AN HMM-BASED AUDIO CHORD DETECTION SYSTEM ATTENUATING THE MAIN MELODY

Jan Weil
Technische Universität Berlin
Communication Systems Group
Einsteinufer 17
10587 Berlin, Germany
weil@nue.tu-berlin.de

Jean-Louis Durrieu
TELECOM ParisTech
TSI / LTCI
46, rue Barrault
75634 Paris Cedex 13, France
durrieu@enst.fr

ABSTRACT

This extended abstract describes an algorithm which was submitted to the 2008 Music Information Retrieval eXchange in the audio chord detection task. As it is not pre-trained it aims at the n-fold training/testing subtask. Part of this submission is a source separation algorithm which is used to attenuate the main melody. [1]

1 INTRODUCTION

The algorithm presented in this extended abstract is similar in principle to others following up on the system described by Bello and Pickens in 2005 [1], e. g., [5]. The chords are modelled as states of an ergodic HMM. For this task only major and minor chords are considered resulting in 24 states. The observation sequences are computed directly from the audio signal. The features used are tonal centroid vectors as described in [4]. Their distribution is modelled as a single multi-variate Gaussian for each state. The sequence of chords is given by decoding the HMM. What makes our system distinct is an additional pre-processing step which attenuates the main melody. As we found, this helps to further improve recognition. We believe that this is due to the fact that the main melody often contains intentionally in-harmonic notes. While these notes are crucial for the perceived richness of the global timbre, they also blur the accompanying harmonies, which makes recognizing them harder. The drawback of our approach is its high computational burden. There are, however, applications for which the separation of the main melody is worthwhile anyway, e. g., the automatic generation of lead sheets [3].

[1] This research was supported by the European Commission under contract *FP6-027026-K-SPACE*.

2 TRAINING

2.1 Pre-processing

To make the feature extraction process more efficient the audio data is mixed down to a single channel by taking the mean of all channels. Afterwards it is downsampled to 11025 Hz.

2.2 Main Melody Attenuation

The resulting signal is fed into the main melody separation algorithm, which incorporates a model of the human voice [2]. The short-time Fourier transform of the music signal is approximated using non-negative matrix factorization while the singer signal is modeled by a combination of a Gaussian mixture and a source/filter model. The separation yields an estimated singer voice signal $v(t)$ along with the music signal $m(t)$. Both components are recombined to

$$x(t) = m(t) + \alpha \cdot v(t), \tag{1}$$

where $\alpha = 0.15$ was determined empirically. Since the separation algorithm is computationally hard it is an optional pre-processing step.

2.3 Tuning Frequency Estimation

For algorithms using chromagram-based features to be efficient it is important to estimate the offset of the tuning frequency relative to $A_4 = 440\,\mathrm{Hz}$. Otherwise, in the worst case when all the instruments are detuned by 50 cents, all the energy is placed exactly in the middle between adjacent frequency bins. To estimate the tuning frequency a constant Q transform of $x(t)$ is computed starting from note *Eb2* up to *F#8* with 36 bins per octave and a hop size of 1024 samples. The absolute value of this transform is interpolated to cent resolution using cubic splines. Afterwards peak picking is applied. The tuning offset in cents is chosen so that the number of peaks caught by the regularly spaced half-tone grid is maximized.

2.4 Feature Extraction

The basis of the observation features is another constant Q transform of $x(t)$ from note *E2* up to *E6*, in which the minimum frequency is adapted using the afore estimated tuning offset. The hop size is again 1024 samples. A chromagram is computed by summing all frequency bins representing the same pitch class resulting in a 12-dimensional vector. The chromagram is used to compute the tonal centroid feature vector as described in [4]. Both the chromagram and the tonal centroid are saved for further processing.

2.5 HMM Parameters

The actual training phase begins when the features for all files in the training set have been computed. The HMM parameter set consists of the prior probabilities π, the transition probabilities A and those parameters describing the distribution of the observation vectors. In our system this distribution is modeled by a single multi-variate Gaussian for each state. It is determined by its mean vector μ and its (full) covariance matrix Σ. Since the key of each piece is not part of the ground-truth annotation we cannot train key-specific models. We have a single global model instead which does not take into account key-specific priors and transition probabilities. Therefore, π is initialized to

$$\pi(k) = 1/24, \quad k \in [1, 24]. \tag{2}$$

The transition probabilities are initialized based on the ground-truth annotation. It is sampled corresponding to the feature hop size. Since we cannot assume that all chord classes are distributed equally among the test set, only two generic classes are trained, a major and a minor chord class. For each frame the relative distance, the interval, to its successor is counted. This results in a histogram of all 24 possible transitions for a major as well as a minor chord. This histogram is converted to quasi probabilities by dividing by the number of frames. These probabilities are circularly shifted to form A. Similarly, the chromagrams of all frames are shifted to normalize all chords relative to *C* major and minor respectively. Both the major and the minor chord chromagram frames are now circularly shifted back to all of the 12 pitch classes. The mean vector and the covariance matrix of the tonal centroid, computed given the circularly shifted chromagram frames, determine μ and Σ for each pitch class. The HMM parameters are saved to be used during the subsequent testing state.

3 TESTING

For each file of the test set the tonal centroid features are extracted as described before. The HMM parameters computed during the training stage are loaded. Viterbi decoding is used to determine the most likely sequence of states which is transformed to match the task's submission format and saved.

4 IMPLEMENTATION

This submission is coded in MATLAB. It includes Kevin Murphy's HMM toolbox [2] and uses the `decimate` function from the Signal Processing Toolbox to downsample the signal. On a single core of a 'Dual Core AMD Opteron(tm) Processor 275' (2210.209 MHz) the system is about ten times faster than real-time when the separation step is skipped. With the separation active it is about three times slower than real-time. For n-fold cross-validation the features are saved for consecutive runs. Since the feature extraction accounts for most of the computational cost the classification is significantly faster once the needed features have been computed.

5 REFERENCES

[1] Bello, J. P. and Pickens, J. "A robust mid-level representation for harmonic content in music signals", ISMIR05, London, UK, 2005.

[2] Durrieu, J.-L., Richard, G., and David, B. "Singer melody extraction in polyphonic signals using source separation methods", *ICASSP 2008*, Las Vegas, USA, 2008.

[3] Durrieu, J.-L. and Weil, J. "Automatic beat-synchronous generation of music lead sheets", *Proc. of 2nd K-Space PhD Jamboree*. Paris, France, 2008.

[4] Harte, C., Sandler, M., and Gasser, M. "Detecting harmonic change in musical audio", *AMCMM'06*, Santa Barbara, USA, 2006.

[5] Papadopoulos, H. and Peeters, G. "Large-scale study of chord estimation algorithms based on chroma representation and hmm", *CBMI 2007*, Bordeaux, France, 2007.

[2] `http://www.cs.ubc.ca/~murphyk/Software/HMM/hmm.html`