A MUSIC RECOMMENDATION SYSTEM BASED ON SEMANTIC AUDIO SEGMENTS SIMILARITY

Alessandro Bozzon, Giorgio Prandi, Giuseppe Valenzise and Marco Tagliasacchi
Politecnico di Milano - Piazza Leonardo Da Vinci 32, 20133 Milano, Italy
{bozzon,prandi,valenzise,tagliasa}@elet.polimi.it

ABSTRACT
In this paper we propose a novel approach for content-based music recommendation. The main innovation of the proposed technique consists of a similarity function that, instead of considering entire songs or their thumbnail representations, analyzes audio similarities between semantic segments from different audio tracks. The rationale of our idea is that a song similarity and recommendation technique, to be more meaningful to the user from a semantic point of view, may evaluate and exploit similarities on semantic units between audio tracks. Our similarity algorithm consists of two main stages: the first step performs segmentation of the song in semantic parts. The latter assigns a similarity and recommendation score to a pair of songs, by computing the distance between the representations of their segments. To assign the global similarity and recommendation score, we consider a consistent subset of all the inter-segment distances. By adopting a graph-bases framework, we propose a graph-reduction algorithm on weighted edges that connect segments of different songs to optimize the similarity score with respect to our recommendation goal. Experiments conducted on a database of 200 audio tracks of various authors and genres show promising results.

KEY WORDS
Content-based Multimedia Retrieval, Music Recommendation, Genre Classification

1 Introduction

The world of digital music is dramatically growing and the availability of huge song collections opens novel scenarios for their use. For instance, Web music stores require more and more advanced services, such as recommendation systems, where the Web store advises customers on related products that might be of particular interest. Usually, recommendation systems are based on some relevant product features that are compared to the same set of features of other products to find similarities. A music recommendation system is based on the same principle: it proposes to the users a song (or a set of songs) being similar to the audio tracks they like. The simplest yet most reliable technique for determining music similarity is performed by tagging each audio track manually. In practice, such approach lacks in feasibility: due to the huge amount of available songs, labeling audio tracks and their similarities is an expensive and time-consuming task. A more complete and democratic technique makes use of collaborative-filtering algorithms: they compute the similarity between two songs by evaluating also the related similarity score given by other people. Being based on human opinion, these approaches capture many intangible factors that are unlikely to be obtained from the original audio. The disadvantage of these techniques, however, is that they are only applicable to music for which a reasonable amount of reliable Web data is available. For new or undiscovered artists, an audio content-based technique can be more suitable. To combine the benefits of both approaches, there exists a third type of systems, named hybrid recommendation systems, which provide an integrated model that incorporates both content-based and collaborative characteristics.

In this paper, we propose a novel approach for content-based music recommendation. The main innovation of the proposed technique consists of a similarity function that, instead of considering entire songs or their thumbnail representations, analyzes audio similarities between semantic segments from different audio tracks. The idea is that, by slicing the audio track in significant sections, two songs can be compared on a finer level of granularity, enabling also the implementation of higher level, semantic targeted similarity functions. The proposed approach can be implemented as a part of a hybrid recommendation system: in these cases, it is clear that a well-functioning audio-based recommender, which takes into consideration the semantic structure of the audio track, allows to reach more consistent results also when other meta-data information are lacking or missing.

The rest of the paper is organized as follows. The next section provides an overview of the existing works concerning semantic segmentation and music recommendation systems. Section 3 describes the theoretical and technical foundations of our work. Section 4 reports the results of our experiments and, finally, Section 5 provides some concluding remarks.

2 Related Work

In the literature, there are three possible approaches for recommendation systems [1]: content-based [2], collaborative [3] and hybrid recommendation [4]. Our work focuses only on content-based systems. Several methods for content-based music recommendation from large song databases have been proposed. Cano et al. [5] present MusicSurfer, a metadata-free system for the interaction with massive collections of music. They extract descriptions related to instrumentation, rhythm and harmony from music signals using similarity metrics. The authors of [6] analyze the problem of music recommendation by considering groups of related songs: from each song, a group of Mel-frequency cepstral coefficients (MFCCs) is extracted. Afterwards, they model such features using K-means clustering, learning the mean, the covariance and the weight of each cluster. Similarity is computed by comparing the models using the Earth-Mover's distance (EMD), which calculates the cost of "moving" probability mass between clusters to make them equivalent. In [7], music is classified using features extracted from the raw music file such as pitch, tempo and loudness. These features are then used to build up a recommendation system. A different approach is proposed in [8], where authors try to learn the user's preferences by mining the melody patterns from the music access behavior. Music recommendation is achieved through a melody preference classifier which is trained on the whole audio track. However, the performance of this system cannot be easily compared with similar works, since the results are based on midi files and not on real music.

To the best of our knowledge, there is no proposal in literature for recommendation systems fully exploiting the potential of semantic segmentation of audio tracks. Semantic music segmentation refers to the process of detecting significant musical structure in an audio stream, such as *intro*, *verse*, *chorus*, *ending*, etc. Many approaches have been proposed in the literature (e.g. [9], [10]), with different application domains. Previous works explore the use of segmentation for music content-based similarity, but only to provide a short representative thumbnail audio clip for a given track. The authors of [11] present a system leveraging on a thumbnailing process to enable easier and faster user evaluation of search results. Conversely, our approach makes use of a semantic segmentation process to enhance the similarity comparison between different tracks, by evaluating the similarity between pairs of audio segments.

3 Technical Realization

The rationale of the proposed approach is that a song similarity and recommendation technique, to be more meaningful to the user under a semantic point of view, has to evaluate and exploit similarities on semantic audio parts, using a finer granularity analysis; in our system this is achieved by means of semantic partitions of the original audio tracks. Figure 1 shows the detailed steps on which the proposed technique is based. When a new song is added to the collection, it is first converted into a raw audio format (e.g. WAV) and pre-processed in order normalize the amplitude of the zero-mean audio signal.

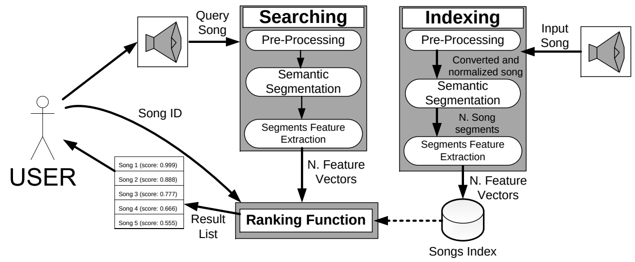


Figure 1: Top level view of the proposed content-based recommendation system.

The semantic segmentation module receives in input the pre-processed song and builds a series of consistent semantic segments. The *Segment Feature Extraction* module is responsible for creating a suitable compact representation for each segment in a song by computing a set of low level features from the raw audio. Such representation consists of a global feature vector for each segment; then, every computed vector is stored into the song database, to compose the song collection.

When, on the other hand, a user performs a query, he provides to the system a query song, in the form of an audio file or as a unique identifier for a track already contained in the database. In the former case, the query song is pre-processed, segmented and its segments' representations are sent to the ranking function; the ranking function produces an ordered list of similar songs by assigning a similarity score to each track in the database. In the latter case, the feature vectors associated with the given song identifier are extracted from the database and given in input to the ranking function.

3.1 Song Segmentation

Our recommendation system requires the semantic segmentation of a song. We implemented the algorithm described in [9]. In this approach, the segment boundaries detection task is divided into a two-phase process: Phase 1 focuses on detecting the possible segment boundaries, which denote structural changes in the audio signal; Phase 2, instead, focuses on refining the detected boundaries obtained from Phase 1 by aggregating contiguous segments while keeping those which mark true structural changes in the music audio. The algorithm returns as output the boundaries of the found semantic segments. In our experiments, the segment duration varies from 4 to approximately 50 seconds. The reader may refer to [9] for a detailed description of the segmentation process.

3.2 Feature Extraction and Segment Modeling

The second step of our recommendation system requires the extraction of significant audio features from each segment. Unlike simple monophonic sounds such as musical instrument tones or sound effects, music is a complex, non stationary polyphonic signal, and therefore it has to be characterized by its overall "texture" features [12].

In our approach, a segment $s_{k,i} \in S_k$ (where S_k is the segment set for a given song k, and i is the intra-song segment index $1 \leq i \leq |S_k|$), is broken into small, overlapping temporal *analysis windows*. Each window is processed separately and it needs to be short enough for the frequency characteristics of the magnitude spectrum to be relatively stable. In our approach we used analysis windows of 100 ms with 40 ms of overlap. For each analysis window, we extract a feature vector of 25 low-level audio features, summarized in Table 1. We made use of *temporal*, *spectral* and *perceptual* features, which, for their nature, are tightly related to the timbral texture of an audio signal.

Feature Type	Features	Ref.
Temporal	ZCR	[13]
Spectral	Centroid, rolloff, flux, flatness	[14]
Perceptual	first 20 MFCC	[15]

Table 1: Audio features used for semantic segment representation.

Each segment $s_{k,i}$ is therefore represented in a 25-dimensional vector space by a cloud of feature vectors. According to [16], a single Gaussian Distribution Model suffices for representation of such cloud; hence, we can represent each vector by its sample mean vector $\mathbf{m}_{k,i}$ and covariance matrix $\mathbf{C}_{k,i}$. Finally, we refer to $\hat{s}_{k,i} = \langle \mathbf{m}_{k,i}, \mathbf{C}_{k,i} \rangle$ as the Gaussian representation for a segment $s_{k,i}$.

3.3 Inter-Segment Similarity Evaluation

To evaluate the similarity for a generic pair of segments belonging respectively to songs k and x, me make use of Kullback-Leibler divergence $D_{KL}(\hat{s}_{k,i}, \hat{s}_{x,j})$, reported in Equation (1) [17] and computed on the Gaussian representations $\hat{s}_{k,i}, \hat{s}_{x,j}$ for the considered segments $s_{k,i}, s_{x,j}$.

$$D_{KL}(\hat{s}_{k,i}, \hat{s}_{x,j}) = \frac{1}{2}(\log\left(\frac{|\mathbf{C}_{x,j}|}{|\mathbf{C}_{k,i}|}\right) + \mathrm{tr}\left(\mathbf{C}_{x,j}^{-1}\mathbf{C}_{k,i}\right) + \\ + (\mathbf{m}_{x,j} - \mathbf{m}_{k,i})^T \mathbf{C}_{x,j}^{-1}(\mathbf{m}_{x,j} - \mathbf{m}_{k,i}) - M) \quad (1)$$

Where M is the dimensionality of the feature space (in our case, $M = 25$). Being the KL-divergence not symmetrical, we used the symmetric log-version defined in Equation (2):

$$D^*_{SKL}(\hat{s}_{k,i}, \hat{s}_{x,j}) = \\ \log\left(1 + \frac{D_{KL}(\hat{s}_{k,i}, \hat{s}_{x,j}) + D_{KL}(\hat{s}_{x,j}, \hat{s}_{k,i})}{2}\right) \quad (2)$$

Using a slight abuse of terminology, from now on we refer to $D^*_{SKL}(\hat{s}_{k,i}, \hat{s}_{x,j})$ as the similarity score for segments $s_{k,i}, s_{x,j}$. Note that low values of D^*_{SKL} indicate high similarity: the maximum similarity of the Gaussian representations is achieved when $D^*_{SKL} = 0$.

3.4 Computation of the Inter-Song Similarity and Recommendation Score

In order to provide recommendation features for a given collection of songs, we define a ranking function $R = rank(q, B)$, where q is the input query song for the recommendation system and B is the collection of songs to query. R is a set of tuples $\langle b, r_b \rangle$, where $b \in B$ is a song belonging to the collection and r_b is the similarity score computed against the query song q. Tuples in R are sorted in increasing order by their r_b score.

To calculate r_b we proceed as follows. Let S_q and S_b be, respectively, the set of segments belonging to the query and the compared songs. We first delete from S_q and S_b those segments which are considered less important for evaluation purposes. We assume as not useful for similarity purposes such segments having a duration lower than a given threshold (in our experiments we discarded segments shorter than 10 s), for which is difficult to have a significant estimation of the Gaussian model due to insufficient number of data samples. Afterwards, the similarity evaluation process proceeds by creating a set $E(q, b)$ of edges, where each edge connects two generic segments $s_{q,i}, s_{b,j}$, respectively belonging to songs q and b. Given $N_q = |S_q|$ and $N_b = |S_b|$, the total number the edges in $E(q, b)$ is $|E(q, b)| = N_q \cdot N_b$.

An edge $e_a \in E(q, b)$, with $1 \leq a \leq |E(q, b)|$, is defined as a tuple $\langle s_{q,i}, s_{b,j}, D^*_{SKL}(\hat{s}_{q,i}, \hat{s}_{b,j}) \rangle$, where $D^*_{SKL}(\hat{s}_{q,i}, \hat{s}_{b,j})$ acts as the weighting term of the edge.

For global inter-song similarity computation, we do not use all the edges we have identified during the previous step. Instead, we adopt a graph-reduction algorithm that discards edges having low similarity value while preserving the complete coverage of the two songs (i.e., each segment of a song is connected to at least another segment of the other song), as reported in Algorithm 1: first, we sort the edges in $E(q, b)$ in descending order with respect to their weighting (similarity) term. Being the edges ordered, for every loop the set-reduction algorithm checks whether it exists in $E(q, b)$ one or more edges which depart from the segments connected by the current edge e_a. If such edge(s) exists, we delete the current edge. The algorithm proceeds with the iteration until all the edges of $E(q, b)$ have been analyzed. At the end, the final subset of non-deleted edges $A(q, b)$ will contain a set of low-weight edges which grant a full coverage on the segment sets S_q and S_b. In Algorithm 1 we denote as $s_{q,i(o)}$ and $s_{b,j(p)}$ respectively the segments of q and b connected by the edges $o, p \in T$.

Note that, in order to discard possible high-weighting outliers, we drop from the final subset $A(q, b)$ the two edges with lowest similarity; this choice is motivated by

Algorithm 1 Edges reduction algorithm

$E(q,b) \leftarrow \text{sort}(E(q,b), D^*_{SKL}(\hat{s}_{q,i}, \hat{s}_{b,j}), \text{"desc"})$
$A(q,b) \leftarrow E(q,b)$
for $a = 1$ to $|E(q,b)|$ **do**
  $T \leftarrow A - e_a$
  if $\exists o, p \in T | s_{q,i(o)} = s_{q,i(e_a)} \wedge s_{b,j(p)} = s_{b,j(e_a)}$ **then**
    $A \leftarrow T$
  end if
end for

empirical analysis that has shown how the presence of such edges heavily affects the performance of the recommendation system. Finally, the resulting similarity score is calculated as

$$r_b = \frac{1}{|A(q,b)|} \sum_{u=1}^{|A(q,b)|} w_u \tag{3}$$

where w_u is the weight of the edge $e_u \in A(q,b)$.

In the following, we describe a simple example to demonstrate the edges reduction technique depicted in Algorithm 1. Suppose to have two songs q and b for which $N_q = 2$ and $N_b = 3$. We have a total of $N_q \cdot N_b = 6$ edges which connect semantic segments of q and b. Table 2a contains an example of weighting values for each edge connecting the segments $s_{q,i}$ of q and $s_{b,j}$ of b.

Then, the edges are sorted in descending order with respect to their weights and the result is assigned to $A(q,b)$. The output of these operations is shown in Table 2b. Starting from the top of the table, the edge e_1 is deleted because there are other edges that fall into segments corresponding to $i = 1$ (edges e_4 and e_5) and $j = 1$ (edge e_3). In the next iteration, also the edge e_2 is deleted because the segment 2 of q is covered also by the edges e_3 and e_6, and the segment 2 of b is covered also by the edge e_5. The new situation of set $A(q,b)$ after two iterations is shown in Table 2c. In the third iteration, instead, the edge e_3 has to be maintained because the segment 1 of b is not covered by any other edge. Edge e_4 is the last one which is deleted (in the fourth iteration), because the segment 1 of q is already covered by edge e_5 and the segment 3 of b is already covered by edge e_6. The final $A(q,b)$ subset is presented in Table 2d.

Note that the final number of edges is only 3 instead of the starting 6, therefore the reduction algorithm has deleted the 50% of the edges in $E(q,b)$.

D^*_{SKL}	$j=1$	$j=2$	$j=3$
$i=1$	5	1	2
$i=2$	3	4	1

(a)

a	i	j	D^*_{SKL}
1	1	1	5
2	2	2	4
3	2	1	3
4	1	3	2
5	1	2	1
6	2	3	1

(b)

a	i	j	D^*_{SKL}
3	2	1	3
4	1	3	2
5	1	2	1
6	2	3	1

(c)

u	a	i	j	$w_u = D^*_{SKL}$
1	3	2	1	3
2	5	1	2	1
3	6	2	3	1

(d)

Figure 2: An example of the edge reduction algorithm.

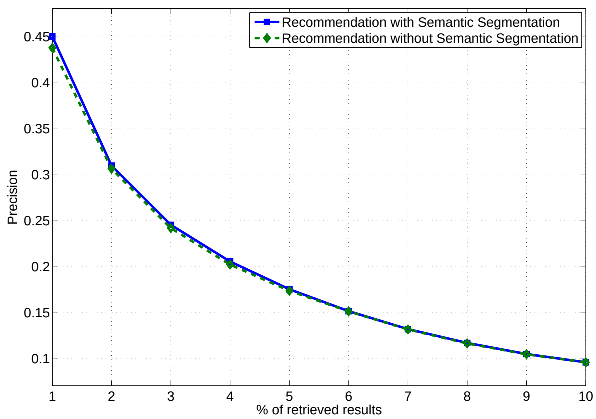
4 Experimental Evaluation

In a content-based music recommendation system, deciding if a document (i.e. a song) is relevant to a given query document is a tricky task: the relevance strictly depends on the characteristics of the raw audio file and deciding whether a song is similar to another one (or not) can be complicated also for a human being; in this situation, finding an objective measure for the evaluation of the system's effectiveness is tedious. Therefore, we decided to reduce the evaluation problem to a *genre coherency* test, where the tested property of the system is its ability to retrieve songs of the same genre as the one given as query. To achieve this goal, the test collection should be characterized by a high degree of timbral similarity between songs of the same genre: therefore we have created a testbed consisting of 200 songs, equally divided in 10 genres (*Classic*, *Dance*, *Electronic*, *Hip Hop*, *Jazz*, *Metal*, *New Age*, *Organ*, *Pop* and *Rock*) and selected them in order to maximize the timbral homogeneity inside each genre. Each song has been converted from its original format to a 22050 Hz, 16 bit, monochannel audio track and has been labeled with its genre.

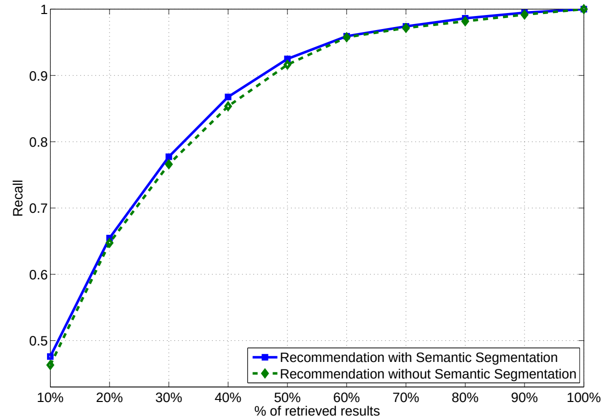
4.1 Evaluation metrics

In order to measure in a quantitative manner the performance of our recommendation system, we have adopted standard evaluation metrics for information retrieval systems. In detail, we used *Precision*, defined as the fraction of retrieved documents that are relevant, *Recall*, the fraction of relevant documents that are retrieved, and *F measure*, computed through the weighted harmonic mean between precision and recall. The advantage of having precision and recall scores is that one is more important than the other in many circumstances: in a typical on-line music recommendation system (e.g. www.last.fm), users requires high precision in order to get the first k retrieved results to be strictly relevant. In contrast, users of off-line recommendation systems (based on browsing and navigation activities) are very concerned with good recall performance, tolerating lower precision results in order to have a wider view on the collection.

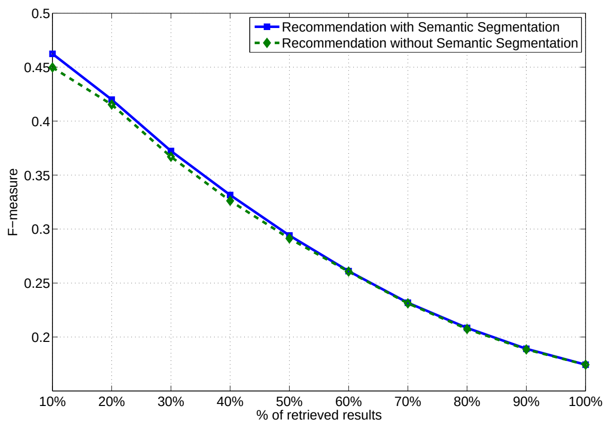
To evaluate the system with metrics that will consider its ranking performances, we adopted the *Precision-recall* curve, which provide an indicator for the precision achieved by retrieving the top-$k\%$ relevant documents in the collection. Finally, to provide a single-figure measure of effectiveness, we also measured the average precision value obtained for the top set of k documents existing after each relevant document is retrieved.



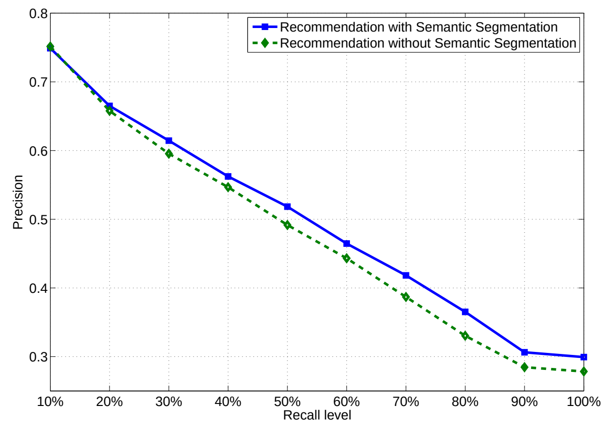
(a) Precision Curve



(b) Recall Curve



(c) F measure Curve



(d) Precision-recall Curve

Figure 3: Results comparison for quantitative analysis.

4.2 Results

To validate our system, we used a *leave one out* strategy over the existing collection, using as query a song and testing its similarity with the remaining ones, resulting in a total number of 200 queries. Results are computed as the arithmetic mean of the metrics evaluated for each query song. This has the effect of weighting each query equally, so that the final result is averaged across different songs and genres. As baseline performance comparison, we developed over the same collection a similar recommendation system which does not make use of semantic segmentation: the related similarity function models a song as a single Gaussian distribution computed over the whole track and implements feature extraction, distance metrics and evaluation methodologies as described in the previous sections.

Experimental results show that our system outperforms the baseline implementation (i.e. the one without segmentation) by increasing the retrieval average precision of more than 2% (51.7% against 49.5%), showing a good ability in retrieving with higher rank the songs actually related to the current query. Such conclusions are confirmed by the precision-recall curves for the two systems: as shown in Figure 3(d), using semantic segmentation results in better precision in almost every recall level besides the first one, which is relative to the first 10% of retrieved results, where the performances are comparable. Similar considerations can be derived by analyzing the precision curves of Figure 3(a), the recall curves of Figure 3(b) and the F-measure curves of Figure 3(c): while our system slightly improves the results for every metrics, noticeable improvements are observed on the recall curve, where our system achieves better performance by “pushing up” in the rank the songs of the same genre of the query. We can interpret such behavior as an indicator of the finer similarity resolution achieved by semantically segmenting the audio tracks: comparing songs’ parts instead of the whole tracks enhances similarities that might be lost due to the approximations introduced by the Gaussian modeling of the entire track.

As a final consideration, it has to be pointed out that increasing the granularity-level for the similarity computation (by considering similarity evaluations between semantic segments) might introduce some distortion effects deriving from the divergence function. As described in [18], the Kullback-Leibler divergence has some undesirable properties. For example, it can be observed that some particular songs, called hubs, are frequently "similar" (i.e. have a small distance) to many other pieces in the collection without sounding similar. On the other side, some pieces are never similar to others. Such behavior can be amplified on song comparison when computing (and averaging) the similarity between semantic segments; nonetheless, our graph reduction algorithm has proved effective in smoothing such effects, resulting in good overall performances.

5 CONCLUSIONS AND FUTURE WORKS

In this paper a novel approach for content-based song recommendation systems has been proposed. The main innovation consists of analyzing audio similarities between semantic segments of different audio tracks in order to make the similarity measure more meaningful to the user under a semantic point of view. Our tests show promising results for all the considered evaluation metrics. Our future work will focus on improving the performance of the recommendation system both by considering an extended set of descriptive features for audio files (including, for example, rhythmic and melodic features) and by improving the graph reduction algorithm by means of ad-hoc weighting functions for edges. Finally, it is our goal to implement the proposed approach inside a real-world prototype application, possibly also leveraging on meta-data and Web-driven retrieval.

References

[1] G. Adomavicius and A. Tuzhilin, "Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, 2005.

[2] R. Typke, F. Wiering, and R. C. Veltkamp, "A survey of music information retrieval systems," in *6th International Conference on Music Information Retrieval*, 2005.

[3] J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl, "Evaluating collaborative filtering recommender systems," *ACM Transactions on Information Systems*, vol. 22, pp. 5–53, 2004.

[4] R. Burke, "Hybrid recommender systems: Survey and experiments," *User Modeling and User-Adapted Interaction*, vol. 12, no. 4, pp. 331 – 370, (November 2002).

[5] P. Cano, M. Koppenberger, and N. Wack, "Content-based music audio recommendation," in *ACM Multimedia*, Singapore, 2005.

[6] B. Logan, "Music recommendation from song sets," in *5th International Conference on Music Information Retrieval*, Barcelona, Spain, 2004.

[7] H. C. Chen and L.P. Chen, "A music recommendation system based on music data grouping and user interests," in *ACM International Conference on Information and Knowledge Management*, 2001.

[8] F. F. Kuo and M. K. Shan, "A personalized music filtering system based on melody style classification," in *International Conference on Data Mining*, Japan, 2002.

[9] B. S. Ong and P. Herrera, "Semantic segmentation of music audio contents," in *Proceedings of International Computer Music Conference*, Barcelona, 2005, ICMA.

[10] W. Chai, "Semantic segmentation and summarization of music," *IEEE Signal Processing Magazine*, pp. 124–132, 2006.

[11] M. Levy and M. Sandler, "Lightweight measures for timbral similarity of musical audio," in *1st ACM workshop on Audio and music computing multimedia*, 2006.

[12] G. Tzanetakis and P. Cook, "Musical genre classification of audio signals," *IEEE Transactions on Speech and Audio Processing*, 2002.

[13] L. Lu, H. Zhang, and H.Jiang, "Content analysis for audio classification and segmentation," *IEEE Transaction on Speech and Audio processing*, vol. 10, pp. 504–516, 2002.

[14] G. Peeters, "A large set of audio features for sound description (similarity and classification) in the cuidado project," Tech. Rep., CUIDADO project report, 2004.

[15] S. Sigurdsson, K.B. Petersen, and T. Lehn-Schiler, "Mel frequency cepstral coefficients: An evaluation of robustness of mp3 encoded music," in *Seventh International Conference on Music Information Retrieval (ISMIR)*, 2006.

[16] M. Mandel and D. Ellis, "Song-level features and support vector machines for music classification," in *6th International Conference on Music Information Retrieval*, London, UK, 2005.

[17] W.D. Penny, "Kl-divergences of normal, gamma, dirichlet and wishart densities," Tech. Rep., Wellcome Department of Cognitive Neurology, 2001.

[18] J.-J. Aucouturier, *Ten Experiments on the Modelling of Polyphonic Timbre*, Ph.D. thesis, University of Paris, 2006.