Efficient Search Strategies in Hierarchical Pattern Recognition Systems

Neeraj Deshmukh
Dept. of EE
Boston University
Boston, MA 02215

Joseph Picone
Inst. for Signal & Info. Proc.
Mississippi State University
MS State, MS 39762

Yu-Hung Kao
Systems & Info. Sci. Lab.
Texas Instruments
Dallas, Texas 75243

Abstract

We describe the generalized N-best search algorithm as applied to hierarchical pattern recognition, and discuss its limitations for a broad class of problems. We then introduce a new algorithm, called the Frame-Synchronous Viterbi Search, that prunes hypotheses by actively organizing system memory after each step in the search. This algorithm is shown to save memory and computation for a specific class of problems involving large search spaces and small memory resources. We also discuss generalizations of this algorithm to provide true N-best scoring and intelligent pruning while preserving the hierarchical structure of the hypotheses. Example of a practical speech recognition system using this algorithm will be given.

1 Introduction

The statistical approach to pattern recognition exploits the statistical relationships among various features in a pattern W. The features are modeled using probabilistic distributions and the available data or observations A are compared with these models. The recognition system generates a number of potential solutions or hypotheses with the objective of maximizing the probability of occurrence of W given A. The observation $\hat{W}$ corresponding to the highest probability score is then chosen as the recognized pattern.

$$p(\hat{W}/A) = \max_{W} p(W/A). \tag{1}$$

We first estimate the correct parameter values for our models by maximizing $p(W/A)$ over a known database of observation sequences. This is called "training" in speech recognition. During training, the recognizer also performs an automatic segmentation of data to determine the beginning and end of particular features. This enables the system to continue processing all hypotheses from the previous frame of data while restarting hypotheses from the top in the current frame. All of these result in an increase in the computational complexity of the system which now rises non-linearly with the duration of features.

Over the years Hidden Markov Models (HMMs) have evolved as the prime tool for modeling of features in a variety of pattern recognition problems. The popularity of HMMs lies in the availability of computationally efficient algorithms for both training and decoding. As HMMs have been applied to more complex recognition tasks, more efficient methods of modeling and search have necessitated for practical implementations. An upshot of this are systems that use a hierarchy of models for different levels of features. Each level of such systems is implemented as an HMM where the states of one level consist of HMMs of the previous level [Figure 1]. For example, in a continuous speech recognition system, sentences can be developed in terms of words, words in terms of phonemes etc.

Viterbi search is an efficient decoding technique to use with HMM-based models, but it is inadequate to handle situations where multiple hypotheses need to be passed on to another system. Recently, N-best search has emerged as a viable solution for such systems. We will now discuss these decoding strategies and their role in hierarchical pattern recognition systems. We then propose a frame-synchronous search algorithm to facilitate application of N-best search in hierarchical systems.

2 Search Techniques in Pattern Recognition

In pattern recognition the search paradigm chooses a pattern that has the highest likelihood for our feature models given the observed evidence. The number of possible hypotheses grows exponentially with the number of feature models, and imposes formidable requirements of computation and storage capability on the decoding implementation. Therefore techniques that save on computation by modifying the search

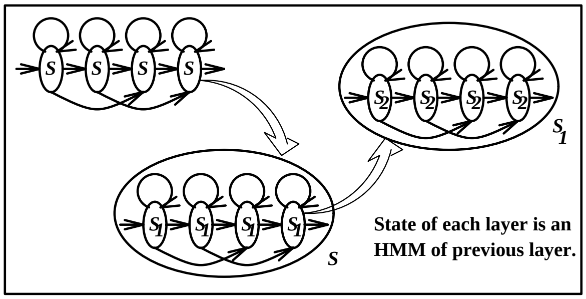


Figure 1: Hierarchical HMM network for recognition

space are vital from an implementation perspective. Such modifications cause the system to make suboptimal decisions, but these are known not to affect the accuracy of recognition significantly. A few popular techniques are briefly described here.

2.1 Viterbi Beam Decoding

The recognition system can be treated as a recursive transition network composed of the states of HMMs in which any state can be reached from any other [Figure 1]. The Viterbi search algorithm [6] builds a breadth-first search tree out of this network. Each state of the model is updated as the input data is processed frame by frame. At each frame all hypotheses are compared and only those having scores above a threshold value are allowed to continue forward.

The computational requirements of Viterbi search are proportional to the number of states of the model and the number of frames of data. Though its time-synchronous nature allows for comparison of competing hypotheses at any frame, this results in evaluation of a state model at every occurrence of the corresponding feature and represents superfluous computation.

2.2 Stack Decoding

Stack decoding [4] is a depth-first technique that maintains a sorted stack of most likely hypotheses. It effectively combines all available information into a single unified one-pass search. Fast match algorithms can be easily incorporated with stack decoding to advance the best hypothesis more efficiently. However, it suffers from problems of convergence and robustness.

2.3 N-Best Search

The optimal N-best decoding algorithm [1] is quite similar to the Viterbi decoder. However, N-best search finds all hypothesis sequences within the specified beam. It keeps track of hypotheses with different histories at each state. It then allows only the top N hypotheses to be retained for further processing. This state-dependent pruning is independent of the global Viterbi beam threshold.

The N-best search paradigm has found use in a variety of applications to incorporate information from various sources into a single framework. Knowledge sources that provide more constraint at a lesser cost are used to guide the initial search to generate the list of top N hypotheses. These hypotheses can later be re-evaluated with other, more expensive knowledge sources to arrive at the best hypothesis.

N-best search has the advantage of using only a subset of the available information to reduce the search space. However, in its pure form it has the flaw of being partial to shorter hypotheses. Also, most of the hypotheses picked in the first stage differ only marginally, and therefore result in much duplicated computation. This is overcome by using some generalizing approximations.

3 Generalized N-Best Search

Several modifications to the exact N-best algorithm [3] have been proposed in the continuous speech recognition (CSR) problem to make it more efficient and accurate. These modifications allow for some approximations and generate a list of sentence hypotheses with much less computation. Such approximations are justified as long as the correct hypothesis is assured to be in this list. Even if it does not hold a very high rank in this preliminary list, the correct hypothesis can be detected later by rescoring on other knowledge sources.

3.1 Forward-Backward Search

Forward-backward search algorithms [5] use an approximate time-synchronous search in the forward direction to facilitate a more complex and expensive search in the backward direction. This generally results in speeding up the search process on the backward pass as the number of hypotheses to be explored is greatly reduced by the forward search. A simplified model is used to perform a fast and efficient forward-pass search in which the scores of all partial hypotheses that fall above a threshold value are stored at every state. Then a simple beam search is performed in the backward direction. This uses more detailed models and scores high on a hypothesis only if it had a correspondingly good score on the forward pass.

3.2 Progressive Lattice Search

The progressive lattice search [2] is a generalization of the N-best algorithm; instead of generating an ordered list of hypotheses it generates a graph or lattice of connected features. The size of this lattice is controlled by the pruning threshold of the initial search. This lattice is then iteratively scaled down using a *forward-backward word-life algorithm* that prunes away nodes belonging to hypotheses with poor scores. This reduces the time required by the backward pass while adjusting the size of the resultant lattice.

3.3 Application to Hierarchical Systems

For pattern recognition systems using multi-level representations of knowledge, application of above search algorithms is not straightforward. Here the information of the best path has to propagate not only along but also across different strata of models. The Viterbi decoding scheme, being inherently 1-best, can be implemented in this setup; but requires excessive computations and large amounts of memory space with no appreciable gain in performance. Moreover, incorporating an N-best algorithm requires backtracing of N different best paths and further degrades the performance of the Viterbi search. Determination of the value of N and the relative weights of the N-best scores at different levels constitute further problems for optimization of the algorithm.

An obvious step in the direction of reducing computational requirement is to optimize the process of hypothesis generation itself and thus reduce the problem space. We propose a *frame-synchronous Viterbi search* algorithm that attempts to achieve this.

4 Frame-Synchronous Viterbi Search

In frame-synchronous Viterbi search (FSVS) we control the number of hypotheses generated in the conventional Viterbi beam search by limiting the number of models evaluated at each frame of the input data. This is done by sorting all active hypotheses in decreasing order of path score and pruning away all but a few top-scoring hypotheses at the end of every frame. Thus at any level the total number of hypotheses forwarded to the next frame is limited.

This pruning of hypotheses is different from the Viterbi beam pruning. While the Viterbi scores are compared with the pruning threshold only at the top level in the hierarchy, the FSVS pruning is carried out at all levels. FSVS pruning can be dynamic where the pruning threshold is decided according to the level and/or in an adaptive fashion. Alternatively, it can be static i.e. fixed by some upper limit on the number of possible hypotheses. The choice of pruning strategy and the value of the threshold (or limit on number of hypotheses) are specific to the application.

Care must be taken in choosing the threshold for frame-level pruning in order to avoid over-pruning of hypotheses, for this will stop even potentially correct hypotheses from advancing and affect the accuracy of recognition.

4.1 Computation and Storage Issues

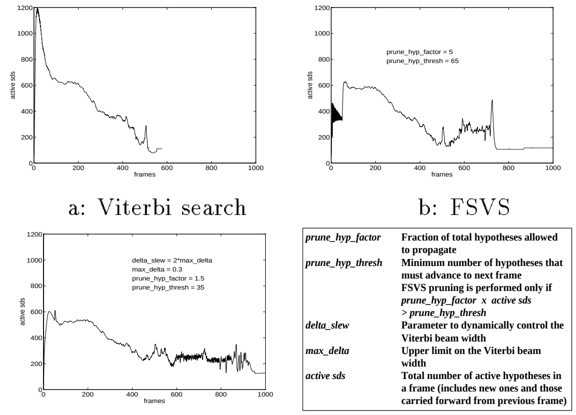
While the goal of FSVS is to save on computation by reducing the number of hypotheses, frame-level pruning requires hypothesis scores to be sorted at every frame. We have observed that though this brings down the net gain in computational expenditure, the savings in memory are considerable. The computational load can be further reduced by exploiting the pattern of hypothesis generation [Figure 2a]. In the first few frames of the observed data the recognizer has insufficient knowledge to decide on more likely patterns. Therefore it generates a very large number of hypotheses, most of which are gradually pruned away due to extremely poor scores in presence of further evidence. The initial frames constitute the region where about 95% of the new hypotheses are generated and hence have the maximum demands on memory. We need to limit FSVS pruning only to these frames. Thus FSVS is most suitable for recognition tasks where the available memory is not sufficient to manage decoding over a large search space.

5 Results

The FSVS has been implemented on HG (*H*ierarchical *G*rammar), a continuous speech recognizer developed at Texas Instruments that uses a multi-level description of acoustic and language models. Speech units such as sentences, words, syllables, phones etc. are represented as HMM levels. Each state in the HMM grammar is associated with a statistical acoustic model which is a random vector of acoustic parameters with an underlying continuous multivariate Gaussian probability density. Various states of HMMs can share the same statistical acoustic model, thus enabling construction of a complex HMM topology that models the temporal course of speech in a compact fashion.

The algorithm was evaluated on a set of 1390 files from the VAA02 database[1], each file containing a 10-digit string. The number of hypotheses generated at

[1] The VAA02 database is part of Texas Instruments' ongoing *Voice Across America (VAA)* corpus building program and contains speech recorded over standard telephone lines.



a: Viterbi search

b: FSVS

prune_hyp_factor	**Fraction of total hypotheses allowed to propagate**
prune_hyp_thresh	**Minimum number of hypotheses that must advance to next frame** **FSVS pruning is performed only if *prune_hyp_factor x active sds* > *prune_hyp_thresh***
delta_slew	**Parameter to dynamically control the Viterbi beam width**
max_delta	**Upper limit on the Viterbi beam width**
active sds	**Total number of active hypotheses in a frame (includes new ones and those carried forward from previous frame)**

c: Dynamic FSVS

Figure 2: Patterns of hypothesis generation

any frame is counted in terms of *scoring data structures (sds)* which are slots in the memory buffer keeping track of the hypothesis scores and the models associated with them. The Viterbi beam can be constant or set adaptively during search. We fixed the number of hypotheses to advance at any frame as a fraction of the total number of hypotheses. Total available memory was fixed at 2500 slots and number of slots available at each frame was fixed at 600.

FSVS pruning was carried out on the first 50 frames of each file. Figure 2 describes the effect of FSVS on hypothesis generation. The experimental results are displayed in Table 1.

Type of pruning	Sent. over flow	Comptn. (fracn of realtime)	Mem. slots/ frame	Word error %	Sent. error %
Viterbi	308	0.289	590	24.2	36.1
FSVS	77	0.259	440	9.2	26.5
DFSVS	0	0.274	424	3.5	21.2

Table 1: Results of FSVS

It can be observed that FSVS works best in association with dynamic Viterbi pruning (DFSVS). Even with static Viterbi beam search it manages to reduce memory overflow and improves upon the error rate.

6 Conclusion

Hierarchical pattern recognition systems, though potentially powerful in solving complex tasks of large magnitude, suffer from acute problems of excessive computational and storage requirements for decoding. Search techniques which have proven advantageous in unilayered systems suffer from loss of efficiency in a multi-layered environment. The implementation framework also needs to be reworked for these algorithms to suit a magnified problem space.

The Frame-Synchronous Viterbi Search algorithm attempts to reduce the problem space by modifying the process of hypothesis generation. Though the reduction in computation is partly offset by the overhead required for frame-level pruning, the gain in memory is substantial making this technique particularly attractive to memory-critical pattern matching applications.

There are numerous applications of hierarchical pattern recognition systems ranging from target recognition in radar, defense, and security systems, to more exotic problems in fingerprint matching, very low rate image compression, and intelligent access of video databases. Efficient search strategies enable the implementation of more sophisticated statistical signal models, in many cases replacing technology based on extensive sets of heuristics. Overall system performance can be significantly improved through powerful closed-loop training techniques. Our future research will be oriented towards developing more efficient search strategies for hierarchical systems, and to introduce N-best search algorithms into such systems.

References

[1] Chow Y. L. and R. M. Schwartz, "The N-Best Algorithm: An Efficient Procedure for Finding Top N Sentence Hypotheses", *Proceedings DARPA Speech and Natural Language Workshop*, pp. 199-202, October 1989.

[2] Murveit H., J. Butzberger, V. Digalakis and M. Weintraub, "Progressive-Search Algorithms for Large-Vocabulary Speech Recognition", *Proceedings DARPA Human Language Technology Workshop*, March 1993.

[3] Nguyen L., R. Schwartz, F. Kubala and P. Placeway, "Search Algorithms for Software-Only Real-Time Recognition with Very Large Vocabularies", *Proceedings DARPA Human Language Technology Workshop*, pp. 91-95, March 1993.

[4] Paul D. B., "An Efficient A^* Stack Decoder Algorithm for Continuous Speech Recognition with a Stochastic Language Model", *Proceedings ICASSP*, pp. 405-409, March 1992.

[5] Schwartz R. M. and S. Austin, "Efficient, High-Performance Algorithms for N-Best Search", *Proceedings DARPA Speech and Natural Language Workshop*, pp. 6-11, June 1990.

[6] Viterbi A. J., "Error Bounds for Convolutional Codes and an Asymptotically Optimal Decoding Algorithm", *IEEE Transactions on Information Theory*, Vol. IT-13, pp. 260-269, April 1967.