COMBINING TIMBRIC AND RHYTHMIC FEATURES FOR SEMANTIC MUSIC TAGGING

Nicola Orio
Department of Cultural Heritage
University of Padua, Italy
orio@dei.unipd.it

Roberto Piva
Department of Information Engineering
University of Padua, Italy
piva.roberto.88@gmail.com

ABSTRACT

In this paper we propose a novel approach to music tagging. The approach uses a statistical framework to model two acoustic features: timbre and rhythm. A collection of tagged music is thus represented as a graph where the states correspond to the songs and the models probabilities are related to the timbric and rhythmic similarity. Under the assumption that acoustically similar songs have similar tags, we infer the tags of a new song by adding it to the graph structure and observing the tags visited in acoustically meaningful random walks. The approach has been tested using the CAL500 dataset, with encouraging results in terms of precision.

1. INTRODUCTION

The ability of humans to associate tags or generic metadata with multimedia content is a difficult task to simulate, because it relies on subjective judgments and on the identification of connections between abstract concepts. In the case of music content, tagging has always been a feature of online streaming services like LastFM, Apple Genius, Pandora or Grooveshark, since these services rely on music descriptors to deliver the right songs to the right user. Their tags have different origins though: while Pandora pays music experts to annotate music with reliable and expressive terms, the other services rely on user generated tags and playlists and exploit statistics tools like collaborative filtering [8] for annotating and recommending music. A typical problem of manual tagging regards the annotation of new items. While songs by renowned artists may easily get proper tagging by their advertisers, there are thousands of tracks – for instance produced by small independent labels – that are likely to be unreachable because of the lack of good descriptors.

Another interesting problem is the long tail distribution: the analysis of listening charts highlights that the distribution of play counts over artists follows a power law. This means that a restricted number of artists gets the majority of play counts. However, the total play counts of the long tail largely outnumbers the one of the top artists. This situation is also common for the global charts of a music streaming service and has a reflection on how these services make money [6]. The problem is related to tagging because in most of the times the long tail consists of poorly tagged music pieces.

Early works on automatic music tagging addressed the recognition of the music genre [11,12,25]. The subjectiveness of "genre" classification led researchers to propose a novel genre taxonomy [18] and to identify music tagging as a more broad concept that includes other types of information – like track's pace and dance-ability. To this end, tagging has also been defined as "music semantic annotation" [10, 24]. Other approaches have explored the association between related tags [16], or the inclusion of mined social tags to infer some relation between tags and audio features [2].

A variety of features were considered in search of the right combination to correctly annotate novel songs. First attempts started with Mel Frequency Cepstral Coefficients (MFCC), which were successfully used for music classification while improvements were obtained with the aid of other sources such as social tags [5], or temporal features [15]. More recent work investigates the improvement of MFCC by using Principal Component Analysis [13] while learning frameworks have been applied as well, such as Support Vector Machines [3] and Artificial Neural Networks [9].

This paper describes a method for semantically tagging music clips by exploiting timbre and rhythm features represented in a single statistical framework, where audio features are related to different model parameters that have been developed on top of a previously defined retrieval model based on content and context descriptors [17].

2. THE TAGGING MODEL

Our basic assumption is that the relation between tags and music features can be better exploited by using multiple features in a single tagging model. Our framework, inspired by Hidden Markov Models (HMMs), accounts for music similarity in terms of two different audio features: timbre represented by MFCC and rhythm represented by Rhythm Histograms (RH).

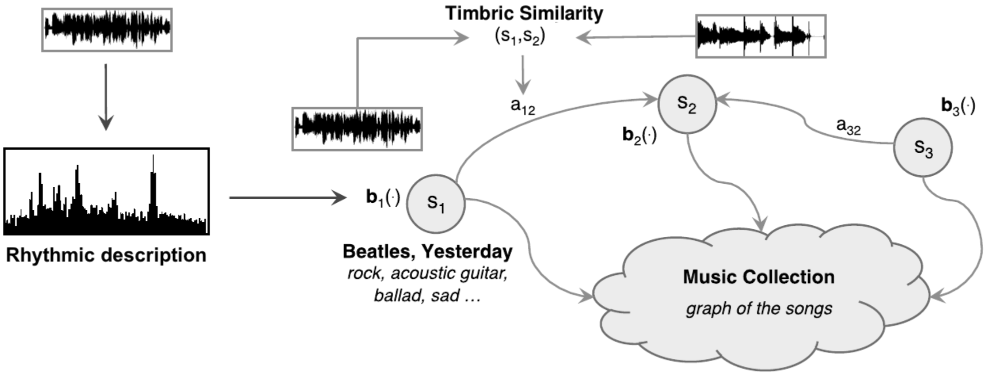

Figure 1. A graphical representation of the tagging model

2.1 Model Elements

Given the close relationship with HMMs, we introduce the different elements of the model using, when possibile, the same notation introduced in [21].

States: A song is represented as a state S_i from a set $\mathcal{S} = \{S_1, \ldots, S_N\}$, and state at time t is denoted as $s(t)$.

Prior Probabilities: The query song to be tagged is added to the model as state S_q, requiring the computation of the transition-wise similarity with each song in the collection to obtain a_{qi} for all i. Each path in the model is forced to start from the query song.

Transition Probabilities: They are related to the acoustic similarity between the songs using one of the two features, thus they can be related either to timbre or to rhythm similarity. Referring to the usual naming convention for HMMs, a_{ij} stands for the acoustic similarity of song S_i with song S_j.

Observation Probabilities: They are related to the acoustic similarity between the songs and the query song, computed from a different music dimension than the on used for transitions. We introduce symbol $\phi_i(q)$ as the probability for song S_i of emitting the audio features of query song S_q.

Tags: The information regarding the tags of the known songs is stored in a separate variable, named $b_i(w)$, that associate tag w to song S_i in the collection. We assume that the vocabulary of tags consists in $|\mathcal{V}|$ different symbols belonging to $\mathcal{V} = \{w_1, w_2, \ldots, w_{|\mathcal{V}|}\}$. The vector b_i is in binary form, it contains a 1 in position j if tag j is present, 0 otherwise.

The model is then exploited to generate random walks across the collection, starting from the song to be tagged. Transition probabilities guarantee acoustic consistency of the songs in the paths, while observation probabilities guarantee acoustic similarity with the query song. A graphical representation of the model can be found in Figure 1.

Although in principle $a_{ij} > 0$ for each (i, j) pair, we artificially keep, for each song S_i, only a subset $\mathcal{R}(S_i)$ of non-zero transitions. That is, we keep the top P highest transition probabilities for each state, and we set to 0 the remaining transitions, with an increase in model scalability. The value of P has been set to 10% of the global number of the songs. Transitions are normalized to maintain the stochastic properties.

As a consequence of the analogy with HMMs, the observation's feature plays a more important role than the transition's feature in discriminating the songs. This happens because observation's feature is compared to the query at each step in a random walk across the model, while the influence of transition similarity decreases as the path gets longer. Having the tags in a separate structure let us grow the model easily if there is a novel tag.

2.2 Acoustic Features

We use MFCC to capture the timbre signature of each song. MFCC are computed using 23 ms, half-overlapping windows over 6 seconds of music after the first 30 seconds. We thought 6 seconds may be a good amount of data to be used for automatic tagging for online services and for fast tagging of large collections. The output is a 13-sized vector for each window, but since we include the first and second derivatives, we end up with a matrix of 39 elements multiplied by the number of windows.

We use Rhythm Histograms (RH) and Rhythm Patterns (RP) to hold information regarding the rhythm of a given piece of music, as described in [19, 22] as *psychoacoustically motivated Rhythm Patterns*. Rhythm Patterns themselves are a variation of Fluctuation Pattern [7]. RH are computed using the same 6 seconds of audio as MFCC. Simplifying some psychoacoustic adjustments steps in the definition of RP and slightly varying the calculation parameters, we ended up with a matrix descriptor of 120 modulation frequencies by 24 frequency bins.

2.3 Similarity Measures

We adopted two approaches to compute the similarity between two songs. The first one considers each frame as a fixed-width word in a dictionary [14]. We calculate mean and covariance of the (letters of the) words over the length of the song and the result is a representative multivariate Gaussian distribution of the song. To compare these two distribution we use the Kullback-Leibler divergence (KL divergence):

$$KL(p\,||\,q) \equiv \int p(x) \log \frac{p(x)}{q(x)}\,\mathrm{d}x, \tag{1}$$

where $p(x)$ and $q(x)$ are the two distributions. The KL divergence is then transformed to a $[0, 1]$ value (with 1 representing two identical songs) by exponentiating the divergence:

$$sim(p, q) = e^{-\gamma KL(p\,||\,q)}, \tag{2}$$

where γ is a parameter to be tuned (for this work it has been set to 0.009, see [20]). In this paper this similarity is referred to as "single gaussian".

We investigated also a number of alternative similarity measures [4]: the Cosine similarity and its varied version Modified Chord, a similarity based on Euclidean distance, two biology-based similarities (Bray-Curtis and Ruzicka), the Similarity Ratio, the Kulczynski similarity and the exponentiated version of the discrete KL-divergence (very similar to the one from Equation 2). This second group of measures is based on a vectorized version of the matrix representations. For MFCC we keep the absolute value of the time-wise sum, obtaining a 39-sized vector representing the average timbre across the piece of music. For the RH we sum along the frequency bins (barks) to obtain a 118 sized vector representing the influence of each modulation frequency in the $[0, 10]$Hz range.

2.4 Querying the model

We developed a modified version of the Viterbi algorithm, which application to HMMs can be found in [21] where δ represents the probability of the optimal path and ψ is used to keep track of its actual state sequence. We refer to the query song using the subscript q (e.g. S_q for the song, $b_q()$ for the tags, and so on).

Initialization: As introduced in Section 2.1 the song to be tagged S_q is inserted in the model and the initialization step is, for all $i = 1, \ldots, N$:

$$\delta_1(i) = \begin{cases} 1 & i = q \\ 0 & i \neq q \end{cases} \tag{3}$$

$$\psi_1(i) = 0. \tag{4}$$

Recursion: Here we present a variation from the original Viterbi. For $t = 2, \ldots, T$, and $i = 1, \ldots, N$:

$$\delta_t(i) = \max_{1 \leq i \leq N} [\delta_{t-1}(i)\, a_{ji}]\, \phi_q(i) \tag{5}$$

$$\psi_t(i) = \arg\max_{1 \leq i \leq N} [\delta_{t-1}(i)\, a_{ji}] \tag{6}$$

$$a_{ri} = \frac{a_{ri}}{\eta} \ \text{ for } \ r = \psi_t(i), \ \eta = 10. \tag{7}$$

Where $\phi_q(i)$ is the similarity between S_q and S_i and can be computed using one of the possible similarity measures introduced in Section 2.3. Equation 7 aims at preventing looping by lowering the probability of performing a transition twice [20].

Decoding: The most probable path is computed as in the original Viterbi algorithm:

$$s(t)^* = \begin{cases} \arg\max_{1 \leq i \leq N} [\,\delta_t(i)\,] & \text{if } t = T \\ \psi_{t+1}(s(t+1)^*) & \text{if } t = T-1, \ldots, 1, \end{cases} \tag{8}$$

$$p(t)^* = \begin{cases} \max_{1 \leq i \leq N} [\,\delta_t(i)\,] & \text{if } t = T \\ \delta_t(s(t)^*) & \text{if } t = T-1, \ldots, 1. \end{cases} \tag{9}$$

Where $p(t)^*$ represents the probability of the optimal path at each step. The parameter T is the maximum length of the subpath, which is discussed in the following section.

Tagging: The tags of states belonging to the optimal path are used to tag the query song, because they are likely to be acoustically similar to the query and to share some of their textual descriptors. We take the T songs extracted by Equation 8 and we calculate a vector of tag weights $b_q(w)$ for all tags in $j = 1, \ldots, M$ as in:

$$b_q(w) = \sum_{t=1}^{T} b_t(w)\omega(t). \tag{10}$$

Where $\omega(t)$ is a decreasing monotonic function of the path position i. That is, tags from songs acoustically similar – reached early in the optimal path – to the query should have more importance than songs that are not that similar – observed at the latter steps of a random walk.

Iteration: To keep a high transition-wise similarity, the procedure is split in substeps. After having computed a path of length T, with $T = 4$ giving the best experimental results, we restart Viterbi decoding from the initialization step without resetting the modified transitions (i.e. keeping the effects of Equation 7). The procedure is iterated a number of times until enough songs are visited in order to correctly infer the tags for the query song. The final weight of a tag is computed as the sum of the weights computed at each iteration.

It has to be noted that there is no real control of the paths extracted at each iteration, in a sense that there can be songs that appear multiple times on different iterations or maybe multiple times on the same iteration. This as a desired behavior: if a song is chosen multiple times we assume that it is particularly relevant to the query, and so its tags may be better related to the it. In fact, we sum the tags contribution of these songs every time they are chosen, regardless of the number of times and the position in the path where they appear.

The proposed approach allows us to exploit the influence of two features at once. It can be noted that we could calculate query's tag by simply using all the neighbor songs computed from a forward exploration of the model starting from the query. The advantage of using Viterbi decoding relies on the fact that it finds an acoustically meaningful "music path" from the query, which helps us avoiding non-related songs. Another important advantage is that we can also decide which weight we want to assign to each song in the path, which could lead to better results.

2.5 Weighting the Tags

As of the weighting function in Equation 10 we explored different options, that have been tested experimentally. For each time step $t = 1, \ldots, T$ the function $\omega(t)$ can be computed according to:

Path probability: As a first option, the path probability at step t can be used directly a the tag weight, using $\omega(t) = p(t)^*$.

Linear decay: The relevance of tags can decrease linearly with the length of the path required to obtain them, according to $\omega(t) = 1 - m(t-1)$, with $0 < m < 1$.

Exponential decay: Since the probability of a path across HMMs decreases exponentially with the length of the path, tag weight can be computed also according to $\omega(t) = a^{(t-1)}$, with $0 < a < 1$.

Hyperbolic decay: In order to obtain, for small values of T, intermediate weights between linear and exponential, tag weight can be also computed as $\omega(t) = 1/t$.

3. RESULTS

An automatic tagging system is expected to put meaningful tags on novel songs in a reliable way. We have already seen the importance of automatic tagging in the introduction and we wanted to test how our model performs in this difficult task.

3.1 Data Source

For the experimental evaluation we focus on the quality of the source data, as we need to rely on it to put the right tags to songs. For these reasons we have chosen to use the CAL500 dataset [23], which consists of 502 popular songs of Western music by different artists. Songs from this dataset have been tagged, through a controlled survey, by at least three human annotators each. The semantic vocabulary consists of 149 tags spacing from genre classification (e.g. "rock", "pop") to vocal and acoustic characteristics (e.g. "female lead vocals"), as well as emotions (e.g. "aggressive") and song usages (e.g. "studying"). The survey results is a binary annotation of each song.

Acoustic features (i.e. MFCC and RH) have been computed from a degraded version of the clips in the dataset, with an encoding quality which was still high enough for our purposes. The availability of the audio motivates the choice of CAL500. Moreover, we are more interested in the reliability of the tagging procedure, so we prefer to evaluate our approach with a small yet controlled collection. Experimental evaluation with larger collections, such as CAL10k or Magnatagatune, will be part of our future work.

3.2 Evaluation Measures

What the users expect from an automatic tagging system is that proposed tags are relevant, in a sense that they truly describe the content of the song, and the tags are complete, so there is no lack of information. Any extra tag is counted as an error, or at least as noise. We also expect that the system has to be concise, that is, if the output of the system is a ranked list of tags, where the rank is a relevance measure, we may want to keep only the top most tags as the query result. In other words, we need to measure whether the system is able to propose the most relevant tags at the top of the ranked list, while other tags (not-so-relevant ones and wrong ones) should have lower ranks. With this aim in mind we choose to use the precision metric: we measured the precision at 10 (P@10), which reports the fraction of relevant tags of the top 10 results from the ranked list, and the mean average precision (MAP), which averages the precision at each point of the ranked list of tags.

3.3 Testing Procedure

Since the role of this model is to tag a new song we can test it using the ground truth (the CAL500 dataset) in a leave-one-out fashion on all the songs minus one. What we have done is calculating the acoustic similarity for each song in the dataset, and, in turns, we simulated the querying of each song against the rest of the dataset.

To this end, the tagging procedure ignores the tag contribution from the query song as it assumes that it does not have tags on it.

3.4 Parameters Tuning

The tagging model as proposed in this thesis has some parameters which have to be tuned.

We tested some combinations of the values of T (length of the optimal path) in relation to the number of effectively retrieved songs per path and the total number of iterations (see section 2.4). What we have seen is that short paths give better results, and the number of effectively retrieved songs should be as close as possible to the length of the path: this could mean that the conservation of acoustic similarity is preferred over the number of retrieved results. We ended up choosing a path length of 4 with 3 retrieved song per iteration. This approach can produce iterations where as little as 1 song are retrieved, that is, there can be a loop of length 3 with the query and one song. Other combinations of path length and retrieved songs led us to worse results MAP-wise.

Regarding the total number of iterations of the Viterbi algorithm we have seen that, for our data, the best results were obtained with 9 iteration. Of course the influence of these parameters should be further discussed and optimal values may change for different datasets or as the model grows integrating the tagged songs.

3.5 Experimental Results

The approach was tested using MFFC for transitions and RH for observations and viceversa. Moreover, we wanted to evaluate the effect of individual features and compare them with a baseline approach. To measure this we tried three strategies: first, we measured the influence of the observation similarity by imposing uniform transitions. Then we measured the influence of the transition similarity by uniforming the observation probabilities. The last test simulated a completely random walk by uniforming both probabilities, in order to obtain a true baseline.

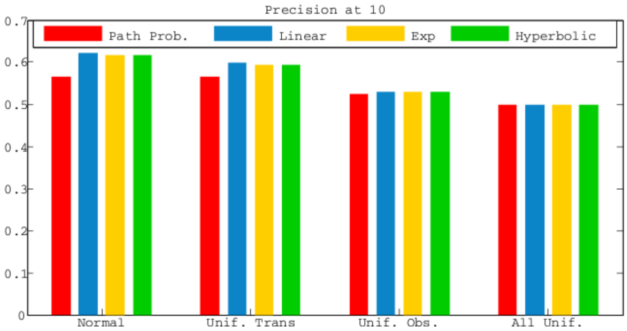

Figure 2. P@10 from MFCC on transitions with KL (single gaussian) similarity, RH on observations with Modified Chord similarity, comparing the effects of different weighting and the single features. Note: The span of this and the following figures is set in the range $[0; 0.7]$ in order to better appreciate the differences between the values

In Figure 2 are shown the results of P@10 comparing different combinations of uniform probabilities and tag weighting schemes. We can see that the exploitation of both features gives consistently better results than uniform probabilities. In turn, using only observations gives better results than using only transitions while the baseline gives always poorer results. It can also be observed that linear weighting performs slightly better than all weighting strategies and that the simple use of path probability – although a natural choice – gives the lowest results.

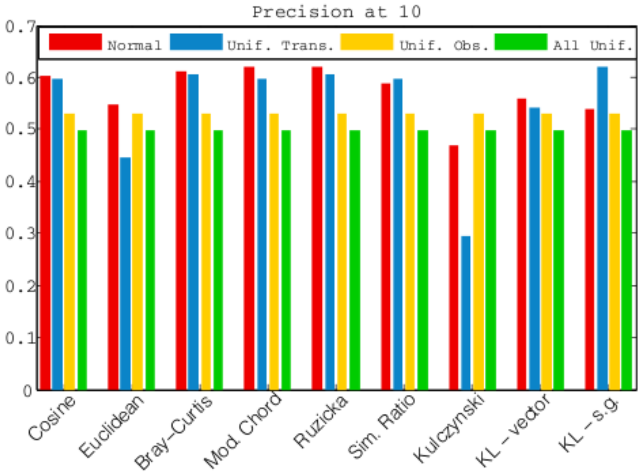

Figure 3. P@10 from MFCC on transitions with KL (single gaussian) similarity, RH on observations with multiple similarities, with linear tag weighting

In Figure 3 and Figure 4 are shown the results of P@10 comparing different similarity measures, using linear tag weighting. Our best results lead to over 0.6 precision, which means that if we pick a random song and we keep only the top 10 proposed tags, we can expect 6 of them to be correct on average, which is a good result for the first implementation of a novel approach. As it can be seen, not all similarities combinations have the same ratio between "normal" approach and uniform probabilities approach. For some combinations "normal" approach has worse results than "uniform transition probabilities" which underlines the importance of the choice of the distance measure.

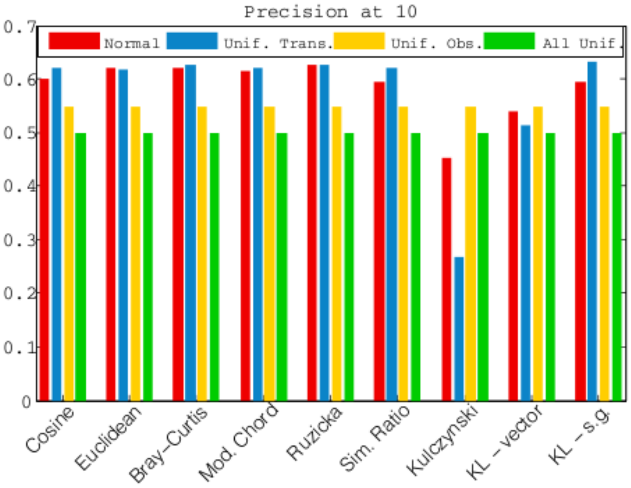

Figure 4. P@10 from RH on transitions with KL (single gaussian) similarity, MFCC on observations with multiple similarities, with linear tag weighting

Our MAP performance is in line with the one of other works at initial stage described in the literature. Table 1 shows a MAP results summary for the same configuration as Figure 3 and Figure 4 respectively.

<table>
<tr><td colspan="5">MFCC on transitions (KL - single gaussian),
RH on observations (Modified Chord)</td></tr>
<tr><th></th><th>Normal</th><th>Un. Tr.</th><th>Un. Obs.</th><th>All Un.</th></tr>
<tr><td><b>Path P.</b></td><td>0.481</td><td>0.474</td><td>0.442</td><td>0.442</td></tr>
<tr><td><b>Linear</b></td><td><b>0.536</b></td><td>0.507</td><td>0.450</td><td>0.442</td></tr>
<tr><td><b>Exp.</b></td><td>0.533</td><td>0.504</td><td>0.451</td><td>0.453</td></tr>
<tr><td><b>Hyperb.</b></td><td>0.533</td><td>0.504</td><td>0.452</td><td>0.450</td></tr>
<tr><td colspan="5">RH on transitions (KL - single gaussian),
MFCC on observations (Euclidean)</td></tr>
<tr><th></th><th>Normal</th><th>Un. Tr.</th><th>Un. Obs.</th><th>All Un.</th></tr>
<tr><td><b>Path P.</b></td><td>0.478</td><td>0.489</td><td>0.447</td><td>0.446</td></tr>
<tr><td><b>Linear</b></td><td><b>0.528</b></td><td>0.523</td><td>0.465</td><td>0.442</td></tr>
<tr><td><b>Exp.</b></td><td>0.526</td><td>0.520</td><td>0.468</td><td>0.453</td></tr>
<tr><td><b>Hyperb.</b></td><td>0.526</td><td>0.520</td><td>0.467</td><td>0.450</td></tr>
</table>

Table 1. Mean average precision summary table

4. CONCLUSIONS

We describe a novel approach for semantic music tagging based on a statistical framework that combines two music features: timbre and rhythm. Tagging is based on a modified Viterbi algorithm to carry out iterated random walks in the graph that represents a collection of tagged songs. The approach was evaluated using the CAL500 dataset. Experiments have shown encouraging results in terms of precision at 10 and Mean Average Precision of the ranked lists of tags. Performance contribution of each feature has been also measured separately and compared to a random baseline. The effects of different similarity measures for observations have been tested as well, together with four approaches to weight tags according to the number of steps required to obtain them. To the best of our knowledge, we think that our performances are in line with other early stage approaches. As pointed out in [1], purely audio-based approaches are instrinsically limited because they cannot capture all the music dimensions perceived by listeners. We think that additional parameter tuning, possibly using other collections, will give further improvements before the "glass ceiling" is reached.

One issue that will be addressed in future work is the effect of loops in the optimal path, which at the moment is minimized with the modified Viterbi algorithm (Equation 7) but can be improved by alternative strategies to modify the transition probabilities. The current tagging procedure suggests a way to grow the collection by adding the newly tagged song to the graph, thus we aim also at measuring how performances degrade with the increase of the collection size.

5. REFERENCES

[1] J.-J. Aucouturier and Pachet F. Improving timbre similarity: How high is the sky? *Journal of Negative Results in Speech and Audio Sciences*, 1(1), 2004.

[2] T. Bertin-Mahieux, D. Eck, F. Maillet, and P. Lamere. Autotagger: A model for predicting social tags from acoustic features on large music databases. *Journal of New Music Research*, 37(2):115–135, 2008.

[3] S. Bourguigne and P.D. Agüero. Audio tag classification using feature trimming and grid search for SVM. In *Proc. of MIREX*, 2011.

[4] S.-H. Cha. Comprehensive survey on distance/similarity measures between probability density functions. *City*, 1(2):1, 2007.

[5] D. Eck, P. Lamere, T. Bertin-Mahieux, and S. Green. Automatic generation of social tags for music recommendation. *Advances in neural information processing systems*, 20(20):1–8, 2007.

[6] A. Elberse. Should you invest in the long tail? *Harvard business review*, 86(7/8):88, 2008.

[7] H. Fastl. Fluctuation strength and temporal masking patterns of amplitude-modulated broadband noise. *Hearing Research*, 8(1):59–69, 1982.

[8] D. Goldberg, D. Nichols, B.M. Oki, and D. Terry. Using collaborative filtering to weave an information tapestry. *Communications of the ACM*, 35(12):61–70, 1992.

[9] P. Hamel. Multi-timescale pmscs for music audio classification. In *Proc. of MIREX*, 2012.

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

[11] T. Li, M. Ogihara, and Q. Li. A comparative study on content-based music genre classification. In *Proc. 26th ACM SIGIR conference*, pages 282–289. ACM, 2003.

[12] T. Lidy and A. Rauber. Evaluation of feature extractors and psycho-acoustic transformations for music genre classification. In *Proc. of ISMIR*, pages 34–41, 2005.

[13] S.-C. Lim, K. Byun, J.-S. Lee, S.-J. Jang, and Moo Young Kim. Music genre/mood classification. In *Proc. of MIREX*, 2012.

[14] M.I. Mandel and D.P.W. Ellis. Song-level features and support vector machines for music classification. In *Proc. of ISMIR*, pages 594–599, 2005.

[15] M.I. Mandel and D.P.W. Ellis. Multiple-instance learning for music information retrieval. In *Proc. of ISMIR*, pages 577–582, 2008.

[16] R. Miotto and G. Lanckriet. A generative context model for semantic music annotation and retrieval. *Audio, Speech, and Language Processing, IEEE Transactions on*, 20(4):1096–1108, 2012.

[17] R. Miotto and N. Orio. A probabilistic model to combine tags and acoustic similarity for music retrieval. *ACM Transactions on Information Systems (TOIS)*, 30(2):8, 2012.

[18] F. Pachet, D. Cazaly, et al. A taxonomy of musical genres. In *Proc. Content-Based Multimedia Information Access (RIAO)*, pages 1238–1245, 2000.

[19] E. Pampalk. Islands of music: Analysis, organization, and visualization of music archives. *Master's thesis, Vienna University of Technology*, 2001.

[20] R. Piva. Combining timbric and rhythmic features for semantic music tagging. *Master's thesis, University of Padua, Padova, Italy*, 2013.

[21] L.R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

[22] A. Rauber, E. Pampalk, and D. Merkl. The som-enhanced jukebox: Organization and visualization of music collections based on perceptual models. *Journal of New Music Research*, 32(2):193–210, 2003.

[23] D. Turnbull, L. Barrington, D. Torres, and G. Lanckriet. Towards musical query-by-semantic-description using the CAL500 data set. In *Proc. of ACM-SIGIR*, pages 439–446. ACM, 2007.

[24] D. Turnbull, L. Barrington, D. Torres, and G. Lanckriet. Semantic annotation and retrieval of music and sound effects. *Audio, Speech, and Language Processing, IEEE Transactions on*, 16(2):467–476, 2008.

[25] G. Tzanetakis and P. Cook. Musical genre classification of audio signals. *Speech and Audio Processing, IEEE transactions on*, 10(5):293–302, 2002.