In Defense of O/E*

Juliet Stanton
New York University

John F. Stanton
University of Florida

January 2, 2022

Abstract

Wilson & Obdeyn (2009) claim that the use of the Observed/Expected measure (O/E) is mathematically unsound. They argue that O/E fails to reconstruct the underlying structure of a dataset, and that it should be abandoned in favor of the loglinear model. In this paper we defend O/E by arguing that the underlying grammatical structure it diagnoses, as well as the structure diagnosed by a loglinear model, are just two possible solutions to a problem that has infinitely many. Based on purely mathematical criteria, the grammar found by O/E is no more or less plausible than that found by a loglinear model.

1 Background and overview

Aspects of the lexica of natural languages can be characterized by matrices that summarize the co-occurrence of segments or types of segments. The matrix in (1), for example, is derived from Maes's (1959) dictionary of Ngbaka Minagende and summarizes the distribution of coronal, dorsal, and labial nasal and oral stops (Danis 2019:577). Here, C_1 is the first consonant in a transvocalic sequence of two consonants, and C_2 is the second; thus there are 28 coronal-coronal sequences, 43 dorsal-coronal sequences, 33 labial-coronal sequences, and so on.

(1) Consonant co-occurrence matrix for toy language

		C_2		
		coronal	dorsal	labial
C_1	coronal	28	34	17
	dorsal	43	26	27
	labial	33	31	12

A major goal of quantitative and computational phonology is to understand what kind of grammar the learner would induce given data like those summarized in (1). Would the learner posit (gradient)

*We thank Gillian Gallagher for comments.

positional restrictions on individual segments? Would the learner posit (gradient) restrictions on combinations of segments? What would the strength of these restrictions look like?

A number of different methods claim to discover the underlying structure of matrices like (1), to give us a sense of what the learner might expect to see if there were no restrictions on combinations, and how (1) deviates from those expectations. The two methods under discussion in this paper are O/E (first used in phonology by Pierrehumbert 1992, and made explicit by Coetzee & Pater 2008) and the loglinear model (as popularized in phonology by Hayes & Wilson 2008). These measures differ markedly in implementation. The O/E statistic is a simple method that can be done on pen and paper, and its results are easy to interpret. If O/E is over 1, the relevant combination of segments is overattested, relative to expectation; if O/E is under 1, the relevant combination of segments is underattested. The loglinear model, by contrast, is harder to implement and interpret, because doing either of these requires a certain degree of statistical sophistication. Just given these differences, one might prefer O/E: it is easier to implement and accessible to a wider range of linguists.

Wilson & Obdeyn (2009) (henceforth WO), however, claim that O/E is an inferior and mathematically unsound method, as it is confounded by differences in positional frequency. This, in turn, makes it unable to accurately identify restrictions on combinations. Their paper has been influential: various journal articles have cited it as a reason to turn away from the O/E statistic (e.g. Breiss & Hayes 2020, Gouskova & Gallagher 2020, Stanton 2020), and this conclusion has influenced how quantitative phonology is taught to students (see e.g. Hayes 2020). In this paper, we argue against WO's conclusion by showing that their argument is flawed as formulated: both O/E and the loglinear model discover possible solutions, neither of which may be the intended one. Section 2 walks through WO's argument. Section 3 summarizes our response. We make three main points: the solution discovered by O/E is a possible solution, the loglinear model does not always identify the intended solution, and the space of correct solutions is vast. Section 4 discusses and concludes.

It is important to note at the outset that our goal is not to argue for O/E over the loglinear model. The goal of the present paper is more modest: we aim only to show that O/E and the loglinear model are two different ways of solving the same problem, neither of which is necessarily the right way. Which of these methods we prefer, if either, should ultimately depend on which model better-resembles the way humans generalize over data.

2 Wilson & Obdeyn's (2009) argument

In this section, we reproduce WO's argument that the loglinear model should be preferred to O/E. In Section 2.1, we walk through WO's example demonstrating that the O/E method fails to discover an underlying co-occurrence matrix. In Section 2.2, we show that a loglinear model succeeds where O/E fails (see also Hayes 2020 for a similar demonstration, using different tools).

2.1 Apparent failure of O/E

WO's (2009) argument begins with a toy grammar (2). The toy grammar in (2) is one in which there are three consonants, [p], [t], and [k]. In the absence of any constraints on combination, in both first (C_1) and second (C_2) position, [p] would occur $\frac{1}{3}$ of the time, [t] would occur $\frac{1}{2}$ of the time, and [k] would occur $\frac{1}{6}$ of the time. In addition to these asymmetries, there are restrictions on consonant combination: words in which C_1 and C_2 are identical occur at half the frequency they would be expected to occur, given the individual frequencies of the consonants under consideration.

(2) Terms for margins and cells

		C_2		
		[p] $\frac{1}{3}$	[t] $\frac{1}{2}$	[k] $\frac{1}{6}$
C_1	[p] $\frac{1}{3}$	$\frac{1}{2}$	1	1
	[t] $\frac{1}{2}$	1	$\frac{1}{2}$	1
	[k] $\frac{1}{6}$	1	1	$\frac{1}{2}$

To determine how frequent we expect each combination of consonants to be, we need to multiply the terms in the margins by the terms in the cells. Thus words in which C_1 and C_2 are [p] will make up $\frac{1}{2} \times \frac{1}{3} \times \frac{1}{3}$, or $\frac{1}{18}$, of the lexicon. Words in which C_1 is [p] and C_2 is [t] will make up $\frac{1}{3} \times \frac{1}{2}$, or $\frac{1}{6}$, of the lexicon. If we follow this procedure for each of the cells in (2), and then normalize the values so that they sum to one, we get the normalized products in (3).

(3) Normalized products

		C_2		
		[p]	[t]	[k]
C_1	[p]	0.069	0.207	0.069
	[t]	0.207	0.155	0.103
	[k]	0.069	0.103	0.017

We can now use these normalized products to construct a toy lexicon. Let us assume that the lexicon contains 5,000 words. By multiplying 5,000 by each of the normalized products in (3), we reach a lexicon with the word frequencies in (4). This is the only aspect of this procedure that is observable to the linguist; the goal of the linguist is then to discover (2) from (4).

(4) Frequencies in a sample

		C_2		
		[p]	[t]	[k]
C_1	[p]	345	1034	345
	[t]	1034	776	517
	[k]	345	517	86

One well-known method of working from (4) to a matrix like (2) is Observed/Expected. To calculate the Observed/Expected ratio, we first need to calculate what the expected frequencies of each cell should be, assuming no constraints on consonant combination. We illustrate first with a calculation for words in which [p] is C_1 and C_2.

Summing across the top row, we see that there are 1724 words in which [p] is C_1; summing down the first column, we see that there are 1724 words in which [p] is C_2. Words in which C_1 is [p] therefore make up 34.48% of the lexicon, as do words in which C_2 is [p]. Assuming no constraints on consonant combination, we expect words in which [p] is C_1 and C_2 to make up roughly 11.89% of the lexicon, or 594 words. To get the O/E value, we divide the observed number in (4), 345, by the expected number, 594. The result is 0.58, suggesting that words in which C_1 and C_2 are [p] are underattested.

We get a different result by considering words in which C_1 is [p] and C_2 is [t]. The first step is the same: summing across the top row, we see that 1724 words, or 34.48% of the lexicon, have [p] as C_1. Summing across the second column, we see that 2327 words, or 46.54% of the lexicon, have [t] as C_2. Assuming no constraints on consonant combination, we expect words in which [p] is C_1 and [t] is C_2 to make up roughly 16.05% of the lexicon, or 802 words. To get the O/E value, we divide the observed number in (4), 1034, by the expected number, 802. The result here is 1.29, suggesting that words in which C_1 is [p] and C_2 is [t] are overattested.

By calculating the O/E for each cell in (4), we reach the result in (5).

(5) O/E values for the sample

		C_2		
		[p]	[t]	[k]
C_1	[p]	0.58	1.29	1.06
	[t]	1.29	0.72	1.17
	[k]	1.06	1.17	0.48

The problem, as Wilson & Obdeyn (2009) point out, is that the result of the O/E calculation suggests a different grammar than the one that was used to generate the dataset. Recall that in (2), each identical combination occurrs at $\frac{1}{2}$ the rate it was expected to. While the O/E calculation does suggest that combinations of identical consonants are underattested relative to combinations of different consonants, the result of the O/E calculation in (5) suggests that the degree of underattestation is different for words with two [p]s, words with two [t]s, and words with two [k]s. The O/E method has thus failed to discover the intended solution in (2).

2.2 Apparent success of a loglinear model

Wilson & Obdeyn (2009) do not show that a loglinear model discovers the intended solution in (2), so we demonstrate that here (see also Hayes 2020). The structure of the data submitted to the

loglinear model is in Table 1. We included one predictor per consonant per position (so p_1, p_2, t_1, etc.), as well as one predictor for each combination: pp, tt, kk, pt, pk, and so on. The count of each sequence (in bold) is the dependent variable, while the rest of the factors in Table 1 are independent variables. Factors were included in the model in the (left-to-right) order that they are presented here.

Table 1: Input to loglinear model

Count	p_1	p_2	t_1	t_2	k_1	k_2	pp	tt	kk	pt	pk	tp	tk	kp	kt
345	1	1	0	0	0	0	1	0	0	0	0	0	0	0	0
1034	1	0	0	1	0	0	0	0	0	1	0	0	0	0	0
345	1	0	0	0	0	1	0	0	0	0	1	0	0	0	0
1034	0	1	1	0	0	0	0	0	0	0	0	1	0	0	0
776	0	0	1	1	0	0	0	1	0	0	0	0	0	0	0
517	0	0	1	0	0	1	0	0	0	0	0	0	1	0	0
345	0	1	0	0	1	0	0	0	0	0	0	0	0	1	0
517	0	0	0	1	1	0	0	0	0	0	0	0	0	0	1
86	0	0	0	0	1	1	0	0	1	0	0	0	0	0	0

We fit a loglinear model to the data in Table 1 using R's glm function and the poisson link function. The results of the fitted model are in Table 3.

Table 2: Results of loglinear model

Predictor	**Coefficient**	***z* value**	**Significant?**
Intercept	5.15	–	–
p_1	0.69	7.43	Yes ($p < .001$)
p_2	0.69	12.87	Yes ($p < .001$)
t_1	1.10	17.65	Yes ($p < .001$)
t_2	1.10	12.48	Yes ($p < .001$)
k_1	NA	NA	NA
k_2	NA	NA	NA
pp	-0.69	-7.43	Yes ($p < .001$)
tt	-0.69	-8.21	Yes ($p < .001$)
kk	-0.70	-5.27	Yes ($p < .001$)
pt	3.83^{-16}	0.00	No
pk	NA	NA	NA
tp	NA	NA	NA
tk	NA	NA	NA
kp	NA	NA	NA
kt	NA	NA	NA

The intercept of this model is the log predicted frequency of kk assuming no constraints on co-occurrence. This is why k_1 and k_2 are not defined in (3): the model has included their values in

the baseline. To evaluate the fit of the model, we can compare the values it returns with the common logarithm (ln) of the observed frequencies. To derive the log predicted frequency of any consonant pair, all that's necessary is to add or subtract from the intercept according to the consonants in that pair (plus any penalty for co-occurrence). To derive the predicted frequency of kk, for example, we simply subtract the penalty for co-occurrence (-0.70) from the intercept (5.15). This gives us 4.45; the log of the observed frequency (86) is 4.45 as well. To give another example: the log predicted frequency of pp is $5.15+0.69+0.69-0.69 = 5.84$; the log of the observed frequency (345) is 5.84 as well. More generally, the predictions of the loglinear model match exactly the observed frequencies; this was confirmed by using R's fitted.values function.

The important point here is that the loglinear model succeeds in reconstructing the matrix (2) originally used to generate the data. Reconstructing the multipliers for segment co-occurrence is simple: given a coefficient c, the multiplier for a given segment pair is e^c. Thus the multiplier for pp, tt, and kk is 0.5; the multipliers for the rest of the cells are 1. While the model does not give us the terms for the frequencies of individual segments, we can calculate these in a roundabout way by finding the expected frequencies of each combination, were there no restrictions on each combination. For each C_1C_2 pair, where x and y are the coefficients for C_1 and C_2 respectively, the expected frequency is $e^{Intercept+x+y}$. For example, the expected frequency for pp, given no constraints on combination, is $e^{5.15+0.69+0.69} = 685$. Performing this calculation for each cell yields the counts in (6).

(6) Expected counts, given no constraints on co-occurrence

		C_2		
		[p]	[t]	[k]
C_1	[p]	685	1033	344
	[t]	1033	1556	518
	[k]	344	518	172

From here, we can calculate the frequencies of [p], [t], and [k] in each position. For p_1, for example, we divide the number of forms where C_1 is [p] (2062) by the total number of forms (6203), yielding 0.33. If we perform this calculation for each consonant in each position, we can complete the matrix. The result, in (7), is identical to the intended solution (2).

(7) Reconstructed grammar from the loglinear model

		C_2		
		[p] 0.33	[t] 0.50	[k] 0.17
C_1	[p] 0.33	0.50	1	1
	[t] 0.50	1	0.50	1
	[k] 0.17	1	1	0.50

It appears, then, that the loglinear model is capable of doing something that O/E cannot; that is, finding the intended solution for a set of data. If this conclusion holds more broadly, it would provide a reason to favor the loglinear model over alternatives, such as O/E.

3 Our response

Our response to WO's argument has three parts. In Section 3.1, we show that the solution found by O/E is a possible solution: it can be used to regenerate the toy dataset. In Section 3.2, we show that the success of the loglinear model in Section 2.2 has to do with properties of the underlying grammar; a loglinear model does not always discover the correct solution. Finally, in Section 3.3, we show that the space of possible solutions to a co-occurrence matrix is much larger than the two solutions discovered by O/E and the loglinear model; they should be thought of as two possible ways of solving a problem, neither of which is necessarily correct.

3.1 The solution found by O/E is a possible solution

WO conclude that, because the O/E statistic does not reconstruct the underlying grammar, it does not find the correct solution. Here we show, however, that the O/E statistic finds a solution that is equally possible as the one discovered by the loglinear model. We begin by considering again the O/Es that the statistic generates, reproduced in (8).

(8) O/E values for the sample

		C_2		
		[p]	[t]	[k]
C_1	[p]	0.58	1.29	1.06
	[t]	1.29	0.72	1.17
	[k]	1.06	1.17	0.48

To re-generate a toy lexicon, we need to first know what the positional probabilities of the consonants are in each position. To calculate this, we need to know what the expected counts of the lexicon would be, given no constraints on combination. We can find these expected numbers by simply dividing the observed numbers (4) by the O/E. The result of this calculation is in (9).

(9) Expected values for the sample

		C_2		
		[p]	[t]	[k]
C_1	[p]	595	802	325
	[t]	802	1078	442
	[k]	325	442	179

From here, we can calculate the positional probabilities of each segment by following the procedure outlined in Section 2.2. To discover the expected frequency of p_1, for example, we can divide the total number of forms where C_1 is [p] (1722) by the total number of forms (4990), yielding 0.355. If we perform this calculation for each of the segments in (9), we reach the matrix in (10).

(10) Grammar found by the O/E statistic

		C_2		
		[p] 0.345	[t] 0.465	[k] 0.190
C_1	[p] 0.345	0.58	1.29	1.06
	[t] 0.465	1.29	0.72	1.17
	[k] 0.190	1.06	1.17	0.48

We can now follow the procedure outlined in Section 2.1 to generate a toy lexicon given the underlying grammar in (10). First, we calculate the probability of each segment pair by multiplying the terms in the margins by the term in the cell; the probability of [pp], for example, is $0.345 \times 0.345 \times 0.58 = 0.069$. Normalizing these so that the probabilities sum to 1 results in (11).

(11) Normalized products

		C_2		
		[p]	[t]	[k]
C_1	[p]	0.069	0.207	0.069
	[t]	0.207	0.155	0.103
	[k]	0.069	0.103	0.017

If we multiply these by 5,000, we find the counts in (12). Notice that these are identical to the frequencies in (3), which were generated using the grammar in (2).

(12) Frequencies in a sample

		C_2		
		[p]	[t]	[k]
C_1	[p]	345	1034	345
	[t]	1034	776	517
	[k]	345	517	86

The significance of this exercise is that there are at least two possible analyses of the count data in (12): the solution found by the loglinear model (7) and the solution found by O/E (10). Instead of asking why the O/E statistic is incapable of finding (7), we could ask why the loglinear model is incapable of finding (10). Both are possible solutions; there is no reason, internal to this exercise, to prefer one over the other.

3.2 Some matrices are not reconstructed by O/E or a loglinear model

A further piece of evidence that WO's (2009) argument lacks rigor comes from consideration of matrices that do not resemble the matrix in (2). Namely, it is possible to construct a toy lexicon whose underlying grammar neither O/E nor a loglinear model can reproduce. We provide one example here, though it is possible to construct many more.

In this example, we consider a matrix that is minimally different from the matrix in (2). The probabilities of individual segments remain the same, as do the restrictions on the co-occurrence of identical consonants. The difference is that combinations of [p]s and [k]s are overattested; they occur at $\frac{3}{2}$ times the rate that would be expected, given the independent probabilities of [p] and [k].

(13) Terms for margins and cells

		C_2		
		[p] $\frac{1}{3}$	[t] $\frac{1}{2}$	[k] $\frac{1}{6}$
C_1	[p] $\frac{1}{3}$	$\frac{1}{2}$	1	$\frac{3}{2}$
	[t] $\frac{1}{2}$	1	$\frac{1}{2}$	1
	[k] $\frac{1}{6}$	$\frac{3}{2}$	1	$\frac{1}{2}$

By calculating the probability of each combination and normalizing the products so that they sum to 1, and then multiplying those products by 5,000, we generate a lexicon with the counts in (14).

(14) Frequencies in a sample

		C_2		
		[p]	[t]	[k]
C_1	[p]	323	968	484
	[t]	968	726	484
	[k]	484	484	81

3.2.1 Apparent failure of O/E

To find the grammar discovered by O/E, we first need to calculate how frequent each combination of consonants is predicted to be, if there were no constraints on co-occurrence. The result of this calculation, following the description of calculating expected frequencies in Section 2.1, is in (15).

(15) Expected frequencies, given observed frequencies in (14)

		C_2		
		[p]	[t]	[k]
C_1	[p]	630	773	372
	[t]	773	948	457
	[k]	372	457	220

By using these expected frequencies, we can find the underlying grammar, or the solution, that the O/E statistic assumes. To calculate the multipliers, we divide the observed numbers in (14) by the observed numbers in (15). To find the assumed frequencies of each consonant in each position, we follow the procedure outlined in Section 2.2. Doing this results in the grammar summarized in (16).

(16) Grammar discovered by O/E

		C_2		
		[p] 0.354	[t] 0.435	[k] 0.210
C_1	[p] 0.354	0.513	1.252	1.301
	[t] 0.435	1.252	0.766	1.059
	[k] 0.210	1.301	1.059	0.368

The reader can verify that this is a possible solution to the problem by following the procedure outlined in Section 3.1: finding the expected frequency of each cell, normalizing these frequencies so that they sum to one, and then multiplying them by 5,000. Minor differences between the numbers found by following this procedure and those in (14) are due to rounding in (16).

3.2.2 Apparent failure of the loglinear model

The loglinear model, in this case, faces the same problem as the O/E statistic: it finds a possible grammar that could underlie (14), but the grammar that it finds is not the same as the grammar in (13), which was used to generate the data.

We fit a loglinear model to the data in (14) that was identical to the loglinear model in Section 2.2 (see Table 1), save of course for the different count numbers. The loglinear model's fit to the data is summarized in Table 3. R's fitted.values function confirms that the model fits the data.

To reconstruct the grammar that the loglinear model finds, we can first find the multipliers by taking e^c, where c is the coefficient for pp, tt, and kk. Following this, we can find the rest of the grammar by finding numbers that the loglinear model expects to occur, given no constraints on co-occurrence. The expected value for [pp], for example, is $e^{5.49+0.69+0.69}$. Crucially, the penalty for co-occurrence (-1.10) is not added in here. The expected values found using this method are in (17).

(17) Expected numbers, given no constraints on co-occurrence

		C_2		
		[p]	[t]	[k]
C_1	[p]	963	963	483
	[t]	963	963	483
	[k]	483	483	242

To find the assumed frequencies of each consonant in each position, we again followed the proce-

Table 3: Results of loglinear model

Predictor	**Coefficient**	***z* value**	**Significant?**
Intercept	5.49	–	–
p_1	0.69	8.15	Yes ($p < .001$)
p_2	0.69	12.45	Yes ($p < .001$)
t_1	0.69	12.45	Yes ($p < .001$)
t_2	0.69	8.15	Yes ($p < .001$)
k_1	NA	NA	NA
k_2	NA	NA	NA
pp	-1.10	-12.08	Yes ($p < .001$)
tt	-0.29	-3.56	Yes ($p < .001$)
kk	-1.09	-8.27	Yes ($p < .001$)
pt	1.083^{-15}	0.00	No
pk	NA	NA	NA
tp	NA	NA	NA
tk	NA	NA	NA
kp	NA	NA	NA
kt	NA	NA	NA

dure outlined in Section 2.2. Doing this results in the grammar summarized in (18).

(18) The whole grammar discovered by loglinear model

		C_2		
		[p] 0.400	[t] 0.400	[k] 0.200
C_1	[p] 0.400	0.33	1	1
	[t] 0.400	1	0.75	1
	[k] 0.200	1	1	0.33

As with the grammar discovered by O/E, the reader can verify that this is a possible solution to the problem by following the procedure outlined in Section 3.1: finding the expected frequency of each cell, normalizing these frequencies so that they sum to one, and then multiplying them times 5,000.

3.2.3 Local summary

In this subsection, we have shown that there are at least three possible grammars that can underlie the toy lexicon in (14). The first, (19), was originally used to generate the data. While both O/E (20) and the loglinear model (21) discovered possible analyses of the data, and both found that pairs of identical consonants are underattested relative to expectation, neither of them found a matrix identical to the one originally used to generate the data.

(19) Terms for margins and cells

		C_2		
		[p] $\frac{1}{3}$	[t] $\frac{1}{2}$	[k] $\frac{1}{6}$
C_1	[p] $\frac{1}{3}$	$\frac{1}{2}$	1	$\frac{3}{2}$
	[t] $\frac{1}{2}$	1	$\frac{1}{2}$	1
	[k] $\frac{1}{6}$	$\frac{3}{2}$	1	$\frac{1}{2}$

(20) Grammar discovered by O/E

		C_2		
		[p] 0.354	[t] 0.435	[k] 0.210
C_1	[p] 0.354	0.513	1.252	1.301
	[t] 0.435	1.252	0.766	1.059
	[k] 0.210	1.301	1.059	0.368

(21) Grammar discovered by the loglinear model

		C_2		
		[p] 0.400	[t] 0.400	[k] 0.200
C_1	[p] 0.400	0.33	1	1
	[t] 0.400	1	0.75	1
	[k] 0.200	1	1	0.33

In particular, note that while the co-occurrence restriction used in the matrix used to generate the data was consistent across [pp], [tt], and [kk]: the identical combinations occur at $\frac{1}{2}$ they would be expected to, assuming no constraints on co-occurence. Neither O/E nor the loglinear model is able to recover this consistency, instead finding (in both cases) that the co-occurrence restriction targeting [tt] is stronger than the restriction targeting [pp] and [kk].

This finding reveals a core weakness in WO's argument: it does not generalize. A loglinear model, just like O/E, cannot necessarily discover the intended underlying structure of a toy lexicon.

3.3 There are many more solutions

In the above section we have demonstrated that there are at least two possible grammars that can underlie the toy lexicon in (4). In this section we demonstrate that these two possible grammars form part of a vastly larger set. In brief, for any set of positional frequencies (i.e. [p] occurs at $\frac{1}{3}$, [t] occurs at $\frac{1}{3}$, and [k] occurs at $\frac{1}{3}$), it is possible to solve for a set of co-occurrence multipliers. To verify this, we wrote a Fortran program that first calculates the values in (4), given the underlying grammar in (2). It then asks for possible positional frequencies of the consonants, and from this, calculates the co-occurrence matrix. Valid inputs include those in which the individual frequencies of the consonants are between 0 and 1, and in which the frequencies, together, sum to 1.

We include three possible solutions returned by this program. In the first, [p t k] occur at equal

frequency. The co-occurrence matrix discovered by the program suggests that combinations of [p]s and [k]s are underattested, while combinations of [t]s are overattested.

(22) Example 1

		C_2		
		[p] $\frac{1}{3}$	[t] $\frac{1}{3}$	[k] $\frac{1}{3}$
C_1	[p] $\frac{1}{3}$	0.621	1.862	0.621
	[t] $\frac{1}{3}$	1.862	1.397	0.931
	[k] $\frac{1}{3}$	0.621	0.931	0.155

In the second, the rate of occurrence of [p] and [t] are reversed from the original grammar: [p] occurs at a rate of $\frac{1}{2}$, while [t] occurs at a rate of $\frac{1}{3}$. The co-occurrence matrix discovered by the the program is qualitatively similar to that in (22): pairs of [p]s and [k]s are overattested, while pairs of [t]s are overattested.

(23) Example 2

		C_2		
		[p] $\frac{1}{2}$	[t] $\frac{1}{3}$	[k] $\frac{1}{6}$
C_1	[p] $\frac{1}{2}$	0.276	1.241	0.828
	[t] $\frac{1}{3}$	1.241	1.397	1.862
	[k] $\frac{1}{6}$	0.828	1.862	0.621

Finally, we considered a situation in which [k] is twice as frequent as [p] and [t] (in contrast to the intended solution, where it is far less frequent). In this situation, the co-occurrence matrix discovered by the program suggests that combinations of [k]s are underattested, combinations of [t]s are overattested, and combinations of [p] are about as frequent as would be expected.

(24) Example 3

		C_2		
		[p] $\frac{1}{4}$	[t] $\frac{1}{4}$	[k] $\frac{1}{2}$
C_1	[p] $\frac{1}{4}$	1.103	3.310	0.552
	[t] $\frac{1}{4}$	3.310	2.483	0.828
	[k] $\frac{1}{2}$	0.552	0.828	0.069

The reader can verify that each of these grammars generates the toy lexicon in (4), using the procedure outlined in Section 2.1. More generally, it is possible to input any set of frequencies and solve for the co-occurrence matrix in the center. The space of possible grammars that underlie the lexicon in (4) is thus much larger than the two solutions found by O/E and the loglinear model. Again, there is no reason internal to this exercise to prefer any of these solutions to the others.

4 Discussion and conclusion

This paper makes two main points.

The first is that WO's argument for the superiority of loglinear models over O/E is flawed, for two reasons. First, the solution that O/E discovers is a possible solution: it can be used to re-generate the lexicon that it analyzes. Second, the loglinear model, just like O/E, is not always successful in discovering the analyst's intended solution. As shown in Section 3.2, it is possible to construct toy grammars that neither O/E nor the loglinear model can reconstruct. Arguments for loglinear models over O/E must rest on claims other than the mathematical insufficiency of O/E, because in terms of their ability to discover possible solutions underlying a dataset, the two are indistinguishable.

The second point is more nuanced. The primary conclusion of Section 3.3 is that the space of possible solutions is vast, and likely infinite: given any set of valid positional frequencies, it is possible to solve for a co-occurrence matrix. And given that the solutions discovered by O/E and the loglinear model are just two of these possible solutions, there is no reason to prefer one over the other, or to indeed prefer either of them at all. Put more bluntly, it is entirely possible that neither O/E nor the loglinear model is the correct way of analyzing count data. Arguments for one way of analyzing count data over the other need to be based not on mathematical sufficiency and elegance but rather on how closely they resemble the generalizations that humans draw when they are exposed to this data. (See Gallagher et al. 2019 for an example of this kind of argumentation.) A major goal of theoretical linguistics, after all, is to build models that reflect speakers' knowledge.

Returning to the main thrust of this paper: if we are to prefer the loglinear model over O/E, it needs to be shown that the way humans generalize better-resembles a grammar discovered by the loglinear model than it does a grammar discovered by O/E. One way of achieving this goal would be to compare the grammars discovered by these two methods with aspects of speakers' grammars discovered through experimental tasks. To the best of our knowledge, this has not been done. Thus there is no reason that we know of to disfavor O/E.

References

Breiss, Canaan & Bruce Hayes. 2020. Phonological markedness effects in sentence formation. *Language* 96. 338–370.

Coetzee, Andries W. & Joe Pater. 2008. Weighted constraints and gradient restrictions on place co-occurrence in Muna and Arabic. *Natural Language and Linguistic Theory* 26. 289–337.

Danis, Nick. 2019. Long-distance major place harmony. *Phonology* 36. 573–604.

Gallagher, Gillian, Maria Gouskova & Gladys Camacho Rios. 2019. Phonotactic restrictions and morphology in Aymara. *Glossa* 4(1). 29.

Gouskova, Maria & Gillian Gallagher. 2020. Inducing Nonlocal Constraints from Baseline Phonotactics. *Natural Language and Linguistic Theory* 38. 77–116.

Hayes, Bruce. 2020. Class notes from Linguistics 219: Phonological Theory III. Online at https://linguistics.ucla.edu/people/hayes/219/index.html.

Hayes, Bruce & Colin Wilson. 2008. A maximum entropy model of phonotactics and phonotactic learning. *Linguistic Inquiry* 39. 379–440.

Maes, Védaste. 1959. *Dictionnaire Ngbaka-Francais-Neerlandais*. Tervuren: Commissie voor Afrikaanse Taalkunde / la Commission de Linguistique Africaine.

Pierrehumbert, Janet. 1992. Dissimilarity in the Arabic Verbal Roots. In *Proceedings of the 23rd Meeting of the Northeastern Linguistic Society*, Amherst, MA: GLSA.

Stanton, Juliet. 2020. Aggressive reduplication and dissimilation in Sundanese. *Phonological Data and Analysis* 2. 1–35.

Wilson, Colin & Marieke Obdeyn. 2009. Simplifying subsidiary theory: statistical evidence from Arabic, Muna, Shona, and Wargamay. Ms. Johns Hopkins University.