TIMBRAL MODELING FOR MUSIC ARTIST RECOGNITION USING I-VECTORS

Hamid Eghbal-zadeh, Markus Schedl and Gerhard Widmer

Department of Computational Perception
Johannes Kepler University of Linz, Austria

ABSTRACT

Music artist (i.e., singer) recognition is a challenging task in Music Information Retrieval (MIR). The presence of different musical instruments, the diversity of music genres and singing techniques make the retrieval of artist-relevant information from a song difficult. Many authors tried to address this problem by using complex features or hybrid systems. In this paper, we propose new song-level timbre-related features that are built from frame-level MFCCs via so-called i-vectors. We report artist recognition results with multiple classifiers such as K-nearest neighbor, Discriminant Analysis and Naive Bayes using these new features. Our approach yields considerable improvements and outperforms existing methods. We could achieve an **84.31%** accuracy using MFCC features on a 20-classes artist recognition task.

Index Terms— music artist recognition, timbral modeling, song-level features, i-vectors, mfcc

1. INTRODUCTION AND RELATED WORK

Digital music is becoming more and more abundant and music streaming services can be easily used on smart phones, personal computers and smart TVs. As a result, technologies are required for efficient retrieval of this digital data to provide tools for browsing the musical content. The identification of music artists[1] from analysis of the music signal is one of these technologies.

As modeling the characteristics of an artist is crucial in artist recognition, features that give a good representation of an artist are very important. Different audio features have been used for modeling an artist. Mel-Frequency Cepstrum Coefficients (MFCCs) have shown great success in modeling the human voice [1] and are found useful for different music classification tasks [2, 3], yet using extra information like chroma can still improve the performance [4]. Basically, three main approaches have been followed in feature extraction. 1) frame-level, 2) segment-level and 3) song-level features.

In order to have a good timbre representation, features are often extracted from short-time frames of audio data. Methods following the first approach, first classify directly the frames themselves, and then combine these frame-based decisions into a song label by majority voting [5]. This approach was successful on small datasets or solo singers.

The second approach aggregates frame-level features over an audio segment that is longer than a frame but still shorter than a song. Similar to above, distinct segment classifiers are combined for the final decision about a song. In [6], a neural network summarizes the audio features over musically significant timescales using an unsupervised pre-training and in [7] an ensemble learner selects from a set of audio features. While promising results have been reported in [7] using this approach for genre recognition, the effect of the segment size is not clearly known for artist recognition.

The third approach builds a single set of song-level features. In [8], a song-level feature is made using full-covariance Gaussian densities and in [9], GMM super-vectors extracted from a song and a distance measure are used to find the similar songs. Compact signatures are generated for a song in [10], then are compared using bipartite graph matching. Also, multivariate kernels [11] have been used to build a model of an artist and assign songs to artists using a sequence of features, with promising results.

Besides these three approaches, other techniques such as sparse modeling and vocal separation have been used to improve artist recognition performance. For instance, [12] investigated sparse modeling techniques for singing voice separation and unsupervised feature learning of group-delay functions for an artist recognition task.

The task of speaker verification is to either accept or reject the identity claimed by an speaker, based on a sample of his voice. This task is similar to music artist recognition since both try to find similarities between different instances of an individual's audio sample. Recently in the field of speaker verification, Dehak et al. [13] introduced **i-vectors** which significantly outperformed the state of the art. I-vector is a feature-modeling technique that builds utterance-level features using MFCCs. It has been successfully used in other areas such as emotion recognition [14], language recognition [15], accent recognition [16] and audio scene detection [17]. In this paper, we propose new song-level features for music artist recognition based on i-vectors. For our experiments, we use the standard artist20 dataset [4]. Multiple classifiers are tested, using a 6-fold leave-one-album-out

[1] From now on, we use the term **music artist** or **artist** to refer to the singer or the band of a song.

978-0-9928626-3-3/15/$31.00 ©2015 IEEE

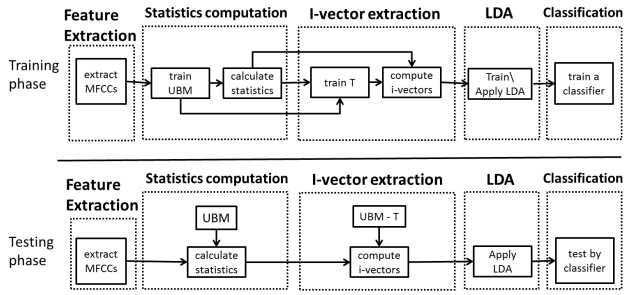

Fig. 1. Block diagram of our artist recognition system employing proposed modeling technique.

validation procedure as proposed in [4].

2. THEORETICAL BACKGROUND

In this section, we describe the basic concept of i-vectors, which we will use for timbral modeling. Our method consists of 5 steps: (i) feature extraction, (ii) computation of Baum-Welch statistics, (iii) i-vector extraction, (iv) linear discriminant analysis and (v) classification. A block diagram of the proposed system can be found in Figure 1. After MFCC feature extraction, a set of statistics are computed for each song and are used as a high-dimensional super-vector. A similar approach using Gaussian super-vectors can be found in [9]. We apply a post-processing method called i-vector extraction [13] to these statistics super-vectors, which transforms them into an information-rich low-dimensional vector, providing a space that best separates different artists and also reduces the dimensionality from a couple of thousand dimensions (the super-vector) to a few hundred (the i-vector). Then, Linear Discriminant Analysis (LDA) is carried out to remove the irrelevant dimensions and at the end, the output is fed into the classifier.

2.1. Feature extraction

MFCCs have proven to be useful features for many audio and music processing tasks [3, 4, 18]. They provide a compact representation of the spectral envelope and are a musically meaningful representation. Even though there are other representations based on MFCCs such as [7], we stay away from feature-engineering and focus on the timbral modeling technique. For the experiments at hand, we have extracted two sets of 13- and 20-dimensional MFCCs. We used the 20-dimensional MFCCs provided in the artist20 [4] dataset and also extracted 13-dimensional MFCC features to assess how much performance drops when less information is used.

2.2. Statistics computation

After extracting MFCCs, a set of statistics are computed for each song. These statistics are known as 0^{th} and 1^{st} order **Baum-Welch** (BW) statistics and are calculated using a Universal Background Model (UBM) [19]. UBM is a Gaussian Mixture Model (GMM) composed of hundreds of Gaussians which are trained on the MFCCs of songs from all the singers, aiming at modeling the overall MFCC distribution over all songs. Using a sequence of L MFCC frames from a specific song and UBM component c, where $c = 1, \ldots, C$ and C is the total number of Gaussian components, the Baum-Welch statistics of the song are computed as follows:

$$(\ 0^{\text{th}} \text{ order statistics})\ \ N_c = \sum_{t=1}^{L} \gamma_t(c) \tag{1}$$

$$(\ 1^{\text{st}} \text{ order statistics})\ \ F_c = \sum_{t=1}^{L} \gamma_t(c) Y_t \tag{2}$$

where $\gamma_t(c)$ is the posterior probability of Gaussian component c for frame t and Y_t is the MFCC feature vector at frame t. These statistics are then centered by removing the mean. The dimension of a single N_c for a component c is 1; for each c, F_c has $D \times 1$ dimensions where D is the dimension of a MFCC vector (see an example in Section 3.1).

2.3. I-vector extraction

The term **identity vectors** or **i-vectors** was introduced by Dehak et al. [13]. An i-vector refers to vectors of a low-dimensional space called **Total Variability Space (TVS)**. The TVS models both artist and session variability [20] where, in our context, the session variability would be the variability exhibited by a given artist from one song to another. The TVS is obtained by factor analysis, via a similar procedure as in [21]. In the resulting new space, a given song is represented by an **i-vector** which indicates the directions that best separate different artists. A rectangular matrix T of low rank is used to extract i-vectors from the statistical super-vector of a song. Conceptually, given a T matrix, the super-vector M extracted from a song of artist α decomposes as follows:

$$M = m + Tw \tag{3}$$

where M is obtained by appending the first-order statistics for all Gaussian components, m is the artist- and session-independent vector and is estimated using UBM and $w \sim \mathcal{N}(0, 1)$ is the artist- and session-dependent vector, referred to as the i-vector.

The subspace matrix T is estimated via expectation maximization using statistics extracted from the training set. More information about the training procedure of T can be found in [13, 22]. The actual computation of an i-vector w for a given song can be done using the following equation:

$$w = (I + T^t \Sigma^{-1} N(s) T)^{-1} \cdot T^t \Sigma^{-1} F(s) \tag{4}$$

We define $N(s)$ as a diagonal matrix with $CD \times CD$ dimensions with diagonal blocks of $N_c * I$ ($c = 1, \ldots, C$ and I has $D \times D$ dimensions). $F(s)$ is defined as a vector with $CD \times 1$ dimensions and generated by concatenating all first-order Baum-Welch statistics F_c for a given song s (N_c and F_c are described in Section 2.2 above). M is the super-vector of the song, and Σ is a diagonal covariance matrix of dimension $CD \times CD$ estimated during factor analysis training; it models the residual variability not captured by the total variability matrix T.

2.4. Linear Discriminant Analysis (LDA)

After extracting and centralizing the i-vectors, LDA [23] is applied to remove unnecessary or irrelevant dimensions in the TVS. If different songs from a given artist are assumed to represent one class, LDA minimizes the intra-class variance caused by artist-independent effects and maximizes the variance between artists.

2.5. Classification

Multiple classifiers were used to classify our song-level features: (i) K-Nearest Neighbor (KNN), (ii) Naive Bayes (NB), (iii) Discriminant Analysis (DA), (iv) Probabilistic Linear Discriminant Analysis (PLDA). Cosine distance has been successfully used with i-vectors [13] to calculate the similarity between train and test i-vectors. Hence, we use the cosine distance with our KNN classifier. Naive Bayes classifiers have been successfully tested with i-vectors in [16]. Discriminant Analysis (DA) assumes different classes have different Gaussian distributions. It is a suitable method since i-vectors are assumed to be normally distributed. Probabilistic Linear Discriminant Analysis (PLDA) [24] is a generative model which models both intra-class and inter-class variance as multidimensional Gaussian and proved to be successful with i-vectors [25]. In our experiments, i-vectors are length normalized [26] before apply PLDA, DA is used with a linear discriminant function and a uniform prior, and the KNN classifier with a cosine distance and k=3.

3. PROPOSED TIMBRAL MODELING METHODS

In this section, using the theoretical background described above, we introduce our specific method for computing song-level features that models timbre for the artist recognition task. To illustrate the importance of the (complex) i-vector component of the method, we also test an alternative modeling system that extracts similar song-level features without using i-vectors. The resulting features are supplied to the same classifiers and the results are compared to each other.

3.1. Timbral modeling method: I-vector - LDA

A 400-dimensional TVS is proposed to extract i-vectors from statistical super-vectors. A UBM with 1024 components is trained to compute statistical super-vectors (for example when 20-MFCCs are used, the 0^{th} order BW statistics have 1024×1 dimensions and 1^{st} order BW statistics have 1024×20 dimensions). I-vectors are extracted from these super-vectors, fed into a LDA and the dimension reduced to 19, then the output is used to train our classifiers. A block-diagram of our proposed method is shown in Figure 1. This method is applied on two sets of 13- and 20-dimensional MFCC features. Below, the proposed method using the DA classifier is named **ivecDA**, and analogously for the other three classifiers (3NN, NB, and PLDA).

3.2. Alternative timbral modeling method: PCA - LDA

In this alternative timbral modeling method, the same procedure as described in Section 3.1 is used, but instead of the i-vector extraction block, a Principal Component Analysis (PCA) is applied on statistical super-vectors to reduce the dimensionality to 400. A 1024 components GMM is used to compute statistical super-vectors , in the same way as in Section 3.1. The same classifiers as described in Section 3.1 are used to classify these song-level features. In the results section, this alternative method is entitled as **ALTpcaDA**.

3.3. Resources

The MSR Identity Toolbox [27] was modified for i-vector extraction and PLDA. We use drtoolbox [28] to apply LDA and PCA. For the 20-dimensional MFCCs, we use the features provided with the dataset, which are also used by one of our baseline methods [4]. For the 13-dimensional MFCCs, we use MIRTOOLBOX [29] with 40 frequency bands, 25 ms window length and 50% overlap to extract features from 32kbps mp3 files provided in the dataset. This is because another baseline method [11] also used it to extract 13-MFCCs. We use 2000 randomly-selected frames from the middle area of each song to compute Baum-Welch statistics, assuming that this middle area of the song contains the most singing voice data.

4. EXPERIMENTS

4.1. Dataset and Evaluation method

All the experiments reported in this paper are done using the artist20 dataset [4]. It contains 1413 tracks, mostly rock and pop, composed of six albums each from 20 artists. We perform 6-fold cross-validation, with five albums from each artist used for training and one for testing in each iteration, as proposed in [4]. We report mean class-specific accuracy, F1, precision and recall, first averaging over the classes, then over the folds. In each iteration, only the training folds are used

to train T and UBM, which are then used in classifying the independent test cases. To speed up the process, we use only a randomly selected $\frac{1}{3}$ of all the songs in the training folds to train the UBM; for learning T, all the training songs are used.

4.2. Baseline methods

Multiple baseline methods from the literature are compared to our method. Results are reported for a 20-class artist recognition task on the artist20 [4] dataset. The first baseline (*BLGMM*) models artists with Gaussian mixture models [4] whose frame-level feature representation is MFCCs. The second baseline (*BLsparse*) applies a sparse feature learning method [12] with a 'bag of features' (bof) using both the magnitude and phase parts of the spectrum. The third baseline (*BLsignature*) generates compact signatures for each music track using a 15-dimensional MFCC feature set and compares these using bipartite graph matching [10]. The fourth baseline (*BLmultivar*) uses multivariate kernels [11] with the direct uniform quantization of the 13-dimensional MFCC features. The results for the latter three are taken from their publications, while the results for *BLGMM* baseline are reproduced using the implementation provided with the dataset. All baselines reported performance on the artist20 dataset using the same songs, and the same fold splits in the cross-validation.

4.3. Results and discussion

Table 1 summarizes the results. As can be seen, our method clearly outperformed the baselines: compared to the 13-MFCC variant of our method, the accuracies achieved by BLGMM, BLsparse, BLsignature and BLmultivar are below our results by, respectively, **4.66**, **2.25**, **1.42** and **0.84 standard deviations** (the standard deviation of the accuracy over the 6 folds for our method (ivecDA) was 4.82). When we use 20-dimensional MFCC features, the differences are **3.86**, **2.29**, **1.74** and **1.36** std. deviations (the standard deviation of the accuracy of our method (ivecDA) was 7.35). As expected, using more coefficients in MFCCs improves the performance: 20-MFCCs achieved better results than 13-MFCCs. Comparing the performance of different classifiers using the proposed song-level features in Table 1, we see that the proposed features yield stable results using different classifiers and manage to achieve good performances using different classification models. The proposed method with DA classifier (ivecDA) performs best.

Table 2 gives the results of our method using the DA classifier with different number of Gaussian components. It shows that increasing the number of Gaussian components improves the classification accuracy. The maximum number of 1024 Gaussians is used in this paper due to computation limits and long training time.

Our final observation refers back to Table 1: comparing the results of our proposed method to the alternative method with PCA instead of i-vectors (*ALTpcaDA*) clearly reveals that i-vector extraction is more effective than the PCA in finding the best artist directions in feature space, thus justifying the increased computational effort.

Method	Feats.	Acc %	F1 %	Prec %	Rec %
BLGMM	20-mfcc	55.90	55.18	58.74	58.20
BLsparse	bof	67.50	-	-	-
BLsignature	15-mfcc	71.50	-	-	-
BLmultivar	13-mfcc	74.33	74.79	-	-
ivecDA	**13-mfcc**	**78.37**	**77.83**	**80.38**	**77.94**
ivec3NN	13-mfcc	78.32	76.94	79.13	77.96
ivecNB	13-mfcc	77.89	77.21	79.35	77.51
ivecPLDA	13-mfcc	77.79	76.55	77.86	77.83
ALTpcaDA	13-mfcc	60.55	59.54	65.08	60.35
ivecDA	**20-mfcc**	**84.31**	**83.68**	**84.92**	**84.67**
ivec3NN	20-mfcc	83.70	82.56	83.28	83.91
ivecNB	20-mfcc	83.90	83.28	84.91	83.97
ivecPLDA	20-mfcc	83.22	82.02	82.88	83.57
ALTpcaDA	20-mfcc	67.95	67.08	71.39	68.12

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

Gauss. #	Feats.	Acc %	F1 %	Prec %	Rec %
128	13-mfcc	71.18	73.6	72.90	71.00
256	13-mfcc	74.47	73.71	75.62	74.41
512	13-mfcc	76.71	75.88	77.79	76.64
1024	**13-mfcc**	**78.37**	**77.83**	**80.38**	**77.94**
128	20-mfcc	80.07	79.37	81.35	80.3
256	20-mfcc	82.13	81.62	83.32	82.38
512	20-mfcc	83.53	82.94	83.93	83.81
1024	**20-mfcc**	**84.31**	**83.68**	**84.92**	**84.67**

Table 2. Artist recognition results for **different Gaussian numbers** with the **proposed method** and the **DA** classifier on the **artist20** dataset.

5. CONCLUSION AND FUTURE WORK

In this paper, a new timbral modeling technique was proposed to extract song-level features for the task of music artist recognition. Using these song-level features, an **84.31%** accuracy and **83.68%** F1 on the artist20 dataset were achieved. To the best of our knowledge, these results are the highest artist recognition results published so far for the artist20 dataset. The new features were evaluated on a variety of classifiers and proved to yield stable results. We can conclude that our timbre modeling method outperforms other current approaches. We also observed that using more coefficients in MFCCs improves the recognition performance and 20-MFCCs outperformed 13-MFCCs. The effect of the number of Gaussians is reported by using multiple components. We found that the accuracy increases as the number of Gaussian rises, which indicates that the number of Gaussian components plays a sig-

nificant role in the modeling process. A comparison with a system using PCA instead of i-vector extraction supported the superiority of the i-vector modeling approach. In the future, we will investigate the use of a singing voice detection system instead of randomly choosing the frames from the middle of a song. Also, we would like to study the performance of our method in a more complex problem by increasing the number of the classes (i.e., singers).

6. ACKNOWLEDGMENTS

We would like to acknowledge the tremendous help by Dan Ellis of Columbia University, who provided tools and resources for feature extraction and shared the details of his work, which enabled us to reproduce his experiment results. Thanks also to Pavel Kuksa from University of Pennsylvania for sharing the details of his work with us. And at the end, we appreciate helpful suggestions of Marko Tkalcic from Johannes Kepler University of Linz. This work was supported by the EU-FP7 project no.601166 (PHENICX).

REFERENCES

[1] L. R Rabiner and B. H Juang, *Fundamentals of speech recognition*, PTR Prentice Hall Englewood Cliffs, 1993.

[2] T. Zhang, "Automatic singer identification," in *Multimedia and Expo, ICME'03. Proceedings.* IEEE, 2003.

[3] B. Logan et al., "Mel frequency cepstral coefficients for music modeling.," in *ISMIR*, 2000.

[4] D. PW Ellis, "Classifying music audio with timbral and chroma features," in *ISMIR*, 2007.

[5] Y. E Kim and B. Whitman, "Singer identification in popular music recordings using voice coding features," in *ISMIR*. 2002, IEEE.

[6] S. Dieleman, P. Brakel, and B. Schrauwen, "Audio-based music classification with a pretrained convolutional network," in *ISMIR*, 2011.

[7] J. Bergstra, N. Casagrande, D. Erhan, D. Eck, and B. Kégl, "Aggregate features and adaboost for music classification," *Machine learning*, 2006.

[8] M. I Mandel and D. PW Ellis, "Song-level features and support vector machines for music classification," in *ISMIR*, 2005.

[9] C. Charbuillet, D. Tardieu, G. Peeters, et al., "Gmm supervector for content based music similarity," in *DAFx-11*, 2011.

[10] S. Shirali-Shahreza, H. Abolhassani, and M Shirali-Shahreza, "Fast and scalable system for automatic artist identification," *Consumer Electronics, Transactions on*, 2009.

[11] P Kuksa, "Efficient multivariate kernels for sequence classification," *CoRR*, 2014.

[12] L. Su and Y. H Yang, "Sparse modeling for artist identification: Exploiting phase information and vocal separation.," in *ISMIR*, 2013.

[13] N. Dehak, P. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *Audio, Speech, and Language Processing, Transactions on*, 2011.

[14] R. Xia and Y. Liu, "Using i-vector space model for emotion recognition.," in *INTERSPEECH*, 2012.

[15] N. Dehak, P. A Torres-Carrasquillo, D. A Reynolds, and R. Dehak, "Language recognition via i-vectors and dimensionality reduction.," in *INTERSPEECH*. Citeseer, 2011.

[16] M. Hasan Bahari, R. Saeidi, D. 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.

[17] B. Elizalde, H. Lei, and G. Friedland, "An i-vector representation of acoustic environments for audio-based video event detection on user generated content," in *Multimedia (ISM), International Symposium on*. IEEE, 2013.

[18] G. Tzanetakis and P. Cook, "Musical genre classification of audio signals," *Speech and Audio Processing, transactions on*, 2002.

[19] D. A Reynolds, T. F Quatieri, and R. B Dunn, "Speaker verification using adapted gaussian mixture models," *Digital signal processing*, 2000.

[20] P. Kenny, G. Boulianne, P. Ouellet, and P. Dumouchel, "Speaker and session variability in gmm-based speaker verification," *Audio, Speech, and Language Processing, Transactions on*, 2007.

[21] P. Kenny, "Joint factor analysis of speaker and session variability: Theory and algorithms," *CRIM, Montreal,(Report) CRIM-06/08-13*, 2005.

[22] D. Matrouf, N. Scheffer, B. GB Fauve, and J. Bonastre, "A straightforward and efficient implementation of the factor analysis model for speaker verification.," in *INTERSPEECH*, 2007.

[23] B. Scholkopft and K. Mullert, "Fisher discriminant analysis with kernels," in *Signal Processing Society Workshop Neural Networks for Signal Processing*, 1999.

[24] S. JD Prince and J. H Elder, "Probabilistic linear discriminant analysis for inferences about identity," in *Computer Vision, ICCV: 11th International Conference on*. IEEE, 2007.

[25] L. Burget, O. Plchot, S. Cumani, O. Glembek, P. Matejka, and N. Brummer, "Discriminatively trained probabilistic lda for speaker verification," in *ICASSP*. IEEE, 2011.

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

[27] S. O Sadjadi, M. Slaney, and L. Heck, "Msr identity toolbox v1.0: A matlab toolbox for speaker-recognition research," *Speech and Language Processing Technical Committee Newsletter*, 2013.

[28] LJP Van der Maaten, EO Postma, and HJ van den Herik, "Matlab toolbox for dimensionality reduction," *MICC*, 2007.

[29] O. Lartillot, P. Toiviainen, and T. Eerola, "A matlab toolbox for music information retrieval," in *Data analysis, machine learning and applications*. Springer, 2008.