THE HYDRA SYSTEM OF UNSTRUCTURED COVER SONG DETECTION

Suman Ravuri

Intl. Computer Science Inst., UC Berkeley
Dept. of Electrical Engineering
Berkeley, CA

Daniel P.W. Ellis

LabROSA, Columbia University
Dept. of Electrical Engineering
New York, NY

ABSTRACT

We describe the Hydra cover song detector. The system aims to solve this problem: given the reference track and a test track and no prior knowledge of the structure of the dataset, identify the reference/test pair as either a reference/cover or reference/non-cover. While this is not the specification of the MIREX audio cover song contest, we have found that the system has a 25.9% relative improvement from the highest published score on the "covers80" test set in the MIREX-style evaluation.

***Index Terms*—** MIREX, Automatic Cover Song Detection

1. INTRODUCTION

The goal of automatic cover song detection over the past few years, in large part due to the MIREX competition, is - given a reference track - to rank a list of m songs so that the top n songs are covers to the reference. The MIREX community has made substantial progress in the performance of these systems over the past few years, increasing scores from 761/3300 in the 2006 competition to 2422/3300 in the 2008 one.

While there has been great work in this high-score setting, ultimately, the goal of cover song detection should be - given a reference song and a test song - to classify the pair as either a reference/cover or reference/non-cover. The proposed system builds upon the knowledge of previous systems to perform this sort of general classification.

Since our system focuses on classification instead of maximizing high scores, the algorithm is handicapped compared to other systems in the MIREX evaluation. In particular, there exist two problems with general classification that do not exist with its high score counterpart. The first is that scores need to be normalized so that there exists a single threshold that identifies reference/covers. Other systems may have a distance of .001 for covers of one particular query and .005 for covers of a different query. Since we have to create a single threshold no matter the query, the normalization scheme, if not done properly, could degrade performance in a high-score evaluation. Luckily, the proposed normalization scheme actually seems to improve performance in a high-score setting. See Section 5 for more details.

A far bigger handicap compared to other systems is that the submitted system is not using the structure of the test data to help performance. [1] and the upcoming paper [2] outline how one can improve cover song detection knowing that there exist sets of 10 covers for each reference in the test set. While these ideas should be included in any system in order to achieve the highest scores in the MIREX evaluation, general classification does not have that sort of structured knowledge. Hence, we did not include any cover set detection components in our algorithm.

2. COVER SONG DETECTION SYSTEM OVERVIEW

Existing cover song detection systems[1] suffer from two problems. The first is that they only calculate one feature and use that one feature to identify all types of covers. A cover in one genre, however, may not have the same properties as a cover in another. We have found that a multistream approach will lead to better and more robust classification. Different features can be optimized for different types of covers and with smart feature combination, we can build a better cover song detector.

The second problem with most algorithms is that they try to calculate features and perform classification in one step. In every previous method, the system creates score/distance features and simply outputs the best scores as reference/cover pairs. This approach, however, makes general classification very difficult because scores for one reference/cover pair may vastly differ from another reference/cover pair. By viewing the matching scores as features and training a separate classifier (such as a support vector machine), we can not only do general classification, but also improve performance in high score evaluation. Moreover, this framework has the added advantage that multiple features can be combined with minimal effort.

Figure 2 shows a block diagram of the proposed system. The first step is to calculate beat-synchronous chroma representations (as described in [3]) of the reference and test

[1] Figure 1 shows the block diagram of current approaches

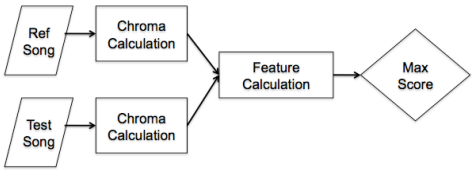

Fig. 1. Block diagram of current approaches to cover song identification.

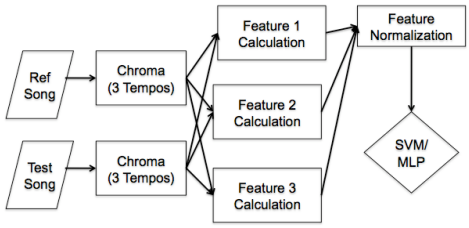

Fig. 2. Block Diagram of my approach to cover song identification.

tracks. Next, the system calculates three features (two cross-correlation features and one dynamic programming feature) at three different tempo levels, yielding 9 different features.[2] The system then performs feature normalization and finally, a support vector machine classifies this multidimensional feature as either of a reference/cover pair or not.

3. FEATURES

3.1. Feature 1: Cross-correlation Feature 1

This feature is the one used in the 2006 LabROSA system and details of its calculation can be found in [4]. The system square-root compresses the reference and test chroma and cross-correlates the two resulting chroma. The utility of the cross-correlation arises from the observation that if fragments of the reference and test track match - as often happens in a cover - the cross-correlation will exhibit rapidly-changing peaks at different time lags. This cross-correlation is performed for all twelve circular shifts of the test chroma and the shift for which the highest cross-correlation peak score occurs is selected. Then, that cross-correlation is high-pass filtered to remove the general triangular structure, leaving only the peaks. Finally, the score of the maximum peak is used as the feature.

3.2. Feature 2: Cross-correlation Feature 2

This cross-correlation feature, used in the 2007 system and described in [5], is a minor variant of that described in Section 3.1. For this feature, each beat vector of the chromagram is normalized to sum to one after square-root compression. Then, the chroma itself is high-pass filtered to de-emphasize a same note being played for multiple beats. Then, cross-correlation is performed as above, and the maximum value of all twelve cross-correlations is outputted as the feature. [5] cited a performance improvement of this feature as a reason to switch features, but we have since found that keeping both features leads to better performance in both the high score and general classification settings.

3.3. Feature 3: Dynamic Programming Feature

This is a feature re-implemented from [6] and modified for use with beat-synchronous, 12-dimensional chroma instead of 93ms-windowed, 36-dimensional Harmonic Pitch Class Profiles.

The dynamic programming feature is a two stage process. First, the cover song detection system calculates a "binary similarity matrix" of the reference/test pair. Then, the Smith-Waterman algorithm is run on the binary similarity matrix, and the highest value of the dynamic program is returned as a feature. Details of the calculation can be found in [7].

4. CHROMA: MULTIPLE TEMPO LEVELS

Sometimes, an incorrect tempo level in the beat-tracking algorithm will lead to poor representation of the musical progression of the reference or cover track. This will invariably lead to bad feature scores, even if the features themselves are somewhat robust to changes in melody. In order to circumvent this problem, we calculate the three above features from chroma beat-tracked at 240 beats/minute, 120 beats/minute, and 60 beats/minute. We also experimented with mixing tempo levels (i.e. using 240 beats/minute for the reference track and 120 beats/minutes for the test track), but including these cross-tempos resulted in no performance improvement.

5. SYSTEM: FEATURE NORMALIZATION

In order to introduce the idea of feature normalization, consider a chromagram of a test song such that all the semitones were of equal value and the beats had equal energy to each other (this would be a "white noisy" track). Such an "impostor" track would score highly on all three aforementioned feature calculations and the test track would be classified a cover for every reference song.

In order to combat this problem, for every test track, we calculate features with random reference tracks and take a mean and standard deviation of these features. Since the prior probability of a reference/test pair being a cover is much less than 1%, we can consider this feature normalization to be a form of crude modeling on how the test track performs with

[2]Hence the name of the system as Hydra.

Feature Name	unnorm score	norm score	relative impr
feat 1 240 bpm	42/80	46/80	9.5%
feat 1 120 bpm	42/80	49/80	16.7%
feat 1 60 bpm	43/80	45/80	4.7%
feat 2 240 bpm	45/80	50/80	11.1%
feat 2 120 bpm	48/80	50/80	4.2%
feat 2 60 bpm	51/80	54/80	5.9%
feat 3 240 bpm	44/80	48/80	9.1%
feat 3 120 bpm	41/80	49/80	19.5%
feat 3 60 bpm	41/80	51/80	24.4%

Table 1. Performance on covers80 test set for unnormalized and normalized features

random non-cover reference tracks. We then mean/variance normalize the features to obtain a z-score for use during classification.

Table 1 shows the unnormalized and normalized scores on the covers80 test set. Normalized features provide between a 4.7% and 24.4% improvement over its unnormalized counterparts. Feature 3 had the most dramatic performance gains, while all the features show at least some modest improvement.

6. SYSTEM: CLASSIFICATION

We train a support vector machine to classify cover songs. Training is done on hypeful.com's 25 best covers of 2008.[3] The set consists of 34 original tracks and 39 covers. Most songs are pop music and between 1 and 3 cover songs exist for each original track.

For training, one has to be careful not to train the classifier on too many reference/non-cover pairs. If, for example, one trains the classifier on all possible combinations of the 34 reference tracks and 39 covers, there would be 39 reference/cover pairs and 1287 reference/non-covers in the training set and during test the classifier would determine that every reference/test pair is a non-cover. We found that removing 75% of the reference/non-cover training examples yields weights that perform well in both general classification and high-score evaluations.

Table 2 shows the covers80 high score results for the 2007 LabROSA system (which has the highest published covers80 score to date), each feature with the maximum score classifier, all features with a maximum score classifier, and Hydra. Hydra performs roughly 25% better than the 2007 LabROSA system and offers an improvement over other systems.

[3] http://www.hypeful.com/2008/12/23/25-best-cover-songs-of-2008/

System Name	covers80 score
LabROSA 2007 MIREX sub.	54/80
feat 1 (3 tempo) Max Score	55/80
feat 2 (3 tempo) Max Score	59/80
feat 3 (3 tempo) Max Score	57/80
Max Score all feats	64/80
Hydra	**67/80**

Table 2. High score performance for different systems

7. OPEN SOURCE SYSTEM AVAILABLE

We have made the source code for our system available at http://www.icsi.berkeley.edu/~ravuri under the GPL license in order to reduce the barrier for entry for participants for future years. We hope that other competitors will also make their code available so that we can improve performance on various tasks at a much greater rate.

8. ACKNOWLEDGMENTS

I would like to thank Professor Nelson Morgan for discussing ideas and suggesting improvements for the algorithm.

9. REFERENCES

[1] A. Egorov and G. Linetsky, "Cover song identification with if-f0 pitch class profiles," *Music Information Retrieval Evaluation eXchange (MIREX)*, 2008.

[2] J. Serra, M. Zanin, C. Laurier, and M. Sordo, "Unsupervised detection of cover song sets: Accuracy improvement and original identification," *International Society for Music Information Retrieval (ISMIR)*, 2009.

[3] D.P.W. Ellis, "Identifying 'cover songs' with beat-synchronous chroma features," *Music Information Retrieval Evaluation eXchange (MIREX)*, 2006.

[4] D.P.W. Ellis and G. Poliner, "Identifying 'cover songs' with chroma features and dynamic programming beat tracking," *IEEE Transactions on Audio, Speech, and Language Processing*, pp. 1429–1432, April 2007.

[5] D.P.W. Ellis and C. Cotton, "The 2007 labrosa cover song detection system," *Music Information Retrieval Evaluation eXchange (MIREX)*, 2007.

[6] J. Serra and E. Gomez, "A cover song identification system based on sequences of tonal descriptors," *Music Information Retrieval Evaluation eXchange (MIREX)*, 2007.

[7] J. Serra, E. Gomez, P. Herrera, and X. Serra, "Chroma binary similarity and local alignment applied to cover song identification," *IEEE Transactions on Audio, Speech, and*

Language Processing, vol. 16, pp. 1138–1151, August 2008.