Nonparametric Learning of Phonological Constraints in Optimality Theory

Gabriel Doyle
Department of Linguistics
UC San Diego
La Jolla, CA, USA 92093
gdoyle@ucsd.edu

Klinton Bicknell
Department of Linguistics
Northwestern University
Evanston, IL, USA 60208
kbicknell@northwestern.edu

Roger Levy
Department of Linguistics
UC San Diego
La Jolla, CA, USA 92093
rlevy@ucsd.edu

Abstract

We present a method to jointly learn features and weights directly from distributional data in a log-linear framework. Specifically, we propose a non-parametric Bayesian model for learning phonological markedness constraints directly from the distribution of input-output mappings in an Optimality Theory (OT) setting. The model uses an Indian Buffet Process prior to learn the feature values used in the log-linear method, and is the first algorithm for learning phonological constraints without presupposing constraint structure. The model learns a system of constraints that explains observed data as well as the phonologically-grounded constraints of a standard analysis, with a violation structure corresponding to the standard constraints. These results suggest an alternative data-driven source for constraints instead of a fully innate constraint set.

1 Introduction

Many aspects of human cognition involve the interaction of constraints that push a decision-maker toward different options, whether in something so trivial as choosing a movie or so important as a fight-or-flight response. These constraint-driven decisions can be modeled with a log-linear system. In these models, a set of constraints is weighted and their violations are used to determine a probability distribution over outcomes. But where do these constraints come from?

We consider this question by examining the dominant framework in modern phonology, Optimality Theory (Prince and Smolensky, 1993, OT), implemented in a log-linear framework, MaxEnt OT (Goldwater and Johnson, 2003), with output forms' probabilities based on a weighted sum of constraint violations. OT analyses generally assume that the constraints are innate and universal, both to obviate the problem of learning constraints' identities and to limit the set of possible languages.

We propose a new approach: to learn constraints with limited innate phonological knowledge by identifying sets of constraint violations that explain the observed distributional data, instead of selecting constraints from an innate set of constraint definitions. Because the constraints are identified as sets of violations, this also permits constraints specific to a given language to be learned. This method, which we call IBPOT, uses an Indian Buffet Process (IBP) prior to define the space of possible constraint violation matrices, and uses Bayesian reasoning to identify constraint matrices likely to have generated the observed data. In identifying constraints solely by their extensional violation profiles, this method does not directly identify the intensional definitions of the identified constraints, but to the extent that the resulting violation profiles are phonologically interpretable, we may conclude that the data themselves guide constraint identification. We test IBPOT on tongue-root vowel harmony in Wolof, a West African language.

The set of constraints learned by the model satisfy two major goals: they explain the data as well as the standard phonological analysis, and their violation structures correspond to the standard constraints. This suggests an alternative data-driven genesis for constraints, rather than the traditional assumption of fully innate constraints.

2 Phonology and Optimality Theory

2.1 OT structure

Optimality Theory has been used for constraint-based analysis of many areas of language, but we focus on its most successful application: phonology. We consider an OT analysis of the mappings

Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, pages 1094–1103,
Baltimore, Maryland, USA, June 23-25 2014.

between underlying forms and their phonological manifestations – i.e., mappings between forms in the mental lexicon and the actual vocalized forms of the words.[1]

Stated generally, an OT system takes some input, generates a set of candidate outputs, determines what constraints each output violates, and then selects a candidate output with a relatively unobjectionable violation profile. To do this, an OT system contains four major components: a generator GEN, which generates candidate output forms for the input; a set of constraints CON, which penalize candidates; a evaluation method EVAL, which selects an winning candidate; and H, a language-particular weighting of constraints that EVAL uses to determine the winning candidate. Previous OT work has focused on identifying the appropriate formulation of EVAL and the values and acquisition of H, while taking GEN and CON as given. Here, we expand the learning task by proposing an acquisition method for CON.

To learn CON, we propose a data-driven markedness constraint learning system that avoids both innateness and tractability issues. Unlike previous OT learning methods, which assume known constraint definitions and only learn the relative strength of these constraints, the IBPOT learns constraint violation profiles and weights for them simultaneously. The constraints are derived from sets of violations that effectively explain the observed data, rather than being selected from a preexisting set of possible constraints.

2.2 OT as a weighted-constraint method

Although all OT systems share the same core structure, different choices of EVAL lead to different behaviors. In IBPOT, we use the log-linear EVAL developed by Goldwater and Johnson (2003) in their MaxEnt OT system. MEOT extends traditional OT to account for variation (cases in which multiple candidates can be the winner), as well as gradient/probabilistic productions (Anttila, 1997) and other constraint interactions (e.g., cumulativity) that traditional OT cannot handle (Keller, 2000). MEOT also is motivated by the general MaxEnt framework, whereas most other OT formulations are ad hoc constructions specific to phonology.

In MEOT, each constraint C_i is associated with a weight $w_i < 0$. (Weights are always negative in OT; a constraint violation can never make a candidate more likely to win.) For a given input-candidate pair (x, y), $f_i(y, x)$ is the number of violations of constraint C_i by the pair. As a maximum entropy model, the probability of y given x is proportional to the exponential of the weighted sum of violations, $\sum_i w_i f_i(y, x)$. If $\mathcal{Y}(x)$ is the set of all output candidates for the input x, then the probability of y as the winning output is:

$$p(y|x) = \frac{\exp\left(\sum_i w_i f_i(y, x)\right)}{\sum_{z \in \mathcal{Y}(x)} \exp\left(\sum_i w_i f_i(z, x)\right)} \quad (1)$$

This formulation represents a probabilistic extension of the traditional formulation of OT (Prince and Smolensky, 1993). Traditionally, constraints form a strict hierarchy, where a single violation of a high-ranked constraint is worse than any number of violations of lower-ranked constraints. Traditional OT is also deterministic, with the optimal candidate always selected. In MEOT, the constraint weights define hierarchies of varying strictness, and some probability is assigned to all candidates. If constraints' weights are close together, multiple violations of lower-weighted constraints can reduce a candidate's probability below that of a competitor with a single high-weight violation. As the distance between weights in MEOT increases, the probability of a suboptimal candidate being chosen approaches zero; thus the traditional formulation is a limit case of MEOT.

2.3 OT in practice

Figure 1 shows *tableaux*, a visualization for OT, applied in Wolof (Archangeli and Pulleyblank, 1994; Boersma, 1999). We are interested in four Wolof constraints that combine to induce vowel harmony: *ɪ, PARSE[rtr], HARMONY, and PARSE[atr]. The meaning of these constraints will be discussed in Sect. 4.1; for now, we will only consider their violation profiles. Each column represents a constraint, with weights decreasing left-to-right. Each tableau looks at a single input form, noted in the top-left cell: ete, ɛtɛ, ɪte, or itɛ.

Each row is a candidate output form. A black cell indicates that the candidate, or input-candidate pair, violates the constraint in that column.[2] A white cell indicates no violation. Grey stripes are

[1] Although phonology is usually framed in terms of sound, sign languages also have components that serve equivalent roles in the physical realization of signs (Stokoe, 1960).

[2] In general, a constraint can be violated multiple times by a given candidate, but we will be using binary constraints (violated or not) in this work. See Sect. 5.2 for further discussion.

ete	*ɪ	Parse(rtr)	Harmony	Parse(atr)	Score
ete					0
ɛte			■	▨	-24
etɛ			■	▨	-24
ɛtɛ				■	-8

ɪte	*ɪ	Parse(rtr)	Harmony	Parse(atr)	Score
ite		■			-32
ɪte	■	▨	▨	▨	-80
itɛ		■	■	▨	-56
ɪtɛ	■	▨	▨	▨	-72

ɛtɛ	*ɪ	Parse(rtr)	Harmony	Parse(atr)	Score
ete		■	▨	▨	-32
ɛte		■	▨	▨	-48
etɛ		■	▨	▨	-48
ɛtɛ			▨	▨	0

itɛ	*ɪ	Parse(rtr)	Harmony	Parse(atr)	Score
ite		■	▨	▨	-32
ɪte	■	▨	▨	▨	-120
itɛ			▨	▨	-16
ɪtɛ	■	▨	▨	▨	-72

Figure 1: Tableaux for the Wolof input forms ete, ɛtɛ, ɪte, and itɛ. Black indicates violation, white no violation. Scores are calculated for a MaxEnt OT system with constraint weights of -64, -32, -16, and -8, approximating a traditional hierarchical OT design. Values of grey-striped cells have negligible effects on the distribution (see Sect. 4.3).

overlaid on cells whose value will have a negligible impact on the distribution due to the values of higher-ranked constraint.

Constraints fall into two categories, faithfulness and markedness, which differ in what information they use to assign violations. Faithfulness constraints penalize mismatches between the input and output, while markedness constraints consider only the output. Faithfulness violations include phoneme additions or deletions between the input and output; markedness violations include penalizing specific phonemes in the output form, regardless of whether the phoneme is present in the input.

In MaxEnt OT, each constraint has a weight, and the candidates' scores are the sums of the weights of violated constraints. In the ete tableau at top left, output ete has no violations, and therefore a score of zero. Outputs ɛte and etɛ violate both HARMONY (weight 16) and PARSE[atr] (weight 8), so their scores are 24. Output ɛtɛ violates PARSE[atr], and has score 8. Thus the log-probability of output ɛtɛ is 1/8 that of ete, and the log-probability of disharmonious ɛte and etɛ are each 1/24 that of ete. As the ratio between scores increases, the log-probability ratios can become arbitrarily close to zero, approximating the deterministic situation of traditional OT.

2.4 Learning Constraints

Choosing a winning candidate presumes that a set of constraints CON is available, but where do these constraints come from? The standard assumption within OT is that CON is innate and universal. But in the absence of direct evidence of innate constraints, we should prefer a method that can derive the constraints from cognitively-general learning over one that assumes they are pre-specified. Learning appropriate model features has been an important idea in the development of constraint-based models (Della Pietra et al., 1997).

The innateness assumption can induce tractability issues as well. The strictest formulation of innateness posits that virtually all constraints are shared across all languages, even when there is no evidence for the constraint in a particular language (Tesar and Smolensky, 2000). Strict universality is undermined by the extremely large set of constraints it must weight, as well as the possible existence of language-particular constraints (Smith, 2004).

A looser version of universality supposes that constraints are built compositionally from a set of constraint templates or primitives or phonological features (Hayes, 1999; Smith, 2004; Idsardi, 2006; Riggle, 2009). This version allows language-particular constraints, but it comes with a computational cost, as the learner must be able to generate and evaluate possible constraints while learning the language's phonology. Even with relatively simple constraint templates, such as the phonological constraint learner of Hayes and Wilson (2008), the number of possible constraints expands exponentially. Depending on the specific formulation of the constraints, the constraint identification problem may even be NP-hard (Idsardi, 2006; Heinz et al., 2009). Our approach of casting the learning problem as one of identifying violation profiles is an attempt to determine the amount that can be learned about the active constraints in a paradigm without hypothesizing intensional constraint definitions. The violation profile informa-

tion used by our model could then be used to narrow the search space for intensional constraints, either by performing post-hoc analysis of the constraints identified by our model or by combining intensional constraint search into the learning process. We discuss each of these possibilities in Section 5.2.

Innateness is less of a concern for faithfulness than markedness constraints. Faithfulness violations are determined by the changes between an input form and a candidate, yielding an independent motivation for a universal set of faithfulness constraints (McCarthy, 2008). Some markedness constraints can also be motivated in a universal manner (Hayes, 1999), but many markedness constraints lack such grounding.[3] As such, it is unclear where a universal set of markedness constraints would come from.

3 The IBPOT Model

3.1 Structure

The IBPOT model defines a generative process for mappings between input and output forms based on three latent variables: the constraint violation matrices F (faithfulness) and M (markedness), and the weight vector w. The cells of the violation matrices correspond to the number of violations of a constraint by a given input-output mapping. F_{ijk} is the number of violations of faithfulness constraint F_k by input-output pair type (x_i, y_j); M_{jl} is the number of violations of markedness constraint $M_{\cdot l}$ by output candidate y_j. Note that M is shared across inputs, as M_{jl} has the same value for all input-output pairs with output y_j. The weight vector w provides weight for both F and M. Probabilities of output forms are given by a log-linear function:

$$p(y_j|x_i) = \frac{\exp\left(\sum_k w_k F_{ijk} + \sum_l w_l M_{jl}\right)}{\sum_{y_z \in \mathcal{Y}(x_i)} \exp\left(\sum_k w_k F_{izk} + \sum_l w_l M_{zl}\right)} \quad (2)$$

Note that this is the same structure as Eq. 1 but with faithfulness and markedness constraints listed separately. As discussed in Sect. 2.4, we assume that F is known as part of the output of GEN (Riggle, 2009). The goal of the IBPOT model is to learn the markedness matrix M and weights w for both the markedness and faithfulness constraints.

As for M, we need a non-parametric prior, as there is no inherent limit to the number of markedness constraints a language will use. We use the Indian Buffet Process (Griffiths and Ghahramani, 2005), which defines a proper probability distribution over binary feature matrices with an unbounded number of columns. The IBP can be thought of as representing the set of dishes that diners eat at an infinite buffet table. Each diner (i.e., output form) first draws dishes (i.e., constraint violations) with probability proportional to the number of previous diners who drew it: $p(M_{jl} = 1|\{M_{zl}\}_{z<j}) = n_l/j$. After choosing from the previously taken dishes, the diner can try additional dishes that no previous diner has had. The number of new dishes that the j-th customer draws follows a Poisson(α/j) distribution. The complete specification of the model is then:

$$M \sim IBP(\alpha); \qquad \mathcal{Y}(x_i) = Gen(x_i)$$
$$w \sim -\Gamma(1,1); \quad y|x_i \sim LogLin(M, F, w, \mathcal{Y}(x_i))$$

3.2 Inference

To perform inference in this model, we adopt a common Markov chain Monte Carlo estimation procedure for IBPs (Görür et al., 2006; Navarro and Griffiths, 2007). We alternate approximate Gibbs sampling over the constraint matrix M, using the IBP prior, with a Metropolis-Hastings method to sample constraint weights w.

We initialize the model with a randomly-drawn markedness violation matrix M and weight vector w. To learn, we iterate through the output forms y_j; for each, we split $M_{-j\cdot}$ into "represented" constraints (those that are violated by at least one output form other than y_j) and "non-represented" constraints (those violated only by y_j). For each represented constraint $M_{\cdot l}$, we re-sample the value for the cell M_{jl}. All non-represented constraints are removed, and we propose new constraints, violated only by y_j, to replace them. After each iteration through M, we use Metropolis-Hastings to update the weight vector w.

Represented constraint sampling We begin by resampling M_{jl} for all represented constraints $M_{\cdot l}$, conditioned on the rest of the violations $(M_{-(jl)}, F)$ and the weights w. This is the sampling counterpart of drawing existing features in the IBP generative process. By Bayes' Rule, the

[3] McCarthy (2008, §4.8) gives examples of "ad hoc" intersegmental constraints. Even well-known constraint types, such as generalized alignment, can have disputed structures (Hyde, 2012).

posterior probability of a violation is proportional to product of the likelihood $p(Y|M_{jl} = 1, M_{-jl}, F, w)$ from Eq. 2 and the IBP prior probability $p(M_{jl} = 1|M_{-jl}) = n_{-jl}/n$, where n_{-jl} is the number of outputs other than y_j that violate constraint $M_{\cdot l}$.

Non-represented constraint sampling After sampling the represented constraints for y_j, we consider the addition of new constraints that are violated only by y_j. This is the sampling counterpart to the Poisson draw for new features in the IBP generative process. Ideally, this would draw new constraints from the infinite feature matrix; however, this requires marginalizing the likelihood over possible weights, and we lack an appropriate conjugate prior for doing so. We approximate the infinite matrix with a truncated Bernoulli draw over unrepresented constraints (Görür et al., 2006). We consider in each sample at most K^* new constraints, with weights based on the auxiliary vector w^*. This approximation retains the unbounded feature set of the IBP, as repeated sampling can add more and more constraints without limit.

The auxiliary vector w^* contains the weights of all the constraints that have been removed in the previous step. If the number of constraints removed is less than K^*, w^* is filled out with draws from the prior distribution over weights. We then consider adding any subset of these new constraints to M, each of which would be violated only by y_j. Let M^* represent a (possibly empty) set of constraints paired with a subset of w^*. The posterior probability of drawing M^* from the truncated Bernoulli distribution is the product of the prior probability of M^* $\left(\frac{\frac{\alpha}{K^*}}{N_Y + \frac{\alpha}{K^*}}\right)$ and the likelihood $p(Y|M^*, w^*, M, w, F)$, including the new constraints M^*.

Weight sampling After sampling through all candidates, we use Metropolis-Hastings to estimate new weights for both constraint matrices. Our proposal distribution is $Gamma({w_k}^2/\eta, \eta/w_k)$, with mean w_k and mode $w_k - \frac{\eta}{w_k}$ (for $w_k > 1$). Unlike Gibbs sampling on the constraints, which occurs only on markedness constraints, weights are sampled for both markedness and faithfulness features.

4 Experiment

4.1 Wolof vowel harmony

We test the model by learning the markedness constraints driving Wolof vowel harmony (Archangeli and Pulleyblank, 1994). Vowel harmony in general refers to a phonological phenomenon wherein the vowels of a word share certain features in the output form even if they do not share them in the input. In the case of Wolof, harmony encourages forms that have consistent tongue root positions.

The Wolof vowel system has two relevant features, tongue root position and vowel height. The tongue root can either be advanced (ATR) or retracted (RTR), and the body of the tongue can be in the high, middle, or low part of the mouth. These features define six vowels:

	high	mid	low
ATR	i	e	ə
RTR	ɪ	ɛ	a

We test IBPOT on the harmony system provided in the Praat program (Boersma, 1999), previously used as a test case by Goldwater and Johnson (2003) for MEOT learning with known constraints. This system has four constraints:[4]

- Markedness:
  - *ɪ: do not have ɪ (high RTR vowel)
  - HARMONY: do not have RTR and ATR vowels in the same word
- Faithfulness:
  - PARSE[rtr]: do not change RTR input to ATR output
  - PARSE[atr]: do not change ATR input to RTR output

These constraints define the phonological standard that we will compare IBPOT to, with a ranking from strongest to weakest of *ɪ >> PARSE[rtr] >> HARMONY >> PARSE[atr]. Under this ranking, Wolof harmony is achieved by changing a disharmonious ATR to an RTR, unless this creates an ɪ vowel. We see this in Figure 1, where three of the four winners are harmonic, but with input itɛ, harmony would require violating one of the two higher-ranked constraints. As in previous MEOT work, all Wolof candidates are faithful

[4]The version in Praat includes a fifth constraint, but its value never affects the choice of output in our data and is omitted in this analysis.

with respect to vowel height, either because height changes are not considered by GEN, or because of a high-ranked faithfulness constraint blocking height changes.[5]

The Wolof constraints provide an interesting testing ground for the model, because it is a small set of constraints to be learned, but contains the HARMONY constraint, which can be violated by non-adjacent segments. Non-adjacent constraints are difficult for string-based approaches because of the exponential number of possible relationships across non-adjacent segments. However, the Wolof results show that by learning violations directly, IBPOT does not encounter problems with non-adjacent constraints.

The Wolof data has 36 input forms, each of the form $V_1 t V_2$, where V_1 and V_2 are vowels that agree in height. Each input form has four candidate outputs, with one output always winning. The outputs appear for multiple inputs, as shown in Figure 1. The candidate outputs are the four combinations of tongue-roots for the given vowel heights; the inputs and candidates are known to the learner. We generate simulated data by observing 1000 instances of the winning output for each input.[6] The model must learn the markedness constraints *ɪ and HARMONY, as well as the weights for all four constraints.

We make a small modification to the constraints for the test data: all constraints are limited to binary values. For constraints that can be violated multiple times by an output (e.g., *ɪ twice by ɪtɪ), we use only a single violation. This is necessary in the current model definition because the IBP produces a prior over binary matrices. We generate the simulated data using only single violations of each constraint by each output form. Overcoming the binarity restriction is discussed in Sect. 5.2.

4.2 Experiment Design

We run the model for 10000 iterations, using deterministic annealing through the first 2500 iterations. The model is initialized with a random markedness matrix drawn from the IBP and weights from the exponential prior. We ran versions of the model with parameter settings between 0.01 and 1 for α, 0.05 and 0.5 for η, and 2 and 5 for K^*. All these produced quantitatively similar results; we report values for $\alpha = 1$, $\eta = 0.5$, and $K^* = 5$, which provides the least bias toward small constraint sets.

To establish performance for the phonological standard, we use the IBPOT learner to find constraint weights but do not update M. The resultant learner is essentially MaxEnt OT with the weights estimated through Metropolis sampling instead of gradient ascent. This is done so that the IBPOT weights and phonological standard weights are learned by the same process and can be compared. We use the same parameters for this baseline as for the IBPOT tests. The results in this section are based on nine runs each of IBPOT and MEOT; ten MEOT runs were performed but one failed to converge and was removed from analysis.

4.3 Results

A successful set of learned constraints will satisfy two criteria: achieving good data likelihood (no worse than the phonological-standard constraints) and acquiring constraint violation profiles that are phonologically interpretable. We find that both of these criteria are met by IBPOT on Wolof.

Likelihood comparison First, we calculate the joint probability of the data and model given the priors, $p(Y, M, w|F, \alpha)$, which is proportional to the product of three terms: the data likelihood $p(Y|M, F, w)$, the markedness matrix probability $p(M|\alpha)$, and the weight probability $p(w)$. We present both the mean and MAP values for these over the final 1000 iterations of each run. Results are shown in Table 1.

All eight differences are significant according to t-tests over the nine runs. In all cases but mean M, the IBPOT method has a better log-probability. The most important differences are those in the data probabilities, as the matrix and weight probabilities are reflective primarily of the choice of prior. By both measures, the IBPOT constraints explain the observed data better than the phonologically standard constraints.

Interestingly, the mean M probability is lower for IBPOT than for the phonological standard. Though the phonologically standard constraints

[5] In the present experiment, we assume that GEN does not generate candidates with unfaithful vowel heights. If unfaithful vowel heights were allowed by GEN, these unfaithful candidates would incur a violation approximately as strong as *ɪ, as neither unfaithful-height candidates nor ɪ candidates are attested in the Wolof data.

[6] Since data, matrix, and weight likelihoods all shape the learned constraints, there must be enough data for the model to avoid settling for a simple matrix that poorly explains the data. This represents a similar training set size to previous work (Goldwater and Johnson, 2003; Boersma and Hayes, 2001).

	MAP		Mean	
	IBPOT	PS	IBPOT	PS
Data	-1.52	-3.94	-5.48	-9.23
M	-51.7	-53.3	-54.7	-53.3
w	-44.2	-71.1	-50.6	-78.1
Joint	-97.4	-128.4	-110.6	-140.6

Table 1: Data, markedness matrix, weight vector, and joint log-probabilities for the IBPOT and the phonological standard constraints. MAP and mean estimates over the final 1000 iterations for each run. All IBPOT/PS differences are significant ($p < .005$ for MAP M; $p < .001$ for others).

exist independently of the IBP prior, they fit the prior better than the average IBPOT constraints do. This shows that the IBP's prior preferences can be overcome in order to have constraints that better explain the data.

Constraint comparison Our second criterion is the acquisition of meaningful constraints, that is, ones whose violation profiles have phonologically-grounded explanations. IBPOT learns the same number of markedness constraints as the phonological standard (two); over the final 1000 iterations of the model runs, 99.2% of the iterations had two markedness constraints, and the rest had three.

Turning to the form of these constraints, Figure 2 shows violation profiles from the last iteration of a representative IBPOT run.[7] Because vowel heights must be faithful between input and output, the Wolof data is divided into nine separate *paradigms*, each containing the four candidates (ATR/RTR $\times$ ATR/RTR) for the vowel heights in the input.

The violations on a given output form only affect probabilities within its paradigm. As a result, learned constraints are consistent within paradigms, but across paradigms, the same constraint may serve different purposes.

For instance, the strongest learned markedness constraint, shown as *M1* in Figure 2, has the same violations as the top-ranked constraint that actively distinguishes between candidates in each paradigm. For the five paradigms with at least one high vowel (the top row and left column), *M1* has the same violations as *ɪ, as *ɪ penalizes some but not all of the candidates. In the other four paradigms, *ɪ penalizes none of the candidates, and the IBPOT learner has no reason to learn it. Instead, it learns that *M1* has the same violations as HARMONY, which is the highest-weighted constraint that distinguishes between candidates in these paradigms. Thus in the high-vowel paradigms, *M1* serves as *ɪ, while in the low/mid-vowel paradigms, it serves as HARMONY.

The lower-weighted *M2* is defined noisily, as the higher-ranked *M1* makes some values of *M2* inconsequential. Consider the top-left paradigm of Figure 2, the high-high input, in which only one candidate does not violate *M1* (*ɪ). Because *M1* has a much higher weight than *M2*, a violation of *M2* has a negligible effect on a candidate's probability.[8] In such cells, the constraint's value is influenced more by the prior than by the data. These inconsequential cells are overlaid with grey stripes in Figure 2.

The meaning of *M2*, then, depends only on the consequential cells. In the high-vowel paradigms, *M2* matches HARMONY, and the learned and standard constraints agree on all consequential violations, despite being essentially at chance on the indistinguishable violations (58%). On the non-high paradigms, the meaning of *M2* is unclear, as HARMONY is handled by *M1* and *ɪ is unviolated. In all four paradigms, the model learns that the RTR-RTR candidate violates *M2* and the ATR-ATR candidate does not; this appears to be the model's attempt to reinforce a pattern in the lowest-ranked faithfulness constraint (PARSE[atr]), which the ATR-ATR candidate never violates.

Thus, while the IBPOT constraints are not identical to the phonologically standard ones, they reflect a version of the standard constraints that is consistent with the IBPOT framework.[9] In paradigms where each markedness constraint distinguishes candidates, the learned constraints match the standard constraints. In paradigms where only one constraint distinguishes candidates, the top learned constraint matches it and the second learned constraint exhibits a pattern consistent with a low-ranked faithfulness constraint.

[7] Specifically, from the run with the median joint posterior.

[8] Given the learned weights in Fig. 2, if the losing candidate violates *M1*, its probability changes from 10^{-12} when the preferred candidate does not violate *M2* to 10^{-8} when it does.

[9] In fact, it appears this constraint organization is favored by IBPOT as it allows for lower weights, hence the large difference in w log-probability in Table 1.

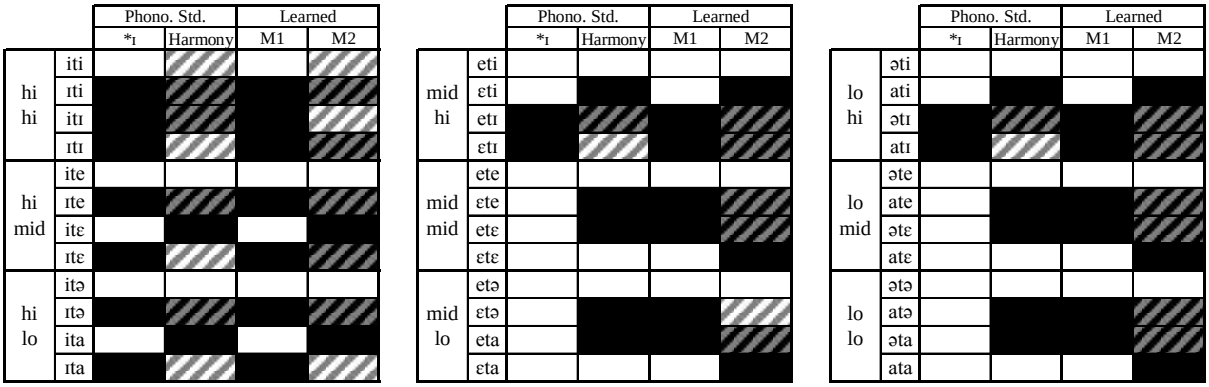

Figure 2: Phonologically standard (*ɪ, HARMONY) and learned (*M1,M2*) constraint violation profiles for the output forms. Learned weights for the standard constraints are -32.8 and -15.3; for *M1* and *M2*, they are -26.5 and -8.4. Black indicates violation, white no violation. Grey stripes indicate cells whose values have negligible effects on the probability distribution.

5 Discussion and Future Work

5.1 Relation to phonotactic learning

Our primary finding from IBPOT is that it is possible to identify constraints that are both effective at explaining the data and representative of theorized phonologically-grounded constraints, given only input-output mappings and faithfulness violations. Furthermore, these constraints are successfully acquired without any knowledge of the phonological structure of the data beyond the faithfulness violation profiles. The model's ability to infer constraint violation profiles without theoretical constraint structure provides an alternative solution to the problems of the traditionally innate and universal OT constraint set.

As it jointly learns constraints and weights, the IBPOT model calls to mind Hayes and Wilson's (2008) joint phonotactic learner. Their learner also jointly learns weights and constraints, but directly selects its constraints from a compositional grammar of constraint definitions. This limits their learner in practice by the rapid explosion in the number of constraints as the maximum constraint definition size grows. By directly learning violation profiles, the IBPOT model avoids this explosion, and the violation profiles can be automatically parsed to identify the constraint definitions that are consistent with the learned profile. The inference method of the two models is different as well; the phonotactic learner selects constraints greedily, whereas the sampling on M in IBPOT asymptotically approaches the posterior.

The two learners also address related but different phonological problems. The phonotactic learner considers phonotactic problems, in which only output matters. The constraints learned by Hayes and Wilson's learner are essentially OT markedness constraints, but their learner does not have to account for varied inputs or effects of faithfulness constraints.

5.2 Extending the learning model

IBPOT, as proposed here, learns constraints based on binary violation profiles, defined extensionally. A complete model of constraint acquisition should provide intensional definitions that are phonologically grounded and cover potentially non-binary constraints. We discuss how to extend the model toward these goals.

IBPOT currently learns extensional constraints, defined by which candidates do or do not violate the constraint. Intensional definitions are needed to extend constraints to unseen forms. Post hoc violation profile analysis, as in Sect. 4.3, provides a first step toward this goal. Such analysis can be integrated into the learning process using the Rational Rules model (Goodman et al., 2008) to identify likely constraint definitions compositionally. Alternately, phonological knowledge could be integrated into a joint constraint learning process in the form of a naturalness bias on the constraint weights or a phonologically-motivated replacement for the IBP prior.

The results presented here use binary constraints, where each candidate violates each constraint only once, a result of the IBP's restriction to binary matrices. Non-binarity can be handled by using the binary matrix M to indicate whether a candidate violates a constraint, with a second

distribution determining the number of violations. Alternately, a binary matrix can directly capture non-binary constraints; Frank and Satta (1998) converted existing non-binary constraints into a binary OT system by representing non-binary constraints as a set of equally-weighted overlapping constraints, each accounting for one violation. The non-binary harmony constraint, for instance, becomes a set {*(at least one disharmony), *(at least two disharmonies), etc.}.

Lastly, the Wolof vowel harmony problem provides a test case with overlaps in the candidate sets for different inputs. This candidate overlap helps the model find appropriate constraint structures. Analyzing other phenomena may require the identification of appropriate abstractions to find this same structural overlap. English regular plurals, for instance, fall into broad categories depending on the features of the stem-final phoneme. IBPOT learning in such settings may require learning an appropriate abstraction as well.

6 Conclusion

A central assumption of Optimality Theory has been the existence of a fixed inventory of universal markedness constraints innately available to the learner, an assumption by arguments regarding the computational complexity of constraint identification. However, our results show for the first time that nonparametric, data-driven learning can identify sparse constraint inventories that both accurately predict the data and are phonologically meaningful, providing a serious alternative to the strong nativist view of the OT constraint inventory.

Acknowledgments

We wish to thank Eric Baković, Emily Morgan, Mark Myslín, the UCSD Computational Psycholinguistics Lab, the Phon Company, and the 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

Arto Anttila. 1997. *Variation in Finnish phonology and morphology*. Ph.D. thesis, Stanford U.

Diana Archangeli and Douglas Pulleyblank. 1994. *Grounded phonology*. MIT Press.

Paul Boersma. 1999. Empirical tests of the Gradual Learning Algorithm. *Linguistic Inquiry*, 32:45–86.

Paul Boersma and Bruce Hayes. 2001. Optimality-theoretic learning in the Praat program. In *Proceedings of the Institute of Phonetic Sciences of the University of Amsterdam*.

Stephen Della Pietra, Vincent Della Pietra, and John Lafferty. 1997. Inducing features of random fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19:380–393.

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, Frank 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.

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

Bruce Hayes. 1999. Phonetically driven phonology: the role of optimality theory and inductive grounding. In Darnell et al, editor, *Formalism and Functionalism in Linguistics, vol. 1*. Benjamins.

Jeffrey Heinz, Gregory Kobele, and Jason Riggle. 2009. Evaluating the complexity of Optimality Theory. *Linguistic Inquiry*.

Brett Hyde. 2012. Alignment constraints. *Natural Language and Linguistic Theory*, 30:789–836.

William Idsardi. 2006. A simple proof that Optimality Theory is computationally intractable. *Linguistic Inquiry*, 37:271–275.

Frank Keller. 2000. *Gradience in grammar: Experimental and computational aspects of degrees of grammaticality*. Ph.D. thesis, U. of Edinburgh.

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

Daniel Navarro and Tom Griffiths. 2007. A nonparametric Bayesian method for inferring features from similarity judgments. In *Advances in Neural Information Processing Systems 19*.

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.

Jennifer Smith. 2004. Making constraints compositional: toward a compositional model of Con. *Lingua*, 114:1433–1464.

William Stokoe. 1960. *Sign Language Structure*. Linstok Press.

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