Learning within- and between-word variation in probabilistic OT grammars

Aleksei Nazarov
University of Huddersfield

The existence of within- and between-word variation (Coetzee and Pater 2011) makes learning a phonological grammar more than finding a single constraint ranking compatible with all data points. Rather, this task includes finding a partition of inputs and a range of constraint rankings that together define just the attested range of within- and between-word variation in the language.

For instance, in Modern Hebrew (Temkin-Martinez 2010 = TM), there is a spirantization process that optionally applies post-vocally (within-word variation; (1a)). However, there are also words in which spirantization never applies, (1b), and words in which spirantization always applies, (1c) (TM: Appendix B; forms with p≤0.1 assumed ungrammatical). Evidence from other words shows that all bolded segments are underlyingly stops. Thus, the grammar partitions words into non-undergoing, (1b), always-undergoing, (1c), and default items, (1a). In addition, for default words, it must allow variation between stops and fricatives in just the right environment.

\begin{equation}
\begin{align*}
&\text{(1) } a. \text{/lictov/} \rightarrow \text{[licktov, liktov]} \mid b. \text{/likro/} \rightarrow \text{[likro] *χ} \mid c. \text{/makan/} \rightarrow \text{[mαχa] *k}
\end{align*}
\end{equation}

To deal with within-word variation, learners with probabilistic ranking (Boersma 1998, Jarosz 2015) or weighted constraints (Pater 2009, Goldwater and Johnson 2003) have been proposed. For between-word variation, learners exist (Becker 2009, Coetzee 2009) for inferring lexically indexed constraints (Kraska-Szlenk 1995, Pater 2000). However, these learners depend on the Constraint Demotion family of approaches (Tesar 1995), which cannot learn within-word variation.

Thus, to my knowledge, there is no existing OT learner that can learn both within-word and between-word variation while starting from an unindexed constraint set. I propose exactly such a learner, by showing that the logic of the existing indexed constraint learners (see above) can be implemented in Jarosz’s (2015) probabilistic Expectation Driven Learning (EDL) framework, which represents within-word variation by probabilities over constraint pair rankings:

\begin{equation}
\begin{align*}
&\text{(2) Constraints: } \{A,B,C\} \quad P(A >> B) = 0.7 \quad P(A >> C) = 0.5 \quad P(B >> C) = 0.2
\end{align*}
\end{equation}

Every time the grammar is used, a full ranking is sampled (cf. Boersma 1998) that uses these pairwise probabilities as well as conditional relations between rankings (see Jarosz 2015 for details). The probabilities themselves are learned as follows: given an initial state (in this case, a uniform distribution for each ranking), it iterates the following procedure. For each /input/ and each constraint pair \{A,B\}, it temporarily sets A >> B and then B >> A while keeping all other probabilities constraint, and for each temporary grammar takes a fixed number (r = 50) of samples; for both sets of samples, it determines the number of matches: (3a). From these numbers of matches, it computes expected success counts: (3b). These expected success counts are then turned into probabilities as in (3c).

\begin{equation}
\begin{align*}
&\text{(3) a. } N \text{ of successes in } r \text{ samples from temp. grammar with categorical } A \gg B = \text{Success}_{A \gg B} \\
&\text{N of successes in } r \text{ samples from temp. grammar with categorical } B \gg A = \text{Success}_{B \gg A} \\
&\text{b. Expected success counts: } E[\text{Success} A \gg B] = \text{Success}_{A \gg B} \times P(A >> B) \\
&\quad E[\text{Success} B \gg A] = \text{Success}_{B \gg A} \times [1 - P(A >> B)]
\end{align*}
\end{equation}

\begin{equation}
\begin{align*}
&\text{c. } P(A \gg B)_{\text{dataset}} = \frac{\sum/\text{inputs} / E[\text{Success}_A >> B]}{\sum/\text{inputs} / (E[\text{Success}_A >> B] + E[\text{Success}_B >> A])}
\end{align*}
\end{equation}

It is these pairwise probabilities that make it possible to incorporate the criterion for indexed constraint induction presented in Pater (2010), Becker (2009), and Coetzee (2009). This criterion is that an indexed constraint is induced whenever inconsistency is detected, i.e., whenever two groups of words require two opposite rankings: e.g., *Stop >> Ident vs. Ident >> *Stop.
In EDL, a “soft inconsistency” criterion can be used that approximates this intuition of opposite ranking preferences: if the lexicon assigns at least 60% probability to ranking A >> B, then any word that assigns 40% or less probability to ranking A >> B is inconsistent with the lexicon w.r.t. constraint pair \{A,B\}. At every iteration of the EDL algorithm defined above, the learner finds the constraint pair \{X,Y\} that has the highest total difference \(\sum_{\text{words}} P(X \gg Y)_{\text{lexicon}} - P(X \gg Y)_{w}\), and creates or updates a lexically indexed constraint \(Y_i\) that is indexed to the words inconsistent with respect to the pair \{X,Y\}.

For instance, if the lexicon prefers *Stop >> *Fricative with a probability of 60%, while 3 out of 10 words prefer *Stop >> *Fricative with a probability of 10%, then these 3 words are inconsistent w.r.t. this constraint pair. If \((3 \times (0.6 - 0.1)) = 1.5\) is the highest total difference between exceptions and rule-obeying forms of any constraint pair, these 3 words will be assigned to an indexed constraint *Fricative, which the learner will rank above *Stop given sufficient evidence.

This learner was tested on a simplified abridged version of the Hebrew data: it had only variation in postvocalic position, as in (1) – non-postvocalic variation was leveled. The relative frequencies of each type of variation were estimated from the 29 roots with a single, non-word-final stop in TM’s experimental data, and scaled to a dataset of 12 words.

In 20 runs of up to 80 iterations each with \(r = 50\), the learner reached \(\leq 5\%\) likelihood of error within 11.35 iterations on average. The rate of exceptional words’ being connected to an indexed constraint was 91% (100% for the 3 words like (1c), 65% for the single word like (1b)), while the corresponding rate for a non-exceptional word was 3% (0% for words like (1a)). Furthermore, for default words, the grammar predicted the attested range of within-word variation in postvocalic position with 96% accuracy, and predicted lack of variation in non-postvocalic position with 95% accuracy. Thus, the learner was able to distinguish within-word from between-word variation: words that underwent within-word variation, (1a), were never given an exceptionality index, while words that did not undergo within-word variation post-vocally, (1b,c), were given such an index at an overall rate of 91%.

References