AN ADAPTIVE-BEAM PRUNING TECHNIQUE FOR CONTINUOUS SPEECH RECOGNITION

Hugo Van hamme and Filip Van Aelten

Lernout & Hauspie Speech Products N.V.
Koning Albert 1 laan, 64
B 1780 WEMMEL
Belgium
email: vanhamme@brussels.lhs.be

ABSTRACT

Pruning is an essential paradigm to build HMM-based large vocabulary speech recognisers that use reasonable computing resources. Unlikely sentence, word or subword hypotheses are removed from the search space when their likelihood falls outside a beam relative to the best scoring hypothesis. A method for automatically steering this beam such that the search space attains a predefined size is presented.

1. INTRODUCTION

The core of a HMM-based recogniser is the dynamic programming, which evaluates decoding hypotheses and selects the single best one. Alternatively, a list of likely candidates (N-best) can be extracted. When the vocabulary size becomes large, the perplexity increases and/or long N-best lists are requested, the search space becomes prohibitive. The idea of pruning is to exclude hypotheses from further investigation if they turn out to be unlikely based on an evaluation on partial data. Of course, a path that was unlikely up to now due to a mismatch with the model may become more likely in the future. Hence, the challenge of designing a good pruning mechanism is to reduce the search space size maximally, without removing the hypotheses that would have been withheld without it.

2. PRUNING STRATEGIES - NOTATIONS

Consider a frame-based HMM speech recognition system. At each frame or time t, the search space is made up of HMM states q which correspond to a well-known state sequence(s) ending in that state q. In turn, a state sequence corresponds to a (partial) sentence hypothesis. The negative logarithm of the likelihood of the observed speech data over frames $1 \cdots t$ given that state sequence is associated to each q and will be called the (cumulative) score $l_t(q)$. The "best state" at time t, $\tilde{q}_t$, is the one with the minimal score. States with a score which is more than the pruning beam T_t above the score of the best state are removed from the search space. The choice of T_t defines the pruning strategy.

The simplest option is to have a time-independent T_t [1]. Although sensible from an acoustical point of view, it only gives indirect control over the search space size. When a partial hypothesis has many possible extensions, the number of states will increase dramatically, although many of these extended hypotheses will be removed later. Hence, there is an instantaneous peak in memory and CPU requirements.

In histogram pruning, T_t is chosen as to keep N_{set} states after pruning. The score of each state is computed, a histogram is made and the threshold to retain the targetted number of states can be determined with an accuracy determined by the histogram bin size [2]. The main disadvantage of this strategy is that two passes through all states are required.

In the present algorithm, pruning is regarded as an adaptive control problem [3]. Automatic steering of T_t limits the size of the search space. The bound on the number of states after pruning is however less rigid than in histogram pruning, but the cost of pruning is lower. Section 3 formulates the pruning problem in the context of adaptive control theory. Section 4 handles some details and section 5 analyses the performance of adaptive pruning in a context of large vocabulary recognition.

3. ADAPTIVE PRUNING

The main purpose of pruning is the limitation of the search space size. A good measure for this size at time t is the number of states n_t that are currently taken into account in the dynamic programming (the active states), since it has a direct impact on the CPU and memory requirements of the recogniser. The pruning process is regarded as a non-linear time-variant dynamical system (the plant) with T_t as its input and n_t as its output. The non-linear, time-variant and dynamic nature of the plant is obvious when realising that its input/output characteristics depend on the hypotheses currently withheld, on the amount of new sentence hypotheses they can expand into and on the speech data. The control goal is to withhold N_{set} states, i.e. to drive n_t to N_{set}. In order to track a step in n_t without asymptotic error, at least one integration is required in the closed loop system. Since it is possible to control the average n_t roughly with a fixed

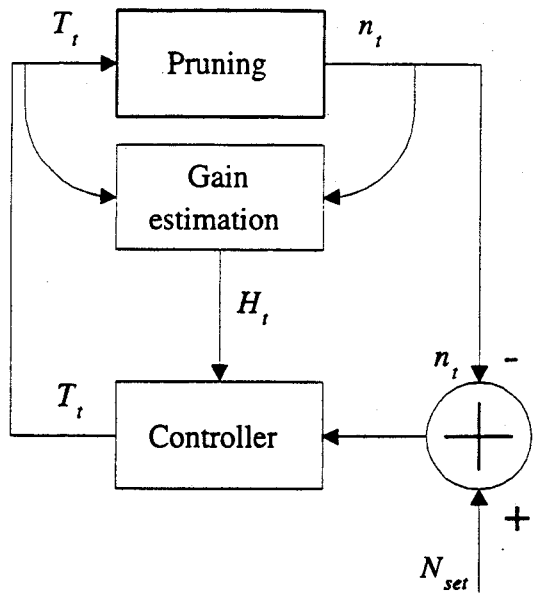

Figure 1: Topology of the adaptive controller for search space pruning.

pruning beam [1], the plant does not contain this integration and it must be put in the controller.

For simplicity, the plant is modeled as a zeroth order linear system with "slowly" varying gain: $T_t = G_t n_t$, where G_t is the reciprocal of the time-variant plant gain. The controller is an integrator: $T_{t+1} = T_t + \alpha H_t(N_{set} - n_t)$, where H_t is an estimate of G_t and α is a non-critical parameter that affects the response time of the closed loop system. The topology of the system is shown in figure 1.

Assuming that the plant model is correct and that the plant gain is estimated correctly, the closed loop system will track N_{set} like a first order system with pole at 1-α. Hence, a reasonable value for α is 0.2. The estimate H_t is a least-squares fit of G_t based on the past L observations of the beam and the number of active states:

$$G_t = \frac{\sum_{i=1}^{L} n_{t-i} T_{t-i}}{\sum_{i=1}^{L} n_{t-i}^2} \quad (1)$$

Since the plant is apt to change quickly, L cannot be chosen too large. On the other hand, a noisy gain estimate based on few data points may lead to bad closed loop plant behaviour [3]. A good trade-off is $L = 5$. Equation 1 must be computed only once per frame, which is a negligible cost.

4. EXTENSIONS

In some cases, especially at the beginning of an utterance, it is impossible to reach N_{set} states. The controller will however increase the beam to attempt the impossible. Therefore, activation of the controller must be delayed until enough states are active. In other situations, very unlikely hypotheses must be included in order to activate N_{set} states. Hence, it is good to limit the beam above. On the other hand, the gain estimation may fail or the plant model may be inappropriate. Therefore, it is also good to limit the beam below. If the purpose of pruning is to control the CPU load rather than the peak memory requirements, this minimal beam can be chosen conservatively.

To achieve pruning as stated above, more than one pass through the active states is required. The pruning threshold above which a score is considered unlikely is computed as $\theta_t = l_t(\tilde{q}_t) + T_t$. Pruning based on this threshold requires the expansion of all states from $t-1$ to t, finding the best score, then reducing the search space using a second pass over all states. To achieve pruning in a single pass, the pruning threshold at time t can be computed with respect to the best cumulative score at time $t-1$: $\theta_t = l_t(\tilde{q}_{t-1}) + T_t$. The bias created by this approximation is automatically removed due to the integration in the controller. The lower limit on the beam must be chosen conservatively to avoid that no states would be withheld upon observing an unlikely frame (high local scores) at time t and when pruning needs to be tight. An alternative is to first expand only part of the states from $t-1$ to t. Interesting options are expansion of the best state at $t-1$ and/or expansion of the self-loops on the active states. This will yield an upper bound on $l_t(\tilde{q}_t)$ and the risk of pruning all states has vanished for positive T_t.

5. EXPERIMENTS

Adaptive pruning is evaluated on an isolated word recognition task and on a continuous speech recognition task, both using discrete-density HMMs. No cost-reducing measures such as phoneme-lookahead are taken. The lexicon is organised "linearly", i.e. all initial states of follower words are activated simultaneously. This is in strong contrast to the use of lexical trees where the activation of states is more progressive in time, making the control problem easier. In the experiments reported below, N_{set} is chosen to have no loss in recognition performance.

In isolated word recognition, the active vocabulary consists of 500 names (first and last). Without pruning, the search space consists of over 24000 HMM states. The behaviour of n_t and B_t is given in figure 2 for a randomly selected utterance. The adaptive controller is able to steer the average n_t to the desired value of 5000 active states. However, some overshoot must be allowed. Hence, adaptive pruning is more suitable for CPU control than for hard limitation of the memory resources. Simple strategies of minimal damage upon hitting the memory bounds can however be proposed. Notice that the beam in figure 2 is steered to extremely high values near the end of the utterance. In that situation, the acoustics allow the elimination of many hypotheses and there is no need to maintain a large search space. Hence, it is wise to limit the pruning beam above - see figure 3.

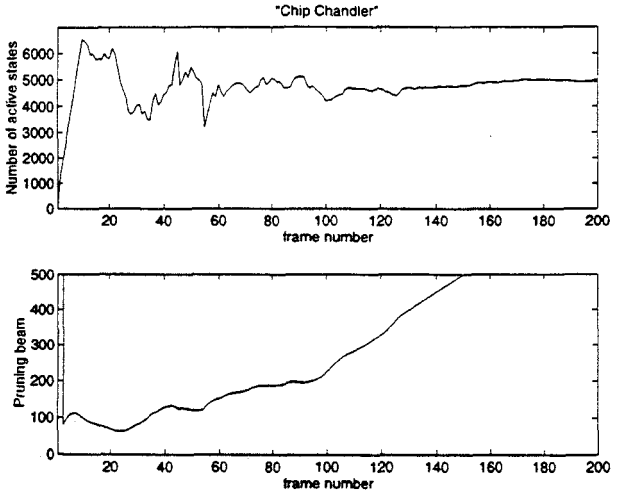

Figure 2: Number of active states and pruning beam for an isolated word syntax - no upper beam limit.

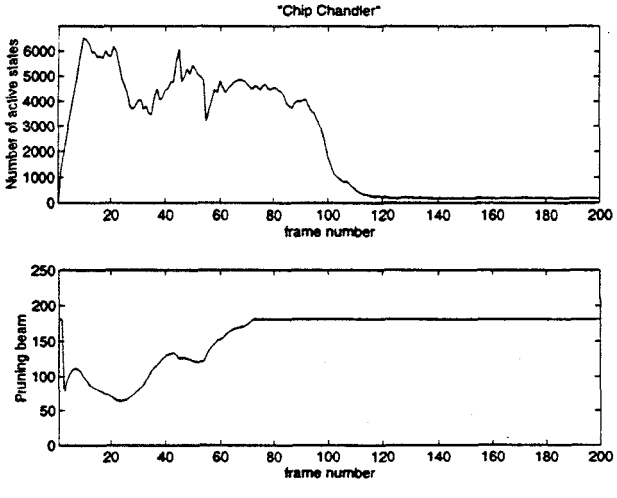

Figure 3: Number of active states and pruning beam for an isolated word syntax - upper beam limited.

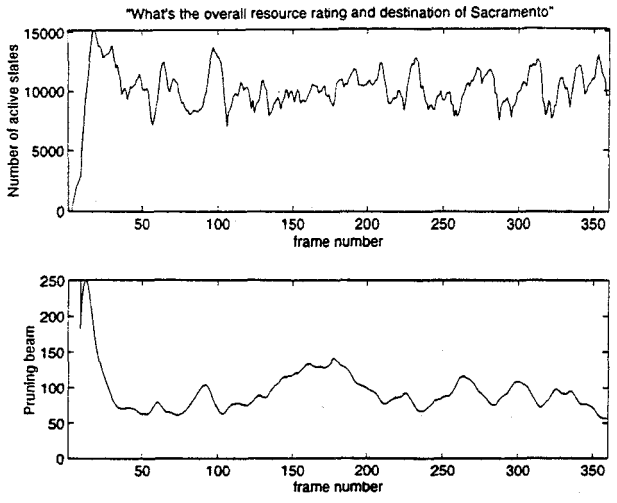

Figure 4: Number of active states and pruning beam for a continuous speech syntax.

In continuous speech, new word hypotheses are created at word endings. Since the lexicon is not organised in a network or tree, sudden increases in the number of active states for a fixed pruning beam are expected each time a word ending is reached. Adaptive pruning is evaluated on the 1000-word continuous speech Resource Management task, which uses a word-pair syntax. Figure 4 shows the behaviour on a particular sentence ("What's the overall resource rating and destination of Sacramento"). Again, the system is able to steer the number of active states to the setting value of 10000, but with some overshoot at the points of large ambiguity. However, some overshoot is helpful in preserving the recognition performance at a lower average number of states. On the average, adaptive pruning attains the same goals as histogram pruning, but it is cheaper. The benefits of histogram pruning over fixed beam pruning were already argumented in [2] and need not be repeated here.

In a last series of experiments, the validity of the assumptions made in section 3 is verified. Figure 5 shows the characteristics of the plant. The static transfer of pruning beam to the number of active states for the continuous speech task is plotted for 10 consecutive frames, taken over different time intervals. These plots show clearly that the plant is strongly nonlinear. This nonlinearity is grossly approximated by a straight line in the plant model. The changes in gain over short periods seem to be less important. Because the continuous speech recognition task generates new word hypotheses almost every frame, the long-term changes in plant gain are quite modest. In figure 6, the same transfer is shown for the isolated word task at several frames distributed over the whole utterance. In this task, the ambiguity is large in the beginning of the utterance. As more data becomes available, the plant gain decreases. This shows the necessity of the adaptivity of the controller.

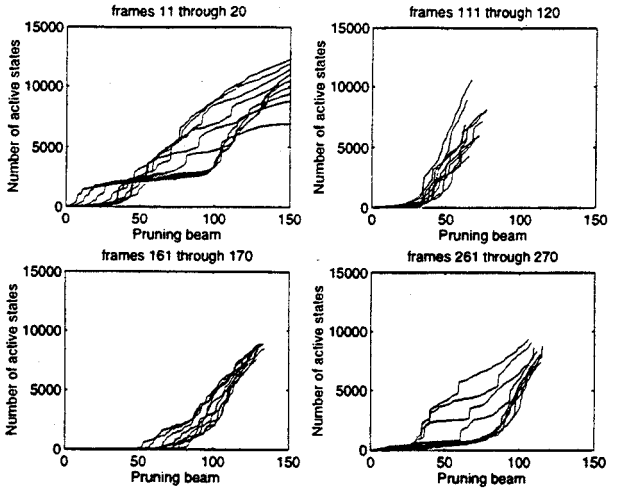

Figure 5: Transfer of the pruning plant for 10 consecutive frames, starting at different times - continuous speech task.

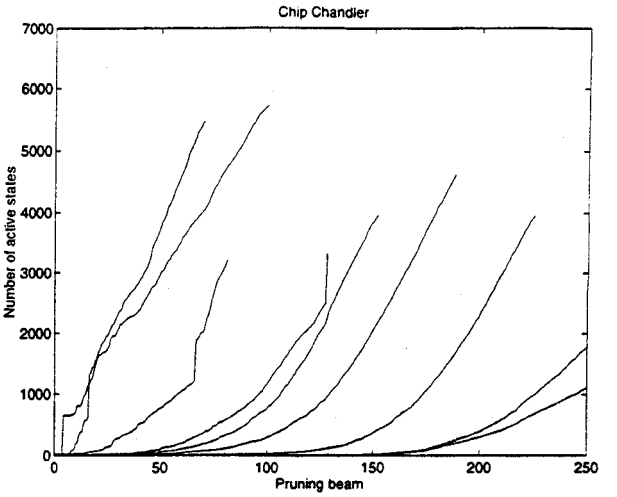

Figure 6: Transfer of the pruning plant - isolated word task.

6. CONCLUSIONS

Application of adaptive control theory to the problem of search space size reduction in the dynamic programming of HMM recognisers was shown to yield a cheap pruning strategy that attains its control goals well and compares favourably to histogram pruning. Although the memory resources can be restricted successfully, its prime application is in CPU load control.

7. REFERENCES

1. A. Noll, H. Ney, D. Mergel and A. Paeseler. Data driven search organization for continuous speech recognition. *IEEE Trans. SP*, 40:272–281, February 1992.

2. B.-H. Tran, V. Steinbiss and H. Ney. Improvements in beam search. In *Proceedings ICSLP*, pages 2143–2146, 1994.

3. G. C. Goodwin and K. W. Sin. *Adaptive Filtering, Prediction and Control.* Prentice-Hall, Englewood Cliffs, New Jersey, 1984.