

I-VECTORS FOR TIMBRE-BASED MUSIC SIMILARITY AND MUSIC ARTIST CLASSIFICATION

Hamid Eghbal-zadeh Bernhard Lehner Markus Schedl Gerhard Widmer

Department of Computational Perception, Johannes Kepler University of Linz, Austria

hamid.eghbal-zadeh@jku.at

ABSTRACT

In this paper, we present a novel approach to extract song-level descriptors built from frame-level timbral features such as Mel-frequency cepstral coefficient (MFCC). These descriptors are called identity vectors or *i-vectors* and are the results of a factor analysis procedure applied on frame-level features. The *i-vectors* provide a low-dimensional and fixed-length representation for each song and can be used in a supervised and unsupervised manner.

First, we use the *i-vectors* for an unsupervised music similarity estimation, where we calculate the distance between *i-vectors* in order to predict the genre of songs.

Second, for a supervised artist classification task we report the performance measures using multiple classifiers trained on the *i-vectors*.

Standard datasets for each task are used to evaluate our method and the results are compared with the state of the art. By only using timbral information, we already achieved the state of the art performance in music similarity (which uses extra information such as rhythm). In artist classification using timbre descriptors, our method outperformed the state of the art.

1. INTRODUCTION AND RELATED WORK

In content-based music similarity and classification, acoustic features are extracted from audio and characteristics of a song are projected into a new space called feature space. In this space, different attributes can be captured based on the features used. For example, features such as Fluctuation Pattern (FP) [26], reflect the variability related to the rhythm; and features such as MFCCs, demonstrate the timbral perspective of a song. However, the diversity of music genres, the presence of different musical instruments and singing techniques make the capturing of these variabilities difficult. Different modeling techniques and machine learning approaches are used to find the factors in the feature space that best represent these variabilities.

Multiple approaches have been followed in the literature for extracting the features from songs in which 1) clas-

sical frame-level features, 2) block-level features and 3) song-level features are the most frequently used methods in MIR.

1.1 Frame-level features

In the frame-level approach, features are often extracted from short-time frames of a song. In this approach, frames are first classified directly, and then the results are combined to make a decision for a song.


1.2 Block-level features

Block-level features process the frames in terms of blocks, where each block consists of a fixed number of frames. They are built in two steps: first, the block processing step and second, the generalization step. In the first step, by selecting a collection of frames using a pattern, blocks are built. Then in the second step, the feature values of all blocks are combined into a single representation for the whole song. In [29], six different block-level features are introduced and a method is proposed to fuse all the blocks together. Block-level features [5, 24, 26, 29] have shown considerable performances in the MIREX¹ challenges.

1.3 Song-level features

Song-level features are found useful in artist recognition as well as music similarity estimation. In [30], a compact signature is generated for each song, and then is compared to the other songs using a graph matching approach for artist recognition. In [21] multivariate kernels have been used to model an artist. Recently, [5, 29] proposed methods to extract a fixed-length vector from a song to be used in music similarity estimation and genre classification.

The advantage of methods based on song-level features is that different tools such as dimensionality reduction (e.g. Principal Components Analysis (PCA) [15]) and projections can be applied to songs. For example, in [5], super-vectors extracted via a Gaussian Mixture Model (GMM) are found useful to represent songs and calculate the similarity using Euclidean distance. In [24] a method using song-level features is presented, which models frame-level descriptors such as MFCCs and FP with a single Gaussian and then the similarity between songs is calculated using Kullback Leibler divergence. In [26], rhythm descriptors

 © Hamid Eghbal-zadeh, Bernhard Lehner, Markus Schedl, Gerhard Widmer. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Hamid Eghbal-zadeh, Bernhard Lehner, Markus Schedl, Gerhard Widmer. “I-Vectors for Timbre-Based Music Similarity and Music Artist Classification”, 16th International Society for Music Information Retrieval Conference, 2015.

¹ Annual Music Information Retrieval eXchange (MIREX). More information is available at: <http://www.music-ir.org>

are introduced to improve the performance of music similarity measures in [24].

1.3.1 GMM and GMM-supervectors

GMMs have been frequently used for acoustic modeling in music processing [4, 5, 12]. In [4, 5], a GMM is used as a *Universal Background Model* (UBM) for content-based music similarity estimation and genre classification.

Gaussian-based features used in [5, 24] are other examples of song-level features which use a Gaussian model to create a statistical representation of a song from frame-level features. Similar to [4, 5], a GMM supervector is computed for each song. This representation is a fixed-length vector, and is computed using a UBM (which is a GMM, trained on a database of songs) via a procedure described in [4, 5].

The first drawback of GMM-based methods is that when the rank of the GMM space (number of Gaussian components) increases, the dimensionality of GMM supervectors rises which causes problems such as the curse of dimensionality. One solution to this issue would be to use dimensionality reduction methods such as PCA. In our previous work [9], we showed that this is not effective. Another solution would be to decompose these high-dimensional supervectors into multiple terms with lower ranks which we will discuss in the following section.

1.3.2 Session and Speaker variability

As described in [18], there exists a second drawback of GMM-based methods. The performance of these frameworks suffer from their inability to capture the variability known as *session variability* in the field of speaker verification. In contrast to *speaker variability* which is the variability that appears between different speakers, session variability is defined as the variability that appears for a speaker from one recording to another [18]. This variability is called *session* because it appears inside a recording session of a speaker.

1.3.3 Song, Genre and Artist variability

In MIR, similar to session variability, we define *song variability* as the variability that appears between songs. Also, similar to speaker variability, we define *genre variability* for genre classification as the variability that appears between different genres, and *artist variability* for artist recognition as the variability appears between different artists.

The second drawback of GMM-based methods is that they can not distinguish between song variability and genre (or artist) variability. If we can provide a decomposition of GMM supervectors in a way that separates the desired factors, such as genre variability from undesired ones, such as song variability, and at the same time decreases the dimensionality of GMM supervectors, then as a result a better representation of GMM supervectors with lower dimensionality and better discrimination power will be obtained. Factor Analysis (FA) provides the means to produce such representations where a GMM supervector de-

composes into multiple factors. An advantage of the features obtained by FA compared to block-level features and Gaussian-based features is that FA can be performed in a way that after decomposition, each component can exhibit a specific variability such as artist or genre. Thus, desired factors can be kept and undesired factors can be removed from the song's GMM supervector. By applying such decomposition on top of the GMM space, another space with bases of desired factors (e.g. genre space, with genre factors) can be created.

Recently, in the field of speaker verification, Dehak et al. [7] introduced **i-vectors** which outperformed the state of the art and provided a solution for the problem of session variability in the GMM-UBM frameworks. The i-vector extraction is a feature-modeling technique that builds utterance-level features, and it has been successfully used in other areas such as emotion recognition [34], language recognition [8], accent recognition [1] and audio scene detection [10].

The i-vector method applies a FA procedure to extract low-dimensional features from GMM supervectors. This FA procedure estimates hidden variables in GMM supervector space, which provides better discrimination ability and lower dimensionality than GMM supervectors. These hidden variables are the i-vectors and even though **the i-vector extraction procedure is totally unsupervised**, they can be used for both supervised and unsupervised tasks. The aim of this paper is to introduce the i-vectors to the MIR community and show their performance on two of the major tasks in content-based MIR.

2. FACTOR ANALYSIS PROCEDURE

In this paper, examples are given from a **genre classification** point of view. The definitions and the method are extendable to other tasks in MIR such as artist classification.

2.1 Overview of Factor Analysis Methods

A FA model can be viewed as a GMM supervector space, where genre and song factors are its hidden variables. Genre and song factors are defined in a way that for a given genre, the values of the genre factors are assumed to be identical for all songs within that genre. The song factors may vary from one song to another.

Let's assume we have a C mixture components GMM and let F be the dimension of the acoustic feature vectors. For each mixture component $c = 1, \dots, C$, let m_c denote the corresponding genre-independent mean vector (UBM mean vector) and let m denote the $C \cdot F \times 1$ supervector obtained by concatenating m_1, \dots, m_C .

Maximum a posteriori (MAP) [14] is a method that is used to extract genre-dependent GMM supervectors. In MAP, it is assumed that each genre g can be modeled only by a single genre-dependent GMM supervector $M(g)$. This supervector is calculated from a genre-independent vector m which is then adapted to a couple of songs from a specific genre known as the genre-adaptation data.

Similar to speaker modeling in speaker verification [19], the MAP approach to genre modeling assumes that for each mixture component c and genre g , there is an unobservable offset vector O_g such that:

$$M(g) = m + O_g \quad (1)$$

O_g is unknown and can not be learned during the MAP training procedure.

Further, *eigenvoice MAP* [17] assumes the row vectors of the matrix O_g are independent and identically distributed. A rectangular matrix V of dimensions $C \cdot F \times R$ is assumed where R is a parameter such that $R \ll C \cdot F$. The V matrix has a lower rank than $C \cdot F$ and can be learned from the training data. The supervector $M(g)$ decomposes into factors $y(g)$ which have lower ranks using V . For genre g , the FA used in eigenvoice MAP is as follows:

$$M(g) = m + Vy(g) \quad (2)$$

where $y(g)$ is a hidden $R \times 1$ vector which has a standard normal distribution. Eigenvoice MAP trains faster than MAP, yet training V properly needs a very large amount of data, also song factors are not considered in the decomposition of $M(g)$.

A solution for separation between song and genre factors was first suggested in [19], and later improved in [16] as Joint Factor Analysis (JFA). JFA decomposition model can be written as follows:

$$M = m + Vy + Ux + Dz \quad (3)$$

where M is a song GMM supervector, m is a genre- and song-independent supervector which can be calculated using a UBM, V and D define a genre subspace (genre matrix and diagonal residual, respectively), and U defines a song subspace. The vectors y , z are the genre-dependent factors, and x is the song-dependent factor in their respective subspaces. They are assumed to be a random variable with a standard normal distribution. Unlike eigenvoice MAP, JFA gives us a modeling with separated genre and song factors with low ranks, where they can be used to better separate songs from different genres by removing song variability.

Even though JFA showed better performance than previous FA methods, in terms of separation between song and genre factors, experimental results in [6] proved that if we extract song and genre factors using JFA, song factors also contain information about genres. Based on this finding, another FA model is proposed in [7], which defines a new low-dimensional space called Total Variability Space (TVS). The vectors in this new space, are called i-vectors. In the TVS, both song and genre factors are considered, but modeled together as a new factor named *total factor*. Total factors have lower dimensionality than GMM supervectors and one can represent a song by extracting total factors from its GMM supervector. Because i-vector FA showed the best results in speaker verification [7], in this paper we use it for multiple tasks in MIR. The FA procedure used to obtain i-vectors is described in the next section.

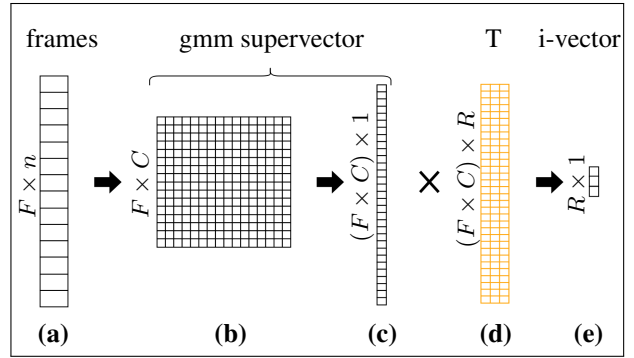


Figure 1: Graphical representation of different vectors extracted during i-vector FA. F is the dimensionality of acoustic features, C is the number of Gaussian components, and R is the rank of the TVS matrix. a) frame-level features of a song. b) and c) GMM supervector. d) TVS matrix T . e) i-vector.

2.2 Overview of I-vectors

TVS refers to total factors that contain both genre and song factors. In the TVS, a given song is represented by a low-dimensional vector called **i-vector**, which provides a good genre separability. This i-vector is known as point estimate of the hidden variables in a FA model similar to JFA. This describes these hidden variables and their characteristics.

In Figure 1, a graphical representation of vectors used in different steps during i-vector FA is provided. From each song, first frame-level features of dimensionality F are extracted as shown in Figure 1-a. Then, a C mixture components GMM trained on a large number of songs is used to extract GMM supervectors of dimension $F \times C$. This rectangular vector (Figure 1-b) then reshapes to a $(F \cdot C) \times 1$ vector (Figure 1-c). A matrix of $(C \cdot F) * R$ known as TVS matrix (T) is learned from a set of songs. T matrix is used to reduce the dimensionality of GMM supervectors to R where R is the rank of T , as can be observed in Figure 1-d. The resulting vectors are i-vectors having a low rank of R (Figure 1-e).

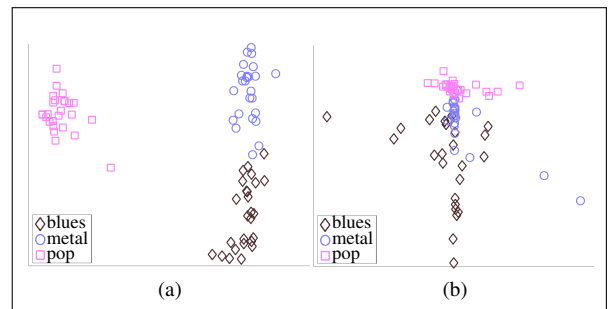


Figure 2: 2D PCA projected vectors extracted from songs of 3 different genres in GTZAN dataset. a) i-vectors. b) GMM supervectors.

A comparison between GMM representation and i-vector representation is provided in Figure 2. This visualization is prepared by projecting GMM supervectors and i-vectors using PCA into a 2 dimensional plane. Multiple

songs of 3 different genres from the GTZAN dataset² are selected, then both their GMM supervectors and i-vectors are extracted. In Figure 2-a, a scatter plot of the song’s projected i-vectors are shown. Also, in Figure 2-b, GMM supervectors projected using PCA are displayed. It can be observed that i-vector extraction was successful at increase the discrimination between songs of different genres. In the following paragraphs, the i-vector FA is described.

A C mixture components GMM ($c = 1, \dots, C$) called UBM can be trained on a large amount of data from multiple genres, where for component c , w_c , m_c and Σ_c denote mixture weight, mean vector and covariance matrix respectively. Given a song of genre g , a GMM supervector $M(g)$ can be calculated from a sequence of X_1, \dots, X_τ frames. The i-vector FA equation decomposes the vector $M(g)$ as follows:

$$M_c(g) = m_c + Ty \quad (4)$$

where $M_c(g)$ corresponds to a subvector of $M(g)$ for component c , m_c is the genre- and song-independent vector, and $y \sim \mathcal{N}(0, 1)$ is the genre- and song-dependent vector, known as the i-vector. A rectangular matrix T of low rank known as TVS matrix is used to extract i-vectors from the vector $M_c(g)$.

The i-vector y is a hidden variable, but we can find it using the mean of its posterior distribution. This posterior distribution is Gaussian and is conditioned to the BaumWelch (BW) statistics for a given song [17]. The zero-order and the first-order BW statistics used to estimate y , are called N_c and P_c respectively (see Equation 6). Similar to [20], the BW statistics are extracted using the UBM as follows.

A closed form of an i-vector y looks as follows:

$$y = (I + T^t \Sigma^{-1} N(s) T)^{-1} \cdot T^t \Sigma^{-1} P(s) \quad (5)$$

where we define $N(s)$ as a diagonal matrix of dimension $C \cdot F \times C \cdot F$ with $N_c \times I$ ($c = 1, \dots, C$ and I has $F \times F$ dimensions) diagonal blocks. $P(s)$ is a vector with $C \cdot F \times 1$ dimensions and is made by concatenating all first-order BW statistics P_c for a given song s ; also Σ is a diagonal covariance matrix of dimension $C \cdot F \times C \cdot F$ estimated during the factor analysis procedure; it models the residual variability not captured by the TVS matrix T . The BW statistics N_c and P_c are defined as follows.

Suppose we have a sequence of frames X_1, \dots, X_τ and a UBM with C mixture components defined in a feature space of dimension F . The BW statistics needed to estimate the i-vector for a given song are obtained by:

$$\begin{aligned} N_c &= \sum_t \gamma_t(c) \\ P_c &= \sum_t \gamma_t(c) X_t \end{aligned} \quad (6)$$

where, for time t , $\gamma_t(c)$ is the posterior probability of X_t generated by the mixture component c of the UBM.

² <http://marsyas.info/downloads/datasets.html>

Since BW statistics are calculated using a GMM, they are called **GMM supervectors** in i-vector modeling.

TVS matrix T is estimated via a expectation maximization procedure using BW statistics. More information about the training procedure of T can be found in [7, 22].

3. I-VECTORS FOR UNSUPERVISED MUSIC SIMILARITY ESTIMATION

In this section, i-vectors are used for music similarity estimation task. Genre and song variability are the factors used in this task.

3.1 Dataset

The 1517Artists³ dataset is used for training UBM and T matrix. This dataset consists of freely available songs and contains 3180 tracks by 1517 different artists distributed over 19 genres. The GTZAN dataset is used for music similarity estimation which contains 1000 song excerpts of 30 seconds, evenly distributed over 10 genres.

3.2 Frame-level Features

We use MFCCs as one of our timbral features. MFCCs are the most utilized timbre-related frame-level features in MIR. They are a compact, and perceptually motivated representation of the spectral envelope.

For the extraction of the MFCCs, we use an observation window of 10 ms, with an overlap of 50%. We extract 25 MFCCs with the rastamat toolbox [11]. The first and second order derivatives (deltas and double-deltas) of the MFCCs are also added to the feature vector.

Additionally, we use the first order derivative of a cent-scaled spectrum, calculated in the same way as explained in [29]. These features are called Spectrum Derivatives (SD).

3.3 Baselines

Four different baselines are used to be compared to our method. The first baseline is fusing block-level similarity measure (BLS) [29], which uses 6 different block-level features containing spectral pattern, delta spectral pattern, variance delta spectral pattern, logarithmic fluctuation pattern, correlation pattern and spectral contrast pattern. These features are used with a similarity function and a distance normalization method to calculate a pairwise distance matrix between songs. The second baseline is called Rhythm Timbre Bag of Features (RTBOF) [26]. RTBOF has two components of rhythm and timbre which are modeled over local spectral features. The third baseline is MARSYAS (Music Analysis, Retrieval and Synthesis for Audio Signals) which has an open source toolbox to calculate various audio features.⁴ A similarity function is used to calculate a distance matrix of features extracted as described in [32]. The last baseline (CMB) is a combination of BLS and RTBOF, which reported in [29] as the best similarity method in case of genre classification measures.

³ This dataset can be downloaded from www.seyerlehner.info.

⁴ <http://marsyas.info>

3.4 Experimental Setup

A UBM with 1024 Gaussian components is trained on the 1517Artists dataset using 2000 consecutive frames from the middle area of each song. No labels are used during the training procedure of UBM and T matrix. The TVS matrix T is trained using 400 total factors, and used during the i-vector extraction procedure. The number of factors and Gaussian components was chosen after a parameter analysis step on a small development dataset which differs from the datasets used in this paper.

Two sets of different i-vectors are used to calculate two similarity matrices for the GTZAN dataset. First, MFCC features are used to extract i-vectors, and cosine distance is used to calculate a pair-wise distance matrix between all songs, since in [7] cosine distance has been successfully used with i-vectors. UBM and T matrix are also trained using MFCC features of 1517Artists.

Second, SD features are used to extract another set of i-vectors to calculate our second distance matrix using cosine distance. Similar to MFCC i-vectors, a new UBM and T matrix is trained using SD features extracted from 1517Artist dataset.

Pair-wise distance matrices are normalized using a distance space normalization (DSN) proposed in [25]. The distance matrices for baseline methods are downloaded from the website⁵ of the author of [29].

3.5 Evaluation

We evaluate the music similarity measures using genre classification via k -nearest neighbor (KNN) classification. This method is also used in [5, 24, 26, 29]. We use different values of k that vary from 1 to 20. Also, we use a leave-one-out scenario for genre classification using pair-wise distance matrices.

3.6 Results and Discussion

The KNN genre classification accuracy calculated using our method is compared to the baseline methods, and the results are shown in Figure 3. As can be seen, our method using MFCC features achieved the performance of the BLS baseline and outperformed MARSYAS. By combining the distance matrices calculated using MFCC and SD i-vectors with equal weights after applying DSN, we could achieve the performance of RTBOF baseline.

Since the authors of the BLS method in [29] reported a combination of BLS and RTBOF (named as CMB in [29]) to perform best, we also combined our MFCC+SD i-vector distance matrix with RTBOF with equal weights after applying DSN and achieved the performance of CMB. Furthermore, by combining MFCC+SD i-vector and CMB distance matrix (with equal weights after DSN), we could achieve a better performance than the best combined method reported in [29].

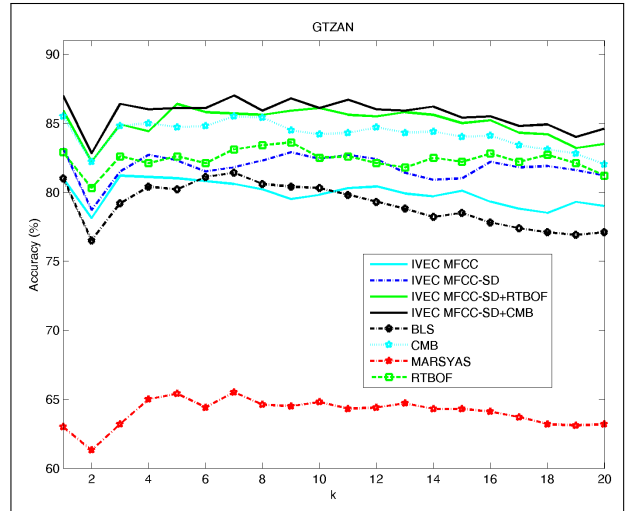


Figure 3: Evaluation results of KNN genre classification on GTZAN dataset.

3.7 Resources

The MSR Identity Toolbox [28] was used for i-vector extraction. We also used drtoolbox [33] to apply PCA for visualization in Figure 2.

4. I-VECTORS FOR SUPERVISED ARTIST CLASSIFICATION

In this section, i-vectors are used for artist recognition task. Artist and song variability are the factors used in this task. More details about artist recognition using i-vectors can be found in our previous work [9].

4.1 Dataset

The artist classification experiments were conducted using the artist20 dataset [12]. It contains 1413 tracks, mostly rock and pop songs, composed of six albums from each of the 20 different artists.

4.2 Frame-level Features

Instead of extracting the MFCCs ourselves, we use the ones provided as part of the dataset in [12]. Neither first nor second order derivatives of the MFCCs are used. Similar to the approach already discussed in Section 3.2, we also include the first order derivative of a cent-scaled spectrum (SD features).

4.3 Baselines

Multiple baseline methods from the literature are selected and their performance is compared to that achieved by our method. Results are reported for a 20-class artist classification task on the artist20 dataset [12]. The first baseline (*BLGMM*) models artists with GMMs using MFCCs [12]. The second baseline (*BLsparse*) uses a sparse feature learning method [31] of ‘bag of features’ (BOF). Both the magnitude and phase parts of the spectrum are used in this

⁵www.seyerlehner.info

method. The third baseline is (*BLsign*). It generates a compact signature for each song using MFCCs, and then compares these by a graph matching technique [30]. The fourth baseline (*BLmultiv*) uses multivariate kernels [21] with the direct uniform quantization of the MFCC features. The results for the latter three are taken from their publications, while the results for the *BLGMM* baseline are reproduced using the implementation provided with the dataset. The performance of all baselines on the artist20 dataset are reported using the same songs, and the same fold splits in the 6-fold cross-validation.

4.4 Experimental Setup

Similar to the setup followed in Section 3.4, a UBM with 1024 Gaussian components and a T matrix with 400 factors are used for i-vector extraction. Unlike the setup in music similarity estimation, no other dataset is used to train T and the UBM. Instead, in each fold the training set is used to train UBM and T matrix. Unlike the setup described in Section 3.4, we apply a Linear Discriminant Analysis (LDA) [23] to the i-vectors to reduce the dimensionality from 400 to 19. The reason we didn't use LDA for music similarity estimation is that the whole procedure of i-vector extraction in Section 3 was unsupervised, and no labels were used during the i-vector extraction process.

In each fold, the LDA is trained on the same data that UBM and T matrix are trained. I-vectors are centered by removing the mean calculated from training i-vectors, then length-normalized [13] before applying LDA. After applying LDA, once again i-vectors are length-normalized since iterative length-normalization was found useful in [2]. The length normalization provides a standard form of i-vectors.

We fuse MFCC and SD i-vectors of a song simply by concatenating the dimensionality-reduced i-vectors and subsequently feed them into the classifiers investigated.

First, a Probabilistic Linear Discriminant Analysis (PLDA) [27] is used to find the artist for each song (iv-PLDA). PLDA is a generative model which models both intra-class and inter-class variance as multidimensional Gaussian and showed significant results with i-vectors [3]. Second, a KNN classifier with $k = 3$ (3NN) and a cosine distance is considered (iv3NN). Third, a Discriminant Analysis (DA) classifier is investigated with a linear discriminant function and a uniform prior (ivDA).

4.5 Evaluation

A 6-fold cross-validation proposed in [12] is used to evaluate the artist classification task. In each fold, five albums from each artist are used for training and one for testing. We report mean class-specific accuracy, F1, precision and recall, all averaged over folds.

4.6 Results and Discussion

The results of artist classification are reported in Table 1. Using MFCC i-vectors, our proposed method outperformed all the baselines with all three classifiers. Also by

using MFCC+SD i-vectors, the results of artist classification from all 3 classifiers improved. The best artist classification performance is achieved using MFCC+SD i-vectors and a DA classifier yielding 11 percentage point improvement in accuracy and 10 percentage point improvement in F1 compared to the best known results among all the baselines.

Method	Feat.	Acc%	F1%	Pr%	Rec%
BLGMM	20mfcc	55.90	55.18	58.74	58.20
BLsparse	BOF	67.50	n/a	n/a	n/a
BLsign	15mfcc	71.50	n/a	n/a	n/a
BLmultiv	13mfcc	74.30	74.79	n/a	n/a
ivPLDA	20mfcc	83.30	82.58	83.72	84.02
iv3NN	20mfcc	82.43	81.70	83.06	83.03
ivDA	20mfcc	83.36	82.67	84.07	83.78
ivPLDA	20mfcc+sd	85.27	84.58	85.87	85.68
iv3NN	20mfcc+sd	83.68	83.05	84.10	84.55
ivDA	20mfcc+sd	85.45	84.59	85.80	85.68

Table 1: Artist classification results for **different methods** on the **artist20** dataset.

4.7 Resources

We used the same resources as reported in Section 3.7. In addition, we used the PLDA implementation from MSR Identity Toolbox [28] and LDA from drtoolbox [33].

5. CONCLUSION

In this paper, we propose an i-vector based factor analysis (FA) technique to extract song-level features for unsupervised music similarity estimation and supervised artist classification. In music similarity estimation, our method achieved the performance of state-of-the-art methods by using only timbral information. In artist classification, our method was evaluated on a variety of classifiers and proved to yield stable results. The proposed method outperformed all the baselines on the artist20 dataset and improved the best known artist classification measures among baselines. To the best of our knowledge, our results are the highest artist classification results published so far for the artist20 dataset.

6. ACKNOWLEDGMENT

We would like to acknowledge the tremendous help by Dan Ellis from Columbia University, who shared the details of his work, which enabled us to reproduce his experiment results. Thanks to Pavel Kuksa from University of Pennsylvania for sharing the details of his work with us. Also thank to Jan Schlüter from OFAI for his help with music similarity baselines. And at the end, we appreciate helpful suggestions of Rainer Kelz and Filip Korzeniowski from Johannes Kepler University of Linz to this work. This work was supported by the EU-FP7 project no.601166 (PHENICX), and by the Austrian Science Fund (FWF) under grants TRP307-N23 and Z159.

7. REFERENCES

- [1] Mohamad Hasan Bahari, Rahim Saeidi, David Van Leeuwen, et al. Accent recognition using i-vector, gaussian mean supervector and gaussian posterior probability supervector for spontaneous telephone speech. In *ICASSP*. IEEE, 2013.
- [2] Pierre-Michel Bousquet, Driss Matrouf, and Jean-François Bonastre. Intersession compensation and scoring methods in the i-vectors space for speaker recognition. In *INTER-SPEECH*, 2011.
- [3] Lukas Burget, Oldrich Plchot, Sandro Cumani, Ondrej Glembek, Pavel Matejka, and Niko Brummer. Discriminatively trained probabilistic lda for speaker verification. In *ICASSP*. IEEE, 2011.
- [4] Chuan Cao and Ming Li. Thinkits submissions for mirex2009 audio music classification and similarity tasks. In *MIREX*. Citeseer, 2009.
- [5] Christophe Charbuillet, Damien Tardieu, Geoffroy Peeters, et al. Gmm-supervector for content based music similarity. In *DAFx, Paris, France*, 2011.
- [6] Najim Dehak. *Discriminative and generative approaches for long-and short-term speaker characteristics modeling: application to speaker verification*. Ecole de Technologie Supérieure, 2009.
- [7] Najim Dehak, Patrick Kenny, Réda Dehak, Pierre Dumouchel, and Pierre Ouellet. Front-end factor analysis for speaker verification. *Audio, Speech, and Language Processing, IEEE Transactions on*, 2011.
- [8] Najim Dehak, Pedro A Torres-Carrasquillo, Douglas A Reynolds, and Reda Dehak. Language recognition via i-vectors and dimensionality reduction. In *INTERSPEECH*. Citeseer, 2011.
- [9] H Eghbal-zadeh, M Schedl, and G Widmer. Timbral modeling for music artist recognition using i-vectors. In *EUSIPCO*, 2015.
- [10] Benjamin Elizalde, Howard Lei, and Gerald Friedland. An i-vector representation of acoustic environments for audio-based video event detection on user generated content. In *ISM*. IEEE, 2013.
- [11] Daniel PW Ellis. PLP and RASTA (and MFCC, and inversion) in Matlab, 2005. online web resource.
- [12] Daniel PW Ellis. Classifying music audio with timbral and chroma features. In *ISMIR*, 2007.
- [13] Daniel Garcia-Romero and Carol Y Espy-Wilson. Analysis of i-vector length normalization in speaker recognition systems. In *INTERSPEECH*, 2011.
- [14] Jean-Luc Gauvain and Chin-Hui Lee. Maximum a posteriori estimation for multivariate gaussian mixture observations of markov chains. *Speech and audio processing, IEEE Transactions on*, 1994.
- [15] Ian Jolliffe. *Principal component analysis*. Wiley Online Library, 2002.
- [16] Patrick Kenny. Joint factor analysis of speaker and session variability: Theory and algorithms. *CRIM, Montreal, (Report CRIM-06/08-13)*, 2005.
- [17] Patrick Kenny, Gilles Boulianne, and Pierre Dumouchel. Eigenvoice modeling with sparse training data. *Speech and Audio Processing, IEEE Transactions on*, 2005.
- [18] Patrick Kenny, Gilles Boulianne, Pierre Ouellet, and Pierre Dumouchel. Speaker and session variability in gmm-based speaker verification. *Audio, Speech, and Language Processing, IEEE Transactions on*, 2007.
- [19] Patrick Kenny, Mohamed Mihoubi, and Pierre Dumouchel. New map estimators for speaker recognition. In *INTER-SPEECH*, 2003.
- [20] Patrick Kenny, Pierre Ouellet, Najim Dehak, Vishwa Gupta, and Pierre Dumouchel. A study of interspeaker variability in speaker verification. *Audio, Speech, and Language Processing, IEEE Transactions on*, 2008.
- [21] Pavel P. Kuksa. Efficient multivariate kernels for sequence classification. *CoRR*, 2014.
- [22] Driss Matrouf, Nicolas Scheffer, Benoit GB Fauve, and Jean-François Bonastre. A straightforward and efficient implementation of the factor analysis model for speaker verification. In *INTERSPEECH*, 2007.
- [23] Sebastian Mika, Gunnar Ratsch, Jason Weston, Bernhard Scholkopf, and Klaus-Robert Muller. Fisher discriminant analysis with kernels. In *Signal Processing Society Workshop Neural Networks for Signal Processing*, 1999.
- [24] Elias Pampalk. Audio-based music similarity and retrieval: Combining a spectral similarity model with information extracted from fluctuation patterns. In *ISMIR*, 2006.
- [25] Tim Pohle and Dominik Schnitzer. Striving for an improved audio similarity measure. *Music information retrieval evaluation exchange*, 2007.
- [26] Tim Pohle, Dominik Schnitzer, Markus Schedl, Peter Knees, and Gerhard Widmer. On rhythm and general music similarity. In *ISMIR*, 2009.
- [27] Simon JD Prince and James H Elder. Probabilistic linear discriminant analysis for inferences about identity. In *Computer Vision, ICCV*. IEEE, 2007.
- [28] Seyed Omid Sadjadi, Malcolm Slaney, and Larry Heck. Msr identity toolbox-a matlab toolbox for speaker recognition research. *Microsoft CSRC*, 2013.
- [29] Klaus Seyerlehner, Gerhard Widmer, and Tim Pohle. Fusing block-level features for music similarity estimation. In *DAFx*, 2010.
- [30] Sajad Shirali-Shahreza, Hassan Abolhassani, and M Shirali-Shahreza. Fast and scalable system for automatic artist identification. *Consumer Electronics, IEEE Transactions on*, 2009.
- [31] Li Su and Yi-Hsuan Yang. Sparse modeling for artist identification: Exploiting phase information and vocal separation. In *ISMIR*, 2013.
- [32] George Tzanetakis and Perry Cook. Musical genre classification of audio signals. *Speech and Audio Processing, IEEE transactions on*, 2002.
- [33] LJP Van der Maaten, EO Postma, and HJ van den Herik. Matlab toolbox for dimensionality reduction. *MICC*, 2007.
- [34] Rui Xia and Yang Liu. Using i-vector space model for emotion recognition. In *INTERSPEECH*, 2012.