THE MACQUARIE SPEAKER DIARISATION SYSTEM FOR RT04S

Steve Cassidy

Centre for Language Technology
Macquarie University
Sydney

ABSTRACT

The primary purpose of the Macquarie participation in RT04s was to establish a baseline diarisation system bringing together recent work done in our group. As such, the performance of the system was not expected to be competitive, but rather to help highlight the issues in the diarisation task where we could usefully concentrate our efforts.

The system combined a simple speech/silence decision, delta BIC speaker segmentation and single Gaussian speaker models to perform the diarisation task. Hierarchical clustering was used to derive between 3 and 10 speaker clusters; models based on each number of speakers were then compared using a BIC corrected likelihood measure on the data.

1. INTRODUCTION

The NIST RT04 Spring evaluation programme has provided us with an excellent motivation to bring together various threads of work on meeting room speech processing into a complete end-to-end speaker diarisation system. Our previous work has involved an investigation of the delta BIC segmentation algorithm [1, 2] in multi-party teleconference and face to face meetings and the use of directional cues from multiple microphones in this context.

While we have investigated a number of multi-microphone methods we were not able to integrate these into this system due to the lack of information about microphone placement in the various data sources used for evaluation. Consequently the system we built was a very basic one using data from a single distant microphone and using established techniques for speaker segmentation and clustering. Building this system has served to highlight the areas of this task that we might be able to do well and those where significant interesting problems remain.

This paper serves to document the system built for this evaluation task. Had more time and resources been available we might have been able to do more experimentation leading up to running the evaluation data. However, circumstances only allowed us to build a very basic system and to make some very rudimentary parameter selections using the devtest data set. There are many parts of this problem which have been solved more effectively in previously published work; many of the choices made in building this system were to allow us to have a working system in time for the evaluation. Hence we use simple Gaussian models when mixture models would be more appropriate.

2. SYSTEM DESCRIPTION

The Macquarie system consists of four components:

- Speech/Silence classifier
- Delta BIC speaker segmentation
- Speaker Clustering
- Speaker Identification

2.1. Speech/Silence Classifier

An earlier version of our system relied entirely on the delta BIC segmentation algorithm to find both speech and silence segments in the data. While this was largely effective we found that many of the 'speech' segments found would include significant portions of silence which would then bias the speaker models produced. To prevent this a simple RMS and zero crossing based speech/silence classifier was built to first find the segments of the signal corresponding to speech which would then subsequently be segmented by the delta BIC algorithm. This improved the purity of the speech segments and had the side effect of significantly speeding up the system since only smaller speech segments had to be dealt with by the more complex segmentation algorithm.

In order to compensate for the various recording conditions found in the evaluation data the speech/silence classifier was tuned on the part of each recording *before* the target section. In this section of audio, the minimum and maximum RMS and ZCR values were recorded and a calibration factor of one tenth of the range was calculated for both RMS and ZCR. The speech silence decision was then made according to the metric:

$$M_i = \frac{r_i - r_{min}}{r_{max} - r_{min}} + 2\frac{z_i - z_{min}}{z_{max} - z_{min}} \quad (1)$$

The factor of 2 for ZCR was chosen rather than being found via an optimisation process. A threshold of 4.0 was used in the results reported here.

2.2. Delta BIC Speaker Segmentation

The speech segments output by the previous step were further segmented using the Bayesian Inference Criterion segmentation algorithm [3]. BIC is an acoustic change detection algorithm that is commonly used in speaker change detection; the assumption being that a qualitative change in the acoustic signal will often correlate with a change in speaker.

The input data was parameterised as 12 MFCC coefficients calculated every 10ms on a 25.6ms window. The sequence of MFCC vectors was processed using the BIC segmentation algorithm as follows:

1. A buffer of 200 MFCC vectors was initialised
2. Each point was evaluated as a potential cut point using the BIC criterion (leaving a 15 point margin at each end of the buffer).
3. If at least one point in the buffer had a BIC value of greater than zero, the point of maximum BIC within the buffer was added to the list of boundaries. The MFCC vectors *before* the boundary were discarded from the buffer.
4. If the buffer contained less than 500 vectors, 100 more vectors were added otherwise 50 new vectors were added. If the buffer contained more than 500 vectors long, vectors were removed from the start until only 500 remained.
5. Repeat from 2 until end of sequence.

In earlier work [1] that the delta BIC algorithm would find many false positive boundaries because it was sensitive to acoustic changes such as the onset of frication or voicing within one speaker turn. In these experiments, the BIC λ parameter was set slightly higer to 1.1 which seemed to eliminate many of these false positives. Time did not permit detailed experimentation to find an optimum value for this parameter.

It's possible to evaluation this part of the overall system independently of subsequent speaker clustering and identification stages. In our earlier paper we reported two kinds of error: **False Positive** errors are automatically detected boundaries that occur within a speaker turn, **Missed Boundary** errors are speaker turn boundaries that aren't detected by our algorithm. The first kind of errors are not too serious in this context since they just mean that a speaker turn has been too finely segmented. Missed boundary errors are more important since they represent missed acoustic changes. The following table summarises the (post-hoc) analysis of our system's performance on the evaluation data set.

	False Positive	Missed Boundary
CMU-20030109-1530	28	66
CMU-20030109-1600	23	65
ICSI-20000807-1000	27	59
ICSI-20011030-1030	19	59
LDC-20011121-1700	16	75
LDC-20011207-1800	20	57
NIST-20030623-1409	19	76
NIST-20030925-1517	28	55
Overall	**22**	**64**

Table 1. Speaker change point detection errors (%) from the evaluation data

From the raw results, it is clear that our system is finding too few change points – 1506 as opposed to 3334 real turns (6373 boundaries). Given more time, it would be useful to attempt to tune the segmentation algorithm to find a larger number of segment boundary hypotheses; as mentioned above, the λ parameter was adjusted to remove spurious breaks but this may have resulted in the removal of too many good break hypotheses.

2.3. Speaker Clustering

Once speaker change points had been generated a speaker clustering algorithm is needed to identify how many speakers are involved in the meeting. According to the rules of RT04s, the system can have no prior knowledge of speaker identity or number of speakers. In a real world application one might expect to have some indication of the context of the meeting which might provide known speaker models or an estimate of speaker numbers. However, the challenge here is to discover this from the audio data.

The standard approach to clustering is to define a distance measure between clusters and iteratively merge the most-similar clusters until the required number of clusters is reached. The two variables in this algorithm are the distance measure used and the stopping criterion.

In our system, only speech segments longer than 1.5 seconds were used for speaker clustering. The motivation here being that shorter segments are more likely to contain idiosyncratic speech and hence would be less useful in characterising speakers. Clusters were represented by single Gaussian models built from the MFCC parameters of the cluster members. In order to combine clusters a distance measure between Gaussians is needed. A useful review of a number of Gaussian distance measures is given in [4]; we

chose to use the Mahalanobis distance which has the advantage of only requiring the diagonal covariance matrix and thus simplifying the distance computation.

Clustering was performed via the `hclust` function in the R statistical package [5] which supports a number of clustering methods. We chose to use Ward's minimum variance method because it seemed to give a good distribution of speaker clusters through the data. Other methods tested tended to give one or two very large clusters which seemed inappropriate; however no thorough testing of the clustering methods performance was possible within the time-frame of the evaluation.

The `hclust` function performs a hierarchical cluster analysis and generates a tree representation of the clusters in the data. It is then possible to derive any given number of clusters by cutting the tree at some depth below the root. Our approach to determining the optimal number of clusters for the data was to generate cluster sets with between 3 and 10 clusters from the hierarchical clustering and then test each one for goodness of fit to the data.

Testing a set of cluster models against the data is straightforward since we can calculate the overall probability of the data given the set of Gaussian models. However these probabilities need to be corrected for the complexity of the models; this can be done via the Bayesian Inference Criterion (BIC) which provides a correction factor according to the difference in the number of parameters between the models being compared [3, 6]. In this manner the corrected likelihoods of the different numbers of speaker models can be compared and the best fit selected.

Using the BIC correction factor the decision as to which number of cluster models to select is based on:

$$argmax_i[\log(p(D|\theta_i)) - \frac{1}{2}\lambda K(1+K)\log(N)] \quad (2)$$

here θ_i is the set of cluster models for i clusters, K_i is the dimensionality of the input data, N is the length of the observation sequence and λ is a *fudge factor* which is ideally set to one but which can be adjusted to optimise the algorithm for a particular problem. In our systems λ was set to 1.2, a value that was derived by trial and error adjustments on trial runs with the devtest data set. Figure 2.3 shows the log probability values for the CMU_20030109-1530 recording for which the correct number of speakers is 4. It can be seen that our algorithm will choose 10 clusters in this case as the maximum corrected likelihood.

Varying the λ parameter changes the shape of the curve emphasising smaller numbers of clusters and so for most curves it is possible to find a value of λ which will give the correct answer. The trick is to find a value which gives a good answer most of the time. It is fairly clear from our results that we did not find this value since in all cases our system chose 10 clusters as the best answer.

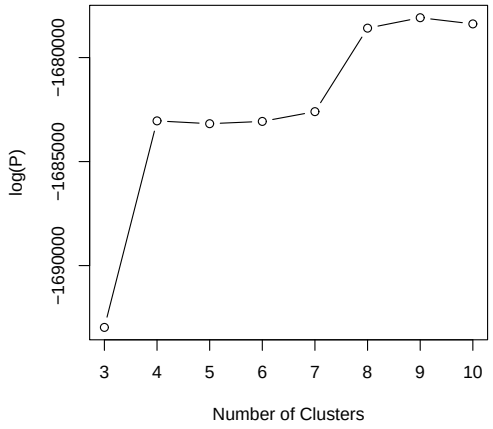

Fig. 1. Corrected likelihoods for different numbers of speaker clusters in CMU_20030109-1530

2.4. Speaker Identification

Once speaker clustering had been performed we have a set of speaker models derived from the segments in each cluster. As mentioned above only simple Gaussian models were used in our system. These models were then used to classify all of the speech segments in the audio data – recall that only those longer than 1.5 seconds were included in the clustering step. The resulting labelling was then output as the final result of our system.

3. RESULTS

	No Overlap	Overlap
Missed Speaker Time	2.7	21.3
FAlarm Speaker Time	2.8	1.8
Speaker Error Time	56.5	45.9
Overall Error	62.0	69.1
Weighted Error	78.3	106.9

Table 2. Summary of Results, percentages

Not surprisingly, the overall results from our system are quite poor. The main source of error derives from getting the number of clusters wrong; this gives rise to the large Speaker Error Time since we always chose nine speakers there was a high likelihood that a speaker would be misslabelled. This is confounded by the simple Gaussian models which are not strong enough to properly characterise speakers and hence will cause clustering errors and speaker identification errors. We are heartened by the low error score for

missed and false alarm speech which mainly seems to indicate that our speech/silence detection strategy is working well.

4. CONCLUSION

This has been an incredibly useful exercise in building our work from a collection of isolated experiments with meeting room speech to an end-to-end system capable of speaker diarisation. It was very clear all the way through the process that the compromises we were making to build the system in time for the evaluation would mean that performance would suffer. However now that it exists, we can return to the system and experiment with published methods and new techniques to improve the overall performance.

5. ACKNOWLEDGEMENTS

Many thanks to the NIST team for supplying the devtest data to us at short notice to enable us to participate in the evaluation. A significant part of the coding for this project was done by Max Whittman, an undergraduate student at Macquarie supported by a Computing Department Summer Vacation Scholarship.

6. REFERENCES

[1] Steve Cassidy and Catherine Watson, "Detecting backchannel intrusions in multi-party teleconferences," in *Proceedings of the 9th Australian International Speech Science and Technology Conference*, Melbourne, 2002, Australian Speech Science and Technology Association.

[2] Catherine Watson and Steve Cassidy, "Speaker change detection in multi-party meetings," in *Proceedings of the Eighth Western Pacific Acoustics Conference*, Melbourne, April 2003.

[3] S. Chen and P. Gopalakrishnan, "Speaker, environment and channel change detection and clustering via the bayesian information criterion," in *Proceedings of Broadcast News Transcription and Understanding Workshop*, Feb 1998.

[4] O. Pietquin, L. Couvreur, and P. Couvreur, "Applied clustering for automatic speaker-based segmentation of audio material," in *Belgian Journal of Operations Research, Statistics and Computer Science (JORBEL) Special Issue : OR and Statistics in the Universities of Mons*, 2001, vol. 41, `http://tcts.fpms.ac.be/publications/regpapers/2002/jorbel2002_oplcpc.pdf`.

[5] R Development Core Team, *R: A language and environment for statistical computing*, R Foundation for Statistical Computing, Vienna, Austria, 2003, ISBN 3-900051-00-3.

[6] Jitendra Ajmera and Charles Wooters, "A robust speaker clustering algorithm," Tech. Rep. 38, IDIAP, 2003, `ftp://ftp.idiap.ch/pub/reports/2003/rr03-38.pdf` To appear in IEEE ASRU 2003.