Capturing gradience in long-distance phonology using probabilistic tier-based strictly local grammars

Connor Mayer
Department of Linguistics
University of California, Los Angeles
Los Angeles, CA 90046, USA
`connormayer@ucla.edu`

Abstract

Phonological processes often exhibit *gradience*, both in response frequencies and in acceptability judgments. This paper presents a variation of tier-based strictly local grammars, *probabilistic tier-based strictly local* (pTSL) grammars, which calculate the conditional probability that a given input string has some grammatical projection. pTSL grammars are well-suited to modeling gradience, particularly for long-distance processes, and naturally extend categorical tier-based strictly local grammars by probabilizing the projection function. After describing the formal properties of pTSL, I illustrate its application using data from Hungarian and Uyghur. pTSL is able to capture distance-based decay in these languages without an explicit notion of distance, and provides a unified account of gradient blocking and distance-based decay. I finish by outlining some of the limitations of pTSL, and how further extensions may overcome these.

1 Introduction

Subregular phonology attempts to find proper subclasses of the finite-state languages and transductions that are sufficiently powerful to model natural language phenomena (see Heinz, 2018). These models provide a strong mathematical foundation for phonological analysis, establish tighter bounds on the range of observed cross-linguistic variation, and have implications for theories of phonological learning (e.g., Lai, 2015; McMullin, 2016; McMullin and Hansson, 2019).

The class of *tier-based strictly local languages* (TSL; Heinz et al., 2011) has proven useful for modeling long-distance phonotactic phenomena

that more restricted classes like the *strictly local languages* (SL) cannot. Grammars that generate the TSL languages remove all symbols from input strings that do not belong to a specified subset of the alphabet before identifying phonotactic violations, allowing non-local dependencies to be regulated in a local manner.

Long-distance phonology frequently exhibits *gradience*, both in response frequencies and in speaker acceptability judgments (e.g., Albright and Hayes, 2003; Daland et al., 2011; Zuraw and Hayes, 2017, a.o.). This gradience often manifests as *distance-based decay*, where long-distance dependencies hold less strongly as the amount of irrelevant material between relevant segments increases. While several accounts of distance-based decay (e.g., Kimper, 2011; Zymet, 2014) have been presented in the framework of Optimality Theory (Prince and Smolensky, 1993/2004), to my knowledge none have been presented using models from subregular phonology.

This paper presents *probabilistic tier-based strictly local* (pTSL) grammars, a natural extension of TSL that probabilizes the projection function used to construct tier representations. This allows the conditional probability of any grammatical projection given an input string to be assigned in a way that permits gradience in long-distance patterns to be captured without an explicit notion of distance (see McMullin, n.d., for a similar observation) and presents a unified account of distance-based decay with other types of blocking. This method of probabilizing the projection function may also be applied with minimal modifications to other extensions of TSL, such as MTSL and SS-TSL/IO-TSL (e.g., De Santo and Graf, 2017; Graf and Mayer, 2018).

The paper is structured as follows. Section 2 provides background on the SL and TSL classes. Section 3 defines pTSL and discusses some of its

---

properties, including its relationship to TSL. Sections 4 and 5 apply pTSL models to vowel harmony in Hungarian and Uyghur, demonstrating that these models can effectively capture speaker judgments and production frequencies in an interpretable way, while also discussing some shortcomings of the model. Section 6 summarizes the paper and proposes several natural extensions of pTSL that overcome its limitations.

2 Background

2.1 Strictly local languages

ε denotes the empty string and S^* the Kleene closure of the set S. S^k denotes the proper subset of S^* that only contains strings of length k, and s^k represents a string consisting of k occurrences of the symbol s.

Let Σ be some fixed alphabet and $s \in \Sigma^*$. The set $f_k(s)$ of k-*factors* of s consists of all the length-k substrings of $\rtimes^{k-1} s \ltimes^{k-1}$, where $\rtimes, \ltimes \notin \Sigma$ and $k \geq 1$. For example, $f_2(ababac) = \{\rtimes a, ab, ba, ac, c\ltimes\}$.

Definition 1. A *strictly k-local grammar* is a set $G \subseteq (\Sigma \cup \{\rtimes, \ltimes\})^k$. A stringset $L \subseteq \Sigma^*$ is *strictly k-local* (SL-k) iff there is some strictly k-local grammar G such that $L = \{s \in \Sigma^* \mid f_k(s) \cap G = \emptyset\}$.

Intuitively, G defines a *grammar* of forbidden substrings that no well-formed string may contain. The class SL of strictly local stringsets is $\bigcup_{k \geq 1}$ SL-k.

Example 1. Consider a language with the following stress pattern: (a) words must have primary stress on the final syllable; (b) words must contain exactly one syllable with primary stress.

Assuming $\Sigma := \{\acute{\sigma}, \sigma\}$, this pattern can be generated using the SL-2 grammar $G := \{\acute{\sigma}\acute{\sigma}, \acute{\sigma}\sigma, \sigma\ltimes, \rtimes\ltimes\}$, and thus is SL-2. For example, the string $\acute{\sigma}\sigma$ is illicit because $f_2(\acute{\sigma}\sigma) \cap G = \{\acute{\sigma}\sigma, \sigma\ltimes\} \neq \emptyset$, whereas the string $\sigma\acute{\sigma}$ is licit because $f_2(\sigma\acute{\sigma}) \cap G = \emptyset$.

2.2 Tier-based strictly local languages

For every $T \subseteq \Sigma$, a *simple tier projection* π_T is a transduction that deletes all symbols not in T:

$$\pi_T(\varepsilon) := \varepsilon$$

$$\pi_T(\sigma u) := \begin{cases} \sigma \pi_T(u) & \text{if } \sigma \in T \\ \pi_T(u) & \text{otherwise} \end{cases}$$

where $\sigma \in \Sigma$ and $u \in \Sigma^*$.

Definition 2. A *tier-based strictly k-local* (TSL-k) grammar over an alphabet Σ is a tuple (T, G), where $T \subseteq \Sigma$ and $G \subseteq (T \cup \{\rtimes, \ltimes\})^k$. A stringset $L \subseteq \Sigma^*$ is TSL-k iff there exists a TSL-k grammar such that $L = \{s \in \Sigma^* \mid f_k(\pi_T(s)) \cap G = \emptyset\}$. It is TSL iff it is TSL-k for some k.

In other words, TSL languages are string languages that are SL once one masks out all irrelevant symbols or, alternatively, languages that are SL over a *tier* to which a subset of relevant symbols are projected (cf. Goldsmith, 1976).

Example 2. Consider a language over the alphabet $\Sigma := \{\acute{\sigma}, \sigma\}$ such that no word may contain more than one primary stress (i.e., more than one $\acute{\sigma}$). This language cannot be characterized by any SL-k grammar: for example, for any value of k we can produce strings of the form $\acute{\sigma}\sigma^{k-1}\acute{\sigma}$, which violate the restriction on multiple primary stresses but cannot be prohibited by a SL-k grammar because the window of length k is not large enough to "see" both stresses at the same time.

We can define a TSL-2 grammar that accepts this language. Let $T := \{\acute{\sigma}\}$ and $G := \{\rtimes\ltimes, \acute{\sigma}\acute{\sigma}\}$. Any illicit string of the form $\acute{\sigma}\sigma^*\acute{\sigma}$, for instance, will first be projected to $\acute{\sigma}\acute{\sigma}$. This projection will be rejected because $f_2(\acute{\sigma}\acute{\sigma}) \cap G = \{\acute{\sigma}\acute{\sigma}\} \neq \emptyset$.

3 Probabilistic tier-based strictly local languages

3.1 Probabilistic tier projection functions

Definition 3. A discrete probabilistic function $f : X \to (Y \to [0, 1])$ assigns to each $x \in X$ a conditional probability distribution over the set Y. For a particular x, $\sum_{y \in Y} f(x)(y) = 1$.

The simple tier projection function π_T discussed in the previous section can be generalized to a *probabilistic tier projection function* $\pi_P : \Sigma^* \to (\Sigma^* \to [0, 1])$. This is a discrete probabilistic function that, given a string, returns a probability distribution over projections of that string to a tier. π_T can be thought of as a special case of π_P where a single output has a probability of 1 and all other outputs have a probability of 0.

The distribution over outputs given an input is calculated based on probabilities associated with projecting individual symbols. Let $P_{proj} : \Sigma \to [0, 1]$ represent the probability that a symbol in Σ is projected to the tier: for example, $P_{proj}(a) := 0.5$ indicates that there is a 50% chance that each a symbol will be projected.

We may notate a sequence of symbols x as $(x_n)_{n \in I}$ where I is the *index set* of the sequence. A sequence y that is a subsequence of x can be written $(y_n)_{n \in J}$ where $J \subseteq I$. Using this notation, we can define the probability of projecting a particular subsequence $y = (y_n)_{n \in J}$ from the input $x = (x_n)_{n \in I}$ as follows:

$$\pi_P(x)(y) := \prod_{k \in J} P_{proj}(x_k) \cdot \prod_{k \in I \setminus J} [1 - P_{proj}(x_k)]$$

That is, the probability of projecting the output y given the input x is the product of the probabilities associated with projecting each symbol that is projected and with not projecting each symbol that is not projected. Any projection that is not a subsequence of x will receive a probability of zero.

The probabilities of all possible projections for an input string x sum to one:

$$\sum_{y \in \Sigma^*} \pi_P(x)(y) = 1$$

Example 3. Let $\Sigma := \{a, b, c\}$, and assume that π_P is defined using the following projection probabilities:

$$P_{proj}(a) := 1.0$$
$$P_{proj}(b) := 0.5$$
$$P_{proj}(c) := 1.0$$

For the input $abbc$ the probability of projecting ac, $\pi_P(abbc)(ac)$, is

$$= P_{proj}(a)[1 - P_{proj}(b)][1 - P_{proj}(b)]P_{proj}(c)$$
$$= 1.0 \cdot 0.5 \cdot 0.5 \cdot 1.0$$
$$= 0.25$$

The complete distribution over possible projections of $abbc$ is:

$$\pi_P(abbc)(abbc) = 0.25$$
$$\pi_P(abbc)(abc) = 0.5$$
$$\pi_P(abbc)(ac) = 0.25$$

All other projections have probabilities of zero.

3.2 Probabilistic tier-based strictly local grammars

Definition 4. A *probabilistic tier-based strictly k-local* (pTSL-k) grammar over an alphabet Σ is a tuple (π_P, G), where π_P is a probabilistic projection function with specified projection probabilities for each $s \in \Sigma$ and $G \subseteq (\Sigma \cup \{\rtimes, \ltimes\})^k$ is a set of prohibited k-factors.

Definition 5. The function $val_{(\pi_P, G)}$ computes the probability that is assigned to a input string x by the corresponding pTSL-k grammar. $val_{(\pi_P, G)}(x)$ is defined as

$$\sum_{y : f_k(y) \cap G = \emptyset} \pi_P(x)(y)$$

In other words, the probability computed by $val_{(\pi_P, G)}(x)$ is the sum of the probabilities of all possible subsequences (or projections) of the input string x that do not contain a prohibited k-factor. Note that val does not constitute a probability distribution over input strings: in general $\sum_{x \in \Sigma^*} val_{(\pi_P, G)}(x) \neq 1$. Instead, val may be interpreted as the conditional probability of any well-formed tier projection given the input.

Example 4. Assume a pTSL-2 grammar defined over the alphabet $\Sigma := \{a, b, c\}$. Let π_P be defined as in Example 3 and $G := \{ac\}$. $val_{(\pi_P, G)}(abbc) = 0.75$, because the sum of the probabilities of all projections of abc that do not contain the 2-factor ac is $0.25 + 0.5 = 0.75$.

3.3 Some properties of pTSL

A stringset $L \subseteq \Sigma^*$ is pTSL-k iff there is some pTSL-k grammar (π_P, G) such that $L = \{w \in \Sigma^* | val_{(\pi_P, G)}(w) > 0\}$. Alternatively, we may say that L is *accepted* by this grammar. The set of pTSL stringsets is the union of all pTSL-k stringsets where $k > 0$.

It is straightforward to show that for every L that is TSL, it is possible to define a pTSL grammar that accepts L. This relationship does not hold in the opposite direction: given a pTSL stringset L, it is not always possible to construct a TSL grammar that accepts L, though certain subclasses of pTSL will always have corresponding TSL grammars. Thus TSL is a proper subset of pTSL. See Appendix A for additional discussion.

3.4 Relating pTSL probabilities to production frequencies

Studies of gradience in phonology typically use as empirical data either speaker ratings of individual forms (e.g., Albright and Hayes, 2003; Daland et al., 2011) or response frequencies of particular word forms (Hayes and Londe, 2006; Zuraw and Hayes, 2017). To model empirical data using pTSL, we must be explicit about how the conditional probabilities assigned to inputs relate to these measurements. In the case of word ratings,

it seems sensible to suggest that assigned probabilities should be positively correlated with ratings. The case of response frequencies is more difficult. Consider a case where a particular word occurs in two forms: y_1 or y_2 (for example, with the front or back form of a suffix under a vowel harmony system). It will not in general be the case that $val_{(\pi_P,G)}(y_1) + val_{(\pi_P,G)}(y_2) = 1$, so the probabilities supplied by the model cannot be treated as response frequencies.

For a pTSL grammar (π_P, G), I relate probabilities assigned by $val_{(\pi_P,G)}$ to response frequencies as follows:

$$P(y_1) := \frac{val_{(\pi_P,G)}(y_1)}{val_{(\pi_P,G)}(y_1) + val_{(\pi_P,G)}(y_2)}$$
$$P(y_2) := 1 - P(y_1)$$

The next two sections provide some empirical justification for this assumption and use pTSL to analyze two cases of gradient, long-distance phonological processes.

4 Gradient transparency in Hungarian vowel harmony

The Hungarian vowel harmony system requires suffix vowels to match the backness of the final front (/y y: ø ø:/; Table 1) or back (/u u: o o: ɔ a:/; Table 2) vowel in the stem (e.g., Hayes and Londe, 2006; Hayes et al., 2009).

yʃt-n**ɛ**k/*-nɔk	'cauldron-DAT'
sɛm**ø**ltʃ-n**ɛ**k/*-nɔk	'wart-DAT'
ʃof**ø**:r-n**ɛ**k/*-nɔk	'chauffeur-DAT'

Table 1: Simple front harmonizing forms (Hayes et al., 2009, p. 829)

ɔbl**ɔ**k-n**ɔ**k/*-nɛk	'window-DAT'
bi:r**o**-n**ɔ**k/*-nɛk	'judge-DAT'
glyk**o:**z-n**ɔ**k/*-nɛk	'glucose-DAT'

Table 2: Simple back harmonizing forms (Hayes et al., 2009, p. 829)

The front unrounded vowels /i i: e: ɛ/ are *transparent* to harmony, meaning that they do not serve as harmony triggers for suffixes, but allow the harmonic values of preceding segments to "pass through" them. Stems with only transparent vowels generally take front suffixes (Table 3), but a small set takes back suffixes (Table 4). Stems with

front vowels followed by transparent vowels invariably take front suffixes (Table 5), while stems with back vowels followed by transparent vowels vary in whether they take front or back suffixes (Table 6).

kɛrt-nɛk/*-nɔk	'garden-DAT'
tsi:m-nɛk/*-nɔk	'address-DAT'
rɛpɛs-nɛk/*-nɔk	'splinter-DAT'

Table 3: Transparent forms that take front suffixes (Hayes et al., 2009, p. 830)

hi:d-nɔk/*-nɛk	'bridge-DAT'
ʃi:p-nɔk/*-nɛk	'whistle-DAT'
dɛre:k-nɔk/*-nɛk	'waist-DAT'

Table 4: Transparent forms that take back suffixes (Hayes et al., 2009, p. 830)

fysɛr-n**ɛ**k/*-nɔk	'spice-DAT'
ø:rizɛt-n**ɛ**k/*-nɔk	'custody-DAT'

Table 5: Front harmonizing forms with transparent vowels (Hayes et al., 2009, p. 829)

dome:n-nɔk/-nɛk	'domain-DAT'
bohe:m-nɔk/-nɛk	'easy going-DAT'
honve:d-nɔk/-nɛk	'Hungarian soldier-DAT'
poe:n-nɔk/-nɛk	'punch line-DAT'

Table 6: Back harmonizing forms with transparent vowels (Hayes et al., 2009, p. 830)

The variation in forms like those in Table 6 is sensitive to the height and count of transparent vowels: harmony from the back trigger is more likely to be blocked (and a front suffix attached) if the intervening transparent vowels are lower (i.e., /i/ and /i:/ are less likely to block than /e:/, which is less likely to block than /ɛ/), and as their number increases (Hayes and Londe, 2006; Hayes et al., 2009).

As part of a broader study, Hayes et al. (2009) administered an online wug test (Berko, 1958) to 131 Hungarian speakers. Participants were presented with 13 wug words from a set of about 1600 wug words embedded in frame paragraphs and asked to choose the form of the dative suffix to attach (front [-nɛk] or back [-nɔk]). Participants were then asked to rate each form and suffix combination on a scale of 1-7, with 7 being the best.

Each wug word belonged to one of the following templates: BN, BNN, and N, where each B and N were sampled from the set of back and transparent vowels respectively according to their lexical frequencies. For simplicity, I treat consonants as completely transparent, ignoring the effects of final consonant on suffix choice observed by Hayes et al. (2009).

To test whether the equation relating word ratings to response frequencies given in Section 3.4 holds for real data, I calculated the predicted proportion of back suffixes per stem type based on speakers' ratings of front and back suffixed forms. The correlation between the proportion of back responses predicted using this method and the proportion observed was extremely high ($r = 0.99$), suggesting that this method is a good characterization of the link between ratings and response frequencies.

To test whether pTSL can adequately model this data, I defined a pTSL-2 grammar (π_P, G) over $\Sigma := \{B, I, \text{e:}, \text{e}, S_f, S_b\}$, where $I := \{i, i :\}$, $\text{e} = \varepsilon$ and S_f and S_b are front and back suffixes. Note that although TSL (and hence pTSL, for the same reasons discussed in de Santo and Graf, 2017) is not in general closed under relabeling, it is closed if no segment corresponds to more than one abstract symbol. This is the case here, and hence the relabelings used here and in the next section do not influence the expressiveness of the models.

The set of prohibited 2-factors G was defined to be $\{BS_f, IS_b, \text{e:}S_b, \text{e}S_b\}$. P_{proj} was fixed to 1 for $\{F, B, S_f, S_b\}$. Maximum likelihood optimization was used to calculate the values of P_{proj} for the remaining symbols using the `minimize` function from the Python library `scipy.optimize` (Virtanen et al., 2020). Optimization was performed 100 times with random starting probabilities. The difference in maximum log likelihood between the best and worst fits was less than 10^{-6}, and the differences between optimal projection probabilities similarly small, indicating that the search space is largely convex. The optimal values were approximately:

$$P_{proj}(\text{I}) = 0.39$$
$$P_{proj}(\text{e:}) = 0.66$$
$$P_{proj}(\text{e}) = 0.82$$

These probabilities directly reflect the tendencies of these segments to act as harmony blockers.
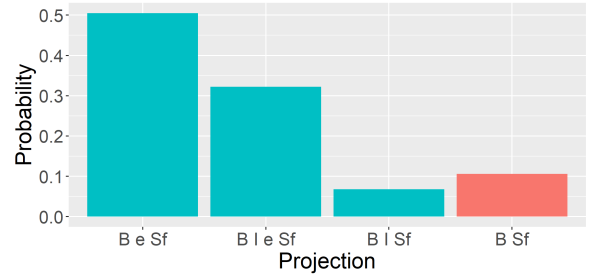

Figure 1: Probability distribution over projections of $BI\text{e}S_f$. Ungrammatical projections are colored red. $val_{(\pi_P, G)}(BI\text{e}S_f) = 0.895$, which is the sum of the probabilities of the grammatical projections. Note that the probability of a particular projection is independent of whether or not it is grammatical: here, ungrammatical BS_f has a higher probability of being projected than grammatical BIS_f.

Fig. 1 shows the probability distribution over projections of the input form $B\text{e:}IS_f$, demonstrating how $val_{(\pi_P, G)}$ calculates its probabilities.

Fig. 2 plots the proportion of back suffix responses predicted by the model against the observed proportions in wug tests. The correlation between the two is strong ($r = 0.83$). The correlation between probabilities assigned to each word by the model and human ratings is also strong ($r = 0.88$). Note that although the predicted frequencies of the model differ somewhat from the observed responses, it captures the observed effects of height and count on responses. Direct comparison between the maximum entropy Optimality Theory models fit to wug test data in Hayes et al. (2009) and this model is difficult for several reasons: Hayes et al. fit their data to individual forms rather than averages over templates, include constraints that model stem-final consonant effects on vowel harmony, and do not present predicted frequency data. I note, however, that the authors report a correlation of $r = 0.575$ between predicted and observed frequencies, which suggests this model may be more successful in capturing height and count effects on vowel harmony.

Note that the pTSL model has the least success predicting the frequency of neutral forms with back suffixes: it predicts they should be more common than they are. The final section of this paper discusses how pTSL may be extended to capture this phenomenon.
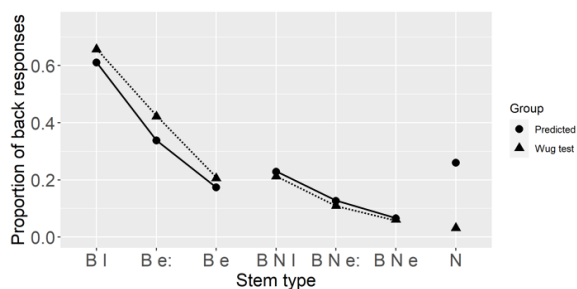
Figure 2: Observed against predicted proportion of back responses by stem template.

5 Gradient judgments in Uyghur backness harmony

Like Hungarian, Uyghur (Turkic: China) displays backness harmony (e.g., Lindblad, 1990; Hahn, 1991a,b; Abdulla et al., 2010).

The basic characterization of this process is that suffixes must agree in backness with the final front (/æ ø y/; Table 7) or back (/u o ɑ/; Table 8) harmonizing stem vowel.

t**y**r-d**æ**/*-dɑ	'type-LOC'
p**æ**n-l**æ**r/*-lɑr	'science-PL'
munb**æ**r-g**æ**/*-ʁɑ	'podium-DAT'

Table 7: Simple front harmonizing forms

p**u**l-**ʁɑ**/*-gæ	'money-DAT'
t**o**p-**qɑ**/*-kæ	'ball-DAT'
ætr**ɑ**p-t**ɑ**/*-tæ	'surroundings-LOC'

Table 8: Simple back harmonizing forms

The vowels /i e/ are *transparent* to harmony (Tables 9 and 10).

m**æ**stʃit-t**æ**/*-tɑ	'mosque-LOC'
ymid-l**æ**r/*-lɑr	'hope-PL'
m**ø**min-g**æ**/*-ʁɑ	'believer-DAT'

Table 9: Front stems with transparent vowels

st**u**dent-l**ɑ**r/*-lær	'student-PL'
uniwersitet-t**ɑ**/*-tæ	'university-LOC'
ɑmil-**ʁɑ**/*-gæ	'element-DAT'

Table 10: Back stems with transparent vowels

If a stem contains no harmonizing vowels, the front dorsals /k g/ (Table 11) and back dorsals /q ʁ χ/ (Table 12) may serve as harmony triggers.

kishi-l**æ**r/*-lɑr	'person-PL'
ne**g**iz-g**æ**/*-ʁɑ	'basis-DAT'

Table 11: Front dorsal stems that take front suffixes

qiz-l**ɑ**r/*-lær	'girl-PL'
ji**ʁ**in-d**ɑ**/*-dæ	'meeting-LOC'
χiris-**qɑ**/*-kæ	'grimace-DAT'

Table 12: Back dorsal stems that take back suffixes

There are a small number of stems with front dorsals that take back suffixes (Table 13). The opposite case (stems with only back dorsals that take front suffixes) does not appear to occur.

ingliz-l**ɑ**r	'English person-PL'
etnik-l**ɑ**r	'ethnic group-PL'
rentgen-**ʁɑ**	'x.ray-DAT'
gips-**qɑ**	'plaster-DAT'

Table 13: Front dorsal stems that take back suffixes

When a stem contains a harmonizing vowel with a following dorsal that conflicts in backness, the final vowel generally takes precedence (Table 14), although there are a small number of stems containing front vowels with following uvulars that take back suffixes (Table 15).

m**æ**ntiq-q**æ**	'logic-DAT'
æqil-g**æ**	'intelligence-DAT'
r**ɑ**k-l**ɑ**r	'shrimp-PL'
p**ɑ**kit-l**ɑ**r	'fact-PL'

Table 14: Conflicting vowels and dorsals, vowel takes precedence

The uvular consonants may thus be characterized as *gradient blockers*: they generally allow the backness of the preceding vowel to pass through to the suffix, but will occasionally impose their own, blocking harmony with the vowel. See Mayer and Major (2018) for a discussion of the challenges of modeling this pattern using TSL.

5.1 Wug-testing backness harmony

Mayer et al. (2019, 2020) present results from wug tests performed on 23 speakers of Uyghur living in Kazakhstan. These tests included nonce words of the templates shown in Table 16, where C, K, and Q are transparent, velar, and uvular consonants, and N, F, and B are transparent, front, and back vowels respectively.

tæsti**q**-**qa**	'approval-DAT'
tæʃwi**q**-lɑr	'publicity-PL'
tætqi**q**-lɑr	'research-PL'

Table 15: Conflicting vowels and dorsals, dorsal takes precedence

F stems	CFC, CFCNC, CFCNCNC
B stems	CBC, CBCNC, CBCNCNC
FQ stems	CFQ, CFCNQ, CFCNCNQ
BK stems	CBK, CBCNK, CBCNCNK

Table 16: Stem templates used in the wug task.

These templates vary the distance between final harmonizing vowel and suffix, as well as the presence or absence of a conflicting dorsal between the two. Participants produced four words from each template (48 words total) in unsuffixed and suffixed form in paragraphs designed to provide a naturalistic context. Suffixed forms were coded for whether they contained a back or a front suffix. See Mayer et al. (2019, 2020) for more detail.

The proportion of back responses is shown in Fig. 3. These results indicate that (a) disharmonic suffix forms become more likely as the distance between the final harmonizing vowel and the suffix increases, and that this effect is significantly stronger for front stems, and (b) an intervening uvular between a front vowel and suffix skews responses towards back, but an intervening velar between a back vowel and a suffix does not.

5.2 Modeling wug test data using pTSL

I defined a pTSL grammar (π_P, G) over $\Sigma := \{C, K, Q, N, F, B, S_f, S_b\}$, with $G := \{FS_b, BS_f, KS_b, QS_f, NS_f\}$. The first four k-factors prohibit backness clashes
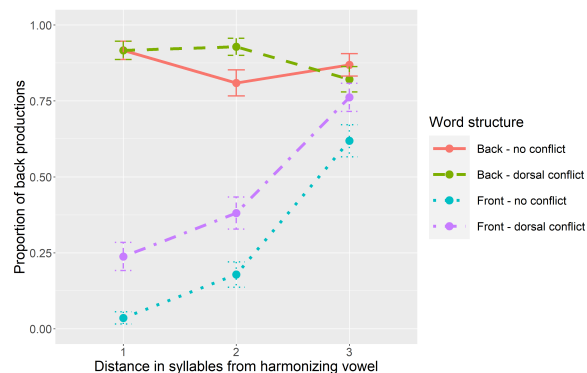

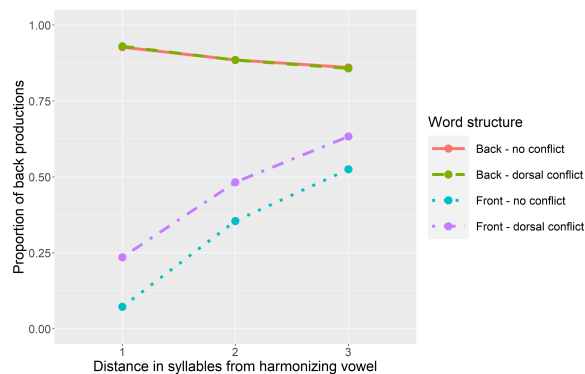Figure 3: Wug word responses from Mayer et al. (2019). Error bars are +/- one standard deviation.

Figure 4: Predicted responses by the optimal pTSL model.

between suffixes and stem vowels and consonants. The final k-factor captures the overall tendency towards back suffixes as the distance between the harmonizing vowel and suffix increases. Note that in Uyghur the neutral vowels behave as gradient triggers for back harmony, while in Hungarian they trigger front harmony.

P_{proj} was fixed to 1 for $\{F, B, S_f, S_b\}$. Maximum likelihood optimization on the remaining parameters was performed in the same manner as for the Hungarian data.

The optimal values were approximately:

$$P_{proj}(C) = 0.08$$
$$P_{proj}(K) = 0.07$$
$$P_{proj}(Q) = 0.24$$
$$P_{proj}(N) = 0.30$$

Fig. 4 shows the proportion of back responses predicted by the model.

The model captures both the gradient blocking displayed by uvulars and the distance-based decay over transparent vowels. Although the model assigns virtually identical frequencies to back stems with and without blockers, this is perhaps not a major shortcoming, since the observed differences in the wug tests are fairly small and difficult to account for in a principled way. More worrying is that the model incorrectly predicts the rate of distance-based decay introduced by transparent vowels. The wug test responses for front stems in Fig. 3 show a smaller decrease in front responses between 1 and 2 syllables, followed by a larger decrease between 2 and 3 syllables. The predictions of the model shown in Fig. 4 display a larger decrease in front responses between 1 and 2 syllables than between 2 and 3. This is an unavoidable consequence of the mathematical properties
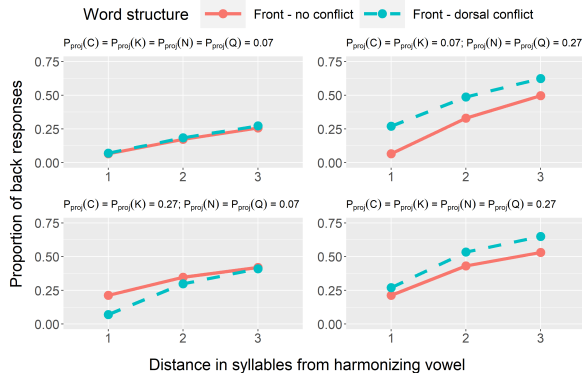
Figure 5: Predicted responses for front stems by models with different projection probabilities.

of pTSL: as independent probabilities are multiplied together to form a joint probability, the rate of change of the joint probability slows with each multiplication (see Fig. 5).

The failure of the model to represent this pattern should not be seen as a significant shortcoming, however. Zymet (2014) identifies decay rates for a range of phonological patterns, and finds that all of them exhibit the kinds of exponential properties that pTSL can represent. Thus the Uyghur wug test results present an interesting exception to general rates of distance-based decay. I discuss this issue in more detail in the next section, and show how pTSL can be extended to cope with it.

6   Conclusion

pTSL grammars allow conditional probabilities to be assigned to stringsets in ways that capture gradient effects in long-distance phonological patterns, and without exceeding the generative capacity of the class of TSL grammars. The parameters of these grammars have a simple and intuitive interpretation from the perspective of autosegmental phonology: the set of prohibited k-factors G corresponds to a set of inviolable local markedness constraints, and the probabilistic projection function π_P defines how likely each segment is to be projected to the tier on which these violations are evaluated. This allows superficially disparate effects such as distance-based decay and gradient blockers to be treated uniformly.

McMullin (n.d.) observed independently that the idea of modeling distance-based decay as a function of probabilities associated with intervening material, as pTSL does, is similar to the decay functions used in Kimper (2011) and Zymet (2014). For example, he shows that the decay

function in Zymet (2014), $\frac{1}{k^x}$, where k is a constant and x is the number of transparent segments between trigger and target, can be implemented by assigning all intervening segments a probability $\frac{1}{k}$ of serving as a blocker. A detailed comparison of these optimality theoretic models with pTSL is beyond the scope of this paper: however, I note that the decay functions proposed in the literature treat all intervening segments alike, while pTSL allows individual projection frequencies for each segment (though this is complicated by the effects of constraint definitions and weights in the OT models). In addition, pTSL lends itself to natural extensions that overcome the limitations of both pTSL and decay functions, while extensions of OT models with decay functions are less straightforward.

6.1   Extending pTSL to handle contextual projection and biases

Generalizations of TSL where the projection function is sensitive to input context (e.g., De Santo and Graf, 2017), output (or tier) context (e.g., Mayer and Major, 2018), or both (e.g., Graf and Mayer, 2018) may be probabilized in a way similar to what has been presented here. This will allow projection probabilities to be conditioned on context. For example, instead of $P_{proj}(x_i)$, we may use $P_{proj}(x_i|x_{i-1})$, $P_{proj}(x_i|y_{j-1})$ (where y_{j-1} is the previously projected symbol), $P_{proj}(x_i|x_{i-1}, y_{j-1})$, etc.

This extension is useful to address two shortcomings of pTSL observed in the Hungarian and Uyghur examples above. In Hungarian, the pTSL model assigns substantially higher ratings to NS_b forms than speakers do. These ratings are simply $1 - P_{proj}(N)$: if the transparent vowel projects, the form will contain an illicit k-factor, while if it does not project, the stem will be licit. A better fit for these forms could be achieved by conditioning on the preceding vowels: that is, transparent vowels are more likely to project when not preceded by a back vowel. pTSL may also be extended to capture lexically-specific phonology (e.g., Pater, 2000) such, as the differences in the effect of uvulars on suffix form between the Uyghur stems in Tables 14 and 15, by conditioning projection probabilities on word identity rather than local context.

The pTSL model of Uyghur wug test data successfully captures distance-based decay and gradient blocking effects of uvular consonants. It fails, however, to predict the correct rate of de-

cay: specifically, the observed productions show a small increase in back responses when a single transparent vowel intervenes between a front vowel and suffix and a larger increase when an additional transparent vowel is added. The model predicts the opposite (see also Mayer et al. (2019), where a similar result emerges from the use of decay functions). The correct rate can be achieved by conditioning the projection probability of neutral vowels on the preceding vowel: neutral vowels after a harmonizing vowel are less likely to project than those after a neutral vowel.

Finally, biases towards particular constraint weights have been employed in previous optimality theoretic models of phonology to explore how biased learning differs from simpler optimization methods like maximum likelihood estimation (e.g., Wilson, 2006). It is straightforward to incorporate similar biases into pTSL models by defining priors over projection probabilities and incorporating them into the learning process. Additionally, it may be interesting to explore pTSL modeling of feature-based representations. I leave these as interesting areas for future research.

A Relating TSL and pTSL grammars

Recall that a stringset $L \subseteq \Sigma^*$ is TSL if there is some TSL-k grammar (T, G) such that $L = \{s \in \Sigma^* | f_k(\pi_T(s)) \cap G = \emptyset\}$ (Section 2.2), and that a stringset $L' \subseteq \Sigma^*$ is pTSL if there is some pTSL-k grammar (π_P, G') such that $L' = \{w \in \Sigma^* | val_{(\pi_P, G')}(w) > 0\}$ (Section 3.3). This appendix demonstrates that the class of TSL stringsets is a proper subclass of the pTSL stringsets. It also describes algorithms that are successful for converting certain subclasses of pTSL grammars to equivalent TSL grammars.

Theorem 1. *TSL $\subsetneq$ pTSL*

To prove this theorem, we must show that TSL $\subseteq$ pTSL and TSL $\neq$ pTSL.

First, we show that TSL $\subseteq$ pTSL. Consider an arbitrary TSL grammar (T, G) over Σ that accepts the stringset L. In order to define a pTSL grammar (π_P, G') that also accepts L, we define the projection probabilities for each $s \in \Sigma$ as follows

$$P_{proj}(s) := \begin{cases} 1 & \text{if } s \in T \\ 0 & \text{otherwise} \end{cases}$$

and set $G' := G$. Under this definition of π_P, each input will have exactly one possible projec-

tion, and this input will receive a non-zero probability only if this projection contains none of the prohibited k-factors in G'. This evaluation procedure is identical to that used by the corresponding TSL grammar.

Example 5. Consider the TSL-2 grammar (T, G) presented in Example 2. The corresponding pTSL-2 grammar (π_P, G') has $G' := G$, and defines the projection probabilities as:

$$P_{proj}(\sigma) = 0$$
$$P_{proj}(\acute{\sigma}) = 1$$

Next, we show that TSL $\neq$ pTSL. This can be demonstrated by counterexample.

Consider a pTSL-2 grammar (π_P, G) over $\Sigma := \{a, b, c\}$. Let π_P be defined such that b and c always project, a sometimes projects, and $G := \{ba, bc, cc\}$. Table 17 shows a number of inputs, their possible projections, and whether they are accepted by the grammar.

Block	Input	Projections	Accepted?
1	b	b	✓
	c	c	✓
	bc	<u>bc</u>	✗
2	cc	<u>cc</u>	✗
	cac	<u>cc</u>, cac	✓
3	ba	b, <u>ba</u>	✓
	ac	c, ac	✓
	bac	<u>bc</u>, <u>ba</u>c	✗

Table 17: Sample inputs to the pTSL grammar. Prohibited 2-factors are underlined.

Now suppose that there is some corresponding TSL-2 grammar (T, G') that accepts the same stringset. At the very least, this grammar must assign the same acceptability judgments to the inputs in Table 17 as those shown in the rightmost column. What remains to be worked out is which symbols such a grammar would need to (deterministically) project and which 2-factors it would need to prohibit in order to do this.

To correctly handle the inputs in Block 1 of Table 17, this grammar must project both b and c: if b were not projected, the projections of c and bc would be identical, meaning either both would be accepted or both would be rejected. Similarly, c must be projected to differentiate between the projections of b and bc. Finally, G' must contain the prohibited 2-factor bc, since this is the only 2-factor of the projection of bc that is not contained in the projections of b or c.

The inputs in Block 2 of Table 17 demonstrate two facts: first, the corresponding TSL-2 grammar must project a: otherwise, the projections of the inputs cc and cac will be identical. Because all symbols are projected, this effectively means the corresponding TSL-2 grammar is an SL-2 grammar. Second, G' must contain cc, since this is the only 2-factor of the projection of cc not also present in the projection of cac.

The inputs in Block 3 of Table 17 dash any hopes for an equivalent TSL-2 grammar. Consider the 2-factors of the projections of each of these inputs:

$$f_2(\pi_T(bac)) = \{ \rtimes b, ba, ac, c \ltimes \}$$
$$f_2(\pi_T(bab)) = \{ \rtimes b, ba, ab, b \ltimes \}$$
$$f_2(\pi_T(ac)) = \{ \rtimes a, ac, c \ltimes \}$$

Note that every 2-factor of the projection of the ungrammatical input bac is present in the 2-factors of the projections of the grammatical inputs ba or ac: there is no 2-factor we can use to reject bac without rejecting either ba or ac, and thus our TSL-2 grammar cannot accept the same stringset as the pTSL-2 grammar defined above (this can also be demonstrated using suffix substitution closure; Rogers and Pullum, 2011).

We can also demonstrate that there is no TSL-k grammar for $k > 2$ that accepts the same stringset as this pTSL-2 grammar (the case of $k = 1$ is trivially true). Note that the considerations of which symbols to project exemplified by Blocks 1 and 2 in Table 17 still apply regardless of our choice of k, so the question becomes whether we can define k-factors for $k > 2$ that produce the correct behavior.

Input	Projections	Accepted?
baa*	b, ba, baa, …	✓
aa*c	c, ac, aac, …	✓
baa*c	bc, bac, baac, …	✗

Table 18: Additional sample inputs to the pTSL grammar. Prohibited 2-factors are underlined.

Consider a generalization of the input forms in Block 3 of Table 17, shown in Table 18. The original pTSL-2 grammar will accept all inputs of the form baa^* and aa^*c, and will reject all inputs of the form baa^*c. For any value of $k > 2$ we might choose for an equivalent TSL-k grammar, every k-factor of the projection of the input $baa^{k-2}c$ will be present in the projections of the inputs baa^{k-2}

or $aa^{k-2}c$, and so the former input cannot be rejected without also rejecting one of the latter two. Thus there is no value of k for which a TSL-k grammar will reject the set baa^*c while accepting the sets baa^* and aa^*c.

Thus no TSL grammar can be defined that accepts the same stringset as the pTSL-2 grammar described above, and so TSL $\neq$ pTSL.

A.1 Algorithms for converting subclasses of pTSL to TSL

There are at least two subclasses of pTSL for which equivalent TSL grammars can always be constructed. The first is the trivial subclass of pTSL where every prohibited k-factor contains only symbols whose projection probabilities are < 1. This means that every input will have at least one possible projection that contains none of the prohibited k-factors, and thus all inputs will be accepted. Accordingly, the equivalent TSL grammar will contain no prohibited k-factors (and so the choice of T is unimportant).

A more interesting subclass of pTSL for which equivalent TSL grammars can always be constructed consists of pTSL grammars where every prohibited k-factor contains only symbols that are always projected. In these cases, I conjecture that the following algorithm will always generate an equivalent TSL grammar, though I do not provide a formal proof here.

Consider an arbitrary pTSL grammar (π_P, G) such that all k-factors in G contain only symbols with a projection probability of 1. This can be converted to an equivalent TSL grammar (T, G') by performing two steps:

1. $G' := G$

2. $T := \{ s \in \Sigma | P_{proj}(s) > 0 \}$

Recall that an input will only receive a probability of 0 if all of its possible projections contain prohibited k-factors. Consider an input that does not contain a prohibited k-factor, but where one may be created if certain symbols are deleted. If any of these symbols have projection probabilities greater than 0, there will be at least one possible projection where they are not deleted, and the prohibited k-factor is not produced. Hence this input will not receive a probability of zero. Defining T to be the set of all symbols with non-zero projection probabilities ensures that such strings will be accepted by the corresponding TSL grammar.

Example 6. Consider the pTSL-2 grammar defined in Example 4, which meets the criterion described above. The corresponding TSL-2 grammar has $G := \{ac\}$ and $T := \{a, b, c\}$. This grammar accepts all input strings over $\{a, b, c\}$ except those that contain the substring ac. This is exactly the set to which the original pTSL-2 grammar will assign non-zero probabilities.

Example 7. The pTSL-2 grammar used as a counterexample in the previous section does not meet the criterion for this algorithm to be applied: the prohibited 2-factor ba contains the symbol a, which does not have a projection probability of 1. The TSL grammar generated by this algorithm will erroneously reject the class baa^*.

The trouble in Example 7 arises because always projecting a can both prevent a prohibited k-factor from being formed (as for the input cac) and retain a prohibited k-factor present in the input (as for the input ba). In the pTSL grammar we consider simultaneously the cases where a does and does not project, while in the corresponding TSL grammar we must choose one or the other. This will produce the incorrect output for one of these forms, and we have seen above that no choice of k-factors can mitigate this conflict without producing incorrect results for other forms.

For pTSL grammars where k-factors contain only symbols with projection probabilities of 1 (as in Example 6), choosing to project, in the corresponding TSL grammar, all symbols that sometimes project in the pTSL grammar can only have the effect of *preventing* the formation of prohibited k-factors. The projection of a given input where all sometimes-projecting symbols are projected will never contain more prohibited k-factors than projections where some or none of these symbols project. Thus choosing to project such symbols in the corresponding TSL grammar deterministically generates the "most grammatical" projection: if this projection contains no prohibited k-factors, the input will be accepted by both the original pTSL grammar and its corresponding TSL grammar, and so the two will behave equivalently.

It may be the case that there are additional subclasses of pTSL for which equivalent TSL grammars may always be constructed. I leave this as an interesting area for future research.

References

Arslan Abdulla, Yari Ebeydulla, and Abduréhim Raxman. 2010. *Hazirqi Zaman Uyghur Tili [Modern Uyghur]*. Xinjiang Xelq Neshriyati [Xinjiang People's Publishing House], Ürümchi.

Adam Albright and Bruce Hayes. 2003. Rules vs. analogy in English past tenses: a computational/experimental study. *Cognition*, 90:119–161.

Jean Berko. 1958. The child's learning of English morphology. *Word*, 14:150–177.

Robert Daland, Bruce Hayes, James White, Marc Garellek, Andreas Davis, and Ingrid Normann. 2011. Explaining sonority projection effects. *Phonology*, 28:197–234.

Aniello De Santo and Thomas Graf. 2017. Structure sensitive tier projection: Applications and formal properties. Ms., Stony Brook University.

John Goldsmith. 1976. *Autosegmental Phonology*. Ph.D. thesis, MIT.

Thomas Graf and Connor Mayer. 2018. Sanskrit n-Retroflexion is Input-Output Tier-Based Strictly Local. In *Proceedings of the 15th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 151–160. The Special Interest Group on Computational Morphology and Phonology, Brussels, Belgium.

R. F. Hahn. 1991a. Diachronic aspects of regular disharmony in modern Uyghur. In W. Boltz and M. Shapiro, editors, *Studies in the Historical Phonology of Asian Languages*. John Benjamins.

R. F. Hahn. 1991b. *Spoken Uyghur*. University of Washington Press, Seattle, WA.

Bruce Hayes and Zsuzsa Londe. 2006. Stochastic phonological knowledge: the case of Hungarian vowel harmony. *Phonology*, 23:59–104.

Bruce Hayes, Kie Zuraw, Peter Siptar, and Zsuzsa Londe. 2009. Natural and unnatural constraints in Hungarian vowel harmony. *Language*, 85:822–863.

Jeffrey Heinz. 2018. The computational nature of phonological generalizations. In Larry Hyman and Frank Plank, editors, *Phonological Typology*, Phonetics and Phonology, chapter 5, pages 126–195. Mouton De Gruyter.

Jeffrey Heinz, Chetan Rawal, and Herbert G. Tanner. 2011. Tier-based strictly local constraints in phonology. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*, pages 58–64.

Wendell A. Kimper. 2011. *Competing triggers: Transparency and opacity in vowel harmony*. Ph.D. thesis, University of Massachusetts, Amherst.

Regine Lai. 2015. Learnable vs. unlearnable harmony patterns. *Linguistic Inquiry*, 46:425–451.

Vern M. Lindblad. 1990. *Neutralization in Uyghur*. University of Washington.

Connor Mayer and Travis Major. 2018. A challenge for tier-based strict locality from Uyghur backness harmony. In *Formal Grammar 2018. Lecture Notes in Computer Science, vol. 10950*, pages 62–83. Springer, Berlin, Heidelberg.

Connor Mayer, Travis Major, and Mahire Yakup. 2019. Wug-testing Uyghur vowel harmony. Talk presented at the 27th Manchester Phonology Meeting. Manchester, England. May.

Connor Mayer, Travis Major, and Mahire Yakup. 2020. Conflicting trigger effects in Uyghur backness harmony. Talk presented at the 5th Workshop on Turkic and languages in contact with Turkic. Newark, Delaware. February.

Kevin McMullin. 2016. *Tier-based locality in long-distance phonotactics: Learnability and typology*. Ph.D. thesis, University of British Columbia.

Kevin McMullin. n.d. A distance-free approach to distance-based phonology. Unpublished manuscript.

Kevin McMullin and Gunnar Ó. Hansson. 2019. Inductive learning of locality relations in segmental phonology. *Laboratory Phonology: Journal of the Association for Laboratory Phonology*, 10:14.

Joe Pater. 2000. Non-uniformity in English secondary stress: the role of ranked and lexically specific cosntraints. *Phonology*, 17:237–274.

Alan Prince and Paul Smolensky. 1993/2004. *Optimality theory: Constraint interaction in generative grammar*. Blackwell, Cambridge, MA. Technical Report CU-CS-696-93, Department of Computer Science, University of Colorado at Boulder, and Technical Report TR-2, Rutgers Center for Cognitive Science, Rutgers University, New Brunswick, NJ, April 1993.

James Rogers and Geoffrey K. Pullum. 2011. Aural pattern recognition experiments and the subregular hierarchy. *Journal of Logic, Language and Information*.

Aniello de Santo and Thomas Graf. 2017. Structure sensitive tier projection: Applications and formal properties. Ms., Stony Brook University.

Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. 2020. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272.

Colin Wilson. 2006. Learning phonology with substantive bias: An experimental and computational study of velar palatalization. *Cognitive Science*, 30:945–982.

Kie Zuraw and Bruce Hayes. 2017. Intersecting constraint families: An argument for harmonic grammar. *Language*, 93:497–548.

Jesse Zymet. 2014. Distance-based decay in long-distance phonological processes. In *Proceedings of the 32nd West Coast Conference on Formal Linguistics*. Cascadilla Press.