A Generative Context Model for Semantic Music Annotation and Retrieval

Riccardo Miotto and Gert Lanckriet

***Abstract*—While a listener may derive semantic associations for audio clips from direct auditory cues (e.g., hearing "bass guitar") as well as from "context" (e.g., inferring "bass guitar" in the context of a "rock" song), most state-of-the-art systems for *automatic* music annotation ignore this context. Indeed, although contextual relationships *correlate* tags, many auto-taggers model tags *independently*. This paper presents a novel, generative approach to improve automatic music annotation by modeling contextual relationships between tags. A Dirichlet mixture model (DMM) is proposed as a second, additional stage in the modeling process, to supplement *any* auto-tagging system that generates a semantic multinomial (SMN) over a vocabulary of tags when annotating a song. For each tag in the vocabulary, a DMM captures the broader context the tag defines by modeling tag co-occurrence patterns in the SMNs of songs associated with the tag. When annotating songs, the DMMs refine SMN annotations by leveraging contextual evidence. Experimental results demonstrate the benefits of combining a variety of auto-taggers with this generative context model. It generally outperforms other approaches to modeling context as well.**

***Index Terms*—Audio annotation and retrieval, context modeling, Dirichlet mixture models, music information retrieval.**

I. Introduction

DURING the last decade, the Internet has reinvented the music industry. Physical media have evolved towards online products and services. As a consequence of this transition, online music corpora have reached a massive scale and are constantly being enriched with new content. This has created a need for music search and discovery technologies that allow users to interact with these extensive collections efficiently and effectively.

Automated semantic annotation of musical content with descriptive tags—keywords or short phrases that capture relevant characteristics of music pieces, ranging from genre and instrumentation, to emotions, usage, etc.—is a core challenge in designing fully functional music retrieval systems [1]. In these systems, semantic tags can be used for keyword search (e.g., searching for "mellow rock songs with acoustic guitar") or example-based retrieval based on high-level semantic representations (e.g., generating playlists based on songs with similar annotations).

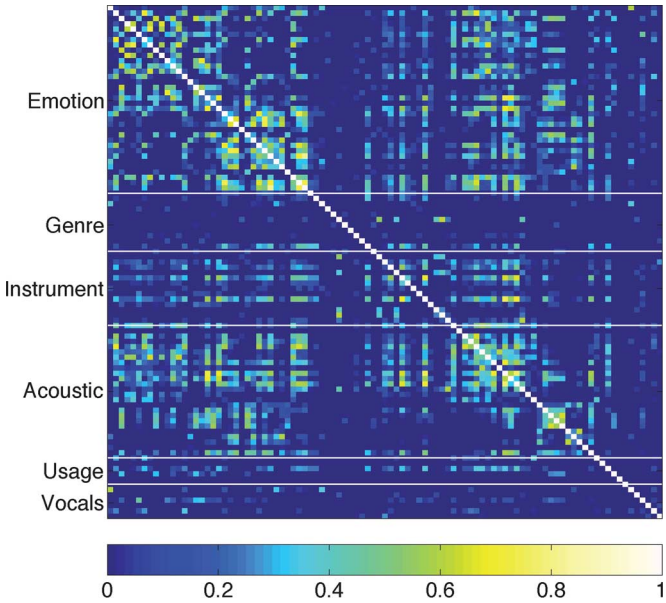


Fig. 1. Tag co-occurrence in the CAL500 dataset. Each row and column corresponds to exactly one tag of the CAL500 vocabulary, ordered by tag category. Entries indicate correlation between the corresponding tags, computed as the Jaccard coefficients, with values ranging from blue (mutually exclusive tags) to white (perfectly correlated tags). Emotion, instrument and acoustic tags exhibit significant co-occurrence with other tags.

To automatically annotate songs with semantic tags, based on audio content, auto-taggers model the characteristic acoustic patterns that are associated with each tag in a vocabulary. State-of-the-art auto taggers are based on discriminative approaches, [e.g., boosting [2] and support vector machines (SVMs) ([3], [4])] as well as generative models (e.g., Gaussian mixture models (GMMs) [5] and the codeword Bernoulli average (CBA) model [6]). Based on these tag models, most auto-taggers generate a vector of tag weights when annotating a new song for music search and retrieval. After normalizing, so its entries sum to one, this vector may be interpreted as a *semantic multinomial* (SMN), i.e., a multinomial probability distribution characterizing the relevance of each tag to a song, as depicted in Fig. 3, on the left-hand side. A song is annotated by selecting the top-ranked tags in its SMN (i.e., sampling the most likely tags from its SMN). To retrieve songs given a tag query, songs in a database are ranked by the tag's probability in their SMN.

Manuscript received October 15, 2010; revised April 03, 2011 and July 17, 2011; accepted September 08, 2011. Date of publication October 17, 2011; date of current version February 10, 2012. This work was supported in part by the National Science Foundation under Grants DMS-MSPA 0625409, CCF-0830535, and IIS-1054960 and in part by the Hellman Fellowship Program. Part of this work was done while R. Miotto was a visiting scholar at U.C. San Diego. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Bryan Pardo.

R. Miotto is with the Department of Information Engineering, University of Padova, Padova 35131, Italy (e-mail: miottori@dei.unipd.it).

G. Lanckriet is with the Department of Electrical and Computer Engineering, University of California at San Diego, La Jolla, CA 92093 USA (e-mail: gert@ece.ucsd.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TASL.2011.2172423

1558-7916/$31.00 © 2011 IEEE

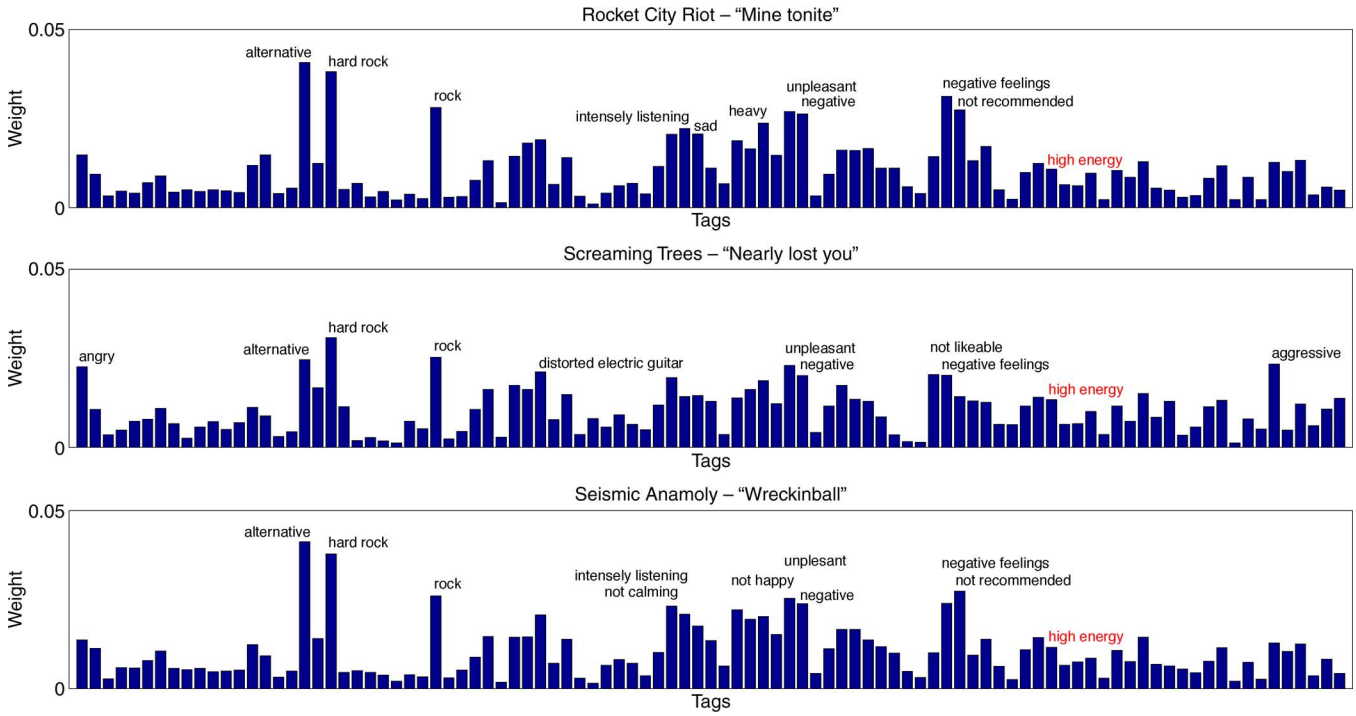


Fig. 2. Examples of semantic multinomials (as predicted by a GMM-based auto-tagger) of three songs associated with the tag "high energy" in the CAL500 dataset. While the content-based auto-tagger fails at predicting the relevance of the tag "high energy," a context model could learn to correct this by observing contextual evidence, e.g., the clear co-occurrence of the tags "alternative" and "hard rock" in "high energy" songs.

TABLE I
TOP-5 CO-OCCURRING TAGS FOR A SAMPLE OF CAL500 TAGS

Tag	**Top-5 co-occurring tags**
hard rock	angry, aggressive vocals, unpleasant, negative feelings, male lead vocals
acoustic guitar	acoustic, not exciting, folk, mellow, light beat
happy emotion	festive, positive feelings, optimistic, carefree, catchy
very danceable song	fast tempo, using at a party, cheerful, awakening happy
going to sleep	calming, tender, mellow, slow beat, low energy

While some semantic associations in music are inspired by direct auditory cues (e.g., hearing a "violin"), others are inferred through contextual relationships (e.g., inferring "cello" and "bassoon," when listening to "orchestral classic music"). These contextual relationships *correlate* tags. This is illustrated in Figs. 1 and 2 and Table I.

Fig. 1 highlights tag co-occurrence patterns (e.g., "rock" songs also tagged as "guitar") in the CAL500 dataset [5], one of the annotated music collections used as *training* data in later experiments (see Section VI-A1 for more details). The pairwise correlation between two tags is computed as the Jaccard coefficient [7], which measures the number of times both tags co-occur, over all songs, normalized by the total number of times the two tags appear.[1] As Fig. 1 indicates, tags in the "Emotion," "Instrument" and "Acoustic" categories correlate significantly with other tags. Additionally, Table I shows the top five co-occurrences for a sample of tags. Fig. 2, on the other hand, shows that the SMNs (generated by a GMM-based auto-tagger) of three CAL500 songs that are associated with the tag "high energy" exhibit similar tag co-occurrences. While the auto-tagger predicts the relevance of the tag "high energy" to be small, based on each song's audio content, this could be corrected by leveraging contextual evidence provided, e.g., by the tags "alternative" and "hard rock," which often co-occur in the SMNs of "high energy" songs.

Auto-tagging systems are expected to benefit from recognizing this context and the tag correlations it induces. For example, the prediction of a tag may be facilitated by the presence or absence of other tags—if a song has been tagged with "drums," the tag "electric guitar" is significantly more likely than "violin." Most state-of-the-art auto-taggers, however, model each semantic tag *independently* and thereby ignore contextual tag correlations.

In this paper, we introduce an additional layer of semantic modeling that supplements existing auto-tagging models by explicitly capturing tag correlations in SMNs. This is shown in Fig. 3. Each tag in the vocabulary is considered to define a broader context that causes multiple, related tags to co-occur in a song's SMN. For each tag, we capture this broader context

[1]In particular, for n_{ij} the number of times tags w_i and w_j co-occur over all songs in the dataset, the Jaccard coefficient c_{ij} is defined as $c_{ij} = n_{ij}/(n_i + n_j - n_{ij})$, where n_i represents the number of songs annotated with the tag w_i. The Jaccard coefficients range between 0 and 1 and are strictly positive if the tags are not mutually exclusive.

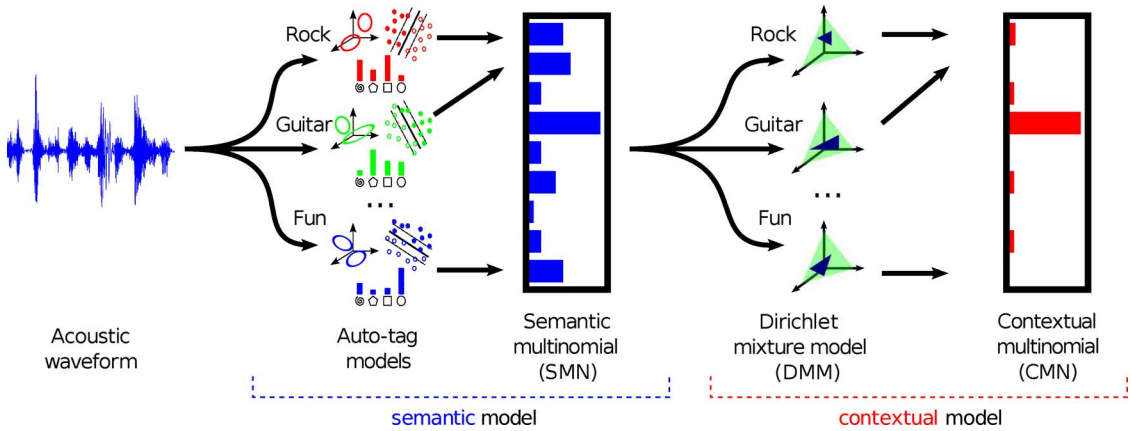


Fig. 3. Overview of the system: DMMs model context by considering co-occurrence patterns between tags in the semantic multinomials.

by estimating a *generative* context model.[2] More specifically, the SMNs predicted for songs associated with the tag are modeled as samples from a Dirichlet mixture model (DMM), i.e., a mixture of Dirichlet distributions [9]. One DMM is estimated for each tag in the vocabulary and models which tags are typically co-occurring or missing for songs associated with that tag. To annotate a new, unlabeled song, the DMMs are used to post process the auto-tagger's SMN, based on semantic context. In particular, a DMM tag model will adjust (i.e., reduce or boost) a tag's SMN weight given contextual evidence provided by other co-occurring tags. This will allow to correct auto-tagger errors in the SMNs, while reinforcing good posterior probability estimates. The resulting tag weights define a *contextual* multinomial (see Fig. 3).

The proposed approach is flexible in that it can be combined with *any* auto-tagger that generates SMNs. As a *generative* model, it also handles weakly labeled data[3] well: unlike discriminative models (e.g., SVMs, boosting, decision trees), which also require reliable negative training examples, generative models only require training examples that have been positively associated with a semantic tag, to estimate class-conditional distributions. Moreover, generative models are better suited at estimating distributions that naturally emerge around relevant contextual tag correlations in the SMNs, while down-weighting irrelevant outliers (e.g., accidental tag co-occurrences), and ranking tags probabilistically for a song (by applying Bayes' rule). Finally, as the Dirichlet distribution is the conjugate prior of the multinomial distribution, it is a natural choice to model the distribution of SMNs generated by an auto-tagger. Estimating a *mixture* of Dirichlet distributions (i.e., a DMM) instead of a single Dirichlet distribution allows to model various contextual relationships within each tag model.

[2]This approach to context modeling is called "generative" as it approaches a multi-class labeling problem by estimating class-conditional distributions (i.e., distributions of SMNs associated with each tag), as opposed to "discriminative" approaches that focus on directly optimizing decision functions and/or probabilities of interest (e.g., discriminant functions to label SMNs with tags). See, e.g., [8] for more details on this nomenclature. We note that using the term "generative" is not intended to suggest that a probabilistic process or graphical model is being proposed that explains and models the generation of music clips from semantic tags, from start to finish.

[3]In weakly labeled data, the presence of a tag implies it applies; the absence of a tag, however, does not guarantee it does not apply.

The remainder of this paper is organized as follows. After a discussion of related work in Section II, Section III describes the music annotation and retrieval problem. In Section IV, we introduce the DMM as a model to capture contextual tag correlations in semantic multinomials. Section V provides an overview of various auto-tagging systems that generate SMNs. Experimental results for combining each of these systems with DMM context models are presented in Sections VI and VII.

II. Related Work

The prohibitive cost of manual labeling of multimedia content (e.g., images, music, movies, etc.) made automatic annotation a major challenge in various fields of research. In the computer vision community, the automatic annotation and retrieval of images has been a topic of ongoing research (see, e.g., [10]–[18]). Recently, Rasiwasia and Vasconcelos [19] proposed a framework that combines object-centric and scene-centric methods to model contextual relationships between visual concepts.

In music information retrieval, auto-tagging has also received a significant amount of attention.[4] The design of auto-taggers has mainly focused on predictive statistical models that capture the acoustic content of songs that are associated with a specific tag (see, e.g., [4]–[6], [21]–[26]), where different tags are often modeled independently. When annotating a new, unlabeled song, most of these semantic tag models allow to compute tag weights, which can be interpreted as a semantic multinomial. Some recent work has started to consider tag correlation ([2], [3], [27]–[29]). Most of this work trains a *discriminative* context model for each tag, based on the tag weights (i.e., SMN) output by the semantic tag models. This context model is often very similar in nature to the semantic model it is "stacked" on top of. For example, Yang *et al.* [27] propose a discriminative approach based on ordinal regression, for both the semantic and the context models (for each tag, training songs are assigned to four groups of decreasing relevance, based on empirically observed tag correlations in the ground truth annotations); Ness *et al.* [3] use support vector machines (SVMs) for semantic and context models, while Bertin-Mahieux *et al.* [2] apply two

[4]As a consequence, in 2008, the "Audio Tag Classification" task has been introduced in the Music Information Retrieval Evaluation eXchange (MIREX) [20].

stages of boosting. Aucouturier *et al.* [28] use a decision tree to refine the result of individual detectors. More recently, Chen *et al.* [29] proposed to estimate not only *word* models, but also *anti-word* models, using GMMs. Word models are regular semantic tag models, modeling the presence of a tag, as in the work of Turnbull *et al.* [5]. Anti-word models, on the other hand, capture acoustic content that is *not* usually associated with a word, by characterizing content that is associated with tags of opposite semantic meaning (i.e., tags that are negatively correlated with the word, on the training set). Predictions from both types of models are combined to obtain the final tag weights. While this approach refines the GMM-based semantic multinomials by explicitly accounting for acoustic patterns that may indicate a tag's absence, it does not provide a holistic contextual model for each tag, to leverage positive, negative and more complex tag correlation patterns in the SMNs.

In this work, we focus on developing a *generative* context model that can be combined with *any* existing auto-tagger that generates SMNs. Compared to the previously proposed discriminative approaches, a generative context model is expected to handle better weakly labeled data and has some other advantages, as discussed before (e.g., the model naturally provides probabilities to rank predictions). Since our context model is not being developed for a specific auto-tagger, we measure its benefit for a variety of auto-taggers, by comparing their performance with and without being combined with the context model. We also explore how other approaches to capturing context ([2], [3], [27]) generalize in combination with auto-taggers other than the ones they were developed for.

III. Music Annotation and Retrieval

This section discusses the related tasks of annotation and retrieval of audio data as a supervised multi-class labeling (SML) problem ([5], [17]). In this setting, each tag in a vocabulary represents a class label, and each song is tagged with multiple labels. Many existing auto-tagging approaches fit within this framework. Section V describes several of them in more detail.

A. Notation

The acoustic content of a song $\mathcal{X}$ is represented as a bag of features, $\mathcal{X}^{\boldsymbol{x}} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_T\}$, where $\boldsymbol{x}_t$ is a vector of audio features extracted from a short snippet (e.g., 20–50 ms) of the audio signal, and T depends on the length of the song.

The dataset $\mathcal{D} = \{\mathcal{X}_d^{\boldsymbol{x}}\}_{d=1}^{|\mathcal{D}|}$ is a collection of $|\mathcal{D}|$ songs. Each of these songs may have been positively associated with one or more tags w_i from a vocabulary $\mathcal{V} = \{w_i\}_{i=1}^{|\mathcal{V}|}$ of size $|\mathcal{V}|$. For each tag w_i, we denote as $\mathcal{D}_{w_i} \subset \mathcal{D}$ the subset of songs that have been positively associated with w_i. The absence of a song in $\mathcal{D}_{w_i}$ does not imply the tag w_i does not apply to the song (i.e., the data may be weakly labeled).

B. Annotation

We treat annotation as a supervised multi-class labeling problem, where each class corresponds to a tag w_i from the vocabulary $\mathcal{V}$ of $|\mathcal{V}|$ unique tags ("rock," "drum," "tender," etc.). In this setting, annotation involves assigning a song to one of a mutually exclusive set of classes, i.e., tags. So, from a modeling perspective, we associate one random variable with each song and its value is the (unique) tag for that song.

First, a set of models (classifiers, class-conditional densities, etc., depending on the type of auto-tagger) over the audio feature space is trained, to recognize the acoustic content associated with each tag w_i in the vocabulary. In most auto-tagging systems, one model is trained per tag and models for different tags are trained *independently*. Then, given a (new) song $\mathcal{X}$, we leverage these models to infer the posterior distribution over tags, $\{w_i\}_{i=1}^{|\mathcal{V}|}$, for that song. As a result, the song is represented as a semantic multinomial (SMN), $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_{|V|})$, on the probability simplex (i.e., $\sum_i \pi_i = 1$ with $\pi_i \geq 0$), where $\pi_i = P(w_i|\mathcal{X}^{\boldsymbol{x}})$ represents the posterior probability of the i^{th} tag for the song $\mathcal{X}$. The SMN provides an ordering of the tags by posterior probability, $P(w_i|\mathcal{X}^{\boldsymbol{x}})$—this, in some sense, reflects the competition between tags, to annotate the song. The maximum of the SMN, i.e., $\arg\max_i P(w_i|\mathcal{X}^{\boldsymbol{x}})$, provides the (unique) class assignment.

For practical annotation purposes, instead of assigning one tag per song, we usually leverage a song's SMN to annotate the song with multiple tags by selecting the, e.g., m tags with the largest posterior probabilities according to the SMN.

C. Retrieval

To retrieve songs given a tag-based query, all songs in a database are ranked based on their relevance to the query and the top-ranked results are returned to the user. More specifically, we determine the relevance of a song $\mathcal{X}$ to a query with tag w_i based on the posterior probability of the tag for that song, i.e., $P(w_i|\mathcal{X}^{\boldsymbol{x}})$. Hence, songs in the database are ranked based on the i^{th} entry, π_i, of their semantic multinomials $\boldsymbol{\pi}$. Although we focus on single-tag queries, this framework easily extends to multiple-tag queries [5].

IV. Dirichlet Mixture as Generative Context Model

Instead of modeling tags independently, in this paper, we recognize contextual tag *correlations* and explicitly model them. We want the model to be 1) generally applicable, so it can be combined with *any* auto-tagger that generates SMNs, 2) compatible with the SML framework proposed in Section III—i.e., generate tag multinomials for annotation and retrieval—, and 3) generative (to handle weakly labeled data and estimate class-conditional distributions around meaningful tag co-occurrence patterns in SMNs, while down-weighting accidental co-occurrence patterns as outliers).

To achieve these objectives, each tag is considered as a "source" of a broader *context*, that causes several related tags to co-occur in a song's SMN, and this context is modeled with a Dirichlet mixture model (DMM), i.e., a mixture of Dirichlet distributions [9]. So, the DMM captures the typical co-occurrence patterns of tags in the SMNs for songs associated with the tag. As the conjugate prior of the multinomial distribution, the Dirichlet distribution is an appropriate model for this generative approach. The SMNs it models could be generated by any algorithm for automatic annotation. This gives rise to the two-level architecture depicted in Fig. 3, applying DMM in tandem with any first-stage auto-tagger generating SMNs. After training the auto-tagger and estimating DMMs for all tags, a new, unlabeled song is tagged as follows. First, the auto-tagger of choice annotates the song with an SMN (left hand side of

Fig. 3). Then, in the right half of Fig. 3, each tag's SMN weight is adjusted based on contextual evidence, i.e., by evaluating the likelihood of co-occurring tags in the SMN based on the DMM for that tag. This maps the song's *semantic* multinomial $\boldsymbol{\pi}$ into a *contextual* multinomial (CMN) $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_{|V|})$, where $\theta_i = P(w_i|\boldsymbol{\pi})$ is the (context adjusted) relevance of the i^{th} tag, given the song's semantic multinomial $\boldsymbol{\pi}$. This new multinomial representation of songs can directly be used in the annotation and retrieval framework presented in Section III, where each song is annotated with the most likely tags according to $\boldsymbol{\theta}$ (i.e., we select the tags showing the largest probability θ_i).

The remainder of this section explains in more detail how the DMMs are estimated (to model context) and then applied (leveraging context) to annotate new songs with a contextual multinomial.

A. Learning DMM Context Models for Each Tag

To model the context induced by a tag w_i, a Dirichlet mixture model is estimated based on the SMNs of all training songs in $\mathcal{D}_{w_i}$ (i.e., all training songs in $\mathcal{D}$ associated with w_i). Note that to annotate a new song, as depicted in Fig. 3, the DMM tag models are applied to the SMN *predicted* by some first-stage auto-tagger. To estimate the DMMs consistently, they are therefore estimated based on the SMNs predicted by the first-stage auto-tagger, for the songs associated with w_i. Moreover, to increase the size and diversity of the training dataset, we will represent each song as a collection of SMNs, by extracting (overlapping) 5-s segments from the song, every 3 s, and annotating each segment with a SMN. This results in a training set of N_{w_i} semantic multinomials, $\{\boldsymbol{\pi}^n\}_{n=1}^{N_{w_i}}$, associated with the tag w_i.

Finally, to better emphasize the peaks in the training SMNs, we consider only the top-h tag weights in each SMN and decrease all others to very low values. For more peaked SMNs, with high kurtosis,[5] h is set to $0.05 \cdot |\mathcal{V}|$, while for more uniform SMNs, with low kurtosis, $h = 0.1 \cdot |\mathcal{V}|$ is used. This is discussed in more detail in Section VII-A.

These SMNs are modeled as samples from a mixture of Dirichlet distributions [30]

$$P(\boldsymbol{\pi}|w_i; \Omega^{w_i}) = \sum_{k=1}^{K} \beta_k^{w_i} \mathrm{Dir}\left(\boldsymbol{\pi}; \boldsymbol{\alpha}_k^{w_i}\right) \tag{1}$$

where K is the number of mixture components and $\Omega^{w_i} = \{\beta_k^{w_i}, \boldsymbol{\alpha}_k^{w_i}\}_{k=1}^{K}$ the model parameters, with $\boldsymbol{\alpha}_k^{w_i}$ the parameters for the k^{th} Dirichlet mixture component and $\beta_k^{w_i}$ the corresponding component weight. The component weights are positive and normalized to sum to 1, i.e., $\sum_k \beta_k^{w_i} = 1$ with $\beta_k^{w_i} \geq 0$. A Dirichlet distribution $\mathrm{Dir}(\boldsymbol{\pi}; \boldsymbol{\alpha})$ with parameters $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_{|V|})$ is given by

$$\mathrm{Dir}(\boldsymbol{\pi}; \boldsymbol{\alpha}) = \frac{\Gamma\left(\sum_{i=1}^{|V|} \alpha_i\right)}{\prod_{i=1}^{|V|} \Gamma(\alpha_i)} \prod_{i=1}^{|V|} (\pi_i)^{\alpha_i - 1} \tag{2}$$

where $\Gamma(\cdot)$ denotes the Gamma function.

[5]Kurtosis provides a measure of the "peakedness" of the distribution, giving an indication whether a SMN is characterized by a few dominant peaks (high kurtosis value) or by a more uniform distribution (low kurtosis value).

Given the training set of SMNs associated with the tag w_i, i.e., $\{\boldsymbol{\pi}^n\}_{n=1}^{N_{w_i}}$, the parameters Ω^{w_i} of the contextual tag model are estimated by adopting the generalized expectation–maximization (GEM) algorithm for maximum-likelihood estimation [31]. GEM is an extension of standard EM and can be applied when the M-step in standard EM is intractable. Just like EM, GEM is an iterative algorithm that iterates between an E-step and a (generalized) M-step. The E-step is identical to the E-step of standard EM: expectations are computed for each mixture component. The (generalized) M-step updates the estimates of the model parameters $\Omega^{w_i} = \{\beta_k^{w_i}, \boldsymbol{\alpha}_k^{w_i}\}_{k=1}^{K}$. Instead of solving for the parameters that maximize the likelihood, which is intractable, this M-step generates a parameter estimate for which the likelihood is higher than the one in the previous iteration. This is known to be sufficient to guarantee convergence of the overall EM procedure [31]. To implement this M-step, we apply the Newton–Raphson algorithm, as in the work of Minka [32] (who applied it for single component Dirichlet distributions). Unlike some other Newton algorithms, the latter is more efficient since it does not to require storing or inverting the Hessian matrix (the update for $\boldsymbol{\alpha}$ in (5) can be computed more efficiently). Details are provided in Algorithm 1 (wherein $\mathbf{1}$ denotes the vector of all ones). Fig. 4 illustrates the complete training procedure.

Algorithm 1 GEM algorithm for DMM

1: **Input**: N semantic multinomials $\{(\pi_1^n, \ldots, \pi_{|V|}^n)\}_{n=1}^{N}$.

2: Randomly initialize DMM parameters $\Omega = \{\beta_k, \boldsymbol{\alpha}_k\}_{k=1}^{K}$.

3: Define

$$\Psi(x) = \frac{\mathrm{d}\log\Gamma(x)}{\mathrm{d}x} \quad \text{and} \quad \Psi'(x) = \frac{\mathrm{d}^2\log\Gamma(x)}{\mathrm{d}x^2}. \tag{3}$$

4: **repeat**

5: {E-step}

6: Compute responsibilities γ_k^n, for $k = 1, \ldots, K$ and $n = 1, \ldots, N$, based on current parameter values

$$\gamma_k^n = \frac{\beta_k \cdot \mathrm{Dir}(\boldsymbol{\pi}^n; \boldsymbol{\alpha}_k)}{\sum_{j=1}^{K} \beta_j \cdot \mathrm{Dir}(\boldsymbol{\pi}^n; \boldsymbol{\alpha}_j)}. \tag{4}$$

7: {M-step}

8: Update DMM parameters. For $k = 1, \ldots, K$ (applying Ψ, Ψ', / and log component-wise)

$$\begin{aligned}
\boldsymbol{g}_k &= N\Psi(\mathbf{1}^T\boldsymbol{\alpha}_k)\mathbf{1} - N\Psi(\boldsymbol{\alpha}_k) + \sum_n \log(\boldsymbol{\pi}^n),\\
\boldsymbol{q}_k &= -N\Psi'(\boldsymbol{\alpha}_k),\\
\boldsymbol{b}_k &= \frac{\mathbf{1}^T(\boldsymbol{g}_k/\boldsymbol{q}_k)}{(N\Psi'(\mathbf{1}^T\boldsymbol{\alpha}_k))^{-1} + \mathbf{1}^T(\mathbf{1}/\boldsymbol{q}_k)} \cdot \mathbf{1},\\
\beta_k^{\mathrm{new}} &= \frac{1}{N}\sum_{n=1}^{N} \gamma_k^n,\\
\boldsymbol{\alpha}_k^{\mathrm{new}} &= \boldsymbol{\alpha}_k^{\mathrm{old}} - \frac{\boldsymbol{g}_k - \boldsymbol{b}_k}{\boldsymbol{q}_k}.
\end{aligned} \tag{5}$$

9: **until** convergence

10: **Output**: DMM parameters $\Omega = \{\beta_k, \boldsymbol{\alpha}_k\}_{k=1}^{K}$.

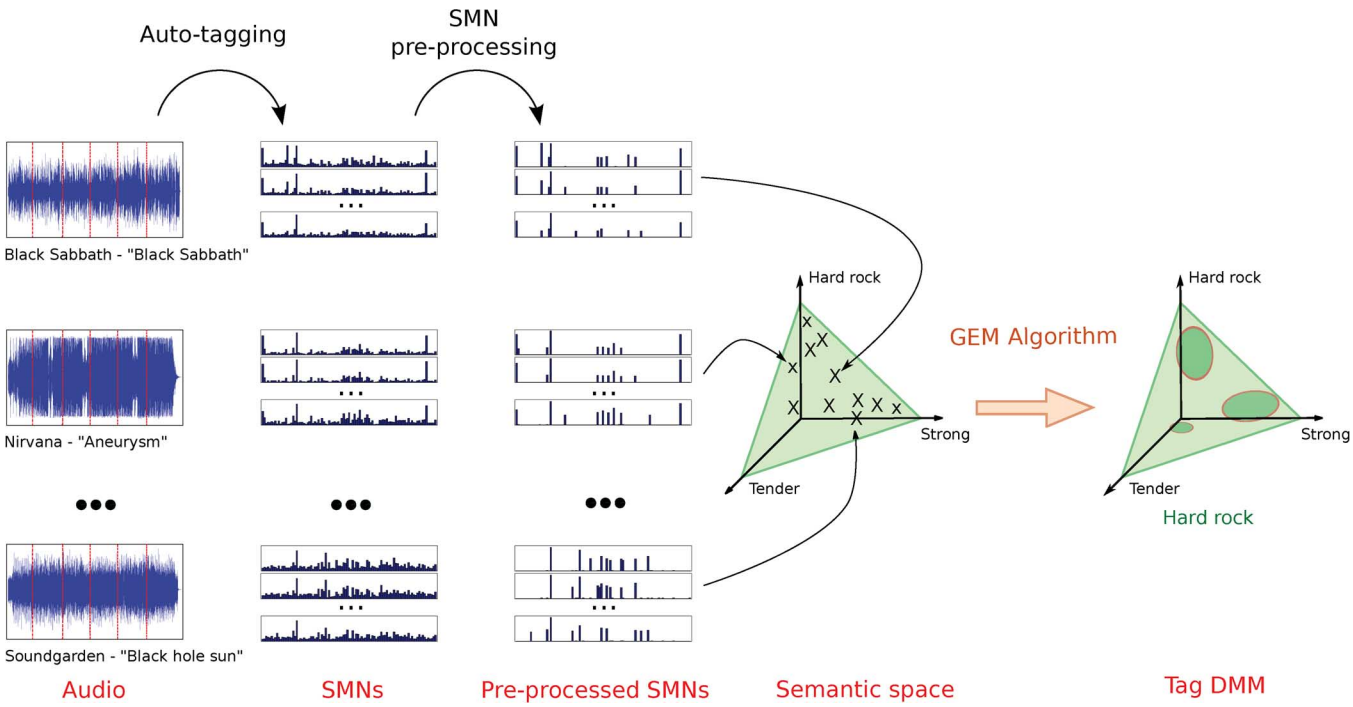


Fig. 4. Learning a DMM context model for the tag "Hard rock." First, 5-s segments, extracted from each "Hard rock" song in the training set (the segments are extracted every 3 s, which, for clarity, is not precisely represented in the figure) are automatically annotated with semantic multinomials. Next, these SMNs are preprocessed, to highlight their peaks, as explained in the text. Finally, the distribution of the SMNs, in the semantic space, is modeled by a Dirichlet mixture model, using a generalized EM algorithm (GEM).

B. Contextual Auto-Tagging

Once the contextual tag models, $P(\boldsymbol{\pi}|w_i;\Omega^{w_i})$, have been estimated for all w_i, $i = 1,\ldots,|\mathcal{V}|$ in the vocabulary $\mathcal{V}$, they can be used to annotate a new song.

First, for an unseen test song $\mathcal{X}$, (overlapping) 3-s segments are extracted from its audio signal, every 2 s. Each segment is annotated with an SMN by the first-stage auto-tagger. This represents the test song as a collection of S semantic multinomials, i.e., $\mathcal{X}^{\boldsymbol{\pi}} = \{\boldsymbol{\pi}^1,\ldots,\boldsymbol{\pi}^S\}$, where $\boldsymbol{\pi}^s = (\pi^s_1,\ldots,\pi^s_{|\mathcal{V}|})$ and S depends on the length of the song. Since we are extracting segments from one song only at this stage (as opposed to extracting them from many during training), we extract slightly shorter segments than from training songs to generate a reasonably large and rich set of SMNs for the test song. Given the set of SMNs representing $\mathcal{X}$, the most relevant tags are the ones with highest posterior probability, computed using Bayes' rule

$$\theta_i = P(w_i|\mathcal{X}^{\boldsymbol{\pi}}) = \frac{P(\mathcal{X}^{\boldsymbol{\pi}}|w_i)P(w_i)}{P(\mathcal{X}^{\boldsymbol{\pi}})} \tag{6}$$

where $P(w_i)$ is the prior of the i^{th} tag and $P(\mathcal{X}^{\boldsymbol{\pi}})$ the song prior. We assume a uniform prior, i.e., $P(w_i) = 1/|\mathcal{V}|$ for $i = 1,\ldots,|\mathcal{V}|$, to promote annotation using a diverse set of tags. The song prior, $P(\mathcal{X}^{\boldsymbol{\pi}})$, is computed as $P(\mathcal{X}^{\boldsymbol{\pi}}) = \sum_{j=1}^{|\mathcal{V}|} P(\mathcal{X}^{\boldsymbol{\pi}}|w_j)P(w_j)$. As in the work of Turnbull *et al.* [5], we estimate the likelihood term of (6), $P(\mathcal{X}^{\boldsymbol{\pi}}|w_i)$, by assuming that song segments are conditionally independent (given w_i) and compensating for the inaccuracy of this naïve Bayes assumption by computing the geometric average of the S segment likelihoods

$$P(\mathcal{X}^{\boldsymbol{\pi}}|w_i) = \left(\prod_{s=1}^{S} P(\boldsymbol{\pi}^s|w_i)\right)^{\frac{1}{S}}. \tag{7}$$

Finally, collecting all posterior probabilities

$$\theta_i = P(w_i|\mathcal{X}^{\boldsymbol{\pi}}) = \frac{\left(\prod_{s=1}^{S} P(\boldsymbol{\pi}^s|w_i)\right)^{\frac{1}{S}}}{\sum_{j=1}^{|\mathcal{V}|}\left(\prod_{s=1}^{S} P(\boldsymbol{\pi}^s|w_j)\right)^{\frac{1}{S}}} \tag{8}$$

provides the *contextual* multinomial $\boldsymbol{\theta}$, which can be used for annotation and retrieval tasks following the approach outlined in Section III.

In Section VII, we will demonstrate that this contextual representation of songs improves annotation and retrieval for a variety of first-stage auto-taggers. First, we overview some auto-taggers that are amenable to DMM context modeling.

V. Auto-Taggers to Compute Semantic Multinomials

We briefly review four state-of-the-art auto-tagging systems, which allow to compute semantic multinomials as described in Section III and, therefore, may be leveraged with the contextual model presented in the previous section.

A. Gaussian Mixture Models (GMM)

This generative model was proposed by Turnbull *et al.* [5]. Each tag w_i, $i = 1,\ldots,|\mathcal{V}|$, in the vocabulary $\mathcal{V}$ is modeled

with a probability distribution $P(\boldsymbol{x}|w_i)$ over the space of audio features $\boldsymbol{x}$, which is a Gaussian mixture model and captures the acoustic patterns that are associated with w_i:

$$P(\boldsymbol{x}|w_i) = \sum_{r=1}^{R} a_r^{w_i} \mathcal{N}\left(\boldsymbol{x}|\boldsymbol{\mu}_r^{w_i}, \boldsymbol{\Sigma}_r^{w_i}\right) \tag{9}$$

where R is the number of mixture components, $\mathcal{N}(\cdot|\boldsymbol{\mu}, \boldsymbol{\Sigma})$ a multivariate Gaussian distribution with mean $\boldsymbol{\mu}$ and (diagonal) covariance matrix $\boldsymbol{\Sigma}$, and $a_r^{w_i}$ the mixing weights. The parameters $\{a_r^{w_i}, \boldsymbol{\mu}_r^{w_i}, \boldsymbol{\Sigma}_r^{w_i}\}_{r=1}^{R}$ of each tag model $P(\boldsymbol{x}|w_i)$ are *estimated* from the audio features of songs that are positively associated with w_i, using an efficient hierarchical expectation–maximization algorithm [5].

Given the audio content $\mathcal{X}^{\boldsymbol{x}} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_T\}$ of a new song $\mathcal{X}$, a Bayes decision rule based on the posterior tag probabilities achieves the minimum probability of error to infer relevant tags

$$\pi_i = P(w_i|\mathcal{X}^{\boldsymbol{x}}) = \frac{P(\mathcal{X}^{\boldsymbol{x}}|w_i)P(w_i)}{P(\mathcal{X}^{\boldsymbol{x}})} \tag{10}$$

where $P(w_i)$ is the prior of the i^{th} tag (assumed to be uniform) and $P(\mathcal{X}^{\boldsymbol{x}})$ the song prior, computed as $P(\mathcal{X}^{\boldsymbol{x}}) = \sum_{j=1}^{|\mathcal{V}|} P(\mathcal{X}^{\boldsymbol{x}}|w_j)P(w_j)$. The likelihood term in (10), $P(\mathcal{X}^{\boldsymbol{x}}|w_i)$, is estimated with the geometric average $\left(\prod_{t=1}^{T} P(\boldsymbol{x}_t|w_i)\right)^{1/T}$. Finally, we have:

$$\pi_i = P(w_i|\mathcal{X}^{\boldsymbol{x}}) = \frac{\left(\prod_{t=1}^{T} P(\boldsymbol{x}_t|w_i)\right)^{\frac{1}{T}}}{\sum_{j=1}^{|\mathcal{V}|}\left(\prod_{t=1}^{T} P(\boldsymbol{x}_t|w_j)\right)^{\frac{1}{T}}}. \tag{11}$$

We run experiments using the code of Turnbull *et al.* [5], estimating song models with 8 and tag models with 16 mixture components.

B. Codeword Bernoulli Average (CBA)

CBA [6] is a probabilistic model to predict the probability that a tag applies to a song based on a vector-quantized representation of the audio features with M codewords. So, rather than representing a song as a bag of audio feature vectors $\boldsymbol{x}_t$, CBA represents a song $\mathcal{X}$ as a vector $\mathbf{n}_{\mathcal{X}}$ of counts of M codewords.

For a song $\mathcal{X}$, the CBA model generates a set of binary random variables $y_i \in \{0,1\}, i = 1, \ldots, |\mathcal{V}|$, which determine whether or not each tag w_i applies to $\mathcal{X}$, in two steps. First, for each tag w_i, $i = 1, \ldots, |\mathcal{V}|$, a codeword $z_i \in \{1, \ldots, M\}$ is selected with probability proportional to the number of times it appears in the song, i.e., proportional to the corresponding entry of $\mathbf{n}_{\mathcal{X}}$. Then, y_i is sampled from a Bernoulli distribution with parameter $\beta_{z_i i}$ (denoting entry (z_i, i) of the parameter matrix $\boldsymbol{\beta}$)

$$\begin{aligned} P(y_i = 1|z_i, \boldsymbol{\beta}) &= \beta_{z_i i}, \\ P(y_i = 0|z_i, \boldsymbol{\beta}) &= 1 - \beta_{z_i i}. \end{aligned} \tag{12}$$

The Bernoulli parameters $\boldsymbol{\beta}$ are estimated from training data using the expectation-maximization algorithm [31].

The probability that a tag w_i applies to an unseen test song $\mathcal{X}$ can be inferred as

$$\pi_i = P(w_i|\mathcal{X}^{\boldsymbol{x}}) = P(y_i = 1|\mathbf{n}_{\mathcal{X}}, \boldsymbol{\beta}) = \frac{1}{T}\sum_{m=1}^{M} (\mathbf{n}_{\mathcal{X}})_m \beta_{mi} \tag{13}$$

where $\mathbf{n}_{\mathcal{X}}$ is the codebook representation of $\mathcal{X}$, and T its total number of features, with $\mathcal{X}^{\boldsymbol{x}} = \{\boldsymbol{x}_t\}_{t=1}^{T}$.

We obtained the authors' code [6] to run our experiments. We modified it so the codebook is constructed using only songs from the training set and set the codebook size $M = 500$.

C. Support Vector Machines (SVM)

Mandel and Ellis use a collection of SVM classifiers for music auto-tagging [4]. An SVM is a discriminative, supervised learning algorithm that learns the maximum margin hyperplane separating two classes of data to construct a binary classifier. For a vocabulary with $|\mathcal{V}|$ tags, an SVM is learned for each tag w_i, to predict the presence or absence of the tag. Each SVM is trained from a collection of positive and negative training examples for the tag it models, using a radial basis kernel.

To tag an unseen test song $\mathcal{X}$, it is classified by each of the $|\mathcal{V}|$ SVMs. Platt scaling [33] is used to convert distance from the separating hyperplane to a probability estimate $\pi_i = P(w_i|\mathcal{X}^{\boldsymbol{x}})$ for each tag w_i.

We implement this model using LibSVM [34], setting the width of the radial basis kernel to 1 and the regularization parameter, C, to 10, after normalizing the data, as in the work of Mandel and Ellis [4].

D. Boosting (BST)

The boosting approach of Eck *et al.* [21] is similar to the previous, SVM-based approach in that it is a discriminative approach that learns a binary classifier for each tag w_i in the vocabulary $\mathcal{V}$, from positive and negative training examples for that tag. More specifically, it constructs a *strong classifier* from a set of simpler classifiers, called *weak learners*, in an iterative way. As weak learners, we use single stumps (i.e., binary thresholding on one feature).

Again, a test song $\mathcal{X}$ is classified by each of the binary classifiers and Platt scaling applied to produce a probability estimate $\pi_i = P(w_i|\mathcal{X}^{\boldsymbol{x}})$ for each tag w_i.

We obtained the authors' code [21] to run our experiments.

VI. Experimental Setup

For two different datasets (one strongly labeled, one weakly labeled), we evaluate how adding context modeling with Dirichlet mixture models affects the performance of the various auto-taggers presented in the previous section, for music annotation and retrieval. We also investigate how adding a DMM-based context model compares to combining these auto-taggers with other, previously proposed context models. In this section, we discuss the datasets, the audio features and the context models considered in these experiments, as well as the metrics to evaluate the results.

A. Music Datasets

For our experiments, we use two different annotated music collections: CAL500 and CAL10k.

1) CAL500 Database: This dataset consists of 502 popular Western songs by as many different artists [5]. Through a controlled survey, each song has been tagged by at least three human annotators using a semantic vocabulary of 149 tags. The vocabulary is diverse and spans genres, instruments, vocal characteristics, acoustic characteristics, emotions, and song usages. The CAL500 dataset provides binary annotations, which are 1 when a tag applies to the song (i.e., at least two subjects voted for the tag) and 0 otherwise. It's a strongly labeled dataset in that both positive (1) and negative (0) associations have been verified. To accurately fit the DMM models, we restrict ourselves to the subset of 97 tags that have at least 30 training songs positively associated with them (11 genre, 14 instrument, 25 acoustic quality, 6 vocal characteristics, 35 emotion, and 6 usage tags).

2) CAL10k Database: CAL10k [35] is a collection of 10 870 songs from 4597 different artists, weakly labeled from a vocabulary composed of 153 "genre" tags and 475 "acoustic tags." Each song is labeled with 2 to 25 tags. The song-tag associations for this dataset have been mined from the Pandora website. As a result, CAL10k is a weakly labeled dataset: while the presence of a tag means musicologists involved with Pandora's Music Genome Project assigned the tag, the absence of a tag does not imply whether it applies or not. Each CAL10k tag has at least 30 example songs positively associated with it.

B. Audio Features

The audio content of CAL500 songs is represented by timbral descriptors, in particular Mel-frequency cepstral coefficients (MFCCs) [36]. MFCCs are a popular feature for content-based music analysis. They summarize the spectral content of a short-time window (e.g., 20–50 ms) of an acoustic waveform by using the discrete cosine transform to decorrelate the bins of a Mel-frequency spectral histogram.

Since we have access to all audio clips of the CAL500 corpus, we extract from the audio signal those features that the respective auto-tagging algorithms (GMM, CBA, SVM, and BST) were originally proposed with. For GMM and CBA, audio segments are represented as a bag of 39-dimensional Delta-MFCC feature vectors. These feature vectors are obtained by extracting half-overlapping, short-time windows (of 23 ms) from the audio signal, computing the first 13 MFCCs for each window, and appending the first and second instantaneous derivatives of the MFCCs (Deltas).

For the boosting algorithm, the audio signal is described as a collection of 20-dimensional MFCCs, to capture timbral aspects, and a series of auto-correlation coefficients (computed for lags spanning from 250 ms to 2000 ms at 10-ms intervals), to describe tempo and pitch. Both sets of features are computed from 100-ms windows that are extracted from the audio signal every 75 ms. Finally, for the SVM-based auto-tagger, the audio signal is described using timbral descriptors and short-term temporal features (to summarize beat, tempo, and rhythmic patterns). The timbral descriptors are obtained as the mean and unwrapped covariance of a clip's 18-dimensional MFCCs, computed from 25-ms windows, extracted every 10 ms. The temporal features are obtained by first combining the Mel frequency bands (of the same 25-ms windows) into low, low-mid, high-mid, and high frequencies, and then modeling the total magnitude in each of these four (large) frequency bands over time (by using a DCT to decorrelate the magnitudes of the Fourier transform of each band).

More details about the audio features used by each of the previous auto-tagging algorithms can be found in the corresponding references, mentioned in Section V. For our experiments, we generally set all parameters to the values reported in those earlier works.

As copyright issues prevent us to obtain all CAL10k songs, we represent their audio content using Echo Nest timbre features (ENTs). This alternative representation can be obtained through the Echo Nest service[6] and does not require the user to own the songs. ENTs are derived from slightly longer windows (generally between 100 and 500 ms). For each window, the Echo Nest service calculates 12 "timbre" features (their exact calculation is a trade secret of the company). Computing the first and second instantaneous derivatives of these features leads to a 36-dimensional Delta-ENT feature vector.

For experiments involving the CAL10k dataset, all auto-tagging algorithms are trained and evaluated based on the timbral Delta-ENT feature vectors only.

C. Models

To evaluate the benefit of context modeling with DMMs, we first combine each of the semantic tag models, described in Section V, with our DMM-based context model, as depicted in Fig. 3. For each combination, we compare the annotation and retrieval accuracy to using the semantic tag models alone, without DMM.

Second, we compare our generative, DMM-based approach to context modeling with several recently proposed discriminative approaches based on SMNs. Most of these discriminative approaches were developed and evaluated in combination with a specific auto-tagger. In our experiments, we provide a more general evaluation. As for the evaluation of DMM, we combine each of these context models with all four semantic tag models and report the resulting annotation and retrieval performance. In particular, we compare with the following approaches:

1) Support Vector Machines (cSVM): Ness *et al.* [3] propose to make tag predictions using SVMs based on semantic multinomials.[7] We refer to this discriminative approach as cSVM (context-level SVMs). To train cSVM, we first re-scale the SMNs so that the minimum tag weight is 0 and the maximum 1 (min-max normalization). We implement cSVM using LibSVM [34].

2) Boosting (cBST): Similar to cSVM, this discriminative approach adopts a set of binary classifiers to predict tags based on semantic multinomials[8] [2]. The classifiers are based on boosting and we refer to this model as cBST (context-level boosting).

[6]http://developer.echonest.com.

[7]In their work, the SMNs are computed by auto-taggers based on SVMs as well.

[8]In the work of Bertin-Mahieux *et al.* [2], the semantic tag models generating the SMNs were also based on boosting, as described in Section V-D.

3) Ordinal Regression (OR): Yang *et al.* [27] also propose a discriminative approach to predict the relevance of a tag, based on semantic multinomials.[9] They model several levels of relevance (as opposed to relevant versus irrelevant in binary classification) by formulating tag prediction as an ordinal regression problem. For each tag w_i, training songs are assigned to one of four ordinal classes of decreasing relevance. The assignment is based on all tags training songs are annotated with and the empirically observed correlations of those tags with w_i (by computing Pearson coefficients over all training data). Moreover, before training a tag model, the training SMNs are preprocessed by removing the tags that are weakly correlated (Pearson coefficient 0.2 or lower) to the target tag and, thus, probably irrelevant for predicting it. Finally, an ordinal regression model is trained for each tag w_i, using the listNet algorithm [37]. This results in $|\mathcal{V}|$ ordinal regression models. Given a new, unseen song, the relevance of each tag in the vocabulary $\mathcal{V}$ is computed based on these models. In our implementation, all the parameters have been set to the values reported in the work by Yang *et al.* [27].

D. Evaluation Metrics for Annotation and Retrieval

Annotation performance is measured following the procedure described by Turnbull *et al.* [5]. Test set songs are annotated with the m most likely tags in their semantic or contextual multinomial. Annotation accuracy is measured by computing precision, recall and F-score for each tag,[10] and then averaging over all tags. Per-tag precision is the probability that the model correctly uses the tag when annotating a song. Per-tag recall is the probability that the model annotates a song that should have been annotated with the tag. F-score is the harmonic mean of precision and recall. Precision, recall and F-score for a tag w are defined as

$$p = \frac{|w_C|}{|w_A|}, \quad r = \frac{|w_C|}{|w_H|}, \quad f = \frac{2}{\frac{1}{p} + \frac{1}{r}} \tag{14}$$

where $|w_H|$ is the number of songs that are annotated with w in the ground truth, $|w_A|$ is the number of songs automatically annotated by the model with the tag w, and $|w_C|$ is the number of times w is correctly used. In case a tag is never selected for annotation, the corresponding precision (that otherwise would be undefined) is set to the tag prior from the training set, which equals the performance of a random classifier.

To evaluate retrieval performance, we rank-order test songs for each single-tag query in our vocabulary, as described in Section III-C. We report top-k precision (Pk), mean average precision (MAP), and area under the receiver operating characteristics curve (AROC), averaged over all the query tags [7]. Pk is the precision when the top-k songs are retrieved, i.e., the fraction true positives in the top-k of the ranking. We consider $k = 3, 5, 10$. MAP averages the precision at each point in the ranking list where a song is correctly retrieved. The ROC curve is a plot of true positive rate versus false positive rate as we move down the ranked list. AROC is computed by integrating the ROC curve. It is upper bounded by 1. Random guessing would result in an AROC of 0.5.

[9]To obtain the SMNs, Yang *et al.* [27] adopt an ordinal regression model that is similar to their context model.

[10]We compute annotation metrics on a *per-tag* basis, as our goal is to build an automatic tagging algorithm with high stability over a wide range of semantic tags. *Per-song* metrics may get artificially inflated if a system consistently annotates songs with a small set of highly frequent tags, while ignoring less common tags.

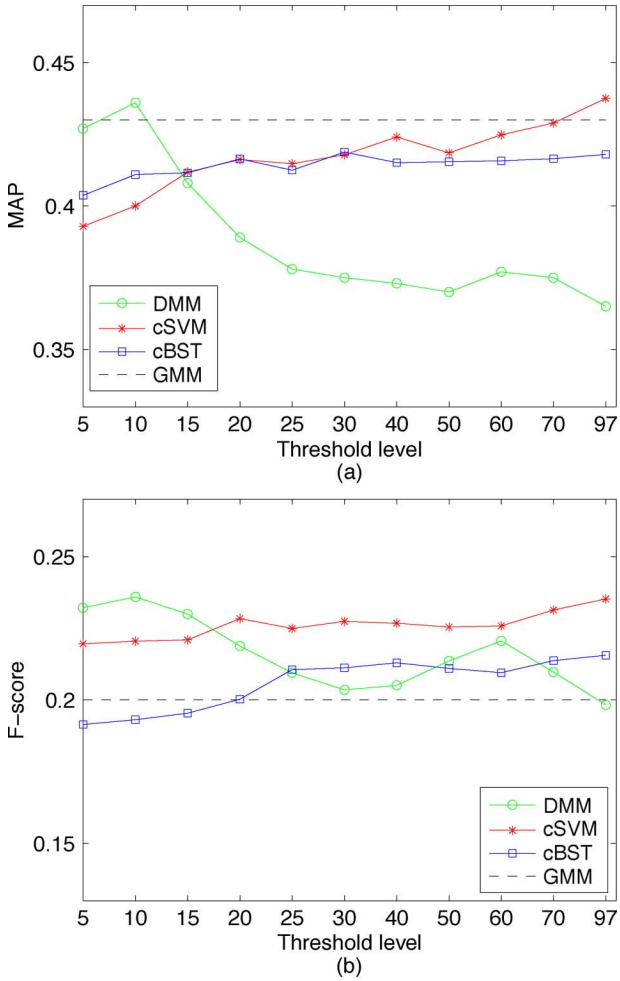


Fig. 5. Effect of preprocessing SMNs before training DMM, cSVM and cBST models. The SMNs are generated by a GMM-based auto-tagger. (a) Retrieval performance (MAP) and (b) annotation performance (F-score). The dashed line represents the performance achieved without context modeling, i.e., using SMNs rather than CMNs.

VII. Results

After discussing the preprocessing of SMNs for DMM in more detail, we provide annotation and retrieval results on the CAL500 and CAL10k datasets.

A. Highlighting the Semantic Peaks for Training DMMs

The goal of the generative, DMM-based approach to context modeling is to estimate class-conditional densities that detect relevant contextual patterns in the SMNs while ignoring accidental tag co-occurrences as noise. Highlighting the peaks in the training SMNs is expected to aid this process by improving the detection of relevant co-occurrence patterns and reducing noise. As mentioned in Section IV-A, this is achieved by limiting each training SMN to its h most relevant peaks and reducing the weights of all other tags to very low values. In particular, for each tag w_i, we compute the kurtosis k_n for each training SMN, $\boldsymbol{\pi}^n$, where $n = 1, \ldots, N_{w_i}$. For peaked SMNs (with $k_n > \bar{k}$),

TABLE II
ANNOTATION AND RETRIEVAL RESULTS FOR CAL500. THE PERFORMANCE IS MEASURED WITH A VARIETY OF METRICS. THE FOUR FIRST-STAGE AUTO-TAGGERS FROM SECTION V (GMM, CBA, BST, SVM) ARE EVALUATED WITHOUT CONTEXT MODEL (INDICATED AS "ALONE"), IN COMBINATION WITH THREE PREVIOUSLY PROPOSED, DISCRIMINATIVE CONTEXT MODELS (CSVM, CBST, AND OR), AND IN COMBINATION WITH OUR GENERATIVE CONTEXT MODEL BASED ON DIRICHLET MIXTURE MODELS (DMMS), RESPECTIVELY. THE BEST RESULTS FOR EACH FIRST-STAGE AUTO-TAGGER ARE INDICATED IN BOLD

Semantic	Context	**Annotation**			**Retrieval**				
Level	Level	Precision	Recall	F-Score	P3	P5	P10	MAP	AROC
GMM	alone	0.405	0.202	0.219	0.456	0.455	0.441	0.433	0.698
	cSVM	0.380	0.225	0.235	0.510	0.490	0.450	0.435	0.686
	cBST	0.412	0.163	0.215	0.465	0.460	0.440	0.429	0.689
	OR	0.390	0.222	0.222	0.464	0.470	0.450	0.434	0.700
	DMM	**0.443**	**0.228**	**0.251**	**0.510**	**0.500**	**0.460**	**0.443**	**0.700**
CBA	alone	0.342	0.217	0.205	0.482	0.462	0.436	0.424	0.687
	cSVM	0.342	0.219	0.224	0.480	0.466	0.440	0.425	0.682
	cBST	0.384	0.154	0.195	0.461	0.453	0.432	0.426	0.693
	OR	0.376	0.209	0.200	**0.502**	0.456	0.437	0.426	0.688
	DMM	**0.400**	**0.223**	**0.235**	0.495	**0.477**	**0.450**	**0.436**	**0.698**
BST	alone	0.430	0.168	0.219	0.508	0.485	0.449	0.440	0.698
	cSVM	0.424	0.240	0.266	0.527	0.513	0.480	0.460	0.713
	cBST	0.440	0.171	0.229	0.481	0.457	0.440	0.430	0.690
	OR	0.438	0.239	0.261	0.551	0.525	0.491	0.470	0.723
	DMM	**0.450**	**0.243**	**0.280**	**0.560**	**0.535**	**0.495**	**0.475**	**0.730**
SVM	alone	0.379	0.230	0.248	0.517	0.500	0.469	0.450	0.707
	cSVM	0.400	**0.240**	0.256	0.473	0.463	0.447	0.440	0.710
	cBST	0.440	0.182	0.242	0.452	0.447	0.436	0.430	0.680
	OR	0.410	0.226	0.234	0.520	0.509	0.480	0.457	0.715
	DMM	**0.475**	0.235	**0.285**	**0.525**	**0.520**	**0.495**	**0.475**	**0.730**

we select $h = 0.05 \cdot |\mathcal{V}|$ tags, while for more uniform SMNs (with $k_n \leq \bar{k}$), we select $h = 0.1 \cdot |\mathcal{V}|$ tags. $\bar{k}$ is computed as the average kurtosis over all training SMNs associated with w_i, i.e., $\bar{k} = (\sum_{n=1}^{N_{w_i}} k_n)/N_{w_i}$.

To evaluate the effectiveness of this preprocessing step, we first consider a simpler approach that selects the top-h peaks for all training SMNs (irrespective of their statistical properties). We vary the threshold h between 5 and $|\mathcal{V}|$ and analyze how this affects the annotation (F-score) and retrieval (MAP) performance on the CAL500 dataset, when learning a DMM context model over GMM-based SMNs. We use five-fold cross validation and annotate songs with the ten most likely tags in the annotation task. For comparison, we also learn cSVM and cBST context models from the same preprocessed training SMNs.

The results, reported in Fig. 5, indicate that DMM-based context modeling benefits from SMN preprocessing. The discriminative models (cBST, cSVM), on the other hand, perform worse when preprocessing training SMNs. Therefore, in the remainder of this section, SMN preprocessing is applied only for DMM training.

The kurtosis-based approach adapts the preprocessing to the "shape" of the SMN. It selects more peaks for more uniform SMNs and less for SMNs with more outspoken peaks. Compared to using the same h for all training SMNs, this improves the F-score further to 0.251 and the MAP to 0.443.

B. Results on CAL500

Annotation and retrieval results for the CAL500 dataset are presented in Table II, using five-fold cross-validation, annotations with the 10 most likely tags, and kurtosis-based SMN preprocessing for estimating DMMs as specified in Section VII-A. The results show that all first-stage (semantic) auto-taggers (GMM, CBA, BST, and SVM) benefit from adding DMM-based context modeling (compared to being used "alone," i.e., without context model), across all performance metrics for both annotation and retrieval. For retrieval, context modeling often more clearly improves the precision-at-k metrics, which focus on the top of the ranked list of retrieval results. For the end user of a semantic music search engine, the quality of the top of the ranking is usually most important.

Most other approaches that leverage contextual tag correlations (i.e., cSVM and OR) improve a majority of the performance metrics, but often not as many as DMM improves: the generative DMM approach is generally superior, irrespective of the underlying semantic model. cBST, on the other hand, frequently performs worse than all other context models and combining it with a first-stage auto-tagger often does not improve performance compared to not adding cBST.

In Table III, we analyze the retrieval results per tag category (emotion, genre, instrument, etc.), for combining a GMM-based semantic model with a variety of context models (qualitatively, the results are similar for other semantic models). For categories of tags that regularly co-occur with other tags in the vocabulary (e.g., "Emotion," "Instrument," and "Acoustic," as can be seen in Fig. 1), most context models improve the semantic model ("alone"), when combined with it. Combining semantic models with the DMM-based context model provides the best results, across all metrics. Categories of tags that co-occur

TABLE III
RETRIEVAL RESULTS PER TAG CATEGORY, FOR CAL500. SEMANTIC MULTINOMIALS ARE COMPUTED BY A GMM-BASED AUTO-TAGGER AND EVALUATED WITHOUT CONTEXT MODEL ("ALONE"), AND IN COMBINATION WITH A VARIETY OF CONTEXT MODELS (CSVM, CBST, OR, AND DMM), RESPECTIVELY. THE BEST RESULTS FOR EACH CATEGORY ARE INDICATED IN BOLD

Category	# Tags	Model	P5	P10	MAP
Emotion	35	alone	0.513	0.506	0.477
		cSVM	0.539	0.510	0.480
		cBST	0.538	0.509	0.477
		OR	0.554	0.515	0.490
		DMM	**0.560**	**0.530**	**0.492**
Genre	11	alone	0.367	0.325	**0.355**
		cSVM	**0.392**	**0.336**	0.350
		cBST	0.363	0.330	0.344
		OR	0.360	0.316	0.339
		DMM	0.360	0.330	0.340
Instrument	14	alone	0.460	0.431	0.441
		cSVM	0.480	0.450	0.450
		cBST	0.460	0.444	0.442
		OR	0.490	0.450	0.448
		DMM	**0.495**	**0.458**	**0.460**
Usage	6	alone	0.253	0.233	0.258
		cSVM	0.250	0.220	0.235
		cBST	0.253	0.190	0.239
		OR	0.226	0.216	0.252
		DMM	**0.280**	**0.270**	**0.282**
Acoustic	25	alone	0.508	0.501	0.472
		cSVM	0.548	0.510	0.486
		cBST	0.524	0.503	0.477
		OR	0.561	0.524	0.500
		DMM	**0.566**	**0.533**	**0.500**
Vocals	6	alone	0.253	0.240	0.261
		cSVM	0.233	0.220	0.245
		cBST	0.273	0.246	0.263
		OR	0.280	**0.265**	0.270
		DMM	**0.287**	0.260	**0.275**

less frequently with other tags (e.g., the "Usage" and "Vocals" categories) still clearly benefit from DMM-based context modeling. Adding other context models provides mixed results and improves performance less than adding DMM. This indicates that even only few co-occurrence patterns in the training data provide enough "detectable" information for the generative, DMM-based approach to improve auto-tagging performance, as opposed to other approaches which do not seem to pick up much contextual information. For the "Genre" category, co-occurrence patterns with other tags are not frequently observed in the CAL500 dataset, as shown in Fig. 1. For this category, adding any of the contextual models leads to mixed results and improvements, if any, are small. A single outlier is cSVM significantly improving P5. We do not have a good explanation for the latter.

Table IV shows the top-10 retrieval results for the query "piano," both for GMM and the combination GMM-DMM.

TABLE IV
TOP-10 RETRIEVED SONGS FOR "PIANO." SONGS WITH PIANO ARE MARKED IN BOLD

Rank	GMM	
1	Yakshi	Chandra
2	Bread	If
3	**Charlie Rich**	**Behind closed doors**
4	Grateful Dead	High time
5	Queen	We will rock you
6	Wes Montgomery	Bumpin
7	Curandero	Aras
8	**Tim Buckley**	**Morning glory**
9	**Bonnie Tyler**	**Total eclipse of the heart**
10	**George Harrison**	**All things must pass**
Rank	GMM-DMM	
1	**Tim Buckley**	**Morning glory**
2	**Charlie Rich**	**Behind closed doors**
3	Queen	We will rock you
4	**George Harrison**	**All things must pass**
5	**Bonnie Tyler**	**Total eclipse of the heart**
6	Bread	If
7	Wes Montgomery	Bumpin
8	**Steely Dan**	**Rikki don't lose that number**
9	**Mazzy Star**	**Fade into you**
10	**Carpenters**	**Rainy days and mondays**

Finally, Table V reports automatic 10-tag annotations for some songs from the CAL500 collection, with GMM and GMM-DMM.

C. Results on CAL10k

To analyze the previous systems when dealing with weakly labeled data, we train them on the CAL10k dataset. Given the weakly labeled nature of this corpus, it is not well suited to evaluate performance. Indeed, since the absence of a tag does not imply it is not applicable to a song, true negative and false positive predictions cannot be reliably verified. Therefore, this experiment is conducted by training models on CAL10k and evaluating them, reliably, on the (strongly labeled) CAL500 dataset. The tags considered are the 55 tags that the CAL10k and the CAL500 corpus have in common (spanning 22 genres and 33 acoustic qualities). Because of the smaller size of the vocabulary, we automatically annotate each song with 5 (instead of 10) tags, in the annotation task. Again, kurtosis-based SMN preprocessing is applied for estimating DMMs (as specified in Section VII-A). As mentioned in Section VI-B, audio clips are represented using Delta-ENT feature vectors for all experiments involving CAL10k, to accommodate copyright issues.

Annotation and retrieval results are reported in Table VI. The performance obtained after adding a DMM-based context model is better than for adding any of the other context models, for all metrics. In fact, none of the other approaches to modeling context is really convincing: half or more of the performance metrics in Table VI decreases when adding a cSVM, cBST, or OR context model to a first-stage auto-tagger. These three discriminative approaches are clearly suffering from the weak labeling of the training data. The generative

TABLE V
AUTOMATIC 10-TAG ANNOTATIONS FOR DIFFERENT SONGS. CAL500 GROUND TRUTH ANNOTATIONS ARE MARKED IN BOLD

Jerry Lee Lewis "Great balls of fire"	
GMM	calming, pleasant, folk, backing vocals, female lead vocals, **not angry**, not exciting, constant energy level, undanceable, **acoustic**
GMM-DMM	**lighthearted**, **emotional**, pleasant, **positive**, folk, **male lead vocals**, **not angry**, **like**, **acoustic**, vocal harmonies
Alicia Keys "Fallin"	
GMM	**emotional**, **tender**, **female lead vocals**, **piano**, acoustic guitar, slow, light beat, not recommended, **undanceable**, **vocal harmonies**
GMM-DMM	**emotional**, **pleasant**, **female lead vocals**, **piano**, **catchy**, **like**, **recommend**, **acoustic**, **emotional vocals**, **vocal harmonies**
Lynyrd Skynyrd "Sweet home Alabama"	
GMM	classic rock, rock, drum set, depressed, not happy, heavy, negative, negative feelings, not recommend, **undanceable**
GMM-DMM	alternative, classic rock, **country**, **piano**, drum set, **clean electric guitar**, **heavy**, not touching, **driving**, hanging with friends
Bryan Adams "Cuts like a knife"	
GMM	angry, alternative, hard rock, **rock**, distorted electric guitar, **not calming**, unpleasant, **negative**, **not likeable**, **negative feelings**
GMM-DMM	angry, alternative, hard rock, **rock**, **bass**, distorted electric guitar, **not calming**, **not mellow**, unpleasant, **heavy**
Glenn Miller "In the mood"	
GMM	romantic, sad, touching, soft rock, piano, light, negative, not likeable, negative feelings, undanceable
GMM-DMM	**emotional**, **positive**, soft rock, **jazz**, female lead vocals, piano, **catchy**, changing energy, recommend, high pitched

TABLE VI
ANNOTATION AND RETRIEVAL RESULTS FOR TRAINING ON CAL10K AND EVALUATING PERFORMANCE ON CAL500, FOR THE 55 TAGS IN COMMON BETWEEN BOTH DATASETS. GMM, CBA, BST, AND SVM ARE EVALUATED WITHOUT CONTEXT MODEL ("ALONE"), AND IN COMBINATION WITH CSVM, CBST, OR, AND DMM, RESPECTIVELY. THE BEST RESULTS FOR EACH FIRST-STAGE (SEMANTIC) AUTO-TAGGER ARE INDICATED IN BOLD

Semantic Level	Context Level	**Annotation** F-Score	**Retrieval** P10	MAP	AROC
GMM	alone	0.217	0.325	0.352	0.729
	cSVM	0.185	0.319	0.349	0.718
	cBST	0.200	0.311	0.344	0.716
	OR	0.171	0.283	0.330	0.700
	DMM	**0.234**	**0.331**	**0.361**	**0.730**
CBA	alone	0.145	0.268	0.315	0.685
	cSVM	0.151	0.280	0.316	0.680
	cBST	0.155	0.278	0.314	0.679
	OR	0.130	0.245	0.290	0.650
	DMM	**0.180**	**0.287**	**0.324**	**0.689**
BST	alone	0.179	0.297	0.333	0.706
	cSVM	0.175	0.321	0.353	0.728
	cBST	0.152	0.292	0.325	0.697
	OR	0.150	0.275	0.318	0.670
	DMM	**0.195**	**0.325**	**0.359**	**0.732**
SVM	alone	0.200	0.300	0.355	0.720
	cSVM	0.190	0.311	0.356	0.725
	cBST	0.200	0.290	0.335	0.690
	OR	0.170	0.290	0.334	0.697
	DMM	**0.220**	**0.315**	**0.371**	**0.730**

DMM-based approach, on the other hand, improves annotation and retrieval performance for all reported metrics and first-stage auto-taggers, compared to using the semantic level "alone," without DMM. Also, at the semantic level, it is observed that the GMM-based generative approach outperforms all other first-stage auto-taggers, for all annotation and retrieval metrics. Finally, combining generative models at the semantic and the contextual level, the tandem GMM-DMM results in the best overall performance for training on this weakly labeled corpus.

VIII. CONCLUSION

In this paper, we have proposed a novel, generative approach to modeling contextual relationships between tags to improve automatic music tagging. In particular, for each tag, we estimate a Dirichlet mixture model to capture typical tag co-occurrence patterns (i.e., "context") in semantic multinomials that songs associated with that tag have been annotated with. This results in a two-step approach: in a first stage, an existing auto-tagging system that allows to annotate a song with a semantic multinomial is applied; in a second stage, the resulting semantic multinomial is refined by a stage of Dirichlet mixture models, based on contextual evidence. This results in a set of posterior tag probabilities that provides a contextual description of the song, i.e., a contextual multinomial. The generative character of this approach makes it well suited for weakly labeled datasets and naturally allows us to rank tags probabilistically for a song.

Experimental results demonstrate that modeling context with DMMs improves performance when combined with a variety of first-stage (semantic) auto-taggers, compared to using the first-stage auto-taggers alone. Moreover, the generative DMM-based approach generally outperforms other, discriminative approaches to modeling context. Its superiority is more outspoken when training on weakly labeled data. Examining the performance per tag category reveals that DMM-based context modeling most significantly benefits those categories of tags that have been empirically observed to frequently co-occur with other tags.

Finally, modeling patterns at the SMN level has the advantage of working with a representation that is independent of the low-level audio feature representation and the actual semantic auto-tagging model used. Even more, SMNs obtained from different auto-taggers or with different audio features may be concatenated, re-normalized, and modeled, once again, as a sample

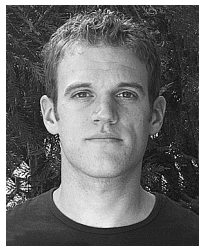
from a (larger) Dirichlet mixture model. Future work will explore this approach as a way to integrate predictions from different auto-taggers or based on different audio features, at different time resolutions.

ACKNOWLEDGMENT

The authors would like to thank the editor and reviewers for their constructive comments. R. Miotto would like to thank M. Mandel for assistance with the implementation of the SVM auto-tagging algorithm [4], T. Bertin-Mahieux and M. Hoffman for providing code for the boosting [2] and CBA [6] algorithms, respectively, and L. Barrington, E. Coviello, N. Orio, and N. Rasiwasia for helpful discussions and support.

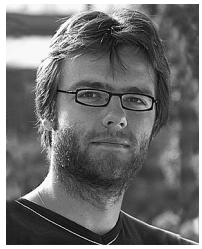
REFERENCES

[1] M. Goto and K. Hirata, "Recent studies on music information processing," *Acoust. Sci. Technol.*, vol. 25, no. 4, pp. 419–425, 2004.
[2] T. Bertin-Mahieux, D. Eck, F. Maillet, and P. Lamere, "Autotagger: A model for predicting social tags from acoustic features on large music databases," *J. New Music Res.*, vol. 37, no. 2, pp. 115–135, 2008.
[3] S. Ness, A. Theocharis, G. Tzanetakis, and L. Martins, "Improving automatic music tag annotation using stacked generalization of probabilistic SVM outputs," in *Proc. ACM MULTIMEDIA*, 2009, pp. 705–708.
[4] M. Mandel and D. Ellis, "Multiple-instance learning for music information retrieval," in *Proc. ISMIR*, 2008, pp. 577–582.
[5] 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, Feb. 2008.
[6] M. Hoffman, D. Blei, and P. Cook, "Easy as CBA: A simple probabilistic model for tagging music," in *Proc. ISMIR*, 2009, pp. 369–374.
[7] C. Manning, P. Raghavan, and H. Schtze, *Introduction to Information Retrieval*. Cambridge, U.K.: Cambridge Univ. Press, 2008.
[8] A. Ng and M. Jordan, "On discriminative vs. generative classifiers: A comparison of logistic regression and naive Bayes," in *Proc. NIPS 14*, 2002, pp. 841–848.
[9] C. Bishop, *Pattern Recognition and Machine Learning*. New York: Springer, 2006.
[10] K. Barnard and D. Forsyth, "Learning the semantics of words and pictures," in *Proc. ICCV*, 2001, vol. 2, pp. 408–415.
[11] P. Duygulu, K. Barnard, N. de Freitas, and D. Forsyth, "Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary," in *Proc. ECCV*, 2002, pp. 349–354.
[12] J. Li and J. Wang, "Automatic linguistic indexing of pictures by a statistical modeling approach," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 9, pp. 1075–1088, Sep. 2003.
[13] D. Blei and M. Jordan, "Modeling annotated data," in *Proc. ACM SIGIR*, 2003, pp. 127–134.
[14] S. Feng, R. Manmatha, and V. Lavrenko, "Multiple Bernoulli relevance models for image and video annotation," in *Proc. IEEE CVPR*, 2004, pp. 1002–1009.
[15] P. Carbonetto, N. de Freitas, and K. Barnard, "A statistical model for general contextual object recognition," in *Proc. ECCV*, 2004, pp. 350–362.
[16] N. Vasconcelos, "From pixels to semantic spaces: Advances in content-based image retrieval," *IEEE Comput.*, vol. 40, no. 7, pp. 20–26, Jul. 2007.
[17] G. Carneiro, A. Chan, P. Moreno, and N. Vasconcelos, "Supervised learning of semantic classes for image annotation and retrieval," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 3, pp. 394–410, Mar. 2007.
[18] C. Tsai and C. Hung, "Automatically annotating images with keywords: A review of image annotation systems," *Recent Patents Comput. Sci.*, vol. 1, pp. 55–68, 2008.
[19] N. Rasiwasia and N. Vasconcelos, "Holistic context modeling using semantic co-occurences," in *Proc. IEEE CVPR*, 2009, pp. 1889–1895.
[20] J. Downie, "The music information retrieval evaluation exchange (2005–2007): A window into music information retrieval research," *Acoust. Sci. Technol.*, vol. 29, no. 4, pp. 247–255, 2008.
[21] D. Eck, P. Lamere, T. Bertin-Mahieux, and S. Green, "Automatic generation of social tags for music recommendation," in *Proc. NIPS 20*, 2007, pp. 385–392.
[22] B. Whitman and R. Rifkin, "Musical query-by-description as a multiclass learning problem," in *Proc. IEEE MMSP*, 2002, pp. 153–156.
[23] M. Slaney, "Semantic-audio retrieval," in *Proc. IEEE ICASSP*, 2002, pp. 4108–4111.
[24] B. Whitman and D. Ellis, "Automatic record reviews," in *Proc. ISMIR*, 2004, pp. 86–93.
[25] G. Tzanetakis and P. Cook, "Musical genre classification of audio signals," *IEEE Trans. Speech Audio Process.*, vol. 10, no. 5, pp. 293–302, Jul. 2002.
[26] E. Pampalk, A. Flexer, and G. Widmer, "Improvements of audio-based music similarity and genre classificaton," in *Proc. ISMIR*, 2005, pp. 628–633.
[27] Y. Yang, Y. Lin, A. Lee, and H. Chen, "Improving musical concept detection by ordinal regression and context fusion," in *Proc. ISMIR*, 2009, pp. 147–152.
[28] J. Aucouturier, F. Pachet, P. Roy, and A. Beurivè, "Signal + context = better classification," in *Proc. ISMIR*, 2007, pp. 425–430.
[29] Z. Chen and J. Jang, "On the use of anti-word models for audio music annotation and retrieval," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 17, no. 8, pp. 1547–1556, Nov. 2009.
[30] R. Miotto, L. Barrington, and G. Lanckriet, "Improving auto-tagging by modeling semantic co-occurrences," in *Proc. ISMIR*, 2010, pp. 297–302.
[31] A. Dempster, N. Laird, and D. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *J. R. Statist. Soc. B*, vol. 39, no. 1, pp. 1–38, 1977.
[32] T. Minka, "Estimating a Dirichlet distribution," 2009 [Online]. Available: http://research.microsoft.com/en-us/um/people/minka/papers/dirichlet/
[33] J. Platt, "Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods," in *Advances in Large Margin Classifiers*. Cambridge, MA: MIT Press, 1999, pp. 61–74.
[34] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," 2001 [Online]. Available: http://www.csie.ntu.edu.tw/~cjlin/libsvm
[35] D. Tingle, Y. Kim, and D. Turnbull, "Exploring automatic music annotation with 'acoustically objective' tags," in *Proc. ACM ICMR*, 2010, pp. 55–61.
[36] B. Logan, "Mel frequency cepstral coefficients for music modeling," in *Proc. ISMIR*, 2000.
[37] F. Xia, T. Liu, J. Wang, W. Zhang, and H. Li, "Listwise approach to learning to rank: Theory and algorithm," in *Proc. ICML*, 2008, pp. 1192–1199.



Riccardo Miotto received the B.S. and M.S. degrees in computer science engineering and the Ph.D. degree in information engineering from the University of Padova, Padova, Italy, in 2004, 2006, and 2011, respectively.

In 2007, he joined the Information Management Systems Research Group, Department of Information Engineering, University of Padova, where he is currently holding a post-doctoral research position. He was a visiting student at the University of Aberdeen, Aberdeen, U.K., in 2005, and a visiting scholar at the Computer Audition Laboratory, Department of Electrical and Computer Engineering, University of California, San Diego, in 2009. His research interests include (music) information retrieval, machine learning applied to computer audition, and data mining.



Gert Lanckriet received the M.S. degree in electrical engineering from the Katholieke Universiteit Leuven, Leuven, Belgium, in 2000 and the M.S. and Ph.D. degrees in electrical engineering and computer science from the University of California, Berkeley, in 2001 and 2005, respectively.

In 2005, he joined the Department of Electrical and Computer Engineering, University of California, San Diego, where he heads the Computer Audition Laboratory. His research focuses on the interplay of convex optimization, machine learning, and signal processing, with applications in computer audition and music information retrieval.

Prof. Lanckriet was awarded the SIAM Optimization Prize in 2008 and is the recipient of a Hellman Fellowship, an IBM Faculty Award, an NSF CAREER Award and an Alfred P. Sloan Foundation Research Fellowship. In 2011, *MIT Technology Review* named him one of the 35 top young technology innovators in the world (TR35).