AN I-VECTOR BASED DESCRIPTOR FOR ALPHABETICAL GESTURE RECOGNITION

You-Chi Cheng[1], *Ville Hautamäki* [2], *Zhen Huang*[1], *Kehuang Li*[1], *Chin-Hui Lee*[1]

[1] Georgia Institute of Technology, Atlanta, GA 30332, USA
[2] School of Computing, University of Eastern Finland, Finland, FI-80101

ABSTRACT

An i-vector approach to extracting features for video camera based gesture recognition is proposed. Conventional low-level raw features, such as position, speed, and acceleration, are low-dimensional feature representations which often suffer from measurement noise and thus are not highly discriminative. High-level features, such as Fourier descriptor, usually take a global transformation on the whole raw features of a gesture, but local statistical information is seldom considered. Moreover, compared with speech recordings, video cameras used to capture data are often at a low frame rate such that it is challenging for proper modeling and recognition. In this paper, we show that the proposed i-vector framework can handle both local statistical information and sparse trajectory representations more efficiently under the sparse data scenarios for an in-car hand-gesturing English letter recognition system. Experimental results confirm the effectiveness of the proposed i-vector features, which can reduce the letter error rate by as much as 36-44% relatively from the results obtained with the conventional location based raw features.

Index Terms— gesture recognition, i-vector, hidden Markov model, support vector machine, orthonormal basis

1. INTRODUCTION

Human computer interface (HCI) systems are drawing increasing attention recently due to the increasing requirement to naturally interact with multiple software or hardware systems in our daily lives. Multi-modal inputs [1] should be considered for an HCI system to enable natural ways of interacting with users. Recently, gesture recognition is becoming more important because it provides some side information that is not easily gained by speech recognition.

To establish an automatic gesture recognition (AGR) system with continuous data sequences, we can use many popular sequential modeling techniques that have been studied in the community of automatic speech recognition (ASR) [2]. Among them, hidden Markov model (HMM) [3] has been reported to be useful for AGR as well [4, 5, 6]. Thus we will use it as our baseline for this study.

In this study, the main focus is to develop a statistical feature representation that is capable of describing the detailed local gesture structure such that the discriminative power can be enhanced for gestures of finger-written English letters. It is important to explore these kind of features because conventional gestural features mostly rely on the raw trajectory information, such as position and velocity sequences [4, 5, 6], and may not be as strong as speech feature vectors in terms of their discriminative power.

Conventional features for gesture recognition can be roughly categorized into two types. The first one contains the previously mentioned raw features of a trajectory, the second one includes the global transformation or statistics of the whole trajectory, such as Fourier descriptors (FD) [7, 8], histogram of trajectory orientations [9], or velocity profiles [10, 11]. Although some successful results are reported, these features seldom consider local information such as sub-unit relationships. For example, the local difference between letter M and N only lies in their last strokes, "↗↓ move-out-stroke" and "↑ move-out-stroke", respectively. This kind of information should be emphasized for better distinguishing letter M and N but is not considered by these global features.

In this paper, we propose to use a statistical feature called i-vector, mostly used in speaker verification literature [12, 13, 14, 15, 16], to represent the desired local statistical information by aligning most likely state boundaries with a Viterbi decoder for HMM [3] and extracting corresponding i-vector for each state. If each HMM has 8 states, for instance, 8 i-vectors will be extracted. We can simply concatenate them to form a big vector to represent a gesture so that classifiers such as support vector machines (SVMs) [17] can be adopted to train a multi-class classifier for i-vector features.

Also, by comparing the i-vector with super-vector representation and the reduced dimension representation using principal component analysis (PCA), we found it will be highly beneficial to orthonormalize the projection matrix of the i-vector extraction process. With this modification, the proposed modified i-vector framework can reduce the letter error rate by up to 36-44% compared to the baseline HMM system on two different data sets.

2. RELATED WORK

The main focus of this work, as mentioned in the introduction, is on building meaningful statistical features for AGR systems. Details for previously mentioned three categories of gesture features are reviewed in this section.

To extract raw features for a gesture trajectory, clues such as color and foreground-background correlation [18] are used to track the hand location sequences followed by proper normalization [6]. Many derived quantities such as velocity, acceleration can then be used. Despite of its easy representation, reasonable results are reported in many indoor gesture recognition systems [4, 5]. However, under noisy environments, object detection is subject to error and extracted gesture trajectory can be questionable for proper modeling [6]. Moreover, no structural or geometric information is explicitly considered, while the advantages of applying local information such as strokes are reported [6]. For example, when whole-letter HMM is applied, the confusion of letter P and D can be observed. It can be further disambiguated by considering the intersection points between curvy stroke (◟ or ◞) and the straight stroke (↑).

Another way to describe a trajectory based gesture is to use global transformations on the data. The histogram of eight different orientations are adopted and histograms for these directions are pooled and fed into a k-nearest neighbor classifier [9]. In addition, Fourier descriptor [7, 8] can also be used as a global feature by resampling the original trajectory points to fixed number of points and

978-1-4799-2893-4/14/$31.00 ©2014 IEEE

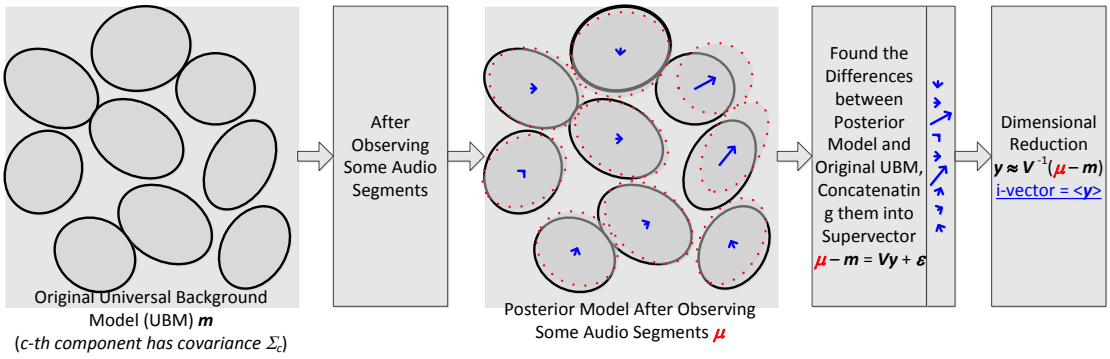

Fig. 1. Concept of the i-vector technique

applying Cosine mapping to alleviate the non-stable results caused by non-periodic characteristics of the trajectory. Fourier transform is then performed on the re-sampled trajectory data or related quantities. These methods are considered as global features and a single feature vector would be generated for each gesture.

Alternatively, we may build the intermediate level local statistical feature by i-vector [12, 13, 14, 15, 16]. It was shown to be highly discriminative for many speaker related problems compared to other statistical features. As far as we know, this technique was mostly used on speaker related works as well as audio concept detectors [19], and was not yet explored in other communities.

3. I-VECTOR

In this section, we will first review the i-vector and introduce the strategy on how we use this technique for our AGR system.

3.1. Conventional I-Vector

The i-vector [12], successfully adopted to the recent NIST Speaker Recognition Evaluation (SRE) tasks [13, 14, 15, 16], is a statistical feature extraction technique developed from joint factor analysis (JFA) [20] and is similar in flavor but more general when compared to probabilistic principal component analysis (PPCA) [21]. The basic idea is to represent the expected variation of the posterior mean vector compared with the mean vectors of a set of reference Gaussian mixtures in a reduced space.

Assume the whole feature space can be roughly described by a Gaussian mixture model (GMM) with C mixture components, a super-vector of these C mixtures can be formed by concatenating mean vectors of these Gaussian densities according to a set of predefined mixture indices. This GMM is called universal background model (UBM) [22] for the feature space. The corresponding super-vector formed by concatenating mean vectors of all mixtures is denoted by $\boldsymbol{m}$. The i-vector is formulated as:

$$\boldsymbol{\mu} = \boldsymbol{m} + \boldsymbol{V}\boldsymbol{y} + \epsilon, \tag{1}$$

where ϵ is a Gaussian noise term with zero mean and a diagonal covariance matrix.

The core of i-vector is to find a projection matrix $\boldsymbol{V}$ so that, after observing a data segment, the expected change of the posterior mean statistic in the super-vector domain when compared to the UBM can be represented in a reduced space effectively. Unlike most dimensional reduction techniques such as PCA which focus on the component analysis of its covariance matrix, i-vector scheme focuses on the posterior statistical behavior. As shown in Fig. 1, it is natural to claim that the posterior membership of a set of input feature vectors should only concentrate on a small subset of the UBM mixtures. Therefore, $(\boldsymbol{\mu} - \boldsymbol{m})$ should be a sparse vector in the super-vector domain. This is why performing a dimension reduction can emphasize these differences and thus have a potential to increase the discriminative power. The actual posterior super-vector $\boldsymbol{\mu}$ however, is a latent variable due to the fact that the actual membership of each feature vector to what specific mixtures is unknown. So the i-vector is defined as the expectation of the vector $\boldsymbol{y}$ in an expectation and maximization (EM) framework [23, 24, 25].

By denoting the feature vector dimension as D, target i-vector dimension as r, the dimension of the UBM super-vector $\boldsymbol{m}$ and posterior super-vector mean $\boldsymbol{\mu}$ will be $CD \times 1$, and in general $r \ll CD$. The target i-vector dimension r, also viewed as the rank of the projection matrix $\boldsymbol{V}$, is a design parameter of this technique. It is reasonable to guess it by doing eigen value analysis on data covariance matrix evaluated in the super-vector domain.

To calculate the i-vector under latent characteristic of posterior super-vector $\boldsymbol{\mu}$, assume the super-vector UBM mean and covariance matrix correspond to the c^{th} mixture are $\boldsymbol{m}_c$ and $\boldsymbol{\Sigma}_c$, respectively. With $\boldsymbol{o}_t$ being the feature vector at the t^{th} frame, and $\gamma_t(c)$ being the posterior probability of the mixture c after observing $\boldsymbol{o}_t$, the training of the projection matrix $\boldsymbol{V}$ can be done in the following way [12, 25]:

1. Randomly initialize $\boldsymbol{V}$,
2. For each gesture s with T_s frames from N training gestures, estimate Baum-Welch statistics with Eq. (2) and (3),
3. Estimate expected i-vector for each gesture by Eq. (4),
4. Estimate $\boldsymbol{V}_c$, the component of $\boldsymbol{V}$ corresponds to the c^{th} mixture of UBM with Eq. (5),
5. Iterate until stop criteria, say 10 iterations, for V are met.

The effective count for mixture c:

$$N_c(s) = \sum_{t=1}^{T_s} \gamma_t(c), \tag{2}$$

and the expected changes on mixture c:

$$\boldsymbol{F}_c(s) = \sum_{t=1}^{T_s} \gamma_t(c)(\boldsymbol{o}_t - \boldsymbol{m}_c), \tag{3}$$

and the iterative update equations:

$$\langle \boldsymbol{y}(s) \rangle = (\boldsymbol{I} + \sum_{c=1}^{C} N_c(s) \boldsymbol{V}_c^* \Sigma_c^{-1} \boldsymbol{V}_c)^{-1} \cdot (\sum_{c=1}^{C} \boldsymbol{V}_c^* \boldsymbol{\Sigma}_c^{-1} \boldsymbol{F}_c(s)), \quad (4)$$

$$\boldsymbol{V}_c = (\sum_{s=1}^{N} \boldsymbol{F}_c(s) \langle \boldsymbol{y}^*(s) \rangle) \cdot (\sum_{s=1}^{N} N_c(s) \langle \boldsymbol{y}(s) \boldsymbol{y}^*(s) \rangle)^{-1}. \quad (5)$$

Once the UBM and the projection matrix $\boldsymbol{V}$ are ready, we can compute i-vector for each testing segment with Eq. (4). In this study, ALIZE toolkit [26] is used.

3.2. Proposed Modifications on I-Vector for AGR systems

Despite its effectiveness reported in the speaker verification task, directly apply this GMM-based technique for all observed trajectories will not keep the structural information of the underlying gestures, as we will see its negative impact in Section 5.

To resolve this problem, the direct solution is inspired by an original work of i-vector [25] and similar strategies [27, 28]. For each gesture, we can just apply the i-vector on the Viterbi aligned state sequence. That is, we first use standard HMM recognition to get the state sequence given by the model with maximal log likelihood score and apply i-vector extraction on segment aligned to each state, and eventually concatenate i-vectors generated by these states to form a super-vector of the i-vectors aligned to these states. We call it "one i-vector per state" strategy in the following paragraphs.

An induced problem is how to select the UBM for each state. One easiest way is to pool all mixtures from the same state of different models and assume these mixtures are the only candidates to be chosen within this specific state. Or we can make a tied-mixture system and apply this tied-mixture pool as UBM mixtures. In this work, we took the former one for it can be computed more efficiently and the results are still quite reasonable.

Besides doing HMM state alignment and extract i-vectors corresponding to these states, an alternative solution is to use a sliding window to extract i-vectors so that we will have a set of i-vectors for each gesture.This method will be more computationally expensive and unfortunately cannot provide extra gain so we will not discuss it here. Therefore, the "one i-vector per state" strategy will be used as our primary feature to be fed into SVM classifiers.

Moreover, note that in Eq. (4), the column vectors of the projection matrix $\boldsymbol{V}_c$ are not guaranteed to form an orthonormal basis. As will be discussed in Section 5, it is quite crucial to orthonormalize these vectors. We believe this is because when these column vectors are not mutually independent, they may not be able to achieve the best representation of the original posterior variation in low dimensional space. Moreover, since classifiers based on Euclidean distance are sensitive to the data scales, the normalization process is also believed to be able to make the resulting i-vector more robust to estimation noise. For practical data, we found the $\boldsymbol{V}_c$ matrix can have small norms for some of its column vectors and the inner product terms for these column vectors with other column vectors can sometimes be even at scales larger than their norms. This make it more sensible to ensure the projection matrix $\boldsymbol{V}_c$ to have orthonormal column vectors.

4. CLASSIFIERS FOR AGR SYSTEMS

In this section, we will introduce the classifiers used in our system. For sequence recognition with raw trajectory features, we adopted HMMs, and for vector-based classification with i-vector features, SVMs were utilized.

4.1. Hidden Markov Model

As mentioned in previous studies [4, 5, 6], HMM is one of the most intuitive ways for sequence modeling. A continuous density HMM is the most commonly used model for ASR, which solves the following problem:

$$\arg \max_{\lambda_k \in \Lambda} \{\log[P(\boldsymbol{O}_s|\lambda_k)]\}, \quad (6)$$

where $\boldsymbol{O}_s$ is a sequence of feature vectors of the s^{th} test gesture and model λ_k, representing gesture class k, is a member of the available model set Λ. The probability term $P(.)$ can be decomposed into the product of discrete state transition probability terms and continuous state observation density terms usually modeled by GMMs. In the present work, each λ_k represent a whole-letter gesture corresponding to a pre-defined writing order.

One advantage of HMM is its capability to segment the continuous data into several states. As mentioned in Section 3.2, this state alignment information can be used to extract i-vector for each state accordingly. Hidden Markov model toolkit [29] is used for modeling and recognizing HMM based AGR.

4.2. Support Vector Machine

SVM [17] is widely used in information retrieval problems [30, 31]. Its soft margin version with regularized penalty term is most widely used, which is shown in the following:

$$\min_{\boldsymbol{w},\xi,b} \{\frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{n} \xi_i\}$$
$$\text{subject to } y_i(\boldsymbol{w} \cdot \boldsymbol{x} - b) \geq 1 - \xi_i, \forall i = 1, ..., N, \quad (7)$$

where $y_i \in \{-1, 1\}$ is the class label of the i^{th} sample for a specific target class. Therefore, multiple SVMs are used for a multi-class problem. Once the feature vector $\boldsymbol{x}$ was extracted, it could be applied in a straight-forward way. In this study, we adopted this linear SVM classifier implemented using LIBSVM tool [32].

5. EXPERIMENTS AND RESULTS

We designed a series of experiments to verify the proposed methods. Two in-car data sets, data sets 1 and 2, were used. We first present the common experimental setups for all experiments.

5.1. Common Experimental Setups

The two data sets were recorded with Dragonfly 2 camera at 15 fps frame rate. The controlled set 1 was collected in an uncontrolled manner from 17 users at three different times, morning, noon, and evening, respectively. In addition, set 2 was collected with 10 users only around noon time with controlled lighting conditions, but all gestures in data set 2 did not share the same stroke orders as those in data set 1. Moreover, in set 2, it may have different stroke orders for the same letter, for example, we observed 6 different ways to express the gesture letter "E". About 3,800 and 3,500 available

Table 1. Results for Preliminary Experiments

Feature Type	Dimension	Classifier	Set 1 (1,877) LER(%)	Set 2 (1,717) LER(%)
Raw Feature	6	HMM	14.86	7.16
FD (Global Feature)	64	SVM	26.52	13.80
One i-vector/gesture (Global Feature)	64	SVM	16.41	11.07
	100	SVM	16.46	11.01

labeled gesturing sequences were selected for sets 1 and 2, respectively. We then randomly chose half of the data for training and the rest for testing, with training and testing data sharing the same pool of users. These two data sets were evaluated separately to check the robustness of the i-vector features for AGR systems.

The raw features were consisted of 6-dimension data: the scaled position, corresponding smoothed velocity and acceleration. The position coordinates were scaled into a square region with both axes normalized to be within $[0, 1]$. A smoothing window of size 3 is used for computing both the velocity and acceleration in both axes.

As for HMMs, we chose 8 states and 2 mixtures per state because the configuration with large number of states may leave too few data points in each state. We adopted this configuration and found the baseline system to be slightly worse than the configuration mentioned in our previous study [6] with 16-state HMMs and 1 mixture per state on test set 1. However, even starting from this configuration, we were able to outperform the previously reported HMM-based system with i-vector and SVM.

For the i-vector systems, 64 and 100 dimensions were chosen to test the system performance. Although these are fairly small numbers compared to those used in speaker recognition systems [12, 13, 14, 15, 16], we found both dimensions can contribute to more than 99% of the data covariance so it was adequate to represent the potential data variation.

5.2. Preliminary Experiments

First, we performed a set of preliminary experiments to compare global features with the raw trajectory features. For the global feature, we computed 64 dimensional FD features and only used a single i-vector to represent each gesture in both the 64 and 100 dimensional subspaces. These extracted features were then fed into SVM classifiers mentioned above. Compared to the baseline HMM system with raw features, unfortunately, these two global features are inferior to the baseline, as shown in the rows 2 to 4 of Table 1, with sizes of two testing data sets given in the brackets shown in the columns 4 and 5 of the header row. This is because the FD features could capture and treat meaningless move-in and move-out strokes with random orientations as real gesturing parts. By removing them, the FD system can be comparable to the baseline. Moreover, although the single i-vector system seems to be better than FD as a global feature system, it is still inferior to the baseline for ignoring local structural information. The weakness of these global features and the importance of the local statistical information were now verified. In the remainder of the paper, we will only discuss the local features.

5.3. Experiments with Local Features

Next, the local statistical features with i-vector extracted at each state will be examined. We first ran the baseline HMM system to determine the state boundaries of the most likely state sequences. During the training phase, data segments aligned to each state were pooled to train the $\boldsymbol{V}$ matrix for each state and UBM were formed by mixtures belonging to the corresponding state as in Section 3.2.

Table 2. AGR Results for Two Data Sets with Local Features

Feature Type	Dimension	Classifier	Set 1 (1,877) LER(%)	Set 2 (1,717) LER(%)
Raw Feature	6	HMM	14.86	7.16
One i-vector/state	64×8=512	SVM	11.19	8.10
	100×8=800	SVM	10.82	7.34
Super-Vector	2,496	SVM	8.79	5.24
Super-Vector(PCA)	512	SVM	8.79	5.36
	800	SVM	8.74	5.36
One i-vector/state with orthonormal $\boldsymbol{V}$	64×8=512	SVM	8.63	**4.54**
	100×8=800	SVM	**8.31**	4.83

We compared the projected i-vector with the original super-vector, estimated $\boldsymbol{\mu} - \boldsymbol{m}$, and the naïve dimensional reduction based on PCA. As shown in the rows 2 and 3 of Table 2, the 64 and 100 dimensional i-vector per state setups can respectively give 24.70%-37.34% relative error reductions for test set 1, but increase the error rate by 2.51%-13.13% for test set 2 relatively. We believe these degradations were due to the non-uniform norm of the non-orthogonal column vectors of the projection matrix $\boldsymbol{V}$ discussed in Section 3.2. As shown in Table 2, the original super-vector in row 4 and the features reduced to 512 and 800 dimensions in rows 5 and 6 are comparable for both test sets and consistently better than the i-vector results in rows 2 and 3. These results not only show that PCA can preserve the discriminative power of the super-vector but verify the importance of ensuring the column vectors of $\boldsymbol{V}$ to form an orthonormal basis. Therefore, as we can see in the last two rows of Table 2, the i-vector with an orthonormalized matrix $\boldsymbol{V}$, again, consistently outperformed all other systems with a relative error reduction of up to 44.08% and 36.59% for test sets 1 and 2 compared to the baseline, respectively. The effectiveness of the proposed scheme is thus verified.

6. CONCLUSION AND FUTURE WORK

We proposed an i-vector based feature extraction framework for gesture recognition of finger-written English letters. We also showed that ensuring the column vectors of the projection matrix $\boldsymbol{V}$ of i-vector extraction to be orthonormal is the key to enhance the discriminative power of the resulting i-vector over original super-vector. With this modified framework, experiments confirm the effectiveness of the proposed scheme can achieve up to a relative 36-44% relative reduction on letter error rates for two different data sets.

We believe the i-vector features will open up new research opportunities in gesture recognition. Although it is designed to only extend whole-letter models, its robustness and strong discriminative power is clearly verified. Different factors could also be studied under the JFA framework for letter modeling. Lighting conditions is one of the most critical sources for AGR errors. User modeling in AGR is also a subject that fits into the JFA framework similar to speaker modeling in speaker verification.

7. ACKNOWLEDGMENTS

The authors would like to thank Robert BOSCH Research and Technology Center for funding support during this research. Work of Ville Hautamäki while visiting Georgia Tech in 2012-2013 was supported by Academy of Finland project 253000.

8. REFERENCES

[1] Z. Zeng, M. Pantic, G. I. Roisman, and T. S. Huang, "A survey of affect recognition methods: Audio, visual, and spontaneous expressions," *Pattern Analysis and Machine Intelligence, IEEE Trans. on*, vol. 31, no. 1, pp. 39–58, 2009.

[2] V.I. Pavlovic, R. Sharma, and T. S. Huang, "Gestural interface to a visual computing environment for molecular biologists," in *Proc. of Automatic Face and Gesture Recognition*. IEEE, 1996.

[3] L.R. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, 1989.

[4] M.Y. Chen, A. Kundu, and J. Zhou, "Off-line handwritten word recognition using a hidden Markov model type stochastic network," *Pattern Analysis and Machine Intelligence, IEEE Trans. on*, vol. 16, no. 5, pp. 481–496, 1994.

[5] F.S. Chen, C.M. Fu, and C.L. Huang, "Hand gesture recognition using a real-time tracking method and hidden Markov models," *Image and Vision Computing*, vol. 21, no. 8, pp. 745–758, 2003.

[6] Y.C. Cheng, K. Li, Z. Feng, F. Weng, and C.H. Lee, "Online whole-word and stroke-based modeling for hand-written letter recognition in in-car environments," in *Proc. of ICASSP*. IEEE, 2013.

[7] S. Impedovo, B. Marangelli, and A.M. Fanelli, "A Fourier descriptor set for recognizing nonstylized numerals," *Systems, Man and Cybernetics, IEEE Trans. on*, vol. 8, pp. 640–645, 1978.

[8] Ø. D. Trier, A. K Jain, and T. Taxt, "Feature extraction methods for character recognition-a survey," *Pattern recognition*, vol. 29, no. 4, pp. 641–662, 1996.

[9] Z.L. Bai and Q. Huo, "A study on the use of 8-directional features for online handwritten Chinese character recognition," in *Proc. of Document Analysis and Recognition*. IEEE, 2005.

[10] F. Hofmann, P. Heyer, and G. Hommel, "Velocity profile based recognition of dynamic gestures with discrete hidden Markov models," *Gesture and Sign Language in Human-Computer Interaction*, pp. 81–95, 1998.

[11] M. Kherallah, L. Haddad, A. M. Alimi, and A. Mitiche, "Online handwritten digit recognition based on trajectory and velocity modeling," *Pattern Recognition Letters*, vol. 29, no. 5, pp. 580–594, 2008.

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

[13] N. Dehak, R. Dehak, P. J. Kenny, N. Brummer, P. Ouellet, and P. Dumouchel, "Support vector machines versus fast scoring in the low-dimensional total variability space for speaker verification," in *Proc. of Interspeech*. ISCA, 2009.

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

[15] V. Hautamäki, K. A. Lee, A. Larcher, T. Kinnunen, B. Ma, and H. Li, "Variational bayes logistic regression as regularized fusion for NIST SRE 2010," in *Proc. of Odyssey: The Speaker and Language Recognition Workshop*, 2012.

[16] V. Hautamäki, Y.C. Cheng, P. Rajan, and C.H. Lee, "Minimax i-vector extractor for short duration speaker verification," in *Proc. of Interspeech*. ISCA, 2013.

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

[18] Y.C. Cheng, Z. Feng, F. Weng, and C.H. Lee, "Enhancing model-based skin color detection: From low-Level RGB features to high-level discriminative binary-class features," in *Proc. of ICASSP*. IEEE, 2012.

[19] Z. Huang, Cheng Y.C., K. Li, V Hautamäki, and C.H. Lee, "A blind segmentation approach to acoustic event detection based on i-vector," in *Proc. of Interspeech 2013*. ISCA, 2013.

[20] P.J. Kenny, G. Boulianne, P. Ouellet, and P. Dumouchel, "Joint factor analysis versus eigenchannels in speaker recognition," *Audio, Speech, and Language Processing, IEEE Trans. on*, vol. 15, no. 4, pp. 1435–1447, 2007.

[21] M.E. Tipping and C.M. Bishop, "Probabilistic principal component analysis," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 61, no. 3, pp. 611–622, 1999.

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

[23] A.P. Dempster, N.M. Laird, and D.B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 1–38, 1977.

[24] C. M. Bishop et al., *Pattern recognition and machine learning*, vol. 4, springer New York, 2006.

[25] P. J. Kenny, G. Boulianne, and P. Dumouchel, "Eigenvoice modeling with sparse training data," *Speech and Audio Processing, IEEE Trans. on*, vol. 13, no. 3, pp. 345–354, 2005.

[26] J.F. Bonastre, N. Scheffer, D. Matrouf, C. Fredouille, A. Larcher, A. Preti, G. Pouchoulin, N. Evans, B. Fauve, and J. Mason, "ALIZE/SpkDet: a state-of-the-art open source software for speaker recognition," in *Proc. of Odyssey: the Speaker and Language Recognition Workshop*, 2008.

[27] C.Y. Lin and H.C. Wang, "Language identification using pitch contour information in the ergodic Markov model," in *Proc. of ICASSP*. IEEE, 2006.

[28] Y. Zhang, Z.J. Yan, and Q. Huo, "A new i-vector approach and its application to irrelevant variability normalization based acoustic model training," in *Proc. of Machine Learning for Signal Processing (MLSP)*. IEEE, 2011.

[29] S. Young, G. Evermann, D. Kershaw, G. Moore, J. Odell, D. Ollason, V. Valtchev, and P. Woodland, "The HTK book," *Cambridge University Engineering Department*, vol. 3, 2002.

[30] T. Joachims, *Text categorization with support vector machines: Learning with many relevant features*, Springer, 1998.

[31] Y. Chen, X.S. Zhou, and T.S. Huang, "One-class SVM for learning in image retrieval," in *Proc. of ICIP*. IEEE, 2001.

[32] C.C. Chang and C.J. Lin, "LIBSVM: A library for support vector machines," *ACM Trans. on Intelligent Systems and Technology*, vol. 2, pp. 27:1–27:27, 2011, Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.