EXTENDED VITERBI ALGORITHM FOR SECOND ORDER HIDDEN MARKOV PROCESS

Yang He

Department of Electrical and Computer Engineering
State University of New York at Buffalo
Buffalo, NY 14260, U.S.A.*

Abstract

In this paper, an extended Viterbi algorithm is presented. The algorithm gives a maximum *a posteriori* estimation of the second order hidden Markov process. The advantage of the second order model and the complexity of the extended algorithm are compared with those of the original first order one. The method used to develop the extended algorithm can also be used to extend the Viterbi algorithm further to any higher order.

I. Introduction

Hidden Markov model (HMM) has been successfully used by many researchers in speech recognition and handwritten script recognition [1]-[3]. A solution to this model is the Viterbi algorithm. It gives, through an observation sequence observed in memoryless noise, an optimal estimation of the state sequence in the sense of maximum *a posteriori* probability [4], [5]. But in all previous research in those areas, the underlying Markov process is restricted to the first order one. So is the Viterbi algorithm applied to. If we can draw a higher order HMM from practical problems, we will be able, obviously, to incorporate more information into recognition procedure, which is specially meaningful to knowledge based systems. But to get to the solution, the Viterbi algorithm must be extended to be applied to higher order HMM. In the following sections, we will extend Viterbi algorithm to the second order hidden Markov process and compare its complexity with that of the first order one. At the end of this paper, it will be seen that an algorithm for any higher order HMM can be easily obtained by analogy of the method used in this paper. All examples used are chosen from script recognition.

II. Order Reduction

Let $X = (x_1, x_2, \ldots, x_K)$ represent an N state, K time-long second order Markov process, where x_i, $1 \leq i \leq K$, can be any one of the N states. Let $Z = (z_1, z_2, \ldots, z_K)$ represent an observation sequence of the process, where z_i, $1 \leq i \leq K$, can be any one of the M observation symbols. We assume that the observation is memoryless. In another word, for any i, observation z_i depends only on the time i state in the process sequence. Let following notations represent the probabilities we will use:

$P(X)$: Probability of state sequence X;

$P(X,Z)$: Joint probability of state sequence X and observation sequence Z;

$P(X|Z)$: Probability that state sequence is X, given observation sequence Z;

$P(Z|X)$: Probability that Z is observed when state sequence is X;

$p(x_1)$: Initial state probability;

$p(x_2|x_1)$: Probability of one step transition from time 1 to time 2;

$p(z_i|x_i)$: Probability that z_i is observed at time i, given the time i state x_i;

$p(x_i|x_{i-1}, x_{i-2})$: Two step transition probability.

Our aim is to find a particular state sequence X^*, when observation sequence Z is given, so that $P(X^*|Z)$ is maximum, or equivalently $P(X^*, Z) = P(X^*|Z)P(Z)$ is maximum. From above definition and assumption, it is easy to derive that

$$\begin{aligned} P(X,Z) &= P(X)P(Z|X) \\ &= p(x_1)p(z_1|x_1)p(x_2|x_1)p(z_2|x_2) \\ &\quad \times \prod_{i=3}^{K} p(x_i|x_{i-1}, x_{i-2})p(z_i|x_i) \end{aligned} \tag{1}$$

* Permanent address: Department of Electronic Engineering, Shenzhen University, Shenzhen, Guangdong, P. R. China.

In order to find the X^* with maximum $P(X, Z)$, we introduce a combined state sequence $Y = (y_1, y_2, \ldots, y_K)$, where $y_1 = x_1$ and $y_i = x_{i-1}x_i$,for $2 \leq i \leq K$. For example, for word “seems”, $X = (x_1, x_2, x_3, x_4, x_5) = (s, e, e, m, s)$ and $Y = (y_1, y_2, y_3, y_4, y_5) = (s, se, ee, em, ms)$. With this definition, we have

$$p(y_1) = p(x_1)$$

$$p(y_2|y_1) = p(x_2|x_1)$$

$$p(y_i|y_{i-1}) = p(x_i|x_{i-1}, x_{i-2}), \quad 3 \leq i \leq K$$

For example,

$$\begin{aligned} &p(y_3 = ee|y_2 = se) \\ &= P(\text{the third letter is “}e\text{” , given the second} \\ &\quad \text{letter “}e\text{” and the first letter “}s\text{”.)} \\ &= p(x_3 = e|x_2 = e, x_1 = s) \end{aligned}$$

Substituting these probabilities into Equation (1), we have

$$P(X, Z) = p(y_1)p(z_1|x_1)\prod_{i=2}^{K} p(y_i|y_{i-1})p(z_i|x_i) \quad (2)$$

Equation (2) is in the same form as that of the first order model. We can also draw a trellis to represent combined state transition, which is very similar to one step transition trellis.

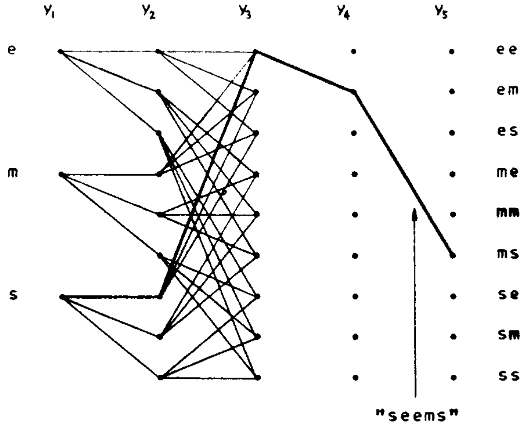

Figure 1. An example of combined state sequence

An example of 5 time–long sequence with 3 original state “e”, “m” and “s” is shown in Figure 1. Only transitions from time 1 to time 3 are drawn except the dark line, which represents the sequence of word “seems”. A complete trellis includes paths of all possible state sequences.

III. Recursive Tracing

Let us consider a partial state sequence $X_i = (x_1, x_2, \ldots, x_i)$ and a partial observation sequence $Z_i = (z_1, z_2, \ldots, z_i)$. For $3 \leq i \leq K$, we have

$$P(X_i, Z_i) = P(X_{i-1}, Z_{i-1})p(y_i|y_{i-1})p(z_i|x_i) \quad (3)$$

Equation (3) suggests that at any time $i \geq 3$, to find the maximum $P(X_i, Z_i)$ for each y_i, we need only to: (a) remember the maximum $P(X_{i-1}, Z_{i-1})$ for each y_{i-1}, (b) extend each of these probabilities to every y_i by computing Equation (3), and (c) select the maximum $P(X_i, Z_i)$ for each y_i. By increasing i by 1 until $i = K$ and repeating (a) through (c), the maximum $P(X, Z)$ for each x_K can finally be found. Then, among all of the $P(X, Z)$ for every y_K, we choose the maximum one and backtrace the sequence leading to this maximum probability. The result is the optimal combined state sequence Y^*. If the first original state in each combined state is omitted, the result is the optimal sequence X^*. At the initial step when $i \leq 2$, we compute

$$P(X_1, Z_1) = p(y_1)p(z_1|x_1) \quad (4)$$

and

$$P(X_2, Z_2) = P(X_1, Z_1)p(y_2|y_1)p(z_2|x_2) \quad (5)$$

From the definition of combined state, we know there are $N \times N$ combined state. But since from any y_{i-1}, the sequence can only transit to such a y_i that its first original state is identical to the second one in y_{i-1}, there are only $N \times N \times N$ non-zero values for $p(y_i|y_{i-1})$, corresponding to two step transition probability $p(x_i|x_{i-1}, x_{i-2})$.

In the following algorithm, combined states are not numbered from 1 to $N \times N$. Instead, two indexes, each from 1 to N, are used to denote combined states.

IV. The Algorithm

In order to describe the algorithm more clearly, we use following notations, which are more similar to the variables used in program:

$a_0(l)$—$p(y_1 = l)$;

$a_1(l, m)$—$p(y_2 = lm|y_1 = l)$;

$a_2(l, m, n)$—$p(y_i = mn|y_{i-1} = lm)$;

$b(z_i|n)$—$p(z_i|x_i = n)$;

$d_1(l)$—$P(X_1 = l, Z_1)$;

$d_i(m, n)$—maximum $P(X_i, Z_i)$ for $y_i = mn$, $2 \leq i \leq N$;

$c_i(m, n)$—state of x_{i-2} that maximizes $P(X_i, Z_i)$ for $y_i = mn$;

$x(i)$—time i state in the optimal sequence X^*, the final result.

In addition, we use symbol

$$\arg_m \max_{1 \leq m \leq M} [expression]$$

to denote a function whose value is the value of m that maximizes the value of the *expression*.

STEP 1. Initialization

a. For $1 \leq l \leq N$,

$$d_1(l) = a_0(l)b(z_1|l)$$

b. For $1 \leq l \leq N$,
for $1 \leq m \leq N$,

$$d_2(l,m) = d_1(l)a1(l,m)b(z_2|m)$$

This step computes $P(X_1, Z_1)$ for each $x_1 = l$ and maximum $P(X_2, Z_2)$ for each $y_2 = lm$.

STEP 2. Recursive Computation

For $3 \leq i \leq K$,
for $1 \leq m \leq N$,
for $1 \leq n \leq N$,

$$d_i(m,n) = \max_{1 \leq l \leq N} [d_{i-1}(l,m)a_2(l,m,n)]b(z_i|n)$$

$$c_i(m,n) = \arg_l \max_{1 \leq l \leq N} [d_{i-1}(l,m)a_2(l,m,n)]$$

At the end of this step, maximum probability $P(X,Z)$ for $y_K = mn$, $1 \leq m,n \leq N$, is found and stored in $d_K(m,n)$.

STEP 3. Determination of the Last Two States

$$x(K) = \arg_n \max_{\substack{1 \leq m \leq N \\ 1 \leq n \leq N}} [d_K(m,n)]$$

$$x(K-1) = \arg_m \max_{\substack{1 \leq m \leq N \\ 1 \leq n \leq N}} [d_K(m,n)]$$

STEP 4. Backtracing to the First State

For $K - 2 \geq i \geq 1$,

$$x(i) = c_{i+2}(x(i+1), x(i+2))$$

Now the optimal sequence is in $x(i)$.

V. Conclusion

In comparison with the Viterbi algorithm for the first order HMM [1], [2], we can see that the computation required for above algorithm is approximately N times as much as for the first order model. If N is not too large (for example, in script recognition, $N = 26$), this will not cause substantial difficulties.

Although experimental results are pending availability of properly calculated two step transition probability in each application field*, the advantage that second order model contains more information can be seen through observing some particular examples. In script recognition, for instance, if the first order model is used, then letter u can be followed by all letters except h, j, k, q and v through z [6]. But when a second order model is used, u can only be followed by a, e, i, o or y, if the u follows a q, the first letter in that word. Thus, the use of two step transition probability can eliminate much more possibilities of erroneous recognition.

From above discussion, we can see that it is easy to extend the algorithm further to any higher order model by following the same method. For example, if we combined each 3 consecutive original state to form combined state, we can easily derive an algorithm for the third order model.

References

[1] L. R. Rabiner, S. E. Levinson and M. M. Sondhi, "On the Application of Vector Quantization and Hidden Markov Models to Speaker Independent Isolated Word Recognition", *Bell System Technical Journal*, Vol. 62, pp. 1075–1105, Apr., 1983.

[2] A. Kundu and P. Bahl, "Recognition of Handprinted Script: a Hidden Markov Model Based Approach", *Proc. of ICASSP*, New York City, pp. 928–932, Apr., 1988.

[3] R. Nag, K. H. Wong and F. Fallside, "Script Recognition Using Hidden Markov Models", *Proc. of ICASSP*, Vol. 3, pp. 2071–2074, 1986.

[4] A. J. Viterbi, "Error Bounds for Convolutional Codes and an Asymptotically Optimum Decoding Algorithm", *IEEE Trans. on Information Theory*, Vol. 13, pp.260-269, Apr., 1967.

[5] G. D. Forney, "The Viterbi Algorithm", *Proc. of IEEE*, Vol. 66, pp. 268-278, No. 3, March, 1973.

[6] A. G. Konheim, "Cryptography: a Primer", Chapter 2, John Wiely and Sons, New York, 1982.

[7] A. Kundu, Y. He and P. Bahl, "Recognition of Handwritten Word: First and Second Order Hidden Markov Model Based Approach", *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, Ann Arbor, Michigan, June, 1988.

* By the time this paper was accepted, an experiment in handwritten word recognition had been done and shown the advantage of the second order model over the first order one [7].