Pattern Recognition 46 (2013) 3452–3459

Contents lists available at SciVerse ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

# Marginalized Viterbi algorithm for hierarchical hidden Markov models

Akira Hayashi*, Kazunori Iwata, Nobuo Suematsu

*Graduate School of Information Sciences, Hiroshima City University, 3–4–1 Ozuka-Higashi, Asa-Minami-Ku, Hiroshima 731–3194, Japan*

ARTICLE INFO

*Article history:*
Received 28 September 2012
Received in revised form
29 May 2013
Accepted 3 June 2013
Available online 14 June 2013

*Keywords:*
Time series data
Hierarchical HMM
Finding the most likely state sequence
Generalized Viterbi algorithm
Marginalized Viterbi algorithm

ABSTRACT

The generalized Viterbi algorithm, a direct extension of the Viterbi algorithm for hidden Markov models (HMMs), has been used to find the most likely state sequence for hierarchical HMMs. However, the generalized Viterbi algorithm finds the most likely whole level state sequence rather than the most likely upper level state sequence. In this paper, we propose a marginalized Viterbi algorithm, which finds the most likely upper level state sequence by marginalizing lower level state sequences. We show experimentally that the marginalized Viterbi algorithm is more accurate than the generalized Viterbi algorithm in terms of upper level state sequence estimation.

## 1. Introduction

Hidden Markov models (HMMs) [1], known for their success in voice recognition, have been widely used to analyze time series data. Fine et al. [2] proposed hierarchical hidden Markov models (HHMMs) as a generalization of HMMs with a hierarchical state space. An HHMM may be represented using a tree structure, where each state at a non-leaf node, called an internal state, is itself a dynamical probabilistic model. Therefore, the internal states of an HHMM emit sequences rather than a single symbol. An HHMM generates sequences by recursive activation of a substate of an internal state, until a leaf node state, called a production state, is reached. Production states are the only states that actually output symbols through the usual HMM mechanism. The original inference algorithm for HHMMs is not efficient, taking $O(T^3)$ time where $T$ is the length of the observation sequence. Murphy et al. [3] devised a dynamic Bayesian network (DBN) representation for HHMMs, thanks to which a linear time ($O(T)$) inference algorithm is now available.

HHMMs can naturally represent the multiple time scale structure of many time series data (for example, voice has three time scale structures: word sequence, phone sequence, and sub-phone sequence), and are gaining much attention in the research community. Some of the applications of HHMMs are hand written character recognition [2], information extraction from texts [4], and video analysis [5,6].

The problem of finding the most likely state sequence from an observation sequence [1] is important and has many applications. To find the most likely state sequence for HHMMs, the generalized Viterbi algorithm (GVA) [2,3], a direct extension of the Viterbi algorithm for HMMs [7,1], has been used. However, GVA finds the most likely whole level state sequence, but not the most likely upper level state sequence.

In this paper, we propose a marginalized Viterbi algorithm (MVA) to overcome the problem associated with GVA. MVA finds the most likely upper level state sequence by marginalizing lower level state sequences. For example, MVA will find the most likely sequence of “word” states in speech recognition by marginalizing the irrelevant “phone” and “sub-phone” state sequences, thus avoiding the problems associated with words having several pronunciations [8].[1]

To explain our motivation for marginalizing irrelevant lower level states, consider the simple two level *static* hierarchical model in Fig. 1. The model can be seen as a Gaussian mixture speaker model for speaker identification [9], where the top level state, $q^1$, stands for a speaker $s$, and the second level state, $q^2$, stands for a component $c$ of the Gaussian mixture model:

$$\begin{cases} p(q^2=c|q^1=s)=\pi_c^s \\ p(o=\mathbf{x}|q^1=s,q^2=c)=\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_c^s,\boldsymbol{\Sigma}_c^s), \end{cases} \tag{1}$$

where $\pi_c^s \geq 0$ is the weight of $c$ and satisfies $\sum_c \pi_c^s = 1.0$, and $\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_c^s,\boldsymbol{\Sigma}_c^s)$ is a Gaussian density with mean vector $\boldsymbol{\mu}_c^s$ and covariance matrix $\boldsymbol{\Sigma}_c^s$. Given an observation $o=\mathbf{x}$, the most likely

---

* Corresponding author. Tel.: +81 82 830 1567.
*E-mail addresses:* akira@hiroshima-cu.ac.jp (A. Hayashi), kiwata@hiroshima-cu.ac.jp (K. Iwata), suematsu@hiroshima-cu.ac.jp (N. Suematsu).

[1] MVA cannot, however, find the most likely word sequence since it does not marginalize over word segmentation boundaries.

0031-3203/$ - see front matter © 2013 Published by Elsevier Ltd.
http://dx.doi.org/10.1016/j.patcog.2013.06.001

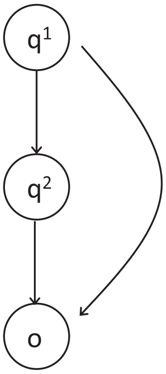

**Fig. 1.** Two level *static* hierarchical model.

estimation for the speaker identification is

$$\hat{s} = \operatorname*{argmax}_{s} p(q^1 = s | o = \mathbf{x}), \tag{2}$$

where $p(q^1 = s | o = \mathbf{x})$ is obtained by

$$p(q^1 = s | o = \mathbf{x}) = \sum_{c} p(q^1 = s, q^2 = c | o = \mathbf{x}), \tag{3}$$

that is, by *marginalizing* $q^2$, an irrelevant second level state.

In this paper, we also propose a fast approximation algorithm for MVA. We show using experiments that MVA is more accurate than GVA in terms of upper level state sequence estimation.

MVA was developed in our lab and first introduced in [10] as a conference proceedings paper with a limited audience. The main theme of the previous paper is not MVA, but HHCRFs, discriminative models corresponding to HHMMs. Most of the results presented in the current paper are new, including the fast approximation algorithm for MVA and the detailed comparison of GVA and MVA through experiments.

Our paper is organized as follows. We explain HHMMs in Section 2. We then explain GVA and MVA in Section 3. In Section 4, we compare the performances of GVA and MVA through experiments. We summarize the paper in Section 5.

## 2. HHMMs

An HHMM may be represented using a tree structure, and generates sequences by recursive activation of a substate of a non-leaf node until a leaf node state, called a production state, is reached. Production states are the only states that actually emit output symbols. The original inference algorithm for HHMMs is not efficient, taking $O(T^3)$ time. Murphy et al. [3] devised a dynamic Bayesian network (DBN) representation for HHMMs, thanks to which a linear time ($O(T)$) inference algorithm is now available.

### 2.1. Overview of HHMMs

An HHMM is represented as a tree structure as shown in Fig. 2. The circles, trapezoids, and rectangles in the figure stand for internal states, production states, and end states, respectively. The arrows connecting the states represent state transitions. A solid line indicates a horizontal transition to a state within the same level, a broken line indicates a vertical transition to a child state in the next level, and a dotted line indicates a forced transition from an end state, after which control is returned to the calling parent state. The state at the top of the hierarchy is called the root node. The level for the root node is 0, and a sequence of state transitions starts at the root state.

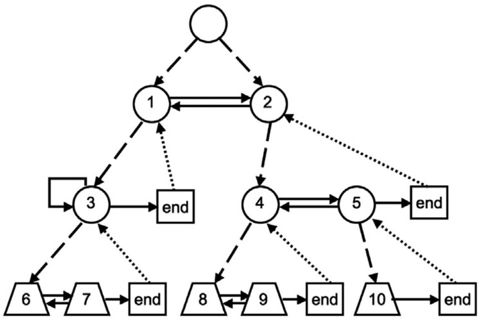

**Fig. 2.** Example of an HHMM with a three-level hierarchy.

An HHMM generates a sequence of observations as follows.

(Step1) Start: we start from the root node at time $t=1$.
(Step2) Vertical transition: a transition occurs from the current state (an internal state) to a child state in the lower level. If the destination is an internal state, further transitions to lower level states occur until a production state is reached.
(Step3) Output symbol emission: the production state emits an output symbol $o_t$. Time $t$ is incremented by 1.
(Step4) Horizontal transition: a transition to a state within the same level occurs. If the destination is an internal state, we go back to Step 2, and if the destination is a production state, we go back to Step 3. If the destination is an end state, we proceed to Step 5.
(Step5) Forced transition: A forced transition occurs to the upper level parent state which has initiated the current level state transitions, and we go back to Step 4.

Fine et al. [2], as well as proposing HHMMs, developed an algorithm for state estimation on the basis of the inside-outside algorithm. This algorithm is not efficient, however, and the time for state estimation and also for the most likely state sequence estimation is $O(T^3)$, where $T$ is the length of the observation sequence.

### 2.2. Representing HHMMs as DBNs

Murphy and Paskin [3] devised a dynamic Bayesian network (DBN) representation for HHMMs. A Bayesian network (BN) is a directed acyclic graph representing conditional independence relationships between random variables, and a DBN is an extension of a BN to a random process, where the random variables are dependent on time $t$. Thanks to the DBN representation, linear time ($O(T)$) algorithms for state estimation and the most likely state sequence estimation have become available.

We show a DBN representation of a three-level HHMM in Fig. 3. (We assume for simplicity that all production states are in the bottom level of the hierarchy.) The random variable $o_t$ in the figure stands for the output from a production state at time $t$ ($t = 1, \dots, T$). The output of an HHMM can be either discrete or continuous, but we consider the case of discrete symbol output in this paper. The state of the HHMM in level $d$ and at time $t$ is denoted by $q_t^d$ ($d \in \{1, \dots, D\}$), where $d$ is the hierarchy index: the top level has $d=1$, and the bottom level has $d=D$.

$f_t^d$ is an indicator variable which is equal to 1 if $q_t^d$ has transitioned to its end state, and is 0 otherwise. The indicator variables play an important role in representing an HHMM as a DBN. As we explained in the previous subsection, a transition to an end state leads to a state transition in the upper level. In other words, $f_t^d = 1$ implies a possible state change in level $d-1$. In addition, if $f_t^d = 1$ then $f_t^{d'} = 1$ for all $d' > d$; hence the number of indicator variables that equal 0 denotes the level of the hierarchy we are currently in.