Available online at www.sciencedirect.com

ScienceDirect

Computer Speech and Language 25 (2011) 481–498

COMPUTER SPEECH AND LANGUAGE

www.elsevier.com/locate/csl

Modeling nuisance variabilities with factor analysis for GMM-based audio pattern classification☆

Driss Matrouf[a,*], Florian Verdet[a,b], Mickaël Rouvier[a], Jean-François Bonastre[a,c], Georges Linarès[a]

[a] University of Avignon, Laboratoire Informatique d'Avignon, 84911 Avignon, France
[b] University of Fribourg, Department of Informatics, 1700 Fribourg, Switzerland
[c] Institut Universitaire de France, France

Received 19 October 2009; received in revised form 2 November 2010; accepted 4 November 2010
Available online 18 November 2010

Abstract

Audio pattern classification represents a particular statistical classification task and includes, for example, speaker recognition, language recognition, emotion recognition, speech recognition and, recently, video genre classification. The feature being used in all these tasks is generally based on a short-term cepstral representation. The cepstral vectors contain at the same time useful information and nuisance variability, which are difficult to separate in this domain. Recently, in the context of GMM-based recognizers, a novel approach using a Factor Analysis (FA) paradigm has been proposed for decomposing the target model into a useful information component and a session variability component. This approach is called Joint Factor Analysis (JFA), since it models jointly the nuisance variability and the useful information, using the FA statistical method. The JFA approach has even been combined with Support Vector Machines, known for their discriminative power. In this article, we successfully apply this paradigm to three automatic audio processing applications: speaker verification, language recognition and video genre classification. This is done by applying the same process and using the same free software toolkit. We will show that this approach allows for a relative error reduction of over 50% in all the aforementioned audio processing tasks.

Keywords: Speaker; Language; Video genre; Session variability; Nuisance variability; Joint Factor Analysis; SVM

1. Introduction

Audio Pattern Classification (APC) includes a number of tasks, such as speech recognition, speaker verification, language recognition, emotion recognition and, since recently, genre recognition. In spite of the research efforts in the fields of audio feature extraction and modeling, the APC must face the problems yielded by the change of acoustic conditions, which vary in an unforeseeable way from one recording to another. This phenomenon is generally referred

☆ This paper has been recommended for acceptance by Hugo Van Hamme, PhD.

[*] Corresponding author. Tel.: + 33 4 90843526; fax: +33 4 90843501.

E-mail addresses: driss.matrouf@univ-avignon.fr (D. Matrouf), florian.verdet@etd.univ-avignon.fr (F. Verdet), mickael.rouvier@univ-avignon.fr (M. Rouvier), jean-francois.bonastre@univ-avignon.fr (J.-F. Bonastre), georges.linares@univ-avignon.fr (G. Linarès).

doi:10.1016/j.csl.2010.11.001

to as *nuisance variability* or *session variability* (Kenny et al., 2005; Vogt et al., 2005; Matrouf et al., 2007) and is one of the most important sources of APC performance degradation.

The term *nuisance variability* encompasses a number of phenomena including transmission channel effects, environment noise (other people, cars, TV, etc.), variable room acoustics (hall, park, etc.), the position of the microphone relative to the mouth and the variability introduced by the speaker himself. The solutions proposed in the literature involve works at various levels of the APC (feature space, model space and score space). In spite of using sophisticated feature extraction modules, the nuisance variability introduces a bias in estimated model parameters. This bias can dramatically influence the classification performance. This is mainly caused by the fact that the training databases cannot offer an exhaustive coverage of all the potential sources of nuisance variability.

The usual approaches in statistical classifier training aim at estimating the parameters that characterize the target pattern itself, while the nuisance variability is neither explicitly modeled in this training process, nor implicitly captured from incomplete training corpora. Recently, in the context of the speaker verification task (Bimbot et al., 2004) based on GMM–UBM (Gaussian Mixture Model–Universal Background Model), a Joint Factor Analysis (JFA) paradigm was introduced in order to model the speaker characteristics and the session variability (Kenny et al., 2005; Vogt et al., 2005; Matrouf et al., 2007) at the same time, but as distinct components.

The basic idea behind the JFA paradigm is that, for one recording, the channel component is not estimated on its data alone, but also on a large number of recordings coming from several sessions that belong to different speakers: Let θ_O be a vector composed of the set of model parameters to be estimated on O (O is the set of data frames extracted from a given audio recording). We consider the following model:

$$\theta_O = \theta_{useful} + \mathbf{U}\mathbf{x}_O \tag{1}$$

where θ_{useful} is a vector of parameters that contain the information of interest (speaker, genre, language, etc.). The session component $\mathbf{U}\mathbf{x}_O$ is composed of two terms. The term $\mathbf{U}$, which is a low rank matrix with respect to the size of θ, is estimated using a large amount of data corresponding to different sessions. The term $\mathbf{x}_O$ is a vector characterizing the current session. In other words, the nuisance variability is assumed to be located in a sub-space of low dimension (the range of $\mathbf{U}$) with respect to the dimension of θ.

The success of the nuisance variability JFA modeling depends mainly on the correctness of the hypothesis that the nuisance variability is located in a sub-space of low dimension and on the correctness of the hypothesis that the speaker and channel effects are additive. The very good results obtained in speaker recognition (Kenny et al., 2005; Vogt et al., 2005; Matrouf et al., 2007) show that this hypothesis is satisfied in this task.

In this paper, we apply the same paradigm to three APC tasks, where the nuisance variability does not have the same meaning and can be very different between them: speaker recognition (SR), language recognition (LR) and video genre classification (VGC). For example, the identity of the speaker, which is the information to be modeled in SR systems, is part of the nuisance variability in LR and VGC. In VGC, the background music in some non-music document classes represents a part of nuisance variability, which does not usually exist in SR and LR systems.

In this study, the model we propose is based on the use of the UBM–GMM approach. The UBM represents the world model: all the speakers in SR, all the languages in LR and all the genres for VGC. The parameterization is based on a short-term cepstral representation. The target GMMs are obtained by adapting the UBM, using the Maximum A Posteriori (MAP) approach (Gauvain and Lee, 1994). Only the means of the Gaussians are adapted, the variances and weights remain unchanged with respect to the UBM. In this case, the vector of parameters, θ in Eq. (1), is simply the concatenation of the GMM means, called Super-Vector (SV).

The JFA decomposition algorithm leads to compensated SVs defining GMM models, which can be used to evaluate likelihood of data in the usual way. The compensated SVs can also directly be used in a Support Vector Machine (SVM) classifier (Campbell et al., 2006b). This association between the JFA and SVM allows one to benefit from the JFA decomposition power and SVM's (discriminative) classification power.

In order to facilitate the understanding of the JFA paradigm, we first introduce the notations and the decomposition algorithm in the context of a speaker verification task. For the other applications presented in this paper, language recognition and video genre classification, these terms will be redefined for matching the specific domain. Basically, the algorithm is the same for all the presented applications—with some differences, which will be emphasized when necessary. All the reported experiments are conducted using the free software framework MISTRAL,[1] which in turn relies on the ALIZE library (Bonastre et al., 2005) and on LIBSVM (Chang and Lin, 2001) for SVMs. All these toolkits are freely available to the community. The same paradigm and programs are used for the three APC tasks presented in this paper.

The goal of this article is to show a common theoretical framework employed in three different tasks under the problem of nuisance variability. We focus on UBM/GMM and Joint Factor Analysis for their intrinsic simplicity. Some other statistical approaches (or variants of the previous ones) propose a higher potential but are less suited to our objective. For example, modeling the speaker using the eigen-speaker factors (Kenny et al., 2008) has showed better results than standard MAP adaptation. However, the speaker factors cannot be used for language recognition and video genre classification. We also wish to have the same framework (and the same free software) for the three applications. First experiences of the JFA method applied to language recognition have been proposed in Verdet et al. (2009); Brümmer et al. (2009) and integrating somehow differing strategies in Castaldo et al. (2007); Hubeika et al. (2008).

A similar approach to deal with nuisance variability was proposed by Campbell et al. (2006a). This approach, named NAP (Nuisance Attribute Projection), works in super-vector space and has hence to be used with an adequate classifier, for example SVMs. The JFA presented here works at GMM–UBM and at frame level. In this case the theoretical framework is more interesting, because it allows joint modeling of different kinds of information, which is not the case of NAP-SVM. The full Joint Factor Analysis for speaker recognition is an example of such extended modeling, where, in addition to nuisance variability modeling, a part of the speaker component is constrained to a low dimensional sub-space, and the rest of the speaker component spans the whole SV space (Kenny et al., 2008; Burget et al., 2009). Besides Audio Pattern Classification applications presented in this paper, recently the JFA has been applied to emotion detection (Dumouchel et al., 2009; Kockmann et al., 2009) and to speech recognition (Povey et al., 2010).

2. Definition of presented audio classification tasks

In this paper we will use the JFA paradigm to handle the acoustic variability in three audio classification tasks. The common characteristic for these three tasks is that the classifier can rely the GMM–UBM paradigm. In the following paragraphs, we will define the three tasks we're interested in.

2.1. Speaker recognition

Speaker recognition (SR) encompasses verification and identification. Automatic speaker verification is the use of a machine to verify a person's claimed identity by his voice. The literature abounds with different terms for speaker verification including voice verification, speaker authentication, voice authentication, talker authentication and talker verification. In automatic speaker identification, there is no *a priori* identity claim and the system decides who the person is, or (in the open-set case) that the person is unknown. General overviews of speaker recognition have been given by Atal (1976); Doddington (1985), and Furui (1994). In this paper, the focused SR task is speaker verification.

2.2. Language recognition

Language recognition (LR) consists in processing a speech signal to detect which language the speaker is talking in (Zissman, 1996; Torres-Carrasquillo et al., 2002). It is a detection task, so we are answering questions if an utterance is of a particular language or not.[2] In our case, the set of languages is known *a priori* for determining the score of a language given the speech signal. The information about the other languages may thus also be used to help making a decision. But the decision is open, which means that the system compares the obtained scores to a fixed threshold and can decide that the language is unknown (NIST, 2005, 2009).

[1] The MISTRAL project, open source platform for biometrics authentication. In: http://mistral.univ-avignon.fr.

[2] Intuitively, it is an identification task, especially in the closed-set case. But we treat it as detection problem in the context of the NIST evaluations.

2.3. Video genre classification

Over the last years, the amount of video available on the Internet or digital TV has increased considerably and users need efficient tools to crawl these large collections. This point motivated many works on structuring audiovisual databases by content analysis, mostly using visual-based approaches (Brezeale and Cook, 2008). Some authors investigated text-based categorization (Weiyu Zhu and Liou, 2001), usually based on viewable text or automatic transcription of speech contents. Nevertheless, web data is strongly variable and recognition rates are generally too low to perform a correct analysis of automatic transcriptions. Audio-only approaches could be more robust to both, acoustic context variability and low speech quality, so some authors proposed audio-only methods for video genre categorization. The conventional approach consists in acoustic-space characterization using statistic classifiers like GMMs, neural networks or SVMs based on spectral information (cepstra) (Roach and Mason, 2001; Jasinschi and Louie, 2001; Wang et al., 2003). In contrast to the previous tasks, video genre classification (VGC) is a identification task. The system thus associates the document with a category among those defined *a priori*.

3. Common models and notations

3.1. GMM–UBM

Gaussian Mixture Models (GMMs) are linear combinations of multivariate Gaussian density functions, generally used for approximating a complex probability density function. A GMM is defined by a set of M Gaussians $\mathcal{N}(.|m_g, \Sigma_g)$ along with their associated weights α_g ($g \in 1, \ldots, M$):

$$\sum_{g=1}^{M} \alpha_g \mathcal{N}(.|m_g, \Sigma_g). \tag{2}$$

The GMM–UBM framework is a standard in speaker verification (Bimbot et al., 2004). It is also used in other audio classification tasks, such as language recognition. The UBM is a GMM that represents all the possible observations. It is sometimes also called the world model. For each target pattern (language, speaker, and video genre), a specific GMM is obtained by adapting the UBM *via* the Maximum A Posteriori (MAP) criterion (Gauvain and Lee, 1994). Only GMM means are adapted. The other GMM parameters (variances and weights) are taken from the UBM without any modification. In our experiments, this system is called GMM–UBM. In the same framework, when JFA is applied, the system is called GMM–UBM–FA.

3.2. Mean super-vector

For the JFA paradigm, we need to define the GMM mean super-vector concept. A GMM mean super-vector is defined as the concatenation of the GMM component means. Let D be the dimension of the feature space. The dimension of a mean super-vector is $M \cdot D$, where M is the number of Gaussians in the GMM. In order to ease the understanding of the JFA development, we introduce the following matrix notation: let $\mathbf{A}$ be a $MD \times K$ matrix formed by vertically concatenating M matrices of dimensions $D \times K$. Let us denote by $\mathbf{A}_g$ the g-th matrix in $\mathbf{A}$ (usually corresponding to the g-th component in the model). Let this GMM be parameterized by $\theta = \{m_g, \Sigma_g, \boldsymbol{\alpha}_g\}_{g=1}^{M}$, where $m_g, \Sigma_g, \boldsymbol{\alpha}_g$ are the mean, the covariance matrix and the weight of the g-th Gaussian in the GMM; m denotes the super-vector obtained by the concatenation of the GMM means m_g. Σ is the block diagonal matrix where the g-th diagonal block is Σ_g.

3.3. SVM

In Campbell et al. (2006b), a probabilistic distance kernel that computes a distance between GMMs was proposed. This distance is well suited for a Support Vector Machine (SVM) classifier. Let $\mathcal{X}_{\mathbf{p}}$ and $\mathcal{X}_{\mathbf{p}'}$ be two sequences of speech data corresponding to the patterns $\mathbf{p}$ and $\mathbf{p}'$ (in our case a speaker, language or video genre). The kernel formulation is given below.

$$K(\mathcal{X}_{\mathbf{p}}, \mathcal{X}_{\mathbf{p}'}) = \sum_{g=1}^{M} \left\langle \sqrt{\alpha_g}\Sigma_g^{-\frac{1}{2}} m_g^{\mathbf{p}}, \sqrt{\alpha_g}\Sigma_g^{-\frac{1}{2}} m_g^{\mathbf{p}'} \right\rangle . \tag{3}$$

where α_g, m_g^p and Σ_g are the weight, the mean and the covariance matrix of the g-th Gaussian in the GMM. This kernel is valid when only the means of the GMM models are varying (the weights and covariances are taken from the world model). All the SVM experiments presented in this paper use the kernel shown in Eq. (3).

4. JFA for speaker verification

This section will present in a first part a straightforward implementation of the JFA decomposition algorithm in the context of a speaker verification task in order to facilitate its understanding. In a second part, the setup of the speaker recognition systems and their results will be presented.

In the following, the GMM mean super-vectors $\mathbf{m}_s$ of all speakers s are assumed to be statistically independent (with respect to s) and having a normal prior distribution with mean m and variance $\mathbf{D}\,\mathbf{D}^T = (\Sigma/\tau)$. m and Σ are the parameters of the GMM–UBM as defined in Section 3.2 and τ is the *relevance factor* required in the standard MAP adaptation (Reynolds et al., 2000). The justification of this form concerning the inter-speaker variability can be found in Kenny and Dumouchel (2004). In this case, the random variable $\mathbf{m}_s$ can be written:

$$\mathbf{m}_s = m + \mathbf{D}\mathbf{y}_s, \tag{4}$$

where $\mathbf{y}_s$ is a latent random vector variable distributed according to the standard normal distribution $\mathcal{N}(\mathbf{0}, \mathbf{I})$. Given some adaptation data for speaker s, MAP adaptation consists in estimating the a posteriori distribution of $\mathbf{m}_s$. The a posteriori distribution of $\mathbf{y}_s$ is shown to be normal (Kenny and Dumouchel, 2004). By this fact, the MAP point estimate of $\mathbf{y}_s$ is the mean of this distribution. Actually, this speaker model (Eq. (4)) is equivalent to the one obtained by Reynold's MAP (Reynolds et al., 2000).

Suppose that for a speaker s we have obtained a MAP point estimate m_s of $\mathbf{m}_s$ by using some speaker adaptation data. Given a collection of recordings for the speaker s, let $\mathbf{m}_{(h,s)}$ denote the super-vector corresponding to channel recording h ($h = 1, 2, \ldots$). For a fixed s, assume that all GMM mean super-vectors $\mathbf{m}_{(h,s)}$ are statistically independent (with respect to h). If we assume that the prior distribution of $\mathbf{m}_{(h,s)}$ is normal, then the basic assumption is that there is a matrix $\mathbf{U}$ of low rank such that for each recording h, the prior distribution of $\mathbf{m}_{(h,s)}$ is given by:

$$\mathbf{m}_{(h,s)} = m_s + \mathbf{U}\,\mathbf{x}_{(h,s)}, \tag{5}$$

where $\mathbf{x}_{(h,s)}$ is a latent random vector variable distributed according to the standard normal distribution $\mathcal{N}(\mathbf{0}, \mathbf{I})$. Given some adaptation data for speaker s and channel h (typically one recording), MAP adaptation consist in estimating the a posteriori distribution of $\mathbf{x}_{(h,s)}$. This distribution is shown to be normal. And the MAP point estimate of $\mathbf{x}_{(h,s)}$ is the mean of this distribution.

In order to integrate speaker and channel variabilities in the same framework we will work with the model obtained by substituting Eq. (4) into Eq. (5) (m_s is replaced by its corresponding random variable $\mathbf{m}_s$). Thus the final model is given by:

$$\mathbf{m}_{(h,s)} = m + \mathbf{D}\mathbf{y}_s + \mathbf{U}\,\mathbf{x}_{(h,s)}, \tag{6}$$

where $\mathbf{m}_{(h,s)}$ is the session-speaker dependent mean super-vector, $\mathbf{D}$ is a ($MD \times MD$) diagonal matrix, $\mathbf{y}_s$ the speaker vector (a MD dimensional vector), $\mathbf{U}$ is the session variability matrix of low rank R (a matrix of size $MD \times R$) and $\mathbf{x}_{(h,s)}$ are the channel factors (an R dimensional vector; theoretically $\mathbf{x}_{(h,s)}$ is independent of s).

Both, $\mathbf{y}_s$ and $\mathbf{x}_{(h,s)}$ are assumed to be normally distributed among $\mathcal{N}(\mathbf{0}, \mathbf{I})$. $\mathbf{D}\,\mathbf{D}^T = \Sigma/\tau$ represents the variability of the speaker mean super-vectors. It is the across-class covariance matrix. $\mathbf{U}\mathbf{U}^T$ represents the session variability and is thus the within-class covariance matrix. Hence, as we assume the speaker and the session variabilities to be independent, the total variability is: $\mathbf{D}\,\mathbf{D}^T + \mathbf{U}\mathbf{U}^T$. The prior distribution of the super-vectors (all the $\mathbf{m}_{(h,s)}$) is thus represented by the mean m of Eq. (6) and this total covariance matrix.

Latent variables $\mathbf{y}_s$ and $\mathbf{x}_{(h,s)}$ are assumed to be sampled from Gaussian prior distributions using Eq. (6). These latent variables define a GMM mean super-vector (SV). The observation sequence is assumed to be generated from the GMM that corresponds to this SV. Given the priors on latent variables $\mathbf{y}_s$ and $\mathbf{x}_{(h,s)}$ and given an observation sequence, one can obtain a posterior distribution (and hence MAP point estimates) of $\mathbf{y}_s$ and $\mathbf{x}_{(h,s)}$, which is what we need for enrolling a speaker model and estimating channel factors. Using segments from many speakers, each recorded over several sessions, the model parameters can be estimated using the EM algorithm, where: (a) in E-step, MAP point estimates of $\mathbf{y}_s$ and $\mathbf{x}_{(h,s)}$ are calculated (Eqs. (10) and (11)); $\mathbf{y}_s$ is constrained not to change for segments of the same speaker. (b) in M-step, model parameters are updated (Eqs. (12) and (13)) to increase likelihood of the training data by maximizing the EM auxiliary function (depending on the posterior distribution of the latent variables). In our implementation, we make some approximations: (a) Alignment of frames to Gaussian components is given by the UBM (rather than the two-level generative model itself), which simplifies the mathematic formulation and allows to work only with sufficient statistics (Eqs. (7) and (8)). (b) Only the $\mathbf{U}$ matrix is estimated—m, Σ and weights are copied from the UBM and not updated during M-step, $\mathbf{D}$ is set in an ad-hoc way, so that without $\mathbf{U}$, MAP point estimation of $\mathbf{y}_s$ becomes equivalent to relevance MAP adaptation. (c) MAP point estimates of $\mathbf{y}_s$ and $\mathbf{x}_{(h,s)}$ are obtained using a Gauss–Seidel-like approximation method as proposed by Vogt et al. (2008) rather than estimating the joint posteriors for $\mathbf{y}_s$ and all $\mathbf{x}_{(h,s)}$ corresponding to all sessions of the given speaker.

4.1. Algorithm for JFA model estimation

The success of the JFA model relies on a good estimation of the nuisance variability matrix $\mathbf{U}$, thanks to a sufficiently high amount of data, where several different recordings per speaker are available. In the following, all details allowing the implementation of JFA within the GMM–UBM framework are given. For more information concerning equation derivations, see for example, Vogt et al. (2008).

General statistics: General statistics on the data have to be gathered for estimating the latent variables and the $\mathbf{U}$ matrix of Eq. (6). These are the zeroth-order and first-order statistics with respect to the UBM model.[3]

Let $\mathbf{N}(s)$ and $\mathbf{N}(h, s)$ be vectors containing the zeroth-order speaker-dependent and session-dependent statistics, respectively (both of dimension M for a particular utterance); namely:

$$\mathbf{N}_g(s) = \sum_{f \in s} \gamma_g(f); \quad \mathbf{N}_g(h, s) = \sum_{f \in (h,s)} \gamma_g(f), \tag{7}$$

where $\gamma_g(f)$ is the *a posteriori* probability of Gaussian g for the cepstral vector of observation f. In Eq. (7), $\sum_{f\in s}$ denotes the sum over all the frames that belong to speaker s and $\sum_{f\in(h,s)}$ denotes the sum over all the frames that belong to session h of speaker s.

Let $\mathbf{X}(s)$ and $\mathbf{X}(h, s)$ similarly be the vectors that contain the first-order speaker-dependent and session-dependent statistics, respectively. For an utterance, the dimensions of $\mathbf{X}(s)$ and $\mathbf{X}(h, s)$ are equal to MD:

$$\mathbf{X}_g(s) = \sum_{f \in s} \gamma_g(f) \cdot f; \quad \mathbf{X}_g(h, s) = \sum_{f \in (h,s)} \gamma_g(f) \cdot f \tag{8}$$

Latent variables estimation: The estimation equations for $x(h, s)$ and $y(s)$ in this section represent MAP point estimates of the channel factors $\mathbf{x}_{(h,s)}$ and the speaker vector $\mathbf{y}_s$, respectively (for an easier notation, the indices have been moved into parentheses).

Let $\bar{\mathbf{X}}(s)$ and $\bar{\mathbf{X}}(h, s)$ be the channel- and the speaker-independent statistics, respectively, defined as follows:

$$\begin{aligned} \bar{\mathbf{X}}_g(s) &= \mathbf{X}_g(s) - \sum_{h \in s} \mathbf{N}_g(h, s) \cdot \{m + \mathbf{U}x(h, s)\}_g \\ \bar{\mathbf{X}}_g(h, s) &= \mathbf{X}_g(h, s) \ - \ \mathbf{N}_g(h, s) \cdot \{m + \mathbf{D}y(s)\}_g \end{aligned} \tag{9}$$

where $\bar{\mathbf{X}}(s)$ is used for estimating the speaker vector (session effects are discarded), while $\bar{\mathbf{X}}(h, s)$ is used for estimating the channel factors (speaker effects are discarded).

[3] All the posterior probabilities are computed on the UBM model.

Let $\mathbf{L}(h, s)$ be a $R \times R$ dimensional matrix and $\mathbf{B}(h, s)$ a vector of dimension R, defined by:

$$\begin{aligned} \mathbf{L}(h, s) &= \mathbf{I} + \sum_{g \in \mathrm{UBM}} \mathbf{N}_g(h, s) \cdot \mathbf{U}_g^T \cdot \Sigma_g^{-1} \cdot \mathbf{U}_g \\ \mathbf{B}(h, s) &= \sum_{g \in \mathrm{UBM}} \mathbf{U}_g^T \cdot \Sigma_g^{-1} \cdot \bar{\mathbf{X}}_g(h, s), \end{aligned} \tag{10}$$

where Σ_g is the covariance matrix of the g-th UBM component. By using $\mathbf{L}(h, s)$ and $\mathbf{B}(h, s)$, we can obtain $x(h, s)$ and $y(s)$ MAP point estimates from the following equations:

$$\begin{aligned} x(h, s) &= \mathbf{L}(h, s)^{-1} \cdot \mathbf{B}(h, s) \\ y_g(s) &= \frac{\tau}{\tau + \mathbf{N}_g(s)} \cdot \mathbf{D}_g \cdot \Sigma_g^{-1} \cdot \ \bar{\mathbf{X}}_g(s), \end{aligned} \tag{11}$$

where $\mathbf{D}_g = (1/\sqrt{\tau})\Sigma_g^{1/2}$ (τ is set to 14.0 in our experiments).

Inter-session matrix estimation: The $\mathbf{U}$ matrix can be estimated row by row, with $\mathbf{U}_g^i$ being the i-th row of $\mathbf{U}_g$; thus:

$$\mathbf{U}_g^i = \mathcal{L}(g)^{-1} \cdot \mathcal{R}^i(g), \tag{12}$$

where $\mathcal{L}(g)$ and $\mathcal{R}^i(g)$ are given by:

$$\begin{aligned} \mathcal{L}(g) &= \sum_s \sum_{h \in s} \mathbf{N}_g(h, s) \cdot (\mathbf{L}(h, s)^{-1} + x(h, s)x(h, s)^T) \\ \mathcal{R}^i(g) &= \sum_s \sum_{h \in s} \bar{\mathbf{X}}_g(h, s)[i] \cdot x(h, s) \end{aligned} \tag{13}$$

Algorithm 1 presents the adopted strategy for estimating the nuisance variability matrix $\mathbf{U}$ with the above developments. The estimation of $\mathbf{U}$ is performed using an independent data corpus with several sessions per speaker. In the client training and the testing phases, the components $\mathbf{x}$ and $\mathbf{y}$ are estimated *via* the same algorithm (Algorithm 1) where the $\mathbf{U}$ matrix is fixed (not re-estimated) and only one iteration is performed.

Algorithm 1. Estimation algorithm for $\mathbf{U}$

```
For each speaker $s$ and session $h$:
  $y(s) \leftarrow \mathbf{0}$, $x(h, s) \leftarrow \mathbf{0}$, $\mathbf{U} \leftarrow random$ ($\mathbf{U}$ is initialized randomly)
Estimate statistics: $\mathbf{N}(s)$, $\mathbf{N}(h, s)$, $\mathbf{X}(s)$, $\mathbf{X}(h, s)$ (Eqs. (7) and (8))
for $i = 1$ to $nb\_iterations$ do
  for all $h$ and $s$ do
    Center statistics: $\bar{\mathbf{X}}(h, s)$ (Eq. (9));
    Estimate $\mathbf{L}(h, s)^{-1}$ and $\mathbf{B}(h, s)$ (Eq. (10)) ;
    Estimate $x(h, s)$ (Eq. (11));
    Center statistics: $\bar{\mathbf{X}}(s)$ (Eq. (9));
    Estimate $y(s)$ (Eq. (11)) ;
  end
  Estimate matrix $\mathbf{U}$ (Eqs. (12) and (13));
end
```

4.2. The verification task

In this paragraph we present the strategy used to estimate the verification score using models in which the effect of nuisance variability is compensated. Let $\mathbf{s}_{tar}$ and $\mathbf{s}_{test}$ be the speakers corresponding to the training and testing data respectively. Applying the JFA decomposition to both, training and testing data, one can write:

$$\begin{aligned} m_{(\mathbf{h}_{tar}, \mathbf{s}_{tar})} &= m + \mathbf{D}y_{\mathbf{s}_{tar}} + \mathbf{U}x_{\mathbf{h}_{tar}}, \\ m_{(\mathbf{h}_{test}, \mathbf{s}_{test})} &= m \quad\ \ + \mathbf{U}x_{\mathbf{h}_{test}}. \end{aligned} \tag{14}$$

The channel factors for the test utterance are obtained using the UBM. This is a good approximation (and speedup) to the accurate way which would include $\mathbf{D}y_{\mathbf{s}_{tar}}$ (Glembek et al., 2009). The JFA decomposition (Eq. (14)) for the training and for the testing data is performed independently. In order to perform the verification test in the standard manner using the Log-Likelihood Ratio (LLR) score, the target JFA model (first line of Eq. (14)) is transformed by replacing the session component of the training data with the test session component (coming from the second line of Eq. (14)):

$$m_{(\mathbf{h}_{test},\mathbf{s}_{tar})} = m + \mathbf{D}y_{\mathbf{s}_{tar}} + \mathbf{U}x_{\mathbf{h}_{test}}. \tag{15}$$

The LLR of a given sequence of test speech frames $\mathcal{Y} = \{\mathcal{Y}_1, \ldots, \mathcal{Y}_T\}$ is then given by the following equation:

$$score(\mathcal{Y}|\mathbf{s}_{tar}) = \log\left(\frac{\mathcal{F}(\mathcal{Y}|m + \mathbf{D}y_{\mathbf{s}_{tar}} + \mathbf{U}x_{\mathbf{h}_{test}})}{\mathcal{F}(\mathcal{Y}|m + \mathbf{U}x_{\mathbf{h}_{test}})}\right). \tag{16}$$

where $\mathcal{F}(.|m)$ indicates the likelihood of the data frames given the GMM having m as mean super-vector and where the weights and variances are the same as those of the UBM. Thus, the estimation of the LLR is done using the modified target model of Eq. (15) and the compensated UBM.

4.3. SVM modeling

By using Algorithm 1, the JFA modeling leads to super-vectors that contain both additive terms: the speaker component and the session component. All super-vectors are compensated by discarding the session component, the retained SVs are of the form $m+\mathbf{D}y$. The resulting compensated super-vectors (SVs) are directly used in the SVM classifier described in Section 3.3. All the SVs used by the SVM are compensated: the SVs corresponding to the target speaker, to the blacklist[4] and to the test segments.

4.4. Experimental protocol for speaker recognition

Speaker verification experiments, presented in Section 4.5, are performed on the NIST SRE 2005 database as a development set and on the NIST SRE 2006 and 2008 databases for the validation set. Male speakers only are used (referred to as the 2005, 2006 and the 2008 protocols). The 2005 protocol consists of 274 speakers, 9012 tests (951 target trails, the rest being impostor trials). The 2006 protocol consists of 354 speakers, 9720 tests (741 target trails and the rest impostors). The 2008 protocol consists of 188 speakers, 6615 tests (of which 439 are target trails).[5]

Results are given in terms of Equal-Error-Rate (EER) and minimum of the Detection Cost Function (DCFmin), as defined by NIST (NIST, 2008). Train and test utterances contain an average of 2.5 minutes of speech (telephone conversations, where around 30% of speech frames per speaker have been retained).

The inter-session variability matrix is enrolled on the NIST-SRE-2004 database with 2938 examples of 124 speakers (around 20 iterations for reaching convergence). From the same database, 200 impostor speakers are used for score normalization and as negative examples in the SVM classifier.

The baseline system is a standard GMM–UBM system (Reynolds et al., 2000). The UBM is trained on the Fisher database[6] using the EM (Expectation Maximization) algorithm. The background model has 512 components; the variance parameters of each component are floored to 50% of the global variance (0.5). The training data consists of about 10 million speech frames. The speaker models are derived by Bayesian adaptation on the Gaussian component means with a relevance factor of 14.

The frames are composed of 19 LFCC (linear frequency cepstral coefficient) parameters, their first-order derivatives and 11 second-order derivatives (and the frequency window is restricted to 300–3400 Hz). A normalization process is applied, so that the distribution of each cepstral coefficient is of 0-mean and 1-variance for a given utterance. In Table 1, we show the results of the baseline system with and without T-norm normalization. Here, the Z- and ZT-norms do not bring any improvement (see Bimbot et al., 2004 for more details concerning score normalization).

[4] The set of negative examples used in training SVMs.

[5] The 2005 protocol corresponds to the core condition, labeled as *det7*, the 2006 protocol corresponds to the core condition, labeled as *det3* and the 2008 protocol corresponds to the core condition, labeled as *det7*.

[6] Fisher English Training Speech Part 1, LDC2004S13.

Table 1
Results of the baseline GMM–UBM speaker verification system on the 2005, 2006 and 2008 protocols. DCFmin (×100), EER (%).

	SRE-05		SRE-06		SRE-08	
	DCFmin	EER	DCFmin	EER	DCFmin	EER
No-norm	3.83	7.15%	3.88	6.79%	4.1	8.12%
T-norm	3.05	8.52%	2.90	5.70%	3.2	7.89%

Table 2
Varying the subspace dimension. 2005 Protocol. DCFmin (×100), EER (%). The best rank in terms of the DCFmin is 40.

	Subspace rank					
	0	20	40	60	80	100
DCFmin (× 100)	3.83	2.05	**1.83**	1.93	1.95	1.99
EER (%)	7.15%	5.1%	4.42%	**4.22**%	4.31%	4.23%

Table 3
Score normalization techniques on the JFA model (rank = 40). The ZT-norm always brings an improvement over the baseline. DCFmin (×100), EER (%)

	SRE-05		SRE-06		SRE-08	
	DCFmin	EER	DCFmin	EER	DCFmin	EER
No-norm	1.83	4.42%	1.61	2.97%	2.00	4.06%
T-norm	1.84	4.72%	1.29	2.83%	1.70	3.60%
ZT-norm	1.72	4.62%	**1.18**	**2.15%**	**1.57**	3.60%
Z-norm	**1.64**	**4.21%**	1.46	2.33%	1.96	**3.46**%

4.5. *Experimental results*

In this section, we present in detail the experimental results obtained with the implementation given in this paper. In Table 2 we show the SR performances with respect to the nuisance variability subspace dimension, without any score normalization. The best performing system has been found to have a rank of 40. For higher ranks we observe a stagnation of the performances.

In Table 3 we show the improvement of the JFA systems with score normalization. Applying the Z-norm is the technique that brings an improvement in all protocols, while the T-norm has an effect only in the 2006 and 2008 protocols. Indeed, the DCFmin drops from 1.83 to 1.64 with the Z-norm in 2005, from 1.61 to 1.18 with the ZT-norm in 2006 and from 2.00 to 1.57 in 2008. While the behavior is different in all years, the ZT-norm seems to be the most efficient choice for score normalization.

In Table 4 we show the results obtained using speaker-dependent super-vectors *via* the kernel described in Eq. (3) along with an SVM classifier. This system obtains performances similar to the one presented above for 2005,

Table 4
GMM JFA (rank = 40) super-vectors with distance kernel. 2005, 2006 and 2008 protocols. DCFmin (×100) and EER (%). The T-norm brings the most significant gain.

	SRE-05		SRE-06		SRE-08	
	DCFmin	EER	DCFmin	EER	DCFmin	EER
Baseline (40)	1.97	4.83%	1.40	2.83%	1.79	3.19%
Z-norm	1.92	5.36%	1.54	2.70%	1.62	2.99%
T-norm	1.61	**4.42**%	**1.03**	2.29%	1.45	2.73%
ZT-norm	**1.58**	4.51%	1.06	**2.16**%	**1.37**	**2.73**%

but outperforms the classical system on 2006 and 2008. It is worth noting that here the Z-norm does not bring any gain, which is as expected, as the blacklist plays the role of the Z-norm cohort. We also observe that the T-norm and ZT-norm bring a similar improvements in all years. The improvement over the classical LLR system is from 1.64 to 1.58 DCFmin in 2005, from 1.18 to 1.03 in 2006 and from 1.57 to 1.37 in 2008. Moreover, in terms of the stability of the decision score, if the threshold is tuned on the 2005 protocol, then the DCF on the 2006 protocol would be 1.23 and 1.40 on 2008 protocol. Finally, by comparing the best performance without JFA and the best performance with JFA, we can see that there is a relative DCFmin gain of about 57% under the 2008 protocol, about 64% under the 2006 protocol and about 48% for the 2005 protocol.

5. Video genre classification

The video genre refers to editorial styles, such as commercials, movies, news, and cartoons. The automatic genre classification recently became a challenging task in the field of video indexing, especially in the context of user-generated Web collections. Thus, this task motivated many research efforts these last years, as well as some contests, such as the Trec-Vid evaluation campaign (Smeaton et al., 2006) or the “Robust, As-Accurate-As-Human Genre Classification for Video” challenge[7] organized by Google in 2009.

Most of the proposed approaches rely on image analysis, but some research efforts investigate audio-only approaches based on low-level signal descriptors or on the analysis of the transcribed spoken contents. Nevertheless, speech recognition systems generally fail on unexpected linguistic domains and in adverse acoustic conditions. (Moncrieff et al., 2003) identify the video genre by tracking audiovisual events such as jingles.

Nevertheless, in more general cases, low-level approaches exhibit a better robustness to the highly variable and unexpected conditions that may be encountered in videos. Roach and Mason (2001); Xu and Li (2003); Roach et al. (2002) propose time-domain features, such as zero crossing rates or energy distributions, but most of the methods rely on cepstral analysis and statistic classifiers, such as GMMs, neural networks or SVMs on cepstral domain features (Roach et al., 2002; Jasinschi and Louie, 2001; Wang et al., 2003).

One of the main difficulties in genre classification in the cepstral domain stems from the diversity of the acoustic patterns for each genre. Unlike speaker identification, genre classification involves a relatively small number of classes that are highly variable. Moreover, useful and useless components are mixed into the cepstral domain without any explicit knowledge about the nature of the noise or about the useful information source. In this section, we investigate the efficiency of the JFA method on such a classification task. We test different configurations according to several JFA meta-parameters and to the classification strategy.

5.1. JFA for genre classification

As in the previous section, we use the GMM–UBM (Bimbot et al., 2004) approach for video genre classification. A world model (UBM) is a GMM representing the whole acoustic space, while genre-specific GMMs (for news, movies, cartoons, music and commercials) are obtained by adapting the world model.

In order to take into account the nuisance variability in the modeling process, a genre-specific model can be decomposed into three different components: a genre- and session-independent component, a genre-dependent component and a session-dependent component. With the same notations as in the previous section, the JFA model, for video genre classification (we replace s with GE), can be written as:

$$\mathbf{m}_{(h,GE)} = m + \mathbf{D}\mathbf{y}_{GE} + \mathbf{U}\mathbf{x}_{(h,GE)}, \tag{17}$$

where $\mathbf{m}_{(h,GE)}$ is the session- and genre-dependent mean super-vector, $\mathbf{y}_{GE}$ the genre-dependent vector, $\mathbf{U}$ is the nuisance variability matrix of low rank R (a $MD \times R$ matrix) and $\mathbf{x}_{(h,GE)}$ are the channel factors.

In the training phase, the $\mathbf{U}$ matrix, the $\mathbf{y}_{GE}$ and $\mathbf{x}_{(h,GE)}$ have to be estimated. The $\mathbf{U}$ matrix is estimated on all the genres and with several sessions by genre; the $\mathbf{y}_{GE}$ component is estimated on all the sessions that belong to genre GE; the $\mathbf{x}_{(h,GE)}$ component is estimated on session h, belonging to genre GE, as described in Algorithm 1.

[7] http://googleresearch.blogspot.com/2009/05/acm-multimedia-2009-grand-challenges.html.

The scoring is done by estimating the likelihood of the test data given the target genre model. The session compensation is performed as described in Section 4.2. For the SVM classifier, we use the same kernel as described in Section 3.

5.2. Experimental protocol

We selected seven categories that are commonly involved in video genre classification tasks: news, movies, cartoons, music, commercials, sports and documentary. The corpus is composed of 1610 manually indexed videos (we will refer to them as *files*), with durations of 2–5 min 1050 of them are used for training and 560 are used for testing (about 150 videos per genre for training and 80 for testing). The speech contained in the different documents is in the French language. In our experiments, we use PLP (Perceptual linear prediction) features, extracted using a 25 ms Hamming window with a shift of 10 ms. Each frame is composed of 39 coefficients (13 PLP coefficients along with their first and second-order derivatives). Cepstral mean subtraction is applied on each audio recording. The next subsections describe the different systems used for our experiments.

5.2.1. GMM–UBM and JFA

Given the UBM and a genre-specific utterance, JFA decomposition is performed (Eq. (17)):

$$m_{GE_{tar}} = m + \mathbf{D}y_{GE_{tar}} + \mathbf{U}x_{(h,GE_{tar})},$$

where $y_{GE_{tar}}$ and $x_{(h,GE_{tar})}$ are the MAP point estimates of $\mathbf{y}_{GE_{tar}}$ and $x_{(h,GE_{tar})}$ (see Algorithm 1). The retained (compensated) model for genre GE_{tar} is given by:

$$m'_{GE_{tar}} = m + \mathbf{D}y_{GE_{tar}}$$

The classification scores are estimated as explained in Section 4.2. This system is called GMM–UBM–JFA.

5.2.2. SVM–UBM and JFA

The system based on the SVM classifier, which uses the genre-specific mean super-vectors, is here called SVM–UBM. When the mean super-vectors are obtained using JFA, the system is called SVM–UBM–JFA. For a given genre, the negative examples are recordings belonging to the other genres. The training and testing stages are performed as explained in Section 4.3.

In the case of SVM–UBM–JFA, we have noticed large score differences between genres. These differences yield performance degradation. In order to solve this problem, a score normalization is applied on the output scores as follows:

Let S_g be the score given by the SVM corresponding to genre g for a given audio file. The score normalization is performed as follows:

$$score(g) = \frac{\mathcal{N}_t(S_g|g)}{\mathcal{N}_t(S_g|g) + \mathcal{N}_n(S_g|g)} \tag{18}$$

where $\mathcal{N}_t(.|g)$ and $\mathcal{N}_n(.|g)$ are two Gaussian distributions that represent the target and non-target scores, respectively. They are estimated on the set of scores obtained by a 6-fold testing on the training corpus where 5 parts were used to train the genre models and the remaining part as utterances to obtain the scores.

5.3. Experimental results for VGC

In the following sections, we will study the impact of the UBM size. After that, the best size will be used for the rest of the experiments. For the experiments using JFA, the dimension of the nuisance variability subspace is fixed to 40. For the SVM classifier, two training strategies are proposed. The results will be presented for GMM–UBM (without JFA), GMM–UBM–JFA and for SVM–UBM–JFA. For the GMM–UBM and GMM–UBM–JFA, there is no need of score normalization. For SVM–UBM–JFA, the score normalization is crucial, it is done using the procedure described in Section 5.2.2.

Table 5
Video genre classification error rates with respect to the GMM–UBM size on Documentary (Doc), Music (Mus), News, Commercial (Com), Cartoon (Cart), Movie (Mov), Sport.

	Doc	Mus	News	Com	Cart	Mov	Sport	Total
64	15%	15%	38%	66%	24%	36%	25%	32%
128	14%	15%	37%	62%	24%	32%	24%	30%
256	13%	14%	35%	64%	24%	32%	22%	**29** %
512	15%	14%	38%	62%	24%	32%	24%	30%

Table 6
SVM–UBM–JFA system compared to GMM; Sys-1 is GMM–UBM–JFA, Sys-2 is SVM–UBM–JFA and Sys-3 is SVM–UBM–JFA with score normalization.

System	Doc	Mus	News	Com	Cart	Mov	Sport	Total
Sys-1	7%	3%	13%	50%	17%	11%	10%	16%
Sys-2	4%	2%	21%	52%	14%	10%	13%	17%
Sys-3	7%	3%	18%	8%	18%	16%	20%	**13** %

5.3.1. *Varying the UBM size*

The first experiment studies the effect of the GMM–UBM size (number of Gaussian components) on the genre classification accuracy. Results are presented for model sizes of 64, 128, 256 and 512 components in Table 5. The results show that a GMM–UBM with 256 components yields the best results on the video genre classification with 29% classification errors. We use this configuration as baseline in our subsequent experiments.

5.3.2. *GMM–UBM–JFA*

The GMM–UBM–JFA and the derived SVM systems are run on a U matrix of rank 40. On previous experiments, this subspace dimensionality has proven to give the best results.

The first row of Table 6 shows the results obtained on the GMM–UBM–JFA system (using 256 Gaussians). The performance is strongly improved by JFA in comparison to the baseline GMM–UBM system, with a relative reduction of error rate of about 45% (from 29% down to 16%).

5.3.3. *SVM–UBM–JFA*

Video genre classification is a multi-class problem and since SVMs solve two-class problems, we need to adapt their usage to the multi-class context. We propose to train an SVM for each class to distinguish it from all the other classes. In our case, there are 7 video genres in total. For each genre we thus have an SVM trained with 150 positive super-vectors and 900 ($=6\times150$) negative super-vectors (blacklist).

Table 6 shows the results for this SVM configuration. We observe that the score normalization is clearly required. The relative error reduction yielded by score normalization is about 24%. Compared to the GMM–UBM–JFA system, we observe a relative classification error reduction of about 19%. The system featuring normalization obtains with 13% classification error rate the best VGC performance presented here. This is 55% relative better than the corresponding GMM–UBM system.

6. Language recognition

The focus of this section is language recognition, which consists in processing a speech signal for detecting the language an unknown speaker is talking in. The nuisance variability, in the case of language recognition, covers speaker particularities including vocal tract configuration, current emotion or health status. It also covers recording conditions with background noise, microphone setup, transmission channel and speech signal encoding.

6.1. JFA for language recognition

As in the SR and VGC systems presented in previous sections, the LR system presented here is based on the GMM–UBM approach with MAP adaptation of the means.

In order to take into account the nuisance variability in the modeling process, a language-specific model can be decomposed into three different components: a language- and session-independent component, a language-dependent component and a session-dependent component. With the same notations as in the previous sections, the JFA model for language recognition (replacing *s* by *l*) can be written as:

$$\mathbf{m}_{(h,l)} = m + \mathbf{D}\mathbf{y}_l + \mathbf{U}\,\mathbf{x}_{(h,l)}, \tag{19}$$

where $\mathbf{m}_{(h,l)}$ is the session- and language-dependent mean super-vector, $\mathbf{y}_l$ the language-dependent vector, $\mathbf{U}$ is the nuisance variability matrix of low rank R (a $MD \times R$ matrix) and $\mathbf{x}_{(h,l)}$ are the channel factors.

6.1.1. Training and testing with JFA

In the training phase, the $\mathbf{U}$ matrix, $\mathbf{y}_l$ and $\mathbf{x}_{(h,l)}$ have to be estimated. The $\mathbf{U}$ matrix is estimated using all the residuals of several sessions per language; the $\mathbf{y}_l$ component is estimated on all sessions belonging to language l; the $\mathbf{x}_{(h,l)}$ component is estimated on session h belonging to language l, as described in Algorithm 1. As for VGC, the $\mathbf{U}$ matrix and the target language models are estimated jointly, using the same training corpus. This is not the case for the speaker verification task (where the target classes are not known *a priori*).

The scoring is done by estimating the likelihood of the test data given a target language model. As defined in Section 2, in this paper LR is a detection task. The session compensation is performed as described in Section 4.2. For the SVM classifier, we use the kernel described in Section 4.2.

In the system that relies on the SVM classifier, the mean super-vectors may also be obtained using JFA. This system is called SVM–UBM–JFA. For a given language, the negative labeled examples are recordings belonging to the other languages. The training and testing stages are performed as explained in Section 4.3.

6.1.2. Scoring and evaluation

The scores are normalized separately for each test utterance among all languages. This is done by dividing each score (usually, the likelihood of the test utterance being of a given language) by the maximum of the scores the utterance obtained against all language models. Expressed in log-likelihood domain, this uniformly shifts all scores in order to assign a log-likelihood of 0 to the hypothesized language yielding the biggest score:

$$score_l(\mathcal{X}) = \log\left(\frac{e^{LLk_l(\mathcal{X})}}{\max\limits_{i \in L} e^{LLk_i(\mathcal{X})}}\right) = LLk_l(\mathcal{X}) - \max_{i \in L} LLk_i(\mathcal{X}) \tag{20}$$

where $LLk_l(\mathcal{X})$ is the log-likelihood of the utterance $\mathcal{X}$ and the hypothesized language *l*. *L* is the set of all languages. Since the scores produced by the SVM systems are log-likelihood like, the same procedure may be applied, although the effect is far less crucial.

System performance is measured using *minimal average cost* ($minC_{avg}$). It is the detection system choosing the decision threshold in such a way that the average expected cost of misses and false acceptances among all target/non-target language pairs is minimal (see Section 4.1 of NIST, 2009).

The cost function that will be minimized is:

$$C_{avg} = \frac{1}{N_L}\sum_{l \in L}\left[0.5 \cdot P_{Miss}(l) + \frac{0.5}{N_L - 1}\sum_{k \neq l \in L} P_{FA}(l, k)\right] \tag{21}$$

where N_L is the number of languages in our set, P_{Miss} is the probability that a language model misses a match and $P_{FA}(l, k)$ is the probability that an utterance of language *l* is mistakenly recognized as being of language *k*.

6.2. Experimental protocol for LR

The experiments are run in the context of two NIST Language Recognition Evaluation (LRE) tasks: LRE 2005 with 7-languages and LRE 2009 with 23 languages.

In our experiments, we use SDC (shifted delta cepstra) parameters in the configuration 7-1-3-7 (akin to what is used in other research efforts in this domain (Campbell et al., 2004; Matějka et al., 2006)). This means that we have 6 cepstral MEL-scale coefficients along with the energy (the cepstra and energy values are kept in the parametric vector) and seven delta blocks stacked, each block calculated on frames $t-1$ and $t+1$ with a t shifted by 3 between each block. This yields feature vectors of size 56.

Speech detection is conducted on all utterances in order to spot speech and non-speech parts. This speech/non-speech classification is based on the energy. A slight smoothing of the speech segmentation is then performed for cleaning up far too short segments. All the features are then normalized in such a way that the features containing speech of one utterance have an average of 0 and a variance of 1.

6.3. Experimental results on NIST LRE 2005

NIST's Language Recognition Evaluation 2005 (NIST, 2005) is a recognition task on 7-languages. Tests were run on this evaluation data in order to be comparable with other systems using the same protocol. The seven languages in LRE 2005 are: English, Hindi, Japanese, Korean, Mandarin, Spanish and Tamil.

While system development has been done on GMMs with 256 Gaussians, results are also presented featuring full systems using 2048 Gaussians. All the results are for 30-second segments, according to the NIST LRE 2005's closed-set primary condition.

6.3.1. Data sets

Training data: For training, all three sets (*train*, *devtest* and *evltest*) of the CallFriend[8] corpus are used. Each of these three subsets of the corpus contains 20 complete two-ended, half-hour conversations per language. The corpora of the named languages are used, including both available dialects for English, Mandarin and Spanish (thus having 40 conversations). 42.2% of the data being detected as speech, we have about 20 (resp. 40) hours of speech for each language.

Testing data: As announced, tests are conducted on the NIST LRE 2005 data. This evaluation set comprises 10 986 utterances, each containing 3, 10 or 30 s of speech. The primary condition of NIST LRE 2005 aggregates just utterances of the seven languages (closed-set condition) with a total of 10 734 utterances. We focus mainly on the 30 s ones; this comes down to 3578 files, giving as many target trials and thus 21 468 non-target trials.

6.3.2. GMM–UBM results

The GMM–UBM language models are obtained from the UBM with 10 iterations of MAP adaptation, where only the mean values are updated (neither Gaussians' weights, nor variances are updated). While seeing the GMM–UBM system as baseline, it obtains $22.40\% minC_{avg}$ with 256 Gaussians and $19.44\% minC_{avg}$ with 2048, which represents about 13% relative gain.

For the JFA system, the **U** matrix is set to have a *rank* of 40 (which is also the number of session factors). The matrix is iteratively estimated during 20 iterations using Algorithm 1. This JFA system performs at $8.57\% minC_{avg}$ using mixtures of 256 Gaussians. As it is shown in Table 7, the $minC_{avg}$ jumps to 5.41% with 2048 Gaussians, which is a far bigger improvement (37% relative) than observed for GMM–UBM systems without JFA. For 2048 Gaussians, the JFA system outperforms the GMM–UBM one by 72% relative. While the capacity of GMM systems seems slowly to exhaust with about 512 Gaussians, JFA systems reveal their power on increased model size. Observing this big performance impact of JFA over the baseline GMM–UBM system validates the profit of JFA for language recognition.

[8] CallFriend corpus, telephone speech of 15 different languages or dialects. In: http://www.ldc.upenn.edu/Catalog.

Table 7
Language Recognition performance given by GMM–UBM and GMM–UBM–JFA systems according to different model sizes (from 256 up to 2048 Gaussians; NIST LRE 2005 task, closed-set 30 seconds; ratings in $\%minC_{avg}$).

System	256	512	1024	2048
GMM–UBM	22.40%	21.25%	20.07%	19.44%
JFA	8.57%	8.38%	6.99%	**5.41%**

Table 8
Language recognition results for SVM systems without and with JFA compared to the generative GMM systems. NIST LRE-2005, in $\%minC_{avg}$.

System	256 G	2048 G
GMM–UBM	22.40%	19.44%
GMM–UBM–JFA	**8.57%**	**5.41%**
SVM–UBM	**11.55%**	—
SVM–UBM–JFA	8.91%	7.21%

Table 9
GMM–UBM and GMM–UBM–JFA systems with 2048 Gaussians on the NIST LRE 2005 and 2009 tasks; ratings in $\%minC_{avg}$.

NIST-LRE	2005	2009
GMM–UBM	19.44%	17.05%
JFA	5.41%	4.74%

6.3.3. SVM systems

In this section, we present SVM systems where every training utterance in the target language is represented by an associated positive SV. The blacklist (negative SVs for SVM training) is composed of one SV for each training utterance (of about 12 min of speech) in the non-target languages and in addition all SVs of non-target language utterances of NIST LRE 2003 evaluation data (containing utterances of 3, 10 and 30 seconds). The blacklists thus count between 2640 and 3240 SVs. In Table 8, we show the results of the 256 Gaussian SVM systems using mean SVs obtained without and with JFA. The test featuring JFA super-vectors has also been conducted on 2048 Gaussians.

The results primarily indicate that in language recognition, the SVM system without JFA largely outperforms the GMM–UBM system, but the JFA-based systems are at about the same level for a model size of 256 Gaussians and do not improve that well for 2048 Gaussians. This could be due to the fact that the tuning of SVMs is more delicate (more parameters that are quite sensible (Chang and Lin, 2001)).

6.4. Validation on NIST LRE 2009

In order to analyze if the improvement of the JFA strategy is at the same level on a different task, we ran systems with the same setup as the 2048 Gaussian systems on the NIST LRE-2009 task. This task comprises 23 languages and data from two quite different channel conditions—namely Conversational Telephone Speech (CTS) and phone bandwidth segments of radio broadcasts (Voice Of America, VOA). For further details of this task, we refer to the evaluation plan (NIST, 2009) and the evaluation overview (Greenberg and Martin, 2009). The results are also for 30-s segments, closed-set primary condition.

The results for the MAP adapted GMM–UBM and for the JFA system are presented in Table 9 along with the corresponding results on LRE 2005. For the 2009 task with the JFA system performing at $4.74\%minC_{avg}$, we also observe a gain of about 72% relative over the GMM–UBM system.

7. Conclusion

In this paper, we have explored the use of JFA for modeling the nuisance variability in three audio pattern classification tasks: speaker verification, language recognition and video genre classification. The nature of the nuisance variability between these three domains is very different. For example, the speaker identity is part of the nuisance variability in LR and VGC but constitutes the useful information for SR. In the VGC, we consider a small number of classes, but with very high intra-class variability compared to SR and LR. Fortunately, the hypothesis of this variability being located in a low-dimensional sub-space seems to be satisfied.

In Section 4, we presented the JFA decomposition algorithm in the context of SR. We proposed a straightforward implementation without referring to other papers or complex mathematics. Moreover, it is implemented in the SpkDet toolkit and the ALIZE library, which are part of the MISTRAL project that is freely available to the community. The use of an SVM classifier with an associated SV-based kernel is straightforward. In most configurations, the relative gain obtained by using JFA ranges from 48% to 64% relative (in DCFmin and EER).

In Section 5, we investigated the use of the JFA method in video genre categorization by acoustic space modeling. We compared various classification schemes and JFA configurations. Experiments on a seven-genre identification task demonstrated the efficiency of the proposed approach: The classification error rate is reduced by about 55% with respect to the standard approach based on GMM–UBM. We finally obtained a classification rate around the 90% mark, corresponding to what is classically obtained by genre-identification methods that combine audio and video information and thus outperforming classical audio-only based techniques.

In Section 6, we investigated the use of the JFA method in language recognition. We compared several approaches: standard GMM–UBM, GMM–UBM combined with JFA and SVM setups. Experiments on NIST LRE 2005 and LRE 2009 demonstrated the efficiency of the proposed methods: The basic GMM–UBM with 2048 Gaussians yields a $minC_{avg}$ of 19% and 17% respectively. With UBM-based JFA, the performance is at 5.4% for 2005 and $4.7\% minC_{avg}$ for 2009, which is a gain of 72% relative. The SVM JFA systems of the 2005 task do not show any improvement over generative GMMs, but the JFA approach proofs its usefulness also in this context.

The presented works confirm that not only the speaker verification issue can benefit from the JFA approach in order to cope with the change of acoustic conditions, but that it is also well suited for other APC tasks like video genre classification and language recognition. The strategy to decompose the modeling into a global, a class-dependent and a session-dependent part, what JFA does, seems to be a general approach to variability reduction that can successfully be applied to various pattern recognition tasks. Reusing the same paradigm for session modeling when the targeted task is varying allows to save a lot of development efforts, as the experience gathered in session modeling for speaker recognition was successfully reused for two other APC tasks, language recognition and video genre classification, even if the kind of nuisance variability is largely different between these applications.

The JFA modeling presented in this work relies on a quite simple paradigm (GMMs using a UBM and MAP adaptation). As a perspective, it seems interesting to combine this JFA approach with more complex approaches like Hidden Markov Models (HMM) or even neural networks. As in GMM-based tasks studied in this work, the HMM-based applications such as speech recognition suffer from the presence of different kind of acoustic variabilities: speaker variability, microphone or telephone variability, etc. Moreover, analyzing the nuisance variabilities in HMM context is far more complex as the variability factors are tied with the HMM states. The JFA approach relies on the training corpora where a large set of examples of nuisance variability should be present. In the HMM context, to model the channel variability tied to each of the HMM states, this constraint should be satisfied at the state level. Collecting such training data is very difficult and expensive.

In this work we presented three audio classification problems in which the explicit modeling of the nuisance variability within generative models has successfully been applied. This approach can be deployed to other classification problems using the same paradigm and, potentially, the same software. For example to the face recognition problem as Prince and Elder (2007) who use Probabilistic Linear Discriminant Analysis for decomposition.

Acknowledgement

This work was partially funded by ANR, RPM2 project (ANR-008 03).

References

Atal, B.S., 1976. Automatic recognition of speakers from their voices. Proceedings of the IEEE. 64 (4), 460–475.

Bimbot, F., Bonastre, J.-F., Fredouille, C., Gravier, G., Magrin-Chagnolleau, I., Meignier, S., Merlin, T., Ortega-García, J., Petrovska-Delacrétaz, D., Reynolds, D.A., 2004. A tutorial on text-independent speaker verification. EURASIP Journal on Applied Signal Processing. Special issue on biometric signal processing 4, 430–451.

Bonastre, J.-F., Wils, F., Meignier, S., 2005. ALIZE, a free toolkit for speaker recognition. In: Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '05), vol. 1 , Philadelphia, PA, USA, pp. 737–740.

Brezeale, D., Cook, D.J., 2008. Automatic video classification: a survey of the literature. Systems, Man, and Cybernetics. 38 (3), 416–430.

Brümmer, N., Strasheim, A., Hubeika, V., Matějka, P., Burget, L., Glembek, O., 2009. Discriminative acoustic language recognition via channel-compensated GMM statistics. In: Proceedings of Interspeech Conference. ISCA , pp. 2187–2190.

Burget, L., Matějka, P., Hubeika, V., Černocký, J.H., 2009. Investigation into variants of Joint Factor Analysis for speaker recognition. In: Proceedings of Interspeech Conference (Interspeech) 2009 , Brighton, UK. No. 9, pp. 1263–1266.

Campbell, W., Sturim, D., Reynolds, D., Solomonoff, A., 2006a. SVM based speaker verification using a GMM supervector kernel and NAP variability compensation. In: Proceedings of International Conference on Acoustics Speech and Signal Processing (ICASSP 2006), vol. 1 , May, pp. 97–100.

Campbell, W.M., Singer, E., Torres-Carrasquillo, P.A., Reynolds, D.A., 2004. Language recognition with support vector machines. In: Proceedings of Odyssey: The Speaker and Language Recognition Workshop , Toledo, Spain, ISCA, June, pp. 285–288.

Campbell, W.M., Sturim, D., Reynolds, D.A., 2006b. Support vector machines using GMM supervectors for speaker verification. IEEE Signal Processing Letters 13 ((5) May), 308–311.

Castaldo, F., Colibro, D., Dalmasso, E., Laface, P., Vair, C., 2007. Compensation of nuisance factors for speaker and language recognition. IEEE Transactions on Audio, Speech, and Language Processing 15 (7 September), 1969–1978.

Chang, C.-C., Lin, C.-J., 2001. LIBSVM: a library for support vector machines. Software available at http://www.csie.ntu.edu.tw/∼cjlin/libsvm.

Doddington, G.R., 1985. Speaker recognition. identifying people by their voices. IEEE Transactions. 73 (11), 1651–1664.

Dumouchel, P., Dehak, N., Attabi, Y., Dehak, R., Boufaden, N., 2009. Cepstral and long-term features for emotion recognition. In: Proceedings of Interspeech Conference (Interspeech) 2009 , Brighton, UK, pp. 344–347.

Furui, S., 1994. An overview of speaker recognition technology. In: Workshop on Automatic Speaker Recognition , Identification, Verification, April, pp. 1–9.

Gauvain, J.-L., Lee, C.-H., 1994. Maximum a posteriori estimation for multivariate gaussian mixture observations of Markov chains. IEEE Transactions on Speech and Audio Processing. 2, 291–298.

Glembek, O., Burget, L., Dehak, N., Brümmer, N., Kenny, P., 2009. Comparison of scoring methods used in speaker recognition with joint factor analysis. In: ICASSP '09: Proceedings of the 2009 IEEE International Conference on Acoustics , Speech and Signal Processing. IEEE Computer Society, Washington, DC, USA, April, pp. 4057–4060.

Greenberg, C., Martin, A., 2009. 2009 NIST Language Recognition Evaluation—Evaluation Overview. Slides presented at NIST LRE 2009 Workshop, June 24–25, 2009. Baltimore, USA http://www.itl.nist.gov/iad/mig/tests/lre/2009/lre09_eval_results/NIST_LRE09_workshop-presentation_website.pdf.

Hubeika, V., Burget, L., Matějka, P., Schwarz, P., 2008. Discriminative training and channel compensation for acoustic language recognition. In: Proceedings of Interspeech 2008. No. 9. International Speech Communication Association , pp. 301–304.

Jasinschi, R.S., Louie, J., 2001. Automatic TV program genre classification based on audio patterns. In: Euromicro Conference, 2001. Proceedings. 27th. IEEE Computer Society , pp. 370–375.

Kenny, P., Boulianne, G., Ouellet, P., Dumouchel, P., 2005. Factor analysis simplified. In: Proceedings of International Conference on Acoustics Speech and Signal Processing (ICASSP 2005), vol. 1 , March 18–23, pp. 637–640.

Kenny, P., Dumouchel, P., 2004. Experiments in speaker verification using factor analysis likelihood ratios. In: Proceedings of Odyssey: The Speaker and Language Recognition Workshop , ISCA, June, pp. 219–226.

Kenny, P., Ouellet, P., Dehak, N., Gupta, V., Dumouchel, P., 2008. A study of interspeaker variability in speaker verification. IEEE Transactions on Speech and Audio Processing 16 (5), 980–988.

Kockmann, M., Burget, L., Černocký, J., 2009. Brno University of technology system for interspeech, emotion challenge. In: Proceedings of Interspeech Conference (Interspeech) 2009 , Brighton, UK. No. 9, pp. 348–351.

Matrouf, D., Scheffer, N., Fauve, B., Bonastre, J.-F., 2007. A straightforward and efficient implementation of the factor analysis model for speaker verification. In: Proceedings of Interspeech Conference , Antewerp, Belgium, pp. 1242–1245.

Matějka, P., Burget, L., Schwarz, P., Černocký, J., 2005. Brno University of technology system for NIST, language recognition evaluation. In: Proceedings of Odyssey 2006: The Speaker and Language Recognition Workshop , pp. 57–64.

Moncrieff, S., Venkatesh, S., Dorai, C., 2003. Horror film genre typing and scene labeling via audio analysis. In: ICME '03: Proceedings of the 2003 International Conference on Multimedia and Expo , IEEE Computer Society, Washington, DC, USA, pp. 193–196.

NIST, 2005. The 2005 NIST Language Recognition Evaluation, Evaluation Plan. http://www.itl.nist.gov/iad/mig/tests/lre/2005.

NIST, 2008. The NIST Year 2008 Speaker Recognition Evaluation Plan. http://www.itl.nist.gov/iad/mig/tests/sre/2008/.

NIST, 2009. The 2009 NIST Language Recognition Evaluation (LRE09), Evaluation Plan. http://www.itl.nist.gov/iad/mig/tests/lre/2009.

Povey, D., et al., 2010. Subspace Gaussian mixture models for speech recognition. In: Proceedings of International Conference on Acoustics Speech and Signal Processing (ICASSP 2010) , March 14–18, pp. 4330–4333.

Prince, S.J.D., Elder, J.H., 2007. Probabilistic linear discriminant analysis for inferences about identity. In: Proceedings of IEEE International Conference on Computer Vision (ICCV 2008) , October, pp. 1–8.

Reynolds, D.A., Quatieri, T.F., Dunn, R.B., 2000. Speaker verification using adapted Gaussian mixture models. Digital Signal Processing 10 (1 January), 19–41.

Roach, M., Mason, J., 2001. Classification of video genre using audio. In: European Conference on Speech Communication and Technology, vol. 4 , pp. 2693–2696.

Roach, M., Xu, L.-Q., Mason, J., Heath, M., Re, I.I., 2002. Classification of non-edited broadcast video using holistic low-level features. In: Proceedings of Tyrrhenian Workshop on Digital Communications (IWDC'2002).

Smeaton, A.F., Over, P., Kraaij, W., 2006. Evaluation campaigns and TRECVid. In: MIR '06: Proceedings of the 8th ACM International Workshop on Multimedia Information Retrieval. ACM , New York, NY, USA, pp. 321–330.

Torres-Carrasquillo, P.A., Singer, E., Kohler, M.A., Greene, R.J., Reynolds, D.A., Deller, J.R.J., 2002. Approaches to language identification using gaussian mixture models and shifted delta cepstral features. In: Proceedings of International Conference on Spoken Language Processing (ICSLP 2002) , pp. 89–92.

Verdet, F., Matrouf, D., Bonastre, J.-F., Hennebert, J., 2009. Factor Analysis and SVM for language recognition. In: Proceedings of International Conference on Speech Communication and Technology (Interspeech) 2009 , ISCA, pp. 164–167.

Vogt, R., Baker, B., Sridharan, S., 2005. Modelling session variability in text-independent speaker verification. In: Proceedings of Interspeech 2005, 9th European Conference on Speech Communication and Technology , September 4–8, 2005, Lisboa, Portugal, pp. 3117–3120.

Vogt, R., Baker, B., Sridharan, S., 2008. Explicit modeling of session variability for speaker verification. Computer Speech & Language 22 (1), 17–38.

Wang, H.-L., Divakaran, A., Vetro, A., Chang, S.-F., Sun, H.-F., 2003. Survey of compressed-domain features used in audio-visual indexing and analysis. Journal of Visual Communication and Image Representation. 14 (2 June), 150–183.

Weiyu Zhu, C.T., Liou, S.-P., 2001. Automatic news video segmentation and categorization based on closed-captioned text. In: Multimedia and Expo, ICME. Vol. 0. IEEE Computer Society , Los Alamitos, CA, USA, pp. 829–832.

Xu, L.-Q., Li, Y., 2003. Video classification using spatial-temporal features and PCA. In: ICME '03: Proceedings of the 2003 International Conference on Multimedia and Expo - Volume 3 (ICME '03) , IEEE Computer Society, Washington, DC, USA, pp. 485–488.

Zissman, M.A., 1996. Comparison of four approaches to automatic language identification of telephone speech. IEEE Transactions. 4 (1), 31–44.