Inducing Synchronous Grammars with Slice Sampling

Phil Blunsom
Computing Laboratory
Oxford University
Phil.Blunsom@comlab.ox.ac.uk

Trevor Cohn
Department of Computer Science
University of Sheffield
T.Cohn@dcs.shef.ac.uk

Abstract

This paper describes an efficient sampler for synchronous grammar induction under a non-parametric Bayesian prior. Inspired by ideas from *slice sampling*, our sampler is able to draw samples from the posterior distributions of models for which the standard dynamic programing based sampler proves intractable on non-trivial corpora. We compare our sampler to a previously proposed Gibbs sampler and demonstrate strong improvements in terms of both training log-likelihood and performance on an end-to-end translation evaluation.

1 Introduction

Intractable optimisation algorithms abound in much of the recent work in Natural Language Processing. In fact, there is an increasing acceptance that solutions to many of the great challenges of NLP (e.g. machine translation, summarisation, question answering) will rest on the quality of approximate inference. In this work we tackle this problem in the context of inducing synchronous grammars for a machine translation system. We concern ourselves with the lack of a principled, and scalable, algorithm for learning a synchronous context free grammar (SCFG) from sentence-aligned parallel corpora.

The predominant approach for learning phrase-based translation models (both finite state or synchronous grammar based) uses a cascade of heuristics beginning with predicted word alignments and producing a weighted set of translation rules (Koehn et al., 2003). Alternative approaches avoid such heuristics, instead learning structured alignment models directly from sentence aligned data (e.g., (Marcu and Wong, 2002; Cherry and Lin, 2007; DeNero et al., 2008; Blunsom et al., 2009)). Although these models are theoretically attractive, inference is intractable (at least $\mathcal{O}(|\mathbf{f}|^3|\mathbf{e}|^3)$). The efficacy of direct estimation of structured alignment models therefore rests on the approximations used to make inference practicable – typically heuristic

constraints or Gibbs sampling. In this work we show that naive Gibbs sampling (specifically, Blunsom et al. (2009)) is ineffectual for inference and reliant on a high quality initialisation, mixing very slowly and being easily caught in modes. Instead, blocked sampling over sentence pairs allows much faster mixing, but done in the obvious way (following Johnson et al. (2007)) would incur a $\mathcal{O}(|\mathbf{f}|^3|\mathbf{e}|^3)$ time complexity.

Here we draw inspiration from the work of Van Gael et al. (2008) on inference in infinite hidden Markov models to develop a novel algorithm for efficient sampling from a SCFG. We develop an auxiliary variable 'slice' sampler which can dramatically reduce inference complexity, and thereby make blocked sampling practicable on real translation corpora. Our evaluation demonstrates that our algorithm mixes more quickly than the local Gibbs sampler, and produces translation models which achieve state-of-the-art BLEU scores without using GIZA++ or symmetrisation heuristics for initialisation.

We adopt the generative model of Blunsom et al. (2009) which creates a parallel sentence pair by a sequence (derivation) of SCFG productions $\mathbf{d} = (r_1, r_2, ..., r_n)$. The tokens in each language can be read off the leaves of the derivation tree while their order is defined hierarchically by the productions in use. The probability of a derivation is defined as $p(\mathbf{d}|\theta) = \prod_{r \in \mathbf{d}} \theta_r$ where θ are the model parameters which are drawn from a Bayesian prior. We deviate from that models definition of the prior over phrasal translations, instead adopting the hierarchical Dirichlet process prior from DeNero et al. (2008), which incorporates IBM Model 1. Blunsom et al. (2009) describe a blocked sampler following Johnson et al. (2007) which uses the Metropolis-Hastings algorithm to correct proposal samples drawn from an approximating SCFG, however this is discounted as impractical due to the $\mathcal{O}(|\mathbf{f}|^3|\mathbf{e}|^3)$ complexity. Instead a Gibbs sampler is used which samples local updates to the derivation structure of each training instance. This avoids the dynamic program of the

blocked sampler but at the expense of considerably slower mixing.

Recently Bouchard-Côté et al. (2009) proposed an auxialliary variable sampler, possibly complementary to ours, which was also evaluated on synchronous parsing. Rather than slice sampling derivations in a collapsed Bayesian model, this model employed a secondary proposal model (IBM Models) and sampled expectations over rule parameters.

2   Slice Sampling a SCFG

It would be advantageous to explore a middle ground where the scope of the dynamic program is limited to high probability regions, reducing the running time to an acceptable level. By employing the technique of *slice sampling* (Neal, 2003) we describe an algorithm which stochastically samples from a reduced space of possible derivations, while ensuring that these samples are drawn from the correct distribution. We apply the slice sampler to the approximating SCFG parameterised by θ, which requires samples from an *inside* chart $p(\mathbf{d}|\theta)$ (for brevity, we omit the dependency on θ in the following).

Slice sampling is an example of *auxiliary variable sampling* in which we make use of the fact that if we can draw samples from a joint distribution, then we can trivially obtain samples from the marginal distributions: $p(\mathbf{d}) = \sum_{\mathbf{u}} p(\mathbf{d}, \mathbf{u})$, where $\mathbf{d}$ is the variable of interest and $\mathbf{u}$ is an auxiliary variable. Using a Gibbs sampler we can draw samples from this joint distribution by alternately sampling from $p(\mathbf{d}|\mathbf{u})$ and $p(\mathbf{u}|\mathbf{d})$. The trick is to ensure that $\mathbf{u}$ is defined such that drawing samples from $p(\mathbf{d}|\mathbf{u})$ is more efficient than from $p(\mathbf{d})$.

We define the variable $\mathbf{u}$ to contain a slice variable u_s for every cell of a synchronous parse chart for every training instance:[1]

$$\mathcal{S} = \{(i, j, x, y) \mid 0 \leq i < j \leq |\mathbf{f}|, 0 \leq x < y \leq |\mathbf{e}|\}$$
$$\mathbf{u} = \{u_s \in \mathbb{R} \mid 0 < u_s < 1, s \in \mathcal{S}\}$$

These slice variables act as cutoffs on the probabilities of the rules considered in each cell s: rule applications r_s with $\theta_{r_s} \leq u_s$ will be pruned from the dynamic program.[2]

Sampling $p(\mathbf{u}|\mathbf{d})$ Unlike Van Gael et al. (2008), there is not a one-to-one correspondence between the spans of the rules in $\mathbf{d}$ and the set $\mathcal{S}$, rather the derivation's rule spans form a subset of $\mathcal{S}$. This complicates our definition of $p(\mathbf{u}|\mathbf{d})$; we must provide separate accounts of how each u_s is generated depending on whether there is a corresponding rule for s, i.e., $r_s \in \mathbf{d}$. We define $p(\mathbf{u}|\mathbf{d}) = \prod_s p(u_s|\mathbf{d})$, where:

$$p(u_s|\mathbf{d}) = \begin{cases} \frac{\mathbb{I}(u_s < \theta_{r_s})}{\theta_{r_s}} & \text{, if } r_s \in \mathbf{d} \\ \beta(u_s; a, b) & \text{, else} \end{cases} \quad (1)$$

which mixes a uniform distribution and a Beta distribution[3] depending on the existence of a rule r_s in the derivation $\mathbf{d}$.[4] Eq. 1 is constructed such that only rules with probability greater than the relevant threshold, $\{r_s \mid \theta_{r_s} > u_s\}$, could have feasibly been part of a derivation resulting in auxiliary variable $\mathbf{u}$. This is critical in reasoning over the reverse conditional $p(\mathbf{d}|\mathbf{u})$ which only has to consider the reduced space of rules (formulation below in (4)). Trivially, the conditioning derivation is recoverable, $\forall r_s \in \mathbf{d}, \theta_{r_s} \geq u_s$. We parameterise the β distribution in (1) with a heavy skew towards zero in order to limit the amount of pruning and thereby include many competing derivations.[5]

Sampling $p(\mathbf{d}|\mathbf{u})$ Recall the probability of a derivation, $p(\mathbf{d}) = \prod_{r_s \in \mathbf{d}} \theta_{r_s}$. We draw samples from the joint distribution, $p(\mathbf{d}, \mathbf{u})$, holding $\mathbf{u}$ fixed:

$$p(\mathbf{d}|\mathbf{u}) \propto p(\mathbf{d}, \mathbf{u}) = p(\mathbf{d}) \times p(\mathbf{u}|\mathbf{d})$$

$$= \left(\prod_{r_s \in \mathbf{d}} \theta_{r_s} \right) \times \left(\begin{array}{c} \prod_{u_s:r_s \in \mathbf{d}} \frac{\mathbb{I}(u_s < \theta_{r_s})}{\theta_{r_s}} \\ \times \prod_{u_s:r_s \notin \mathbf{d}} \beta(u_s; a, b) \end{array} \right)$$

$$= \prod_{u_s:r_s \in \mathbf{d}} \mathbb{I}(u_s < \theta_{r_s}) \prod_{u_s:r_s \notin \mathbf{d}} \beta(u_s; a, b) \quad (2)$$

$$= \prod_{u_s:r_s \in \mathbf{d}} \frac{\mathbb{I}(u_s < \theta_{r_s})}{\beta(u_s; a, b)} \prod_{u_s} \beta(u_s; a, b) \quad (3)$$

$$\propto \prod_{u_s:r_s \in \mathbf{d}} \frac{\mathbb{I}(u_s < \theta_{r_s})}{\beta(u_s; a, b)} \quad (4)$$

In step (2) we cancel the θ_{r_s} terms while in step (3) we introduce $\beta(u_s; a, b)$ terms to the numerator and denominator for $u_s : r_s \in \mathbf{d}$ to simplify the range

---

[1]The dependence on training instances is omitted here and subsequently for simplicity. Each instance is independent, and therefore this formulation can be trivially applied to a set.

[2]Alternatively, we could naively sample from a pruned chart using a fixed beam threshold. However, this would not produce samples from $p(\mathbf{d})$, but some other unknown distribution.

[3]Any distribution defined over $\{x \in \mathbb{R} \mid 0 \leq x \leq 1\}$ may be used in place of β, however this may affect the efficiency of the sampler.

[4]$\mathbb{I}(\cdot)$ returns 1 if the condition is true and 0 otherwise.

[5]We experiment with a range of $a < 1$ while fixing $b = 1$.

System	BLEU	time(s)	LLH
Moses (default settings)	47.3	–	–
LB init.	36.5	–	-257.1
M1 init.	48.8	–	-153.4
M4 init.	49.1	–	-151.4
Gibbs LB init.	45.3	44	-135.4
Gibbs M1 init.	48.2	40	-120.5
Gibbs M4 init. (Blunsom et al., 2009)	49.6	44	-110.3
Slice (a=0.15, b=1) LB init.	47.3	180	-98.9
Slice (a=0.10, b=1) M1 init.	50.4	908	-89.4
Slice (a=0.15, b=1) M1 init.	49.9	144	-90.2
Slice (a=0.25, b=1) M1 init.	49.2	80	-95.6

Table 1: IWSLT Chinese to English translation.

of the second product. The last step (4) discards the term $\prod_{u_s} \beta(u_s; a, b)$ which is constant wrt $\mathbf{d}$. The net result is a formulation which factors with the derivation structure, thereby eliminating the need to consider all $\mathcal{O}(|\mathbf{e}|^2|\mathbf{f}|^2)$ spans in $\mathcal{S}$. Critically $p(\mathbf{d}|\mathbf{u})$ is zero for all spans failing the $\mathbb{I}(u_s < \theta_{r_s})$ condition.

To exploit the decomposition of Equation 4 we require a parsing algorithm that only explores chart cells whose child cells have not already been pruned by the slice variables. The standard approach of using synchronous CYK (Wu, 1997) doesn't posses this property: all chart cells would be visited even if they are to be pruned. Instead we use an agenda based parsing algorithm, in particular we extend the algorithm of Klein and Manning (2004) to synchronous parsing.[6] Finally, we need a Metropolis-Hastings acceptance step to account for intra-instance dependencies (the 'rich-get-richer' effect). We omit the details, save to state that the calculation cancels to the same test as presented in Johnson et al. (2007).[7]

3 Evaluation

In the following experiments we compare the slice sampler and the Gibbs sampler (Blunsom et al., 2009), in terms of mixing and translation quality. We measure mixing in terms of training log-likelihood (LLH) after a fixed number of sampling iterations. Translations are produced using Moses (Koehn et al., 2007), initialised with the word alignments from the final sample, and are evaluated using BLEU(Papineni et al., 2001). The slice sampled models are restricted to learning binary branching one-to-one (or null) alignments,[8] while no restriction is placed on the Gibbs sampler (both use the same model, so have

comparable LLH). Of particular interest is how the different samplers perform given initialisations of varying quality. We evaluate three initialisers: **M4**: the symmetrised output of GIZA++ factorised into ITG form (as used in Blunsom et al. (2009)); **M1**: the output of a heavily pruned ITG parser using the IBM Model 1 prior for the rule probabilities;[9] and **LB**: left-branching monotone derivations.[10]

We experiment with the Chinese→English translation task from IWSLT, as used in Blunsom et al. (2009).[11] Figure 1 shows LLH curves for the samplers initialised with the M1 and LB derivations, plus the curve for Gibbs sampler with the M4 initialiser.[12] Table 1 gives BLEU scores on Test-05 for phrase-based translation models built from the 1500th sample for the various models along with the average time per sample and their final log-likelihood.

4 Discussion

The results are particularly encouraging. The slice sampler uniformly finds much better solutions than the Gibbs sampler regardless of initialisation. In particular, the slice sampled model initialised with the naive LB structure achieves a higher likelihood than the M4 initialised model, although this is not reflected in their relative BLEU scores. In contrast the Gibbs sampler is more significantly affected by its initialisation, only deviating slightly before becoming trapped in a mode, as seen in Fig. 1. With sufficient (infinite) time both sampling strategies will converge on the true posterior regardless of initialisation, however the slice sampler appears to be converging much faster than the Gibbs sampler.

Interestingly, the initialisation heuristics (M1 and M4) outperform the default heuristics (Koehn et al., 2007) by a considerable margin. This is most likely because the initialisation heuristics force the alignments to factorise with an ITG, resulting in more aggressive pruning of spurious alignments which in turn allows for more and larger phrase pairs.

---

[6]Moreover, we only sample values for u_s as they are visited by the parser, thus avoiding the quartic complexity.

[7]Acceptance rates averaged above 99%.

[8]This restriction is not strictly necessary, however it greatly simplifies the implementation and increases efficiency.

[9]The following beam heuristics are employed: alignments to null are only permitted on the longer sentence side; words are only allowed to align to those whose relative sentence position is within ±3 words.

[10]Words of the longer sentence are randomly assigned to null.

[11]We limit the maximum training sentence length to 40, resulting in $\sim 40k$ training sentences.

[12]The GIZA++ M4 alignments don't readily factorise to word-based ITG derivations, as such we haven't produced results for this initialiser using the slice sampler.
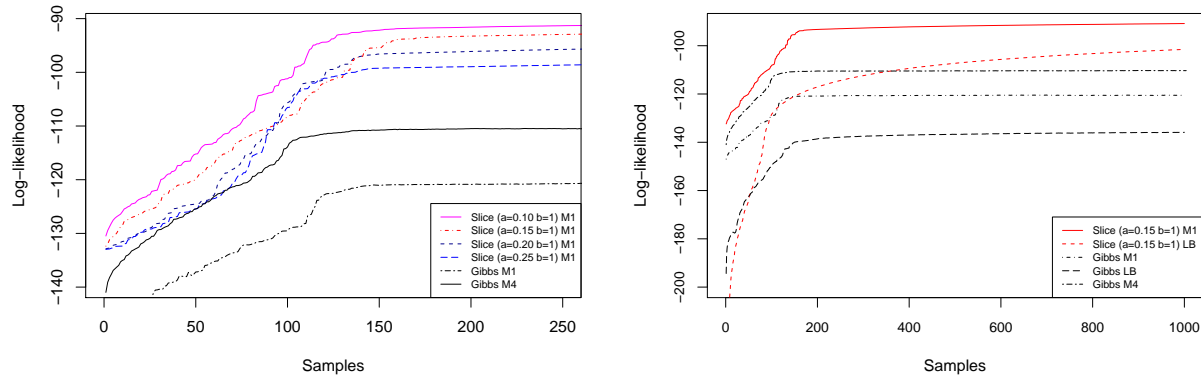
Figure 1: Training log-likelihood as a function of sampling iteration for Gibbs and slice sampling.

While the LLHs for the slice sampled models and their BLEU scores appear correlated, this doesn't extend to comparisons with the Gibbs sampled models. We believe that this is because the GIZA++ initialisation alignments also explain the data well, while not necessarily obtaining a high LLH under the ITG model. Solutions which score highly in one model score poorly in the other, despite both producing good translations.

The slice sampler is slower than the local Gibbs sampler, its speed depending on the parameterisation of the Beta distribution (affecting the width of the beam). In the extreme, exhaustive search using the full dynamic program is intractable on current hardware,[13] and therefore we have achieved our aim of mediating between local and blocked inference.

This investigation has established the promise of the SCFG slice sampling technique to provide a scalable inference algorithm for non-parametric Bayesian models. With further development, this work could provide the basis for a family of principled inference algorithms for parsing models, both monolingual and synchronous, and other models that prove intractable for exact dynamic programming.

References

P. Blunsom, T. Cohn, C. Dyer, M. Osborne. 2009. A Gibbs sampler for phrasal synchronous grammar induction. In *Proc. ACL/IJCNLP*, 782–790, Suntec, Singapore. Association for Computational Linguistics.

A. Bouchard-Côté, S. Petrov, D. Klein. 2009. Randomized pruning: Efficiently calculating expectations in large dynamic programs. In *Advances in Neural Information Processing Systems 22*, 144–152.

C. Cherry, D. Lin. 2007. Inversion transduction grammar for joint phrasal translation modeling. In *Proc. SSST*, Rochester, USA.

J. DeNero, A. Bouchard-Côté, D. Klein. 2008. Sampling alignment structure under a Bayesian translation model. In *Proc. EMNLP*, 314–323, Honolulu, Hawaii.

M. Johnson, T. Griffiths, S. Goldwater. 2007. Bayesian inference for PCFGs via Markov chain Monte Carlo. In *Proc. HLT-NAACL*, 139–146, Rochester, New York.

D. Klein, C. D. Manning, 2004. *Parsing and hypergraphs*, 351–372. Kluwer Academic Publishers, Norwell, MA, USA, 2004.

P. Koehn, F. J. Och, D. Marcu. 2003. Statistical phrase-based translation. In *Proc. HLT-NAACL*, 81–88, Edmonton, Canada.

P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, E. Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proc. ACL*, Prague.

D. Marcu, W. Wong. 2002. A phrase-based, joint probability model for statistical machine translation. In *Proc. EMNLP*, 133–139, Philadelphia.

R. Neal. 2003. Slice sampling. *Annals of Statistics*, 31:705–767.

K. Papineni, S. Roukos, T. Ward, W. Zhu. 2001. Bleu: a method for automatic evaluation of machine translation. Technical Report RC22176 (W0109-022), IBM Research Division, Thomas J. Watson Research Center, 2001.

J. Van Gael, Y. Saatci, Y. W. Teh, Z. Ghahramani. 2008. Beam sampling for the infinite hidden markov model. In *ICML*, 1088–1095, New York, NY, USA.

D. Wu. 1997. Stochastic inversion transduction grammars and bilingual parsing of parallel corpora. *Computational Linguistics*, 23(3):377–403.

---

[13]Our implementation had not completed a single sample after a week.