Learning Stochastic OT Grammars: A Bayesian approach using Data Augmentation and Gibbs Sampling

Ying Lin[*]
Department of Linguistics
University of California, Los Angeles
Los Angeles, CA 90095
yinglin@ucla.edu

Abstract

Stochastic Optimality Theory (Boersma, 1997) is a widely-used model in linguistics that did not have a theoretically sound learning method previously. In this paper, a Markov chain Monte-Carlo method is proposed for learning Stochastic OT Grammars. Following a Bayesian framework, the goal is finding the posterior distribution of the grammar given the relative frequencies of input-output pairs. The Data Augmentation algorithm allows one to simulate a joint posterior distribution by iterating two conditional sampling steps. This Gibbs sampler constructs a Markov chain that converges to the joint distribution, and the target posterior can be derived as its marginal distribution.

1 Introduction

Optimality Theory (Prince and Smolensky, 1993) is a linguistic theory that dominates the field of phonology, and some areas of morphology and syntax. The standard version of OT contains the following assumptions:

- A grammar is a set of ordered constraints ($\{C_i : i = 1, \cdots, N\}, >$);
- Each constraint C_i is a function: $\Sigma^* \rightarrow \{0, 1, \cdots\}$, where Σ^* is the set of strings in the language;
- Each underlying form u corresponds to a set of candidates $GEN(u)$. To obtain the unique surface form, the candidate set is successively filtered according to the order of constraints, so that only the most harmonic candidates remain after each filtering. If only 1 candidate is left in the candidate set, it is chosen as the optimal output.

The popularity of OT is partly due to learning algorithms that induce constraint ranking from data. However, most of such algorithms cannot be applied to noisy learning data. Stochastic Optimality Theory (Boersma, 1997) is a variant of Optimality Theory that tries to quantitatively predict linguistic variation. As a popular model among linguists that are more engaged with empirical data than with formalisms, Stochastic OT has been used in a large body of linguistics literature.

In Stochastic OT, constraints are regarded as independent normal distributions with unknown means and fixed variance. As a result, the stochastic constraint hierarchy generates systematic linguistic variation. For example, consider a grammar with 3 constraints, $C_1 \sim N(\mu_1, \sigma^2)$, $C_2 \sim N(\mu_2, \sigma^2)$, $C_3 \sim N(\mu_3, \sigma^2)$, and 2 competing candidates for a given input x:

	$p(.)$	C_1	C_2	C_3
$x \sim y_1$	.77	0	0	1
$x \sim y_2$	.23	1	1	0

Table 1: A Stochastic OT grammar with 1 input and 2 outputs

[*] The author thanks Bruce Hayes, Ed Stabler, Yingnian Wu, Colin Wilson, and anonymous reviewers for their comments.

Proceedings of the 43rd Annual Meeting of the ACL, pages 346–353,
Ann Arbor, June 2005.

The probabilities $p(.)$ are obtained by repeatedly sampling the 3 normal distributions, generating the winning candidate according to the ordering of constraints, and counting the relative frequencies in the outcome. As a result, the grammar will assign non-zero probabilities to a given set of outputs, as shown above.

The learning problem of Stochastic OT involves fitting a grammar $G \in R^N$ to a set of candidates with frequency counts in a corpus. For example, if the learning data is the above table, we need to find an estimate of $G = (\mu_1, \mu_2, \mu_3)^1$ so that the following ordering relations hold with certain probabilities:

$$\begin{array}{l} \max\{C_1, C_2\} > C_3; \textit{ with probability } .77 \\ \max\{C_1, C_2\} < C_3; \textit{ with probability } .23 \end{array} \tag{1}$$

The current method for fitting Stochastic OT models, used by many linguists, is the Gradual Learning Algorithm (GLA) (Boersma and Hayes, 2001). GLA looks for the correct ranking values by using the following heuristic, which resembles gradient descent. First, an input-output pair is sampled from the data; second, an ordering of the constraints is sampled from the grammar and used to generate an output; and finally, the means of the constraints are updated so as to minimize the error. The updating is done by adding or subtracting a "plasticity" value that goes to zero over time. The intuition behind GLA is that it does "frequency matching", i.e. looking for a better match between the output frequencies of the grammar and those in the data.

As it turns out, GLA does not work in all cases[2], and its lack of formal foundations has been questioned by a number of researchers (Keller and Asudeh, 2002; Goldwater and Johnson, 2003). However, considering the broad range of linguistic data that has been analyzed with Stochastic OT, it seems unadvisable to reject this model because of the absence of theoretically sound learning methods. Rather, a general solution is needed to evaluate Stochastic OT as a model for linguistic variation. In this paper, I introduce an algorithm for learning Stochastic OT grammars using Markov chain Monte-Carlo methods. Within a Bayesian framework, the learning problem is formalized as finding the *posterior distribution* of ranking values (G) given the information on constraint interaction based on input-output pairs (D). The posterior contains all the information needed for linguists' use: for example, if there is a grammar that will generate the exact frequencies as in the data, such a grammar will appear as a mode of the posterior.

In computation, the posterior distribution is simulated with MCMC methods because the likelihood function has a complex form, thus making a maximum-likelihood approach hard to perform. Such problems are avoided by using the *Data Augmentation* algorithm (Tanner and Wong, 1987) to make computation feasible: to simulate the posterior distribution $G \sim p(G|D)$, we augment the parameter space and simulate a joint distribution $(G, Y) \sim p(G, Y|D)$. It turns out that by setting Y as the value of constraints that observe the desired ordering, simulating from $p(G, Y|D)$ can be achieved with a *Gibbs sampler*, which constructs a Markov chain that converges to the joint posterior distribution (Geman and Geman, 1984; Gelfand and Smith, 1990). I will also discuss some issues related to efficiency in implementation.

2 The difficulty of a maximum-likelihood approach

Naturally, one may consider "frequency matching" as estimating the grammar based on the maximum-likelihood criterion. Given a set of constraints and candidates, the data may be compiled in the form of (1), on which the likelihood calculation is based. As an example, given the grammar and data set in Table 1, the likelihood of d="$\max\{C1, C2\} > C3$" can be written as $P(d|\mu_1, \mu_2, \mu_3)$=

$$1 - \int_{-\infty}^{0} \int_{-\infty}^{0} \frac{1}{2\pi\sigma^2} \exp\left\{ -\frac{\vec{f}_{xy} \cdot \Sigma \cdot \vec{f}_{xy}^{T}}{2} \right\} dx\, dy$$

where $\vec{f}_{xy} = (x - \mu_1 + \mu_3, y - \mu_2 + \mu_3)$, and Σ is the identity covariance matrix. The integral sign follows from the fact that both $C_1 - C_2$, $C_2 - C_3$ are normal, since each constraint is independently normally distributed.

If we treat each data as independently generated by the grammar, then the likelihood will be a product of such integrals (multiple integrals if many constraints are interacting). One may attempt to maximize such a likelihood function using numerical

[1] Up to translation by an additive constant.

[2] Two examples included in the experiment section. See 6.3.

methods[3], yet it appears to be desirable to avoid likelihood calculations altogether.

3 The missing data scheme for learning Stochastic OT grammars

The Bayesian approach tries to explore $p(G|D)$, the posterior distribution. Notice if we take the usual approach by using the relationship $p(G|D) \propto p(D|G) \cdot p(G)$, we will encounter the same problem as in Section 2. Therefore we need a feasible way of sampling $p(G|D)$ without having to derive the closed-form of $p(D|G)$.

The key idea here is the so-called "missing data" scheme in Bayesian statistics: in a complex model-fitting problem, the computation can sometimes be greatly simplified if we treat part of the unknown parameters as data and fit the model in successive stages. To apply this idea, one needs to observe that Stochastic OT grammars are learned from *ordinal data*, as seen in (1). In other words, only one aspect of the structure generated by those normal distributions — the ordering of constraints — is used to generate outputs.

This observation points to the possibility of treating the sample values of constraints $\vec{y} = (y_1, y_2, \cdots, y_N)$ that satisfy the ordering relations as missing data. It is appropriate to refer to them as "missing" because a language learner obviously cannot observe real numbers from the constraints, which are postulated by linguistic theory. When the observed data are augmented with missing data and become a *complete data* model, computation becomes significantly simpler. This type of idea is officially known as *Data Augmentation* (Tanner and Wong, 1987). More specifically, we also make the following intuitive observations:

- The complete data model consists of 3 random variables: the observed ordering relations D, the grammar G, and the missing samples of constraint values Y that generate the ordering D.
- G and Y are interdependent:
  - For each fixed d, values of Y that respect d can be obtained easily once G is given: we just sample from $p(Y|G)$ and only keep those that observe d. Then we let d vary with its frequency in the data, and obtain a sample of $p(Y|G, D)$;
  - Once we have the values of Y that respect the ranking relations D, G becomes independent of D. Thus, sampling G from $p(G|Y, D)$ becomes the same as sampling from $p(G|Y)$.

4 Gibbs sampler for the joint posterior — $p(G, Y|D)$

The interdependence of G and Y helps design iterative algorithms for sampling $p(G, Y|D)$. In this case, since each step samples from a conditional distribution ($p(G|Y, D)$ or $p(Y|G, D)$), they can be combined to form a Gibbs sampler (Geman and Geman, 1984). In the same order as described in Section 3, the two conditional sampling steps are implemented as follows:

1. Sample an ordering relation d according to the prior $p(D)$, which is simply normalized frequency counts; sample a vector of constraint values $y = \{y_1, \cdots, y_N\}$ from the normal distributions $N(\mu_1^{(t)}, \sigma^2), \cdots, N(\mu_N^{(t)}, \sigma^2)$ such that y observes the ordering in d;

2. Repeat Step 1 and obtain M samples of missing data: $y^1, \cdots, y^M$; sample $\mu_i^{(t+1)}$ from $N(\sum_j y_i^j/M, \sigma^2/M)$.

The grammar $G = (\mu_1, \cdots, \mu_N)$, and the superscript $^{(t)}$ represents a sample of G in iteration t. As explained in 3, Step 1 samples missing data from $p(Y|G, D)$, and Step 2 is equivalent to sampling from $p(G|Y, D)$, by the conditional independence of G and D given Y. The normal posterior distribution $N(\sum_j y_i^j/M, \sigma^2/M)$ is derived by using $p(G|Y) \propto p(Y|G)p(G)$, where $p(Y|G)$ is normal, and $p(G) \sim N(\mu_0, \sigma_0)$ is chosen to be an non-informative prior with $\sigma_0 \to \infty$.

M (the number of missing data) is not a crucial parameter. In our experiments, M is set to the total number of observed forms[4]. Although it may seem that σ^2/M is small for a large M and does not play

[3] Notice even computing the gradient is non-trivial.

[4] Other choices of M, e.g. $M = 1$, lead to more or less the same running time.

a significant role in the sampling of $\mu_i^{(t+1)}$, the variance of the sampling distribution is a necessary ingredient of the Gibbs sampler[5].

Under fairly general conditions (Geman and Geman, 1984), the Gibbs sampler iterates these two steps until it converges to a unique stationary distribution. In practice, convergence can be monitored by calculating cross-sample statistics from multiple Markov chains with different starting points (Gelman and Rubin, 1992). After the simulation is stopped at convergence, we will have obtained a perfect sample of $p(G, Y|D)$. These samples can be used to derive our target distribution $p(G|D)$ by simply keeping all the G components, since $p(G|D)$ is a marginal distribution of $p(G, Y|D)$. Thus, the sampling-based approach gives us the advantage of doing inference without performing any integration.

5 Computational issues in implementation

In this section, I will sketch some key steps in the implementation of the Gibbs sampler. Particular attention is paid to sampling $p(Y|G, D)$, since a direct implementation may require an unrealistic running time.

5.1 Computing $p(D)$ from linguistic data

The prior probability $p(D)$ determines the number of samples (missing data) that are drawn under each ordering relation. The following example illustrates how the ordering D and $p(D)$ are calculated from data collected in a linguistic analysis. Consider a data set that contains 2 inputs and a few outputs, each associated with an observed frequency in the lexicon:

		C1	C2	C3	C4	C5	Freq.
x_1	y_{11}	0	1	0	1	0	4
	y_{12}	1	0	0	0	0	3
	y_{13}	0	1	1	0	1	0
	y_{14}	0	0	1	0	0	0
x_2	y_{21}	1	1	0	0	0	3
	y_{22}	0	0	1	1	1	0

Table 2: A Stochastic OT grammar with 2 inputs

The three ordering relations (corresponding to 3 attested outputs) and $p(D)$ are computed as follows:

Ordering Relation D	$p(D)$
$\left\{ \begin{array}{r} C1>\max\{C2, C4\} \\ \max\{C3, C5\}>C4 \\ C3>\max\{C2, C4\} \end{array} \right.$	.4
$\left\{ \begin{array}{r} \max\{C2, C4\}>C1 \\ \max\{C2, C3, C5\}>C1 \\ C3>C1 \end{array} \right.$	.3
$\max\{C3, C4, C5\} > \max\{C1, C2\}$	.3

Table 3: The ordering relations D and $p(D)$ computed from Table 2.

Here each ordering relation has several conjuncts, and the number of conjuncts is equal to the number of competing candidates for each given input. These conjuncts need to hold simultaneously because each winning candidate needs to be more harmonic than all other competing candidates. The probabilities $p(D)$ are obtained by normalizing the frequencies of the surface forms in the original data. This will have the consequence of placing more weight on lexical items that occur frequently in the corpus.

5.2 Sampling $p(Y|G, D)$ under complex ordering relations

A direct implementation $p(Y|G, d)$ is straightforward: 1) first obtain N samples from N Gaussian distributions; 2) check each conjunct to see if the ordering relation is satisfied. If so, then keep the sample; if not, discard the sample and try again.

However, this can be highly inefficient in many cases. For example, if m constraints appear in the ordering relation d and the sample is rejected, the $N - m$ random numbers for constraints not appearing in d are also discarded. When d has several conjuncts, the chance of rejecting samples for irrelevant constraints is even greater.

In order to save the generated random numbers, the vector Y can be decomposed into its 1-dimensional components $(Y_1, Y_2, \cdots, Y_N)$. The problem then becomes sampling $p(Y_1, \cdots, Y_N|G, D)$. Again, we may use conditional sampling to draw y_i one at a time: we keep $y_{j \neq i}$ and d fixed[6], and draw y_i so that d holds for y. There are now two cases: if d holds regardless of y_i, then any sample from $N(\mu_i^{(t)}, \sigma^2)$ will do; otherwise, we will need to draw y_i from a truncated

[5] As required by the proof in (Geman and Geman, 1984).

[6] Here we use $y_{j \neq i}$ for all components of y except the i-th dimension.

normal distribution.

To illustrate this idea, consider an example used earlier where d=“$\max\{c_1, c_2\} > c_3$”, and the initial sample and parameters are $(y_1^{(0)}, y_2^{(0)}, y_3^{(0)}) = (\mu_1^{(0)}, \mu_2^{(0)}, \mu_3^{(0)}) = (1, -1, 0)$.

Sampling dist.	Y_1	Y_2	Y_3
$p(Y_1 \mid \mu_1, Y_1 > y_3)$	$\underline{2.3799}$	-1.0000	0
$p(Y_2 \mid \mu_2)$	2.3799	$\underline{-0.7591}$	0
$p(Y_3 \mid \mu_3, Y_3 < y_1)$	2.3799	-0.7591	$\underline{-1.0328}$
$p(Y_1 \mid \mu_1)$	$\underline{-1.4823}$	-0.7591	-1.0328
$p(Y_2 \mid \mu_2, Y_2 > y_3)$	-1.4823	$\underline{2.1772}$	-1.0328
$p(Y_3 \mid \mu_3, Y_3 < y_2)$	-1.4823	2.1772	$\underline{1.0107}$

Table 4: Conditional sampling steps for $p(Y|G, d) = p(Y_1, Y_2, Y_3|\mu_1, \mu_2, \mu_3, d)$

Notice that in each step, the sampling density is either just a normal, or a truncated normal distribution. This is because we only need to make sure that d will continue to hold for the next sample $y^{(t+1)}$, which differs from $y^{(t)}$ by just 1 constraint.

In our experiment, sampling from truncated normal distributions is realized by using the idea of *rejection sampling*: to sample from a truncated normal[7] $\pi_c(x) = \frac{1}{Z(c)} \cdot N(\mu, \sigma) \cdot I_{\{x>c\}}$, we first find an *envelope* density function $g(x)$ that is easy to sample directly, such that $\pi_c(x)$ is uniformly bounded by $M \cdot g(x)$ for some constant M that does not depend on x. It can be shown that once each sample x from $g(x)$ is rejected with probability $r(x) = 1 - \frac{\pi_c(x)}{M \cdot g(x)}$, the resulting histogram will provide a perfect sample for $\pi_c(x)$. In the current work, the exponential distribution $g(x) = \lambda \exp\{-\lambda x\}$ is used as the envelope, with the following choices for λ and the rejection ratio $r(x)$, which have been optimized to lower the rejection rate:

$$\begin{aligned} \lambda &= \frac{c + \sqrt{c + 4\sigma^2}}{2\sigma^2} \\ r(x) &= \exp\left\{\frac{(x+c)^2}{2} + \lambda_0(x+c) - \frac{\sigma^2\lambda_0^2}{2}\right\} \end{aligned}$$

Putting these ideas together, the final version of Gibbs sampler is constructed by implementing Step 1 in Section 4 as a sequence of conditional sampling steps for $p(Y_i|Y_{j \neq i}, d)$, and combining them with the sampling of $p(G|Y, D)$. Notice the order in which Y_i is updated is fixed, which makes our implementation an instance of the *systematic-scan* Gibbs sampler (Liu, 2001). This implementation may be improved even further by utilizing the structure of the ordering relation d, and optimizing the order in which Y_i is updated.

5.3 Model identifiability

Identifiability is related to the uniqueness of solution in model fitting. Given N constraints, a grammar $G \in R^N$ is not identifiable because $G + C$ will have the same behavior as G for any constant $C = (c_0, \cdots, c_0)$. To remove translation invariance, in Step 2 the average ranking value is subtracted from G, such that $\sum_i \mu_i = 0$.

Another problem related to identifiability arises when the data contains the so-called “categorical domination”, i.e., there may be data of the following form:

$c_1 > c_2$ *with probability* 1.

In theory, the mode of the posterior tends to infinity and the Gibbs sampler will not converge. Since having categorical dominance relations is a common practice in linguistics, we avoid this problem by truncating the posterior distribution[8] by $I_{|\mu|<K}$, where K is chosen to be a positive number large enough to ensure that the model be identifiable. The role of truncation/renormalization may be seen as a strong prior that makes the model identifiable on a bounded set.

A third problem related to identifiability occurs when the posterior has multiple modes, which suggests that multiple grammars may generate the same output frequencies. This situation is common when the grammar contains interactions between many constraints, and greedy algorithms like GLA tend to find one of the many solutions. In this case, one can either introduce extra ordering relations or use informative priors to sample $p(G|Y)$, so that the inference on the posterior can be done with a relatively small number of samples.

5.4 Posterior inference

Once the Gibbs sampler has converged to its stationary distribution, we can use the samples to make var-

[7] Notice the truncated distribution needs to be re-normalized in order to be a proper density.

[8] The implementation of sampling from truncated normals is the same as described in 5.2.

ious inferences on the posterior. In the experiments reported in this paper, we are primarily interested in the mode of the posterior marginal[9] $p(\mu_i|D)$, where $i = 1, \cdots, N$. In cases where the posterior marginal is symmetric and uni-modal, its mode can be estimated by the sample median.

In real linguistic applications, the posterior marginal may be a skewed distribution, and many modes may appear in the histogram. In these cases, more sophisticated non-parametric methods, such as kernel density estimation, can be used to estimate the modes. To reduce the computation in identifying multiple modes, a mixture approximation (by EM algorithm or its relatives) may be necessary.

6 Experiments

6.1 Ilokano reduplication

The following Ilokano grammar and data set, used in (Boersma and Hayes, 2001), illustrate a complex type of constraint interaction: the interaction between the three constraints: *COMPLEX-ONSET, ALIGN, and $IDENT_{BR}$([long]) cannot be factored into interactions between 2 constraints. For any given candidate to be optimal, the constraint that prefers such a candidate must simultaneously dominate the other two constraints. Hence it is not immediately clear whether there is a grammar that will assign equal probability to the 3 candidates.

/HRED-bwaja/	$p(.)$	*C-ONS	AL	I_{BR}
bu:.bwa.ja	.33	1	0	1
bwaj.bwa.ja	.33	2	0	0
bub.wa.ja	.33	0	1	0

Table 5: Data for Ilokano reduplication.

Since it does not address the problem of identifiability, the GLA does not always converge on this data set, and the returned grammar does not always fit the input frequencies exactly, depending on the choice of parameters[10].

In comparison, the Gibbs sampler converges quickly[11], regardless of the parameters. The result suggests the existence of a unique grammar that will assign equal probabilities to the 3 candidates. The posterior samples and histograms are displayed in Figure 1. Using the median of the marginal posteriors, the estimated grammar generates an exact fit to the frequencies in the input data.

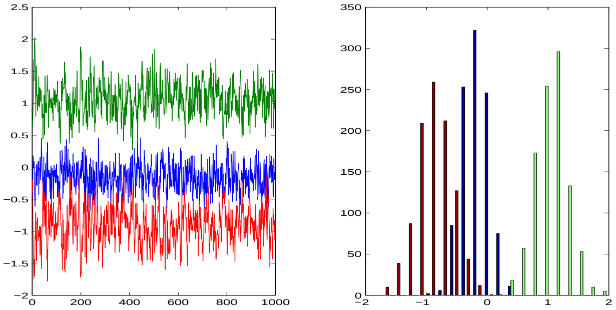

Figure 1: Posterior marginal samples and histograms for Experiment 2.

6.2 Spanish diminutive suffixation

The second experiment uses linguistic data on Spanish diminutives and the analysis proposed in (Arbisi-Kelm, 2002). There are 3 base forms, each associated with 2 diminutive suffixes. The grammar 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 grammar 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 [liryosito]. Yet unlike [lityosito], [ubasita] is not observed.

Input	Output	Freq.	**A**	**M**	**D**	**B**
/uba/	[ubita]	10	0	1	0	1
	[ubasita]	0	1	0	0	0
/mar/	[marEsito]	5	0	0	1	0
	[marsito]	5	0	0	0	1
/liryo/	[liri.ito]	9	0	1	0	0
	[liryosito]	1	1	0	0	0

Table 6: Data for Spanish diminutive suffixation.

In the results found by GLA, [marEsito] always has a lower frequency than [marsito] (See Table 7). This is not accidental. Instead it reveals a problematic use of heuristics in GLA[12]: since the constraint **B** is violated by [ubita], it is always demoted whenever the underlying form /uba/ is encountered during learning. Therefore, even though the expected

[9] Note $G = (\mu_1, \cdots, \mu_N)$, and $p(\mu_i|D)$ is a marginal of $p(G|D)$.

[10] B &H reported results of averaging many runs of the algorithm. Yet there appears to be significant randomness in each run of the algorithm.

[11] Within 1000 iterations.

[12] Thanks to Bruce Hayes for pointing out this problem.

model assigns equal values to μ_3 and μ_4 (corresponding to **D** and **B**, respectively), μ_3 is always less than μ_4, simply because there is more chance of penalizing **D** rather than **B**. This problem arises precisely because of the heuristic (i.e. demoting the constraint that prefers the wrong candidate) that GLA uses to find the target grammar.

The Gibbs sampler, on the other hand, does not depend on heuristic rules in its search. Since modes of the posterior $p(\mu_3|D)$ and $p(\mu_4|D)$ reside in negative infinity, the posterior is truncated by $I_{\mu_i<K}$, with $K = 6$, based on the discussion in 5.3. Results of the Gibbs sampler and two runs of GLA[13] are reported in Table 7.

Input	Output	Obs	Gibbs	GLA$_1$	GLA$_2$
/uba/	[ubita]	100%	95%	96%	96%
	[ubasita]	0%	5%	4%	4%
/mar/	[marEsito]	50%	50%	38%	45%
	[marsito]	50%	50%	62%	55%
/liryo/	[liri.ito]	90%	95%	96%	91.4%
	[liryosito]	10%	5%	4%	8.6%

Table 7: Comparison of Gibbs sampler and GLA

7 A comparison with Max-Ent models

Previously, problems with the GLA[14] have inspired other OT-like models of linguistic variation. One such proposal suggests using the more well-known *Maximum Entropy* model (Goldwater and Johnson, 2003). In Max-Ent models, a grammar G is also parameterized by a real vector of weights $w = (w_1, \cdots, w_N)$, but the conditional likelihood of an output y given an input x is given by:

$$p(y|x) = \frac{\exp\{\sum_i w_i f_i(y,x)\}}{\sum_z \exp\{\sum_i w_i f_i(z,x)\}} \qquad (2)$$

where $f_i(y, x)$ is the violation each constraint assigns to the input-output pair (x, y).

Clearly, Max-Ent is a rather different type of model from Stochastic OT, not only in the use of constraint ordering, but also in the objective function (conditional likelihood rather than likelihood/posterior). However, it may be of interest to compare these two types of models. Using the same data as in 6.2, results of fitting Max-Ent (using conjugate gradient descent) and Stochastic OT (using Gibbs sampler) are reported in Table 8:

Input	Output	Obs	SOT	ME	ME$_{sm}$
/uba/	[ubita]	100%	95%	100%	97.5%
	[ubasita]	0%	5%	0%	2.5%
/mar/	[marEsito]	50%	50%	50%	48.8%
	[marsito]	50%	50%	50%	51.2%
/liryo/	[liri.ito]	90%	95%	90%	91.4%
	[liryosito]	10%	5%	10%	8.6%

Table 8: Comparison of Max-Ent and Stochastic OT models

It can be seen that the Max-Ent model, in the absence of a smoothing prior, fits the data perfectly by assigning positive weights to constraints **B** and **D**. A less exact fit (denoted by ME$_{sm}$) is obtained when the smoothing Gaussian prior is used with $\mu_i = 0$, $\sigma_i^2 = 1$. But as observed in 6.2, an exact fit is impossible to obtain using Stochastic OT, due to the difference in the way variation is generated by the models. Thus it may be seen that Max-Ent is a more powerful class of models than Stochastic OT, though it is not clear how the Max-Ent model's descriptive power is related to generative linguistic theories like phonology.

Although the abundance of well-behaved optimization algorithms has been pointed out in favor of Max-Ent models, it is the author's hope that the MCMC approach also gives Stochastic OT a similar underpinning. However, complex Stochastic OT models often bring worries about identifiability, whereas the convexity property of Max-Ent may be viewed as an advantage[15].

8 Discussion

From a non-Bayesian perspective, the MCMC-based approach can be seen as a randomized strategy for learning a grammar. Computing resources make it possible to explore the entire space of grammars and discover where good hypotheses are likely to occur. In this paper, we have focused on the frequently visited areas of the hypothesis space.

It is worth pointing out that the Graduate Learning Algorithm can also be seen from this perspective. An examination of the GLA shows that when the plasticity term is fixed, parameters found by GLA also form a Markov chain $G^{(t)} \in R^N$, $t = 1, 2, \cdots$. Therefore, assuming the model is identifiable, it

[13] 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.

[14] See (Keller and Asudeh, 2002) for a summary.

[15] Concerns about identifiability appear much more frequently in statistics than in linguistics.

seems possible to use GLA in the same way as the MCMC methods: rather than forcing it to stop, we can run GLA until it reaches stationary distribution, if it exists.

However, it is difficult to interpret the results found by this "random walk-GLA" approach: the stationary distribution of GLA may not be the target distribution — the posterior $p(G|D)$. To construct a Markov chain that converges to $p(G|D)$, one may consider turning GLA into a real MCMC algorithm by designing *reversible jumps*, or the *Metropolis algorithm*. But this may not be easy, due to the difficulty in likelihood evaluation (including likelihood ratio) discussed in Section 2.

In contrast, our algorithm provides a general solution to the problem of learning Stochastic OT grammars. Instead of looking for a Markov chain in R^N, we go to a higher dimensional space $R^N \times R^N$, using the idea of data augmentation. By taking advantage of the interdependence of G and Y, the Gibbs sampler provides a Markov chain that converges to $p(G, Y|D)$, which allows us to return to the original subspace and derive $p(G|D)$ — the target distribution. Interestingly, by adding more parameters, the computation becomes simpler.

9 Future work

This work can be extended in two directions. First, it would be interesting to consider other types of OT grammars, in connection with the linguistics literature. For example, the variances of the normal distribution are fixed in the current paper, but they may also be treated as unknown parameters (Nagy and Reynolds, 1997). Moreover, constraints may be parameterized as mixture distributions, which represent other approaches to using OT for modeling linguistic variation (Anttila, 1997).

The second direction is to introduce informative priors motivated by linguistic theories. It is found through experimentation that for more sophisticated grammars, identifiability often becomes an issue: some constraints may have multiple modes in their posterior marginal, and it is difficult to extract modes in high dimensions[16]. Therefore, use of priors is needed in order to make more reliable inferences. In addition, priors also have a linguistic appeal, since current research on the "initial bias" in language acquisition can be formulated as priors (e.g. *Faithfulness Low* (Hayes, 2004)) from a Bayesian perspective.

Implementing these extensions will merely involve modifying $p(G|Y, D)$, which we leave for future work.

[16] Notice that posterior marginals do not provide enough information for modes of the joint distribution.

References

Anttila, A. (1997). *Variation in Finnish Phonology and Morphology*. PhD thesis, Stanford University.

Arbisi-Kelm, T. (2002). An analysis of variability in Spanish diminutive formation. Master's thesis, UCLA, Los Angeles.

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

Boersma, P. and Hayes, B. P. (2001). Empirical tests of the Gradual Learning Algorithm. *Linguistic Inquiry*, 32:45–86.

Gelfand, A. and Smith, A. (1990). Sampling-based approaches to calculating marginal densities. *Journal of the American Statistical Association*, 85(410).

Gelman, A. and Rubin, D. B. (1992). Inference from iterative simulation using multiple sequences. *Statistical Science*, 7:457–472.

Geman, S. and Geman, D. (1984). Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 6(6):721–741.

Goldwater, S. and Johnson, M. (2003). Learning OT constraint rankings using a Maximum Entropy model. In Spenader, J., editor, *Proceedings of the Workshop on Variation within Optimality Theory*, Stockholm.

Hayes, B. P. (2004). Phonological acquisition in optimality theory: The early stages. In Kager, R., Pater, J., and Zonneveld, W., editors, *Fixing Priorities: Constraints in Phonological Acquisition*. Cambridge University Press.

Keller, F. and Asudeh, A. (2002). Probabilistic learning algorithms and Optimality Theory. *Linguistic Inquiry*, 33(2):225–244.

Liu, J. S. (2001). *Monte Carlo Strategies in Scientific Computing*. Number 33 in Springer Statistics Series. Springer-Verlag, Berlin.

Nagy, N. and Reynolds, B. (1997). Optimality theory and variable word-final deletion in Faetar. *Language Variation and Change*, 9.

Prince, A. and Smolensky, P. (1993). *Optimality Theory: Constraint Interaction in Generative Grammar*. Forthcoming.

Tanner, M. and Wong, W. H. (1987). The calculation of posterior distributions by data augmentation. *Journal of the American Statistical Association*, 82(398).