Music Annotation and Retrieval using Unlabeled Exemplars: Correlation and Sparse Codes

Ping-Keng Jao and Yi-Hsuan Yang, *Member, IEEE*

***Abstract*—Tagging music signals with semantic labels such as genres, moods and instruments is important for content-based music retrieval and recommendation. While considerable effort has been made, automatic music annotation is still considered challenging due to the difficulty of extracting good audio features that capture the characteristics of different tags. To address this issue, we present in this letter two exemplar-based approaches that represent the content of a music clip by referring to a large set of unlabeled audio exemplars. The first approach represents a music clip by the set of audio exemplars that is highly correlated with the short-time feature vectors of the clip, whereas the second approach represents a music clip as sparse linear combinations of its short-time feature vectors over the audio exemplars. Music annotation is then performed by learning the relevance of the audio examples to different tags using labeled data. These two approaches effectively capitalize the availability of unlabeled data to explore the commonality of music signals to find out tag-specific acoustic patterns, without domain knowledge and feature design. Evaluation on the CAL10k music genre tagging dataset for tag-based music retrieval shows that, with thousands of unlabeled audio examples randomly drawn from the Million Song Dataset, the proposed approaches lead to remarkably higher precision rates than existing approaches.**

***Index Terms*—Music tagging, retrieval, sparse representation.**

I. Introduction

AUTOMATIC music annotation, a.k.a. music autotagging, refers to the task of automatically assigning semantic labels (tags) such as genres, moods, and instruments to music objects (*e.g.* artists, albums, tracks, or segments of a track) so as to facilitate applications such as tag-based retrieval, similarity search, recommendation, and visualization [1]–[7]. In the past decade, a great effort has been made to use supervised machine learning algorithms to map signal-level audio features extractable by machine (*e.g.* temporal or spectral features) to high-level semantic labels using manually pre-labeled training samples [8]–[16]. The task, however, remains challenging due to the following three issues: the scarcity of well-labeled training data [17], [18], the complexity involved in formalizing and evaluating the task while taking care of possible confounds [18], [19], and the difficulty of extracting good audio features that capture the characteristics of each tag [20]–[24]. Good feature design is hard to come by, for example for tags that are social and cultural constructs (*e.g.* genres [25]). The goal of this letter is to address the last issue by proposing novel ways of computing audio features.

Our approaches are motivated by the success of exemplar-based approaches for pattern recognition [26]–[30]. The idea is to build a *local* classification model from a few relevant pre-labeled training samples for each test sample, instead of deriving a single *global* model from all the training samples before the test sample is seen [30]. For instance, the k-nearest neighbor (kNN) algorithm classifies a test sample by the majority vote over the labels of its k nearest training samples in a given feature space [26]. The sparse representation-based classification (SRC) algorithm approximates a test sample by a sparse linear combination with respect to an overcomplete dictionary comprising of all the training samples, and then performs classification by evaluating which class-specific sub-dictionary (*i.e.* training samples from the same class) leads to the minimal coding residual [27]. As the sparse representation is given by a small number of relevant training samples, it is likely to consist of samples from the same class. Comparing to kNN, SRC does not fix the number of neighborhood training samples chosen, and appears to be a more robust classifier. It has been shown that SRC performs well even in the presence of occlusion or corruption in face and object recognition [28], and noises in speech recognition [29].

The aforementioned two algorithms, however, are not readily applicable to music autotagging for the following reasons. First, existing labeled datasets for music autotagging usually assign tags to 10–30 seconds clips, if not the whole tracks [7]–[11], whereas audio features are usually extracted over short-time analysis windows referred to as *frames*, over which the signal is considered as stationary [31]. As a clip can consist of hundreds or thousands of frames, directly using frame-level features as exemplars would result in exceedingly large number of potentially redundant exemplars. Moreover, frame-level features from different tags may resemble one another, limiting the discriminative ability of the classifier. On the other hand, designing effective *temporal integration* algorithms that summarize the content of successive frames in the clip or chunk (*e.g.* per second) level remains difficult [31], especially for tags that apply to only short fragments of a clip (*e.g.* the tag 'guitar solo'). Second, as a music object can be associated with not only one but multiple tags, the idea of class-specific subdictionary becomes vague. It is uncertain which tags are relevant if the sparse linear approximation of a test sample involves training samples associated with multiple tags.

In light of the above observations, we propose to leverage the abundance of *unlabeled* data as exemplars in an overcomplete dictionary to compute the feature representations, and then use *labeled* training data to learn a discriminative classifier for

Manuscript received February 06, 2015; revised April 01, 2015; accepted April 28, 2015. Date of publication May 13, 2015; date of current version May 20, 2015. The associate editor coordinating the review of this manuscript was Prof. Paris Smaragdis.

The authors are with the Research Center for Information Technology Innovation, Academia Sinica, Taipei 11564, Taiwan (e-mail: nafraw@citi.sinica.edu.tw; yang@citi.sinica.edu.tw).

Digital Object Identifier 10.1109/LSP.2015.2433061

music autotagging, using the computed feature representations. The dictionary is formed by frame-level feature vectors *randomly* drawn from a large and diverse set of unlabeled music clips, so as to avoid losing short-time audio information by temporal integral and the redundancy of using feature vectors from a limited number of clips. The unlabeled exemplars provide a universal basis to represent frame-level music content. We can then compute feature representations for both the training and test samples by referring to this dictionary, and use algorithms such as the support vector machine (SVM) [32] to learn a classifier. As the classification is performed on a feature representation comprising of unlabeled exemplars, the classification algorithm is actually learning the relevance of the exemplars to each tag, and the prediction for a test sample actually depends on the set of exemplars in the dictionary that are relevant to that test sample. As the exemplars *themselves* are used to represent music content, this approach does not involve much feature design.

As embodiment of the aforementioned idea, we propose two exemplar-based approaches to audio feature computation. The first approach resembles the idea of kNN and represents a music sample (*i.e.* a clip or a track) by a binary vector indicating the subset of exemplars in the dictionary that are most correlated with the frame-level feature vectors of the music sample. The second approach resembles the idea of SRC and represents a music sample by the sparse representation of its frame-level feature vectors with respect to the dictionary. Accordingly, the first approach assumes two samples share similar tags if their short-time features are correlated with similar sets of exemplars, whereas the second approach assumes two samples share similar tags if their short-time features can be approximated by similar sparse linear combinations of the exemplars. The multi-label classification problem of music autotagging is cast to multiple binary classification problems [33], using one linear kernel SVM for each tag to predict the relevance between a tag and a test sample. Evaluation on a music autotagging dataset for tag-based music retrieval (a.k.a. query-by-tag) [12]–[15] shows the superiority of the proposed approaches over prior arts in the precision rates.

II. Problem Statement

We are given a labeled dataset $\{X_i, \mathbf{y}_i\}_{i=1}^{l}$ with l clips (or tracks) and an unlabeled dataset $\{X_i\}_{i=l+1}^{l+u}$ with u clips, where $X_i = \{\mathbf{x}_i^{(1)}, \ldots, \mathbf{x}_i^{(\tau_i)}\}$ is a collection of p-dimensional frame-level feature vectors $\mathbf{x}_i^{(t)} \in \mathbb{R}^p$ for the clip with τ_i short-time frames, and $\mathbf{y}_i \in \{0,1\}^q$ is a label vector indicating which of the q possible tags can be applied to the corresponding clip. The problem of music autotagging is to learn a function $f_{X_i} : \mathbb{R}^{p\times\tau_i} \mapsto \{0,1\}^q$ from the labeled dataset.

The main idea of the proposed approach is to employ an overcomplete dictionary $\mathbf{D} \in \mathbb{R}^{p\times m}$ composing of $m > p$ frame-level feature vectors $\mathbf{d}_j \in \mathbb{R}^p$ randomly drawn from the unlabeled dataset $\{X_i\}_{i=l+1}^{l+u}$, such that we can use $\mathbf{D}$ to compute exemplar-based clip-level representation $\boldsymbol{\psi}_i \in \mathbb{R}^m$ that nicely captures the audio characteristics for each X_i. Using the new representation, we learn a function $f_{\boldsymbol{\psi}} : \mathbb{R}^m \mapsto \{0,1\}^q$ from the labeled dataset for autotagging. In this letter, this is achieved by learning q binary classifiers, one for each tag.

Without loss of generalizability, we drop the subscript i for the index of clips in the following discussion. Moreover, we assume the frame-level feature vectors are normalized to unit energy (*i.e.* $\|\mathbf{x}^{(t)}\|_2 = \|\mathbf{d}_j\|_2 = 1$). We use bold upper case to represent matrices and bold lower case for column vectors.

III. Exemplar-based Correlation Codes

We can measure the linear dependence between two variables by the Pearson product-moment correlation coefficient. As the frame-level feature vectors are normalized, the correlation between $\mathbf{x}^{(t)}$ and an exemplar $\mathbf{d}_j$ is given by

$$r_j^{(t)} = \mathbf{d}_j^T \mathbf{x}^{(t)}, \tag{1}$$

where $r > 0$ implies that the two vectors are pointing into the same half-space of $\mathbb{R}^p$, $r = \pm 1$ indicates they are in the same or opposite direction, and $r = 0$ if they are orthogonal. To obtain a clip-level representation for X, we first calculate the frame-level binary indicator vector $\boldsymbol{\sigma}^{(t)} \in \mathbb{R}^m$ for each $\mathbf{x}^{(t)}$:

$$\sigma_j^{(t)} = (h(r_j^{(t)}) - \theta^{(t)})_+, \tag{2}$$

where $\sigma_j^{(t)}$ denotes the jth element of $\boldsymbol{\sigma}^{(t)}$, $h(\cdot)$ is a mapping function such as $h(z) = z$ or $h(z) = |z|$ (*i.e.* considering both positive and negative correlation), the operator $(z)_+$ returns 1 if $z > 0$ and 0 otherwise, and $\theta^{(t)}$ is a threshold controlling the number of dictionary atoms considered as relevant to $\mathbf{x}^{(t)}$. Then, we compute the clip-level feature representation $\boldsymbol{\psi}^{\mathrm{c}}$ by

$$\boldsymbol{\psi}^{\mathrm{c}} = \left(\sum\nolimits_{t=1}^{\tau} \boldsymbol{\sigma}^{(t)} - \vartheta\right)_+, \tag{3}$$

where the scalar ϑ decides the number of non-zero elements in $\boldsymbol{\psi}^{\mathrm{c}}$. There are at least two possible methods for setting ϑ:

- *Union-based*: we set $\boldsymbol{\psi}_j^{\mathrm{c}} = 1$ as long as $\mathbf{d}_j$ is considered relevant to at least one of the input frames (*i.e.* $\exists t \in [1,\tau], \sigma_j^{(t)} > 0$). This is equivalent to setting $\vartheta = 0$.
- *Voting-based*: we set $\boldsymbol{\psi}_j^{\mathrm{c}} = 1$ if $\sum_{t=1}^{\tau} \boldsymbol{\sigma}_j^{(t)}$ is among the top k ones for $j \in [1,m]$, or, equivalently, when $\mathbf{d}_j$ is considered relevant to sufficiently large number of input frames. This can be achieved by properly setting ϑ.

The voting-based method allows for exactly the same number (*i.e.* k) of non-zero elements in $\boldsymbol{\psi}^{\mathrm{c}}$ for different clips, while the union-based method adaptively uses different number of non-zero elements according to signal complexity and variability. We refer to $\boldsymbol{\psi}^{\mathrm{c}}$ as the exemplar-based correlation codes (ECC).

Remark on Efficiency: The operations of computing a clip is almost linear in the m and τ of that clip. Computing (1) is already fast, but it can be made even faster by exploiting the identity $\|\mathbf{a} - \mathbf{b}\|_2^2 = \|\mathbf{a}\|_2^2 + \|\mathbf{b}\|_2^2 - 2\mathbf{a}^T\mathbf{b}$ and the fact that the frame-level feature vectors have been normalized; since the exemplar with the largest correlation to an input is also the one with the smallest Euclidean distance to the input, one can use nearest-neighbor search algorithms such as k-D Tree and PCA Tree [34] to speed up the computing of ECC.

IV. Exemplar-based Sparse Codes

It has been shown that a sparse linear approximation of an input $\mathbf{x}^{(t)}$ with respect to a dictionary $\mathbf{D}$ can be calculated by solving the following l_1-minimization problem [35]–[37]:

$$\boldsymbol{\alpha}^{(t)} = \arg\min_{\boldsymbol{\alpha}} \frac{1}{2}\|\mathbf{x}^{(t)} - \mathbf{D}\boldsymbol{\alpha}\|_2^2 + \lambda\|\boldsymbol{\alpha}\|_1, \tag{4}$$

where $\boldsymbol{\alpha} \in \mathbb{R}^m$ is the combination (or activation) coefficients and the parameter λ controls the tradeoff between the reconstruction error $\|\mathbf{x}^{(t)} - \mathbf{D}\boldsymbol{\alpha}\|_2^2$ and the sparsity-promoting l_1-regularizer $\|\boldsymbol{\alpha}\|_1 = \sum_{j=1}^m |\alpha_j|$. Many optimization algorithms have been proposed to solve this problem [38]–[40]. In this letter, we employ the homotopy-based least angle regression and shrinkage (LARS)-Lasso algorithm [41] for its demonstrated efficiency and effectiveness for pattern recognition problems [38]. The clip-level representation for X is obtained by taking the average across the coefficients $\boldsymbol{\alpha}^{(t)}$ for each frame $\mathbf{x}^{(t)}$,

$$\boldsymbol{\psi}^{\mathrm{s}} = \frac{1}{\tau} \sum\nolimits_{t=1}^{\tau} \boldsymbol{\alpha}^{(t)}. \tag{5}$$

Taking the average would preserve short-time music characteristics, because $\boldsymbol{\psi}^{\mathrm{s}}_j \neq 0$ as long as the corresponding exemplar $\mathbf{d}_j$ is chosen in the approximation of at least one of the input frames (*i.e.* $\exists t \in [1, \tau], \alpha_j^{(t)} > 0$). Our pilot study shows that this mean pooling is better than sum or max pooling [23]. We refer to $\boldsymbol{\psi}^{\mathrm{s}}$ as the exemplar-based sparse codes (ESC). Please note that ECC is a binary representation, but ESC is not.

Remark on Efficiency: Using LARS Lasso with a Cholesky-based implementation, the time complexity of computing $\boldsymbol{\alpha}^{(t)}$ is $O(pm\iota + m\iota^2)$ [42], [43], where ι is the number of iterations needed to reach a local minimum and in general ι equals to the number of non-zero elements in $\boldsymbol{\alpha}^{(t)}$ and $\iota \leq \min(p, m) = p$. A number of algorithms have been proposed in the literature for better efficiency. In this letter, we extend the frame-level Lasso screening algorithm proposed by Xiang *et al.* [42], [43] to the clip-level for speeding up, as described below.

A. Clip-level Lasso Screening

The idea of Lasso screening is to adaptively reduce the size of the dictionary $\mathbf{D}$ for each input $\mathbf{x}^{(t)}$ by removing atoms that are unlikely to be non-zero in $\boldsymbol{\alpha}^{(t)}$ [43]. It has been proven that we can remove $\mathbf{d}_j$ from $\mathbf{D}$ if the following inequality holds, without affecting the optimal solution of (4) [42]:

$$|r_j - (r_* - \lambda)\mathbf{d}_j^T \mathbf{d}_*| < \lambda - (r_* - \lambda)\sqrt{r_*^{-2} - 1}, \tag{6}$$

where $r_* \equiv \max_j |r_j|$ and $\mathbf{d}_* \in \{\pm \mathbf{d}_j\}_{j=1}^m$ is the possibly negated dictionary atom that leads to r_*. As this algorithm is designed to find a frame-specific subdictionary $\widehat{\mathbf{D}^{(t)}} \subset \mathbf{D}$ for each $\mathbf{x}^{(t)}$, we refer to it as frame-level Lasso screening.

In practice, however, we found that frame-level screening cannot accelerate the computation of ESC for two reasons [44]. First, the inequality (6) is seldom met as r_* is usually large (*i.e.* at least some atoms are highly correlated with the input) and λ is usually small (*e.g.* λ is usually set to $p^{-1/2}$ in solving (4) [45]). In consequence, the size of the subdictionary remains large. Second, using τ different subdictionaries demands τ times of memory transfer efforts for each clip, which is not memory efficient. Given the repetition nature of music, it is advisable to find a *clip-specific subdictionary* $\widehat{\mathbf{D}} \subset \mathbf{D}$ to compute the $\boldsymbol{\alpha}^{(t)}$ for all the frames in the same clip.

We propose to achieve this by borrowing the idea of ECC described in Section III. Specifically, since the ECC $\boldsymbol{\psi}^{\mathrm{c}}$ is a binary vector indicating the dictionary atoms that are deemed relevant to the input clip X, we can use the indices $\mathcal{I} \subset [1, m]$ of the non-zero entries in $\boldsymbol{\psi}^{\mathrm{c}}$ to sample the columns of $\mathbf{D}$ and form a clip-specific subdictionary $\widehat{\mathbf{D}}$, whose size is equal to the cardinality of $\mathcal{I}$ (*i.e.* number of non-zeros entries in $\boldsymbol{\psi}^{\mathrm{c}}$). We use $\widehat{\mathbf{D}}$ to solve for $\boldsymbol{\alpha}_{\mathcal{I}}^{(t)} \in \mathbb{R}^{|\mathcal{I}|}$ by LARS-Lasso for every frame of the clip, get the clip-level representation by $\widehat{\boldsymbol{\psi}^{\mathrm{s}}} = \frac{1}{\tau}\sum_{t=1}^{\tau} \boldsymbol{\alpha}_{\mathcal{I}}^{(t)}$, and finally obtain $\widetilde{\boldsymbol{\psi}^{\mathrm{s}}} \in \mathbb{R}^m$ by taking the values corresponding to $\mathcal{I}$ from $\widehat{\boldsymbol{\psi}^{\mathrm{s}}}$ and filling zeros for the other indices (*i.e.* $[1, m] - \mathcal{I}$). Due to the last *upsampling* step, the representation $\widetilde{\boldsymbol{\psi}^{\mathrm{s}}}$ of every clip lies in the same feature space spanned by the m unlabeled exemplars in $\mathbf{D}$. We refer to the aforementioned idea of using the same $\widehat{\mathbf{D}}$ for all the frames in a clip as clip-level Lasso screening, and the resulting $\widetilde{\boldsymbol{\psi}^{\mathrm{s}}}$ as approximated ESC (AESC).

Remark on Efficiency: As the complexity of LARS-Lasso is linear in the dictionary size, the computation of AESC is supposed to be faster than ESC by $m/|\mathcal{I}|$ folds (note this value is clip-dependent), assuming that the runtime of computing ECC is neglige comparing to that of solving (4). According to (3), we see that the value of $|\mathcal{I}|$ depends on the value of ϑ. We refer readers to the Table I and Fig. 1 in [44] for evaluation of the actual runtime of the encoding stage of ESC and AESC.

Remark on the Difference Between ESC and AESC: Because $\widehat{\mathbf{D}}$ is not constructed by using theoretically motivated rules [43] such as (6), clip-level Lasso screening can affect the solution of (4). In general, $\widehat{\mathbf{D}}\boldsymbol{\alpha}_{\mathcal{I}}^{(t)}$ cannot approximate $\mathbf{x}^{(t)}$ as well as $\mathbf{D}\boldsymbol{\alpha}^{(t)}$ does, and $\widetilde{\boldsymbol{\psi}^{\mathrm{s}}} \neq \boldsymbol{\psi}^{\mathrm{s}}$. Nevertheless, it may be fine to use $\widetilde{\boldsymbol{\psi}^{\mathrm{s}}}$, as the goal is to obtain a feature representation for classification, not for perfect reconstruction of the input.

Remark on the Frame-Level Indicator Vector of ECC: Finally, we note that the way we construct the frame-level indicator vector $\sigma_j^{(t)}$ of ECC in (2) can be connected with the frame-level Lasso screening rule in (6). For example, if we set $\lambda = r_*$, the rule (6) would reduce to $|r_j| < r_*$, which implies only the exemplar that has the maximal absolute correlation with the input frame would be kept. This is equivalent to using $h(z) = |z|$ and selecting only the maximal absolute correlated exemplar in (2). Although such a screening rule may seem stringent, it is fine as the clip-level $\boldsymbol{\psi}^{\mathrm{c}}$ (and $\widehat{\mathbf{D}}$) is formed by taking into account all the frames in the clip. From (4) we also see that setting λ to r_* instead of the smaller $p^{-1/2}$ stresses more on the sparsity of $\boldsymbol{\alpha}^{(t)}$ than the reconstruction error.

Remark on the Use of Sparse Codes in Audio: Sparse representation of audio signals has received much attention in recent years for its excellent empirical performance in classification and retrieval problems [21]–[24]. Different dictionary learning algorithms (*i.e.* to optimize $\mathbf{D}$) [39], encoding methods (*i.e.* to solve for $\mathbf{x}^{(t)}$) [23], [46], and temporal pooling methods (*i.e.* to get $\boldsymbol{\psi}^{\mathrm{s}}$) [16], [47] have been proposed and compared. The focus of this letter, however, is to investigate efficient ways of exploiting unlabeled exemplars themselves as the dictionary atoms, a topic that is seldom addressed before.

V. Experiment

Our evaluation uses the Million Song Dataset (MSD) [48] as the unlabeled dataset and the CAL10k (a.k.a. Swat10k) dataset [10] as the labeled dataset. MSD is a publicly-available collection of audio features and metadata for a million contemporary popular music tracks. To construct a dictionary of size m, we pick m random clips from MSD and randomly draw one exemplar from each clip. Due to the size and diversity of MSD, these exemplars are assumed to provide an adequate basis for the dictionary. CAL10k, on the other hand, consists of the human annotation of 147 genre tags for 10,267 songs. Popular tags with around 1,000 positive samples include genres such as 'Rock,' 'Jazz' and 'Pop,' whereas less popular tags (*e.g.* with less than 200 positive samples) are usually sub-genres, radios or musical events. The proposed approaches bypass the difficulty of feature design for these tags, as the exemplars themselves may already represent characteristics such as sound quality, playing techniques, recording environment, *etc*.

The short-time signal-level feature representation $\mathbf{x}^{(t)}$ can be obtained by extracting for example the magnitude spectra or the Mel-frequency spectra from audio signals [14], [23]. In this work, we use the 12-D timbre descriptors (ENT) and 12-D pitch descriptors (ENP) computed by the Echo Nest API as the short-time features. ENT is a descriptor of the magnitude spectra and ENP is chroma-like [49]. To incorporate temporal information, we also take their 1st- and 2nd-order instantaneous derivatives [10], leading to 36-D timbre (ENTΔ) and 36-D pitch descriptors (ENPΔ). Using these features encourages reproducibility, because everyone can obtain the features by querying the API with the artist names and song titles, without having the audio files for MSD or CAL10k.

We use the five training and testing splits specified in [10] for CAL10k and report the average result. We use the linear SVM implemented by LIBLINEAR [32] and train $q = 147$ binary classifiers, one for each tag, with parameters optimized by cross validating on the training split. Instead of evaluating the performance of music annotation, we directly evaluate the performance of using the predicted tags for tag-based music retrieval [12]–[15], in terms of the area under the receiver operating characteristic curve (AUC), mean average precision (MAP) and precision at rank 10 (P10). The values for these metrics all fall within [0,1], and larger values indicate better performance. For each tag, we rank the test clips in descending order of the decision values computed by SVM and calculate the above measures according to the ranking [16]. We select only one exemplar for each frame in (2) with $h(z) = |z|$, and use the voting-based method in (3), such that $|\boldsymbol{\psi}^{\mathrm{c}}|_1 = |\mathcal{I}| = \max(m/10, p)$. Moreover, we set λ to $p^{-1/2}$ in (4) [45].

Table I compares the performance of ECC, ESC and AESC, using ENT Δ and ENPΔ to build two m-atom dictionaries and concatenating the resulting two representations $\boldsymbol{\psi}$s for each clip for training the classifier. We see that the performance of the three algorithms increases as the dictionary size m grows, for all metrics. We also find that ESC performs the best among the three, and that AESC outperforms ECC when $m \geq 4,000$.

Table II compares AESC with existing approaches, using only ENTΔ as the short-time feature. We see that AESC obtains much higher precision rates (i.e. MAP and P10) than the state-of-the-art [10] when $m > 1,000$. From Tables I and II, we also see that even the simplest ECC can outperform the prior arts in MAP and P10, validating the use of the exemplars. The proposed approaches are on par with prior arts in AUC.

TABLE I
Performance of Music Annotation and Retrieval on CAL10k, Using Both ENTΔ and ENPΔ as Short-Time Features

m	ECC			ESC			AESC		
	AUC	MAP	P10	AUC	MAP	P10	AUC	MAP	P10
128	0.859	0.214	0.279	0.879	0.241	0.305	0.848	0.201	0.263
250	0.866	0.232	0.295	0.884	0.258	0.323	0.850	0.206	0.272
1k	0.871	0.246	0.315	0.888	0.275	0.342	0.862	0.236	0.301
2k	0.872	0.253	0.323	0.892	0.286	0.353	0.869	0.253	0.319
4k	0.874	0.260	0.330	0.892	0.290	0.354	0.876	0.265	0.334
10k	0.872	0.262	0.328	0.894	0.298	0.368	0.884	0.279	0.348
20k	0.873	0.261	0.325	0.895	0.301	0.370	0.891	0.296	0.368

TABLE II
Performance Comparison Between AECS (ENTΔ) With Prior Arts for Music Annotation and Retrieval on CAL10k

Approach	AUC	MAP	P10
AESC, ENTΔ, $m = 250$	0.836	0.187	0.247
AESC, ENTΔ, $m = 1$k	0.851	0.221	0.283
AESC, ENTΔ, $m = 4$k	0.869	0.249	0.312
AESC, ENTΔ, $m = 20$k	0.884	0.279	0.349
random guess [10]	0.501	0.018	0.015
GMM, ENTΔ [10]	0.887	0.211	0.266
HEM-DTM, Mel-spectra [12]	0.870	0.140	0.180
SC, magnitude spectra [16]	0.854	0.202	0.253
SC, Mel-spectra [24]	0.874	0.195	0.246

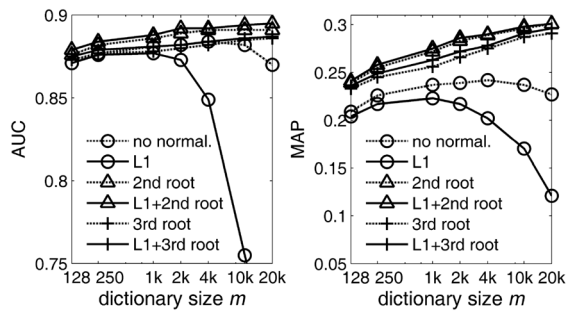

Fig. 1. Performance of music annotation and retrieval on CAL-10k, using both ENTΔ and ENPΔ features for computing ESC but different normalization methods before classifier training.

Finally, Fig. 1 shows the effect of applying different normalization methods on ESC, before using it for classifier training: L1 ($\boldsymbol{\psi}^{\mathrm{s}} \leftarrow \boldsymbol{\psi}^{\mathrm{s}}/|\boldsymbol{\psi}^{\mathrm{s}}|_1$), 2nd root ($\psi^{\mathrm{s}}_j \leftarrow \sqrt{\psi^{\mathrm{s}}_j}$) and 3rd root ($\psi^{\mathrm{s}}_j \leftarrow \sqrt[3]{\psi^{\mathrm{s}}_j}$). We see that both the 2nd and 3rd root power normalization improve the performance of ESC, especially when the dictionary size is large. A possible reason is power normalization can improve the noise robustness of the resulting representation [50]. The results of ESC and AESC in Tables I and II are those with L1 and 2nd root normalization.

VI. Conclusion

In this letter, we have presented approaches to compute exemplar-based representation for music, and shown that high precision rates in tag-based music retrieval is obtained by using the representation to train simple linear SVMs. The proposed approaches are easy to implement, conceptually intuitive, and may be applied to other temporal signals beyond music.

REFERENCES

[1] D. Eck, P. Lamere, T. Bertin-Mahieux, and S. Green, "Automatic generation of social tags for music recommendation," in *Proc. Conf. Advances in Neural Information Processing Systems*, 2007.

[2] E. L. M. Law, L. V. Ahn, R. B. Dannenberg, and M. Crawford, "TagATune: A game for music and sound annotation," in *Proc. Int. Soc. Music Information Retrieval Conf.*, 2007, pp. 361–364.

[3] M. I. Mandel and D. Ellis, "A web-based game for collecting music metadata," *J. New Music Res.*, vol. 37, no. 2, pp. 151–165, 2008.

[4] P. Lamere, "Social tagging and music information retrieval," *J. New Music Res.*, vol. 37, no. 2, pp. 101–114, 2008.

[5] 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 *Proc. Int. Soc. Music Information Retrieval Conf.*, 2011, pp. 85–90.

[6] R. Miotto and N. Orio, "A probabilistic model to combine tags and acoustic similarity for music retrieval," *ACM Trans. Inf. Syst.*, vol. 30, no. 2, pp. 8:1–8:29, 2012.

[7] J.-C. Wang, H.-M. Wang, and S.-K. Jeng, "Playing with tagging: A real-time tagging music player," in *Proc. IEEE. Int. Conf. Acoustics, Speech and Signal Processing*, 2012, pp. 77–80.

[8] D. Turnbull, L. Barrington, D. Torres, and G. Lanckriet, "Semantic annotation and retrieval of music and sound effects," *IEEE Trans. Audio, Speech, Lang, Process.*, vol. 16, no. 2, pp. 467–476, 2008.

[9] M. Levy and M. Sandler, "Music information retrieval using social tags and audio," *IEEE Trans. Multimedia*, vol. 11, pp. 383–395, 2009.

[10] D. Tingle, Y. E. Kim, and D. Turnbull, "Exploring automatic music annotation with acoustically-objective tags," in *Proc. ACM Int. Conf. Multimedia Information Retrieval*, 2010, pp. 55–62.

[11] M. I. Mandel, R. Pascanu, D. Eck, Y. Bengio, L. M. Aiello, R. Schifanella, and F. Menczer, "Contextual tag inference," *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 7S, no. 1, pp. 1547–1556, 2011.

[12] E. Coviello, A. B. Chan, and G. R. G. Lanckriet, "Time series models for semantic music annotation," *IEEE Trans. Audio, Speech, Lang, Process.*, vol. 19, no. 5, pp. 1343–1359, 2011.

[13] K. Ellis, E. Coviello, and G. R. G. Lanckriet, "Semantic annotation and retrieval of music using a bag of systems representation," in *Proc. Int. Soc. Music Information Retrieval Conf.*, 2011, pp. 723–728.

[14] J. Nam, J. Herrera, M. Slaney, and J. Smith, "Learning sparse feature representations for music annotation and retrieval," in *Proc. Int. Soc. Music Information Retrieval Conf.*, 2012, pp. 565–560.

[15] R. Miotto and G. R. G. Lanckriet, "A generative context model for semantic music annotation and retrieval," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 20, no. 4, pp. 1096–1108, 2012.

[16] C.-C. M. Yeh and Y.-H. Yang, "Towards a more efficient sparse coding based audio-word feature extraction system," in *Proc. Asia Pacific Signal and Info. Proc. Association Annu. Summit and Conf.*, 2013.

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

[18] G. Marques, M. A. Domingues, T. Langlois, and F. Gouyon, "Three current issues in music autotagging," in *Proc. Int. Soc. Music Information Retrieval Conf.*, 2011, pp. 795–800.

[19] B. L. Sturm, "A simple method to determine if a music information retrieval system is a "horse"," *IEEE Trans. Multimedia*, vol. 16, no. 6, pp. 1636–1644, 2014.

[20] P. Hamel and D. Eck, "Learning features from music audio with deep belief networks," in *Proc. Int. Soc. Music Information Retrieval Conf.*, 2010, pp. 339–344.

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

[22] M. Henaff, K. Jarrett, K. Kavukcuoglu, and Y. LeCun, "Unsupervised learning of sparse features for scalable audio classification," in *Proc. Int. Soc. Music Information Retrieval Conf.*, 2011, pp. 681–686.

[23] 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*, vol. 16, no. 5, pp. 1188–1200, 2014.

[24] Y. Vaizman, B. McFee, and G. Lanckriet, "Codebook based audio feature representation for music information retrieval," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 22, no. 10, pp. 1483–1493, 2014.

[25] N. Scaringella, G. Zoia, and D. Mlynek, "Automatic genre classification of music content: A survey," *IEEE Signal Process. Mag.*, vol. 23, no. 2, pp. 133–141, 2006.

[26] S. Li, "Content-based audio classification and retrieval using the nearest feature line method," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 8, no. 5, pp. 619–625, 2000.

[27] J. Wright, A. Yang, A. Ganesh, S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 31, no. 2, pp. 210–227, 2009.

[28] J. Wright, Y. Ma, J. Mairal, G. Sapiro, T. S. Huang, and S. Yan, "Sparse representation for computer vision and pattern recognition," *Proc. IEEE*, vol. 98, no. 6, pp. 1031–1044, 2010.

[29] J. F. Gemmeke, T. Virtanen, and A. Hurmalainen, "Exemplar-based sparse representations for noise robust automatic speech recognition," *IEEE Trans. Audio, Speech, Lang, Process.*, vol. 19, no. 7, pp. 2067–2080, 2011.

[30] T. N. Sainath, B. Ramabhadran, D. Nahamoo, D. Kanevsky, D. Van Compernolle, K. Demuynck, J. F. Gemmeke, J. R. Bellegarda, and S. Sundaram, "Exemplar-based processing for speech recognition: An overview," *IEEE Signal Process. Mag,*, vol. 29, no. 6, pp. 98–113, 2012.

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

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

[33] M.-L. Zhang and Z.-H. Zhou, "A review on multi-label learning algorithms," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 8, pp. 1819–1837, 2014.

[34] Y. Bachrach, Y. Finkelstein, R. Gilad-Bachrach, L. Katzir, N. Koenigstein, N. Nice, and U. Paquet, "Speeding up the Xbox recommender system using a Euclidean transformation for inner-product spaces," in *Proc. ACM Conf. Recommender Systems*, 2014, pp. 257–264.

[35] S. S. Chen, D. L. Donoho, and M. A. Saunders, "Atomic decomposition by basis pursuit," *SIAM J. Sci. Comput.*, vol. 20, pp. 33–61, 1998.

[36] D. L. Donoho, "For most large underdetermined systems of linear equations the minimal l_1-norm solution is also the sparsest solution," *Commun. Pure Appl. Math.*, vol. 59, pp. 797–829, 2006.

[37] E. J. Candès, J. K. Romberg, and T. Tao, "Stable signal recovery from incomplete and inaccurate measurements," *Commun. Pure Appli. Math.*, vol. 59, no. 8, pp. 1207–1223, 2006.

[38] A. Y. Yang, S. S. Sastry, A. Ganesh, and Y. Ma, Fast l_1-minimization algorithms and an application in robust face recognition: A review Univ. Urbana-Champaign, Tech. Rep., 2010.

[39] F. Bach, R. Jenatton, J. Mairal, and G. Obozinski, "Optimization with sparsity-inducing penalties," *Found. Trends Mach. Learn.*, vol. 4, pp. 1–106, 2012.

[40] M. Tan, I. W. Tsang, and L. Wang, "Matching pursuit LASSO part I: Sparse recovery over big dictionary," *IEEE Trans. Signal Process.*, vol. 63, no. 3, pp. 727–741, 2015.

[41] R. Tibshirani, "Regression shrinkage and selection via the Lasso," *J. Roy. Statist. Soc.*, vol. 58, pp. 267–288, 1996.

[42] Z. J. Xiang, H. Xu, and P. J. Ramadge, "Learning sparse representations of high dimensional data on large scale dictionaries," in *Proc. Advances in Neural Information Processing Systems*, 2011, pp. 900–908.

[43] Z. J. Xiang, Y. Wang, and P. J. Ramadge, "Screening tests for Lasso problems," *CoRR abs/1405.4897*, 2014.

[44] P.-K. Jao, C.-C. M. Yeh, and Y.-H. Yang, "Modified Lasso screening for audio word-based music classification using large-scale dictionary," in *Proc. IEEE Int. Conf. Acoustics, Speech & Signal Processing*, 2014, pp. 5207–5211.

[45] P. J. Bickel, Y. Ritov, and A. B. Tsybakov, "Simultaneous analysis of Lasso and Dantzig selector," *Ann. Statist.*, vol. 37, no. 4, pp. 1705–1732, 2009.

[46] C.-C. M. Yeh, P.-K. Jao, and Y.-H. Yang, The AWtoolbox for characterizing audio information Academia Sinica, Tech. Rep., 2015.

[47] S. Zubair, F. Yan, and W. Wang, "Dictionary learning based sparse coefficients for audio classification with max and average pooling," *Dig. Signal Process.*, vol. 23, no. 3, pp. 960–970, 2013.

[48] T. Bertin-Mahieux, D. Ellis, B. Whitman, and P. Lamere, "The Million Song Dataset," in *Proc. Int. Soc. Music Information Retrieval Conf.*, 2011, pp. 591–596.

[49] A. Schindler and A. Rauber, "Capturing the temporal domain in echonest features for improved classification effectiveness," in *Int. Workshop on Adaptive Multimedia Retrieval*, 2012.

[50] 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.