Multipitch Tracking Using A Factorial Hidden Markov Model

Michael Wohlmayr, Franz Pernkopf

Signal Processing and Speech Communication Laboratory
Graz University of Technology, Graz, Austria
michael.wohlmayr@tugraz.at, pernkopf@tugraz.at

Abstract

In this paper, we present an approach to track the pitch of two simultaneous speakers. Using a well-known feature extraction method based on the correlogram, we track the resulting data using a factorial hidden Markov model (FHMM). In contrast to the recently developed multipitch determination algorithm [1], which is based on a HMM, we can accurately associate estimated pitch points with their corresponding source speakers. We evaluate our approach on the “Mocha-TIMIT” database [2] of speech utterances mixed at 0dB, and compare the results to the multipitch determination algorithm [1] used as a baseline. Experiments show that our FHMM tracker yields good performance for both pitch estimation and correct speaker assignment.

Index Terms: Factorial hidden Markov model (FHMM), pitch estimation, multipitch tracking, graphical model

1. Introduction

Estimation and tracking of pitch is important for many algorithms and applications in speech and audio signal processing, e.g. one-channel blind source separation [3]. While well performing algorithms do exist for the case of a single speaker in a clean recording [4], the same task of pitch estimation is more difficult for noisy speech and multiple speakers talking simultaneously.

In [1], an approach for robust multipitch tracking has been proposed. It is based on the unitary model of pitch perception [5], upon which several improvements are introduced to yield a probabilistic representation of the periodicities in the signal. Semi-continuous pitch trajectories are then obtained by tracking these likelihoods using a hidden Markov model (HMM). Although this model provides an excellent performance in terms of accuracy, it is not possible to correctly link each pitch estimate to its source speaker.

We show that factorial hidden Markov models (FHMM) provide a natural framework to track the pitch of multiple speakers. FHMMs have been introduced in [6] and constitute a generalization to HMMs. They provide a probabilistic framework to track the states of multiple Markov processes evolving in parallel over time, where the available observations are considered as a joint effect of all single Markov processes. Given these observations only, the task of inferring the most likely state sequence of each hidden Markov chain is more complex than for the HMM case.

Although the junction tree algorithm [7] provides exact solutions to this problem, its computational complexity increases exponentially with the number of hidden Markov chains. Various methods to approximate the solutions for the sake of reduced complexity exist and can often be formulated as message passing algorithms [8]. The Sum-Product Algorithm [9] is exact on graphical models with tree structure, and approximates solutions for graphs with cycles (loopy Sum-Product Algorithm).

In this paper, we propose a FHMM for tracking two simultaneous users using the feature extraction method of [1]. We apply the Junction tree algorithm and the loopy Sum-Product algorithm to estimate the pitch track. In contrast to [1], the usage of a FHMM allows to link pitch estimates correctly to their respective source speakers.

In section 2, we review the main steps of feature extraction proposed in [1]. Section 3 gives a short introduction to FHMMs and presents its application for multipitch estimation. We shortly overview message passing algorithms for tracking. In section 4, the experimental setup is discussed and the performance of the system is compared to the approach in [1]. Finally, section 5 concludes the paper.

2. Feature Extraction

To calculate the conditional observation probabilities used in the FHMM for tracking, we employ the same method as proposed in [1]. In this section, we review some of the main aspects only. For details, we refer the reader to the original paper. The input signal is decomposed into 128 subbands using a gammatone filterbank with center frequencies uniformly spaced along logarithmic frequency. For high frequency channels (center frequency above 800Hz), the amplitude envelope is extracted using the Teager energy operator followed by a lowpass filter with cutoff frequency at 800Hz. The normalized autocorrelation is then computed on frames for every channel with frame-length 16ms and step-size 10ms.

Up to this stage, these processing blocks resemble the method to calculate the correlogram [10]. Summation of the periodicity information across all channels would result in the ‘summary autocorrelation’. In contrast, [1] employs a scheme to discard channels whose periodicity information is likely to be unreliable due to noise. For selecting a low frequency channel, the maximum peak at nonzero lag must exceed a certain threshold. For high frequency channels, the periodicity information must be consistent with the autocorrelation computed on a long time frame of 30ms. If a high frequency channel is selected, an additional peak selection routine is employed. A peak is only selected if a second peak at double time lag exists. Further, if the peak with smallest nonzero lag exceeds a threshold, all peaks at a multiple lag will not be selected.

The final set of peaks selected from various channels serves as a basis to create a probabilistic representation of zero, one or two pitch periodicities at each time frame. Let $\Phi^t$ denote the set of peak locations extracted from various channels at time frame $t$. In brief, the method in [1] calculates the likelihood of pitch periodicities under the observation $\Phi^t$ for the hypothesis of one and two pitches. We denote these quantities by $p(\Phi^t|d_1)$ and $p(\Phi^t|d_1, d_2)$, respectively, where $d_1$ and $d_2$ indicate the corresponding pitch periodicities.
3. Tracking with FHMM

A factorial HMM is a graphical model that employs several Markov chains in parallel. For simplicity, we present the case of two Markov chains, which is shown in Figure 1. The hidden random variables are denoted by $x_k^{(i)}$, where $k$ indicates the Markov chain and $t$ the time index from 1 to $T$. Similarly, the observed random variables are denoted by $y^{(i)}$. Each hidden node represents a discrete random variable, while the observed nodes can be either discrete or continuous. For simplicity, all hidden variables are assumed to have cardinality $|X|$. The arcs between nodes indicate a direct conditional dependency between two random variables. Specifically, the dependency of hidden variables between two consecutive time instances is defined for each Markov chain by the transition probability $p(x_k^{(i)}|x_k^{(i-1)})$. The dependency of the observable variables on the hidden variables of the same time frame are defined by the observation probability $p(y^{(i)}|x_k^{(i)}_1,x_k^{(i)}_2)$. Finally, the prior distribution of the hidden variables in every chain is denoted by $p(x_k^{(1)})$. Denoting the whole sequence of variables in boldface, i.e. $\mathbf{x} = \bigcup_{i=1}^{T} \{x_1^{(i)}, x_2^{(i)}\}$ and $\mathbf{y} = \bigcup_{i=1}^{T} \{y^{(i)}\}$, the joint distribution of all variables is given by

$$p(\mathbf{x}, \mathbf{y}) = p(\mathbf{y}|\mathbf{x})p(\mathbf{x}),$$

$$p(\mathbf{x}) = \prod_{k=1}^{2} p(x_k^{(1)}) \prod_{t=2}^{T} p(x_k^{(t)}|x_k^{(t-1)}),$$

$$p(\mathbf{y}|\mathbf{x}) = \prod_{t=1}^{T} p(y^{(t)}|x_1^{(t)}, x_2^{(t)}).$$

The number of possible hidden states per time frame is $|X|^2$. As pointed out in [6], this could also be accomplished by an ordinary HMM. The main difference, however, is the constraint placed upon the transition structure. While a HMM with $|X|^2$ states would allow any $|X|^2 \times |X|^2$ transition matrix between two hidden states, the FHMM is restricted to two $|X| \times |X|$ transition matrices. It is reasonable to assume that this modelling approach fits well to the problem of multipitch tracking, where the observable signal is generated from the superposition of several distinct sources, each with their own dynamics.

### 3.1. Parameter settings

In this work, we use two Markov chains, one for each speaker. Each hidden variable has $|X| = 200$ states, where state value ‘1’ refers to ‘no pitch’, and state values ‘2’–‘200’ encode correlation lags corresponding to different pitch periodicities ranging from less than 1ms to 12.5ms. Accordingly, each $200 \times 200$ likelihood matrix $\Psi_{ij} = p(y^{(t)}|x_k^{(i)}, x_2^{(k)})$ is constructed from the quantities $p(\Phi^{(1)}|d_1)$ and $p(\Phi^{(2)}|d_1, d_2)$, which were obtained during feature extraction (see section 2). Specifically, the elements of $\Psi_{ij}$ are set to

$$\ln \Psi_{ij} = \ln \left( \frac{\gamma_i^1 - \max \{\gamma_i^1 + \alpha \}}{\ln \frac{\beta_j}{\beta_i}} \right) + \beta \right)$$

$$i = 1, j = 1, i = 2, ... , 200$$

$$\ln \left( \frac{\min \left( \gamma_i^2, ... , \gamma_i^j \right)}{\beta_i \beta_j} \right)$$

$$i = 1, j = 2, ... , 200$$

where $\gamma_i^k = -\ln \sum_{d_k} p(\Phi^{d_k}|d_1)$ and parameters $\alpha$ and $\beta$ are set to ensure proper scaling.

Both transition matrices $\Omega^{(k)} = p(x_k^{(1)}|x_k^{(t-1)})$ are obtained from training data. After initialization to an identity matrix, element $\Omega_{ij}^{(k)}$ is increased by one for each occurrence of a transition from state $j$ to state $i$ at speaker $k$. Finally, both matrices are normalized such that each column sums to one ($\sum_i \Omega_{ij}^{(k)} = 1$). Both prior distributions are set to $p(x_k^{(1)} = i) = \sum_j \Omega_{ij}^{(k)}$.

### 3.2. Tracking

The task of tracking involves searching the sequence of hidden states $\mathbf{x}^*$ that maximizes the conditional distribution $p(\mathbf{x}|\mathbf{y})$.

$$\mathbf{x}^* = \arg \max_{\mathbf{x}} p(\mathbf{x}|\mathbf{y})$$

For HMMs, the exact solution to this problem is found by the Viterbi algorithm. For FHMMs, an exact solution can be found using the Junction tree algorithm [7], however this approach gets intractable for increasing $K$ and $|X|$. Several algorithms for approximate solutions are derived in [6] from the framework of variational inference. The Sum-Product algorithm [9] can be derived under a similar setting of variational principles [8], although more intuitive derivations exist for graphs without loops. When applied on a graph with loops, as is the case for a FHMM, the solutions are in general not guaranteed to converge and can only approximate the optimal solution.

In this work, we explore both the Junction Tree (JT) algorithm and the Max-Sum (MS) algorithm (a variant of Sum-Product algorithm) to obtain solutions for equation 5. We will refer to these approaches as FHMM-JT and FHMM-MS, respectively.

As the FHMM used here has only two Markov chains, the Junction Tree algorithm is still computationally tractable. This allows to compare approximate solutions to exact ones.

We give a short review on the message passing algorithms used. For a detailed discussion, we refer the reader to [9, 7, 8]. Both
Figure 3: Multipitch tracking results for one test example. The first row shows the spectrogram and both reference pitch trajectories of a test example, where a male and a female speaker are mixed at 0dB. The rows beneath display tracking results of our FHMM approach and the baseline algorithm.

the Max-Sum and the Junction Tree algorithm are based on passing messages between nodes of a graph. Among various types of graphs, factor graphs [9] have become a popular tool to depict the mechanisms of message passing. Figure 2 shows a FHMM as factor graph, where the functional dependency of each variable node, for brevity called $x$, is made explicit by “factor nodes”, shown as shaded rectangles, i.e., each rectangle denotes a function $f(\hat{x})$ of its neighbouring variable nodes $\hat{x}$. For the Max-Sum algorithm, each node sends to every neighbour a vector valued message $\mu$, which is itself a function of the messages it received, (as well as $f(\hat{x})$, for the case of a factor node). A message from variable node $x$ to factor node $f$ is

$$\mu_{x \rightarrow f}(x) = \sum_{g \in n(x) \setminus \{f\}} \mu_{g \rightarrow x}(x),$$

while a message from factor $f$ to variable $x$ is

$$\mu_{f \rightarrow x}(x) = \max_{x \in \mathcal{X}} \left( \ln f(x) + \sum_{y \in n(f) \setminus \{x\}} \mu_{y \rightarrow f}(y) \right).$$

Here, $n(x)$ denotes the neighbour nodes of $x$. We normalize each message and restrict each node to send a maximum of 15 messages per link. Further, each node only re-sends a message to a neighbour if it is significantly different from the previously sent message in terms of the $L_2$ norm.

Although the set of maxima $x^* = \arg \max_{x} p^*(x) \forall x \in \mathcal{X}$ does not necessarily yield the global maximum $x^{\ast}$, as multiple global maxima might be present, a backtracking stage may lead to inconsistencies due to the loops in the factor graph. For this reason, we simply set the global maximum $x^{\ast}$ to the set of individual maxima $x^*$. For the Junction tree algorithm, the factor graph from figure 2 is turned into a single chain factor graph where each variable node is a union of two variables (i.e., $x_1(t), x_2(t)$, $x_1(t), x_2(t+1)$, $x_1(t+1), x_2(t+1)$, ...). Then, the message passing mechanism from equation 6 and 7 is applied. This results in an increased computational complexity due to the higher dimensionality of each node, however only a single pass from left to right is needed. As in the Viterbi algorithm, the global maximum is then obtained by backtracking [7].

4. Experimental Results

The performance of FHMM-JT and FHMM-MS was evaluated and compared with the HMM baseline system proposed in [1]. Again, we emphasize that the feature extraction scheme is the same for both our approach and the baseline. The main difference is the tracking method. While the baseline system uses a single HMM, our approach is based on a FHMM with one Markov chain per speaker. Experiments have been performed using the “Mocha-TIMIT” database [2]. It consists of 460 English utterances from both a male and a female speaker, sampled at 16kHz. In addition, laryngograph signals are available for all recordings, from which the pitch ground truth $f_0[t]$ was acquired using the ESPS “get_f0” method [4] together with manual removal of erroneous pitch estimates in nonaudible regions. A training and test set
of 400 and 60 sentences, respectively, was assembled, were each training and test instance was obtained by mixing a random male and female utterance at 0dB. The baseline system does not require any training and is used as provided\(^\dagger\). Training of our proposed system involves estimation of the transition and prior distributions, as well as of parameters \(\alpha\) and \(\beta\), which are set to \(-131\) and 16, respectively.

The baseline system, as well as FHMM-JT and FHMM-MS, are then applied on the test set. For every test instance, each method estimates two pitch trajectories, \(f_0^{(1)}[t]\) and \(f_0^{(2)}[t]\). For performance evaluation, each of the two estimated pitch trajectories needs to be assigned to its ground truth trajectory, \(f_0^{(1)}[t]\) or \(f_0^{(2)}[t]\). From the two possible assignments, \([f_0^{(1)} \rightarrow f_0^{(1)}, \hat{f}_0^{(2)} \rightarrow \hat{f}_0^{(2)}]\) or \([\hat{f}_0^{(1)} \rightarrow \hat{f}_0^{(2)}, f_0^{(2)} \rightarrow f_0^{(1)}]\)

the one is chosen for which the overall quadratic error is smaller. To evaluate the resulting estimates, we use the error measure proposed in [1]: \(E_{ij}\) denotes the percentage of time frames were \(i\) pitch points are misclassified as \(j\) pitch points, i.e., \(E_{ij}\) means the percentage of frames with \(i\) pitches estimated whereas only one pitch is present. The pitch frequency deviation is defined as

\[
\Delta f[k][t] = \frac{|\tilde{f}_0^{(k)}[t] - f_0[t]|}{f_0[t]},
\]

\(\tilde{f}_0[t]\) denotes the reference chosen for \(f_0^{(k)}[t]\). The gross detection error rate \(E_{Gross}\) is the percentage of time frames where the algorithm correctly detects the presence of one pitch (two pitches), but the corresponding frequency deviation \(\Delta f[k][t]\) (either of \(\Delta f^{(1)}[t]\) or \(\Delta f^{(2)}[t]\)) is larger than 20%. The fine detection error \(E_{Fine}\) is the average frequency deviation in percent at time frames where \(\Delta f[k][t]\) is smaller than 20%. The overall error, \(E_{Total}\), is defined as the sum of all error terms:

\[
E_{Total} = E_{01} + E_{02} + E_{10} + E_{12} + E_{20} + E_{21} + E_{Gross} + E_{Fine}.
\]

Table 1 shows the error measure of all methods on the test set. FHMM-JT achieves the best overall performance. The baseline system performs comparably in terms of the \(E_{ij}\) measures, however, it has a worse gross error rate. In contrast to FHMM-JT, the estimates from the baseline are not able to give speaker consistent trajectories. This is demonstrated in figure 3, where the results from tracking a test example are displayed. As shown in the bottom tracking result, the points of each estimated pitch trajectory provided by the baseline swap between the true trajectories, such that it is not possible to link estimated points correctly to their source speaker. This is also partially true for the results of FHMM-MS, as the loopy Max-Sum algorithm is only

\(\dagger\)C-Code available at http://www.cse.ohio-state.edu/ dwang/pnl/software.html

5. Conclusions

We have investigated the performance of a FHMM for multipitch tracking. In comparison to the baseline system using a single HMM, a FHMM models the pitch of every speaker in a separate Markov chain. This allows to identify the source speaker of every estimated pitch point. In contrast to the baseline, however, this approach relies on a training stage. Especially, it is necessary to obtain reliable estimates of the transition probabilities of every Markov chain. Experiments have shown that our FHMM tracker yields a good performance in terms of the accuracy of the estimated pitch trajectories as well as the correct assignment of all estimates to their corresponding speakers.

6. Acknowledgements

This work was funded by the Austrian Science Fund under the project number P19737-N15.

7. References


