FEATURE DIMENSIONALITY REDUCTION THROUGH GENETIC ALGORITHMS FOR FASTER SPEAKER RECOGNITION

M. Zamalloa[†‡], *L. J. Rodriguez-Fuentes*[†], *M. Penagarikano*[†], *G. Bordel*[†], *J. P. Uribe*[‡]

†Grupo de Trabajo en Tecnologías del Software, DEE, ZTF/FCT, University of the Basque Country
Barrio Sarriena s/n, 48940 Leioa, SPAIN

‡Ikerlan – Technological Research Centre
Paseo J.M. Arizmendiarrieta 2, 20500 Arrasate-Mondragón, SPAIN

phone: +34 946012716, fax: +34 946013500, email: luisjavier.rodriguez@ehu.es
web: http://gtts.ehu.es/TWiki/bin/view (in Spanish)

ABSTRACT

Mel-Frequency Cepstral Coefficients and their derivatives are commonly used as acoustic features for speaker recognition. Reducing the number of features leads to more robust estimates of model parameters, and speeds up the classification task, which is crucial for real-time speaker recognition applications running on low-resource devices. In this paper, a feature selection procedure based on Genetic Algorithms (GA) is presented and compared to two well-known dimensionality reduction techniques, namely PCA and LDA. Evaluation is carried out for two speech databases, containing laboratory read speech and telephone spontaneous speech, applying a standard speaker recognition system. Results suggest that dynamic features are less discriminant than static ones, since the low-size optimal subsets found by the GA did not include dynamic features. GA-based feature selection outperformed PCA and LDA when dealing with clean speech, whereas PCA and LDA outperformed GA-based feature selection for telephone speech, probably due to some kind of noise compensation implicit in linear transforms, which cannot be accomplished just by selecting a subset of features.

1. INTRODUCTION

Smart environments with pervasive computing capabilities require automatic adaptation/customization of the products and services they provide. Speaker identification is a natural way of customizing many services, by first identifying and then retrieving information about clients. After identification, client choices or activities can be tracked and stored, improving their profiles for further interactions. In these applications, transparency and naturalness are critical. Clients should be aware that their activities are being tracked, their voice recorded and their profile information stored, but these actions should not interfere with the service itself. Depending on the interface, speakers might be continuously tracked, or just identified when the service is started. In any case, they cannot be asked even for just a few seconds of speech data to create accurate profiles, nor interactions delayed due to a computationally expensive search of their profiles. Moreover, clients may be accessing the service through a portable or embedded device with low storage and computational capabilities. In this case, real-time operation becomes the most critical issue to allow natural interactions. For these latter applications, high dimensional feature vectors do not seem suitable, and some kind of dimensionality reduction technique must be applied to save as much time as possible with no or little performance degradation.

State-of-the-art speaker recognition systems use short-term spectrum features, the Mel-Frequency Cepstral Coefficients (MFCC) [4], because they convey not only the frequency distribution identifying sounds, but also speaker specific features. Additionally, it has been shown that dynamic information improves the performance of recognizers, so MFCC, energy and their first and second derivatives are commonly used as features. The resulting feature vectors may consist of up to 50 components, all of them conveying a certain amount of relevant information.

Literature is plenty of comparative studies which consider various feature extraction techniques and then select that yielding the best performance in a target task (see [15] for reference). In this work, instead, we aim to smartly reduce feature dimension to speed up computation while keeping performance. MFCC, energy and their first and second derivatives are taken as the baseline acoustic features. Then, the K features most relevant to the classification task are extracted. In other words, the D-dimensional feature space is transformed into a K-dimensional subspace ($K < D$) which minimizes the loss of relevant information. A robustness issue is also involved, since a limited amount of data is available to estimate speaker models.

The problem of dimensionality reduction is sometimes formulated as a linear transform which projects feature vectors on a transformed subspace defined by relevant directions. Given a D-dimensional feature vector X, a $K \cdot D$ matrix A is applied to get a K-dimensional vector Y of transformed features ($K < D$). The matrix A is estimated so that, from the point of view of classification, redundancy is removed and relevant information retained. This should, at least, optimize the performance for the target value of K, but it may even outperform the baseline feature set, due to the removal of harmful or confusing features and, more probably, to better (more robust) estimates of model parameters. The following methods have been proposed (among others):

– *Principal Component Analysis* (PCA) [12], an old technique of multivariate statistical analysis, consists of computing the eigenvectors of the $D \cdot D$ covariance matrix Σ, then sorting them according to the corresponding eigenvalues, in descending order, and finally building the projection matrix A (called *Karhunen-Loeve Transform*, KLT) with the largest K eigenvectors (i.e. the K directions of greatest variance). Each feature vector X is then pre-processed according to the expression $Y = A(X - \mu)$, where μ represents the mean feature vector. KLT decorrelates the features and provides the smallest possible reconstruction error among all linear transforms, i.e. the smallest possible mean-square error between the data vectors in the original D-dimensional space and the data vectors in the projected K-dimensional subspace. Unfortunately, this does not guarantee minimizing classification error.

– *Linear Discriminant Analysis* (LDA) [7] attempts to find the transform A that maximizes a criterion of class separability. This is done by computing the within-class and between-class variance matrices, Σ_{wc} and Σ_{bc}, then finding the eigenvectors of $\Sigma_{wc}^{-1}\Sigma_{bc}$, sorting them according to the eigenvalues in descending order, and finally building the projection matrix A with the first K eigenvectors (which define the K most discriminant hyperplanes). LDA assumes that all the classes share a common within-class covariance matrix, and that each class is modelled by a single Gaussian distribution. LDA also assumes that classes are linearly separable. Additionally, as any supervised approach, it requires labelling samples with class identities.

Linear transforms combine in an elegant way feature extraction and feature selection. However, these two steps can be also applied in an uncoupled way. Strictly speaking, feature selection consists of determining an optimal subset of features by exhaustively exploring all the 2^D possible combinations. Most feature selection procedures use the classification error as the evaluation function. This makes exhaustive search computationally unfeasible in practice, even for moderate values of D. The simplest method consists of evaluating the D features individually and selecting the K most discriminative ones, but it does not take into account dependencies among features. So a number of suboptimal heuristic search techniques have been proposed in the literature, which essentially trade-off the optimality of the selected subset for computational efficiency [11].

Genetic Algorithms (GA) suitably fit this kind of complex optimization problems. Candidate solutions are represented as individuals in a large population. Initial solutions (which may be randomly generated) are iteratively driven by the GA to an optimal point according to a complex metric 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 GA 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 [18] [14]. GA can easily encode decisions about selecting or not selecting features as sequences of boolean values, allow to smartly explore the feature space by retaining those decisions that benefit the classification task, and simultaneously avoid local optima due to their intrinsic randomness. GA have been recently applied to feature extraction [3], feature weighting [20] an feature selection [5] [19] in speaker recognition.

In [5], a reduced set of features was determined on a speaker-by-speaker basis by applying GAs to maximize a measure of discrimination between each speaker and her/his two closest neighbours. Speaker recognition performance was measured on a small dataset containing only 15 speakers, and using a very simple speaker identification algorithm. In [19], feature weighting was used as an intermediate step towards feature selection. GAs were applied to search for feature weights maximizing speaker recognition performance on a validation dataset. Speaker models were based on empirical distributions of acoustic labels, obtained through vector quantization. Finally, features were sorted according to their weights and the K features with greatest average ranks were retained and evaluated.

In this paper, a feature selection procedure based on a GA-driven search is presented and compared to PCA and LDA in a speaker recognition task. Experiments are carried out for two speech databases, containing laboratory read speech and telephone spontaneous speech, respectively. A standard GMM-based speaker recognition system is applied. The rest of the paper is organized as follows. The speaker recognition system and the feature selection approach are described in Sections 2 and 3, respectively. The experimental setup is outlined in Section 4, including details about the speech databases, the computation of MFCC, the speaker models and the implementations of GA, PCA and LDA. Section 5 presents the results of the GA-based feature selection approach in speaker recognition experiments, and compares them to those of PCA and LDA. Finally, conclusions are summarized in Section 6.

2. SPEAKER RECOGNITION

In this work, the distribution of feature vectors extracted from a speaker's speech is represented by a linear combination of M multivariate Gaussian densities, known as *Gaussian Mixture Model* (GMM) [16]. GMM parameters are estimated from speaker samples by applying the *Maximum Likelihood* (ML) criterion. Each sample X consists of a sequence of D-dimensional feature vectors: $X = (x_1, x_2, \ldots, x_T)$. The conditional probability of a feature vector x, given the speaker model $\lambda = \{w_j, \mu_j, \Sigma_j | j = 1, \ldots, M\}$, is computed as follows:

$$p(x|\lambda) = \sum_{j=1}^{M} w_j \mathscr{N}(x; \mu_j, \Sigma_j) \qquad (1)$$

where $\mathscr{N}(x; \mu, \Sigma)$ denotes the D-dimensional normal density function of mean vector μ and covariance matrix Σ, and the mixture weights satisfy the constraint $\sum_{j=1}^{M} w_j = 1$.

We assume that input utterances are produced by S known speakers, represented by their corresponding models $\lambda_1, \lambda_2, \ldots, \lambda_S$. Then, for any input utterance $X = (x_1, x_2, \ldots, x_T)$, the most likely speaker $\hat{i}(X)$ is selected according to the following expression:

$$\hat{i}(X) = \arg \max_{i=1,\ldots,S} \log p(\lambda_i | X) \qquad (2)$$

Applying the Bayes rule, taking into account that maximizing over the set of speakers does not depend on the acoustic sequence, assuming that all the speakers have equal *a priori* probabilities and that acoustic vectors are statistically independent, it follows:

$$\begin{aligned} \hat{i}(X) &= \arg \max_{i=1,\ldots,S} \log(p(X|\lambda_i)p(\lambda_i)) \\ &= \arg \max_{i=1,\ldots,S} \log \prod_{t=1}^{T} p(x_t|\lambda_i) \\ &= \arg \max_{i=1,\ldots,S} \sum_{t=1}^{T} \log p(x_t|\lambda_i) \qquad (3) \end{aligned}$$

According to Eq. 3, the computational cost of speaker recognition depends linearly on the number of speakers (S) and on the length of the input utterance (T). Since GMM are used as speaker models, the computational cost also depends linearly on the number of mixtures (M) and on the dimension of the feature space (D).

3. FEATURE SELECTION USING GENETIC ALGORITHMS

In this study, the well-known *Simple Genetic Algorithm* (SGA) [10] is applied to search for the optimal feature set. The evaluation of feature sets (i.e. the fitness function used by the GA) is based on the classification accuracy obtained in speaker recognition experiments for development data.

The GA-driven selection process begins by fixing the target size K of the reduced feature subspace. Then, an initial population of candidate solutions (K-feature subsets) is randomly generated. To evaluate the K-feature subset $\Gamma = \{f_1, f_2, \ldots, f_K\}$, the following steps are carried out: (1) the acoustic vectors of the whole speech database are reduced to the components enumerated in Γ; (2) speaker models are estimated using a training corpus; (3) utterances in a development corpus are classified by applying the speaker models; and (4) the speaker recognition accuracy obtained for the development corpus is used to evaluate Γ.

Each candidate solution is represented by a D-dimensional vector of positive integers $R = \{r_1, r_2, \ldots, r_D\}$, the K highest values determining what features are selected. Note that the same feature set Γ may be represented by different vectors, that is, modifications to a given candidate solution R might not change the selection of features. This redundancy in representation makes the genetic algorithm to evolve smoothly and facilitates its convergence.

At the end of each iteration/generation, after all the K-feature subsets in the population are evaluated, some of them (usually the fittest ones), are selected, mixed and mutated in order to get the population for the next generation. Mutation is used to introduce small variations that help decrease the chances of getting local optima. On the other hand, *elitism* (copying some of the fittest individuals to the next generation) is applied to guarantee that the fitness function 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 optimal K-feature subset $\hat{\Gamma} = \{\hat{f}_1, \hat{f}_2, \ldots, \hat{f}_K\}$ is returned. Finally, $\hat{\Gamma}$ is evaluated on a test corpus. The three datasets used in this procedure: training, development and test, are independent and composed of disjoint sets of utterances.

4. EXPERIMENTAL SETUP

4.1 Speech databases

Two series of experiments were carried out for two different databases, emphAlbayzín and *Dihana*, each partitioned in three sets: (1) the training set, used to estimate the speaker models; (2) the development set, used by the GA to compute the fitness function; and (3) the test set, used to evaluate the performance of the optimal K-feature subset.

Albayzín is a phonetically and gender-balanced database in Spanish, recorded at 16 KHz in laboratory conditions [2]. It contains 204 speakers, each speaker contributing 25 utterances in a single session, each utterance lasting an average of 3.55 seconds. The 25 utterances corresponding to each speaker are distributed as follows: 10 are taken for training, 7 for development and 8 for testing. So, the training, development and test sets are composed of 2040, 1428 and 1632 utterances, respectively.

Dihana is a spontaneous task-specific speech corpus in Spanish, recorded at 8 kHz through telephone lines [1]. It contains 900 human-machine dialogues from 225 speakers (153 men, 72 women), acquired through a *Wizard of Oz* setup [9]. Additionally, each speaker recorded 8 phonetically balanced and 8 task-specific read utterances. So, each speaker contributes with 4 dialogues and 16 read utterances, recorded in one or more sessions. The training set consists of 2 dialogues and 8 phonetically balanced read utterances per speaker, and both the development and test sets consist of 1 dialogue and 4 task-specific read utterances per speaker. The training, development and test sets contain 4598, 2379 and 2897 utterances, respectively.

4.2 Speech processing

Speech is analysed in 25-millisecond frames, at intervals of 10 milliseconds. A Hamming window is applied and an FFT computed, whose length depends on the sampling frequency: 256 points for signals sampled at 8 khz and 512 points for signals sampled at 16 kHz. FFT amplitudes are then averaged in 20 (8 kHz) or 24 (16 kHz) 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 10 (8 kHz) or 12 (16 kHz) Mel-Frequency Cepstral Coefficients (MFCC). To increase robustness against channel distortion, Cepstral Mean Normalization (CMN) [17] is applied on an utterance-by-utterance basis. The first and second derivatives of the MFCC, the frame energy (E) and its first and second derivatives are also computed, thus yielding a 33-dimensional (8 khz) or a 39-dimensional (16 kHz) feature vector.

4.3 Speaker models

The baseline system uses 32-mixture diagonal covariance GMM as speaker models. The number of mixtures was tuned in preliminary experiments, aiming to get a suitable trade-off between computational load and performance. ML estimates of model parameters are computed from speaker samples by applying the iterative *Expectation-Maximization* (EM) algorithm [6], starting from random values. Though few iterations are enough for the model parameters to converge, the random nature of initialization implies that different runs of the EM algorithm can lead to different parameter estimates. We discuss in Section 5 some issues related to this fact.

4.4 GA implementation

The genetic algorithm was implemented by means of ECJ [8], a Java-based Evolutionary Computation and Genetic Programming Research System, developed at George Mason University's Evolutionary Computation Laboratory and released under a special open source license. Preliminary experimentation was carried out to adjust the parameters that control the performance and convergence of the GA. Population size is one of the most critical parameters: high volume populations make the convergence of the algorithm too slow, whereas too small populations could limit the search performance. An optimal population size was determined for each K. On the other hand, it was observed that 40 generations were enough to converge in most cases, so no other convergence criterion was applied. The allowed gene values ranged from 0 to 255 (8 bits). Offspring was bred by first selecting and then mixing two parents in the current population. The first parent was selected according to the fitness-proportional criterion, by picking the fittest from seven randomly chosen individuals. The second parent was chosen in the same way, but only from two randomly chosen individuals, to allow diversity and avoid local optima. One-point crossover was applied and the mutation probability was set to 0.01. Finally, the simplest case of elitism was applied by keeping the fittest individual for the next generation.

4.5 PCA and LDA implementations

LNKnet [13], a public domain software developed at MIT Lincoln Laboratory, was used to perform PCA. Regarding LDA, a custom implementation was developed in Java. It computes the within-class covariance matrix Σ_{wc} as a weighted average of the covariance matrices of speakers, using the fraction of training samples corresponding to each speaker as weight. Covariance matrices of speakers are estimated from training data, assuming a single Gaussian density model. The between-class covariance matrix is computed by subtracting the within-class covariance matrix from the global covariance matrix: $\Sigma_{bc} = \Sigma_g - \Sigma_{wc}$.

5. RESULTS AND DISCUSSION

5.1 Performance of the feature sets provided by the GA

Table 1 shows the mean speaker recognition error rates and the 95% confidence intervals obtained with the optimal feature sets provided by the GA. Recognition results for three reference sets (MFCC, MFCC+E and the full feature vector) are shown too. To illustrate the consistency of the optimal sets provided by the GA, error rates for both the development and test sets are shown. Note that the GA looks for the best K-dimensional feature subset by performing speaker recognition experiments on a development dataset, whereas the performance of the optimal subset is measured on an independent test set. The close correlation between the rates for both sets supports the use of genetic algorithms for this kind of optimization problems.

Confidence intervals allow significant performance comparisons among different feature sets. This deserves a brief explanation. Model estimations start from random initializations. Preliminary experimentation showed that, fixed the set of features and the training database, random initializations led to slightly different model parameters after convergence, and therefore slight differences in speaker recognition performance were observed. This uncertainty can be taken into account in performance comparisons by computing the mean error rate and the corresponding confidence interval in a significant number of experiments. In this study, the whole process of training speaker models and carrying out speaker recognition experiments on the test set was repeated 20 times, and the 95% confidence interval was computed, assuming a Gaussian distribution of error rates.

In the experiments for clean speech, the recognition error rate decreases consistently as the number of features increases from 6 to 12, but performance improvements become relatively smaller for $K > 12$. Since optimal feature sets for $K \leq 12$ consist exclusively of a number of MFCCs plus the frame energy, this suggests that,

Table 1: Mean error rates and 95% confidence intervals in speaker recognition experiments for clean and telephone speech using the optimal K-dimensional feature subsets provided by the GA, for K = 6, 8, 10, 11, 12, 13, 20 and 30. Results using MFCC, MFCC+E and the full feature vector are shown too, for reference.

K	**Clean speech**		**Telephone speech**	
	Development	**Test**	**Development**	**Test**
6	7.64±0.12	5.71±0.09	31.76±0.16	34.23±0.12
8	2.86±0.12	1.81±0.09	21.99±0.13	23.90±0.14
10	2.24±0.11	0.94±0.04	17.91±0.16	19.70±0.12
11	0.81±0.06	0.35±0.04	17.64±0.11	19.32±0.14
12	1.23±0.07	0.30±0.04	17.37±0.09	19.27±0.14
13	1.05±0.06	0.36±0.03	17.30±0.12	19.12±0.14
20	0.67±0.09	0.16±0.02	17.59±0.09	19.99±0.11
30	0.57±0.05	0.13±0.02	16.05±0.14	19.10±0.14
MFCC	1.27±0.08	0.40±0.06	17.91±0.16	19.70±0.12
MFCC+E	0.90±0.05	0.22±0.04	19.76±0.14	22.34±0.10
Full feature vector	0.77±0.09	0.20±0.03	15.66±0.16	18.69±0.15

when dealing with clean laboratory speech, the information about speaker characteristics contained in dynamic features (first and second derivatives) is less relevant than that contained in static features. It does not mean that dynamic features are useless. Many studies have demonstrated that including them improves performance. It only means that when reduced sets must be defined, static features are the best choice.

Results for telephone speech also support this conclusion: the optimal feature sets for $K \leq 10$ are composed exclusively by MFCCs, and performance improvements for $K > 10$ are very small. It is worth noting the case of the reference subset composed of 10 MFCC and the frame energy, whose performance is 1.85 absolute points worse than that of the subset composed exclusively by 10 MFCC. This result reveals the lack of robustness of the frame energy when dealing with telephone speech, an issue that was already discovered by the GA in the selection experiments, since the optimal feature subsets for $K \leq 13$ did not include the frame energy.

5.2 Comparing GA to PCA and LDA

GA-based feature selection projects the original D-dimensional feature space into a reduced K-dimensional subspace by just selecting K features. PCA and LDA not only reduce but also scale and rotate the original feature space, through a transformation matrix A which optimizes a given criterion on the training data. From this point of view, PCA and LDA generalize feature selection, but the criteria applied to compute A (the highest variance in PCA, and the highest ratio of between to within class variances in LDA) do not match the criterion applied in evaluation (the highest speaker recognition accuracy). This is the strong point of GA, since feature selection is performed in order to maximize the speaker recognition rate on an independent development corpus.

GA-based feature selection, PCA and LDA were tested in speaker recognition experiments on clean and telephone speech. First, D-dimensional feature vectors were transformed into reduced K-dimensional feature vectors, according to the optimal subset/transformation given by GA, PCA or LDA, then speaker models were estimated on the training corpus and finally speaker recognition experiments were carried out on the test corpus. Results are shown in Table 2 (results for GA are the same shown in Table 1). Again, the mean error rate and the 95% confidence interval in 20 different experiments are given, to account for the uncertainty intrinsic to the estimation of GMM parameters.

In the case of clean speech, neither PCA nor LDA outperformed GA. PCA yielded lower error rates than LDA for $K > 12$. For $K \leq 12$, LDA outperformed PCA. However, the error rates are too low and the differences in performance too small for these conclusions to be statistically significant.

Error rates for telephone speech were much higher than those obtained for clean speech. Besides considering the presence of channel and environment noise, it can be argued that a large part of that corpus consists of spontaneous speech. The presence of noise makes PCA and LDA more suitable than GA, because feature selection cannot compensate for noise, whereas linear transforms can do it to a certain extent. This may explain why either PCA or LDA outperformed GA in all cases but for $K = 8$. LDA was the best approach in most cases (for K = 10, 11, 12, 13 and 20), whereas GA was the second best approach for K = 6, 10, 11, 12 and 13. On the other hand, the lowest error rate (15.97%) was obtained for $K = 30$ using PCA.

In summary, the GA-based feature selection scheme proposed in this paper seems to be competitive only when dealing with clean speech, though it performs quite well even for telephone-channel speech when the target K is small. Authors that argue against GA optimization say that it is too costly, since it requires iteratively evaluating candidate solutions in classification experiments over a development dataset. It must be noted, however, that GA optimization is done off-line, so the computational cost is not an issue in practice. Moreover, during recognition, feature selection is less costly than feature transformation.

5.3 Empirical time savings

To check empirically the time savings that could be attained by reducing the number of features, recognition times were recorded for several values of K in two different computers (see Figure 1). As expected, the running time t grew linearly with K. In the case of Albayzín (clean/laboratory/read speech), using 13-dimensional feature vectors took on average around 40% the time of using full 39-dimensional feature vectors. In the case of Dihana (telephone/office/spontaneous speech), similar savings were observed when comparing the running times of 10-dimensional and 33-dimensional feature vectors.

6. CONCLUSIONS

In this work, genetic algorithms were applied to search for the subset of K features maximizing the recognition performance. Alternatively, two well-known feature dimensionality reduction techniques, PCA and LDA, were applied and their performance compared to that of the GA-based feature selection approach. Experiments were carried out for two speech databases in Spanish, containing read speech in laboratory conditions and spontaneous speech through telephone lines, respectively, applying a standard GMM-based speaker recognition system.

Feature selection based on GA suggests that static features are more discriminant than dynamic features for speaker recognition applications. If a reduced set of features had to be selected (due to storage or computational restrictions), MFCC would be the best choice, augmented with the frame energy when dealing with clean-laboratory speech. In the case of telephone speech, the smallest feature subsets ($K \leq 13$) did not include the frame energy, which reveals that channel and/or environment noise is distorting the infor-

Table 2: Mean error rates and 95% confidence intervals in speaker recognition experiments on test data for clean and telephone speech, using the optimal K-dimensional feature sets provided by GA, PCA nd LDA, for K = 6, 8, 10, 11, 12, 13, 20 and 30.

K	**Clean speech**			**Telephone speech**		
	GA	**PCA**	**LDA**	**GA**	**PCA**	**LDA**
6	**5.71±0.09**	14.37±0.15	8.11±0.14	34.23±0.16	**33.23±0.12**	35.52±0.14
8	**1.81±0.09**	5.86±0.12	2.64±0.09	**23.90±0.14**	24.19±0.13	25.06±0.13
10	**0.94±0.04**	2.73±0.12	1.21±0.06	19.70±0.12	20.67±0.12	**19.43±0.12**
11	**0.35±0.04**	1.61±0.07	1.12±0.06	19.32±0.14	20.27±0.13	**18.10±0.13**
12	**0.30±0.04**	0.94±0.06	0.79±0.06	19.27±0.14	19.75±0.16	**18.18±0.12**
13	**0.33±0.05**	0.56±0.05	0.88±0.04	19.12±0.11	19.63±0.10	**17.66±0.10**
20	**0.16±0.02**	0.19±0.02	0.39±0.04	19.99±0.11	17.61±0.13	**17.24±0.11**
30	**0.13±0.02**	0.15±0.03	0.33±0.04	19.10±0.14	**15.97±0.15**	18.17±0.12

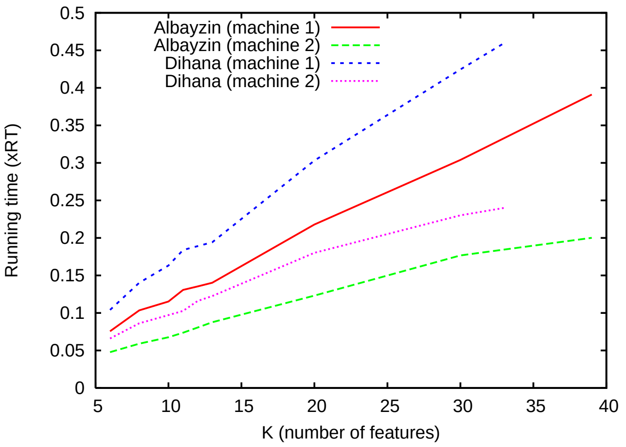

Figure 1: Average running times (real-time factor, xRT) for several values of K, in speaker recognition experiments carried out in two different computers (machine 1: 2 x Quad Core Intel Xeon E5320, 1.86GHz, 1066MHZ FSB, 4GB FB 667MHz; machine 2: 2 x AMD Opteron270 64bit Dual Core 2.0Ghz, 4GB).

mation it conveys. Regarding the methodology, the consistency of the feature selection results across the development and test datasets validates the use of GA for this kind of optimization problems.

GA outperformed PCA and LDA only when dealing with clean speech, whereas PCA and LDA outperformed GA in most cases when dealing with telephone speech, probably due to some kind of noise compensation implicit in linear transforms, which cannot be accomplished just by selecting a subset of features. In any case, since applying a linear transform is more costly than selecting a subset of features, depending on the target K, the gain in performance might not be worth the additional effort.

At the end of this study, we were tempted to combine the strong points of GA and linear transforms by applying GA to search for the linear transform that maximized the speaker recognition rate on a development set. However, such an approach was found unfeasible in practice, because determining $K \cdot D$ floating-point transform coefficients (instead of just K feature indices) requires a huge amount of training and development data (and a shocking amount of processing time) for the GA to converge and provide a robust transform.

7. ACKNOWLEDGEMENTS

This work has been partially funded by the Spanish MEC, under Plan Nacional de I+D+i, project TSI2006-14250-C02-01; the Government of the Basque Country, under program SAIOTEK, projects S-PE06UN48, S-PE06IK1, S-PE07UN43 and S-PE07IK3; and the University of the Basque Country, under project EHU06/96.

REFERENCES

[1] N. Alcocer, M. J. Castro, I. Galiano, R. Granel, S. Grau, and D. Griol. Adquisición de un Corpus de Diálogo: DIHANA. In *Actas de las III Jornadas en Tecnología del Habla (in Spanish)*, pages 131–134, Valencia (Spain), November 2004.

[2] F. Casacuberta, R. García, J. Llisterri, C. Nadeu, J. M. Pardo, and 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*, pages 26–28, Chiavari, Italy, September 1991.

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

[4] S. B. Davis and P. Mermelstein. Comparison of Parametric Representations for Monosyllabic Word Recognition in Continuously Spoken Sentences. *IEEE Transactions on Acoustic, Speech and Signal Processing*, 28(4):357–366, 1980.

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

[6] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society – Series B*, 39(1):1–38, September 1977.

[7] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification (Second Edition)*. Wiley Interscience, 2000.

[8] ECJ 16. http://cs.gmu.edu/ eclab/projects/ecj/.

[9] N. M. Fraser and G. N. Gilbert. Simulating speech systems. *Computer Speech and Language*, 5:81–99, 1991.

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

[11] A. K. Jain, R. P. W. Duin, and J. Mao. Statistical Pattern Recognition: A Review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(1):4–37, January 2000.

[12] I. T. Jolliffe. *Principal Component Analysis (Second Edition)*. Springer, 2002.

[13] R. P. Lippmann, L. Kukolich, and E. Singer. LNKnet: Neural Network, Machine Learning and Statistical Software for Pattern Classification. *Lincoln Laboratory Journal*, 6(2):249–268, 1993.

[14] L. S. Oliveira, R. Sabourin, F. Bortolozzi, and C. Y. Suen. A Methodology for Feature Selection Using Multiobjective Genetic Algorithms for Handwritten Digit String Recognition. *International Journal of Pattern Recognition and Artificial Intelligence*, 17(6):903–929, 2003.

[15] D. A. Reynolds. Experimental Evaluation of Features for Robust Speaker Identification. *IEEE Transactions on Speech and Audio Processing*, 2(4):639–643, October 1994.

[16] D. A. Reynolds and R. C. Rose. Robust Text-Independent Speaker Identification Using Gaussian Mixture Speaker Models. *IEEE Transactions on Speech and Audio Processing*, 3(1):72–83, January 1995.

[17] A. E. Rosenberg, C. H. Lee, and F. K. Soong. Cepstral Channel Normalization Techniques for HMM-Based Speaker Verification. In *Proceedings of the ICSLP'94*, pages 1835–1838, Yokohama, Japan, 1994.

[18] J. Yang and V. Honavar. Feature subset selection using a genetic algorithm. *IEEE Intelligent Systems*, 13(2):44–49, March 1998.

[19] M. Zamalloa, G. Bordel, L. J. Rodríguez, and M. Peñagarikano. Feature Selection Based on Genetic Algorithms for Speaker Recognition. In *IEEE Speaker Odyssey: The Speaker and Language Recognition Workshop*, pages 1–8, Puerto Rico, June 2006.

[20] M. Zamalloa, G. Bordel, L. J. Rodríguez, M. Peñagarikano, and J. P. Uribe. Using Genetic Algorithms to Weight Acoustic Features for Speaker Recognition. In *Proceedings of the ICSLP'06*, Pittsburgh (USA), September 2006.