State Clustering Improvements for Continuous HMMs in a Spanish Large Vocabulary Recognition System

R. Córdoba, J. Macías-Guarasa, J. Ferreiros, J.M. Montero, J.M. Pardo

Grupo de Tecnología del Habla. Departamento de Ingeniería Electrónica. Universidad Politécnica de Madrid
E.T.S.I. Telecomunicación. Ciudad Universitaria s/n, 28040 Madrid, Spain
cordoba@die.upm.es
http://www-gth.die.upm.es

ABSTRACT

In this paper we present a whole set of improvements that have been applied to a large vocabulary isolated-word recognition system using continuous models. This system has been used in the EU funded IDAS project (LE4-8315), where an automated interactive telephone-based directory assistance service has been developed.

We cover both improvements in the techniques for continuous HMM reestimation and agglomerative clustering for context-dependent models, all of them applied to our database in Spanish. Specifically, we will show how a new distance between states can greatly improve the performance of the clustering process. We show a new strategy for the clustering itself based in multiple Gaussian clustering which improved the results too. And finally, we present a new way to find the optimum number of Gaussians for each state that can be applied to both context dependent and context independent models.

Keywords: large vocabulary recognition, telephone-based, continuous HMMs, agglomerative clustering.

1. INTRODUCTION

In the IDAS project [4], we address the challenging problem of automating the provision of directory assistance services to the public over the telephone network.

The speech recognizer module plays a decisive role in the overall performance of the system. There are two main difficulties for this module. The first one is the noise and the reduced signal to noise ratio that is common in a telephone line. The second difficulty is the high degree of confusability that arises when you consider 10,000 surnames in Spanish, because we need their exact transcription.

In this paper, we will describe a series of improvements that have been applied to a large vocabulary isolated-word recognition system using context-dependent continuous models. When using context-dependent models, a key problem is the need to balance the desired model detail with the amount of training data. The total number of states is too high, there are too many parameters to be estimated, and so we need different techniques to reduce the number of states. There are two main methods to do this: using decision trees [5][6] or making an agglomerative clustering [7].

In this paper, we present an agglomerative algorithm to cluster and tie acoustically similar states. As this technique is well known, we will focus the paper in the most critical aspects or techniques that improved the algorithm over the standard procedures.

2. RECOGNITION MODULE

2.1 Database

We have used the SpeechDat database, the isolated speech part, with 1,000 speakers who utter the following items: application words, isolated digits, cities, companies, names, surnames, and spelled names. We have divided this database into two parts: 9,069 words for training and 3,840 for recognition. The size of the dictionary for all experiments in sections 3 and 4 is 1,000 words.

2.2 System architecture

As real-time is a must in the system, we decided to use a hypothesis-verification approach to reduce the number of candidates that have to be considered with detailed modeling in our Large Vocabulary recognition system.

- In the hypothesis step, we use a fast preselection module [3]: with context-independent semi-continuous HMMs (CI-SC), we obtain a sequence of recognized phones which is followed by a lexical access module. This module passes N-best candidates to the verification step.
- In the verification step, using context-dependent semi-continuous HMMs (CD-SC) or continuous HMMs (CD-C), and whole word recognition we obtain the recognized word.

In this paper, we will concentrate in the verification step. We had previously used CD-SC with success [1] [2], so we concentrated our efforts in the CD-C, which we had used before for another task in [1]. We considered first context-independent continuous HMMs (CI-C) for the initial fine-tuning of the system.

3. CONTEXT-INDEPENDENT CONTINUOUS MODELS

We considered the CI system as a baseline for our development. Using it, we could check the suitability of decisions about the reestimation and the general topology of the system very quickly. The objective is to obtain the best possible multiple Gaussian monophone models. Although some of the aspects mentioned here are well known, they are enumerated to clarify many problems that are usually faced by newcomers to this modeling. In any case, they are not usually mentioned in the literature but, as we will see, the performance can increase if they are applied correctly. In the following sections we will see the main areas of work in the CI system.

3.1 Set of allophones

Although rarely mentioned, it can be a key factor of the performance. We considered three different sets of allophones for Spanish:

- 23 units: it is the basic set of allophones for Spanish
- 51 units: it is a detailed set of allophones, where we differentiate between stressed/unstressed/nasalized vowels, we include different variants for the vibrant 'r' in Spanish, different units for the diphthongs, the fricative version of 'b', 'd', 'g', and the affricates version of 'y' (like 'a*y*er' and 'cón*y*uge')
- 45 units: detailed set, suppressing six units with a very low number of appearances in the training database (one of the variants for the vibrant 'r' and the 'stressed nasal vowels' which are joined to the 'unstressed nasal vowels')

We obtained a **12%** improvement with 45 units over the set of 23 and only a minor improvement over the set of 51.

3.2 Increasing the number of Gaussians

We began with single Gaussian monophone models and wanted to find an optimum way to increase the number of Gaussians in each state of the model.

We compared two approaches:

A. Split the single Gaussian in multiple Gaussians in a single step, and then reestimate the models.

B. Follow a stepwise approach, splitting the Gaussians one by one, and perform 4 reestimations in each step.

This last method obtained an improvement of **6.2%** in our system, so it is the approach we have followed.

Another question that has to be answered is which Gaussian should be split (which is the "largest" one?). We have considered the following possibilities for the largest Gaussian:

A. More vectors assigned to it during reestimation.

B. Highest average distance between its vectors.

C. Highest global distance between its vectors.

Again, using the last option we got an improvement of **6.7%** over A, and **7%** over B.

A third point of improvement we detected in this process is to apply the k-means algorithm after the splitting of the largest Gaussian and before making the reestimation. This way, the average vectors have much better initial values, reducing the possibility of what is called "defunct mixtures", and fewer iterations are needed.

The improvement we obtained using this approach was **9%** in average.

4. CONTEXT-DEPENDENT CONTINUOUS MODELS

We have followed an agglomerative clustering approach at the state level, as in [1]. One restriction we apply is that we only cluster states with the same central allophone.

The clustering process has the following steps:

1. Bootstrap: We train single Gaussian context-dependent models. We have 21,543 different states in our system and 1 cluster for each state.
2. We first cluster all states with a very low number of vectors available in training (less than 30).
3. Find clusters x and y for which distance(x,y) is minimum.
4. While distance(x,y) < threshold and number of clusters > minimum
   a. Merge clusters x and y
   b. Compute the distance between the new cluster and all the other clusters with the same central allophone.
   c. Find clusters x and y for which distance(x,y) is minimum.

In the following sections we will see the new techniques considered for this process.

4.1 Best "distance between states"

As we want to cluster the acoustically most similar models, the measure of similarity between states is a critical factor. We need to define a distance that helps us to determine which are the most similar states.

One classical solution is the one adopted in [7], which is related to the square root of the divergence between the two Gaussian pdfs:

$$D(i,j)=\left(\frac{1}{codes}\right)\sum_{s=1}^{codes}\left[\left(\frac{1}{param}\right)*\sum_{K=1}^{param}\left(\frac{(\mu_{isk}-\mu_{jsk})^2}{\sigma_{isk}*\sigma_{jsk}}\right)\right]^{1/2}$$

We made the clustering described above with single Gaussian models and the results we got were low: **15%** error rate in a system with 3,600 states and did not improve as the clustering progressed. So, we decided to look for new alternatives.

From our previous experiments in agglomerative clustering [1], we knew the importance of including the number of vectors available in training for each state in the distance measure. So, we decided to weigh the distance by the number of vectors available in training for each state to favor the clustering of states with little training data:

$$D'(i,j)=\sqrt{\left(\frac{n_i*n_j}{n_i+n_j}\right)}\left(\frac{1}{codes}\right)\sum_{s=1}^{codes}\left[\left(\frac{1}{param}\right)*\sum_{K=1}^{param}\left(\frac{(\mu_{isk}-\mu_{jsk})^2}{\sigma_{isk}*\sigma_{jsk}}\right)\right]^{1/2}$$

where n_i, n_j are the number of vectors for state i and j.

In the best experiment we got a **9.56%** error rate, which is a significant improvement over the previous distance.

When we made the agglomerative clustering with discrete HMM models, we found that a distance based on the minimum loss of information had a very good performance [1]. So, we decided to use a new distance, which is based in the minimum loss of information produced by the clustering,

weighed by the number of vectors assigned to the states in the training. This is the expression of this distance in the case of single Gaussian clustering:

$$D''(i,j) = (n_i + n_j) * \sum_{k=1}^{d} \ln(\sigma_k) - n_i * \sum_{k=1}^{d} \ln(\sigma i_k) - n_j * \sum_{k=1}^{d} \ln(\sigma j_k)$$

where n_i, n_j are the number of vectors for state i and j, σ_k is the standard deviation of the distribution obtained after the clustering, σi_k, σj_k are the standard deviations of the original distributions *i* and *j*, and *d* is the dimension of the vector of parameters.

This distance has produced better results: **8.32%** error rate with a similar number of total Gaussians (**13%** improvement over D'), so we decided to use it in our system.

4.2 Best strategy for the clustering

We compared two alternatives:

A) *Clustering with single Gaussian models.*

It is the classical procedure [7]. When the clustering is over, we reestimate the models increasing the number of Gaussians in each state (in a similar way to the increasing of Gaussians made in context-independent (CI) models). The error rate obtained with the best models is 8.32% (the experiment from the previous section), with 1200 states and 6 Gaussians/state.

B) *Clustering with multiple Gaussian models.*

We followed the following iterative approach:

1. Reduce the number of units: find the closest states and merge them.
2. Increase the number of Gaussians (similar to CI).
3. Reestimate the models. If 'number of units' > objective, go to step 1. If not, stop.

This approach offers the following advantages:

a) The multiple Gaussian models will be more robust, so our decisions during the clustering process will be based in more robust estimates.

b) The intermediate reestimation allows us to optimize the models in successive steps.

We had to change the expression of the distance to account for the multiple Gaussian components for each state:

$$D' = \sum_{l=1}^{N} n_l * \sum_{k=1}^{d} \ln\sigma_{lk} - \sum_{l=1}^{Ni} ni_l * \sum_{k=1}^{d} \ln\sigma i_{lk} - \sum_{l=1}^{Nj} nj_l * \sum_{k=1}^{d} \ln\sigma j_{lk}$$

where N is the number of Gaussians in the distribution obtained after the clustering, Ni and Nj are the number of Gaussians in the two original distributions. n_l, ni_l and nj_l reflect the number of vectors assigned to each Gaussian in each distribution. σ refer to standard deviations of the distribution obtained after the clustering, and σi, σj refer to standard deviations of the original distributions *i* and *j*.

This distance can now have positive values when the number of Gaussians obtained after the clustering is equal to the number of original Gaussians. The iterative approach has followed the steps described in Table 1. Steps 1, 3, 5, 7, 9, and 11 are clustering steps; steps 2, 4, 6, 8, 10, and 12 are reestimation steps with an increase in the number of Gaussians.

Step	Total number of clusters	Maximum number of Gaussians/state
0	21,543	1
1-2	7,000	$1 \rightarrow 2$
3-4	3,600	$2 \rightarrow 3$
5-6	2,400	$3 \rightarrow 4$
7-8	1,800	$4 \rightarrow 5$
9-10	1,440	$5 \rightarrow 6$
11-12	1,200	$6 \rightarrow 7$

Table 1. Steps followed in the multiple Gaussian clustering

In this process, we obtain different sets of models with a similar performance: from step 8 until 12 the results are very similar.

The best error rates obtained with this approach can be seen in Table 2 (both with a similar number of Gaussians, 7,200).

Set	Single Gaussian clustering	Multiple Gaussian clustering	Impro-vement
Test set	8.32	7.70	7%
Training set	2.40	1.97	18%

Table 2. Comparison between single Gaussian and multiple Gaussian clustering

These results show that the improvement for the recognition set is relatively low, but there is a great adaptation to the training data (18% improvement with the same number of parameters). So, we can expect a more significant improvement when we increase the size of the database, as there is a great difference in performance for both sets.

In any case, both techniques show a great adaptation to the training data.

4.3 Optimum number of Gaussians/state

In the process of increasing the number of Gaussians it is difficult to know which is the best Gaussian to be split and if it should be split at all. We have compared three alternatives:

A) Use the same number of Gaussians for all states.

B) Number of Gaussians proportional to the number of vectors assigned to the state in the training. This way, more detailed models are used when there is enough training data.

C) An innovative approach: we split the Gaussians that provide the largest reduction in entropy (or largest increase in information). So we adapt the process of increasing the Gaussians to the training data.

The formula we have used for the reduction in entropy (option C) is:

$$G = \sum_{j=1}^{NF} ni_j * \sum_{i=1}^{d} \ln \sigma f_{ji} - \sum_{j=1}^{NI} ni_j * \sum_{i=1}^{d} \ln \sigma i_{ji}$$

where NF is the final number of Gaussians in the state, NI is the initial number of Gaussians, ni and nf are the weights of each Gaussian, σi and σf are the standard deviations of each Gaussian and *d* is the dimension of the vector of parameters.

G measures the increase in information that we obtain when we increase the number of Gaussians in the state. G can not be negative, because when we add a new Gaussian we (almost) always obtain more information.

G is then multiplied by a factor, proportional to the number of vectors of the state divided by the number of Gaussians to favor the election of states with more data available.

In the global process we increase the number of Gaussians of the states which give the highest value of G (50% of the states in each step) and reestimate the models. We stop when we get to the desired number of total Gaussians or when we get an optimum in recognition rate.

This technique can be applied to both context-dependent and context-independent systems.

4.3.1 Results for the CD system

We used the result of the clustering obtained with single Gaussian models and then increased the number of Gaussians, as we wanted to analyze only the increase of Gaussians, not the clustering process itself.

We can see the results for the three options considered using some 7,200 Gaussians for all of them in Table 3.

	Option A	Option B	Option C
Test set	8.32	7.41	7.52
Training set	2.40	1.84	1.33

Table 3. Error rates for the 3 options considered (CD-models)

We can see that applying option B, with a variable number of Gaussians/state between 2 and 18, the improvement was 11% over option A.

With option C, the improvement was 9.6% over option A, but with a decrease of 1.5% over option B. This last result was a bit disappointing, although not significant. Nevertheless, we considered that with CD models we could have insufficient data to demonstrate that option C was better than B. This effect can be seen in the results for the training set, where option C is 27.7% better than option B.

So, we decided to compare both techniques with CI models, where the amount of training data available is more adequate in relation to the number of parameters estimated.

4.3.2 Results for the CI system

In Table 4 we can see the results of these techniques for CI models, with an average of 9 Gaussians/state and a total of 1270 Gaussians approximately in all of them.

	Option A	Option B	Option C
Test set	13.27	12.69	12.26

Table 4. Error rates for the 3 options considered (CI-models)

This time, option C is better than both option A (8.2% improvement) and option B (3.5% improvement), which demonstrates the effectiveness of option C when we have enough data available.

In any case, we are now preparing a larger database to repeat the test for context-dependent models.

5. FINAL RESULTS

We applied our best systems to the final vocabulary used in the IDAS system (10,000 words) and we obtained the following WER for each system:

A. Context-dependent Continuous (CD-C): 23.1%.

B. Context-dependent Semicontinuous (CD-SC): 25.4%.

C. Context-independent Continuous (CI-C): 34.7%.

We can see that the best results correspond to the CD-C system. We are also confident that results will improve significantly in this system as we increase the size of the database. The reason for this affirmation is that the recognition results for the training set for the CD-C system show an improvement of 44% over the CD-SC, which means that the modeling is very adapted to the training set and there is place to improvements with a bigger database (our database can be considered small).

Another aspect that should be remarked is that results could be better as there are some mistakes in the database that have not been excluded from the test.

6. CONCLUSIONS

We have presented new techniques for continuous HMMs and agglomerative clustering with very good results. The new distance improves significantly the performance of the system. The strategy of clustering with multiple Gaussian models provides good results, specially for the training set, which indicates that there is a great adaptation to the training set and that the results can improve when a bigger database is used. We have compared three ways of assigning the optimum number of Gaussians/state, with very good results for the entropy-based one, specially for context-independent models.

7. REFERENCES

[1] R. Córdoba, J.M. Pardo. “Different strategies for distribution clustering using discrete, semicontinuous and continuous HMMs in CSR”, ICSLP 96, pp. 1097-1100.

[2] R. Córdoba, X. Menéndez-Pidal, J. Macías, A. Gallardo, J.M. Pardo. “Development and improvement of a real-time ASR system for isolated digits in Spanish over the telephone line”. Eurospeech 95. Vol. II, pp. 1537-1540.

[3] J.Ferreiros, J. Macías-Guarasa, et al. “Recent Work on Preselection Module for Flexible Large Vocabulary Speech Recognition System in Telephone Enviroment”, ICSLP 98, pp. 1689-1692.

[4] R. Córdoba, et al. “An Interactive Directory Assistance Service for Spanish with Large-Vocabulary Recognition”, Eurospeech 01, pp. 1279-1282.

[5] M.Y. Hwang, F. Alleva, X.D. Huang, “Senones, multi-pass search and unified stochastic modeling in Sphinx-II”, Eurospeech 93, pp. 2143-2146.

[6] P.C. Woodland, J.J. Odell, V. Valtchev, S.J. Young, “Large vocabulary CSR using HTK”, ICASSP’94, pp. II-125-128.

[7] S.J.Young, P.C. Woodland, “State clustering in hidden Markov model-based continuous speech recognition”. Computer Speech and Language 1994 vol. 8, pp. 369-383.