

Supplementary Materials 1.2: Finding the right combination of UR probabilities and constraint weights

For Yang Wang and Bruce Hayes, “Learning phonological underlying representations: the role of abstractness” published in *Linguistic Inquiry*

1. Purpose

This material provides the technical details for a key component of our modeling approach: the calculation of the optimal UR probabilities and phonological constraint weights. As noted, this is one instance of the “hidden structure” problem and as such represents a particular analytic challenge. As in earlier work (e.g. Jarosz 2006, Pater et al. 2012, Cotterell et al. 2015, Johnson et al. 2015, Nazarov and Pater 2017, Nelson 2019, Tan 2022, Pater and Prickett 2022), our approach to hidden structure is based on the *expectation-maximization* algorithm (EM; Dempster et al., 1977).

As noted in the main text, we treat underlying representations (URs) as latent variables, maintaining for each morpheme a probability distribution over all candidate URs, designated as θ . For phonology, we employ Maximum Entropy grammars (MaxEnt; Goldwater and Johnson, 2003); the vector of constraint weights deployed in a grammar is denoted W . EM is applied to find the best values for θ and W .

Our work resembles that of Jarosz and Cotterell et al. in that we employ probability distributions over candidate URs, as opposed to using UR constraints, as in Pater et al. (2012). Our work resembles that of Pater et al. Nelson, and Tan in adopting MaxEnt as the phonological grammar. It is the combination of the two that requires the particular deployment of EM described below.

The rest of this document is organized as follows. In Section 2, we define the objective function employed for model optimization. Section 3 gives a top-down view of how model finds the values for θ and W that optimize this function, and Section 4 derives the essential formulae needed in this process.

2. Defining the objective function

Probability of an SR given a UR. This is computed by the MaxEnt phonological grammar, which consists of a set of constraints C (functions that assign violation counts to UR-SR pairs), along with their weights W . Formula (1) (Goldwater and Johnson 2003:2) uses these constraints and weights to derive SR probabilities given a UR.

(1) *Computing the probability of a surface candidate given an underlying representation*

$$P(s \mid u; W) = \frac{1}{Z} \exp(-\sum_i W_i C_i(u, s))$$

$$\text{where } Z = \sum_{s' \in \text{GEN}(u)} \exp(-\sum_i W_i C_i(u, s'))$$

This formula computes the various expressions seen in tableau (8) in the main text: Harmony (\mathcal{H} ; $\sum_i W_i C_i(u, s)$), $e\mathcal{H}$ ($\exp(-\mathcal{H})$), and Z .

Aggregating the probability of surface forms with multiple URs. When multiple URs are in contention, the probability of a surface form s must be computed by forming a weighted sum of the probabilities of s being derived from any one of these URs; hence (2).

(2) *Probability of a surface candidate for a word*

$$\begin{aligned} P(s \mid \omega; \boldsymbol{\theta}, \mathbf{W}) &= \sum_{u \in \text{UR}(\omega)} P(s, u \mid \omega; \boldsymbol{\theta}, \mathbf{W}) \\ &= \sum_{u \in \text{UR}(\omega)} P(s \mid u; \mathbf{W}) P(u \mid \omega; \boldsymbol{\theta}) \\ &= \sum_{u \in \text{UR}(\omega)} P(s \mid u; \mathbf{W}) \prod_{i=1}^n \theta_{(\mu_i, v_i)} \end{aligned}$$

where u is a possible UR for ω , consisting of the concatenation ($v_1 v_2 v_3 \dots v_n$) of the URs of its morphemes ($\mu_1 \mu_2 \mu_3 \dots \mu_n$).

The expansion of $P(u \mid \omega; \boldsymbol{\theta})$ in the third line expresses the idea that the probability of a candidate UR for a word is the product of the probabilities of the various URs for the morphemes that comprise it. Our use of the KK Hierarchy always yields a finite number of URs for each word, so that the summation in is over a finite discrete set.

Calculating likelihood. The objective function is based on the conditional *likelihood* of the model as applied to the data. This is defined as the product of the probabilities assigned to every observation in D , under a particular setting of the model's parameters.

(3) *Calculating likelihood*

$$P(D \mid \boldsymbol{\theta}, \mathbf{W}) = \prod_{(s, \omega) \in D} P(s \mid \omega; \boldsymbol{\theta}, \mathbf{W})^{f(s, \omega)}$$

where $f(s, \omega)$ is the frequency with which ω is realized as s in the learning data
 $P(s \mid \omega; \boldsymbol{\theta}, \mathbf{W})$ is the probability that the grammar assigns to s as the surface output for ω , as defined in (2).

Instead of maximizing likelihood directly, it is convenient instead to maximize the log of the likelihood, calculated as in (4).

(4) *Defining log likelihood*

$$\ln(P(D \mid \boldsymbol{\theta}, \mathbf{W})) = \sum_{(s, \omega) \in D} f(s, \omega) \ln(P(s \mid \omega; \boldsymbol{\theta}, \mathbf{W}))$$

Lastly, following standard practice, we augment (4) with a regularization term, which serves to avoid infinite weights and overfitting. With this term included, the objective function is now defined as in (5).

(5) *Defining the objective function*

$$L = \sum_{(s, \omega) \in D} f(s, \omega) \ln(P(s \mid \omega; \boldsymbol{\theta}, \mathbf{W})) - \sum_i \frac{(w_i - \mu_i)^2}{2\sigma_i^2}$$

3. Optimizing θ and W with Expectation-Maximization

It is at this stage that the problem of hidden structure is addressed by using Expectation-Maximization. EM is an iterative method that breaks down an optimization task involving hidden structure into a set of smaller steps and alternates between them. The two steps are the Expectation (E) step and the Maximization (M) step.

E-step. The E-step fills in the missing UR information in the input by calculating expected values. Intuitively, it calculates how much “responsibility” each UR u should take for any observed datum (s, ω) . This is done by calculating a posterior probability distribution over the hidden structures, using Bayes’ Theorem, as shown in (6a).

(6) a. *The probability of the UR u , given observation (s, ω)*

$$\begin{aligned} P(u \mid s, \omega) &= \frac{P(s|u, \omega) P(u \mid \omega)}{P(s \mid \omega)} && \text{by Bayes' Theorem} \\ &= \frac{P(s|u)P(u \mid \omega)}{\sum_{u' \in UR(\omega)} P(s, u' \mid \omega)} && \text{by law of total probability} \\ &= \frac{P(s|u) P(u \mid \omega)}{\sum_{u' \in UR(\omega)} P(s|u')P(u' \mid \omega)} \end{aligned}$$

b. *E-step: expected frequencies of a UR in observation (s, ω)*

$$E(u, s, \omega) = f(s, \omega) \frac{P(s|u) P(u \mid \omega)}{\sum_{u' \in UR(\omega)} P(s|u')P(u' \mid \omega)}$$

where $P(s \mid u)$ and $P(s \mid u')$ are as given in (1).

M-steps. During the M-steps, the algorithm obtains a better estimate for the parameters W and θ by maximizing the likelihood of “filled-in” data, based on the guess made at the Expectation step. The calculations for W , which are justified in detail below, are given in (7); those for θ in (8).

(7) *M-step: Using estimated frequencies to calculate constraint weights*

$$W^{t+1} = \operatorname{argmax}_W \left\{ \sum_{(s, \omega) \in D} \sum_{u \in UR(\omega)} E(u, s, \omega) \ln(P(s \mid u; W)) - \sum_i \frac{(w_i - \mu_i)^2}{2\sigma_i^2} \right\}$$

where $P(s \mid u; W)$ is defined in (1)

(8) *M-step: Using estimated frequencies to re-estimate UR probabilities*

$$\theta_{(\mu, v)}^{t+1} = \frac{\sum_{(s, \omega) \in M(\mu)} \sum_{u \in UR(\omega)} E(u, s, \omega)}{\sum_{v' \in UR(\mu)} \sum_{(s, \omega) \in M(\mu)} \sum_{u' \in UR(\omega)} E(u', s, \omega)}$$

where μ is a morpheme, and v is a possible UR for μ
 u, u' are members of the word form UR sets containing v or v'
 $M(\mu)$ is the set of word forms containing μ ,
 E is as defined as in (6b).

The complete learning process. With both the E-step and the M-steps complete, a single iteration is accomplished. The next iteration begins by inputting the new parameter values θ^{t+1} and W^{t+1} to (6b), and the process continues.

The discussion oversimplifies slightly, giving the classical layout of EM. Our own implementation achieves slightly faster convergence by carrying out the E-steps (6) not just before (7) but also before (8); see Meng and Rubin (1993). The overall procedure is this: the first iteration commences by assigning initial values θ^0 and W^0 to θ and W , and the process terminates when an iteration ceases to improve log likelihood by more than some small threshold amount.

Initialization and other details. For θ , all candidate URs generated for a given morpheme (at a specific KK-level) were initially set as equiprobable. For W , following e.g. Hayes and Wilson (2008), Nelson (2019), we started out all constraint weights at 1, and reimposed this default at the start of each EM iteration. For MaxEnt optimization there are multiple algorithms; we employed L-BFGS-B, described in Byrd et al. (1995) and Nazarov and Pater (2017). Optimization was carried out with a Gaussian prior, with μ_i at 0 and $2\sigma_i^2$ at 10^5 for all constraints. Learning was terminated on the first iteration at which log-likelihood increased by less than 10^{-3} .

4. Deriving M-steps

The formulae (7) and (8) above implement the M-steps in the Expectation-Maximization search. We show here that they can be derived starting with (5), the objective function. The derivations follow methods given in McLachlan and Peel (2000: 47-50), Pavlov et al. (2003), and Bishop (2006, ch. 9).

4.1 Deriving (7) from (5): Constraint weights (W)

Equation (5) gives the objective function for learning, repeated below in adapted form as (9). For simplicity we omit prior terms here, since they do not affect the rest of the derivation.

$$\ln(P(D | \theta, W)) = \sum_{(s, \omega) \in D} f(s, \omega) \ln(P(s | \omega; \theta, W)) \quad (9)$$

$$= \sum_{(s, \omega) \in D} f(s, \omega) \ln \left(\sum_{u \in UR(\omega)} P(s | u; W) P(u | \omega; \theta) \right) \quad (10)$$

We know that at the maximum value, the partial derivatives of log likelihood with respect to all model parameters are zero, as stated in (11).

$$\frac{\partial}{\partial W} \ln(P(D | \theta, W)) = 0 \quad (11)$$

For a particular constraint weight W_j , this partial derivative can be computed as follows.

$$\frac{\partial}{\partial W_j} \ln(P(D | \theta, W)) = \sum_{(s, \omega) \in D} f(s, \omega) \frac{\partial}{\partial W_j} \ln \left(\sum_{u \in UR(\omega)} P(s | u; W) P(u | \omega; \theta) \right)$$

$$\begin{aligned}
\text{chain rule} \quad &= \sum_{(s,\omega) \in D} f(s, \omega) \frac{1}{\sum_{u'} P(s | u'; \mathbf{W}) P(u' | \omega; \boldsymbol{\theta})} \frac{\partial}{\partial W_j} \left(\sum_{u \in UR(\omega)} P(s | u; \mathbf{W}) P(u | \omega; \boldsymbol{\theta}) \right) \\
\text{sum rule} \quad &= \sum_{(s,\omega) \in D} f(s, \omega) \sum_{u \in UR(\omega)} \frac{P(u | \omega; \boldsymbol{\theta})}{\sum_{u'} P(s | u'; \mathbf{W}) P(u' | \omega; \boldsymbol{\theta})} \left(\frac{\partial}{\partial W_j} P(s | u; \mathbf{W}) \right) \\
(12) \quad &= \sum_{(s,\omega) \in D} \sum_{u \in UR(\omega)} \boxed{f(s, \omega) \frac{P(s | u; \mathbf{W}) P(u | \omega; \boldsymbol{\theta})}{\sum_{u'} P(s | u'; \mathbf{W}) P(u' | \omega; \boldsymbol{\theta})}} \frac{1}{P(s | u; \mathbf{W})} \left(\frac{\partial}{\partial W_j} P(s | u; \mathbf{W}) \right)
\end{aligned}$$

The boxed term in (12) represents the expected counts of each UR u for a given (s, ω) pair, since the fraction within the box represents the responsibility of the UR u in deriving the surface form s for a given word ω . Based on the definition (6b) above we can rewrite the right-hand side of equation (12) as follows.

$$\begin{aligned}
\frac{\partial}{\partial W_j} \ln(P(D | \boldsymbol{\theta}, \mathbf{W})) &= \sum_{(s,\omega) \in D} \sum_{u \in UR(\omega)} E(u, s, \omega) \frac{1}{P(s | u; \mathbf{W})} \left(\frac{\partial}{\partial W_j} P(s | u; \mathbf{W}) \right) \\
&= \sum_{(s,\omega) \in D} \sum_{u \in UR(\omega)} E(u, s, \omega) \left(\frac{\partial}{\partial W_j} \ln(P(s | u; \mathbf{W})) \right) \quad (13)
\end{aligned}$$

It is formula (13) that makes it possible to deploy the iterative method EM: one first fixes the parameter values of $\boldsymbol{\theta}$ and \mathbf{W} to calculate $E(u, s, \omega)$, and the resulting expected values are then used to re-estimate \mathbf{W} , as in (14).

$$\mathbf{W}^{t+1} = \text{argmax}_{\mathbf{W}} \sum_{(s,\omega) \in D} \sum_{u \in UR(\omega)} E(u, s, \omega) \ln(P(s | u; \mathbf{W})) \quad (14)$$

With (14) we have derived the result stated in (7) above, excepting that we are omitting the prior term.

4.2. Deriving (8) from (5): UR parameters (θ)

We can perform a similar calculation to derive (8) (best-fit values for UR probabilities) from (5). The starting point is (10), repeated below:

$$\ln(P(D | \boldsymbol{\theta}, \mathbf{W})) = \sum_{(s,\omega) \in D} f(s, \omega) \ln \left(\sum_{u \in UR(\omega)} P(s | u; \mathbf{W}) P(u | \omega; \boldsymbol{\theta}) \right) \quad (10)$$

The probability of a UR of a word is the product of the probabilities of the morphemes of which it is comprised. That is, if $\omega = \mu_1 \mu_2 \mu_3 \dots \mu_n$ and $u = v_1 v_2 v_3 \dots v_n$, we can re-express (10) as (15):

$$\ln(P(D | \boldsymbol{\theta}, \mathbf{W})) = \sum_{(s,\omega) \in D} f(s, \omega) \ln \left(\sum_{u \in UR(\omega)} P(s | u; \mathbf{W}) \prod_{i=1}^n \theta_{(\mu_i, v_i)} \right) \quad (15)$$

We also know that the probabilities for all the possible URs of any given morpheme sum to 1; that is, for each morpheme μ , $\sum_v \theta_{(\mu, v)} = 1$. Using the method of the Lagrange multipliers (see e.g. Chong and Zak 2001: 374-379), we can set up the following objective to maximize.

$$\mathcal{L}'(\boldsymbol{\theta}, \mathbf{W}, \boldsymbol{\lambda}) = \ln(P(D | \boldsymbol{\theta}, \mathbf{W})) + \sum_{\mu} \lambda_{\mu} [\sum_v \theta_{(\mu, v)} - 1] \quad (16)$$

This time, we need to find partial derivatives for each value in θ , designated as $\theta_{(\mu,v)}$. We start by substituting (15) into (16):

$$\mathcal{L}'(\theta, \mathbf{W}, \lambda) = \sum_{(s,\omega) \in D} f(s, \omega) \ln\left(\sum_{u \in UR(\omega)} P(s | u; \mathbf{W}) \prod_{i=1}^n \theta_{(\mu_i, v_i)}\right) + \sum_{\mu} \lambda_{\mu} [\sum_v \theta_{(\mu,v)} - 1]$$

Then, we differentiate \mathcal{L}' by $\theta_{(\mu,v)}$:

$$\frac{\partial \mathcal{L}'}{\partial \theta_{(\mu,v)}} = \sum_{(s,\omega) \in D} f(s, \omega) \frac{\partial}{\partial \theta_{(\mu,v)}} \ln\left(\sum_{u \in UR(\omega)} P(s | u; \mathbf{W}) \prod_{i=1}^n \theta_{(\mu_i, v_i)}\right) + \lambda_{\mu}$$

The remaining steps are parallel to (11)-(12) above:

$$\begin{aligned} \frac{\partial \mathcal{L}'}{\partial \theta_{(\mu,v)}} &= \sum_{(s,\omega) \in D} f(s, \omega) \frac{1}{\sum_{u' \in UR(\omega)} P(s | u'; \mathbf{W}) P(u' | \omega; \theta)} \frac{\partial}{\partial \theta_{(\mu,v)}} \left(\sum_{u \in UR(\omega)} P(s | u; \mathbf{W}) \prod_{i=1}^n \theta_{(\mu_i, v_i)}\right) + \lambda_{\mu} \quad \text{chain rule} \\ &= \sum_{(s,\omega) \in D} f(s, \omega) \sum_{u \in UR(\omega)} \frac{P(s | u; \mathbf{W})}{\sum_{u' \in UR(\omega)} P(s | u'; \mathbf{W}) P(u' | \omega; \theta)} \left(\frac{\partial}{\partial \theta_{(\mu,v)}} \prod_{i=1}^n \theta_{(\mu_i, v_i)}\right) + \lambda_{\mu} \quad \text{sum rule} \\ &= \sum_{(s,\omega) \in M(\mu)} f(s, \omega) \sum_{u \in UR(\omega)} \frac{P(s | u; \mathbf{W})}{\sum_{u' \in UR(\omega)} P(s | u'; \mathbf{W}) P(u' | \omega; \theta)} \frac{P(u | \omega; \theta)}{\theta_{(\mu,u)}} + \lambda_{\mu} \\ &= \sum_{(s,\omega) \in M(\mu)} \sum_{u \in UR(\omega)} \boxed{f(s, \omega) \frac{P(s | u; \mathbf{W}) P(u | \omega; \theta)}{\sum_{u'} P(s | u'; \mathbf{W}) P(u' | \omega; \theta)}} \frac{1}{\theta_{(\mu,v)}} + \lambda_{\mu} \end{aligned} \quad (17)$$

Following the same practice above, given that the fraction in the boxed term is the posterior probability that s is derived from u , based on the definition (6b), the right-hand side of (17) can be further expanded as follows:

$$\frac{\partial \mathcal{L}'}{\partial \theta_{(\mu,v)}} = \sum_{(s,\omega) \in M(\mu)} \sum_{u \in UR(\omega)} E(u, s, \omega) \frac{1}{\theta_{(\mu,v)}} + \lambda_{\mu} = 0 \quad (18)$$

For any morpheme μ , all such equations for the UR parameters share the same λ_{μ} term. Solving them together leads us to the following value:

$$\theta_{(\mu,v)} = \frac{\sum_{(s,\omega) \in M(\mu)} \sum_{u \in UR(\omega)} E(u, s, \omega)}{\sum_{v' \in UR(\mu)} \sum_{(s,\omega) \in M(\mu)} \sum_{u' \in UR(\omega)} E(u', s, \omega)}$$

where u, u' are members of the word form UR sets containing v or v'

$M(\mu)$ is the set of word forms containing μ

Again, this formula is part of the EM iterative method to find the solution of θ : one can calculate the expected values $E(u, s, \omega)$ by fixing the parameters θ and \mathbf{W} ; then those expected values can be used for better estimates. Applied at any stage t , we obtain the following.

$$\theta_{(\mu,v)}^{t+1} = \frac{\sum_{(s,\omega) \in M(\mu)} \sum_{u \in UR(\omega)} E(u, s, \omega)}{\sum_{v' \in UR(\mu)} \sum_{(s,\omega) \in M(\mu)} \sum_{u' \in UR(\omega)} E(u', s, \omega)}$$

where u, u' are members of the word form UR sets containing v or v'

$M(\mu)$ is the set of word forms containing μ

This is identical to (8) above.

References

- Bishop, Christopher M. 2006. *Pattern Recognition and Machine Learning* (Information Science and Statistics). Springer-Verlag, Berlin, Heidelberg.
- Byrd, Richard H., Peihuang Lu, Jorge Nocedal, and Ciyu Zhu. 1995. A limited memory algorithm for bound constrained optimization. *SIAM Journal on Scientific Computing* 16. 1190–1208.
- Chong, Edwin. K. P., and Zak, Stanislaw H. 2001. *An Introduction to Optimization* (2nd ed.) Wiley-Interscience Series in Discrete Mathematics and Optimization, John Wiley & Sons, Inc.
- Cotterell, Ryan, Nanyun Peng and Jason Eisner. 2015. Modeling word forms using latent underlying morphs and phonology. *Transactions of the Association for Computational Linguistics*, 3:433–447.
- Dempster, Arthur P., Nan M. Laird, and Donald B. Rubin. 1977. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)* 39:1–22.
- Goldwater, Sharon and Mark Johnson. 2003. Learning OT constraint rankings using a maximum entropy model. In Jennifer Spenader, Anders Eriksson & Östen Dahl (eds.), *Proceedings of the Stockholm Workshop on Variation within Optimality Theory*, 111–120. Stockholm: Stockholm University.
- Hayes, Bruce and Colin Wilson. 2008. A maximum entropy model of phonotactics and phonotactic learning. *Linguistic Inquiry* 39:379–440.
- Jarosz, Gaja. 2006. Rich lexicons and restrictive grammars: Maximum likelihood learning in Optimality Theory. Doctoral dissertation, Johns Hopkins University.
- Johnson, Mark, Joe Pater, Robert Staubs, and Emmanuel Dupoux. 2015. Sign constraints on feature weights improve a joint model of word segmentation and phonology. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 303–313. Denver: Association for Computational Linguistics.
- McLachlan, Geoffrey J. and David Peel. 2000. *Finite Mixture Models*. Wiley Series in Probability and Statistics, John Wiley & Sons, Inc.
- Meng, Xiao-Li, and Donald B. Rubin. 1993. Maximum likelihood estimation via the ECM algorithm: A general framework. *Biometrika* 80:267–78.
- Nazarov, Aleksei, and Joe Pater. 2017. Learning opacity in stratal maximum entropy grammar. *Phonology* 34:299–324.
- Nelson, Max. 2019. Segmentation and UR acquisition with UR constraints. *Proceedings of the Society for Computation in Linguistics*: Vol. 2, Article 8.
- Pater, Joe, and Brandon Prickett. 2022. Typological gaps in iambic nonfinality correlate with learning difficulty. In *Proceedings of the Annual Meetings on Phonology*, vol. 9.
- Pavlov, Dmitry, Alexandrin Popescul, David M. Pennock, and Lyle H. Ungar. 2003. Mixtures of conditional maximum entropy models. In *Proceedings of the 20th International Conference on Machine Learning (ICML-03)*, 584–591.

Tan, Adeline. 2022. Concurrent hidden structure & grammar learning. In *Proceedings of the Society for Computation in Linguistics*: Vol. 5, Article 5.