Data-driven learning of symbolic constraints for a log-linear model in a phonological setting

Gabriel Doyle
Department of Psychology
Stanford University
Stanford, CA 94305
gdoyle@stanford.edu

Roger Levy
Department of Brain and Cognitive Sciences
Massachusetts Institute of Techonology
Cambridge, MA 02139
rplevy@mit.edu

Abstract

We propose a non-parametric Bayesian model for learning and weighting symbolically-defined constraints to populate a log-linear model. The model jointly infers a vector of binary constraint values for each candidate output and likely definitions for these constraints, combining observations of the output classes with a (potentially infinite) grammar over potential constraint definitions. We present results on a small morphophonological system, English regular plurals, as a test case. The inferred constraints, based on a grammar of articulatory features, perform as well as theoretically-defined constraints on both observed and novel forms of English regular plurals. The learned constraint values and definitions also closely resemble standard constraints defined within phonological theory.

1 Introduction

Constraint-based models of language, often in the form of "maximum entropy" or "log-linear" models, are prominent in many applications and theoretical analyses in computational linguistics and psycholinguistics, including in text segmentation (Beeferman et al., 1999; Poon et al., 2009), machine translation (Och and Ney, 2002), syntactic alternation choice (Bresnan et al., 2007), and phonology (Goldwater and Johnson, 2003). Building successful models – and learning about human behavior from them – relies on the ability to identify relevant constraints, and this can be a difficult problem.

In this paper, we propose a system for learning both the values of and symbolic definitions for such constraints. We present a framework that combines observed data about linguistic outcomes with a flexible probabilistic context-free grammar of constraint structure to jointly infer (binary) feature values for multiple constraints and likely symbolic definitions for those constraints. We ground the model in a morphophonological setting, using the model to infer what phonological constraints affect the output form of regular English plurals, although it can be applied to other problems for which a constraint grammar can be defined.

The inference procedure moves beyond existing methods for learning *extensional* definitions of constraint values (Griffiths and Ghahramani, 2005; Görür et al., 2006; Doyle et al., 2014) from observational data to incorporate top-down information about likely *intensional* constraint definitions, improving both the applicability of the constraints and the theoretical basis for their values. We show that learning the constraints through this model performs as well as using pre-specified phonologically-standard constraints in explaining both observed and novel regular plural morphophonology. In addition, the structure of the learned constraints is similar to standard phonological constraints, showing that the model can be useful in both applications and theory-building.

2 Constraint-based models and the phonological test case

Our core problem is how to learn appropriate identities and weights for log-linear features in linguistic applications. In general, we assume some set of input types $\{x_i\}$, with $n_{i\cdot}$ instances of each type observed. The input type x_i is observed to produce n_{ij} instances of each outcome type y_j, and, as we are using a log-linear model, we assume that the number of observed input-output pairs (x_i, y_j) is proportional to the exponential of the weighted sum of the constraint values v_{ijk} over all constraints k. At least

Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, pages 2217–2226, Osaka, Japan, December 11-17 2016.

some subset of these constraints k are unknown, and our goal is to learn the number of and values for these unknown constraints, as well as weights for both the known and unknown features.

Furthermore, we assume that the values of the unknown constraints are based on definitions that are generated from a symbolic grammar. This allows the model to inject theory- or observation-based structure into the learning process, improving the plausibility of the constraint values and allowing the researcher to identify likely definitions for the constraints to apply to unobserved inputs. At present, we limit ourselves to the case where the unknown constraints are binary and depend only on the outcome type y_j, a simplified case that is particularly relevant to phonological constraint acquisition. We discuss avenues for relaxing the binarity limitation in Section 7.

Gaps in log-linear phonological modeling We consider phonological theory as a test case because it has a well-established constraint-based framework, Optimality Theory (OT; Prince and Smolensky (1993)). But there is a gap in learning methods for OT-style phonology. Multiple methods have been proposed within OT for learning constraint weights or orderings (Tesar and Smolensky, 2000; Boersma and Hayes, 2001; Goldwater and Johnson, 2003) when the constraint definitions are known. None of these can learn constraint definitions, though three general tracks of research have pushed toward this goal. One track builds phonetically-grounded constraints based on the difficulty of producing or understanding the sound sequences (Hayes, 1999), but cannot produce constraints that lack such grounded motivations (Hayes, 1995). A second track learns constraints within a phonotactic problem, looking solely at attested output forms (Hayes and Wilson, 2008; Berent et al., 2012), but the phonotactic learning problem does not take input forms into account, and searches over a finite constraint set (instead of an infinite grammar). A third track uses data-driven learning to infer constraints (Doyle et al., 2014), but this method only learns which words violate a given constraint, and not a symbolic or intensional definition to apply it to novel words.

We propose a model to fill the gaps between these research tracks, by inferring constraints: 1) in the absence of articulatory motivation, 2) in the presence of input forms, and 3) with explicit, symbolic constraint definitions. The model uses a simple (but infinite) grammar of constraints to jointly learn a matrix of constraint violations, likely definitions for the constraints, and relative weights on the constraints that adequately explain the observed phonological forms.

Phonological constraints and log-linearity Traditional versions of OT do not employ log-linearity, so we work with the MaxEnt OT (MEOT; Goldwater and Johnson (2003)) framework, an extension that connects constraint-based phonology to the general class of log-linear models. (Traditional, non-log-linear, OT is approximated as the difference between weights on the MEOT constraints grow.) Some existing work on phonotactic and phonological constraint learning (Hayes and Wilson, 2008; Doyle et al., 2014) has been based in such a log-linear framework.

As with all OT frameworks, the core structure supposes that phonological forms are produced by starting with an input form, generating a set of output candidates, determining what constraints each candidate input-output pair violates, and selecting an output form based on the number and strength of the candidates' constraint violations. There are two types of constraints: those that depend on both the input and output ("faithfulness"), and those that depend only on the output ("markedness"). Each constraint has an associated weight, which is always non-positive; no constraint violations can make an output form more likely to be chosen. MEOT is a log-linear model, so summing the weights of all violated constraints provides each candidate's linear predictor, which is logit-transformed to a probability.

In terms of the general framework from the start of this section, faithfulness constraints are known, while the markedness constraints and weights for both constraint types are unknown.[1] In addition, we assume that the definitions for the markedness constraints are generated by a PCFG over phonological features of the sounds of the output candidates. Our specific grammar is discussed in Sect. 5.2.

[1]We limit ourselves to the learning of markedness constraints in this paper, as faithfulness constraints appear to be less arbitrary than markedness constraints (McCarthy, 2008), and may be representable as part of the output candidate generation process (Riggle, 2009).

3 Model structure

We represent the constraints as two matrices: F, the observed faithfulness constraints, which depend on both input and output forms; and M, the unobserved markedness constraints, which depend only on the output form. Each cell of F and M tells the number of violations of a constraint by a given input-output mapping. F_{ijk} is the number of violations of faithfulness constraint k by input-output pair type (x_i, y_j); M_{jl} is the number of violations of markedness constraint l by output candidate y_j. For each input x_i, some subset of the output forms $\{y_j\}$ are possible; this subset will be denoted $\mathcal{Y}(x_i)$. The weight vector w provides weights for both F and M, and is unobserved.

M is a non-parametric binary matrix with a known number of rows (candidates) but an unknown number of columns (constraints). Each column $M_{\cdot l}$ of the matrix M, which we will refer to as a "violation profile", is a binary vector of length J, the number of output candidates, specifying whether the candidate y_j violates this constraint. w is a vector of real numbers; within OT, weights are strictly negative, so we draw from $-\exp(\eta_w)$.

Previous work on constraint learning (Doyle et al., 2014) generated M through an Indian Buffet Process (Griffiths and Ghahramani, 2005), with the number of constraints L generated by a Poisson prior (with parameter α) and the violation profiles generated by a rich-get-richer scheme. In the present work, we retain the Poisson prior over L, but we want the violation profiles to be derived from symbolic constraint definitions d instead. The definitions are built from the underlying grammar G and specify whether each candidate y_j violates d. Within our model, we assume that a candidate y_j can be an exception to the definition d (switching a one to zero or vice versa in M_{jl}), and the number of exceptions is drawn from a exponential prior (Rational Rules framework; Goodman et al. (2008)). Thus, given a constraint definition d_l, the probability of it producing a violation profile $M_{\cdot l}$ is given by

$$p(M_{\cdot l}|d) \propto \exp(-bQ(M_{\cdot l}; y_\cdot, d_l)) \tag{1}$$

where $Q(M_{\cdot l}; y_\cdot, d_l)$ is the number of exceptions in $M_{\cdot l}$ given candidates $\{y_j\}$ and definition d_l, and b is the exception parameter, with larger b penalizing exceptions more strongly. As neither the true violation matrix M nor the true constraint definitions d are observed, we estimate the probability of a violation profile $M_{\cdot l}$ by marginalizing over possible constraint definitions (see Sect. 4.1).

The probability of whole observed corpus Y is the product of the probabilities across all observed input-output pairs:

$$p(Y|M, F, W) \propto \prod_i \frac{\left(\exp\left(\sum_{jk} w_{Fk} F_{ijk} + \sum_{jl} w_{Ml} M_{jl}\right)\right)^{n_{ij}}}{\left(\sum_{y_z \in \mathcal{Y}(x_i)} \exp\left(\sum_k w_{Fk} F_{izk} + \sum_l w_{Ml} M_{zl}\right)\right)^{n_{i\cdot}}} \tag{2}$$

In summary: F is observed, $w \sim -\exp(\eta_w)$, $L \sim \text{Poiss}(\alpha)$, $M_{\cdot l} \sim \exp(-bQ(M_{\cdot l}; y_\cdot, d_l))$, and $d_l \sim \text{PCFG}(G)$. We infer likely constraint matrices and weights M and w from their joint posterior distribution, which is proportional to the product of the probabilities of the data (Eqn. 2), constraints M, and weights w:

$$p(M, w|Y, F, \alpha, b, \eta_w, G) \propto p(Y|M, F, w)p(M|b, G)p(w|\eta_w) \tag{3}$$

4 Model Inference

For the model to find appropriate constraint structures, we use Markov Chain Monte Carlo (MCMC) inference over the space of constraint definitions d, markedness matrices M, and weight vectors w.

4.1 Inference over d

Inference on M requires knowledge of the prior over violation profiles (i.e., columns of M), but this is a sum over the infinite set of constraint definitions. To estimate this, we use importance sampling over constraint definitions d. For a given profile m, we start by drawing a constraint definition d from the PCFG, then Metropolis-Hastings sample through the space of constraint definitions, with three possible

move types of equal probability: subtree replacement, incision, and excision. In terms of the constraint definitions in Sect. 5.2, replacement changes a feature value, a phoneme, or a phoneme sequence; excision removes a feature, phoneme or phoneme sequence; and insertion adds a feature, phoneme, or phoneme sequence.

Subtree replacement The first move, subtree replacement, comes from Goodman et al. (2008). Subtree replacement chooses a non-terminal node uniformly randomly in the tree, and re-draws all of its children according to the PCFG probabilities. If a subtree replacement is to be made, the jump probability of moving from tree T to T' by redrawing the subtree S_X at node X is:

$$J_R(T';T) = \frac{1}{N_R} \cdot \prod_{r \in S'_X} p(r), \tag{4}$$

where N_R is the number of non-terminal nodes in T, S'_X is the new subtree with root X, and r ranges over the rules triggered by S'_X.

Node excision The second move is node excision, which promotes a subtree one level up in the tree, eliminating its parent node and sibling subtree. It selects a node X uniformly randomly from the set of nodes that can be excised (nodes with at least one grandchild Z that is also a valid child of X under the CFG). If no excisable nodes exist in the tree, the model attempts a different move type (replacement or insertion) instead. Excision removes a node Y – the child of X and parent of Z – from the tree, as well as the current sibling of Z (with its subtree). If an excision is to be made, the jump probability of choosing to excise between X and Z in tree T to yield tree T' is:

$$J_E(T';T) = \frac{1}{N_E} \cdot \frac{1}{N_{E;X}}, \tag{5}$$

where N_E is the number of nodes in T that have at least one excisable grandchild, and $N_{E;X}$ is the number of excisable grandchildren of X in T.

Node insertion The third move, node insertion, reverses node excision. A new node is inserted between a parent and child node, and the child node gets a new sibling subtree. Node selection for insertion works similarly to excision; a node is drawn uniformly randomly from the set of insertable nodes, those that have at least one child that could also be its grandchild. As with excision, if no insertable nodes exist, a different move type is attempted. Once an insertable node X is chosen, the model chooses a child node Z uniformly among its children that could be a grandchild of X. That node becomes a grandchild of X, and the model draws a new node Y from the PCFG, such that Y is a valid child of X, parent of Z, and sibling of the remaining child node of X (call this A). Finally, Z draws a new sibling B in its new lower position, according to the PCFG. Given that an insertion is to be made to the tree T, the probability of that insertion being node Y between X and Z is:[2]

$$J_I(T';T) = \frac{1}{N_I} \cdot \frac{1}{N_{I;X}} \cdot \frac{p(X \to AY)}{p(X \to A*)} \cdot \frac{p(Y \to ZB)}{p(Y \to (Z * \mid * Z))} \cdot \prod_{r \in S_B} p(r), \tag{6}$$

where N_I is the number of nodes in T that have at least one insertable child, and $N_{I;X}$ is the number of insertable children of X in T. The third fraction is the probability of choosing Y as the new child in T', and the fourth fraction is the probability of choosing B as the new sibling of Z, as well as whether Z is the left- or right-hand child of Y. The final term is the probability of the subtree S_B.

Acceptance probability Using the jump probabilities between trees given by the above equations, we can calculate the acceptance probability of a possible Metropolis move from T to T'. This is the product of the ratio of the forward and backward jump probabilities and the ratio of the trees given the current violation profile m:

[2]This equation assumes that Z is the right-hand child of X and the left-hand child of Y. If Z is the left-hand child of X or the right-hand child of Y or both, the probability is calculated similarly, but the third or fourth fraction changes to reflect the actual structure.

$$\frac{p(m|T')p(T')J(T;T')}{p(m|T)p(T)J(T';T)}. \tag{7}$$

The Metropolis method samples constraint definitions $\{d^{(1)}, \cdots, d^{(n)}\}$ from the posterior distribution $p(d|M_{\cdot l})$. These samples are used to estimate the probability of the violation profile m given the constraint grammar G by taking the harmonic mean of $p(M_{\cdot l}|d^{(t)})$ over all samples (Newton and Raftery, 1994).[3] This provides a prior for the columns of the matrix; coupled with the Poisson prior on the number of columns, we have a phonologically-motivated prior over matrices with an indefinite number of columns.

4.2 Inference over M

Inference on M uses five possible sampling moves, all of which rely on the estimates of $p(M_{\cdot l}|G)$ obtained above. Three of the sampling moves are equivalent to previous work with non-parametric binary constraint matrices (Görür et al., 2006; Doyle et al., 2014): columns may be removed or added, and each cell M_{jl} is Gibbs sampled, potentially changing whether candidate y_j violates constraint l.

We introduce two new moves – splitting or combining columns – to more efficiently move between constraint definitions. These can shift violations that explain the data well but are exceptions within their current column into a column where they fit better. Without them, moving violations between columns requires first removing them via Gibbs sampling, which may be very unlikely due to the loss in data likelihood from the loss of critical constraint violations.

A proposed split and its acceptance probability are drawn as follows. The likelihood of a violation M_{jl} being an exception within its profile $M_{\cdot l}$ is estimated from the proportion of samples from $p(d|M_{\cdot l})$ that mark the violation as exceptional. The set of violations V to be moved is drawn as a sequence of independent Bernoulli draws based on each violation's likelihood of being an exception. The exception likelihood is smoothed using a Beta-binomial distribution with parameter β, by taking the maximum a posteriori estimate of the likelihood:

$$p(m_{jl} \in V) = \frac{N_E + \beta - 1}{N_E + N_N + 2\beta - 2} \tag{8}$$

The number of Metropolis samples in which M_{jl} was an exceptional violation is N_E and a non-exceptional violation is N_N. Higher β increases the overall smoothing, and the effect of the smoothing decreases as more Metropolis samples are drawn. We set $\beta = 100$, as we expect substantial noise due to the size of the sample space.

4.3 Inference over w

After each matrix sample, we apply Metropolis-Hastings sampling on w. Our proposal distribution is $-\Gamma({w_k}^2/\eta_M, \eta_M/-w_k)$, which the current weight w_k as its mean. We set $\eta_M = 1$ as a default.

5 Experiment

5.1 English regular plural morphophonology

We test this model on the English regular plural system, which has one underlying form (/z/) with three attested output realizations: [z], [s], or [əz] (as in *hugs, huts*, and *hushes*, respectively). Two markedness constraints drive this alternation in the standard phonological analysis, which can be written in terms of the phonetic feature sequences they penalize: [-VOI][+VOI] and [+STR][+STR]. The former penalizes outputs where consecutive consonants do not agree in voicing, and the latter penalizes outputs where consecutive consonants are both strident (s,z,sh,ch). These are coupled with three faithfulness constraints, which penalize removing, adding, or changing a phoneme (MAX, DEP, and IDENT in OT terminology).

For this experiment, we consider four candidate outputs for each input: the bare singular form, plus forms with each of the three attested allomorphs of the regular plural suffix. The candidates for plural

[3] Harmonic mean estimation can be noisy and take a large number of iterations to converge (Neal, 1994), so we tested a range of violation profiles and found consistent convergence to the expected constraint definitions and profile probabilities within a few thousand samples.

hug (underlying /hʌgz/), for instance, are [hʌg], [hʌgz], [hʌgs], or [hʌgəz]. In general, the [əz] candidate wins only when the singular ends in a strident, the [s] candidate wins only when the singular ends in a voiceless non-strident, and the [z] candidate wins the rest of the time. The training set consists of the plural forms of 26 nouns, each understood by at least 89% of 18-month-old English learners (Dale and Fenson, 1996).[4] The model is given 100 examples of each plural, always using the standard pluralization.

5.2 The constraint grammar

Potential constraint definitions are sequences of phonological feature bundles. There are 23 phonological features, each capturing different characteristics of a sound; for instance, the phoneme [s] has phonological features including [+consonantal, +strident, –voiced], while the similar phoneme [z] has features including [+consonantal, +strident, +voiced]. A feature bundle within a constraint definition matches all phonemes with all of the bundle's features. Thus a definition [+consonantal, +strident][+consonantal, +voiced] matches the strings sz and zz, but not zs. Phoneme-to-feature mappings are based on Riggle (2012).[5] To make sure that model's success is not based on the grammar generating only definitions relevant to the English plural problem, we include Kleene stars, matching zero or more consecutive occurrences of a feature bundle. While some other phonological constraint definitions, such as vowel harmony, require Kleene stars, the English plural does not.

Note that the constraints are not necessarily binary when defined as sequences of feature bundles; a candidate can contain multiple sequences that violate a constraint. But because we are considering the effect of adding a suffix to a stem word, we can subtract the stem's violations from each candidate. Since all the candidates from a given input share the singular form as a stem, the same number of violations are subtracted from all of them, and the candidate probabilities within the log-linear model are unchanged.

5.3 Model parameters and implementation

We ran the model for 200 iterations in three trial runs, with deterministic annealing on the first 100. For estimating $p(M_{.l})$, 1000 burn-in samples were taken and discarded, and 250 additional samples (every second sample out of 500 to reduce autocorrelation) of d are averaged. For violation profiles $M_{.l}$ that reoccured, each time $p(M_{.l})$ was re-calculated, half as many additional samples were drawn (125, 62, ..., to a minimum of 25) and incorporated into the average. The parameters α and η_w are set to 1 and b is set to 10, to encourage fewer constraints and parity between violations and definitions. These fit the standard phonological assumption of phonologically-motivated and parsimonious constraints.

6 Results

We tested the model's performance in four ways: how well the learned structures explain observed plurals, how well they predict novel plurals, how accurately they reflect the standard violation profiles, and how interpretable the constraint definitions are. In all cases, we compared against a baseline of the standard phonological constraints that phonological theory suggests. This baseline M was derived from the two standard English markedness constraints, [+STR][+STR] and [-VOI][+VOI], with no exceptions. Baseline weights were sampled as in Sect. 4.3, with M held constant.

Explaining observed plurals The first test is to show that the model can learn a phonological system for the observed plurals. The model satisfies this goal if it predicts the observed forms at least as well as the baseline model. We calculate both the mean and MAP values of the data likelihood (Eq. 2) over the final 100 iterations of each of the three model runs, and report the across-run means in Table 1. t-tests found no significant differences between the learned and baseline performance on the training data.

Predicting novel plurals The second test of the model is whether its learned constraints extend to newly-encountered words. This is a crucial feature for human acquisition; children quickly learn to generalize morphophonological systems. It also represents an important model improvement, as Doyle

[4]Training words: *baby, ball, balloon, banana, bath, bird, blanket, book, car, chair, daddy, diaper, door, drink, eye, hug, key, kiss, kitty, mommy, nap, nose, phone, shoe, spoon, toothbrush*

[5]There is one deviation from Riggle's system: we do not specify voicing on sonorants, because sonorants do not have voiceless versions and do not trigger [-VOI][+VOI] violations.

	MAP LL	Mean LL	Min. Cand. Prob.	Mean Cand. Prob.
Model	-2.94	-7.91	.986	.995
Baseline	-5.16	-10.6	.974	.991

Table 1: Comparing the performance of the learned constraints to the baseline of the standard phonological constraint definitions. On the left, training data log-likelihoods on the left, based on values from the final 100 iterations for the three model runs. On the right, test set probability masses for the correct plural forms. The learned constraints perform as well as the phonologically standard constraints.

et al. (2014)'s constraint learning model was incapable of making such predictions due to its strictly-extensional constraints. For the test set, we used the 25 most frequent countable nouns in the Corpus of Contemporary American English (COCA; Davies (2008)) that take regular plurals, none of which were in the training set.[6] Five of these nouns end with phonemes that did not occur word-finally in the observed data, requiring the model to have made phonological generalizations from the training data.

To assess the predictive power of the learned constraints, we obtained constraint definitions by using the $p(d|m)$ Metropolis sampler to generate a distribution over definitions for each violation profile. Violation profiles $M_{\cdot l}$ are taken from the final iteration of each model run. For a new candidate y, the probability $m_{y;l}$ that y violates constraint l was estimated using constraints d drawn from the $p(d|M_{\cdot l})$ Metropolis sampler. $m_{y;l}$ was then used as the constraint value for the log-linear predictor. Both the model and baseline constraints correctly put the highest probabilities on the correct plural forms, as shown in Table 1. All correct plural forms received at least 98% of the probability mass under the model constraints, and there was no significant difference between the model and baseline predictions.

Violation profile accuracy The previous test showed that the constraint definitions effectively extend to unobserved forms. Now we want to examine their correspondence with phonological theory. First, we want to see if the right number of constraints was learned. Two of the model runs had two markedness constraints throughout the final 100 iterations, like the baseline. The third model run used four markedness constraints over its final 100 iterations, but the extra markedness constraints supplied violations that matched two of the faithfulness constraints (DEP and IDENT). Those faithfulness constraints' weights dropped to near zero in this run, though, meaning that all learning and baseline runs had five active constraints. In the runs with two markedness constraints, we tested how their violation profiles and definitions mapped to the baseline constraints.[7] Over the final 100 iterations, the learned violation profiles agree with their corresponding baseline violation profiles on an average of 98.9% of all candidates, showing that both constraint sets have similar phonological meanings.

Similarities in constraint definitions We compared the likely constraint definitions, as estimated by the Metropolis sampler for $p(d|m)$, for the learned and baseline violation profiles to their phonologically standard counterparts. Table 2 shows the most likely constraint definitions for each violation profile, given either the baseline violation profiles (based on the standard constraint definitions) and the two-constraint runs of the model. On three of the four learned constraints, the model agrees with the inferred definition given the baseline violations. Reasons for the deviations from the phonologically standard definitions are discussed in Sect. 7.

Experiment summary We performed four tests of the constraint learner. The model learned a set of constraints and weights that could explain observed data and effectively generalize to unobserved forms. In addition, we find that the constraint definitions it learns correspond with the definitions that come from a baseline set of constraints, although additional information is needed to identify the exact same constraints as the baseline set.

[6]These words are: *time, year, way, day, thing, world, school, state, family, student, group, country, problem, hand, part, place, case, week, company, system, program, question, government, number, night*

[7]The remaining analyses are limited to the two-constraint learning runs; the four-constraint solution represents convergence failure to a local optimum with joint probability (Eq. 3) well below the two-constraint solutions because of its lower probability M. Better exploration of the constraint space would move this run toward the two-constraint solution.

Standard	Baseline	Model Run 1	Model Run 2
[+STR][+STR]	[+STR][-SYL]	[+STR][-SYL]	[+STR][-SYL]
[-VOI][+VOI]	[-VOI][+VOI]	[-VOI][+VOI]	[-VOI,-STR][-SG,-HI]*[-NAS,+VOI]
Key: voi=voicing, str=strident, sg=spread glottis, hi=high, nas=nasal, syl=syllabic			

Table 2: The phonological standard definitions and the most likely constraint definitions inferred in the baseline and model runs. Baseline/model d likelihoods based on 10000 samples from $p(d|m)$.

7 Discussion

Definitional ambiguity Although the constraints extend seamlessly to new data and their violation profiles mostly match, Table 2 showed the constraint definitions don't quite match the standard definitions. This is because multiple definitions can have identical violation profiles, as there are many phonological features; for instance, based on the first constraint's violation profile, the model has learned to penalize sz and zz sequences, but not səz. Phonological theory says that this constraint's definition is [+STR][+STR], but given the available data, any feature that is negative for [z] and [s] but positive for [ə] (or vice versa) will produce the same violation pattern, and the model has no reason to prefer one to the other.[8]

The complex definition of the second constraint in the second model run arises similarly. Small differences (8% of violations) between the model and baseline violation profiles lead the model to infer this more complicated definition, which penalizes stems ending in voiceless non-stridents getting either the [əz] suffix (with [ə] matching the [-SG,-HI] bundle and [z] matching [-NAS,+VOI]) or the [z] suffix (with the Kleene star vacuously satisfied). Such stems should get the [s] suffix, so this constraint definition is consistent with the observed data, and overreaching by handling two constraints' function: penalizing [z] like the [-VOI][+VOI] constraint would, but also penalizing [əz], which is covered by the faithfulness constraint DEP.

Such definitional ambiguity can be reduced through simultaneous learning of multiple phonological phenomena. The [+STR][-SYL] definition could be ruled out by observing the faithful manifestation of s-initial onset clusters in English, as in *stop* or *spin*; the Kleene-star definition could be ruled out by faithful realizations of non-harmonious *kid* or *peg*. Such learning would also be more realistic, as learners generally observe and learn a range of phonological phenomena simultaneously.

Relaxing binarity One important remaining step is to allow for non-binary constraints in the model, which could be introduced in multiple ways. One possibility is to mimic non-binary constraints through multiple, overlapping binary constraints (Frank and Satta, 1998), though this would require changes to the current PCFG. Another possibility is to treat the existing binary matrix as an indicator of whether a constraint is violated and add a second matrix, with positive integer values, corresponding to the number of violations of that constraint. Griffiths and Ghahramani (2011) use a similar design to overcome the binary nature of an Indian Buffet Process for object recognition.

Theory testing Our model also represents a way to investigate the plausibility of different theoretical statements of a constraint, casting constraint selection through the lens of model comparison. In addition, if the underlying constraint grammar is varied, this model could be used to investigate the plausibility and effectiveness different potential grammars.

8 Conclusion

We presented a model for learning binary, symbolically-defined constraints in a log-linear model from a combination of observational data and an infinite grammar over constraint definitions. We tested this model on a morphophonological problem and showed that it accurately inferred the values of the constraints, and found appropriate constraint definitions (though with some issues of definitional ambiguity).

[8] In fact, $p(d|m)$ is approximately equal for a range of constraint definitions that include [+STR][-SYL], [+STR][+STR], [+STR][-LABIAL], and others.

Acknowledgements

We wish to thank Eric Baković, Klinton Bicknell, Dave Barner, Charles Elkan, Andy Kehler, the UCSD Computational Psycholinguistics Lab, the Phon Company, and the COLING reviewers for their discussions and feedback on this work. This research was supported by NSF award IIS-0830535 and an Alfred P. Sloan Foundation Research Fellowship to RL.

References

Doug Beeferman, Adam Berger, and John Lafferty. 1999. Statistical models for text segmentation. *Machine Learning*, 34:177–210.

Iris Berent, Colin Wilson, Gary F. Marcus, and Douglas K. Bemis. 2012. On the role of variables in phonology: Remarks on Hayes and Wilson 2008. *Linguistic Inquiry*, 43:97–119.

Paul Boersma and Bruce Hayes. 2001. Empirical tests of the Gradual Learning Algorithm. *Linguistic Inquiry*, 32:45–86.

Joan Bresnan, Anna Cueni, Tatiana Nikitina, and R. Harald Baayen. 2007. Predicting the dative alternation. In G. Bourne, I. Kraemer, and J. Zwarts, editors, *Cognitive Foundations of Interpretation*. Royal Netherlands Academy of Science, Amsterdam.

Philip S. Dale and Larry Fenson. 1996. Lexical development norms for young children. *Behavioral Research Methods, Instruments, and Computers*, 28:125–127.

Mark Davies. 2008. The Corpus of Contemporary American English: 450 million words, 1990-present.

Gabriel Doyle, Klinton Bicknell, and Roger Levy. 2014. Nonparametric learning of phonological constraints in Optimality Theory. In *Proceedings of the Association for Computational Linguistics*.

Robert Frank and Giorgio Satta. 1998. Optimality theory and the generative complexity of constraint violability. *Computational Linguistics*, 24:307–315.

Sharon Goldwater and Mark Johnson. 2003. Learning OT constraint rankings using a Maximum Entropy model. In *Proceedings of the Workshop on Variation within Optimality Theory*.

Noah Goodman, Joshua Tenebaum, Jacob Feldman, and Tom Griffiths. 2008. A rational analysis of rule-based concept learning. *Cognitive Science*, 32:108–154.

Dilan Görür, F. Jäkel, and Carl Rasmussen. 2006. A choice model with infinitely many latent features. In *Proceedings of the 23rd International Conference on Machine Learning*.

Thomas Griffiths and Zoubin Ghahramani. 2005. Infinite latent feature models and the Indian buffet process. Technical Report 2005-001, Gatsby Computational Neuroscience Unit.

Thomas Griffiths and Zoubin Ghahramani. 2011. The Indian Buffet Process: An introduction and review. *Journal of Machine Learning Research*, 12:1185–1224.

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

Bruce Hayes. 1995. *Metrical Stress Theory: Principles and Case Studies*. U. of Chicago, Chicago.

Bruce Hayes. 1999. Phonetically driven phonology: the role of optimality theory and inductive grounding. In M. Darnell, E. Moravcsik, M. Noonan, F. Newmeyer, & K. Wheatley, editor, *Formalism and Functionalism in Linguistics, vol. 1*. Benjamins.

John McCarthy. 2008. *Doing Optimality Theory*. Blackwell.

Radford Neal. 1994. Response to approximate Bayesian inference with the weighted likelihood bootstrap. *Journal of the Royal Statistical Society. Series B (Methodological)*, 56:3–48.

Michael Newton and Adrian Raftery. 1994. Approximate Bayesian inference with the weighted likelihood bootstrap. *Journal of the Royal Statistical Society. Series B (Methodological)*, 56:3–48.

Franz Josef Och and Hermann Ney. 2002. Discriminative training and maximum entropy models for statistical machine translation. In *Proceedings of the 40th Annual Meeting of the ACL.*

Hoifung Poon, Colin Cherry, and Kristina Toutanova. 2009. Unsupervised morphological segmentation with log-linear models. In *Proceedings of the North American Chapter of the Association for Computational Linguistics*.

Alan Prince and Paul Smolensky. 1993. Optimality Theory: Constraint interaction in generative grammar. Technical report, Rutgers Center for Cognitive Science.

Jason Riggle. 2009. Generating contenders. *Rutgers Optimality Archive*, 1044.

Jason Riggle. 2012. Phonological feature chart (v. 12.12). December.

Bruce Tesar and Paul Smolensky. 2000. *Learnability in Optimality Theory*. MIT Press.