SPEAKER VERIFICATION USING KERNEL-BASED BINARY CLASSIFIERS WITH BINARY OPERATION DERIVED FEATURES

Hung-Shin Lee[1, 2], Yu Tso[3], Yun-Fan Chang[3], Hsin-Min Wang[2], Shyh-Kang Jeng[1]

[1]Department of Electrical Engineering, National Taiwan University, Taiwan
[2]Institute of Information Science, Academia Sinica, Taiwan
[3]Research Center for Information Technology Innovation, Academia Sinica, Taiwan

ABSTRACT

In this paper, we study the use of two kinds of kernel-based discriminative models, namely support vector machine (SVM) and deep neural network (DNN), for speaker verification. We treat the verification task as a binary classification problem, in which a pair of two utterances, each represented by an i-vector, is assumed to belong to either the "within-speaker" group or the "between-speaker" group. To solve the problem, we employ various binary operations to retain the basic relationship between any pair of i-vectors to form a single vector for training the discriminative models. This study also investigates the correlation of achievable performances with the number of training pairs and the various combinations of basic binary operations, using the SVM and DNN binary classifiers. The experiments are conducted on the male portion of the core task in the NIST 2005 Speaker Recognition Evaluation (SRE), and the results are competitive or even better, in terms of normalized decision cost function (minDCF) and equal error rate (EER), while compared to other non-probabilistic based models, such as the conventional speaker SVMs and the LDA-based cosine distance scoring.

Index Terms— speaker verification, SVM, DNN, i-vector

1. INTRODUCTION

As a key component in speaker recognition tasks, speaker verification, which verifies whether the speech utterances pronounced by an unknown speaker correspond to the claimed identity or not, has become more and more indispensable in many security-related applications, such as telephone banking [1] and forensic analysis [2]. In general, the design of a conventional speaker verification system needs to take two distinct phases into consideration while processing the extracted speech features [3, 4]. In the enrollment phase, each target speaker to be verified by the system has to provide some speech samples to train the model for that particular speaker either by means of the generative approach that captures the empirical probability density function corresponding to the acoustic feature vectors or through the discriminative way, seeking to minimize the error on a set of true and impostor training samples. In the test phase, an individual makes a claim about his/her identity, then the system proceeds to authenticate whether that claim is true or false through the clamed speaker model.

Going into detail while facing practical issues in the real world, in order to make the speaker-specific models more reliable and immune to the problem caused by an inadequate amount of training data, a universal background model (UBM), such as a Gaussian mixture model (GMM) [5, 6] and a hidden Markov model (HMM) [7, 8], is often trained in advance for efficiently learning the target speaker models by adaptation. In recent years, as the number of target speakers is increasing and some discriminative models, such as the support vector machine (SVM) [9] and artificial neural network (ANN) [10], become more and more attractive and effective, the aforementioned generative models no longer serve as the only choice for verification. On the contrary, they are often used as tokenizers to represent speech utterances of varying durations by fixed-length supervectors, concatenations of multiple vectors [11], or the well-known i-vectors [12], where the speaker-dependent information of the whole utterance and session/channel compensation are considered simultaneously. These representations, making the way to compare one utterance with another more straightforward, are effective for most of backend discriminative classifiers. For example, many researchers exploited using an SVM to model the boundary between the target speaker and the imposter speakers. Some of them constructed kernel functions of supervectors based on the Kullback-Leibler (KL) divergence [13] or the Bhattacharyya -based distance [14] between two GMMs, and some simply harnessed the dot-product kernel along with the within class covariance normalization (WCCN) [15] or linear discriminant analysis (LDA) [16] based projection matrices on i-vectors [12, 17].

Therefore, the classic structure for speaker verification conveys a simple three-point message. First, each target speaker needs to be independently and individually modeled, resulting that the whole system comprises different models, so that score normalization techniques are often necessary before an acceptance or a rejection decision threshold is made [18]. Second, in order to train speaker models, a set of utterances from cohort speakers needs to be involved to form a set of negative examples [13], even if we cannot completely assure whether the examples are all *true* negative or not. Finally, without any utterance-partitioning techniques [17], the number of enrollment utterances is usually much less than the size of the cohort set, because in most of real applications, verification systems cannot expect that the data obtained from the target speaker are always sufficient for model training.

These characteristics might also cause some problems in practice. For instance, how to build up a robust discriminative classifier in a small-sample-size situation? How to make scores generated by different models comparable? And how to effectively maintain the consistency of a specific speaker model when its representative voice has to be replaced? In this paper, we attempt to address these issues by translating the verification problem into a binary (or two-class) classification problem: given a pair of utterances, in which one is from the claimed speaker and the other from an unknown speaker, decision should be made regarding which group it falls into: the "same speaker" group or the "different speakers" group. The holist concept results in only one classifier in the verification

system. To our knowledge, the interpretation and the corresponding treatment can be dated back to the work proposed by Moghaddam *et al.* [19] and the afterwards extended version based on factor analysis [20, 21] on face recognition (or might be much earlier in other fields like information retrieval). In [19], the authors considered *pixel-wise* difference between probe and gallery images and modeled distributions of "within-individual" and "between-individual" differences in the Bayesian sense. Therefore, for two new images, they used the posterior probability that the difference belonged to each distribution. Along a similar vein in speaker verification, Cumani *et al*. used a suitable dot-product based kernel derived from the two-covariance generative model to train a single *linear* SVM in the primal form, which classified a pair of utterances into either "same speaker" or "different speakers" [22, 23]. In light of the concept that an i-vector can be decomposed into a speaker factor and a Gaussian distributed channel component, they extracted the expanded vectorial form from the formulation of the speaker detection log-likelihood, shown in [31], to express the feature-wise interaction between the i-vector pair.

Although modeling the relationship between the "within-speaker" and "between-speaker" groups is conceptually similar to the above work, what we propose in the paper still differs in some aspects, which can be illustrated with two questions. First, how to make a pair of utterances joined into a single input that conforms to the backend classifier? We use commutable binary operations in a more general way for the purpose to capture the pure relationship between two utterances, without regard to any similarity measures like the feature difference derived by the division operation in [19] and the speaker likelihood considered in [22]. Second, how to model the "within-speaker" and "between-speaker" groups? In recent research in speaker verification, deep neural network (DNN) [27] and its fundamental building block, restricted Boltzmann machine (RBM), have proven effective for feature representation [32]. In this paper, apart from the probabilistic schemes used in [20, 21] and the linear SVM, we use the *nonlinear* SVM and DNN as binary classifiers for structurally modeling the two groups.

The remainder of this paper is organized as follows. In Section 2, we briefly describe the principle of the SVM and its conventional usage in speaker verification. Section 3 presents our proposed framework, which is divided into two parts: binary operation based feature formulation and kernel-based discriminative binary classification. Finally, experiments, conclusions and future work are outlined in Sections 4 and 5, respectively.

2. SVM AND ITS CONVENTIONAL USAGE

Since the utterances vary in duration, the first step for speaker verification is to represent each set of feature vectors as a fixed-length single vector. One of the widely-used approaches is to stack the d-dimensional mean vectors of a K-component adapted GMM into a Kd-dimensional Gaussian supervector [4, 13], which, by considering the channel and session variability, can be further reduced into a lower dimensional vector, the well-known i-vector [12]. After being scaled and normalized, the supervectors or i-vectors can be applied as inputs to the SVM.

2.1. Principle of SVM

As illustrated Figure 1, an SVM is a binary classifier that fits a separating hyperplane between two groups, labeled as "positive" and "negative", when the data are linearly separable [9]. The optimal hyperplane, represented by the solid black line, is chosen according to a maximum margin criterion. That is, the optimal hyperplane is chosen to maximize the distance to the nearest data points on each side of the plane that we call "margin" and minimizes the chance of causing a misclassification nearby as well. The closest data points to the separating hyperplane, which lay on the dashed black lines, are known as the support vectors, since removing them will change the location of the separating hyperplane.

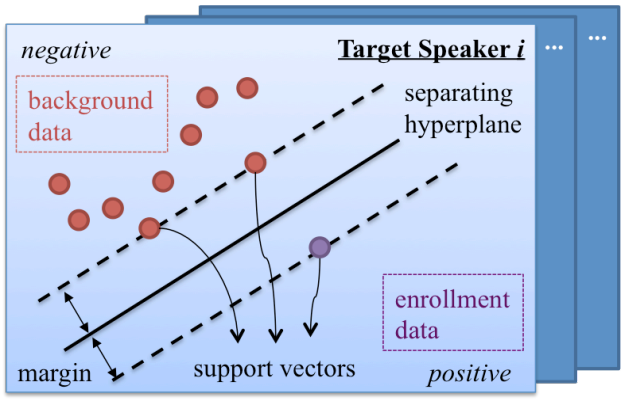

Figure 1. The conventional speaker SVMs.

Moreover, the SVM can reveal its powerful classification ability especially when the data are not linearly separable by using slack variables that allow an example to be in the margin and a kernel function that projects the data to a high-dimensional feature space where the data become linearly separable. In such a case, the problem of margin maximization in the high-dimensional feature space equally turns out to be the optimization of a function of the support vectors, in which "the kernel trick", a method of expanding up from a linear classifier to a non-linear one in an efficient manner, makes the problem tractable [24].

2.2. Speaker SVMs

The goal of the SVM-based speaker verification is to optimally separate the enrollment data (purple circles in Figure 1), i.e., samples of a target speaker, from those of background or imposter speakers (red circles in Figure 1). In the enrollment phase, given one or several utterances spoken by speaker i and labeled as "positive", along with a set of background data consisting of thousands of samples extracted from impostor speakers and labeled as "negative", a speaker-specific SVM is then trained using these samples. This results in support vector selection and weight (i.e., the contribution of a support vector) determination from the enrollment data and background data, which can be used to obtain the verification score while an unknown utterance is claimed to be spoken by speaker i in the test phase [13].

Apart from the problem caused by the insufficiency of positive data, which might be much more critical in generative models than that in discriminative models, we can still see that, when the positive group contains only one example, as shown in Figure 1, the support vector selection is strongly dominated by the nearest negative examples to the positive examples.

3. OUR PROPOSED METHOD

We propose an alternative way to apply kernel-based classifiers to speaker verification. As a member of the typical pattern recognition techniques, our method also contains two parts: a feature formulation mechanism and a classification model.

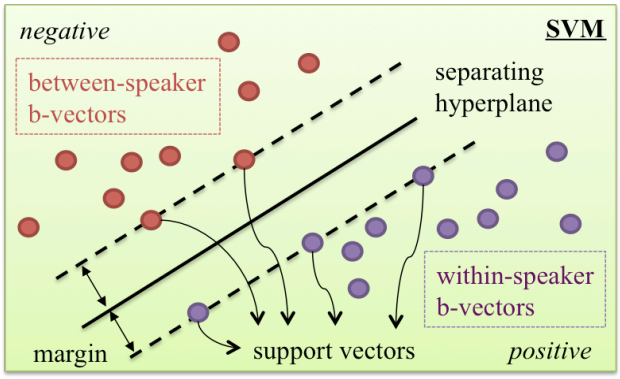

Figure 2. The b-vector based SVM.

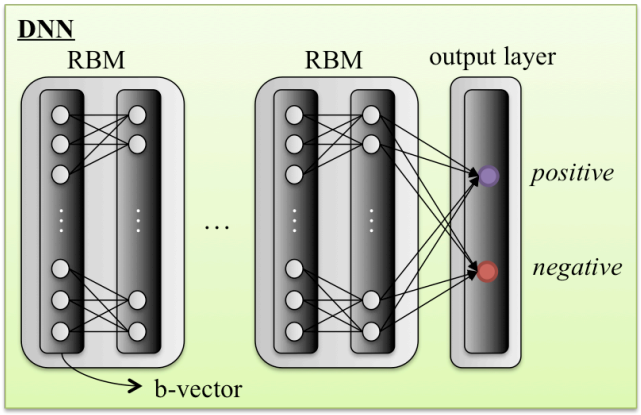

Figure 3. The b-vector based DNN.

3.1. Feature formulation by binary operations

As mentioned in Section 1, the first step for a classifier to indicate whether a pair of samples belongs to some class or not is to take this pair as a new sample, which contains some information about the relationship between the two samples. Given two vectors $\boldsymbol{w}_1$ and $\boldsymbol{w}_2$, which represent two utterances u_1 and u_2, respectively, we attempt to find a binary operation $B(\boldsymbol{w}_1, \boldsymbol{w}_2)$ that not only applies to any two d-dimensional vectors $\boldsymbol{w}_1 = [w_{11}, \dots, w_{1d}]^T$ and $\boldsymbol{w}_2 = [w_{21}, \dots, w_{2d}]^T$ but also serves as a map $B: W \times W \to W$, where W is a nonempty set, such that B is defined for every pair of elements in W, and *uniquely* associates each pair of elements in W [25]. Different from some ensemble-based learning models or score fusion techniques, the binary operation based feature representation is not a combination of similarity measures or discriminant scores that are generated from several weak learners or subsystems; on the contrary, it serves as an information package that contains raw materials that describe the relationship between two samples, so that a complex classifier, such as kernel machines, can unpack it and take out what is helpful for classification.

In this paper, we use three kinds of basic binary operations, $\boldsymbol{w}_1 \oplus \boldsymbol{w}_2$, $\boldsymbol{w}_1 \otimes \boldsymbol{w}_2$, and $|\boldsymbol{w}_1 \ominus \boldsymbol{w}_2|$, where $\oplus$, $\otimes$, and $\ominus$ denote element-wise addition, multiplication, and subtraction, respectively, and $|\cdot|$ denotes the element-wise absolute value function. For example, $[(w_{11} + w_{21}), \dots, (w_{1d} + w_{2d})]^T$ is the result of $\boldsymbol{w}_1 \oplus \boldsymbol{w}_2$. Any combinations of the above operations can be simultaneously performed to form a higher-dimensional vector by augmentation. For instance, with addition and multiplication, the binary operation-derived vector $\boldsymbol{b}_{12}$, b-vector for short, can be expressed by $[(\boldsymbol{w}_1 \oplus \boldsymbol{w}_2)^T, (\boldsymbol{w}_1 \otimes \boldsymbol{w}_2)^T]^T$. We can also easily prove that the function that maps $\boldsymbol{w}_1$ and $\boldsymbol{w}_2$ to their corresponding b-vector is one-to-one (injective) and onto (surjective), that means each pair of vectors has its b-vector by all means, and no two distinct b-vectors are produced from the same vector pair. Thus, the b-vector strongly guarantees the uniqueness for each vector pair. It is worth pointing out that, all of above operations are commutative or order independent, that is, $B(\boldsymbol{w}_1, \boldsymbol{w}_2) = B(\boldsymbol{w}_2, \boldsymbol{w}_1)$, and moreover, to avoid the situations of arithmetic overflow and divide-by-zero errors during implementation, the operation of element-wise division is not considered in this paper.

3.2. SVM with b-vectors

The training mechanism for binary discriminative models in the b-vector based scheme is as follows. Let $\boldsymbol{b}_{ij} = B(\boldsymbol{w}_i, \boldsymbol{w}_j)$ be the b-vector related to the pair of $\boldsymbol{w}_i$ and $\boldsymbol{w}_j$. The positive and negative examples, P and N, for training the discriminative model can be collected through the labeled portion in the background data. For each pair of vectors, $\boldsymbol{b}_{ij} \subseteq P$ if $\boldsymbol{w}_i$ and $\boldsymbol{w}_j$ belong to the same speaker; in the same way, $\boldsymbol{b}_{ij} \subseteq N$ if $\boldsymbol{w}_i$ and $\boldsymbol{w}_j$ are generated from different speakers. Figure 2 shows the structure of the b-vector based SVM. Given the SVM trained according to the similar principle described in Section 2.1, for each trial, the corresponding enrollment utterance of the target speaker is first combined with the test utterance to form a b-vector. Then, the score output by the decision function of the SVM, signifying the degree to which the b-vector belongs to the "with-speaker" group, is obtained by processing the b-vector into the SVM.

Note that if the background data contain N speakers, each providing at least M conversations, then the total number of background utterances is at least MN, which is also the size of negative samples for training each speaker-specific SVM in Figure 1. However in the SVM shown in Figure 2, the numbers of positive and negative examples amount to at most $N \times C_2^M$ and $C_2^N \times M^2/2$, respectively, where C_k^n denotes the combinatorial number or the binomial coefficient [26]. We can easily see that the number of positives in our proposed SVM is much larger than any of those in the conventional speaker SVMs in Figure 1, and that the data imbalance problem can be well solved with an appropriate choice of M or through a mechanism of random selection.

3.3. DNN with b-vectors

The second discriminative classifier designed for b-vectors is the recent popular deep neural network (DNN) [27]. The DNN is a feed-forward artificial neural network model, which consists of multiple layers of neurons. The neurons in each layer are fully connected to the neurons in the next layer. As shown in Figure 3, the DNN maps the input b-vector of an utterance pair into the posterior probabilities of positive and negative groups, and the verification score is obtained by the posterior probability output from the positive node that embodies the "within-speaker" group.

With a set of training data, the parameters in the DNN are first initialized by restricted Boltzmann machines (RBMs) in a layer-by-layer manner, followed by a logistic layer with Gaussian kernel placed on top of it, and then the back-propagation algorithm fine-tunes those parameters. When the DNN is trained on a small training set, it typically performs poorly on the test data. This "overfitting" is greatly reduced by randomly "dropping out" some hidden units in the feed-forward phase of back-propagation because each neuron is forced to learn effectively due to the "unreliability"

Table 1. *Numbers of pairs in the "within-speaker" (P) and "between-speaker" (N) groups for various settings of R in the male portion of SRE04.*

R	2	3	4	5	6
P	15,109				
N	14,762	22,143	29,524	36,905	44,286

Table 2. *EER (%) of the b-vector based SVM system for various settings of R while using four kinds of combinations of binary operations on SRE05.*

Operations	$R = 2$	$R = 3$	$R = 4$	$R = 5$	$R = 6$
$\oplus$, $\otimes$	**9.57**	9.74	9.99	9.99	9.99
$\oplus$, $\ominus$	10.31	10.25	10.31	10.23	10.22
$\otimes$, $\ominus$	10.64	10.80	10.80	10.72	10.65
$\oplus$, $\otimes$, $\ominus$	10.48	10.30	10.31	10.29	10.23

of other neurons. With the "dropout" technique, even though the test data are mismatched to the training data or corrupted by noises, the DNN can still provide very effective performance. The random "dropout" technique has been verified to give big improvements on many benchmark tasks and set new records for speech and object recognition [28].

4. EXPERIMENTS

All the experiments in this paper were carried out on the male portion of the core condition (1conv4w-1conv4w) in NIST SRE05, where each target speaker provided only one 5-min conversational utterance for enrollment [29]. The evaluation task contains 1,222 true trials and 12,367 false trials. We used equal error rate (EER) and normalized minimum decision cost function (minDCF) as acknowledged metrics for evaluation. With the frame length of 25 ms and the frame shift of 10 ms, speech parameters were represented by a 60-dimensional feature vector of Mel-frequency cepstral coefficients (MFCC) with first and second derivatives appended using a 2-frame window, followed by data distribution-based feature warping with a 300-frame window in order to compensate for the effects of environmental mismatch [30].

A gender-dependent UBM consisting of 2,048 Gaussian components with diagonal covariance matrices and an i-vector extractor with dimensionality 300 were trained using data drawn from SRE04. Each speech utterance is finally tokenized by a length-normalized i-vector [33]. In the conventional speaker SVM system, we take 274 male models in SRE05 to carry out T-normalization [18] and 1,875 male SVM background impostors in SRE04 to train the SVM in the dual form with the radial basis function (RBF) kernel. The 1,875 male utterances, each of which is labeled with one of the 122 male speakers in SRE04, are used for training our proposed b-vector based kernel machines and generating a 300×120 LDA projection matrix that reduces the dimensionality of each i-vector from 300 to 120.

Table 1 shows the size of P and N, which denote the numbers of i-vector pairs in the "within-speaker" and "between-speakers" groups, respectively, in the male portion of SRE04. Since the total number of pairs in the "between-speakers" group, drawn from 122 speakers, are over 1,740,000 such that the training of the SVM becomes intractable, since the derivation of a Gram matrix would be clearly unfeasible. Furthermore, unlike the dot-product kernel, the RBF kernel in the SVM does not theoretically allow us to transform the SVM solver from dual forms to primal forms, and if so, the SVM might lose its discriminatory ability when the data are non-linearly separable [22, 23]. Thus, we tried to reduce the size of the training data by randomly choosing R pairs of i-vectors from each pair of speakers not only to reduce the size of N, but also to make the training data more balanced. In Table 2, we can see that in the SVM system, the best EER is achieved when the b-vectors are formed through binary operations of $\oplus$ and $\otimes$, and $R = 2$, where the training data are most balanced in all situations. We can also see that the results in the combination of $\oplus$, $\otimes$, and $\ominus$ are worse than those in the combination of $\oplus$, $\otimes$. This implies that the binary operation $\ominus$ in the framework might produce redundant features although it is a basic operation for similarity measures.

Table 3. *EER (%) of the b-vector based DNN system for various settings of R while using binary operations $\oplus$ and $\otimes$ on SRE05.*

Operations	$R = 2$	$R = 3$	$R = 4$	$R = 5$	$R = 6$
$\oplus$, $\otimes$	9.81	9.72	**9.33**	9.57	9.50

Table 4. *EER (%) and minDCF for various approaches on SRE05.*

Methods	EER (%)	minDCF
LDA with cosine scoring	12.36	0.71
speaker SVMs	10.07	0.38
b-vector SVM ($R = 2$, $\oplus$, $\otimes$)	**9.57**	**0.43**
b-vector DNN ($R = 4$, $\oplus$, $\otimes$)	**9.33**	**0.48**

Table 3 presents EERs for the DNN system based on b-vectors that are formed by the operations of $\oplus$ and $\otimes$. The best result occurs when $R = 4$, which means that the DNN system is less vulnerable to the data imbalance situation while compared to the SVM system. From Table 4, we can see that in terms of EER, our proposed kernel-based discriminative models outperform the state-of-the-art speaker SVMs and the LDA-based cosine scoring methods. Next by comparing minDCF scores in Table 4, we can notice that both b-vector SVM and b-vector DNN outperform LDA with cosine scoring, whereas slightly underperform speaker SVMs. As is shown in [29], the minDCF, defined by NIST SRE05, is obtained by allowing the evaluator, who has access to the true class labels, to choose an optimal threshold at the operating point that inclines to penalizing the false acceptance (type-II error) more than the false rejection (type-I error). Therefore, it is reasonable that speaker SVMs, where the much fewer enrollment data participate in the model training with the result that much more impostor support vectors are produced, can achieve lower minDCF than b-vector SVM and b-vector DNN. However, please be noted that, it is not required to prepare multiple speaker-specific models for both b-vector SVM and b-vector DNN, which considerably reduces the computational complexity.

5. CONCLUSIONS AND FUTURE WORK

This paper has presented a non-probabilistic scheme to directly solve the problem of speaker verification without building an individual model for each target speaker. Based on the idea of binary operations, each pair of utterances can be represented by a single vector that is suitable for any applications involving the comparison between two examples. The integration of such feature vectors and discriminative classifiers has shown to perform well on the standard NIST SRE corpus. In our future work, model adaptation with enrollment data and involvement of feature correlation for formulating b-vectors will be considered. Additionally, some probabilistic approaches, such as PLDA and its modifications [34], give us an insight that careful modeling of the measurement noise in our proposed methods might be beneficial to verification performance.

12. REFERENCES

[1] H. Melin, A. Sandell, and M. Ihse, "CTT-bank: A speech controlled telephone banking system - an initial evaluation," *TMH-QPSR*, 2001.

[2] M. I. Mandasari, M. McLaren, and D. Leeuwen, "Evaluation of i-vector speaker recognition systems for forensic application," in *Proc. Interspeech 2011*, pp. 21-24, 2011.

[3] F. Bimbot, J.-F. Bonastre, C. Fredouille, G. Gravier, I. Magrin-Chagnolleau, S. Meignier, T. Merlin, J. Ortega-García, D. Petrovska-Delacrétaz, and D. A. Reynolds, "A tutorial on text-independent speaker verification," *EURASIP journal on applied signal processing*, vol. 2004, pp. 430-451, 2004.

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

[5] D. A. Reynolds and R. Rose, "Robust text-independent speaker identification using Gaussian mixture speaker models," *IEEE Trans. Speech Audio Process.*, vol. 3, pp. 72-83, 1995.

[6] D. A. Reynolds, T. F. Quatieri, and R. B. Dunn, "Speaker verification using adapted Gaussian mixture models," *Digital Signal Process.*, vol. 10, no. 1, pp. 19-41, 2000.

[7] J. Naik, L. Netsch, and G. Doddington, "Speaker verification over long distance telephone lines," in *Proc. the International Conference on Acoustics, Speech, and Signal Processing (ICASSP) 1989*, pp. 524-527, 1989.

[8] M. BenZeghiba and H. Bourland, "User-customized password speaker verification using multiple reference and background models," *Speech Comm.*, vol. 48, no. 9, pp. 1200-1213, 2006.

[9] V. Vapnik, *The Nature of Statistical Learning Theory*, New York: Springer-Verlag, 1995.

[10] S. Haykin, *Neural Networks: A Comprehensive Foundation*. New York: Macmillan, 1994.

[11] W. Campbell, D. Sturim, D. A. Reynolds, and A. Solomonoff, "SVM based speaker verification using a GMM supervector kernel and NAP variability compensation," in *Proc. ICASSP 2005*, pp. 637-640, 2005.

[12] N. Dehak, P. J. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *IEEE Trans. Audio Speech Lang. Process.*, vol. 19, no. 4, pp. 788-798, 2011.

[13] W. M. Campbell, D. E. Sturim, and D. A. Reynolds, "Support vector machines using GMM supervectors for speaker verification," *IEEE Signal Process. Lett.*, vol. 13, no. 5, pp. 308-311, 2006.

[14] C. H. You, K. A. Lee, and H. Li, "GMM-SVM kernel with a Bhattacharyya-based distance for speaker recognition," *IEEE Trans. Audio Speech Lang. Process.*, vol. 18, no. 6, pp. 1300-1312, 2010.

[15] A. O. Hatch, S. Kajarekar, and A. Stolcke, "Within-class covariance normalization for SVM-based speaker recognition," in *Proc. ICASSP 2006*, pp. 1471-1474, 2006.

[16] C. M. Bishop, *Pattern Recognition and Machine Learning*. New York, NY, USA: Springer, 2006.

[17] W. Rao and M.-W. Mak, "Boosting the performance of i-vector based speaker verification via utterance partitioning," *IEEE Trans. Audio Speech Lang. Process.*, vol. 21, no. 5, pp. 1012-1022, 2013.

[18] R. Auckenthaler, M. Carey, and H. Lloyd-Thomas, "Score normalization for text-independent speaker verification systems," *Digital Signal Process.*, vol. 10, pp. 42-54, 2000.

[19] B. Moghaddam, T. Jebara, and A. Pentland, "Bayesian Face Recognition," *Pattern Recognition*, vol. 33, pp. 1771-1782, 2000.

[20] S. J. Prince and J. H. Elder, "Probabilistic linear discriminant analysis for inferences about identity," in *Proc. the International Conference on Computer Vision (ICCV) 2007*, pp. 1-8, 2007.

[21] P. Li, Y. Fu, U. Mohammed, J. H. Elder, and S. J. D. Prince, "Probabilistic models for inference about identity," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 1, pp. 144-157, 2012.

[22] S. Cumani, N. Brummer, L. Burget, and P. Laface, "Fast discriminative speaker verification in the i-vector space," in *Proc. the International Conference on Acoustics, Speech, and Signal Processing (ICASSP) 2011*, pp. 4852-4855, 2011.

[23] S. Cumani and P. Laface, "Analysis of large-scale SVM training algorithms for language and speaker recognition," *IEEE Trans. Audio Speech Lang. Process.*, vol. 20, no. 5, pp. 1585-1596, 2012.

[24] B. Scholkopf and A. Smola. *Learning with Kernels*. MIT Press, 2002.

[25] E. W. Weisstein, "Binary operation," from *MathWorld--A Wolfram Web Resource*, http://mathworld.wolfram.com/BinaryOperation.html.

[26] Eric W. Weisstein, "Binomial coefficient," from *MathWorld--A Wolfram Web Resource*, http://mathworld.wolfram.com/BinomialCoefficient.html.

[27] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural Comput.*, vol. 18, pp. 1527-1554, 2006.

[28] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," *arXiv*, 2012.

[29] The NIST Year 2005 Speaker Recognition Evaluation Plan, http://www.nist.gov/speech/tests/spk/2005/index.htm.

[30] J. Pelecanos and S. Sridharan, "Feature warping for robust speaker verification," in *Proc. Odyssey: Speaker Lang. Recognit. Workshop 2011*, pp. 213-218, 2001.

[31] N. Brummer and E. de Villiers, "The speaker partitioning problem," in *Proc. Odyssey: Speaker Lang. Recognit. Workshop 2011*, pp. 194-201, 2010.

[32] T. Stafylakis, P. Kenny, M. Senoussaoui, and P. Dumouchel, "PLDA using Gaussian restricted Boltzmann machines with application to speaker verification," in *Proc. Interspeech 2012*, 2012.

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

[34] P. Kenny, "Bayesian speaker verification with heavy–tailed priors," in *Proc. Odyssey: Speaker Lang. Recognit. Workshop 2010*, 2010.