Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence

A Family of Latent Variable Convex Relaxations for IBM Model 2

Andrei Simion
Columbia University
IEOR Department
New York, NY, 10027
aas2148@columbia.edu

Michael Collins
Columbia University
Computer Science
New York, NY, 10027
mc3354@columbia.edu

Clifford Stein
Columbia University
IEOR Department
New York, NY, 10027
cs2035@columbia.edu

Abstract

Recently, a new convex formulation of IBM Model 2 was introduced. In this paper we develop the theory further and introduce a class of convex relaxations for latent variable models which include IBM Model 2. When applied to IBM Model 2, our relaxation class subsumes the previous relaxation as a special case. As proof of concept, we study a new relaxation of IBM Model 2 which is simpler than the previous algorithm: the new relaxation relies on the use of nothing more than a multinomial EM algorithm, does not require the tuning of a learning rate, and has some favorable comparisons to IBM Model 2 in terms of F-Measure. The ideas presented could be applied to a wide range of NLP and machine learning problems.

Introduction

The IBM translation models (Brown et al. 1993) were the first Statistical Machine Translation (SMT) systems; their primary use in the current SMT pipeline is to seed more sophisticated models which need alignment tableaus to start their optimization procedure. Although there are several IBM Models, only IBM Model 1 can be formulated as a convex optimization problem. Other IBM Models have non-concave objective functions with multiple local optima, and solving a non-convex problem to optimality is typically a computationally intractable task. Recently, using a linearization technique, a convex relaxation of IBM Model 2 was proposed (Simion, Collins, and Stein 2013; 2014). In this work we generalize the methods introduced in (Simion, Collins, and Stein 2013) to yield a richer set of relaxation techniques. Our algorithms have comparable performance to previous work and have the potential for more applications.

We make the following contributions in this paper:

- We introduce a convexification method that may be applicable to a wide range of probabilistic models in NLP and machine learning. In particular, since the likelihood we are optimizing and the metric we are testing against are often not the same (e.g. for alignment tasks we want to maximize F-Measure, but F-Measure is not directly in the likelihood function), different relaxations should potentially be considered for different tasks. The crux of our approach relies on approximating the product function $\prod_{i=1}^{n} x_i$ with a concave function and as a supplement we present some theoretical analysis characterizing concave functions h that approximate this function.
- As a specific application, we introduce a generalized family of convex relaxations for IBM Model 2.[1] Essentially, the relaxation is derived by replacing the product $t(f_j|e_i) \times d(i|j)$ with $h(t(f_j|e_i), d(i|j))$ where $h(x_1, x_2)$ is a concave upper envelope for $x_1 x_2$. We show how our results encompass the work of (Simion, Collins, and Stein 2013) as a special case.
- We detail an optimization algorithm for a particularly simple relaxation of IBM Model 2. Unlike the previous work in (Simion, Collins, and Stein 2013) which relied on a exponentiated subgradient (EG) optimization method and required the tuning of a learning rate, this relaxation can be approached in a much simpler fashion and can be optimized by an EM algorithm that is very similar to the one used for IBM Models 1 and 2. We show that our method achieves a performance very similar to that of IBM Model 2 seeded with IBM 1.

Notation. Throughout this paper, for any positive integer N, we use $[N]$ to denote $\{1 \dots N\}$ and $[N]_0$ to denote $\{0 \dots N\}$. We denote by $\mathbb{R}^n_+$ and $\mathbb{R}^n_{++}$ the set of nonnegative and strictly positive n dimensional vectors, respectively. We denote by $[0,1]^n$ the $n-$dimensional unit cube.

Related Work

The IBM Models were introduced in (Brown et al. 1993) and since then there has been quite a bit of research on them and their variants (e.g. (Vogel, Ney, and Tillman 1996; Och and Ney 2003; Toutanova and Galley 2011; Moore 2004; Liang, Taskar, and Klein 2006)). For example, (Dyer, Chahuneau, and Smith 2013) recently proposed a (non-convex) variant of IBM Model 2 that focuses on generating "diagonal" alignments, allows for very fast parameter optimization, and empirically was shown to be superior to IBM Model 4 in generating quality translations. Moreover,

[1] We note that there are negative results which show that certain latent variable problems will have convex relaxations having the uniform solution as optimal (Guo and Schuurmans 2007). However, for IBM Model 2, the data breaks such symmetries, so any relaxation will be nontrivial.

(Simion, Collins, and Stein 2013) introduced the first convex relaxation of a model beyond IBM Model 1, design an algorithm for its optimization, and showed that it gives the same level of performance as IBM Model 2 (Simion, Collins, and Stein 2014).

Background on Alignment Models

In this section we give a brief survey of IBM Models 1 and 2 and the convex relaxations of Model 2 , I2CR-1 and I2CR-2 (Simion, Collins, and Stein 2013). The standard approach in training parameters for IBM Models 1 and 2 is EM, whereas for models I2CR-1 and I2CR-2 an EG algorithm was developed.

We assume that our set of training examples is $(e^{(k)}, f^{(k)})$ for $k = 1 \dots n$, where $e^{(k)}$ is the k'th English sentence and $f^{(k)}$ is the k'th French sentence. The k'th English sentence is a sequence of words $e_1^{(k)} \dots e_{l_k}^{(k)}$ where l_k is the length of the k'th English sentence, and each $e_i^{(k)} \in E$; similarly the k'th French sentence is a sequence $f_1^{(k)} \dots f_{m_k}^{(k)}$ where m_k is the length of the k'th French sentence, and each $f_j^{(k)} \in F$. We define $e_0^{(k)}$ for $k = 1 \dots n$ to be a special NULL word (note that E contains the NULL word). For each English word e, we will assume that $D(e)$ is a dictionary specifying the set of possible French words that can be translations of e. Finally, we define $L = \max_{k=1}^{n} l_k$ and $M = \max_{k=1}^{n} m_k$.

More details on the convex and non-convex IBM Models 1 and 2 optimization problems can be found in (Simion, Collins, and Stein 2013). The IBM Model 2 optimization problem is shown in Figure 1. We note that for this model the constraints are convex but the objective is not concave, causing the problem to be non-convex. IBM Model 1 has the same lexical parameters t as IBM Model 2 but the distortion parameters d are set to be uniform throughout, yielding a model with constraints given by Eq. 1 - 2 and concave objective given by

$$\frac{1}{n} \sum_{k=1}^{n} \sum_{j=1}^{m_k} \log \sum_{i=0}^{l_k} t(f_j^{(k)} | e_i^{(k)}) \ .$$

Since the objective function for IBM Model 1 is concave, the EM algorithm will converge to a global maximum. A common heuristic is to initialize the $t(f|e)$ parameters in EM optimization of IBM Model 2 using the output from the weaker IBM Model 1. Although this initialization heuristic has been shown to be effective in practice (Och and Ney 2003), there are no formal guarantees on its performance and there are non-convex IBM Model 2 variants which empirically do not work well with this type of initialization (Dyer, Chahuneau, and Smith 2013).

The I2CR-2 optimization problem is a convex relaxation of IBM Model 2. The constraints for I2CR-2 are those of IBM Model 2 while its objective is

$$\frac{1}{2n} \sum_{k=1}^{n} \sum_{j=1}^{m_k} \log' \sum_{i=0}^{l_k} t(f_j^{(k)} | e_i^{(k)})$$
$$+ \quad \frac{1}{2n} \sum_{k=1}^{n} \sum_{j=1}^{m_k} \log' \sum_{i=0}^{l_k} \min\{t(f_j^{(k)} | e_i^{(k)}), d(i|j)\} \ ,$$

Input: Define E, F, L, M, $(e^{(k)}, f^{(k)}, l_k, m_k)$ for $k = 1 \dots n$, $D(e)$ for $e \in E$.
Parameters:
• A parameter $t(f|e)$ for each $e \in E, f \in D(e)$.
• A parameter $d(i|j)$ for each $i \in [L]_0, j \in [M]$.
Constraints:

$$\forall e \in E, f \in D(e), \quad t(f|e) \geq 0 \tag{1}$$
$$\forall e \in E, \quad \sum_{f \in D(e)} t(f|e) = 1 \tag{2}$$
$$\forall i \in [L]_0, j \in [M], \quad d(i|j) \geq 0 \tag{3}$$
$$\forall j \in [M], \quad \sum_{i \in [L]_0} d(i|j) = 1 \tag{4}$$

Objective: Maximize

$$\frac{1}{n} \sum_{k=1}^{n} \sum_{j=1}^{m_k} \log \sum_{i=0}^{l_k} t(f_j^{(k)} | e_i^{(k)}) d(i|j) \tag{5}$$

with respect to the $t(f|e)$ and $d(i|j)$ parameters.

Figure 1: The IBM Model 2 Optimization Problem.

where $\log'(z) = \log(z + \lambda)$ and $\lambda = .001$ is a small positive smoothing constant. The I2CR-2 objective is a convex combination of the concave IBM Model 1 objective and a direct (concave) relaxation of the IBM2 Model 2 objective, and hence is itself concave. The direct relaxation of IBM Model 2 is known as I2CR-1; the only difference between I2CR-1 and I2CR-2 is that I2CR-1 does not have an IBM Model 1 objective appended to its objective.

A Class of Concave Functions based on the Generalized Weighted Mean

In (Simion, Collins, and Stein 2013), model I2CR-2 is studied and, at a high level, the key component is to replace the non-concave function $f(x) = \prod_{i=1}^{n} x_i$ by the concave function $h(x) = \min_{i=1}^{n} x_i$. This is only one possible convexification; we now explore a much larger set of ways to convexify a product.

Definition 1. *Let $(\alpha_1, \dots, \alpha_n) \in (0,1)^n$ be such that $\sum_{i=1}^{n} \alpha_i = 1$. For $p \neq 0$ denote $f_p : \mathbb{R}_{++}^n \to \mathbb{R}_+$ given by*

$$f_p(x_1, \dots, x_n) = \left(\sum_{i=1}^{n} \alpha_i x_i^p \right)^{1/p} \tag{6}$$

as the generalized weighted mean function. For $p = 0$ denote $f_0 : \mathbb{R}_{++}^n \to \mathbb{R}_+$ given by

$$f_0(x_1, \dots, x_n) = \prod_{i=1}^{n} x_i^{\alpha_i} \tag{7}$$

as the generalized weighted geometric mean function.

Although the above definition restricts the domain to $\mathbb{R}_{++}^n$, we extend the domain of f_p to $\mathbb{R}^n$ by setting $f_p(x)$

to $-\infty$ for any $x \notin \mathbb{R}^n_{++}$. With this definition, f_p is defined everywhere and is a *concave* function (Boyd and Vandenberghe 2004). The results we need on the generalized weighted mean are detailed next along with some new material that serves as supplement. Theorems 1-2 and Lemma 1 are implicit in several sources in the literature ((Boyd and Vandenberghe 2004; Zalinescu 2002; Bullen, Mitrinovic, and Vasic 1987)).

Theorem 1. *If $p \leq 1$ then any f_p within the class of functions in Definition 1 is concave.*

Proof. See Appendix C. □

Using Theorem 1 and extending f_p to $\mathbb{R}^n$, the generalized mean function thus gives us a family of concave functions defined everywhere. Interestingly, we note that the extremes

$$\lim_{p\to 0} f_p(x) = \prod_{i=1}^{n} x_i^{\alpha_i} = f_0(x)$$

and

$$\lim_{p\to -\infty} f_p(x) = \min\{x_1, \ldots, x_n\} = f_{-\infty}(x)\,,$$

are both concave and belong to this family.

Lemma 1. *Let $f_p(x)$ be defined as in Definition 1. We have*

$$\lim_{p\to 0} f_p(x) = \prod_{i=1}^{n} x_i^{\alpha_i}\,,$$

and

$$\lim_{p\to -\infty} f_p(x) = \min\{x_1, \ldots, x_n\}\,.$$

Proof. See Appendix C. □

Lemma 1 implies that $f_p(x)$ can be identified for any $p \leq 1$ and all x as being concave. Moreover, for $x \in [0,1]^n$, $f_p(x)$ provides a monotonic concave upper envelope for $\prod_{i=1}^n x_i$.

Theorem 2. *Let $f_p(x)$ be defined as in Definition 1. For $x \in [0,1]^n$ the generalized weighted mean function $f_p(x)$ provides a monotonic concave envelope for $\prod_{i=1}^n x_i$. In particular, we have*

$$\prod_{i=1}^{n} x_i \leq f_p(x) \leq f_q(x)$$

for any $p \leq q \leq 1$.

Proof. See Appendix C. □

We next show that $f_{-\infty}(x) = \min_{i=1}^n x_i$ is a special function when used to bound $\prod_{i=1}^n x_i$ above by a positive-valued concave envelope. Specifically, we have that $\min_{i=1}^n x_i$ is the tightest such upper bound, *regardless* of the class of functions we consider.

Theorem 3. *Consider any concave function $h : \mathbb{R}^n_{++} \to \mathbb{R}_+$ such that*

$$\prod_{i=1}^{n} x_i \leq h(x)$$

for all $x \in [0,1]^n$. Then

$$\min_{i=1}^{n} x_i \leq h(x)$$

for all $x \in [0,1]^n$.

Proof. Due to space limitations, we consider only the case $n = 2$. For a full proof, see Appendix D. Note that for any point of the type $(x_1, 1)$, $(x_1, 0)$, $(1, x_2)$, or $(0, x_2)$ the result follows easily, so without loss of generality consider $x = (x_1, x_2) \in (0,1)^2$ with $x_1 \leq x_2$ and suppose by way of contradiction that $h(x_1, x_2) < x_1$ and note that we have

$$x_1 \leq x_2 h\left(\frac{x_1}{x_2}, 1\right)$$

and

$$0 \leq (1 - x_2)h(0,0)$$

by the positivity of h and the fact that h bounds the product of its arguments. Adding the above and using Jensen's inequality we then have

$$x_1 \leq x_2 h\left(\frac{x_1}{x_2}, 1\right) + (1 - x_2)h(0,0) \leq h(x_1, x_2) < x_1\,.$$

The above result yields a contradiction, and we now have that $\min\{x_1, x_2\}$ is the tightest positive-valued upper bound on $x_1 x_2$. □

Although several upper bounds for $\prod_{i=1}^n x_i$ with $x \in [0,1]^n$ are detailed above, we note that bounding $\prod_{i=1}^n x_i$ *below* by a nontrivial positive-valued concave function is not possible, if $n \geq 2$.

Theorem 4. *Let $n \geq 2$ and $h : \mathbb{R}^n_{++} \to \mathbb{R}_+$ be a concave function such that*

$$h(x) \leq \prod_{i=1}^{n} x_i$$

for all $x \in [0,1]^n$. Then $h(x)$ is identically equal to zero.

Proof. If x has a component which is zero then $h(x) \leq 0$ and hence $h(x) = 0$ since $h(x) \geq 0$. Choosing $\theta \in (0,1)$ and $x \in (0,1]^n$ yields that

$$\theta h(x) + (1 - \theta)h(0) \leq h(\theta x) \leq \theta^n \prod_{i=1}^{n} x_i\,,$$

and we note that the left hand side above is equal to $\theta h(x)$. Dividing both sides by θ we next have

$$h(x) \leq \theta^{n-1} \prod_{i=1}^{n} x_i\,.$$

Letting $\theta \to 0$ in the last equation we get $h(x) \leq 0$. Since we also have $h(x) \geq 0$, we now have $h(x) = 0$ for any $x \in [0,1]^n$, as needed. □

The main takeaway of the above is that positive valued concave envelopes for $\prod_{i=1}^n x_i$ are limited to *upper* envelopes such as those provided by f_p in Definition 1.

Input: Define E, F, L, M, $(e^{(k)}, f^{(k)}, l_k, m_k)$ for $k = 1 \ldots n$, $D(e)$ for $e \in E$. A positive-valued concave function $h : \mathbb{R}^2_{++} \rightarrow \mathbb{R}_+$ such that

$$x_1 x_2 \leq h(x_1, x_2) \ ,$$

$\forall\ (x_1, x_2) \in [0, 1]^2$.
Parameters: Same as IBM Model 2.
Constraints: Same as IBM Model 2.
Objective: Maximize

$$\frac{1}{n} \sum_{k=1}^{n} \sum_{j=1}^{m_k} \log \sum_{i=0}^{l_k} h(t(f_j^{(k)}|e_i^{(k)}), d(i|j)) \qquad (8)$$

with respect to the $t(f|e)$ and $d(i|j)$ parameters.

Figure 2: The I2CR (IBM 2 Convex Relaxation) Problem. For any function h that is concave, the resulting optimization problem is a convex problem. I2CR-1 results from using $h(x_1, x_2) = f_{-\infty}(x_1, x_2) = \min\{x_1, x_2\}$ in the above while I2CR-3 arises from using $h(x_1, x_2) = f_0(x_1, x_2) = x_1^{\beta} x_2^{1-\beta}$ with $\beta \in [0, 1]$.

A Family of Convex IBM Model 2 Alternatives

(Simion, Collins, and Stein 2013) call their relaxations of IBM Model 2 I2CR-1 and I2CR-2. Since our methods subsume theirs, we use I2CR to denote the general optimization problem class that arises by using a special concave h instead of $x_1 x_2$ in IBM Model 2; see Figure 2. I2CR-3 and I2CR-4 are based on the particular concave function $f_0(x_1, x_2) = x_1^{\beta} x_2^{1-\beta}$ (with $\beta \in [0, 1]$) from Definition 1.[2] Although the focus is on the special case I2CR-3, the convexity proof we present is general and will imply that I2CR is a family of convex optimization problems. For a fixed h, any new relaxation of IBM Model 2 could then be optimized using a mini-batch EG method as discussed in (Simion, Collins, and Stein 2013). Because of the convexity of the problems that result, the optimization methods above are guaranteed to converge to a global solution.

The I2CR-3 Problem

The I2CR-3 problem is a special case of I2CR shown in Figure 2 using $h = f_0$. The key difference between this model and IBM Model 2 is that in the objective of IBM Model 2 we have replaced terms of the type $t(f_j|e_i) \times d(i|j)$ by $t^{\beta}(f_j|e_i) \times d^{1-\beta}(i|j)$, where $\beta \in [0, 1]$. We now state the main result needed to show that the objective of I2CR-3 is concave:

Lemma 2. *Let $\mathcal{T}$ be a subset of $[n]$ and consider $h : \mathbb{R}^n_{++} \rightarrow \mathbb{R}_+$ given by*

$$h(x_1, \ldots, x_n) = \prod_{i \in \mathcal{T}} x_i^{\alpha_i} \ ,$$

where $\alpha_i \in (0, 1)\ \forall i \in \mathcal{T}$ and $\sum_{i \in \mathcal{T}} \alpha_i = 1$. Then h is concave.

Proof. Let $g : \mathbb{R}^{|\mathcal{T}|}_{++} \rightarrow \mathbb{R}_+$ be given by

$$g(x_1, \ldots, x_{|\mathcal{T}|}) = \prod_{i=1}^{|\mathcal{T}|} x_i^{\alpha_i}$$

and note that g is concave by Theorem 4.1. Next we note that $h(x) = g(Ax + b)$ where $b = 0$ and $A \in \mathbb{R}^{n \times |\mathcal{T}|}$ is a suitably chosen matrix which projects down from dimension n to $|\mathcal{T}|$. By the composition rule of a concave function with a linear transformation, h is a concave function (Boyd and Vandenberghe 2004). □

Using the above Lemma, we can prove that functions such as

$$h(x_1, x_2, x_3) = \sqrt{x_1 x_2} + \sqrt{x_2 x_3}$$

are concave since they are the sum of two concave functions. We use this observation in the following theorem.

Theorem 5. *The objective of I2CR-3 is concave.*

Proof. Fix a specific training pair index k and target word position j within the objective of I2CR-3 given by Eq. 8. We first note that the $\log$ is an increasing concave function (we define $\log(x)$ to be $-\infty$ if $x \leq 0$). Using Lemma 2 repeatedly the sum inside the logarithm in the objective of I2CR-3 (Eq. 8) is a sum of concave functions, and is hence itself concave. It is a well known rule that composing a concave increasing function (such as the logarithm) and a concave function yield a concave function (Boyd and Vandenberghe 2004). Hence, for a fixed k and j, the objective of I2CR-3 is concave. Since the objective in Eq. 8 is a sum of concave functions, the result now follows. □

Theorem 5 implies that I2CR-3 is a convex optimization problem since its objective is concave and the constraints form a polytope. In fact, note that an analogous Lemma 2 would hold for any concave function h. With this observation we now have a recipe that can be carried out for any positive-valued concave function h thus yielding our main result: I2CR is a family of convex relaxations for IBM Model 2. In particular, this recipe is more general than the linearization technique in (Simion, Collins, and Stein 2013) and can be carried out for any concave function h in Figure 2. By using $h = f_{-\infty}$ and applying Theorem 2 we have the tightest such relaxation: I2CR-1 (Simion, Collins, and Stein 2013). Interestingly, we will see later that a tighter relaxation does not necessarily give better alignment quality.

As a final comment, we remark that the new relaxation is not *strictly* convex for *all* datasets. However, similar to IBM Model 2, our sense is that the symmetries in the data that would result in non-strict convexity will be rare in real datasets — much more rare than the case of IBM Model 1, for which it is well known that the objective is not strictly

[2] Note that there is some similarity of the resulting objective function to methods that use deterministic annealing for EM ((Smith and Eisner 2004); (Rose 1998)) In annealing approaches the objective would be $(x_1 x_2)^{\beta}$ where β is initially close to 0, and is then progressively increased to a value of 1. This prior work does not make the connection to convex objectives when $\beta = 1/2$, and unlike our approach varies β between 0 and 1 within their algorithm.

convex for real-world datasets (Toutanova and Galley 2011). We leave further study of this to future work.[3]

The I2CR-4 Problem

Our initial experiments with I2CR-3 lead to better performance than IBM Model 1, but did not yield results as good as those of Model 2. (Simion, Collins, and Stein 2013) obtained better performance by appending an IBM Model 1 objective to the original convex relaxation I2CR-1 that they derived, and we felt that this might work for our model as well. To this end we call our new model I2CR-4 and note that its objective is the sum of one likelihood which places all its importance on the lexical terms (IBM1) and another (I2CR-3) that distributes weight on the lexical and distortion term via the geometric weighted mean:

$$\frac{1}{2n}\sum_{k=1}^{n}\sum_{j=1}^{m_k}\log\sum_{i=0}^{l_k}t(f_j^{(k)}|e_i^{(k)}) + \frac{1}{2n}\sum_{k=1}^{n}\sum_{j=1}^{m_k}\log\sum_{i=0}^{l_k}t^{\beta}(f_j^{(k)}|e_i^{(k)})d^{1-\beta}(i|j)\ ,$$

This new model is still a convex optimization problem since its objective is concave (the sum of two concave functions is concave).

An EM Algorithm for I2CR-4

We describe an EM algorithm for optimizing the I2CR-4 problem in Figure 3, and note that the memory and time requirements are the same as those of IBM Model 2's EM algorithm. We find it appealing to introduce a relaxation based on the weighted geometric mean specifically because a simple EM algorithm can be derived. For a proof and further discussion of the algorithm, see Appendix E.

Decoding with I2CR-3 and I2CR-4

To obtain the highest probability alignment of a pair $(e^{(k)}, f^{(k)})$ using an IBM Model we need to find the $a^{(k)} = (a_1^{(k)}, \ldots, a_{m_k}^{(k)})$ which yields the highest probability $p(f^{(k)}, a^{(k)}|e^{(k)})$. There are various ways to use the estimated parameters from the IBM Models in decoding. For one, we could find the optimal alignment for I2CR-4 using IBM Model 2's rule (this is the optimal rule for I2CR-3 as well). On the other hand, using the same methods as presented in (Simion, Collins, and Stein 2014) we can find the optimal vector $a^{(k)}$ by splitting the maximization over the components of $a^{(k)}$ and focusing on finding $a_j^{(k)}$ given by

$$a_j^{(k)} = \text{argmax}_a\{t^{1+\beta}(f_j^{(k)}|e_a^{(k)})d^{1-\beta}(a|j)\}\ .$$

Finally, we also decode using IBM Model 1's rule. Since the EM updates for IBM Model 1 do not take position at all into account, any reasonable convex relaxation of IBM Model 2 should always beat IBM Model 1 in lexical parameter quality.

[3]Noting that for $(\alpha_1, \alpha_2) \in (0,1)^2$ with $\alpha_1 + \alpha_2 < 1$ $f_0(x_1, x_2) = x_1^{\alpha_1}x_2^{\alpha_2}$ is strictly concave ((Zalinescu 2002)), there is an easy remedy to guarantee strict convexity. In particular, using a degenerate f_0 we get the same EM algorithm as in Figure 3 (change $(\beta, 1-\beta)$ to (α_1, α_2)), but now have a strictly convex relaxation. Besides this, we could also use an l_2 regularizer. For more, see Appendix F.

Input: Define E, F, L, M, $(e^{(k)}, f^{(k)}, l_k, m_k)$ for $k = 1 \ldots n$, $D(e)$ for $e \in E$. An integer T specifying the number of passes over the data. A weighting parameter $\beta \in [0, 1]$.

Parameters:

- A parameter $t(f|e)$ for each $e \in E, f \in D(e)$.
- A parameter $d(i|j)$ for each $i, \in [L]_0, j \in [M]$.

Initialization:

- $\forall e \in E,\ f \in D(e)$, set $t(f|e) = 1/|D(e)|$.
- $\forall i \in [L]_0, j \in [M]$, set $d(i|j) = 1/(L+1)$.

EM Algorithm:

for all $t = 1 \ldots T$ **do**
  $\forall e \in E,\ f \in D(e),\ count(f, e) = 0$
  $\forall e \in E,\ count(e) = 0$
  $\forall i \in [L]_0, j \in [M], count(i, j) = 0$
  $\forall j \in [M], count(j) = 0$
  EM Algorithm: Expectation
  for all $k = 1 \ldots n$ **do**
    for all $j = 1 \ldots m_k$ **do**
      $\delta_1[i] = \delta_2[i] = 0\ \forall i \in [l_k]_0$
      $\Delta_1 = \Delta_2 = 0$
      for all $i = 0 \ldots l_k$ **do**
        $\delta_1[i] = t(f_j^{(k)}|e_i^{(k)})$
        $\delta_2[i] = t^{\beta}(f_j^{(k)}|e_i^{(k)})d^{1-\beta}(i|j)$
        Δ_1 += $\delta_1[i]$
        Δ_2 += $\delta_2[i]$
      for all $i = 0 \ldots l_k$ **do**
        $\delta_1[i] = \frac{\delta_1[i]}{\Delta_1}$
        $\delta_2[i] = \frac{\delta_2[i]}{\Delta_2}$
        $count(f_j^{(k)}, e_i^{(k)})$ += $\delta_1[i] + \beta\delta_2[i]$
        $count(e_i^{(k)})$ += $\delta_1[i] + \beta\delta_2[i]$
        $count(i, j)$ += $(1-\beta)\delta_2[i]$
        $count(j)$ += $(1-\beta)\delta_2[i]$
  EM Algorithm: Maximization
  for all $e \in E$ **do**
    for all $f \in D(e)$ **do**
      $t(f|e) = \frac{count(e,f)}{count(e)}$
  for all $\forall i \in [L]_0, j \in [M]$, **do**
    $d(i|j) = \frac{count(i,j)}{count(j)}$

Output: t, d parameters.

Figure 3: Pseudocode for T iterations of the EM Algorithm for the I2CR-4 problem.

Experiments

In this section we describe experiments using the I2CR-3 and I2CR-4 optimization problems combined with the EM algorithm for these problems. For our experiments we only used $\beta = \frac{1}{2}$, but note that β can be cross-validated for optimal performance.

Data Sets

For our alignment experiments, we used a subset of the Canadian Hansards bilingual corpus with 247,878 English-French sentence pairs as training data, 37 sentences of development data, and 447 sentences of test data (Michalcea and Pederson 2003). As a second corpus, we considered a

training set of 48,706 Romanian-English sentence-pairs, a development set of 17 sentence pairs, and a test set of 248 sentence pairs (Michalcea and Pederson 2003). For our SMT experiments, we choose a subset of the English-German Europarl bilingual corpus, using 274,670 sentences for training, 1,806 for development, and 1,840 for test.

Methodology

For each of the models we follow convention in applying the following methodology: first, we estimate the t and d parameters using models in both source-target and target-source directions; second, we find the most likely alignment for each development or test data sentence in each direction; third, we take the intersection of the two alignments as the final output from the model.

For our experiments, we report results in both AER (lower is better) and F-Measure (higher is better) (Och and Ney 2003). There is evidence (Fraser and Marcu 2007) that F-Measure is better correlated with translation quality when the alignments are used in a full system.

Model	IBM2	IBM2	I2CR-3	I2CR-3	I2CR-4	I2CR-4	I2CR-4
Decoding Rule	t	$t \times d$	t	$t \times d$	t	$t \times d$	$t \times \sqrt{t \times d}$
Iteration				AER			
0	0.2141	0.2141	0.9273	0.9273	0.9273	0.9273	0.9273
1	0.2128	0.1609	0.3697	0.3786	0.3669	0.3790	0.3569
2	0.2013	0.1531	0.2614	0.2235	0.2408	0.2090	0.2038
3	0.1983	0.1477	0.2333	0.1879	0.2209	0.1769	0.1754
4	0.1950	0.1458	0.2116	0.1783	0.2153	0.1668	0.1646
5	0.1941	0.1455	0.2088	0.1753	0.2067	0.1632	0.1592
6	0.1926	**0.1436**	0.2063	0.1739	0.2058	0.1600	0.1559
7	0.1912	**0.1436**	0.2048	0.1726	0.2046	0.1566	0.1551
8	0.1904	0.1449	0.2044	0.1730	0.2044	0.1549	0.1540
9	0.1907	0.1454	0.2041	0.1727	0.2047	0.1527	0.1534
10	0.1913	0.1451	0.2042	0.1721	0.2045	0.1524	0.1524
11	0.1911	0.1452	0.2042	**0.1718**	0.2039	0.1515	0.1520
12	0.1901	0.1454	**0.2040**	0.1722	0.2035	0.1513	0.1514
13	0.1899	0.1462	0.2041	0.1721	0.2032	0.1510	0.1511
14	**0.1898**	0.1471	0.2041	0.1724	0.2032	0.1509	0.1508
15	0.1900	0.1474	0.2041	0.1727	**0.2031**	**0.1505**	**0.1505**
Iteration				F-Measure			
0	0.7043	0.7043	0.0482	0.0482	0.0482	0.0482	0.0482
1	0.7049	0.7424	0.5610	0.5446	0.5664	0.5455	0.5712
2	0.7127	0.7468	0.6603	0.6910	0.6818	0.7059	0.7149
3	0.7116	0.7489	0.6838	0.7201	0.6977	0.7302	0.7385
4	**0.7130**	**0.7501**	0.7036	0.7255	0.7020	0.7369	0.7471
5	0.7124	0.7495	0.7060	0.7252	0.7102	0.7394	0.7515
6	0.7121	0.7501	0.7079	0.7257	0.7103	0.7411	0.7531
7	0.7132	0.7493	0.7084	0.7260	0.7111	0.7443	0.7531
8	0.7132	0.7480	**0.7085**	0.7252	0.7113	0.7457	0.7541
9	0.7127	0.7473	0.7084	0.7254	0.7115	0.7476	0.7547
10	0.7116	0.7474	0.7082	**0.7261**	0.7113	0.7482	0.7559
11	0.7113	0.7466	0.7080	**0.7261**	0.7117	0.7493	0.7563
12	0.7123	0.7463	0.7081	0.7256	0.7118	0.7496	0.7568
13	0.7119	0.7460	0.7081	0.7257	0.7121	0.7497	0.7571
14	0.7122	0.7451	0.7081	0.7253	0.7121	0.7497	0.7575
15	0.7122	0.7447	0.7081	0.7250	**0.7122**	**0.7501**	**0.7577**

Table 1: Intersected results on the English-French data for IBM Model 2, I2CR-3, and I2CR-4 trained for 15 EM using either the IBM1 (t), IBM2 ($t \times d$), or I2CR-4 ($t \times \sqrt{t \times d}$) decoding.

In training IBM Model 2 we first train IBM Model 1 for 5 iterations to initialize the t parameters, then train IBM Model 2 for a further 15 iterations (Och and Ney 2003). For the I2CR models, we use 15 iterations over the training data and seed all parameters to uniform probabilities. Since the development data we use is rather small, for all models considered we report F-Measure and AER results for each of the 15 iterations, rather than picking the results from a single iteration. Table 1 contains our results for the Hansards data. For the Romanian data, we obtained similar behavior, but we leave out these results due to space limitations.

From our experiments, we see that both I2CR-4 and I2CR-3 converge to solutions which give better alignment quality than those of IBM Model 1. Moreover, I2CR-3 is strictly speaking worse than IBM Model 2 and its performance lies in-between that of IBM Model 1 and IBM Model 2. On the other hand, extracting the alignments from I2CR-4 with its natural decoding rule (using $t \times \sqrt{t \times d}$) produces better F-Measure scores than those of IBM Model 2. We feel that even though our convex models are not superior in every way to IBM Model 2, their relatively easy structure and similarity to IBM Model 2 offer some deep insights into what can be accomplished with a convex relaxation. Lastly, we note that it is possible that the balance between the t and d parameters in I2CR-3 should be more carefully chosen within the weighted geometric mean (recall that we used $\beta = 1/2$) to produce the optimal results. Indeed, if we had set $\beta = 1$ in I2CR-3 we get IBM Model 1; on the other hand, setting $\beta = 0$ gives a model that ignores lexical parameters and has weak performance.

So as to better understand the need for an IBM Model 1 objective within our convex relaxation, we also compared I2CR-3 with I2CR-1 trained via the setup in (Simion, Collins, and Stein 2013). Our analysis found that I2CR-1 got AER and F-Measure scores that were very close to those of IBM Model 1 (using the same setup as (Simion, Collins, and Stein 2013), I2CR-1 has AER and F-Measure numbers that hover around .19 and .71, respectively, while IBM Model 1 has AER and F-Measure numbers close to .21 and .70, respectively). Since I2CR-3 performs better than I2CR-1, what this says is that even though the $\min$ is a stronger relaxation of the product of two probabilities than the square root (c.f. Theorem 2), the objective (value) difference between a convex relaxation and the original problem it estimates is not the most important feature when picking between various relaxations.

Lastly, we also conducted SMT experiments using the cdec system (Dyer et al. 2010) on a subset of the Europarl English-German data using BLEU as our metric (Papineni et al. 2002) along with the "grow-diagonal-final" heuristic (Och and Ney 2003). In computing BLEU, we ran cdec three times over the data and report the average test BLEU score achieved. Using alignments generated by IBM Model 2 and I2CR-4 we respectively obtained BLEU scores of **0.175202** and **0.1751417**. With the default FastAlign system cdec obtained **0.177983** BLEU.

Conclusions and Future Work

Generalizing the work of (Simion, Collins, and Stein 2013), we have introduced a class of convex relaxations for the unsupervised learning of alignments in statistical machine translation with performance comparable to the commonly-used IBM Model 2. Extending the convexity results of (Simion, Collins, and Stein 2013) allows us to better understand the old results and develop further applications. Future work will consider different relaxations within the class we have introduced, and apply our method to other NLP tasks and problems beyond IBM Model 2.

Acknowledgments

Andrei Simion was supported by NSF grant 1161814, and by a Google research award. Michael Collins was partially supported by NSF grant 1161814. Cliff Stein is supported in part by NSF grants CCF-1349602 and CCF-1421161. The first author thanks Chris Dyer and Phillip Koehn for their help in answering several Giza++ and cdec questions. We also thank the anonymous reviewers for many useful comments; we hope to pursue the comments we were not able to address in a followup paper.

References

Boyd, S., and Vandenberghe, L. 2004. Convex optimization. Cambridge University Press.

Brown, P.; Della-Pietra, V.; Della-Pietra, S.; and Mercer, R. L. 1993. The mathematics of statistical machine translation: Parameter estimation. In *Computational Linguistics*.

Bullen, P.; Mitrinovic, D.; and Vasic, M. 1987. Means and their inequalities. Springer.

Dyer, C.; Lopez, A.; Ganitkevitch, J.; Weese, J.; Ture, F.; Blunsom, P.; Setiawan, H.; Eidelman, V.; and Resnik, P. 2010. cdec: A decoder, alignment, and learning framework for finite-state and context-free translation models. In *Proceedings of ACL*.

Dyer, C.; Chahuneau, V.; and Smith, N. 2013. A simple, fast, and effective reparameterization of ibm model 2. In *Proceedings of NAACL*.

Fraser, A., and Marcu, D. 2007. Measuring word alignment quality for statistical machine translation. In *Journal Computational Linguistics*.

Guo, Y., and Schuurmans, D. 2007. Convex relaxations of latent variable training. In *Proceedings of NIPS*.

Liang, P.; Taskar, B.; and Klein, D. 2006. Alignment by agreement. In *Proceedings of HLT-NAACL*.

Michalcea, R., and Pederson, T. 2003. An evaluation exercise in word alignment. In *HLT-NAACL 2003: Workshop in building and using Parallel Texts: Data Driven Machine Translation and Beyond*.

Moore, R. 2004. Improving ibm word-alignment model 1. In *Proceedings of the ACL*.

Och, F., and Ney, H. 2003. A systematic comparison of various statistical alignment models. In *Journal of Computational Linguistics*.

Papineni, K.; Roukos, S.; Ward, T.; and Zhu, W. 2002. "bleu: A method for automatic evaluation of machine translation". In *Proceedings of the ACL*.

Rose, K. 1998. Deterministic annealing for clustering, compression, classification, regression, and related optimization problems. In *Proceedings of the IEEE*.

Simion, A.; Collins, M.; and Stein, C. 2013. A convex alternative to ibm model 2. In *Proceedings of EMNLP*.

Simion, A.; Collins, M.; and Stein, C. 2014. Some experiments with a convex ibm model 2. In *Proceedings of EACL*.

Smith, N., and Eisner, J. 2004. Annealing techniques for unsupervised statistical language learning. In *Proceedings of ACL*.

Toutanova, K., and Galley, M. 2011. Why initialization matters for ibm model 1: Multiple optima and non-strict convexity. In *Proceedings of ACL*.

Vogel, S.; Ney, H.; and Tillman, C. 1996. Hmm-based word alignment in statistical translation. In *Proceedings of COLING*.

Zalinescu, C. 2002. Convex analysis in general vector spaces. World Scientific.