JOINT OPTIMIZATION OF AN HIDDEN MARKOV MODEL - NEURAL NETWORK HYBRID FOR CHORD ESTIMATION

Nikolaas Steenbergen
University of Amsterdam
NFWI

John Ashley Burgoyne
University of Amsterdam
ILLC

ABSTRACT

In the following we present a system for chord estimation, based on a combination of a neural network and an Hidden Markov Model. The approach is based on Bengio et al.'s [1] system for automated speech recognition and modified for musical chord estimation. The system consists of a neural network with softmax activation, that is trained to approximate Pitch Class Profiles from a constant Q transform. An Hidden Markov Model is used to classify the chords. Both are trained separately at first and then are jointly optimized.

1. INTRODUCTION

Chord estimation describes the process of extracting musical chord labels from (wave form) encoded Music Pieces. Hereby the specific chord and temporal position and duration have to be automatically determined. Recent approaches include rule based recognition (e.g. [3]) or machine learning. A common approach for extraction of features for machine learning based approaches are so called pitch class profiles. In which the wave form signal is transformed to the Fourier space, and the resulting frequency distribution is aggregated according to the twelve base tones of western tonality music (i.e. c,c#,d, etc.), where octave multiples (frequency multiples) are aggregated to the same "base-tone" bin. These resulting pitch class profiles are then used as basis for classification.
There are different approaches for classification of chords according to extracted features. Matthias Mauch uses dynamic Bayesian Networks [4] with extended pitch class profile features (for base notes of the chords). A common approach for classification are variations of Hidden Markov Models (e.g. [5, 8]). Or combination of classifiers and rule based algorithms [2]. A different approach is described in [6], in which an artificial neural net based on pitch class profiles described above is used for chord recognition (on a train and test set of only individual instruments with individual chords, not complete mixed songs with several instruments and several chords).

A similar domain is automatic speech recognition. Similar to Chord estimation a wave form audio signal is to be analyzed and classified into different entities (e.g. words). As in chord recognition a temporal dependency of phones (atomic acoustic utterances) forming a word, and/or words forming a sentence is a characteristic of the problem. In [1] Bengio, De Mori, Flammia and Kompe propose a system of a combination of Neural Networks (for feature extraction as basis for later classification) and an continuous density HMM for the final classification to incorporate time dependency. Both are trained separately at first, thereafter the authors describe a method of joint optimization through gradient descent according to a global optimization criterion (maximum likelihood).
In the following a proposition for a combined Neural Network / Continuous density Hidden Markov Model as described in [1] applied to musical chord estimation is given: We first give an overview of the system's components in section 2, we then describe how to compute the optimization criteria and computation of the gradient for the HMM in section 3, section 4 describes how to update the neural network according to the computed gradient and section 5 describes how the HMM can be updated, hereafter the specific implementation is briefly described in section 6.

2. BASIC SYSTEM OUTLINE

Our system consists of two main components:

1. A continuous density HMM which estimates time the temporal correlation of chord progressions and performs the final classification.

2. A neural network with softmax activation, which is trained to approximate the computation of normalized Pitch Class Profiles from a constant Q transform, which will be computed in a preprocessing step.

Both are trained separately at first, the neural network according to precomputed training data and the HMM on basis of of the neural network output and ground truth chord data.

After this a joint optimization is performed, based on the gradient of the HMM according to a global optimization criteria (maximum likelihood), the neural network's weights are adjusted. The emission probabilities are updated on basis of the new neural network output, until the system does not improve further.

3. GRADIENT OF THE HIDDEN MARKOV MODEL

We define the emission probability b_t of the HMM as follows:

$$b_t = P(Y_t|S_t) \tag{1}$$

the probability of emitting the neural network output Y_t in state S_t at time t according to our state sequence determined by the training data.

The joint probability of state and observation sequence is defined as:

$$\pi_1 b_1 \prod_{t=2}^{T} b_t a_{t-1,t} \tag{2}$$

with π_1 being the initial state probability, b_t the probability of emission as stated in equation 1, and $a_{t-1,t}$ the transition probability from state S_{t-1} to S_t.

We want to maximize the log likelihood of the model our optimization criteria:

$$C = log(\pi_1 b_1 \prod_{t=2}^{T} b_t a_{t-1,t}) \tag{3}$$

Similar to Bengio et al. in [1].

Since the transition probabilities are fixed by the provided ground truth, we take the partial derivative in respect to b_t leaving us thus with:

$$\frac{\partial C}{\partial b_t} = \frac{\partial log(\pi_1 b_1 \prod_{t=2}^{T} b_t a_{t-1,t})}{\partial b_t} \tag{4}$$

We rewrite the logarithm of the product as a sum of logarithms. Since the derivative in respect to b_t does not affect the initial state probability distribution, transition probabilities or emission probabilities of the other states, these are dropped, leaving us with:

$$\frac{\partial C}{\partial b_t} = \frac{\partial log(b_t)}{\partial b_t} = \frac{1}{b_t} \tag{5}$$

Since we are using a Continuous densities HMM, the emission probability b_t can be represented as a mixture of Gaussians as described in [1]:

$$b_{i,t} = \sum_k \frac{Z_k}{\sqrt{(2\pi)^n |\Sigma_k|}} exp(-\frac{1}{2}(Y_t - \mu_k)\Sigma_k^{-1}(Y_t - \mu_k)^T) \tag{6}$$

where n is the number of Gaussian components per state of the HMM, Z_k, μ_k and Σ_k the gain, mean and Covariance matrix of Gaussian component k respectively.

4. ADJUSTING NEURAL NETWORK PARAMETER

Since we are aiming to change the neural network parameters according to the HMM optimization gradient, we need to adjust the Neural network parameters as described in bengio et al in [1].

Using the chain rule we take partial derivative of the optimization criterion C in respect to the neural network output $Y_{j,t}$ for the j^{th} component of the output at time t:

$$\frac{\partial C}{\partial Y_{j,t}} = \frac{\partial C}{\partial b_{i,t}} \frac{\partial b_{i,t}}{\partial Y_{j,t}} \tag{7}$$

Where $\frac{\partial b_{i,t}}{\partial Y_{j,t}}$ by differentiating equation 6, $\frac{\partial b_{i,t}}{\partial Y_{j,t}}$ can be written as follows:

$$\frac{\partial b_t}{\partial Y_{jt}} = \sum_k \frac{Z_k}{\sqrt{2\pi|\Sigma_k|}} (\sum_l d_{k,lj}(\mu_{kl} - Y_{lt})) \exp(-\frac{1}{2}(Y_t - \mu_k)\Sigma_k^{-1}(Y_t - \mu_k)^T) \tag{8}$$

5. UPDATING HMM PARAMETERS

Rabbiner et al in [7] provides us with methods to update continuous densities HMMs we can update the gain Z_{jk} for state j and component k as follows:

$$Z'_{jk} = \frac{\sum_{t=1}^{T} \gamma_t j, k}{\sum_{t=1}^{T} \sum_{k=1}^{M} \gamma_t(j,k)} \tag{9}$$

The mean μ_{jk} for state j and component k can be computed with:

$$\mu'_{jk} = \frac{\sum_{t=1}^{T} \gamma_t(j,k) O_t}{\sum_{t=1}^{T} \gamma_t(j,k)} \tag{10}$$

where O_t the observation, specific neural network output at time t.

The Covariance Σ_{jk} for state j and component k can be computed with:

$$\Sigma'_{jk} = \frac{\sum_{t=1}^{T} \gamma_t(j,k)(O_t - \mu_{jk}(O_t - \mu_{jk})^T}{\sum_{t=1}^{T} \gamma_t(j,k)} \tag{11}$$

$\gamma_t(j,k)$ describes the probability of being in state j at time t with the kth Gaussian mixture component:

$$\gamma_t(j,k) = \delta_{tj} \frac{Z_{jk}\mathcal{N}(O_t, \mu_{jk}, \Sigma_{jk})}{\sum_m Z_{jm}\mathcal{N}(O_t, \mu_{jm}, \Sigma_{jm})} \tag{12}$$

where the term δ_{tj} is 1 if j is equal to the state in our ground truth data and 0 otherwise.

6. IMPLEMENTATION

6.1 Data Preprocessing

We first take the Fourier transform of the 44100 Hz with a window size of 8192 samples with 2048 samples overlap. Thereafter the frequencies are further processed with a constant Q transform with 36 bins per octave over 6 octaves, ranging from approx 32.7 Hz (midi note 24) to approx. 2093 Hz (midi note 69. To take into account minor

pitch shifts, and further reduce the input space, we choose a frame wise maximum of the respective bins to decide which of the three CQT bin components to use for the aggregation of a normalized Pitch Class Profile for this frame.

6.2 Neural Network

The neural network is trained to approximate the normalized Pitch Class Profiles from the constant Q transform values described the above (we only supply the constant Q transform bins that are taken into account after the maximization for minor pitch shifts).

Since the Pitch Class Profile is normalized, we use a softmax activation function for the output of the neural network. The other nodes have a sigmoidal activation function.

The neural network contains 100 hidden nodes.

6.3 Hidden Markov Model

In the current system we try to estimate only major, minor and none chords, thus leaving us with 25 possible chords, one for each root node and chord type. These are modeled as states in an ergodic Hidden Markov Model. Since major and minor chords are determined by three musical notes, the emission probabilities of each state in the HMM are modeled by a mixture of three Gaussian. The HMM in turn is trained on the output of the pretrained neural network.

6.4 Combined training

For the joint optimization of the neural network and the HMM we iteratively adjust the neural network weights according to the HMM gradient for the global optimization criterion (as described above). After the neural network weights are adjusted, we updated the HMM with the methods described in section 5 Every alternating neural network weight adjustment and HMM update a test is performed and the training is completed when the change in performance of the system falls below a prior specified threshold.

7. REFERENCES

[1] Yoshua Bengio, Renato De Mori, Giovanni Flammia, and Ralf Kompe. Global optimization of a neural network-hidden markov model hybrid. *Neural Networks, IEEE Transactions on*, 3(2):252–259, 1992.

[2] W Bas De Haas, José Pedro Magalhães, and Frans Wiering. Improving audio chord transcription by exploiting harmonic and metric knowledge. In *Proceedings of the 13th International Society for Music Information Retrieval Conference (ISMIR)*, pages 295–300, 2012.

[3] Nikolay Glazyrin. Audio chord estimation using chroma reduced spectrogram and self-similarity. *Proceedings of the Music Information Retrieval Evaluation Exchange (MIREX)*, 2012.

[4] Matthias Mauch. Automatic chord transcription from audio using computational models of musical context. 2010.

[5] Yizhao Ni, Matt Mcvicar, Raul Santos-Rodriguez, and Tijl De Bie. Harmony progression analyzer for mirex 2011. 2011.

[6] Julien Osmalsky, Jean-Jacques Embrechts, Marc Van Droogenbroeck, and Sébastien Pierard. Neural networks for musical chords recognition. In *Journées d'informatique musicale*, 2012.

[7] Lawrence R Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

[8] Yushi Ueda, Yuuki Uchiyama, Takuya Nishimoto, Nobutaka Ono, and Shigeki Sagayama. Hmm-based approach for automatic chord detection using refined acoustic features. In *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*, pages 5518–5521. IEEE, 2010.