Using Genetic Algorithms to Weight Acoustic Features for Speaker Recognition

Maider Zamalloa[1,2], Germán Bordel[1], Luis Javier Rodríguez[1], Mikel Penagarikano[1], Juan Pedro Uribe[2]

(1) GTTS, Departamento de Electricidad y Electrónica, Universidad del País Vasco
(2) Ikerlan - Technological Research Centre

mzamalloa001@ikasle.ehu.es

Abstract

The Mel-Frequency Cepstral Coefficients (MFCC) are widely accepted as a suitable representation for speaker recognition applications. MFCC are usually augmented with dynamic features, leading to high dimensional representations. The issue arises of whether some of those features are redundant or dependent on other features. Probably, not all of them are equally relevant for speaker recognition. In this work, we explore the potential benefit of weighting acoustic features to improve speaker recognition accuracy. Genetic algorithms (GAs) are used to find the optimal set of weights for a 38-dimensional feature set. To evaluate each set of weights, recognition error is measured over a validation dataset. Naive speaker models are used, based on empirical distributions of vector quantizer labels. Weighting acoustic features yields 24.58% and 14.68% relative error reductions in two series of speaker recognition tests. These results provide evidence that further improvements in speaker recognition performance can be attained by weighting acoustic features. They also validate the use of GAs to search for an optimal set of feature weights.[1]

Index Terms: speaker recognition, feature extraction, genetic algorithms

1. Introduction

Feature extraction is a key issue for efficient speaker recognition. Redundant and harmful information should be removed from speech, retaining only those features relevant to classification. Many state-of-the-art speaker recognition systems use a set of short-term spectrum features called Mel-Frequency Cepstral Coefficients (MFCC). Not surprisingly, MFCC's are also used for speech recognition, since they not only convey information about the glottal source and the vocal tract shape and length, which are speaker specific features, but also the frequency distribution identifying sounds. Additionally, it has been shown that dynamic information improves significantly the performance of recognizers. So, MFCC, energy and their first and second derivatives are commonly used as features. Depending on the acoustic front-end, the resulting feature vectors may have from 20 to 50 components. The issue arises of whether some of those features are redundant or dependent on other features. Probably, not all of them are equally relevant for speaker recognition. Some of them may be even discarded. In this work, the problem is addressed from the point of view of feature weighting. An exhaustive search of the optimal weights is too costly even for a moderate number of features, so heuristic approaches must be applied.

[1] Work partially funded by the Government of the Basque Country, under program SAIOTEK, project S-PE04UN18.

Genetic Algorithms (GAs for short), introduced by Holland in 1975 [1], are randomized heuristic search techniques based on biological evolution strategies, with three basic operations: selection of the fittest, crossover and mutation. GAs are usually applied in complex optimization problems. Candidate solutions are represented by individuals (or chromosomes) in a large population. Initial solutions may be randomly generated or obtained by other means. Then GAs iteratively drive the population to an optimal point according to a complex metric (called fitness or evaluation function) that measures the performance of the individuals in a target task. The fittest individuals are selected and their chromosomes mixed, mutated or taken unchanged to the next generation. A major advantage of the GAs over other heuristic search techniques is that they do not rely on any assumption about the properties of the evaluation function. Multiobjective evaluation functions (e.g. combining the accuracy and the cost of classification) can be defined and used in a natural way. In particular, GAs can easily encode feature weights as sequences of integer or real values in a chromosome, allow to smartly explore the feature space by retaining those values that benefit the classification task and simultaneously avoid local optima due to their intrinsic randomness.

In this work, a genetic algorithm is used to find the optimal set of weights for a 38-dimensional feature set, consisting of 12 MFCC, their first and second derivatives, energy and its first derivative. Weights are encoded as 8-bit integers, so each individual (representing a set of weights) is encoded by 304 bits. Speaker recognition error, measured over a validation dataset, is used as evaluation function. Naive speaker models are used, based on empirical distributions of acoustic labels. A database of read speech in Spanish, including 204 speakers, is used for the experiments. The approach presented in this paper is related to other works applying GAs to feature selection [2], feature extraction [3] and feature weighting [4] in speaker recognition tasks.

The rest of the paper is organized as follows. The speaker recognition system and the GA-based search of the optimal weights are described in Section 2. Experimental setup is described in Section 3, including the speech database used to train and test speaker models and the tuning phase of the GA. Speaker recognition results using non-weighted and weighted features are presented and discussed in Section 4. Finally, Section 5 summarizes our approach and outlines future work.

2. Methodology

2.1. The speaker recognition system

2.1.1. Acoustic front-end

Speech, acquired at 16 kHz, is analysed in frames of 25 milliseconds (400 samples), at intervals of 10 milliseconds. A Hamming window is applied and a 512-point FFT computed. The FFT amplitudes are then averaged in 24 overlapped triangular filters, with central frequencies and bandwidths defined according to the Mel scale. A Discrete Cosine Transform is finally applied to the logarithm of the filter amplitudes, obtaining 12 Mel-Frequency Cepstral Coefficients (MFCC). To increase robustness against channel distortion, Cepstral Mean Normalization (CMN) [5] is applied on an utterance-by-utterance basis. The first and second derivatives of the MFCC, the frame energy (E) and its derivative are also computed, thus yielding a 38-dimensional feature vector.

2.1.2. Vector Quantization

Vector Quantization (VQ) [6][7] is applied, yielding an optimal codebook of $L=256$ centroids which minimizes the average distortion (Euclidean distance) in quantifying feature vectors in the training set. Then, each feature vector in the database is replaced by the index of the closest centroid. So, each speech utterance is first analysed to get a sequence of feature vectors $X=\{x_1,x_2,...,x_T\}$, and then, by applying VQ, transformed into a sequence of acoustic labels $Y=\{y_1,y_2,...,y_T\}$ corresponding to the closest VQ centroids.

2.1.3. Speaker models

Speaker models are distributions of VQ labels. These simple models have been successfully used for speaker adaptation through speaker clustering in speech recognition tasks [8]. Let $U(i)$ be the training subset corresponding to speaker i, $c(i)$ the number of VQ labels in $U(i)$, and $c(k,i)$ the number of times the label k appears in $U(i)$. Then, the conditional probability $P(k|i)$ can be empirically estimated as follows:

$$P(k \mid i) = \frac{c(k,i)}{c(i)}. \quad (1)$$

Finally, assuming that successive labels are independent, the conditional probability of a sequence of labels $Y=\{y_1,y_2,...,y_T\}$, given speaker i, can be computed as follows:

$$P(Y \mid i) = \sum_{t=1}^{T} P(Y(t) \mid i). \quad (2)$$

2.1.4. Speaker recognition

Assuming that input utterances are produced by S known speakers, given the sequence of labels $Y=\{y_1,y_2,...,y_T\}$, the most likely speaker is selected:

$$\hat{i}(Y) = \arg\max_{i=1,...,S} \{P(i \mid Y)\}. \quad (3)$$

Applying the Bayes rule, it follows:

$$\hat{i}(Y) = \arg\max_{i=1,...,S} \left\{ \frac{P(Y \mid i)P(i)}{P(Y)} \right\} = \arg\max_{i=1,...,S} \{P(Y \mid i)P(i)\}, \quad (4)$$

because maximizing over the set of speakers does not depend on the acoustic sequence. Then, assuming that all speakers have equal *a priori* probabilities, it follows:

$$\hat{i}(Y) = \arg\max_{i=1,...,S} \{P(Y \mid i)\}, \quad (5)$$

and introducing (2) into (5):

$$\begin{aligned} \hat{i}(Y) &= \arg\max_{i=1,...,S} \left\{ \prod_{t=1}^{T} P(Y(t) \mid i) \right\} \\ &= \arg\max_{i=1,...,S} \left\{ \log \prod_{t=1}^{T} P(Y(t) \mid i) \right\} \\ &= \arg\max_{i=1,...,S} \left\{ \sum_{t=1}^{T} \log P(Y(t) \mid i) \right\} \end{aligned}. \quad (6)$$

So, the computational cost of speaker recognition is linear with the number of speakers (S) and with the length of the input utterance (T), involving $S \times T$ memory accesses, $S \times (T-1)$ sums and $S-1$ comparisons. For convenience, it is assumed that label probabilities are stored in logarithmic form.

2.1.5. Feature weighting and speaker recognition

Feature weighting do not affect speaker models directly, but only through the VQ process. For each candidate set of feature weights $W=\{w_1,w_2,...,w_D\}$, a different codebook $C(W)$ is computed by first weighting feature vectors in the training set and then using the Euclidean distance to measure distortion in the weighted space. So, $C(W)$ consists of the L centroids that minimize VQ distortion in the weighted space. Obviously, labelling the database also depends on weighting: W is applied to each feature vector X, and the resulting vector $X'=(w_1x_1,w_2x_2,...,w_Dx_D)$ is replaced by the index k corresponding to the closest centroid in $C(W)$.

2.2. GA-based search for the optimal weights

The well-known *Simple Genetic Algorithm* (SGA) [9] is employed to search for the optimal set of weights. As noted above, for each candidate set of weights $W=\{w_1,w_2,...,w_D\}$, a codebook $C(W)$ is computed and the whole database labelled according to W and C(W). Then, $c(k,i)$ and $c(i)$ are counted for each training subset $U(i)$, and speaker models $M=\{P(k|i)|k \in [1..L],\ i \in [1..S]\}$ estimated using (1). Finally, utterances in the validation set $V=\{V(1),V(2),...,V(S)\}$ are classified, and the classification accuracy used as fitness function:

$$F(W) = \sum_{i=1}^{S} \sum_{\forall Y \in V(i)} \delta\left(\hat{i}(Y) = i\right), \quad (7)$$

where:

$$\delta(x) = \begin{cases} 1 & if\ x = true \\ 0 & if\ x = false \end{cases}. \quad (8)$$

Once all the candidates are evaluated, some of them (usually the fittest ones) are selected, mixed and mutated in order to get the population for the next generation. SGA allows a number of individuals to survive for the next generation, which is called *elitism*. The simplest case of elitism, which consists of keeping the fittest individual, is applied. This guarantees that the fitness of the fittest individual increases monotonically with successive generations. If that increase is smaller than a given threshold, or a maximum number of generations is reached, the algorithm

stops and the fittest individual (i.e. the set of weights yielding the highest classification accuracy over validation data) is returned. Finally, a third independent set of utterances is used to test the optimal set of weights.

3. Experimental setup

3.1. The speech database

A phonetically balanced database in Spanish, called Albayzin [10], is used for the experiments. Albayzin was recorded at 16 kHz in laboratory conditions and was originally designed to train acoustic models for speech recognition. It contains 204 speakers, each contributing at least 25 utterances, each utterance lasting 3.55 seconds on average.

Using the classification rate to evaluate feature weights makes the optimization process very costly. To speed up the evaluation of the proposed methodology, instead of using the whole database, 10 partially overlapped subsets are defined, each containing 20 speakers, which amounts to 164 speakers all together. Each subset is further divided into three independent datasets: *training* (5 utterances per speaker), *validation* (10 utterances per speaker) and *test* (10 utterances per speaker). The training set is used to compute the VQ codebook and to estimate speaker models. The validation set is used by the GA to search for the optimal feature weights. Finally, the test set is used to evaluate the performance of weighted features in speaker recognition experiments.

3.2. Tuning the GA

The SGA is implemented by using ECJ, a *Java-based Evolutionary Computation and Genetic Programming Research System*, presently developed at George Mason University's Evolutionary Computation Laboratory and released under a special open source license [11]. ECJ shows very interesting features, including a flexible breeding architecture, arbitrary representations, fixed and variable length genomes, several multiobjective optimization methods and many selection operators.

Preliminary experimentation has been carried out to adjust the parameters that control the performance and the convergence of the SGA. It has been observed that populations of 50 individuals need at most 30 generations to converge, so no convergence criterion is applied and a fixed number of 30 generations is established. Chromosomes consist of 38 genes, each encoding a feature weight. To reduce computational costs as much as possible, 8 bits have been allocated for each weight. So, allowed gene values range from 0 to 255. Offspring is bred by first selecting and then mixing two parents in the current population. One of the parents is selected according to the fitness-proportional criterion. The second is selected according to the tournament method, by picking the fittest of 7 randomly chosen individuals (the choice of 7 has proven good in the experiments and is also suggested by the manufacturers of the toolkit). This mixed approach seems suitable because fitness-proportional selection guarantees that the fittest individuals are picked, and tournament selection introduces diversity, which is good for avoiding local optima. Crossover type and rate and mutation rate have been heuristically established to get a good balance between performance and convergence. Finally, as noted above, the simplest case of elitism is applied by keeping the fittest individual for the next generation. Tuned settings are summarized in Table 1.

4. Experimental results

Two series of experiments were carried out, by using 2 and 3 training utterances per speaker, respectively, over 10 subsets of 20 speakers. Hereafter, we will refer to them as the 2U and 3U experiments, respectively. GA-based optimization was applied, using the settings shown in Table 1, to get the optimal set of feature weights for each configuration.

Table 1. Tuned settings of the SGA parameters.

Parameter			Setting
Population size			50
Number of generations			30
Chromosome size (number of genes)			38
Gene Values	Minimum		0
	Maximum		255
Genetic operations	Crossover	Type	Two-point
		Rate	0.8
	Mutation	Rate	0.05
Selection	First Parent		Fitness-Proportional
	Second Parent		Tournament (Size: 7)
Elitism			1

Tables 2 and 3 show speaker recognition results using non-weighted and weighted features for the test and validation sets in the 2U and 3U experiments, respectively. Note that using weighted features did not always improve performance over the test sets, as in the case of the set of speakers ss09 in the 2U experiments, and the sets ss03, ss08 and ss10 in the 3U experiments. This is probably due to a lack of robustness of speaker models, which do not generalize well to data on the test sets. However, weighted features led in most cases (16/20, 80%) to higher recognition rates. This reveals that searching for the feature weights that maximize classification performance over validation data can help to compensate for the lack of robustness of speaker models. As a result, classification performance was also improved over independent test sets. On average, recognition rates using weighted features were 2.95 and 1.05 points better than those achieved using non-weighted features, for the 2U and 3U experiments, representing error reductions of 24.58% and 14.68%, respectively. As may be expected, the more robust the speaker models are the more difficult is to improve their performance by feature weighting.

Table 2. Speaker recognition error rates with non-weighted and weighted features for the 2U experiments (2 utterances for training, 10 utterances for validation and 10 utterances for test) over 10 different sets of speakers (ss01, ss02, etc.).

	Test set		Validation set	
	Non-Weighted	**Weighted**	**Non-Weighted**	**Weighted**
ss01	18.00	13.00	18.50	8.00
ss02	13.50	10.00	20.00	7.50
ss03	14.00	13.50	12.50	4.50
ss04	16.00	9.50	13.00	4.00
ss05	11.00	7.50	10.00	4.00
ss06	10.50	7.50	5.50	3.00
ss07	11.00	6.00	11.00	4.50
ss08	6.50	4.00	8.00	3.00
ss09	10.00	11.50	15.50	6.00
ss10	9.50	8.00	8.50	3.00
Average	**12.00**	**9.05**	**12.25**	**4.75**

Error rates over validation sets were significantly lower than those achieved over test sets, since feature weights were searched specifically to maximize the performance over validation data. So, as shown in Tables 2 and 3, weighted features always provided better performance than non-weighted features in the 2U and 3U experiments over validation sets. In the 2U experiments, recognition rates using weighted features were, on average, 7.5 points better than those achieved using non-weighted features, which represents a 61.22% error reduction. The average error reduction was even larger in the 3U experiments (70.63%, 4.45 points). These results provide evidence that further improvements in speaker recognition performance can be attained by weighting acoustic features. They also validate the use of GAs to search for an optimal set of feature weights.

Table 3. Speaker recognition error rates with non-weighted and weighted features for the 3U experiments (3 utterances for training, 10 utterances for validation and 10 utterances for test) over 10 different sets of speakers (ss01, ss02, etc.).

	Test set		Validation set	
	Non-Weighted	Weighted	Non-Weighted	Weighted
ss01	8.50	6.00	9.00	1.50
ss02	6.50	3.50	9.00	2.50
ss03	5.00	11.50	4.50	3.50
ss04	12.00	4.00	7.50	1.50
ss05	6.50	6.00	8.50	1.00
ss06	7.00	3.50	3.50	1.00
ss07	9.00	8.50	4.00	2.00
ss08	4.50	6.00	6.00	1.50
ss09	7.50	6.00	6.50	0.50
ss10	5.00	6.00	6.50	3.50
Average	**7.15**	**6.10**	**6.30**	**1.85**

5. Conclusions and future work

Mel-Frequency Cepstral Coefficients and their derivatives are commonly used as acoustic features in speaker recognition systems. However, some of them may be redundant, dependent on other features or even harmful. In this work, the issue is addressed from the point of view of feature weighting. Genetic Algorithms are applied to search for an optimal set of feature weights, and simple distributions of acoustic labels (obtained through vector quantization) are used as speaker models.

The proposed methodology was evaluated over a database of clean speech in Spanish containing 204 speakers. To speed up the evaluation process, 10 partially overlapped subsets were defined, each containing 20 speakers, and results were averaged over them. Two series of experiments were carried out, with 2 and 3 training utterances per speaker. A GA-based search was run to get the optimal weights, i.e. those minimizing speaker recognition errors over validation data. Finally, weighted features yielded, on average, error reductions of 24.58% and 14.68% in the two series of speaker recognition experiments, respectively. This demonstrates that improved performance can be attained by weighting acoustic features, and validates the use of Genetic Algorithms to search for an empirically optimal set of feature weights.

Future work includes finding the optimal weights for the whole database, which may require the use of more robust speaker models, since the probability of guessing decreases drastically from 1/20 to 1/204. Also, a selection procedure is being designed based on weights to retain the K most relevant features, thus reducing storage and computational costs, which is crucial for speaker recognition applications running on low-resource devices.

6. References

[1] J.H. Holland. *"Adaptation in natural and artificial systems"*. University of Michigan Press, 1975 (reprinted in 1992 by MIT Press, Cambridge, MA).

[2] M. Demirekler, A. Haydar. *"Feature Selection Using a Genetics-Based Algorithm and its Application to Speaker Identification"*, Proceedings of the IEEE ICASSP'99, pp. 329—332, Phoenix, Arizona, 1999.

[3] C. Charbuillet, B. Gas, M. Chetouani, J.L. Zarader. *"Filter Bank Design for Speaker Diarization Based on Genetic Algorithms"*, to appear in Proceedings of the IEEE ICASSP'06, Toulouse, France, May 2006.

[4] D. Charlet, D. Jouvet. *"Optimizing feature set for speaker verification"*, Pattern Recognition Letters, Vol. 18, No. 9, pp. 873—879, September 1997.

[5] A.E. Rosenberg, C.H. Lee, F.K. Soong. *"Cepstral Channel Normalization Techniques for HMM-Based Speaker Verification"*, Proceedings of the ICSLP'94, pp. 1835—1838, Yokohama, Japan, 1994.

[6] Y. Linde, A. Buzo, R.M. Gray. *"An Algorithm for Vector Quantizer Design"*, IEEE Transactions on Communications, Vol. 28, No. 1, pp. 84—95, January 1980.

[7] G. Patanè and M. Russo. *"The enhanced LBG algorithm"*. Neural Networks, Vol. 14, No. 9, pp. 1219—1237, November 2001.

[8] L.J. Rodríguez, M.I. Torres. *"A Speaker Clustering Algorithm for Fast Speaker Adaptation in Continuous Speech Recognition"*, in P. Sojka, I. Kopecek and K. Pala Eds., Proceedings of the 7th International Conference on Text, Speech and Dialogue (Brno, Czech Republic, September 2004), pp. 433—440, LNCS/LNAI 3206, Springer-Verlag, 2004.

[9] D.E. Goldberg. *"Genetic Algorithms in Search, Optimization and Machine Learning"*, Addison-Wesley, 1989.

[10] F. Casacuberta, R. García, J. Llisterri, C. Nadeu, J.M. Pardo, A. Rubio. *"Development of Spanish Corpora for Speech Research (Albayzín)"*, in G. Castagneri Ed., Proceedings of the Workshop on International Cooperation and Standardization of Speech Databases and Speech I/O Assessment Methods, Chiavari, Italy, 26-28 September 1991, pp. 26—28.

[11] ECJ 13, http://cs.gmu.edu/~eclab/projects/ecj/.