A Blind Segmentation Approach to Acoustic Event Detection Based on I-Vector

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

[1] School of ECE, Georgia Institute of Technology, Atlanta, GA. 30332-0250, USA
[2] School of Computing, University of Eastern Finland, Finland, FI-80101

Abstract

We propose a new blind segmentation approach to acoustic event detection (AED) based on i-vectors. Conventional approaches to AED often required well-segmented data with non-overlapping boundaries for competing events. Inspired by block-based automatic image annotation in image retrieval tasks, we blindly segment audio streams into equal-length pieces, label the underlying observed acoustic events with multiple categories and with no event boundary information, extract i-vector for them, and perform classification using support vector machine and maximal figure-of-merit based classifiers. Experiments on various sets of audio data show promising results with an average of 8% absolute gain in F_1 over the conventional hidden Markov model based approach. An enhanced robustness at different noise levels is also observed. The key to the success lies in the enhanced discrimination power offered by the i-vector representation of the acoustic data.

Index Terms: acoustic event detection, i-vector, blind segmentation, support vector machine, maximal figure-of-merit

1. Introduction

Acoustic event detection (AED) aims at detecting different types of events like speech, music, dog barking, etc. in a long and unstructured audio stream. It has become a challenging part of the multimedia event detection (MED) task conducted annually by National Institute of Standards and Technology's (NIST). The MED evaluation data often consist of uncontrolled, real-life audio recordings obtained at low signal-to-noise-ratio (SNR) enviroments with highly-mixed events in a single acoustic segment. Research in AED [1] is drawing a growing attention recently because it can be an important source of semantic descripion in MED-related tasks [2, 3, 4]. Together with evidence from visual sources, essential information about target multimedia events can be inferred from observed videos. The most popular approach to AED is based on multi-class supervised [1, 5, 6] or unsupervised [7] hidden Markov model (HMM) [8] learning and decoding.

However, in HMM training, some rough boundary information is needed. In the NIST MED task, the data are real-world videos with uncontrolled recording conditions. As a result, the target acoustic events may overlap with each other and mixed with various loud noise sources. Fig. 1 is a typical example structure of those audio streams. The overlapping parts will lead to multiple labels causing problems in HMM learning. Even we ignore the overlapping structure of the events, the manual labeling effort of the temporal information could be still quite labor intensive due to the complicated structures and strong noises. Although correctly labeled data with temporal information can give a good performance for HMM based systems [1, 5, 6, 7] under non-overlapping scenarios, incorrectly labeled time stamps may greatly degrade the system performance. With the increasing number of events, it becomes infeasible to directly label the acoustic data with good time stamps in our AED task.

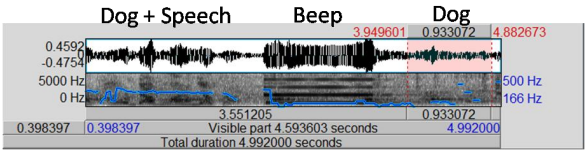

Figure 1: structure of an example audio stream

To handle practical situations, we propose to blindly segment audio clips with equal-length chunks, label each segment with events observed in it without boundary information. That is inspired by an approach to automatic image annotation AIA [9, 10, 11] in which the labeling effort is restricted to just labeling the concerned objects without detail locations in an image. A similar idea is also used in [12], where a spoken utterance (like an audio segment in this paper) is represented by a feature vector and then vector base classifier is used to perform spoken language recognition.

In our system, we train a binary SVM classifier for each particular event with the feature vectors and labels. Then this classifier is used to determine whether a segment contains that particular event. We also adopt the MFoM [13, 14] framework which can handle multi-class multi-label classification well.

One essential part of our system is feature extraction. In this paper, we adopt i-vector, inspired by a recently framework called joint factor analysis [15] widely used in NIST Speaker Recognition Evaluation [16, 17]. It represents an entire audio stream with a relatively low dimensional feature vector while retaining the relevant statistical information. Thus it is considered as a "bag of statistics" feature type. We extract one i-vector for each audio segment and use the i-vector as the feature vector for the segment.

2. I-vector Based Feature Extraction

The i-vector technique [18] is developed based on Joint Factor Analysis [15] and has the flavor of Probabilistic Principal Component Analysis (PPCA) [19]. It is widely used in speaker recognition and verification and is recently reported in the NIST Speaker Recognition Evaluation (SRE) task [16, 17, 20]. Its detailed mathematic explanation can be found in [21]. In this paper, we introduce a different viewpoint to understand it.

Assume the whole audio space can be roughly described by a Gaussian mixture model (GMM) with C components, a

* This work is done while Ville Hautamäki was visiting Georgia Tech.

Copyright © 2013 ISCA

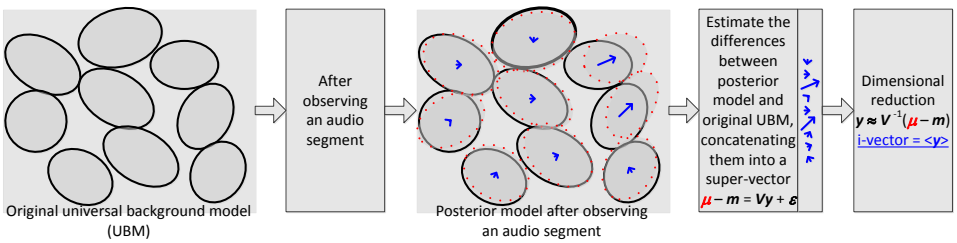

Figure 2: Concept of i-vector technique

super-vector of these C mixtures can be formed by concatenating mean vectors of these Gaussian densities. We called this C-mixture GMM a universal background model (UBM) for the audio space. The corresponding super-vector is denoted by $\boldsymbol{m}$ in the following paragraphs.

One goal of the i-vector technique is to reduce the dimension of the changes of the posterior mean statistics in the super-vector space when compared to the UBM after observing an audio segment. That is, $\boldsymbol{\mu}$ - $\boldsymbol{m} = \boldsymbol{V}\boldsymbol{y}$, where $\boldsymbol{\mu}$ and $\boldsymbol{m}$ are the posterior and original GMM mean super-vectors, respectively; $\boldsymbol{V}$ is the eigenmatrix and $\boldsymbol{y}$ is the i-vector with less dimension than $\boldsymbol{\mu}$ and $\boldsymbol{m}$. It is claimed that an audio segment is only highly related to a subset of the Gaussian densities in the UBM, reflected by the posterior responsibility of certain mixture for generating each feature vector in that audio segment [22]. So the changes of the posterior mean statistics in the super-vector space should be sparse. The actual posterior super-vector $\boldsymbol{\mu}$ is latent due to the fact that the actual membership of each acoustic feature vector to the mixtures is unknown, so the i-vector is defined as the expectation of the vector $\boldsymbol{y}$ in an expectation-maximization (EM) framework [21, 22, 23].

Note that although $\boldsymbol{\mu}$ is not directly observed, the expected changes $\boldsymbol{\mu}$ - $\boldsymbol{m}$ should be still sparse in the super-vector space. This is one reason that performing dimension reduction can emphasize the differences and thus have a potential to increase the discriminative power in the i-vector representation.

The target i-vector dimension, equal to the "rank" of the eigenmatrix $\boldsymbol{V}$, is a design parameter for this framework. Although the concept of i-vector looks quite simple as shown in Fig. 2, the actual algorithm is more complicated because of the latent characteristics of the posterior super-vector $\boldsymbol{\mu}$.

To calculate the i-vector, we assume the UBM mean super-vector and covariance matrix corresponding to the cth mixture are $\boldsymbol{m_c}$ and $\boldsymbol{\Sigma_c}$, respectively, training of the eigenmatrix $\boldsymbol{V}$ can then be performed in the following manner [21]: given $\mathbf{o_t}$ as the feature vector at the tth frame, and $\gamma_t(c)$ as the posterior probability of the mixture component c after observing $\mathbf{o_t}$,

1. Randomly initialize $\boldsymbol{V}$ in Eq. (4);
2. For each segment s with T_s frames from a total of N training segments, compute the Baum-Welch statistics with Eqs. (1) and (2);
3. Estimate the expected i-vector for each file with Eq. (3);
4. Estimate $\boldsymbol{V_c}$, the component of $\boldsymbol{V}$ corresponds to c^{th} mixture of UBM with Eq. (4);
5. Iterate until the stop criteria for $\boldsymbol{V}$ are met.

The effective count for mixture c:

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

and the expected changes on mixture c:

$$\mathbf{F}_c(s) = \sum_{t=1}^{T_s} \gamma_t(c)(\mathbf{o}_t - \mathbf{m}_c). \tag{2}$$

$$\langle \mathbf{y}(s) \rangle = (\mathbf{I} + \sum_{c=1}^{C} N_c(s) V_c^* \Sigma_c^{-1} \mathbf{V}_c)^{-1} \cdot (\sum_{c=1}^{C} \mathbf{V}_c^* \Sigma_c^{-1} \mathbf{F}_c(s)) \tag{3}$$

$$\mathbf{V}_c = (\sum_{s=1}^{N} \langle y(s) \rangle \mathbf{F}_c^*(s)) \cdot (\sum_{s=1}^{N} N_c(s) \langle \mathbf{y}(s) \mathbf{y}^*(s) \rangle)^{-1} \tag{4}$$

Once the UBM and the eigenmatrix $\boldsymbol{V}$ are ready, we can compute the i-vectors for each testing segment with Eq. (3). In this study, we use the ALIZE toolkit [24].

3. Performance Metrics and Classifiers

Next, we describe the classifiers used in this study and the performance metrics used for our evaluation.

3.1. Performance Metrics

To evaluate the results, several performance metrics, such as precision, recall, and F_1, can be used. Given statistics of true positive (TP), false positive (FP), false negative (FN), and true negative (TN), for a specific event i, the class recall R_i, precision P_i and F_{1i} can be computed as in Eqs. (5), (6) and (7), shown below:

$$R_i = \frac{TP_i}{TP_i + FN_i} \tag{5}$$

$$P_i = \frac{TP_i}{TP_i + FP_i} \tag{6}$$

$$F_{1i} = \frac{2TP_i}{2TP_i + FN_i + FP_i} \tag{7}$$

The average performance over all event classes can be summarized by macro- and micro-F_1, F_1^M and F_1^μ, shown below:

$$F_1^M = \frac{2\sum_{i=1}^{K} R_i \sum_{i=1}^{K} P_i}{K(\sum_{i=1}^{K} R_i + \sum_{i=1}^{K} P_i)} \tag{8}$$

$$F_1^\mu = \frac{2\sum_{i=1}^{K} 2TP_i}{\sum_{i=1}^{K} FP_i + \sum_{i=1}^{K} FN_i + 2\sum_{i=1}^{K} 2TP_i} \tag{9}$$

3.2. Support Vector Machine

Support vector machine (SVM) [25] is widely adopted in information retrieval tasks, such as MED [2, 3, 26]. Its soft margin version for risk minimization is shown in the following:

$$\min_{\mathbf{w},\xi,b}\{\frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n}\xi_i\}$$
$$\text{subject to } y_i(\mathbf{w}\cdot\mathbf{x} - b) \geq 1 - \xi_i, \forall i = 1, ..., N \quad (10)$$

3.3. MFoM learning

MFoM is designed to optimize a specific performance metric directly and does well with multi-class multi-label problems [13, 14, 27]. It has been applied to various classification problems such as text categorization [13, 14] and automatic image annotation [27]. The MFoM learning approach uses a differentiable objective function to approximate error counting often required in performance metrics, such as recall, precision, and F_1 defined above. This way, we can optimize certain task-driven performance metrics in different applications.

4. Experiments and Results

To verify the performance of i-vector based classifiers, we conducted 2 different sets of experiments, the first was on the labeled NIST data, and the second was on some artificially-mixed data aiming at simulating the actual scenarios of the future NIST data and demonstrating the essentials of our proposed blind segmentation and i-vector based framework.

4.1. Common settings for the two experiments

In both experiments, one single UBM was trained with data outside the training and testing sets used for the two experiments. A single GMM with 128 mixtures serves as the UBM. The dimension of the i-vector, equal to the "rank" of eigenmatrix V, was chosen to be 64 in this study because it was determined empirically that a larger i-vector dimension would not give much improvement to the classification performance but it greatly increased the computation effort.

In actual scenarios, each acoustic segment may contain multiple labels. For example, both speech and animal sounds may exist in one segment. So we trained a binary SVM classifier for each particular event respectively and we used the classifier to determine whether a segment contains that particular event or not. By combining binary classification results from each classifier we can get the multi-class results. MFoM is originally designed for binary and multi-class classification. We used it here to compare with the SVM-based classifiers. The libsvm toolbox [28] was used for all SVM-related experiments.

HTK [29] was used to build the baseline HMM system using the whole observed segments to train the event HMMs, with 5 states and 15 mixtures per state. Other similar parameter settings gave comparable performance results.

4.2. Experiments on the NIST MED data

We randomly chose a collection of MED video clips from the NIST's TRECVID 2010, 2011, and 2012 MED event kits [26], extracted and segmented the audio streams into 5-second segments for labeling the observed acoustic events. 39 dimension mel-frequency cepstral coefficient (MFCC) [30] vectors were computed with a window size of 25 ms and a 10 ms shift.

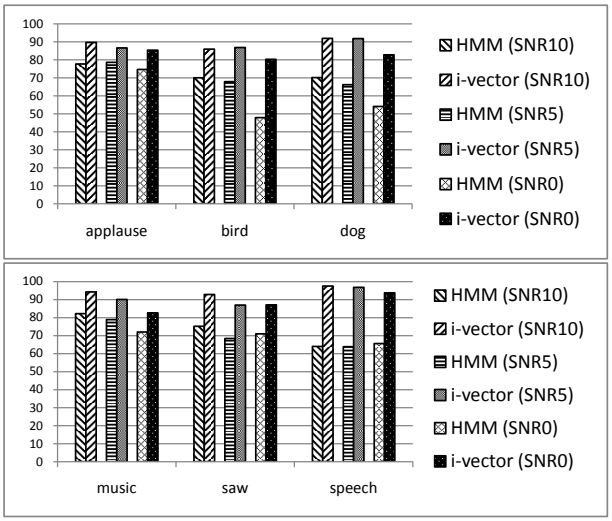

Figure 3: F_1 results (Gaussian SVM) for MED data

There were 6 acoustic events in our categorization framework, as shown in Table 1. We labeled only a total of 1,021 segments.

4.2.1. Experimental setup and results on NIST MED data

We did 5-fold cross-validation for SVM-based classification. For every fold, we used the training segments to compute the eigenmatrix V and then extracted the i-vectors for all the training and testing segments. Table 1 lists the comparison results of the HMM baseline and the proposed i-vector based systems (SVM(L) and SVM(G) denote Linear and Gaussian kernel SVM, repectively).

It was observed that the i-vector systems consistently outperform HMM on the F_1 metric with about an average of 8% absolute gain and were better than HMM for almost all metrics. We also applied linear MFoM [13] to the i-vectors maximizing the macro-F_1 and compared the performance with linear SVM. It is interesting to note that MFoM learning gave considerable improvements over SVM in cases where the positive samples are relatively few, e.g., with 11% for machine sound, and even more seriously at only 4% for animal sound as shown in the rightmost column of Table 1. The macro-F_1 over the 6 categories was 0.46 for SVM and was increased to 0.53 for MFoM.

4.2.2. Potential problems with NIST MED data

To fully explore the proposed framework of blind segmentation and i-vector based AED systems, we found our MED data set suffered from the following 3 shortcomings, namely: (1) there are too few positive samples for some events like "machine sound" and "animal sound". For example, there are only less than 4% positive samples in the total training data. The number of positive data is not enough to train meaningful models for these events; (2) some audio concept labels like "human made sound" and "animal sound" are too general. For example, dog bark and bird sound are both labeled as "animal sound", but we know that they are quite different in their acoustic characteristics. The relatively low performance of these two events shows that more careful categorization of the event concepts is needed; and (3) there may be wrongly labeled segments because categorizing a long segment into one single audio concept can be misleading as shown in Fig. 1.

Table 1: Averaged 5-fold Cross Validation results using HMM, and i-vector based binary SVM and linear MFoM

% / event	recall				precision				F_1				$\frac{\#Positive}{\#Total}$
	HMM	SVM(G)	SVM(L)	MFoM	HMM	SVM(G)	SVM(L)	MFoM	HMM	SVM(G)	SVM(L)	MFoM	
speech	76.37	79.32	83.54	82.49	80.47	84.62	87.85	87.08	78.32	84.08	83.37	84.72	46.43
human voice	64.00	64.39	71.59	66.29	47.32	54.00	62.73	57.95	54.39	61.56	63.55	61.85	25.86
human made sound	64.40	66.18	65.83	60.43	46.88	51.54	56.61	59.57	54.06	57.82	61.02	60.00	27.24
machine sound	58.77	60.52	65.79	64.91	36.10	27.17	43.12	32.74	44.30	38.46	50.36	43.53	11.17
music	81.85	77.99	77.51	72.73	46.98	57.65	69.96	69.22	59.56	66.12	7375	71.19	20.47
animal sound	42.50	60.52	47.37	39.47	16.48	13.14	18.85	23.44	23.46	20.57	28.75	29.41	3.72

4.3. Experiments on artificially-mixed data

To alleviate some of the difficulties mentioned above, we designed a second set of experiments using artificially-mixed audio segments to balance the binary split with about 20% positive and 80% negative data for all event categories and to choose more meaningful event categories with reliable tagging information to experiment on. Moreover, we can add known noise to audio data to simulate low SNR conditions.

We therefore collected audio clips from 6 categories (applause, bird, dog, music, saw, speech) of acoustic events from FindSounds.com [31]. The choice of the 6 categories was based on the availability of data and the fact that they possess different acoustic characteristics and should not be grouped into one single category. The assembled audio clips were divided into 1-second segments, called signal segments, and randomly mixed with 5-second noise segments from the Aurora2 noise database [32]. Each 5-sec mixed segment can contain up to 2 signal segments in our experimental setting. We labeled each mixed segment with the event category using the original signal labels. If one noise segment is not mixed with any signal segment, we labeled it as "noise". We controlled the percentage of the positive samples to be around 20% for each event category. 3 SNR levels, 0 dB, 5 dB and 10 dB, were used and 1,374 segments in 5-sec lengths were obtained for each SNR level. The choice of these levels were made by actually listening to some NIST videos.

4.3.1. Experimental setup and results

We divided the 1,374 segments evenly into training and testing sets. The other experimental settings were the same as those in the NIST MED data experiments described earlier. Fig. 3 displays the F_1 results for each category using HMMs and i-vector based Gaussian kernel SVMs at the three SNR levels.

First, we examine the results closely at the 10dB SNR level for the two leftmost vertical bars in Fig. 3 for each of the 6 categories. Note that the second bar (i-vector based result) for each event is consistently better than that for the first bar (HMM based result). It shows that the i-vector based system achieved a better F_1 at a relatively low SNR level.

Experiments on the 5dB and 0dB SNR data also showed good robustness and performance for the i-vector based systems. They again outperformed the HMM based systems for every event. Let us focus our attention on the category "bird" in the upper middle panel of Fig. 3. When SNR drops from 5dB (the two middle bars) to 0dB (the two rightmost bars), in the HMM experiment, F_1 for the bird category degraded greatly from 67.84% to 47.89%. On the other hand, in the i-vector experiment, F_1 dropped not as significantly.

When comparing F_1^M and F_1^μ in Fig. 4 using linear SVM and linear MFoM, it can also be seen that the results of linear MFoM learning were slightly better than those for linear

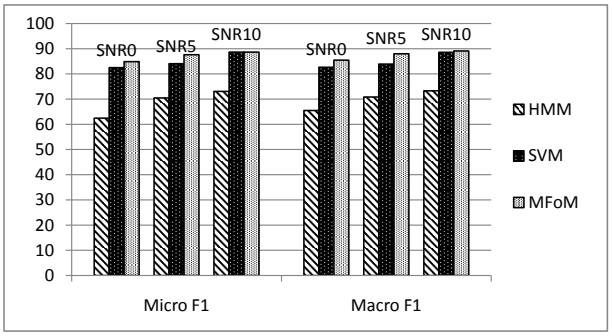

Figure 4: Micro and macro F_1 results for mixed data

SVM, and both SVM and MFoM based classifiers outperformed HMM based classifiers considerably in both macro-average and micro-average F_1 when the ratio of positive to negative training examples is not as severe as shown in Table 1.

5. Summary and Discussion

We have demonstrated the good performance of the proposed blind segmentation and i-vector based approach to audio event detection. By blind segmentation, the difficult issues of manual event labeling and overlapping events are somewhat alleviated. The i-vector based systems with SVM and MFoM based classifiers show much better performance with an average of 8% absolute gain in F_1 than a conventional HMM based system. An enhanced robustness over HMM based systems across low SNR conditions is also observed. For future work, more event categories would be incorporated. Utilizing the AED results from the audio segments in a particular video clip to help describing the semantics of the whole clip would also be studied. We will also investigate Gaussian MFoM learning and compare it with Gaussian SVM.

6. Acknowledgments

The authors would like to thank Dr. Yu Tsao of Academia Sinica for fruitful discussions. This work was supported by the Intelligence Advanced Research Projects Activity (IARPA) via Department of Interior National Business Center contract number D11PC20069. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright thereon. Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, DOI/NBC, or the U.S. Government. Work of Ville Hautamäki was supported by Academy of Finland Project 253000.

7. References

[1] X. Zhuang, X. Zhou, M. A. Hasegawa-Johnson, and T. S. Huang, "Real-world acoustic event detection," *Pattern Recognition Letters*, vol. 31, no. 12, pp. 1543–1551, 2010.

[2] A. G. A. Perera, S. Oh, M. Leotta, I. Kim, B. Byun, C.-H. Lee, S. McCloskey, J. Liu, B. Miller, Z. F. Huan, A. Vahdat, W. Yang, G. Mori, K. Tang, D. Koller, L. Fei-Fei, K. Li, G. Chen, J. Corso, Y. Fu, and R. Srihari, "2011 Multimedia Event Detection: Late-Fusion Approaches to Combine Multiple Audio-Visual features," in *Proc. NIST TRECVID Workshop*, 2011.

[3] N. Inoue, Y. Kamishima, T. Wada, K. Shinoda, and S. Sato, "TokyoTech+ Canon at TRECVID 2011," in *Proc. NIST TRECVID Workshop*, 2011.

[4] A. Kumar, P. Dighe, R. Singh, S. Chaudhuri, and B. Raj, "Audio event detection from acoustic unit occurrence patterns," in *Proc. ICASSP*. IEEE, 2012, pp. 489–492.

[5] R. Cai, L. Lu, H.-J. Zhang, and L.-H. Cai, "Highlight sound effects detection in audio stream," in *Proc. ICME*, vol. 3. IEEE, 2003, pp. III–37.

[6] X. Zhuang, X. Zhou, T. S. Huang, and M. Hasegawa-Johnson, "Feature analysis and selection for acoustic event detection," in *in Proc. ICASSP*. IEEE, 2008, pp. 17–20.

[7] B. Byun, I. Kim, S. M. Siniscalchi, and C.-H. Lee, "Consumer-level multimedia event detection through unsupervised audio signal modeling," in *Proc. INTERSPEECH*, 2012.

[8] 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.

[9] S. Gao, D.-H. Wang, and C.-H. Lee, "Automatic image annotation through multi-topic text categorization," in *Proc. ICASSP*, vol. 2. IEEE, 2006, pp. II–II.

[10] B. Byun and C.-H. Lee, "An incremental learning framework combining sample confidence and discrimination with an application to automatic image annotation," in *Proc. ICIP*. IEEE, 2009, pp. 1441–1444.

[11] I. Kim and C.-H. Lee, "A hierarchical grid feature representation framework for automatic image annotation," in *Proc. ICASSP*. IEEE, 2009, pp. 1125–1128.

[12] H. Li, B. Ma, and C.-H. Lee, "A vector space modeling approach to spoken language identification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 1, pp. 271–284, 2007.

[13] S. Gao, W. Wu, C.-H. Lee, and T.-S. Chua, "A MFoM learning approach to robust multiclass multi-label text categorization," in *Proc. ICML*. ACM, 2004, p. 42.

[14] C. Ma and C.-H. Lee, "A Regularized Maximum Figure-of-Merit (rMFoM) Approach to Supervised and Semi-Supervised Learning," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 5, pp. 1316–1327, 2011.

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

[16] N. Dehak, R. Dehak, P. 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. Interspeech*, 2009, pp. 1559–1562.

[17] 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. Odyssey: the Speaker and Language Recognition Workshop*, 2012.

[18] N. Dehak, P. J. 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.

[19] 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.

[20] D. Garcia-Romero and C. Y. Espy-Wilson, "Analysis of i-vector length normalization in speaker recognition systems," in *Proc. International Conference on Speech Communication and Technology*, 2011, pp. 249–252.

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

[22] C. M. Bishop *et al.*, *Pattern recognition and machine learning*. springer New York, 2006, vol. 4, no. 4.

[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] 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. Odyssey: the Speaker and Language Recognition Workshop*, 2008.

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

[26] A. Vahdat, K. Cannons, H. Hajimirsadeghi, G. Mori, S. McCloskey, B. Miller, S. Venkatesha, P. Davalos, P. Das, C. Xu *et al.*, "TRECVID 2012 GENIE: Multimedia event detection and recounting," in *Proc. NIST TRECVID Workshop*, 2012.

[27] B. Byun, C. Ma, and C.-H. Lee, "An experimental study on discriminative concept classifier combination for trecvid high-level feature extraction," in *Proc. ICIP*. IEEE, 2008, pp. 2532–2535.

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

[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] F. Zheng, G. Zhang, and Z. Song, "Comparison of different implementations of mfcc," *Journal of Computer Science and Technology*, vol. 16, no. 6, pp. 582–589, 2001.

[31] S. V. Rice and S. M. Bailey, "Searching for sounds: A demonstration of Findsounds. com and Findsounds palette," in *Proc. the International Computer Music Conference*, 2004, pp. 215–218.

[32] H.-G. Hirsch and D. Pearce, "The Aurora experimental framework for the performance evaluation of speech recognition systems under noisy conditions," in *ASR2000-Automatic Speech Recognition: Challenges for the new Millenium ISCA Tutorial and Research Workshop (ITRW)*, 2000.