Learning Hidden Structure with Maximum Entropy Grammar

Brandon Prickett and Joe Pater

University of Massachusetts Amherst

27th Manchester Phonology meeting

May 25th, 2019

MaxEnt Grammars in Phonological Analysis

- In Maximum Entropy grammars (Goldwater and Johnson 2003), underlying representations map to a probability distribution over possible surface representations.
- This allows phonologists to analyze variable processes.

“Categorical” Deletion Process

/bat/	NoCoda	Max			
Weights	50	1	**H**	**e^H**	**p(SR|UR)**
[bat]	-1	0	-50	~0	~0
[ba]	0	-1	-1	0.368	~1

Variable Deletion Process

/bat/	NoCoda	Max			
Weights	3	2	**H**	**e^H**	**p(SR|UR)**
[bat]	-1	0	-3	0.050	.27
[ba]	0	-1	-2	0.135	.73

- However, finding the weights that optimally describe a dataset often can’t be easily done by hand.

Finding the Weights for a Grammar

- Boersma (1997) introduced the Gradual Learning Algorithm for learning variation in ranking-based OT grammars (see also Boersma and Hayes 2001).
  - The closely related HG-GLA was developed to handle Harmonic Grammar (Legendre et al. 1990).
  - As long as a model has access to all of the relevant information about inputs and output candidates, the HG-GLA is guaranteed to converge on a categorical distribution (Boersma and Pater 2008).
- MaxEnt grammars typically use Gradient Descent or Conjugate Gradient Descent to find the optimal set of weights to describe a dataset.
  - Gradient Descent is related both to the Perceptron Update Rule (Rosenblatt 1958) and the algorithms discussed above.
  - It is guaranteed to converge on both probabilistic and categorical distributions, as long as the model has access to all of the information that's relevant to the pattern at hand (Berger et al. 1996, Fischer 2005: ROA).
  - Other optimizers, like L-BFGS-B (Byrd et al. 1995) have also been successfully applied to learning MaxEnt grammar weights (e.g. Pater et al. 2012, Culbertson et al. 2013).

Hidden Structure and Learning

- The convergence guarantees mentioned above hold only when the learner is provided with the full structure of the data.
- Footing is a common example of hidden structure: overt [babába] is compatible with at least two full structures, each violating different constraints (e.g. Trochee and Iamb).
- In phonological analysis, just as in learning, we are typically not given the full structure of the data.
- Given the overt forms of the language data, we have to infer hidden structures like Underlying Representations and prosodic structures.

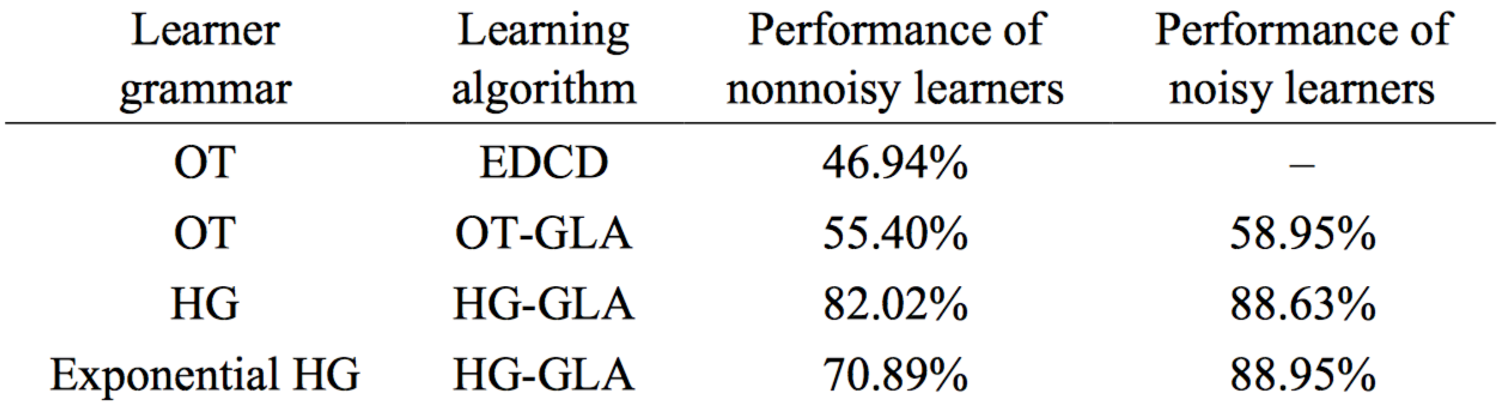
/bababa/	Trochee	Iamb
(babá)ba	-1	0
ba(bába)	0	-1

Hidden Structure and Analysis

- As another – this time real – example, consider *t/d*-deletion in English, as in [wɛs.bɛŋk] for "west bank".
- Given the observed faithful pronunciation in "west end", what is the structure? (Period = syllable boundary). Each one satisfies some constraint(s) that the other violates – there is no harmonic bounding.
  - [wɛs.tɛnd] ?
  - [wɛst.ɛnd] ?
- Coetzee and Pater (2011) show how some varieties of *t/d*-deletion can be analyzed in Stochastic OT, Noisy HG, and MaxEnt, using Praat-supplied learning algorithms to construct the analyses.
- Because C&P were working with learners that had no way of coping with hidden structure, they were limited to analyses without syllable structure, with constraints like Max-Prevocalic.
- This is of course not a general solution.

Robust interpretive parsing

- Tesar and Smolensky (2000) introduce Robust Interpretive Parsing, (RIP) and a test set of 124 stress patterns (12 constraints, 62 overt forms per language).
- Boersma and Pater (2008/2016) test RIP with a set of different grammar formalisms and learning algorithms:



Learner grammar	Learning algorithm	Performance of nonnoisy learners	Performance of noisy learners
OT	EDCD	46.94%	–
OT	OT-GLA	55.40%	58.95%
HG	HG-GLA	82.02%	88.63%
Exponential HG	HG-GLA	70.89%	88.95%

MaxEnt and hidden structure learning

- MaxEnt grammar is popular for phonological analysis at least in part because it is convenient: it's much more difficult in other approaches to probabilistic OT / HG to calculate the probabilities of candidate outcomes.
- MaxEnt learning can also be more convenient than learning in other probabilistic frameworks: under a fully batch approach, it is deterministic, so a single run is all that's needed for a given starting state (other approaches, which use sampling, need averaging of multiple runs).
- While hidden structure learning has been studied in MaxEnt (e.g. Pater et al. 2012, Nazarov and Pater 2017 and references therein), no one has provided results on the Tesar and Smolensky (2000) benchmarks.
- Here we show that a fully batch approach provides results as good as the best from Boersma and Pater's (2008/2016) study of on-line learners, and nearly as good as Jarosz's (2013, 2015) more recent state of the art results.

The Model

- In MaxEnt models, learning involves finding the optimal set of weights for a grammar—we define these to be weights that assign a probability distribution over overt forms that is similar to the distribution seen in the training data (formalized using KL-Divergence; Kullback and Leibler 1951).

- Our model uses two mechanisms to find these optimal weights for hidden structure patterns:
  - *L-BFGS-B Optimization* (Byrd et al. 1995): this is a quasi-Newton method that uses a limited amount of memory to find the optimal values for a set of parameters (in this case, constraint weights) whose values are bounded in some way (in this case, greater than 0).
  - *Expectation Maximization* (Dempster et al. 1977): this is a way of estimating the probability of data (in this case, of UR→SR mappings), when you don't have all of the information that's relevant to that estimate (in this case, constraint violations).

- Why L-BFGS-B?
  - We tried more standard optimization algorithms (like gradient descent and stochastic gradient descent) as well as more efficient ones (like Adam; Kingma and Ba 2014), but L-BFGS-B outperformed all the alternatives we checked.
  - It's also relatively easy to implement, since there are packages in most programming languages (e.g. Python and R) that perform the algorithm for you.

Expectation Maximization and MaxEnt

- Why expectation maximization? In hidden structure problems, you don’t necessarily know what constraints a given form will violate.

- Returning to our previous example, if you see the word [babába], you wouldn’t know which of the following foot structures to assign to it:

/bababa/	Trochee	Iamb
Weights	5	1
(babá)ba	-1	0
ba(bába)	0	-1

Expectation Maximization and MaxEnt

- Why expectation maximization? In hidden structure problems, you don’t necessarily know what constraints a given form will violate.

- Returning to our previous example, if you see the word [babába], you wouldn’t know which of the following foot structures to assign to it:

/bababa/	Trochee	Iamb			
Weights	5	1	**H**	**e^H**	**p(SR|UR)**
(babá)ba	-1	0	-5	0.007	.02
ba(bába)	0	-1	-1	0.368	.98

- Expectation Maximization allows us to estimate the probability of each structure, based on the current weights of our constraints.
  - So in the example above, we would assign a probability of 2% to the iambic parsing and a probability of 98% to the trochaic one, because our current grammar prefers trochees.
  - This is related to Robust Interpretive Parsing (Tesar and Smolensky 1998), RRIP, and EIP (Jarosz 2013).

The Learning Task

- To test how well our model learned patterns with hidden structure, we trained it on the 124 stress patterns laid out by Tesar and Smolensky (2000).
- These patterns are a sample of the factorial typology for 12 constraints:

- *WSP*: stress heavy syllables.
- *FootNonFinal*: head syllables must not come foot final.
- *Iambic*: head syllables must come foot final.
- *Parse*: Each syllables must be footed.
- *FtBin*: feet must be one heavy syllable or two syllables of either weight.
- *WordFootLeft*: align feet with the left edge of the word.
- *WordFootRight*: align feet with the right edge of the word.

- *MainLeft*: align the head foot with the left edge of the word.
- *MainRight*: align the head foot with the right edge of the word.
- *AllFeetLeft*: align all feet with the left edge of the word.

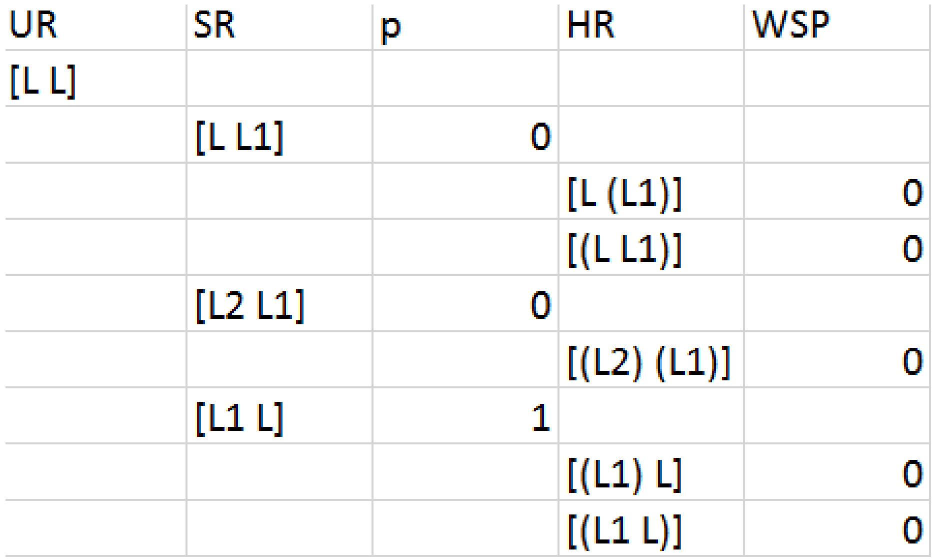
  AllFeetRight: align all feet with the right edge of the word.
- *NonFinal*: the final syllable in a word must not be footed.

Past Work

- Jarosz (2013, 2015) compared a number of models' performance on these languages, following the previous work by Boersma and Pater (2008).
- In these studies, a model was considered successful for a given language if it assigned the correct primary and secondary stress to every word in that language's data.
- The best three models were:
  - HG (Legendre et al. 1990), optimized using HG-GLA (Boersma 1997, Jesney 2007) with Expected Interpretive Parsing (EIP; Jarosz 2013), which succeeded an average of **93.95%** of the time.
  - Probabilistic Ranking Grammars (Jarosz 2015), optimized using Online Expectation Driven Learning (Jarosz 2015), which succeeded an average of **95.65%** of the time.
  - Probabilistic Ranking Grammars (Jarosz 2015), optimized using Batch Expectation Driven Learning (Jarosz 2015), which succeeded an average of **95.73%** of the time.
- To our knowledge, no one has proposed a model that performs better on this dataset.

The Structure of the Data

- In the Tesar and Smolensky (2000) dataset, each language consists of a lexicon of 62 words.
  - The underlying representation (UR) for each word is made up of sequences of /L/'s and /H/'s, which stand for light and heavy syllables.
  - The overt representations (SR) include stress information—for example, [L1 L2] would be a word with primary stress on the first syllable and secondary stress on the second.
  - Each overt representation is also associated with a probability (1 if the UR maps to the SR, 0 if it doesn't).
- For each possible overt form, there can be a number of different footings (HR).
  - All possible foot structures for each SR are provided in the data, along with their constraint violations.



UR	SR	p	HR	WSP
[L L]				
	[L L1]	0		
			[L (L1)]	0
			[(L L1)]	0
	[L2 L1]	0		
			[(L2) (L1)]	0
	[L1 L]	1		
			[(L1) L]	0
			[(L1 L)]	0

Results

- Unlike past work on this dataset, our model's learning is deterministic (i.e. it doesn't sample its outputs over the course of acquisition), so only one run from each language was necessary to gauge its performance.

- The model was run on each of the 124 languages until it stopped improving on the objective function (see the documentation at scipy.optimize.minimize for how this was determined) or reached 15,000 weight updates.
  - For an objective function, we used the KL-Divergence (Kullback and Leibler 1951) between the model's probability distribution over fully structured forms and probability of these in the training data (estimated using expectation maximization).

- A language was considered to be successfully learned if the final grammar assigned a conditional probability of more than 90% to each of the correct surface forms.
  - For our purposes, this was analogous to the criterion used by Jarosz (2013, 2015).

- The model met this criterion for **91.94%** (114/124) of the languages in the dataset.

- Moreover, the model did so very efficiently—it can run through all 124 languages on a relatively average laptop in around 2-2.5 hours.
  - Past approaches that have higher proportions of success take much longer to pass through the entire data set (Jarosz, p.c.) and must be run multiple times to get an accurate sample of their performance.

Applying the Model to Real Languages

- As an example of how our model can be used on real language data, we've begun applying it to the languages in the StressTyp2 database (Goedemans et al. 2014), using the constraints from Tesar and Smolensky (2000).

- While this work is still ongoing, our initial goal was to see if the constraints used by Tesar and Smolensky (2000) have the expressive power to represent the languages in StressTyp2.

- We've found that this isn't the case—when we add additional constraints, the model is able to successfully converge on more of the database's languages.

- Specifically, when we added in the constraints "MainStressWordLeft" and "MainStressWordRight", the model was able to learn languages in which the primary stress and secondary stress occur in different parts of a foot and opposite ends of a word.

Revisiting Coetzee and Pater (2011)

- We've also applied our model to a set of toy data representing the kind of variation+hidden structure analysis that Coetzee and Pater (2011) avoided.
- Our toy languages were similar to two dialects of English that demonstrate variable /t/-deletion processes that are sensitive to syllabic structure: African American Vernacular English and Chicano English.
  - (An example tableau from our pseudo-Chicano is show to the right.)
- Pseudo-AAVE deletes prevocalically with a probability of .29, preconsonantally .76, and phrase finally .73.
- Pseudo-Chicano deletes prevocalically with a probability of .45, preconsonantally .62, and phrase finally .37.

UR	Overt Form	Training Data Probability	Full Structure
Ct#V			
	CtV	0.55	
			Ct.V
			C.tV
	CV	0.45	
			C.V

Variation and Hidden Structure Results

- The model converged successfully (i.e. matched the training data probabilities) for both of these toy languages.
  - Pseudo-AAVE

Constraint:	*Comp-Coda	Max-Phr-Fin	Max	Align
Final Weight:	2.020898	0.158049	0.868204	0.110898

  - Pseudo-Chicano

Constraint:	*Comp-Coda	Max-Phr-Fin	Max	Align
Final Weight:	1.161199	1.021771	0.671661	1.16714

- The weight of Max-Phr-Fin is relatively higher in Pseudo-Chicano to represent its tendency for not deleting /t/’s phrase-finally.
- The relatively lower weight for Align in Pseudo-AAVE represents its tendency to not delete prevocalically.

Future Work

- While we plan to continue exploring the model's ability to learn stress, there are a number of problems in phonology that can be conceptualized as hidden structure (see Nazarov 2016 for more on this).
- We hope that others will find the model useful when analyzing these phenomena, and we've made the code public at https://github.com/blprickett/Hidden-Structure-MaxEnt .
- Feel free to contact us (bprickett@umass.edu) if you have any issues using it, or any questions that aren't answered in the software's README file.

References

Boersma, P. (1997). How we learn variation, optionality, and probability. *Proceedings of the Institute of Phonetic Sciences of the University of Amsterdam*, *21*, 43–58. Amsterdam.

Boersma, P., & Hayes, B. (2001). Empirical tests of the gradual learning algorithm. *Linguistic Inquiry*, *32*(1), 45–86.

Boersma, P., & Pater, J. (2008). *Convergence properties of a gradual learning algorithm for Harmonic Grammar*.

Byrd, R. H., Lu, P., Nocedal, J., & Zhu, C. (1995). A limited memory algorithm for bound constrained optimization. *SIAM Journal on Scientific Computing*, *16*(5), 1190–1208.

Coetzee, A. W., & Pater, J. (2011). The Place of Variation in Phonological Theory. In J. Goldsmith, J. Riggle, & A. Yu (Eds.), *The handbook of phonological theory* (pp. 401–431). Blackwell.

Dempster, A. P., Laird, N. M., & Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, *39*(1), 1–22.

Goldwater, S., & Johnson, M. (2003). Learning OT constraint rankings using a maximum entropy model. *Proceedings of the Stockholm Workshop on Variation within Optimality Theory*, *111120*.

Jarosz, G. (2013). Learning with hidden structure in optimality theory and harmonic grammar: Beyond robust interpretive parsing. *Phonology*, *30*(1), 27–71.

Jarosz, G. (2015). Expectation driven learning of phonology. *Ms., University of Massachusetts Amherst*.

Jesney, K. C. (2011). *Cumulative constraint interaction in phonological acquisition and typology*. University of Massachusetts Amherst.

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *ArXiv Preprint ArXiv:1412.6980*.

Kullback, S., & Leibler, R. A. (1951). On information and sufficiency. *The Annals of Mathematical Statistics*, *22*(1), 79–86.

Legendre, G., Miyata, Y., & Smolensky, P. (1990). Can Connectionism Contribute to Syntax? Harmonic Grammar, with an Application. Report CU-CS–. *Proceedings of the 26th Regional Meeting of the Chicago Linguistic Society*, 237–252.

Nazarov, A. (2016). *Extending Hidden Structure Learning: Features, Opacity, and Exceptions* (Dissertation, University of Massachusetts Amherst). Retrieved from https://scholarworks.umass.edu/dissertations_2/782

Pater, J. (2014). Categorical correctness in MaxEnt hidden structure learning. Retrieved from https://blogs.umass.edu/comphon/2014/09/24/success-maxent/

Pater, J., Jesney, K., Staubs, R., & Smith, B. (2012). Learning probabilities over underlying representations. *Proceedings of the Twelfth Meeting of the Special Interest Group on Computational Morphology and Phonology*, 62–71. Association for Computational Linguistics.

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

Tesar, B. (2004). Using inconsistency detection to overcome structural ambiguity. *Linguistic Inquiry*, *35*(2), 219–253.

Tesar, B., & Smolensky, P. (2000). *Learnability in optimality theory*. Mit Press.

Thank you!

We would like to thank the members of UMass's Sound Workshop, Robert Staubs, David Smith, Mark Johnson, and Gaja Jarosz for helpful discussion in various stages of this project.

Work on this project was supported by NSF grant BCS-424077 to the University of Massachusetts Amherst.