B6

F0 Contour Parameterization Using Optimal Regression Chains

Aaron Albin (Kobe University)
albin@people.kobe-u.ac.jp

1. Background

In a wide range of applications, both practical and scientific, it is useful to discretize an F0 track into a finite set of parameters that collectively represent a 'stylized' version of the shape of the F0 contour in the raw data. Such algorithms are used, for example, in machine-learning, automatic speech recognition, and computer-assisted language learning. The specific case examined in the present study is the automatic classification of a Japanese word (or phrase) based on its pitch-accent type. After first providing an overview of one common approach to this problem in Section 1, a novel approach that avoids several of its shortcomings is described in detail in Section 2. Section 3 then illustrates the method by applying it to a test dataset. Finally, in Section 4, the paper concludes with a discussion of promising directions for future development.

1.1. Mora-based F0 contour parametrization

One popular approach to the F0 contour parametrization problem involves first calculating some form of representative F0 for each mora (e.g., by averaging across the F0 points therein), then calculating the change in these values between each pair of adjacent moras - each resulting change value being one parameter. This approach is described, for example, in Ishi et al. (2003) as the "CV-average" operationalization of "F0mora". While this method has proven useful in a wide range of contexts, at least three problems can be identified. First, consonantal 'microprosody' often creates unreliable F0 information and may lead to distorted parameter values. Such cases are not uncommon for [s], where most frames therein are voiceless, and for voiced stops like [g], where most frames are often voiced but merely represent a perturbation. Second, the number of parameters extracted for a given word is relatively small, e.g. only two parameters for a three-mora word like *nimono* ('stew'). Representing all possible F0 contour shapes over the six segments in a word like *nimono* with only two parameters involves a significant loss of information. Third, a similarly nontrivial amount of information is lost by summarizing by representing each mora's F0 with a single average. This is most problematic in cases where important F0 changes occur inside a single mora, e.g. if the F0 rises during the onset consonant and then falls during the vowel.

1.2. Present study

The following section describes an alternative approach to F0 contour parametrization, called Optimal Regression Chains (or "ORC"), that overcomes these three problems. With the proposed approach, one parameter is calculated for each individual segment (rather than each mora). The goal of this method is to extract a set of parameters from the F0 contour of an utterance in a way that preserves the separation between phonologically distinctive categories (e.g., Japanese accent types).

2. Proposed method

The proposed algorithm is implemented as a function in the R programming language that takes four pieces of information as input for any given file: (1) the soundfile itself (in .wav format), (2) a file containing F0 information (e.g., a Praat Pitch object saved in plain text format), plus a matrix containing (3) segmentation boundaries (i.e., timestamps of the beginning/end of every segment) and (4) labels for each segmentation interval indicating which phone is contained therein.

2.1. Reliable vs. unreliable F0 information

The discussion above alluded to the fact that the F0 information for certain segments is inherently more reliable than for others. This information is built directly into the ORC algorithm by making a distinction between 'reliable' and 'unreliable' portions of the F0 contour. Doing so makes the modeling more conservative by avoiding using F0 information that is likely to be influenced by well-documented sources of noise. Recall from above that any token to be analyzed must first be parsed into labeled intervals. Each such interval is classified as [+/- reliable] by applying two 'checks' to its associated information.

The first check involves cross-referencing each segment label with two (non-overlapping) exhaustive lists of [+ reliable] and [- reliable] labels, both specified by hand. Note that the since the inventory of segments is language-specific, and conventions for segment labeling can be researcher-specific, this information should be prepared specifically for each individual analysis. In informal testing, the following lists were found to be effective at maximally separating segments in Japanese with reliable and unreliable F0:

Reliable: (1) Vowels like /a,i,u,e,o/, (2) Nasals like /m,n/, (3) Approximants like /w,j/.

Unreliable: (4) Voiceless obstruents like /p,t,k,ts,tʃ,ɸ,s,ʃ,h/, (5) Voiced obstruents like /b,d,g,dʒ,z/, (6) flap consonants like /ɾ/

The second check involves calculating the percentage of voiced frames in each interval and confirm whether it falls above some minimum threshold. Even vowels can occasionally lack robust F0 for a variety of reasons, e.g., phonological vowel devoicing in Japanese, or utterance-final creaky voice. For this reason, it may be wise to treat the F0 information as unreliable if too many frames have missing/NA F0 values. The threshold itself can be set to any arbitrary percent, but informal testing suggests a minimum threshold around 20% is effective. That is, if less than 20% of the frames within an interval are voiced, then the F0 information in that interval is treated as unreliable. (Note that this second check can be 'turned off' by simply setting the threshold to 0%.)

In order for a given interval to be treated as reliable, it needs to pass both of the above checks. In other words, it needs to have not only an appropriate label but also a sufficiently high percentage of voiced frames. Any interval failing either (or both) of these two criteria is treated as unreliable. In intervals thus determined to be unreliable, all F0 points are changed to N/A, i.e. making it identical to intervals containing 0% voiced frames in the raw data.

2.2. Creating line segments

Next, a line segment is created for every interval. The exact details of how the line segments are determined depends on how many F0 points fall into that interval (i.e., how many voiced frames there are). In the majority case where there are 2 or more F0 points in the interval, a linear regression is fit to these points (with x=time and y=F0). In the rare case that there are exactly 2 points, this regression is trivially identical to simply connecting those two points with a straight line. If there is only 1 point in the interval, a perfectly horizontal (zero-slope) line passing through that point is used – i.e., with a single F0 value held constant throughout the interval.

Line segments are created even for intervals with no F0 points - either from lacking F0 points in the raw F0 track or due to being NA-ed out for having unreliable F0 as discussed in Section 2.1. Since such intervals have no usable F0 information, F0 information from neighboring intervals is used to fill in the gap. If the interval in question is initial or final within the word/sentence token, the nearest regression endpoint is copied to fill in the missing F0 values (via 'constant extrapolation'). For example, in a word like *ki* 'tree', if the first interval (/k/) is missing F0 and the fitted regression line begins at 123 Hz in the second interval (/i/), then the beginning and ending F0 values for the first interval are set to 123 Hz as well. If there are multiple intervals with missing F0 (the first 3 segments in a token of *suki* 'like' with a devoiced /u/), constant extrapolation is applied across all of them.

Alternatively, if the interval without F0 points is medial within the token (i.e., anything but the first or last), the line segment is created by copying the values of the regression endpoints in the adjacent intervals. For example, in a word like *aki* 'autumn', with missing F0 for the /k/, if the regression line of the interval to the left (/a/) ended at 234.5 Hz, the beginning of the target interval (/k/) is assumed to be 234.5 Hz as well. If there are multiple medial intervals with missing F0, linear interpolation is used to fill in the gaps. For example, in a token of *deshita* 'was' with the entire sequence [ɕi̥t] deemed unreliable due to devoiced /i/, if /e/ ends at 190 Hz and /a/ begins at 100 Hz, then the line segments would be filled in as follows: /ʃ/=190-160 Hz, /i/=160-130 Hz, /t/=130-100 Hz.

2.3. Optimizing the junctions

The linear regressions described in Section 2.2 are fit on an interval-by-interval basis, in isolation of the F0 points in all other intervals. As such, the 'junctions' (i.e., points of union) between two neighboring intervals almost never match up. For instance, in the word *ao* 'blue', a linear regression fit to the F0 over the /a/ may end up at 140.1 Hz, and the regression over /o/ may begin at 149.9 Hz. The resulting model is physically unrealistic since, at the moment of the junction, it implies the speaker's pitch needs to be at two different values simultaneously. Moreover, by unnecessarily having two parameters at each junction rather than one, the model is arguably overfitting the data.

The proposed method overcomes this problem by using an optimization algorithm – the `optim()` function in R – to determine, for each junction, which exact F0 value would fit the raw data the best. The `optim()` function is set to `method="L-BFGS-B"` so that the parameter search is 'box-constrained', i.e. only certain ranges of values are considered. More specifically, for each junction, of the two competing regression endpoints, the smaller one (i.e. the one with the lower Hertz value) is rounded down, and this is used as the lower bound. Likewise, the higher one is rounded up, and this is used as the upper bound. For the above *ao* example, the lower value is 140.1, rounded down to 140, and the upper value is 149.9, rounded up to 150, hence the ultimately-chosen best-fitting parameter must be between 140 and 150 Hz. The values used to initialize the parameter search are the midpoints between each such pair of (unrounded) bounds, i.e., 145 Hz in this example. In this way, in the set of parameters used in `optim()`, there is one parameter for each junction, totaling to the number of segments/intervals minus one (e.g., 6-1=5 junctions for *nimono* 'stew'). Note that the very beginning of the utterance (more precisely, the beginning of the regression line inside the first interval containing F0 points, e.g., the beginning of /n/ in *nimono*), is <u>not</u> treated as a free parameter. Rather, this value is determined based on a linear regression constrained to pass through the [time,F0] point of the first junction (e.g., the junction between the first /n/ and the /i/ in *nimono*). The same is true of the last interval containing F0 points, whose regression endpoint is likewise determined based on a constrained regression that must pass through the last junction. (Note that the regressions run initially, as described in Section 2.2, are free of such a constraint.)

At each step in the search through the parameter space, the reconstructed model for the contour as a whole is created through linear interpolation between the F0 targets represented by the various parameters. Each resultant model (one for each step in the search) is then evaluated in terms of goodness-of-fit by calculating the median absolute deviation ("MAD") between the model F0 and the raw F0. Since the model contour is created through linear interpolation, which can generate an F0 point at any arbitrary point in time, the time sampling between the model F0 and the raw F0 is kept identical, making it possible to directly subtract one from the other. Since sign (positive vs. negative) is not important, the absolute value is then calculated for each deviation. In an effort to mimic perception, the resulting values are weighted so that high-intensity frames impact the resulting statistic more than low-intensity frames. (For further details on MAD, see Albin (2015, pp.83-84), which uses the same method.)

The overall end result of applying the ORC algorithm is the matrix with one row for each segment/interval and the following columns: (1) Label, (2) Reliable, (3) Time0, (4) Time1, (5) Hertz0, (6) Hertz1, (7) Cents, (8) Voicing, (9) Utilized. Column (1) contains the label for the interval in question, and (2) indicates whether that label is classified as reliable. Columns (3) through (6) contain the time and F0 ("Hertz") information for the beginning ("0") and end ("1") of the line segment inside that interval. Column (7) represents the change in F0 from Hertz0 to Hertz1 in cents (i.e. semitones ×100). Column (8) indicates the percentage of frames inside the interval that are voiced, from 0 to 1 (0% to 100% voiced). Column (9) indicates whether the F0 information inside the interval in question was utilized or not, based on the [+/-reliable] and voicing threshold criteria.

Of this rich information, at a bare minimum, only two pieces of information are necessary to represent a 'skeleton' of the entire contour shape: (A) the 'Hertz0' value of the very first segment, i.e. where the contour as a whole begins in Hertz space, and (B) the 'Cents' values for every segment in the contour. The former likely indexes things like speaker sex and emotional arousal, whereas the latter contains phonologically-relevant information about contour shape.

3. Application to test dataset

The materials used for the present test application were 160 tri-moraic Japanese nonwords, originally designed for another purpose. These words represented three different pitch-accent types: initial-accented (on the first mora), medial-accented (on the second mora), and unaccented. The 160 words were split into four subgroups of 40 words, each consisting of 20 minimal pairs: (1) unaccented-medial minimal pairs (*hetoya-he**tó**ya*), (2) unaccented-initial minimal pairs (*wakumi-**wá**kumi*), (3) initial-medial minimal pairs (***kó**zabe-ko**zá**be*), and vocalic minimal pairs (20 pairs like *dohesa-dohosa*). Globally, the 160 words were roughly balanced for accent type: 52 were initial-accented, 52 were medial-accented, and 56 were unaccented. 140 words were CVCVCV, 14 words were VCVCV (e.g., *ateyu*), 4 words were CVVCV (e.g., *meobi*), and 2 words were CVCVV (e.g., *dotsua*). In terms of segmental makeup, across all 160 words, there were 480 vowels, 154 voiced obstruents (including the flap /r/), 162 voiceless obstruents, and 144 sonorant consonants. The approximately equal representation among the three different classes of consonants is crucial for illustrating the effectiveness of the proposed method. These 140 words were read aloud in a quiet room by three female native speakers of Japanese: two from Shizuoka prefecture and one from Hiroshima prefecture. Words were blocked so that everything within a block had the same accent pattern, thus making it easier to pronounce the non-words with the intended accentuation. In total, with tokens from 3 speakers for each of 160 words, 480 soundfiles in total were analyzed with the proposed method.

4. Results

Figure 1 is an example of what the output of ORC looks like, as applied to an initial-accented word (as evidenced by the peak at the end of the first [e]). The top panel is the waveform, the middle panel is the F0 track (where thicker, redder portions of the contour indicate higher-quality F0 information), and the bottom panel is the segmentation superimposed over the spectrogram. The solid black lines in the F0 track are those created through regression. Since there are no F0 points in the initial [z], the dotted grey line over [z] was created through extrapolation leftward from the beginning of the regression line for the first [e]. Likewise, the unreliable F0 information during the flap [ɾ] is ignored and instead the dotted grey line is filled based on the regressions in the surrounding vowels. Note that, on the whole, the straight-line model fits the raw data quite well, and the 'filled-in' values are plausible representations of what the missing F0 information might have looked like.

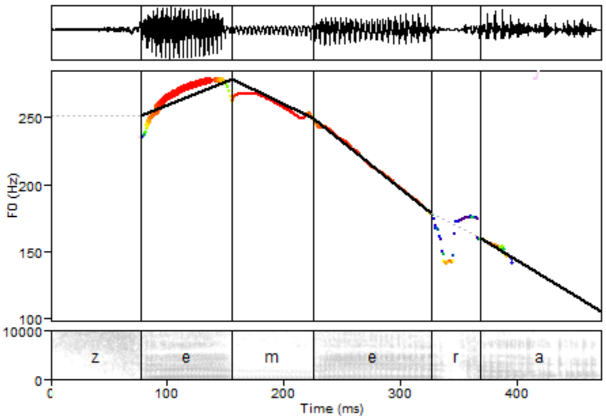

Figure 1: Example of output, applied to nonword ***zé**mera*

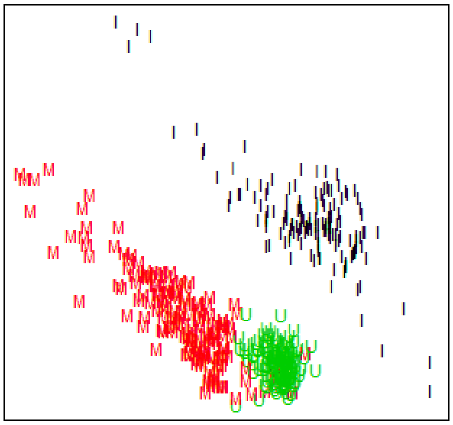

Figure 2: Multidimensional scaling results

The straight-line model in Figure 1 can be represented with the following seven parameters: 251, 0, 178.2, -192.1, -576.8, -188.9, -722. The first of these (251) is the initial F0 value (in Hertz), and the remaining six correspond to the size of F0 change over each of the six segments (in cents). Due to the makeup of the test dataset, these seven parameters can be estimated for nearly every token. (The only exception is the minority of VCVCV, CVVCV, and CVCVV words, for which the parameters for the missing consonants are N/A.) Setting aside the 'initial F0 value' parameter, the remaining six main parameters were visualized using multidimensional scaling, the output of which appears in Figure 2. There are 480 points in the plot – one for every token in the dataset. Initial-accented tokens are marked with black "I", medial-accented ones with red "M", and unaccented ones with green "U".

5. Conclusion

The clear separation between the 3 point clouds in Figure 2 is a testament to the effectiveness of the proposed method in extracting from the signal a set of parameters that maintains separation between the different phonological categories in question – thus attesting to how the proposed method successfully achieves its stated goal. This method holds much promise in a wide range of contexts, e.g., automated analysis of hard-to-classify productions by second language learners. Among numerous directions for future research, of particular importance is a systematic side-by-side comparison of the performance of the proposed method alongside other traditional methods.

References

Albin, A. (2015). Typologizing native language influence on intonation in a second language: Three transfer phenomena in Japanese EFL learners. Ph.D. dissertation. Indiana University.

Ishi, C. T., Hirose, K., and Minematsu, N. (2003) "Mora F0 representation for accent type identification in continuous speech and considerations on its relation with perceived pitch values", Speech Communication 41, 441-453.