DISCRETE EXPECTED LIKELIHOOD KERNEL FOR SVM-BASED SPEAKER VERIFICATION

Kong Aik Lee[1], *Haizhou Li*[1], *Chang Huai You*[1], *Tomi Kinnunen*[2], and *Khe Chai Sim*[1]

[1]Human Language Technology Department, Institute for Infocomm Research,
Agency for Science, Technology and Research (A*STAR), Singapore
[2]Department of Computer Science and Statistics, University of Eastern Finland, Finland
kalee@i2r.a-star.edu.sg

ABSTRACT

The construction of kernel functions to handle sequences of speech feature vectors is crucial in using support vector machine (SVM) for speaker verification. Previous studies have reported the idea of representing speech signals as sequences of discrete acoustic or phonotactic events. This paper introduces a class of SVM kernels derived based on the expected likelihood measure between the probability distributions of discrete event sequences. We investigate and compare the effectiveness of three expected likelihood kernels using the universal background model (UBM) as the discrete event detector. Experiments conducted on the NIST 2006 speaker verification task indicate that the proposed kernel outperforms the popular rank-normalized kernel.

1. INTRODUCTION

Speaker verification [1] is the task of verifying the identity of persons using their voices. Recent advances reported in [2, 3, 4, 5] have shown successful application of support vector machine (SVM) [6] for speaker modeling.

The key issue in using SVM for classifying speech signals, which have a varying number of spectral vectors, is how to represent them in a suitable form as SVM can only use input of a fixed dimensionality. A common approach is to map the sequences explicitly into fixed-dimensional vectors known as *supervectors*. Classifying variable-length sequences is thereby translated into a simpler task of classifying the supervectors. For instance, in [3] speech vectors are mapped to a high-dimensional space via time-averaged polynomial expansion. In [4], speech vectors are used to train a Gaussian mixture model (GMM) via the adaptation of a so-called *universal background model* (UBM). The supervector is then formed by concatenating the mean vectors of the adapted GMM. In [7], the *maximum likelihood linear regression* (MLLR) transform is used to form the supervectors comprising of the transform coefficients. It should be mentioned that the term *supervector* was originally used in [4, 8] to refer to the GMM supervector. Here, we use similar term in a broader sense referring to any fixed-dimensional vector that represents a whole speech sequence as a single point in the vector space, having a much higher dimensionality than the original input space.

This paper advocates the use of discrete events and their probabilities to construct supervectors. Discrete events arise naturally in modeling many types of data, for example, letters, words, and DNA sequences. Speech signals can also be represented as sequences of discrete symbols. For instance, high-level features extraction (e.g., idiolect, phonotactic, prosody) usually produces discrete symbols. The idea of using such discrete representation as features for SVM has been examined previously in [5, 9, 10, 11]. Their results showed that the greatest difficulty lies at the construction of kernel function (i.e., inner product function) which should give a proper similarity measure between two event sequences. In this paper, we solve this problem by first model the distribution of the discrete events as probability mass function (PMF). SVM kernel is then derived based on the expected likelihood measure between PMFs. We report three discrete expected likelihood kernels, analyze and compare their efficacy on the NIST 2006 speaker recognition tasks. To the best of our knowledge, the idea of expected likelihood kernel was first reported in [12] for continuous distribution. Here, we extend the idea for discrete distribution and investigate its relevance to speaker recognition.

2. SUPERVECTOR OF DISCRETE PROBABILITIES

We first review the general concept of discrete events and the estimation of discrete probabilities. We then apply this framework for constructing supervector from the discrete probabilities of some acoustic events.

2.1. Discrete events and the estimation of discrete probabilities

Let some discrete events $S = \{e_i, i = 1, 2, \ldots, M\}$ have M possible outcomes, and let ω_i be the probability of observing the ith event e_i. Given a sequence of speech feature vectors, $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T\}$, our goal is to estimate the probabilities $\Omega = \{\omega_1, \omega_2, \ldots, \omega_M\}$ of the events observed in the speech signal. These discrete events could correspond to abstract linguistic units such as

 ISSN 2076-1465

phonemes, syllables, words, or subsequences of *n* symbols (i.e., *n*-grams). In its simplest form, a discrete event could also correspond to a Gaussian density in the acoustic space as we shall see in Section 2.2.

Using the maximum *a posteriori* (MAP) criterion [13], the discrete probabilities can be estimated as

$$\begin{aligned}\tilde{\Omega} &= \arg\max_{\Omega}\left\{\log\prod_{i=1}^{M}\omega_i^{n_i(\mathcal{X})}+\log g(\omega_1,\ldots,\omega_M)\right\}\\ &= \arg\max_{\Omega}\left\{\sum_{i=1}^{M}n_i(\mathcal{X})\log\omega_i+\log g(\omega_1,\ldots,\omega_M)\right\}\end{aligned} \quad (1)$$

subject to the constraints $\sum_{i=1}^{M}\omega_i=1$ and $\omega_i\geq 0$. In the above equation $n_i(\mathcal{X})$ denotes the number of occurrences of the events e_i in $\mathcal{X}$, and $g(\Omega)$ is the prior density for the parameters Ω. Taking the prior as a Dirichlet density [13], the MAP estimate $\tilde{\Omega}=\{\tilde{\omega}_1,\tilde{\omega}_2,\ldots,\tilde{\omega}_M\}$ can be easily solved via the method of Lagrange multiplier, as follows

$$\tilde{\omega}_i=\frac{n_i(\mathcal{X})+\nu_i}{\sum_{i=1}^{M}[n_i(\mathcal{X})+\nu_i]},\ i=1,2,\ldots,M\ . \quad (2)$$

The MAP estimate is simply the sum of the observed statistics n_i and the so-called hyperparameters ν_i. The denominator ensures that the probabilities $\tilde{\omega}_i$ always sum to one.

2.2. UBM as soft quantizer

The universal background model, or UBM, is a Gaussian mixture model (GMM) trained by pooling together the speech feature vectors from a number of different speakers. The UBM, denoted by Θ, is characterized by the density function,

$$p(\mathbf{x}|\Theta)=\sum_{i=1}^{M}\lambda_i\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_i,\boldsymbol{\Sigma}_i), \quad (3)$$

where $\Theta=\{\lambda_i,\boldsymbol{\mu}_i,\boldsymbol{\Sigma}_i,i=1,2,\ldots,M\}$ denotes the parameters of the *M* Gaussians: the mixture weights (λ_i), the mean vectors ($\boldsymbol{\mu}_i$) and the covariance matrices ($\boldsymbol{\Sigma}_i$).

The UBM represents a speaker-independent distribution in the acoustic space, where similar acoustic features are grouped together and represented with Gaussian densities. Let each of the Gaussian densities represent a discrete event e_i. Given a speech segment $\mathcal{X}$, the number of occurrences of event e_i is computed by accumulating the posterior probabilities

$$P(i|\mathbf{x}_t,\Theta)=\frac{\lambda_i\mathcal{N}(\mathbf{x}_t|\boldsymbol{\mu}_i,\boldsymbol{\Sigma}_i)}{\sum_{j=1}^{M}\lambda_j\mathcal{N}(\mathbf{x}_t|\boldsymbol{\mu}_j,\boldsymbol{\Sigma}_j)} \quad (4)$$

for the whole utterance, as follows

$$n_i(\mathcal{X})=\sum_{t=1}^{T}P(i|\mathbf{x}_t,\Theta), \quad (5)$$

where *T* is the number of speech frames. The UBM quantizes the input vectors into discrete symbols; each corresponds to a Gaussian density. Since the Gaussian densities can be overlapped, rather than partitioned, soft membership can be computed based on the Bayes rule as given in (4).

Finally, using the occupancy count in (5), the MAP estimate $\tilde{\Omega}$ can be obtained by substituting the results into (2) with the parameters ν_i set to

$$\nu_i-1=\tau\cdot M\cdot\lambda_i, \quad (6)$$

where λ_i are the weights of the UBM and the controlled parameter τ has to be greater or equal to 0. This is known as the τ-initialization method in [13]. Feasible values for τ range from 0 to 1, which we have found effective for this application. For $\tau=0$, the MAP estimate reduces to the maximum likelihood (ML) estimate.

2.3. Constructing supervector

The set of discrete events $S=\{e_1,e_2,\ldots,e_M\}$ and their estimated probabilities $\tilde{\Omega}=\{\tilde{\omega}_1,\tilde{\omega}_2,\ldots,\tilde{\omega}_M\}$ can be conveniently represented in functional form as $P(h|\tilde{\Omega})$, where $P(h=e_i|\tilde{\Omega})=\tilde{\omega}_i$ and the variable $h\in S$ represents any possible event in the set. The function $P(h|\tilde{\Omega})$ is known as the probability mass function (PMF). We can further express the PMF in vector form as

$$\mathbf{p}=[P(e_1),P(e_2),\ldots,P(e_M)]^{\mathrm{T}}=[\tilde{\omega}_1,\tilde{\omega}_2,\ldots,\tilde{\omega}_M]^{\mathrm{T}}, \quad (7)$$

where the superscript $^{\mathrm{T}}$ denotes transposition. The vector **p** has a fixed dimensionality, *M*, equivalent to the cardinality of the event set *S*. It represents the speech segment $\mathcal{X}$ in terms of the distribution of discrete events observed in $\mathcal{X}$. These attributes fulfill our requirement of supervector representation. In this paper, $\tilde{\Omega}$ is obtained from $\mathcal{X}$ using (2), (4), (5), and (6). The dimensionality of the resulting supervector **p** is determined by the number of Gaussian densities in the UBM.

3. DISCRETE EXPECTED LIKELIHOOD KERNEL

For a kernel to be admissible for SVM, it has to be symmetric and represent an inner product in the feature space [6]. In this section, we introduce and compare three different expected likelihood measures and show how to use them for constructing SVM kernel.

3.1. Expected likelihood

Consider two PMFs parameterized by $\tilde{\Omega}_a$ and $\tilde{\Omega}_b$ corresponding to speech signals $\mathcal{X}_a$ and $\mathcal{X}_b$, respectively. We are interested in finding a symmetric measure that defines the similarity between the PMFs. A natural measure that satisfies the symmetric property is the expected value of the first PMF with respect to the second one:

$$E_b\{P(h|\tilde{\Omega}_a)\}=\sum_{h\in S}P(h|\tilde{\Omega}_a)P(h|\tilde{\Omega}_b)=E_a\{P(h|\tilde{\Omega}_b)\}. \quad (8)$$

Note that the expectation operation over a discrete distribution is the summation of $P(h|\tilde{\Omega}_a)$ by using

$P(h|\tilde{\Omega}_b)$ as a weighting function. Since the probability $P(h=e_i|\tilde{\Omega}_a)=\mathcal{L}(\tilde{\Omega}_a|h)$ is also known as the likelihood of the model $\tilde{\Omega}_a$, (8) is referred to as the *expected likelihood* (EL). By using the notations $P(h=e_i|\tilde{\Omega}_a)=\tilde{\omega}_{i,a}$ and $P(h=e_i|\tilde{\Omega}_b)=\tilde{\omega}_{i,b}$ in (8), we arrive at the following EL kernel

$$\kappa_{\mathrm{EL}}(\mathcal{X}_a,\mathcal{X}_b)=\sum_{i=1}^{M}\tilde{\omega}_{i,a}\tilde{\omega}_{i,b}=\mathbf{p}_a^{\mathrm{T}}\mathbf{p}_b\,, \tag{9}$$

which is simply the inner product between two supervectors of discrete probabilities.

3.2. Expected likelihood ratio

The EL kernel in (9) may be suboptimum as rare acoustic events might be outweighed by those with higher probabilities (which tend to dominate the inner product). A normalization term based on the prior weights of the UBM $\Lambda=\{\lambda_1,\lambda_2,\ldots,\lambda_M\}$ can be included, in which case we arrive at the *expected likelihood ratio* (ELR):

$$E_b\left\{\frac{P(h|\tilde{\Omega}_a)}{P(h|\Lambda)}\right\}=\sum_{h\in S}\frac{P(h|\tilde{\Omega}_a)P(h|\tilde{\Omega}_b)}{P(h|\Lambda)}=E_a\left\{\frac{P(h|\tilde{\Omega}_b)}{P(h|\Lambda)}\right\}. \tag{10}$$

Substituting $P(h=e_i|\tilde{\Omega}_a)=\tilde{\omega}_{i,a}$, $P(h=e_i|\tilde{\Omega}_b)=\tilde{\omega}_{i,b}$ and $P(h=e_i|\Lambda)=\lambda_i$ into (10), we arrive at the following ELR kernel

$$\kappa_{\mathrm{ELR}}(\mathcal{X}_a,\mathcal{X}_b)=\sum_{i=1}^{M}\left\{\frac{\tilde{\omega}_{i,a}\tilde{\omega}_{i,b}}{\lambda_i}\right\}. \tag{11}$$

The scaling factor $1/\lambda_i$ allows rare acoustic events to be emphasized while suppressing those with high probabilities. The kernel function can also be written in vector notation as follows

$$\kappa_{\mathrm{ELR}}(\mathcal{X}_a,\mathcal{X}_b)=(\mathbf{p}_{\mathrm{ELR}})^{\mathrm{T}}\,\mathbf{p}_{\mathrm{ELR}}\,, \tag{12}$$

where

$$\mathbf{p}_{\mathrm{ELR}}=\left[\frac{\tilde{\omega}_1}{\sqrt{\lambda_1}},\frac{\tilde{\omega}_2}{\sqrt{\lambda_2}},\ldots,\frac{\tilde{\omega}_M}{\sqrt{\lambda_M}}\right]^{\mathrm{T}} \tag{13}$$

could be seen as the normalized version of the supervector $\mathbf{p}$. Similar normalization method was introduced in [14] for the use of *n*-gram probabilities in spoken language recognition application and was referred to as the term-frequency log-likelihood ratio (TFLLR) scaling. Here, we derive and interpret the same normalization method from the expected likelihood perspective.

3.3. Expected square-root likelihood ratio

In the ELR kernel, the likelihood ratio is taken with respect to the UBM. Considering that the likelihood ratio is now computed between the two PMFs to be compared, we arrive at

$$\begin{aligned}E_b\left\{\sqrt{\frac{P(h|\tilde{\Omega}_a)}{P(h|\tilde{\Omega}_b)}}\right\}&=\sum_{h\in S}\sqrt{P(h|\tilde{\Omega}_a)P(h|\tilde{\Omega}_b)}\\&=E_a\left\{\sqrt{\frac{P(h|\tilde{\Omega}_b)}{P(h|\tilde{\Omega}_a)}}\right\}.\end{aligned} \tag{14}$$

Notice that a square-root operator is used so that the resulting measure will be symmetric. Substituting $P(h=e_i|\tilde{\Omega}_a)=\tilde{\omega}_{i,a}$ and $P(h=e_i|\tilde{\Omega}_b)=\tilde{\omega}_{i,b}$ in (14), we obtain the following *expected square-root likelihood ratio* (ESLR) kernel:

$$\begin{aligned}\kappa_{\mathrm{ESLR}}(\mathcal{X}_a,\mathcal{X}_b)&=\sum_{i=1}^{M}\sqrt{\tilde{\omega}_{i,a}\tilde{\omega}_{i,b}}\\&=(\mathbf{p}_{\mathrm{ESLR}})^{\mathrm{T}}\,\mathbf{p}_{\mathrm{ESLR}},\end{aligned} \tag{15}$$

where

$$\mathbf{p}_{\mathrm{ESLR}}(\mathcal{X})=\left[\sqrt{\tilde{\omega}_1},\sqrt{\tilde{\omega}_2},\ldots,\sqrt{\tilde{\omega}_M}\right]^{\mathrm{T}}. \tag{16}$$

Comparing (15) to (9), the only difference is the square-root operator. Numerically, the square-root operator has an effect in applying higher gain to rare events; the gain reduces gradually for higher probabilities. The ESLR kernel is also simpler compared to the ELR kernel, which normalizes individual dimension with a constant scaling factor instead of warping. It is worth mentioning that the statistical measure in (14) is commonly referred to as the Bhattacharyya coefficient in the literature [15]. Here, we interpret the same statistical measure from expected likelihood perspective.

4. USING THE EXPECTED LIKELIHOOD KERNELS WITH SVM

We deliberately write the kernel functions in (9), (12) and (15) in terms of $\mathcal{X}_a$ and $\mathcal{X}_b$ so that the kernel function represents (i) a mapping from $\mathcal{X}$ to $\tilde{\Omega}$ and (ii) a similarity measure between two sequences. Using this notation, the discriminant function of the SVM [6] can be expressed as,

$$f(\mathcal{X})=\sum_{l=1}^{L}\alpha_l y_l\kappa(\mathcal{X}_l,\mathcal{X})+b\,, \tag{17}$$

where $\kappa(\mathcal{X}_l,\mathcal{X})$ is any of the three kernels above, b is the bias parameter, L is the number of support vectors, and α_l are the weights assigned to the *l*th support vector with its label given by $y_l\in\{-1,+1\}$.

Since a supervector represents a speech utterance as a single point in the vector space, it becomes possible to remove the unwanted variability, due to different handsets, channels and phonetic content, from the supervector by linear projection. Let **E** be an *M*-by-*N* matrix representing the unwanted subspace that causes the variability. Nuisance attribute projection (NAP) [16] removes the unwanted variability from a supervector via a projection to the subspace complementary to **E**, as follows

Table 1: EER and MinDCF on the core test of NIST SRE 2006 for the EL [eq. (9)], ELR [eq. (12)], ESLR [eq. (15)], rank normalization [10], and GMM supervector (GSV) [4].

System	EER (%)	MinDCF (×100)
EL	6.81	3.31
ELR	6.59	3.30
ESLR	5.11	2.52
Rank Normalization	5.51	2.61
GSV	4.85	2.33
ESLR + GSV	4.50	2.22

$$\mathbf{p}' = \left(I - \mathbf{E}\mathbf{E}^{\mathrm{T}}\right)\mathbf{p}\,. \tag{18}$$

NAP assumes that the variability is confined in a relatively low dimensional subspace such that $N \ll M$. The columns of $\mathbf{E}$ are the eigenvectors of the within-speaker covariance matrix estimated from a development dataset with large number of speakers, each having several training sessions.

In (18), the supervector $\mathbf{p}$ depends on the kernel function (EL, ELR, or ESLR) used. SVM modeling is then performed using the supervectors $\mathbf{p}'$ that have been compensated for session variability. The SVM discriminant function can be expressed in terms of the compensated supervector as follows

$$f(\mathbf{p}') = \sum_{l=1}^{L} \alpha_l y_l \left(\mathbf{p}_l'\right)^{\mathrm{T}} \mathbf{p}' + b\,. \tag{19}$$

5. EXPERIMENTAL RESULTS

The experiments were carried out on the NIST speaker recognition evaluation (SRE) 2006 corpus [17]. The core test consists of 3612 genuine and 47836 imposter trials. There are 810 target speakers each enrolled with one side of a 5 minutes conversation. Our development data were drawn from the NIST SRE 2004 and SRE 2005. All speech samples are first pre-processed to remove silence and converted into sequences of 36-dimensional MFCCs with deltas and double deltas. RASTA and utterance-level mean and variance normalization were performed. NAP and test normalization (T-norm) were applied to compensate for session variability at the model and score levels, respectively. The NAP projection matrix has a rank of 40 and was derived from NIST SRE 2004 data, while the T-norm cohorts were selected from NIST SRE 2005 data.

The UBM used in the experiments contains M = 16384 mixtures. The parameter τ in the discrete probabilities estimation is set to 0.01. Due to a relatively large number of mixtures, *Gaussian selection* technique [18] is used for speeding up likelihood evaluation. Here, the hash model size is set to 512 while the length of the shortlists is 2048. This leads to approximately six times faster computation. We also tie the Gaussian components such that they share the same global covariance matrix to further simplify the computation of the posterior probabilities.

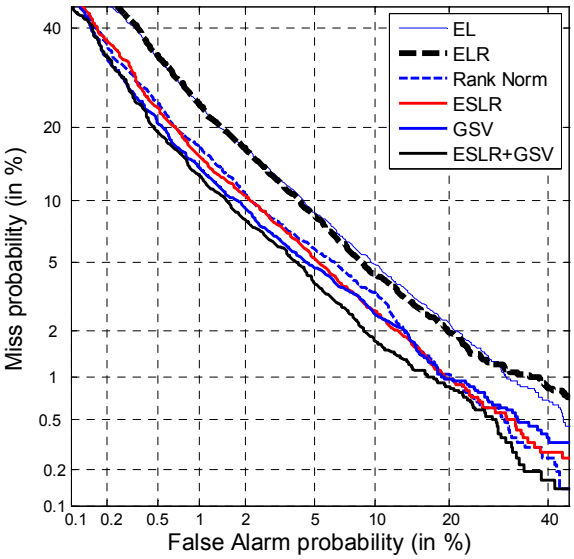

Figure 1: Detection error tradeoff (DET) curves of individual systems and fusion evaluated on the NIST SRE 2006 core task. The order of the system in the legend indicates the performance with the bottom (i.e., ESLR + GSV) being the best, which corresponds to the curve closest to the origin.

We compare the performance of the proposed kernels (EL, ELR, and ESLR) and include the rank normalization [10] in comparison as well. The scaling factor $1/\sqrt{\lambda_i}$ in (13) and the square-root operator in (16) impose a feature normalization step on the supervectors. Proper feature normalization is desirable as SVMs are not invariant to scaling in the feature space. To this end, r*ank normalization* was proposed in [10] for high-level feature having similar count-based characteristic. Rank normalization replaces each dimension of the supervector by its rank in a background data, which is more complicated than any of the EL kernels.

Table 1 shows the performance in terms of equal error rates (EERs) and minimum detection cost function (MinDCF), while Fig. 1 shows the detection error trade-off (DET) curve. It is evident that the ESLR and ELR kernels perform better than the EL kernel, which indicates that kernel normalization is important. Comparing the ESLR and ELR kernels, on the other hand, shows that the square-root operator is more robust than the inverse probability weighting in the ERL kernel, where nuisance features may be unintentionally amplified. The ESLR kernel performs consistently better than rank normalization in terms of EER and MinDCF showing the efficiency of our approach.

Table 1 also shows the performance of the GMM mean supervector (GSV) [4]. For the GSV system, the UBM consists of 512 mixtures leading to supervectors of dimensionality 18432. Recall that the supervector of discrete probabilities has a comparable dimensionality of $M = 16384$. The datasets used for UBM training, SVM

background data, NAP and t-norm are the same for all systems. The proposed ESLR kernel exhibits competitive performance compared to the GSV system, both having approximately the same dimensionality in the kernel space. We fused the two systems using equal weights. The fusion gives relative EER improvement of 7.2% over the best single system. Since the same datasets were used for system development, the fusion result shows the diversity in speaker modeling, suggesting that the two approaches capture complementary aspects (continuous vs. discrete) of speaker characteristics.

6. CONCLUSION

We have introduced a class of SVM kernels derived based on the expected likelihood measure between the distributions of discrete speech events. We demonstrate the usefulness of the expected likelihood kernels on the discrete speech events representation derived from the frame posterior probabilities of UBM. Experimental results show that the expected square-root likelihood ratio (ESLR) kernel performs better than the rank-normalized kernel using the same feature on the 2006 NIST speaker verification task. The expected likelihood kernel gives equally good accuracy compared to, and fuses with, the state-of-the-art GMM mean supervector approach.

It is worth emphasizing that, even though the current work uses a UBM quantizer to construct supervectors, this is not necessarily the case; the proposed method can be used with other types of front-end quantizer. In particular, we expect the method to be readily applicable for spoken language recognition, and applications beyond speech technology that operate on discrete symbols, such as natural language processing (NLP) and bioinformatics.

ACKNOWLEDGEMENT

The work of T. Kinnunen was supported by the Academy of Finland (project no 132129).

REFERENCES

[1] T. Kinnunen and H. Li, "An overview of text-independent speaker recognition: from features to supervectors," *Speech Communication*, vol. 52, no. 1, pp. 12-40, Jan. 2010.

[2] V. Wan and S. Renals, "Speaker verification using sequence discriminant support vector machines," *IEEE Trans. Speech Audio Process.*, vol. 13, no. 2, pp. 203-210, Mar. 2005.

[3] W. M. Campbell, J. P. Campbell, D. A. Reynolds, E. Singer, and P. A. Torres-Carrasquillo, "Support vector machines for speaker and language recognition," *Computer Speech and Language*, vol. 20, no. 2-3, pp. 210-229, 2006.

[4] W. M. Campbell, D. E. Sturim, D. A. Reynolds, and A. Solomonoff, "SVM based speaker verification using a GMM supervector kernel and NAP variability compensation," in *Proc ICASSP*, pp. I-97- I-100, 2006..

[5] K. A. Lee, C. You, H. Li, and T. Kinnunen, "A GMM-based probabilistic sequence kernel for speaker recognition," in *Proc. Interspeech*, pp. 294-297, 2007.

[6] V. Kecman, Learning and Soft Computing: Support Vector Machines, Neural Networks, and Fuzzy Logic Models. MA: MIT Press, 2001.

[7] A. Stolcke, S. S. Kajarekar, L. Ferrer, and E. Shriberg, "Speaker recognition with session variability normalization based on MLLR adaptation transforms," *IEEE Trans. Audio, Speech, and Lang. Process.*, vol. 15, pp. 1987-1998, Sep. 2007.

[8] P. Kenny, M. Mihoubi, and P. Dumouchel, "New MAP estimates for speaker recognition," in *Proc. EUROSPEECH*, pp. 2964-2967, 2003.

[9] E. Shriberg, L. Ferrer, S. Kajarekar, A. Venkataraman and A. Stolcke, "Modeling prosodic feature sequences for speaker recognition," *Speech Communication,* 46(3-4), pp. 455-472, Jul. 2005.

[10] A. Stolcke, S. Kajarekar, and L. Ferrer, "Nonparametric feature normalization for SVM-based speaker verification," in *Proc.ICASSP,* pp. 1577-1580, 2008.

[11] N. Scheffer and J. -F. Bonastre, "UBM-GMM driven discriminative approach for speaker verification," in *Proc. Odyssey*, 2006.

[12] T. Jebara and R. Kondor, "Bhattacharyya amd expected likelihood kernels," in *Proc*. 16th Annual Conference on Computational Learning Theory, 2003.

[13] C. H. Lee and Q. Huo, "On adaptive decision rules and decision parameter adaptation for automatic speech recognition," *Proceedings of the IEEE*, vol. 88, no. 8, pp. 1241-1269, Aug. 2000.

[14] W. M. Campbell, J. P. Campbell, Terry P. Gleason, D. A. Reynolds, and Wade Shen, "Speaker verification using support vector machines and high-level features," *IEEE Trans. Audio, Speech, and Language Process.*, vol. 15, no. 7, pp. 2085-2094, Sep. 2007.

[15] T. Kailath, "The divergence and Bhattacharyya distance measures in signal selection," *IEEE Trans. Commun. Technol.*, vol. COM-15, no. 1, pp. 52-60, Feb. 1967.

[16] A. Solomonoff, W. M. Campbell, and C. Quillen, "Channel compensation for SVM speaker recognition," in *Proc. Odyssey*, pp. 57-62, 2004.

[17] *The NIST Year 2006 Speaker Recognition Evaluation Plan*, National Institute of Standards and Technology, Mar. 2006.

[18] R. Auckenthaler and J. S. Mason, "Gaussian selection applied to text-independent speaker verification," in *Proc. Odyssey*, 2001.