SPARSE CEPSTRAL CODES AND POWER SCALE FOR INSTRUMENT IDENTIFICATION

Li-Fan Yu, Li Su and Yi-Hsuan Yang

Research Center for Information Technology Innovation, Academia Sinica, Taiwan

ABSTRACT

This paper presents a novel feature representation called *sparse cepstral codes* for instrument identification. We first motivate the approach by discussing why cepstrum is suitable for instrument identification. Then we propose the use of sparse coding and power normalization to derive compact codes that better represent the information of the cepstrum. Our evaluation on both uni-source and multi-source instrument identification tasks show that the proposed feature leads to significantly better accuracy than existing methods. We further show that cepstrum obtained from power-scaled spectrum can do better than typical cepstrum especially in multi-source signal. The proposed system achieves 0.955 F-score in uni-source dataset and 0.688 F-score in multi-source dataset.

***Index Terms*—** cepstrum, power scale, sparse coding, dictionary learning, instrument identification

1. INTRODUCTION

The *cepstrum* of a time-domain, finite-length signal $\mathbf{x}$ is obtained by taking the logarithm of the magnitude of the Fourier transform of the signal and then computing the inverse Fourier transform [1],

$$\mathcal{C}_{\log}(\mathbf{x}) = \mathcal{F}^{-1}(\log|\mathcal{F}(\mathbf{x})|)\,, \tag{1}$$

where $\mathcal{F}(\cdot)$ denotes the Fourier transform. Features derived from the cepstrum, such as Mel-frequency cepstral coefficients (MFCCs), are often employed in contemporary audio signal processing systems [2–4]. Due to the log function in Eq. 1, the cepstrum converts signals combined by convolution (such as excitation and transfer functions in a source-filter model) into additive terms for linear separation. For example, for human voices, the low-frequency periodic excitation from the vocal cords and the formant filtering of the vocal tract would be in different regions in the cepstrum [5, 6].

To extract patterns from the cepstrum, a number of cepstral features have been proposed in the literature, with MFCC being possibly the most famous one [7]. MFCC is a computationally light representation because its dimension is usually low. However, such cepstral features might not capture every details of the cepstrum [8] because of the series of abstraction and transformation operations that convert raw cepstrum into meaningful statistics. For example, it is well-known that MFCC ignores much high-frequency information because of the logarithmic filter spacing above 1 kHz [2]. Although high-frequency information is less important for speech, this is not the case for music. Moreover, conventionally feature design processes are usually hand-crafted, making good feature extraction hard to design and difficult to optimize [9].

To come up with a feature representation that preserves the important information of the cepstrum, the *sparse coding* (SC) technique [10, 11] is investigated in this paper. SC seeks a succinct representation of raw features as a combination of only a few atoms (codewords) of a dictionary, which is assumed to be representative of the music universe [12]. Comparing to conventional audio features, SC offers greater flexibility in capturing the nuance of music signals, in that each dictionary atom can be considered as a quantization of the music universe, and that the quantization goes finer as the size of the dictionary increases. Because the sparse representation is high-dimensional but sparse, using linear support vector machine (SVM) [13] for training audio classifiers has been found effective. For instance, applying SC to the raw spectrum $\mathcal{F}(\mathbf{x})$ has led to state-of-the-art performance on music genre classification and music auto-tagging [14–16]. To our best knowledge, however, SC has rarely been applied to the cepstrum. We refer to the SC result of the cepstrum as "sparse cepstral codes" and evaluate its effectiveness as a feature representation for automatic instrument identification (i.e., asking a machine to reproduce the human labels of instrument in a music signal) [17–20].

The second contribution of the paper lies in the application of power normalization [21] to the cepstrum. It has been recently found that modifying the scale from conventional log scale to power scale such as square-root enhances the noise robustness of a cepstrum-based speaker identification system [22, 23]. Therefore, whether power normalization is beneficial for music signal processing is worth a study. In particular, we are interested in the case of predominant instrument identification from multi-source music signals [18], as the interference from subordinate instruments also challenges the robustness of the identification system.

Our performance study includes instrument identification from both uni-source (monophony) and multi-source (polyphony) music signals. We report an empirical comparison among spectrum and cepstrum features, with or without sparse coding. The evaluation result confirms the effective-

ness of the proposed sparse cepstral codes from power-scaled spectrum. We are able to improve the accuracy (measured in F-score) from 0.820 to 0.955 for a uni-source dataset [17], and from 0.630 to 0.661 for a multi-source dataset [18].

Section 2 describes the cepstrum and the techniques we proposed to compute the sparse cepstral codes. After presenting a system overview in Section 3, we report the experiments in Section 4. Finally, Section 5 concludes the paper.

2. CEPSTRAL FEATURE

2.1. Sparse cepstral codes

The computation of sparse cepstral codes involves three technical components: the computation of the cepstrum, SC, and dictionary learning. SC employs an l_1-regularizer in finding a sparse representation $\alpha \in \mathbb{R}^k$ of the input $\mathbf{y} \in \mathbb{R}^m$ over a dictionary $\mathbf{D} \in \mathbb{R}^{m \times k}$. This is often referred to as the LASSO problem [11],

$$\hat{\alpha} = f_{\text{SC}}(\mathbf{D}, \mathbf{y}) = \arg\min_{\alpha} \ \|\mathbf{y} - \mathbf{D}\alpha\|_2^2 + \lambda\|\alpha\|_1 \,. \quad (2)$$

Being a generic algorithm, SC can take any feature representation as the input $\mathbf{y}$. We refer to the SC result of the cepstrum $\mathbf{y} = \mathcal{C}_{\log}(\mathbf{x})$ as the sparse cepstral codes. While a straightforward linear-programming solver is computationally intensive, efficient algorithms that better exploit the properties of Eq. 2 have been proposed [24]. We adopted the least angle regression (LARS)-lasso algorithm [10] in this work.

The dictionary $\mathbf{D}$ is learned from an external, possibly unlabeled, dataset referred to as the *training corpus* $\mathbf{Y} = \{\mathbf{y}_1, \ldots, \mathbf{y}_N\}$, in off-line. It involves the following joint optimization problem,

$$\hat{\mathbf{D}} = \arg\min_{\mathbf{D}} \frac{1}{N} \sum_{i=1}^{N} \left(\frac{1}{2}\|\mathbf{y}_i - \mathbf{D}\alpha_i\|_2^2 + \lambda\|\alpha_i\|_1 \right), \quad (3)$$

where the energy of any dictionary atom $\mathbf{d}_j \in \mathbb{R}^m, j \in \{1, k\}$ is constrained to $\mathbf{d}_j^T \mathbf{d}_j \leq 1$. One can solve for Eq. 3 by iteratively updating $\mathbf{D}$ and α_i, holding one fixed while updating the other variable [12]. In this work, we employed the online dictionary learning (ODL) solver [12] because of its time-efficient and memory-friendly mini-batch mechanism, which learns the dictionary incrementally by using a part of the training corpus in each update. The open-source package SPAMS (`http://spams-devel.gforge.inria.fr/`) is employed for its ODL implementation.

2.2. Source-filter modeling perspective

When log scale is applied to the spectral representation as in Eq. 1, the transformation is a homomorphic one that maps convolution to addition [1], which facilitates the separation of the source and filter components. A number of cepstrum-based source-filter separation techniques has been proposed in the literature [5, 6]. Such technique usually estimates the transfer function of the filter as a white excitation source pass through a filter. Consequently, the spectral envelope is almost independent of pitch.

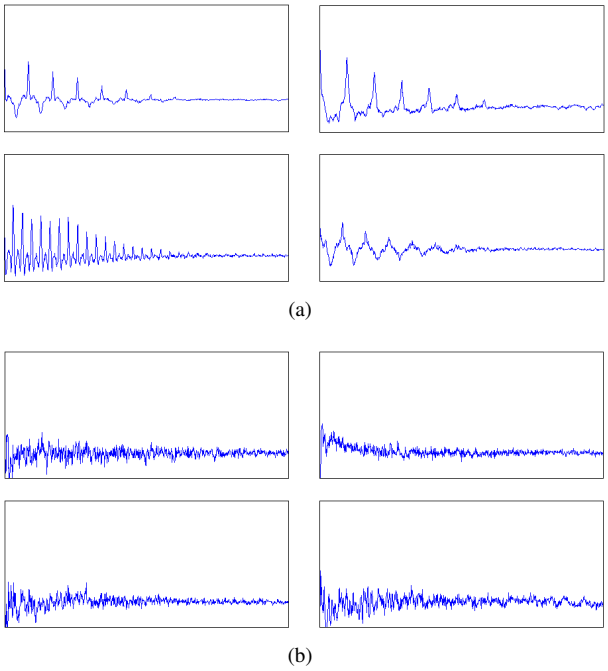

Fig. 1. The (a) first and (b) second type of most frequently used codewords (i.e., dictionary atoms) in the sparse representation of the cepstra of violin signals.

We found that sparse cepstral codes inherit the strength of the cepstrum in separating pitch and timbre. Fig. 1 show the atoms $\mathbf{d}_j$ that are most frequently used for encoding violin signals, whose spectrum is known to be well modeled by the multiplication of harmonic series (the excitation source) and the overall frequency response contributed by the violin resonator and string. We found that the atoms can be distinctively divided into two types: the first type features an apparent harmony structure and, whereas the second type has no harmony component. The first type appears to be strongly related to the pitch component contributed by the excitation source, whereas the second type characterizes the timbre component contributed by the resonator and string.

Nevertheless, the source and filter components might still overlap in the cepstrum. Learning a dictionary of cepstrum codewords helps resolve this issue by using different atoms to represent the two components. Since the filter component is more related to the discrimination of different instrument timbres, using sparse cepstral codes as input to SVM might lead to better accuracy in instrument identification, comparing to the raw cepstrum or spectrum. This is supported by our empirical study reported in Section 4.

2.3. Power normalization

In addition to using the log function, we can also apply power scales $g(\cdot)$ in computing the cepstrum. That is, Eq. 1 can be expressed in the following more general form,

$$\mathcal{C}(\mathbf{x}) = \mathcal{F}^{-1}(g(\mathcal{F}(\mathbf{x}))) \,. \tag{4}$$

We experimented with $g(\mathbf{x}) = |\mathbf{x}|^{1/2}$, $|\mathbf{x}|^{1/3}$, $|\mathbf{x}|^{1/4}$, and $|\mathbf{x}|^{1/5}$ in this study, which are referred to as $\mathcal{C}_2$ (square root), $\mathcal{C}_3$ (cubic root), $\mathcal{C}_4$, and $\mathcal{C}_5$, respectively.

The aforementioned scales change the function of the cepstrum. On one hand, these power scales no longer map multiplication to addition. Instead, the power scales retain the multiplication factor and are therefore scale-variant, a property which may be useful when the energy level of the signal carries relevant information. For example, as the predominant instrument in a multi-source signal usually has higher energy level [18], replacing the log function by a power scale in computing the cepstrum might be helpful.

On the other hand, according to the Stevens' power law [25], the magnitude of a subjective sensation (e.g., the perception of instrument timbre) increases proportionally to the power of the magnitude of physical stimulus (e.g., a music signal). Therefore, it is possible that the power-scaled variant Eq. 4 better fits human perception of instrument.

In what follows, we denote the SC of spectrum as $f(\mathcal{F})$ and that of cepstrum, including both the log-scaled and power-scaled ones, as $f(\mathcal{C})$.

3. SYSTEM OVERVIEW

Fig. 3 shows the diagram of the proposed classification system based on sparse cepstral codes. It begins with extracting frame-level features (e.g., the cepstrum) from each song. Then, it applies SC to compute the sparse representation for each frame using a dictionary, which is trained from the training corpus. The frame-level codes are pooled along the temporal dimension by sum pooling [15], leading to the song-level feature representation for the song, which is used as input for classification. The song-level feature is further normalized by Manhattan normalization (i.e., sum-to-one normalization) to account for the length of the different songs [16]. Finally, the LIBLINEAR library [13] is employed for training a linear SVM model for classification.

4. EXPERIMENT

4.1. Datasets & experimental setup

Two datasets were employed to evaluate the accuracy of identifying instrument: the ParisTech dataset [17] for uni-source signal and the Music Technology Group (MTG) dataset [18] for the multi-source case. The ParisTech dataset contains 273 solo pieces with different genes; each piece is played with

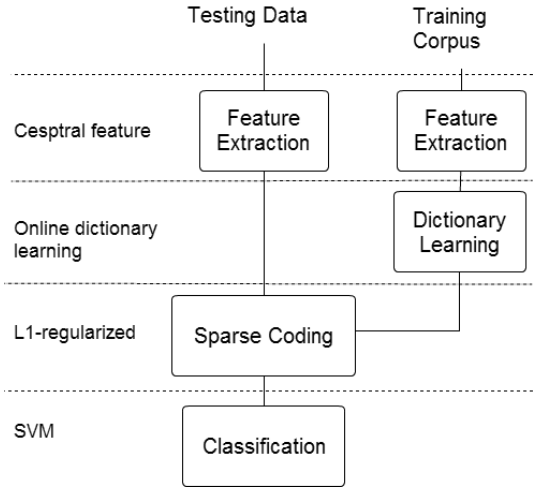

Fig. 2. Diagram of a classification system based on the proposed sparse cepstral codes.

one of the following 10 possible instruments — bassoon, double bass, clarinet, cello, flute, guitar, oboe, piano, violin, and trumpet. Each piece is 111 seconds in length on average, totaling 30,350 seconds [17]. The MTG dataset consists of about 2,500 music pieces with multiple instruments sounded at the same time. Each piece comes with a label of the predominant pitched instrument played in the piece, selecting from one of the following 11 possible instruments — cello, clarinet, flute, acoustic guitar, electric guitar, Hammond organ, piano, saxophone, trumpet, violin and singing voice. The songs are of various genres and are mostly shorter than 10 seconds, totaling 13,400 seconds [18]. For the MTG dataset we found it a non-trivial task to determine the predominant instrument even for human listeners.

As for the training corpora for dictionary learning, we used the musical instrument sound subset of the RWC dataset [26] for uni-source and the USPOP2002 dataset [27] for multi-source. The former contains complete pitch ranges of 50 instruments with different instrument manufactures, musicians, playing styles, and dynamic levels, lasting 330,000 seconds in total. The latter is a collection of nearly 8,000 contemporary pop songs from approximately 400 artists.

The low-level feature representation $\mathbf{y}$ includes the raw spectrum, the raw cepstrum, and 30-band MFCC [4], all of which are based on a Fourier transform using Hanning window with 1,025 samples and 50% overlaps. We experimented with different scaling functions for the cepstrum and tried SC on the spectrum and cepstrum. We did not perform SC on MFCC for this has been shown suboptimal in [14].

To avoid overfitting, a six-fold jack-knife cross-validation scheme was adopted. Each fold contains at least one file per instrument. Ten runs of cross-validation evaluation with random partitions were performed to get the average result. The accuracy was evaluated in terms of the classification accuracy

Table 1. Results of different feature representations, where the first row shows accuracies and the second row shows F-scores

Task	existing work		low-level feature			SC feature					
	[17]	[18]	MFCC	$\mathcal{F}$	$\mathcal{C}_{\log}$	$f(\mathcal{F})$	$f(\mathcal{C}_{\log})$	$f(\mathcal{C}_2)$	$f(\mathcal{C}_3)$	$f(\mathcal{C}_4)$	$f(\mathcal{C}_5)$
Uni-source (ParisTech)	0.820	–	0.541	0.684	0.705	0.934	0.945	0.941	0.939	0.948	0.957
Multi-source (MTG)	–	0.630	0.229	0.256	0.276	0.611	0.610	0.603	0.637	0.649	0.661

Table 2. Classification accuracies of different SC features with varying codebook size k for the uni-source dataset [17]. The last row shows the differences between $k = 1024$ and 64

k	$f(\mathcal{F})$	$f(\mathcal{C}_{\log})$	$f(\mathcal{C}_2)$	$f(\mathcal{C}_3)$	$f(\mathcal{C}_4)$	$f(\mathcal{C}_5)$
64	0.818	0.833	0.910	0.892	0.919	0.916
128	0.873	0.875	0.926	0.924	0.918	0.921
256	0.902	0.920	0.936	0.935	0.936	0.938
512	0.907	0.926	0.947	0.942	0.944	0.947
1024	0.928	0.944	0.939	0.939	0.946	0.955
$\triangle$	0.110	0.111	0.029	0.047	0.027	0.039

for the uni-source dataset according to [17] and in terms of the F-score, the harmonic mean of precision and recall, for the multi-source dataset according, following [18].

4.2. Result

We first evaluated the accuracy of instrument identification with different feature representations, with the dictionary size k used in SC fixed to 1,024, a commonly adopted size [15].

The first row of Table 1 shows the accuracies for the uni-source case. The following observations can be made. First, a comparison between low-level features and SC features clearly illustrates the effectiveness of SC in representing audio information. Second, SC on cepstrum with log scale ($f(\mathcal{C}_{\log})$) performs better than SC on spectrum ($f(\mathcal{F})$), showing that cepstrum contains relevant information of instrument timbre. Third, higher accuracy can be obtained by using higher-order power normalization (e.g., $f(\mathcal{C}_5)$), suggesting replacing the log function by power scales might be advisable. Through a two-tailed t-test we validated that the performance of $f(\mathcal{C}_5)$ (0.957) is significantly better than that of the state-of-the-art result (0.820) reported for this dataset [17]. Moreover, the performance difference between either $f(\mathcal{C}_5)$ and $f(\mathcal{F})$ or $f(\mathcal{C}_5)$ and $f(\mathcal{C}_3)$ is also significant.

The second row of Table 1 shows the F-scores for the multi-source case. We see that either $f(\mathcal{C}_{\log})$ or $f(\mathcal{C}_2)$ exhibits no advantage over $f(\mathcal{F})$. However, better result was obtained again by using higher-order power normalization. When $f(\mathcal{C}_5)$ was used, significantly better accuracy (0.661) was obtained, comparing to either the use of $f(\mathcal{F})$ or the state-of-the-art result (0.630) [18]. The improvement of using higher-order power scales appears to be more pronounced for the multi-source case, comparing to the uni-source case.

Next, we evaluate the performance of different SC features as dictionary size k varies. As mentioned in Section 1, we expected better performance can be obtained by increasing k, which in effect increases the granularity of the sparse representation [12]. From Table 2, we indeed see a positive correlation between k and the classification accuracy that was obtained. Comparing to the spectrum, the performance of the proposed cepstral codes appears to be relatively less sensitive to the dictionary size k, except for $f(\mathcal{C}_{\log})$. This result indicates that SC on power-scaled cepstrum is a competitive feature representation for instrument identification, even with low-dimensional codes (which is more efficient to compute [15]). Although we are not able to provide direct proofs in this work, it seems that power normalization enhances the robustness of instrument identification as it does for speaker identification [22, 23].

5. CONCLUSION

In this paper, we have presented a novel system that incorporates cepstral feature and sparse coding for instrument identification. The resulting feature representation, coined as the sparse cepstral codes, exhibits promising performance for two different instrument identification datasets. When high-order power normalization is applied, significant better performance in comparison to the state-of-the-art is obtained. It appears that the use of power scales enhances the robustness of an instrument identification system, an observation that has been made in speaker identification problems. Although the performance study presented here might be at best preliminary, it provides empirical evidences that the proposed sparse cepstral codes feature is a competitive representation for automatic instrument identification.

6. REFERENCES

[1] A. V. Oppenheim and R. W. Schafer, *Discrete-Time Signal Processing*, Prentice Hall, 2010.

[2] M. Müller, D. P. W. Ellis, A. Klapuri, and G. Richard, "Signal processing for music analysis," *IEEE J. Sel. Topics Signal Processing*, vol. 5, no. 6, pp. 1088–1110, 2011.

[3] A. Eronen and A. Klapuri, "Musical instrument recognition using cepstral coefficients and temporal features," in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing*, 2000, pp. 753–756.

[4] M. A. Casey, R. Veltkamp, M. Goto, M. Leman, C. Rhodes, and M. Slaney, "Content-based music information retrieval: Current directions and future challenges," *Proceedings of the IEEE*, vol. 96, no. 4, pp. 668–696, 2008.

[5] T. Galas and X. Rodet, "An improved cepstral method for deconvolution of source filter systems with discrete spectra: Application to musical sound signals," in *Proc. Int. Computer Music Conf.*, 1990, pp. 82–84.

[6] A. Röbel, "Efficient spectral envelope estimation and its application to pitch shifting and envelope preservation," in *Proc. Int. Conf. Digital Audio Effects*, 2005, pp. 30–35.

[7] S. B. Davis and P. Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," *IEEE Trans. Acoustics, Speech & Signal Processing*, vol. 28, no. 4, pp. 357–366, 1980.

[8] A. Diment, P. Rajan, T. Heittola, and Tuomas Virtanen, "Modified group delay feature for musical instrument recognition," in *Proc. Int. Symp. Computer Music Multidisciplinary Research*, 2013, pp. 431–438.

[9] E. J. Humphrey, J. P. Bello, and Y. LeCun, "Moving beyond feature design: Deep architectures and automatic feature learning in music informatics," in *Proc. Int. Society of Music Information Retrieval*, 2012, pp. 403–408.

[10] B. Efron, T. Hastie, L. Johnstone, and R. Tibshirani, "Least angle regression," *Annals of Statistics*, vol. 32, pp. 407–499, 2004.

[11] R. Tibshirani, "Regression shrinkage and selection via the lasso," *J. Royal Statistical Soc.*, vol. 58, pp. 267–288, 1996.

[12] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, "Online dictionary learning for sparse coding," in *Proc. Int. Conf. Machine Learning*, 2009, pp. 689–696.

[13] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, "LIBLINEAR: A library for large linear classification," *J. Machine Learning Research*, vol. 9, pp. 1871–1874, 2008.

[14] C.-C. M. Yeh and H. Yang Y, "Supervised dictionary learning for music genre classification," in *Proc. ACM Int. Conf. Multimedia Retrieval*, 2012.

[15] L. Su, C.-C. M. Yeh, J.-Y. Liu, J.-C. Wang, and Y.-H. Yang, "A systematic evaluation of the bag-of-frames representation for music information retrieval," *IEEE Trans. Multimedia*, 2013.

[16] Chin-Chia Michael Yeh, Li Su, and Yi-Hsuan Yang, "Dual-layer bag-of-frames model for music genre classification," .

[17] C. Joder, S. Essid, and G. Richard, "Temporal integration for audio classification with application to musical instrument classification," *IEEE Trans. Audio, Speech & Language Processing*, vol. 17, no. 1, pp. 174–186, 2009.

[18] F. Ferdinand, *Automatic musical instrument recognition from polyphonic music audio signals*, Ph.D. thesis, Universitat Pompeu Fabra, 2012.

[19] A. P. Glennon E. J. Humphrey and J. P. Bello, "Non-linear semantic embedding for organizing large instrument sample libraries," in *Int. Conf. Machine Learning and Applications*, 2011, pp. 142–147.

[20] J. J. Bosch, J. Janer, F. Fuhrmann, and P. Herrera, "A comparison of sound segregation techniques for predominant instrument recognition in musical audio signals," in *Proc. Int. Society of Music Information Retrieval*, 2012, pp. 559–564.

[21] H. Jégou, F. Perronnin, M. Douze, J. Sánchez, P. Pérez, and C. Schmid, "Aggregating local image descriptors into compact codes," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2012.

[22] Y. Shao X. Zhao and D. Wang, "CASA-based robust speaker identification," *IEEE Trans. Audio, Speech & Language Processing*, vol. 20, no. 5, pp. 1608–1616, 2012.

[23] X. Zhao and D. Wang, "Analyzing noise robustness of mfcc and gfcc features in speaker identification," in *Proc. IEEE. Int. Conf. Acoustics, Speech & Signal Processing*, 2013, pp. 7204–7207.

[24] A. Y. Yang, Z. Zhou, A. Ganesh, S. Shankar Sastry, and Y. Ma, "Fast L1-Minimization Algorithms For Robust Face Recognition," *ArXiv e-prints*, 2010.

[25] J. J. Zwislocki, *Sensory Neuroscience: Four Laws of Psychophysics*, Springer US, 2009.

[26] M. Goto, H. Hashiguchi, T. Nishimura, and R. Oka, "RWC music database: Music genre database and musical instrument sound database," in *Proc. Int. Society of Music Information Retrieval*, 2003, pp. 229–230.

[27] A. Berenzweig, B. Logan, D. P. W. Ellis, and B. Whitman, "A large-scale evaluation of acoustic and subjective music similarity measures," in *Computer Music J.*, 2004, vol. 28, pp. 63–76.