Hierarchical Algorithm for Hidden Markov Model

SANAA CHAFIK
Laboratory of modelisation and calcul.
University Sultan Moulay Slimane
Béni Mellal, Moroco

DAOUI CHERKI
Laboratory of modelisation and calcul.
University Sultan Moulay Slimane
Béni Mellal, Moroco

***Abstract*—The Forward algorithm is an inference algorithm for hidden Markov models, which often leads to a very large hidden state space. The objective of this work is to reduce the task of solving the Forward algorithm, by offering faster improved algorithm which is based on divide and conquer technique.**

***Keywords* —Hidden Markov Model; Forward; Divide and Conquer; Decomposition; Communicating Class.**

I. INTRODUCTION

A Hidden Markov Model (HMM) is a doubly stochastic process, one of whose components is an unobservable Markov chain; it is used extensively in pattern recognition, speech recognition [1, 2], Handwriting recognition [3, 4, 5], computational biology [6], Machine translation [7]. During the use of HMMs we are led to treat three fundamental problems: Evaluation, decoding and learning [8].

The HMMs fall most often on a large dimension state space that makes interesting use of the Divide and Conquer technique. The principle is based on dividing a large problem into several similar problems which avoids the curse of dimensionality. In this direction, we will propose a decomposition method and improved algorithm to solve large HMMs.

This paper is organized as follows: We briefly present in the second section a general introduction to Hidden Markov Models and their fundamental problems. In the third section the Forward algorithm is described. The problematic and the solution are given in the next section. Finally, we propose an improved version and the complexity of the Forward algorithm in the fourth section.

II. HIDDEN MARKOV MODEL

A. Definition

The HMM is defined by a tuple [9, 10] {N, M, A, B, Π }:

The Model is formed by N states $S=\{S_1,S_2,...,S_N\}$.

The M observations $O=\{O_1,O_2,...,O_M\}$.

The matrix of transition probabilities is denoted by A=$\left[a_{ij}\right]$,where:

$$a_{ij}=P(s_{t+1}=S_j \mid s_t=S_i) \text{ where } \sum_{j\in S} a_{i,j}=1 \text{ , } \forall i\in S \quad (1)$$

a_{ij} Specifies the probability of transitioning from state i to state j.

The observation probability matrix or emission probability, denoted by $B=\left[b_j(m)\right]$:

$$b_j(m)=P(o_t=O_m \mid s_t=S_j) \quad (2)$$

$b_j(m)$ Represents the probability of emitting symbol O_m at the instant t by the state S_j.

The probability distribution of the initial state is denoted by $\Pi=\left[\pi_i\right]$:

$$\pi_i=P(s_1=S_i) \text{ where } \sum_{i\in S}\pi(i)=1 \quad (3)$$

π_i specifies the probability of being in state i at time zero.

B. Fundamental problems of HMM

There are three basic HMM problems that must be solved:

Evaluation: Given an observation sequence $O=O_1,O_2,...,O_T$ and a model $\lambda=\{\Pi,A,B\}$, what is the probability of the model generating that observation sequence?

Decoding: Given the observation $O=O_1,O_2,...,O_T$ and an HMM model $\lambda=\{\Pi,A,B\}$, how do we find the state sequences that best explain the observation?

Learning: How do we adjust the model parameters $\lambda=\{\Pi,A,B\}$, to maximize $P(O\mid\lambda)$?

III. FORWARD ALGORITHM

For each pair (state, time) we associate the Forward variable $\alpha_t(i)$ given in equation (5) which represents the probability of the partial observation sequence $\{O_1,...,O_t\}$ (until time t) and state S_i at time t, given the model λ .

$$\alpha_t(i)=P(o_1=O_1,...,o_t=O_t,s_t=S_i \mid \lambda) \quad (5)$$

Algorithm 3.1.

Step1: Initialization, let $i\in S$

$$\alpha_1(i)=\pi_i\times b_i(o_1) \quad (6)$$

Step2: Induction, let $t\in\left[1,T-1\right], j\in\left[1,N\right]$

$$\alpha_{t+1}(j)=\left[\sum_{i\in S}\alpha_t(i)\times a_{ij}\right]b_j(o_{t+1})$$

Step3: Termination

(7)

$$P(O \mid \lambda)=\sum_{i\in S}\alpha_T(i) \qquad (8)$$

IV. PROBLEMATIC AND SOLUTION

A. The curse of dimensionality

The statistical learning algorithms such as those dedicated to hidden Markov chains they are suffering from the exponentially increase of the cost when the volume of data grows, which is known as the curse of dimensionality [11].

B. Divide and Conquer

The term Divide and Conquer algorithmic technique [12, 13] yields elegant, simple and very efficient algorithms, their principle is based on dividing a large problem into several similar problems which avoids the curse of dimensionality.

C. Principe of decomposition

Decomposition technique [14] consists of the following steps. First, the algorithm of decomposition to levels is applied, thereafter the restricted HMMs are constructed, eventually, we combine all the partial solutions in order to construct the global solution of the HMM.

In this section, we consider HMM, Let G=(S, U) be the graph associated with the HMM, that is, the state space S represents the set of nodes and $U=\{(i,j)\in S^2 : a_{ij}>0\}$ the set of directed arcs.

1) Decomposition into levels

The state space can be partitioned into strongly connected classes $C_1, C_2, ..., C_H$. Note that the strongly connected classes are defined to be the classes with respect to the relation on G defined by: i is strongly connected to j if and only if i=j or there exist a directed path from i to j and directed path from j to i. There are many good algorithms in graph theory for the computations of such partition, see [15]. Now we construct by induction the levels of the graph G. The level L_0 is formed by all classes C_i such that C_i is closed, that is, any arc emanating from C_i has both nodes in C_i. The path level L_p is formed by all classes C_i such that the end of any arc emanating from C_i is in some level $L_{p-1}, L_{p-2}, ..., L_0$.

Remark 4.1. Let C_i be strongly connected class in the level L_p then C_i is closed with respect to the restricted HMM to the state space $S-(L_{p-1}, L_{p-2}, ..., L_0)$.

It is clear that, from Remark 4.1, the following algorithm finds the levels

Algorithm 4.1.

```
Beginning
$\Omega \leftarrow S$
$p \leftarrow 0$
$L_p = \{C_i : C_i$ closed at $\Omega\}$
Repeat
    $\Omega \leftarrow \Omega \setminus L_p$
    If $\Omega \neq \varnothing$ then
        $L_{p+1} = \{C_i : C_i$ closed for HMM restricted to $\Omega\}$
        $p \leftarrow p+1$
    End if
Until $\Omega = \varnothing$
```

Example: The classes $C_i; i=1,...,5$ shown in Fig. 1, construct three levels L_0, L_1 and L_2, the class C_1 which is situated in the level L_1, is closed by way of contribution of the level L_2.

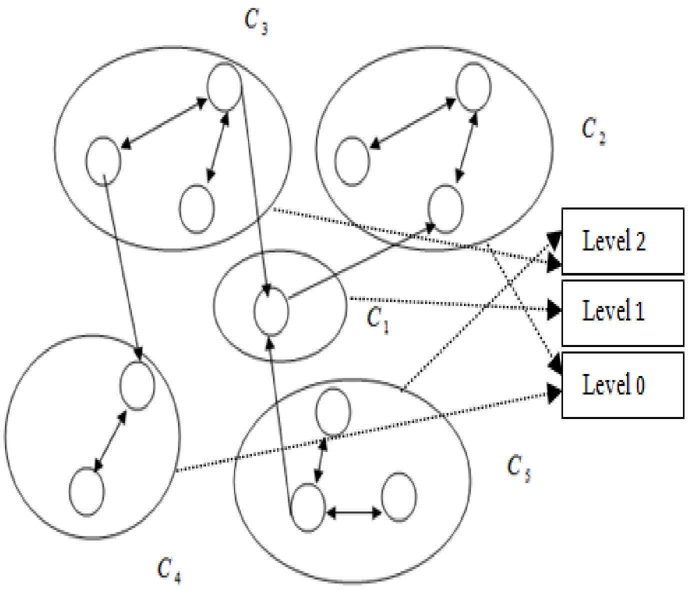

Fig. 1. Construction of levels

2) Restricted HMM for decoding problem using Forward algorithm

In what follows, we construct, by induction, the restricted HMMs corresponding to each level $L_p; p=n,...,0$ .Let (C_{pk}), $k\in\{1,2,...,K(p)\}$ be the K^{th} strongly connected class corresponding to the nodes in level p, where K(p) represent the maximum of the classes included in the level p.

a) Construction of the restricted HMM in level L_n

For each $k=1,2,\ldots K(n)$, we denote by HMM_{nk} the restricted HMM corresponding to the class C_{nk} that is the restricted HMM in which the state space restricted is $S_{nk}=C_{nk}$; the same M symbols $O_1,O_2,...,O_M$; the matrix of transition probabilities and the matrix of observation probability are restricted to S_{nk}.

b) Construction of the restricted HMM in level $L_p, p=n-1,...,0$

For each $k=1,2,\ldots K(p)$ and $p=n-1,...,0$ we denote by HMM_{pk} the restricted HMM corresponding to the class C_{pk}. Let $\Gamma^{-1}(C_{pk})$ be the set of predecessors for each state $i\in C_{pk}$ .The restricted HMM_{pk} defined by:

The state space: $S_{pk}=C_{pk}\cup\Gamma^{-1}(C_{pk})$.

The matrix of transition probabilities A: for each $j,\ i\in S_{pk}$, $A=\left[a_{ij}\right]_{i\in S_{pk}}$ where $a_{ij}=P_{pk}(j\,|\,i)$ if $j\in C_{pk}$.

The same symbols $O=\{O_1,O_2,...,O_M\}$.

The probability distribution of the initial state $\Pi=\left[\pi_i\right]_{i\in C_{pk}}$, where $\pi(i)=P(s_1=S_i)$ if $i\in C_{pk}$.

The matrix of observation probability $B=\left[b_j(m)\right]_{j\in C_{pk}}$, where $b_j(m)=P(o_t=O_m\,|\,s_t=S_j)$ if $j\in C_{pk}$.

V. Improved Forward Algorithm

We denote by $\alpha_{t,pk}(i),\ t\in\{1,...,T\}$, p=n,...,0 and k={1,..,K(p)} the Forward variable in state $i\in C_{pk}$.

Lemma 5.1. Let $j\in C_{pk}$, the Forward variable for j at time t+1 is defined by:

$$\alpha_{t+1,pk}(j)=\left[\sum_{i\in S_{pk}}\alpha_{t,pk}(i)\times a_{ij}\right]b_j(o_{t+1}) \qquad (9)$$

Proof. From equation (7) to calculate the Forward variable $\alpha_{t+1}(j)$ we need only the states i such as $P(x_t=j\,|\,x_{t-1}=i)\neq 0$, $\forall\ j\in C_{pk}$, these states belongs to the original set states of the class C_{pk} or $i\in\Gamma^{-1}(C_{pk})$.

Remark 5.1. To calculate the Forward variable $\alpha_{t+1,pk}(j)$ we need the Forward variable $\alpha_{t,pk}(i)$ for each $i\in\Gamma^{-1}(C_{pk})$, therefore, we always need some values that have been already calculated in the upper levels.

Algorithm 5.1.

Step1 : Initialization

For p=n,...,0 and $k=1,2,\ldots K(p)$; let $i\in S_{pk}$

$$\begin{aligned}&\alpha_{1,pk}(i)=\pi_i\times b_i(o_1) \text{ if } i\in C_{pk}\\&\alpha_{1,pk}(i)=\alpha_{1,mk'}(i) \text{ if } i\in\Gamma^{-1}(C_{pk}),\ (i\in C_{mk'}, m>p)\end{aligned} \qquad (10)$$

Step2 : Iteration

For $t\in[1,\ T-1]$, p=n,...,0 and $k=1,2,\ldots K(p)$; let $j\in S_{pk}$

$$\begin{aligned}&\alpha_{t+1,pk}(j)=\left[\sum_{i\in S_{pk}}\alpha_{t,pk}(i)\times a_{ij}\right]b_j(o_{t+1}) \text{ if } j\in C_{pk}\\&\alpha_{t+1,pk}(j)=\alpha_{t+1,mk'}(j) \text{ if } j\in\Gamma^{-1}(C_{pk}),\ (j\in C_{mk'}, m>p)\end{aligned} \qquad (11)$$

Step2 : Termination

$$P(O\,|\,\lambda)=\sum_{p=0}^{n}\sum_{k=1}^{K(p)}\sum_{i\in C_{pk}}\alpha_{T,pk}(i) \qquad (12)$$

VI. Complexity

The classical Forward algorithm generate N (N +1) (T-1) + N multiplication and N (N-1) (T-1) addition, it takes on the order of N^2T computations, which represents the quadratic complexity in the number of state. Whereas, the complexity of improved Forward algorithm can be calculated by using Akra Bazzi theorem [16] (A generalization to the well known Master Theorem [17]) which allows calculating the complexity for this type of problem. Therefore, it will be quasi-linear, equal to $N\log(N)$.

VII. Conclusion

The Forward algorithm progressively calculate the probability of an observation sequence, it is used in the recognition and learning because it represents the basis for revaluation in the Baum-Welch algorithm. We benefited from the method divide and conquer to reduce the charge of calculation of the Forward algorithm.

References

[1] Pellegrini T., Duée R., "Suivi de Voix Parlée grace aux Modèles de Markov Cachés," IRCAM, PARIS, 2003.

[2] Juang B. H., Rabiner L. R., "Hidden Markov Models for Speech Recognition," Technometrics, Vol. 33, No. 3. 1991, pp. 251-272.

[3] Ramy Al-Hajj M., Chafic Mokbel, Laurence Likforman-Sulem, "Reconnaissance de l'écriture arabe cursive: combinaison de classifieurs MMCs à fenêtres orientées," Université de Balamand, Faculté de Génie, 2006, pp 1-6.

[4] Ben Amara N., Belaïd A., Ellouze N., "Utilisation des modèles markoviens en reconnaissance de l'écriture arabe état de l'art," Colloque International Francophone sur l'Ecrit et le Document (CIFED'00), Lyon, France.

[5] Nakai M., Akira N., Shimodaira H., Sagayama S., "Substroke Approach to HMM-based On-line Kanji Handwriting Recognition," Sixth International Conference on Document Analysis and Recognition (ICDAR 2001), pp.491-495.

[6] Krogh A., I.Saira Mian, Haussler D., "A hidden Markov model that finds genes in E.coli DNA," Nucleic Acids Research, Vol. 22, No. 22.

[7] Morwal S., Jahan N., Chopra D., "Named Entity Recognition using Hidden Markov Model (HMM)," International Journal on Natural Language Computing (IJNLC), 2012, Vol. 1, No.4.

[8] Rabiner R., "A tutorial on Hidden Markov Models and Selected Applications in Speech Recognition," Proceding of the IEEE, Vol. 77, No. 2. 2010.

[9] Dequier J., "Chaînes de Markov et applications," Centre D'enseignement de Grenoble, 2005, pp.9-35.

[10] Khreich W., Granger E., Miri A., Sabourin R., "On the memory complexity of the forward–backward algorithm," Pattern Recognition Letters 31. 2010.

[11] Rust J., "Using Randomization to Break the Curse of Dimensionality," Econometrica, Vol. 65, No. 3, May, 1997, pp. 487-516.

[12] Van Caneghem M., "Algorithmes recursifs Diviser pour régner,", 2003.

[13] Canet L., "Algorithmique, graphes et programmation dynamique," 2003, pp 23-25.

[14] Cherki D., "decomposition des problemes de decision markoviens," Thesis of doctorat in Faculty of science Rabat, 2002, pp 64-69.

[15] Gondran M., Minoux M., "Graphes et Algorithmes" 2nd edition, 1990.

[16] Drmota M., Szpankowski W., "A Master Theorem for Discrete Divide and Conquer Recurrences," Austrian Science Foundation FWF Grant No. S9604.

[17] Roura S. (March 2001), "Improved Master Theorems for Divide-and-Conquer Recurrences," Journal of the ACM, Vol. 48, No. 2, pp. 170–205.