A phoneme-space-representation heuristic to improve the performance in a cryptographic-speech-key generation task

L. Paola García-Perera, Juan A. Nolazco-Flores, and Carlos Mex-Perera

Computer Science Department, ITESM, Campus Monterrey
Av. Eugenio Garza Sada 2501 Sur, Col. Tecnológico
Monterrey, N.L., México, C.P. 64849
{paola.garcia, carlosmex, jnolazco}@itesm.mx

Abstract. In this paper we propose an improvement in the generation of the cryptographic-speech-key by using an heuristic consisting on the selection of the dimensions with the best performance for each of the phonemes. This selection can be made thanks that we know the phonemes of the spoken user passphrase. First, the mel frequency cepstral coefficients, (first and second derivatives) of the speech signal are calculated. Then, an Automatic Speech Recogniser, which models are previously trained, is used to detect the phoneme limits in the speech utterance. Afterwards, the feature vectors are built using both the phoneme-speech models and the information obtained from the phoneme segmentation. Next, the Support Vector Machines classifier, relying on an RBF kernel, computes the cryptographic key. By applying the phoneme-space-representation heuristic our results show an improvement of 24.26%, 18.85%, 16.56% for 10, 20 and 30 speakers, from the YOHO database, respectively, compared with our previous results.

1 Introduction

Biometrics have been widely developed for access control purposes, but they are also becoming generators of cryptographic keys [14]. From all the biometrics voice was chosen for this research since it has the advantage of being flexible. For instance, if a user utters different phrases the key produced must be different. This means that by changing a spoken sentence or word the key automatically changes. Furthermore, the main benefit of using voice is that it can simultaneously act as a passphrase for access control and as a key for encryption of data that will be stored or transmitted. Moreover, having a key generated by a biometric is highly desirable since the intrinsic characteristics are unique for each individual, therefore, it will be difficult to guess.

Given the biometric information it is also possible to generate a private key and a public key. As an application we can propose the following scenario. A user utters its pass phrase that operates in two ways: as a generator of a private and public key and as a passphrase for accessing his files. If and unauthorised user tries to access the files with a wrong pass phrase the access will be denied,

but even if the passphrase is correct the access will be denied since the phonetic features are not the ones that first generated the cryptographic keys. With this example we can have a view of the potentiality of using the voice to generate such keys. Similar applications can be found in [10].

The results obtained in our previous work explained our proposed system architecture [4, 5, 7]. In those studies we tested different types of kernels, as result we obtained that the RBF kernel was superior than the linear and polynomial kernels [7]. In other of our works, we examined the parametrisation of the speech with different kinds of parameters; from here we concluded that the best feature vector for the SVM is based on the mel frequency cepstral coefficients - as it is for speech recognition [4, 5]. We also experimented using different number of Gausssians, and have found that eight was the best number of Gaussians [6]. Finally, we have also investigated the benefit of tuning the SVM model per phoneme [6].

For Automatic Speech Recognition (ASR) task is well known that the optimal number of parameters is around twelve. It is very common to use this number that SPHINX, one of the most important software for automatic speech recognition employs it. Influenced by this trend, in our previous work, we had used this number. Moreover, since we know the phonemes of the passphrase, we can also select the dimensions in the feature space with the best performance for each of the phonemes. Consequently, the main purpose of this work is to improve the generation of the cryptographic-speech-key by selecting the dimensions with the best performance for each of the phonemes.

The system architecture is depicted in Figure 1 and will be discussed in the following sections. For a general view, the part under the doted line shows the training phase that is performed offline. The upper part shows the online phase. In the training stage the *speech processing* and *recognition* techniques are used to obtain the model parameters and the segments of the phonemes in each user utterance. Afterwards, using the model parameters and the segments the feature generation is performed. Next, the *Support Vector Machine* (SVM) classifier produces its own models according to a specific kernel and bit specifications. From all those models, the ones that give the best results per phoneme are selected and form the final SVM model. Finally, using the last SVM model the key is generated. The online stage is very much similar to the training and will repeatedly produce the same key if a user utters the same passphrase.

2 Speech Processing and Phoneme Feature Generation

Firstly, the speech signal is divided into short windows and the *mel frequency cepstral coefficients* (MFCC) are obtained. As a result an n-dimension vector, $(n-1)$-dimension MFCCs followed by one energy coefficient is formed. To emphasize the dynamic features of the speech in time, the time-derivative (Δ) and the time-acceleration (Δ^2) of each parameter are calculated [13].

Afterwards, a forced alignment configuration of an ASR is used to obtain a model and the starts and ends of the phonemes per utterance. The ASR is

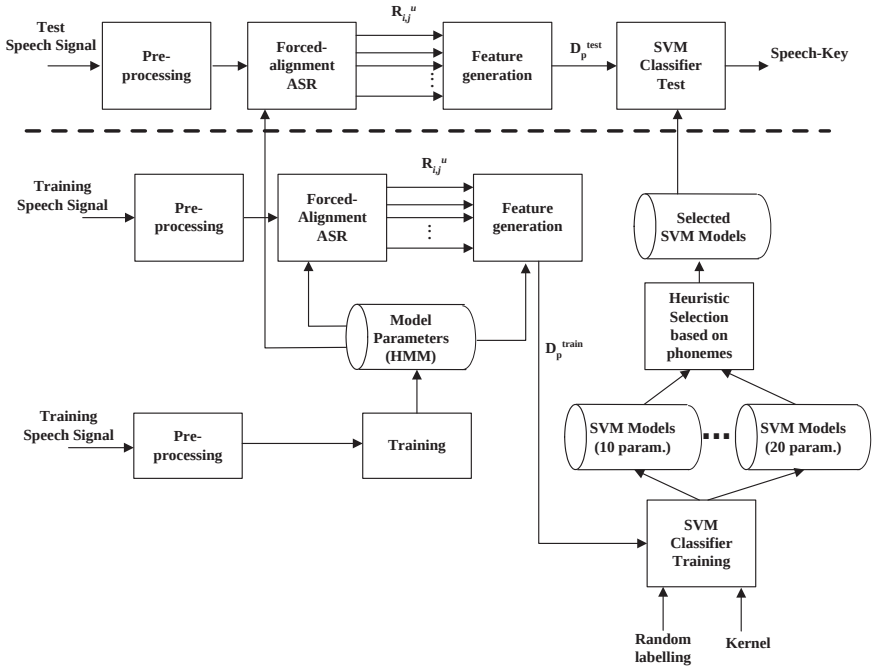

Fig. 1. System Architecture

based on a 3 state, left-right, Gaussian-based continuous Hidden Markov Model (HMM). For this research, the phonemes were selected instead of words since it is possible to generate larger keys with shorter length sentences.

Assuming the phonemes are modelled with a three-state left-to-right HMM, and assuming the middle state is the most stable part of the phoneme representation, let,

$$C_i = \frac{1}{K} \sum_{l=1}^{K} W_l G_l, \tag{1}$$

where G is the mean of a Gaussian, K is the total number of Gaussians available in that state, W_l is the weight of the Gaussian and i is the index associated to each phoneme.

Given the phonemes' starts and ends information, the MFCCs for each phoneme in the utterances can be arranged forming the sets $R^u_{i,j}$, where i is the index associated to each phoneme, j is the j-th user, and u is an index that starts in zero and increments every time the user utters the phoneme i.

Then, the feature vector is defined as

$$\psi^u_{i,j} = \mu(R^u_{i,j}) - C_i$$

where $\mu(R^u_{i,j})$ is the mean vector of the data in the MFCC set $R^u_{i,j}$, and $C_i \in \mathcal{C}_P$ is known as the matching phoneme mean vector of the model. Let us denote the set of vectors,

$$D_p = \{\psi^u_{p,j} \mid \forall\, u, j\}$$

where p is a specific phoneme.

Afterwards, this set is divided in subsets: D^{tr}_p and D^{test}_p. 80% of the total D_p are elements of D^{tr}_p and the remaining 20% form D^{test}_p. Then, $D^{train}_p = \{[\psi^u_{p,j}, b_{p,j}] \mid \forall\, u, j\}$ where $b_{p,j} \in \{-1, 1\}$ is the key bit or class assigned to the phoneme p of the j-th user.

3 Support Vector Machine

The *Support Vector Machine* (SVM) *Classifier* is a method used for pattern recognition, and was first developed by Vapnik and Chervonenkis [1, 3]. For this technique, given the observation inputs and a function-based model, the goal of the basic SVM is to classify these inputs into one of two classes. Firstly, the following set of pairs are defined $\{x_i, y_i\}$; where $x_i \in \mathbb{R}^n$ are the training vectors and $y_i = \{-1, 1\}$ are the labels. The SVM learning algorithm finds an hyperplane (w, b) such that,

$$\min_{x_i, b, \xi} \frac{1}{2} w^T w + C \sum_{i=1}^{l} \xi_i$$

$$\text{subject to } y_i(w^T \phi(x_i) + b) \geq 1 - \xi_i$$

$$\xi_i \geq 0$$

where ξ_i is a slack variable and C is a positive real constant known as a tradeoff parameter between error and margin.

To extend the linear method to a nonlinear technique, the input data is mapped into a higher dimensional space by function ϕ. However, exact specification of ϕ is not needed; instead, the expression known as kernel $K(x_i, x_j) \equiv \phi(x_i)^T \phi(x_j)$ is defined. There are different types of kernels as the linear, polynomial, radial basis function (RBF) and sigmoid.

Although SVM has been used for several applications, it has also been employed in biometrics [12, 11]. In this research, we study just SVM techinque using radial basis function (RBF) kernel to transform a feature, based on a MFCC-vector, to a binary number (key bit) assigned randomly. The RBF kernel is denoted as $K(x_i, x_j) = e^{(-\gamma ||x_i - x_j||^2)}$, where $\gamma > 0$.

For our research, the methodology used to implement the SVM training is as follows. Firstly, the training set for each phoneme (D^{train}_p) is formed by assigning a one-bit random label ($b_{p,j}$) to each user. Since a random generator of the values (-1 or 1) is used, the assignation is different for each user. The advantage of this random assignation is that the key entropy grows significantly. Afterwards, by employing a grid search the parameters C and γ are tuned.

Next, a testing stage is performed using D^{test}_p. This research considers just binary classes and the final key could be obtained by concatenating the bits

produced by each phoneme. For instance, if a user utters two phonemes: */F/* and */AH/*, the final key is $K = \{f(D_{/F/}), f(D_{/AH/})\}$, thus, the output is formed by two bits.

Finally, the SVM average classification accuracy is computed by the ratio

$$\eta = \frac{\alpha}{\beta}. \tag{2}$$

where α is the classification matches on test data and β is the total number of vectors in test data.

It is possible to choose the appropiate SVM model that corresponds to a specific phoneme by making the proper selection of the number of dimensions of the MFCCs. The SVM model should satisfy that the best average classification accuracy is obtained by all users in the SVM classifier outcome for that specific phoneme.

4 Experimental Methodology and Results

The YOHO database was used to perform the experiments [2, 8]. YOHO contains clean voice utterances of 138 speakers of different nationalities. It is a combination lock phrases (for instance, "Thirty-Two, Forty-One, Twenty-Five") with 4 enrollment sessions per subject and 24 phrases per enrollment session; 10 verification sessions per subject and 4 phrases per verification session. Given 18768 sentences, 13248 sentences were used for training and 5520 sentences for testing.

Afterwards, the utterances are processed using the Hidden Markov Models Toolkit (HTK) by Cambridge University Engineering Department [9] configured as a forced-alignment automatic speech recogniser. The important results of the speech processing stage are the twenty sets of mean vectors of the mixture of gaussians per phoneme given by the HMM and the phoneme starts and ends of the utterances. The phonemes used are: /AH/, /AX/, /AY/, /EH/, /ER/, /EY/, /F/, /IH/, /IY/,/K/, /N/, /R/, /S/, /T/, /TH/, /UW/, /V/, /W/. Following the method already described, the D_p sets are formed. It is important to note that the cardinality of each D_p set can be different since the number of equal phoneme utterances can vary from user to user. Next, subsets D_p^{train} and D_p^{test} are constructed. For the training stage, the number of vectors picked per user and per phoneme for generating the model is the same. Each user has the same probability to produce the correct bit per phoneme. However, the number of testing vectors that each user provided can be different.

Following the method shown the key bit assignation is needed. For this research, the assignation is arbitrary. Thus, the keys have liberty of assignation, therefore the keys entropy can be easily maximised if they are given in a random fashion with a uniform probability distribution.

The classification of D_p vectors was performed using SVMlight [15]. The training and the testing stage needed the selection and the tuning of the parameters γ and C. The behaviour of the SVM is given in terms of Equation 2.

After selecting the dimension with the best performance for each of the phonemes, the results for η are depicted in Figure 2.

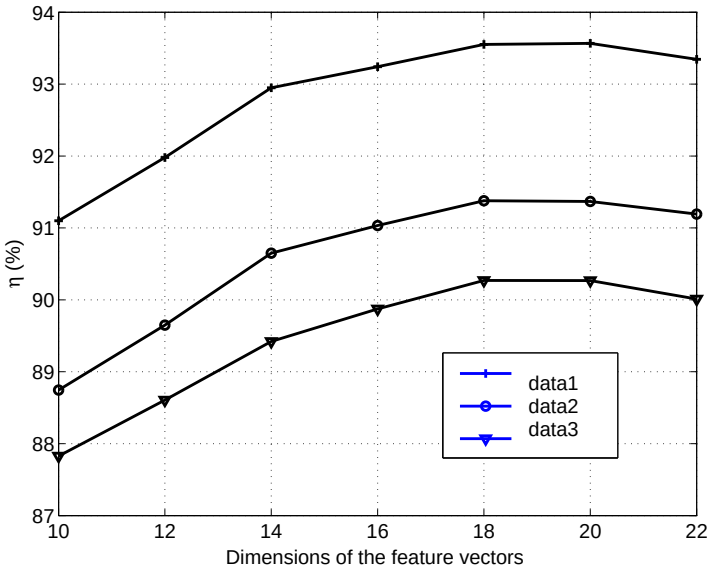

Fig. 2. η for different number of parameters

Number of users	η (%)
10	93.65966316
20	91.60074211
30	90.30173158

Table 1. Global average for η, after the best perfomance models

Figure 2 show the values for η for several number of users. The statistics were computed as follows: 500 trials were performed for 10 and 20 users, and 1000 trails were performed for 30 and 50 users.

As shown in the Tables the increment of the number of MFCC coefficients gives better results than just adjusting to speech known specifications.

5 Conclusion

We have presented a method to improve the generation of a cryptographic key from the speech signal based on the selection of the best performance for each of the phonemes. With this method we obtained an improvement of 24.26%, 18.85%, 16.56% for 10, 20 and 30 speakers, from the YOHO database, respectively, compared with our previous results. In addition, it is important to note that for this task the 18 dimension vector shows better performance than 12 dimension vector which is the most common parameter number used in speech recognition.

For future research, we plan to study the classification techniques, either improving the SVM kernel or by using artifitial neural networks. Moreover, it is important to study the robustness of our system under noisy conditions. Besides, future studies on a M-ary key may be useful to increase the number of different keys available for each user given a fixed number of phonemes in the passphrase.

References

1. Boser, B., I. Guyon, and V. Vapnik. A training algorithm for optimal margin classifiers. In Proceedings of the Fifth Annual Workshop on Computational Learning Theory, 1992.
2. Campbell, J. P., Jr. Features and Measures for Speaker Recognition. Ph.D. Dissertation, Oklahoma State University, 1992.
3. Cortes, C. and V. Vapnik. Support-vector network. Machine Learning 20, 273-297, 1995.
4. Garcia-Perera L. P., C. Mex-Perera and J. A. Nolazco-Flores. Multi-speaker voice cryptographic key generation. Accepted for publication in the 3rd ACS/IEEE International Conference on Computer Systems and Applications - January 2005.
5. Garcia-Perera L. P., C. Mex-Perera and J. A. Nolazco-Flores. Cryptographic-speech-key generation using the SVM technique over the lp-cepstra speech space. INTERNATIONAL SUMMER SCHOOL "NEURAL NETS E. R. CAIANIELLO" IX COURSE as a TUTORIAL RESEARCH WORKSHOP on Nonlinear Speech Processing: Algorithms and Analysis. October 2004.
6. L. Paola Garcia-Perera, Carlos Mex-Perera, and Juan A. Nolazco-Flores. Sent to the 2nd Iberian Conference on Pattern Recognition and Image Analysis, June 7-9, 2005 Estoril, Portugal.
7. Garcia-Perera L. P., C. Mex-Perera and J. A. Nolazco-Flores. SVM Applied to the Generation of Biometric Speech Key A. Sanfeliu et al. (Eds.): CIARP 2004, LNCS 3287, pp. 637-644, 2004. Springer-Verlag Berlin Heidelberg 2004
8. Higgins, A., J. Porter and L. Bahler. YOHO Speaker Authentication Final Report. ITT Defense Communications Division, 1989.
9. Young,S., P. Woodland HTK Hidden Markov Model Toolkit home page. http://htk.eng.cam.ac.uk/
10. F. Monrose, M. K. Reiter, Q. Li , S. Wetzel. Cryptographic Key Generation From Voice. Proceedings of the IEEE Conference on Security and Privacy, Oakland, CA. May, 2001.
11. E. Osuna, R. Freund, and F. Girosi. Support vector machines: Training and applications. Technical Report AIM-1602, MIT A.I. Lab., 1996.

12. E. Osuna, R. Freund, and F. Girosi, Training Support Vector Machines: An Application to Face Recognition, in IEEE Conference on Computer Vision and Pattern Recognition, pp. 130-136, 1997.
13. L.R. Rabiner and B.-H. Juang. Fundamentals of speech recognition. Prentice-Hall, New-Jersey, 1993.
14. U. Uludag, S. Pankanti, S. Prabhakar and A.K. Jain, Biometric cryptosystems: issues and challenges, Proceedings of the IEEE , Volume: 92 , Issue: 6 , June 2004.
15. T. Joachims, SVMLight: Support Vector Machine, SVM-Light Support Vector Machine http://svmlight.joachims.org/, University of Dortmund, November 1999.

6 Acknowledgments

The authors would like to acknowledge the Cátedra de Seguridad, ITESM, Campus Monterrey and the CONACyT project CONACyT-2002-C01-41372 who partially supported this work.