MULTIBAND STATISTICAL LEARNING FOR F_0 ESTIMATION IN SPEECH

Fei Sha, J. Ashley Burgoyne, and Lawrence K. Saul

Department of Computer and Information Science
University of Pennsylvania, Philadelphia, PA 19104

ABSTRACT

We investigate a simple algorithm that combines multiband processing and least squares fits to estimate f_0 contours in speech. The algorithm is untraditional in several respects: it makes no use of FFTs or autocorrelation at the pitch period; it updates the pitch incrementally on a sample-by-sample basis; it avoids peak picking and does not require interpolation in time or frequency to obtain high resolution estimates; and it works reliably, in real time, without the need for postprocessing to produce smooth contours. We show that a baseline implementation of the algorithm, though already quite accurate, is significantly improved by incorporating a model of statistical learning into its final stages. Model parameters are estimated from training data to minimize the likelihood of gross errors in f_0, as well as errors in classifying voiced versus unvoiced speech. Experimental results on several databases confirm the benefits of statistical learning.

1. INTRODUCTION

There exists a large body of work on pitch determination of speech signals[1, 2, 3, 4]. The goal of algorithms for pitch tracking is to estimate the fundamental frequency, f_0, of speech as generated by the quasi-periodic vibration of the vocal cords (and as corresponds typically to its perceived pitch).

Most algorithms for pitch tracking involve one or more of the following components: (i) preprocessing to enhance the periodicity of the waveform (e.g., lowpass filtering, center clipping), (ii) short-time analysis of speech to obtain initial estimates of f_0, and (iii) postprocessing to correct isolated errors and produce smooth contours (e.g., median filtering, dynamic programming). Methods in the time domain[2, 3, 4] typically rely on autocorrelation at the pitch period, detecting f_0 from one or more peaks in the autocorrelation function. Likewise, methods in the frequency domain[1, 4] typically rely on FFTs, detecting f_0 (for example) from peaks the magnitude cepstrum or harmonic power spectrum. Peak-picking in either domain can be difficult, requiring special heuristics to handle the complexities of voiced speech (e.g., quasiperiodicity, nonstationarity). Initial f_0 estimates from peak-picking are also limited in resolution by the sampling rate (in the time domain) or FFT size (in the frequency domain), though in either domain they can be subsequently refined by interpolation.

Recently, we introduced a multiband least squares algorithm[5] for pitch tracking that takes a different approach. In particular, the algorithm makes no use of FFTs or autocorrelation at the pitch period; it does not require interpolation to obtain high resolution estimates of f_0; and it works reliably, in real-time, without the need for postprocessing to produce smooth contours. The algorithm is based on the assumption that the low frequency spectrum of voiced speech can be modeled as a sum of (noisy) sinusoids occurring at integer multiples of f_0. Using a nonlinearity to concentrate energy at f_0 and a bank of overlapping bandpass filters with carefully arranged passbands, the algorithm detects voiced speech by ascertaining that the output of one filter resembles a sinusoid at frequency f_0, while the others do not. Sinusoids are detected by simple, one-parameter least-squares fits.

In this paper, we significantly extend our previous work. Not only do we benchmark the algorithm on much larger data sets, but we also incorporate a model of statistical learning into its final stages. The model is a multiway classifier trained to select the bandpass filter whose output reveals the fundamental frequency f_0 of the speech waveform. The parameters of the model are estimated from the TIMIT database[6], containing over two hours of speech. The classifier adds little computational overhead to the original algorithm, but yields significantly fewer errors in estimating f_0 and classifying voiced versus unvoiced speech. We have also incorporated the multiband classifier into a real-time implementation for pitch tracking.

2. MULTIBAND LEAST SQUARES METHOD

In this section, we briefly review the multiband least squares method for pitch tracking; more details can be found in our earlier work[5]. We first explain how least squares fits can be used to detect sinusoids in individual subbands, then extend this approach to the more general problem of estimating the fundamental frequency of a periodic (though not sinusoidal) waveform.

2.1. Detecting sinusoids

Consider the problem of detecting sinusoids. The approach we describe here is a simple variant of Prony's method[7]. Note that for a discretely sampled sinusoid $s_n = A\sin(\omega n+\theta)$, each sample is proportional to the average of its neighbors, with the constant of proportionality given by:

$$s_n = (\cos\omega)^{-1}\left[\frac{s_{n-1}+s_n}{2}\right]. \qquad (1)$$

We can use eq. (1) to measure how well an unknown signal x_n is described by a sinusoid. In particular, consider the error function:

$$\varepsilon(\alpha) = \sum_n \left[x_n - \alpha\left(\frac{x_{n-1}+x_{n+1}}{2}\right)\right]^2. \qquad (2)$$

If x_n is well described by a sinusoid, then the right hand side of eq. (2) will achieve a small value when the coefficient α is tuned to match its frequency, as in eq. (1). The minimum error least squares fit is given by:

$$\alpha^* = \frac{2\sum_n x_n(x_{n-1}+x_{n+1})}{\sum_n (x_{n-1}+x_{n+1})^2}. \qquad (3)$$

We can judge whether x_n is sinusoidal by checking two conditions: first, $\varepsilon(\alpha^*) \ll \varepsilon(0)$, and second, that $|\alpha^*| > 1$. The first condition establishes that the residual error is small relative to the overall energy of the signal, while the second establishes that the signal oscillates with an estimated (real-valued) frequency:

$$\omega^* = \cos^{-1}(1/\alpha^*). \qquad (4)$$

The above scheme has several useful properties for our purposes. First, the frequency estimate is the solution to a least squares problem; hence, its resolution is not limited by the sampling rate, as (say) the location of the peak of an autocorrelation function. Second, it relies only on the zero-lagged and one-lagged autocorrelation in eq. (3), namely $\sum_n x_n^2$ and $\sum_n x_n x_{n\pm 1}$. Third, the method is easily adapted to tracking the frequency of nonstationary signals; we simply analyze the signal with windows that shift one sample at a time, incrementally updating the autocorrelations that appear in eq. (3) for adjacent windows.

Eq. (1) can be viewed as a one-parameter autoregression, a predictive model that forecasts s_{n+1} from s_n and s_{n-1}. The second derivative of its error function thus characterizes the uncertainty in the fit due to noises in the observation process[8]. Intuitively, the closer the signal x_n is to a sinusoid, the sharper the fit and the less its uncertainty. Let $\mathcal{N}(\alpha) = \varepsilon(\alpha)/\varepsilon(0)$ be the dimensionless, normalized error function (which is insensitive to the amplitude of the signal), and let $\Delta\mu$ denote the uncertainty in the estimated log-frequency $\mu^* = \log\omega^*$, characterized by:

$$\Delta\mu \;=\; \left[\frac{\partial^2 \log \mathcal{N}}{\partial \mu^2}\right]^{-\frac{1}{2}}_{\mu=\mu^*} \;=\; \frac{\cos^2\omega^*}{\omega^* \sin\omega^*}\left[\frac{1}{\varepsilon}\frac{\partial^2 \varepsilon}{\partial \alpha^2}\right]^{-\frac{1}{2}}_{\alpha=\alpha^*} \qquad (5)$$

By working in the log domain, we measure uncertainty in units proportional to the distance between notes on the musical scale. We shall see in later sections that this measure of uncertainty is a useful feature for pitch tracking.

2.2. Estimating f_0 in speech

The multiband least squares method for pitch tracking is a simple extension of the method in the previous section. The algorithm operates in a number of stages, as shown in Fig. 1, and as summarized below.

2.2.1. *Preprocessing*

In the first stage, the signal is lowpass filtered to remove energy above 1 kHz, then transformed by a pointwise nonlinearity such as squaring or half-wave rectification. The lowpass filter is used to remove the aperiodic components of voiced fricatives, while the nonlinearity helps to concentrate energy at f_0 in the case of a weak or missing fundamental. The signal can also be downsampled at this stage for faster processing.

2.2.2. *Filterbank*

In the second stage, the signal is analyzed by a bandpass filterbank whose filters are designed to satisfy two competing criteria. On one hand, we make them sufficiently narrow to resolve the fundamental at f_0, while on the other hand, we make them sufficiently wide to integrate higher-order harmonics. An idealized two-octave filterbank with these properties is shown in Fig. 1. The result of this analysis for voiced speech is that the output of the filterbank consists of either sinusoids at f_0 (and not any other frequency) or signals do not resemble sinusoids at all. For example, consider a segment of voiced speech with fundamental frequency $f_0 = 180$ Hz. In this case, the second filter with a passband of 50–200 Hz will output a sinusoid at $f_0 = 180$ Hz. By contrast, the first filter will output low frequency noise; the third filter will pass the first and the second harmonics at 180 Hz and 360 Hz, and the fourth filter will pass the second, third, and fourth harmonics at 360, 540, and 720 Hz. Thus, the outputs of the first, third, and fourth filters bear no resemblance to pure sinusoids. While the idealized filters in Fig. 1 have unrealizably steep rolloffs, we obtain the same effect in practice by implementing a larger bank of six filters with narrower (1.6 octave) passbands.

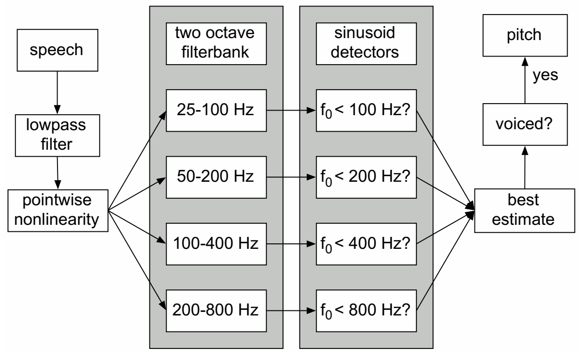

Fig. 1. Multiband least squares method for estimating f_0.

2.2.3. *Sinusoid detection*

The final stage of the algorithm is sinusoid detection at the outputs of the filterbank, using the method described in section 2.1. Running least squares fits and their uncertainties are computed from eqs. (3–5). The fits are updated on a sample by sample basis for the output of each filter. A voicing decision is then made for frames whose total energy exceeds a minimum “silence” threshold. Nonsilent frames are labeled as voiced if the uncertainty $\Delta\mu$ of any subband f_0 estimate from the filterbank is less than a specified threshold; in this case, the value of f_0 is estimated from the subband with the sharpest least squares fit. Otherwise, the frame is labeled as unvoiced.

3. MULTIBAND STATISTICAL LEARNING

In previous work[5], we showed that the algorithm from section 2 works well for batch and real-time f_0 estimation in speech. Nevertheless, the algorithm hinges in an unsatisfying way on two heuristics—namely, the manual tuning of energy and sharpness-of-fit thresholds to minimize errors in voiced/unvoiced classification, and the use of the uncertainty criterion in eq. (5) to select the correct subband from which to estimate f_0. It is obvious that additional features could be used for these purposes, but it is not obvious (from a priori knowledge) how to combine them in a sensible way. In this section, we describe a multiband classifier that can be trained from reference f_0 contours to provide better decisions for voiced/unvoiced classification and subband selection in the final stages of our algorithm.

3.1. Inputs and outputs

We considered various features as input to the classifier, eventually settling on the following. For each frame, and in each subband, we computed three features: the square root of the normalized residual error $\mathcal{N}(\alpha)$, the cube root of the uncertainty measure $\Delta\mu$, and the logarithm of the energy. The nonlinear transformations in these features were chosen to equalize their dynamic ranges. The same features at the preceding frame were also included as input to the classifier. Thus, for each frame, the classifier input consisted of a 36-dimensional feature vector $\mathbf{x} \in \mathcal{R}^{36}$ consisting of six features from each subband and reflecting not only the signal properties in that frame but also their first derivatives in time. (Note that features from succeeding frames were deliberately excluded to minimize the latency of a real-time implementation.)

The classifier output was a discrete label $y \in \{0,1,2,3,4,5,6\}$ indicating either that the frame was unvoiced ($y = 0$) or that the output of the y^{th} subband should be used to estimate f_0. Thus, whereas in the original algorithm the signal was classified as unvoiced or voiced based on simple energy and sharpness-of-fit thresholds, the multiband classifier was trained to make these decisions by analyzing the much richer input provided by each frame's feature vector.

3.2. Model and training

The classifier was trained by multinomial logistic (or "softmax") regression[8] on the feature vectors. Target labels for the classifier outputs were provided by reference f_0 contours from a large database of speech (described in section 4).

Softmax regression is a simple model for multiway classification. The model computes the posterior probability that a feature vector $\mathbf{x}$ has class label y as:

$$\Pr(y = i|\mathbf{x}) = \frac{e^{\mathbf{w}_i \cdot \mathbf{x}}}{\sum_j e^{\mathbf{w}_j \cdot \mathbf{x}}}, \tag{6}$$

where $\mathbf{w}_j$ is the weight vector attributed to class j, and the sum in the denominator—over all class labels—ensures that the right hand side defines a properly normalized distribution. In practice, the feature vector $\mathbf{x}$ is labeled by whichever class maximizes the right hand side of eq. (6), or equivalently, whichever class maximizes the dot product $\mathbf{w}_i \cdot \mathbf{x}$ that appears in the exponent. Because of this, eq. (6) gives rise to piecewise linear class boundaries in the feature space.

The weight vectors $\mathbf{w}_j$ are the parameters of the softmax regression. They can be estimated from a large training set of labeled examples. Let $(\mathbf{x}_t, y_t)$ denote the t^{th} example in the training set. We choose the parameters to maximize the total log-likelihood of correct classification:

$$\mathcal{L} = \sum_t \log \Pr(y = y_t|\mathbf{x}_t). \tag{7}$$

The log-likelihood $\mathcal{L}$ is a concave function of the parameters $\mathbf{w}_j$, and its global maximum can be computed by iterative procedures such as gradient ascent or Newton's method. In our experiments, we used a variant of Newton's method that updated one weight vector at a time, as opposed to all the weight vectors at once: this was done to avoid computing the entire hessian matrix.

The above model is easily incorporated into the final stages of the multiband least squares algorithm. Whereas our original implementation only considered the minimum uncertainty $\Delta\mu$ across subbands, the classifier in eq. (6) uses much more information to make voiced/unvoiced decisions and to determine (in voiced frames) the subband from which to estimate f_0. Also, unlike the manually tuned energy and sharpness-of-fit thresholds in section 2, here the classifier parameters have the benefit of being optimized over a large training set of labeled examples.

4. EVALUATION

We evaluated the multiband least squares algorithm for pitch tracking on several data sets of speech. To assess the benefits of statistical learning, we collected results both before and after incorporating the multiband classifier into the final stages of the algorithm.

4.1. TIMIT data

To train the classifier, a large data set of speech with reference pitch contours was needed. We used the training portion of the TIMIT data set[6] for this purpose. The TIMIT utterances are not distributed with f_0 contours, so we used two independent, state-of-the-art pitch tracking algorithms (get_f0 from ESPS[2] and YIN[3]) in conjunction with the TIMIT phonetic alignments to derive voiced/unvoiced labels and reference f_0 contours. Both get_f0 and YIN were used with their default parameter settings, except for adjustments of the frame rate. Classifier targets were derived as followed. Frames were labeled as unvoiced (with a classifier target of $y = 0$) if both the phonetic alignment and get_f0 labeled them as unvoiced. Frames were labeled as voiced if the f_0 estimates of get_f0 and YIN were within 20% of each other. For frames labeled as voiced, the f_0 estimates from get_f0 were converted into targets $y \in \{1, 2, 3, 4, 5, 6\}$ for the multiband classifier based on the subbands that contained them. Ambiguous frames, including those in voiced-unvoiced and unvoiced-voiced transitions, were discarded from training and testing.

The training portion of the TIMIT data set consists of 4620 utterances from adult male and female speakers from the major dialect regions of the US; with an analysis window of 40 ms, and a 10 ms shift between frames, a total of 1015630 frames were collected for training (not including discarded frames). The testing portion of the TIMIT data set consists of 1680 utterances, giving rise to 369378 frames for testing.

Experimental results on the TIMIT data set are shown in Table 1. In the table, MLS and MLS$^+$ refer respectively to the multiband least squares algorithm before and after the incorporation of the statistical learning model. The first two rows report the percentage of unvoiced frames misclassifed as voiced ("unvoiced in error") and the percentage of voiced frames misclassified as unvoiced ("voiced in error"). The third row ("gross errors") reports the percentage of voiced frames where the f_0 estimates from MLS and MLS$^+$ differed from the ground-truth estimate (supplied by get_f0) by over 20%. Finally, the last row reports the root-mean-squared (RMS) difference between the estimated f_0 and the get_f0 value for frames without gross errors.

The benefits of statistical learning are apparent. The incorporation of the multiband classifier leads to roughly a halving of errors in the classification of unvoiced/voiced frames and grossly incorrect estimates of f_0. Note that comparative results for get_f0 and YIN do not appear in Table 1 because these algorithms were used to derive the "ground-truth" estimates of f_0.

Table 1. Results on TIMIT data.

	train		test	
error type	MLS	MLS$^+$	MLS	MLS$^+$
unvoiced in error(%)	4.64	1.42	4.28	1.27
voiced in error (%)	2.25	1.58	2.34	1.65
gross errors (%)	1.31	0.69	1.31	0.70
rms (Hz)	3.41	3.49	3.41	3.53

Table 2. Results on Keele and Edinburgh data.

Keele				
error type	MLS	MLS$^+$	get_f0	YIN
unvoiced in error(%)	8.60	7.90	6.83	–
voiced in error (%)	8.87	7.03	3.24	–
gross errors (%)	1.68	1.5	2.29	3.28
rms (Hz)	4.68	4.54	4.5	3.62

Edinburgh				
error type	MLS	MLS$^+$	get_f0	YIN
unvoiced in error(%)	4.86	5.65	8.84	–
voiced in error (%)	7.97	5.38	4.29	–
gross errors (%)	0.39	0.67	2.86	3.48
rms (Hz)	5.88	5.88	5.83	6.2

4.2. Keele and Edinburgh data

We also evaluated the MLS algorithms on two smaller data sets with reference f_0 contours derived from laryngograph signals: the Keele data set[9] and the Edinburgh data set[4]. The Keele data set contains roughly five minutes of speech from five male and female adult speakers, while the Edinburgh data set contains roughly five minutes of speech from two adult speakers, one male and one female. Both data sets have somewhat different acoustic, phonetic and linguistic characteristics from those of TIMIT.

The results for the Keele and Edinburgh data sets are shown in Table 2. Note that the classifier parameters in MLS$^+$ were not adapted to these data sets; their values were frozen after being estimated from the TIMIT training data. The only allowance made for the Keele and Edinburgh data sets was to scale the energy features computed from their waveforms so that they had a similar histogram as those from the TIMIT waveforms. Here again, the results show that the incorporation of a statistical learning model leads to generally improved performance of the MLS algorithm.

For these data sets, it was also possible to evaluate get_f0 and YIN in the same experimental setup. Note that YIN does not classify frames as voiced/unvoiced, so these error rates are not reported for YIN. Among MLS$^+$, get_f0, and YIN, no method uniformly outperforms the others. In these comparisons, however, it is worth remembering that get_f0 is not suited to real-time implementations (because it relies on dynamic programming for smoothing of pitch contours), and that YIN does not classify frames as voiced/unvoiced. The MLS$^+$ algorithm is well suited to applications with both these requirements[5]. More generally, we believe that it provides an interesting, competitive alternative to two leading algorithms based on autocorrelation at the pitch period.

5. DISCUSSION

We have shown how to improve a pitch-tracking algorithm based on multiband least squares fits by incorporating a model of statistical learning into its final decision process. In addition to benchmarking the algorithm on the TIMIT, Keele, and Edinburgh data sets, we have also implemented a real-time version of the improved algorithm on a 1 GHz Macintosh PowerBook G4. This implementation is being used as a front-end to interactive applications with voice-driven agents and real-time audiovisual feedback[5]. Finally, while the algorithm described in this paper was conceived only for clean speech, we are investigating how to apply similar ideas to noisy environments, polyphonic music, and "cocktail parties" with overlapping speakers. We hope that the ideas in this paper will ultimately inspire novel approaches to f_0 estimation in these more challenging settings.

6. ACKNOWLEDGMENTS

The authors are very grateful to Alain de Cheveigné for providing the f_0 estimation software for the YIN algorithm[3].

7. REFERENCES

[1] W. J. Hess, "Pitch and voicing determination," in *Advances in Speech Signal Processing*, S. Furui and M. M. Sondhi, Eds., pp. 3–48. Marcel Dekker, Inc., New York, 1992.

[2] D. Talkin, "A robust algorithm for pitch tracking (RAPT)," in *Speech Coding and Synthesis*, W. B. Kleijn and K. K. Paliwal, Eds., pp. 497–518. Elsevier Science, 1995.

[3] A. de Cheveigné, "YIN, a fundamental frequency estimator for speech and music," *Journal of the Acoustical Society of America*, vol. 111, no. 4, pp. 1917–1930, 2002.

[4] P. C. Bagshaw, S. M. Hiller, and M. A. Jack, "Enhanced pitch tracking and the processing of f0 contours for computer aided intonation teaching," in *Proceedings of the 3rd European Conference on Speech Communication and Technology*, 1993, vol. 2, pp. 1003–1006.

[5] L. K. Saul, D. D. Lee, C. L. Isbell, and Y. LeCun, "Real time voice processing with audiovisual feedback: toward autonomous agents with perfect pitch," in *Advances in Neural Information Processing Systems 15*, S. Becker, S. Thrun, and K. Obermayer, Eds. 2003, MIT Press.

[6] J. S. Garofolo, L. F. Lamel, W. M. Fisher, J. G. Fiscus, D. S. Pallett, N. L. Dahlgren, and V. Zue, "The DARPA TIMIT acoustic-phonetic continuous speech corpus," NTIS order number PB91-100354, 1992.

[7] J. G. Proakis, C. M. Rader, F. Ling, M. Moonen, I. K. Proudler, and C. L. Nikias, *Algorithms for Statistical Signal Processing*, Prentice Hall, 2002.

[8] C. M. Bishop, *Neural Networks for Pattern Recognition*, Oxford University Press, 1995.

[9] F. Plante, G. F. Meyer, and W. A. Ainsworth, "A pitch extraction reference database," in *Proceeding of the 4th European Conference on Speech Communication and Technology*, 1995, vol. 2, pp. 837–840.