Accurate max-margin training for structured output spaces

Sunita Sarawagi SUNITA@IITB.AC.IN
IIT Bombay, India

Rahul Gupta GRAHUL@CSE.IITB.AC.IN
IIT Bombay, India

Abstract

Tsochantaridis et al. (2005) proposed two formulations for maximum margin training of structured spaces: margin scaling and slack scaling. While margin scaling has been extensively used since it requires the same kind of MAP inference as normal structured prediction, slack scaling is believed to be more accurate and better-behaved. We present an efficient variational approximation to the slack scaling method that solves its inference bottleneck while retaining its accuracy advantage over margin scaling.

We further argue that existing scaling approaches do not separate the true labeling comprehensively while generating violating constraints. We propose a new max-margin trainer PosLearn that generates violators to ensure separation at each position of a decomposable loss function. Empirical results on real datasets illustrate that PosLearn can reduce test error by up to 25% over margin scaling and 10% over slack scaling. Further, PosLearn violators can be generated more efficiently than slack violators; for many structured tasks the time required is just twice that of MAP inference.

1. Introduction

The max-margin framework for training structured prediction models generalizes the benefits of support vector machines (SVMs) to predicting complex objects. A popular member of this framework is the margin scaling method (Tsochantaridis et al., 2005; Taskar, 2004; LeCun et al., 2006; Crammer & Singer, 2003) that tries to ensure that the score of the correct prediction is separated from the score of an incorrect prediction by a margin equal to the error of the prediction. This method has been used extensively in many applications, including sequence labeling (Taskar, 2004; Tsochantaridis et al., 2005), image segmentation (Taskar, 2004; Ratliff et al., 2007), grammar parsing (Taskar et al., 2004), dependency parsing (McDonald et al., 2005b), bipartite matching (Taskar, 2004) and text segmentation (McDonald et al., 2005a). A reason for its wide-spread use is that it can exploit the decomposability of the error function to find the most violating constraint using the maximum a-posteriori (MAP) inference algorithm used for prediction.

An alternative formulation (Tsochantaridis et al., 2005) is to ensure that all labelings are separated by a fixed margin of one but penalize violations in proportion to their errors. This method, called slack scaling, generally provides higher accuracy than margin scaling which gives too much importance to labelings with large errors even after they are well-separated, sometimes at the expense of instances that are not even separated. Another shortcoming of margin scaling is that it requires an error function that is linearly comparable with the feature values, whereas slack scaling is invariant to scaling of the error function. In spite of the advantages, slack scaling is not popular because it requires inferring the labeling which maximizes a non-decomposable metric – difference of score and error inverse.

In this paper we make two contributions in max-margin training of structured models.

First, we address the computational challenge of infering the labelings required when training via slack scaling. We propose a variational approximation of the slack loss so that the most violating labeling is found using the same loss augmented MAP inference as in margin scaling. We demonstrate that accuracy-wise

Appearing in *Proceedings of the 25th International Conference on Machine Learning*, Helsinki, Finland, 2008.

our slack approximation is much better than margin scaling and close to the more expensive slack scaling.

Second, we propose a new max-margin framework for training models with decomposable error functions that, like the slack scaling method, is scale invariant and discounts labelings well-separated from the margin. The inference step it requires is much simpler than required by slack scaling. In particular, for Markov models we show that the inference of the most violating labelings is only a factor of two more expensive than in margin scaling. The basic idea of the new learner, that we call PosLearn, is to associate a different slack variable for each error position of a decomposable error function. We show that this leads to a better characterization of the loss than both the slack and margin scaling methods that define loss in terms of a *single* most violating labeling. Empirically, PosLearn reduces error by up to 25% over margin scaling and 10% over slack scaling in various tasks.

2. Existing methods for max-margin training

We consider structured prediction problems that associate a score $\mathrm{s}(\mathbf{x},\mathbf{y})$ for each output $\mathbf{y} \in \mathcal{Y}$ of an input $\mathbf{x}$, and predict the output $\mathbf{y}^*$ with maximum score. The scoring function $\mathrm{s}(\mathbf{x},\mathbf{y})$ is a dot product of a feature vector $\mathbf{f}(\mathbf{x},\mathbf{y})$ defined jointly over the input $\mathbf{x}$ and output $\mathbf{y}$, and the corresponding parameter vector $\mathbf{w}$. The space of possible outputs $\mathcal{Y}$ can be exponentially large. Thus, efficient solutions for $\mathbf{y}^* = \mathrm{argmax}_{\mathbf{y}\in\mathcal{Y}}\mathbf{w}^T\mathbf{f}(\mathbf{x},\mathbf{y})$ crucially depend on the decomposability of the feature vector $\mathbf{f}$ over components of $\mathbf{y}$. During training the goal is to find a $\mathbf{w}$ using a set of labeled input-output pairs $(\mathbf{x}_i,\mathbf{y}_i) : i = 1\ldots N$ so as to minimize prediction error. The error of predicting $\mathbf{y}$ for an instance $\mathbf{x}_i$ whose correct label is $\mathbf{y}_i$ is user-provided. We denote it by $L_i(\mathbf{y})$. In max-margin methods, the training goal is translated to finding a $\mathbf{w}$ that minimizes the sum of the loss on the labeled data while imposing a regularization penalty for overfitting. The loss is a computationally convenient combination of the user-provided error function and feature-derived scores so as to both minimize training error and maximize the margin between correct and incorrect outputs. There are two popular loss functions for structured learning tasks: margin scaler and slack scaler. We review them briefly.

2.1. Margin scaling

In margin scaling, the goal is to find $\mathbf{w}$ such that the difference in score $\mathbf{w}^T\delta\mathbf{f}_i(\mathbf{y}) = \mathbf{w}^T\mathbf{f}(\mathbf{x}_i,\mathbf{y}_i) - \mathbf{w}^T\mathbf{f}(\mathbf{x}_i,\mathbf{y})$ of the correct output $\mathbf{y}_i$ from an incorrect labeling $\mathbf{y}$ is at least $L_i(\mathbf{y})$. This is formulated as:

$$\begin{aligned}
&\min_{\mathbf{w},\xi} \frac{1}{2}||\mathbf{w}||^2 + C\sum_{i=1}^{N}\xi_i \\
&\text{s.t. } \mathbf{w}^T\delta\mathbf{f}_i(\mathbf{y}) \geq L_i(\mathbf{y}) - \xi_i \quad \forall \mathbf{y} \neq \mathbf{y}_i, i : 1\ldots N \\
&\xi_i \geq 0 \quad i : 1\ldots N
\end{aligned}$$

Two category of methods have been proposed to optimize the above QP. The first category is based on the cutting plane algorithm to avoid generating the exponentially many constraints. This involves incrementally finding the output $\mathbf{y}^M = \mathrm{argmax}_{\mathbf{y}}(\mathbf{w}^T\mathbf{f}(\mathbf{x}_i,\mathbf{y}) + L_i(\mathbf{y}))$ which most violates the constraint. $\mathbf{y}^M$ can be found using the same inference algorithm as MAP $\mathbf{y}^* = \mathrm{argmax}_{\mathbf{y}}\mathbf{w}^T\mathbf{f}(\mathbf{x}_i,\mathbf{y})$ when $L_i(\mathbf{y})$ decomposes over variable subsets no larger than the subsets in the decomposition of $\mathbf{f}(\mathbf{x}_i,\mathbf{y})$. This category includes exact gradient ascent methods (Tsochantaridis et al., 2005), stochastic gradient methods (Bordes et al., 2007) and online sub-gradient methods (Ratliff et al., 2007). The online structured learning methods of (Crammer & Singer, 2003) follow a perceptron based framework but their constraints are identical to the margin scaling method described here. The second category of methods (Taskar, 2004; Taskar et al., 2006) exploit the decomposability of the error function to create a combined program for the inference and parameter learning task.

2.2. Slack scaling

Slack scaling demands a margin of one but scales the slacks of violating outputs in proportion to their errors. The corresponding optimization problem is:

$$\begin{aligned}
&\min_{\mathbf{w},\xi} \frac{1}{2}||\mathbf{w}||^2 + C\sum_{i=1}^{N}\xi_i \\
&\text{s.t. } \mathbf{w}^T\delta\mathbf{f}_i(\mathbf{y}) \geq 1 - \frac{\xi_i}{L_i(\mathbf{y})} \quad \forall \mathbf{y} \neq \mathbf{y}_i, i : 1\ldots N \\
&\xi_i \geq 0 \quad i : 1\ldots N
\end{aligned}$$

The optimization of the above QP via the cutting plane algorithm requires the inference of the labeling $\mathbf{y}^S = \mathrm{argmax}_{\mathbf{y}}(1 - \mathbf{w}^T\delta\mathbf{f}_i(\mathbf{y}) - \frac{\xi_i}{L_i(\mathbf{y})})$. Unlike for margin scaling, even with decomposable loss and scoring functions, it is not easy to find $\mathbf{y}^S$ efficiently. For this reason, the slack scaling approach is not popular.

However, the slack loss is in many ways better behaved than margin loss (Tsochantaridis et al., 2005). Margin scaling gives too much importance to instances which are already well-separated from the margin. This hurts because the loss ξ_i is determined by a single most violating labeling. If a labeling imposes a difficult margin

requirement because of its large error, the optimizer will appropriately increase ξ_i. After that, there is no incentive to improving separability of any other labeling of that instance. In contrast, the slack scaling loss will ignore instances that are separated by a margin of 1, and ξ_i is determined by labelings that matter because of their being close to the margin. Empirically, we found slack scaling to give better accuracy than margin scaling (Section 5). Slack scaling also makes it convenient for an end-user to define an error function and a feature vector and tune C because the error function can be arbitrarily scaled vis-a-vis the feature vector.

3. Approximate slack scaling

We present a variational approximation to the slack inference problem that is applicable for any structured model for which we can only solve for the MAP efficiently. The slack inference problem is to find

$$\begin{aligned}\mathbf{y}^S &= \text{argmax}_{\mathbf{y}\in\overline{\mathcal{Y}}}\left(1-\mathbf{w}^T\delta\mathbf{f}_i(\mathbf{y})-\frac{\xi_i}{L_i(\mathbf{y})}\right)\\ &= \text{argmax}_{\mathbf{y}\in\overline{\mathcal{Y}}}\left(\text{s}_i(\mathbf{y})-\frac{\xi_i}{L_i(\mathbf{y})}\right) \quad (1)\end{aligned}$$

where $\text{s}_i(\mathbf{y}) = \mathbf{w}^T\mathbf{f}(\mathbf{x}_i,\mathbf{y})$, $\overline{\mathcal{Y}} = \{\mathbf{y} : \mathbf{y}\neq\mathbf{y}_i,\ \text{s}_i(\mathbf{y}) - \frac{\xi_i}{L_i(\mathbf{y})} > \text{s}_i(\mathbf{y}_i)-1\}$ is the set of all violating labelings.

We approximate $\mathbf{y}^S$ with another labeling $\mathbf{y}^A$. Our approximation is based on the observation that $\text{s}_i(\mathbf{y}) - \frac{\xi_i}{L_i(\mathbf{y})}$ is concave in $L_i(\mathbf{y})$ and its variational approximation can be written as a linear function of $L_i(\mathbf{y})$ (Jordan et al., 1999). Here on, we drop the subscript i wherever possible.

Claim 3.1. $s(\mathbf{y})-\frac{\xi}{L(\mathbf{y})} = \min_{\lambda\geq 0} s(\mathbf{y})+\lambda L(\mathbf{y})-2\sqrt{\xi\lambda}$

Proof. Any concave function $f(z)$ can be expressed as $\min_{\lambda\geq 0}(z\lambda - f^*(\lambda))$ where $f^*(\lambda) = \min_z(z\lambda - f(z))$ is the conjugate function of $f(z)$. The result follows from the fact that the conjugate function of $\frac{-\xi}{z}$ is $2\sqrt{\xi\lambda}$. □

Let $F'(\mathbf{y};\lambda) \triangleq \text{s}(\mathbf{y}) + \lambda L(\mathbf{y}) - 2\sqrt{\xi\lambda}$ and $F(\lambda) \triangleq \max_{\mathbf{y}\neq\mathbf{y}_i} F'(\mathbf{y};\lambda)$

We now approximate the exact slack MAP objective with an upper bound as follows:

$$\begin{aligned}\max_{\mathbf{y}\in\overline{\mathcal{Y}}}\left(\text{s}(\mathbf{y})-\frac{\xi}{L(\mathbf{y})}\right) &= \max_{\mathbf{y}\in\overline{\mathcal{Y}}}\min_{\lambda\geq 0} F'(\mathbf{y};\lambda) \quad (2)\\ &\leq \min_{\lambda\geq 0}\max_{\mathbf{y}\in\overline{\mathcal{Y}}} F'(\mathbf{y};\lambda) \quad (3)\\ &\leq \min_{\lambda\geq 0} F(\lambda) \quad (4)\end{aligned}$$

For a fixed λ, we can compute $F(\lambda)$ using the loss augmented MAP algorithm employed in margin scaling to first find $\mathbf{y}_\lambda = \text{argmax}_{\mathbf{y}\neq\mathbf{y}_i} s(\mathbf{y}) + \lambda L(\mathbf{y})$ and then setting $F(\lambda) = F'(\mathbf{y}_\lambda;\lambda)$. The constraint $\mathbf{y}\neq\mathbf{y}_i$ can be met by asking the loss augmented MAP algorithm to return top two MAPs. The algorithmic extension to return top two MAPs is straight forward in many structured tasks.

We search for the λ for which the upper bound $F(\lambda)$ is minimized by exploiting the fact that $F(\lambda)$ is convex in λ.

Claim 3.2. *$F(\lambda)$ is convex in λ.*

Proof. It can be seen that $F'(\mathbf{y};\lambda)$ is convex in λ. Since $F(\lambda)$ is a max of finitely many convex functions, and max is also convex, $F(\lambda)$ is convex. □

We can compute $\min_{\lambda\geq 0} F(\lambda)$ using efficient line search algorithms such as Golden Search. During the search phase, for each λ that we encounter, we evaluate $F(\lambda)$ and thus get one labeling. Of all these labelings, we return the one with the highest $\text{s}(\mathbf{y}) - \frac{\xi}{L(\mathbf{y})}$ as $\mathbf{y}^A$. We show in the next section the range $[\lambda_l, \lambda_u]$ within which it is sufficient to perform the line search.

3.1. Upper and lower bounds for λ

Since $\lambda \geq 0$, we can use $\lambda_l = 0$ as the lower limit. However, with $\lambda = 0$, $F'(\mathbf{y};\lambda)$ is not able to distinguish between high and loss labelings with same scores commonly seen in early training iterations. It can be shown that with $\lambda_l = \frac{\epsilon}{L_{max}}$ where L_{max} is the maximum possible loss, $F(\lambda)$ for any $\lambda < \lambda_l$ will not return any violating labeling with slack score more than ϵ of the score of a labeling returned with $\lambda \geq \lambda_l$. By setting ϵ to the tolerance of the cutting-plane algorithm, we get a provably correct lower bound.

For the upper bound, it is sufficient to pick a λ_u such that for any $\lambda \geq \lambda_u$, either $F(\lambda)$ gets the same violator as $F(\lambda_u)$ or a non-violator that we are not interested in. It is sufficient to pick a λ_u such that $\text{argmax}_{\mathbf{y}\in\overline{\mathcal{Y}}} F'(\mathbf{y};\lambda_u)$ ($= \mathbf{y}'$ say) has the maximum loss among all violators in $\overline{\mathcal{Y}}$. Hence we need:

$$\text{s}(\mathbf{y}') + \lambda_u L(\mathbf{y}') \geq \max_{\mathbf{y}\in\overline{\mathcal{Y}},\ L(\mathbf{y})<L(\mathbf{y}')} \text{s}(\mathbf{y}) + \lambda_u L(\mathbf{y})$$

Let $\mathbf{y}_1 \triangleq \text{argmax}_{\mathbf{y}} s(\mathbf{y})$ and L_ε be the minimum difference between two distinct loss values (e.g. $L_\varepsilon = 1$ for Hamming loss). Then the right side can be atmost $\text{s}(\mathbf{y}_1) + \lambda_u(L(\mathbf{y}') - L_\varepsilon)$. So we require $\lambda_u \geq \frac{\text{s}(\mathbf{y}_1)-\text{s}(\mathbf{y}')}{L_\varepsilon}$.

Now, $\mathbf{y}' \in \overline{\mathcal{Y}} \Rightarrow \text{s}(\mathbf{y}') \geq \text{s}(\mathbf{y}_i) - 1 + \frac{\xi}{L(\mathbf{y}')} \geq$

$\mathrm{s}(\mathbf{y}_i) - 1 + \frac{\xi}{L_{max}}$, so we can conservatively set $\lambda_u = \frac{1}{L_\varepsilon}\left(\mathrm{s}(\mathbf{y}_1) - \mathrm{s}(\mathbf{y}_i) + 1 - \frac{\xi}{L_{max}}\right)$.

3.2. Limitation of approximate slack

In the worst case, it is possible that the exact slack MAP $\mathbf{y}^S$ violates the inequality but $\mathbf{y}^A$ does not, as we show next.

Claim 3.3. $s(\mathbf{y}^A) - \frac{\xi}{L(\mathbf{y}^A)} < s(\mathbf{y}_i) - 1 + \epsilon \not\Rightarrow s(\mathbf{y}^S) - \frac{\xi}{L(\mathbf{y}^S)} < s(\mathbf{y}_i) - 1 + \epsilon$

Proof. We prove the claim with a counter example. Let $\mathbf{y}_j, j = 1, 2, 3$ be three labelings with scores $s_j = -\frac{1}{2}, -\frac{13}{18}, -\frac{5}{6}$ and losses $L_j = 1, 2, 3$. Note that $s_1 > s_2 > s_3$ and $L_1 < L_2 < L_3$. Let the score of the true labeling be $s = 0$, the slack be $\xi = \frac{19}{36}$, and let $\epsilon \approx 0$. By computing $\mathrm{sgn}(s_j - \frac{\xi}{L_j} - s + 1 - \epsilon)$, we can see that labelings $\mathbf{y}_1$ and $\mathbf{y}_3$ are not violators but $\mathbf{y}_2$ is. In order to return $\mathbf{y}_2$ as the worst violator, there must exist λ such that $s_2 + \lambda L_2 \geq s_j + \lambda L_j,\ j = 1, 3$. This translates to the constraints $\lambda > \frac{2}{9}$ and $\lambda < \frac{1}{9}$, which are infeasible. □

The above counter example showed that it is impossible to approximate the slack scaled constraint by any method that depends on finding MAP with varying weights on error. This limitation though seemingly restrictive, only slightly hampers the performance in practice, as evident in our experimental results.

4. Position learner

We next propose a new formulation for max-margin training that directly exposes the decomposability of the error function so as to require solving a considerably simpler inference problem. We show that this new formulation not only addresses the computational problem of slack scaling inference, but also provides a more accurate characterization of the loss of scoring functions.

The basic premise of the new learner, which we call PosLearn, is that when error is additive over a set of positions, the loss should also additively reflect margin violations at each possible error position. This is in contrast to both the margin and slack scaling where loss is in terms of a single most violating labeling.

Let $L_i(\mathbf{y}) = \sum_{c \in C} L_{i,c}(\mathbf{y}_c)$ denote a decomposition of the error function. Our goal during training is to ensure that at each possible error position c, the correct labeling has a margin over all labelings where c is incorrectly labeled. If not, we add a hinge loss on the difference in score between the correct labeling $\mathbf{y}_i$ and the best labeling $\mathrm{argmax}_{\mathbf{y}:\mathbf{y}_c \neq \mathbf{y}_{i,c}} \mathbf{w}^T \mathbf{f}(\mathbf{x}_i, \mathbf{y})$ incorrect at c. This yields the following constrained optimization

$$\begin{aligned} \min_{\mathbf{w},\xi} \quad & \frac{1}{2}||\mathbf{w}||^2 + C \sum_{i=1}^{N} \sum_{c} \xi_{i,c} \\ \text{s.t} \quad & \mathbf{w}^T \delta\mathbf{f}_i(\mathbf{y}) \geq 1 - \frac{\xi_{i,c}}{L_{i,c}(\mathbf{y}_c)} \quad \forall \mathbf{y} : \mathbf{y}_c \neq \mathbf{y}_{i,c} \\ & \xi_{i,c} \geq 0 \quad i : 1 \ldots N, \forall c \end{aligned}$$

In the above program, the number of slack variables is equal to the total number of error positions over all instances. Otherwise, the form of the QP is the same as in Section 2.2 and therefore can be solved via similar cutting plane algorithms. For a given position c of an instance i, the most violating constraint is the labeling

$$\mathbf{y}^{P:c} \triangleq \mathrm{argmax}_{\mathbf{y}:\mathbf{y}_c \neq \mathbf{y}_{i,c}} \left(\mathrm{s}_i(\mathbf{y}) - \frac{\xi_{i,c}}{L_{i,c}(\mathbf{y}_c)}\right) \qquad (5)$$

This inference problem can be solved efficiently by any structured learning task in which MAP can be found efficiently since

$$\max_{\mathbf{y}:\mathbf{y}_c \neq \mathbf{y}_{i,c}} \mathrm{s}_i(\mathbf{y}) - \frac{\xi_{i,c}}{L_{i,c}(\mathbf{y}_c)} = \max_{\mathbf{y}_c \neq \mathbf{y}_{i,c}} \left(\max_{\mathbf{y} \sim \mathbf{y}_c} \mathrm{s}_i(\mathbf{y}) - \frac{\xi_{i,c}}{L_{i,c}(\mathbf{y}_c)}\right)$$

where the outer max is over a small number of values as the size of c is typically small and the inner max is MAP inference with label of c constrained to $\mathbf{y}_c$. The MAPs $\mathbf{y}^{P:c}$ will typically be evaluated simultaneously for each c. In many structured learning tasks, all these MAPs can be found in just twice the amount of time it takes to compute a single unrestricted MAP, as we show in Section 4.3.1.

In addition to these computational advantages, PosLearn also provides better loss characterization than slack scaling.

4.1. Comparison with slack scaling

First, we claim that the PosLearn loss is an upper bound of the slack loss.

Claim 4.1. *The slack loss* $\sum_i \max_{\mathbf{y}} L_i(\mathbf{y})[1 - \mathbf{w}^T \delta\mathbf{f}_i(\mathbf{y})]_+$ *is upper bounded by the PosLearn loss* $\sum_i \sum_c \max_{\mathbf{y}:\mathbf{y}_c \neq \mathbf{y}_{i,c}} L_{i,c}(\mathbf{y}_c)[1 - \mathbf{w}^T \delta\mathbf{f}_i(\mathbf{y})]_+$

Proof. Let $\mathbf{y}^S = \mathrm{argmax}_{\mathbf{y} \neq \mathbf{y}_i} L_i(\mathbf{y})[1 - \mathbf{w}^T \delta\mathbf{f}_i(\mathbf{y})]_+$.

$$\begin{aligned} & L_i(\mathbf{y}^S)[1 - \mathbf{w}^T \delta\mathbf{f}_i(\mathbf{y}^S)]_+ \\ & \quad = \sum_c L_{i,c}(\mathbf{y}^S_c)[1 - \mathbf{w}^T \delta\mathbf{f}_i(\mathbf{y}^S)]_+ \\ & \quad \leq \sum_c \max_{\mathbf{y}:\mathbf{y}_c \neq \mathbf{y}_{i,c}} L_{i,c}(\mathbf{y}_c)[1 - \mathbf{w}^T \delta\mathbf{f}_i(\mathbf{y})]_+ \end{aligned}$$

□

Next, we show that slack scaling by defining the total loss in terms of a *single* most violating labeling, cannot discriminate amongst scoring functions as well as the PosLearn loss that involves different labelings at different error positions.

Consider one example where $\mathbf{w}$ is such that three labelings $\mathbf{y}_0 = [0\ 0\ 0\ 0], \mathbf{y}_1 = [1\ 1\ 0\ 0], \mathbf{y}_2 = [0\ 0\ 1\ 0]$, all have the same score of 1. Let $\mathbf{y}_0$ be the correct labeling, then $L(\mathbf{y}_1) = 2, L(\mathbf{y}_2) = 1$, assuming Hamming error. Let the score of all remaining labelings be 0. The total slack loss in this case is 2 whereas the PosLearn loss is 3. Now consider the case where $\mathbf{y}_2$ has score 0. The slack loss remains unchanged whereas PosLearn loss reduces to 2.

An important consequence of the reduced error coverage is that, when the cutting plane algorithm terminates in slack scaling, PosLearn could continue to find violating constraints. The reverse is not true.

4.2. Comparison with M^3N training

The PosLearn program appears similar to the M^3N program of (Taskar, 2004) because both decompose the slack variable over multiple positions. However, the similarity is only superficial. The training objective of M^3N is Margin scaling and the position specific slack variables are for integrating training with inference for loss augmented MAP. In PosLearn the position specific slacks lead to a very different training objective.

4.3. Common decomposable error functions

We show examples of decomposable error functions in several structured learning tasks and show how to efficiently find the most violating constraints over all error positions simultaneously.

4.3.1. Markov models

Many structured prediction tasks can be modeled as Markov models. Popular examples are sequence labeling for information extraction (Lafferty et al., 2001), and grid models for image segmentation (Taskar, 2004; Boykov et al., 2001). A natural error function here is Hamming loss that decomposes over the nodes of the Markov network. Typical MAP inference algorithms based on belief propagation also give max-marginals at each node. The max-marginals gives us at each (node c, label y) pair, the best labeling $\mathbf{y}^{c:y}$ with node c labeled y. We can now find the most violating labeling at each position c via

$$\max_{y \neq \mathbf{y}_{i,c}} (1 - \mathbf{w}^T \delta \mathbf{f}_i(\mathbf{y}^{c:y})) L_{i,c}(y)$$

where $L_{i,c}(y) = 1$ when $y \neq \mathbf{y}_{i,c}$ for Hamming loss. In general $L_{i,c}(y)$ can be any arbitrary real-value, for example a mis-classification matrix $M(y', y)$ could give the cost of misclassifying a y' node as y.

4.3.2. Segmentation

The output space $\mathcal{Y}$ consists of all possible labeled segmentations of an input sequence $\mathbf{x}$. A segmentation $\mathbf{y}$ consists of a sequence of segments $s_1 \ldots s_p$ where each $s_j = (t_j, u_j, y_j)$ with t_j = segment start position, u_j = segment end position, and y_j= segment label. Segmentation models have been proposed as alternative models for information extraction that allows for more effective use of entity-level features (McDonald et al., 2005a; Sarawagi & Cohen, 2004).

The feature vector decomposes over segments and is a function of the segment and the label of the previous segment. Thus $\mathbf{f}(\mathbf{x}, \mathbf{y}) = \sum_{j=1}^{p} \mathbf{f}(\mathbf{x}, s_j, y_{j-1})$. The error function also decomposes over segments as $L_i(\mathbf{y}) = \sum_{s \in \mathbf{y}} L_i(s)$ where for a segment $s = (t, u, y)$, $L_i((t, u, y))$ is defined as

$$L_i((t,u,y)) = \begin{cases} p_y + \sum_{\substack{(l',u',y') \in \mathbf{y}_i \\ t \leq u' \leq u}} r_{y'} & (t,u) \notin \mathbf{y}_i \\ M(y', y) & (t,u,y') \in \mathbf{y}_i \end{cases}$$

where p_y is the precision penalty of labeling a segment as y and $r_{y'}$ is the recall penalty of missing a true segment of label y' and $M(y', y)$ is the misclassification cost matrix applicable when the same span appears in both segmentations.

The number of slack variables is the number of possible segment spans (t, u), which is $O(nm)$ for a sequence of length n and maximum segment size m.

The MAP segmentation can be found using an extension of the Viterbi algorithm (Sarawagi & Cohen, 2004). Viterbi also gives the highest scoring segmentation of the sequence from 1 to i with the last segment ending at i with label y for all possible i and y. Call this $\gamma(i, y)$. Similarly, we can use a backward Viterbi pass to get $\beta(i, y)$ the highest scoring segmentation from $i + 1$ to n with label y on the segment ending at i. These can be combined to find the most violating constraint for a slack variable corresponding to segment (t, u) as:

$$\max_{y', y:(t,u,y) \notin \mathbf{y}_i} L_i((t,u,y))[1 - \mathrm{s}(\mathbf{y}_i) + \gamma(t-1, y') + \mathbf{w}^T \mathbf{f}(\mathbf{x}, (t,u,y), y') + \beta(u, y)]_+$$

4.3.3. Unlabeled dependency parsing

In unlabeled dependency parsing, the goal is to assign each token to its 'head' token (or to a dummy token),

such that the head links form a directed spanning tree. The feature vector for a tree $\mathbf{y}$ over a sentence $\mathbf{x}$ is decomposable over the edges (McDonald et al., 2005b): $\mathbf{f}(\mathbf{x}, \mathbf{y}) = \sum_t \mathbf{f}(y_t, \mathbf{x}, t)$ where t is a token and y_t is its head. A natural error function for a dependency parse tree is then the number of words that are assigned an incorrect head word. In this case, the error and features decompose in exactly the same way, over individual words. The only coupling amongst the predictions of different words is that they need to form a tree.

We use the combinatorial non-projective parsing algorithm of (McDonald et al., 2005b), which cannot be easily extended to simultaneously return MAP for each position. For PosLearn we return the worst violator for each position by first finding the unrestricted MAP $\mathbf{y}^*$. Then, for each position where $\mathbf{y}^*$ is correct, we re-invoke MAP with the correct assignment disabled. In the worse case, this will lead to n MAP invocations.

5. Experiments

We present experimental results on three tasks — sequence labeling, text segmentation and dependency parsing, performed on the following datasets and settings:

CoNLL'03: We use the English benchmark from the CoNLL'03 shared task on named entity recognition. The corpus consists of train, development and test sets of $\approx$ 14000, 3200 and 3400 sentences respectively. We used exactly the same features as in the trained model from Stanford's Named Entity Recognizer [1].

Cora: This is a database of $\approx$ 500 citations (McCallum et al., 2000), containing entities such as Author, Journal, Title, Year and Volume. We used standard extraction features defined over the neighborhoods of each token and the label of the previous token (Peng & McCallum, 2004). For the segmentation task on this dataset, we also used the segment length feature.

Address: This is a collection of $\approx$ 400 non-US postal addresses. Unlike US addresses, these addresses are highly irregular and relatively difficult to segment. The features for sequence labeling and segmentation tasks are as defined in (Sarawagi & Cohen, 2004).

CoNLL-X: We use the freely available treebanks for Swedish, Dutch and Danish from the CoNLL X Shared Task for unlabeled dependency parsing. The training sets contain $\approx$ 11000, 13350, and 5200 sentences respectively. We use the first-order features, the online MIRA trainer (Crammer & Singer, 2003), and the non-projective parsing algorithm provided in the MSTParser package[2].

[1] `http://nlp.stanford.edu/software/stanford-ner-2008-05-07.tar.gz`

Table 1. Token mis-classifications (in %) of all approaches on all tasks. For sequence labeling and segmentation we also report span F1 (after '/').

	Margin	Slack	Approx	PosLearn
		Sequence Labeling		
Cora	12.3/74.9	10.0/82.9	9.9/83.0	**9.5/83.4**
Address	17.1/71.0	15.7/76.7	15.1/78.1	**14.2/78.4**
CoNLL	2.89/84.7	2.95/84.7	2.96/84.6	**2.82/85.1**
		Segmentation		
Cora	17.7/81.8	17.4/81.9	17.3/81.9	**16.2/83.1**
Address	15.4/77.6	15.4/77.5	15.4/77.6	**13.8/79.0**
		Dependency Parsing		
Danish	**12.4**	-	-	12.5
Dutch	**16.3**	-	-	16.9
Swedish	12.9	-	-	**12.8**

5.1. Results

Table 1 shows test errors (as defined in Section 4.3) and Span F1 (where ever applicable) of all four training approaches on all the tasks. For Cora and Address results are averaged over ten splits of 25% train — 75% test, the rest are with the standard training and test files as available in the benchmark. For sequence and segmentation tasks, we are able to solve the Slack inference problem exactly using a quadratic algorithm that finds the MAP for each possible error value. For dependency parsing, it was not easy to find MAP with a pre-specified error. Hence, numbers for Slack scaling methods are missing for this task.

Sequence labeling

We note that the errors go down in the order Margin > Slack > ApproxSlack > PosLearn, and PosLearn achieves $\approx$ 20% error reduction over Margin and 5-10% over Slack. The difference between PosLearn and Margin is statistically significant (p-value from paired t-test is < 0.001), while that between ApproxSlack and Slack is not. This confirms that the approximations done in ApproxSlack are empirically good.

We also reports the entity span F1 values in Table 1 (numbers after the "/"). PosLearn provides signifi-

[2] `http://sourceforge.net/mstparser`

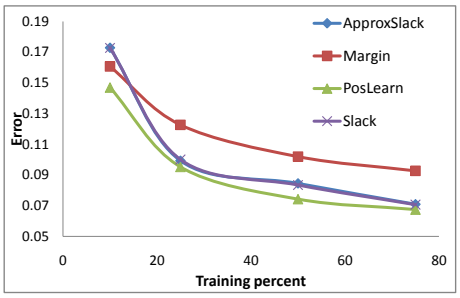

Figure 1. Sequence labeling error (in %) of all approaches on Cora as training size is increased.

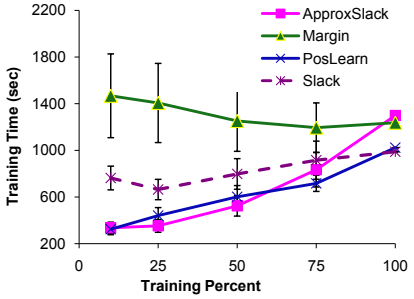

Figure 2. Comparison of training times on Cora over various training percentages. The error bars denote one standard deviation over ten random splits.

cant improvements over Margin for Cora and Address, going from 75 to 83 and 71 to 78 respectively. This shows that optimizing for the error directly translates to significantly better span F1 scores. For CoNLL'03 the gains are modest both for error and Span F1 for reasons we will highlight in Section 5.2. Figure 1 investigates the effect of increasing training size on the sequence labeling errors of all the approaches on Cora. PosLearn remains the best approach for all training sizes, with a 25% error reduction over Margin even for 75% training data. ApproxSlack and Slack are almost identical for all training sizes.

Figure 2 compares the training time of the four approaches on Cora over various training sizes. PosLearn and ApproxSlack turn out to be the cheapest of all the approaches. Two key observations here are (a) PosLearn is up to five times faster than Margin in spite of generating many more constraints, and (b) The training time of Margin reduces with an increase in data. These can be attributed to two reasons. First, PosLearn quickly generates a lot of relevant constraints and terminates in much fewer iterations, whereas Margin spends too much time in separating high loss labelings which are already far enough. Second, when data is scarce, Margin is not able to find good support vectors early on and takes many more iterations. This provides another empirical support for the recent observations in (Bottou & Bousquet, 2008) on the inverse dependence of training time on data sizes.

Segmentation

The results for segmentation are similar to sequence labeling. Again, PosLearn provides 7-10% decrease in error over Margin and Slack. ApproxSlack again turns out to be a close approximation to Slack.

Unlabeled dependency parsing

The difference between PosLearn and Margin turns out to be very insignificant in this case. We cannot evaluate Slack as its MAP inference algorithm is not feasible in this setting. Our discussion in the next section shows that we do not expect ApproxSlack to score over Margin either.

Note our baseline numbers are competitive with the state of the art for these tasks. For Swedish and Danish, the errors for Margin scaling are significantly lower than the average errors of the CoNLL X Shared Task participants — 15.8% and 15.5% respectively . For Dutch, Margin scaling model is better than the best model in the Shared Task (error 16.4%).

5.2. Discussion

We observed that Margin scaling was significantly worse than other loss functions for tasks like sequence labeling on the Address and Cora datasets, while being the highest performing on tasks like dependency parsing. We explain the reasons behind the varying gains of Margin relative to other loss functions, in particular PosLearn, based on the decomposition of the error function compared to the feature function.

We argue that margin scaling is a bad loss function only when the model comprises of features that strongly couple larger subset of variables than the error function. Consider the case when the feature function decomposes over each position of $\mathbf{y}$, exactly as in the error function. This is true for dependency parsing, and for sequence labeling models with no edge features. In such cases, a structured formulation adds little value, and a multi-class SVM with independent constraints over the local features and loss at each position, is just as adequate. The constraints of structured margin scaling turn out to be a summation of the constraints of multi-class SVM and the two solve equivalent objectives as shown in (Joachims, 2006). Interestingly, (McDonald et al., 2005b) indeed finds that such a model (which they call the factored model) is very close to the structured model using margin scal-

ing. We verify that for sequence labeling, if we disable all edge features, then for the Address dataset, span F1 drops from 71 to 62 and for Cora from 75 to 44 with Margin scaling. This indicates the strong importance of structured features for these datasets. In contrast, for CoNLL'03 where Margin is competitive with PosLearn, removal of edge features causes only a small drop in Span F1, from 84.7 to 81. Without edge features, PosLearn shows little or negative improvement over Margin scaling for all three datasets.

This indicates that in domains where the feature function does not induce strong coupling amongst variables, there is no reward in going beyond simple margin scaling, and possibly even multiclass SVMs. In truly structured problems where features strongly couple multiple variables, margin scaling gets adversely affected by the unnecessary margin requirements of high error labelings due to shared slack variables. PosLearn ignores labelings separated from the margin, and by defining per-position slacks instead of a single shared slack, handles such structured cases better.

6. Conclusion

We presented an efficient variational approximation to the slack scaling approach, which only requires a slightly modified loss augmented MAP algorithm, instead of the inefficient slack scaling inference algorithm. We demonstrated that in practice it performs much better than margin scaling and closely approximates slack scaling.

Next, we argued that all existing approaches that define loss in terms of a single most violating labeling achieve inadequate separation from the correct labeling. We proposed a new trainer, PosLearn that involves multiple labelings in trying to ensure max-margin separation at each possible error position in the structured output. The PosLearn constraints can be generated using only the MAP algorithm, and for many structured models the time required is no more than twice the time taken to find MAP. Empirically, this leads to significant error reduction over Margin scaling on structured models that induce strong coupling amongst output variables.

A compelling future direction is theoretically analyzing the generalizability of PosLearn vis-a-vis other loss scaling methods.

References

Bordes, A., Bottou, L., Gallinari, P., & Weston, J. (2007). Solving multiclass support vector machines with larank. *ICML* (pp. 89–96).

Bottou, L., & Bousquet, O. (2008). The tradeoffs of large scale learning. *NIPS*.

Boykov, Y., Veksler, O., & Zabih, R. (2001). Fast approximate energy minimization via graph cuts. *IEEE Trans. Pattern Anal. Mach. Intell.*, *23*, 1222–1239.

Crammer, K., & Singer, Y. (2003). Ultraconservative online algorithms for multiclass problems. *J. Mach. Learn. Res.*, *3*, 951–991.

Joachims, T. (2006). Training linear SVMs in linear time. *KDD*.

Jordan, M. I., Ghahramani, Z., Jaakkola, T. S., & Saul, L. (1999). An introduction to variational methods for graphical models. In M. I. Jordan (Ed.), *Learning in graphical models*. MIT Press.

Lafferty, J., McCallum, A., & Pereira, F. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data. *Proceedings of the International Conference on Machine Learning (ICML-2001)*. Williams, MA.

LeCun, Y., Chopra, S., Hadsell, R., Marc'Aurelio, R., & Huang, F. (2006). A tutorial on energy-based learning. *Predicting Structured Data*. MIT Press.

McCallum, A., Nigam, K., Reed, J., Rennie, J., & Seymore, K. (2000). Cora: Computer science research paper search engine. http://cora.whizbang.com/.

McDonald, R., Crammer, K., & Pereira, F. (2005a). Flexible text segmentation with structured multilabel classification. *HLT/EMNLP*.

McDonald, R., Crammer, K., & Pereira, F. (2005b). Online large-margin training of dependency parsers. *ACL* (pp. 91–98).

Peng, F., & McCallum, A. (2004). Accurate information extraction from research papers using conditional random fields. *HLT-NAACL* (pp. 329–336).

Ratliff, N., Bagnell, J., & Zinkevich, M. (2007). (online) subgradient methods for structured prediction. *AIStats*.

Sarawagi, S., & Cohen, W. W. (2004). Semi-markov conditional random fields for information extraction. *NIPS*.

Taskar, B. (2004). *Learning structured prediction models: A large margin approach*. Doctoral dissertation, Stanford University.

Taskar, B., Klein, D., Collins, M., Koller, D., & Manning, C. (2004). Max-margin parsing. *EMNLP*.

Taskar, B., Lacoste-Julien, S., & Jordan, M. I. (2006). Structured prediction, dual extragradient and bregman projections. *J. Mach. Learn. Res.*, *7*, 1627–1653.

Tsochantaridis, I., Joachims, T., Hofmann, T., & Altun, Y. (2005). Large margin methods for structured and interdependent output variables. *Journal of Machine Learning Research (JMLR)*, *6(Sep)*, 1453–1484.