Parameterization and automatic labeling of Hungarian intonation

Uwe D. Reichel[1], *Alexandra Markó*[2], *Katalin Mády*[3]

[1]Institute of Phonetics and Speech Processing, University of Munich, Germany
[2]Eötvös Loránd University Faculty of Humanities, Budapest, Hungary
[3]Hungarian Academy of Sciences, Budapest, Hungary

reichelu@phonetik.uni-muenchen.de, marko.alexandra@btk.elte.hu, mady@nytud.hu

Abstract

In Hungarian intonation research the goal of a common framework developed by Varga (2002; [1]) is to categorize the intonation within the domain of accent groups by *character contours*. We propose a linear parameterization of a subset of these contours derived from polynomial stylization. These parameters were used to train classification trees and support vector machines for contour prediction. Parameter extraction and training was carried out on the original F0 contours of spontaneous speech data as well as on three differently normalized variants suppressing fundamental frequency level and range effects. The highest accuracies were obtained for classification trees and F0 residuals after midline subtraction, but the overall performances were rather poor. Nevertheless, a significant improvement of the results was achieved by a Hidden Markov model to predict the correct label sequence from the partly erroneous classification output.

Index Terms: intonation, Hungarian, character contours, stylization, labeling

1. Introduction

An established approach in Hungarian intonation research is to describe fundamental frequency (F0) curves in terms of *character contours (CC)*. This framework was developed by Varga [2, 1] and follows the tradition of contour-based intonation representations [3, 4] that focus on the contour properties of F0 rather than treating it as a sequence of tone targets [5].

Varga [1, p. 33] defines a CC as a *"discrete, meaningful speech melody"* with a *"characteristic shape"*. Its domain is a syllable sequence consisting of an initial accented (*"major-stressed"* [1, p. 33]) syllable and all following syllables till the next accented one or till the end of the intonation phrase. We refer to this sequence by the term *accent group* (AG) in the following.

According to [1] in Hungarian eleven character contours (and an appended contour that is not necessarily related to an AG) can be distinguished. The nine main contours are illustrated in Figure 1. They can be divided into three major classes: *i. front falling* (left column) *ii. sustained* (middle column), and *iii. end-falling* (right column). The abstract meanings assigned to these major classes are *self-contained*, *forward-pointing*, and *yes-no interrogative*, respectively.

For contour-based as well as for tone-sequence intonation models numerous machine learning techniques have been developed to derive the intonation representation automatically from the signal. Among the established techniques to learn categorical intonation labels from acoustic features are neural networks [6], decision trees and predicate logic learning rules [7], Hidden Markov models [7, 8] classification and regression trees [9], support vector machines, and instance-based learning [10]. [11] and [12] extract intonation categories in an unsupervised bottom-up fashion by means of clustering. The extraction of non-categorical parametric intonation representations is generally treated as an optimization task in an analysis-by-synthesis framework to minimize the distance between the observed F0 contour and the contour generated by the parameterization [13, 14, 15].

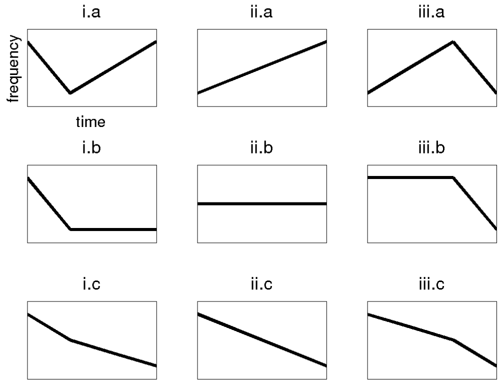

Figure 1: *Nine of the eleven character contours proposed by Varga [1, p. 43].*

Despite of this extensive literature on automatic intonation labeling, to our knowledge no approach has yet been published to extract the described character contours from the signal automatically. The aims of this study are thus to develop an appropriate F0 parameterization linking the signal to the character contour inventory and to train classifiers for automatic F0 labeling.

2. Data and preprocessing

The examined data consists of 50 Hungarian spontaneous speech utterances from collaborative dialogs by 10 Hungarian speakers. Each utterance forms a single intonation phrase (IP). Within each IP the accent groups were manually segmented following the definition of section 1, and the character contours were manually labeled by phonetic expert native speakers (the second and the third author of this study). In total, the data contains 140 AGs each linked to a character contour.

F0 was extracted by autocorrelation (Praat 5.3, sample rate

100 Hz). Voiceless utterance parts and F0 outliers were interpolated by piecewise cubic splines. The contour was then smoothed by Savitzky Golay filtering using third order polynomials in 5 sample windows and transformed to semitones relative to a base value. This base value was set to the F0 median below the 5th percentile of an utterance and serves to normalize F0 with respect to its overall level.

2.1. Normalization

To reduce the influence of F0 register on the local character contour shapes we generated three F0 residual variants. To capture the register in terms of its level and range [16] we fitted a base-, a mid-, and a topline for the IP. The baseline and the midline represent different aspects of the F0 level, whereas the F0 range information is provided by the time-varying span between the base- and topline. The robust fitting procedure that is motivated and explained in greater detail in [17] has already been applied for boundary strength [17] and prosodic phrase examinations [18]. The procedure is illustrated in the left panel of Figure 2 and consists of the following steps:

- A window of length 200 ms is shifted along the F0 vector with a stepsize of 10 ms.
- Within each window the F0 median is calculated
  - of the values below the 10th percentile for the baseline,
  - of the values above the 90th percentile for the topline, and
  - of all values for the midline.

  This gives 3 sequences of medians, one for the base-, the mid-, and the topline, respectively.
- Within each median sequence outliers are replaced by linear interpolation.
- Finally, for all three median sequences linear polynomials are fitted.

From this register stylization three F0 residuals were generated as can be seen in the right panel of Figure 2:

- the **baseline residual** by subtraction of the baseline from the F0 contour,
- the **midline residual** by subtraction of the midline from the F0 contour, and
- the **range residual** by normalizing each F0 value to the range between base- and topline at the corresponding position. These local reference points are set to 0 (baseline) and 1 (topline), respectively.

For the first two residuals the influence of register level is suppressed, whereas the impact of range is reduced for the third residual.

3. Parameterization

3.1. Method

Within each accent group segment we parametrized the original F0 contour as well as all residuals to derive a character contour representation. As can be seen in Figure 1 all contours can prototypically be represented by single lines or line pairs. We derived these lines the following way:

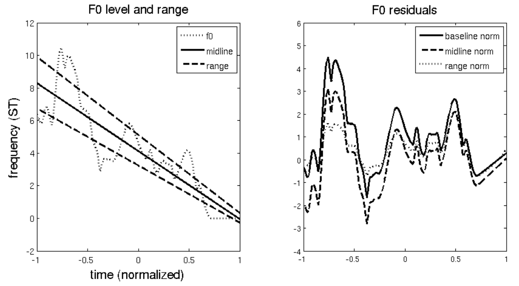

Figure 2: **Left:** *Extraction of a base-, mid- and topline within an intonation phrase to account for F0 level and range varying over time.* **Right:** *F0 residuals after level subtraction (baseline or midline) and range normalization.*

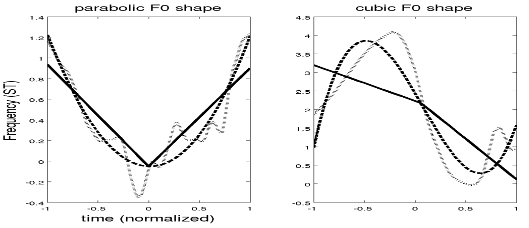

Figure 3: *Third-order polynomial stylization of F0 shapes.* **Left:** *Parabolic shape indicated by the absence of a turning point within the given time interval. The extreme value is selected as the split point* **Right:** *Cubic shape, where the turning point serves as the split point.*

- A third order polynomial was fitted to the F0 contour within an AG. Time was normalized to the interval from -1 to 1.
- The extreme values and the turning point of this polynomial within the normalized time interval were calculated from the polynomial's first and second derivative, respectively.
- If the polynomial neither contains a turning point nor an extreme value in the examined time interval, a **linear** F0 shape can be deduced that can sufficiently be characterized by a single line.
- If the polynomial contains only an extreme value but no turning point, this indicates a **parabolic** F0 shape in the examined time window. The extreme value is taken as the split point for subsequent line pair fitting. An stylization example for parabolic F0 shape is shown in the left panel of Figure 3.
- The existence of a turning point gives indication for a **cubic** shape. Since for these shapes two extreme values can occur in the examined interval, instead of choosing one of them in an ad-hoc manner, we select the turning point as the split point for subsequent line pair fitting. The right panel of Figure 3 shows a stylization example for cubic F0 shapes.
- The split point serves to divide the polynomial into two parts. Separately for the first and the second part a straight line is fitted passing through the split point by means of linear regression with zero intercept.

From this parameterization single line character contours can be described by the slope s of the regression line. Two-line

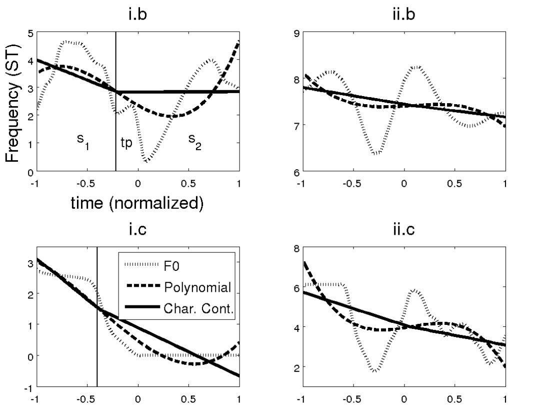

Figure 4: *F0 parameterization: Prototypical examples for each character contour.*

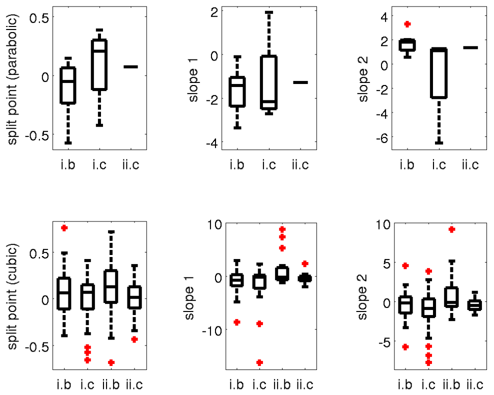

Figure 5: *Parameter distributions for the available character contour classes for parameterizations of parabolic (top row) and cubic shapes (bottom row).*

CCs are represented by a set of three features: sp: the position of the split point within the normalized time interval, s_1: the slope of the regression line form the beginning of the AG to the turning point, and s_2, the regression line from the turning point to the end of the AG. Since in our data only a single F0 shape was identified as linear it was subsumed to the set of parabolic shapes simply treating the line midpoint as the split point. Due to the lack of comparability between the different split point definitions the subsequent examinations and classifications were carried out separately for parabolic and cubic contours. Prototypical character contour stylization examples are given in Figure 4.

3.2. Relation between parameters and character contours

For the parabolic shape parameterizations none of the parameters differed significantly with respect to the character contour. For the cubic shape parameterization in contrast both slopes s_1 and s_2 showed significant differences across the contour classes for the original F0 contours and the residuals resulting from base- and midline subtraction (ANOVAs with s_1 resp. s_2 as dependent and character contour as independent variable. For s_1: $F[3, 113] = 4.08$, $p < 0.01$, and for s_2: $F[3, 113] = 3.51, p < 0.05$). With reference to Figure 1 one would expect more strongly negative slopes for $ii.c$ as opposed to $ii.b$ and for $i.c$ as opposed to $i.b$, but a Tukey-Kramer post-hoc test ($\alpha = 0.05$) only revealed significant differences for both slopes between the major classes i and ii. Nevertheless the expected tendencies can be seen in Figure 5.

4. Automatic Labeling

4.1. Features and preprocessing

We defined two different feature sets: a CC feature set as well as an extended set, which are presented in Table 1. For the extended feature set a sequential feature selection was carried out using the Matlab function *sequentialfs* to find the feature subset which is best in the sense of an optimality criterion. In our case the error to be minimized was calculated from the mean silhouette over all data points. We adopted this cluster evaluation measure to judge the goodness of separability of CC types by the respective feature combination. The mean silhouette $\hat{S}$, whose values range from -1 (bad) to 1 (good separability) was transferred to an error measure e ranging from 0 to 1 by $e = \frac{1-\hat{S}}{2}$. Subsequently the CC features as well as the selected features of the extended set were orthogonalized by means of a principal component analysis.

Table 1: *Feature sets for character contour classification.*

Feature	Description	CC	Extended
sp	position of split point	+	+
s_1	slope of first line	+	+
s_2	slope of second line	+	+
s_d	s_1-s_2	−	+
sp_y	F0 value of split point	−	+
coeff$_{0..3}$	polynomial coefficients	−	+

4.2. Classifiers

Separately for parabolic and cubic contours we trained classification trees (CART) [19] and support vector machines (SVM) [20] with a linear Kernel function to predict the character contour type from the respective feature set. For the training the Matlab functions *classregtree* and *svmtrain* with their default initializations were used.

4.3. Automatic label correction

In a postprocessing step we tried to improve the classification performance by treating the task of automatic labeling as a noisy channel problem: a correct label sequence (the expert annotation C) cannot be observed directly but only in form of a defective channel output O, which is given by the outputs of our classifiers. Thus label correction is basically the same as revealing the most probable hidden label sequence C that underlies the classifier output O: $\hat{C} = \arg\max_C[P(C|O)]$. We addressed this problem by Hidden Markov modeling (HMM). For the transition probabilities a linear interpolated uni- and bigram model was trained. Counts were smoothed by Good-Turing discounting.

5. Results

5.1. Classification accuracies

The accuracies for both classifiers CART and SVM, for the parabolic and cubic F0 contours and all its residuals, and for the CC and the extended feature set are presented in Tables 2 and 3. The accuracies were measured by means of leave-one-out evaluation. It turned out that:

- parabolic contours can be classified with higher accuracy than cubic contours. This can mainly be explained by the fact that parabolic contours were observed only for three of the four contour classes.
- F0 level normalization has a positive impact on CART performance while for the SVM performance rather range normalization is crucial.
- The best results were obtained by a CART trained on the reduced CC feature set and on midline residuals.

Table 2: *Character contour classification accuracies for classification trees.*

	CC feature set		extended feature set	
	parabolic	cubic	parabolic	cubic
original	69.23	29.91	53.85	23.08
baseline residual	76.92	30.77	69.23	36.75
midline residual	**84.62**	**43.59**	69.23	30.77
range residual	62.50	35.96	62.50	39.47

Table 3: *Character contour classification accuracies for support vector machines.*

	CC feature set		extended feature set	
	parabolic	cubic	parabolic	cubic
original	69.23	34.18	53.85	33.33
baseline residual	69.23	36.75	53.85	30.77
midline residual	69.23	35.04	69.23	32.48
range residual	75.00	41.22	68.75	42.98

5.2. Accuracies after error correction by HMMs

We trained HMMs to map the partly erroneous output of our best classifier (the CART classifying F0 midline residuals based on CC features) to the correct labels of the manual annotation.

In a tenfold cross validation it turned out that the performance could be improved with high significance (Mann-Whitney test, $z = -2.5809, p < 0.005$). The result is presented in Figure 6.

6. Discussion and Conclusion

The findings of this study suggest that the application of the character contour framework on spontaneous speech is challenging. First, in our data only four character contour types had been observed which is only a subset of the contour inventory of [1]. This is partly to be explained by the shortcoming that our data does not contain questions, so that end-falling characters are very unlikely. But it also indicates that the other contour types are very unevenly distributed in spontaneous speech (*i.a* and *ii.a* from Figure 1 have not been observed at all). This uneven distribution in spontaneous speech was already observed by [1, p. 52]. In his data 60% of the contours were of type *ii.b*, whereas all *iii*-types taken together only occurred in less than 2% of the accent groups. Second, in spontaneous speech the extracted character contour parameters, slopes and turning point, overlap to a high extent across different contour classes so that it is difficult to relate a realized contour to its underlying prototype.

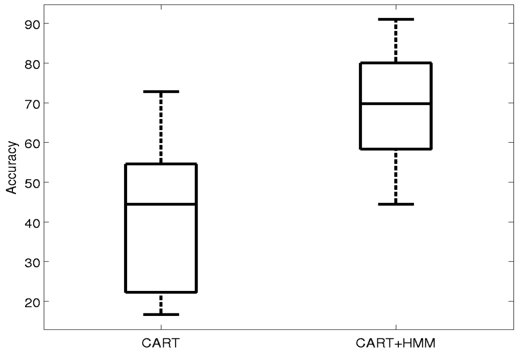

Figure 6: *The classification correction by means of an HMM leads to a significant accuracy improvement.*

Principally, the parameterization proposed here is not strongly affected by the first drawback that not all contour types are contained in our data. The parameterization is extendable to the remaining contours, since for the observed contours their prototypes can be described by means of one or two line slopes.

Automatic labeling in contrast is heavily affected by the second drawback of parameter value overlap making it difficult to distinguish and identify character types in spontaneous speech. Also the usage of additional features as polynomial coefficients did not turn out to be gainful. Thus, the automatic labeling of character contours remains a difficult task.

Nevertheless, a significant improvement was achieved by postprocessing the classifier output with HMMs. From this finding we draw two conclusions. First, since postprocessing accounts for the left label context in terms of transition probabilities, one can infer that context plays a role for label assignment. Neither standard CARTs nor SVMs consider context, if it is not explicitly contained in the feature vectors. Context might be introduced by additional features (which requires more training data than available for this study) or by an appropriate normalization of the given features. The latter would be an interesting issue to be addressed in a follow-up study. Alternatively, one could directly use HMMs for intonation label assignment. However, using HMMs as a separate postprocessing classifier also has its benefits which brings us to our second conclusion: the learnability of correct labels from erroneous ones indicates that there are systematic correspondences between the classification output and reference labels. This finding might be of more general use for applications that automatically correct the output of labeling tools.

Finally, comparing the results between different classifiers and F0 normalization methods, one can see that normalization generally increases classification performance, but not in a uniform way for all classifiers. More systematic examinations are needed to get better insight into these complex relations.

7. References

[1] L. Varga, *Intonation and Stress: Evidence from Hungarian*. Hampshire, New York: Palgrave Macmillan, 2002.

[2] ——, “Prozodémák a magyar beszédben és jelölésük az intonációs átiratban,” in *Műhelymunkák a nyelvészet és társtudományai köréből III*, MTA Nyelvtudományi Intézet, Budapest, 1987, pp. 91–119.

[3] D. Bolinger, “Intonation: Levels Versus Configurations,” *Word*, vol. 7, pp. 199–210, 1951.

[4] M. A. K. Halliday, *Intonation and Grammar in British English*. Den Haag: Mouton, 1967.

[5] J. Pierrehumbert, “The phonology and phonetics of Englisch intonation,” Ph.D. dissertation, MIT, Cambridge, MA, 1980.

[6] S. Ananthakrishnan and S. S. Narayanan, “Automatic Prosodic Event Detection Using Acoustic, Lexical, and Syntactic Evidence,” *IEEE Transactions on Audio, Speech & Language Processing*, vol. 16, no. 1, pp. 216–228, 2008.

[7] S. Rapp, “Automatic labelling of German prosody,” in *Proc. ICSLP*, 1998, pp. 1267–1270.

[8] C. Brindöpke, G. Fink, F. Kummert, and G. Sagerer, “A HMM-based recognition system for perceptive relevant pitch movements of spontaneous German speech,” in *Proc. ICSLP*, Sydney, 1998, pp. 2895–2898.

[9] I. Bulyko and M. Ostendorf, “Joint Prosody Prediction and Unit Selection for Concatenative Speech Synthesis,” in *Proc. ICASSP*, 2001, pp. 781–784.

[10] A. Schweitzer and B. Möbius, “Experiments in Automatic Prosodic Labeling,” in *Proc. Eurospeech*, Brighton, 2009, pp. 2515–2518.

[11] G. Möhler and A. Conkie, “Parametric modeling of intonation using vector quantization,” in *Proc. 3rd ESCA Workshop on Speech Synthesis*, 1998, pp. 311–316.

[12] U. Reichel, “Automatisation of intonation modelling and its linguistic anchoring,” in *Proc. Speech Prosody*, Shanghai, 2012, pp. 63–66.

[13] H. Mixdorff, “An Integrated Approach to Modeling German Prosody,” Ph.D. dissertation, TU Dresden, 2002.

[14] H. Pfitzinger, H. Mixdorff, and J. Schwarz, “Comparison of Fujisaki-model extractors and F0 stylizers,” in *Proc. Interspeech*, Brighton, 2009, pp. 2455–2458.

[15] P. Taylor, “Analysis and Synthesis of Intonation using the Tilt Model,” *Journal of the Acoustical Society of America*, vol. 107, pp. 1697–1714, 2000.

[16] T. Rietveld and P. Vermillion, “Cues for Perceived Pitch Register,” *Phonetica*, vol. 60, pp. 261–272, 2003.

[17] U. Reichel and K. Mády, “Parameterization of F0 register and discontinuity to predict prosodic boundary strength in Hungarian spontaneous speech,” in *Elektronische Sprachsignalverarbeitung*, ser. Studientexte zur Sprachkommunikation, P. Wagner, Ed., vol. 65. Bielefeld: TUDpress, 2013, pp. 223–230.

[18] K. Mády, U. Reichel, and v. Beňuš, “Accentual phrase in languages with fixed word stress: a study on Hungarian and Slovak,” in *Workshop Advancing Prosodic Transcription for Spoken Language Science and Technology II, Phonetics and Phonology in Iberia 2013*, Lisbon, 2013.

[19] L. Breiman, J. Friedman, C. Stone, and R. Olshen, *Classification and Regression Trees*. Pacific Grove, CA.: Wadsworth & Brooks, 1984.

[20] C. Cortes and V. Vapnik, “Support-vector networks,” *Machine Learning*, vol. 20, 1995.