Exact Maximum Inference for the Fertility Hidden Markov Model

Chris Quirk
Microsoft Research
One Microsoft Way
Redmond, WA 98052, USA
chrisq@microsoft.com

Abstract

The notion of fertility in word alignment (the number of words emitted by a single state) is useful but difficult to model. Initial attempts at modeling fertility used heuristic search methods. Recent approaches instead use more principled approximate inference techniques such as Gibbs sampling for parameter estimation. Yet in practice we also need the single best alignment, which is difficult to find using Gibbs. Building on recent advances in dual decomposition, this paper introduces an exact algorithm for finding the single best alignment with a fertility HMM. Finding the best alignment appears important, as this model leads to a substantial improvement in alignment quality.

1 Introduction

Word-based translation models intended to model the translation process have found new uses identifying word correspondences in sentence pairs. These word alignments are a crucial training component in most machine translation systems. Furthermore, they are useful in other NLP applications, such as entailment identification.

The simplest models may use lexical information alone. The seminal Model 1 (Brown et al., 1993) has proved very powerful, performing nearly as well as more complicated models in some phrasal systems (Koehn et al., 2003). With minor improvements to initialization (Moore, 2004) (which may be important (Toutanova and Galley, 2011)), it can be quite competitive. Subsequent IBM models include more detailed information about context. Models 2 and 3 incorporate a positional model based on the absolute position of the word; Models 4 and 5 use a relative position model instead (an English word tends to align to a French word that is nearby the French word aligned to the previous English word). Models 3, 4, and 5 all incorporate a notion of "fertility": the number of French words that align to any English word.

Although these latter models covered a broad range of phenomena, estimation techniques and MAP inference were challenging. The authors originally recommended heuristic procedures based on local search for both. Such methods work reasonably well, but can be computationally inefficient and have few guarantees. Thus, many researchers have switched to the HMM model (Vogel et al., 1996) and variants with more parameters (He, 2007). This captures the positional information in the IBM models in a framework that admits exact parameter estimation inference, though the objective function is not concave: local maxima are a concern.

Modeling fertility is challenging in the HMM framework as it violates the Markov assumption. Where the HMM jump model considers only the prior state, fertility requires looking across the whole state space. Therefore, the standard forward-backward and Viterbi algorithms do not apply. Recent work (Zhao and Gildea, 2010) described an extension to the HMM with a fertility model, using MCMC techniques for parameter estimation. However, they do not have a efficient means of MAP inference, which is necessary in many applications such as machine translation.

This paper introduces a method for exact MAP inference with the fertility HMM using dual decomposition. The resulting model leads to substantial improvements in alignment quality.

Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, pages 7–11,
Sofia, Bulgaria, August 4-9 2013.

2 HMM alignment

Let us briefly review the HMM translation model as a starting point. We are given a sequence of English words $\mathbf{e} = e_1, \ldots, e_I$. This model produces distributions over French word sequences $\mathbf{f} = f_1, \ldots, f_J$ and word alignment vectors $\mathbf{a} = a_1, \ldots, a_J$, where $a_j \in [0..J]$ indicates the English word generating the jth French word, 0 representing a special NULL state to handle systematically unaligned words.

$$\Pr(\mathbf{f}, \mathbf{a}|\mathbf{e}) = p(J|I) \prod_{j=1}^{J} p(a_j|a_{j-1})\, p\big(f_j\big|e_{a_j}\big)$$

The generative story begins by predicting the number of words in the French sentence (hence the number of elements in the alignment vector). Then for each French word position, first the alignment variable (English word index used to generate the current French word) is selected based on only the prior alignment variable. Next the French word is predicted based on its aligned English word.

Following prior work (Zhao and Gildea, 2010), we augment the standard HMM with a fertility distribution.

$$\begin{aligned}\Pr(\mathbf{f}, \mathbf{a}|\mathbf{e}) =& p(J|I) \prod_{i=1}^{I} p(\phi_i|e_i) \\ & \prod_{j=1}^{J} p(a_j|a_{j-1})\, p\big(f_j\big|e_{a_j}\big)\end{aligned} \tag{1}$$

where $\phi_i = \sum_{j=1}^{J} \delta(i, a_j)$ indicates the number of times that state j is visited. This deficient model wastes some probability mass on inconsistent configurations where the number of times that a state i is visited does not match its fertility ϕ_i. Following in the footsteps of older, richer, and wiser colleagues (Brown et al., 1993),we forge ahead unconcerned by this complication.

2.1 Parameter estimation

Of greater concern is the exponential complexity of inference in this model. For the standard HMM, there is a dynamic programming algorithm to compute the posterior probability over word alignments $\Pr(\mathbf{a}|\mathbf{e}, \mathbf{f})$. These are the sufficient statistics gathered in the E step of EM.

The structure of the fertility model violates the Markov assumptions used in this dynamic programming method. However, we may empirically estimate the posterior distribution using Markov chain Monte Carlo methods such as Gibbs sampling (Zhao and Gildea, 2010). In this case, we make some initial estimate of the $\mathbf{a}$ vector, potentially randomly. We then repeatedly resample each element of that vector conditioned on all other positions according to the distribution $\Pr(a_j|\mathbf{a}_{-j}, \mathbf{e}, \mathbf{f})$. Given a complete assignment of the alignment for all words except the current, computing the complete probability including transition, emission, and jump, is straightforward. This estimate comes with a computational cost: we must cycle through all positions of the vector repeatedly to gather a good estimate. In practice, a small number of samples will suffice.

2.2 MAP inference with dual decomposition

Dual decomposition, also known as Lagrangian relaxation, is a method for solving complex combinatorial optimization problems (Rush and Collins, 2012). These complex problems are separated into distinct components with tractable MAP inference procedures. The subproblems are repeatedly solved with some communication over consistency until a consistent and globally optimal solution is found.

Here we are interested in the problem of finding the most likely alignment of a sentence pair $\mathbf{e}, \mathbf{f}$. Thus, we need to solve the combinatorial optimization problem $\arg\max_{\mathbf{a}} \Pr(\mathbf{f}, \mathbf{a}|\mathbf{e})$. Let us rewrite the objective function as follows:

$$\begin{aligned}h(a) = & \sum_{i=1}^{I} \left(\log p(\phi_i|e_i) + \sum_{j, a_j = i} \frac{\log p(f_j|e_i)}{2} \right) \\ & + \sum_{j=1}^{J} \left(\log p(a_j|a_{j-1}) + \frac{\log p\big(f_j\big|e_{a_j}\big)}{2} \right)\end{aligned}$$

Because $\mathbf{f}$ is fixed, the $p(J|I)$ term is constant and may be omitted. Note how we've split the optimization into two portions. The first captures fertility as well as some component of the translation distribution, and the second captures the jump distribution and the remainder of the translation distribution.

Our dual decomposition method follows this segmentation. Define $\mathbf{y_a}$ as $y_{\mathbf{a}}(i, j) = 1$ if $a_j = i$, and 0 otherwise. Let $\mathbf{z} \in \{0, 1\}^{I \times J}$ be a binary

$u^{(0)}(i,j) := 0 \ \ \forall i \in 1..I, j \in 1..J$
for $k = 1$ to K
  $\mathbf{a}^{(k)} := \arg\max_{\mathbf{a}} \left(f(\mathbf{a}) + \sum_{i,j} u^{(k-1)}(i,j) y_{\mathbf{a}}(i,j) \right)$
  $\mathbf{z}^{(k)} := \arg\max_{\mathbf{z}} \left(g(\mathbf{z}) - \sum_{i,j} u^{(k-1)}(i,j) z(i,j) \right)$
  if $\mathbf{y_a} = \mathbf{z}$
    return $\mathbf{a}^{(k)}$
  end if
  $u^{(k)}(i,j) := u^{(k)}(i,j) + \delta_k \left(y_{\mathbf{a}^{(k)}}(i,j) - z^{(k)}(i,j) \right)$
end for
return $\mathbf{a}^{(K)}$

Figure 1: The dual decomposition algorithm for the fertility HMM, where δ_k is the step size at the kth iteration for $1 \leq k \leq K$, and K is the max number of iterations.

matrix. Define the functions f and g as

$$f(\mathbf{a}) = \sum_{j=1}^{J} \left(\log p(a_j | a_{j-1}) + \frac{1}{2} \log p\big(f_j \big| e_{a_j}\big) \right)$$

$$g(\mathbf{z}) = \sum_{i=1}^{I} \Big(\log p(\phi(\mathbf{z}_i) | e_i) + \sum_{j=1}^{J} \frac{z(i,j)}{2} \log p(f_j | e_i) \Big)$$

Then we want to find

$$\arg\max_{\mathbf{a},\mathbf{z}} f(\mathbf{a}) + g(\mathbf{z})$$

subject to the constraints $y_{\mathbf{a}}(i,j) = z(i,j) \forall i,j$. Note how this recovers the original objective function when matching variables are found.

We use the dual decomposition algorithm from Rush and Collins (2012), reproduced here in Figure 1. Note how the langrangian adds one additional term word, scaled by a value indicating whether that word is aligned in the current position. Because it is only added for those words that are aligned, we can merge this with the $\log p\big(f_j \big| e_{a_j}\big)$ terms in both f and g. Therefore, we can solve $\arg\max_{\mathbf{a}} \left(f(\mathbf{a}) + \sum_{i,j} u^{(k-1)}(i,j) y_{\mathbf{a}}(i,j) \right)$ using the standard Viterbi algorithm.

The g function, on the other hand, does not have a commonly used decomposition structure. Luckily we can factor this maximization into pieces that allow for efficient computation. Note that g sums over arbitrary binary matrices. Unlike the HMM, where each French word must have exactly one English generator, this maximization allows each

$z(i,j) := 0 \ \forall (i,j) \in [1..I] \times [1..J]$
$v := 0$
for $i = 1$ **to** I
  for $j = 1$ **to** J
    $x(j) := (\log p(f_j | e_i), j)$
  end for
  sort x in descending order by first component
  $max := \log p(\phi = 0 | e_i), arg := 0, sum := 0$
  for $f = 1$ **to** J
    $sum := sum + x[f, 1]$
    if $sum + \log p(\phi = f | e_i) > max$
      $max := sum + \log p(\phi = f | e_i)$
      $arg := f$
    end if
  end for
  $v := v + max$
  for $f = 1$ **to** arg
    $z(i, x[f, 2]) := 1$
  end for
end for
return $\mathbf{z}, v$

Figure 2: Algorithm for finding the arg max and max of g, the fertility-related component of the dual decomposition objective.

French word to have zero or many generators. Because assignments that are in accordance between this model and the HMM will meet the HMM's constraints, the overall dual decomposition algorithm will return valid assignments, even though individual selections for this model may fail to meet the requirements.

As the scoring function g can be decomposed into a sum of scores for each row $\sum_i g_i$ (i.e., there are no interactions between distinct rows of the matrix) we can maximize each row independently:

$$\max_{\mathbf{z}} \sum_{i=1}^{I} g_i(\mathbf{z}_i) = \sum_{i=1}^{I} \max_{\mathbf{z}} g_i(\mathbf{z}_i)$$

Within each row, we seek the best of all 2^J possible configurations. These configurations may be grouped into equivalence classes based on the number of non-zero entries. In each class, the max assignment is the one using words with the highest log probabilities; the total score of this assignment is the sum those log probabilities and the log probability of that fertility. Sorting the scores of each cell in the row in descending order by log probability allows for linear time computation of the max for each row. The algorithm described in Figure 2 finds this maximal assignment in $O(IJ \log J)$ time, generally faster than the $O(I^2 J)$ time used by Viterbi.

We note in passing that this maximizer is picking from an unconstrained set of binary matri-

ces. Since each English word may generate as many French words as it likes, regardless of all other words in the sentence, the underlying matrix have many more or many fewer non-zero entries than there are French words. A straightforward extension to the algorithm of Figure 2 returns only $\mathbf{z}$ matrices with exactly J nonzero entries. Rather than maximizing each row totally independently, we keep track of the best configurations for each number of words generated in each row, and then pick the best combination that sums to J: another straightforward exercise in dynamic programming. This refinement does not change the correctness of the dual decomposition algorithm; rather it speeds the convergence.

3 Fertility distribution parameters

Original IBM models used a categorical distribution of fertility, one such distribution for each English word. This gives EM a great amount of freedom in parameter estimation, with no smoothing or parameter tying of even rare words. Prior work addressed this by using the single parameter Poisson distribution, forcing infrequent words to share a global parameter estimated from the fertility of all words in the corpus (Zhao and Gildea, 2010).

We explore instead a feature-rich approach to address this issue. Prior work has explored feature-rich approaches to modeling the translation distribution (Berg-Kirkpatrick et al., 2010); we use the same technique, but only for the fertility model. The fertility distribution is modeled as a log-linear distribution of F, a binary feature set: $p(\phi|e) \propto \exp(\theta \cdot F(e, \phi))$. We include a simple set of features:

- A binary indicator for each fertility ϕ. This feature is present for all words, acting as smoothing.
- A binary indicator for each word id and fertility, if the word occurs more than 10 times.
- A binary indicator for each word length (in letters) and fertility.
- A binary indicator for each four letter word prefix and fertility.

Together these produce a distribution that can learn a reasonable distribution not only for common words, but also for rare words. Including word length information aids in for languages with compounding: long words in one language may correspond to multiple words in the other.

Algorithm	AER (G→E)	AER (E→G)
HMM	24.0	21.8
FHMM Viterbi	19.7	19.6
FHMM Dual-dec	18.0	17.4

Table 1: Experimental results over the 120 evaluation sentences. Alignment error rates in both directions are provided here.

4 Evaluation

We explore the impact of this improved MAP inference procedure on a task in German-English word alignment. For training data we use the news commentary data from the WMT 2012 translation task.[1] 120 of the training sentences were manually annotated with word alignments.

The results in Table 1 compare several different algorithms on this same data. The first line is a baseline HMM using exact posterior computation and inference with the standard dynamic programming algorithms. The next line shows the fertility HMM with approximate posterior computation from Gibbs sampling but with final alignment selected by the Viterbi algorithm. Clearly fertility modeling is improving alignment quality. The prior work compared Viterbi with a form of local search (sampling repeatedly and keeping the max), finding little difference between the two (Zhao and Gildea, 2010). Here, however, the difference between a dual decomposition and Viterbi is significant: their results were likely due to search error.

5 Conclusions and future work

We have introduced a dual decomposition approach to alignment inference that substantially reduces alignment error. Unfortunately the algorithm is rather slow to converge: after 40 iterations of the dual decomposition, still only 55 percent of the test sentences have converged. We are exploring improvements to the simple sub-gradient method applied here in hopes of finding faster convergence, fast enough to make this algorithm practical. Alternate parameter estimation techniques appear promising given the improvements of dual decomposition over sampling. Once the performance issues of this algorithm are improved, exploring hard EM or some variant thereof might lead to more substantial improvements.

[1] www.statmt.org/wmt12/translation-task.html

References

Taylor Berg-Kirkpatrick, Alexandre Bouchard-Côté, John DeNero, and Dan Klein. 2010. Painless unsupervised learning with features. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 582–590, Los Angeles, California, June. Association for Computational Linguistics.

Peter F. Brown, Vincent J. Della Pietra, Stephen A. Della Pietra, and Robert L. Mercer. 1993. The mathematics of statistical machine translation: parameter estimation. *Computational Linguistics*, 19(2):263–311.

Xiaodong He. 2007. Using word-dependent transition models in HMM-based word alignment for statistical machine translation. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 80–87, Prague, Czech Republic, June. Association for Computational Linguistics.

Philipp Koehn, Franz J. Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*.

Robert C. Moore. 2004. Improving ibm word alignment model 1. In *Proceedings of the 42nd Meeting of the Association for Computational Linguistics (ACL'04), Main Volume*, pages 518–525, Barcelona, Spain, July.

Alexander M Rush and Michael Collins. 2012. A tutorial on dual decomposition and lagrangian relaxation for inference in natural language processing. *Journal of Artificial Intelligence Research*, 45:305–362.

Kristina Toutanova and Michel Galley. 2011. Why initialization matters for ibm model 1: Multiple optima and non-strict convexity. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 461–466, Portland, Oregon, USA, June. Association for Computational Linguistics.

Stephan Vogel, Hermann Ney, and Christoph Tillmann. 1996. HMM-based word alignment in statistical translation. In *COLING*.

Shaojun Zhao and Daniel Gildea. 2010. A fast fertility hidden markov model for word alignment using MCMC. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 596–605, Cambridge, MA, October. Association for Computational Linguistics.