PROGRESS IN SPEAKER RECOGNITION AT DRAGON SYSTEMS

Andres Corrada-Emmanuel, Michael Newman, Barbara Peskin, Larry Gillick, Robert Roth

Dragon Systems, Inc.
320 Nevada Street, Newton, MA 02160

ABSTRACT

We present a new algorithm for speaker recognition (the Sequential Non-Parametric system, or SNP) that has the potential to overcome two limitations of the current approaches. It uses sequences of frames instead of one frame at a time; and it avoids the need to model a speaker with mixtures of Gaussians by scoring the data non-parametrically. Although at an early stage in its development, SNP's output can be interpolated with that of our GMM system to outperform state-of-the-art GMM's. Comparative results are presented for the 1998 NIST Speaker Recognition Evaluation test set.

1. INTRODUCTION

The most popular algorithm for speaker recognition systems is the Gaussian Mixture Model (GMM). It is fast, relatively simple, and offers good performance. Its simplicity, however, means that it ignores useful linguistic information that should allow one to improve recognition performance.

Dragon Systems believes that using Large Vocabulary Continuous Speech Recognition (LVCSR) is one way to develop future algorithms that can outperform the GMM approach. We have discussed this approach in several places (e.g. [1]); in particular, we presented an LVCSR-based system in [2]. We have been able to improve the performance of this system to the point that it yields comparable performance to a GMM system when there is enough training and test data. We begin by briefly describing these improvements in section 2.

We will then discuss a second approach to using speech recognition. This new algorithm, the Sequential Non-Parametric system (SNP), abandons the use of parametric speaker models and instead relies on non-parametric comparisons between the training and test speech data.

Dragon Systems is not the first to use a non-parametric approach to speaker recognition (see, for example, [3] and [4]). The novel feature of SNP is its use of sequential information at the frame level. We close by discussing a series of experiments exploring the relevance of this information to system performance.

2. GMM AND LVCSR IMPROVEMENTS

At ICSLP in 1996, we presented early results (too late to be included in the proceedings) comparing our LVCSR speaker identification system (LVCSR) with one using a Gaussian Mixture Model (GMM). At the time, our GMM was far superior to our LVCSR, but in the last two years, we have made significant improvements to both, and especially to the LVCSR, so that the gap in performance has almost disappeared for sufficiently long test utterances.

In our LVCSR system, we use a somewhat simplified version of our standard speech recognizer to transcribe the speech data and then time-align the speech to these (errorful) transcripts. These time-alignments are then scored with speaker-adapted monophone models. (See [2] for details.)

Our GMM was modeled after the system designed by Doug Reynolds (described in [5]), to allow us to study the differences between the two systems in as controlled a manner as possible. For example, the signal-processing and adaptation algorithms were the same for the two systems.

We have since made three major changes, each of which improves the performance considerably. The effect of these changes is illustrated here on a subset of the 1996 NIST Speaker Recognition database.

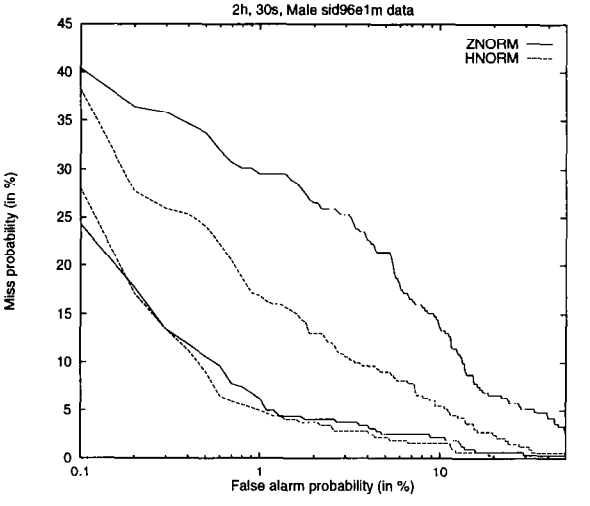

Figure 1: Gain from HNORM (GMM) (matched and mismatched handsets).

First, we implemented an algorithm for normalizing the scores from different speakers, to allow pooling of scores before a decision threshold is applied. The simpler version of this (commonly referred to as ZNORM) uses a set-aside corpus of development (impostor) data to compute for each target speaker the mean and variance of non-target test utterances. Given the score of a test utterance against a particular target, we normalize the score by subtracting the mean, and dividing by the standard deviation. A more sophisticated version of this algorithm (HNORM) was developed by Reynolds [5], where he showed how to take advantage of side-knowledge of the handset type, by computing separate normalizations for carbon and electret

handsets. At test time, a handset detector is used to determine which normalization should be applied to a given test utterance. Subsequently, scores are pooled in the usual fashion.

In figure 1, we show the effect of HNORM vs. ZNORM on our GMM for both matched (training and test data come from the same handset type) and mismatched test data. We see that matched test data (the lower two lines) are essentially unchanged, but HNORM helps considerably on mismatched test utterances.

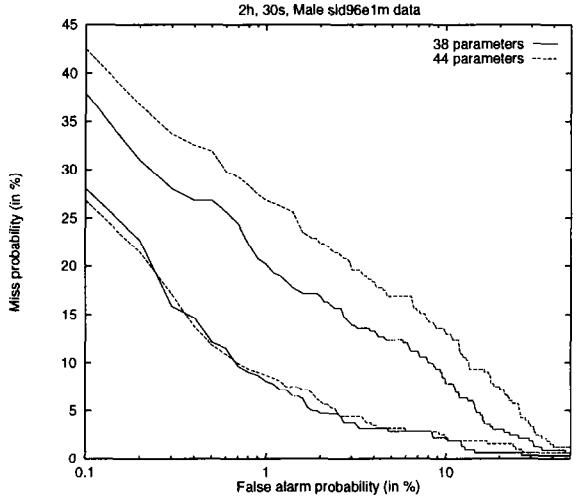

Figure 2: Effect of parameter set (GMM) (matched and mismatched handsets).

Next, we changed our signal-processing feature set to keep all 19 cepstral coefficients plus first differences (for a total of 38), instead of our conventional recognition parameter set of the first 12 each of cepstrals, first, and second differences, plus 8 spectrals (for a total of 44). The results are shown in figure 2. This was a clear improvement on the mismatched data. We had tried this before, but we did not see any improvement until we implemented HNORM. We speculate that the gross errors arising from the lack of handset normalization masked the improvement from the better signal processing.

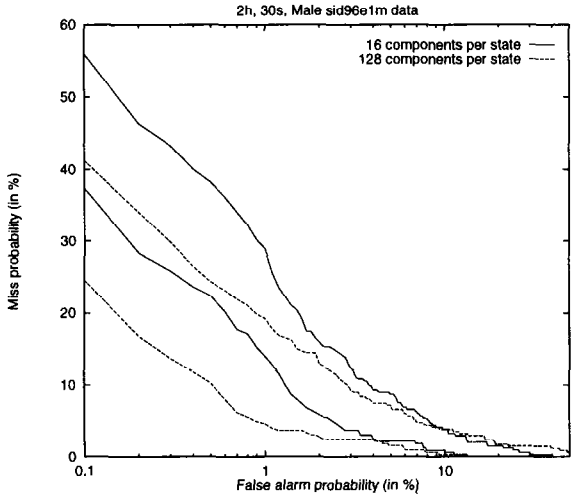

Figure 3: Number of components in each state (LVCSR) (matched and mismatched handsets).

These two changes helped our LVCSR and GMM systems equally. For the LVCSR, we got an additional substantial gain from increasing the number of Gaussians in each mixture. We had previously used 16 Gaussians for each state of each phoneme, for a total of about 1900 Gaussians, which was about the same size as the GMM. However, it turns out that performance increases dramatically with number of Gaussians, saturating at around 128 per mixture. This change is shown in figure 3, and by itself is enough to narrow the gap between the two systems so that for the 30-second test condition, it has almost disappeared. For shorter test utterances, the GMM is still clearly ahead, which we speculate is due to the higher word error rate on the short utterances. (Of course, this raises the intriguing possibility that the LVCSR might win for longer test utterances, or if the word error rate were lower.)

3. THE SNP ALGORITHM

3.1 Motivation

Our development of the SNP algorithm was motivated by two ideas. The first is that the way a person's speech traverses acoustic space carries additional information about his or her identity. A GMM system is unable to utilize this information since it scores each speech frame independently of all others. Our LVCSR system implicitly uses some of this sequential information since it scores time-alignments produced by a word-level recognizer. However, any HMM-based system has some smallest unit below which it assumes frame independence.

Our second motivating idea was that Gaussian mixture models might be too coarse for encoding a speaker's speech. This led us to consider a non-parametric method of comparing speech segments.

3.2 Phoneme-level labels for data

We begin by using the same transcription and time-alignments as in our LVCSR speaker recognition system. These time-alignments are highly errorful since they are based on recognition transcripts produced by a recognizer with a WER of nearly 50%.

The time-alignments allow us, in principle, to compare sequences at many levels. We could, for example, compare words instead of phonemes. Our actual choice of phoneme-level comparisons was driven by two considerations: we wanted a unit small enough that many comparisons could be made between two speakers, but with enough frames to capture the sequential information we wanted to use. Phonemes are a reasonable compromise that satisfies both constraints. We call the resulting units into which the speech stream is partitioned "tokens".

3.3 Comparing a target and test speaker

The heart of the SNP algorithm is the comparison of speech tokens. When we compare two tokens, we use a standard dynamic programming algorithm to find the best alignment between them, using the Euclidean metric to calculate the distance between frames.

A comparison between a target and test speaker begins by scoring each test phoneme token against all target training tokens that belong to the same phoneme. For each test token, we keep only the best match with all the corresponding target tokens.

We now have a score for each token present in the test data. We expect some of these matching scores to be unsatisfactory for a variety of reasons: mislabeling of tokens, small number of comparison tokens for rare phonemes, etc. Thus, for robustness, we keep only a percentage of the best scores (75% in our current implementation). The selection is made fair by normalizing each token score by its duration length. Finally, we assign a single score to a test-target comparison by summing the unnormalized scores from the selected tokens and dividing by their total number of frames.

3.4 Normalizing the scores

The raw score between a target and test speaker is not enough to obtain good performance. It is well known that the scores need to be normalized to take into account such factors as channel and speaker variations.

The first normalization corrects for the variability of the test pieces. This is done in our GMM and LVCSR systems by subtracting from the score of each target-test piece the equivalent score that the test piece receives when compared to a background model. Since we have no such model in this algorithm, we correct for test variability by scoring the test pieces against a set of cohorts, subtracting from each target-test raw score the average cohort score for that test piece. Finally, we apply ZNORM to the resulting target scores.

4. COMPARISON OF SYSTEMS

We tested all three speaker recognition systems on the 1998 NIST Speaker Recognition Evaluation test set. The task consists of the recognition of 250 females and 250 males under a variety of training and test conditions. All data is drawn from the Switchboard-II collection of telephone conversations [6].

There are three training conditions: 1S, 2S, and 2F. The "one-session" (1S) condition consists of two minutes of speech taken from a single conversation. The "two-session" (2S) condition uses one minute from the 1S condition and an additional minute from a different conversation from the same telephone number. Finally, the "two-session-full" (2F) condition uses all available data from the same sessions, bringing up the training data for this condition to a nominal 5 minutes. The test data consists of 3-, 10- and 30-second pieces, with 2,500 pieces for each duration. Test impostors were taken from the 1997 NIST Speaker Recognition Evaluation test set. The cohort speakers necessary to carry out the SNP algorithm were also taken from the 1997 database.

At this early stage of its development, the SNP algorithm is inferior to both the GMM and LVCSR systems for each of the nine possible train/test pairs. Figure 4 shows the results for the 2F training condition tested with the 3-second pieces (upper set of curves) and the 30-second pieces (lower set of curves).

SNP, however, can be profitably used by interpolating it with the GMM system (see figure 5). We saw a comparable improvement from GMM-SNP interpolation on our 1997 development set. In contrast, we saw no benefit from interpolating GMM and LVCSR on both the 1996 and 1997 NIST Evaluation sets and a modest improvement in our 1997 development set. However, GMM-LVCSR interpolation came close to GMM-SNP interpolation on the 1998 Evaluation set. We believe that the GMM-SNP system offers state-of-the-art performance for all but the smallest test pieces.

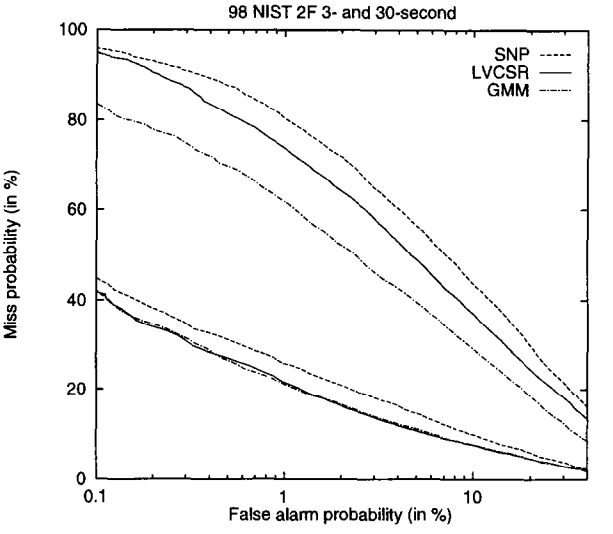

Figure 4: System comparison for the 2F training condition on 3- and 30-second test pieces (upper set: 3-second, lower set: 30-second).

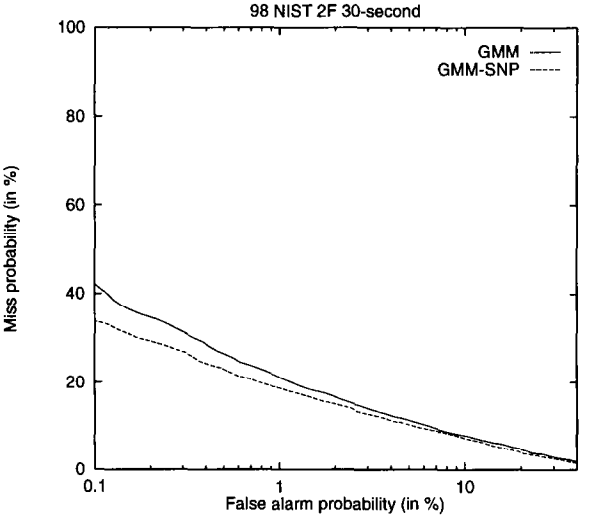

Figure 5: Interpolation of GMM and SNP for 2F/30-second.

5. FURTHER EXPLORATIONS OF SNP

Generally, non-parametric approaches have focused on individual frames and have neglected sequential information. Since the belief that this information is useful motivated us to develop SNP, we wanted to test our hypothesis.

One obvious way to test how important sequences are to the algorithm is to eliminate them. Instead of comparing sequences, we can compare individual frames. Each test frame receives a score corresponding to its distance to the closest frame in the target's training data. The resulting scores are then normalized exactly as in the SNP algorithm. This frame-by-frame scoring can be carried out in two ways to provide further insight into the workings of SNP. We can restrict the comparison of a test frame to target frames belonging to the same phoneme (same-phoneme scoring) or score a test frame against all target frames irrespective of phonemic label (all-frame scoring).

We compared these two scoring methods for the 1S training condition on the 3-, 10-, and 30-second test pieces using the female half of our 1997 NIST Speaker Recognition development set. The results show SNP lagging behind both frame scoring schemes for the smaller test pieces, but beginning to overtake them for the 30-second pieces (figure 6: upper set shows the 3-second result, middle set the 10-second result, and lower set the 30-second result).

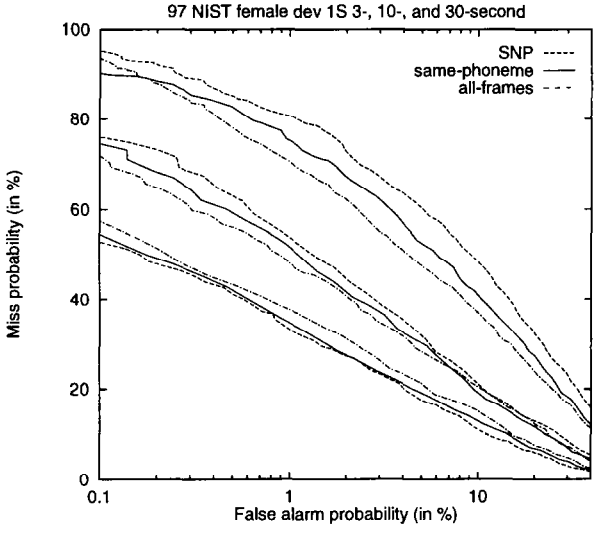

Figure 6: SNP compared to individual frame scoring methods (upper set: 3-second, middle set: 10-second, and lower set: 30-second).

One way to understand these results is to hypothesize that the advantage of sequential information is masked in the smaller test pieces because of a greater error rate in their time-alignments; i.e. segment boundaries and labels are more likely to be misleading. This is consistent with our observation on GMM vs. LVCSR performance at the end of section 2. In addition, this hypothesis would explain the difference between all-frame and same-phoneme scoring in the smaller test pieces. We ran a preliminary test of the hypothesis on the 3-second test pieces by retaining the integrity of test phoneme sequences but allowing them to match against any equal-sized sequence in the training data ("slide" scoring). The result (figure 7) already shows an improvement over SNP even though we did not use dynamic programming, which we consider to be an essential part of sequence matching. Clearly, there remains much work to do in realizing the full potential of the SNP approach.

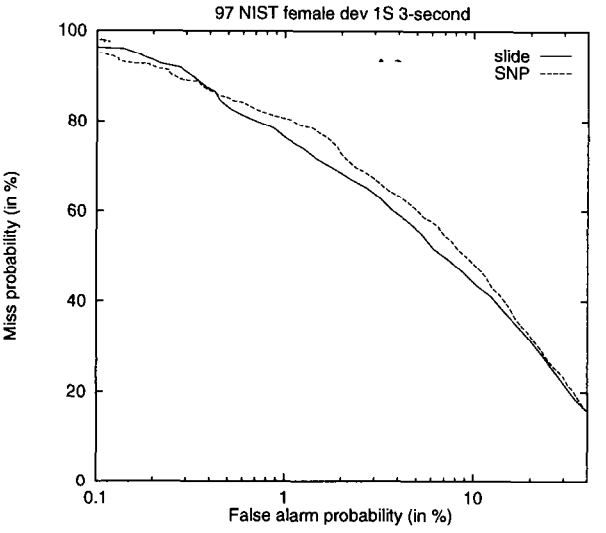

Figure 7: SNP versus "slide" scoring.

6. CONCLUSIONS

Our LVCSR system is now competitive with a GMM system when there is sufficient data. In addition, we presented a new algorithm that uses a non-parametric method to compare sequences of frames (SNP). This new system is still under development, but already allows us to outperform standard GMM systems when interpolated with them.

7. REFERENCES

1. B. Peskin et al., "Topic and Speaker Identification via Large Vocabulary Continuous Speech Recognition," ARPA Workshop on Human Language Technology, Princeton, March 1993.

2. M. Newman et al., "Speaker Verification Through Large Vocabulary Continuous Speech Recognition," Proc. ICSLP-96, Philadelphia, November 1996.

3. A. L. Higgins et al., "Voice Identification Using Nearest-Neighbor Distance Measure," Proc. ICASSP-93, Minneapolis, May 1993.

4. L. G. Bahler et al., "Improved Voice Identification Using a Nearest-Neighbor Distance Measure," Proc. ICASSP-94, Adelaide, May 1994.

5. D. Reynolds, "Comparison of Background Normalization Methods for Text-Independent Speaker Verification," Proc. Eurospeech97, Rhodes, September 1997.

6. M. A. Przybocki and A. F. Martin, "NIST Speaker Recognition Evaluation – 1997," RLA2C Workshop, Avignon, April 1998.