Genre Classification and the Invariance of MFCC Features to Key and Tempo

Tom LH. Li and Antoni B. Chan

Department of Computer Science, City University of Hong Kong, Hong Kong
lihuali2@cityu.edu.hk, abchan@cityu.edu.hk

Abstract. Musical genre classification is a promising yet difficult task in the field of musical information retrieval. As a widely used feature in genre classification systems, MFCC is typically believed to encode timbral information, since it represents short-duration musical textures. In this paper, we investigate the invariance of MFCC to musical key and tempo, and show that MFCCs in fact encode both timbral and key information. We also show that musical genres, which should be independent of key, are in fact influenced by the fundamental keys of the instruments involved. As a result, genre classifiers based on the MFCC features will be influenced by the dominant keys of the genre, resulting in poor performance on songs in less common keys. We propose an approach to address this problem, which consists of augmenting classifier training and prediction with various key and tempo transformations of the songs. The resulting genre classifier is invariant to key, and thus more timbre-oriented, resulting in improved classification accuracy in our experiments.

1 Introduction

Musical information retrieval is a field that is growing vigorously in recent years thanks to the thriving digital music industry. As a promising yet challenging task in the field, musical genre classification has a wide-range of applications: from automatically generating playlists on an MP3 player to organizing the enormous billion-song database for major online digital music retailers. In many genre classification systems, the Mel-frequency cepstral coefficients (MFCCs) [3] have been used as a timbral descriptor [15,12,10,7]. While it is common to think of MFCCs as timbre-related features, due to the short-duration frame on which they are extracted (e.g., 20 milliseconds), it is still uncertain how the key and tempo of a song affects the MFCC features, and hence the subsequent genre classification system.

In this paper, we attempt to address the following question: are MFCCs invariant to key and tempo? In other words, is MFCC a purely timbral feature set? If the MFCCs are purely timbral features, then they should be invariant to the changes in musical keys and tempo. Otherwise, changes in the musical key and tempo of a song will affect the MFCCs, which may adversely affect the training of genre classifiers. The contributions of this paper are three-fold. First, we show that musical genres, which *should* be independent of key, are in

K.-T. Lee et al. (Eds.): MMM 2011, Part I, LNCS 6523, pp. 317–327, 2011.

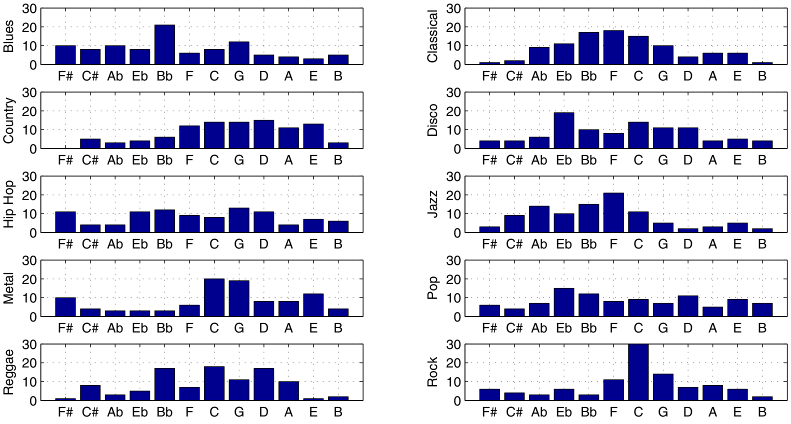

Fig. 1. Key histograms of the GTZAN dataset on the circle of fifths scale. The vertical axis is the number of songs with a certain key.

fact influenced by the fundamental keys of the instruments involved. Second, we show that MFCCs indeed encode both timbral and key information, i.e., they are not invariant to shifts in musical key. As a result, genre classifiers based on the MFCC features will be influenced by the dominant keys of the genre, resulting in poor performance on songs in less common keys. Third, we propose an approach to build key-independent genre classifiers, which consists of augmenting the classifier training and prediction phases with various key and tempo transformations of the songs. The resulting genre classifier is invariant to key, and thus more timbre-oriented, resulting in improved classification accuracy in our experiments.

The rest of this paper is organized as follows. In Section 2, we explore the distribution of musical key for different genres. In Section 3, we study the invariance of MFCC to musical key and tempo shifts. In Section 4, we propose a data-augmented genre classification scheme, based on key and tempo transformations, while in Section 5 we present experiments on genre classification using our data-augmented system.

2 Key Histograms of the GTZAN Dataset

In this section, we explore the relationship between musical genres and musical keys. We manually annotate each song in the GTZAN dataset [15] with their musical "keys". In this section, we define the concept of "key" as the pitch of the "Do" sound of the song in the solfège scale (Do-Re-Mi scale). Such definition is different from the more common definition — the tonic sound of the scale (e.g., in minor scales the tonic sound is the La sound rather than the Do sound). Because a major scale and its relative minor scale share the identical composition of pitches, it is simpler to annotate both scales with the same label to show that they actually have the same pitch ingredients in the songs (e.g., songs in C major and A minor are both labeled with "C"). In cases where the scale is unapparent, we annotate the key based on the most repeated pitch.

Figure 1 shows the key histograms for different genres in the GTZAN dataset, using our annotation criteria, with keys ordered by the circle of fifths (C is in the center). We observe that genre is indeed key-related with the distribution centered around particular keys based on the instrumentation.

- Blues: peaks at B♭ and G. B♭ is the fundamental pitch of many horn instruments. G corresponds to the Do sound for the blues scale in E, which is the fundamental key for guitar.
- Classical: distribution around F, which is in between the horn instrument fundamental B♭ and the piano fundamental C.
- Country: broad distribution around D, with keys that are easy to play on guitars (e.g. G, D, A, E, C).
- Disco: peaks at E♭ and C. Disco frequently employs Blues scale. For C Blues, the Do sound is E♭.
- Hip Hop: distribution is not obvious. This genre typically does not have a key, as the main instruments are human voice and drums.
- Jazz: distribution is skewed towards flat keys (Db, Ab, Eb, Bb), which are the fundamental horn pitches. The peak at F is similar to that of Classical.
- Metal: peaks at C, G, E and F♯. The G key correspond to E Blues. E is the pitch of the lowest string on guitar. In Metal, the lowest string is used extensively to create a massive feeling. The peak at F♯, corresponding to E♭ Blues, can be explained by the common practice of Metal artists to lower the tuning by one semi-tone, creating an even stronger metal feeling.
- Pop: distribution is not obvious. The peak at E♭ is the Blues-scale of the C key. The distributions of Pop and Disco are similar, due to similar instrumentation.
- Reggae: peaks at C (keyboard), D (guitar), B♭ (horns) and C♯ (B♭ Blues).
- Rock: significant distribution around C. The distribution is be related to the dominance of guitar and piano in this genre. Rock is arguably the most key-related genre in the GTZAN dataset.

In summary, there is a strong correlation between genre and key, with each genre having a unique key distribution. Such correlation most likely stems from the fundamental keys associated with the instruments used in each genre. For instance, the most common kind of clarinet is in the key of B♭, while the alto saxophone is in E♭. The four strings of a violin are tuned by standard to G, D, A and E. The piano has all its white keys in C major. Although it is entirely possible to play a song in any key, some keys are arguably easier to play than others, depending on the instruments used. Hence, the key characteristics of instruments could unexpectedly associate musical keys to specific genres.

3 Are MFCCs Invariant to Key and Tempo?

In this section we study the invariance of MFCCs to shifts in musical key and tempo.

3.1 Mel-Frequency Cepstral Coefficients

The mel-frequency cepstral coefficients (MFCC) [3] are a widely adopted audio feature set for various audio processing tasks such as speech recognition [13], environmental sound recognition [9], and music genre classification [15,2,7,11]. [11] investigated the MFCC features on various time scales and with different modeling techniques, such as autoregressive models. [15,8] compared the MFCCs to the short-time Fourier transform (STFT), beat histogram and pitch histogram feature sets, concluding that MFCCs give best performance as an independent feature set.

Given a frame of audio, the computation of MFCC involve the following steps that mimic the low-level processing in the human auditory system [3]: 1) transformation of the audio frame to the frequency domain using the STFT; 2) mapping the frequency bins to the mel-scale, using triangular overlapping windows; 3) taking the logs of the mel-band responses; 4) applying a discrete cosine transform (DCT) to the mel-bands. In this paper, the MFCCs are extracted with the CATBox toolbox [4], using 40 mel-bands and 13 DCT coefficients. The frame size is 18 milliseconds, taken every 9 milliseconds.

3.2 Key and Tempo Transformations

To examine the changes of MFCC values to shifts in keys and tempos, we apply key shifting and tempo shifting musical transforms to each song in the GTZAN dataset. These transformations consist of sharpening/flattening the song up to 6 semitones, and changing the tempo 5% and 10% faster/slower. The transformations are performed with the WSOLA algorithm [16], which is implemented in the open-source audio editor Audacity [1]. The musical transforms are analogous to affine transforms of images, which deform an image without changing the general shape (e.g. rotating and skewing the number 1). Augmenting the dataset with affine transforms is a common technique in digit recognition tasks [14], where the enlarged training set improves classification accuracy by encouraging invariance to these deformations.

There are doubts that transforming a song to approximate the key-shifted and tempo-shifted version of the songs might not be appropriate, since such transforms might also contaminate the timbral characteristics of the songs. We argue that such an effect is minor for the following three reasons: 1) qualitatively speaking, the transformed songs sound perceptually very similar to the original song recorded in different key and tempo, with critical information for genre classification, such as instruments, musical patterns and rhythm characteristics, still preserved; 2) considering that musical instruments have different timbre in different registers, we limit the key shifts to the range of half an octave (from ♭6 to ♯6); 3) we compared the MFCC values extracted from MIDI songs and their perfect key-transposed versions, and observed that the MFCC values vary in similar ways as in the key-transformed songs.

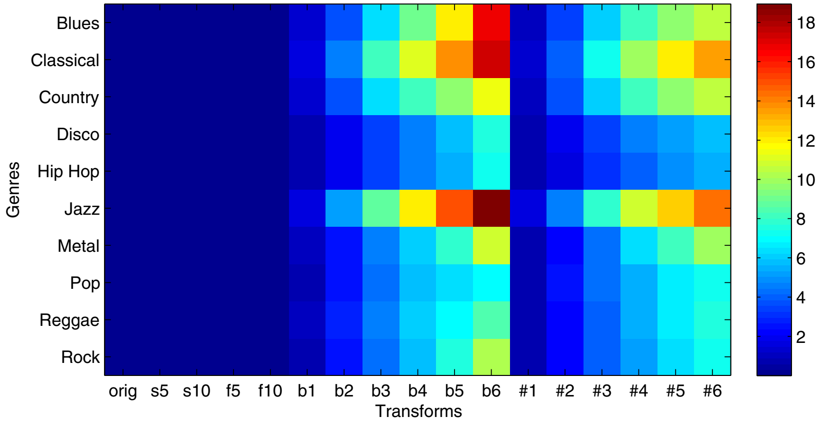

Fig. 2. MFCC KL-divergence: the horizontal axis represents the key and tempo transforms, from left to right, original, 5% slower, 10% slower, 5% faster, 10% faster, key transform ♭1 to ♭6 and ♯1 to ♯6. The color represents the average KL divergence between corresponding frames in the original and transformed songs.

3.3 Comparison of MFCCs under Key and Tempo Transforms

For genre classification, MFCCs are often aggregated over a long-duration window using statistical methods [15,2]. Motivated by this fact, we compare the original songs and their transformed versions by computing the Kullback-Leibler (KL) divergence [5] between corresponding windowed excerpts (3.5 seconds). Assuming that the MFCCs in a window follow a Gaussian distribution (e.g., as in [15]), the calculation of KL divergence between two windows is given by:

$$D_{KL}\left(\boldsymbol{N}_0 \parallel \boldsymbol{N}_1\right) = \frac{1}{2}\left(\log\frac{|\Sigma_1|}{|\Sigma_0|} + \mathrm{tr}\left(\Sigma_1^{-1}\Sigma_0\right) + \left(\mu_1 - \mu_0\right)^\top \Sigma_1^{-1}\left(\mu_1 - \mu_0\right) - d\right) \quad (1)$$

where (μ_0, Σ_0) and (μ_1, Σ_1) are the mean and covariance for the two Gaussian distributions, and d is the dimension.

Figure 2 shows the KL divergence between different musical transforms of the same songs, averaged over each genre. From the figure, we see that key transforms affect the MFCC distribution, with larger key shifts affecting the distribution more. Interestingly, MFCCs for some genres are more sensitive to the changes in key, such as blues, jazz and metal. This can be explained by the fact that these genres have instruments with richer harmonic structure, and therefore the MFCCs change more since they model timbre. On the other hand, tempo transforms do not have a great effect on the distribution of MFCC values. This is because transforming a song in time does not change the frequency characteristics, but only the number of MFCC frames. Compressing a song subsamples the MFCC frame set, while stretching it adds new MFCC frames by interpolation. In both cases, the distribution of the MFCCs over the window remains about the same.

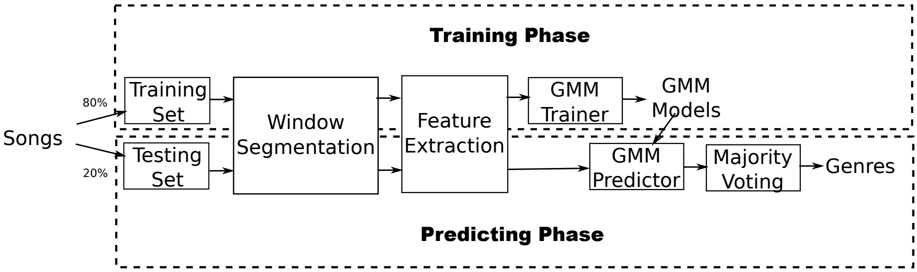

Fig. 3. System architecture

In the previous, we showed that genres have dominant keys, due to the instrumentation of the genre. On the other hand, in this section, we have shown that MFCCs, which are common features for genre classification, are not invariant to key transformations. This brings forward an interesting dilemma. Because genre is key dependent and MFCCs are not key invariant, then a classifier based on MFCCs may overfit to the dominant keys of the genre. The resulting classifier will then have poor accuracy on songs in the less common keys. In the next section, we look at learning a key-invariant genre classifier, by augmenting the classifier with different musical transforms.

4 Genre Classification with Musical Transforms

In this paper, we adopt the genre classification system of [15,2,11]. Figure 3 shows the architecture of the system, which contains four steps. First, the input song is split into non-overlapping windows of equal length (as in [2], we use window length of 3.5 seconds). These windows then go through a feature extraction process, producing feature vectors which are compact representations of those windows. In particular, MFCCs are first extracted from the audio signal, and the mean and standard deviation of the MFCCs over the window are calculated as the feature vector. In the third step, the feature vector is fed to a Gaussian mixture model (GMM) classifier. The parameters of the GMM classifier are learned from the training set using the EM algorithm [6], which iteratively estimates the parameters by maximizing the likelihood of the training set. One GMM is learned for each genre. Given a feature vector extracted from a window, the GMM with the largest likelihood is selected as the genre label for the window. The labels for all the windows in a song are then aggregated with a majority voting process to produce a genre label for the song.

We can modify the genre classification system in two ways to make it invariant to musical transforms. First, in the training phase, we can expand the training set by adding transformed versions of the training songs, hence generating more examples for learning the genre classifier. Second, in the prediction phase, we can augment the classifier by processing the test song along with its transformed versions. The final label for the test song is the majority vote over all windows of all versions of the songs. The data augmentation step can be seen as adding a sample diffusion layer before either the training or the predicting phase of the system.

5 Experiments

In this section we present our experimental results on genre classification in the context of key and tempo augmentation.

5.1 Dataset and Experimental Setup

In our experiments, we use the GTZAN dataset [15], which contains 1000 song clips of 30 seconds each, with a sampling rate of 22050 Hz at 16 bits. There are 10 musical genres, each with 100 songs: Blues, Classical, Country, Disco, Hip hop, Jazz, Metal, Pop, Reggae, and Rock. We augment the original GTZAN dataset (denoted as the "Orig" dataset) using different combinations of musical transforms. The "Tempo" dataset contains the Orig dataset and its tempo variants, 5% and 10% faster/slower. The "Key" dataset contains the Orig dataset and its key variants from $\flat 6$ to $\sharp 6$. The "Tempokey" dataset is the union of the Tempo and Key datasets. We also augment our dataset with key transforms that are based on the circle of fifths. The "Fifth1" dataset contains the Orig dataset and its key variants with one step on the circle of fifths, i.e. $\flat 5$ and $\sharp 5$, while the "Fifth2" dataset contains variants with one more step, i.e. $\flat 2$ and $\sharp 2$. The circle of fifths augmented datasets are strict subsets of the Key dataset.

We carried out three different sets of experiments in combination with the 6 augmentations listed above. In the first experiment, denoted as AugTrain, the classifiers are trained using the augmented dataset, while genre prediction is performed using only the original songs. In the second experiment, denoted as AugPredict, the classifiers are trained only on the original dataset, while prediction is performed by pooling over the augmented song data. In the final experiment, denoted as AugBoth, both the classifier training and prediction use the augmented song data. Genre classification is evaluated using five random splits of the dataset, with 80% of the songs (and its variants) used for training, and the remaining 20% used for testing. The experiments are carried out on a range of parameters. We use MFCC lengths from 1 to 13 (i.e., the number of DCT coefficients), and vary the number of components in the GMM (K) from 1 to 20. We also assume diagonal covariance matrices in the GMM.

5.2 Experimental Results

We first examine the effects of the system parameters, such as the size of the GMM and the length of the MFCCs. Figure 4a shows the classification accuracy, averaged over all the data augmentations and MFCC lengths, while varying the number of components in the GMM. In general, the classification accuracy increases with K, and there does not seem to be an over-fitting problem for large K, such as 20. Figure 4b shows the accuracy, averaged over all data augmentations and GMMs, while varying the length of the MFCCs. Similarly, the

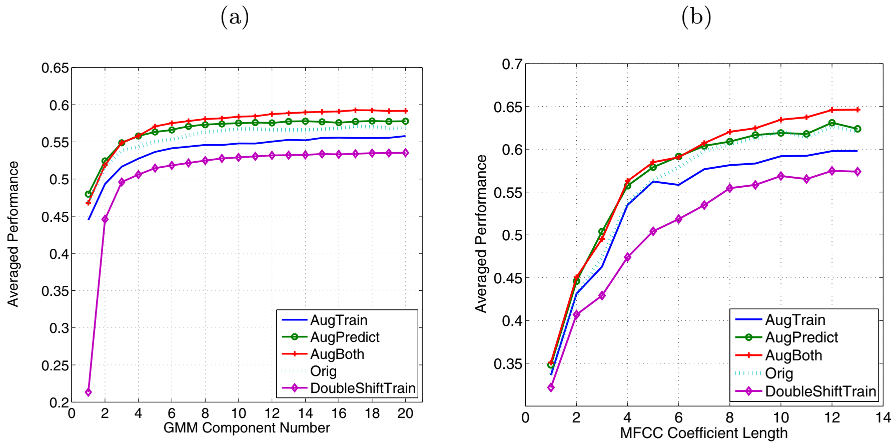

Fig. 4. (a) Averaged accuracy for all datasets and MFCC lengths, while varying the number of GMM components (K); (b) Averaged accuracy for all datasets and GMM components, while varying the MFCC length

accuracy improves as more MFCCs are added. In fact, despite their sensitivity to noise, these high-order coefficients provide useful details for genre classification. As a comparison, [15] limited their system to the first 5 MFCC coefficients and GMMs with K=5, and achieved 61% classification accuracy when using MFCCs with three other types of features. In contrast, our system scores 66.3% on the Orig dataset when using 13 MFCC features.

Next, we look at the effect of signal degradation when using the music transformation. In particular, we add noise to the Orig training set by applying a “double-shift” to each training song. This consists of first shifting the key of the song, and then shifting it back to the original scale. The result is a training song with noise added due to the musical transformation. The double-shifted training set is used to train the genre classifier, which then predicts genres on the Orig test data. This result is denoted as DoubleShiftTrain in Figure 4. In particular, using the noisy training data degrades the accuracy, when compared to the Orig performance (e.g, the accuracy drops 5% to 53.5% for K=20). However, in spite of this added noise to the training set, the system is still able to do genre classification, albeit with reduced accuracy.

Finally, we look at the effect of using the proposed data-augmented classifiers. From Figure 4, we observe that the AugTrain classifier gives constantly better performance than the DoubleShiftTrain classifier, while its performance is still lower than that of the Orig dataset. This suggests that using augmented training data improves the accuracy, at least compared to the unaugmented classifier using similar noisy training data. This improvement, however, is not enough to overcome the transformation noise. On the other hand, using data-augmented prediction (AugPredict) gives constantly better performance than the Orig dataset. Finally, using both data-augmented classification and prediction (AugBoth) achieves the best accuracy, dominating both AugPredict and

Table 1. Genre classification accuracy for different data-augmentation schemes and transformed datasets, for K=20 and MFCC length 13

	Tempo	Key	Tempokey	Fifth1	Fifth2	Average
Orig	–	–	–	–	–	64.5%
DoubleShiftTrain	–	–	–	–	–	61.9%
AugTrain	65.1%	62.0%	64.5%	60.5%	62.8%	63.0%
AugPredict	66.2%	63.6%	66.4%	61.0%	63.7%	64.2%
AugBoth	66.6%	67.8%	**68.9%**	67.5%	67.3%	67.6%

Table 2. AugBoth Classification Rates for different genres, with K = 20 and MFCC length 13

	blues	classical	country	disco	hip-hop	jazz	metal	pop	reggae	rock	average
Orig	59	92	62	41	64	86	77	58	61	45	64.5
Tempo	64	97	62	46	66	85	75	64	68	39	66.6
Key	62	99	67	55	65	90	83	64	60	33	67.8
Tempokey	63	98	67	55	65	91	87	61	63	39	68.9
Fifth1	61	98	67	52	63	88	83	63	62	38	67.5
Fifth2	64	94	63	58	63	90	79	64	66	32	67.3

Orig. Table 1 shows the average classification accuracy using different transformed datasets and data-augmentation schemes for K=20 and MFCC length 13. The best performance achieved for all experiments is 69.3%, using the AugBoth classifier with the Key transformations, K=18 and MFCC length 13.

Table 2 shows the classification accuracy for different genres using the AugBoth classifier. Comparing the genres, Classical has the highest accuracy, scoring over 90% on all datasets, followed by Jazz and Metal. In contrast, Disco and Rock are the two worst performing genres. In general, the augmentation of the dataset improves the genre classification. The only exception is the Rock genre, where augmentation always lowers the classification accuracy. Looking at the confusion matrix for AugBoth, we found that more instances of Rock are misclassified as Metal. On the other hand, Disco performs significantly better because less instances are misclassified as Blues, Pop and Rock.

5.3 Discussion

From these experimental results we have three conclusions. First, the MFCC feature set is largely a timbral feature set. From the confusion matrices we found that confusable genres have similar instrumentation. Additionally, genres with distinct instrumentation stand out from others easily, e.g., Classical uses orchestral instruments, while Metal has high frequency distorted guitar.

Second, in addition to timbral information, MFCCs also encodes key information, which eventually affects the genre classification accuracy. We observed that the key and tempo augmented classifiers have a significant change in performance over the baseline. Rock and Metal both use guitars and drums as the main instruments, but they have very different key distributions as shown in Figure 1. The confusion between Rock and Metal after key augmentation suggest that the classification of Rock music is partly due to musical keys. If we blur the lines between keys for these two genres, we are likely to lose such information, leading to a degradation of classification performance.

Third, making the genre classifier tempo- and key-invariant, via data augmentation, generally improves the classification accuracy. The accuracies of the AugTrain, AugPredict and AugBoth classifiers are significantly better than the noise-added DoubleShiftTrain baseline. Despite the noise from the imperfect musical transforms, the accuracy of the AugPredict and AugBoth classifiers are constantly better than the Orig baseline. These results suggest a method for boosting overall genre classification performance, by artificially generating transformed songs to augment the classifier training and prediction phases, thus strengthening the timbre-orientation of the classifier. However, some genres (e.g. Rock) will suffer from such augmentation since the recognition of that genre is partly due to musical keys.

While the concept of "musical genre" is perceptual and largely based on timbre information, there is still a strong correlation between genre and key, due to instrumentation, which should also be considered. Future work will look at combining timbral and key information, using appropriate machine learning models, to push the performance further. In addition, reducing the noise introduced by the musical transform will also likely improve the classification accuracy.

6 Conclusion

MFCCs are widely used audio features in music information retrieval. Extracted over a short-duration frame, MFCCs are typically perceived as a timbral descriptor. In this paper, we have shown that the MFCCs are not invariant to changes in key, and hence they encode both timbral and key information. On the other hand, we found that musical genres, which should be independent of key, are in fact influenced by the fundamental keys of the instruments involved. As a result, genre classifiers based on the MFCC features will be influenced by the dominant keys of the genre, resulting in poor performance on songs in less common keys. We suggested an approach to address this problem, which consists of data-augmentation during the classifier training and prediction phases, with key and pitch transformations of the song. The resulting genre classifier is invariant to key, and thus more timbre-oriented, resulting in improved classification accuracy in our experiments.

References

1. Audacity, the free, cross-platform sound editor, `http://audacity.sourceforge.net/`
2. Bergstra, J., Casagrande, N., Erhan, D., Eck, D., Kégl, B.: Aggregate features and Adaboost for music classification. Machine Learning 65(2), 473–484 (2006)

3. Bridle, J.S., Brown, M.D.: An experimental automatic word recognition system. JSRU Report, 1003 (1974)
4. Computer audition toolbox, http://cosmal.ucsd.edu/cal/projects/catbox/catbox.htm
5. Cover, T.M., Thomas, J.A.: Elements of information theory. John Wiley and Sons, Chichester (2006)
6. Dempster, A.P., Laird, N.M., Rubin, D.B., et al.: Maximum likelihood from incomplete data via the EM algorithm. Journal of the Royal Statistical Society. Series B (Methodological) 39(1), 1–38 (1977)
7. Ellis, D.: Classifying music audio with timbral and chroma features. In: Int. Symp. on Music Information Retrieval (ISMIR), pp. 339–340 (2007)
8. Li, T., Tzanetakis, G.: Factors in automatic musical genre classification of audio signals. In: IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, pp. 143–146 (2003)
9. Lu, L., Zhang, H.J., Li, S.Z.: Content-based audio classification and segmentation by using support vector machines. Multimedia Systems 8(6), 482–492 (2003)
10. Mandel, M., Ellis, D.: Song-level features and support vector machines for music classification. In: Proc. ISMIR, pp. 594–599. Citeseer (2005)
11. Meng, A., Ahrendt, P., Larsen, J.: Improving music genre classification by short time feature integration. In: Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2005), vol. 5 (2005)
12. Pachet, F., Aucouturier, J.J.: Improving timbre similarity: How high is the sky?. Journal of negative results in speech and audio sciences 1(1) (2004)
13. Pearce, D., Hirsch, H.G.: The AURORA experimental framework for the performance evaluation of speech recognition systems under noisy conditions. In: Sixth International Conference on Spoken Language Processing. Citeseer (2000)
14. Simard, P.Y., Steinkraus, D., Platt, J.: Best practices for convolutional neural networks applied to visual document analysis. In: International Conference on Document Analysis and Recogntion (ICDAR), pp. 958–962. IEEE Computer Society, Los Alamitos (2003)
15. Tzanetakis, G., Cook, P.: Musical genre classification of audio signals. IEEE Transactions on speech and audio processing 10(5), 293–302 (2002)
16. Verhelst, W., Roelands, M.: An overlap-add technique based on waveform similarity (WSOLA) for high quality time-scale modification of speech. In: IEEE International Conference on Acoustic Speech and Signal Processing, vol. 2, Institute of Electrical Engineers Inc., (IEE) (1993)