A Speaker Clustering Algorithm for Fast Speaker Adaptation in Continuous Speech Recognition

Luis Javier Rodríguez Fuentes and M. Inés Torres*

Pattern Recognition & Speech Technology Group
DEE. Facultad de Ciencia y Tecnología. Universidad del País Vasco
Apartado 644. 48080 Bilbao. Spain
Email: luisja@we.lc.ehu.es

Abstract. In this paper a speaker adaptation methodology is proposed, which first automatically determines a number of speaker clusters in the training material, then estimates the parameters of the corresponding models, and finally applies a fast match strategy – based on the so called *histogram models* – to choose the optimal cluster for each test utterance. The fast match strategy is critical to make this methodology useful in real applications, since carrying out several recognition passes – one for each cluster of speakers – , and then selecting the decoded string with the highest likelihood, would be too costly. Preliminary experimentation over two speech databases in Spanish reveal that both the clustering algorithm and the fast match strategy are consistent and reliable. The histogram models, though being suboptimal – they succeeded in guessing the right cluster for unseen test speakers in 85% of the cases with read speech, and in 63% of the cases with spontaneous speech – , yielded around a 6% decrease in error rate in phonetic recognition experiments.

1 Introduction

One of the most challenging issues posed by current applications of continuous speech recognition is the speaker variability. The availability of large databases with hundreds or even thousands of speakers allows to train very robust speaker-independent acoustic models. These generic models behave quite well with most speakers – those falling in the average *way of speaking* – , but may show a significant decrease in performance with some specially difficult speakers. Clearly, improved performance may result from adapting speaker-independent models to each particular speaker. Various strategies have been proposed in the literature, remarkably *speaker normalization* [1], *speaker adaptation* [2,3], and *speaker clustering* [4,5].

In some applications, like automatic dictation, only one speaker uses the system, so it seems reasonable to incrementally adapt the models to that speaker. In other applications, like information kiosks or automated ticket machines with spoken dialogue interfaces, many speakers, very different to each other, successively access the system and use it during just a short time. In these conditions it would be useless to adapt the models in an incremental way – or based on a few utterances – , because the users change very frequently. Instead,

* This work was partially supported by the University of the Basque Country, under grant 9/UPV 00224.310-13566/2001, and the Spanish MCYT, under project TIC2002-04103-C03-02.

Petr Sojka, Ivan Kopeček and Karel Pala (Eds.): TSD 2004, LNAI 3206, pp. 433–440, 2004.

the adaptation should be done on a utterance-by-utterance basis, and should be done fastly. Performing speaker clustering and training specific models for the resulting clusters allows instantaneous adaptation by selecting the most suitable set of models [6,7]. The key issue is to find a fast and reliable way of selecting the most suitable set of models for any given speech utterance, since carrying out several recognition passes – one for each cluster of speakers – , and then selecting the decoded string with the highest likelihood, would be too costly.

This work aims to automatically find a set of speaker clusters and train specific HMMs which may be either selected or combined during recognition; also, it looks for a fast and reliable way of selecting the most suitable cluster during recognition, which usually relies on smartly reducing the dimensionality of the feature space. We apply Vector Quantization (VQ) to the acoustic features and define each cluster model as a discrete probability distribution – that we call *Histogram Model* – , which is applied to the input utterances in a straightforward manner, without recognizing them.

The rest of the paper is organized as follows: section 2 describes the histogram models used to represent the speaker clusters; section 3 addresses the speaker clustering algorithm; section 4 considers different ways of selecting/composing the speaker-adapted model during recognition, along with related computational issues; experimental evaluation of the clustering algorithm and phonetic recognition results using both the histogram models and the acoustic likelihoods after recognition are presented in section 5; finally, section 6 briefly reviews the presented work.

2 The Histogram Model

Let M be the number of acoustic vectors used to represent the speech signal at each time t. Then each sample $X(t)$ consists of M vectors, $X_j(t)$ with $j = 1, \ldots, M$. First, for each acoustic representation a VQ codebook is built, using the standard LBG algorithm to minimize the average distortion in quantifying the acoustic vectors of a training corpus. Let N be the size of these codebooks. Then each vector $X_j(t)$ can be replaced by a single symbol $Y_j(t) \in \{1, \ldots, N\}$, corresponding to the index of the nearest centroid.

Now, assuming that the training corpus is partitioned into S speaker clusters, consider the cluster i, for which $c(i)$ samples are available. We store in $c(k, j, i)$ the number of times $Y_j(t) = k$ in the set of samples corresponding to the cluster i, and define the discrete distribution $P_j(k|i)$ as:

$$P_j(k|i) = \frac{c(k, j, i)}{c(i)} \quad . \tag{1}$$

This is an empirical distribution based on the histograms of the symbols at each acoustic stream. Note that for any j $\sum_{k=1}^{N} c(k, j, i) = c(i)$, so that $\sum_{k=1}^{N} P_j(k|i) = 1$. The probability of a quantified speech sample $Y(t)$ being generated by a speaker in cluster i is defined as the joint discrete distribution:

$$P(Y(t)|i) = \prod_{j=1}^{M} P_j(Y_j(t)|i) \quad . \tag{2}$$

Finally, the probability of a speech utterance $Y = \{Y(t)|t = 1, \ldots, T\}$ being generated by a speaker in cluster i is given by:

$$P(Y|i) = \prod_{t=1}^{T} P(Y(t)|i) \ . \tag{3}$$

3 The Clustering Algorithm

A top-down clustering scheme was applied – a variation on LBG [8] – , starting from a single cluster, iteratively splitting one of the clusters and readjusting the allocation of speakers, until not enough samples/speakers were available, or the average distortion decreased below a certain threshold.

Before writing the algorithm, we must give some definitions. Assuming that a histogram model has been constructed for each speaker l – based on the set of quantified samples corresponding to that speaker, $\Upsilon(l) = \{Y(t)|t = 1, \ldots, s(l)\}$, $s(l)$ being the number of samples – , the distance from l to a given cluster i, $d(l; i)$, is defined as follows:

$$d(l; i) = -\log\left\{\frac{P(\Upsilon(l)|i)}{P(\Upsilon(l)|l)}\right\} \ , \tag{4}$$

where $P(\Upsilon(l)|\cdot)$ is computed as the joint probability of all the quantified speech samples corresponding to the speaker l, given a histogram model (equation 3). Note that $d(l; m) \neq d(m; l)$. So, to verify the conmutative property, the distance between any pair of speakers l and m is given by the following expression:

$$\begin{aligned} D(l, m) &= d(l; m) + d(m; l) \\ &= -\log\left\{\frac{P(\Upsilon(l)|m)P(\Upsilon(m)|l)}{P(\Upsilon(l)|l)P(\Upsilon(m)|m)}\right\} \ . \end{aligned} \tag{5}$$

Given a cluster i, the speaker centroid $l^{(i)}$ is defined as that for which the average distance to other speakers in that cluster is minimum:

$$l^{(i)} = \arg\min_{l}\{\bar{D}(l|i)\} \tag{6}$$

$$\bar{D}(l|i) = \frac{1}{L(i) - 1}\sum_{m \in i} D(l, m) \ , \tag{7}$$

where $L(i)$ is the number of speakers in the cluster i. Finally, to stop the splitting process, a criterion based on the decrease of the clustering distortion must be defined. Assuming that each speaker l was assigned to a cluster $i_n^{(l)}$ at the iteration n of the clustering algorithm, then the *average distortion* is defined as:

$$R(n) = -\log\left[\prod_{l=1}^{L}\frac{P(\Upsilon(l)|i_n^{(l)})}{P(\Upsilon(l)|l)}\right]^{1/C} \ , \tag{8}$$

where $C = \sum_{l=1}^{L} s(l)$ is the number of samples in the training corpus. The clustering algorithm is described in detail in the following paragraphs:

1. For each speaker $l \in \{1, \ldots, L\}$ and for each acoustic stream $j \in \{1, \ldots, M\}$, the *speaker histograms* $s(k, j, l)$ are counted, and the normalizing factor $s(l) = \sum_{k=1}^{N} s(k, 1, l)$ computed.
2. Compute and store the distance between any pair of speakers l and m, $D(l, m)$. Note that only $L(L-1)/2$ values must be computed, since $D(l, l) = 0$ and $D(l, m) = D(m, l)$.
3. Initially ($n = 0$), a single cluster is defined ($S = 1$) including all the speakers: $\forall l,\ i_0^{(l)} = 1$. The clustering distortion $R(0)$ is computed. Also, for each acoustic representation $j \in \{1, \ldots, M\}$ the histogram model of the initial cluster is computed as follows: $c(k, j, 1) = \sum_{l=1}^{L} s(k, j, l)$ and $c(1) = \sum_{l=1}^{L} s(l)$.
4. **repeat**
   4.1 $n \leftarrow n + 1$
   4.2 For each cluster $g \in \{1, \ldots, S\}$, obtain the centroid speaker $l^{(g)}$, the average distance from any speaker in the cluster to the centroid, $\bar{D}(l^{(g)}|g)$, and the nearest speaker to the centroid, $m^{(g)}$. Add this information to a list of *cluster split candidates*, c_{cand}, in descending order of $\bar{D}(g)$.
   4.3 **while** $c_{cand} \neq \emptyset$ **do**
      4.3.1 Extract the first item of the list: $(g, l^{(g)}, m^{(g)})$, and split in two the cluster g, taking as histogram models of the new clusters those of $l^{(g)}$ and $m^{(g)}$, respectively.
      4.3.2 **repeat**
         - For each speaker l, assign it to the nearest cluster, i.e. that for which $d(l; i)$ is minimum.
         - For each cluster i, recompute the histogram model using the counts $s(k, j, l)$ and $s(l)$ of the speakers assigned to it.

      until maximum number of iterations **or** speaker clusters unchanged
      4.3.3 **if** partition is valid **then**
      { $S \leftarrow S + 1$; Compute $R(n)$; Empty c_{cand}; }
      else
      { Recover the cluster partition at $n - 1$; $R(n) \leftarrow R(n-1)$; }

   until $(R(n-1) - R(n))/R(n) < \tau$
5. Store the speaker cluster partition and the corresponding histogram models.

In the above algorithm $\tau > 0$ is an empirical threshold for the relative decrease in the average distortion. Also, each time a candidate partition is generated, all the clusters must contain a minimum number of speakers and samples to guarantee the trainability of the acoustic models. As noted in step 4.3.3, when not enough speakers or samples are available for any of the clusters, the previous partition is recovered and another splitting explored. The candidate splittings are explored in descending order of $\bar{D}(g)$, so that the cluster with the highest distortion is split in first place.

4 Speaker Adaptation Alternatives

Once the training material is grouped into, say, S speaker clusters, acoustic models must be trained for each cluster. We accomplished this by applying the well known MAP re-estimation procedure [2], starting from speaker independent models and heuristically tuning

the adaptation learning rate. As usual when dealing with Continuous-Density HMMs, only the Gaussian means and weights were re-estimated. During recognition we considered four possible ways of selecting/composing the HMM set for an input utterance:

Maximum likelihood. The most expensive approach – which we consider here as a reference – consists of carrying out S recognition passes, one for each HMM set, and selecting that yielding the highest likelihood. This multiplies by S the computational cost of the baseline speaker-independent approach.

Maximum histogram probability. A second approach consists of applying the histogram models to the input utterance and selecting the cluster that yields the highest probability. Then a single recognition pass is run using the HMM set corresponding to the selected cluster.

Beam of histogram probabilities. The third approach is a variation on the previous one. It consists of selecting not only that cluster yielding the highest histogram probability, but also those whose histogram probabilities are higher than, say, 70% the maximum value, then carry out recognition passes for them and select the decoded string that yields the highest likelihood. This is a sort of beam selection, motivated by the fact that sometimes the *true* cluster yields histogram probabilities near but below the maximum. This approach will require more than one recognition pass on average – typically between 2 and 3 – , but the recognition performance might reach that of the *true* likelihoods.

Weighted combination of HMMs. The fourth approach consists of composing the speaker-dependent HMM as a linear combination of the cluster HMMs, as other authors have previously done [4,5]. For a given speech utterance Y, the weight of each cluster i is computed in a straightforward way, based on the histogram probabilities, as follows:

$$w_i = \frac{P(Y|i)}{\sum_{g=1}^{S} P(Y|g)} \quad . \tag{9}$$

As in the speaker-independent case, a single recognition pass is run in this approach, but S times more parameters will be used in the computation of the observation probabilities. So computational costs will be close to those of the approach based on likelihoods.

5 Experimental Results

5.1 Databases

A read speech database in Spanish, called SENGLAR – phonetically and gender-balanced, acquired at 16 kHz in laboratory conditions – , was used in first place to tune the clustering algorithm. The training corpus consisted of 1529 utterances, pronounced by 57 speakers and including 60399 phone samples, with a total duration of around 80 minutes. The test corpus consisted of 700 utterances, pronounced by 33 speakers, and included 32034 phones, with a total duration of around 40 minutes.

A spontaneous speech task-specific database in Spanish, called INFOTREN – composed of human-computer spoken dialogues, acquired at 8 kHz across telephone lines in office environment – was used in second place, to test the proposed methodology in a real-life application. The training corpus consisted of 1349 utterances, pronounced by 63 speakers and including 62729 phone samples, with a total duration of around 117 minutes. The test

corpus consisted of 308 utterances, pronounced by 12 speakers, and included 13683 phones, with a total duration of around 30 minutes.

5.2 Conditions

The mel-scale cepstral coefficients (MFCC) and energy (E) – computed in frames of 25 milliseconds, taken each 10 milliseconds – were used as acoustic features. The first and second derivatives of the MFCCs and the first derivatives of E were also computed. Four acoustic streams were defined: MFCC, ΔMFCC, Δ^2MFCC and (E,ΔE). Vector quantization (LBG, [8]) was applied to get four codebooks, each one consisting of 256 centroids.

In the case of SENGLAR the set of sublexical units consisted of 23 context-independent phones (CIP) plus one extra unit for *silences*. In the case of INFOTREN, besides the 23 CIP, 14 extra units were defined to model spontaneous speech events like noises, lengthenings, filled pauses, silent pauses, etc. Each sublexical unit was represented with a left-right Continuous-Density HMM consisting of three states with self-loops but no skips. No phonological restrictions were applied. After recognition, the extra units were either filtered or mapped into the 23 CIP set, for both the recognized and the correct strings, and finally the phonetic recognition rate was computed.

5.3 Results of Speaker Clustering

The clustering algorithm was run using the training corpora of the two databases described above. At least 2 speakers and 30000 speech frames (5 minutes) were required for each cluster to be valid. The maximum number of convergence iterations (step 4.3.2) was set to 20, and the threshold for the relative decrease in the average distortion was fixed to $\tau = 0.01$. This resulted in 5 speaker clusters for SENGLAR and 8 speaker clusters for INFOTREN. Most clusters were gender-specific, i.e. formed almost exclusively either by male or by female speakers, which means that speaker characteristics were effectively working to separate clusters each other.

On the other hand, Continuous-Density HMMs were trained for each cluster, and the training corpus recognized with them. It was found that the HMM set corresponding to the *right* cluster yielded the best likelihood in 99.6% of the cases for SENGLAR, and in 94.4% of the cases for INFOTREN. Using the histogram models to select the most suitable set of HMMs – instead of the *true* likelihoods – , the *right* models were selected in 94.5% of the cases for SENGLAR, and in 75.3% of the cases for INFOTREN. This fall in performance for INFOTREN may be explained by the intrinsic lack of acoustic information due to a lower sampling rate (8 kHz) and to the background/channel noise, which increases acoustic variability. However, the clustering algorithm still produced very consistent speaker groups.

Finally, when dealing with speech data from unseen speakers, as those included in the test corpora, though the decisions about the best cluster were not homogeneous, histogram probabilities led to the same decision than the *true* likelihoods in 84.7% of the cases for SENGLAR, and in 63.0% of the cases for INFOTREN. Since test speakers did not participate in the clustering process, they were not clearly classified in one of the clusters. More often two or three clusters appeared as candidates.

5.4 Phonetic Recognition Results

Phonetic recognition experiments were carried out, using MAP-adjusted Continuous-Density HMMs and applying the adaptation alternatives described in Section 4. Recognition rates, as well as the average number of recognition passes and the CPU time – relative to the speaker-independent case – are shown in Table 1.

Table 1. Phonetic recognition rates using speaker-independent and speaker-adapted CDHMMs for the speech databases SENGLAR and INFOTREN. The average number of recognition passes (#REC) and the CPU time relative to the speaker-independent case are shown too.

	%PhREC (#REC,CPU)	
	SENGLAR	**INFOTREN**
Speaker-independent	72.72 (1,1.00)	61.34 (1,1.00)
Max-likelihood	74.41 (5,5.00)	63.61 (8,8.00)
Max-hprob	73.04 (1,0.92)	63.13 (1,1.01)
Beam-hprob (70%)	**74.41 (2.08,2.16)**	**63.60 (3.37,3.33)**
Weighted-hprob	73.68 (1,4.90)	62.14 (1,7.63)

All the speaker adaptation alternatives based on the clustering algorithm proposed in this paper outperformed the baseline speaker-independent approach. The adaptation approach based on the *true* likelihoods yielded around a 6% decrease in phonetic error rate, but CPU times were multiplied by 5 and 8 for SENGLAR and INFOTREN, respectively. The approach based on the histogram probabilities slightly improved the performance in the case of SENGLAR, but showed a much better behaviour in the case of INFOTREN, with a 4.6% decrease in error rate. Note again that this approach did not increase the computational costs. The approach which selected a beam of clusters – those whose histogram probabilities were higher than 70% the maximum – was a good compromise between performance and computational cost, since it yielded the same performance than likelihoods with only two o three recognition passes on average. Finally, the approach based on a weighted combination of the cluster HMMs did not improve the performance of the beam approach, and needed almost as much CPU time as the apprroach based on likelihoods.

6 Conclusion

This paper presents a new speaker clustering algorithm, which uses a discrete distribution of VQ labels in various acoustic streams as speaker/cluster model – the so called histogram model. Also, various speaker adaptation schemes are described based on Continuous-Density HHMs and histogram models, specifically obtained for a set of speaker clusters. Results of clustering are presented for two speech databases in Spanish with around 60 training speakers. Phonetic recognition results reveal that a 6% decrease in error rate can be attained at the expense of two or three times the computational cost of the speaker-independent baseline approach. More remarkable improvements should be expected when applying this methodology to a larger database, with hundreds or even thousands of speakers.

References

1. Lee, L., Rose, R.: A frequency warping approach to speaker normalization. IEEE Transactions on Speech and Audio Processing **6** (1998) 49–60.
2. Gauvain, J., Lee, C.: Maximum A Posteriori Estimation for Multivariate Gaussian Mixture Observations of Markov Chains. IEEE Transactions on Speech and Audio Processing **2** (1994) 291–298.
3. Leggetter, C., Woodland, P.: Maximum Likelihood Linear Regression for Speaker Adaptation of Continuous Density Hidden Markov Models. Computer, Speech and Language **9** (1995) 171–185.
4. Gales, M.: Cluster Adaptive Training of Hidden Markov Models. IEEE Transactions on Speech and Audio Processing **8** (2000).
5. Kuhn, R., Junqua, J., Nguyen, P., Niedzielski, N.: Rapid Speaker Adaptation in Eigenvoice Space. IEEE Transactions on Speech and Audio Processing **8** (2000) 695–707.
6. Faltlhauser, R., Ruske, G.: Robust Speaker Clustering in Eigenspace. In: Proceedings of the IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU), Madonna di Campiglio (Italy) (2001) CD-ROM, paper n. 86.
7. Naito, M., Deng, L., Sagisaka, Y.: Speaker clustering for speech recognition using vocal tract parameters. Speech Communication **36** (2002) 305–315.
8. Linde, Y., Buzo, A., Gray, R.: An algorithm for vector quantizer design. IEEE Transactions on Communications **28** (1980) 84–95.