SVM based Speaker Recognition: Harnessing Trials with Multiple Enrollment Sessions

Jason Pelecanos[1], *Weizhong Zhu*[1], *Sibel Yaman*[2]

[1]IBM T.J. Watson Research Center, Yorktown Heights, NY, 10598, USA.
[2]Apple Inc., 1 Infinite Loop, Cupertino, CA, 95014, USA.

jwpeleca@us.ibm.com, zhuwe@us.ibm.com, syaman@apple.com

10.21437/Interspeech.2014-173

Abstract

In this paper we extend a variation of the trial-based SVM speaker verification work proposed by Cumani et al to exploit multiple enrollment sessions. Specifically, Cumani proposed the use of a 2nd order SVM kernel for the binary classification of basic trials. In this new work, trials with multiple enrollment sessions are modelled by stacking the i-vectors of the test and enrollment sessions. We further exploit the fact that the score should be independent of the enrollment recording order and present a simplified 2nd order polynomial kernel scoring function accordingly.

In the second part of this work we examine the utility of enrollment pruning for multi-session enrollments. Past work demonstrates that pruning can be beneficial for PLDA based systems. We examine the effects of enrollment pruning in the context of the proposed SVM model.

The results demonstrate that the multi-session enrollment SVM kernel is generally better than the model trained using single sessions. The model is also comparable in performance to the PLDA based approach. Further gains are observed through combination of the PLDA and SVM scores.

Index Terms: speaker recognition, SVM, 2nd order polynomial, multi-session analysis

1. Introduction

Over the last few years, Probabilistic Linear Discriminant Analysis (or PLDA) modelling of i-vectors [1, 2] has become one of the most commonly used approaches for speaker recognition. Recently Cumani [3, 4] showed a relationship between the structure of the scoring function for a two-Gaussian PLDA model and how it can be represented by a kernel with up to 2nd order terms. Their work effectively showed that for single session enrollments the performance of their discriminatively trained Support Vector Machine (SVM) kernel function was similar to the performance of PLDA. It was also shown that these terms may be described by a subset of the terms of a standard 2nd order polynomial kernel SVM [5, 3, 6].

Yaman previously showed that SVMs using higher order polynomial models could be successfully exploited for language identification [7]. This work was followed by trial based modeling for speaker recognition in the dual domain for SVMs [6]. Here a 3rd order polynomial kernel was shown to provide an incremental improvement in performance over the second order.

In comparing the two SVM based approaches (either primal or dual approaches), it should be noted that Cumani's approach is limited by the kernel complexity (or polynomial order) and to a lesser degree the number of training examples. This is because optimization is performed explicitly in the SVM feature space to yield a primal formulation. However, as the polynomial order increases, the feature space dimension increases exponentially. By contrast, the approach by Yaman is limited by the number of training examples and not the polynomial kernel order 'complexity' in the SVM feature space since optimization is performed on the dual formulation.

Other earlier work related to SVMs included training an SVM using a GMM-based kernel for each enrollment model [8]. This requires the use of SVM training during evaluation rather than the techniques discussed here which require a single SVM to be trained once, offline.

In this paper we extend a variation of Cumani's primal optimization approach to include multiple enrollment sessions. (This work is based on the standard 2nd order polynomial kernel instead of a PLDA kernel derivation.) Additionally, we view the problem as attempting to correctly classify an example comprised of stacking the test i-vector and the corresponding set of enrollment session i-vectors. For a second order polynomial kernel, a significant reduction in the computational overhead can be achieved by applying the constraint that the score is independent of enrollment order. We optimize the formulation using stochastic gradient descent on a GPU which also enables the method to be practical for large scale datasets.

After presenting the extended SVM representation, we consider the effect of enrollment pruning on performance. Previous work by McLaren (discussed in more detail in section 3.4) demonstrates that enrollment pruning can improve speaker recognition performance. Stemming from this work, we propose a variation of this technique and apply it to both the PLDA and SVM representations.

The remainder of the paper is as follows. Section 2 provides an outline of the multi-session enrollment SVM formulation building upon the work of Cumani [3, 4]. Section 3 presents the experimental results for the SVM representation and enrollment session pruning which is then followed by the conclusions.

2. Multi-session enrollment

When building a speaker recognition system with multi-session enrollment capabilities, there are not only several ways a system can be trained, but also how speaker trials are scored in evaluation. For example, for the PLDA model, the model itself

This work was supported in part by Contract No. D11PC20192 DOI/NBC under the RATS program. The views, opinions, findings and recommendations contained in this article are those of the author(s) and should not be interpreted as representing the views or policies, either expressed or implied, of the DOI/NBC.

can be trained jointly on multiple instances for each training data speaker. In evaluation, a trial (with multiple enrollment sessions) can also be scored (i) jointly or (ii) the score can be determined by finding average of the pairwise scores between the test segment and each of the enrollment sessions.

Similarly, there are multiple ways to configure the SVM based system. Let us introduce the SVM for the primal formulation [9, 10, 11]. Given an enrollment vector $\boldsymbol{e}$ and a test vector $\boldsymbol{v}$ which can be concatenated to form a training or test example, $\boldsymbol{x} = [\boldsymbol{e}^T \boldsymbol{v}^T]^T$, the speaker recognition score for the SVM may be determined as:

$$f(\boldsymbol{x}) = \boldsymbol{\Phi}(\boldsymbol{x})^T \boldsymbol{w} + b \tag{1}$$

where $\boldsymbol{w}$ is the SVM normal vector to the classification hyperplane in the SVM feature space and b is the SVM offset or bias parameter. Here, $\boldsymbol{\Phi}(\cdot)$ is the kernel expansion vector, with $(^T)$ representing its transpose.

In the primal form, the SVM training function is described by a combination of the square of the L-2 norm cost of the weight vector $\boldsymbol{w}$ and the hinge loss based on the SVM score from Equation 1. Specifically, the SVM soft-margin cost function [11, 10] can be given by:

$$\begin{aligned} \Psi &= \Psi_{L2} + \Psi_{Hinge} && (2) \\ &= \frac{C}{2} \|\boldsymbol{w}\|^2 + \frac{1}{N} \sum_{n=1}^{N} \max(0, 1 - y_n f(\boldsymbol{x_n})) && (3) \end{aligned}$$

where y_n is the output label (either -1 for a non-target or 1 for a target trial) for the given training example, $\boldsymbol{x_n}$. The regularization cost is given as C. The model cost Ψ is to be minimized by optimizing the model parameters $\{\boldsymbol{w}, b\}$.

The method proposed by Cumani [3] addressed the single-enrollment single-test scenario for both training and testing as described above. For the multiple enrollment session scenario, the average score of the test against multiple enrollments would need to be calculated. For multi-session enrollments a drawback is that the model is trained in a different manner to how it is tested and would be sub-optimal.

Alternatively, we extend past work and propose an approach to view an SVM example as the concatenation of the test with all enrollment session statistics. We discuss this in more detail in following sections.

2.1. Cost Function Derivation

In this section we present the optimization details for the proposed multi-session method and also cover some specifics of the previous single-session enrollment approach.

Let us suppose that we have a second order polynomial kernel function with i-vectors described by $\boldsymbol{x}$ and $\boldsymbol{y}$ and scaling parameter ρ:

$$K(\boldsymbol{x}, \boldsymbol{z}) = (\boldsymbol{x}^T \boldsymbol{z} + \rho)^2 \tag{4}$$

This kernel function can be represented as a dot product in the expanded SVM feature space as:

$$K(\boldsymbol{x}, \boldsymbol{z}) = \boldsymbol{\Phi}(\boldsymbol{x})^T \boldsymbol{\Phi}(\boldsymbol{z}) \tag{5}$$

where one possibility for $\boldsymbol{\Phi}(\boldsymbol{x})$ is:

$$\begin{pmatrix} vec(\boldsymbol{x}\boldsymbol{x}^T) \\ \sqrt{2\rho}\, \boldsymbol{x} \\ \rho \end{pmatrix} \tag{6}$$

For simplicity, let us focus exclusively on the second order terms only, i.e. set ρ to 0 and ignore the zero terms. This gives:

$$\boldsymbol{\Phi}(\boldsymbol{x}) = vec(\boldsymbol{x}\boldsymbol{x}^T) \tag{7}$$

Let's first assume that $\boldsymbol{x}$ consists of a single enrollment session trial. The structure for $\boldsymbol{\Phi}$ becomes

$$\begin{aligned} \boldsymbol{\Phi}(\boldsymbol{x}) &= vec\left(\begin{pmatrix} \boldsymbol{e} \\ \boldsymbol{v} \end{pmatrix} \begin{pmatrix} \boldsymbol{e} \\ \boldsymbol{v} \end{pmatrix}^T \right) && (8) \\ &= vec\left(\begin{bmatrix} \boldsymbol{e}\boldsymbol{e}^T & \boldsymbol{e}\boldsymbol{v}^T \\ \boldsymbol{v}\boldsymbol{e}^T & \boldsymbol{v}\boldsymbol{v}^T \end{bmatrix} \right) && (9) \end{aligned}$$

This configuration for $\boldsymbol{\Phi}(\cdot)$ can be used to efficiently calculate the scores across all training example pairs by considering that these elements will be multiplied with the SVM weights vector $\boldsymbol{w}$.

Now, for R enrollment sessions for a speaker, we have the following for the expansion $\boldsymbol{\Phi}(\boldsymbol{x})$:

$$vec\left(\begin{bmatrix} \boldsymbol{e_1 e_1^T} & \cdots & \boldsymbol{e_1 e_R^T} & \boldsymbol{e_1 v^T} \\ \vdots & \ddots & \vdots & \vdots \\ \boldsymbol{e_R e_1^T} & \cdots & \boldsymbol{e_R e_R^T} & \boldsymbol{e_R v^T} \\ \boldsymbol{v e_1^T} & \cdots & \boldsymbol{v e_R^T} & \boldsymbol{v v^T} \end{bmatrix} \right) \tag{10}$$

with $vec(\cdot)$ being the matrix vectorization operator.

It is apparent, for a large number of enrollment sessions per test speaker, that the computational complexity can become quite high. For example, in our scenario which requires the evaluation of 6 enrollment sessions, the size of the matrix is increased from 4 blocks of terms to 49 blocks. This is an increase of more than an order of magnitude in the number of terms.

A very useful approach here is to partition the matrix into its various function blocks and exploit the fact that the speaker recognition score should be independent of the order of the enrollment sessions. Correspondingly, the SVM weights vector $\boldsymbol{w}$ can then be created from the vectorization of 4 matrices $\{\boldsymbol{W_s}, \boldsymbol{W_a}, \boldsymbol{W_{ve}}, \boldsymbol{W_v}\}$:

$$vec\left(\begin{bmatrix} \boldsymbol{W_s} & \boldsymbol{W_a} & \cdots & \boldsymbol{W_a} & \boldsymbol{W_{ve}^T} \\ \boldsymbol{W_a} & \ddots & & \vdots & \vdots \\ \vdots & & \ddots & \boldsymbol{W_a} & \vdots \\ \boldsymbol{W_a} & \cdots & \boldsymbol{W_a} & \boldsymbol{W_s} & \boldsymbol{W_{ve}^T} \\ \boldsymbol{W_{ve}} & \cdots & \cdots & \boldsymbol{W_{ve}} & \boldsymbol{W_v} \end{bmatrix} \right) \tag{11}$$

where $\boldsymbol{W_s}$ is the enrollment session self-scatter weights matrix, $\boldsymbol{W_a}$ is the across enrollment session scatter weights matrix, $\boldsymbol{W_{ve}}$ is the verification against enrollment session scatter weights matrix, and $\boldsymbol{W_v}$ is the verification segment self-scatter weights matrix. Given the enrollment reordering constraint, all matrices are symmetric except for $\boldsymbol{W_{ve}}$. If the test segment set has the same statistical properties as the enrollment segment set, then $\boldsymbol{W_v}$ and $\boldsymbol{W_s}$ can also be set to be equivalent (although we did not do this). If we frame the work of Cumani [3] within the standard 2nd order polynomial kernel structure, it can be observed that there is an additional matrix term relating to the across enrollment session scatter $\boldsymbol{W_a}$.

The SVM score, which is also required to determine the hinge loss, may be calculated as follows:

$$f(\boldsymbol{x_n}) = c_s + c_a + c_{ve} + c_v \quad (12)$$

where

$$c_s = \sum_{r=1}^{R} \boldsymbol{e_r^T W_s e_r} \quad (13)$$

$$c_a = R^2 \bar{\boldsymbol{e}}^{\boldsymbol{T}} \boldsymbol{W_a} \bar{\boldsymbol{e}} - \sum_{r=1}^{R} \boldsymbol{e_r^T W_a e_r} \quad (14)$$

$$c_{ve} = 2R \boldsymbol{v^T W_{ve}} \bar{\boldsymbol{e}} \quad (15)$$

$$c_v = \boldsymbol{v^T W_v v} \quad (16)$$

Here, $\bar{\boldsymbol{e}}$ represents the mean of the R enrollment session vectors for the trial. It is useful to note that once all the calculations involving multiplications of the form $\boldsymbol{We}$ and $\boldsymbol{Wv}$ are determined, it is computationally efficient to determine the final SVM score for the trial. Additionally, if a mini-batch of enrollment sessions and test examples are used, once these statistics have been determined, it is relatively efficient to produce the exhaustive trial scores for test/enrollment combinations within the mini-batch.

The L2-Norm in Equation 3 can be calculated by the summation of Frobenius or vector norms as follows:

$$\|\boldsymbol{w}\|^2 = R\|\boldsymbol{W_s}\|_F^2 + R(R-1)\|\boldsymbol{W_a}\|_F^2 + 2R\|\boldsymbol{W_{ve}}\|_F^2 + \|\boldsymbol{W_v}\|_F^2 \quad (17)$$

2.2. Stochastic Gradient Descent

The equation to be optimized is from Equation 3. We approach this problem by using stochastic gradient descent which involves calculating the slope of this equation for a mini-batch of examples.

For a mini-batch of M trials, we first determine the subset of trials $\{m\}$ that would incur hinge loss based on the hinge loss portion of Equation 3. We then calculate the estimated slopes:

$$\frac{\partial \Psi}{\partial \boldsymbol{W_s}} = CR\boldsymbol{W_s} - \frac{1}{M} \sum_{\forall m} y_m \sum_{r=1}^{R} \boldsymbol{e_{r_m} e_{r_m}^T} \quad (18)$$

$$\frac{\partial \Psi}{\partial \boldsymbol{W_a}} = CR(R-1)\boldsymbol{W_a} - \frac{1}{M} \sum_{\forall m} y_m \left(R^2 \bar{\boldsymbol{e}}_{\boldsymbol{m}} \bar{\boldsymbol{e}}_{\boldsymbol{m}}^{\boldsymbol{T}} - \sum_{r=1}^{R} \boldsymbol{e_{r_m} e_{r_m}^T} \right) \quad (19)$$

$$\frac{\partial \Psi}{\partial \boldsymbol{W_{ve}}} = 2CR\boldsymbol{W_{ve}} - \frac{2R}{M} \sum_{\forall m} y_m \boldsymbol{v_m} \bar{\boldsymbol{e}}_{\boldsymbol{m}}^{\boldsymbol{T}} \quad (20)$$

$$\frac{\partial \Psi}{\partial \boldsymbol{W_v}} = C\boldsymbol{W_v} - \frac{1}{M} \sum_{\forall m} y_m \boldsymbol{v_m v_m^T} \quad (21)$$

At each step, the estimates for the model weights are adjusted accordingly:

$$\boldsymbol{W_{new}} = \boldsymbol{W_{old}} - \triangle \frac{\partial \Psi}{\partial \boldsymbol{W}} \Big|_{\boldsymbol{W}=\boldsymbol{W_{old}}} \quad (22)$$

where $\triangle$ is the step size. In our experiments, we reduce the step size, $\triangle$, by 2% if the cost of the last $200k$ examples (200 sets of 1000 sample mini-batches) processed is 0.1% worse than the previous $200k$ sample blocks. We also note that the Pegasos approach proposed by Shalev-Shwartz [12] could also be used to train this system to provide guaranteed error constraints.

2.3. Optimization Considerations

Past approaches in speaker recognition have viewed the data set as a limited set whereby the system trains over the set *epoch* of examples. This is feasible for data sets that are not too large and only pairwise training is performed. In our scenario, we train the SVM on trials consisting of multiple enrollment examples. It becomes apparent that the number of examples becomes prohibitive for a relatively large number of speakers and training examples per speaker.

Given the vast number of possible examples to model, we approach the problem in a different manner by attempting to obtain a result close to the solution using stochastic gradient descent. Work by Shalev-Shwartz [12] (the method known as Pegasos) showed that for problem formulations of this type, an error bound can be obtained within a certain order of computational time (or samples processed). Pegasos is basically stochastic gradient descent with a zero vector starting point and a scheduled decay in the step size. This algorithm is optimized for single sample at a time adaptation but can also be applied in a more limited manner to mini-batches.

For our problem we generated dynamically (and randomly) created mini-batches for processing on a GPU. If the cost didn't improve after many steps were taken, we reduced the step size slightly. We found that we could obtain a result comparable to Pegasos within a couple of hours.

3. Experiments

In this section we present the experimental results. We begin by discussing the data used and provide an overview of the system.

3.1. Data

The data [13] used in this evaluation consists of the Robust Automatic Transcription of Speech (RATS) data from the DARPA RATS program. It comprises conversational telephone speech recordings passed through eight HF Radio Transmitter-Receiver pairs. The audio consists of five languages: Levantine Arabic, Dari, Farsi, Pashto, and Urdu. There are no cross-gender or cross-language trials. The audio has significant noise and distortions present with some distortions that could be described as amplitude compression effects and frequency shift effects.

Our training set consists of $65k$ segments ranging in duration from $10 - 120$ seconds of speech. There are 390 speakers in this set. For the internal evaluation set, there are $95k$ segments and 314 speakers to produce 1.5 million trials total across eight conditions. There are four conditions shown in these results. Each of the conditions presented here consists of trials formed from segments of approximately equal speech duration with six sessions for enrollment and one for test. The four conditions relate to 120, 30, 10, and 3 seconds of speech per session with 333k, 163k, 173k, and 172k trials correspondingly. We also note that for the 120 second condition (and correspondingly, other conditions) there are 6 enrollment sessions and 1 test session of 120 seconds of speech. More specifically, there is $6 \times 120 = 720$ seconds of enrollment speech and 120 seconds of test speech.

3.2. System Overview

The system developed here is based on similar components documented in our earlier RATS data work [14]. In this particular system, we perform feature extraction using MFCCs (32ms window length, 10ms frame shift), deltas and acceleration coef-

ficients to extract 57 dimensional feature vectors. Session level mean and variance normalization is applied based on the speech frames determined by a speech detection component using energy, voicing and spectral deviation parameters [15, 16]. A 1024 component Gaussian Mixture Model (GMM) [17, 18] is trained from these features. Using Dehak and Kenny's factor analysis formulation we extract a low rank *total variability* representation of the signal [1, 19]. Accordingly, we further reduce the dimensionality from 400 to 200 dimensions using LDA, followed by Within-Class Covariance Normalization (WCCN [20]), and finally unit length normalization (see [21] for information on length normalization effects) to produce the fixed dimension *i-vectors*. These 200 dimensional session representations are used in either the PLDA [2, 1] or the SVM formulations.

3.3. SVM and PLDA comparison

This section presents the performance comparison for the various SVM and PLDA approaches. This discussion refers to the results included in Table 1. Line (a) of Table 1 refers to the regular approach for PLDA training, and in testing, the average score is taken between the test and each of the 6 enrollment sessions. The alternate approach that we have used in evaluations is the approach on Line (b) which performs the regular PLDA training followed by the full hypothesis testing procedure proposed by Prince [2] which directly encompasses the six enrollment sessions in the hypothesis test. Line (c) describes the results for the SVM pairwise (enroll-test) training approach based on the work by Cumani applied to multiple enrollment sessions by averaging the six enroll-test pairs to obtain the final score. Line (d) describes the full joint SVM training approach proposed in this paper. The result on Line (e) presents the performance for the system score combination of the best PLDA and SVM approaches. Here, linear score combination was applied with the weight learned from held-out *Dry-run* data.

There are two observations we cover here. The first is that for both PLDA and SVM, the jointly estimated scores perform better than the average of the six pairwise scores. The second observation is that the SVM approaches seem to perform better for the shorter durations compared with their corresponding PLDA systems.

In summary, the proposed jointly estimated SVM system performs better than the pairwise SVM approach across the four task durations. This SVM is comparable in performance to the jointly estimated PLDA approach for long durations and better than PLDA for the short durations. This result emphasizes the importance of training a system based on a criterion as close as possible to the target objective; that is, to recognize a speaker given all six enrollment sessions together.

3.4. Enrollment Pruning

This section discusses the results of enrollment segment pruning. The technique applied here only applies to the testing phase and does not require retraining of the PLDA or SVM backends. The idea is to deemphasize or remove the influence of enrollment sessions that are problematic for one reason or another. In the RATS speaker recognition evaluation, each speaker model has 6 enrollment segments. McLaren[1] first proposed a trial-based enrollment pruning method. For each test trial, the algorithm picks the maximum score from all possible combinations by filtering 0, 1 or 2 segments from the candidate enrollment data. Building from this work, our method is to have an equal weight score combination of a score using **all** 6 enrollment segments and a score based on picking the maximum score from all possible combinations of 1, 2 or 3 segments being pruned from the candidate enrollment data.

[1]We thank Mitch McLaren from SRI International for permitting us to share his approach in this paper.

Table 1: *Table showing the performance comparison of PLDA and SVM models. In particular, we compare three training and test methodologies: (a) PLDA joint multi-session training and then average scoring, (b) PLDA joint multi-session training and testing, (c) SVM pairwise training and average scoring, and (d) SVM joint multi-session training and testing. (Miss (%) at 2.5% False Accept Rate)*

System Task	120s	30s	10s	3s
a) PLDA (joint, avg)	1.86	5.99	18.17	52.11
b) PLDA (joint, joint)	1.34	**4.38**	15.32	48.02
c) SVM (pairwise, avg)	1.95	6.07	16.63	45.70
d) SVM (joint, joint)	**1.29**	4.46	**14.09**	**43.79**
e) Combination (b, d)	1.11	3.72	13.24	43.99

The results for this method are presented in Table 2. It is interesting to note that, in comparing Tables 1 and 2, PLDA enrollment pruning improves the PLDA performance but not the SVM related result. Enrollment pruning for PLDA enables the PLDA system to beat the joint SVM result for the two longer duration tasks. It is speculated that since the SVM was optimized explicitly for six sessions (including problematic enrollment sessions) segment pruning may not benefit. Although we have not experimentally validated this, it is hypothesized that pruning for multi-session SVMs should be considered as part of the SVM training process and not at the testing stage only.

Table 2: *Table comparing the performance of the PLDA and SVM models with enrollment (p)runing. (Miss (%) at 2.5% False Accept Rate)*

System Task	120s	30s	10s	3s
a) PLDA$_p$ (joint, joint)	**1.21**	**4.09**	14.83	47.12
b) SVM$_p$ (joint, joint)	1.25	4.42	**14.12**	**44.09**
c) Combination (a, b)	1.04	3.55	12.91	43.65

4. Conclusions

This work demonstrated how SVM kernels can be reformulated to model speaker recognition trials with more than one enrollment session. The two SVM approaches evaluated here performed particularly well for the shorter duration tasks compared to the PLDA approaches. The proposed joint SVM training method (jointly trained using 6 enrollment session trials) performed consistently better than the SVM trained with the single enrollment setup. Additionally, it performed comparably to PLDA for the long durations and significantly better for the shorter durations. The combination of PLDA and the new SVM approach provided additional gains. The experiment also demonstrates that aligning the cost function more closely to the actual classification task being performed is beneficial. For enrollment segment pruning, we found that PLDA based models benefitted from this while the SVM did not. This may be because the SVM is focussed on the correct classification of the trials and less influenced by problematic enrollment sessions.

5. References

[1] 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.

[2] S. Prince and J. Elder, "Probabilistic linear discriminant analysis for inferences about identity," *ICCV*, 2007.

[3] S. Cumani, N. Brümmer, L. Burget, P. Laface, O. Plchot, and V. Vasilakakis, "Pairwise discriminative speaker verification in the i-vector space," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, no. 6, pp. 1217–1227, 2013.

[4] L. Burget, O. Plchot, S. Cumani, O. Glembek, P. Matejka, and N. Brummer, "Discriminatively trained probabilistic linear discriminant analysis for speaker verification," *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4832–4835, 2011.

[5] S. Yaman, J. Pelecanos, and W. Zhu, "Unifying PLDA and polynomial kernel SVMs," *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 7698–7701, 2013.

[6] S. Yaman and J. Pelecanos, "Using polynomial kernel support vector machines for speaker verification," *IEEE Signal Processing Letters*, vol. 20, no. 9, pp. 901–904, 2013.

[7] S. Yaman, J. Pelecanos, and M. Omar, "On the use of nonlinear polynomial kernel SVMs in language recognition," *Interspeech*, 2012.

[8] W. Campbell, D. Sturim, D. Reynolds, and A. Solomonoff, "SVM based speaker verification using a GMM supervector kernel and NAP variability compensation," *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2006.

[9] B. Schölkopf and A. Smola, *Learning with kernels: Support vector machines, regularization, optimization, and beyond*. MIT press, 2002.

[10] V. Vapnik, *The nature of statistical learning theory*. Springer, 2000.

[11] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.

[12] S. Shalev-Shwartz, Y. Singer, N. Srebro, and A.Cotter, "Pegasos: Primal estimated sub-gradient solver for SVM," *Mathematical programming*, vol. 127, no. 1, pp. 3–30, 2011.

[13] K. Walker and S. Strassel, "The RATS radio traffic collection system," in *Odyssey*, 2012.

[14] W. Zhu, S. Yaman, and J. Pelecanos, "The IBM RATS Phase II speaker recognition system: Overview and analysis," *Interspeech*, 2013.

[15] 3GPP2, "Enhanced variable rate codec, speech service option 3 for wideband spread spectrum digital systems," *http://www.3gpp2.org/public_html/specs/C.S0014-A_v1.0_040426.pdf*, 2004.

[16] Unknown, "Enhanced variable rate codec," *http://en.wikipedia.org/wiki/Enhanced_Variable_Rate_Codec*, 2014.

[17] D. Reynolds, "Speaker identification and verification using Gaussian mixture speaker models," *Speech Communication*, vol. 17, no. 1-2, pp. 91–108, 1995.

[18] D. Reynolds, T. Quatieri, and R. Dunn, "Speaker verification using adapted Gaussian mixture models," *Digital Signal Processing*, vol. 10, no. 1/2/3, pp. 19–41, 2000.

[19] P. Kenny, "Joint factor analysis of speaker and session variability: Theory and algorithms," *Technical Report CRIM-06/08-13*, 2005.

[20] A. Hatch, S. Kajarekar, and A. Stolcke, "Within-class covariance normalization for SVM-based speaker recognition," *International Conference on Spoken Language Processing*, 2006.

[21] D. Garcia-Romero and C. Espy-Wilson, "Analysis of i-vector length normalization in speaker recognition systems," *Interspeech*, pp. 249–252, 2011.