFB achieve the identical log-likelihood through the training. This result provides experimental evidence that our algorithm estimates HHMM parameters exactly identically to the flattening method does. Furthermore, the FBA enables us to conduct the parameter estimation of HHMMs which has non-zero self-transition parameters. To evaluate the computational costs empirically, we compare four methods of HHMM parameter estimation. Two are based on the EM algorithm with inference by the forward-backward activation algorithm (FBA), and by the flattening forward-backward method (FFB). Another two are based on a sampling approach: direct Gibbs sampling for the flat HMMs (DGS) and forward-backward activation sampling (FBAS). FBAS is a straightforward application of the forward-backward sampling scheme to the translated DBN presented in figure 2. In FBAS, we first calculate forward activation probabilities. Then we sample state activation variables from e1T to b11 in the backward order with respect to forward activation probabilities. We evaluate four methods based on three aspects: execution time, convergence speed, and scalability of the state space size. We apply each method to four different HHMMs of (D = 3,N = 3), (D = 3,N = 4), (D = 4,N = 3), and (D = 4,N = 4). We examine the log-likelihood of the training dataset achieved at each iteration to ascertain the learning convergence. As a training dataset, we use 100 documents from the Reuters corpus as word sequences. The dataset includes 36,262 words in all, with a 4,899 word vocabulary. Figure 3 presents the log-likelihood of the training data. The horizontal axis shows the logarithmically scaled execution time. Table 2 presents the average execution time for a single iteration. From these results, we can say primarily that FBA outperforms FFB in terms of execution time. The improvement is remarkable, especially for the HHMMs of large state space size because FBA has less time complexity for N and D than FFB has. 7

-180000

-190000

-190000

-200000

-200000

-210000

-210000

Log-Likelihood

Log-Likelihood

-180000

-220000 -230000 -240000

-230000 -240000

FB Activation Flattening FB FB Activation Sampling Direct Gibbs Sampling

-250000 -260000 100

1000

10000 Execution Time (ms)

100000

FB Activation Flattening FB FB Activation Sampling Direct Gibbs Sampling

-250000 -260000 100

1e+006

-180000

-180000

-190000

-190000

-200000

-200000

-210000

-210000

Log-Likelihood

Log-Likelihood

-220000

-220000 -230000 -240000

1000

10000 Execution Time (ms)

100000

1e+006

-220000 -230000 -240000

FB Activation Flattening FB FB Activation Sampling Direct Gibbs Sampling

-250000 -260000 100

1000

10000 Execution Time (ms)

100000

FB Activation Flattening FB FB Activation Sampling Direct Gibbs Sampling

-250000 -260000 1000

1e+006

10000

100000

1e+006

Execution Time (ms)

Figure 3: Convergence of log-likelihood for the training data on the Reuters corpus. Log-likelihood (vertical) is shown against the log-scaled execution time (horizontal) to display the execution time necessary to converge the learning of each algorithm. (top-left) HHMM of D = 3, N = 3. (topright) D = 3, N = 4. (bottom-left) D = 4, N = 3. (bottom-right) HHMM of D = 4, N = 4.

Table 3: Average execution time for a single iteration (ms).

Method	FBA	FFB	FBAS	DGS
D = 3, N = 3 (N D = 27)	186.65	1729.90	82.45	24.19
D = 3, N = 4 (N D = 64)	391.73	9242.35	142.20	37.50
D = 4, N = 3 (N D = 81)	476.92	19257.80	183.39	45.43
D = 4, N = 4 (N D = 256)	1652.03	220224.00	581.58	265.98

The results show that the likelihood convergence using DGS is much slower than that of other methods.The execution time of DGS is less than that of other methods for a single iteration, but this cannot compensate for the low convergence speed. However, FBAS achieves a competitive likelihood in comparison to FBA. Results show that FBAS might be appropriate for some situations because FBAS finds a better solution than that FBA do in some results.

6 Conclusion

In this work, we proposed a new inference algorithm for HHMMs based on the activation probability. Results show that the performance of our proposed algorithm surpasses that of existing methods. The forward-backward activation algorithm described herein enables us to conduct unsupervised parameter learning with a practical computational cost for HHMMs of larger state space size.

2 References

[1] C. Bishop. Pattern Recognition and Machine Learning. Springer, 2007. [2] S. Chib. Calculating posterior distributions and modal estimates in markov mixture models. Journal of Econometrics, 1996. [3] S. Fine, Y. Singer, and N. Tishby. The hierarchical hidden markov model: Analysis and applications. Machine Learning, 1998. 8

[4] T. Griffiths and M. Steyvers. Finding scientific topics. Proc. the National Academy of Sciences of the United States of America, 2004. [5] K. Heller, Y. Teh, and D. Gorur. Infinite hierarchical hidden markov models. In Proc. International Conference on Artificial Intelligence and Statistics, 2009. [6] S. Luhr, H. Bui, S. Venkatesh, and G. West. Recognition of human activity through hierarchical stochastic learning. In Proc. Pervasive Computing and Communication, 2003. [7] K. Murphy and M. Paskin. Linear time inference in hierarchical hmms. In Proc. Neural Information Processing Systems, 2001. [8] N. Nguyen, D. Phung, and S. Venkatesh. Learning and detecting activities from movement trajectories using the hierarchical hidden markov models. In Proc. Computer Vision and Pattern Recognition, 2005. [9] L. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. Proc. IEEE, 1989. [10] S. Scott. Bayesian methods for hidden markov models: Recursive computing in the 21st century. Journal of the American Statistical Association, 2002. [11] M. Skounakis, M. Craven, and S. Ray. Hierarchical hidden markov models for information extraction. In Proc. International Joint Conference on Artificial Intelligence, 2003. [12] M. Weiland, A. Smaill, and P. Nelson. Learning musical pitch structures with hierarchical hidden markov models. In Proc. Journees Informatiques Musicales, 2005. [13] L. Xie, S. Chang, A. Divakaran, and H. Sun. Learning hierarchical hidden markov models for video structure discovery. Technical report, Columbia University, 2002.

9