A Probabilistic Model to Combine Tags and Acoustic Similarity for Music Retrieval

RICCARDO MIOTTO and NICOLA ORIO, University of Padova

The rise of the Internet has led the music industry to a transition from physical media to online products and services. As a consequence, current online music collections store millions of songs and are constantly being enriched with new content. This has created a need for music technologies that allow users to interact with these extensive collections efficiently and effectively. Music search and discovery may be carried out using tags, matching user interests and exploiting content-based acoustic similarity. One major issue in music information retrieval is how to combine such noisy and heterogeneous information sources in order to improve retrieval effectiveness. With this aim in mind, the article explores a novel music retrieval framework based on combining tags and acoustic similarity through a probabilistic graph-based representation of a collection of songs. The retrieval function highlights the path across the graph that most likely *observes* a user query and is used to improve state-of-the-art music search and discovery engines by delivering more relevant ranking lists. Indeed, by means of an empirical evaluation, we show how the proposed approach leads to better performances than retrieval strategies which rank songs according to individual information sources alone or which use a combination of them.

Categories and Subject Descriptors: H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*Search process*; H.4.0 [**Information Systems Applications**]: General

General Terms: Design, Algorithms, Experimentation

Additional Key Words and Phrases: Music information retrieval, tags, acoustic similarity, graph structure, probabilistic model, music discovery

ACM Reference Format:
Miotto R. and Orio N. 2012. A probabilistic model to combine tags and acoustic similarity for music retrieval. ACM Trans. Inf. Syst. 30, 2, Article 8 (May 2012), 29 pages.
DOI = 10.1145/2180868.2180870 http://doi.acm.org/10.1145/2180868.2180870

1. INTRODUCTION

The increasing availability of multimedia user-generated content, such as Flickr[1] for photos, MySpace[2] for music, and YouTube[3] for videos, requires a parallel increase in the development of technologies to search and retrieve this content. A common approach is to exploit information retrieval techniques based on textual descriptors. These can be provided by the user uploading the digital item in the form of free text or

[1] http://www.flickr.com/
[2] http://www.myspace.com/
[3] http://www.youtube.com/

Authors' address: R. Miotto (corresponding author), Department of Information Engineering, University of Padova, via Gradenigo, 6/B, 35131, Padua, Italy; email: miotto.r@gmail.com; N. Orio, Department of Cultural Heritage, University of Padova, Piazza Capitaniato, 7, 35139, Padua, Italy; email: orio@dei.unipd.it.

DOI 10.1145/2180868.2180870 http://doi.acm.org/10.1145/2180868.2180870

of keywords belonging to predefined categories and by users accessing the digital item in the form of comments, ratings, and *tags*.

Tags are short phrases of one to about three words aimed at describing digital items; in the case of music, they may define the genre, the instrumentation, the evoked mood, the use a song is particularly suitable for, and so on. Tags are important because they facilitate the interaction between users and music technologies; in fact, typical users of music engines are more comfortable in expressing their requirements with descriptive words rather than using musical terms (e.g., note and chord names, time signatures, musical forms). Listeners extensively use tags to obtain playlist and recommendations, particularly when using a *music discovery* system. In this scenario, a user is not looking for a specific song or artist, but may have some general criteria that he wishes to satisfy. Examples of such criteria may be "*pop* songs with *high energy* that help *driving* at night" as well as "songs similar to *Yesterday, Beatles*". Common commercial music discovery systems are Last.fm[4] and Pandora[5]. Both use human-based annotations to describe songs, but while Pandora hires expert musicologists to annotate the songs, Last.fm relies on tags provided by final users.

However, the human-based annotation process does not scale well with present-day music collections, where a large amount of new songs are released daily. This problem is particularly relevant in the case of little known artists, whose songs are initially provided without semantic descriptions and can hardly be included in any search algorithms before being tagged. This is among the several factors that may lead to the phenomenon of the *long tail* of songs and artists which are not included (or are included after a long time) in the recommendation lists delivered by many state-of-the-art music engines [Celma 2008]. For this reason, several automatic approaches have been introduced in order to speed up the music tagging process, such as the content-based autotaggers, the propagation of tags from similar songs, or the analysis of users' social behavior [Turnbull et al. 2008a].

Nevertheless, state-of-the-art systems that automatically collect tags can lead to noisy representations, and this may negatively affect the effectiveness of retrieval algorithms. For instance, content-based tagging systems need highly reliable training data (i.e., verified tag-song associations) to obtain effective models for predicting the tags of previously unseen songs. Additionally, these models sometimes have to handle robustness issues such as parameter overfitting or term normalization, which are typically due to data sparseness. The use of social behavior to obtain tags may also result in poor descriptors. Social tags are sometimes referred to as the *wisdom of the crowd* since they are assigned to the songs by a large number of nonexpert humans. However, if a vocabulary of tags is not defined in advance, a human user of these systems can annotate songs without restrictions. Therefore, tags can contain typos (e.g., "classickjazz", "mellou"), be redundant (e.g., "hip-hop" - "hiphop", "hard rock" - "hardrock" - "harder rock"), or be simply not useful for retrieval purposes (e.g., "favorite artist", "mistagged").

Alternatively, the retrieval mechanism can be carried out directly on the acoustic content of the songs, without relying on a high-level semantic representation. This was the initial and typical approach of Music Information Retrieval (MIR) researchers, where audio and music processing is applied to compute acoustic similarity relationships between songs of a collection. Such similarity connections can then be used to rank songs in a content-based music retrieval system [Casey et al. 2008b]. However, in music applications the problem of selecting an optimal similarity measure is difficult because of the intrinsic subjectivity of the task: users may not consistently agree

[4] `http://www.last.fm/`

[5] `http://www.pandora.com/`

upon whether or to what degree a pair of songs or artists are acoustically similar. For instance, the perceived similarity between two songs can be due to external factors, such as being used in the same movie or aired in the same period of time. In this case, tags can be used to leverage acoustic similarity because they contextualize a song, for instance, describing its genre or a geographical area related to it. Hence, both approaches have some limitations when taken singularly; a combined use of them may instead lead to a reduction of such disadvantages.

This article presents a general model for searching songs in a music collection where both content-based acoustic similarity and autotags[6] are combined together in a probabilistic retrieval framework. That is, the goal is to partially overcome the limitations of these two automatic strategies by applying them together in a unified graph-based representation. In particular, each song is represented as a state of a graph and is described by a set of tags, whereas each edge is weighted by the acoustic similarity between the two songs it connects. The documents relevant for a given query are retrieved by searching, in the space of possible paths through the graph, the one that most likely is related to the request. Retrieval is performed efficiently and does not require any preprocessing steps or parameter estimation (i.e., training). The model is applicable in different music discovery scenarios, such as item-based recommendation, playlist generation, and qualitative reranking of fast retrieval approaches.

The remainder of this article is organized as follows. After a discussion of related work in Section 2, the model and the retrieval framework are defined in Sections 3 and 4, respectively. Sections 5, 6, and 7 address the description of the evaluation, in particular the music representation used, the experimental setup, and the results. Lastly, Section 8 provides a general conclusive discussion and possible future works.

2. RELATED WORK

A considerable amount of research has been devoted to semantic music search and discovery engines. These systems allow queries in natural language, such as tags or song metadata (e.g., title, artist, year) and return songs that are semantically related to this query [Celma et al. 2006; Knees et al. 2007; Turnbull et al. 2007]. Barrington et al. [2009] compare a content-based semantic retrieval approach with Apple iTunes Genius, which recommends music by mainly exploiting user behavior (i.e., collaborative filtering). They showed that, while Genius generally performs better, the content-based research system achieves competitive results in the *long tail* of undiscovered and little recommended songs.

The graph-based framework proposed in this article aims at improving the state-of-the-art music discovery technologies by combining autotags with content-based acoustic similarity. Berenzweig et al. [2004] show that it is important to combine both subjective (e.g., semantic labels) and acoustic representation when dealing with music similarity. In the following, Sections 2.1 and 2.2 describe several techniques for content-based acoustic similarity and automatic music tagging, respectively. A variety of state-of-the-art retrieval approaches combining different sources of music information are reviewed in Section 2.3, which also reports a brief summary of the original model proposed in this article.

2.1. Acoustic Similarity

Acoustic similarity is an *objective* relationship, in the sense that it does not consider the subjectivity of music perception. Typical approaches to compute the content-based

[6] Hereafter we focus on music collections where each song has already been annotated by any noisy automatic tagging approach.

acoustic similarity between two songs rely on comparing the features automatically extracted from the audio signal of each song [Casey et al. 2008b]. Yet the choice of the features to be extracted and of the similarity measure to be used is crucial in the design of a MIR system, and strictly depends on the applicative scenario. For example, automatic playlist generation gains little benefit from the use of melodic features, while beat tracking and rhythm similarity may be particularly useful. Conversely, the identification of different versions of a same music work requires focusing on the melodic content, with the rhythm playing a secondary role. In any case, the audio descriptors of a song are generally extracted from short snippets (e.g., 20–50 ms) of the audio signal and result in a vector of features $\boldsymbol{x}_t$, where t is related to time in the audio signal where the snippet occurs. The acoustic content of a song $\mathcal{X}$ is then represented as a series of audio features, that is, $\mathcal{X} = (\boldsymbol{x}_1, ..., \boldsymbol{x}_T)$, where T depends on the length of the song. This series of vectors can then be processed to compute acoustic similarity relationships between the songs.

In the literature, most of the state-of-the-art systems for computing acoustic similarity rely on timbral descriptors. The *timbre* can be defined as “the sound characteristics that allow listeners to perceive as different two sounds with the same pitch (i.e., the perceived fundamental frequency of a sound) and intensity (i.e., the energy of the music vibration) but played by different instruments” [Orio 2006]. In our experiments, we relied on timbral descriptors as well, sometimes additionally supported by temporal descriptors; we refer the reader to Section 5.1 for the details.

Early approaches to define acoustic similarity from audio features used mixtures of Gaussians to represent the songs and computed similarity as the divergence between the models. In particular, interesting results were achieved using a single Gaussian to model the timbral features of the songs, and Kullback-Leibler (KL) divergence to compute pairwise similarity [Mandel and Ellis 2005]. Later, Jensen et al. [2009] showed that this approach may have some weaknesses in the case of many different instruments being played simultaneously, and proposed introducing a source separation component to improve its performances. Recently, other competitive algorithms have been proposed. Hoffman et al. [2008] use the hierarchical Dirichlet process to automatically discover the latent structure within and across groups of songs in order to generate a compact representation of each song, which could be efficiently compared using KL divergence. Alternatively, Seyerlehner et al. [2008] represent songs through a nonparametric variant based on vector quantization; this representation allows for more powerful search strategies, such as Locality Sensitive Hashing (LSH) and KD-trees. LSH was used by Casey et al. [2008a] as well to compute acoustic similarity between music excerpts represented as a series of temporal features. Music similarity has been one of the tasks of the MIREX (Music Information Retrieval Evaluation eXchange) since 2005 [Downie 2008]; other approaches for computing acoustic similarity can be found in the records of the yearly campaigns.

See Section 5.2 for the details about the specific methods used in the evaluation process. Yet, regardless of the strategy, when processing a music collection, acoustic similarity is generally computed as a distance measure between the audio features of each song and all the other documents in the collection. Given N songs, similarity can be expressed by a N^2 *similarity matrix*, where each generic entry (i, j) represents the acoustic similarity of song i with song j [Pampalk 2006].

2.2. Music Automatic Tagging

In recent years, a variety of strategies have been proposed to automatically collect tags for the songs of a music collection. In particular, early approaches concerned social tagging (i.e., the users of the system assign the tags to the songs they listen

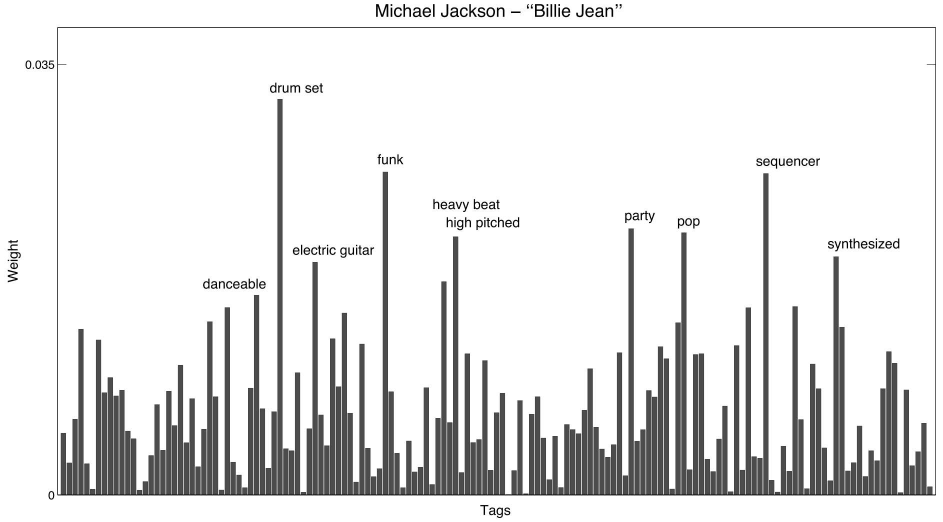


Fig. 1. The semantic multinomial distribution over a set of 149 tags for the song "Billie Jean", as performed by Michael Jackson; the ten most probable tags are labeled.

to) [Lamere 2008], Web mining (i.e., the tags are searched by mining the Web pages linked to a given song) [Knees et al. 2007] and tag propagation from similar songs (i.e., assign the tags which already describe the artist, similar songs, similar artists, etc.) [Sordo et al. 2007]. Each of these has advantages and disadvantages as described by Turnbull et al. [2008a].

Recently, MIR researchers have proposed content-based automatic tagging systems (autotaggers), which focus on modeling the characteristic acoustic patterns associated with each tag of a semantic vocabulary in an annotated database. State-of-the-art autotaggers are based on discriminative approaches, for example, boosting [Bertin-Mahieux et al. 2008] and support vector machines [Mandel and Ellis 2008], as well as generative models, for example, the Gaussian mixture models [Turnbull et al. 2008b], the codeword Bernoulli average model [Hoffman et al. 2009], and the time-series model [Coviello et al. 2011]. Each tag model is used to compute the relevance of that tag to a new song, and the most likely tags can then be assigned to the song. Additionally, approaches that add a second modeling layer to capture the tag co-occurrences in the space of the autotagger outputs showed improvements in the quality of the final predictions (e.g., see Yang et al. [2009], Ness et al. [2009], and Miotto and Lanckriet [2012]).

Regardless of the strategy, automatic tagging systems may generally output a vector of tag weights, one for each tag of a semantic vocabulary. After normalizing, so that entries sum to one, this vector may be interpreted as a *Semantic MultiNomial* (SMN), that is, a probability distribution characterizing the relevance of each tag to a song. A song can then be annotated by selecting the top-ranked tags in its SMN, or the SMN itself can be used in the retrieval function (e.g., given a tag query, retrieval can be performed by ranking songs according to the tag's probability in their SMN) [Barrington et al. 2009; Turnbull et al. 2007]. As an example, Figure 1 shows the semantic multinomial achieved by a content-based autotagger over 149 different tags for the song "Billie Jean", as performed by Michael Jackson.

Depending on the tagging strategy, an SMN describing a song can be sparse or dense. On the one hand, content-based autotaggers lead to a dense representation, because they can estimate a model for each tag in the vocabulary. On the other hand, strategies such as social tags and Web mining lead to a sparse representation because it is not possible to collect a relevance value for each tag in the vocabulary (e.g., it is unlikely that a user would tag a “classical” song as “heavy metal”). In this last case, the tag not collected takes a zero value.

Refer to Section 5.3 for details about the autotagging strategies used in the evaluation process.

2.3. Combining Music Descriptions to Improve Retrieval

As already mentioned in Section 1, this article presents a novel model to retrieve songs by combining acoustic similarity and the semantic descriptions given by the autotags. Again, our main goal is to propose a general methodology which overreaches the limitations of these two automatic strategies by using them together for delivering higher-quality ranking lists in response to a given semantic query. That is, our retrieval system attempts to return a list of music items which are acoustically coherent and pertinent to the request and, at the same time, are ranked according to their relevance to one or more semantic labels that give context to the user information need.

In the literature, approaches that merge different heterogeneous descriptions of music information were proposed in Slaney et al. [2008] for music classification, in Turnbull et al. [2009] and Tomasik et al. [2009] for semantic retrieval, in McFee and Lanckriet [2009] for artist similarity, and in Wang et al. [2010] for artist style clustering. These methodologies generally learn the parameters of a function that is used to join the different sources of information in order to provide the user with more subjective song descriptions and similarity relationships. Our approach is consistently different because it is built on a graph-based representation that combines different music descriptions, and it does not rely on additional processing or training to carry out retrieval. This point is particularly important in scenarios where it is difficult to collect reliable ground-truth data for the training step. Moreover, rather than providing a more meaningful description of songs or better similarity relationships, our goal is to deliver a unified retrieval mechanism located on a higher level than acoustic and semantic representations considered individually. This means that it can also exploit these more sophisticated descriptions. A graph-based representation for music recommendation is also used by Bu et al. [2010]; however, instead of a graph, they use a unified hypergraph to model and combine social media information and music acoustic-based content. This approach is an application of ranking on graph data [Agarwal 2006] and requires learning a ranking function; again, it can be highlighted that we do not need training operations to perform retrieval. Additionally, Bu et al. [2010] cover a wider application than semantic retrieval, since they focus on the design of a hybrid recommender which also considers user-profile data. However, we do not investigate the extension of our model in a hybrid recommendation scenario where user-profile is also taken into account. Social data and graph analysis were also exploited by Fields et al. [2011] to combine acoustic similarity with musician social networks to computing hybrid similarity measures for music recommendation.

Knees et al. [2009] examine the effect of incorporating audio-based similarity into a tag-based ranking process, either by directly modifying the retrieval process or by performing post-hoc audio-based reranking of the search results. The general idea is to include in the ranking list songs that sound similar to those already collected through tag-only retrieval; hence, acoustic similarity is used as a correction factor that improves the ranking list. Again, our approach is different because tags and acoustic similarity are used together at the same time in the retrieval process (i.e., one representation is not used to correct the ranking scheme of the other).

The framework is proposed as the core component of a semantic music discovery engine. However, the approach is very general and can be used in other scenarios as well as with other media. The main requirement of the model is that a medium may be represented by semantic descriptors, and that it is possible to compute a similarity measure between digital items. In particular, image and video retrieval seem potential scenarios where the proposed framework may be applied [Feng et al. 2004; Rasiwasia and Vasconcelos 2007; Tsai and Hung 2008].

3. PROBABILISTIC MODELING OF A MUSIC COLLECTION

The goal of semantic music search and discovery engines is to retrieve a list of songs in response to the description of a user information need. The latter can be represented either directly by a general semantic indication, such as the tag “rock”, or indirectly by a seed song, such as the audio content and the set of tags assigned to “My Sharona”. In both cases, it can be assumed that the goal of the user is to *observe* consistently the fulfilment of his information need during the time of his access to the music collection. In particular, the user query can be interpreted as a sequence of observations through time, that is, $\boldsymbol{O} = (\boldsymbol{o}_1, ..., \boldsymbol{o}_T)$, where T gives the number of retrieved songs and $\boldsymbol{o_t}$ expresses what the user is expected to listen to at time t. Therefore, the framework here discussed provides a model to rank the songs of a music collection in a way that maximizes the probability to *observe* over time music items that are relevant to the user query in terms of both contextual and acoustic sources of information.

With this aim in mind, we propose to represent the music collection as a graph, where each state is a song and the retrieval process is considered as a *doubly embedded stochastic process*, where an underlying stochastic process can only be observed through another set of stochastic processes that “*emit*” the sequence of observations. This model can represent either content and context information, under the following assumptions [Miotto and Orio 2010].

- If each state represents a song in the collection, acoustic content-based similarity can be modeled by *transition probabilities*, which weight each edge connecting two generic states of the model.
- If the symbols emitted by the states are semantic labels, the context that describes each state can be represented by *emission probabilities*, which describe the relationship between the tags and the song mapped into each generic state of the model.

Therefore, in such a process, at each time step the model performs a transition to a new state according to transition probabilities and it emits a new symbol according to emission probabilities. Thus each path across the graph has a total likelihood which depends on both transition and emission probabilities. When a sequence of observations is given to the model, the states composing the most probable path need to be computed through decoding.

This structure takes its ground from the definition of a Hidden Markov Model (HMM), a double embedded stochastic process that is often used to represent physical processes that evolve over time (e.g., speech, audio signal) [Rabiner 1989]. HMMs have been widely used in MIR applications, such as segmentation [Raphael 1999], query-by-singing [Shifrin et al. 2002], and automatic identification and alignment [Miotto et al. 2010]. In particular, an HMM defines a structure that combines two stochastic processes (i.e., transition and emission probabilities) to generate a sequence of observations, or dually, to highlight the most likely state sequence which has generated a given series of observations. In our case, the user query defines the observations (i.e., $\boldsymbol{O}$), while the hidden sequence of states to be highlighted is the list of songs to be retrieved. Despite this similarity in the model definition, we do not refer to our

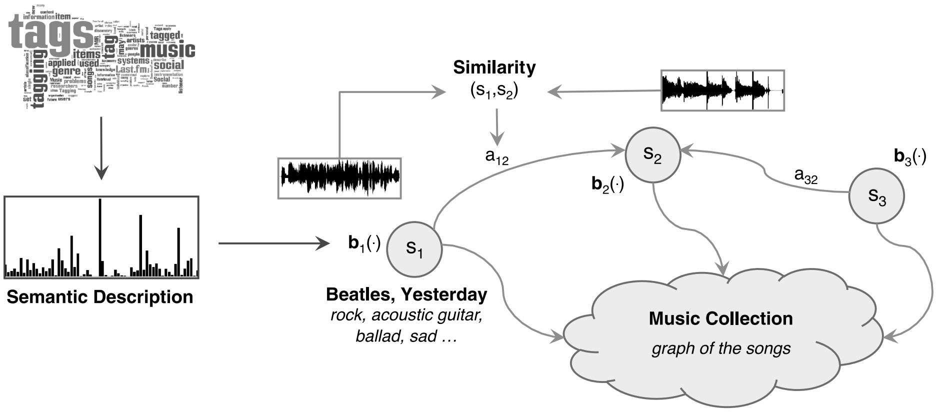


Fig. 2. General structure of the graph-based model: each song included in a music collection is represented by a state and is described by a set of tags. States are linked together by edges weighted according to acoustic similarity between the songs.

framework as an HMM because, compared to the formulation of Rabiner [1989], we introduce some variations at the standard structure of the model as well as at the algorithm to highlight the most likely path in order to better fit the applicative scenario.

A suitably built model can be used to address the examples provided at the beginning of this section. On the one hand, the model can be used to create a path across songs while observing, for a defined number of time steps, the semantic label "rock". On the other hand, the model can start the path from the state associated with the song "My Sharona" and proceed to new states while observing the semantic labels associated with the seed song. In both cases, the songs in the path are likely to have a similar content because of transition probabilities and are likely to be in the same context because of emission probabilities. A graphical representation of the model is shown in Figure 2; as can be seen, songs are linked together through edges which are weighted according to the similarity between the audio signals. In addition, each state is also described by a set of tags which contextualize each song.

3.1. Definition of the Model

The graph-based structure representing a collection of autotagged songs can be formally defined through a set of parameters; in particular, the model λ is defined by:

- the number of songs N in the collection, each song represented by a state of the graph. Individual states are denoted as $\mathcal{S} = \{S_1, ..., S_N\}$, where each generic state S_i represents the song i, while a state at time t is generally denoted as $s(t)$;
- the number of distinct tags that can be used to semantically describe a song. The set of symbols is defined by a finite size vocabulary $\mathcal{V} = \{w_1, ..., w_{|\mathcal{V}|}\}$;
- the state transition probability distribution $\boldsymbol{A} = \{a_{ij} \mid i, j = 1, ..., N\}$, which defines the probability of moving from state S_i to state S_j in a single step. This distribution is related to the acoustic similarity between songs; therefore a_{ij} depends on the audio similarity of song i with song j (i.e., $\boldsymbol{A}$ is a similarity matrix as defined in Section 2.1);
- the emission symbol probability distribution $\boldsymbol{B} = \{b_i^1, ..., b_i^{|\mathcal{V}|} \mid i = 1, ..., N\}$, which represents the probability that each tag $w \in \mathcal{V}$ is associated with song i. Each probability value represents the strength of the relationship *song-tag*; the vector $\boldsymbol{b}_i(\cdot)$ is the semantic description of the general song i (i.e., an SMN as defined in Section 2.2).

We use the compact notation $\lambda = (\boldsymbol{A}, \boldsymbol{B})$ to denote the model; the parameters N and $|\mathcal{V}|$ can be inferred from the probability distributions.

Common approaches for computing acoustic similarity usually give a positive value for each pair of songs, implying $a_{ij} > 0$ for each i, j. However, in particular with the aim of improving scalability, at retrieval time, we consider each generic state S_i to be connected directly to only the subset of states $\mathcal{R}(S_i)$, which includes the most acoustically similar songs to song i; conversely, the transition probabilities with all the other states are set to zero. This leads to a sparse transition matrix which can help to speed up the retrieval process. Additionally, we show in Section 7.1 how the size of subsets $\mathcal{R}(S_i)$, with $i = 1, ..., N$, may also affect the retrieval performances of the model. Self-transitions are set to zero as well, because self-similarity is not a relevant factor in retrieval applications. Both transition and emission probabilities are normalized to one to satisfy the stochastic propriety of the model formulation and to reduce the numerical sparsity of the different relationships (i.e., song-tag and song-song), that is

$$\sum_{j=1}^{N} a_{ij} = 1 \ , \quad \text{and} \quad \sum_{k=1}^{|\mathcal{V}|} b_i^k = 1 \ , \quad \text{with} \quad i = 1, ..., N \ . \tag{1}$$

Because of these steps, transition probabilities are usually not symmetric (i.e., $a_{ij} \neq a_{ji}$), which, at least in the case of music, is a reasonable assumption. As an example, a complex and extremely popular song such as “Stairway to Heaven” may have inspired hundreds of songs that reprise part of its content. While each of them can be considered similar to “Stairway to Heaven” in its original version, the latter may be only loosely related to these songs.

After setting all the parameters, the model can be used to decode the most likely sequence of states which may have generated a given series of observations as discussed in the following section.

4. RANKING THE SONGS

The goal of the retrieval function is to highlight a subset of songs that are relevant to a particular query, either expressed by tags or by a seed song. The general problem can be stated as follows [Rabiner 1989].

> [G]iven the observation sequence $\boldsymbol{O} = \{\boldsymbol{o}_1, ..., \boldsymbol{o}_T\}$, and the model λ, find a corresponding state sequence $\{s^*(1), ..., s^*(T)\}$ which is optimal in some sense.

The description of the user information need is represented by the observations sequence $\boldsymbol{O}$; the value of T defines the number of songs which are retrieved. Yet, the output of the search process is a list of songs ranked according to their relevance with the given observation sequence.

4.1. The Retrieval Algorithm

The decoding problem of highlighting the list of states which most likely has generated a sequence of observations can be solved by using a variant of the Viterbi algorithm [Forney 1973]. The latter acts as a max-sum algorithm and searches efficiently in the space of possible paths to find the optimal one (i.e., the most likely one). In particular, the algorithm is composed by a forward computation which finds the maximization for the most probable path, and by a backward computation which decodes the sequence of states composing such most probable path. The cost of the process grows only linearly with the length of the desired ranking list; this means that retrieval can be performed more efficiently simply by keeping the value of T relatively small (i.e., shorter ranking lists).

ALGORITHM 1: The Retrieval Algorithm

```
1: Input: the model $\lambda = (\boldsymbol{A}, \boldsymbol{B})$, the vocabulary of tags $\mathcal{V}$ and the observation sequence
   $\boldsymbol{O} = \{\boldsymbol{o}_1, ..., \boldsymbol{o}_T\}$, such that, for each $t = 1, ..., T$,
   $$\boldsymbol{o}_t(\cdot) = \begin{cases} \{(o_t^1, ..., o_t^k) \mid o_t^y = w \in \mathcal{V} \text{ for } y = 1, ..., k\} \text{ if k-tag query-by-description} \\ \boldsymbol{b}_q(\cdot) \ \text{ if query-by-example with song q} \end{cases}$$
2: Define the KL divergence $D_{KL}(\boldsymbol{x} \parallel \boldsymbol{y})$ between general discrete probability distributions $\boldsymbol{x}$
   and $\boldsymbol{y}$:
   $$D_{KL}(\boldsymbol{x} \parallel \boldsymbol{y}) = \sum_{k=1}^{|x|} x_k \cdot \log \frac{x_k}{y_k} \,, \quad \text{where} \quad |\boldsymbol{x}| = |\boldsymbol{y}|$$
3: Define $\mathcal{R}(S_i)$ for each state $S_i \in \lambda$, i.e. the subset of states to which $S_i$ is connected, according
   to $\boldsymbol{O}$, edit $\boldsymbol{A}$ such that,
   $$a_{ij} = \begin{cases} a_{ij} & \text{if } j \in \mathcal{R}(S_i) \\ 0 & \text{elsewhere,} \end{cases} \quad \text{for each } \ i, j = 1, ..., N \,,$$
   and then normalize the edited $\boldsymbol{A}$ such that,
   $$\sum_{j=1}^{N} a_{ij} = 1 \quad \text{for} \quad i = 1, ..., N \,.$$
4: Define $OBS_i(t)$ with $i = 1, ..., N$, according to $\boldsymbol{O}$ and $t$:
   query-by-description: $OBS_i(t) = b_i(o_t^1) \cdot b_i(o_t^2) \cdot \ ... \ \cdot b_i(o_t^k)$ for all $t > 0$
   query-by-example: $OBS_i(t) = \begin{cases} 1 & \text{if } t = 1 \text{ and } i = q \\ 0 & \text{if } t = 1 \text{ and } i \neq q \\ \frac{1}{D_{KL}(\boldsymbol{b}_i(\cdot) \parallel \boldsymbol{b}_q(\cdot))} & \text{if } t > 1 \end{cases}$
5: Initialization: for $i = 1, ..., N$:
   $$\delta_1(i) = OBS_i(1) \,, \quad \psi_1(i) = 0$$
6: Recursion: for $t = 2, ..., T$, $i = 1, ..., N$:
   $$\delta_t(i) = \max_{1 \leq j \leq N}[\delta_{t-1}(j) \cdot a_{ji}] \cdot OBS_i(t)$$
   $$\psi_t(i) = \arg\max_{1 \leq j \leq N}[\delta_{t-1}(j) \cdot a_{ji}]$$
   $$a_{ri} = \frac{a_{ri}}{\eta} \quad \text{with} \quad r = \psi_t(i) \,, \ \eta = 10$$
7: Decoding: highlight the optimal path
   $$s(t)^* = \begin{cases} \arg\max_{1 \leq j \leq N}[\delta_t(j)] & \text{if } t = T \\ \psi_{t+1}(s(t+1)^*) & \text{if } t = T-1, T-2, ..., 1 \end{cases}$$
8: Output: the ranking list $\{s(1)^*, ..., s(T)^*\}$.
```

The details of the retrieval function are provided in Algorithm 1. The overall time complexity is $\mathcal{O}(TN^2)$, whereas the space complexity is $\mathcal{O}(TN)$. In particular, as can be seen in step 6 (first and second line), for each state S_i, the forward computation at time t seeks the next best movement across the graph, which is the one maximizing the probability of the path at that time in terms of: (i) path at time $t-1$, (ii) acoustic similarity, and (iii) semantic description.

Algorithm 1 (step 4) defines $OBS_i(\cdot)$, a general function related to the type of query (tags or seed song) which specifies how the semantic description of the generic song i and the initial conditions of the model λ are considered during the retrieval process. Motivations and details are discussed in Section 4.2.

The recursion step expressed in step 6 of Algorithm 1 introduces a variation of the role of transition probabilities. In fact, because of the structure of the model, it could happen that the optimal path enters a loop between the same subset of songs or, in the worst case, jumps back and forth between two states. This would lead to retrieved lists that present the same set of songs multiple times. Moreover, the loop could be infinite, meaning that the algorithm cannot exit from it and the retrieval list would only be composed by very few items. We addressed this problem by considering a penalty factor that is applied to the transition probabilities when they have already been chosen during the forward step. In fact, when a transition is chosen, we decrease the corresponding probability value by factor η (we used $\eta = 10$), as shown in the third line of step 6; this makes it unlikely for the states' sequence to pass again through that transition. Attenuation is carried out *temporarily*, meaning that it affects the structure of the model only during the current retrieval operation. It should be noted that after this modification there is no guarantee that the path is globally optimal; however, what is relevant to our aims is that the path has a high probability of being relevant to the query from both content and context information and, at the same time, covers a large part of the collection.

Preliminary experiments pointed out another limitation of the model, which concerns long sequences of observations; in fact, we noted a strong decrease in the retrieval precision when decoding long paths through the graph. Consequently, the model appears to be generally poor at capturing long-range correlations between the observed variables (i.e., between variables that are separated by many time steps). In order to tackle this problem, we consider the retrieval process composed by many shorter substeps, each one retrieving a subsequence of the final ranking list. When one of the subsequences is delivered, the following retrieval substep restarts from the last song previously suggested and is performed only on the songs not yet included in the ranking. Given the locality of the approach, in this way we try to keep constant the correlation between the query and the retrieved songs along the whole list. Additionally, splitting the retrieval in substeps allows the system to better interact with the final users, who can obtain the ranking list of songs incrementally, instead of having to wait for the whole computation. For instance, the model can be set to deliver enough songs at each substep to fill a Web page; in this way the remaining items may then be retrieved while the user is already checking the first delivered results.

4.2. Querying the Model

In the interaction with music discovery and search engines, a user can generally submit a query in two alternative ways: by providing tags (*query-by-description*) or a seed song (*query-by-example*). The proposed framework can easily handle both scenarios; however, according to the topology of the query, some parameters need to be set differently, as also highlighted in step 4 of Algorithm 1.

In the *query-by-description* scenario, the model has to rank the songs according to their relevance with the provided tags. The query may be composed of a single tag (e.g., “rock”) or a combination of tags (e.g., “mellow”, “acoustic guitar”, “female lead vocals”). In this case, the observation at iteration t, with $t = 1, ..., T$, is the vector $\boldsymbol{o}_t = \{o_t^1, \cdots, o_t^k\}$, where each entry is a tag of the vocabulary and k is the number of tags in the query. We consider the tags in the query having the same importance; however, tags may be also weighted in relation to the query string. For example, it is possible to give more weight to the words that appear earlier in the query string as is commonly done by Web search engines. The probability of starting a path from a given state depends only on the probability of emitting the observation. Therefore, the function $OBS_i(t)$ of Algorithm 1 is defined as

$$OBS_i(t) = b_i(o_t^1) \cdot b_i(o_t^2) \cdot \ldots \cdot b_i(o_t^k) , \tag{2}$$

for each state S_i and all $t > 0$. It can be noted that observations may be composed of the same tags for all the time steps, or they may also change over time (e.g., starting with {"rock"} for the first iterations, going on with {"pop", "tender"} in the following steps, and so on)[7].

In the *query-by-example* scenario, the system is required to provide the user with a list of songs similar to a seed song q. In this case all the paths are forced to start from the seed song, that is $OBS_i(1) = 1$ when $i = q$, while $OBS_i(1) = 0$ elsewhere[8]. The observation sequence to be decoded is modeled as the tags of the seed song, that is $\boldsymbol{b}_q(\cdot)$; so, in this case $OBS_i(t)$ is defined as the inverse of the KL divergence between the semantic description of a chosen state and $\boldsymbol{b}_q(\cdot)$ [Kullback and Leibler 1951]. The choice of the KL divergence aims at generalizing the terms used for the tags, because it is related to the similarity of concepts associated with the tags rather than the pure distance between lists of tags. We consider the inverse because the goal is to have similarity relationships (i.e., to maximize the probability when the divergence is small). Therefore,

$$OBS_i(t) = \frac{1}{D_{KL}(\boldsymbol{b}_i(\cdot) \parallel \boldsymbol{b}_q(\cdot))} , \quad \text{where} \quad D_{KL}(\boldsymbol{b}_i(\cdot) \parallel \boldsymbol{b}_q(\cdot)) = \sum_{k=1}^{|\mathcal{V}|} b_i^k \cdot \log \frac{b_i^k}{b_q^k} , \tag{3}$$

for the generic state i, a seed q, and all $t > 1$. In preliminary experiments we also attempted to compute the similarity between two songs as the negative exponentiation of their KL divergence; however, this solution led to consistently worse empirical results.

5. MUSIC REPRESENTATION

This section overviews the methodologies used to represent the songs in the evaluation process reported in Sections 6 and 7; more details can be found in the corresponding references. It is important to note that the proposed model can be used with any of the approaches reviewed in Sections 2.1 and 2.2 for computing content-based acoustic similarity and autotags. The approaches here described represent only the experimental setup we built to validate the proposed model.

5.1. Audio Features

Available music datasets are released either as a set of audio clips or as a list of songs (title and artist) to tackle copyright issues. According to the case, we use different music content features. However, as mentioned in Section 2.1 we mostly relied on timbral descriptors.

When we have access to the audio clips, we represent the songs by vectors of Mel Frequency Cepstral Coefficients (MFCCs) and Fluctuation Patterns (FPs). MFCCs are

[7] In this work we only treat queries composed by the same tags for all the time steps. We mention the possibility of queries which dynamically change over time just to highlight that the formulation of the model can naturally handle this applicative scenario as well. However, there are currently no standard datasets to test such a task; therefore, we consider the use of queries dynamic over time as a possible future application.

[8] We assume that the seed song q is stored in the collection. Possible extensions to scenarios where q does not belong to the collection are not taken into account in this work.

a popular feature for content-based music analysis for representing the timbre of the audio signal [Logan 2000]. They summarize the spectral content of a short-time window of an acoustic waveform by using the discrete cosine transform to decorrelate the bins of a Mel Frequency spectral histogram. We represent each song by a bag of 13-dimensional MFCC vectors computed on half-overlapping windows of 23 ms. MFCCs are computed on at most 4 minutes of music taken from the middle of the song; at the end, we randomly subsample 15,000 feature vectors to generally obtain a same-size representation of each song. Additionally, we also include the first and second instantaneous derivatives of MFCCs, achieving a final 39-dimensional representation (delta-MFCC vectors). In contrast, FPs highlight characteristics of the audio signal which are not captured by the spectral representation (e.g., the periodicity of the music signal) by describing the amplitude modulation of the loudness per-frequency band. The combined use of MFCCs and FPs is common in MIR, for instance, for hubness reduction [Flexer et al. 2010]. We compute an FP by: (i) cutting a 20-band spectrogram into 3-second half-overlapped segments, (ii) using an FFT to compute amplitude modulation frequencies of loudness (range 0–10 Hz) for each segment and frequency band, (iii) weighting the modulation frequencies based on a model of perceived fluctuation strength, (iv) applying filters to emphasize certain patterns and smooth the result [Pampalk 2006]. The resulting FP is a 12 (frequency bands according to 12 critical bands of the Bark scale) times 30 (modulation frequencies, ranging from 0–10 Hz) matrix for each song.

When copyright issues prevent the owner from releasing the audio clips, a dataset can still be published as a list of songs specifying title, artist, and unique identification parameters (ID) that can be used for content access. In this scenario, descriptors can be obtained using public Web-based repositories, such as The Echo Nest Web service[9]. The latter provides a free music analysis API that users can exploit to collect information about songs and artists. In particular, a user uploads an audio clip for analysis; once the track has been uploaded and automatically processed, the Echo Nest provides this user with a unique song ID that can be exploited to access the information about that song (i.e., content-based features as well as metadata). Releasing the unique song IDs makes it possible for everyone to access the same audio descriptions of a given song. Among the different content descriptors available, the Echo Nest Timbre (ENT) features provide the timbral descriptors. ENTs are derived from slightly longer windows than MFCCs (generally between 100 and 500 ms), where, for each window, the service calculates 12 "timbre" features[10]. Each song is then represented by a bag of 12-dimensional vectors; computing the first and second instantaneous derivatives of these vectors leads to the final 36-dimensional delta-ENT vectors. ENT features have been already successfully used in different applications related to music similarity and access (e.g., see Tingle et al. [2010], Tomasik et al. [2010], and Bertin-Mahieux et al. [2010]).

5.2. Acoustic Similarity

We collect acoustic similarity relationships by using two distances computed over different audio features and, when possible, by deriving a single similarity measure through a weighted combination between them. Both approaches are fast in computation and easy to implement. Since we aim at showing the general performances of the model and not a particular configuration, we prefer such aspects to the greater precision that could be achieved using some of the alternative models reviewed in

[9] `http://developer.echonest.com/`

[10] The 12 timbre coefficients highly abstract the spectral surface of the audio signal; however, we do not have access to the details of the exact calculation of the ENTs because they are a trade secret of the company.

Section 2.1, which, however, require a more sophisticated parameter estimation, and whose implementation could be more error-prone.

Timbral similarity is computed following the approach described in Mandel and Ellis [2005], that is, representing the timbre descriptors of each song (i.e., delta-MFCCs or delta-ENTs) as the single Gaussian (full covariance) with the maximum likelihood of fitting the song's features. The distance between two Gaussians is computed as the KL divergence, that is, $D_{KL}(x \parallel y)$, for each general pair of songs x and y, rescaled to turn into a similarity value (i.e., negative exponentiation of each divergence). Since the standard KL divergence is asymmetric, we have that $D_{KL}(x \parallel y) \neq D_{KL}(y \parallel x)$, which generally fits our applicative scenario of nonsymmetric music similarity relationships.

In contrast, similarity of FPs is computed as the rescaled Euclidean distance between two FPs [Pampalk 2006].

To obtain an overall acoustic similarity measure $sim(x, y)$ between the two songs x and y, all the described similarity values are combined by a simple arithmetic weighting

$$sim(x, y) = 0.7 \cdot z_G(x, y) + 0.3 \cdot z_{FP}(x, y) \,, \tag{4}$$

where $z_G(x, y)$ and $z_{FP}(x, y)$ are the value of the timbre-based and FP-based similarity after z-normalization. We set the weighting coefficients after some preliminary tuning experiments.

Lastly, we used the combination shown in Eq. (4) only when we had access to the audio clips of a dataset. Conversely, in cases where we had the ENT features only, the similarity computation refers to the single Gaussian-based timbral similarity alone. In any case, all the similarity values related to each experiment are collected together to form the transition matrix of the graph-based retrieval model.

5.3. Tags

We use two different automatic strategies to collect tags: content-based autotaggers and social tags. We separately address these two different representations in order to validate the model with different semantic sources; however, they could be combined into a single richer one as described by Turnbull et al. [2009]. For both approaches, we consider a vocabulary $\mathcal{V}$ of $|\mathcal{V}|$ distinct tags w_i, for $i = 1, ..., |\mathcal{V}|$. As mentioned in Section 2.2, at the end, each song is described by an SMN, which is used as the emission probability distribution in the corresponding state of the model.

5.3.1. Content-Based Autotaggers. The problem of content-based automatic music tagging is widely tackled as a supervised multiclass labeling problem [Carneiro et al. 2007], where each class corresponds to a tag w_i of the vocabulary $\mathcal{V}$. Among all the different approaches of the literature, we used the Gaussian Mixture Model (GMM) [Turnbull et al. 2008b], leveraged by a contextual model based on the Dirichlet Mixture Model (DMM) [Miotto and Lanckriet 2012].

In particular, Turnbull et al. [2008b] proposed modeling the acoustic features associated with each tag w_i in $\mathcal{V}$ with a probability distribution $p(\boldsymbol{x}|w_i)$ over the space of audio features $\boldsymbol{x}$ as a GMM

$$p(\boldsymbol{x}|w_i) = \sum_{r=1}^{R} a_r^{w_i} \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_r^{w_i}, \boldsymbol{\Sigma}_r^{w_i}) \,, \tag{5}$$

where R is the number of mixture components, $\mathcal{N}(\cdot|\boldsymbol{\mu}, \boldsymbol{\Sigma})$ a multivariate Gaussian distribution with mean $\boldsymbol{\mu}$ and 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 extracted from the songs that are positively associated with w_i in an annotated database, using the hierarchical Expectation Maximization (EM) algorithm [Vasconcelos and Lippman 1998].

Given the audio content of a new song $\mathcal{X} = \{\boldsymbol{x}_1, ..., \boldsymbol{x}_T\}$, the relevance of each tag w_i is computed using the Bayes' rule

$$\pi_i = P(w_i|\mathcal{X}) = \frac{p(\mathcal{X}|w_i)\, P(w_i)}{p(\mathcal{X})}, \tag{6}$$

where $P(w_i)$ is the tag prior (assumed uniform) and $p(\mathcal{X})$ the song prior, that is, $p(\mathcal{X}) = \sum_{j=1}^{|\mathcal{V}|} p(\mathcal{X}|w_j)P(w_j)$. Collecting all the posterior probability π_i leads to the song SMN, that is, $\boldsymbol{\pi} = (\pi_1, ..., \pi_{|\mathcal{V}|})$, which is intended normalized to one.

The GMM alone models each tag independently, without taking into account the latent correlation between them (e.g., "rock" may often co-occur with "guitar"). For this reason, we also process the SMNs' output by the GMM through a second modeling layer which captures the tag context and leads to improved semantic descriptions. With this aim in mind we use the DMM [Miotto and Lanckriet 2012], which is a generative model that assumes the SMNs $\boldsymbol{\pi}$ of the songs positively associated to a tag w_i being distributed accordingly to a mixture of Dirichlet distributions, that is

$$p(\boldsymbol{\pi}|w_i; \Omega^w) = \sum_{r=1}^{R} \beta^{w_i} \mathrm{Dir}(\boldsymbol{\pi}|\alpha_r^{w_i}) \;, \tag{7}$$

where R is the number of mixtures, $\beta_k^{w_i}$ are the mixing weights, and $\mathrm{Dir}(\cdot|\boldsymbol{\alpha})$ is a Dirichelet distribution of parameters $\boldsymbol{\alpha} = (\alpha_1, ..., \alpha_{|\mathcal{V}|})$. Each DMM is estimated from the SMNs of the songs positively associated with w_i in an annotated database, via the generalized EM algorithm [Dempster et al. 1977].

Given the SMN $\boldsymbol{\pi}$ of a new song, the final relevance value of each tag w_i, that is, $P(w_i|\boldsymbol{\pi})$, is computed as posterior probability using Bayes' rule (as in Eq. (6), with $\boldsymbol{\pi}$ in place of $\mathcal{X}$). The vector of the relevance values output by the DMM for each tag, after normalization to one, is the final semantic representation of the song which is used as emission distribution in the corresponding state of the model.

With regard to the implementation, we obtained the code of both Turnbull et al. [2008b] and Miotto and Lanckriet [2012]; the songs have been represented by using timbral content only (as in Turnbull et al. [2008b]), then delta-MFCCs or delta-ENTs.

5.3.2. Social Tags. Social tags were gathered from Last.fm, as available in November 2010. The users of this service can assign tags to the songs they are listening to, so these tags provide a valuable source of information on how people perceive and describe music. However, the data collection process is noisy since it involves a large community of nonexpert fans annotating music with an unconstrained vocabulary of free-text tags. Therefore, many tags may be redundant and inconsistent. Additionally, the vocabulary of tags that emerges from the community tends to be very simple and focused on social rather than acoustic aspects of the music.

For each song of the collection, we gathered two lists of social tags using the Last.fm public data sharing AudioScrobbler Web site[11]. We gathered both the list of tags related to a song and the list of tags related to an artist. The overall list of scores is given by summing the scores in both lists.

Since we are working using an experimental and of limited size vocabulary, the gathered social tags have to be mapped into the equivalent classes in $\mathcal{V}$. In order to

[11] http://ws.audioscrobbler.com/2.0/

match the largest part of the classes, we preprocess the social tags through stemming (e.g., joining “rock”, “rockers”, “rocking”), noise reduction (i.e., joining “r&b”, “rhythm and blues”, “r & b”), and synonyms detection (e.g., annotating with “down tempo” if the social tag is “slow beat”). When annotating a song, a tag in the vocabulary that correctly matches a social tag takes the corresponding Last.fm-based score; otherwise, it takes a zero value. Therefore, if no social tags of a song match any tags in the vocabulary, that song is represented by a uniform description where all the tags share the same relevance.

6. EXPERIMENTAL SETUP

One of the challenges of designing a music search engine is how to evaluate the novel methodology. Although several efforts have been made within the MIREX campaigns, data of past contests are not always freely available to test new approaches, because of well-known copyright issues. Ideally, the recommended songs should be evaluated by humans, in order to consider the subjective nature of the music similarity concept. Since human evaluation is a time-consuming task, we use an automatic approach by considering that reliable annotations on songs can be exploited to measure the quality of a ranking list. In particular, we use human-based annotated datasets, considering these annotations as ground truth for retrieval evaluation purposes. In this section we present these datasets, and we discuss the evaluation tasks, the metrics used to evaluate the results, as well as the models included in the comparison.

6.1. Music Datasets

We use two different annotated music collections: CAL500 and CAL10k.

6.1.1. CAL500 Database. This dataset consists of 502 popular songs of Western music by as many different artists [Turnbull et al. 2007]. Through a controlled survey, each song has been tagged by at least 3 human annotators using a semantic vocabulary of 149 tags. The vocabulary is diverse and spans genres, instruments, vocal and 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 2 subjects voted for the tag) and 0 otherwise.

Since we have full access to all the audio clips of this collection, we could extract both MFCCs and FPs. Therefore, the acoustic similarity is computed considering the combined similarity defined in Eq. (4).

6.1.2. CAL10k Database. CAL10k is a collection of 10,870 songs from 4,597 different artists, labeled from a vocabulary composed of 137 “genre” tags and 416 “acoustic” tags. Each song is labeled with 2–25 tags. The song-tag associations for this dataset have been mined from the Pandora Web site; since Pandora claims that their musicologists maintain a high level of agreement, these annotations are considered highly *objective* [Tingle et al. 2010].

As copyright issues prevent us from obtaining all CAL10k songs, we represent the audio content using the ENT features described in Section 5.1. Therefore, in this case, acoustic similarity relies on the single Gaussian-based timbral similarity alone.

6.2. Retrieval Tasks

The proposed framework is evaluated in the retrieval tasks introduced in Section 4.2, that is *query-by-description* and *query-by-example*.

In the *query-by-description* task, a combination of k distinct tags is provided as query and the songs are ranked by their relevance to them. The tags are taken from the finite size vocabulary $\mathcal{V}$; we consider queries composed by $k = 1, 2, 3$ tags. We believe that a real applicative scenario is very unlikely to have queries with more than three tags. The retrieval performances are measured by scrolling down the list and searching the rank of the songs that have been human-annotated with the query-tags in the ground truth; metrics are averaged through all the queries considered.

In the *query-by-example* task, the query is a song and the retrieval function ranks the songs according to their similarity with it. The quality of the ranking list is measured with respect to the songs having the most similar semantic description with the seed in the human-based ground truth. This set is built by computing the KL divergence between the semantic descriptions of the query and of all the other items in the collection, and by retaining the items showing the least divergence. In particular, for each song, we generally consider 30 and 50 relevant documents for CAL500 and CAL10k, respectively (we assume that a larger collection may imply a larger number of relevant songs). Additionally, Section 7.2.2 shows the results achieved using a different number of relevant documents in CAL500. Hence, the purpose of the evaluation is to search the rank of the relevant songs in the ranking list, and to average the metrics over the size of the query set.

6.3. Evaluation Metrics for Retrieval

For both tasks, we evaluate the ranking lists using standard information retrieval metrics [Manning et al. 2008]. We mostly focus on the top of the list, because it is the most interesting part for the user of a semantic music discovery engine. In fact, unlike other media where evaluation is almost immediate (e.g., many retrieved images can be presented at the same time, and it takes less than one second to address the relevance of an image), evaluating song relevance is a more time-consuming task. In fact, the user has to select and play the songs in the retrieved list and listen to at least several seconds of each song. Therefore, it is unlikely that a user would listen to more than the first 20–25 songs (i.e., which may correspond to about 2 hours of music).

In particular, we report the following metrics.

- Precision at k (Pk), which measures how many good results there are at the beginning of the list, reporting the fraction of relevant documents in the top-k positions. The values of k used in the experiments are 1, 3, 5, 10, and 20.
- Mean Reciprocal Rank (MRR), which averages the inverse of the rank of the first correct answer for each single query. MRR is a measure of the level of the ranking list at which the information need of the user is first fulfilled.
- Mean Average Precision (MAP), which averages the precision at each point in the ranking list where a song is correctly retrieved. Precision is defined as the fraction of retrieved documents which is relevant. MAP is a measure of the quality of the whole ranking list.

6.4. Compared Models

We compare the proposed model with some alternative approaches, which use either tags or acoustic similarity alone, as well as their combinations. We implemented these strategies in order to have a number-based comparison using the same music representations. A more general comparison with another state-of-the-art approach is provided in Section 8. In the experiments, we consider the following approaches.

- *TAG-Based* (TAG). This carries out retrieval using the semantic descriptions alone [Turnbull et al. 2007, 2008b]. This model ranks the songs according to their KL divergence with the query-tags. In the query-by-description scenario, a k-tag query is mapped into a query multinomial $\boldsymbol{q} = \{q_1, ..., q_{|\mathcal{V}|}\}$, where $q_i = 1$ if tag w_i of $\mathcal{V}$ is in the query, and $q_i = \epsilon$ where $1 >> \epsilon > 0$ otherwise. This multinomial is then normalized to one. In the query-by-example scenario $\boldsymbol{q}$ is the semantic multinomial of the seed song. Once we have a query multinomial $\boldsymbol{q}$, we rank all the songs in the database by their KL divergence with it.

—*Acoustic-Based* (AB). This ranks the songs by their direct acoustic similarity with the query [Mandel and Ellis 2005]. This model easily fits the query-by-example paradigm; conversely, it needs an extra effort to be adapted to the query-by-description scenario. In fact, we need to define a song to be used as starting point for the acoustic similarity (i.e., we rank songs according to their similarity with this song). In these experiments, for each query-tag we consider as seed the song ranked first by the TAG model, and we rank the other songs by the acoustic similarity with it (note that the ranking lists resulting with the TAG and AB models achieve the same P1).

—*Weighted Linear Combination* (WLC). This combines acoustic and semantic similarities in a single measure through a weighted linear combination[12]. In particular, if TAG(x, y) and AB(x, y) are the similarity values between the general songs x and y in the corresponding models, the WLC-based similarity is WLC$(x, y) = 0.6 \cdot \text{TAG}(x, y) + 0.4 \cdot \text{AB}(x, y)$. We chose the weighting coefficients that maximized the retrieval results. Retrieval is carried out in the same way as the AB model, leading to the same considerations. Simple linear combination may also be performed on the ranks of TAG and AB models (i.e., use independently the two models to rank the songs and then combine the ranking lists). However, in preliminary experiments, we did not obtain good results and we prefer to discard this option.

—*Post-Hoc Audio-Based Reranking* (PAR). This incorporates audio similarity into an already existing ranking [Knees et al. 2009]. It was originally proposed to improve a text-based music search engine that indexes songs based on related Web documents[13]. In our case, the algorithm modifies the ranking achieved by the TAG model by also including the acoustic similarity. Briefly, for each song x the PAR approach computes a new score that combines the tag-based rank of x, the tag-based rank of all the songs having x in their acoustic neighborhood, and the rank of x in all these neighborhoods. The songs are then sorted according to this new score. In the implementation, we followed the details reported in the referred paper.

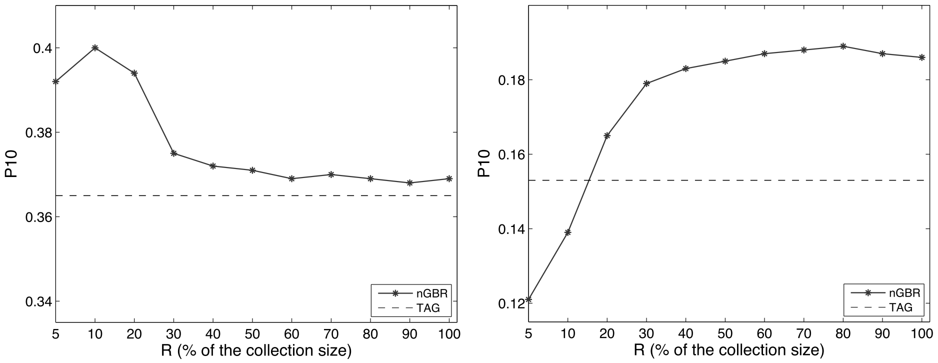
Additionally, each experiment includes as baseline a *random* model (Rand). The latter annotates songs by sampling ten tags from the vocabulary according to their prior distribution (i.e., it stochastically generates tags from a pool including the most frequently used annotations in the ground truth). These random semantic descriptions are then used to retrieve songs in the same way as the TAG model.

7. RESULTS

After discussing some preliminary results, this section provides the retrieval results on the CAL500 and CAL10k datasets. In all the tables reported, the symbol (*) after a numeric value means that the difference with the corresponding second best measurement in that experiment is statistically significant ($p < 0.05$, sign-test). The same statistical assumption is also implied when we talk about "*significant improvements*"

[12] In preliminary experiments we tested other different simple ways of combining acoustic and semantic similarities, including max, min, median, and geometric mean. For the sake of brevity we do not report these preliminary experiments; however, the weighted sum was the approach that achieved the best results.

[13] Knees et al. [2009] also propose a mechanism that incorporates the acoustic similarity directly in the tag-based scoring scheme. However, this approach is less general and relies on information about the number of Web pages mined when associating a tag to a song. Additionally the results described in that paper for this methodology are not significantly different to the PAR ones. For these reasons, we did not include it in this study.



(a) (b)

Fig. 3. The effects of R, that is, the number of edges outgoing from each state, on retrieval performances in terms of P10 (CAL500 dataset, social tags). (a) 1-tag query-by-description (149 tags) and (b) query-by-example (502 songs). The dashed line represents the performances of the TAG model, which is considered as baseline.

in the text without providing any supporting number. Along this section, for brevity, we will refer to our model as “nGBR” (novel Graph-Based Ranking).

7.1. Preliminary Experiment

Although the proposed framework is a general model, it is expected that some choices on the parameters affect the quality of the delivered ranking lists. In particular, as introduced in Section 3.1, one possible tuning option regards the connectivity of the model, that is the possibility of limiting the number of edges in the graph in order to improve scalability. In particular, the idea is to include in the model (i.e., in the transition matrix) for each state S_i only a subset $\mathcal{R}(S_i)$ of its outgoing transitions, that is, the ones towards the $|\mathcal{R}(S_i)|$ states which represent the most acoustically similar songs to song i.

In this experiment, we consider each state in the model having the same number of outgoing edges (i.e., $R = |\mathcal{R}(S_i)| = |\mathcal{R}(S_j)|$ for each generic states S_i and S_j), we range this value from 5% to 100% of the collection size, and we measure the P10 on the obtained ranking lists[14]. Results are reported in Figure 3(a) for 1-tag query-by-description (149 queries, i.e., all the tags in the vocabulary) and Figure 3(b) for query-by-example (502 query-songs, i.e., all the songs in the collection). In this preliminary experiment we use only the social tags (i.e., the Last.fm tags); the dashed line shows the performances of the TAG model which is considered as baseline.

As can be seen, the value of R strongly affects the retrieval performances. In the query-by-description, the model works better when R is small (between 5%–20% of the collection size), meaning that a little connected model gives higher-quality results. We believe this may depend on the fact that a few connections produce more discriminative similarity transitions which better combine with the observations. Conversely, in the query-by-example task, the graph needs to be more connected to obtain satisfactory ranking lists (i.e., R greater than 40% of the collection size).

[14] We consider each state having the same number of outgoing edges. Nevertheless, more sophisticated techniques could be exploited to set this number differently according to the characteristics of the song mapped in each state.

Table I. The Retrieval Results for 1-Tag Query-by-Description with CAL500

1-tag query-by-description - 149 cases								
Semantic	Model	P1	P3	P5	P10	P20	MRR	MAP
	Rand	0.214	0.185	0.186	0.174	0.161	0.358	0.179
cb-auto-tag	TAG	0.295	0.315	0.334	0.340	0.336	0.463	0.304
	AB	0.295	0.352	0.347	0.331	0.300	0.481	0.272
	WLC	0.295	0.341	0.360	0.340	0.319	0.471	0.279
	PAR	0.369	0.395	0.373	0.359	0.341	0.519	0.306
	nGBR	**0.405***	**0.420***	**0.405***	**0.380**	**0.355**	**0.561***	**0.331**
Last.fm	TAG	0.375	0.396	0.390	0.365	0.340	0.537	0.270
	AB	0.375	0.331	0.311	0.301	0.269	0.532	0.251
	WLC	0.375	0.341	0.319	0.313	0.282	0.536	0.255
	PAR	0.411	0.399	0.389	0.370	0.345	0.569	**0.276**
	nGBR	**0.461***	**0.440***	**0.431***	**0.400***	**0.347**	**0.629***	0.275

We consider 149 queries, that is all the distinct tags in the vocabulary.

Following these results and considerations, from now on, we generally set $R = 0.1 \cdot N$ for query-by-description, and $R = 0.6 \cdot N$ for query-by-example, where N is the number of states in the model.

7.2. Results on CAL500

This section presents the retrieval results for the CAL500 dataset; the nGBR model refers to the parameters setting discussed in the previous section. In this experiment, we computed the content-based autotags independently using a 5-fold cross-validation process (i.e., 100 different songs to be tagged in each fold), though we evaluate the retrieval algorithms over the entire dataset. It could be argued that applying the retrieval algorithms to each annotating set and then averaging the results over the folds would be a more appropriate procedure in order to keep train and test sets completely separate (as we do in Section 7.3 for the larger CAL10k dataset). Nevertheless, the challenging nature of multilabel classification makes overfitting an unlikely side-effect [Ness et al. 2009] that should not particularly affect the retrieval experiments. Therefore, considering the limited size of the collection, we prefer to run the experiments using all the songs. In fact, we believe that a retrieval experiment over the only 100 songs composing an annotation set may not be particularly relevant. At the same time, reducing the number of folds by using less examples for training could lead to not reliable autotags. Yet, in this way we can also provide a more relevant numerical comparison with the performances achieved by the models using the social tags[15].

7.2.1. Query-by-Description. Retrieval results for query-by-description are reported in Tables I, IIa, and IIb, for queries composed of 1, 2, and 3 tags, respectively. While the 1-tag task is performed over all the 149 tags of the vocabulary, in the other two cases we prefilter the query sets by considering all the tag combinations having at least 10 relevant songs in the ground truth. This is mainly done for discarding combinations that associate contradictory descriptions (e.g., “bitter” - “sweet”, “rock” - “classical music”). This leads to 3,684 distinct queries composed of 2 tags, and about 11,000 of

[15] For the sake of completeness, in a preliminary experiment we tested all the retrieval algorithms over the 5-fold annotation sets in the 1-tag query-by-description task, and we found that the results are comparable to those achieved when using the whole collection. Indeed, for example, nGBR achieves a MAP of 0.38, while TAG and PAR (which generally are the other best working models) achieve 0.33 and 0.34, respectively. In a similar way, the MRR achieved by nGBR improves by about 0.08 and 0.12 the performances of PAR and TAG, respectively.

Table II. The Retrieval Results for Query-by-Description Using a Combination of 2 (a) and 3 (b) Tags over All the 149 Tags of the CAL500 Dataset

2-tag query-by-description - 3,684 cases				
Semantic	Model	P5	P10	MRR
	Rand	0.074	0.074	0.190
cb-auto-tag	TAG	0.185	0.190	0.335
	AB	0.195	0.183	0.344
	WLC	0.196	0.188	0.341
	PAR	0.199	0.201	0.361
	nGBR	**0.234***	**0.233***	**0.401***
Last.fm	TAG	0.148	0.140	0.309
	AB	0.149	0.151	0.316
	WLC	0.166	0.159	0.327
	PAR	0.152	0.144	0.319
	nGBR	**0.201***	**0.190***	**0.384***

3-tag query-by-description - 3,000 cases				
Semantic	Model	P5	P10	MRR
	Rand	0.047	0.048	0.140
cb-auto-tag	TAG	0.106	0.101	0.238
	AB	0.151	0.124	0.261
	WLC	0.134	0.136	0.261
	PAR	0.139	0.126	0.298
	nGBR	**0.169***	**0.178***	**0.341***
Last.fm	TAG	0.081	0.085	0.233
	AB	0.091	0.096	0.222
	WLC	0.101	0.099	0.233
	PAR	0.099	0.096	0.241
	nGBR	**0.129***	**0.119***	**0.296***

Each query has at least 10 relevant songs. For the 3-tag scenario, we sample a random subset of 3,000 queries.

3 tags; in this last case, we retain a random sample of 3,000 queries, assuming them generally enough to evaluate the task.

As can be seen, the proposed model generally outperforms all the other algorithms over all the experiments, both with content-based autotags and Last.fm tags. The major benefits are at the top of the ranking list; in particular, the improvements in P1, P3, P5, and MRR are statistically significant compared to the PAR algorithm (which generally is the second best model), as well as to the TAG model. Conversely, retrieval along the full ranking list tends to decrease the effectiveness, as can be inferred in Table I by the minor improvement of the MAP. Nevertheless, we argue again that the most important aspect of a music ranking list is the quality at its top. For this reason Tables IIa and IIb report top-ranking measures only; in these scenarios, nGBR works even better with significant improvements also for P10, showing a good robustness to multitag queries.

Note that most of the results based on Last.fm tags tend to show competitive precision at the top and worse along the whole list with respect to the results obtained with the content-based autotags. This depends on the fact that the Last.fm representation is rather sparse and noisy, since tags may also not have been assigned. When a tag is not assigned to a song, the retrieval algorithms rank that song randomly or just solely on the basis of acoustic similarity. This leads to a less precise bottom part of the list, which affects the measurements. Conversely, tags generated through the autotaggers are more stable since each song has a relevance value for each tag, and no song is retrieved randomly or according to acoustic similarity alone.

A deeper analysis of the results in the 1-tag query-by-description showed that the improvement in P10 with respect to the TAG model involves 123 and 120 tags for the content-based autotags and the Last.fm tags, respectively (about 80% of the queries); conversely, the PAR algorithm improves only about 70% of the queries. The improvement generally happens in all the tag categories, meaning that there are no types of tags that gain a major benefit (CAL500 spans different categories of tags, such as genre, emotion, usage, etc.). A similar result is achieved with 2-tag queries, while the improvement in the 3-tag scenario is even greater with about 88% of the queries outperforming the TAG model in terms of P10.

Lastly, Table III compares the top five positions in the ranking list achieved by nGBR and TAG for 5 representative 1-tag queries (content-based autotags). This table shows some examples of the results delivered to a user of a semantic music discovery engine, especially highlighting the benefits introduced by the proposed framework. For example, in response to the query “emotional”, nGBR ranks five relevant songs at the top, compared with only one ranked by the TAG model.

Table III. The Top 5 Songs in the Ranking Lists Obtained by the nGBR and TAG Models for 5 Representative 1-Tag Queries from the CAL500 Vocabulary

Rank	nGBR	TAG
	classic rock	
1	**The Cure - Just like heaven (*)**	Electric Frankenstein - Teenage shut down
2	**Bruce Springsteen - Badlands (*)**	**The Cure - Just like heaven (*)**
3	**The Adverts - Gary Gilmore's eyes (*)**	Joy Division - Love will tear us apart
4	**The Metallica - One (*)**	**The Adverts - Gary Gilmore's eyes (*)**
5	Electric Frankenstein - Teenage shut down	**Boston - More than a feeling (*)**
	trumpet	
1	**B.B. King - Sweet little angel (*)**	Paul McCartney - Ebony and ivory
2	Paul McCartney - Ebony and ivory	Carl Perkins - Matchbox
3	Beach Boys - I get around	**B.B. King - Sweet little angel (*)**
4	**The Doors - Touch me (*)**	Captain Beefheart - Safe as milk
5	Carl Perkins - Matchbox	The Jefferson Airplane - Somebody to love
	driving	
1	Buzzcocks - Everybody's happy nowadays	Radiohead - Karma police
2	**Weezer - Buddy Holly (*)**	Tom Waits - Time
3	**Big Star - In the street (*)**	Drevo - Our watcher, show us the way
4	**Pearl Jam - Yellow Ledbetter (*)**	Buzzcocks - Everybody's happy nowadays
5	Radiohead - Karma police	The Napoleon Blown Aparts - Higher education
	happy	
1	**Jerry Lee Lewis - Great balls of fire (*)**	**Creedence CR - Travelin' band (*)**
2	**Creedence CR - Travelin' band (*)**	**Jackson 5 - ABC (*)**
3	Louis Armstrong - Hotter than that	Ray Charles - Hit the road Jack
4	Rolling Stones - Little by little	Louis Armstrong - Hotter than that
5	**Stevie Wonder - For once in my life (*)**	ABC - Poison Arrow
	emotional	
1	**Alicia Keys - Fallin' (*)**	Van Morrison - And it stoned me
2	**Shakira - The one (*)**	Clarence Ashley - The house carpenter
3	**Chantal Kreviazuk - Surrounded (*)**	Booker T and the MGS - Time is tight
4	**Evanescence - My immortal (*)**	Stooges - Dirt
5	**Carpenters - Rainy days and Mondays (*)**	**Alicia Keys - Fallin' (*)**

We refer to the content-based autotags experiments; relevant songs are listed in bold and marked as "significant" (i.e., (*)).

7.2.2. Query-by-Example. The query-by-example results for CAL500 are presented in Table IV. We use all the songs of the dataset as query; therefore the results are averaged over 502 results. We consider 30 relevant songs for each query, which have been gathered as described in Section 6.2.

As can be seen, also in this scenario nGBR generally outperforms all the other algorithms. Major improvements are achieved with the Last.fm tags (significant for P1, P3, P10, and MRR). These results show that using the proposed model to combine completely different sources of information (in this case social tags and music content) is a competitive strategy. The improvement of P5 is not significant at the 5% level; however, the sign-test obtains $p = 0.056$, that is, very close to being significant. Conversely, results with the content-based autotags show significant improvement for the P3, P10, and MRR only.

It should be noted that this evaluation may be slightly affected by the algorithm used to choose the relevant documents. In fact, for each query we automatically choose in a heuristic way the first 30 most similar songs according to the human-based annotations. Nevertheless, some queries could not have 30 "true" similar songs. For example, the tag "swing" has only 5 human-annotated examples in the ground truth; this means that these songs could not have up to 30 songs which can be truly

Table IV. The Retrieval Results in Query-by-Example with 502 Queries over the CAL500 Dataset

query-by-example - 502 cases								
Semantic	Model	P1	P3	P5	P10	P20	MRR	MAP
	Rand	0.065	0.071	0.073	0.071	0.066	0.205	0.074
	AB	0.178	0.163	0.161	0.141	0.129	0.341	0.119
cb-auto-tag	TAG	0.260	0.217	0.187	0.141	0.133	0.410	0.133
	WLC	0.254	0.208	0.184	0.143	0.139	0.406	0.134
	PAR	0.265	0.210	0.189	0.146	0.138	0.407	0.136
	nGBR	**0.286**	**0.230***	**0.201**	**0.176***	**0.152**	**0.458***	**0.145**
Last.fm	TAG	0.197	0.177	0.163	0.153	0.139	0.359	0.131
	WLC	0.204	0.184	0.179	0.159	0.146	0.370	0.132
	PAR	0.201	0.176	0.171	0.158	0.146	0.359	0.126
	nGBR	**0.281***	**0.222***	**0.197**	**0.187***	**0.161**	**0.443***	**0.145**

considered similar in the dataset ("swing" is quite a strong discriminative music genre). Therefore, the evaluation could have also been done by searching songs that are not truly relevant (i.e., these songs are in the top-30 only because less *dissimilar* to the query than the others).

For this reason we also provide Figure 4, which depicts the performances of the different models in terms of P10 and MRR by ranging the automatic number of relevant songs from 5–100. As can be seen, nGBR generally outperforms all the other models in any case. Note that when the number of relevant documents is higher, all the models tend to converge to similar performances.

7.3. Results on CAL10k

This section presents the results with the CAL10k database. The purpose of the following experiment is two-fold. On the one hand, we aim at testing the framework with a larger collection, both in terms of tags and songs. On the other hand, the test also involves the music content descriptors; in fact, as mentioned in Section 5.1, the CAL10k songs must be represented using delta-ENTs to accommodate copyright issues.

From the original semantic vocabulary we retain only the tags with at least 10 examples in the ground truth in order to have a reasonable minimum number of songs to search during the evaluation. The final vocabulary is then composed of 485 distinct tags (119 "genre" and 366 "acoustic" tags). Note that no subjective tags such as emotions or usages are included in this database. With this dataset, we use only the content-based autotags for semantically describing each song in order to have more dense descriptions. In fact, the Pandora-based CAL10k tags are assigned to songs by experts and musicologists, and sometimes prove too specific and technical to be also given by normal Last.fm users (e.g., "slide/pedal steel guitars", "electric guitar wall-o-sound", "dominant melodic hooks"). Therefore, the resulting Last.fm-based descriptions would be too sparse to deliver reliable ranking lists (i.e., where a few random-based sorts are involved) for evaluation purposes.

In this case, the size of the collection makes it possible to perform a proper cross-validation experiment, thus keeping the training songs used to estimate the tag models completely separate from the songs used in the retrieval experiments. With this aim in mind we run a 5-fold cross-validation process, using about 2,000 songs in each training set and the remaining 8,000 in the test set (i.e., we assign the content-based autotags using the estimated tag models to the songs in the test set and we apply the retrieval algorithms on this set only). The split is done by considering each song being in the training set exactly once, and additionally by applying an artist-filter which prevents having the same artist on both training and test sets.

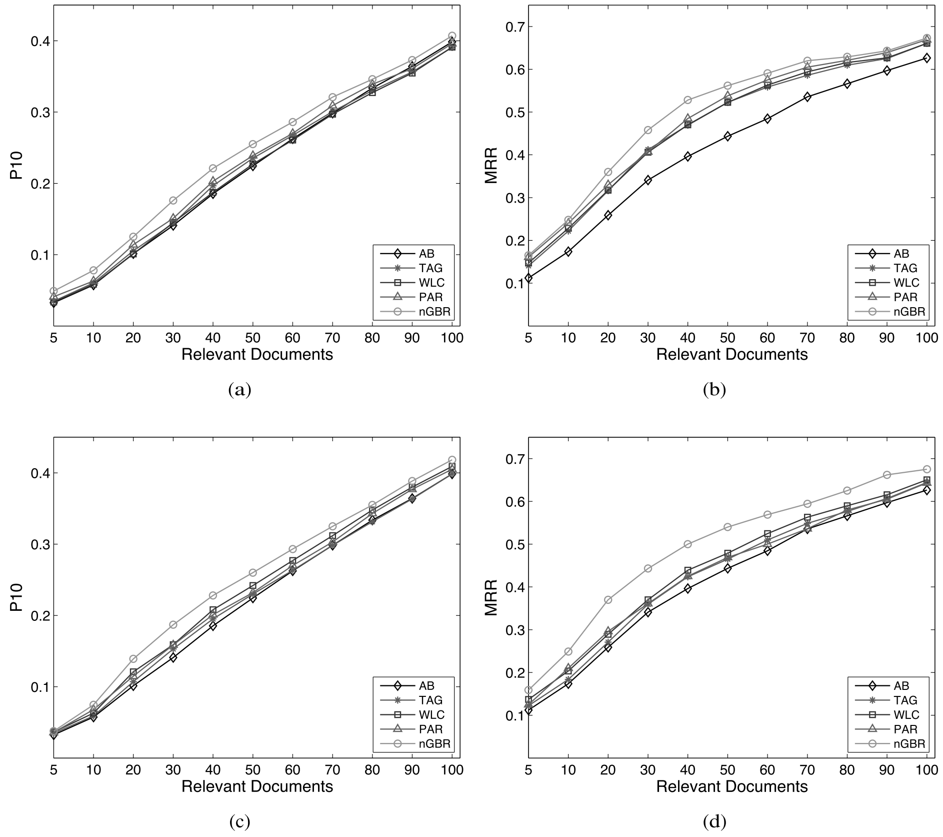


Fig. 4. Retrieval results in query-by-example for different numbers of relevant songs in terms of P10 and MRR; (a),(b) content-based autotags, (c),(d) Last.fm tags.

During preliminary experiments, not reported here for brevity, we saw that the framework built using all the songs of the database leads to low retrieval results. We believe this may depend on the normalization performed on the transition probabilities to satisfy the stochastic requirement that help to reduce the numerical sparsity of the relationships. In fact, with many edges outgoing from each state (e.g., about 4,800 for query-by-example), the normalization leads to similar transition values which do not discriminate well the paths across the model.

Nevertheless, we argue that in the real scenario of commercial search engines, it would be computationally complex to build the graph with all the millions of songs indexed as well. For this reason, we propose using the nGBR model to refine the retrieval on a reliable subset of the collection (i.e., reliable in the sense that ideally all the relevant songs are in this subset). One method to extract this subset is to precluster the songs using metadata such as title, artist, year, user preferences, etc. Since we search by tags and we do not have many other additional pieces of information to exploit in this scenario (e.g., artists similarity), we use the TAG model as clustering algorithm. In particular, for each query, we consider the top 3,000 results achieved by the TAG ranking and we use nGBR to rerank this subset. We measured that the recall-at-3,000 (i.e., the number of relevant documents retrieved over the total number of relevant documents) achieved by the TAG model is about 85% for both tasks;

Table V. The Retrieval Results for the 1-Tag Query-by-Description with the 485 Tags of the CAL10k Dataset

1-tag query-by-description - 485 cases								
Semantic	Model	P1	P3	P5	P10	P20	MRR	MAP
	Rand	0.031	0.030	0.025	0.024	0.022	0.079	0.024
cb-auto-tag	TAG	0.219	0.212	0.218	0.228	0.226	0.364	0.148
	AB	0.219	0.256	0.251	0.211	0.193	0.392	0.089
	WLC	0.219	0.249	0.253	0.221	0.195	0.378	0.119
	PAR	0.277	0.259	0.258	0.236	0.235	0.436	0.108
	nGBR	**0.321***	**0.296***	**0.278**	**0.251**	**0.248**	**0.473***	**0.165**

Table VI. The Retrieval Results in Query-by-Example with 5,000 Random Queries in the CAL10k Dataset

query-by-example - 5,000 cases								
Semantic	Model	P1	P3	P5	P10	P20	MRR	MAP
	Rand	0.006	0.006	0.006	0.007	0.005	0.028	0.007
cb-auto-tag	TAG	0.125	0.112	0.109	0.099	0.081	0.239	0.041
	AB	0.119	0.102	0.099	0.088	0.074	0.225	0.049
	WLC	0.166	0.136	0.121	0.102	0.091	0.283	0.055
	PAR	0.171	0.141	0.125	0.103	**0.093**	0.287	0.057
	nGBR	**0.219***	**0.172***	**0.150***	**0.125***	0.090	**0.371***	**0.058**

therefore the resulting clusters can be considered reliable enough to be used as input for the proposed algorithm.

We use nGBR with the same parameters guideline defined in Section 7.1. The query-by-description task is performed over 485 1-tag queries (i.e., all the tags of the vocabulary); results are reported in Table V. We did not investigate the 2-tag and 3-tag queries scenario with this dataset. As can be seen, the proposed model generally leads to the best results again. First, we obtained a significant improvement with respect to TAG and AB in most of the metrics; additionally, results are significantly better on P1, P3, and MRR, and never worse than the PAR model (which generally is the second best model at the top of the ranking lists). It can also be noted that AB works better than with CAL500; this is due to the more qualitative timbre descriptions provided by the delta-ENT features.

Lastly, Table VI shows results for query-by-example; in this experiment we pseudo-randomly sampled 1,000 songs from each test set to use as seed queries (i.e., globally we evaluate the system over 5,000 different queries)[16]. Again, nGBR outperforms the other models with significant improvements at the top of the ranking list.

8. DISCUSSION AND FUTURE WORKS

This article presents a novel approach for music retrieval based on a graph-based representation of a music collection that combines acoustic similarity and tags in a single probabilistic framework. Each state of the model represents a song of the music collection, where the weight of the edges is ruled by acoustic similarity between the songs, while tags define the state emissions. A decoding algorithm is used to retrieve the list of songs that best responds to a user query, which is expressed either as a combination of tags or as a seed song. In the presence of tagged music library, the framework does not require any additional preprocessing steps (e.g., training); additionally, efficiency

[16] With the term *pseudo-random* we mean that choosing the query songs for each test set was actually based on a random process, but then a picked song that was discarded and replaced (with the same process) if already used as a query in a previous fold.

is guaranteed by subretrieval steps which reduce the waiting time of the user. This approach aims at integrating and improving state-of-the-art systems for semantic music search and discovery engines, recommendation, and playlist generation.

Experimental evaluation shows that the proposed model generally outperforms other state-of-the-art approaches that rank songs by a single source of information alone, or by a combination of them. Improvements are more significant at the top of the ranking lists and when the size of the collection is smaller. In contrast, a larger collection brought up some performance issues; however, the model is particularly suitable for reranking a cluster of relevant candidates efficiently retrieved from a collection. An interesting consideration that may be inferred from the experimental results regards the significant improvement achieved with respect to the tag-based retrieval. In fact, Barrington et al. [2009] show that a recommender system built on content-based autotags (i.e., GMM model) alone can expect to perform similarly or even better than Apple iTunes Genius when exploring the *long tail*. Genius is based on collaborative filtering of huge amounts of user data; however, in the case of less well-known music this massive quantity of data is missing and Genius is unable to make good recommendations. In the content-based autotags scenario, our system improves the tag-based retrieval alone and it is therefore expected to improve the state-of-the-art for music discovery of the long tail also with respect to commercial systems built over collaborative filtering techniques.

To the best of our knowledge, the models included in the comparison represent the state-of-the-art concerning systems that combine acoustic similarity and tags for music search and discovery. Alternative approaches in the literature use other sources of information, such as artist- and user-related data, or Web documents, and combine them for other retrieval tasks (e.g., artist clustering and recommendation, classification). However, a comparison with these methodologies goes beyond the scope of this article. We believe that the high generality of the model makes it possible to integrate such different descriptions as well.

Nevertheless, one additional approach that should be mentioned is Turnbull et al. [2009]; in this work, the authors combine semantic descriptions (i.e., SMNs) for a music discovery task. In particular, they consider content-based autotags, social tags, and tags mined from Web pages, while the combination is carried out through three different machine learning approaches. Experiments on 72 tags of CAL500, one of the collections used in our experiments, show that the combination approach called Calibrated Score Averaging (CSA) outperforms the others (including the use of single descriptors) in terms of MAP (1-tag query-by-description). When examining each tag individually, they report that the percentage of tags that are improved by each combined model with respect to the best retrieval achieved using individual descriptors ranged from 15%–27%; in particular, CSA improves 22% of the tags. While a global direct comparison based on the absolute value of the measures cannot be done because of the smaller vocabulary and of the shorter ranking lists due to the cross-validation procedure, we can compare improvements when tags are taken individually. To this end, we detect that for experiments in query-by-description using a single tag, the proposed model improves MAP for 67% of the content-based autotags and 54% for Last.fm. This higher improvement may be due to the fact that our approach exploits the graph structure induced by the acoustic similarity between all the songs in the collection.

Future work can be carried out in different directions. First, the integration with collaborative filtering techniques can lead to a more powerful hybrid discovery system which could provide better recommendations for all music. For instance, this can be done by exploiting user-related data in the definition of transition probabilities to take into account the subjectivity of music similarity, thus without modifying the retrieval algorithm. Second, it is possible to study the application of the retrieval framework when dealing with query tags which dynamically change over time; however, this would first require the definition of an appropriate ground-truth dataset for the task.

Another possible future direction concerns the inversion of the roles played by acoustic descriptors and tags. That is, emissions can be related to the acoustic content, whereas transitions probabilities can be based on semantic similarity. In the formulation proposed in this article, we use tags as state emissions because this is more naturally related to the way humans describe music (which is by textual semantic labels), and because in this way we can query the model using both tags and seed songs. Conversely, the reversed formulation allows for querying the model using only seed songs, because the sequence of observations can be defined by audio feature vectors only. Nevertheless, using the acoustic content as state emissions, and therefore as means to query the model, could be exploited in other audio-related search operations.

Lastly, the high generality of the model makes it suitable for other media, such as images and videos. In particular, we intend to explore its applicability to image retrieval tasks, where both textual descriptors and content-based features are available.

ACKNOWLEDGMENTS

The authors would like to thank the editor and reviewers for their constructive and insightful comments, as well as Giorgio Maria Di Nunzio, Gert Lanckriet, Nicola Montecchio, Emanuele Di Buccio, and Gianmaria Silvello for helpful discussions and support. The authors also wish to acknowledge the support provided by the Information Management Systems research group led by Professor Maristella Agosti at the Department of Information Engineering of the University of Padova, Italy.

REFERENCES

AGARWAL, S. 2006. Ranking on graph data. In *Proceedings of the International Conference on Machine Learning (ICML'06)*. 25–32.

BARRINGTON, L., ODA, R., AND LANCKRIET, G. 2009. Smarter than genius? Human evaluation of music recommender systems. In *Proceedings of the International Society for Music Information Retrieval (ISMIR'09)*. 357–362.

BERENZWEIG, A., LOGAN, B., ELLIS, D., AND WHITMAN, B. 2004. A large-scale evaluation of acoustic and subjective music-similarity measures. *Comput. Music J. 28*, 63–76.

BERTIN-MAHIEUX, T., ECK, D., MAILLET, F., AND LAMERE, P. 2008. Autotagger: A model for predicting social tags from acoustic features on large music databases. *J. New Music Res. 37*, 2, 115–135.

BERTIN-MAHIEUX, T., WEISS, R., AND ELLIS, D. 2010. Clustering beat-chroma patterns in a large music database. In *Proceedings of the International Society for Music Information Retrieval (ISMIR'10)*. 111–116.

BU, J., TAN, S., CHEN, C., WANG, C., WU, H., ZHANG, L., AND HE, X. 2010. Music recommendation by unified hypergraph: Combining social media information and music content. In *Proceedings of the ACM Multimedia Conference*. 391–400.

CARNEIRO, G., CHAN, A., MORENO, P., AND VASCONCELOS, N. 2007. Supervised learning of semantic classes for image annotation and retrieval. *IEEE Trans. Pattern Anal. Mach. Intell. 29*, 3, 394–410.

CASEY, M., RHODES, C., AND SLANEY, M. 2008a. Analysis of minimum distances in high-dimensional musical spaces. *IEEE Trans. Audio, Speech Lang. Process. 5*, 16, 1015–1028.

CASEY, M., VELTKAMP, R., GOTO, M., LEMAN, M., RHODES, C., AND SLANEY, M. 2008b. Content-Based music information retrieval: Current directions and future challenges. *Proc. IEEE 96*, 4, 668–696.

CELMA, O. 2008. Music recommendation and discovery in the long tail. Ph.D. thesis, Universitat Pompeu Fabra, Barcelona.

CELMA, O., CANO, P., AND HERRERA, P. 2006. Search sounds: An audio crawler focused on Web-logs. In *Proceedings of the International Society for Music Information Retrieval (ISMIR'06)*. 365–366.

COVIELLO, E., CHAN, A., AND LANCKRIET, G. 2011. Time series models for semantic music annotation. *IEEE Trans. Audio, Speech Lang. Process. 19*, 5, 1343–1359.

DEMPSTER, A., LAIRD, N., AND RUBIN, D. 1977. Maximum likelihood from incomplete data via the EM algorithm. *J. Roy. Statist. Soc. B39*, 1, 1–38.

Downie, J. 2008. The music information retrieval evaluation exchange (2005–2007): A window into music information retrieval research. *Acoust. Sci. Technol. 29*, 4, 247–255.

Feng, S., Manmatha, R., and Lavrenko, V. 2004. Multiple Bernoulli relevance models for image and video annotation. In *Proceedings of the IEEE Conference on Computerc Vision and Pattern Recognition (CVPR'04)*. 1002–1009.

Fields, B., Jacobson, K., Rhodes, C., d'Inverno, M., Sandler, M., and Casey, M. 2011. Analysis and exploitation of musician social networks for recommendation and discovery. *IEEE Trans. Multimedia 13*, 4, 674–686.

Flexer, A., Schnitzer, D., Gasser, M., and Pohle, T. 2010. Combining features reduces hubness in audio similarity. In *Proceedings of the International Society for Music Information Retrieval (ISMIR'10)*. 171–176.

Forney, G. 1973. The Viterbi algorithm. *Proc. IEEE 61*, 3, 268–278.

Hoffman, M., Blei, D., and Cook, P. 2008. Content-Based musical similarity computation using the hierarchical Dirichlet process. In *Proceedings of the International Society for Music Information Retrieval (ISMIR'08)*. 349–354.

Hoffman, M., Blei, D., and Cook, P. 2009. Easy as CBA: A simple probabilistic model for tagging music. In *Proc. of ISMIR*. 369–374.

Jensen, J., Christensen, M., Ellis, D., and Jensen, S. 2009. Quantitative analysis of a common audio similarity measure. *IEEE Trans. Audio, Speech Lang. Process. 17*, 4, 693–703.

Knees, P., Pohle, T., Schedl, M., and Widmer, G. 2007. A music search engine built upon audio-based and Web-based similarity measures. In *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval*. 23–27.

Knees, P., Pohle, T., Schedl, M., Schnitzer, D., Seyerlehner, K., and Widmer, G. 2009. Augmenting text-based music retrieval with audio similarity. In *Proceedings of the International Society for Music Information Retrieval (ISMIR'09)*. 579–584.

Kullback, S. and Leibler, R. 1951. On information and sufficiency. *Ann. Math. Statist. 12*, 2, 79–86.

Lamere, P. 2008. Social tagging and music information retrieval. *J. New Music Res. 37*, 2, 101–114.

Logan, B. 2000. Mel frequency cepstral coefficients for music modeling. In *Proceedings of the International Society for Music Information Retrieval (ISMIR'00)*.

Mandel, M. and Ellis, D. 2005. Song-level features and support vector machines for music classification. In *Proceedings of the International Society for Music Information Retrieval (ISMIR'05)*. 594–599.

Mandel, M. and Ellis, D. 2008. Multiple-instance learning for music information retrieval. In *Proceedings of the International Society for Music Information Retrieval (ISMIR'08)*. 577–582.

Manning, C., Raghavan, P., and Schtze, H. 2008. *Introduction to Information Retrieval*. Cambridge University Press.

McFee, B. and Lanckriet, G. 2009. Heterogenous embedding for subjective artist similarity. In *Proceedings of the International Society for Music Information Retrieval (ISMIR'09)*. 513–518.

Miotto, R. and Lanckriet, G. 2012. A generative context model for semantic music annotation and retrieval. *IEEE Trans. Audio, Speech Lang. Process. 20*, 4, 1096–1108.

Miotto, R. and Orio, N. 2010. A probabilistic approach to merge context and content information for music retrieval. In *Proceedings of the International Society for Music Information Retrieval (ISMIR'10)*. 15–20.

Miotto, R., Montecchio, N., and Orio, N. 2010. Statistical music modeling aimed at identification and alignment. In *Advances in Music Information Retrieval*, Z. Ras and A. Wieczorkowska Eds., Springer, 187–212.

Ness, S., Theocharis, A., Tzanetakis, G., and Martins, L. 2009. Improving automatic music tag annotation using stacked generalization of probabilistic SVM outputs. In *Proceedings of the ACM Multimedia Conference*. 705–708.

Orio, N. 2006. Music retrieval: A tutorial and review. *Found Trends Inf. Retriev. 1*, 1, 1–90.

Pampalk, E. 2006. Computational models of music similarity and their application to music information retrieval. Ph.D. thesis, Vienna University of Technology.

Rabiner, L. 1989. A tutorial on hidden Markov models and selected application. *Proc. IEEE 77*, 2, 257–286.

Raphael, C. 1999. Automatic segmentation of acoustic musical signals using hidden Markov models. *IEEE Trans. Pattern Anal. Mach. Intell. 21*, 4, 360–370.

Rasiwasia, N. and Vasconcelos, N. 2007. Bridging the semantic gap: Query by semantic example. *IEEE Trans. Multimedia 9*, 5, 923–938.

SEYERLEHNER, K., WIDMER, G., AND KNEES, P. 2008. Frame level audio similarity – A codebook approach. In *Proceedings of the International Conference on Digital Audio Effects (DAFx'08)*. 349–356.

SHIFRIN, J., PARDO, B., MEEK, C., AND BIRMINGHAM, W. 2002. HMM-based musical query retrieval. In *Proceedings of the ACM/IEEE Joint Conference on Digital Libraries (JCDL'02)*. 295–300.

SLANEY, M., WEINBERGER, K., AND WHITE, W. 2008. Learning a metric for music similarity. In *Proceedings of the International Society for Music Information Retrieval (ISMIR'08)*. 313–318.

SORDO, M., LAURIER, C., AND CELMA, O. 2007. Annotating music collections: How content-based similarity helps to propagate labels. In *Proceedings of the International Society for Music Information Retrieval (ISMIR'07)*. 531–534.

TINGLE, D., KIM, Y., AND TURNBULL, D. 2010. Exploring automatic music annotation with "acoustically-objective" tags. In *Proceedings of the ACM International Conference on Multimedia Retrieval (ICMR'10)*. 55–61.

TOMASIK, B., KIM, J., LADLOW, M., AUGAT, M., TINGLE, D., WICENTOWSKI, R., AND TURNBULL, D. 2009. Using regression to combine data sources for semantic music discovery. In *Proceedings of the International Society for Music Information Retrieval (ISMIR'09)*. 405–410.

TOMASIK, B., THIHA, P., AND TURNBULL, D. 2010. Beat-sync-mash-coder: A web application for real-time creation of beat-synchronous music mashups. In *Proc. of IEEE ICASSP*. 437–440.

TSAI, C. AND HUNG, C. 2008. Automatically annotating images with keywords: A review of image annotation systems. *Recent Patents Comput. Sci 1*, 55–68.

TURNBULL, D., BARRINGTON, L., TORRES, D., AND LANCKRIET, G. 2007. Towards musical query-by-semantic description using the CAL500 data set. In *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval*. 439–446.

TURNBULL, D., BARRINGTON, L., AND LANCKRIET, G. 2008a. Five approaches to collecting tags for music. In *Proceedings of the International Society for Music Information Retrieval (ISMIR'08)*. 225–230.

TURNBULL, D., BARRINGTON, L., TORRES, D., AND LANCKRIET, G. 2008b. Semantic annotation and retrieval of music and sound effects. *IEEE Trans. Audio, Speech Lang. Process. 16*, 2, 467–476.

TURNBULL, D., BARRINGTON, L., LANCKRIET, G., AND YAZDANI, M. 2009. Combining audio content and social context for semantic music discovery. In *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval*. 387–394.

VASCONCELOS, N. AND LIPPMAN, A. 1998. Learning mixture hierarchies. In *Proceedings of the Conference on Advances in Neutral Information Processing Systems (NIPS'98)*. 606–612.

WANG, D., LI, T., AND OGIHARA, M. 2010. Are tags better than audio features? The effects of joint use of tags and audio content features for artistic style clustering. In *Proceedings of the International Society for Music Information Retrieval (ISMIR'10)*. 57–62.

YANG, Y., LIN, Y., LEE, A., AND CHEN, H. 2009. Improving musical concept detection by ordinal regression and context fusion. In *Proceedings of the International Society for Music Information Retrieval (ISMIR'09)*. 147–152.

Received February 2011; revised September 2011; accepted December 2011