DUAL-LAYER BAG-OF-FRAMES MODEL FOR MUSIC GENRE CLASSIFICATION

Chin-Chia Michael Yeh, Li Su, and Yi-Hsuan Yang

Research Center for Information Technology Innovation,
Academia Sinica, Taiwan.
chiadnaoh@gmail.com, lisu@citi.sinica.edu.tw, yang@citi.sinica.edu.tw

ABSTRACT

This paper concerns the development of a music dictionary-based model for summarizing local feature descriptors computed over time. Comparing to a holistic representation, this text-like, bag-of-frames representation better captures the rich and time-varying information of music. However, the dictionary used in classical bag-of-frames model only captures frame-level elements of the music; thus, there exists a semantic gap between the dictionary element and commonly seen music description. In order to reduce the gap, a new feature representation called *dual-layer bag-of-frames* is proposed in this paper. It models the music with a two layer structure, where the first-layer dictionary captures the frame-level characteristics, and the second-layer dictionary captures the segment-level semantics. This hierarchical structure resembles the alphabet-word-document structure of text. Our result demonstrates that the proposed dual-layer bag-of-frames feature achieves state-of-the-art accuracy of music genre classification. The classification accuracy for the GTZAN benchmark reaches 86.7% with dictionary trained from GTZAN, and 83.6% with dictionary trained from another data set USPOP.

Index Terms— Sparse coding, deep structure, audio alphabets, audio words, music genre classification

1. INTRODUCTION

Over the recent years, the "bag-of-frames" (BoF) model has been shown useful in a variety of music information retrieval (MIR) tasks [1–6], owing to its ability to represent music information that happens in a short temporal moment (e.g., "guitar solo") [2]. It preserves the information by representing each music piece as a histogram over a dictionary of music "elements," or codewords, selected or learnt from a music collection [7, 8].

However, in the classical BoF model, the dictionary is trained from frame-based features/representations such as MFCC, sonograme, or spectrogram,[1] which capture the characteristics of the sound only at *frame-level* [9]. Accordingly, such frame-level dictionary elements might be unable to directly represent some of the music descriptions (e.g., "metal", "blast beat") due to its lack of long term information. This can be viewed analogously as trying to capture the semantic of a text document with a dictionary containing only letters.

In order to close the gap and produce more meaningful dictionary elements, we propose a new BoF-based feature, called *dual-layer BoF* (DLBoF), which attempts to capture the *segment-level* semantics of music signals by a two layer structure. The first-layer dictionary captures the frame-level music characteristics, and the second-layer dictionary captures the combinations of frame-level music characteristics. Similar to the alphabet-word relation in text document, if we define the frame-level music characteristic as *audio-alphabet*, the second-layer dictionary contains a list of combinations of audio-alphabets, giving rise to the notion of *audio-words*. Comparing to the conventional single-layer, flat structure, this *deep* architecture better represents the information of music [10]. Evaluation on the a genre classification benchmark data set shows that fusing the audio-word based BoF with the audio-alphabet based BoF (i.e., DLBoF) leads to considerable improvement in the accuracy of genre classification, comparing to using the audio-alphabet based BoF alone.

Such a multi-layer structure is conceptually similar to deep belief net (DBN), where each layer of DBN extracts salient information at different timescales [10–13], and related to multi-scale temporal fusion, where the frame-based features/representations are pooled at different time scale and then fused together [14, 15]. However, we opt for the sparse-coding (SC) based approach for one can combine any dictionary learning method with sparse coding [16].

2. RELATED WORK

A number of algorithms have been proposed for implementing BoF models for music feature extraction. For instance, McFee *et al*. [4] employed *k*means to cluster a collection of frame-level MFCC vectors and used the cluster centers for vector quantization (VQ). The histogram representation of a song is constructed by counting the frequency with which each dictionary element quantizes the bag of MFCC vectors of that song. Yeh *et al.* [7] performed a systematic evaluation on various BoF-related algorithms, and found that coupling log-power spectrogram with sparse coding and online dictionary learning (ODL) [17] shows relatively better result. Based on their finding, the proposed DLBoF model is implemented by referring to their best setup, which consists of using spectrogram as the local descriptor, sparse coding as the encoding algorithm, and ODL for dictionary learning.

Sparse coding (SC) algorithms have also been utilized for constructing the codebook for music [18–24]. SC seeks to mimic the human's sensory system by forming codes that are sparse in support (with most coefficients being zero), yet contain sufficient information to reconstruct or to interpret the input signals. People are interested in sparse representations or sparse models, because they lead to codewords that are "neurally plausible," or that can be explained [21]. As first demonstrated by Smith *et al.* [25], audio codewords learnt by using the matching pursuit (MP) algorithm for sparse decomposition show striking similarities to time-domain cochlear filter estimates. Therefore, in addition to the discriminative power, SC leads to codewords that are higher in interpretability.

[1] The frame size setting for these features/representations is usually 23–92 ms for MIR appreciations [9].

To the best of our knowledge, this work represents one of the first works that learns multiple layers of codebooks for music signals using online dictionary learning [17]. In addition, the notions of frame-level audio alphabets and segment-level audio words have not been proposed before.

3. DICTIONARY-BASED FRAME WROK

The dictionary-based classification system requires a set of labeled songs for classifier training, and a set of unlabeled songs for dictionary training. It uses the unlabeled set to train the dictionary; then uses the dictionary on the labeled set to generate BoF-based features. After that, the extracted feature can be used with the label information to train a classifier, and the resultant classifier can be tested on a testing set to measure the performance. The detail of each system component is described below.

3.1. Sparse Coding and Dictionary Learning

Given an input signal vector $x \in \mathbb{R}^m$, the sparse representation problem can be mathematically formulated as

$$\alpha^* = \underset{\alpha}{\operatorname{argmin}} \frac{1}{2}\|x - D\alpha\|_2^2 + \lambda\|\alpha\|_1, \tag{1}$$

where $\alpha \in \mathbb{R}^k$ is the sparse coding of x, $D \in \mathbb{R}^{m\times k}$ is a given dictionary, and λ is a turning parameter for the trade-off between α's sparsity and the representation accuracy. Typically λ is set to $1/\sqrt{m}$ for that is the classical normalization factor [17], where m is the feature dimension of x. This problem is usually referred to as basis pursuit or Lasso in the machine learning and statistics literature [26]. It can be solved efficiently by off-the-shelf programs such as LARS-lasso [27].

It has been shown that using a learnt dictionary instead of a predefined one improves the performance of sparse coding as the learnt one is more adapt to the data being processed [17]. The dictionary learning problem can be formulated as

$$D^* = \underset{D\in\mathcal{C}}{\operatorname{argmin}} \frac{1}{n}\sum_{i=1}^{n}\left(\frac{1}{2}\|x_i - D\alpha_i\|_2^2 + \lambda\|\alpha_i\|_1\right), \tag{2}$$

where x_i denotes the i-th signal among a dataset of n signals, and $\mathcal{C} \triangleq \{D \in \mathbb{R}^{m\times k}\}$ is a set of convex matrices in which the l_2 norm of each column d_j is not larger than one, i.e., $d_j^T d_j \leq 1, \forall j$. This constraint is imposed to constrain the energy of the dictionary elements. The formulation in Eq. 2 is a joint optimization problem in α and D, and a natural solution is to optimize the two variables in an alternating fashion.

In this work, we employ the first-order stochastic gradient descent algorithm called online dictionary learning (ODL) [17] to solve this joint optimization problem. ODL is known to be more scalable than standard second-order batch algorithms for its relatively lower computational cost, memory consumption, and capability of learning in an online, rather than batch, fashion.

3.2. Single-layer Bag-of-Frames Model

The system diagram of the classical BoF model encoding is shown in Fig. 1. We use the term *single-layer BoF* (SLBoF) to describe this model oppose to the proposed DLBoF model.

For the dictionary training process, all the training song is first converted into spectrograms. The frame size and overlap for Fast Fourier transform are 1,024 and 50% respectively. Then, the dictionary is trained with ODL from the extracted spectrograms (cf. Section 3.1). Once we have the dictionary, any given digital music can be encoded by first converting it into spectrogram. Then, the sparse coding of the music is calculated from the spectrogram in a frame-by-frame fashion. Next, the individual frame-level sparse coding is aggregated together (BoF aggregation) to form a histogram representation (cf. Section 3.5). Finally, the histogram is first normalized with power normalization, then Manhattan normalization (i.e., sum-to-one normalization); the result vector is the SLBoF of the input digital music.

3.3. Dual-layer Bag-of-Frames Model

As shown in Fig. 1, after the first-layer dictionary is constructed, the first-layer dictionary is used to encode all the training songs with sparse coding. The resultant sparse coding is aggregated with *bag-of-histogram aggregation* (BoH aggregation), converting the frame-level sparse coding into segment-level histogram representation. In consequence, each training song is represented with multiple histograms. Next, each histogram is power normalized, and used to train the second-layer dictionary with ODL. In the end, we have a first-layer dictionary trained from spectrograms and a second-layer dictionary trained from histograms.

With both dictionaries trained, the encoding process starts with converting the given digital music's spectrogram to first-layer sparse coding with the first-layer dictionary. After BoH aggregation and power normalization, the resultant vectors are converted to second-layer sparse coding with the second-layer dictionary. Lastly, the sparse coding from both layers are aggregated and concatenated to form a histogram representation, which is normalized with power normalization and Manhattan normalization again, leading to the DLBoF representation of the input digital music.

3.4. Power Normalization

Given an input feature vector $w \in \Re^k$, the power normalization can be calculated with

$$w^* = \operatorname{sign}(w)|w|^a, \tag{3}$$

where $\operatorname{sign}(\cdot)$ is the sign function and $a \in [0, 1]$ is a pre-set parameter, and Jégou *et al.* [28] has empirically determined that $a = 0.5$ constantly leads to near-optimal results. Such transformation has been shown to increase the performance for a BoF based image search system, due to its ability to reduce the influence of bursty visual elements. It can also be interpreted as variance stabilizing transform, which corrects the dependence between the variance and the mean. It has been applied to BoF, GMM, and Fisher vector; these power normalized feature vectors yield improved performances comparing to their original version [28].

3.5. Bag-of-Histogram Aggregation

It has been found that partitioning a song into short segments, each span a number of frames, and generate feature based on the segmentation usually improves the classification accuracy [29, 30]. These segments are called "texture windows" by Tzanetakis *et al.* as it should correspond to the minimum time amount of music that is necessary to identify a particular music's timbre, pitch, and loudness [29]. To capture the change of music texture, we aggregate the first-layer sparse coding over a texture window and represent a song as a *bag-of-histograms*.

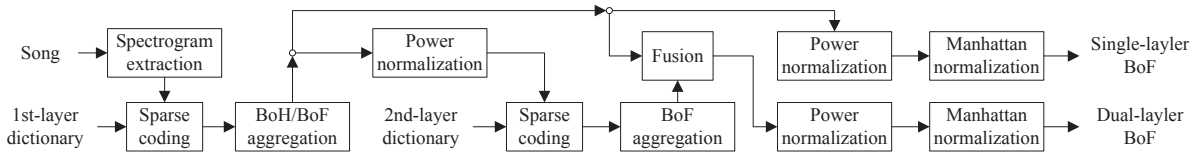

Fig. 1. The system diagrams of the encoding for bag-of-frames models.

In addition to using a fixed-length texture window for segmentation, another segmentation method we adopted for BoH aggregation is the MIRsegment algorithm implemented in the MIRtoolbox [31]. Specifically, we compute the similarity of the low-level feature vectors of every two frames in a song to construct a similarity matrix, from which the so-called "checkerboard" patterns can be observed from the main diagonal if there are segmental boundaries. The two segments beside the boundary produce two adjacent square regions of high within-segment similarity along the main diagonal and two square regions of low between-segment similarity off the main diagonal. To detect such pattern, a Gaussian-tapered checkerboard kernel is correlated along the main diagonal of the similarity matrix, and the so-called novelty curve can be calculated [32]. Since the peaks of the novelty curve indicate points of highly dissimilarity adjacent segments, these peaks can be identified as possible boundaries. Typically, the segment size is between 2–10 seconds.

4. EXPERIMENTS

We evaluate the performances of the BoF-based features on music genre classification, one of the most well studied problems in MIR. GTZAN is the most used benchmark dataset; it is composed of 1,000 30-second clips covering ten genres, with 100 clips per genre [29].[2] Each song is converted to a standard mono-channel and 22,050 Hz sampling rate WAV format before experiments, a common practice in MIR. Evaluation on GTZAN is typically conducted using a stratified 10-fold cross validation. The performance is measured in terms of the (average) classification accuracy.

However, GTZAN is problematic when used as an exemplary dataset for testing music genre classification systems. The three problems brought up by Sturm *et al.* [33] are repetition, mislabeling, and distortion. Additionally, it is debatable on whether genre recognition is a well-defined problem since the definitions of certain genres varied as the society changes. As a 10-year-old data set, 10.6% of GTZAN clips are mislabeled based on contemporary standards. Consequently, it may not be an ideal data set for studying music genre recognition, but a handy preliminary test bed for comparing a newly proposed method with existed methods because all the MIR systems tested on GTZAN had to face the same problems. As the main goal of this work is to perform an initial study on the proposed BoF feature, we use GTZAN for performance evaluation. In order to better assess the new feature, conducting experiments on other MIR tasks is in order.

4.1. Single-layer Model, Normalization, Kernel & Efficiency

We first examine the effectiveness of power normalization on GTZAN. The feature vector is generated based on the SLBoF model describe in Section 3.2. The dictionary is trained with GTZAN, and the dictionary size is set to 1,024. The experiments is performed with both linear and histogram intersection kernel (HIK) support vector machine (SVM). HIK is a kernel designed specifically for histograms features (e.g., bag-of-frames) [34]; we use the implementation from Maji *et al.* [34]. For linear SVM, we use two different implementations, LIBSVM and LIBLINEAR [35, 36]. LIBSVM uses a more general way to solve the SVM optimization problem, which works for both linear and non-linear kernels. In contrast, LIBLINEAR utilizes a more efficient solver design specifically for linear kernel. Therefore, LIBLINEAR is a more efficient linear SVM implementation comparing to LIBSVM. For LIBLINEAR, we use two different settings; one solves the primal formulation (Primal), and the other solves the dual formulation (Dual.) Although the two formulations are mathematically equivalent [36], we found the dual form is more efficient for our problem, as Table 1 shows. Overall, using power normalization yields an improvement in both average accuracy and speed, and the best result for the power normalized SLBoF is 82.2%, which outperforms conventional VQ and is on par with the state-of-the-art results for GTZAN genre classification [7, 37].

For HIK SVM, power normalization improves the classification accuracy, but the increment is subtle (1.1%). In contrast, for linear SVM, the effect of power normalization is more relevant (6.43%). Power normalization improves the discriminant power of BoF more

Table 1. The effect of power normalization on the genre classification accuracy; where Pow stand for power normalization.

	Pow		**No Pow**	
	Acc	**Time**	**Acc**	**Time**
LIBSVM+HIK	82.2	119.6	81.1	146.2
LIBSVM+Linear	77.8	67.5	71.9	76.0
LIBLINEAR(Primal)	78.3	19.7	71.7	31.8
LIBLINEAR(Dual)	78.4	6.8	71.6	16.6

Table 2. Performance of dual-layer bag-of-frames model comparing to other models under different experiment settings. The bold font face indicates that the setup is significantly better than its first-layer counterpart (p-value < 0.05).

		Linear SVM			**HIK SVM**		
		2.5s	**5s**	**MIR**	**2.5s**	**5s**	**MIR**
GTZ	**First**	76.8			82.7		
	Second	70.7	66.6	72.7	81.0	79.2	82.4
	Dual	79.2	78.1	78.6	**85.2**	84.4	**85.7**
	Late	77.8	78.1	77.5	84.1	83.8	**85.5**
USP	**First**	81.5			83.3		
	Second	74.2	70.1	77.3	73.8	69.3	76.2
	Dual	82.2	81.6	81.7	81.9	82.3	82.7
	Late	80.7	78.4	80.7	81.5	79.3	81.4

[2]The datasets are available at `http://opihi.cs.uvic.ca/sound/genres.tar.gz`. The genre classes contain classical, country, disco, hiphop, jazz, rock, blues, reggae, pop, and metal.

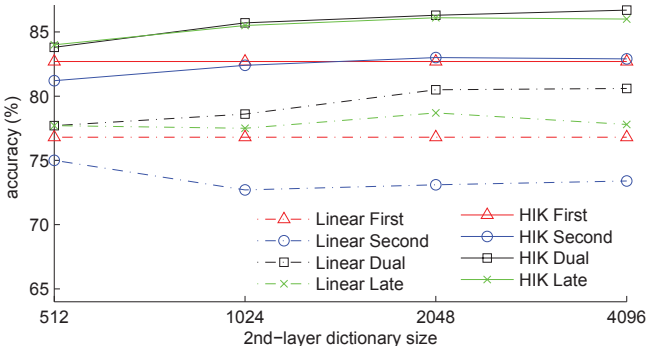

Fig. 2. Classification accuracy as we vary the second-layer dictionary size for different bag-of-frames models with linear kernel or histogram intersection kernel.

in the linear space than in the non-linear space. Although LIBSVM + HIK leads to better results, it requires roughly 17 times more process time than LIBLINEAR (Dual). LIBLINEAR considerably boosts the speed of training and testing process.

Although LIBLINEAR with dual formulation has better performance comparing to primal formulation, the decision about which formulation to use depends on the size of dictionary (dimension of the feature vector) and number of training instances. According to Fan *et al.* [36], primal is more efficient than dual when # of instances $\gg$ feature dimension, and vice versa when # of instances $\ll$ feature dimension. As a result, with power normalized BoF, LIBLINEAR (Primal) is a more viable solution in a large scale setting.

4.2. Dual-layer Model, Segmentation & Transductive vs. Inductive Learning

Next, we evaluate the performance of the proposed DLBoF model under $2 \times 2 \times 3 = 12$ different setups on GTZAN genre classification. These setups are designed form varying dictionary training set, classifier, and segmentation method. The two dictionary training sets we used in this experiment are GTZAN (**GTZ**) and USPOP (**USP**). USPOP is a data set consisting of 8,764 tracks from 400 manually selected popular artists [38]. While the use of GTZAN for dictionary learning adopts the "transductive learning" scenario (assuming that the test set of the target task is available during feature learning), the use of USPOP increases the inductive power of the experiment results. The two classifiers we used are HIK SVM and linear SVM, and the three segmentation methods considered are fixed segments of 2.5 seconds with 50% hop (**2.5s**), fixed segments of 5 seconds with 50% hop (**5s**), and variable-length segmentation using MIRsegment (**MIR**) [31]. The size of the dictionaries is set to 1,024. The four BoF related models we tested include the BoF constructed by first-layer sparse coding (**First**-layer BoF), BoF constructed by second-layer sparse coding (**Second**-layer BoF), the proposed DLBoF (**Dual**), and the late fusion alternative (**Late**) that builds two independent classifiers for the first and second-layer BoF and makes prediction based on the average estimated probability of the two classifiers.

Table 2 shows that using the two-layer structure improves the performance in most cases. When the dictionary is learned from GTZAN and the music is segmented using MIRsegment, the classification accuracy reaches 85.7%. The performance difference is significant (p-value < 0.05) under the two-tailed t-test. We also see that using a fixed-length window (e.g., 2.5 sec) is also competitive (85.2% for Dual). Using second-layer BoF alone does not lead to performance improvement, possibly because frame-level information is more important for genre classification. Moreover, we see that using USPOP for learning the dictionary leads to higher accuracy for the single-layer (First) setting but lower for the dual-layer setting, suggesting that for such a deep structure it may be preferable to adopt a transductive learning setting, as usually done in the literature (e.g., [12, 23]).

As for large-scale processing, linear kernel is usually preferable. Interestingly, we note that the performance difference between linear SVM and HIK SVM is smaller when USPOP is used for dictionary training. This suggests that, when the dictionary training set and the classifier training set are independent and the target data set is large, cooperating DLBoF with linear SVM is an attractive setup. In addition, although MIRsegment always performs the best among the three tested segmentation methods, it is computational more demanding comparing to fixed segmentation. To deal with a large-scale data set, using fix segmentation is a sound option.

4.3. Dictionary Size & Fusion

According to Yeh *et al.*, setting the first-layer dictionary size to 1,024 leads to near-optimal result [7]. However, whether 1,024 is optimal for second-layer dictionary deserves investigation. To this end, we perform another experiment that fixes the size of the first-layer dictionary to 1,024 but varies the size of the second-layer dictionary. For this evaluation, the dictionaries are trained from GTZAN, and the segmentation is done by MIRsegment. The result is shown in Fig. 2. Overall, the accuracy of fusion increases as the second-layer dictionary size grows.

For linear SVM, the increment of DLBoF's performance saturates about 2,048, and there seems to be no clear relationship between DLBoF and second-layer BoF. In addition, second-layer BoF is always worse than the other features. For HIK, the performance seems to continue improving slightly even when the dictionary size reaches 4,096. Nevertheless, in view of computational complexity, it seems feasible to set the dictionary size simply to 1,024.

Finally, we note that, while late fusion weights both layers equally, the proposed DLBoF adopts an early fusion scheme and does not assume equal weight between the two layers. It turns out that early fusion consistently outperforms its late fusion counterpart in our evaluation, showing that it is better not to assign equal weights to the two layers.

5. CONCLUSIONS

In this paper, we have proposed the DLBoF model, which improves conventional BoF model by considering both frame-level music characteristics and segment-level music semantics. We have shown that the proposed DLBoF is effective and efficient, and it obtains 86.7% accuracy for music genre classification on the GTZAN data set. The result is highly competitive in comparison to the state-of-the-art. We have also shown the effectiveness of power normalization, and the advantages gained from using linear SVM instead of non-linear SVM. The two-layer structure can be easily implemented by cascading two layers of dictionary learning and is readily applicable to other MIR or audio classification problems.

6. ACKNOWLEDGMENT

This work was supported by the National Science Council of Taiwan under Grant NSC 101-2221-E-001-017 and the Academia Sinica Career Development Program.

7. REFERENCES

[1] M. Casey and M. Slaney, "The importance of sequences in musical similarity," in *ICASSP*, 2006, pp. 14–19.

[2] M. I. Mandel, D. Eck, and Y. Bengio, "Learning tags that vary within a song," in *ISMIR*, 2010, pp. 399–404.

[3] J.-C. Wang, H.-S. Lee, H.-M. Wang, and S.-K. Jeng, "Learning the similarity of audio music in bag-of-frames representation from tagged music data," in *ISMIR*, 2011.

[4] B. McFee, L. Barrington, and G. R. G. Lanckriet, "Learning content similarity for music recommendation," *IEEE Trans. Audio, Speech and Lang. Processing*, vol. 20, no. 8, 2012.

[5] J. Wülfing and M. Riedmiller, "Unsupervised learning of local features for music classification," in *ISMIR*, 2012, pp. 139–144.

[6] J.-Y. Liu, C.-C. M. Yeh, Y.-C. Teng, and Y.-H. Yang, "Bilingual analysis of song lyrics and audio words," in *ACM Multimedia*, 2012, pp. 829–832.

[7] C.-C. M. Yeh and Y.-H. Yang, "Supervised dictionary learning for music genre classification," in *ACM ICMR*, 2012.

[8] I. Tosic and P. Frossard, "Dictionary learning: What is the right representation for my signal?," *IEEE Signal Processing Magazine*, vol. 28, no. 2, pp. 27–38, 2011.

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

[10] E. J. Humphrey, J. P. Bello, and Y. LeCun, "Deep architectures and automatic feature learning in music informatics," in *ISMIR*, 2012, pp. 403–408.

[11] H. Lee, Y. Largman, P. Pham, and A. Y. Ng, "Unsupervised feature learning for audio classification using convolutional deep belief networks," in *NIPS*, 2009, pp. 1096–1104.

[12] P. Hamel and D. Eck, "Learning features from music audio with deep belief networks," in *ISMIR*, 2010, pp. 339–344.

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

[14] P.-S. Huang, J. Yang, M. Hasegawa-Johnson, F. Liang, and T. S. Huang, "Pooling robust shift-invariant sparse representations of acoustic signals," in *Interspeech*, 2012.

[15] R. Foucard, S. Essid, M. Lagrange, and G. Richard, "Multi-scale temporal fusion by boosting for music classification," in *ISMIR*, 2011, pp. 663–668.

[16] F. Bach, R. Jenatton, J. Mairal, and G. Obozinski, "Optimization with sparsity-inducing penalties," *Foundations and Trends in Machine Learning*, vol. 4, pp. 1–106, 2012.

[17] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, "Online dictionary learning for sparse coding," in *ICML*, 2009, pp. 689–696.

[18] M. D. Plumbley, T. Blumensath, L. Daudet, R. Gribonval, and M. E. Davies, "Sparse representations in audio and music: from coding to source separation," *Proceedings of the IEEE*, vol. 98, no. 6, pp. 995–1005, 2010.

[19] S. Scholler and H. Purwins, "Sparse approximations for drum sound classification," *IEEE J. Sel. Topics Signal Processing*, vol. 5, no. 50, pp. 933–940, 2011.

[20] M. Henaff, K. Jarrett, K. Kavukcuoglu, and Y. LeCun, "Unsupervised learning of sparse features for scalable audio classification," in *ISMIR*, 2011, pp. 681–686.

[21] P. Depalle C. Kereliuk, "Sparse atomic modeling of audio: A review," in *Int. Conf. Digital Audio Effects*, 2011, pp. 81–92.

[22] C.-T. Lee, Y.-H. Yang, and H. H. Chen, "Multipitch estimation of piano music by exemplar-based sparse representation," *IEEE TMM*, vol. 14, pp. 608–618, 2012.

[23] J. Nam, J. Herrera, M. Slaney, and J. Smith, "Learning sparse feature representations for music annotation and retrieval," in *ISMIR*, 2012, pp. 565–560.

[24] Y.-H. Yang, "Towards real-time music auto-tagging using sparse features," in *IEEE ICME*, 2013, to be published.

[25] E. C. Smith and M. S. Lewicki, "Efficient auditory coding," *Nature*, vol. 439, no. 7079, pp. 978–982, 2006.

[26] S. S. Chen, D. L. Donoho, Michael, and A. Saunders, "Atomic decomposition by basis pursuit," *SIAM J. Scientific Computing*, vol. 20, pp. 33–61, 1998.

[27] B. Efron, T. Hastie, L. Johnstone, and R. Tibshirani, "Least angle regression," *Annals of Statistics*, 2004.

[28] H. Jégou, F. Perronnin, M. Douze, J. Sánchez, P. Pérez, and C. Schmid, "Aggregating local image descriptors into compact codes," *IEEE TPAMI*, 2012.

[29] G. Tzanetakis and P. Cook, "Musical genre classification of audio signals," *IEEE Trans. Speech and Audio Processing*, 2002.

[30] J. Bergstra and B. Kegl, "Aggregate features and adaboost for music classification," in *Machine Learning*, 2006, vol. 65, pp. 473–484.

[31] O. Lartillot and P. Toiviainen, "MIR in Matlab (II): A toolbox for musical feature extraction from audio," in *ISMIR*, 2007.

[32] J. Foote, "Automatic audio segmentation using a measure of audio novelty," in *IEEE ICME*, 2000, pp. 452–455.

[33] B. L. Sturm, "An analysis of the GTZAN music genre dataset," in *ACM Workshop on Music information retrieval with user-centered and multimodal strategies*, 2012, pp. 7–12.

[34] S. Maji, A. C. Berg, and J. Malik, "Efficient classification for additive kernel svms," *IEEE TPAMI*, 2013.

[35] C.-C. Chang and C.-J. Lin, *LIBSVM: A library for support vector machines*, 2001.

[36] 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*, 2008.

[37] B. L. Sturm, "A survey of evaluation in music genre recognition," in *Adaptive Multimedia Retrieval*, 2012.

[38] 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 Journal*, 2003.