Stochastic Optimality Theory, local search, and Bayesian learning of hierarchical models

Ying Lin

Abstract

The Gradual Learning Algorithm (GLA) (Boersma and Hayes, 2001) can be seen as a stochastic local search method for learning Stochastic OT grammars. This paper tries to achieve the following goals: first, in response to the criticism in (Keller and Asudeh, 2002), we point out that the computational problem of learning stochastic grammars does have a general approximate solution (Lin, 2005). Second, we argue that the Bayesian framework on which the general solution is based connects the perspective of learning a probability distribution over grammars with local search strategies. Third, we also suggest that a general class of hierarchical probabilistic models may be suitable for marrying linguistic formalism with probability distribution.

Keywords:
Optimality Theory, Learning algorithm, Stochastic model, Local search, Bayesian methods.

1 Stochastic OT defines a probability distribution over grammars

A natural idea for building quantitative models of linguistic variation is to use the language of probability. Recent efforts in marrying linguistic formalism with probability distributions have resulted in the Stochastic Optimality Theory, independently proposed in (Boersma, 1997) and (Hayes and MacEachern, 1998). One key idea underlying stochastic OT and other similar proposals is a probability distribution over all possible grammars. Since each grammar may generate its own linguistic forms
with some probability, a distribution over a set of grammars is able to generate linguistic variation in a systematic manner. To distinguish individual grammars from a distribution over those grammars, we will use “OT grammars” for the former, and “stochastic OT models” for the latter.

An important issue that arises with the “distribution-over-the-grammars” approach is constraining such distributions. Suppose the universal grammar consists of $N$ constraints. There then would be an enormous number of distributions over the $N!$ permutations$^1$ of constraints, since each distribution is a way of dividing the probability mass into each of the $N!$ grammars. If the distributions over grammars are completely unconstrained, then the majority of these distributions will tend to be rather arbitrary and possibly of little interest to linguists. As a proposal to constrain the range of possible distributions, Stochastic OT characterizes a distribution over rankings with only a few parameters. The way such a distribution is determined by the parameters can be described as follows: first, constraints are represented by normal distributions with fixed variance and unknown means, thus giving them a continuous ranking scale. The mean values of those normal distributions, also called “ranking values”, are the parameters in Stochastic OT. Second, the normal distributions centered around the ranking values will determine the probability of any of the $N!$ grammars, thus inducing a far more constrained distribution over the space of possible grammars. In addition to the examples given in (Boersma and Hayes, 2001), Figure 1 illustrates a stochastic OT model with 3 constraints and the distribution it generates over the 6 possible rankings:

$^1$The permutations are also called “ranking” in literature.
Figure 1: Distributions of OT grammars generated by a Stochastic OT model. The ranking values are 0, 1, and 2 respectively, and the standard error = 1.

To see how stochastic OT constrains the range of such distributions, consider a distribution that assigns 0.5 to each of the ranking $C_3 > C_2 > C_1$ and $C_1 > C_2 > C_3$, and 0 to all the other rankings. No matter what ranking values are chosen for the constraints, this distribution does not correspond to any Stochastic OT model.

Unfortunately, no closed-form formula is known for calculating the probability of a certain ranking, yet such probabilities are necessary for calculating the predicted output frequencies of the grammar\(^2\). In (Boersma and Hayes, 2001), they suggest a simulation procedure that numerically computes the frequencies by repeating the following steps: first, a set of constraint values (called “selection points”) are generated independently from each normal distribution; these constraint values are then placed in an descending order to produce a ranking, which is used in the standard OT evaluation. For each input, the counts of output forms are then normalized after the simulation, which is regarded as relative frequency patterns predicted by the grammar.

\(^2\)A similar point is made with an example in (Lin, 2005).
From a computational perspective, Boersma and Hayes made the rational choice of inferring the output frequencies by conducting computer simulations, given the complex form of their distribution function. In a broader context for scientific computing, this strategy is known as an instance of Monte-Carlo methods (Liu, 2001), a strategy that can also be applied to learn parameters of the stochastic model. The close connection between Graduate Learning Algorithm, proposed by Boersma and Hayes, and the Monte-Carlo methods is a topic that the current paper intends to explore.

2 Stochastic OT grammars are learned from relational data

The learning problem of stochastic OT can be described as follows: how can the learner infer the ranking values of the constraints from frequencies of candidates? This problem is illustrated through the following example:

<table>
<thead>
<tr>
<th></th>
<th></th>
<th>$C_1$</th>
<th>$C_2$</th>
<th>$C_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$p(.)$</td>
<td>Ident(voice)</td>
<td>*VoiceObs</td>
<td>*VoiceObs(Coda)</td>
</tr>
<tr>
<td>/ad/</td>
<td>[at]</td>
<td>0.55</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>[ad]</td>
<td>0.45</td>
<td></td>
<td>*</td>
<td>*</td>
</tr>
</tbody>
</table>

Table 1: An example of a Stochastic OT learning problem

Since the constraints have continuous ranking scales, the principles of OT implies that the above data can be translated to statements like follows:

$$\max\{C_1, C_2\} > C_3; \text{ with probability 0.55}$$

$$\max\{C_1, C_2\} < C_3; \text{ with probability 0.45}$$

(1)
Here the maximum of $C_1$ and $C_2$ corresponds to the assumption that if either $C_1$ or $C_2$ dominates $C_3$, then, the candidate [ad] would be less optimal than [at]. Thus, the learning problem of stochastic OT can be simply stated as: given data that encode the stochastic relationships between constraints, such as in (1), how to find the desired parameters that constrain the distribution over the rankings? It should be noted that such a problem is quite distinct from the problem of learning “probabilistic grammars” in the sense used in computational linguistics: first, (1) represents a type of relational data, the input-output pairs encode ordering relations between constraints. Moreover, the goal is not learning a distribution over strings, but a distribution over grammars. Hence, learning continuous parameters from relational data presents the main challenge for applying Stochastic OT to linguistic data analysis.

3 Gradual Learning Algorithm and stochastic local search

The Gradual Learning Algorithm is a proposal for learning stochastic OT as well as standard OT grammars. The following algorithm summarizes what was described in (Boersma and Hayes, 2001):
Require: : total number of iterations $T$; elasticity $\Delta_1, \cdots, \Delta_T$

1: $t \leftarrow 1$

2: repeat

3: Pick an input-output pair $(x, y)$ randomly according to the frequency in the data

4: Randomly generate a ranking $R$ from the current hypothesis, for the given input $x$, let $z \leftarrow$ the output selected by $R$

5: if $z \neq y$ then

6: add $\Delta_t$ to all constraints that prefer $z$ over $y$

7: subtract $\Delta_t$ from all constraints that prefer $y$ over $z$

8: end if

9: $t \leftarrow t + 1$

10: until $t > T$

Why does GLA work? In order to address this question, it is useful to distinguish two separate problems to which GLA can be applied. The first problem is learning a standard OT grammar from noisy data: is there a procedure that will find a single constraint ranking despite the exceptions in the data? Although the problem can be solved efficiently in the absence of noise (Tesar and Smolensky, 1996), it can be shown that noise makes the problem much harder, if a systematic search is used to look for the exact grammar with the fewest exceptions\(^3\). For the noisy ranking problem, GLA can be seen as a combination of two proposals:

1. The first proposal is transforming the original discrete problem to a continuous one. Instead of searching among the $N!$ rankings, GLA searches within the

\(^3\)The specific claim can be stated as follows: if one allows exceptions to arise from arbitrary re-ranking of constraints, then the constraint ranking problem can be shown to be intractable (Lin, 2002). But the problem can be solved efficiently if one only seeks approximate solutions.
larger, continuous space of stochastic OT models\textsuperscript{4}. An approximate solution is thus pursued within the continuous space, and the result can be mapped into the original space by ordering the ranking values of a stochastic OT model. This strategy, also called relaxation, is often adopted in numerical approximations of optimization problems and provides a view of stochastic OT as a relaxed version of standard OT.

2. The second proposal is a local search strategy with a stochastic component. In order to minimize the error incurred by the grammar, in each step the GLA searches for the next hypothesis within a local neighborhood of the current hypothesis. Compared to a deterministic local search, which faces local maxima problems, the random factor in a stochastic local search allows the algorithm to move “downwards” to a less optimal solution, thereby helping to escape local maxima (Hoos and Stützle, 2005). In GLA, the randomness arises from the selection of the input-output pair and the generation of rankings, and the standard error of ranking values can serve as a tuning parameter that controls the “temperature” of a stochastic search.

Notice that these two proposals are somewhat independent, and that this allows for the development of other optimization techniques that are distinct from GLA\textsuperscript{5}. However, our main point here is distinguishing the problem that GLA tries to solve, and the search strategy of the learning algorithm. For the first problem of learning discrete OT grammar from noisy data, GLA can be seen as a local search algorithm that finds a solution in the relaxed, continuous hypothesis space.

The second problem that GLA can be applied to is inferring the parameters in

\textsuperscript{4}This space can be formally represented as $R^{N-1}$.

\textsuperscript{5}For example, we could introduce stochastic searches in the original space of $N!$ rankings, by using techniques similar to Simulated Annealing. Yet this option is not explored here.
stochastic OT models from frequency data, a problem that is distinct from the one
described above. As discussed in Section 2, although the search space – all possible
distributions over the $N!$ rankings – is infinite, these distributions are controlled by
$N - 1$ parameters$^6$. As an attempt to learn those parameters, GLA is justified as a
way to perform “frequency matching” (Boersma and Hayes, 2001): the local search
strategy adjusts the parameters gradually so that they generate frequency patterns
similar to those of the learning data. As noted in previous work (Eisner, 2000),
this claim is closely related to the well-known maximum likelihood criterion of model
fitting. Under this criterion, the distribution that makes the observed frequencies
most likely should be chosen as the best hypothesis learned from the data. However,
maximum likelihood learning requires explicit probability calculations. Partly due to
the lack of explicit formulae for doing such calculations (also discussed in Section 1),
this connection has not been explored further in literature.

In sum, the above discussion argues for GLA as a sensible optimization algorithm.
Nevertheless, it does not take the place of a formal analysis, which is crucial for
establishing the desirable theoretical guarantees for a learning algorithm. For either
of the problems listed above, no result is known regarding the quality of the answer
that GLA finds, or under what conditions this algorithm will converge. This problem,
noticed by previous authors, is the most serious argument against GLA (Keller and
Asudeh, 2002), and leaves the learning of stochastic OT models an open problem.

4 First try: a “random walk” version of GLA

Part of the difficulty in analyzing the behavior of GLA arises from a changing “elas-
ticity schedule”: $\Delta_t$ is set to be decreasing with time. According to (Boersma and

$^6$Because keeping the distance between ranking values would not change the behavior of the
stochastic OT model, one parameter can be removed from the $N$ constraints.
Hayes, 2001), the reason for setting such a schedule is that for large values of \( \Delta_t \), the algorithm tends to move “quickly” towards the right answer; yet smaller \( \Delta_t \) does a better job in matching the frequencies before the algorithm is forced to stop.

Leaving aside the choice of appropriate plasticity values, we study the following simplification: what if we set plasticity to a _fixed_ value, say \( \Delta \)? Formally, this is equivalent to discretizing the search space into a \( N \)-dimensional grid, with spacing \( \Delta \) between adjacent points. At any given time, the GLA looks for the next hypothesis among a subset of the vertices of the \( N \)-dimensional cube around the current hypothesis\(^7\). The probability of moving in each direction is jointly determined by three factors: the learning data, the current and the new hypothesis. With this modification, the learner will perform a “random walk”, and its behavior is characterized by the transition probability between neighboring points on the grid. As an example, consider a grammar of two constraints and a data set with only 2 competing candidates:

<table>
<thead>
<tr>
<th>/ad/</th>
<th>[at]</th>
<th>0.7</th>
<th>*</th>
</tr>
</thead>
<tbody>
<tr>
<td>[ad]</td>
<td>0.3</td>
<td></td>
<td>*</td>
</tr>
</tbody>
</table>

Table 2: A simple stochastic OT grammar.

The search space of GLA for the above grammar and data set is illustrated in Figure 2. The dark point corresponds to the current hypothesis, while the grey points correspond to possible moves for the next hypothesis. Since the 2-constraint problem is characterized by 1 parameter, the search space is constrained to be the points lying on the same diagonal line, which extends infinitely towards above and below:

\(^7\)When the constraint interactions are simple, then the number of possible moves is even smaller.
The random walk analogy brings up a conceptual problem: when does a learner stop changing her mind? Although the learner moves randomly at each step, there is a kind of invariance among all the moves by the learner, if we consider what happens in the long run: under fairly general conditions, a random walk converges to a unique stationary distribution, regardless of starting point\(^8\). In other words, if we collect the hypotheses of the learner over a long period of time, they form a distribution in the hypothesis space that does not change over time. To illustrate this idea, we run the “random walk-GLA” on Table 2 for a large number of iterations, and the aggregation of the outputs are shown in Figure 3.

---

\(^8\)This is a standard result in Markov chain theory. For a linguistically relevant discussion of Markov chains and language acquisition, see (Berwick and Niyogi, 1996).
Curiously, if we extract the modes of the distribution shown in Figure 3, it actually fits the frequencies rather well\(^9\). This simple example illustrates a “non-deterministic” view of learning algorithms like our modification of GLA: instead of expecting the learner to converge to a unique hypothesis, it is also possible to allow the learner to converge in distribution\(^10\). This view is consistent with proposals of variational learning (Yang, 2000), as well as the observation that learners can make different generalizations from different subsets of the data (Gerken, 2006).

Unfortunately, the above analysis of “random walk-GLA” is not general enough to handle problems at the scale of real linguistic analyses. For any grammar with more than 3 constraints, explicitly calculating the transition probabilities is tedious, and worse still, the probabilities depend on the constraint interactions reflected in each data set. Clearly, if one needs to make an argument about the learner in general (e.g. whether it converges), then such argument should not depend on the specific data. In addition, whereas the perspective of convergence in distribution may be appealing, there is no clear interpretation of what such a distribution actually means. These problems point to the need for a general framework in which stochastic simulation and its results can be understood. The next section introduces such a framework.

\(^9\)To complete this argument, one may explicitly list all the states shown in Figure 2, and calculate the transition probabilities between adjacent states. This matrix of transition probability can be used to calculate the stationary distribution, but this calculation is not attempted here.

\(^10\)Notice we have used the word “distribution” in two contexts. In the first context, a Stochastic OT model considers a distribution over all possible rankings. This distribution is parameterized as a vector of real numbers. In the current context, we are referring to a distribution over the parameters of a stochastic OT model. It is the second distribution to which the learner converges.
5 Second try: the Bayesian perspective and the hierarchical model

The Bayesian approach to learning Stochastic OT addresses two key questions raised above:

1. What is a sensible choice for the stationary distribution over grammars?

2. How do we design a stochastic search strategy that will eventually converge to such a distribution?

The following notations will be used hereafter: $G$ stands for parameters for Stochastic OT, $D$ for a set of relational data as illustrated in Section 2, and $Y$ stands for the selection points that generate the ranking. Upper letters stand for random variables, and lower case letters will be used to represent instances from the corresponding distributions. Square brackets are used to indicate a distribution for which the first symbol is the random variable. For example, the expression $x \sim [X|Y = y]$ can be read as: “$x$ is a sample from the conditional distribution of $X$ when $Y$ is fixed to $y$”.

The goal of Bayesian learning can be stated as inferring the posterior distribution $[G|D]$ over the hypothesis space, from a prior distribution\(^\text{11}\) $[G]$ over the same space and a set of data $\{d_1, \ldots, d_n\}$. The posterior distribution represents the learner’s uncertainty about the underlying hypothesis after seeing evidence (frequencies contained in $\{d_1, \ldots, d_n\}$) in her language, and contains rather rich information. For example, if the posterior distribution is concentrated around one hypothesis, its mode can be

\(^{11}\)At present, we do not discuss the significance of the prior distribution, which has been set to a vague distribution that does prefer any hypothesis. A discussion of the prior distribution is included in Section 7.
extracted to represent such a distribution. A possible confusion may arise with regard to the word “distribution” here, since each hypothesis – the value of $G$ itself – stands for a set of parameters that control another distribution over OT grammars. To clarify, we note that Bayesian learning tries to quantify the uncertainty within the hypothesis space, rather than to identify a single hypothesis. In other words, the objective of the Bayesian approach can be seen as learning a “(posterior) distribution of (parametric) distributions”.

In order to obtain the posterior distribution, Bayesian researchers often rely on computational simulations, especially for problems where many parameters need to be learned from data. Many of those procedures are rather similar to the one sketched in Section 4. Typically, a Markov chain is designed such that it eventually converges to the posterior distribution, and an algorithm following this chain is used to sequentially search through the entire hypothesis space. When the algorithm has run for a sufficiently long time, the congregation of hypotheses explored by the algorithm provide a sample of the posterior distribution. Various properties of the posterior can thus be inferred from this sample.

An implementation of the Bayesian learning method sketched above is presented in detail in (Lin, 2005). This algorithm is also an instance of a Monte-Carlo method and can be seen as the learning counterpart of the generation scheme proposed by Boersma and Hayes: instead of running a “forward” simulation, we run a “backward” simulation to sample from the posterior distribution. To complement the technical presentation in (Lin, 2005), the main idea is illustrated graphically in Figure 4:

---

12 We note, however, that the mode does not have a special status in Bayesian statistics. Here it merely serves as a convenient way of checking the result of learning.

13 Since this method uses a sequence of search steps in Monte-Carlo simulation, it is named Markov chain Monte-Carlo.
Figure 4: Graphical illustration of the generation and learning of stochastic OT. Top: generating data from a Stochastic OT grammar; Bottom: learning stochastic OT grammar from data.

The filled circles represent observed variables of the model, and the unfilled ones represent the hidden variables. Represented by solid arcs, Boersma and Hayes’ scheme is illustrated in the top panel: first a set of selection points are generated from the known ranking values $G = g$, then these points are ordered to generate rankings that determine the frequencies in the data. In statistical terms, these two steps are equivalent to first drawing a sample from the conditional distribution $y \sim [Y|G = g]$, followed by drawing the input-output pairs from $d \sim [D|G = g, Y = y]$. After many rounds of simulation, the obtained selection points and frequencies form a sample of the joint distribution $[Y, D|G = g]$, and the frequencies themselves form a sample of the marginal distribution $[D|G = g]$. The solid arc from a higher-level hidden variable $G$ to $Y$ characterizes the specific parametrization used in the stochastic OT model. This type of architecture, with higher-level hidden variables controlling the lower-level/observed variables, is recognized as hierarchical modeling in statistics literature.

Starting from only the observed data $\{d_1, \ldots, d_n\}$, the learning procedure is done in two iterative steps, illustrated by dotted arcs: first we assume the ranking values
are known, say $g^{(0)}$, and use the observed data and this initial hypothesis together to search for a set of selection points. This is equivalent to sampling from a *conditional* distribution $y^{(1)} \sim [Y | G = g^{(0)}, D = d]$. In other words, we are taking the parameter $G$ as known, and trying to generate a set of grammars from it that are consistent with the data. By letting the learning data $d$ vary according to its relative frequency in the corpus $\{d_1, \ldots, d_n\}$, the generation of the selection points will also depend on the variation contained in the corpus. As the next step, we fix the set of selection points, and search for the new hypothesis from another conditional distribution $g^{(1)} \sim [G | Y = y^{(1)}]^{14}$. In effect, this updates the parameters by summarizing the “attested” grammars obtained from the previous step. Although the initial hypothesis $g^{(0)}$ may be a poor fit to the data, these two search steps are iterated to produce a sequence of \((\text{ranking values, selection points})\) pairs that form a Markov chain themselves: \((g^{(1)}, y^{(1)}), (g^{(2)}, y^{(2)}), \ldots, (g^{(n)}, y^{(n)})\). As the iteration $n$ tends to infinity, this Markov chain converges to the joint posterior distribution $[G, Y | D]$. As a consequence, if we only consider the sequence of ranking values $g^{(1)}, g^{(2)}, \ldots, g^{(n)}$, then they converge to the target of Bayesian learning – the posterior distribution of the ranking values given the data $[G | D]$.

In comparison to *ad hoc* stochastic local search methods, whose problems have been mentioned earlier, Bayesian stochastic search has several advantages. For properly constrained problems with *proper* posterior distributions$^{15}$, Bayesian simulation is guaranteed to converge, under mild conditions that are almost always satisfied in common problems (Tierney, 1994). Although each individual hypothesis explored by the learner does not have much significance, the collection of all the hypotheses can be interpreted as a sample of the posterior distribution after the learner has seen

---

$^{14}$This distribution is the same as $[G | Y = y^{(1)}, D]$.

$^{15}$The posterior is called “improper” if the probability mass does not sum up to 1.
the data. Hence the Bayesian approach addresses both of the the problems of the “random walk-GLA” approach discussed in Section 4, and provides a sound general solution to the stochastic OT learning problem.

In addition to the computational advantage, the Bayesian framework also provides a unified perspective on two separate ideas in the algorithmic approach to language learning. The first idea, discussed in Section 1, introduces probability distributions over a finite set of grammars as the learner’s hypothesis space. Since these distributions are constrained to be parametric (for example, the parameters are the ranking values in stochastic OT), the learning problem is formalized as inferring the parameter values from the observed data. The main challenge of this view is that the distribution over grammars is not observed directly. Rather, a hierarchical model (such as that of Figure 1) is often needed to relate the linguistic data and to the distribution. Because of the use of hidden variables (such as the selection points in Stochastic OT), the Bayesian framework is well-suited for hierarchical models, where maximum likelihood (“frequency matching”) methods often fail. Another example of hierarchical models also appeared in (Yang, 2000)’s variational learning framework, which also discusses parameterized distributions over grammars. We note in passing that although Yang presents his learning model in the context of natural selection, in one version of his model, the parameters can also be learned through a Bayesian approach that is very similar to the one presented above (see Appendix).

The second idea – local search – is also the heart of many learning algorithms. For example, trigger-based learning (Gibson and Wexler, 1994) is a deterministic local search-based method. Stochastic variants of TLA and their formal analysis are based on Markov chain theory (Berwick and Niyogi, 1996). Compared to the parameter-setting framework, where local search is conducted in a discrete hypothesis space by flipping the value of binary parameters, the learning strategy discussed in this section
is also an instance of local search, but in a continuous hypothesis space. Just like the stochastic variants of TLA, the Bayesian local search also eventually settles on a distribution over the hypothesis space. The main difference is that Bayesian local search makes use of an additional space – the space of selection points. By alternating between updating the hypothesis and the selection points in their respective spaces, the Bayesian local search reaches the posterior distribution in the limit.

6 Experiment: Spanish diminutive suffixation

This section examines a diagnostic example of (Lin, 2005) in more detail. The data set intends to capture certain aspects of Spanish diminutive formation. The actual constraints used in the analysis are not important here, since we are focusing on the formal aspect of the learning problem. For comparison\textsuperscript{16}, we apply both the GLA and Bayesian method to a data set of Spanish diminutives, based on the analysis proposed in (Arbisi-Kelm, 2002). There are 3 base forms, each associated with 2 diminutive suffixes. The model consists of 4 constraints: ALIGN(TE,Word,R), MAX-OO(V), DEP-IO and BaseTooLittle. The data presents the problem of learning from noise, since no Stochastic OT model can provide an exact fit to the data: the candidate [ubita] violates an extra constraint compared to [liri.ito], and [ubasita] violates the same constraint as [lirjosito]. Yet unlike [lirjosito], [ubasita] is not observed in the data.

\textsuperscript{16}Thanks to Bruce Hayes for suggesting this example.
In the results found by GLA, \([\text{marsito}]\) always has a lower frequency than \([\text{marEsito}]\) (See Table 4). This is not accidental. Instead, it reveals that the GLA’s local search strategy can be problematic: since the constraint \(B\) is violated by \([\text{ubita}]\), it is always demoted whenever the underlying form /uba/ is encountered during learning. Therefore, even though the expected model assigns equal values to \(\mu_3\) and \(\mu_4\) (corresponding to \(D\) and \(B\), respectively), \(\mu_3\) is always less than \(\mu_4\), simply because there is more chance of penalizing \(D\) rather than \(B\). This problem arises precisely because of the local search heuristic (i.e., demoting the constraint that prefers the wrong candidate) that GLA uses to find the target grammar.

The Bayesian method, on the other hand, suffers from no such problems because it does not rely on heuristics for the local search. Starting from an “uninformative” prior as described above, the posterior distribution found by the learner is shown in Figure 5. Simulated frequencies using the parameters found by the Bayesian learner, as compared to those found by GLA after two runs\(^{17}\), are reported in Table 4.

---

\(^{17}\) The two runs here both use 0.002 and 0.0001 as the final plasticity. The initial plasticity and the iterations are set to 2 and 1.0e7. Slightly better fits can be found by tuning these parameters, but the observation remains the same.
Figure 5: Simulated posterior distribution of the three ranking values on the Spanish diminutive dataset.

<table>
<thead>
<tr>
<th>Input</th>
<th>Output</th>
<th>Obs</th>
<th>Bayesian</th>
<th>GLA₁</th>
<th>GLA₂</th>
</tr>
</thead>
<tbody>
<tr>
<td>/uba/</td>
<td>[ubita]</td>
<td>100</td>
<td>95%</td>
<td>96%</td>
<td>96%</td>
</tr>
<tr>
<td></td>
<td>[ubasita]</td>
<td>0%</td>
<td>5%</td>
<td>4%</td>
<td>4%</td>
</tr>
<tr>
<td>/mar/</td>
<td>[maresito]</td>
<td>50%</td>
<td>50%</td>
<td>38%</td>
<td>45%</td>
</tr>
<tr>
<td></td>
<td>[marsito]</td>
<td>50%</td>
<td>50%</td>
<td>62%</td>
<td>55%</td>
</tr>
<tr>
<td>/lirjo/</td>
<td>[liri.ito]</td>
<td>90%</td>
<td>95%</td>
<td>96%</td>
<td>91.4%</td>
</tr>
<tr>
<td></td>
<td>[lirjosito]</td>
<td>10%</td>
<td>5%</td>
<td>4%</td>
<td>8.6%</td>
</tr>
</tbody>
</table>

Table 4: Comparison of Bayesian method and GLA. The parameters used in the Bayesian simulation are set to modes of the posterior distribution, while the GLA simulation uses the value returned by the algorithm.

7 Summary and remaining issues

Stochastic local search provides a useful perspective for viewing various approaches to the learning problem of Stochastic OT: while GLA can be seen as a kind of stochastic local search based on heuristics, the Bayesian method is another search strategy
that converges to the posterior distribution. The key to the effectiveness of these methods is computing power: computing resources make it possible to explore the entire space of grammars and discover where good hypotheses are likely to occur. Given unlimited run time, the Bayesian simulation will lead to the exact posterior distribution. In reality, the solution is always approximate\textsuperscript{18}, given the limited run time for each program.

In addition, it is worth pointing out that our work did not exploit the full potential of the Bayesian paradigm. Although Bayesian learning has been connected to stochastic local searches, we did not explore the possibility of a “theoretically informed” search through the use of prior distributions, i.e. learner’s uncertainty about the underlying hypothesis before seeing any linguistic evidence. In the current model, the role of prior distribution has been de-emphasized, since flat, uninformative priors have been used to obtain posterior distributions. However, proposals on learning bias in the OT literature (Prince and Tesar, 1999; Hayes, 2004) suggest that an integration of initial bias and prior distributions is quite promising. We also note that a number of other proposals in the OT literature can also be formulated in the Bayesian framework. For example, by using a kind of mixture/multi-modal prior distribution, the formal proposal in (Anttila, 1997) may also be approximated in the Bayesian framework. Potentially, the proposal of “floating constraints” (Nagy and Reynolds, 1997) can also be translated to the Bayesian framework, if the fixed variance (spread) of the normal distribution in the Stochastic OT model is replaced with an unknown variance\textsuperscript{19}. Hence, the Bayesian approach not only solves the computational problem of learning Stochastic OT, but also opens up connections to other OT variants.

\textsuperscript{18}However, there is a large body of work that investigates the convergence rate of these strategies and ways to speed up convergence (Gilks, Richardson, and Spiegelhalter, 1996; Liu, 2001).

\textsuperscript{19}Such unknown parameters can also be learned from the data using the same procedure outlined above.
From a more general perspective, Stochastic OT is an instance of a hierarchical model. As discussed in Section 5, hierarchical modeling introduces linguistic variation by adding a stochastic level of grammar generation on top of a deterministic generative grammar. To some extent, the learner is still strongly biased because the learner is able to tell whether a form is consistent with a grammar. Since the assumption of strong initial bias has also been used in previous work on the parameter setting problem as well as OT learnability, our work does not signify a radical departure from the generative tradition, but is a way of approaching linguistic variation from the perspective of multiple competing grammars that are governed by a distribution.

The assumption of strong initial bias also distinguishes the current work from popular probabilistic models used in natural language processing. Compared to the latter, the hierarchical approach considers probability distribution at a very different level: instead of assigning probabilities to forms (e.g. words or sentences), the hierarchical approach assigns probabilities to a set of grammars. These probabilities are not observed directly through the data, and learning such a distribution is generally non-trivial because each datum may be consistent with several different grammars at the same time. For the special case of stochastic OT distributions, our result shows that local search algorithms within a Bayesian framework solves its learning problem. More empirical tests are required before we can evaluate the utility of the Bayesian framework in analyzing systematic linguistic variation.
Appendix: A Bayesian approach to Yang’s principle-and-parameter model

Similar to Figure 1, a graphical representation of Yang’s parametric model is given in Figure 6:

![Diagram of Yang's parametric model]

Here $Z = (\alpha_1, \cdots, \alpha_n)$, $\alpha_i \in \{0, 1\}$ corresponds to the parameter vector that determines a grammar. $D$ represents a set of sentences. For a fixed grammar, each sentence is either analyzable or unanalyzable by the grammar. The hidden variable $P = (p_1, \cdots, p_n)$ controls the probability of each $z_i$ taking the value of 0 or 1. Yang describes the generation from this model as two steps: first, a binary parameter vector is generated from the binomial distribution, then this grammar is used to analyze a randomly selected sentence. Using a very similar procedure as described in Section 5, the Bayesian learning algorithm for Yang’s model can be given as follows:

- Let $t \leftarrow 0$ and set the initial value of $p^{(0)}$.

- Iterate until convergence:

  - Draw a set of grammars $\{z_1^{(t)}, \cdots, z_M^{(t)}\}$ as follows: first select a sentence $s$, then use the binomial probability $p^{(t)}$ to draw a binary vector (grammar) that can analyze $s$. A grammar is discarded if it is not consistent with $s$, and the sampling continues until the required number of grammar is reached.
– Draw each dimension of the binomial probability $p_i^{(t+1)}$ from a Beta distribution, and let $p^{(t+1)} = (p_1^{(t+1)}, \ldots, p_n^{(t+1)})$.

– Let $t \leftarrow t + 1$.

The first step draws $M$ grammars from one conditional distribution $[Z|P, D]$, while the second step samples from the other $[P|Z, D] = [P|Z]$. The Beta distribution follows from the following derivation: using the conjugate prior for the binomial probability that controls each parameter: $[p_i] \sim Beta(\mu, \nu)$, applying the Bayes formula $[p_i|\alpha_i] \propto [\alpha_i|p_i] \cdot [p_i]$, we have $[p_i|\alpha_i] \sim Beta(\mu + N_i, \nu + M - N_i)$, where $N_i$ represents the number of grammars that have the digit $\alpha_i = 0$. The free parameters $\mu, \nu \geq 0$ can be set to a variety of values. For example, to exclude any a priori preference for the value of $p_i$ on the interval $[0, 1]$, we can set $\mu = \nu = 1$ and the Beta distribution becomes a uniform distribution over $[0, 1]$. When the parameters do not interact, parameter setting can be done in a straightforward way, i.e. each $p_i$ is simply the proportion of sentences that have parameter $z_i = 1$. However, the parameter setting problem becomes non-trivial if the parameters interact, and the Bayesian approach would offer an advantage in terms of computation, just as in the case of constraint interaction in Optimality Theory.

References


**Corresponding address:**

1100 E. University Blvd, 200E, Department of Linguistics, University of Arizona, Tucson, AZ 85721. yinglin@email.arizona.edu