ROBUST MUSIC IDENTIFICATION, DETECTION, AND ANALYSIS

Mehryar Mohri[1,2]**, Pedro Moreno**[2]**, and Eugene Weinstein**[1,2]
[1] Courant Institute of Mathematical Sciences
251 Mercer Street, New York, NY 10012.
[2] Google Inc.
76 Ninth Avenue, New York, NY 10011.

ABSTRACT

In previous work, we presented a new approach to music identification based on finite-state transducers and Gaussian mixture models. Here, we expand this work and study the performance of our system in the presence of noise and distortions. We also evaluate a song detection method based on a universal background model in combination with a support vector machine classifier and provide some insight into why our transducer representation allows for accurate identification even when only a short song snippet is available.

1 INTRODUCTION

Automatic detection and identification of music have been the subject of several recent studies both in research [5, 6, 1] and industry [16, 4]. Music identification consists of determining the identity of the song matching a partial recording supplied by the user. In addition to allowing the user to search for a song, it can be used by content distribution networks such as Google YouTube to identify copyrighted audio within their systems and for recording labels to monitor radio broadcasts to ensure correct accounting.

Music identification is a challenging task because the partial recording supplied may be distorted due to noise or channel effects. Moreover, the test recording may be short and consist of just a few seconds of audio. Since the size of the database is limited another crucial task is that of *music detection*, that is that of determining if the recording supplied contains an in-set song.

Previous work in music identification (see [2] for a recent survey) can be classified into hashing and non-hashing approaches. The hashing approach involves computing local fingerprints, that is feature values over a window, retrieving candidate songs matching the fingerprints from a database indexed by a hash table, and picking amongst the candidates using some accuracy metric. Haitsma et al. [5] used hand-crafted features of energy differences between Bark-scale cepstra. The fingerprints thus computed were looked up in a large hash table of fingerprints for all songs in the database. Ke et al. [6] used a similar approach, but selected the features automatically using boosting. Covell et al. [4] further improve on Ke and extend the technique beyond music to broadcast news identification.

Two main limitations of hashing approaches are the requirement to match a fingerprint exactly or almost exactly, and the need for a disambiguation step to reject false positive matches. In contrast, the use of Gaussian mixtures allows our system to tolerate variations in acoustic conditions more naturally. Our use of finite state transducers (FSTs) allows us to index music event sequences in an optimal and compact way and, as demonstrated in this work, is highly unlikely to yield a false positive match. Finally, this representation permits the modeling and analysis of song structure by locating similar sound sequences within a song or across multiple songs.

An example of a non-hashing approach is the work of Batlle et al [1]. They proposed decoding MFCC features over the audio stream directly into a sequence of audio events, as in speech recognition. Both the decoding and the mapping of sound sequences to songs is driven by hidden Markov models (HMMs). However, the system looks only for atomic sound sequences of a particular length, presumably to control search complexity.

Our own music identification system was first presented in Weinstein and Moreno [17]. Our approach is to automatically select an inventory of music sound events using clustering and train acoustic models for each such event. We then use finite-state transducers to represent music sequences and guide the search process efficiently. In contrast to previous work, ours allows recognition of an arbitrarily long song segment. In previous work, we reported the identification accuracy of our music processing system in ideal conditions. Here, we examine the problem of music detection and identification under adverse conditions, such as additive noise, time stretching and compression, and encoding at low bit rates.

2 MUSIC IDENTIFICATION

2.1 Acoustic Modeling

Our acoustic modeling approach consists of jointly learning an inventory of *music phones* and the sequence of phones best representing each song. We compute mel-frequency cepstral coefficient (MFCC) features for each song. Cepstra have recently been shown to be effective in the analysis of music [1, 15, 7]. We use 100ms windows over the feature stream, and keep the first twelve coefficients, the energy, and their first and second derivatives to produce a 39-dimensional feature vector.

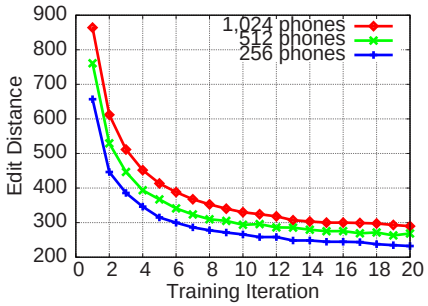

Figure 1. Average edit distance per song vs. training iteration.

Each song is initially broken into pseudo-stationary segments. Single diagonal covariance Gaussian models are fitted to each window. We hypothesize segment boundaries where the KL divergence between adjacent windows is above an experimentally determined threshold. We then apply divisive clustering to the song segments in which all points are initially assigned to one cluster. At each clustering iteration, the centroid of each cluster is perturbed in two opposite directions of maximum variance to make two new clusters. Points are reassigned to the new cluster with the higher likelihood [8]. In a second step we apply k-means clustering. For each cluster, we train a single initial diagonal covariance Gaussian model.

The standard EM algorithm for Gaussian mixture model (GMM) training cannot be used since there are no reference song transcriptions. Instead, we use an unsupervised learning approach similar to that of [1] in which the statistics representing each music phone and the transcriptions are inferred simultaneously. We alternate between finding the best transcription per song given the current model and refining the GMMs given the current transcriptions.

To measure the convergence of our algorithm we use the edit distance, here defined as the minimal number of insertions, substitutions, and deletions of music phones required to transform one transcription into another. For a song set S let $t_i(s)$ be the transcription of song s at iteration i and $\mathrm{ED}(a, b)$ the edit distance of sequences a and b. At each iteration i, we compute the total edit distance $C_i = \sum_{s \in S} \mathrm{ED}(t_i(s), t_{i-1}(s))$ as our convergence measure. Figure 1 illustrates how this quantity changes during training for three phone inventory sizes, and shows that it converges after around twenty iterations.

2.2 Recognition Transducer

Our music identification system is based on weighted finite-state transducers and Viterbi decoding as is common in speech recognition [12]. The decoding is based on the acoustic model described in the previous section and a compact transducer that maps music phone sequences to corresponding song identifiers.

Given a finite set of songs S, the music identification task is to find the songs in S that contain a query song snippet x. Hence, the recognition transducer must map any sequence of music phones appearing in a song to the corresponding song identifiers.

More formally, let Δ denote the set of music phones. The song set $S = \{x_1, \ldots, x_m\}$ is a set of sequences in Δ^*. A *factor*, or *substring*, of a sequence $x \in \Delta^*$ is a sequence of consecutive phones appearing in x. Thus, y is a factor of x iff there exists $u, v \in \Delta^*$ such that $x = uyv$. The set of factors of x is denoted by $\mathrm{Fact}(x)$ and more generally the set of factors of all songs in S is denoted by $\mathrm{Fact}(S)$. A correct transcription of an in-set song snippet is thus an element of $\mathrm{Fact}(S)$. The recognition transducer T must thus represent a mapping from transcription factors to numerical song identifiers:

$$\begin{array}{rccl} [\![T]\!] : \mathrm{Fact}(S) & \to & \mathbb{N} \\ x & \mapsto & [\![T]\!](x) = y_x. \end{array} \quad (1)$$

Figure 2 shows a transducer T_0 mapping each song to its identifier, when S is reduced to three short songs. We can construct a factor transducer from T_0 simply by adding ϵ transitions from the initial state to each state and by making each state final. However, in order for efficient search to be possible, the transducer must further be deterministic and minimal. Determinizing the transducer constructed in this fashion can result in an exponential size blowup. In our previous work [17], we gave a method for constructing a compact recognition transducer T using weights to represent song identifiers with the help of weighted determinization and minimization algorithms [9, 11].

We have empirically verified the feasibility of this construction. For $15{,}455$ songs, the total number of transitions of the transducer T is about 53.0M, only about 2.1 times that of the minimal deterministic transducer T_0 representing all songs. We present elsewhere a careful analysis of the size of the factor automaton of an automaton and provide worst case bounds in terms of the size of the original automaton or transducer representing all songs [13]. These bounds suggest that our method can scale to a larger set of songs, e.g., several million songs.

2.3 Improving Robustness

In the presence of noise or distortions, the recognized music phone sequence x may be corrupted by decoding errors. However, the transducer T associating music phone sequences to song identifiers only accepts correct music phone sequences as inputs. To improve robustness, we can compose a transducer T_E with T that allows corrupted transcriptions to also be accepted, resulting in the mapping $[\![T \circ T_E]\!](x)$. A particular corruption transducer T_E is the edit distance transducer, which associates a cost to each edit operation [10]. In this case, the above composition has the effect of allowing insertions, deletions, and substitutions to corrupt the input sequence x while penalizing any path allowing such corruptions in the Viterbi beam search algorithm. The costs may be determined analytically to reflect a desired set of penalties, or may be learned to maximize identification accuracy.

Robustness can also be improved by including data reflecting the expected noise and distortion conditions in the acoustic model training process. The resulting models are then adapted to handle similar conditions in the test data.

3 MUSIC DETECTION

Our music detection approach relies on the use of a universal background music phone model (UBM) model that

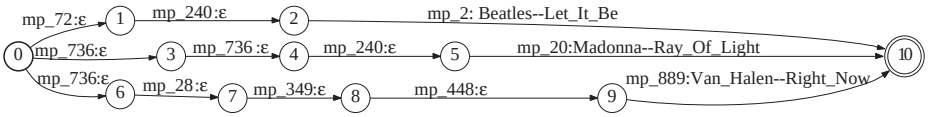

Figure 2. Finite-state transducer T_0 mapping each song to its identifier.

generically represents all possible song sounds. This is similar to the techniques used in speaker identification (e.g., [14]). The UBM is constructed by combining the GMMs of all the music phones. We apply a divisive clustering algorithm to yield a desired number of mixture components.

To detect out-of-set songs, we compute the log-likelihood of the best path in a Viterbi search through the regular song identification transducer and that given a trivial transducer that allows only the UBM. When the likelihood ratio of the two models is large, one can be expect the song to be in the training set. However, a simple threshold on the likelihood ratio is not a powerful enough classifier for accurate detection. Instead, we have been using a discriminative method for out-of-set detection. We construct a three-dimensional feature vector $[L_r, L_b, (L_r - L_b)]$ for each song snippet, where L_r and L_b are the log-likelihoods of the best path and background acoustic models, respectively. These serve as the features for a support vector machine (SVM) classifier [3].

4 EXPERIMENTS

Our training data set consisted of $15{,}455$ songs. The average song duration was 3.9 minutes, for a total of over $1{,}000$ hours of training audio. The test data consisted of $1{,}762$ in-set and $1{,}856$ out-of-set 10-second snippets drawn from 100 in-set and 100 out-of-set songs selected at random. The first and last 20 seconds of each song were omitted from the test data since they were more likely to consist of primarily silence or very quiet audio. Our music phone inventory size was $1,024$ units, each model consisting of 16 mixture components. For the music detection experiments, we also used a UBM with 16 components. We tested the robustness of our system by applying the following transformations to the audio snippets:

a. WNoise-x: additive white noise (using `sox`). Since white noise is a consistently broadband signal, this simulates harsh noise. x is the noise amplitude compared to saturation (i.e., WNoise-0.01 is 0.01 of saturation).

b. Speed-x: speed up or slow down by factor of x (using `sox`). Radio stations frequently speed up or slow down songs in order to produce more appealing sound [1].

c. MP3-x: mp3 reencode at x kbps (using `lame`). This simulates compression or transmission at a lower bitrate.

For the detection experiments we used the `LIBSVM` implementation with a radial basis function (RBF) kernel. The accuracy was measured using 10-fold cross-validation and a grid search for the values of γ in the RBF kernel and the trade-off parameter C of support vector machines [3].

The identification and detection accuracy results are presented in Table 1. The identification performance is almost flawless on clean data. The addition of white noise degrades the accuracy when the mixing level of the noise is increased. This is to be expected as the higher mixing levels result in a low signal-to-noise ratio (SNR). The inclusion of noisy data in the acoustic model training process slightly improves identification quality – for instance, in the WNoise-0.01 experiment, the accuracy improves from 85.5% to 88.4%. Slight variations in playback speed are handled quite well by our system (high 90's); however, major variations such as 0.9x and 1.1x cause the accuracy to degrade into the 40's. MP3 recompression at low bitrates is handled well by our system.

Table 1. Identification accuracy rates under various test conditions

Condition	Identification Accuracy	Detection Accuracy
Clean	99.4%	96.9%
WNoise-0.001 (44.0 dB SNR)	98.5%	96.8%
WNoise-0.01 (24.8 dB SNR)	85.5%	94.5%
WNoise-0.05 (10.4 dB SNR)	39.0%	93.2%
WNoise-0.1 (5.9 dB SNR)	11.1%	93.5%
Speed-0.98	96.8%	96.0%
Speed-1.02	98.4%	96.4%
Speed-0.9	45.7%	85.8%
Speed-1.1	43.2%	87.7%
MP3-64	98.1%	96.6%
MP3-32	95.5%	95.3%

The detection performance of our system is in the 90's for all conditions except the 10% speedup and slowdown. This is most likely due to the spectral shift introduced by the speed alteration technique. This shift results in a mismatch between the audio data and the acoustic models. We believe that a time scaling method that maintains spectral characteristics would be handled better by our acoustic models. We will test this assumption in future work.

5 FACTOR UNIQUENESS ANALYSIS

We observed that our identification system performs well when snippets of five seconds or longer are used. Indeed, there is almost no improvement when the snippet length increases from ten seconds to the full song. To further analyze this, we examined the sharing of factors across songs. Let two song transcriptions $x_1, x_2 \in S$ share a common factor $f \in \Delta^*$ such that $x_1 = ufv$ and $x_2 = afc$; $u, v, a, c \in \Delta^*$. Then the sections in these two songs transcribed by f are similar. Further, if a song x_1 has a repeated factor $f \in \Delta^*$ such that $x_1 = ufvfw$; $u, v, w \in \Delta^*$, then x_1 has two similar audio segments. If $|f|$ is large, then it is unlikely that the sharing of f is coincidental, and likely represents a repeated structural element in the song.

Figure 3 gives the number of non-unique factors over a range of lengths. This illustrates that some sharing of long elements is present, indicating similar music segments across songs. However, factor collisions decrease rapidly as the factor length increases. For example, we can

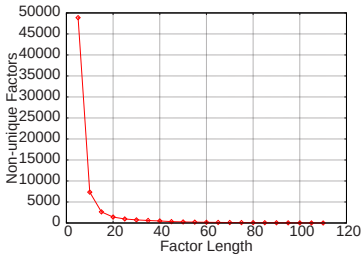

Figure 3. Number of factors occurring in more than one song in S for different factor lengths.

see that for factor length of 50, only 256 out of the 24.4M existing factors appear in more than one song. Considering that the average duration of a music phone in our experiments is around 200ms, a factor length of 50 corresponds to around ten seconds of audio. This validates our initial estimate that ten seconds of music are sufficient to uniquely map the audio to a song in our database. In fact, even with factor length of 25 music phones, there are only 962 non-unique factors out of $23.9M$ total factors. This explains why even a five-second snippet is sufficient for accurate identification.

6 CONCLUSION

We described a music identification system based on Gaussian mixture models and weighted finite-state transducers and its performance in the presence of noise and other distortions. Our approach allows us to leverage the robustness of GMMs to maintain good accuracy in the presence of low to medium noise levels. In addition, the compact representation of the mapping of music phones to songs allows for efficient decoding, and thus high accuracy. We have also implemented a music detection system using the likelihoods of the decoder output as input to a support vector machine classifier and provided an empirical analysis of factor uniqueness across songs, verifying that five-second or longer song snippets are sufficient for very low factor collision and thus accurate identification.

Acknowledgements

We thank the members of the Google speech team, in particular Michiel Bacchiani, Mike Cohen, Michael Riley, and Johan Schalkwyk, for their help, advice, and support. The work of Mehryar Mohri and Eugene Weinstein was partially supported by the New York State Office of Science Technology and Academic Research (NYSTAR). This project was also sponsored in part by the Department of the Army Award Number W81XWH-04-1-0307. The U.S. Army Medical Research Acquisition Activity, 820 Chandler Street, Fort Detrick MD 21702-5014 is the awarding and administering acquisition office. The content of this material does not necessarily reflect the position or the policy of the Government and no official endorsement should be inferred.

7 REFERENCES

[1] E. Batlle, J. Masip, and E. Guaus. Automatic song identification in noisy broadcast audio. In *IASTED International Conference on Signal and Image Processing*, Kauai, Hawaii, 2002.

[2] P. Cano, E. Batlle, T. Kalker, and J. Haitsma. A review of audio fingerprinting. *Journal of VLSI Signal Processing Systems*, 41:271–284, 2005.

[3] C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.

[4] M. Covell and S. Baluja. Audio fingerprinting: Combining computer vision & data stream processing. In *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Honolulu, Hawaii, 2007.

[5] J. Haitsma, T. Kalker, and J. Oostveen. Robust audio hashing for content identification. In *Content-Based Multimedia Indexing (CBMI)*, Brescia, Italy, September 2001.

[6] Y. Ke, D. Hoiem, and R. Sukthankar. Computer vision for music identification. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 597–604, San Diego, June 2005.

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

[8] M.Bacchiani and M. Ostendorf. Joint lexicon, acoustic unit inventory and model design. *Speech Communication*, 29:99–114, November 1999.

[9] M. Mohri. Finite-state transducers in language and speech processing. *Computational Linguistics*, 23(2):269–311, 1997.

[10] M. Mohri. Edit-distance of weighted automata: General definitions and algorithms. *International Journal of Foundations of Computer Science*, 14(6):957–982, 2003.

[11] M. Mohri. Statistical Natural Language Processing. In M. Lothaire, editor, *Applied Combinatorics on Words*. Cambridge University Press, 2005.

[12] M. Mohri, F. C. N. Pereira, and M. Riley. Weighted Finite-State Transducers in Speech Recognition. *Computer Speech and Language*, 16(1):69–88, 2002.

[13] Mehryar Mohri, Pedro Moreno, and Eugene Weinstein. Factor automata of automata and applications. In *12th International Conference on Implementation and Application of Automata (CIAA)*, Prague, Czech Republic, July 2007.

[14] A. Park and T.J. Hazen. ASR dependent techniques for speaker identification. In *International Conference on Spoken Language Processing (ICSLP)*, Denver, Colorado, USA, September 2002.

[15] D. Pye. Content-based methods for the management of digital music. In *ICASSP*, pages 2437–2440, Istanbul, Turkey, June 2000.

[16] A. L. Wang. An industrial-strength audio search algorithm. In *International Conference on Music Information Retrieval (ISMIR)*, Washington, DC, October 2003.

[17] E. Weinstein and P. Moreno. Music identification with weighted finite-state transducers. In *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Honolulu, Hawaii, 2007.