Unsupervised HMM classification of F0 curves

Damien Lolive, Nelly Barbot, Olivier Boeffard

IRISA / University of Rennes 1 - ENSSAT
6 rue de Kerampont, B.P. 80518, F-22305 Lannion Cedex
France
{damien.lolive,nelly.barbot,olivier.boeffard}@irisa.fr
http://www.irisa.fr/cordial

Abstract

This article describes a new unsupervised methodology to learn F0 classes using HMM models on a syllable basis. A F0 class is represented by a HMM with three emitting states. The clustering algorithm relies on an iterative gaussian splitting and EM retraining process. First, a single class is learnt on a training corpus (8000 syllables) and it is then divided by perturbing gaussian means of successive levels. At each step, the mean RMS error is evaluated on a validation corpus (3000 syllables). The algorithm stops automatically when the error becomes stable or increases. The syllabic structure of a sentence is the reference level we have taken for F0 modelling even if the methodology can be applied to other structures. Clustering quality is evaluated in terms of cross-validation using a mean of RMS errors between F0 contours on a test corpus and the estimated HMM trajectories. The results show a pretty good quality of the classes (mean RMS error around 1Hz).

Index Terms: prosody, fundamental frequency, unsupervised classification, Hidden Markov Model

1. Introduction

Technologies linked to speech processing widely use intonational speech models. We can particularly cite Text-to-Speech Synthesis (TTS) or a more emerging field as Voice Transformation. A TTS system needs prosodic models in order to create intelligible speech from text and elocution style. Most of works on this subject rely on a strong expertise in phonology and acoustic phonetic. A great challenge for a TTS system would be to offer a wide variety of prosodic models so as to diversify voice catalogs.

Nowadays, the majority of voice transformation systems use global prosodic adjustment (elocution rate and melody).[1]. An important issue would be to transform prosodic models between source and target speakers, notably of melodic contours. In order to easily adapt these models from various speakers and to limit manual expertise, an unsupervised methodology is necessary.

Although intonation is a combination of numerous linguistic factors, this article focuses on the acoustic parameter recognized to be the most prominent suprasegmental factor, the fundamental frequency or F0. F0 contours, extracted from the speech signal, represent the vibration of the vocal folds over time. A wide range of publications have reported on efforts in modelling F0 evolution. We can particularly cite MoMel [2], Tilt [3], B-spline models [4], as well as Sakai et Glass’s work [5] which use regular spline functions. Such stylizations offer a direct or parametric description of the F0. A consequent literature deals with the fundamental frequency prediction problem from linguistic information [6]. This kind of modelling is supervised insofar as a segmentation in prosodic units is imposed and associated to F0 curves.

As for the melodic contour classification issue, few works deal with an unsupervised F0 clustering. The problem is to derive a set of basic melodic patterns from a set of sentences from which F0 has been previously computed. The idea is that concatenation of elementary F0 contours can characterize a complete melodic sentence [7]. We assume that an atomic element of the melodic space is linked to the syllable. Thus, the objective is to learn a coherent set of melodic contour classes at the syllabic level. The major difficulty is to take into account the syllable duration. Two melodic contours with different temporal supports can represent the same elementary melodic pattern. Consequently, we choose to use Hidden Markov Models (HMM) which intrinsically integrate the elasticity of the representation support of an elementary form.

In this article, an unsupervised classification methodology for melodic contours is described. This methodology is based on the use of HMM models used in an unsupervised mode. The increase of the number of classes is realized using a variant of gaussian splitting on a HMM set.

The HMM model structure and the procedure carried out to split a class are introduced in section 2. In section 3, the unsupervised learning algorithm applied to determine a set of melodic contour classes is described. The experimental methodology is then presented in section 4, as well as the evaluation method of class quality. The results are discussed in section 5.

2. Unsupervised HMM modelling

2.1. The model

In this article, we are interested in finding out a partition of a set of syllable melodic contours thanks to HMM models. In our approach, a HMM characterizes a class and models F0 contours which are monodimensional signals. Figure 1 shows the topology of the HMM used. Their construction is based on syllable structure. Indeed, linguistics teaches us that a syllable can be divided into three parts: onset, kernel and coda. This structure leads us to consider a model with three emitting states. Moreover, as onset and coda are optional, the state transition graph includes jumps which allow to avoid the first and last emitting states.

A HMM Mi is composed of five states and does not have any backward state transition. States q0 and q4 are respectively the start and end nodes of the HMM. These two states are
non-emitting and have a null sojourn time. As for the states \( q_{ij} \), for \( i \) from 1 to 3, their output values are distributed according to a gaussian law with mean \( \mu_{ij} \) and variance \( \sigma_{ij}^2 \).

\[
\mathcal{N}(\mu_{ij}, \sigma_{ij}^2) \quad \mathcal{N}(\mu_{2j}, \sigma_{2j}^2) \quad \mathcal{N}(\mu_{3j}, \sigma_{3j}^2)
\]

Figure 1: Structure of HMM \( M_j \).

For a contour class \( M_j \), the associated HMM parameters are trained using a standard Baum-Welch algorithm. Melodic contours are labeled thanks to the Viterbi algorithm that provides the global mean RMS error is computed on the validation corpus.

\[
\text{RMS} = \frac{1}{d} \sum_{t=1}^{d} (F_0(x_t) - \mu_{T_{ij}})^2
\]

\[d\]

\[e_{\text{prev}}\]

\[\epsilon = 1 \times 10^{-4};\]

\[\text{converged} = \text{false};\]

\[5 \text{ repeat}\]

\[6 \text{ foreach HMM model } M_i \in M \text{ do}\]

\[7 \text{ - learn } M_i \text{ using the Baum-Welch algorithm on the training corpus}\]

\[8 \text{ end}\]

\[9 \text{ - re-label all syllables of the validation corpus with the new HMM models } M \text{ (Viterbi)};\]

\[10 \text{ - compute the mean RMS error } e_{\text{cur}} \text{ between each syllable and its HMM class model};\]

\[11 \text{ if } e_{\text{prev}} - e_{\text{cur}} < \epsilon \text{ then}\]

\[12 \text{ converged} = \text{true};\]

\[13 \text{ else}\]

\[14 \text{ - divide } M \text{ into two HMM sets } M_1 \text{ and } M_2 \text{ with } \text{card}(M_1) = NbToSplit;\]

\[15 \text{ - split each HMM of } M_j \text{ into } M_1^{\text{new}};\]

\[16 \text{ - merge } M_1^{\text{new}} \text{ and } M_2 \text{ into a new HMM set } M^{\text{new}};\]

\[17 \text{ - re-label all syllables according to the new HMM set } M^{\text{new}};\]

\[18 M = M^{\text{new}};\]

\[19 e_{\text{prev}} = e_{\text{cur}};\]

\[20 \text{ end}\]

\[21 \text{ until converged} = \text{true} ;\]

Algorithm 1: Unsupervised algorithm used to learn the melodic contour classes.

The algorithm first considers one class to which a HMM is associated. At each step of the algorithm, we split a subset of the existing classes to create new classes. Considering the algorithm has done a certain number of iterations, we then have a HMM set \( M \). After the learning step of the models in \( M \), the global mean RMS error is computed on the validation corpus. For a \( F_0 \) contour of length \( d \), the RMS error calculation is done in the following way:

- We compute the optimal state sequence \((T_i)_i \in \{q_{ij}, q_{j2}, q_{j3}\}\) of the HMM \( M_j \) associated to the syllable using the Viterbi algorithm.
- To each state \( T_i \), we associate the mean value \( \mu_{T_{ij}} \) of the gaussian in the state \( T_i \) of the HMM \( M_j \).
- The RMS error (Root Mean Square error) is then computed between the \( F_0 \) observations and that sequence of mean values:

\[
\text{RMS}^2 = \frac{1}{d} \sum_{t=1}^{d} (F_0(x_t) - \mu_{T_{ij}})^2
\]

The algorithm convergence is then evaluated in function of the mean RMS error on the validation corpus: we consider that the convergence is achieved if the mean RMS increases or is stable. If the algorithm has not converged at this step, we construct the subset \( M' \) constituted by the NbToSplit HMM that have the highest cumulative MSE (Mean Square Error). These HMM are then split each one into two HMM, in order to obtain
4. Methodology

4.1. F0 corpus
Experiments are conducted on a set of syllables randomly extracted from a 7,000 sentence corpus. The acoustic signal was recorded in a professional recording studio; the speaker was asked to read the text. The acoustic signal was annotated and segmented into phonetic units. The fundamental frequency, \( F_0 \), was analyzed in an automatic way according to an estimation process based primarily on the autocorrelation function of the speech signal. Next, an automatic algorithm was applied to the phonetic chain pronounced by the speaker so as to find the underlying syllables. The corpus of the selected syllables is divided into a training corpus (8,000 syllables) and a validation corpus (3,000 syllables).

4.2. Data pre-processing
The first step concerns the conversion of the \( F_0 \) values in cents. The cent, which is the hundredth of a semi-tone, is a unit that makes a parallel with the logarithmic scale of the ear. The conversion from Hertz to cent is given by equation 4, where \( F_0^{\text{cent}} = 110 \text{Hz} \).

\[
F_0^{\text{cent}} = 1200 \times \log_{10} \left( \frac{F_0^{\text{hertz}}}{F_0^{\text{ref}}} \right)
\]  
(4)

The second step is similar to the processing achieved in [10]. It realizes a linear interpolation of unvoiced parts of the \( F_0 \) curves at the sentence level. This interpolation comes from the hypothesis according to which a continuous melodic segment and the fundamental frequency value is then masked during unvoiced parts. Moreover, a linear regression is done on the interpolated \( F_0 \) curves in order to suppress microprosodic variations.

4.3. Experiment
The main goal of this study is to establish unsupervised classes from a speech corpus. Thus, the use of common evaluation methodology in order to evaluate the quality of the classes is impractical.

In our case, we propose to evaluate the overall quality of the clustering in relation to the similarity of the contours grouped according to their shape and independently of their duration. To do that, we use a RMS error calculation between a syllable and the optimal trajectory of the associated HMM. We can obtain a RMS error for an entire class, that we want as small as possible and notably smaller than the common JND threshold for the \( F_0 \) (about 4Hz).

Moreover, to be able to compute the RMS error and compare the results to the JND threshold (for \( F_0 \)), we convert the more accurate classes in terms of cumulative MSE. The number of HMM to split \( NbToSplit \) is a parameter of the algorithm.

Once we have the new set of classes \( M'_{\text{new}} \) coming from the splitting of \( M' \), the Viterbi algorithm is applied to modify the \( F_0 \) contour labels in the training corpus and to make them correspond to the new classes. Thenceforth, we can learn the new HMM on the modified training corpus. The gaussian splitting process is repeated until the algorithm reaches a convergence threshold. During the splitting step, if a HMM does not capture a sufficient number of contours, then the algorithm goes on without splitting it.

4.4. Results and discussion

Figure 2 shows the example of a melodic contour and the trajectory of the HMM associated to its class. We can observe the sequence of the HMM states over time. For this example, the HMM stays in state \( q_1 \) during the first four observations. The gaussian mean that corresponds to this state is approximately 107Hz. In this example, the RMS error between the \( F_0 \) contour and the HMM trajectory is around 1Hz. The analysis of this figure shows that the states of the HMM reflect the general shape of the contour. The time evolution and thus the length of the contour is caught by the loops at the level of each HMM state. Consequently, each HMM reflects a particular form which is independent of duration and enables the modelling of melodic contours of different lengths but of similar shape.

A HMM state models a constant melodic segment and the first derivative could be useful to better follow the evolution of the melodic contour. For practical purposes, this would be realized by the joint use of the \( F_0 \) values and the first derivative values. However, taking into account this problem is relatively complex and leads us to difficulties concerning the estimation of the class quality. Instead of taking into account explicitly the first derivative, we can also increase the number of the states to better model \( F_0 \) inflexions. In this case, the estimation process turns out to be an over-estimated solution considering the high number of parameters.

Mean RMS errors in function of the number of classes are presented in tables 1 and 2. This experiment is carried out with three different \( NbToSplit \) values:

- **Split-1**: \( NbToSplit = 1 \), we divide only one HMM at each iteration.
- **Split-2**: \( NbToSplit = 2 \), two HMM are divided at each iteration.
- **Split-n**: all the HMM are split into two parts at each iteration.

In table 1, we can see that, on the validation corpus, the RMS error decreases while the number of HMM increases for all the three split methods. However, the error does not evolve in the same manner for the three cases. Concerning **split-1** and **split-2**, the number of HMM split at each iteration is small. The
Table 1: Mean RMS error (Hz) with 95% confidence intervals for the three split variants on the validation corpus

<table>
<thead>
<tr>
<th>N. of HMM</th>
<th>Split-1</th>
<th>Split-2</th>
<th>Split-n</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>11.44 ± 0.18</td>
<td>11.44 ± 0.18</td>
<td>11.44 ± 0.18</td>
</tr>
<tr>
<td>2</td>
<td>9.87 ± 0.16</td>
<td>9.87 ± 0.16</td>
<td>9.87 ± 0.16</td>
</tr>
<tr>
<td>4</td>
<td>9.23 ± 0.15</td>
<td>9.30 ± 0.15</td>
<td>9.30 ± 0.15</td>
</tr>
<tr>
<td>8</td>
<td>7.25 ± 0.15</td>
<td>7.87 ± 0.12</td>
<td>8.26 ± 0.14</td>
</tr>
<tr>
<td>16</td>
<td>5.48 ± 0.12</td>
<td>5.79 ± 0.11</td>
<td>6.74 ± 0.13</td>
</tr>
<tr>
<td>32</td>
<td>4.86 ± 0.11</td>
<td>4.82 ± 0.10</td>
<td>5.76 ± 0.12</td>
</tr>
<tr>
<td>64</td>
<td>4.56 ± 0.10</td>
<td>4.54 ± 0.11</td>
<td>5.15 ± 0.11</td>
</tr>
<tr>
<td>128</td>
<td>4.27 ± 0.10</td>
<td>4.25 ± 0.11</td>
<td>4.68 ± 0.11</td>
</tr>
</tbody>
</table>

Table 2: Mean RMS error (Cent) with 95% confidence intervals for the three split variants on the validation corpus

<table>
<thead>
<tr>
<th>N. of HMM</th>
<th>Split-1</th>
<th>Split-2</th>
<th>Split-n</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>165.50 ± 2.30</td>
<td>165.50 ± 2.30</td>
<td>165.50 ± 2.30</td>
</tr>
<tr>
<td>2</td>
<td>140.89 ± 2.01</td>
<td>140.89 ± 2.01</td>
<td>140.89 ± 2.01</td>
</tr>
<tr>
<td>4</td>
<td>130.96 ± 1.92</td>
<td>131.98 ± 1.90</td>
<td>131.98 ± 1.90</td>
</tr>
<tr>
<td>8</td>
<td>104.80 ± 2.05</td>
<td>113.86 ± 1.71</td>
<td>118.85 ± 1.92</td>
</tr>
<tr>
<td>16</td>
<td>79.81 ± 1.68</td>
<td>84.91 ± 1.58</td>
<td>98.26 ± 1.73</td>
</tr>
<tr>
<td>32</td>
<td>71.37 ± 1.56</td>
<td>70.53 ± 1.40</td>
<td>84.28 ± 1.69</td>
</tr>
<tr>
<td>64</td>
<td>66.97 ± 1.50</td>
<td>66.16 ± 1.53</td>
<td>75.63 ± 1.59</td>
</tr>
<tr>
<td>128</td>
<td>62.62 ± 1.48</td>
<td>62.08 ± 1.49</td>
<td>68.53 ± 1.50</td>
</tr>
</tbody>
</table>

6. Conclusion

In this article, a new unsupervised learning methodology based on HMM models for melodic contour classes is described. The results show a pretty good precision of the classes. The mean RMS error is near 4Hz which is the common JND threshold for the F0. Besides, HMM modelling enables to cluster contours of similar shape independently of their duration.

The experiments presented in this paper are based on melodic contours at a syllabic level. This methodology can be easily adapted to other temporal units like syllable sequences or intonational units.

Naturally, to validate the results and the usability of this method for TTS applications, listening tests would be necessary.

Having a set of melodic contour classes for two speakers, we can estimate a conversion function enabling the transformation from one’s speaker melodic contour classes (source speaker) into the classes of a target speaker. Moreover, the classification of melodic contours gives output labels corresponding to the F0 patterns. These labels could be used in a TTS system to enhance it and diversify the possible synthesized voices at a prosodic level.

7. References