Online Relative Margin Maximization for Statistical Machine Translation

Vladimir Eidelman
Computer Science
and UMIACS
University of Maryland
College Park, MD
vlad@umiacs.umd.edu

Yuval Marton
Microsoft
City Center Plaza
Bellevue, WA
yuvalmarton@gmail.com

Philip Resnik
Linguistics
and UMIACS
University of Maryland
College Park, MD
resnik@umd.edu

Abstract

Recent advances in large-margin learning have shown that better generalization can be achieved by incorporating higher order information into the optimization, such as the spread of the data. However, these solutions are impractical in complex structured prediction problems such as statistical machine translation. We present an online gradient-based algorithm for relative margin maximization, which bounds the spread of the projected data while maximizing the margin. We evaluate our optimizer on Chinese-English and Arabic-English translation tasks, each with small and large feature sets, and show that our learner is able to achieve significant improvements of 1.2-2 BLEU and 1.7-4.3 TER on average over state-of-the-art optimizers with the large feature set.

1 Introduction

The desire to incorporate high-dimensional sparse feature representations into statistical machine translation (SMT) models has driven recent research away from Minimum Error Rate Training (MERT) (Och, 2003), and toward other discriminative methods that can optimize more features. Examples include minimum risk (Smith and Eisner, 2006), pairwise ranking (PRO) (Hopkins and May, 2011), RAMPION (Gimpel and Smith, 2012), and variations of the margin-infused relaxation algorithm (MIRA) (Watanabe et al., 2007; Chiang et al., 2008; Cherry and Foster, 2012). While the objective function and optimization method vary for each optimizer, they can all be broadly described as learning a linear model, or parameter vector $\mathbf{w}$, which is used to score alternative translation hypotheses.

In every SMT system, and in machine learning in general, the goal of learning is to find a model that generalizes well, i.e. one that will yield good translations for previously unseen sentences. However, as the dimension of the feature space increases, generalization becomes increasingly difficult. Since only a small portion of all (sparse) features may be observed in a relatively small fixed set of instances during tuning, we are prone to overfit the training data. An alternative approach for solving this problem is estimating discriminative feature weights directly on the training bitext (Tillmann and Zhang, 2006; Blunsom et al., 2008; Simianer et al., 2012), which is usually substantially larger than the tuning set, but this is complementary to our goal here of better generalization given a fixed size tuning set.

In order to achieve that goal, we need to carefully choose what objective to optimize, and how to perform parameter estimation of $\mathbf{w}$ for this objective. We focus on large-margin methods such as SVM (Joachims, 1998) and passive-aggressive algorithms such as MIRA. Intuitively these seek a $\mathbf{w}$ such that the *separating distance* in geometric space of two hypotheses is at least as large as the *cost* incurred by selecting the incorrect one. This criterion performs well in practice at finding a linear separator in high-dimensional feature spaces (Tsochantaridis et al., 2004; Crammer et al., 2006).

Now, recent advances in machine learning have shown that the generalization ability of these learners can be improved by utilizing second order information, as in the Second Order Perceptron (Cesa-Bianchi et al., 2005), Gaussian Margin Machines (Crammer et al., 2009b), confidence-weighted learning (Dredze and Crammer, 2008), AROW (Crammer et al., 2009a; Chiang, 2012) and Relative Margin Machines (RMM) (Shivaswamy and Jebara, 2009b). The latter, RMM, was introduced as an effective and less computationally expensive way to incorporate the *spread* of the data – second order information about the

Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, pages 1116–1126,
Sofia, Bulgaria, August 4-9 2013.

distance between hypotheses when projected onto the line defined by the weight vector $\mathbf{w}$.

Unfortunately, not all advances in machine learning are easy to apply to structured prediction problems such as SMT; the latter often involve latent variables and surrogate references, resulting in loss functions that have not been well explored in machine learning (Mcallester and Keshet, 2011; Gimpel and Smith, 2012). Although Shivaswamy and Jebara extended RMM to handle sequential structured prediction (Shivaswamy and Jebara, 2009a), their batch approach to quadratic optimization, using existing off-the-shelf QP solvers, does not provide a practical solution: as Taskar et al. (2006) observe, "off-the-shelf QP solvers tend to scale poorly with problem and training sample size" for structured prediction problems.. This motivates an online gradient-based optimization approach—an approach that is particularly attractive because its simple update is well suited for efficiently processing structured objects with sparse features (Crammer et al., 2012).

The contributions of this paper include **(1)** introduction of a loss function for structured RMM in the SMT setting, with surrogate reference translations and latent variables; **(2)** an online gradient-based solver, RM, with a closed-form parameter update to optimize the relative margin loss; and **(3)** an efficient implementation that integrates well with the open source cdec SMT system (Dyer et al., 2010).[1] In addition, **(4)** as our solution is not dependent on any specific QP solver, it can be easily incorporated into practically any gradient-based learning algorithm.

After background discussion on learning in SMT (§2), we introduce a novel online learning algorithm for relative margin maximization suitable for SMT (§3). First, we introduce RMM (§3.1) and propose a latent structured relative margin objective which incorporates cost-augmented hypothesis selection and latent variables. Then, we derive a simple closed-form online update necessary to create a large margin solution while simultaneously bounding the spread of the projection of the data (§3.2). Chinese-English translation experiments show that our algorithm, RM, significantly outperforms strong state-of-the-art optimizers, in both a basic feature setting and high-dimensional (sparse) feature space (§4). Additional Arabic-English experiments further validate these results, even where previously MERT was shown to be advantageous (§5). Finally, we discuss the spread and other key issues of RM (§6), and conclude with discussion of future work (§7).

2 Learning in SMT

Given an input sentence in the source language $x \in \mathcal{X}$, we want to produce a translation $y \in \mathcal{Y}(x)$ using a linear model parameterized by a weight vector $\mathbf{w}$:

$$(y^*, d^*) = \underset{(y,d)\in\mathcal{Y}(x),\mathcal{D}(x)}{\arg\max} \ \mathbf{w}^\top \boldsymbol{f}(x, y, d)$$

where $\mathbf{w}^\top \boldsymbol{f}(x, y, d)$ is the weighted feature scoring function, hereafter $\mathrm{s}(x, y, d)$, and $\mathcal{Y}(x)$ is the space of possible translations of x. While many derivations $d \in \mathcal{D}(x)$ can produce a given translation, we are only able to observe y; thus we model d as a latent variable. Although our models are actually defined over derivations, they are always paired with translations, so our feature function $\boldsymbol{f}(x, y, d)$ is defined over derivation–translation pairs.[2] The learning goal is then to estimate $\mathbf{w}$.

The instability of MERT in larger feature sets (Foster and Kuhn, 2009; Hopkins and May, 2011), has motivated many alternative tuning methods for SMT. These include strategies based on batch log-linear models (Tillmann and Zhang, 2006; Blunsom et al., 2008), as well as the introduction of online linear models (Liang et al., 2006a; Arun and Koehn, 2007).

Recent batch optimizers, PRO and RAMPION, and Batch-MIRA (Cherry and Foster, 2012), have been partly motivated by existing MT infrastructures, as they iterate between decoding the entire tuning set and optimizing the parameters. PRO considers tuning a classification problem and employs a binary classifier to rank pairs of outputs. RAMPION aims to address the disconnect between MT and machine learning by optimizing a structured ramp loss with a concave-convex procedure.

2.1 Large-Margin Learning

Online large-margin algorithms, such as MIRA, have also gained prominence in SMT, thanks to their ability to learn models in high-dimensional feature spaces (Watanabe et al., 2007; Chiang et al., 2009). The usual presentation of MIRA's optimization problem is given as a quadratic program:

[1] https://github.com/veidel/cdec

[2] We may omit d in some equations for clarity.

$$\mathbf{w_{t+1}} = \arg\min_{\mathbf{w}} \frac{1}{2}||\mathbf{w} - \mathbf{w_t}||^2 + C\xi_i$$
$$\text{s.t. } \mathrm{s}(x_i, y_i, d) - \mathrm{s}(x_i, y', d) \geq \Delta_i(y') - \xi_i \quad (1)$$

where y' is the single most violated constraint, the cost $\Delta_i(y)$ is computed using an external measure of quality, such as 1-BLEU(y_i, y), and a slack variable ξ_i is introduced to allow for non-separable instances. C acts as a regularization parameter, trading off between margin maximization and constraint violations.

While solving the optimization problem relies on computing the margin between the correct output y_i, and y', in SMT our decoder is often incapable of producing the reference translation, i.e. $y_i \notin \mathcal{Y}(x_i)$. We must instead resort to selecting a surrogate reference, $y^+ \in \mathcal{Y}(x_i)$. This issue has recently received considerable attention (Liang et al., 2006a; Eidelman, 2012; Chiang, 2012), with preference given to surrogate references obtained through cost-diminished hypothesis selection. Thus, y^+ is selected based on a combination of model score and error metric from the k-best list produced by our current model. A similar selection is made for the cost-augmented hypothesis $y^- \in \mathcal{Y}(x_i)$:

$$(y^+, d^+) \leftarrow \arg\max_{(y,d)\in\mathcal{Y}(x_i),\mathcal{D}(x_i)} s(x_i, y, d) - \Delta_i(y)$$

$$(y^-, d^-) \leftarrow \arg\max_{(y,d)\in\mathcal{Y}(x_i),\mathcal{D}(x_i)} s(x_i, y, d) + \Delta_i(y)$$

In this setting, the optimization problem becomes:

$$\mathbf{w_{t+1}} = \arg\min_{\mathbf{w}} \frac{1}{2}||\mathbf{w} - \mathbf{w_t}||^2 + C\xi_i$$
$$\text{s.t. } \delta\mathrm{s}(x_i, y^+, y^-) \geq \Delta_i(y^-) - \Delta_i(y^+) - \xi_i \quad (2)$$

where $\delta\mathrm{s}(x_i, y^+, y^-)$=$\mathrm{s}(x_i, y^+, d^+)$-$\mathrm{s}(x_i, y^-, d^-)$

This leads to a variant of the structured ramp loss to be optimized:

$$\ell = - \max_{(y^+,d^+)\in\mathcal{Y}(x_i),\mathcal{D}(x_i)} \left(\mathrm{s}(x_i, y^+, d^+) - \Delta_i(y^+)\right) + \max_{(y^-,d^-)\in\mathcal{Y}(x_i),\mathcal{D}(x_i)} \left(\mathrm{s}(x_i, y^-, d^-) + \Delta_i(y^-)\right) \quad (3)$$

The passive-aggressive update (Crammer et al., 2006), which is used to solve this problem, updates $\mathbf{w}$ on each round such that the score of the correct hypothesis y^+ is greater than the score of the incorrect y^- by a margin at least as large as the cost incurred by predicting the incorrect hypothesis, while keeping the change to $\mathbf{w}$ small.

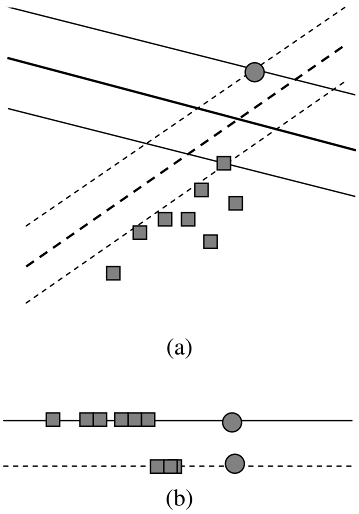

Figure 1: (a) RM and large margin solution comparison and (b) the spread of the projections given by each. RM and large margin solutions are shown with a darker dotted line and a darker solid line, respectively.

3 The Relative Margin Machine in SMT

3.1 Relative Margin Machine

The margin, the distance between the correct hypothesis and incorrect one, is defined by $\mathrm{s}(x_i, y^+, d^+)$ and $\mathrm{s}(x_i, y^-, d^-)$. It is maximized by minimizing the norm in SVM, or analogously, the proximity constraint in MIRA: $\arg\min_{\mathbf{w}} \frac{1}{2}||\mathbf{w} - \mathbf{w_t}||^2$. However, theoretical results supporting large-margin learning, such as the VC-dimension (Vapnik, 1995) or the Rademacher bound (Bartlett and Mendelson, 2003) consider measures of complexity, in addition to the empirical performance, when describing future predictive ability. The measures of complexity usually take the form of some value on the radius of the data, such as the ratio of the radius of the data to the margin (Shivaswamy and Jebara, 2009a). As radius is a way of measuring spread in any projection direction, here we will specifically be interested in the the spread of the data as measured after the projection defined by the learned model **w**.

More formally, the *spread* is the distance between y^+, and the worst candidate $(y^w, d^w) \leftarrow \arg\min_{(y,d)\in\mathcal{Y}(x_i),\mathcal{D}(x_i)} s(x_i, y, d)$, after projecting both onto the line defined by the weight vector $\mathbf{w}$. For each y', this projection is conveniently given by $\mathrm{s}(x_i, y', d)$, thus the spread is calculated as $\delta\mathrm{s}(x_i, y^+, y^w)$.

RMM was introduced as a generalization over SVM that incorporates both the margin constraint and information regarding the spread of the data. The relative margin is the ratio of the absolute, or maximum margin, to the spread of the projected data. Thus, the RMM learns a large margin solution relative to the spread of the data, or in other words, creates a max margin while simultaneously bounding the spread of the projected data. As a concrete example, consider the plot shown in Figure 1(a), with hypotheses represented by two-dimensional feature vectors. The point marked with a circle in the upper right represents $\boldsymbol{f}(x_i, y^+)$, while all other squares represent alternative incorrect hypotheses $\boldsymbol{f}(x_i, y')$. The large margin decision boundary is shown with a darker solid line, while the relative margin solution is shown with a darker dotted line. The lighter lines parallel to each define the margins, with the square at the intersection being $\boldsymbol{f}(x_i, y^-)$. The bottom portion of Figure 1(b) presents an alternative view of each solution, showing the projections of the hypotheses given the learned model of each. Notice that with a large margin solution, although the distance between y^+ and y^- is greater, the points are highly spread, extending far to the left of the decision boundary.

In contrast, with a relative margin, although we have a smaller absolute margin, the spread is smaller, all points being within a smaller distance ϵ of the decision boundary. The higher the spread of the projection, the higher the variance of the projected points, and the greater the likelihood that we will mislabel a new instance, since the high variance projections may cross the learned decision boundary. In higher dimensions, accounting for the spread becomes even more crucial, as will be discussed in Section 6.[3]

Although RMM is theoretically well-founded and improves practical performance over large-margin learning in the settings where it was introduced, it is unsuitable for most complex structured prediction in NLP. Nonetheless, since structured RMM is a generalization of Structured SVM, which shares its underlying objective with MIRA, our intuition is that SMT should be able to benefit as well. But to take advantage of the second-order information RMM utilizes for increased generalizability in SMT, we need a computationally efficient optimization procedure that does not require batch training or an off-the-shelf QP solver.

3.2 RM Algorithm

We address the above-mentioned limitations by introducing a novel online learning algorithm for relative margin maximization, RM. The relative margin solution is obtained by maximizing the same margin as Equation (2), but now with respect to the distance between y^+, and the worst candidate y^w. Thus, the relative margin dictates trading-off between a large margin as before, and a small spread of the projection, in other words, bounding the distance between y^+ and y^w. The additional computation required, namely, obtaining y^w, is efficient to perform, and has likely already happened while obtaining the k-best derivations necessary for the margin update. The online latent structured soft relative margin optimization problem is then:

$$
\begin{aligned}
\mathbf{w_{t+1}} = \arg\min_{\mathbf{w}} \frac{1}{2}||\mathbf{w} - \mathbf{w_t}||^2 + C\xi_i + D\tau_i \\
\text{s.t.:}\ \ \delta\text{s}(x_i, y^+, y^-) \geq \Delta_i(y^-) - \Delta_i(y^+) - \xi_i \\
-B - \tau_i \leq \delta\text{s}(x_i, y^+, y^w) \leq B + \tau_i
\end{aligned}
\tag{4}
$$

where additional bounding constraints are added to the usual margin constraints in order to contain the spread by bounding the difference in projections. B is an additional parameter; it controls the spread, trading off between margin maximization and spread minimization. Notice that when $B \to \infty$, the bounding constraints disappear, and we are left with the original problem in Equation (2). D, which plays an analogous role to C, allows penalized violations of the bounding constraints.

The dual of Equation (4) can be derived as:

$$
\begin{aligned}
\max_{\alpha,\beta,\beta^*} \mathcal{L} = \sum_{y\in\mathcal{Y}(x_i)} \alpha_y - B \sum_{y\in\mathcal{Y}(x_i)} \beta_y - B \sum_{y\in\mathcal{Y}(x_i)} \beta^*_y \\
-\frac{1}{2}\Big\langle \sum_{y\in\mathcal{Y}(x_i)} \alpha_y \omega_i(y^+, y) - \sum_{y\in\mathcal{Y}(x_i)} \beta_y \omega_i(y^+, y) \\
+ \sum_{y\in\mathcal{Y}(x_i)} \beta^*_y \omega_i(y^+, y), \\
\sum_{y'\in\mathcal{Y}(x_j)} \alpha_{y'} \omega_j(y^+, y') - \sum_{y'\in\mathcal{Y}(x_j)} \beta_{y'} \omega_j(y^+, y') \\
+ \sum_{y'\in\mathcal{Y}(x_j)} \beta^*_{y'} \omega_j(y^+, y') \Big\rangle
\end{aligned}
\tag{5}
$$

where the α Lagrange multiplier corresponds to the standard margin constraint, while β and

[3] The motivation of confidence-weighted estimation (Dredze and Crammer, 2008) and AROW (Crammer et al., 2009a) is related in spirit. They use second-order information in the form of a distribution over weights to change the maximum margin solution.

β^* each correspond to a bounding constraint, and $\omega_i(y^+, y')$ corresponds to the difference of $\boldsymbol{f}(x_i, y^+, d^+)$ and $\boldsymbol{f}(x_i, y', d')$. The weight update can then be obtained from the dual variables:

$$\sum \alpha_y \omega_i(y^+, y) - \sum \beta_y \omega_i(y^+, y) + \sum \beta^*_y \omega_i(y^+, y) \tag{6}$$

The dual in Equation (5) can be optimized using a cutting plane algorithm, an effective method for solving a relaxed optimization problem in the dual, used in Structured SVM, MIRA, and RMM (Tsochantaridis et al., 2004; Chiang, 2012; Shivaswamy and Jebara, 2009a). The cutting plane presented in Alg. 1 decomposes the overall problem into subproblems which are solved independently by creating working sets S_i^j, which correspond to the largest violations of either the margin constraint, or bounding constraints, and iteratively satisfying the constraints in each set.

The cutting plane in Alg. 1 makes use of the the closed-form gradient-based updates we derived for RM presented in Alg. 2. The updates amount to performing a subgradient descent step to update **w** in accordance with the constraints. Since the constraint matrix of the dual program is not strictly decomposable across constraint types, we are in effect solving an approximation of the original problem.

Algorithm 1 RM Cutting Plane Algorithm (adapted from (Shivaswamy and Jebara, 2009a))

```
Require: $i^{th}$ training example $(x_i, y_i)$, weight $\mathbf{w}$, margin
    reg. $C$, bound $B$, bound reg. $D$, $\epsilon$, $\epsilon_B$
$S_i^1 \leftarrow \{y^+\}, S_i^2 \leftarrow \{y^+\}, S_i^3 \leftarrow \{y^+\}$
repeat
    $H(y) := \Delta_i(y) - \Delta_i(y^+) - \delta \mathrm{s}(x_i, y^+, y)$
    $y_1 \leftarrow \arg\max_{y \in \mathcal{Y}(x_i)} H(y)$
    $y_2 \leftarrow \arg\max_{y \in \mathcal{Y}(x_i)} G(y) := \delta \mathrm{s}(x_i, y^+, y)$
    $y_3 \leftarrow \arg\min_{y \in \mathcal{Y}(x_i)} -G(y)$
    $\xi \leftarrow \max\{0, \max_{y \in S_i} H(y)\}$
    $V_1 \leftarrow H(y_1) - \xi - \epsilon$
    $V_2 \leftarrow G(y_2) - B - \epsilon_B$
    $V_3 \leftarrow -G(y_3) - B - \epsilon_B$
    $j \leftarrow \arg\max_{j' \in \{1,2,3\}} V_{j'}$
    if $V_j > 0$ then
        $S_i^j \leftarrow S_i^j \cup \{y_j\}$
        OPTIMIZE($\mathbf{w}, S_i^1, S_i^2, S_i^3, C, B$)        ▷ see Alg. 2
    end if
until $S_i^1, S_i^2, S_i^3$ do not change
```

Algorithm 2 RM update with α, β, β^*

```
procedure OPTIMIZE($\mathbf{w}, S_i^1, S_i^2, S_i^3, C, B$)
    while $\mathbf{w}$ changes do
        if $|S_i^1| > 1$ then
            UPDATEMARGIN($\mathbf{w}, S_i^1, C$)
        end if
        if $|S_i^2| > 1$ then
            UPDATEUPPERBOUND($\mathbf{w}, S_i^2, B$)
        end if
        if $|S_i^3| > 1$ then
            UPDATELOWERBOUND($\mathbf{w}, S_i^3, B$)
        end if
    end while
end procedure
procedure UPDATEMARGIN($\mathbf{w}, S_i^1, C$)
    $\alpha_y \leftarrow 0$ for all $y \in S_i^1$
    $\alpha_{y_i^+} \leftarrow C$
    for $n \leftarrow 1...MaxIter$ do
        Select two constraints $y, y'$ from $S_i^1$
        $\gamma_\alpha \leftarrow \frac{\Delta_i(y') - \Delta_i(y) - \delta \mathrm{s}(x_i, y, y')}{||\omega(y,y')||^2}$
        $\gamma_\alpha \leftarrow \max(-\alpha_y, \min(\alpha_{y'}, \gamma_\alpha))$
        $\alpha_y \leftarrow \alpha_y + \gamma_\alpha$ ;  $\alpha'_y \leftarrow \alpha'_y - \gamma_\alpha$
        $\mathbf{w} \leftarrow \mathbf{w} + \gamma_\alpha(\omega(y, y'))$
    end for
end procedure
procedure UPDATEUPPERBOUND($\mathbf{w}, S_i^2, B$)
    $\beta_y \leftarrow 0$ for all $y \in S_i^2$
    for $n \leftarrow 1...MaxIter$ do
        Select one constraint $y$ from $S_i^2$
        $\gamma_\beta \leftarrow \max(0, \frac{B - \delta \mathrm{s}(x_i, y^+, y)}{||\omega(y^+, y)||^2})$
        $\beta_y \leftarrow \beta_y + \gamma_\beta$
        $\mathbf{w} \leftarrow \mathbf{w} - \gamma_\beta(\omega(y^+, y))$
    end for
end procedure
procedure UPDATELOWERBOUND($\mathbf{w}, S_i^3, B$)
    $\beta^*_y \leftarrow 0$ for all $y \in S_i^3$
    for $n \leftarrow 1...MaxIter$ do
        Select one constraint $y$ from $S_i^3$
        $\gamma_{\beta^*} \leftarrow \max(0, \frac{-B - \delta \mathrm{s}(x_i, y^+, y)}{||\omega(y^+, y)||^2})$
        $\beta^*_y \leftarrow \beta^*_y + \gamma_{\beta^*}$
        $\mathbf{w} \leftarrow \mathbf{w} + \gamma_{\beta^*}(\omega(y^+, y))$
    end for
end procedure
```

Alternatively, we could utilize a passive-aggressive updating strategy (Crammer et al., 2006), which would simply bypass the cutting plane and select the most violated constraint for each set, if there is one, and perform the corresponding parameter updates in Alg. 2. We refer to the resulting passive-aggressive algorithm as RM-PA, and the cutting plane version as RM-CP. Preliminary experiments showed that RM-PA performs on par with RM-CP, thus RM-PA is the one used in the empirical evaluation below.

A graphical depiction of the passive-aggressive RM update is presented in Figure 2. The upper right circle represents y^+, while all other squares represent alternative hypotheses y'. As in the standard MIRA solution, we select the maximum margin constraint violator, y^-, shown as the triangle, and update such that the margin is greater than the cost. Additionally, we select the maximum bound-

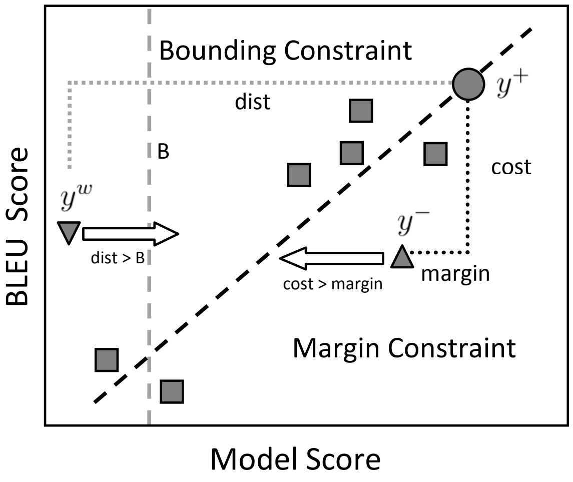

Figure 2: RM update with margin and bounding constraints. The diagonal dotted line depicts cost–margin equilibrium. The vertical gray dotted line depicts the bound B. White arrows indicate updates triggered by constraint violations. Squares are data points in the k-best list not selected for update in this round.

task	Corpus	Sentences	Tokens	
			En	Zh/Ar
Zh-En	training	1.6M	44.4M	40.4M
	tune (MT06)	1664	48k	39k
	MT03	919	28k	24k
	MT05	1082	35k	33k
Ar-En	training	1M	23.7M	22.8M
	tune (MT06)	1797	55k	49k
	MT05	1056	36k	33k
	MT08	1360	51k	45k
	4-gram LM	24M	600M	–

Table 1: Corpus statistics

ing constraint violator, y^w, shown as the upside-down triangle, and update so the distance from y^+ is no greater than B.

4 Experiments

4.1 Setup

To evaluate the advantage of explicitly accounting for the spread of the data, we conducted several experiments on two Chinese-English translation test sets, using two different feature sets in each. For training we used the non-UN and non-HK Hansards portions of the NIST training corpora, which was segmented using the Stanford segmenter (Tseng et al., 2005). The data statistics are summarized in the top half of Table 1. The English data was lowercased, tokenized and aligned using GIZA++ (Och and Ney, 2003) to obtain bidirectional alignments, which were symmetrized using the `grow-diag-final-and` method (Koehn et al., 2003). We trained a 4-gram LM on the English side of the corpus with additional words from non-NYT and non-LAT, randomly selected portions of the Gigaword v4 corpus, using modified Kneser-Ney smoothing (Chen and Goodman, 1996). We used cdec (Dyer et al., 2010) as our hierarchical phrase-based decoder, and tuned the parameters of the system to optimize BLEU (Papineni et al., 2002) on the NIST MT06 corpus.

We applied several competitive optimizers as baselines: hypergraph-based MERT (Kumar et al., 2009), k-best variants of MIRA (Crammer et al., 2006; Chiang et al., 2009), PRO (Hopkins and May, 2011), and RAMPION (Gimpel and Smith, 2012). The size of the k-best list was set to 500 for RAMPION, MIRA and RM, and 1500 for PRO, with both PRO and RAMPION utilizing k-best aggregation across iterations. RAMPION settings were as described in (Gimpel and Smith, 2012), and PRO settings as described in (Hopkins and May, 2011), with PRO requiring regularization tuning in order to be competitive with the other optimizers. MIRA and RM were run with 15 parallel learners using iterative parameter mixing (McDonald et al., 2010). All optimizers were implemented in cdec and use the same system configuration, thus the only independent variable is the optimizer itself. We set C to 0.01, and $MaxIter$ to 100. We selected the bound step size D, based on performance on a held-out dev set, to be 0.01 for the basic feature set and 0.1 for the sparse feature set. The bound constraint B was set to 1.[4] The approximate sentence-level BLEU cost Δ_i is computed in a manner similar to (Chiang et al., 2009), namely, in the context of previous 1-best translations of the tuning set. All results are averaged over 3 runs.

4.2 Feature Sets

We experimented with a small (basic) feature set, and a large (sparse) feature set. For the small feature set, we use 14 features, including a language model, 5 translation model features, penalties for unknown words, the glue rule, and rule arity. For experiments with a larger feature set, we introduced additional lexical and non-lexical sparse Boolean features of the form commonly found in the literature (Chiang et al., 2009; Watan-

[4] We also conducted an investigation into the setting of the B parameter. We explored alternative values for B, as well as scaling it by the current candidate's cost, and found that the optimizer is fairly insensitive to these changes, resulting in only minor differences in BLEU.

Optimizer	Zh	Ar
MIRA	35k	37k
PRO	95k	115k
RAMPION	22k	24k
RM	30k	32k
Active+Inactive	3.4M	4.9M

Table 2: Active sparse feature templates

abe et al., 2007; Simianer et al., 2012).

Non-lexical features include structural distortion, which captures the dependence between reordering and the size of a filler, and rule shape, which bins grammar rules by their sequence of terminals and nonterminals (Chiang et al., 2008). Lexical features on rules include rule ID, which fires on a specific grammar rule. We also introduce context-dependent lexical features for the 300 most frequent aligned word pairs (f,e) in the training corpus, which fire on triples (f,e,f_{+1}) and (f,e,f_{-1}), capturing when we see f aligned to e, with f_{+1} and f_{-1} occurring to the right or left of f, respectively. All other words fall into the default $\langle unk \rangle$ feature bin. In addition, we have insertion and deletion features for the 150 most frequently unaligned target and source words. These feature templates resulted in a total of 3.4 million possible features, of which only a fraction were active for the respective tuning set and optimizer, as shown in Table 2.

4.3 Results

As can be seen from the results in Table 3, our RM method was the best performer in all Chinese-English tests according to all measures – up to 1.9 BLEU and 6.6 TER over MIRA – even though we only optimized for BLEU.[5] Surprisingly, it seems that MIRA did not benefit as much from the sparse features as RM. The results are especially notable for the basic feature setting – up to 1.2 BLEU and 4.6 TER improvement over MERT – since MERT has been shown to be competitive with small numbers of features compared to high-dimensional optimizers such as MIRA (Chiang et al., 2008).

For the tuning set, the decoder performance was consistently the *lowest* with RM, compared to the other optimizers. We believe this is due to the RM bounding constraint being more resistant to overfitting the training data, and thus allowing for improved generalization. Conversely, while PRO had the second lowest tuning scores, it seemed to display signs of underfitting in the basic and large feature settings.

5 Additional Experiments

In order to explore the applicability of our approach to a wider range of languages, we also evaluated its performance on Arabic-English translation. All experimental details were the same as above, except those noted below.

For training, we used the non-UN portion of the NIST training corpora, which was segmented using an HMM segmenter (Lee et al., 2003). Dataset statistics are given in the bottom part of Table 1. The sparse feature templates resulted here in a total of 4.9 million possible features, of which again only a fraction were active, as shown in Table 2.

As can be seen in Table 4, in the smaller feature set, RM and MERT were the best performers, with the exception that on MT08, MIRA yielded somewhat better (+0.7) BLEU but a somewhat worse (-0.9) TER score than RM.

On the large feature set, RM is again the best performer, except, perhaps, a tied BLEU score with MIRA on MT08, but with a clear 1.8 TER gain. In both Arabic-English feature sets, MIRA seems to take the second place, while RAMPION lags behind, unlike in Chinese-English (§4).[6]

Interestingly, RM achieved substantially higher BLEU precision scores in all tests for both language pairs. However, this was also usually coupled had a higher brevity penalty (BP) than MIRA, with the BP increasing slightly when moving to the sparse setting.

6 Discussion

The trend of the results, summarized as RM gain over other optimizers averaged over all test sets, is presented in Table 5. RM shows clear advantage in both basic and sparse feature sets, over all other state-of-the-art optimizers. The RM gains are notably higher in the large feature set, which we take

[5] In the small feature set RAMPION yielded similar best BLEU scores, but worse TER. In preliminary experiments with a smaller trigram LM, our RM method consistently yielded the highest scores in all Chinese-English tests – up to 1.6 BLEU and 6.4 TER from MIRA, the second best performer.

[6] In our preliminary experiments with the smaller trigram LM, MERT did better on MT05 in the smaller feature set, and MIRA had a small advantage in two cases. RAMPION performed similarly to RM on the smaller feature set. RM's loss was only up to 0.8 BLEU (0.7 TER) from MERT or MIRA, while its gains were up to 1.7 BLEU and 2.1 TER over MIRA.

	Small (basic) feature set					Large (sparse) feature set				
Optimizer	Tune	MT03		MT05		Tune	MT03		MT05	
	↑BLEU	↑BLEU	↓TER	↑BLEU	↓TER	↑BLEU	↑BLEU	↓TER	↑BLEU	↓TER
MERT	35.4	35.8	60.8	32.4	63.9	-	-	-	-	-
MIRA	35.5	35.8	61.1	32.1	64.6	36.6	35.9	60.6	32.1	64.1
PRO	34.1	36.0	60.2	31.7	63.4	35.7	34.8	56.1	31.4	59.1
RAMPION	35.1	**36.5**	58.6	33.0	61.3	36.7	36.9	57.7	33.3	60.6
RM	31.3	**36.5**	**56.4**	**33.6**	**59.3**	33.2	**37.5**	**54.6**	**34.0**	**57.5**

Table 3: Performance on Zh-En with basic (left) and sparse (right) feature sets on MT03 and MT05.

	Small (basic) feature set					Large (sparse) feature set				
Optimizer	Tune	MT05		MT08		Tune	MT05		MT08	
	↑BLEU	↑BLEU	↓TER	↑BLEU	↓TER	↑BLEU	↑BLEU	↓TER	↑BLEU	↓TER
MERT	43.8	**53.3**	**40.2**	**41.0**	50.7	-	-	-	-	-
MIRA	43.0	52.8	40.8	**41.3**	50.6	44.4	53.4	40.1	**41.8**	50.2
PRO	41.5	51.3	41.5	39.4	51.5	46.8	53.2	40.0	41.4	49.7
RAMPION	42.4	52.0	40.8	40.0	50.8	44.6	52.9	40.4	41.0	50.4
RM	38.5	**53.3**	**39.8**	40.6	**49.7**	43.0	**55.3**	**37.5**	**41.8**	**48.4**

Table 4: Performance on Ar-En with basic (left) and sparse (right) feature sets on MT05 and MT08.

	Small set		Large set	
Optimizer	BLEU	TER	BLEU	TER
MERT	0.4	2.6	-	-
MIRA	0.5	3.0	1.4	4.3
PRO	1.4	2.9	2.0	1.7
RAMPION	0.6	1.6	1.2	2.8

Table 5: RM gain over other optimizers averaged over all test sets.

as an indication for the importance of bounding the spread.

Spread analysis: For RM, the average spread of the projected data in the Chinese-English small feature set was 0.9±3.6 for all tuning iterations, and 0.7±2.9 for the iteration with the highest decoder performance. In comparison, the spread of the data for MIRA was 5.9±20.5 for the best iteration. In the sparse setting, RM had an average spread of 0.9±2.4 for the best iteration, while MIRA had a spread of 14.0±31.1. Similarly, on Arabic-English, RM had a spread of 0.7±2.4 in the small setting, and 0.82±1.4 in the sparse setting, while MIRA's spread was 9.4±26.8 and 11.4±22.1, for the small and sparse settings, respectively. Notice that the average spread for RM stays about the same when moving to higher dimensions, with the variance decreasing in both cases. For MIRA, however, the average spread increases in both cases, with the variance being much higher than RM. For instance, observe that the spread of MIRA on Chinese grows from 5.9 to 14.0 in the sparse feature setting. While bounding the spread is useful in the low-dimensional setting (0.7-1.5 BLEU gain with RM over MIRA as shown in Table 3), accounting for the spread is even more crucial with sparse features, where MIRA gains only up to 0.1 BLEU, while RM gains 1 BLEU. These results support the claim that our imposed bound B indeed helps decrease the spread, and that, in turn, lower spread yields better generalization performance.

Error Analysis: The inconclusive advantage of RM over MIRA (in BLEU vs. TER scores) on Arabic-English MT08 calls for a closer look. Therefore we conducted a coarse error analysis on 15 randomly selected sentences from MERT, RMM and MIRA, with basic and sparse feature settings for the latter two. This sample yielded 450 data points for analysis: output of the 5 conditions on 15 sentences scored in 6 violation categories. The categories were: function word drop, content word drop, syntactic error (with a reasonable meaning), semantic error (regardless of syntax), word order issues, and function word mistranslation and "hallucination". The purpose of this analysis was to get a *qualitative* feel for the output of each model, and a better idea as to why we obtained performance improvements. RM noticeably had more word order and excess/wrong function word issues in the basic feature setting than any optimizer. However, RM seemed to benefit the most from the sparse features, as its bad word order rate dropped close to MIRA, and its excess/wrong function word rate dropped below that of MIRA with sparse features (MIRA's rate actually doubled from its basic feature set). We conjecture both these issues will be ameliorated with syntactic features such as those in Chiang et al. (2008). This correlates with our observation that RM's overall BLEU score is negatively impacted by the BP, as the BLEU precision scores are noticeably higher.

K-best: RM is potentially more sensitive to the size and order of the k-best list. While MIRA is only concerned with the margin between y^+ and y^-, RM also accounts for the distance between y^+ and y^w. It might be the case that a larger k-best, or revisiting previous strategies for y^+ and y^- selection, such as bold updating, local updating (Liang et al., 2006b), or max-BLEU updating (Tillmann and Zhang, 2006) might have a greater impact. Also, we only explored several settings of B, and there remains a continuum of RM solutions that trade off between margin and spread in different ways.

Active features: Perhaps contrary to expectation, we did not see evidence of a correlation between the number of active features and optimizer performance. RAMPION, with the fewest features, is the closest performer to RM in Chinese, while MIRA, with a greater number, is the closest on Arabic. We also notice that while PRO had the lowest BLEU scores in Chinese, it was competitive in Arabic with the highest number of features.

7 Conclusions and Future Work

We have introduced RM, a novel online margin-based algorithm designed for optimizing high-dimensional feature spaces, which introduces constraints into a large-margin optimizer that bound the spread of the projection of the data while maximizing the margin. The closed-form online update for our relative margin solution accounts for surrogate references and latent variables.

Experimentation in statistical MT yielded significant improvements over several other state-of-the-art optimizers, especially in a high-dimensional feature space (up to 2 BLEU and 4.3 TER on average). Overall, RM achieves the best or comparable performance according to two scoring methods in two language pairs, with two test sets each, in small and large feature settings. Moreover, across conditions, RM always yielded the best combined TER-BLEU score.[7]

These improvements are achieved using standard, relatively small tuning sets, contrasted with improvements involving sparse features obtained using much larger tuning sets, on the order of hundreds of thousands of sentences (Liang et al., 2006a; Tillmann and Zhang, 2006; Blunsom et al., 2008; Simianer et al., 2012). Since our approach is complementary to scaling up the tuning data, in future work we intend to combine these two methods. In future work we also intend to explore using additional sparse features that are known to be useful in translation, e.g. syntactic features explored by Chiang et al. (2008).

Finally, although motivated by statistical machine translation, RM is a gradient-based method that can easily be applied to other problems. We plan to investigate its utility elsewhere in NLP (e.g. for parsing) as well as in other domains involving high-dimensional structured prediction.

Acknowledgments

We would like to thank Pannaga Shivaswamy for valuable discussions, and the anonymous reviewers for their comments. Vladimir Eidelman is supported by a National Defense Science and Engineering Graduate Fellowship. This work was also supported in part by the BOLT program of the Defense Advanced Research Projects Agency, Contract HR0011-12-C-0015.

References

Abishek Arun and Philipp Koehn. 2007. Online learning methods for discriminative training of phrase based statistical machine translation. In *MT Summit XI*.

Peter L. Bartlett and Shahar Mendelson. 2003. Rademacher and gaussian complexities: risk bounds and structural results. *J. Mach. Learn. Res.*, 3:463–482, March.

Phil Blunsom, Trevor Cohn, and Miles Osborne. 2008. A discriminative latent variable model for statistical machine translation. In *Proceedings of ACL-08: HLT*, Columbus, Ohio, June.

[7] We and other researchers often use $\frac{1}{2}(\text{TER} - \text{BLEU})$ as a combined SMT quality metric.

Nicolò Cesa-Bianchi, Alex Conconi, and Claudio Gentile. 2005. A second-order perceptron algorithm. *SIAM J. Comput.*, 34(3):640–668, March.

Stanley F. Chen and Joshua Goodman. 1996. An empirical study of smoothing techniques for language modeling. In *Proceedings of the 34th Annual Meeting of the Association for Computational Linguistics*, pages 310–318.

Colin Cherry and George Foster. 2012. Batch tuning strategies for statistical machine translation. In *Proceedings of NAACL*.

David Chiang, Yuval Marton, and Philip Resnik. 2008. Online large-margin training of syntactic and structural translation features. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Waikiki, Honolulu, Hawaii.

David Chiang, Kevin Knight, and Wei Wang. 2009. 11,001 new features for statistical machine translation. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, NAACL '09, pages 218–226.

David Chiang. 2012. Hope and fear for discriminative training of statistical translation models. *J. Machine Learning Research*.

Koby Crammer, Ofer Dekel, Joseph Keshet, Shai Shalev-Shwartz, and Yoram Singer. 2006. Online passive-aggressive algorithms. *J. Mach. Learn. Res.*, 7:551–585.

Koby Crammer, Alex Kulesza, and Mark Dredze. 2009a. Adaptive regularization of weight vectors. In *Advances in Neural Information Processing Systems 22*, pages 414–422.

Koby Crammer, Mehryar Mohri, and Fernando Pereira. 2009b. Gaussian margin machines. *Journal of Machine Learning Research - Proceedings Track*, 5:105–112.

Koby Crammer, Mark Dredze, and Fernando Pereira. 2012. Confidence-weighted linear classification for text categorization. *J. Mach. Learn. Res.*, 98888:1891–1926, June.

Mark Dredze and Koby Crammer. 2008. Confidence-weighted linear classification. In *In ICML 08: Proceedings of the 25th international conference on Machine learning*, pages 264–271. ACM.

Chris Dyer, Adam Lopez, Juri Ganitkevitch, Jonathan Weese, Ferhan Ture, Phil Blunsom, Hendra Setiawan, Vladimir Eidelman, and Philip Resnik. 2010. cdec: A decoder, alignment, and learning framework for finite-state and context-free translation models. In *Proceedings of ACL System Demonstrations*.

Vladimir Eidelman. 2012. Optimization strategies for online large-margin learning in machine translation. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*.

George Foster and Roland Kuhn. 2009. Stabilizing minimum error rate training. In *Proceedings of the Fourth Workshop on Statistical Machine Translation*, pages 242–249, Athens, Greece, March. Association for Computational Linguistics.

Kevin Gimpel and Noah A. Smith. 2012. Structured ramp loss minimization for machine translation. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics*.

Mark Hopkins and Jonathan May. 2011. Tuning as ranking. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1352–1362, Edinburgh, Scotland, UK., July. Association for Computational Linguistics.

Thorsten Joachims. 1998. Text Categorization with Support Vector Machines: Learning with Many Relevant Features. In Claire Nédellec and Céline Rouveirol, editors, *European Conference on Machine Learning*, pages 137–142, Berlin. Springer.

Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*, NAACL '03, Stroudsburg, PA, USA.

Shankar Kumar, Wolfgang Macherey, Chris Dyer, and Franz Och. 2009. Efficient minimum error rate training and minimum bayes-risk decoding for translation hypergraphs and lattices. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 163–171.

Young-Suk Lee, Kishore Papineni, Salim Roukos, Ossama Emam, and Hany Hassan. 2003. Language model based Arabic word segmentation. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics - Volume 1*, pages 399–406.

Percy Liang, Alexandre Bouchard-Côté, Dan Klein, and Ben Taskar. 2006a. An end-to-end discriminative approach to machine translation. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, ACL-44, pages 761–768.

Percy Liang, Alexandre Bouchard-Côté, Dan Klein, and Ben Taskar. 2006b. An end-to-end discriminative approach to machine translation. In *Proceedings of the 2006 International Conference on Computational Linguistics (COLING) - the Association for Computational Linguistics (ACL)*.

David Mcallester and Joseph Keshet. 2011. Generalization bounds and consistency for latent structural probit and ramp loss. In J. Shawe-Taylor, R.S. Zemel, P. Bartlett, F.C.N. Pereira, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems 24*, pages 2205–2212.

Ryan McDonald, Keith Hall, and Gideon Mann. 2010. Distributed training strategies for the structured perceptron. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 456–464, Los Angeles, California.

Franz Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. In *Computational Linguistics*, volume 29(21), pages 19–51.

Franz Josef Och. 2003. Minimum error rate training in statistical machine translation. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 160–167.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318.

Pannagadatta Shivaswamy and Tony Jebara. 2009a. Structured prediction with relative margin. In *In International Conference on Machine Learning and Applications*.

Pannagadatta K Shivaswamy and Tony Jebara. 2009b. Relative margin machines. In *In Advances in Neural Information Processing Systems 21*. MIT Press.

Patrick Simianer, Stefan Riezler, and Chris Dyer. 2012. Joint feature selection in distributed stochastic learning for large-scale discriminative training in smt. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Jeju Island, Korea, July.

David A. Smith and Jason Eisner. 2006. Minimum risk annealing for training log-linear models. In *Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions*, Sydney, Australia, July. Association for Computational Linguistics.

Ben Taskar, Simon Lacoste-Julien, and Michael I. Jordan. 2006. Structured prediction, dual extragradient and bregman projections. *J. Mach. Learn. Res.*, 7:1627–1653, December.

Christoph Tillmann and Tong Zhang. 2006. A discriminative global training algorithm for statistical MT. In *Proceedings of the 2006 International Conference on Computational Linguistics (COLING) - the Association for Computational Linguistics (ACL)*.

Huihsin Tseng, Pi-Chuan Chang, Galen Andrew, Daniel Jurafsky, and Christopher Manning. 2005. A conditional random field word segmenter. In *Fourth SIGHAN Workshop on Chinese Language Processing*.

Ioannis Tsochantaridis, Thomas Hofmann, Thorsten Joachims, and Yasemin Altun. 2004. Support vector machine learning for interdependent and structured output spaces. In *Proceedings of the twenty-first international conference on Machine learning*, ICML '04.

Vladimir N. Vapnik. 1995. *The nature of statistical learning theory*. Springer-Verlag New York, Inc., New York, NY, USA.

Taro Watanabe, Jun Suzuki, Hajime Tsukada, and Hideki Isozaki. 2007. Online large-margin training for statistical machine translation. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, Prague, Czech Republic, June. Association for Computational Linguistics.