An Efficient Algorithm for Sequentially finding the N-Best List*

Dennis Nilsson
Aalborg University
Fredrik Bajers Vej 7 E
9220 Aalborg Øst
Denmark
nilsson@math.auc.dk

Jacob Goldberger
The Weizmann Institute
of Science
Rehovot, 76100
Israel
jacob@wisdom.weizmann.ac.il

November 1999

Abstract

We propose a novel method to obtain the N-best list of hypotheses produced by a speech recognizer. The proposed procedure is based on efficiently computation of the N most likely state sequences of a hidden Markov model. We show that the entire information needed to compute the N-best list from the HMM trellis graph is encapsulated in entities that can be computed in a single forward-backward iteration that usually yields the most likely state sequence. The hypotheses list can then be extracted in a sequential manner from these entities without the need to refer back to the original data of the HMM.

*This is Research Report R-99-2022.

1 Introduction

In many tasks of large vocabulary speech recognition it is desirable to find from the HMM graph the N most likely word sequences given the observed acoustic data. The recognizer chooses the utterance hypotheses on the basis of acoustic information and a relatively simple language model. The existence of an N-best list enable us to combine additional knowledge sources such as complicated acoustic and language models into the recognition process [3]. Given the additional knowledge sources the list of sentence can be rescored and reordered. Even without additional knowledge sources the N-best paradigm can be used to improve the recognition rate [6]. In this paper we concentrate on the step of computing the N-best list. Several algorithms to compute the N-best list have been proposed in the last decade. These algorithms are based either on a Viterbi search of a trellis or on A^* search [4] [5]. This paper describes an efficient algorithm for finding the N most likely configurations in hidden Markov models (HMM). The algorithm presented is a special instance of the algorithm for finding the N most likely configurations in probabilistic expert systems [1].

2 Basic Structure

Consider a HMM with m hidden Markovian random variables $X_1 \dots X_m$ and m observed variables $Y_1 \dots Y_m$ such that the distribution of Y_t is determined by X_t. Denote $X = \{X_1, \dots, X_m\}$ and $Y = \{Y_1, \dots, Y_m\}$. Typical values that X and Y can take are denoted $x = (x_1, \dots, x_m)$ and $y = (y_1, \dots, y_m)$ respectively. The joint density distribution is :

$$P(x, y) = P(x_1) \prod_{t=2}^{m} P(x_t|x_{t-1}) \prod_{t=1}^{m} P(y_t|x_t) \tag{1}$$

This paper deals with the following decoding problem: Given observations $y_1 \dots y_m$, find the N most likely configurations of the unobserved state-variables. In other words we want to find the N values of x that maximize the joint density function $P(x, y)$ viewed as a function of x. A single iteration of the forward-backward algorithm [2] yields the following terms for each time index t:

$$\begin{aligned} f_t(s) &= \max_{\{x|x_t=s\}} P(x, y) \\ f_{t,t+1}(s, s') &= \max_{\{x|(x_t,x_{t+1})=(s,s')\}} P(x, y) \end{aligned} \tag{2}$$

We shall show that the entire information needed to compute the N-best list is encapsulated in the expressions defined in (2). In other words, once $f_t(s)$ and $f_{t,t+1}(s, s')$ are given there is no need to refer again to the trellis graph.

As a first example of the usefulness of f_t and $f_{t,t+1}$, we apply them to obtain the most likely state sequence. One can observe that the probability of the most likely configuration is :

$$\max_x P(x, y) = \max_s f_t(s)$$

This equality remain the same for each index $t = 1, \ldots, m$. The most likely configuration itself, denoted by $\hat{x} = (\hat{x}_1, \ldots, \hat{x}_m)$, can be obtained in the following manner (when this is uniquely determined) :

$$\hat{x}_t = \arg\max_s f_t(s) \qquad t = 1, \ldots, m$$

3 The N-Best Algorithm

In this section we provide an algorithm for finding the N most likely state configurations. Let $x^L = (x_1^L, \ldots, x_m^L)$ denote the Lth most likely configuration of $P(x, y)$. Denote the entire ensemble of possible state configurations by $\mathcal{X}$.

In the previous section we described a method for finding the most likely configuration. The algorithm described here will identify the configurations $x^2 \ldots x^N$ iteratively: x^L is found by going through the following three phases (by induction we may suppose that $\hat{x}, x^2 \ldots x^{L-1}$ have been identified):

- **Partition–phase** Here we partition the set $\mathcal{X} \setminus \{\hat{x}, x^2, \ldots, x^{L-1}\}$ into subsets. Many possible partitions exist; however a clever partitioning has two conflicting properties:

  1. **Subsets that we can 'conquer'**: The partitioning should consists of subsets that we can conquer, i.e. the maximum of $P(x, y)$ over each subset in the partitioning should be easily obtained.
  2. **Few subsets:** For efficiency reasons we want as few subsets as possible.

- **Candidate–phase** This phase finds the maximum of $P(x, y)$ over the subsets generated in the Partition-phase. These maxima can be regarded as our 'candidates' for the probability of the Lth most likely configuration. The maximum of these candidates is the probability of Lth most likely configuration.

- **Identification-phase** Here we identify the Lth most likely configuration.

First we discuss in detail how the algorithm identifies the second and third most likely configuration and then we describe the general case.

3.1 The second most likely configuration

Assume that the most likely configuration $\hat{x} = (\hat{x}_1, \ldots, \hat{x}_m)$ has been identified.

Partition–phase

The following m subsets form a partition of $\mathcal{X} \setminus \{\hat{x}\}$:

$$\begin{aligned} S_1 &= \{x \in \mathcal{X} |\ x_1 \neq \hat{x}_1\} \\ S_2 &= \{x \in \mathcal{X} |\ x_1 = \hat{x}_1, x_2 \neq \hat{x}_2\} \\ &\vdots \\ S_m &= \left\{ x \in \mathcal{X} |\ \begin{array}{l} x_1 = \hat{x}_1, \ldots, \\ x_{m-1} = \hat{x}_{m-1}, x_m \neq \hat{x}_m \end{array} \right\} \end{aligned}$$

Candidate–phase

From the definition of the functions f_t and f_{t+1} it follows that the maximum of the density function P within each of the subsets $S_1 \ldots S_m$ is given by:

$$\begin{aligned} \max_{x \in S_1} P(x, y) &= \max_{s \neq \hat{x}_1} f_1(s) \\ \max_{x \in S_2} P(x, y) &= \max_{s \neq \hat{x}_2} f_{1,2}(\hat{x}_1, s) \\ &\vdots \\ \max_{x \in S_m} P(x, y) &= \max_{s \neq \hat{x}_m} f_{m-1,m}(\hat{x}_{m-1}, s) \end{aligned}$$

Accordingly the probability of the second most likely state sequence is the highest of the above probabilities. The important result of this phase, however, is the index of the subset where the maximum is obtained.

Identification–phase

Suppose the second most likely state sequence $x^2 = (x_1^2, \ldots, x_m^2)$ belongs to the subset S_i. Thus

$$(x_1^2, \ldots, x_{i-1}^2) = (\hat{x}_1, \ldots, \hat{x}_{i-1}) \text{ and } x_i^2 \neq \hat{x}_i.$$

The sequence $x_i^2, \ldots, x_m^2$ can be found as follows:

$$x_i^2 = \arg \max_{s \neq \hat{x}_i} f_{i-1,i}(x_{i-1}^2, s)$$

When x_i^2 has been identified we can find x_{i+1}^2 as follows:

$$x_{i+1}^2 = \arg \max_{s} f_{i,i+1}(x_i^2, s)$$

Proceeding in this way we eventually identify the second most likely configuration.

3.2 The third most likely configuration

Identifying the third most likely configuration can be done in almost the same way as we identified the second most likely configuration: The procedure only differs slightly in the candidate phase where we use the probability $P(x^2, y)$ of the second most likely configuration to compute the probabilities of our new candidates.

Partition–phase

Suppose the second most likely configuration belongs to subset S_i. Note that the $m-i+1$ subsets partition $S_i \setminus \{x^2\}$.

$$\begin{aligned} T_i &= \{x \in S_i |\ x_i \neq x_i^2\} \\ T_{i+1} &= \{x \in S_i |\ x_i = x_i^2, x_{i+1} \neq x_{i+1}^2\} \\ &\vdots \\ T_m &= \left\{ x \in S_i |\ \begin{array}{l} x_i = x_i^2, \ldots, \\ x_{m-1} = x_{m-1}^2, x_m \neq x_m^2 \end{array} \right\} \end{aligned}$$

Candidate–phase

Now it can be shown (using Theorem 3 [1]) that the maximum of $P(x, y)$ within each of the subsets $T_i, \ldots, T_m$ is given by :

$$\begin{aligned} \max_{T_i} P(x,y) &= P(x^2,y) \frac{\max_{s \notin \{\hat{x}_i, x_i^2\}} f_{i-1,i}(x_{i-1}^2, s)}{f_{i-1,i}(x_{i-1}^2, x_i^2)} \\ \max_{T_{i+1}} P(x,y) &= P(x^2,y) \frac{\max_{s \neq x_{i+1}^2} f_{i,i+1}(x_i^2, s)}{f_{i,i+1}(x_i^2, x_{i+1}^2)} \\ &\vdots \\ \max_{T_m} P(x,y) &= P(x^2,y) \frac{\max_{s \neq x_m^2} f_{m-1,m}(x_{m-1}^2, s)}{f_{m-1,m}(x_{m-1}^2, x_m^2)} \end{aligned}$$

Note that the multiplicative factors:

$$P(x^2, y) \frac{1}{f_{j-1,j}(x_{j-1}^2, x_j^2)} \qquad j = i, ..., m$$

are needed to ensure that the maximization will be taken exactly on the desired subset.

Accordingly the probability of the third most likely configuration is either one of the above probabilities ($\max_{T_j} P(x, y)$) or is one of the probabilities computed earlier ($\max_{S_j : j \neq i} P(x, y)$).

Identification–phase

If the third most likely configuration belongs to one of the subsets S_j $(j \neq i)$ then it is identified in a similar way as the second most likely configuration was identified. Suppose the third most likely configuration belongs to the subset T_j. Clearly

$$(x_1^3, \ldots, x_{j-1}^3) = (x_1^2, \ldots, x_{j-1}^2)$$

The state x_j^3 can be found as follows :

$$x_j^3 = \begin{cases} \arg\max_{s \notin \{\hat{x}_j, x_j^2\}} f_{j-1,j}(x_{j-1}^3, s) & \text{if } j = i \\ \arg\max_{s \neq x_j^2} f_{j-1,j}(x_{j-1}^3, s) & \text{if } j > i \end{cases}$$

When x_j^3 has been identified we can identify x_{j+1}^3 as follows:

$$x_{j+1}^3 = \arg\max_s f_{j,j+1}(x_j^3, s).$$

and proceeding in this way we eventually identify the third most likely configuration.

3.3 The Lth most likely configuration

We shall now present the algorithm for finding x^L. Suppose we have identified $\hat{x}, x^2, \ldots, x^{L-1}$. Furthermore we are given a partitioning of the set $\mathcal{X} \setminus \{\hat{x}, x^2, \ldots, x^{L-2}\}$.

Partition–phase

To explain the general case we shall need a short notation for the subsets that are generated by the algorithm. Let $\mathcal{X}_j$ be the possible states that X_j can have. If $x^* = (x_1^*, \ldots, x_m^*)$ and $\mathcal{X}_j^* \subseteq \mathcal{X}_j$ then the couple $(x^*, \mathcal{X}_j^*)$ denotes the subset

$$(x^*, \mathcal{X}_j^*) = \{x \in \mathcal{X} | x_i = x_i^* \text{ for } i < j \ \wedge x_j \in \mathcal{X}_j^*\}$$

For instance, using this notation we can write $S_2 = \{x \in \mathcal{X} | x_1 = \hat{x}_1, x_2 \neq \hat{x}_2\}$ which was defined in section 3.1 as

$$S_2 = (\hat{x}, \mathcal{X}_2^*) \text{ with } \mathcal{X}_2^* = \mathcal{X}_2 \setminus \{\hat{x}_2\}.$$

As a first step in the partitioning we note that x^{L-1} belongs to one of the elements in the partitioning of $\mathcal{X} \setminus \{\hat{x}, x^2, \ldots, x^{L-2}\}$. Denote this element $(x^*, \mathcal{X}_j^*)$ and let

$$\mathcal{X}_j^{**} = \mathcal{X}_j^* \setminus \{x_j^{L-1}\} \text{ and } \mathcal{X}_i^{**} = \mathcal{X}_i \setminus \{x_i^{L-1}\}, i > j.$$

Note that the subsets $(x^{L-1}, \mathcal{X}_i^{**})$ for $i \geq j$ partition $(x^*, \mathcal{X}_j^*) \setminus \{x^{L-1}\}$.

Candidate–phase

From Theorem 3 [1] it can be shown that the maximum of $P(x,y)$ within each of the subsets $(x^{L-1}, \mathcal{X}_i^{**})$ for $i \geq j$ is given by (writing K for $L-1$)

$$\begin{aligned}
\max_{(x^K, \mathcal{X}_j^{**})} P(x,y) &= P(x^K, y) \frac{\max_{s \in \mathcal{X}_j^{**}} f_{j-1,j}(x_{j-1}^K, s)}{f_{j-1,j}(x_{j-1}^K, x_j^K)} \\
&\vdots \\
\max_{(x^K, \mathcal{X}_m^{**})} P(x,y) &= P(x^K, y) \frac{\max_{s \in \mathcal{X}_m^{**}} f_{m-1,m}(x_{m-1}^K, s)}{f_{m-1,m}(x_{m-1}^K, x_m^K)}
\end{aligned}$$

Accordingly the probability of the Lth most likely configuration is either one of the above probabilities or is one of the probabilities computed earlier. The search over all the probabilities can be efficiently performed in the following way. We can keep all the subsets computed earlier in a list sorted according to the associated probabilities. In the partition phase we split the subset that contained x^{L-1} (and hence was at the top of the list) into several subsets. In the Candidate phase we merge the new subsets into the sorted list according to the probabilities of the best state sequence in each subset. The sequence x^L belongs to the subset at the top of the updated list.

Identification–phase

The Lth most likely configuration belongs to a subset generated in the algorithm. Suppose the subset is $(x^*, \mathcal{X}_j^*)$. To identify x^L we note that

$$(x_1^L, \ldots, x_{j-1}^L) = (x_1^*, \ldots, x_{j-1}^*)$$

The state x_j^L can be found as follows :

$$x_j^L = \arg\max_{s \in \mathcal{X}_j^*} f_{j-1,j}(x_{j-1}^L, s)$$

When x_j^L has been identified we can identify x_{j+1}^L by

$$x_{j+1}^L = \arg\max_s f_{j,j+1}(x_j^L, s),$$

and proceeding in this way we eventually identify the Lth most likely configuration.

4 Conclusion

We have presented in this paper a novel method to compute the N most likely state sequences. The algorithm is most effective in cases where we do

not know in advance how many solutions are needed. The main concept is to perform a small preprocessing computation and then we can produce the sequences in an incremental manner. We have concentrated in this paper on applications of the algorithm to speech recognition problems. The proposed algorithm, however, can be applied to many other sources of information that are organized in a hidden Markov model e.g. analysis of DNA sequences and real time robot navigation.

Acknowledgement

This research was supported by DINA (Danish Informatics Network in the agricultural Sciences), funded by the Danish Research Councils through their PIFT programme.

References

[1] Nilsson, D. "An efficient algorithm for finding the M most probable configurations in probabilistic expert systems", *Statistics and Computing*, **8**, pp. 159–73, 1998.

[2] Rabiner, L.R. "A tutorial on hidden Markov models and selected application in speech recognition", *Proceedings of the IEEE*, vol 37, no. **2** pp. 257-86, 1989.

[3] Ostendorf, M. et al., "Integration of Diverse Recognition Methodologies Through Reevaluation of N-Best Sentence Hypotheses", In *Proceedings, DARPA Speech and Natural language Processing Workshop*, pp. 83–87, 1991.

[4] Schwartz, R. and Chow, Y.L. "A Comparison of Several Approximate Algorithms for Finding Multiple (N-Best) Sentence Hypotheses,", in *Proceedings IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 701–704, 1991.

[5] Schwartz, R., Nguyen, L., and Makhoul, J. "Multiple-pass search strategies". In *Automatic Speech and Speaker recognition*, C-H Lee, F.K. Soong and K.K. Paliwal eds, pp. 429–456, Kluwer Academic Publishers, 1996.

[6] Stolcke, A., Konig Y., and Weintraub M., "Explicit word error minimization in N-best list rescoring", *Proc. EUROSPEECH*, vol. **1**, pp. 163–166, 1997.