Improving Robustness of Speaker Verification Against Mimicked Speech

Kuruvachan K. George[1], *C. Santhosh Kumar*[1], *K. I. Ramachandran*[1], *Ashish Panda* [2]

[1]Machine Intelligence Research Lab.,
Department of Electronics and Communication Engineering,
Amrita School of Engineering, Coimbatore, Amrita Vishwa Vidyapeetham University, India
[2]TCS Innovation Labs., Mumbai, India
{kg_kuruvachan, cs_kumar, ki_ram}@cb.amrita.edu, ashish.panda@tcs.com

Abstract

Making speaker verification (SV) systems robust to spoofed/mimicked speech attacks is very important to make its use effective in security applications. In this work, we show that using a proximal support vector machine backend classifier with i-vectors as inputs (i-PSVM) can help improve the performance of SV systems for mimicked speech as non-target trials. We compared our results with the state-of-the-art baseline i-vector with cosine distance scoring (i-CDS), i-vector with a backend SVM classifier (i-SVM) and cosine distance features with an SVM backend classifier (CDF-SVM) systems. In i-PSVM, proximity of the test utterance to the target and non-target class is the criteria for decision making while in i-SVM, the distance from the separating hyperplane is the criteria for the decision. It was seen that the i-PSVM approach is advantageous when tested with mimicked speech as non-target trials. This highlights that proximity to the target speakers is a better criteria for speaker verification for mimicked speech. Further, we note that weighting the target and non-target class examples helps us further fine tune the performance of i-PSVM. We then devised a strategy for estimating the weights for every example based on its cosine distance similarity with respect to the centroid of target class examples. The final i-PSVM with example based weighting scheme achieved an improvement of 3.39% absolute in EER when compared to the best baseline system, i-SVM. Subsequently, we fused the i-PSVM and i-SVM systems and results show that the performance of the combined system is better than the individual systems.

1. Introduction

Speaker verification (SV) is the process of verifying the identity claim of a person from his/her spoken utterance. Improving the robustness of speaker verification (SV) systems against spoofed/mimicked speech is extremely important to make its use in critical applications effective [1]. Spoofing is a technique in which the data content in the speech utterance is modified or corrupted, maintaining the acoustic characteristics of the speech signal [2, 3]. Spoofing attacks are mainly of four types: mimicry, replay, speaker adapted speech synthesis and voice conversion [1]. In this work, we emphasize on intentional speech modifications caused by mimicking the target speaker's speech. Lack of a standard speaker recognition evaluation database with adequate number of impersonators mimicking the voices of the target speakers is a major bottleneck for pursuing research in this direction.

We created a mimicry database, Amrita Speaker Recognition Evaluation (SRE) Database [2], with 115 target speakers and 76 impersonators mimicking the voices of the target speakers to benchmark the performance of speaker verification algorithms against mimicked speech test conditions. We compared the performance of different state-of-the-art SV systems, i-vector with cosine distance scoring (i-CDS) [4], i-vectors with a backend maximum margin support vector machine classifier (i-SVM) [5, 6] and cosine distance features (CDF) with a backend SVM classifier (CDF-SVM) [7] systems developed using Amrita SRE Database [2]. We also compared the effectiveness of different short term cepstral features, mel frequency cepstral coefficients (MFCC), power normalized cepstral coefficients (PNCC) [8] and delta spectral cepstral coefficients (DSCC) [9], and found that MFCC outperforms other features when tested with mimicked speech. From the experimental results [2], it was seen that the impersonator mimicking the target speakers caused significant degradation in the performance of all speaker verification systems when compared with the systems developed using the target and non-target trials as in NIST SRE, without mimicked voices. In a subsequent work [10], we evaluated the advantage of gammatone frequency cepstral coefficients (GFCC) [11] as an input feature in reducing the false alarm probabilities (FAP) of SV systems. From the experimental results it was observed that the CDF-SVM with an intersection kernel developed using GFCC achieves the minimum FAP. However, the best overall performance was obtained with MFCC as input features and hence we use MFCC for all experiments in this work.

We focus on enhancing the robustness of SV systems against mimicked speech utterances as non-target trials using i-vectors as inputs to a proximal support vector machine backend classifier (i-PSVM). In i-PSVM [12, 13], proximity to the target and non-target class is the criteria for decision making while in i-SVM the distance from the separating hyperplane is the criteria for the decision. It was seen that the i-PSVM approach is advantageous when tested with mimicked speech as non-target trials. In this context, we use PSVM instead of separating margin SVM as a backend classifier and experiment using different weighting factors for the target and non-target speaker classes. We then present the details of the i-PSVM and compare its performance with the state-of-the-art baseline SV systems. Later, we refine and devise an algorithm for an example based weighting factor for further enhancing the performance of i-PSVM. Weights for each example are calculated based on its cosine distance similarity with respect to the centroid of the target class examples. The i-PSVM with example based weighting scheme achieved an improvement of 4.93%, 3.39%, and 4.44% absolutes in EER when compared to the baseline i-CDS, i-SVM and CDF-SVM systems respectively. Finally, we fused the i-PSVM and i-SVM to obtain an additional improvement of 0.51% absolute in EER, making an overall performance improvement of 3.90% absolute in EER compared to the best baseline, i-SVM.

The rest of the paper is organized as follows. Section 2 discusses SVMs, which also includes a brief description of maximum margin SVM, PSVM and PSVM with custom c. A brief review of the state-of-the-art systems, i-CDS, i-SVM, i-PSVM, CDF-SVM and CDF-PSVM is provided in section 3. The details of Amrita SRE Database is presented in section 4. In section 5, experiments and results are discussed and finally, section 6 concludes.

2. Support Vector Machines

Support Vector Machines (SVM) [12, 13] are supervised learning algorithms used for solving binary classification problems. SVMs find the optimum hyperplane by maximizing the separating margin between the two classes, where the margin is the sum of distances from the hyperplane to the nearest data points belonging to each of the two separate classes. A detailed description on the formulation of maximum margin SVM, proximal SVM (PSVM) and PSVM with custom c are described in what follows. We refer to the maximum margin SVM as simply SVM throughout this work. For simplicity, we elaborate only on using linear kernels, and extending the formulation to non-linear kernels is trivial and is not discussed in this work [12, 13], though we used non-linear kernels throughout.

2.1. Maximum Margin SVM

Consider a given training set,

$$S = \{x_i, y_i\}_{i=1}^{m+n} \tag{1}$$

where, $x_i \in R^k$, $y_i \in \{-1, +1\}$ and $A = (x_1, x_2, ..., x_m, x_{m+1}, ..., x_{m+n})^T$ is the training data matrix in which the first m examples correspond to the target class and the next n examples correspond to the non-target class. The objective of SVM is to find the optimum seperating hyperplane,

$$\begin{aligned} &\underset{w,\xi}{min}\ \frac{1}{2}w^Tw + \frac{c}{2}\xi^T\xi \\ &subject\ to:\ D(Aw - eb) \geq e - \xi \\ &\qquad\qquad\qquad \xi \geq 0 \end{aligned} \tag{2}$$

where, $e = (1, 1, ..., 1)^T$, $\xi = (\xi_1, \xi_2, ..., \xi_m, \xi_{m+1}, ..., \xi_{m+n})$ is the non-negative slack variables to account for the misclassified examples, D is a diagonal matrix with its non-zero entries are given by, $D_{ii} = y_i$ and c is a positive scalar weighting factor for examples.

Solving the minimization problem in (2) the optimum values for w and b is obtained. Finally, for an unseen example, x, the decision function is given by,

$$f(x) = sign(w^Tx - b) \tag{3}$$

where,

$$\begin{aligned} &x \in +1,\ if\ f(x) > 0 \\ &x \in -1,\ if\ f(x) < 0 \end{aligned}$$

2.2. Proximal Support Vector Machines

Mangasarian and Fung [12] introduced a fundamental change in the formulation given in (2) by replacing the inequality constraint with an equality as follows:

$$\begin{aligned} &\underset{w,b,\xi}{min}\ \frac{c}{2}\xi^T\xi + \frac{1}{2}(w^Tw + b^2) \\ &subject\ to:\ D(Aw - eb) = e - \xi \end{aligned} \tag{4}$$

This formulation is equally good as the classical formulation with some added advantages such as strong convexity of the objective function [13]. Fig. 1 contrasts the difference between the classical SVM and PSVM. It may be seen that, in the case of classical SVM the bounding hyperplanes are passing through the support vectors while it passes through the data centroids of the corresponding classes for PSVM.

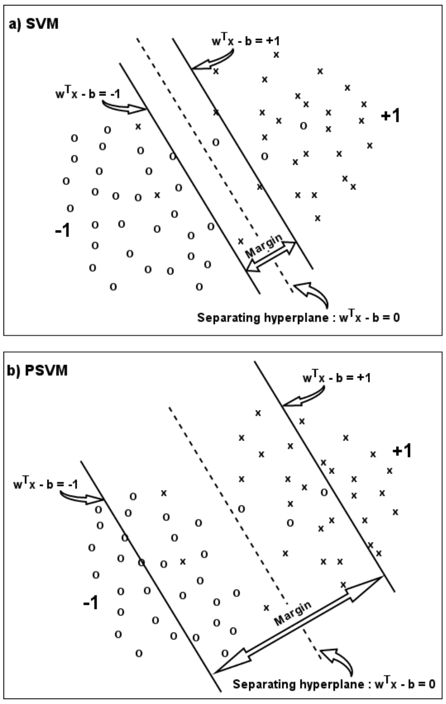

Figure 1: *a) Classical SVM : The bounding hyperplanes are passing through the support vectors, b) PSVM : The bounding hyperplanes are passing through the data centroids of the corresponding classes*

The Lagrangian function for this problem may be obtained as [13]:

$$L(w, b, \xi, u) = \frac{c}{2}\xi^T\xi + \frac{1}{2}(w^Tw + b^2) - u^T(D(Aw - eb) + \xi - e) \tag{5}$$

The KKT optimality conditions for the optimization problem is obtained by setting the derivatives of the above Lagrangian function with respect to (w, b, ξ, u) to zero and substituting the expressions for w, b, ξ in to the constraint equation given in (4), we have,

$$\begin{aligned} &D(AA^TDu + ee^TDu) + \frac{u}{c} = e \\ &\therefore u = (\frac{I}{c} + D(AA^T + ee^T)D)^{-1}e \end{aligned} \tag{6}$$

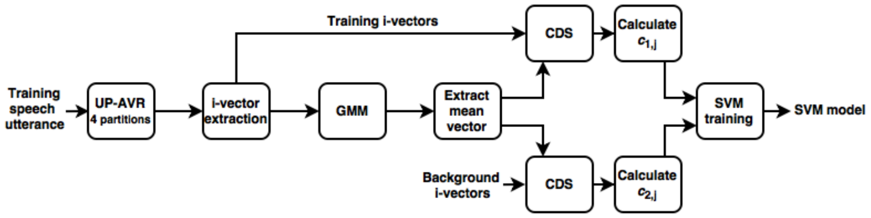

Figure 2: *Block diagram describing the PSVM model training with custom c values for each training example.*

Having u from (6), the optimum values of w and b may be obtained and for an unseen data point, x, the decision can be found using (3).

2.3. PSVM with custom c

In the original PSVM formulation by Mangasarian, et al. [12], examples belonging to the target and non-target classes are treated with equal importance by setting the same c for all training examples. In [14], it is reported that, treating every data point equally in the training process may cause unsuitable overfitting in SVMs.

In this work, we reformulate the PSVM training to provide a means to vary the value of c independently for the positive and negative classes. We are replacing the parameter c in (4) by $\mathbf{C}$, where, $\mathbf{C}$ is the diagonal matrix with its first m diagonal entries are the weights corresponding to the examples of target class, c_1, and the next n diagonal entries are the weights corresponding to the non-target class examples, c_2, we obtain the new objective function and the constraints as:

$$\begin{aligned} &\underset{w,b,\xi}{min}\ \frac{1}{2}\xi^T\mathbf{C}\xi + \frac{1}{2}(w^T w + b^2) \\ &subject\ to:\ D(Aw - eb) = e - \xi \end{aligned} \tag{7}$$

The Lagrangian function for this problem may be obtained as:

$$L(w,b,\xi,u) = \frac{1}{2}\xi^T\mathbf{C}\xi + \frac{1}{2}(w^T w + b^2) - u^T(D(Aw - eb) + \xi - e) \tag{8}$$

The KKT optimality conditions for the expression in (7) is obtained by setting the derivatives of the above Lagrangian function with respect to (w, b, ξ, u) to zero, we have:

$$\begin{aligned} \frac{\partial L}{\partial w} &= 0 \Rightarrow w = A^T Du \\ \frac{\partial L}{\partial b} &= 0 \Rightarrow b = -e^T Du \\ \frac{\partial L}{\partial \xi} &= 0 \Rightarrow \xi = \mathbf{C}^{-1}u \end{aligned} \tag{9}$$

Substituting the expressions for w, b, ξ in (9) to the constraint equation given in (7), we have,

$$\begin{aligned} &D(AA^T Du + ee^T Du) + \mathbf{C}^{-1}u = e \\ &\therefore u = (\mathbf{C}^{-1} + D(AA^T + ee^T)D)^{-1}e \end{aligned} \tag{10}$$

Having u from (10), the optimum values of w and b may be obtained from (9). Therefore, for an unseen data point, x, the classification decision is obtained by (3).

Initially, the optimum model parameters are obtained by empirically calibrating the values of c_1 and c_2 for target and non-target classes. The best values of c for each training example is then estimated algorithmically. The algorithmic estimation of c is described in what follows:

2.3.1. Estimating the example based weights

In Fig. 2, the process of PSVM model training with custom c values for each training example is described. The single speech utterance available for training the speaker model [15] is first subjected to utterance partitioning with acoustic vector resampling (UP-AVR) [6] to generate four sub-utterances. i-vectors extracted from the sub-utterances are then used for training a Gaussian mixture model (GMM) with a single Gaussian and the mean vector extracted from the GMM is further used as a reference i-vector. Subsequently the cosine distance score (CDS) [4] of the reference i-vector with each of the target and non-target class i-vectors are calculated. The CDS values of each of the i-vectors are then used for obtaining the weighting factor for each of the examples as:

$$c_{1,j} = \alpha\left(1 - \frac{< ivec_{ref}, ivec_j >}{\| ivec_{ref} \|\| ivec_j \|}\right) \tag{11}$$

$$c_{2,j} = \beta\left(1 - \frac{< ivec_{ref}, ivec_j >}{\| ivec_{ref} \|\| ivec_j \|}\right) \tag{12}$$

where, $c_{1,j}$ and $c_{2,j}$ corresponds to the c values of $ivec_j$, j^{th} i-vector in the target and non-target classes respectively with respect to the reference i-vector $ivec_{ref}$ of the target class. The value of α and β were selected empirically to compensate for the data imbalance problem, and the best values are reported in Section 5.

It may be noted that the examples far away from the target class will have a higher weighting factor compared to an example closer to the target class. Therefore the centroid of target class is not modified significantly while the centroid of the non-target class is pushed away from the centroid of the target class and hence the discrimination between the two classes is improved.

3. System Description

In this work, we developed speaker recognition systems with the state-of-the-art i-vector with cosine distance scoring (i-CDS), i-vectors as inputs to a backend support vector machine (i-SVM) and proximal support vector machine classifiers (i-PSVM), cosine distance features (CDF) with a backend SVM and PSVM classifiers (CDF-SVM and CDF-PSVM). A brief description of i-CDS, i-SVM, i-PSVM, CDF-SVM and CDF-PSVM systems is briefly explained in what follows:

Table 1: The specifications of Amrita SRE Database [2]

	English	Hindi	Kannada	Malayalam	Tamil	Telugu	Total
Target Speakers	12	17	20	22	14	30	115
Impersonators	20	25	5	7	11	8	76
Target Models	104	141	132	128	117	193	815
Target Trials	1285	996	1004	1167	1000	1542	6994
Non-target Trials	1066	735	355	413	350	1057	3976

3.1. i-CDS

Cosine distance scoring [4] is the most popular and widely used scoring paradigm in the i-vector framework for its computational efficiency. In this approach, the decision score is calculated as the cosine distance between the target i-vector and the test i-vector [4]. The CDS may be obtained by,

$$score(ivec_{tar}, ivec_{test}) = \frac{< ivec_{tar}, ivec_{test} >}{\| ivec_{tar} \| \| ivec_{test} \|} \quad (13)$$

where target speaker i-vector and test speaker i-vector are denoted by $ivec_{tar}$ and $ivec_{test}$ respectively. It may be noted that, i-CDS does not require a target model training as in SVM systems.

3.2. i-SVM and i-PSVM

In i-SVM and i-PSVM systems, i-vectors are given as inputs to a backend SVM or PSVM classifiers [5, 14]. According to the NIST speaker recognition evaluation (SRE) criteria [15], for each training model a single speech utterance is available. Therefore, the number of target class examples are much fewer than the non-target class examples during the model training and thereby a significant degradation in the performance [6] of the i-SVM/i-PSVM is observed when compared to i-CDS. This data imbalance problem is tackled by using utterance partitioning with acoustic vector resampling (UP-AVR) [6] algorithm in which the single training utterance is split into sub-utterances prior to the i-vector extraction. In this work, four sub-utterances are generated from each of the training utterances by repeating the frame index resampling and utterance partitioning. Further the i-vectors corresponding to the sub-utterances as target examples and that of the background utterances as non-target examples are used for training the SVM or PSVM models. The decision score is generated by comparing the test i-vectors with the SVM models. It was seen that i-PSVM outperforms all other state-of-the-art speaker verification systems when tested using the mimicked speech database, Amrita SRE Database.

However, it may be noted that in SVM and PSVM, the training data belonging to both target and non-target classes are given equal importance by setting a constant scalar, c, as a weighting factor for the slack variable, ξ. In this work, we reformulate the PSVM as described in 2.3, by varying the value of c independently for the target and non-target training examples to further enhance the robustness of i-PSVM system when tested with mimicked speech. It is observed from the experimental results that, using the custom c values during the PSVM training significantly improves the robustness of i-PSVM when mimicked voices of the target speakers are used as non-target trials.

3.3. CDF-SVM and CDF-PSVM

In this approach, cosine distance similarities of an i-vector with a set of predefined reference i-vectors are used as features, cosine distance features (CDF) [7], to a backend SVM or PSVM classifier (CDF-SVM or CDF-PSVM). UP-AVR is applied only on the training utterances and hence the background and test speakers are represented by single i-vectors.

The j^{th} element of the CDF vector for the i-vector $ivec_{in}$ is calculated as:

$$CDF(j) = \frac{< ivec_{in}, ivec_{cb}(j) >}{\| ivec_{in} \| \| ivec_{cb}(j) \|} \quad (14)$$

where, $ivec_{cb}(j)$ denotes the j^{th} reference speaker's i-vector in the codebook.

Further, normalization of CDF vector is performed and M largest non-zero values are retained in every CDF vectors while the remaining $N - M$ smallest elements are truncated to zero. where, the total number of reference i-vectors is denoted by N. The final CDF vector obtained is sparse since $M \ll N$.

The final decision score is obtained by comparing the CDF vector corresponding to the test utterance with that of the trained SVM or PSVM model.

4. Amrita SRE Database

Enhancing the robustness of speaker verification systems when tested against mimicked speech of target speakers as non-target trials is extremely important for its use in critical applications. However, speech mimicry test environments are less explored due to the nonavailability of a standard speaker recognition database with mimicked speech utterances of the target speakers.

Amrita SRE Database [2], consists of speech data collected from 115 target speakers and 76 impersonators mimicking the voices of the target speakers. The target speakers and impersonators speak in any of the six different languages: English, Hindi, Malayalam, Kannada, Tamil and Telugu. The specifications of the database is provided in Table 1. In order to ensure that the database meets the existing standards, we made the evaluation criteria same as that of the NIST speaker recognition evaluations (SRE) [15]. A total of 815 target models with 6994 target and 3976 non-target trials are available during the evaluation. The training and test utterances are of an average 5 minutes and 30 seconds durations respectively. Apart from the training and test data, a total number of 2577 speech utterances of 5 minutes duration were also collected as part of this database to be used as the development data. The development data does not contain any speech utterances collected from the speakers involved in the training or testing. All the speech utterances of Amrita SRE Database are saved in single channel 16-bit PCM format at a sampling frequency of 8000 Hz.

5. Experiments and Results

All experiments in this work were performed using the Amrita SRE database. A spectral matching based voice activity

Table 2: The performances of i-CDS, i-SVM, i-PSVM, CDF-SVM and CDF-PSVM systems developed using MFCC are compared when tested with mimicked speech.

System	Kernel	EER
i-CDS	-	21.99
CDF-SVM	Linear	21.93
	Polynomial	31.82
	RBF	33.63
	Sigmoid	34.43
	Intersection	21.5
i-SVM	Linear	20.45
	Polynomial	21.06
	RBF	22.5
	Sigmoid	22.27
	Intersection	20.92
i-PSVM	Linear	21.36
	Polynomial	32.54
	RBF	**18.81**
	Sigmoid	36.86
	Intersection	21.96
CDF-PSVM	Linear	20.87
	Polynomial	34.26
	RBF	35.12
	Sigmoid	23.2
	Intersection	20.67

detection (VAD) [16] algorithm was used for removing the silence segments from the speech data. The popular short term cepstral feature, Mel frequency cepstral coefficients (MFCC) is used for developing the i-CDS, i-SVM, i-PSVM, CDF-SVM and CDF-PSVM speaker recognition systems [7, 17]. For extracting MFCC features, a 19 dimensional mel cepstral coefficients together with log energy were computed. The delta and acceleration coefficients were then appended to make the final feature vector dimension 60. Universal background model (UBM) [4] is a Gaussian mixture model (GMM), comprising 512 multivariate Gaussian components, trained using the development dataset. 400 and 200 respectively are the ranks of the total variability and LDA matrices. Training of the total variability matrix, linear discriminant analysis (LDA), and within class covariance normalization (WCCN) were performed using the development dataset [4]. Subsequently, i-CDS, i-SVM, i-PSVM, CDF-SVM and CDF-PSVM systems were trained using the i-vectors derived from the development and the training datasets.

A codebook consisting of 2577 reference speaker models were used for deriving the CDF. The speaker models from the development dataset only are present in the reference codebook. We chose a total of 1500 non-zero values, empirically obtained, in the CDF vector for the best performance, thus making the CDF vector sparse.

In Table 2, the performances of i-CDS, i-SVM, i-PSVM, CDF-SVM and CDF-PSVM systems developed using MFCC are compared when tested with mimicked speech. For the SVM backend systems, the performance with different kernels: linear, polynomial, radial basis function (RBF), sigmoid and intersection are also presented in Table 2. It is observed that the i-PSVM with an RBF kernel outperforms all other systems. Therefore the subsequent experiments with PSVM were performed using RBF kernel.

On comparing the performance of CDF-SVM systems developed in this work with that in [7, 17], we note that the performance of CDF-SVM deteriorates significantly when tested with mimicked speech as non-target trials. Our further analysis on CDF shows that, the CDF values extracted from the target and impersonators' are very close to each other and hence a reduction in the discriminative capability of the CDF. Therefore, it is worth exploring on how to improve the discriminative capability of the CDF to make it robust to mimicked speech. In [2], we found that impersonators can effectively match the formant frequencies of the target speakers. Therefore, a detailed study needs to be done on the attributes that could be utilized effectively to distinguish the mimicked speech from that of the target speaker. Our initial study using phase, group delay and instantaneous frequency for spoofing detection is encouraging to suggest the use of these features to derive CDF to enhance its robustness against mimicked speech attacks [18].

In Table 3, the performance of different i-PSVM systems with: 1) equal weights for all examples, 2) class based weights, 3) example based weights and 4) fused i-SVM + i-PSVM with example based weights are compared. It is seen that the best performance for i-PSVM with different weights for target and non-target classes is obtained for $c_1 = 0.2$ and $c_2 = 1.1$ by manually varying the values of c_1 and c_2. For i-PSVM with example based weights, the best performance of 17.06% EER was achieved for an $\alpha = 16$ and $\beta = 0.80$. It may also be noted that, i-PSVM with example based weights gives the best overall performance when compared with all other individual systems.

Finally, we fused the i-PSVM and i-SVM systems and the results show that the performance of the combined system is significantly better than all the individual systems and hence enhancing its robustness when tested with mimicked speech as non-target trials. We followed a simple fusion technique by weighting individual scores of the two systems separately and the optimum fusion weights are calculated empirically. It was found that the optimum fusion weights for i-SVM and i-PSVM are 0.01 and 0.99 respectively. However it may also be noted that, algorithmically calculating the optimum fusion weights using a set of calibration data can further improve the performance and it is not explored in this work due to the unavailability of calibration data.

In Fig. 3, the performance of i-CDS, i-SVM, CDF-SVM, CDF-PSVM, i-PSVM with equal weights for all examples, i-PSVM with class based wights, i-PSVM with example based

Table 3: Performance of i-PSVM systems with: 1) equal weights for all examples, 2) class based weights, 3) example based weights and 4) fused i-SVM + i-PSVM.

System	Optimum parameters	EER (in %)
i-PSVM with equal weights for all examples	C=0.2	18.81
i-PSVM with different weights for target and non-target classes	c_1=0.2 and c_2=1.1	17.35
i-PSVM with example based weighting	α=16 and β=0.8	17.06
Fused i-SVM + i-PSVM	Fusion weights: 0.01 for i-SVM and 0.99 for i-PSVM	**16.55**

weights and the fused i-SVM+i-PSVM with example based weights are compared.

ALIZE toolkit [19] was used for training the UBM and i-vector extraction. We used LIBSVM [20] and PSVM [12] for all our experiments with SVM. The performance of different systems developed in this work were evaluated using the detection error tradeoff (DET) curves [21]. The equal error rate (EER) was used for comparing the systems and was calculated according to the NIST speaker recognition evaluation (SRE) criteria [1, 15, 21].

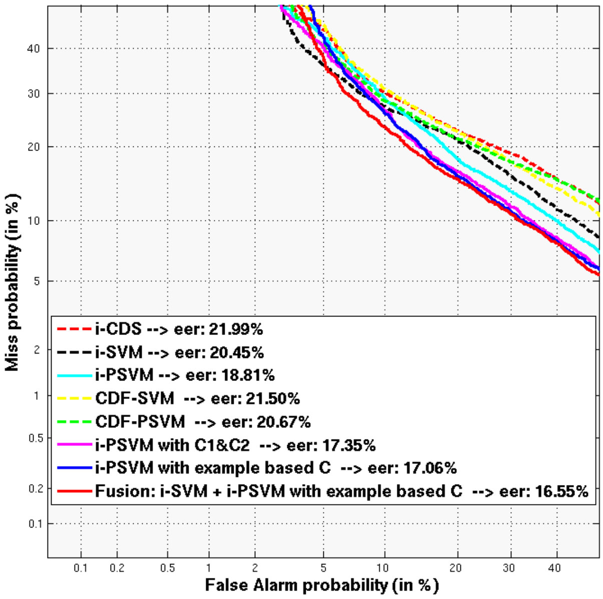

Figure 3: *Performance comparison of i-CDS, i-SVM, CDF-SVM, CDF-PSVM, i-PSVM with equal weights for all examples, i-PSVM with class based wights, i-PSVM with example based weights and the fused i-SVM+i-PSVM with example based weights when tested with mimicked voices of the target speakers as non-target trials.*

6. Conclusion

In this work, we investigated the use of proximal support vector machine backend classifier with i-vectors as inputs (i-PSVM) for enhancing the robustness of speaker verification systems when tested with mimicked voices of the target speakers as non-target trials. From the experimental results on the mimicry database, Amrita SRE Database, it was seen that i-PSVM significantly outperforms the state-of-the-art baseline i-vector with cosine distance scoring (i-CDS), i-vector with a backend SVM (i-SVM) and cosine distance features with an SVM backend classifier (CDF-SVM) systems.

In i-PSVM, proximity to the target and non-target class is the criteria for decision making while in i-SVM the distance from the hyperplane is the criteria. It was seen that the i-PSVM approach is advantageous when tested with mimicked speech as non-target trials. This highlights that the proximity to the target speakers is a better criteria for speaker verification. Further we noted that weighting the target and non-target class examples helps us further improve the performance of the i-PSVM system. We obtained an improvement of absolute 3.10 % in EER for i-PSVM with optimum weights for the target and non-target classes when compared to the best baseline system, i-SVM. We then devised a strategy for estimating weights for each examples based on its cosine distance similarity to the centroid of target class. The i-PSVM with example based weighting scheme achieves an improvement of 4.93%, 3.39%, and 4.44% absolutes in EER when compared to the baseline i-CDS, i-SVM and CDF-SVM systems respectively. Finally, we fused the i-PSVM and i-SVM to obtain an additional improvement of 0.51% in EER, making an overall performance improvement of 3.90% absolute in EER compared to the best baseline, i-SVM.

7. Acknowledgments

Authors would like to thank Tata Consultancy Services (TCS) for supporting the research of Kuruvachan K. George through the TCS Research Scholarship. We also would like to thank Dr. P. Balamuralidhar, TCS Innovation Labs., Bangalore, Aparna S. Raghunath and K. T. Sreekumar for their support during the course of this work.

8. References

[1] R. G. Hautamäki, T. Kinnunen, V. Hautamäki, and A. M. Laukkanen, "Automatic versus human speaker verification: The case of voice mimicry," *Speech Communication*, vol. 72, pp. 13–31, 2015.

[2] K. K. George, C. S. Kumar, K. T. Sreekumar, K. Arun Das, J. T. Alphin, S. K. Meenu and K. I. Ramachandran, "Amrita SRE Database: a database for evaluating speaker recognition systems with mimicked speech," in *O-COCOSDA 2015, Shanghai, China*.

[3] L. Mary, K. A. Babu, and A. Joseph, "Analysis and detection of mimicked speech based on prosodic features," *International Journal of Speech Technology*, vol. 3, pp. 407–417, 2012.

[4] N. Dehak, P. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 4, pp. 788–798, 2011.

[5] W. Rao and M. W. Mak, "Boosting the performance of i-vector based speaker verification via utterance partitioning," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 21, no. 5, pp. 1012–1022, 2013.

[6] M. W. Mak and W. Rao, "Utterance partitioning with acoustic vector resampling for GMM–SVM speaker verification," *Speech Communication*, vol. 53, no. 1, pp. 119–130, 2011.

[7] K. K. George, C.S. Kumar, K.I. Ramachandran, and A. Panda, "Cosine distance features for improved speaker verification," *IET Electronics Letters*, vol. 51, no. 12, pp. 939–941, 2015.

[8] C. Kim and R. M. Stern, "Power-normalized cepstral coefficients (PNCC) for robust speech recognition," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2012, pp. 4101–4104.

[9] K. Kumar, C. Kim, and R. M. Stern, "Delta-spectral cepstral coefficients for robust speech recognition," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2011, pp. 4784–4787.

[10] K. K. George , C. S. Kumar, A. Panda, K. I. Ramachandran, K. Arun Das and S. Veni, "Minimizing the false

alarm probability of speaker verification systems for mimicked speech," in *International Conference on Computing and Network Communications (CoCoNet'15), Thiruvananthapuram, India*.

[11] Y. Shao and D. Wang, "Robust speaker identification using auditory features and computational auditory scene analysis," in *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2008*, pp. 1589–1592.

[12] O. L. Mangasarian and E. W. Wild, "Proximal support vector machine classifiers," in *Proceedings KDD-2001: Knowledge Discovery and Data Mining*, 2001, pp. 77–86.

[13] K. P. Soman, R. Loganathan, and V. Ajay, *Machine learning with SVM and other kernel methods*, PHI Learning Pvt. Ltd., 2009.

[14] J. Ling, "A robust proximal support vector machines for classification," in *International Conference on Neural Networks and Brain*, 2005, vol. 1, pp. 576–580.

[15] "The NIST Year 2008 Speaker Recognition Evaluation Plan," www.itl.nist.gov/iad/mig/tests/sre/2008/sre08_evalplan_release4.pdf, Accessed: 2015-03-20.

[16] K. T. Sreekumar, K. K. George, K. Arunraj, and C. S. Kumar, "Spectral matching based voice activity detector for improved speaker recognition," in *International Conference on Power Signals Control and Computations (EPSCICON)*. IEEE, 2014, pp. 1–4.

[17] K. K. George, C.S. Kumar, K.I. Ramachandran, and A. Panda, "Cosine distance features for robust speaker verification," in *INTERSPEECH 2015, Dresden, Germay*, pp. 234–238.

[18] T. B. Patel and H. A. Patil, "Combining evidences from mel cepstral, cochlear filter cepstral and instantaneous frequency features for detection of natural vs. spoofed speech," in *INTERSPEECH 2015, Dresden, Germay*, pp. 2062–2066.

[19] J. F. Bonastre, F. Wils, and S. Meignier, "ALIZE, a free toolkit for speaker recognition," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2005, pp. 737–740.

[20] C. C. Chang and C. J. Lin, "LIBSVM: a library for support vector machines," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 2, no. 3, pp. 27, 2011.

[21] N. Brümmer and E. De Villiers, "The BOSARIS toolkit: theory, algorithms and code for surviving the new DCF," *arXiv preprint arXiv:1304.2865*, 2013.