TOOLS AND ARCHITECTURE FOR THE EVALUATION OF SIMILARITY MEASURES : CASE STUDY OF TIMBRE SIMILARITY

Jean-Julien Aucouturier
SONY CSL Paris
6, rue Amyot
75005 Paris, France.

Francois Pachet
SONY CSL Paris
6, rue Amyot
75005 Paris, France.

ABSTRACT

The systematic testing of the very many parameters and algorithmic variants involved in the design of high-level music descriptors at large, and similarity measure in particular, is a daunting task, which requires the building of a general architecture which is nearly as complex as a full-fledge Music Browsing system. In this paper, we report on experiments done in an attempt to improve the performance of the music similarity measure described in [2], using the Cuidado Music Browser ([8]). We do not principally report on the actual results of the evaluation, but rather on the methodology and the various tools that were built to support such a task. We show that many non-technical browsing features are useful at various stages of the evaluation process, and in turn that some of the tools developed for the expert user can be reinjected into the Music Browser, and benefit the non-technical user.

1. INTRODUCTION

The domain of Electronic Music Distribution has gained worldwide attention recently with progress in middleware, networking and compression. However, its success depends largely on the existence of robust, perceptually relevant music similarity relations. It is only with efficient content management techniques that the millions of music titles produced by our society can be made available to its millions of users.

1.1. Case study : Timbre Similarity

In [2], we have proposed to compute automatically music similarities between music titles based on their global timbre quality. Typical examples of timbre similarity as we define it are :

- a Schumann sonata ("Classical") and a Bill Evans piece ("Jazz") are similar because they both are romantic piano pieces,
- A Nick Drake tune ("Folk"), an acoustic tune by the Smashing Pumpkins ("Rock"), a bossa nova piece by Joao Gilberto ("World") are similar because they all consist of a simple acoustic guitar and a gentle male voice, etc.

Timbre Similarity has seen a growing interest in the Music Information Retrieval community lately (see [3, 4, 7, 11], and [1] for a complete review). Each contribution often is yet another instantiation of the same basic pattern recognition architecture, only with different algorithm variants and parameters. The signal is cut into short overlapping frames (usually between 20 and 50ms and a 50% overlap), and for each frame, a feature vector is computed, which usually consists of Mel Frequency cepstrum Coefficients (MFCC). The number of MFCCs is an important parameter, and each author comes up with a different number. Then a statistical model of the MFCCs' distribution is computed, e.g. K-means or Gaussian Mixture Models (GMMs). Once again, the number of kmean or GMM centres is a discussed parameter which has received a vast number of answers in the litterature. Finally, models are compared with different techniques, e.g. Monte Carlo sampling, Earth Mover's distance or Asymptotic Likelihood Approximation. All these contributions give encouraging results with a little effort and imply that near-perfect results would just extrapolate by fine-tuning the algorithms' parameters. However, such extensive testing over large, dependent parameter spaces is both difficult and costly.

1.2. Evaluation

The algorithm used for timbre similarity comes with very many variants, and has very many parameters to select. The parameter space for the original algorithm is at least 6-dimensional: sample rate, number of MFCCs (N), number of components (M), distance sample rate (for Monte Carlo), alternative distance (EMD, etc.), window size. Moreover, some of these parameters are not independent, e.g. there is an optimal balance to be found between high dimensionality (N) and high precision of the modeling (M). Additionally, the original algorithm may be modified by a number of classical pre/postprocessing, such as appending delta coefficients or 0th coefficient to the MFCC set. Finally, one would also like to test a number of vari-

ants, such as LPC or Spectral Contrast instead of MFCCs, HMMs or SVMs instead of GMMs, etc.

At the time of [2], the systematic evaluation of the algorithm was so unpractical that the chosen parameters resulted from hand-made parameter twitching. In more recent contributions, such as [3, 11], our measure is compared to other techniques, with similarly fixed parameters that also result from little if any systematic evaluation. More generally, attempts at evaluating different measures in the literature tend to compare individual contributions to one another, i.e. particular, discrete choices of parameters, instead of directly testing the influence of the actual parameters. For instance, [11, 3] compares the settings in [7](19 MFCCs+16Kmeans) to those of [2](8 MFCCs+3GMM).

In [4], Berenzweig et al. describe their experiments at a large-scale comparison of timbre and cultural similarity measures, on a large database on songs (8772). The experiments notably focus on the gathering of ground truth data for such a large quantity of material. Several authors have studied the problem of choosing an appropriate ground truth : [7] considers as a good match a song which is from the "same album", "same artist", "same genre" as the seed song. [11] also proposes to use "styles" (e.g. Third Wave ska revival) and "tones" (e.g. energetic) categories from the All Music Guide AMG [1] . [4] pushes the quest for ground truth one step further by mining the web to collect human similarity ratings. On the other hand, the actual number of algorithmic variants tested in [4] remains small, mainly the dimension of the statistical model, with fixed MFCC number, frame rate, etc.

One of the main obstacles to conducting such a systematic evaluation is that it requires to build a general architecture that is able to :

- access and manage the collection of music signals the measures should be tested on
- store each result for each song (or rather each duplet of songs as we are dealing with a binary operation $dist(a, b) = d$ and each set of parameters
- compare results to a ground truth, which should also be stored
- build or import this ground truth on the collection of songs according to some criteria
- easily specify the computation of different measures, and to specify different parameters for each algorithm variant
- easily manipulate the resulting similarity matrices, so they can be compared and analysed

In the context of the European project Cuidado, which ended in January 2004, the music team at SONY CSL Paris has built a fully-fledged EMD system, the Music Browser ([8]), which is to our knowledge the first system able to handle the whole chain of EMD from metadata extraction to exploitation by queries, playlists, etc. Using the Music Browser (MB), we were able to easily specify and launch a large number of experiments in an attempt to improve the performance of the class of algorithms described above. All the needed operations were done directly from the GUI, without requiring any additional programming or external program to bookkeep the computations and their results.

This paper does not focus on the actual results of the evaluation (although we report on a subset of these experiments in section 3.2). A complete account of the results of the evaluation can be found in [1]. Here, we rather focus on the methodology and the various tools that were built to support such a task. Notably, we discuss how advanced evaluation features that proved useful for the expert user may be reinjected into the MB, and benefit the non-technical user.

2. USING THE MUSIC BROWSER AS AN EVALUATION TOOL

Following our experiments with building the MB and other content-based music systems, we have started developing a general JAVA API, the so-called MCM (Multimedia Content Management), on which the current implementation of the MB relies.

2.1. Building a test database

Thanks to MCM, one can use all the browsing functionalities of the MB (editorial metadata, signal descriptors, automatic playlist generation, and even other similarity functions) to select the items which will populate the test database.

For this study, we have constructed a test database of 350 song items. In order to use the "same artist" ground truth, we select clusters of songs by the same artist. However, we refine this ground truth by hand, using the MB query panel to help us select sets of songs which satisfy 3 additional criteria.

First, clusters are chosen so they are as distant as possible from one another. This is realized e.g. by using the MB to select artists which do not have the same genre and instrumentation metadata. For instance, "Beethoven" and "Ray Charles" were selected because although they both have the same value for their "instrument" Field (i.e. "piano"), but they have distinct values for their "genre" Field ("jazz" adn "classical").

Second, artists and songs are chosen in order to have clusters that are "timbrally" consistent (all songs in each cluster sound the same). This is realized by selecting all songs by the chosen artist, and filtering this result set by the available signal descriptors in the MB (subjective energy, bpm, voice presence, etc.), and by the relevant editorial metadata (e.g. year, album).

Finally, we only select songs that are timbrally homogeneous, i.e. there is no big texture change within each song. This is to account for the fact that we only compute and compare one timbre model per song, which "merges"

[1] www.allmusic.com

all the textures found in the sound. The homogeneity of the songs can be assessed with the MB, which is able to measure some statistics on the metadata of a set of songs.

Once the songs are selected, the MB offers administrative functions to create a new database (e.g. a separated test database) and add the current result set to this new database. This also copies the associated metadata of the items (i.e. the Fields and their values), which is needed to automatically compute the "same artist" ground truth.

2.2. Generating algorithmic variants

As described in section 1.2, the algorithms evaluated here have a vast number of parameters that need to be fine-tuned. The default parameters used in [2] were based on intuition, previous work and limited manual testing during the algorithm design. No further tests had been made, because it was difficult and very-time consuming to insert new descriptors in a db, and parameterize these new descriptors. Basically, the DB had to be edited manually with a client SQL administrative tool.

The MCM API makes the whole process of creating new descriptors (or Fields) a lot easier. Each Field comes with a number of properties, stored in the db, which can be edited to create any wished number of algorithmic variants. The Field properties describe the executables that need to be called, as well as the arguments of these executables.

Figure 1 shows the properties of the Fields available in the Descriptor Manager panel of the MB. As an example, the executable of the selected Field, mfcc_d_2, has 3 arguments, 20 (the number of MFCC), 50 (the number of Gaussian Components), and 2 (the size of the delta coefficient window). The associated distance function also appears with its own parameters, here 2000 (the number of sampled points for the monte-carlo distance).

2.3. Computing similarity matrices

The Fields and the corresponding distances can then be computed on the test database using the Browser's Descriptor Manager (Figure 2). When the method `Field.compute()` is called, MCM automatically creates the database structures and the similarity cache tables to accommodate the new descriptors, as specified by their properties, and launches the corresponding executable.

2.4. Generating a Ground Truth

A ground truth which we want to compare the measure to is simply yet another similarity matrix. Several types of ground truth can be seamlessly integrated in the MB :

- ground truths based on consensual, editorial metadata about songs and artists, such as artist name, album, recording date, etc. Such similarity matrices can be computed with the descriptor manager, using a simple euclidean distance for numerical data, and either exact or approximate string matching for strings.

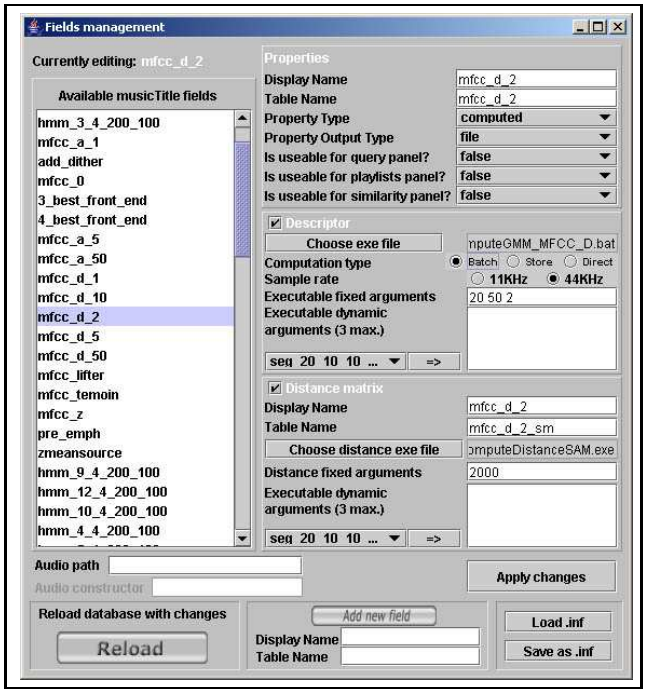

Figure 1. The Field properties as shown in the Descriptor Manager

- ground truths imported from other sources of similarity, e.g. inference of cultural similarity by mining of co-occurrences on the web ([4, 9]).
- ground truths based on user subjective judgments, generated by experiments. We are currently working on a general framework to automatically generate such user tests (in the form of web pages), and collect the results in the form of a MCM similarity matrix ([10]).

For the present study, we generate the "same artist" ground truth on the songs items by testing for equality of the "artist" Field in each song. This ground truth is stored in the form of a similarity matrix, which we will compare successively to all the computed timbre similarity matrices.

3. COMPARING MATRICES

3.1. Metrics

We have identified 2 cases in similarity matrix comparison

3.1.1. Comparing floating point matrices

The two matrices to compare contain floating-point similarity or distances values. In [4], the authors propose to use the *top-N ranking agreement score*: the top N hits from the reference matrix define the ground truth, with exponentially decaying weights so that the top hit has weight = 1, the 2^{nd} hit has weight α_r, the 3^{rd} has weight α_r^2, etc.

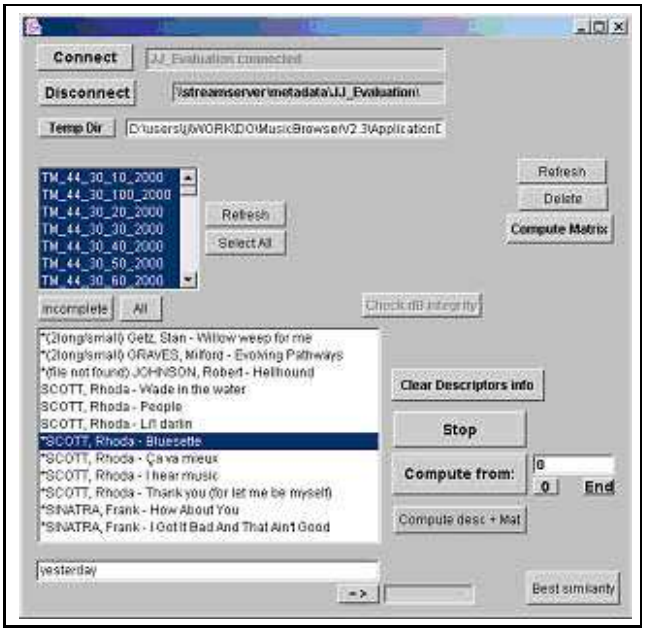

Figure 2. The Descriptor Manager

The score for the i^{th} query is defined by :

$$\mathcal{S}_i = \sum_{r=1}^{N} \alpha_r^r \alpha_c^{k_r} \quad (1)$$

where k_r is the ranking according to the candidate measure of the r^{th}-ranked hit under the ground truth. In [4], the authors use the following values $N = 10$,$\alpha_r = 0.5^{1/3}$ and $\alpha_c = \alpha_r^2$.

3.1.2. Comparing to a class matrix

A class matrix is a binary similarity matrix (similarity is either 0 or 1). This is the case in the evaluation reported here where songs from the same artist are considered relevant to one another, and songs from different artists are non relevant. This framework is very close to traditional IR, where we know the number of relevant documents for each query. The evaluation process for such tasks has been standardized within the Text REtrieval Conference (TREC) [13]. Here are the official values printed for a given retrieval task :

- total number of documents over all queries : Retrieved, Relevant, Relevant & Retrieved
- interpolated recall-precision averages, at 0.00, at 0.10, ..., at 1.00 : Measures precision (i.e. the percentage of retrieved documents that are relevant) at various recall level (i.e. after a certain percentage of all the relevant docs for that query has been retrieved).
- average precision (non interpolated) over all relevant documents
- precision, at 5 docs, at 10 docs, ..., at 1000 docs

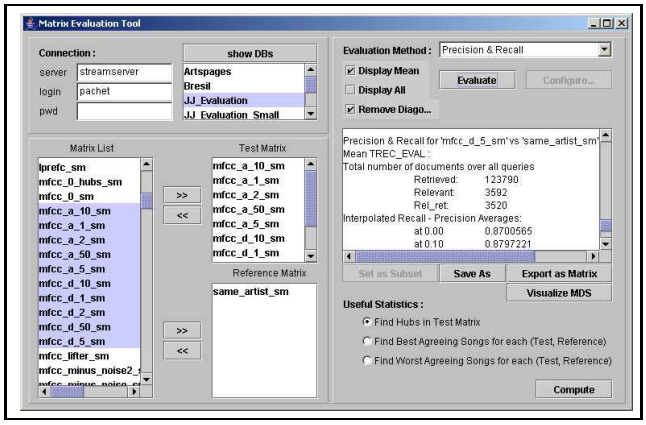

Figure 3. The evaluation tool computes R-precision for each query, and average R-precision fro the whole matrix

- R-precision :Precision after R documents have been retrieved, where R=number of relevant documents for the given query.

Figure 3 shows a screenshot of our matrix evaluation tool used to compute the evaluation values. The tool uses the standard NIST evaluation package TREC_EVAL.

3.2. Results

Here we report on a subset of the evaluation results obtained with the methodology and tools examined in this paper. A complete account and discussion of the results can be found in [1].

3.2.1. (N,M) exhaustive search

As a first evaluation, we explore the space constituted by the following 2 parameters :

- Number of MFCCs (N): The number of the MFCCs extracted from each frame of data.
- Number of Components (M): The number of gaussian components used in the GMM to model the MFCCs.

Figure 4 shows the results of the complete exploration of the (N,M) space, with N varying from 10 to 50 by steps of 10 and M from 10 to 100 by steps of 10. We use the R-precision measure as defined in section 3.1.2. We can see that too many MFCCs ($N \geq 20$) hurt the precision. When N increases, we start to take greater account of the spectrum's fast variations, which are correlated with pitch. This creates unwanted variability in the data, as we want similar timbres with different pitch to be matched nevertheless. We also notice that increasing the number of components at fixed N, and increasing N at fixed M is eventually detrimental to the precision as well. This illustrates the curse of dimensionality mentioned earlier. The best precision $p = 0.63$ is obtained for 20 MFCCs and 50 components. Compared to the original values used in [2] (N=8, M=3), this corresponds to an improvement of over 15% (absolute) R-precision.

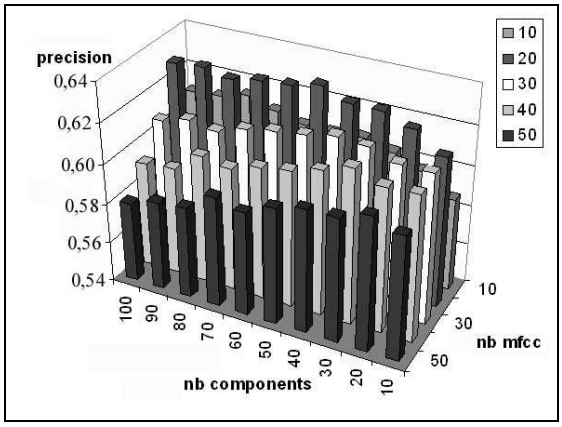

Figure 4. Influence of the number of MFCCs and the number of components

3.2.2. Hidden Markov Models

In an attempt to model the short-term dynamics of the data, we try replacing the GMMs by hidden Markov models (HMMs, see [12]). In figure 5, we report experiments using a single HMM per song, with a varying number of states in a left-right topology. The output distribution of each state is a 4-component GMM (the number of component is fixed) (see [1] for more details). From figure 5, we see that HMM modeling performs no better than static GMM modeling. The maximum R-precision of 0.632 is obtained for 12 states. Interestingly, the precision achieved with this dynamic model with 4*12=48 gaussian components is comparable to the one obtained with a static GMM with 50 states. This suggests that short-term dynamics are not a useful addition to model polyphonic mixtures.

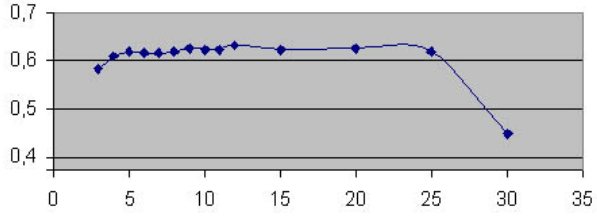

Figure 5. Influence of the number of states in HMM modeling

4. ADVANCED FEATURES

We have added a number of advanced features to the evaluation tool shown in Figure 3, in order to further discuss and analyse evaluation results.

4.1. Hubs

Interestingly, in our test database, a small number of songs seems to occur frequently as false positives. For instance (see [1] for more details), the song `MITCHELL, Joni - Don Juan's Reckless Daughter` occur more than 6 times more than it should, i.e. is very close to 1 song out of 6 in the database (57 out of 350). Among all its occurrences, many are likely to be false positives. This suggests that the errors (about 35%) are not uniformly distributed over the whole database, but are rather due to a very small number of "hubs" (less than 10%) which are close to all other songs. Using the evaluation tool, we could redo the simulation without considering the 15 biggest hubs (i.e. comparing the similarity matrices only on a subset of the items), which yield an absolute improvement of 5.2% of R-precision.

These hubs are reminiscent of a similar problem in Speech Recognition, where a small fractions of speakers (referred to as "goats", as opposed to sheeps) are responsible for the vast majority of errors ([6]). They are especially intriguing as they usually stand out of their clusters, i.e. other songs of the same cluster as a hub are not usually hubs themselves. A further study should be done with a larger test database, to see if this is only a boundary effect due to our small, specific database or a more general property of the measure.

4.2. Meta similarity

Being MCM Objects, similarities can themselves be compared and combined in a number of operations :

4.2.1. Clustering

The evaluation tool has the capacity to represent the various similarity functions as points in a 2-D space, using multidimensional scaling (MDS, see [5]) based on their mutual distances. Figure 6 compares similarities obtained with hmm and delta coefficients (another variant examined in [1]), using the Top-10 ranking agreement distance measure. It appears that while both variants remain equally close from the ground truth as their order increases, they become more and more distant to one another. In other words, low order delta and hmm similarities are fairly interchangeable, whereas higher order really capture different aspects of the data, yielding quite distinct similarity matrices.

Such similarity at the "meta" level (similarity between similarity) may prove useful for a non expert user to quickly have an idea of how various criteria available for browsing differ.

4.2.2. Agreement on queries

Similarity measures can be further compared to find the most or least agreeing queries, i.e. the line indexes in the matrices which have the smallest distance to one another. If we compare a test matrix to a ground truth, we can therefore estimate the domains on which the test matrix is the most accurate : a similarity may be good to compare jazz music, while another more suitable for electro music.

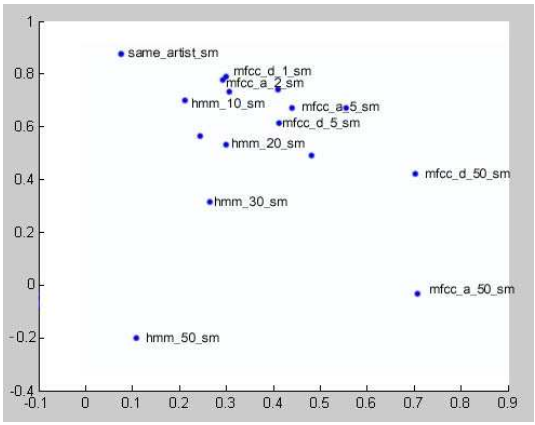

Figure 6. Multidimensional scaling of the delta-coefficient and hmm similarity matrices

4.2.3. Agreement on duplets

Matrices can also be intersected with AND or NAND operations. Combining similarities by only keeping the results on which they agree for each query (e.g. intersecting the 10 nearest neighbors result sets) results in a more consensual similarity, with a better precision but a worse recall. 2 matrices can also be inspected to find duplets of items which are close according to one similarity, but distant according to the other. We have shown in [2] how such disagreement between similarities could mine rare, interesting relations between items. For instance, acoustic cover versions of songs have a distant timbre similarity to the original version, however they are very close according to editorial metadata.

5. CONCLUSION

In this paper, we have reported on tools built in an attempt to improve the performance of the music similarity measure described in [2], using the Cuidado Music Browser ([8]). We have shown that many non-technical browsing features are crucial at various stages of the evaluation process, in order e.g. to build a test database or to generate a ground truth. In turn, we have introduced evaluation tools which can benefit the non-technical user while browsing for music. Matrix distances may help her choose between different available search criteria, while matrix intersection can help her find rare, interesting songs. We believe that this case study shows the importance of designing complete evaluation environments for MIR researchers. Moreover, the convergence in needs and tools of the process of browsing and the process of designing browsing tools opens up the way for a unified language, toward which the MCM API may be a first step.

6. REFERENCES

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

[2] J.-J. Aucouturier and F. Pachet. Music similarity measures: What's the use ? In *Proceedings of the 3rd International Conference on Music Information Retrieval (ISMIR), Paris (France)*, October 2002.

[3] S. Baumann and T. Pohle. A comparison of music similarity measures for a p2p application. In *Proceedings of the Sixth International Conference on Digital Audio Effects DAFX, London (UK)*, September 2003.

[4] A. Berenzweig, B. Logan, D. P. W. Ellis, and B. Whitman. A large-scale evaluation of acoustic and subjective music similarity measures. In *Proceedings of the Fourth International Conference on Music Information Retrieval (ISMIR 2003), Baltimore, Maryland (USA)*, October 2003.

[5] I. Borg and P. Groenen. *Modern Multidimensional Scaling*. Springer-Verlag, New York., 1997.

[6] G. Doddington, W. Liggett, A. Martin, M. Przybocki, and D. Reynolds. Sheep, goats, lambs and wolves, a statistical analysis of speaker performance. In *Proceedings of the 5th International Conference on Spoken Language Processing (ICSLP), Sydney, Australia.*, December 1998.

[7] B. Logan and A. Salomon. A music similarity function based on signal analysis. In *in proceedings IEEE International Conference on Multimedia and Expo (ICME), Tokyo (Japan)*, August 2001.

[8] F. Pachet, A. LaBurthe, A. Zils, and J.-J. Aucouturier. Popular music access: The sony music browser. *Journal of the American Society for Information (JASIS), Special Issue on Music Information Retrieval*, 2004.

[9] F. Pachet, G. Westermann, and D. Laigre. Musical datamining for emd. In *Proceedings Wedelmusic*, 2001.

[10] F. Pachet and A. Zils. Evolving automatically high-level music descriptors from acoustic signals. In *Springer Verlag LNCS, 2771*, 2003.

[11] E. Pampalk, S. Dixon, and G. Widmer. On the evaluation of perceptual similarity measures for music. In *Proceedings of the Sixth International Conference on Digital Audio Effects DAFX, London (UK)*, September 2003.

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

[13] E. Voorhes and D. Harman. Overview of the eighth text retrieval conference. In *Proceedings of the Eighth Text Retrieval Conference (TREC), Gaithersburg, Maryland (USA)*, November 1999. http://trec.nist.gov.