IMPROVING MUSIC AUTO-TAGGING BY INTRA-SONG INSTANCE BAGGING

Chin-Chia Michael Yeh[†] *Ju-Chiang Wang*[‡] *Yi-Hsuan Yang*[†] *Hsin-Min Wang*[‡]

[†] Research Center for Information Technology Innovation, Academia Sinica, Taiwan.
[‡] Institute of Information Science, Academia Sinica, Taiwan.
Emails: {mcyeh@citi, asriver@iis, yang@citi, whm@iis}.sinica.edu.tw

ABSTRACT

Bagging is one the most classic ensemble learning techniques in the machine learning literature. The idea is to generate multiple subsets of the training data via bootstrapping (random sampling with replacement), and then aggregate the output of the models trained from each subset via voting or averaging. As music is a temporal signal, we propose and study two bagging methods in this paper: the *inter-song instance bagging* that bootstraps song-level features, and the *intra-song instance bagging* that draws bootstrapping samples directly from short-time features for each training song. In particular, we focus on the latter method, as it better exploits the temporal information of music signals. The bagging methods result in surprisingly effective models for music auto-tagging: incorporating the idea to a simple linear support vector machine (SVM) based system yields accuracies that are comparable or even superior to state-of-the-art, possibly more sophisticated methods for three different datasets. As the bagging method is a meta algorithm, it holds the promise of improving other MIR systems.

***Index Terms*—** Bagging, ensemble classification, sparse coding, feature pooling, music auto-tagging

1. INTRODUCTION

As the number of digital music continues to grow, recent years have witnessed growing interest in developing auto-tagging systems that label a song according to the music content [1–9]. The goal of such systems is to help users search for music using tags such as genre, emotion, or instrumentation, while reducing the effort in manually labeling music. With music tags, users can query music via natural language, tag list, or tag cloud [10–13].

Due to the so-called semantic gap between audio signals and music tags, music auto-tagging is still considered challenging [6]. In this paper, we propose a novel ensemble learning algorithm called *intra-song instance bagging* for music auto-tagging. Our strategy is to improve the capability of classification by incorporating the idea of bootstrap aggregating (bagging) [14] and adapt it to handle music signals. Typically, ensemble classification will lead to better performance if certain diversity can be taken into account by the multiple base classifiers [15]. As opposed to the "inter-song" counterpart that aggregates models trained from song-level features, intra-song instance bagging diversifies the feature samples by bootstrapping short-time instances from each training song. While we expect that the intra-song method better capitalizes local musical information since information loss inherent to song-level features is avoided, both the inter- and intra- song methods inherit the advantages of bagging [14] in improving the stability of machine learning models and in reducing the risk of overfitting. According to our performance study, the intra-song method outperforms the inter-song one and leads to stat-of-the-art accuracies on three auto-tagging datasets, encompassing CAL500, MajorMiner and CAL10k [16–18]. Despite the bagging idea is relatively simple, it is easy to implement and constitutes a competitive alternative in building MIR models.

The paper is organized as follows. We briefly describe some related work in Section 2 and then present the proposed method in Section 3. We report a comprehensive performance study covering the three datasets in Section 5 and conclude in Section 6.

2. RELATED WORK

Ensemble classification has been adopted and shown effective in many MIR systems. For instance, the winning solution [19] of the audio tag classification task (i.e. auto-tagging) of Music Information Retrieval Evaluation eXchange (MIREX) 2009 used heterogeneous model averaging to combine support vector machine (SVM) and AdaBoost. Meta learning algorithms such as random forest have also been adopted in a variety of MIR problems including genre classification, mood classification and auto-tagging [4, 8, 20–22]. Most of the existing methods adopt the inter-song strategy. In contrast, relatively little attention has been made to the intra-song variant.

The bagging technique was first proposed by Breiman [14]. It is a simple method that uses multiple training subsets for improving the stability of an ensemble system. It is one of the most important components behind popular machine learning algorithms such as random forest and rotation forest [23, 24]. Although the idea of bagging and its variants have been applied in a variety of pattern recognition problems including MIR, this work represents an early attempt that discusses the difference between inter-song and intra-song instance bagging for MIR problems, to our best knowledge.

3. METHODOLOGY

3.1. The "multi-tag classifier" framework

Music auto-tagging is usually formulated as a multi-label classification problem [25]. We follow one of the state-of-the-art frameworks that uses the *binary relevance* scheme together with the *classifier stacking* technique [4, 26, 27]. Binary relevance applies an independent binary classifier for modeling each tag [1–4, 27]. On the other hand, classifier stacking arranges a second layer of binary classifiers to combine the output scores of the first layer classifiers, so that the association between any pair of tags can be modeled [26, 27]. Classifier stacking also complements the tag independence assumption of the binary relevance scheme. We call the abovementioned framework a "*multi-tag classifier*," which is able to generate a series of probabilities for multiple tags, where each probability can be viewed as the *affinity* between a song and a tag. When annotating a test song, the tags with highest affinities will be selected.

978-1-4799-2893-4/14/$31.00 ©2014 IEEE

3.2. Non-bagging versus bagging

Suppose we are given a tagged music training set with N songs, $\mathcal{D} = \{\mathbf{X}_i, \mathbf{y}_i\}_{i=1}^N$. For the i-th song, $\mathbf{y}_i \in \{0,1\}^L$ is the binary multi-tag label, L is the number of tags, and $\mathbf{X}_i = \{\mathbf{x}_{i1}, \ldots, \mathbf{x}_{i\tau_i}\}$ represents the set of intra-song instances, where $\mathbf{x}_{ij} \in \mathbb{R}^M$ is a *short-time* (e.g., frame-level or segment-level) feature vector of length M, and τ_i denotes the number of instances.

For a conventional *non-bagging* auto-tagging method, one exploits all the information of $\mathcal{D}$ to train a single multi-tag classifier. A typical way is to first *pool* (temporally aggregate) each $\mathbf{X}_i$ across time into a feature vector $\mathbf{s}_i \in \mathbb{R}^M$ based on all components of $\mathbf{X}_i$, leading to $\mathcal{F} = \{\mathbf{s}_i, \mathbf{y}_i\}_{i=1}^N$, and then train the multi-tag classifier $f : \mathbb{R}^M \mapsto \mathbb{R}^L$ using the whole $\mathcal{F}$. In Section 4.3 we will present the details about the pooling methods considered in this work; the basic idea is to aggregate short-time feature vectors of a song into a song-level feature vector for classification [28].

To incorporate the bagging scheme, we can perform bootstrap sampling on $\mathcal{D}$ to generate a number of K training subsets $\mathcal{D}_k$, in which each is used to train the k-th "base multi-tag classifiers" f_k. For a test signal, the decisions from the base multi-tag classifiers are combined to give the final result, e.g., $\frac{1}{K}\sum_{k=1}^K f_k$. Two bagging strategies are explored below, namely, inter-song instance bagging and intra-song instance bagging.

3.3. Inter-song instance bagging

Inter-song instance bagging is operated upon $\mathcal{F}$ but partially exploits $\mathcal{F}$ to train a base multi-tag classifier at a time. Specifically, we can bootstrap either subsets of songs (i.e., $\mathcal{F}_k \subseteq \mathcal{F}$) or subsets of features (i.e., $\mathcal{F}_k = \{\hat{\mathbf{s}}_i, \mathbf{y}_i\}_{i=1}^N, \hat{\mathbf{s}}_i \subseteq \mathbf{s}_i$) to train the K number of base multi-tag classifiers. Both bootstrap schemes have been adopted in the literature [20–22, 29]. We consider the former variant (i.e., bootstrapping subsets of songs) in this paper, in order to focus on the effect of bagging rather than the features.

3.4. Intra-song instance bagging

In contrast, intra-song instance bagging is applied to the short-time features $\{\mathbf{x}_{i1}, \ldots, \mathbf{x}_{i\tau_i}\}$ for each training song. As Algorithm 1 presents, we perform the bootstrap directly on $\mathcal{D}$ (instead of $\mathcal{F}$) for K times, and each time executes the following procedures. For the i-th song, we uniformly sample $\lfloor q\tau_i \rfloor$ intra-song instances (q is the sampling percentage) and pool the instances into a feature vector $\mathbf{v}_i^k \in \mathbb{R}^M$ until all the training songs are done. Then, we train the k-th base multi-tag classifier f_k based on $\mathcal{D}_k = \{\mathbf{v}_i^k, \mathbf{y}_i\}_{i=1}^N$. Because Algorithm 1 performs bootstrap on $\mathcal{D}$, it is possible that $\mathcal{D}_{k'} = \mathcal{D}_k$ for $k \neq k'$.

In the test phase, we perform the same bootstrap procedures for a given song $\hat{\mathbf{X}}$ to obtain K feature vectors $\{\hat{\mathbf{v}}^k\}_{k=1}^K$ and then generate the affinity as $\frac{1}{K}\sum_{k=1}^K f_k(\hat{\mathbf{v}}^k)$. Clearly, the method reduces to the non-bagging scheme when $K = 1$ and $q = 1$.

The major difference between inter- and intra-song bagging lies in the input to the base multi-tag classifiers. The former uses a feature vector pooled from the entire song, whereas the latter uses a feature vector pooled from a random subset of intra-song short-time feature vectors for each song. In other words, the inter-song method performs bootstrapping *after* pooling, whereas the intra-song method performs bootstrapping *before* pooling. Because of this difference, the intra-song method better capitalizes diverse subsets of instances from a song, which might in turn help the modeling of local events such as "guitar solo" [16] within a song.

Algorithm 1: Intra-song instance bagging

Input: Training data $\mathcal{D} = \{\mathbf{X}_i, \mathbf{y}_i\}_{i=1}^N$; number of base multi-tag classifiers K; sampling percentage q;
Output: Ensemble of multi-tag classifiers $\{f_k\}_{k=1}^K$

```
for 1 ← k to K do
    for 1 ← i to N do
        Φ ← uniformly sample ⌊qτ_i⌋ intra-song instances from X_i;
        v_i^k ← pool Φ into a vector;
    end
    D_k ← {v_i^k, y_i}_{i=1}^N;
    f_k ∈ R^L ← train a base multi-tag classifier on D_k;
end
```

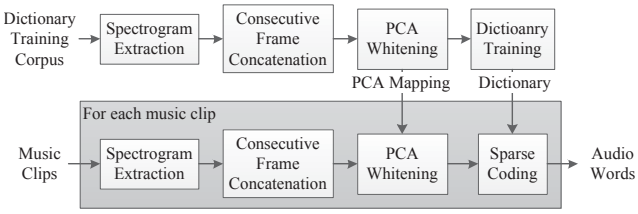


Fig. 1: Feature extraction pipeline for audio word feature [32].

3.5. Implementation details

We used the linear SVM implemented in the `LIBLINEAR` library [30] for a binary classifier in the multi-tag classifier framework. It has been shown that linear SVM has comparable prediction performance to non-linear SVM but better efficiency when the feature dimension is large and sparse (e.g., [29, 31–33]). However, as each binary classifier is trained independently, directly averaging the outputs of different base multi-tag classifiers for ensemble classification might be sub-optimal. Instead, for a binary classifier in a multi-tag classifier, we used the training data to train a sigmoid function that transforms the raw output score into a probability estimate [4,29,34], and aggregated the probability estimates for final decision.

4. FEATURE REPRESENTATION

The following two state-of-the-art audio feature representations were considered in our study — audio word and EchoNest features.

4.1. Audio word representation (AW)

AW is a text-like representation of music that has been found effective for many MIR problems in recent years [31–33]. We show the extraction pipeline in Figure 1 and present some details below.

Spectrogram extraction transforms the time domain music signal into log-powered spectrum for each short-time frame. We used 46ms frame size with 50% overlaps, following [31].

Multiple frame concatenation (MFR) concatenates successive η frames into a feature vector to better incorporate temporal information of a music signal [7, 33]. We used $\eta = 4$ with 50% overlaps, following a prior art [33].

PCA whitening is another simple trick that improves the performance of a feature learning system [7, 28, 33]. It refers to applying principle component analysis (PCA) to the low level representation

before sparse coding. We selected the number of principle component so that 90% of variance is maintained. We use $\mathbf{z}$ to denote the log-powered spectrum after MFR and PCA whitening.

Sparse coding has been found a more effective way of computing AW, in comparison to competing methods such as vector quantization methods [7,31]. Given an input vector $\mathbf{z} \in \mathbb{R}^m$ and a pre-built dictionary $\mathbf{D} \in \mathbb{R}^{m \times k}$, it computes the audio word representation $\alpha^* \in \mathbb{R}^k$ (for each frame) via

$$\alpha^* = \arg\min_{\alpha} \ \|\mathbf{z} - \mathbf{D}\alpha\|_2^2 + \lambda\|\alpha\|_1 \,. \qquad (1)$$

As a result, α^* is a k-dimensional vector with a few non-zero terms that as a whole well approximates the input vector $\mathbf{z}$. The parameter λ is a trade-off between sparsity term and reconstruction error. We set $\lambda = m^{-1/2}$ following [35] and used the LARS-LASSO algorithm [36] to solve Eq. 1. The resulting α^* is adopted as the short-time feature $\mathbf{x}_{ij}$ for some datasets (cf. Section 5.1).

Dictionary learning. As can be seen from Eq. 1, the dictionary $\mathbf{D}$ plays an important role in computing AW. It has been found that learning a dictionary from data usually leads to better performance [31]. In this work, we used the online dictionary learning (ODL) algorithm [35] to learn the dictionary from USPOP2002, an external dataset composed of nearly 7,000 contemporary pop music [37].

4.2. EchoNest feature

Features computed by the EchoNest API (`http://echonest.com/`) has been widely used in MIR in recent years [18, 38, 39]. We employed the EchoNest API to compute segment-level 12-D timbre descriptor (ENT) and 12-D pitch descriptors (ENP), and used the concatenation of them as another type of short-time feature $\mathbf{x}_{ij}$. ENT describes the timbre characteristics of the magnitude spectrogram, whereas ENP is chroma-like [40]. The features of a song are not in the frame-level but in the segment-level computed by the EchoNest API, where each segment corresponds to an acoustically homogenous fragment with multiple consecutive frames. Please note that the EchoNest features can be obtained by querying the API with song titles and artist names, so audio files are not needed.

4.3. Temporal aggregation

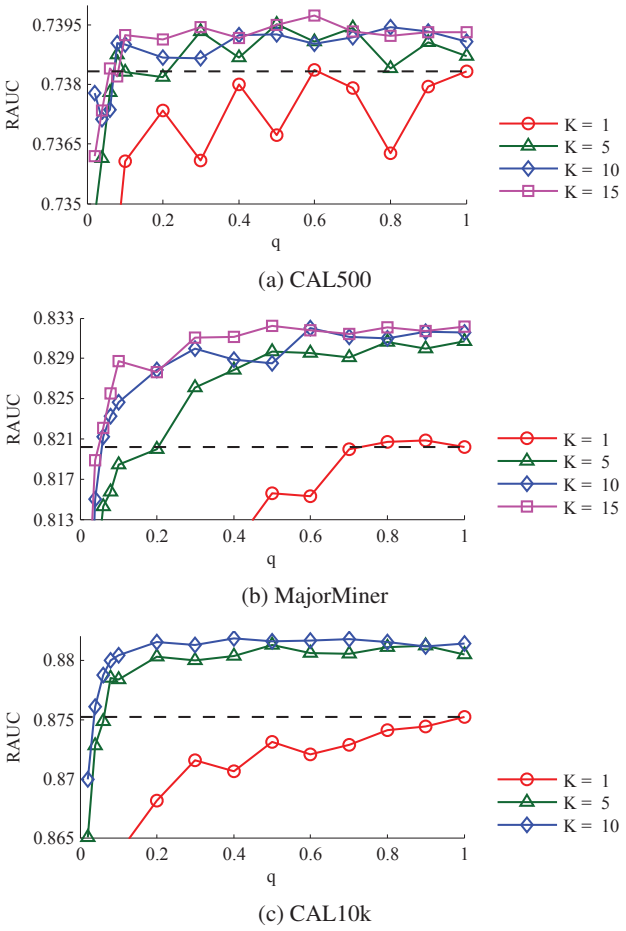
Temporal aggregation is a process that pools short-time feature vectors of a song into a song-level feature vector for classification [28]. Different methods were adopted for the two features in this work. Since AW is a histogram-like feature, we used sum pooling and normalized it by sum-to-one normalization and square-root power normalization, following [41]. On the other hand, we pooled the EchoNest features by calculating the mean and standard divination along the temporal dimension, and then took the z-scores for feature normalization along each feature dimension.

5. EXPERIMENT

5.1. Dataset and evaluation protocol

We evaluated the performance of auto-tagging using the following three datasets to ensure the result is generalizable.

CAL500 contains 502 western popular music manually annotated with a lexicon of 174 pre-defined tags [16]. The length of a music audio clip ranges from 3 second to over 22 min. According to the common protocol in the literature [7], we used a subset of 97 tags and evaluated the performance for both semantic *annotation* and *retrieval*. The accuracies of annotation (i.e., annotating a song with tags) and retrieval (i.e., retrieving relevant songs with respect to a tag query) were evaluated in terms of F-score and the area under the receiver operating characteristic curve (RAUC), respectively [16]. We used AW as the feature representation for this dataset.



(a) CAL500

(b) MajorMiner

(c) CAL10k

Fig. 2: The per-tag retrieval performances on the three datasets. The baseline method is represented by a dashed line. The solid lines with different colors show the performances of the proposed intra-song instance bagging with different Ks.

MajorMiner was crawled from the MajorMiner.org website [17]. Therefore, it may be slightly different from the one used in the MIREX auto-tagging task. It contains 2,472 music clips annotated over 45 tags, with each audio clip being 10 sec long. Following the setup of MIREX, we used three-fold cross validation to evaluate the performance in terms of *per-song* annotation AUC (AAUC) and *per-tag* retrieval AUC (RAUC) for the semantic annotation and retrieval tasks, respectively. For feature representation, AW is used for this dataset.

CAL10k consists of 10,870 partially annotated songs over a lexicon of 1,053 tags by expert music editors of the music service company Pandora (`http://www.pandora.com`) [18]. In this work, we only considered the 153 genre tags defined by Tingle *et al.* [18]. Due to the copyright issues, the audio clips for this dataset are not available. Therefore, we used the EchoNest feature as the feature

Table 1: Performance comparison against the stat-of-the-art

Dataset	Method	Annotation	Retrieval
		Fscore	RAUC
CAL500	HEM+DTM [5]	0.264	0.708
	SVM+DMM [27]	0.285	0.730
	SparseRBM+DNN [7]	0.292	0.754
	non-bagging[a]	0.274	0.738
	intra-song bagging[b]	0.279	0.740
		AAUC	RAUC
MajorMiner	CBA [3]	0.861	0.754
	CSML [4]	0.895	0.828
	RAkEL [4, 42]	0.896	0.814
	non-bagging[a]	0.891	0.820
	intra-song bagging[c]	0.900	0.833
		—	RAUC
CAL10k	GMM+ENTΔ [18]	—	0.887
	SVM+Sparse [33]	—	0.854
	non-bagging[a]	—	0.875
	intra-song bagging[d]	—	0.882

[a] non-bagging ($K = 1, q = 1$)
[b] intra-song ($K = 15, q = 0.6$)
[c] intra-song ($K = 5, q = 0.3$)
[d] intra-song ($K = 10, q = 0.8$)

Table 2: Performance comparison between the inter-song ('Inter') and intra-song ('Intra') bagging methods on the three datasets

Dataset	Setup		Annotation		Retrieval	
			Inter	Intra	Inter	Intra
	K	q	Fscore		RAUC	
CAL500	15	0.2	0.243	**0.276**	0.712	**0.739**
		0.4	0.271	**0.276**	0.729	**0.739**
		0.6	0.276	**0.278**	0.733	**0.740**
		0.8	0.272	**0.279**	0.732	**0.739**
		1	**0.278**	0.278	0.735	**0.739**
	K	q	AAUC		RAUC	
MajorMiner	15	0.2	0.896	**0.896**	0.818	**0.828**
		0.4	**0.903**	0.896	0.829	**0.831**
		0.6	**0.899**	0.898	0.828	**0.832**
		0.8	0.898	**0.898**	0.826	**0.832**
		1	0.895	**0.898**	0.823	**0.832**
	K	q	–		RAUC	
CAL10k	10	0.2	–	–	0.848	**0.882**
		0.4	–	–	0.871	**0.882**
		0.6	–	–	0.877	**0.882**
		0.8	–	–	0.880	**0.882**
		1	–	–	0.879	**0.881**

representation. Following [18], we evaluated the performance only for retrieval and used the pre-defined train and test splits.

5.2. Result and discussion

Figure 2 shows the performance of intra-song instance bagging on the three datasets with respect to different values of K and q. Note that the performances with $K = 1$ (i.e., using only one multi-tag classifier) correspond to the non-bagging scheme, and the performance with $K = 1$ and $q = 1$ is considered as the baseline.

The following two observations are made regarding Figure 2. First, we can observe from the systems without ensemble ($K = 1$) that only a fraction of the frames (e.g., $q = 0.5$) are needed to achieve a comparable performance against the baseline ($q = 1$) on all the tested datasets. In other words, we could apply frame-based feature extraction algorithm to just half of the overall frames and pool them, while still getting a competitive song-level feature. This finding could potentially double the efficiency of many state-of-the-art frame-based feature extraction algorithms. Second, when q and K are sufficiently large, the intra-song instance bagging method outperforms the baseline. One possible reason is that the ensemble technique can usually reduce the variance of feature modeling and at the same time avoid over-fitting. We may state that this improvement is unlikely to be feature dependent or dataset dependent as it happens with three different datasets and two different features.

Table 1 shows that the proposed intra-song instance bagging method is fairly competitive against several state-of-the-art methods, many of which involves advanced feature extraction algorithms (e.g., sparse restricted Boltzmann machine (SparseRBM) [7] and dynamic texture model (DTM) [5]) or sophisticated machine learning algorithms (e.g., random k-labelsets (RAkEL) [25], cost-sensitive multi-label learning (CSML) [4], deep neural network (DNN) [7], and Dirichlet mixture model (DMM) [27]). For example, the proposed method obtains the best accuracy in both annotation and retrieval on MajorMiner, and the second-highest accuracy in retrieval on both CAL500 and CAL10k. From Table 1, we also see that the baseline method (non-bagging, $K = 1$, $q = 1$) are in fact competitive as well, showing the effectiveness of the multiple-tag classifier framework (cf. Section 3.1) and the AW or EchoNest features. However, to reach comparable result with the state-of-the-art, incorporating ensemble techniques such as intra-song instance bagging seems to be advisable. As intra-song instance bagging is a meta algorithm, we believe that it is also applicable to other existing systems for improving the performance.

Lastly, Table 2 compares the intra-song instance bagging with the conventional inter-song counterpart with varying qs and a fixed K. The result validates the advantage of the intra-song method. For the retrieval task, the performance difference with certain setting can reach up to 3.4% (e.g., $q = 0.2$ on CAL10k). As for the annotation task, the performance difference is not pronounced.

6. CONCLUSION

In this paper, we have introduced a novel meta algorithm called intra-song instance bagging and validated its efficiency through testing it on three different datasets and two distinct features. We have also shown that intra-song instance bagging performs better than conventional inter-song instance bagging and many state-of-the-art methods. Performing frame samplings before feature extraction offers an addition gain in efficiency. As intra-song instance bagging is a meta algorithm, it holds the promise of improving other MIR or pattern recognition applications.

7. ACKNOWLEDGMENT

This work was supported by the National Science Council of Taiwan under grants NSC 102-2221-E-001-004-MY3, NSC 101-2221-E-001-019-MY3, and the Academia Sinica Career Development Award 102-CDA-M09.

8. REFERENCES

[1] T. Bertin-Mahieux, D. Eck, F. Maillet, and P. Lamere, "Autotagger: A model for predicting social tags from acoustic features on large music databases," *JNMR*, 2008.

[2] D. Turnbull, L. Barrington, D. Torres, and G. Lanckriet, "Semantic annotation and retrieval of music and sound effects," in *IEEE TASLP*, 2008.

[3] M. D. Hoffman, D. M. Blei, and P. R. Cook, "Easy as CBA: A simple probabilistic model for tagging music," in *ISMIR*, 2009.

[4] H.-Y. Lo, J.-C. Wang, H.-M. Wang, and S.-D. Lin, "Cost-sensitive multi-label learning for audio tag annotation and retrieval," *IEEE TMM*, vol. 13, no. 3, pp. 518–529, 2011.

[5] E. Coviello, A. B. Chan, and G. Lanckriet, "Time series models for semantic music annotation," in *IEEE TASLP*, 2011.

[6] G. Marques, M. A. Domingues, T. Langlois, and F. Gouyon, "Three current issues in music autotagging," in *ISMIR*, 2011.

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

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

[9] J.-C. Wang, H.-M. Wang, and S.-K. Jeng, "Playing with tagging: A real-time music tagging player," in *ICASSP*, 2012.

[10] M. Levy and M. Sandler, "Music information retrieval using social tags and audio," in *IEEE TMM*, 2009.

[11] J.-C. Wang, Y.-C. Shih, M.-S. Wu, H.-M. Wang, and S.-K. Jeng, "Colorizing tags in tag cloud: a novel query-by-tag music search system," in *ACM MM*, 2011, pp. 293–302.

[12] L. Barrington, D. Turnbull, and G. Lanckriet, "Game-powered machine learning," *PNAS*, vol. 109, no. 17, pp. 6411–6416, 2012.

[13] J.-C. Wang, M.-S. Wu, H.-M. Wang, and S.-K. Jeng, "Query by multi-tags with multi-level preferences for content-based music retrieval," in *IEEE ICME*, 2011.

[14] L. Breiman, "Bagging predictors," *Machine learning*, vol. 24, no. 2, pp. 123–140, 1996.

[15] L. Rokach, "Ensemble-based classifiers," *Artificial Intelligence Review*, vol. 33, no. 1-2, pp. 1–39, 2010.

[16] D. Turnbull, L. Barrington, D. Torres, and G. Lanckriet, "Towards musical query-by-semantic-description using the CAL500 data set," in *ACM SIGIR*, 2007, pp. 439–446.

[17] M. I Mandel and D. P W Ellis, "A web-based game for collecting music metadata," in *ISMIR*, 2007.

[18] D. Tingle, Y. E. Kim, and D. Turnbull, "Exploring automatic music annotation with "acoustically-objective" tags," in *ISMIR*, 2010.

[19] H.-Y. Lo, J.-C. Wang, and H.-M. Wang, "Homogeneous segmentation and classifier ensemble for audio tag annotation and retrieval," in *Proc. IEEE ICME*, 2010, pp. 304–309.

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

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

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

[23] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[24] J. J. Rodriguez, L. I. Kuncheva, and C. J. Alonso, "Rotation forest: A new classifier ensemble method," *IEEE TPAMI*, vol. 28, no. 10, pp. 1619–1630, 2006.

[25] G. Tsoumakas and I. Katakis, "Multi-label classification: An overview," *Int. J. Data Warehousing and Mining*, vol. 2007, pp. 1–13, 2007.

[26] S. R. Ness, A. Theocharis, G. Tzanetakis, and L.G. Martins, "Improving automatic music tag annotation using stacked generalization of probabilistic svm outputs," in *Proc. ACM MM*, 2009, pp. 705–708.

[27] R. Miotto and G. Lanckriet, "A generative context model for semantic music annotation and retrieval," *IEEE TASLP*, vol. 20, no. 4, pp. 1096–1108, 2012.

[28] P. Hamel, S. Lemieux, Y. Bengio, and D. Eck, "Temporal pooling and multiscale learning for automatic annotation and ranking of music audio," in *ISMIR*, 2011, pp. 729–734.

[29] J.-C. Wang, H.-Y. Lo, S.-K. Jeng, and H.-M. Wang, "MIREX 2010: Audio classification using semantic transformation and classifier ensemble," in *MIREX*, 2010.

[30] 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.

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

[32] 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," in *IEEE TMM*, 2014.

[33] C.-C. M. Yeh and Y.-H. Yang, "Towards a more efficient sparse coding based audio-word feature extraction system," in *APSIPA ASC*, 2013.

[34] H.-T. Lin, C.-J. Lin, and R. C Weng, "A note on Platts probabilistic outputs for support vector machines," *Machine learning*, vol. 68, no. 3, pp. 267–276, 2007.

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

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

[37] 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.

[38] B. McFee and G. Lanckriet, "The natural language of playlists," in *ISMIR*, 2011.

[39] B. McFee, T. Bertin-Mahieux, D. Ellis, and G. Lanckriet, "The million song dataset challenge," in *AdMIRe*, 2012.

[40] A. Schindler and A. Rauber, "Capturing the temporal domain in echonest features for improved classification effectiveness," in *AMR*, 2012.

[41] C.-C. M. Yeh, L. Su, and Y.-H. Yang, "Dual-layer bag-of-frames model for music genre classification," in *ICASSP*, 2013.

[42] G. Tsoumakas and I. Vlahavas, "Random k-labelsets: An ensemble method for multilabel classification," in *ECML*, 2007.