Accessing Music Digital Libraries by Combining Semantic Tags and Audio Content

Riccardo Miotto and Nicola Orio

Department of Information Engineering, University of Padua
Via Gradenigo, 6/a – 35131 Padua, Italy
{miottori,orio}@unipd.it

Abstract. An interesting problem in accessing music digital libraries is how to combine the information of different sources in order to improve the retrieval effectiveness. This paper introduces an approach to represent a collection of tagged songs through an hidden Markov model with the purpose to develop a system that merges in the same framework both acoustic similarity and semantic descriptions. The former provides content-based information on song similarity, the latter provides context-aware information about individual songs. Experimental results show how the proposed model leads to better performances than approaches that rank songs using both a single information source and a their linear combination.

1 Introduction

The widespread diffusion of digital music occurred during the last years has brought music information retrieval (MIR) to the general attention. A central goal of MIR is to create systems that can efficiently and effectively retrieve songs from collections of music content (e.g. music digital libraries) according to some sense of similarity with a given query. In information retrieval systems, the concept of similarity plays a key role and can dramatically impact performances. Yet, in music applications, the problem of selecting an optimal similarity measure is even more difficult because of the intrinsic subjectivity of the task: users may not consistently agree upon whether, or at which degree, a pair of songs or artists are similar.

In the last years, in order to deal with the subjective nature of music similarity, it became very common to describe songs as a collection of meaningful terms, or *tags*, as done in Last.fm[1] and Pandora[2]. In particular, tags are often, directly or indirectly, provided by end users and can represent a variety of different concepts including genre, instrumentation, emotions, geographic origins, and so on. Many approaches have been developed to collect tags, ranging from mining the Web and exploiting social behavior of users, to automatic annotation of music through machine learning algorithms. Tags are useful because they contextualize

[1] http://www.last.fm
[2] http://www.pandora.com

M. Agosti et al. (Eds.): IRCDL 2011, CCIS 249, pp. 26–37, 2011.

a song – for instance describing an historical period, a geographical area, or a particular use of the song – through an easy high-level representation. This information can then be used to retrieve music documents, to provide recommendations or to generate playlists.

Excluding the case of Pandora, where songs are annotated by human experts to guarantee high quality and consistency, in automatic systems or when the social behavior of users is kept into account, the semantic descriptions may be very noisy. In automatic approaches, for example, the quality of the prediction strictly depends on the quality of the training set, on the quality of the model, and on other issues such as parameter overfitting or term normalization. On the other hand, standard content-based music similarity, computed directly on music features, can be exploited to improve the quality of the retrieval, without requiring additional training operations.

The goal of this paper is to provide a general model to describe a music collection and easily retrieve songs combining both content-based similarity and context-aware tag descriptions. The model is based on an application of hidden Markov models (HMMs) and of the Viterbi algorithm to retrieve music documents. The main applicative scenario is cross-domain music retrieval, where music and text information sources are merged.

1.1 Related Work

There has been a considerable amount of research devoted to the topic of music retrieval, recommender systems and music similarity. Some of the most well-known commercial and academic systems have been described in [1]. The model proposed in this paper fits the scenario of item-based retrieval systems, combining pure acoustic similarity and semantic descriptions.

Methodologies that merge different heterogeneous sources of information have been recently proposed for the task of semantic discovery [2], artist recommendation [3] and music classification [4]. All of these approaches learn a metric space to join and compare the different sources of information in order to provide the user with a single ranking list. Our approach is consistently different, because it is built on a graph-based representation of the collection that model both sources of information and thus it does not rely on an additional processing to combine them. Content-based music similarity can be computed directly on music features [5, 6] or through a semantic space which describes music content with meaningful words [7, 8]. In our work, we exploit the properties of an HMM to combine these two descriptions to improve retrieval performances.

As it is well known, HMMs have been extensively used in many applications, which in particular involve processes through time such as speech recognition [9]. In the music information retrieval research area, they have been used in different scenarios: query-by-example [10], automatic identification [11], alignment [12], segmentation [13], and chord recognition [14]. At the best of our knowledge, this is the first application of HMMs in the task of cross-domain retrieval where music and text information is modeled in a single framework.

2 Statistical Modeling of a Music Collection

The general goal of accessing to music digital libraries is to retrieve a list of songs according to a particular principle. The principle could be described either directly by a general semantic indication, such as the tag "classic rock", or indirectly by a song, such as the set of tags assigned to "Yesterday, The Beatles". In both cases, the principle represents a user information need, and it can be assumed that the goal of an user is to *observe* consistently the application of this principle during the time of his access to the music collection. In the particular case of playlist generation, a system should be able to retrieve a list of music documents that are acoustically similar to the music the user likes and, at the same time, are relevant to one or more semantic labels that give a context to his information need.

The methodology presented in this paper aims at providing a formal and general model to retrieve music documents combining acoustic similarity and semantic descriptions given by social tags. That is, the goal is to propose a model that encompasses both content-based similarity and context-aware descriptors. To this end, HMMs are particularly suitable because they allow us to model two different sources of information. In fact, HMMs represent a doubly embedded stochastic process where, at each time step, the model performs a transition to a new state according to transition probabilities and emits a new symbol according to observation probabilities.

Thus HMMs can represent either content and context information, under the following assumptions:

- if each state represents a song in the collection, acoustic content-based similarity can be modeled by transition probabilities
- if the symbols emitted by the HMM are semantic labels, the context that describes each state can be modeled by observation probabilities.

A suitably built HMM (see Section 2.1) may be exploited to address the examples provided at the beginning of this section. On the one hand, the model can generate a path across songs while observing, for a defined number of time steps, the semantic label "classic rock". On the other hand, the model can start the path from the state associated to "Yesterday" and proceed to new states while observing the semantic labels associated to 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.

Since states of a HMM are not directly observable, the paths across the song collection need to be computed by a decoding step, which highlights the most probable state sequence according to a sequence of observations. A representation of the proposed model is depicted in Figure 1.

2.1 Definition of the HMM

An HMM λ that represents a collections of tagged songs can be formally defined by:

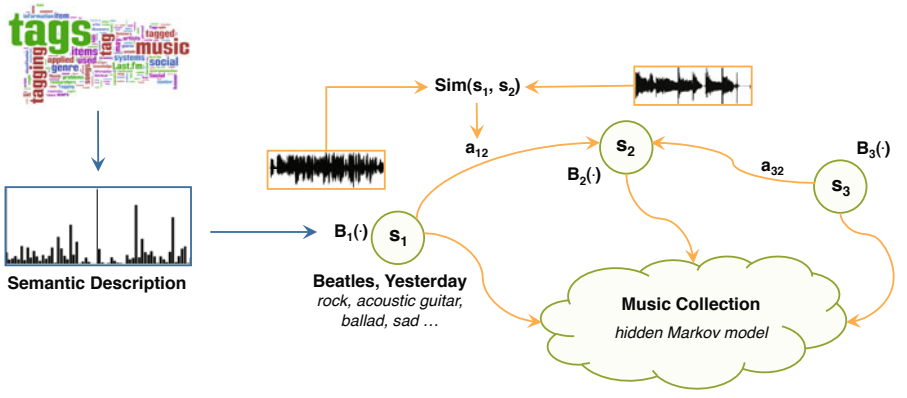

Fig. 1. General structure of the model: each song is represented by a state and is described by a set of tags. States are linked together by edges weighted according to acoustic similarity between the songs.

1. The number of songs N in the collection, each song represented by a state of the HMM. The set of states is denoted as $S = \{s_1, s_2, ..., s_N\}$.
2. The number M of distinct tags that can be used to describe a song. The set of symbols is denoted as $V = \{v_1, v_2, ..., v_M\}$.
3. The state transition probability distribution $A = a_{ij}$, which defines the probability to move from state i to state j in a single step. Transition probabilities a_{ij} depends to the similarity between songs s_i and s_j.
4. The observation probability distribution of each state j, $B = b_j(k)$, which defines the probability that tag v_k is associated to song j. Observation probability values represent the strength of the relationships *song-tag*, which is indicated as *affinity* value.
5. The initial state distribution $\pi = \{\pi_i\}$, that defines the probability to start a path across the model beginning at state s_i. Differently from the standard definition of HMMs, the initial state distribution is computed dynamically at retrieval time, since it is strictly connected to the type of information need, as described in Section 2.3.

Although acoustic similarity is always a positive value, implying $a_{ij} > 0 \quad \forall i, j$, with the aim of improving scalability, each state is directly connected to only the P most similar songs in the collection, while the transition probabilities with all the other states are set to 0. Heuristically, we set P to be the 10% of the global number of songs. At present, no deeper investigation has been carried out to highlight an optimal value of P. In order to obtain a stochastic model, both transition and emission probabilities are normalized, that is $\sum_j a_{ij} = 1$ and $\sum_k b_j(k) = 1$. Because of these two steps, transition probabilities are usually not symmetric, then $a_{ij} \neq a_{ji}$.

After setting all the parameters, the HMM can be used to generate random sequences, where observed symbols are tags. Dually, well known algorithms can

be used to decode the most probable state sequence according to a given observation sequence.

2.2 Computing the Relevance of Songs

The task at retrieval time is to highlight a sub-set of songs in the collection that are relevant to a particular query, either expressed by semantic labels or by a seed song. In the context of HMMs, the general problem can be stated as follows [9]: "given the model λ, and the observation sequence $\bar{O} = \{o(1), \ldots, o(T)\}$ with $o_j \in V$, the goal is to choose a state sequence $\bar{S} = \{s(1), \ldots, s(T)\}$ which is optimal in some sense". Clearly, the observations sequence represents the semantic description specified by the user need.

In literature, this problem is solved using the max-sum algorithm, which in HMMs applications is known as the Viterbi algorithm. The algorithm efficiently searches in the space of paths, in order to find the most probable one, with a computational cost that grows only linearly with the length of the chain. The algorithm is composed by a forward computation to find the maximization for the most probable path, and by a backward computation to decode the sequence of states. Although the general structure of the algorithm has been maintained, some key modifications in the recursion part of the forward computation have been introduced. Following the notation and the algorithm description provided in [9] the normal initialization and the modified recursion steps follow, for $1 \leq j \leq N$:

Initialization: for $t = 1$

$$\delta_t(j) = \pi_j \cdot obs_j(t) \ , \qquad \psi_t(j) = 0 \ . \tag{1}$$

Recursion: for $2 \leq t \leq T$

$$\delta_t(j) = \max_{1 \leq i \leq N} [\delta_{t-1}(i) \cdot a_{ij}] \cdot obs_j(t) \ , \tag{2}$$

$$\psi_t(j) = \arg \max_{1 \leq i \leq N} [\delta_{t-1}(i) \cdot a_{ij}] \ , \tag{3}$$

$$a_{kj} = \frac{a_{kj}}{d} \quad \text{with} \quad k = \psi_t(j) \ . \tag{4}$$

As it can be seen, we introduce $obs_j(t)$, defined in the next section, which is a general function that indicates how the semantic description is considered during the retrieval process. This function plays the role of observations in typical decoding applications.

Equation 4 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. Clearly, this is a problem because the

retrieved list would 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 be composed by only few songs. We addressed this problem by introducing a decreasing factor d, which is applied to the transitions probabilities when they are selected in the forward step. So, when a transition is chosen, the probability a_{ij} is decreased by factor d (we set $d = 10$), as shown in Equation 4, in order to make unlikely that the state sequence would pass again through the corresponding edge. It has to be noted that the attenuation is carried out *locally*, meaning that it affects the structure of the model only during the current retrieval operation.

Another issue that has to be addressed is a limitation in the structure of standard HMMs. Because of the first-order Markov chain assumption, HMMs are generally poor at capturing long-range correlations between the observed variables, that is between variables that are separated by many steps [15]. Earlier experiments showed that this limitation involved a decrease in precision when decoding long paths. In order to solve this problem, we considered the retrieval composed by many sub-retrieval operations, each one retrieving a sub-list of songs. Instead of performing a single backward decoding, the algorithm works for a subset of iterations, from which an optimal sub-path is built. Only the first n songs of this sub-path are considered in the final ranking list; at the end of each iteration the algorithm restarts from the last state of the n suggested. Given the locality of the approach, in this way we aim to keep constant the quality along the retrieved list, avoiding a decrease in precision.

2.3 Querying the Model

As often assumed in the interaction with music search engines, in our scenario a user can submit a query in two distinct ways: by providing a tag or by selecting a seed song in the collection. According to the kind of query, some of the model parameters are set differently.

In the *tag-based scenario*, the goal is to rank the songs according to their relevance with the provided tag and, at the same time, to their acoustic similarity. In this case, the observation sequence is composed simply by the chosen tag. We decided to set the initial state probability equal for all the states, in order to let the algorithm decide the beginning of the retrieved list. This scenario is very related to the standard HMMs case, then the function $obs_j(t)$ of Equations 1 and 2 is defined as

$$obs_j(t) = b_j(o_t) \tag{5}$$

for a generic state j, where observations o_t may be the same tag for all the time steps or it may change over time in case of playlist generation through more complex patterns.

In the *seed-song scenario*, when the query is submitted as a song q, the system is required to provide the user with a list of songs potentially similar to the query. In this case, the initial state distribution is forced to be 1 for the state

representing the seed song and 0 for all the others. The observation sequence to be decoded is modeled as the vector of observations characterizing the seed song. The function $obs_j(t)$ of Equations 1 and 2 is proportional to the inverse of the Kullback-Leibler (KL) divergence between the semantic description of the seed song and the chosen state [16]. 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 to the tags rather than to the pure distance between lists of tags. It is important to note that the KL divergence is required also because each song is described by a set of tags. Clearly, we consider the inverse because the goal is to maximize the probability when the divergence is small. Therefore,

$$obs_j(t) = \frac{1}{\mathrm{KL}(b_j(\cdot), b_q(\cdot))} \ , \quad \text{where} \quad \mathrm{KL}(b_j(\cdot) \parallel b_q(\cdot)) = \sum_{i=1}^{M} b_j(i) \cdot \log \frac{b_j(i)}{b_q(i)} \ , \tag{6}$$

for the generic state j and the initial seed state q; clearly, observations of q do not change over time t being linked to observations of the seed song. Since it is an observation probability, the actual value of $obs_j(t)$ undergoes a normalization process. It is worth noting that the use of KL divergence can be extended also to the tag-based scenario when the user provides a set of tags (instead of a single one) although this extension has not been tested yet.

3 Experimental Evaluation

A big challenge when designing a music retrieval system is how to evaluate a novel methodology. Although several efforts have been made within the MIREX campaigns, because of well-known copyright issues, data of past campaigns are not always available to test new approaches. Ideally, the list of retrieved songs should be evaluated by humans, in order to consider effectively the subjective nature of the concept of music similarity. Being human evaluation a time consuming task, we use an automatic approach considering that reliable annotations on songs can be exploited to measure the quality of a ranking list.

We tested our model through the Computer Audition Lab (CAL500) [17] dataset: 502 songs played by 502 unique artists, each one annotated by a minimum of 3 individuals using a vocabulary of 174 tags. A song is considered to be annotated with a tag if 80% of the human annotators agreed that the tag would be relevant. CAL500 is a reasonable ground truth because annotations are highly reliable, complete and redundant – i.e. multiple persons explicitly evaluated the relevance of every tag for each song. So far, it has been mainly used to evaluate automatic music annotation systems, but we believe that it could be a reasonable ground truth also to evaluate qualitatively a retrieval task. Although the size of the dataset does not allow to perform experiments in terms of scalability, we argue that, at this point, it is more significant to test the effectiveness of the approach, to show if the model can provide improvements in the retrieval process.

In the experiments reported in this section, we require that each tag is associated with at least 30 songs and remove some tags that seemed to be redundant or overly subjective. The semantic space is then composed by 62 tags describing information about: genre, instrument, acoustic qualities, vocal characteristic, emotion, and usage.

Retrieval is evaluated with metrics considering both performances at the top and along the whole ranking list. Since a music retrieval system should maximize the quality of the retrieved items in the first positions, we evaluate the precision at the first 3, 5 and 10 positions (P3, P5, P10). Beside, we include the *mean average precision* (MAP) measure, in order to have also an evaluation along the whole ranking list. All these metrics are extensively used in the literature to assess the effectiveness of a retrieval system [18].

3.1 Acoustic Content-Based Similarity

A number of methodologies have been proposed in literature to compute direct acoustic content-based similarity. In this set of experiments, we rely on the algorithm proposed by Mandel and Ellis [5], which uses a single Gaussian with full covariance to model a song. Although, some alternative approaches have been recently proposed [6], we use this one because of its efficiency and simplicity in the implementation. Songs are represented through vectors of Mel-Frequency Cepstral Coefficients together with their first and second derivatives (MFCC + delta) extracted from about one minute of music content, and the similarity between songs is computed using a symmetrized version of the KL divergence.

Section 2.1 describes how transition probabilities are computed from these similarity values, in particular by selecting for each state s_i the first P most similar songs and performing the normalization $\sum_j a_{ij} = 1$ with $s_j \in P$. It is important to note that we aim at proposing a general approach, which is independent on the way acoustic similarity is actually computed and which can be applied to other audio descriptors and other similarity measures. For this reason the computation of acoustic similarity is presented within the experimental evaluation section.

3.2 Semantic Space

There are several approaches to collect tags for music, each with its own advantages and disadvantages [19]. Among all, we chose two different representations.

A first semantic description has been computed from the music content. We used the Gaussian mixture model described by Turnbull et al. [7] to automatically annotate songs with tags based on audio content analysis. For a given song, the output of this algorithm is a vector of posterior probabilities named *semantic multinomial* that represents the strength of the relationship *tag-song* for each tag in the vocabulary. We refer to this description as "cb-auto-tags".

A second representation has been created by gathering the social tags from Last.FM, as reported on February 2010. For each song of the dataset, we collected

Table 1. The retrieval results using 62 distinct tags as queries

Semantic	Model	P3	P5	P10	MAP
	Random	0.165	0.171	0.166	0.141
cb-auto-tags	HMM	**0.516**	**0.488**	**0.452**	**0.361**
	TAG	0.419	0.431	0.405	0.332
Last.fm	HMM	**0.347**	**0.331**	**0.268**	**0.225**
	TAG	0.303	0.297	0.218	0.207

two lists of social tags using their public data sharing AudioScrobbler[3] website. We gathered both the list of tags related to a song, and the list of tags related to an artist. The relevance score between a song and a tag is given by the sum of the scores in both lists, plus the tag score for any synonym or other wild matches of the tag in both lists [2]. Social tag scores are then mapped to the equivalent class in our semantic description. If no gathered tag for a given song belonged to the semantic space, the semantic description is represented by a uniform distribution, where all the tags share the same score. This lead to a very sparse and noisy description, which is useful to test the effectiveness of our approach. We refer to these tags as "Last.fm".

We addressed these descriptions with two different evaluations, although they could be combined together in a single richer semantic description [2].

3.3 Tag-Based Retrieval

In this first experiment, the model is queried using a tag; a semantic concept is provided to the system, and the goal is to rank all the songs according to their relationships with that term. Metrics are then averaged through all the terms in the vocabulary. Retrieval performances are measured by finding the positions, along the ranking list, of the documents annotated with the considered tag in the ground truth. HMM-based retrieval is compared with the retrieval performed by simply ranking the songs according to their affinity value for that tag (TAG), as well as with a random baseline. Results are reported in Table 1, considering both types of semantic description.

As it can be seen, HMM-based retrieval clearly outperforms the retrieval based on a single tag, with a major improvement in the quality at the top of the ranking list. On the other hand, retrieval along the full list tends to decrease its effectiveness, as it can be inferred by the lower improvement achieved by MAP. This is probably due to the problem, discussed in Section 2.2, of HMMs generally

[3] `http://ws.audioscrobbler.com/2.0/`

Table 2. The retrieval results using 50 random seed songs as queries

Semantic	Model	P3	P5	P10	MAP
	Random	0.113	0.104	0.096	0.050
cb-auto-tags	TAG	0.266	0.270	0.246	0.211
	AB	0.237	0.234	0.236	0.187
	WLC	0.280	0.278	0.244	0.204
	HMM	**0.295**	**0.288**	**0.258**	**0.225**
Last.fm	Tag	0.273	0.272	0.262	0.191
	AB	0.237	0.234	0.236	0.187
	WLC	**0.305**	0.292	0.262	0.198
	HMM	0.304	**0.299**	**0.284**	**0.219**

poor at capturing long-range correlations between the observed variables. Still we believe that the most important aspect to consider in a retrieval system is the quality on the top of the ranking list. Results based on Last.fm tags tend to have lower performances in terms of absolute values. This likely depends on the fact that the semantic descriptions are rather sparse and noisy and that sometimes songs were represented through a uniform distribution.

3.4 Seed Song Retrieval

In this experiment, retrieval is carried out by submitting to the system 50 randomly selected seed songs and considering the sequence of states highlighted by the optimal path as a ranking list of retrieved documents. A ground truth, against which retrieval results are compared, has been created for each query song by selecting the 30 most similar songs according to their human-based annotations. Semantic similarity has been computed using an application of the KL divergence to the set of tags for each pair of songs.

We compare different approaches: the HMM-based retrieval, a direct content-based retrieval where songs have been ranked according to their acoustic similarity with the seed (AB), a semantic similarity measured as KL divergence between the semantic descriptions of the seed song and each document in the collection (TAG), and a linear combination between the two distances (WLC). Additionally we alos include random baseline (Random).

As it can be seen from the results reported in Table 2, even in this case the proposed model leads to outperforming results; the same consideration reported in Section 3.3 can be extend to the current evaluation. The only different aspect is that, in this case, the Last.fm tags better quantize the similarity relationships

among songs; thus, the absolute values of the metrics is not very different between the two semantic representations.

4 Conclusions

We introduce a novel methodology that represents a music collection through an hidden Markov model with the purpose to build a music retrieval system that combines content-based acoustic similarity and context-aware semantic descriptions. In the model, each state represents a song, transitions probabilities depend on acoustic similarity and observation probabilities represent semantic descriptions. An application of the Viterbi algorithm allows us to create paths across the model, which provides a ranking list of the songs. This approach represents an application of cross-domain retrieval combining audio content and text for item-based retrieval. It is important to note that the approach can be generalized also to other multimedia tasks where content can be combined with context, such as video or image retrieval. The model can be used as a part of a music digital library to refine the retrieval functions.

Some issues are still open and will be addressed in future work. First of all, evaluation tested only the effectiveness of the model; scalability needs to be evaluated with a larger collection, in terms of number of songs and tags. Moreover, future research will be also devoted to the analysis of the effects introduced by different content descriptors and similarity measures. Finally, the extension to other music retrieval tasks, such as music recommendation and playlist generation, will be explored.

References

1. Barrington, L., Oda, R., Lanckriet, G.: Smarter than genius? Human evaluation of music recommender systems. In: Proceedings of the International Conference on Music Information Retrieval, pp. 357–362 (2009)
2. Barrington, L., Lanckriet, G., Turnbull, D., Yazdani, M.: Combining audio content and social context for semantic music discovery. In: Proceedings of ACM SIGIR, pp. 387–394 (2009)
3. McFee, B., Lanckriet, G.: Heterogenous embedding for subjective artist similarity. In: Proceedings of the International Conference on Music Information Retrieval, pp. 513–518 (2009)
4. Slaney, M., Weinberger, K., White, W.: Learning a metric for music similarity. In: Proceedings of the International Conference on Music Information Retrieval, pp. 313–318 (2008)
5. Mandel, M., Ellis, D.P.W.: Song-level features and support vector machines for music classification. In: Proceedings of the International Conference on Music Information Retrieval, pp. 594–599 (2005)
6. Hoffman, M., Blei, D., Cook, P.: Content-based musical similarity computation using the hierarchical dirichlet process. In: Proceedings of the International Conference on Music Information Retrieval, pp. 349–354 (2008)

7. Turnbull, D., Barrington, L., Torres, D., Lanckriet, G.: Semantic annotation and retrieval of music and sound effects. IEEE Transactions on Audio, Speech, and Language Processing 16, 467–476 (2008)
8. Ness, S.R., Theocharis, A., Tzanetakis, G., Martins, L.G.: Improving automatic music tag annotation using stacked generalization of probabilistic svm outputs. In: Proceedings of ACM MULTIMEDIA, pp. 705–708 (2009)
9. Rabiner, L.: A tutorial on hidden Markov models and selected application. Proc. of the IEEE 77, 257–286 (1989)
10. Shifrin, J., Pardo, B., Meek, C., Birmingham, W.: HMM-based musical query retrieval. In: Proceedings of ACM/IEEE Joint Conference on Digital Libraries, pp. 295–300 (2002)
11. Miotto, R., Orio, N.: Automatic identification of music works through audio matching. In: Proceedings of the European Conference on Digital Libraries, pp. 124–135 (2007)
12. Montecchio, N., Orio, N.: A discrete filter bank approach to audio to score matching for polyphonic music. In: Proceedings of the International Conference on Music Information Retrieval, pp. 495–500 (2009)
13. Raphael, C.: Automatic segmentation of acoustic musical signals using hidden markov models. IEEE Transactions on Pattern Analysis and Machine Intelligence 21, 360–370 (1999)
14. Khadkevich, M., Omologo, M.: Use of hidden markov models and factored language models for automatic chord recognition. In: Proceedings of the International Conference on Music Information Retrieval, pp. 561–566 (2009)
15. Bishop, C.M.: Pattern Recognition and Machine Learning. Springer, Heidelberg (2006)
16. Kullback, S., Leibler, R.: On information and sufficiency. Annals of Mathematical Statistics 12, 79–86 (1951)
17. Turnbull, D., Barrington, L., Torres, D., Lanckriet, G.: Towards musical query-by-semantic description using the CAL500 data set. In: Proceedings of ACM SIGIR, pp. 439–446 (2007)
18. Manning, C., Raghavan, P., Schtze, H.: Introduction to Information Retrieval. Cambridge University Press (2008)
19. Turnbull, D., Barrington, L., Lanckriet, G.: Five approaches to collecting tags for music. In: Proceedings of the International Conference on Music Information Retrieval, pp. 225–230 (2008)