Speaker Verification Using Accumulative Vectors with Support Vector Machines

Manuel Aguado Martínez, Gabriel Hernández-Sierra, and José Ramón Calvo de Lara

Advanced Technologies Application Center, Havana, Cuba
{maguado,gsierra,jcalvo}@cenatav.co.cu

Abstract. The applications of Support Vector Machines (SVM) in speaker recognition are mainly related to Gaussian Mixtures and Universal Background Model based supervector paradigm. Recently, has been proposed a new approach that allows represent each acoustic frame in a binary discriminant space. Also a representation of a speaker - called accumulative vectors - obtained from the binary space has been proposed. In this article we show results obtained using SVM with the accumulative vectors and Nuisance Attribute Projection (NAP) as a method for compensating the session variability. We also introduce a new method to counteract the effects of the signal length in the conformation of the accumulative vectors to improve the performance of SVM.

Keywords: speaker recognition, binary values, accumulative vectors, Support Vector Machine, Nuisance Attribute Projection.

1 Introduction

Currently SVM is one of the most robust and powerful discriminative classifier in speaker recognition. The applications of SVM are mainly related to Gaussian Mixtures and Universal Background Model (GMM/UBM) based supervector paradigm [1, 2]. Generally a supervector is obtained by concatenating the means of the adapted GMM models. However, these approaches show limitations associated with the GMM/UBM paradigm. First, it is difficult to exploit temporal or sequential information. Second, the supervector space don´t allows working directly with discriminant aspects of speaker.

A new approach that attempts to reduce these limitations was proposed in [3]. It deals directly with the speaker discriminant information in a discrete and binary space. Our method to obtain the binary representation and then the accumulative vectors is similar to [4], it only differs on the normalization process used for reducing the susceptibility of the accumulative vectors to the signal length. At this point we introduce a new method for successfully accomplish this task since it shows better performance combined with the SVM.

In this article we used SVM as a classifier to work with the accumulative vectors and then we compared the results with those obtained with the GMM/UBM based supervector paradigm. We also use Nuisance Attribute Projection (NAP) as a method

J. Ruiz-Shulcloper and G. Sanniti di Baja (Eds.): CIARP 2013, Part II, LNCS 8259, pp. 350–357, 2013.

for compensating the session variability because this algorithm intend to reduce the susceptibility of SVM kernel to this problem [5].

This paper is organized as follows. Section 2 explains the process to obtain the accumulative vectors presented in [4]. Section 3 briefly describes SVM paradigm and NAP algorithm. In Section 4 are introduced the proposals: a new method for scaling the accumulative vectors and the use of SVM as a classifier for accumulative vectors. Section 5 describes the experimental setup, presents the results obtained and show advantages and disadvantages of the proposed approach. Finally, section 6 gives some conclusions.

2 Accumulative Vectors

The process to obtain the accumulative vectors is mainly composed by three components. First, a UBM is trained to divide the acoustic space in acoustic classes. Second, a set of Gaussians components is incorporated to each component or acoustic class of the UBM. These components are known as "speaker specificities" and the set of those as generator model. Finally, for an acoustic frame, each speaker specificity is evaluated and it corresponding binary value is established.

The role of the generator model is to highlight the speaker specificities. As mentioned, each acoustic class of the UBM is represented by a set of Gaussian Components. Those specificities are obtained from the adapted models of the training set, by matching the i components of the adapted models to the i component of the UBM. Since the specificities number is assumed to be very large, it is necessary to reduce it, selecting the most important [4]. As a result the number of specificities per acoustic class could not be the same.

For obtaining the binary representation of a given speaker, first we took each acoustic frame and determine its posterior probability related with each Gaussian component of the UBM by a process similar to Maximum a Posteriori (MAP) [6]. Then the K components with the highest probability were selected, the specificities of these components are the ones represented in the binary vector. We use $K = 3$ based on previous results presented in [4]. Then for each component is compute the likelihood of each acoustic frame with all the corresponding specificities. The equations for determine the posterior probability and the likelihood are detail described in [7]. Finally a binary vector is created by set in 1 the components of the vector corresponding to the specificities with the higher likelihood. These are known as the "active components". After that, a binary vector for each acoustic frame is obtained. Pooling these vectors we have a binary matrix that represents a given speaker. The accumulative vector is then obtained by setting the component of the vector corresponding to a given specificity to the number of activations.

3 Support Vector Machines

At the most basic level, SVM is a binary classifier which models the decision boundary between two classes as a separating hyperplane. In the speaker verification, one

class is the vector of the target speaker (labeled as +1), and the other class is composed for the vectors of a background population (labeled as -1). Using this information, SVM intends to find a separating hyperplane that maximizes the margin of separation between these two classes. This is an optimization problem defined by:

$$Min \ \ \frac{1}{2}||w||^2 + C\sum_i \varepsilon_i \tag{1}$$

$$s.a \ \ t_i(w' * x_i - b) - 1 + \varepsilon_i \geq 0 \ \ \forall_i \tag{2}$$

$$\varepsilon_i \geq 0 \ \forall_i \tag{3}$$

where w is a orthogonal vector to the separating hyperplane, x_i is the accumulative vector i, t_i is the class of the vector i, ε_i are the "slack variables" (allows for violations of the constraints since in practice the data is not linearly separable), b and C are constants.

Here $\sum_i \varepsilon_i$ is the penalty or loss function and could be interpreted as a measure of "how bad" the violations are. The constant C controls the tradeoff between penalty and margin.

This optimization problem is solved in a space of dimension higher than the original space, due to the solution is easier to find in it. To achieve this transformation, SVM use a kernel function $K(x_i, x_j)$. The kernel function should satisfy the Mercer condition [8] (The Kernel should be positive semi definite) and therefore can be expressed as:

$$K(x_i, x_j) = \langle \theta(x_i), \theta(x_j) \rangle = \theta(x_i)^T * \theta(x_j) \tag{4}$$

were $\langle \theta(x_i), \theta(x_j) \rangle$ is the inner product of two vectors. Given a test vector y the discriminant function of SVM is given by:

$$f(y) = w^T * \ \theta(y) + b = \sum_{s=1}^{S} a_s t_s \theta(x_s)^T * \theta(y) + b \tag{5}$$

were x_s are the support vectors determined in the optimization process and S is the number of support vectors.

4 Proposed Methods

In order to improve the results obtained with the similarity measure Intersection and Symmetric Difference (ISDS) [4] we propose to use SVM as a classifier of the accumulative vectors. We train a model for each target speaker using its accumulative vector and a set of background vectors.

We first obtained the generator model described in [4] and extracted the accumulative vectors of the target speakers from its corresponding signal. We took a set of background speakers and extract the corresponding accumulative vectors to be used as impostors in the SVM training process. These background speakers are labeled as -1 and are used to train all the target speakers' models.

Something that has negative impact in the use of accumulative vectors with SVM is their direct dependency with the number of frames of their corresponding signals. For that, before feeding accumulative vectors into SVM we transform them by the procedure described in 4.1.

To improve the results obtained with the SVM we use NAP as a technique to compensate the session variability presented in the accumulative vectors. Therefore we apply the NAP transformation to the accumulative vectors used for the SVM training before starting the training process. We assume that the accumulative vectors holds session variability information and we confirm that in the results. The NAP procedure is similar to the presented in [5] and is described in section 4.2. Finally we use a standard linear kernel to train the support vector machines.

4.1 Scaling the Accumulative Vectors

As we explain in section 2, a binary vector is obtained from each acoustic frame of a given signal. As result the numbers of binary vectors extracted from an acoustic signal depends on the length of it. Since the accumulative vectors are obtained from these binary vectors and their represent the number of times that each specificity was activated, the accumulative vectors of two acoustic signals from the same speaker will be very different if one signal is bigger than the other.

To deal with this problem, in [4] each accumulative vector is divided by the sum of the accumulative values in it. But we face a problem with this method: the resulting accumulative values are too small, and therefore, this phenomenon causes loss of significance in the data during the training of SVM.

To address this trouble we propose to divide each accumulative vector for the number of frames of its corresponding signal. As result, each accumulative value will be equal to the average that specificity was activated by frame. Then the new accumulative vectors are obtained by:

$$s_{acc} = \frac{s_{acc}}{N_f} \tag{6}$$

where s_{acc} is the accumulative vector and N_f is the number of frames of its corresponding signal. Then the new accumulative values are not too small, and with this method we outperform the proposal in [4], using SVM.

4.2 Nuisance Attribute Projection (NAP)

Nuisance Attribute Projection is a compensation technique that successfully removes the session variability in SVM supervectors [5, 9] and we use it with accumulative

vectors. This algorithm is not specific to some kernel and can be applied to any kind of supervectors.

NAP makes the hypothesis that the channel variations tend to lie in a low-dimensional subspace of a speaker s and projecting out these dimensions, most of the speaker-dependent information in s will be unaffected. This transformation is achieved by:

$$s'_{acc} = s_{acc} - U(U^t s_{acc}) \tag{7}$$

where U is the projection matrix and s_{acc} is the accumulative vector of a given speaker. By orthonormality this transformation is idempotent [9]. This means that it is not necessary to transform the test accumulative vectors.

To obtain the U matrix we need a dataset with several speakers and several sessions for each one of them. With this dataset the procedure to obtain U is the following:

1. Extract an accumulative vector of dimension D_{acc} for each session of the training set.
2. Scale these accumulative vectors using the method described in 4.1.
3. For each speaker, calculate the mean accumulative vector and then subtracts this mean from all of its accumulative vectors. Pooling all these accumulative vectors is obtained a large matrix D.
4. Now perform a Principal Component Analysis (PCA) on D to obtain the D_{NAP} principal eigenvectors.
5. The result matrix is the projection matrix U.

In the matrix D most of the speaker variability presented in the accumulative vectors has been removed, however it holds the intersession variability.

5 Experiments

For all the signals used in the experiments we extract 19 Linear Frequency Cepstral Coefficients (LFCC) with the log energy. We add 20 delta coefficients and 10 delta-delta coefficients for a total of 50 features.

A UBM with 512 components was trained using 1661 speakers from NIST SRE 2005. Using this, we train the generator model as was described in [4] with 2450 multilingual signals of 124 speakers from NIST SRE 2004. This set of signals also was used to estimate the NAP projection matrix. To train the SVM we use a subset of 500 signals from the ones selected from NIST SRE 2004.

We use 0.001 as activation threshold to obtain the accumulative vectors. This means that a position in a binary vector was set to 1 if its corresponding likelihood is bigger than this value.

For the test we use det7 core condition test of NIST SRE 2008. This test has 1270 target speakers and 2528 unknown signals. We make 6615 verifications based on this test.

5.1 Results

Firstly we conduct a set of experiments without channel compensation to adjust the parameter C. The results are shown in Table 1.

Table 1. Equal Error Rate (EER) and Detection Cost Function (DCF) results for speaker verification using SVM without channel compensation to adjust the C parameter

C	EER	DCF
100	10.022%	0.0491
250	8.428%	0.0441
500	7.561%	0.0416
750	7.742%	0.0406
1000	7.742%	0.0405
2500	7.289%	0.0407
5000	7.289%	0.0403
6000	7.253%	0.0405
7500	**7.061%**	0.0399
8000	7.205%	0.0398
9000	7.289%	0.0401
10000	7.289%	**0.0395**

For highest values of C we can see a stable behavior in EER. Although the better performance was obtained for C = 7500. We choose this value to adjust the dimension of the NAP matrix projection.

Table 2. Equal Error Rate (EER) and Detection Cost Function (DCF) results for speaker verification using SVM with channel compensation for different dimension of the NAP projection matrix and C=7500

D_{NAP}	EER	DCF
40	6.735%	0.0350
60	6.378%	0.0355
100	6.525%	0.0343
200	6.378%	0.0345
350	6.169%	**0.0325**
450	6.039%	0.0329
550	6.150%	0.0331
600	**5.975%**	0.0337
700	6.150%	0.0362

In Table 2 we show that the best result of our system is obtained with the dimension of NAP projection matrix equal to 600 for C=7500. Therefore the use of NAP improves the results of the SVM by about of 1%. It proves that the accumulative vectors holds information related to session variability.

Finally for comparison purposes we select and develop two different experiments. First an experiment with the similarity measure ISDS [4] was conducted. We apply the normalization of the accumulative vectors described in [4] because ISDS is adjusted to work with this method.

At last a set of experiments using the state of art algorithm i-vector [10] with the compensation techniques Within Class Covariance Normalization (WCCN) [11] and Linear Discriminant Analysis (LDA) [10] was presented. The estimations of the matrixes associated with this experiment uses NIST 2004 speaker set previous described. The rank of the i-vector total variability matrix was equal to 400 and the LDA dimension was set in 390.

Table 3. Equal Error Rate (EER) and Detection Cost Function (DCF) comparison of our proposal with others

D_{NAP}	EER	DCF
SVM C=7500 $D_{NAP} = 600$	5.975%	0.0337
ISDS	11.690%	0.0486
i-vector	7.092%	0.0309
i-vector + LDA	5.828%	0.0290
i-vector+WCCN	6.655%	0.0286
i-vector+WCCN+LDA	5.922%	0.0283

Table 3 shows that our proposal outperforms the similarity measure ISDS and therefore the base line of the accumulative vectors. Also the results are very close to those obtained with the better techniques of the state of art applied to the GMM/UBM based supervector paradigm. Although the experiments only show a slight improvement on a single dataset, the proposed approach seems promising due to that the accumulative vectors paradigm is relatively new, just like its previously mentioned possibility of working directly with the discriminative information of a speaker.

A major drawback of the realized experiments is that we only have one sample of each target speaker and therefore we trained his corresponding SVM model with the accumulative vector extracted from that signal. The use of more than one sample should improve the results obtained. Also the selection of the background signals used to train the SVMs is very crucial. Nevertheless the low cost process of training and scoring the SVM models, its high discriminative power and the advantages relative to the accumulative vectors paradigm, compensate the mentioned disadvantages.

6 Conclusions and Future Work

In this paper we introduce a new scaling method of accumulative vectors because we obtain better results using SVM with it than with the reported in [4]. The obtained results prove that this method successfully removes the effects of signal length in the accumulative vectors. We also show that the accumulative vectors hold information about the session variability and it can be reduced by applying the NAP compensation technique. We demonstrate also that the results obtained with our proposal are much

closed to those obtained with the state of art techniques but using a binary representation of a speaker that allows working directly with it discriminative characteristic and temporal information.

In the future we will try to use others compensation techniques instead on NAP to remove the session variability in the accumulative vectors and make a comparison with the results obtained, just like, use PLDA instead of LDA for comparison purpose. Also in [4] was proposed a trajectory model that represents the temporal information of a speaker and extract more than one accumulative vector, so we intend in futures work to exploit the information relative to this model by using SVM. Furthermore we intend to run more experiments in different datasets to enhance the robustness of our proposal.

References

1. Campbell, W., et al.: Support vector machines for speaker and language recognition. Computer Speech and Language 20, 210–229 (2006)
2. Campbell, W., Sturim, D., Reynolds, D.: Support vector machines using GMM supervectors for speaker verification. IEEE Signal Processing Letters 13, 308–311 (2006)
3. Anguera, X., Bonastre, J.F.: A novel speaker binary key derived from anchor models. In: Proc. Interspeech, pp. 2118–2121 (2010)
4. Hernández-Sierra, G., Bonastre, J.-F., Calvo de Lara, J.R.: Speaker recognition using a binary representation and specificities models. In: Alvarez, L., Mejail, M., Gomez, L., Jacobo, J. (eds.) CIARP 2012. LNCS, vol. 7441, pp. 732–739. Springer, Heidelberg (2012)
5. Solomonoff, A., Campbell, W.M., Boardman, I.: Advances in Channel Compensation for SVM speaker Recognition. In: Proc. Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP), pp. 629–632 (2005)
6. Hautamaki, V., Kinnunen, T., Karkkainen, I., Tuononen, M., Saastamoinen, J., Franti, P.: Maximum a Posteriori estimation of the centroid model for speaker verification (2008)
7. Bonastre, J.F., Bousquet, P.M., Matrouf, D.: Discriminant binary data representation for speaker recognition. In: Proc. ICASSP, pp. 5284–5287 (2011)
8. Cristianini, N., Shawe-Taylor, J.: Support Vector Machines (2000)
9. Campbell, W., Sturim, D., Reynolds, D.: SVM based speaker verification using a GMM supervector kernel and NAP variability compensation. In: Proc. Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP), pp. 637–640 (2005)
10. Dehak, N., et al.: Front-End Factor Analysis For Speaker Verification. IEEE Transactions on Audio, Speech and Language Processing 19(4), 788–798 (2010)
11. Hatch, A.O., Kajarekar, S., Stolcke, A.: Within-Class Covariance Normalization for SVM-based Speaker Recognition. In: Proc. ICSLP, pp. 1471–1474 (2006)