Joint learning of constraint weights and gradient inputs in Gradient Symbolic Computation with constrained optimization

Max Nelson
University of Massachusetts, Amherst
Amherst, MA, USA
manelson@umass.edu

Abstract

This paper proposes a method for the joint optimization of constraint weights and symbol activations within the Gradient Symbolic Computation (GSC) framework. The set of grammars representable in GSC is proven to be a subset of those representable with lexically-scaled faithfulness constraints. This fact is then used to recast the problem of learning constraint weights and symbol activations in GSC as a quadratically-constrained version of learning lexically-scaled faithfulness grammars. This results in an optimization problem that can be solved using Sequential Quadratic Programming.

1 Introduction and background

This paper proposes a method for the joint optimization of constraint weights and symbol activations within the Gradient Symbolic Computation (GSC) framework. The set of grammars representable in GSC is proven to be a subset of those representable with lexically-scaled faithfulness constraints. This fact is then used to recast the problem of learning constraint weights and symbol activations in GSC as a quadratically-constrained version of learning lexically-scaled faithfulness grammars. This results in an optimization problem that can be solved using Sequential Quadratic Programming.

The remainder of this paper proceeds as follows. The rest of this section provides the relevant background on GSC, previous approaches to the same problem, and maximum entropy grammars which are used in the proposed model. §2 describes and proves the relationship between GSC grammars and lexically-scaled faithfulness constraints and then uses this proof to develop the proposed learning algorithm. §3 illustrates with a minimal test case of an example used through the GSC literature, French Liaison. §4 provides a brief discussion and concludes.

1.1 Phonological grammars in Gradient Symbolic Computation

Gradient Symbolic Computation is a general cognitive framework in which structures are represented as gradient blends of multiple symbolic representations. Smolensky and Goldrick (2016) adapt standard optimality-theoretic constraints and optimization procedures to allow for inputs which consist of blends of symbolic structures. They propose that each position in the input is associated with a blend of discrete units, each of which is associated with an activation. In phonological terms an input may be composed of a series of positions, each of which is associated with a set of phonemes with different degrees of activation. The evaluation of constraints that make reference to the input, traditionally only faithfulness constraints, is done with respect to the activations of individual segments in the gradient representation. So if this partially active /t/ is fully realized, then a constraint like Dep, which penalizes epenthesis, will be violated to the degree that reflects the extent of this epenthesis: in this example, a violation of strength 0.3. Phonological grammars that allow for gradient inputs will henceforth be referred to as gradient symbolic grammars (GS grammars).

GS grammars have been employed to capture phonological phenomena that are difficult for traditional representational theories, including opacity (Mai et al., 2018), and exceptionality (Zimmerman, 2018; Hsu, 2018)/subregularity (Rosen, 2016; Smolensky and Goldrick, 2016).

1.2 Learning gradient symbolic grammars

GS grammars present a unique learning problem. In standard constraint-based grammars a phonological learner must discover the discrete underlying forms of the target language as well as the ranking or weighting of the constraints. In GS gram-

Proceedings of the Seventeenth SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology, pages 224–232

https://doi.org/10.18653/v1/P17

mars the learner has to learn these things as well, while also learning the activations of all symbols at all positions in the underlying form. The complete GS grammar learning problem, discovering the discrete units, their activations, and the constraint ordering, has not been addressed in previous literature and will not be addressed here. Previous work has however looked at different subparts of this problem, including the learning of activations in isolation (Rosen, 2019) and the parallel learning of activations and constraint weights (Rosen, 2016; Smolensky et al., 2020). This parallel problem is the topic of the present work.

Rosen (2016) presents an approach to jointly optimizing constraint weights and input activations based on simulated annealing which is able to successfully learn a grammar capturing Japanese rendaku. As will be discussed below, the joint optimization of weights and activations is non-convex so simulated annealing is a promising approach. This work will not attempt to improve on the empirical performance of a simulated annealing model, but rather it will propose an alternative approach which is more closely related to gradient-based methods used elsewhere in the phonological learning literature (Goldwater and Johnson, 2003; Boersma and Pater, 2008; Hayes and Wilson, 2008).

Smolensky et al. (2020) apply the Gradual Learning Algorithm (GLA) for Harmonic Grammar (Boersma and Pater, 2008), which is based on the Perceptron Update Rule (Rosenblatt, 1958), to the problem of learning both constraint weights and input activations. They report promising results, however the convergence proof for the GLA does not necessarily apply to the case of GS grammars, where multiple interacting parameters are being simultaneously optimized. As will be discussed later, activations add quadratic terms to the Harmony function. This means that Harmonies are not linear in the parameters and consequently the relationship between Harmonic Grammar and the Perceptron does not hold between GS grammar and the Perceptron.

This work presents a third approach to jointly learning activations and constraint weights, based on the fact that blended inputs represent a scaling function on faithfulness violations and on previous work which has explored the learning of scaled faithfulness. The presented model is also not guaranteed to converge on a global optimum, so it does not improve on the GLA approach in that respect. It does however have the benefit of casting the GS grammar learning problem as an explicit and well-understood optimization procedure while also relating it to a familiar problem, learning lexically-scaled constraint weights (Hughto et al., 2019).

1.3 Maximum entropy grammars

Unlike previous work in GSC, the learning algorithm in the present work will make use of Maximum Entropy (MaxEnt) Grammars (Goldwater and Johnson, 2003). A MaxEnt grammar is a log-linear model which allows for the probabilistic interpretation of a Harmonic Grammar (HG). In Harmonic Grammar the Harmony $\mathscr{H}$ of a candidate is the dot-product of its constraint violations and the constraint weights. Constraint violations are generally treated as strictly negative and weights as strictly positive, so given an input x a candidate y is optimal if it has the highest Harmony score in the set of all competing candidates $\mathcal{Y}(x)$.

$$\mathscr{H}_{(x,y)} = \sum_i w_i c_i(x, y) \tag{1}$$

A MaxEnt probability distribution is computed by applying the softmax function to the set of Harmonies.

$$p(x) = \frac{e^{\mathscr{H}_{(x,y)}}}{\sum_{\gamma \in \mathcal{Y}(x)} e^{\mathscr{H}_{(x,\gamma)}}} \tag{2}$$

MaxEnt grammars are used for the learning algorithm purely because it is intuitive to define an interpretable loss function when model outputs are a probability distribution, as will be discussed in §2.2. This is an expository choice: the learning algorithm presented below could be equivalently described as learning a Harmonic Grammar by minimizing a loss function that incorporates the softmax function. Because softmax is monotonic, a MaxEnt grammar makes the same prediction about the most well-formed candidate as its corresponding Harmonic Grammar.

2 Optimizing gradient symbolic grammars

2.1 Gradient symbolic computation as lexically scaled faithfulness

The observation driving the proposed learning algorithm for GS grammars is that GS grammars can be rewritten as a special case of lexically-scaled

faithfulness (LSF) grammars. An LSF grammar (Linzen et al., 2013) is a grammar in which all morphemes come with a set of scales which combine additively with constraint weights. This section aims to prove that the set of expressible GS grammars is a subset of the expressible LSF grammars.

In this work I assume all outputs are discrete structures and consequently only faithfulness constraints are gradiently evaluated[1]. Within the faithfulness constraints, Smolensky and Goldrick (2016) describe two classes in terms of how gradient activations in the input influence evaluation. Constraints belonging to the PROPORTIONAL class are violated to a degree proportional to the activation level of a deleted feature or segment, for example MAX constraints in Smolensky and Goldrick. Constraints belonging to the COMPLEMENT class are violated to a degree proportional to one minus the activation level of a realized feature or segment, for example DEP constraints in Smolensky and Goldrick. Introducing gradient inputs to the grammar results in a rescaling of faithfulness constraint violations and in no effect on markedness constraint violations[2].

Consider the simple GS tableau in (1), where α is the activation of the input segment b, M is the weight of a PROPORTIONAL constraint, and Δ is the weight of some COMPLEMENT constraint. Two hypothetical candidates are competing on which of the two constraints is violated. Note that Harmony is a quadratic function of the weights and activations.

(1)

	M	Δ	
$/b_\alpha/$	PROP	COMP	$\mathscr{H}$
ϕ	0	$1-\alpha$	$\Delta - \alpha\Delta$
ψ	α	0	αM

Now consider the grammar in (2), which uses lexically scaled faithfulness (LSF) constraints. The scales are indexed to the input morpheme(s) and combine additively with constraint weights. So the functional weight of PROP when evaluated on the ith morpheme is the general weight of PROP, M, added with the scale brought by morpheme i, μ_i (Linzen et al., 2013). In this case Harmony is a linear function of the weights and scales.

(2)

	μ_i	δ_i	
	M	Δ	
$/b/_i$	PROP	COMP	$\mathscr{H}$
[b]	0	1	$\Delta + \delta_i$
$\emptyset$	1	0	M + μ_i

The tableaux in (1) and (2) make identical predictions as long as the equalities in Eq. (3) hold. In other words if these equalities are true then the two grammars assign the exact same Harmonies to the candidates.

$$\begin{aligned} \Delta - \alpha\Delta &= \Delta + \delta_i \\ \alpha M &= M + \mu_i \end{aligned} \tag{3}$$

Given this fact, any GS grammar can be converted into an LSF grammar by replacing any morpheme's activation values with a set of scales. Scales for COMP and PROP constraints can be computed from activations by rearranging Eq. (3), as in Eq. (4).

$$\begin{aligned} \delta_i &= -\alpha\Delta \\ \mu_i &= \alpha M - M \end{aligned} \tag{4}$$

Eq. (4) proves that any function representable with a GS grammar can be expressed with an equivalent LSF grammar. The converse however is not true – there are functions representable in LSF grammars that are not representable in GS grammars. This is be illustrated by considering how Eq. (3) would be used to convert an arbitrary LSF grammar into a GS grammar. Converting in this direction requires computing activations from the set of lexical scales. By rearranging Eq. (3), we see that there are two ways to compute activations from a given LSF grammar. Activations can be computed either from the MAX constraints or from the DEP constraints.

$$\begin{aligned} \alpha &= \frac{\mu_i}{M} + 1 \\ \alpha &= -\frac{\delta_i}{\Delta} \end{aligned} \tag{5}$$

It is not possible for a single segment or feature to have multiple distinct activation levels. An LSF grammar is a valid GS grammar only if both methods of computing a yield the same result. So while there is an LSF grammar for every GS grammar,

[1] In GSC this is expressed a strong quantization constraint, which pushes outputs into discrete states (Smolensky et al., 2014; Cho et al., 2017)

[2] Zimmerman (2018) advocates for gradient outputs, which will allow for gradiently evaluated markedness constraints. The approach outlined below can be extended to cover this by allowing for lexically scaled markedness constraints as well

there is not a GS grammar for every LSF grammar. Only the subset of LSF grammars that satisfy the equality in Eq. (6) are valid GS grammars.

$$\frac{\mu_i}{M} + 1 = -\frac{\delta_i}{\Delta} \tag{6}$$

For simplicity, Eq. (6) can be rearranged as in Eq. (7).

$$\mu_i \Delta + M\Delta + \delta_i M = 0 \tag{7}$$

This does not necessarily mean anything about the linguistic expressivity of GS and LSF grammars. The conversion from GS to LSF grammar assumes that there are no limits on the constraint set and consequently may require theoretically unwieldy constraints. For example in order to capture the fact that there are separate activations at all positions in the input, there must be separate constraints for every feature at every position in the input. This point is ultimately unimportant for the present work, which aims to address the relationship between the mathematical, rather than linguistic, functions that are representable in the two theories with the purpose of leveraging this relationship to construct a learning algorithm for GS grammars. The next section will outline exactly how this subset-superset relationship can be used to to formulate the problem of simultaneously learning input activations and constraint weights as a quadratically constrained optimization problem.

2.2 Learning gradient symbolic grammars with constrained optimization

The relationship between GS and LSF grammars described above is useful because it allows the problem of learning constraint weights and activations to be related to a well-understood problem, learning constraint weights and additive scales. Additive scales are themselves a special case of another formalism, lexically-indexed constraints. Because the scaled violations combine additively in the Harmony function, lexical scales can be represented as indexed versions of their general form which always incur the same number of violations as the general form. Moore-Cantwell and Pater (2016) show that the problem of learning lexically-indexed constraint weights is no different than the standard MaxEnt optimization problem and Hughto et al. (2019) show that similar approaches can be taken to learning additive lexical scales. So, like in standard MaxEnt (Goldwater and Johnson, 2003), the task of learning an LSF grammar can be cast as optimizing the negative log-likelihood of the training data [3], which is convex in the constraint weights.

Unfortunately, because of the subset-superset relationship between GS and LSF grammars, the problem of learning GS grammars is not similarly reducible to the standard convex MaxEnt learning problem. Rather, the GS learning problem can be reduced to a constrained version of the LSF learning problem. Learning a GS grammar is equivalent to learning an LSF grammar subject to the hard constraint that the LSF grammar represents a possible GS grammar. This can be stated formally as the optimization problem in Eq. (8), where $p(x)$ is computed using the standard MaxEnt probability function in Eq. (2). The weight vector w includes: General PROP and COMP weights M and Δ, i lexically indexed scales on PROP $\mu_1, ..., \mu_i$, i lexically indexed scales on COMP $\delta_1, ..., \delta_i$, and n general markedness constraints $m^1, ..., m^n$. The rightmost term in the objective function is an L2 prior with strength λ.

$$w = [M, \Delta, \mu_1 ..., \mu_i, \delta_1 ..., \delta_i, m^1, ..., m^n]$$

$$\min_w \left[\left(-\sum_x log\ p(x) \right) + \lambda \mid\mid w \mid\mid_2 \right]$$

$$\text{Subject to: } \sum_i (\mu_i \Delta + M\Delta + \delta_i M)^2 = 0 \tag{8}$$

The constraint enforcing that the learned grammar is a viable GS grammar is the equality relationship in Eq. (7) summed over all input phonemes i. The constraint is squared within the sum to prevent positive and negative terms in the summation from canceling out. This ensures that activations computed for a given phoneme and morpheme index from both the PROP and COMP constraints will be guaranteed to return the same value.

There are a number of potential approaches to constrained optimization problems like that posed above. It is worth mentioning here why methods familiar in computational phonology will not work. Maximum Entropy and Harmonic Grammars are generally fit using projected gradient descent, which is itself a method of constrained optimization. This entails computing the weight update, independent of any constraints placed on the weights,

[3] Or other equivalent loss function, such as Kullback-Leibler divergence

and then projecting the updated weights onto the set defined by the constraint. The use familiar in phonology is in the enforcement non-negativity – a restriction against negative weights which maintains the theoretical tenant of Optimality Theory that constraints can penalize but not reward. In this case projected gradient descent is effective. Not only is the projection function simple to compute because the nearest non-negative number to any negative number is 0, but the space defined by the constraint is a convex set, meaning that projected gradient descent with this constraint has the same convergence guarantees as standard gradient descent (Levitin and Polyak, 1966). As defined in Eq. (8) the current problem is quadratically constrained, meaning that the set that satisfies the constraint is non-convex and a projection function onto the set is not easily computable. Consequently projected gradient descent is not only not guaranteed to converge, it is computationally intractable.

Another possible approach would be to treat the constraint as a prior. One simple issue with this is that priors are violable. Given that the goal is learn a GS grammar, the constraint on the solution space defined above cannot be violated. One possible workaround would be to set the strength of the prior arbitrarily high, making it functionally non-violable. However the intersection of the loss function and the space satisfying the constraint is non-convex and is not guaranteed to be connected. Consequently gradient descent and other widely applied optimization techniques are likely to fail.

The proposed solution is to use Sequential Quadratic Programming (SQP), an iterative generalization of Newton's method developed for minimizing a function under quadratic constraints. The general approach is to iteratively take the quadratic approximation of the constrained objective function at w, minimize this subproblem with quadratic programming, and then set w to the solution. This will yield increasingly better approximations and therefore increasingly better solutions. On a practical note, this requires computing the first three terms of the Taylor expansion of the objective function at a given point, meaning that it must be twice differentiable. For detailed derivation and discussion of the method see Boggs and Tolle (1995).

3 An example

To illustrate the promise of the proposed approach to learning GS grammars, this section applies it to a minimal example of the French liaison problem that Smolensky and Goldrick (2016) use to motivate the use of gradient representations in the phonological grammar. Liaison is a phenomenon in which, in certain syntactic contexts, a consonant surfaces between vowel-final and vowel-initial words when hiatus would otherwise occur. The identity of this consonant, the liaison consonant, is not phonologically predictable. There is a long literature on the phonological analysis on liaison and its interacting processes, including competing analyses that propose that the liaison consonant is specified by the first word (Tranel, 1996) and by the second word in the sequence (Morin, 2005).

There is a class of words which are phonologically vowel initial but exceptionally do not trigger the surfacing of a liaison consonant in environments where it is otherwise predicted to surface. These words are always the second word in the pair and are called the h-aspiré words, referencing the fact that they are orthographically h-initial.

Consider the following set of French surface forms. When *petit* comes together with *ami*, a vowel-initial word, the liaison consonant [t] surfaces.

[pøti] *petit* ‘small’	+	[ami] *ami* ‘friend’
[pøti͡tami]	*petit ami*	‘boyfriend’

However, when *petit* is followed by *héros*, an h-aspiré word, no liaison consonant surfaces.

[pøti] *petit* ‘small’	+	[eʁo] *héros* ‘hero’
[pøti͡eʁo]	*petit héros*	‘little hero’

The adjective [pøti] *petit* is associated with a liaison t. When it occurs in isolation the liaison consonant does not surface, however when it occurs before the vowel-initial [ami] *ami* the liaison consonant surfaces, preventing two adjacent vowels from surfacing. Despite being vowel-initial, the h-aspiré word [eʁo] *héros* does not trigger the surfacing of the liaison consonant when it surfaces after *peti*.

Smolensky and Goldrick (2016) offer an analysis of this phenomenon couched in Gradient Symbolic Computation, which suggests that the liaison consonant is specified by both the first and second word in the pair. In their analysis both words contain partially active edge consonants. When the words

surface together the combined activation is enough to get cause the liaison consonant to surface. In this analysis h-aspiré words differ from their liaison-participating counterparts in that they have no or minimal activation on liaison consonants at their left edge, preventing them from contributing to the combined activation.

In terms of the minimal dataset above, they propose that there is a partially-activated /t/ in the input at both the right edge of *peti* and at the left edge of *ami*. When either word occurs in isolation there is not sufficient activation of the /t/ for it to surface. When the two words surface adjacent to one another the combined activation of /t/ in both words overcomes a threshold and liaison [t] surfaces. In the h-aspiré *héros* there is little to no activation on an input /t/ at the left edge. Despite the consequence of realizing a marked vowel-vowel sequence, the liaison [t] does not surface between [pøti] and [eʁo] because the combined activation of the input /t/s is not enough to justify its realization. They argue that this analysis overcomes empirical shortcomings of analyses which place the onus of specifying the liaison consonant on exclusively the first or second consonant, see Smolensky and Goldrick (2016) and Smolensky et al. (2020) for detailed discussion.

As proof of concept a GS grammar was fit to these data using the procedure described above. Model parameters include the weight of three constraints, HIATUS, MAX(t) and DEP(t), as well as the activation levels of liaison /t/ at the left edge of *petit* and at the right edge of *ami* and *héros*. MAX(t) is a PROP constraint and DEP(t) is a COMP constraint. In every tableau there are two competing candidates, one in which [t] surfaces and one in which it does not. Activations were constrained to being positive by adding the constraint in Eq. (9) to the optimization procedure.

$$\sum_i \min\left(\frac{\mu_i}{M} + 1, 0\right) = 0 \tag{9}$$

In practice the Jacobian and Hessian of the objective function are estimated analytically, so the algorithm described above is non-deterministic. The quadratically-constrained optimization problem is also generally non-convex, so variation is expected across runs. Consequently 10 models were fit with weights randomly initialized in [-2,0). An L2 prior is included with $\lambda = 0.01$. Table (1) shows the average final probability of each candidate in the five tableaux across the 10 runs.

	Candidate	avg.	s.d.
▷	[pøti]	0.999	1e-4
	[pøtit]	0.001	
▷	[ami]	0.991	0.003
	[tami]	0.008	
▷	[eʁo]	0.999	2e-7
	[teʁo]	1e-6	
▷	[pøtit ami]	0.980	0.009
	[pøti ami]	0.020	
	[pøtit eʁo]	0.015	0.005
▷	[pøti eʁo]	0.985	

Table 1: Average final probability across 10 runs on all forms. ▷ indicates the target surface forms.

The average activations of input /t/s in all words are shown in Table (2). Recall that there are two possible ways to compute the activations, from the COMP or PROP constraints. To ensure that the model works correctly, both methods of computing activations are shown. Note that these are negligibly different, confirming that the final grammar is indeed a valid GS grammar.

	COMP	PROP
pøti(t)	0.296 (0.062)	0.296 (0.062)
(t)ami	0.614 (0.081)	0.614 (0.081)
(t)eʁo	-2e-5 (6e-5)	-1e-4 (2e-4)

Table 2: Average (s.d.) activation of liaison consonants in all words as computed from the Δ and M constraints.

The activations suggest that the model may be converging on a solution that resembles the analysis proposed by Smolensky and Goldrick. *Petit* and *ami* both have a partially-activated /t/ in the at the relevant edge, while the activation of liaison /t/ in *héros* is approximately 0. The individual tableaux confirm that the learned analysis resembles Smolensky and Goldrick's. For simplicity, and consistency with previous work, all tableaux will be presented without probabilities, as HG tableaux.

While *petit* and *ami* both have partially-activated underlying /t/s, the activation is low enough that when either of these words occur in isolation the /t/ is not realized. This is demonstrated in Tableaux (3) and (4).

(3)

/pøtit$_{0.30}$/	-13.1 DEP(t)	-5.3 HIATUS	-0.4 MAX(t)	$\mathscr{H}$
[pøti]	0	0	0.30	-0.12
[pøtit]	0.70	0	0	-9.59

(4)

/t$_{0.61}$ami/	-13.1 DEP(t)	-5.3 HIATUS	-0.4 MAX(t)	$\mathscr{H}$
[ami]	0	0	0.61	-0.24
[tami]	0.39	0	0	-5.12

In *héros* the underlying liaison /t/ has a 0 activation, so it trivially does not surface in isolation.

(5)

/t$_{0.00}$ɛʁo/	-13.1 DEP(t)	-5.3 HIATUS	-0.4 MAX(t)	$\mathscr{H}$
[ɛʁo]	0	0	0.0	-0.0
[tɛʁo]	1.0	0	0	-13.1

When *petit* and *ami* are realized next to one another, their combined activation, as well as the threat of a HIATUS violation, are enough to make the liaison consonant surface.

(6)

/pøti t$_{0.30+0.61}$ ami/	-13.1 DEP(t)	-5.3 HIATUS	-0.4 MAX(t)	$\mathscr{H}$
[pøtiami]	0	1	1.01	-5.70
[pøtitami]	0.01	0	0	-0.13

However this is not the case when *petit* and *héros* surface together. Because *héros* contributes 0 activation to /t/, the cost of epenthesizing the remaining activation needed for the /t/ to be realized does not outweigh the cost of incurring a HIATUS violation.

(7)

/pøti t$_{0.30+0.00}$ ɛʁo/	-13.1 DEP(t)	-5.3 HIATUS	-0.4 MAX(t)	$\mathscr{H}$
[pøtiɛʁo]	0	1	0.30	-5.42
[pøtitɛʁo]	0.70	0	0	-9.17

The presented learning algorithm for GS grammars reliably converges on the analysis of French liaison offered by Smolensky and Goldrick (2016) as a motivating pattern for the inclusion of gradient inputs in the phonological grammar. This serves to illustrate the fact that the proposed learning algorithm is capable of learning interpretable GS grammars and has promising application in future work, both in finding GSC analyses of linguistic phenomena and in evaluating the learnability of phenomena in the GSC framework.

4 Discussion and Conclusions

This paper has presented a method for the joint optimization of blended inputs and constraint weights in gradient symbolic grammars. The proposed method leverages the fact that the set of functions representable by GS grammars is a subset of those representable by lexically-scaled faithfulness grammars to cast the GS grammar learning problem as a constrained version of the LSF grammar learning problem. The primary aim of this work is to introduce and justify the method, rather than discuss its implications for linguistic theory, however points of interest to linguistic theory will be briefly addressed here.

The subset-superset relationship that was shown to hold between GS and LSF grammars does not make predictions regarding the expressivity of the two theories in terms of the linguistic phenomena they are capable of representing. It does, however, highlight differences between the two theories which may provide a starting point for comparing their linguistic expressivity. For example, representing GS grammars in the LSF framework requires a set of faithfulness constraints which make reference to every position in every input. This differs from standard approaches to positional faithfulness, where faithfulness constraints make reference to prosodic positions (Beckman, 1998), and may yield pathological predictions. Consequently, despite the fact that LSF grammars represent a greater range of functions, it is likely that there are phenomena that can be captured with GS grammars but not with LSF grammars given a limited constraint set. This is left to future work.

This work has also shown that the optimization problem for GS grammars is likely more difficult than the analogous problem in other frameworks designed to capture the same types of phonological phenomena. For example, grammars with lexically-scaled constraints like those mentioned throughout this paper have also been shown to capture lexical exceptionality and subregularity but, as described, they correspond to a convex optimization problem. Similarly, grammars with underlying representation constraints have also been shown to be a viable approach to capturing these phonological phenomena (Apoussidou, 2007; Smith, 2015) and, in learning problems like that described in this paper present a convex optimization problem. The critical difference between these approaches and GS grammars is that Harmony function for GS grammars is quadratic, consequently the optimization problem is not guaranteed to be convex. It is not necessarily the case that the complexity of the related optimization problems is a valid metric along which to compare linguistic theories. Previous work however, has made strong claims regarding the relationship between the numerical optimization of MaxEnt/HG grammars and the learning trajectories of

human language learners (Boersma et al., 2000; Jäger, 2007; Jesney and Tessier, 2008, 2011), in which case there may be merit in comparing the optimization procedure for competing theories.

The broader GSC framework offers a novel theory of phonological grammars, the expressivity and restrictiveness of which has not been thoroughly explored. This work hopes to facilitate further research by introducing a method for simultaneously learning constraint weights and input activations of GS grammars which both relates GS grammars to an existing phonological framework and serves as a tool in finding GS analyses of phonological phenomena.

Acknowledgments

Thank you to Katherine Blake, Gaja Jarosz, Andrew Lamont, Joe Pater, Brandon Prickett and everyone at UMass Sound Workshop for productive discussion of the ideas presented above, as well as to four anonymous SIGMORPHON reviewers for specific comments on this paper. All remaining errors are my own.

References

Diana Apoussidou. 2007. *The learnability of metrical phonology*. Ph.D. thesis, University of Amsterdam.

Jill N. Beckman. 1998. *Positional faithfulness*. Ph.D. thesis.

Paul Boersma, Clara Levelt, et al. 2000. Gradual constraint-ranking learning algorithm predicts acquisition order. In *Proceedings of Child Language Research Forum*, volume 30, pages 229–237. CSLI Publications Stanford, CA.

Paul Boersma and Joe Pater. 2008. Convergence properties of a gradual learning algorithm for harmonic grammar.

Paul T. Boggs and Jon W. Tolle. 1995. Sequential quadratic programming. *Acta numerica*, 4:1–51.

Pyeong Whan Cho, Matthew Goldrick, and Paul Smolensky. 2017. Incremental parsing in a continuous dynamical system: Sentence processing in gradient symbolic computation. *Linguistics Vanguard*, 3(1).

Sharon Goldwater and Mark Johnson. 2003. Learning OT constraint rankings using a maximum entropy model. *Proceedings of the Stockholm Workshop on Variation in Optimality Theory*, pages 111–120.

Bruce Hayes and Colin Wilson. 2008. A maximum entropy model of phonotactic and phonotactic learning. *Linguistic Inquiry*, 39(3):379–440.

Brian Hsu. 2018. Scalar constraints and gradient symbolic representations generate exceptional prosodification effects without exceptional prosody. In *Handout, West Coast Conference on Formal Linguistics*, volume 36.

Coral Hughto, Andrew Lamont, Brandon Prickett, and Gaja Jarosz. 2019. Learning exceptionality and variation with lexically scaled maxent. In *Proceedings of the Second Annual Meeting of the Society for Computation in Linguistics (SCiL)*.

Gerhard Jäger. 2007. Maximum entropy models and stochastic optimality theory. *Architectures, rules, and preferences: variations on themes by Joan W. Bresnan. Stanford: CSLI*, pages 467–479.

Karen Jesney and Anne-Michelle Tessier. 2008. Gradual learning and faithfulness: consequences of ranked vs. weighted constraints.

Karen Jesney and Anne-Michelle Tessier. 2011. Biases in harmonic grammar: the road to restrictive learning. *Natural Language & Linguistic Theory*, 29(1):251–290.

Evgeny S. Levitin and Boris T. Polyak. 1966. Constrained minimization methods. *USSR Computational mathematics and mathematical physics*, 6(5):1–50.

Tal Linzen, Sofya Kasyanenko, and Maria Gouskova. 2013. Lexical and phonological variation in russian prepositions. *Phonology*, 30(3):453–515.

Anna Mai, Eric Bakovic, and Matt Goldrick. 2018. Phonological opacity as local optimization in gradient symbolic computation. *Proceedings of the Society for Computation in Linguistics*, 1(1):219–220.

Claire Moore-Cantwell and Joe Pater. 2016. Gradient exceptionality in maximum entropy grammar with lexically specific constraints. *Catalan Journal of Linguistics*, 15:53–66.

Yves Charles Morin. 2005. La liaison relève-t-elle d'une tendance à éviter les hiatus? réflexions sur son évolution historique. *Langages*, (2):8–23.

Eric R. Rosen. 2016. Predicting the unpredictable: Capturing the apparent semi-regularity of rendaku voicing in japanese through harmonic grammar. In *Proceedings of BLS*, volume 42, pages 235–249.

Eric R. Rosen. 2019. Learning complex inflectional paradigms through blended gradient inputs. In *Proceedings of the Society for Computation in Linguistics (SCiL) 2019*, pages 102–112.

Frank Rosenblatt. 1958. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386.

Brian Smith. 2015. *Phonologically conditioned allomorphy and UR constraints*. Ph.D. thesis, University of Massachusetts Amherst.

Paul Smolensky and Matthew Goldrick. 2016. Gradient symbolic representations in grammar: The case of french liason. Technical report.

Paul Smolensky, Matthew Goldrick, and Donald Mathis. 2014. Optimization and quantization in gradient symbol systems: a framework for integrating the continuous and the discrete in cognition. *Cognitive science*, 38(6):1102–1138.

Paul Smolensky, Eric Rosen, and Matthew Goldrick. 2020. Learning a gradient grammar of French liaison. In *Proceedings of the 2019 Annual Meeting on Phonology, Stonybrook NY*.

Bernard Tranel. 1996. French liaison and elision revisited: A unified account within optimality theory. *Aspects of Romance linguistics*, pages 433–455.

Eva Zimmerman. 2018. Gradient Symbolic Representations in the output: A case study from Moses Columbian Salishan stress. In *Proceedings of the Forty-Eigth Annual Meeting of the North East Linguistic Society*.