Second-Order Belief Hidden Markov Models

Jungyeul Park, Mouna Chebbah, Siwar Jendoubi, Arnaud Martin

Second-order Belief Hidden Markov Models

Jungyeul Park[1], Mouna Chebbah[1,2] and Siwar Jendoubi[1,2], Arnaud Martin[1]

[1] UMR 6074 IRISA, Université de Rennes1, Lannion, France
[2] LARODEC Laboratory, University of Tunis, ISG Tunis, Tunisia
{jungyeul.park,mouna.chebbah,arnaud.martin}@univ-rennes1.fr
jendoubi.siwar@etudiant.univ-rennes1.fr

Abstract. Hidden Markov Models (HMMs) are learning methods for pattern recognition. The probabilistic HMMs have been one of the most used techniques based on the Bayesian model. First-order probabilistic HMMs were adapted to the theory of belief functions such that Bayesian probabilities were replaced with mass functions. In this paper, we present a second-order Hidden Markov Model using belief functions. Previous works in belief HMMs have been focused on the first-order HMMs. We extend them to the second-order model.

Keywords: Belief functions, Dempster-Shafer theory, first-order belief HMM, second-order belief HMM, probabilistic HMM

1 Introduction

A Hidden Markov Model (HMM) is one of the most important statistical models in machine learning [1]. A HMM is a classifier or labeler that can assign label or class to each unit in a sequence [2]. It has been successfully utilized over several decades in many applications for processing text and speech such as Part-of-Speech (POS) tagging [3], named entity recognition [4] and speech recognition [5]. However, such works in the early part of the period are mainly based on first-order HMMs. As a matter of fact, the assumption in the first-order HMM, where the state transition and output observation depend only on one previous state, does not exactly match with the real applications [6]. Therefore, they require a number of sophistications. For example, even though the first-order HMM for POS tagging in early 1990s performs reasonably well, it captures a more limited amount of the contextual information than is available [7]. As consequence, most modern statistical POS taggers use a second-order model [8].

Uncertainty theories can be integrated in statistical models such as HMMs: The probability theory has been used to classify units in a sequence with the Bayesian model. Then, the theory of belief functions is employed to this statistical model because the fusion proposed in this theory simplifies computations of *a posteriori* distributions of hidden data in Markov models. This theory can provide rules to combine evidences from different sources to reach a certain level of belief [9–13]. Belief HMMs introduced in [14–22], use combination rules proposed in the framework of the theory of belief functions. This paper is an extension of previous ideas for second-order belief HMMs. For the current work, we focus on explaining a second-order model. However, the proposed method can be easily extended to higher-order models.

This paper is organized as follows: In Sections 2 and 3, we detail probabilistic HMMs for the problem of POS tagging where HMMs have been widely used. Then, we describe the first-order belief HMM in Section 4. Finally, before concluding, we propose the second-order belief HMM.

2   First-order probabilistic HMMs

POS tagging is a task of finding the most probable estimated sequence of n tags given the observation sequence of v words. According to [1], a first-order probabilistic HMM can be characterized as follows:

N	The number of states in a model $S_t = \{s_1^t, s_2^t, \cdots s_N^t\}$ at the time t.
M	The number of distinct observation symbols. $V = \{v_1, v_2, \cdots, v_M\}$.
$A = \{a_{ij}\}$	The set of N transition probability distributions.
$B = \{b_j(o_t)\}$	The observation probability distributions in state j.
$\pi = \{\pi_i\}$	The initial probability distribution.

Figure 1 illustrates the first-order probabilistic HMM allowing to estimate the probability of the sequence s_i^{t-1} and s_j^t where a_{ij} is the transition probability from s_i^{t-1} to s_j^t and $b_j(o_t)$ is the observation probability on the state s_j^t. Regarding POS tagging, the number of possible POS tags that are hidden states S_t of the HMM is N. The number of words in the lexicons V is M. The transition probability a_{ij} is the probability that the model moves from one tag s_i^{t-1} to another tag s_j^t. This probability can be estimated using a training data set in supervised learning for the HMM. The probability of a current POS tag appearing in the first-order HMM depends only on the previous tag. In general, first-order probabilistic HMMs should be characterized by three fundamental problems as follows [1]:

- Likelihood: Given a set of transition probability distributions A, an observation sequence $O = o_1, o_2, \cdots, o_T$ and its observation probability distribution B, how do we determine the likelihood $P(O|A,B)$? The first-order model relies on only one observation where $b_j(o_t) = P(o_j|s_j^t)$ and the transition probability based on one previous tag where $a_{ij} = P(s_j^t|s_i^{t-1})$. Using the forward path probability, the likelihood $\alpha_t(j)$ of a given state s_j^t can be computed by using the likelihood $\alpha_{t-1}(i)$ of the previous state s_i^{t-1} as described below:

$$\alpha_t(j) = \sum_i \alpha_{t-1}(i) a_{ij} b_j(o_t) \tag{1}$$

- Decoding: Given a set of transition probability distributions A, an observation sequence $O = o_1, o_2, \cdots, o_T$ and its observation probability distribution B, how do we discover the best hidden state sequence? The Viterbi algorithm is widely used for calculating the most likely tag sequence for the decoding problem. The Viterbi algorithm can calculate the most probable path $\delta_t(j)$ which contains the sequence of $\psi_t(j)$. It can select the path that maximizes the likelihood of the sequence as described below:

$$\delta_t(j) = \max \delta_{t-1}(i) a_{ij} b_j(o_t)$$
$$\psi_t(j) = \text{argmax } \psi_{t-1}(i) a_{ij}$$
(2)

– Learning: Given an observation sequence $O = o_1, o_2, \cdots, o_T$ and a set of states $S = \{s_1^t, s_2^t, \cdots, s_N^t\}$, how do we learn the HMM parameters for A and B? The parameter learning task usually uses the Baum-Welch algorithm which is a special case of the Expectation-Maximization (EM) algorithm.

In this paper, we focus on the likelihood and decoding problems by assuming a supervised learning paradigm where labeled training data are already available.

3 Second-order probabilistic HMMs

Now, we explain the extension of the first-order model to a *trigram*[3] in the second-order model. Figure 2 illustrates the second-order probabilistic HMM allowing to estimate the probability of the sequence of three states s_i^{t-2}, s_j^{t-1} and s_k^t where a_{ijk} is the transition probability from s_i^{t-2} and s_j^{t-1} to s_k^t, and $b_k(o_t)$ is the observation probability on the state s_k^t. Therefore, the second-order probabilistic HMM is characterized by three fundamental problems as follows:

– Likelihood: The second-order model relies on one observation $b_k(o_t)$. Unlike the first-order model, the transition probability is based on two previous tags where a_{ijk} = $P(s_k^t | s_i^{t-2}, s_j^{t-1})$ as described below:

$$\alpha_t(k) = \sum_j \alpha_{t-1}(j) a_{ijk} b_k(o_t)$$
(3)

However, it will be more difficult to find a sequence of three tags than a sequence of two tags. Any particular sequence of tags s_i^{t-2}, s_j^{t-1}, s_k^t that occurs in the test set may simply never have occurred in the training set because of data sparsity [2]. Therefore, a method for estimating $P(s_k^t | s_i^{t-2}, s_j^{t-1})$, even if the sequence s_i^{t-2}, s_j^{t-1}, s_k^t never occurs, is required. The simplest method to solve this problem is to combine the trigram $\hat{P}(s_k^t | s_i^{t-2}, s_j^{t-1})$, the bigram $\hat{P}(s_k^t | s_j^{t-1})$, and even the unigram $\hat{P}(s_k^t)$ probabilities [8]:

$$P(s_k^t | s_i^{t-2}, s_j^{t-1}) = \lambda_1 \hat{P}(s_k^t | s_i^{t-2}, s_j^{t-1}) + \lambda_2 \hat{P}(s_k^t | s_j^{t-1}) + \lambda_3 \hat{P}(s_k^t)$$
(4)

Note that $\hat{P}$ is the maximum likelihood probabilities which are derived from the relative frequencies of the sequence of tags. Values of λ are such that $\lambda_1 + \lambda_2 + \lambda_3 = 1$ and they can be estimated by the *deleted interpolation* algorithm [8]. Otherwise, [7] describes a different method for values of λ as below:

$$\lambda_1 = k_3$$
$$\lambda_2 = (1 - k_3) \cdot k_2$$
$$\lambda_3 = (1 - k_3) \cdot (1 - k_2)$$
(5)

---

[3] The trigram is the sequence of three elements, *i.e.* three states in our case.

where $k_2 = \frac{log(C(s_j^{t-1}, s_k^t)+1)+1}{log(C(s_j^{t-1}, s_k^t)+1)+2}$, $k_3 = \frac{log(C(s_i^{t-2}, s_j^{t-1}, s_k^t)+1)+1}{log(C(s_i^{t-2}, s_j^{t-1}, s_k^t)+1)+2}$, and $C(s_i^{t-2}, s_j^{t-1}, s_k^t)$ is the frequency of a sequence $s_i^{t-2}, s_j^{t-1}, s_k^t$ in the training data. Note that $\lambda_1 + \lambda_2 + \lambda_3$ is not always equal to one in [7]. The likelihood of the observation probability for the second-order model uses B where $b_k(o_t) = P(o_k|s_k^t, s_j^{t-1})$.

– Decoding: For second-order model we require a different Viterbi algorithm. For a given state s at the time t, it would be redefined as follows [7]:

$$
\begin{aligned}
&\delta_t(k) = \max \delta_{t-1}(j) a_{ijk} b_k(o_t) \\
&\quad \text{where } \delta_t(j) = \max P(s^1, s^2, \cdots, s^{t-1} = s_i, s^t = s_j, o_1, o_2, \cdots, o_t) \\
&\psi_t(k) = \text{argmax } \psi_{t-1}(j) a_{ijk} \\
&\quad \text{where } \psi_t(k) = \text{argmax } P(s^1, s^2, \cdots, s^{t-1} = s_i, s^t = s_j, o_1, o_2, \cdots, o_t)
\end{aligned}
\tag{6}
$$

– Learning: The problem of learning would be similar to the first-order model except that parameters A and B are different.

With respect to performance measures, different transition probability distributions in [8] and [7] obtain 97.0% and 97.09% tagging accuracy for known words, respectively for the same data (the Penn Treebank corpus). Even though probabilistic HMMs perform reasonably well, belief HMMs can learn better under certain conditions on observations [21].

4   First-order Belief HMMs

In probabilistic HMMs, A and B are probabilities estimated from the training data. However, A and B in belief HMMs are mass functions (bbas) [17, 21]. According to previous works on belief HMMs, a first-order HMM using belief functions can be characterized as follows[4]:

N	The number of states in a model $\Omega_t = \{S_1^t, S_2^t, \cdots, S_N^t\}$.
M	The number of distinct observation symbols V.
$A = \{m_a^{\Omega_t}[S_i^{t-1}](S_j^t)\}$	The set of conditional bbas to all possible subsets of states.
$B = \{m_b^{\Omega_t}[o_t](S_j^t)\}$	The set of bbas according to all possible observations O_t.
$\pi = \{m_\pi^{\Omega_1}(S_i^{\Omega_1})\}$	The bba defined for the the initial state.

Difference between the first-order probabilistic and belief HMMs is presented in Figure 1, the transition and observation probabilities in belief HMMs are described as mass functions. Therefore, we can replace a_{ij} by $m_a^{\Omega_t}[S_i^{t-1}](S_j^t)$ and $b_j(o_t)$ by $m_b^{\Omega_t}[o_t](S_j^t)$. The set Ω_t has been used to denote states for HMMs using belief functions [17, 21]. Note that s_i^t is the single state for probabilistic HMMs and S_i^t is the multi-valued state for belief HMMs. First-order belief HMMs should also be characterized by three fundamental problems as follows:

– Likelihood: The likelihood problem in belief HMMs is not solved by *likelihood*, but by using the combination. The first-order belief model relies on (*i*) only one observation $m_b^{\Omega_t}[o_t](S_j^t)$ and (*ii*) a transition conditional mass function based on

---

[4] In the model Ω_t, S^t are focal elements

one previous tag $m_a^{\Omega_t}[S_i^{t-1}](S_j^t)$. Mass functions of sets A and B are combined using the Disjunctive Rule of Combination (DRC) for the forward propagation and the Generalized Bayesian Theorem (GBT) for the backward propagation [23]. Using the forward path propagation, the mass function of a given state S_j^t can be computed as the combination of mass functions on the observation and the transition as described below:

$$q_\alpha^{\Omega_t}(S_j^t) = \sum m_\alpha^{\Omega_{t-1}}(S_i^{t-1}) \cdot q_a^{\Omega_t}[S_i^{t-1}](S_j^t) \cdot q_b^{\Omega_t}(S_j^t) \tag{7}$$

Note that the mass function of the given state S_j^t is derived from the commonality function $q_\alpha^{\Omega_t}$.

– Decoding: Several solutions have been proposed to extend the Viterbi algorithm to the theory of belief functions [17, 24, 25]. Such solutions maximize the plausibility of the state sequence. In fact, the *credal* Viterbi algorithm starts from the first observation and estimates the commonality distribution of each observation until reaching the last state. For each state S_j^t, the estimated commonality distribution $(q_\delta^{\Omega_t}(S_j^t))$ is converted back to a mass function that is conditioned on the previous state. Then, we apply the *pignistic* transform to make a decision about the current state $(\psi_t(s_j^t))$:

$$\begin{aligned} q_\delta^{\Omega_t}(S_j^t) &= \sum_{S_i^{t-1} \subseteq A^{t-1}} m_\delta^{\Omega_{t-1}}(S_i^{t-1}) \cdot q_a^{\Omega_t}[S_i^{t-1}](S_j^t) \cdot q_b^{\Omega_t}(S_j^t) \\ \psi_t(s_j^t) &= \text{argmax}_{S_i^{t-1} \in \Omega_{t-1}} (1 - m_\delta^{\Omega_t}[S_i^{t-1}](\emptyset)) \cdot P_t[S_i^{t-1}](S_j^t) \end{aligned} \tag{8}$$

where $A^t = \cup_{S_j^t \in \Omega_t} \psi_t(S_j^t)$ [17].

– Learning: Instead of the traditional EM algorithm, we can use the E^2M algorithm for the belief HMM [22].

To build belief functions from what we learned using probabilities in the previous section, we can employ the least commitment principle by using the inverse pignistic transform [26, 27].

5 Second-order Belief HMMs

Like the first-order belief HMM, N, M, B and π are similarly defined in the second-order HMM. The set A is quite different and is defined as follows:

$$A = \{m_a^{\Omega_t}[S_i^{t-2}, S_j^{t-1}](S_k^t)\} \tag{9}$$

where A is the set of conditional bbas to all possible subsets of states based on the two previous states. Second-order belief HMMs should also be characterized by three fundamental problems as follows:

– Likelihood: The second-order belief model relies on one observation $m_b^{\Omega_t}[o_t](S_k^t)$ in a state S_k at time t and the transition conditional mass function based on two previous states S_i^{t-2} and S_j^{t-1}, defined by $m_a^{\Omega_t}[S_i^{t-2}, S_j^{t-1}](S_k^t)$. Using the forward path

propagation, the mass function of a given state S_k^t can be computed as the disjunctive combination (DRC) of mass functions on the transition $m_a^{\Omega_t}[S_i^{t-2}, S_j^{t-1}](S_k^t)$ and the observation $m_b^{\Omega_t}(S_k^t)$ as described below:

$$q_\alpha^{\Omega_t}(S_k^t) = \sum m_\alpha^{\Omega_{t-1}}(S_j^{t-1}) \cdot q_a^{\Omega_t}[S_i^{t-2}, S_j^{t-1}](S_k^t) \cdot q_b^{\Omega_t}(S_k^t) \tag{10}$$

where $q_a^{\Omega_t}[S_i^{t-2}, S_j^{t-1}](S_k^t)$ is the commonality function derived from the conjunctive combination of mass functions of two previous transitions. The conjunctive combination is used to have the conjunction of observations on previous two states S_i^{t-2} and S_j^{t-1}.

The combined mass function $m_a^{\Omega_t}[S_i^{t-2}, S_j^{t-1}](S_k^t)$ of two transitions $m_a^{\Omega_{t-1}}[S_i^{t-2}](S_j^{t-1})$ and $m_a^{\Omega_t}[S_j^{t-1}](S_k^t)$ is defined as follows:

$$m_a^{\Omega_t}[S_i^{t-2}, S_j^{t-1}](S_k^t) = m_a^{\Omega_{t-1}}[S_i^{t-2}](S_j^{t-1}) \, \textcircled{\tiny $\bigcirc$} \, m_a^{\Omega_t}[S_j^{t-1}](S_k^t) \tag{11}$$

The conjunctive combination is required to obtain the conjunction of both transitions. Note that the mass function of the given state S_k^t is derived from the commonality function $q_\alpha^{\Omega_t}$. We use DRC with commonality functions like in [17]. Note that the observation only on one previous state is taken into account in the first-order belief HMM, but the conjunction of observations on two previous states is considered in the second-order belief HMM.

- Decoding: We accept our assumption of the first-order belief HMM for the second-order model. Similarly to the first-order belief HMM, we propose a solution that maximizes the plausibility of the state sequence. The credal Viterbi algorithm estimates the commonality distribution of each observation from the first observation till the final state. For each state S_k^t, the estimated commonality distribution $(q_\delta^{\Omega_t}(S_k^t))$ is converted back to a mass function that is conditioned on a mass function of the two previous states. This mass function is the conjunctive combination of mass functions of the two previous states. Then, we apply the pignistic transform to make a decision about the current state $(\psi_t(s_j^t))$ as before:

$$\begin{aligned} q_\delta^{\Omega_t}(S_k^t) &= \sum_{S_j^{t-1} \subseteq A^{t-1}} m_\delta^{\Omega_{t-1}}(S_j^{t-1}) \cdot q_a^{\Omega_t}[S_i^{t-2}, S_j^{t-1}](S_k^t) \cdot q_b^{\Omega_t}(S_k^t) \\ \psi_t(s_k^t) &= \mathrm{argmax}_{S_j^{t-1} \in \Omega_{t-1}} (1 - m_\delta^{\Omega_t}[S_j^{t-1}](\emptyset)) \cdot P_t[S_i^{t-2}, S_j^{t-1}](S_k^t) \end{aligned} \tag{12}$$

- Learning: Like the first-order belief model, we can still use the E^2M algorithm for the belief HMM [22].

Since the combination of mass functions in the belief HMM is required where the previous observation is already considered in the set of conditional bbas $m_a^{\Omega_t}[S_i^{t-2}, S_j^{t-1}]$, we do not need to refine the observation probability for the second-order model as in the second-order probabilistic model.

6 Conclusion and future perspectives

The problem of POS tagging has been considered as one of the most important tasks for natural language processing systems. We described such a problem based on HMMs

Fig. 1. First-order probabilistic and belief HMMs
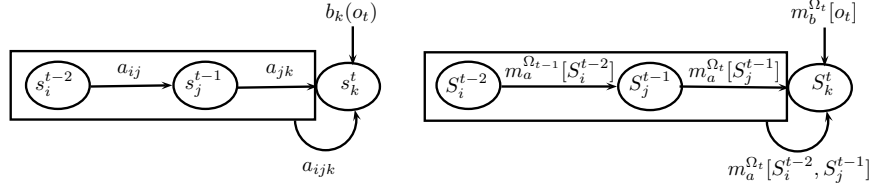
Fig. 2. Second-order probabilistic and belief HMMs

and tried to apply our idea to the theory of belief functions. We extended previous works on belief HMMs to the second-order model. Using the proposed method, we will be able to easily extend the higher-order model for belief HMMs. Some technical aspects still remain to be considered. Robust implementation for belief HMMs are required where in general we can find over one million observations in the training data to deal with the problem of POS tagging. As described before, the choice of inverse pignistic transforms would be empirically verified.[5] We are planning to implement these technical aspects in near future.

The current work is described to rely on a supervised learning paradigm from labeled training data. Actually, the forward-backward algorithm in HMMs can do completely unsupervised learning. However, it is well known that EM performs poorly in unsupervised induction of linguistic structure because it tends to assign relatively equal numbers of tokens to each hidden state [29].[6] Therefore, the initial conditions can be very important. Since the theory of belief functions can take into consideration of uncertainty and imprecision, especially for the lack of data, we might obtain a better model using belief functions on an unsupervised learning paradigm.

References

---

[5] For example, [28] used the inverse pignistic transform in [26] to calculate belief functions from Bayesian probability functions. As matter of fact, the problem of POS tagging can be normalized and inverse pignistic transforms in [26] did not propose the case for $m(\emptyset)$.

[6] The actual distribution of POS tags would be highly skewed as in heavy-tail distributions.