Message Passing for Soft Constraint Dual Decomposition

David Belanger
UMass Amherst
belanger@cs.umass.edu

Alexandre Passos
UMass Amherst
apassos@cs.umass.edu

Sebastian Riedel
University College London
s.riedel@ucl.ac.uk

Andrew McCallum
UMass Amherst
mccallum@cs.umass.edu

Abstract

Dual decomposition provides the opportunity to build complex, yet tractable, structured prediction models using linear constraints to link together submodels that have available MAP inference routines. However, since some constraints might not hold on every single example, such models can often be improved by relaxing the requirement that these constraints always hold, and instead replacing them with soft constraints that merely impose a penalty if violated. A dual objective for the resulting MAP inference problem differs from the hard constraint problem's associated dual decomposition objective only in that the dual variables are subject to box constraints. This paper introduces a novel primal-dual block coordinate descent algorithm for minimizing this general family of box-constrained objectives. Through experiments on two natural language corpus-wide inference tasks, we demonstrate the advantages of our approach over the current alternative, based on copying variables, adding auxiliary submodels and using traditional dual decomposition. Our algorithm performs inference in the same model as was previously published for these tasks, and thus is capable of achieving the same accuracy, but provides a 2-10x speedup over the current state of the art.

1 INTRODUCTION

We often need complex structured prediction models that encode rich global and local dependencies and constraints among the outputs, but this can render efficient prediction difficult. Therefore, *dual decomposition* is quite useful, since it enables efficient inference in models composed of various submodels with available black-box MAP inference routines (Komodakis *et al.*, 2007; Sontag *et al.*, 2011; Rush & Collins, 2012).

In some cases, the flexibility and robustness of such models can be improved by using *soft constraints*, where the model imposes a cost if a constraint is violated, but does not require that it is satisfied. In natural language processing, for example, soft constraints have enabled accuracy gains for named entity recognition (Finkel *et al.*, 2005; Sutton & McCallum, 2006), parsing (Smith & Eisner, 2008; Rush *et al.*, 2012), and citation field segmentation (Chang *et al.*, 2012; Anzaroot *et al.*, 2014). Using soft constraints is reasonable in these applications because the constraints are not required in order to define feasible outputs, but are instead a modeling layer imposed to improve predictive accuracy. Soft constraints are advantageous over hard constraints because they allow the model to trade off evidence for and against a constraint being satisfied.

In all of these examples besides Rush *et al.* (2012) and Anzaroot *et al.* (2014), inference is performed using standard techniques for inference in loopy graphical models such as belief propagation or MCMC. However, these have poor optimality guarantees and can also be difficult to generalize to prediction problems that are not graphical models. An alternative method for handling soft constraints is to make copies of variables participating in soft constraints, constrain each variable to equal its copy, and apply dual decomposition (Rush *et al.*, 2012). While this exhibits better flexibility, scalability, and guarantees, it requires inference in auxiliary submodels and copying variables prevents the feasibility of the output during intermediate iterations before convergence, since the two copies of a variable may have different values.

Recently, Anzaroot *et al.* (2014) employed an attractive alternative algorithm for performing MAP subject to soft constraints that offers the optimality guarantees and generality of dual decomposition, but avoids variable copying and auxiliary models completely. Their algorithm requires an extremely straightforward modification to existing dual decomposition objectives: if the model penalizes the violation of a constraint with a penalty of c, then the dual variable is subject to a *box constraint*, where it can not exceed c. They minimize this objective with projected subgradient descent.

While this projected subgradient algorithm is simple, its convergence can be slow and sensitive to a choice of step size schedule. On the other hand, block coordinate descent algorithms, such as MPLP (Globerson & Jaakkola, 2007), are parameter-free and often converge much faster than subgradient descent for dual decomposition objectives, subject to our ability to obtain *max-marginals* from the subproblems (Sontag *et al.*, 2011).

In response, we contribute the following:

1. An extension of the projected subgradient algorithm of Anzaroot *et al.* (2014) to general pairwise soft constraints (Section 5) that are capable of modeling arbitrary pairwise graphical model factors (Section 8).
2. An adaptation of the MPLP algorithm beyond graphical models to alternative structured prediction problems with certain structure (Section 6).
3. Box-MPLP, a primal-dual message passing algorithm for solving the box-constrained dual decomposition objective for soft constraints (Section 7). Its update rule and derivation differ substantially from MPLP.
4. Experiments on two corpus-wide prediction tasks from natural language processing (Section 2) demonstrating both the advantages of using Box-MPLP v.s. projected subgradient and of using a box-constrained dual objective v.s. variable copying and hard-constraint dual decomposition (Section 10).

2 CORPUS-WIDE INFERENCE

We first motivate the use of soft constraints by describing the application that we will explore in our experiments. In natural language processing, it is common to part-of-speech (POS) tag and dependency parse every sentence in a corpus of documents. Both tasks can be posed as efficient MAP inference, but a drawback of these algorithms is that they process each sentence in isolation, despite the fact that there is discriminative information shared across the corpus. In response, Rush *et al.* (2012) performed *corpus-wide inference*. Specifically, for word types that did not appear in the training data, they introduced global model terms that encouraged every occurrence of the word in the test corpus to receive the same POS tag, or to be assigned a dependency parent with the same POS tag. A similar model appeared in Chieu & Teow (2012).

Rush *et al.* (2012) model these cross-sentence relationships among sets of occurrences that are encouraged to agree, by introducing one *consensus structure*, described in the Figure 1 caption, per set. There is a soft constraint between every variable at the bottom of the consensus set, and the one at the top. If the underlying sentence-level models are graphical models, the corpus-wide inference problem could be posed as a large loopy graphical model and we can perform approximate MAP using standard techniques. An alternative solution, depicted in Figure 2, is to copy variables that participate in consensus sets, introduce an auxiliary tree-structured subproblem, and use dual decomposition for corpus-wide MAP. This has superior optimality guarantees and flexibility to use sentence-level problems that are not graphical models. In practice, this algorithm can be slow to converge, however. In response, we introduce a new approach for performing MAP subject to soft constraints that when applied to corpus-wide inference allows us to work directly in the soft constraint problem of Figure 1, yet yields the same flexibility and optimality guarantees as Rush *et al.* (2012) and substantially faster runtimes. The techniques are general and apply to a wide range of additional applications.

Figure 1: One *consensus set*. The circles at the bottom represent words of the same type, and the boxes represent arbitrary sentence-level prediction problems that they are contained in. The circle at the top is a *consensus variable* introduced to encourage consensus among the bottom circles, where the squares are soft constraints penalizing disagreement. The corpus is linked together by a web of consensus structures.

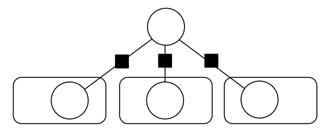
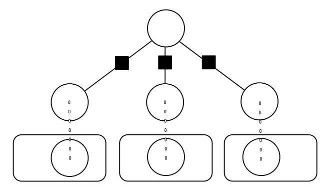


Figure 2: The variable-copying version of Fig. 1, where dashed lines denote equality constraints.



3 NOTATION AND STRUCTURED LINEAR MODELS

Bold-faced lower-case letters, such as $\mathbf{x}$, represent column vectors, and bold-faced upper case letters, such as $\mathbf{A}$, represent matrices. The i-th coordinate of vector $\mathbf{x}$ is $\mathbf{x}(i)$ and the i, jth coordinate of a matrix is $\mathbf{A}$ is $\mathbf{A}(i, j)$. Lower-case greek letters such as $\boldsymbol{\lambda}$ represent either vector-valued or matrix-valued dual variables. We use $\mathbf{x}^{(t)}$ for $\mathbf{x}$ at iteration t. The term 'constraint' either refers to a constraint between scalars or a set of coordinate-wise constraints between vectors (or matrices). In the latter case, the associated dual variable is a vector (or matrix).

We consider structured prediction problems defined by *structured linear models* such as conditional random fields (Lafferty *et al.*, 2001) and maximum spanning tree parsers (McDonald *et al.*, 2005). These assign a score to each possible output labeling by decomposing each candidate output into a collection of *parts*, each of which can be active or inactive in a given labeling. For example, in first-order dependency parsing, each part corresponds to a single dependency arc (Smith, 2011). In a conditional random field, there is a part for each possible setting of each clique.

We write the indicator vector for the parts of a specific labeling of a datacase k as $\mathbf{x}_k$. It is a binary vector with one coordinate per possible part, which is zero if the part is not present in the structured output and one if it is. The model for candidate outputs is called *linear* because the score of a given labeling is the dot product $\langle \mathbf{w}_k, \mathbf{x}_k \rangle$ of a weight vector $\mathbf{w}_k$ and the indicator vector over the parts. In many models, such as conditional random fields, the score of each part is a function of some observed features, and in many cases this mapping from features to weights is also linear. We focus only on inference, however, and make no assumptions about how the weights are set. In non-trivial structured linear models, not all assignments of values to these parts are valid, since they typically represent some over-complete view of the structured output or are subject to global structural constraints, such as projectivity for dependency parsing (Smith, 2011). For an instance k we refer to the set of valid assignments to parts as $\mathcal{U}_k$.

We refer to the problem of finding the highest-scoring valid collection of parts as *MAP inference*:

$$\max_{\mathbf{x}_k} \quad \langle \mathbf{w}_k, \mathbf{x}_k \rangle \quad \text{s.t.} \quad \mathbf{x}_k \in \mathcal{U}_k.$$

4 DUAL DECOMPOSITION

Following Sontag *et al.* (2011); Rush & Collins (2012); Komodakis *et al.* (2007), we consider the problem:

$$\begin{aligned} \max_{\mathbf{x}} \quad & \sum_k \langle \mathbf{w}_k, \mathbf{x}_k \rangle && (1)\\ \textbf{s.t.} \quad & \forall k \quad \mathbf{x}_k \in \mathcal{U}_k && (2)\\ & \sum_k \mathbf{A}_k \mathbf{x}_k = 0, && (3) \end{aligned}$$

where each $\mathbf{x}_k$ represents the vector of parts for a specific structured linear 'submodel.' The formulation can easily be adapted to account for a nonzero right hand side of (3). If (3) did not exist, the problem would reduce to independent MAP inference in each subproblem.

Dualizing the linear constraints in (3), but not the $\mathbf{x}_k \in \mathcal{U}_k$ constraints, results in the Lagrange dual problem:

$$\min_{\boldsymbol{\lambda}} D(\boldsymbol{\lambda}) = \sum_k \max_{\mathbf{x}_k \in \mathcal{U}_k} \left\langle \mathbf{w}_k + \mathbf{A}_k^T \boldsymbol{\lambda}, \mathbf{x}_k \right\rangle. \qquad (4)$$

Algorithm 1 Dual Decomposition with Subgradient Descent

$\boldsymbol{\lambda} \leftarrow \mathbf{0}$
while has not converged **do**
  for submodel i **do**
    $\mathbf{x}_k^* \leftarrow \max_{\mathbf{x}_k \in \mathcal{U}_k} \left\langle \mathbf{w}_k + \mathbf{A}_k^T \boldsymbol{\lambda}, \mathbf{x}_k \right\rangle$
  $\boldsymbol{\lambda} \leftarrow \boldsymbol{\lambda} - \eta^{(t)} \sum_k \mathbf{A}_k \mathbf{x}_k^*$

The dual objective $D(\boldsymbol{\lambda})$ is convex and piece-wise linear, as it is the sum of the supremum of linear functions of $\boldsymbol{\lambda}$, and hence can be solved with known convex optimization techniques, including subgradient methods. Any particular element of the subgradient of the dual function with respect to $\boldsymbol{\lambda}$ can be written as

$$\partial D(\boldsymbol{\lambda}) = \sum_k \mathbf{A}_k \mathbf{x}_k^*, \qquad (5)$$

where each $\mathbf{x}_k^*$ is some maximizer of a MAP inference problem with shifted weights:

$$\mathbf{x}_k^* \in \operatorname*{argmax}_{\mathbf{x}_k \in \mathcal{U}_k} \left\langle \mathbf{w}_k + \mathbf{A}_k^T \boldsymbol{\lambda}, \mathbf{x}_k \right\rangle. \qquad (6)$$

We consider cases, where these MAP subproblems are tractable and solving their linear programming relaxations returns an integral value for any weight vector. Therefore, one can use subgradient descent, Algorithm 1, to minimize the dual problem. Subject to conditions on the sequence of step sizes $\eta^{(t)}$ and the feasibility of the constraints that link the subproblems, the subgradient method is guaranteed to converge to the optimum, where (3) will be satisfied (Nesterov, 2003; Sontag *et al.*, 2011).

5 SOFT DUAL DECOMPOSITION

5.1 PROBLEM STATEMENT

This paper focuses on applications of dual decomposition where the underlying prediction problem has at least two distinct sets of outputs $\mathbf{x}_1 \in \mathcal{U}_1$ and $\mathbf{x}_2 \in \mathcal{U}_2$, and linear constraints are imposed between them not as a requirement to define feasible outputs, but as an extra layer of modeling to encourage global regularity of the outputs. This contrasts with problems with a single output $\mathbf{x}$ subject to the linear constraints $\mathbf{x} \in \mathcal{U}_1 \cap \mathcal{U}_2$, and while these are unmanageable directly, $\mathcal{U}_1$ and $\mathcal{U}_2$ can each be handled in isolation. Here, dual decomposition can be employed via a copy variable $\mathbf{x}_2$, and constraints $\mathbf{x} \in \mathcal{U}_1$, $\mathbf{x}_2 \in \mathcal{U}_2$, and $\mathbf{x}_1 = \mathbf{x}_2$ (Koo *et al.*, 2010; Rush & Collins, 2012). The first family is precisely where it can make sense to employ soft constraints, since they will not threaten the output's feasibility.

Anzaroot *et al.* (2014) recently performed MAP with soft constraints by performing projected gradient descent in a box-constrained dual objective. Our message passing algorithm requires using a slightly more restrictive set of global constraint structures to be converted into soft constraints than what they considered, which are of the form (3). Specifically, we assume the global constraints decompose into sets of pairwise equality constraints between components of submodels:

$$\max_{\mathbf{x}} \quad \sum_k \langle \mathbf{w}_k, \mathbf{x}_k \rangle \tag{7}$$
$$\mathbf{s.t.} \quad \forall k \quad \mathbf{x}_k \in \mathcal{U}_k \tag{8}$$
$$\forall (\mathbf{A}_p, \mathbf{B}_p, p_1, p_2) \in \mathcal{P} \quad \mathbf{A}_p \mathbf{x}_{p_1} = \mathbf{B}_p \mathbf{x}_{p_2}. \tag{9}$$

A given product $\mathbf{A}_p \mathbf{x}_{p_1}$ or $\mathbf{B}_p \mathbf{x}_{p_2}$ is allowed to appear in multiple $p \in \mathcal{P}$, so $\mathcal{P}$ effectively defines a collection of linear measurements of the structured output and a graph of equality constraints among them. These can be defined over differently-size mapping matrices. Define s_p to be the length of the vector $\mathbf{A}_p \mathbf{x}_{p_1}$ (also the length of $\mathbf{B}_p \mathbf{x}_{p_2}$).

Defining a dual variable $\boldsymbol{\lambda}_p \in \mathbb{R}^{s_p}$ for every $p \in \mathcal{P}$, we have the following convex dual decomposition objective:

$$\sum_k \max_{x_k} \left\langle \mathbf{w}_k + \sum_{p:p_1=k} \mathbf{A}_p^T \boldsymbol{\lambda}^p - \sum_{p:p_2=k} \mathbf{B}_p^T \boldsymbol{\lambda}^p, \mathbf{x}_k \right\rangle . \tag{10}$$

A soft constraint formulation of (7) with penalty matrices $\mathbf{c}_p \in \mathbb{R}^{s_p \times s_p}$ subtracts a penalty of $\mathbf{c}_p(i,j)$ from the score of the global MAP problem whenever $\mathbf{A}_p \mathbf{x}_{p_1}$ is set to value i and $\mathbf{B}_p \mathbf{x}_{p_2}$ is *not* set to value j. In the subsequent exposition, we leave the constraints $\mathbf{x}_k \in \mathcal{U}_k$ implicit, since we assume we have available black-box algorithms for maximizing over these constraint sets. Therefore, we have:

$$\max_{\mathbf{x}} \sum_k \langle \mathbf{w}_k, \mathbf{x}_k \rangle - \sum_p \sum_{i,j} \mathbf{c}_p(i,j) \left[\mathbf{A}_p \mathbf{x}_{p_1}(i) - \mathbf{B}_p \mathbf{x}_{p_2}(j) \right]_+ \tag{11}$$

where $[\cdot]_+ = \max(0, \cdot)$. Using a matrix-valued penalty is important in order to support a mapping between arbitrary graphical model factors and soft constraints (see Section 8). In Section 7.1, we consider diagonal $\mathbf{c}_p$, which are sufficient for the model to penalize when certain components of the structured output do not take on the same value.

An alternative to (11) for expressing soft constraints is to create copies of both of the terms appearing in each $p \in \mathcal{P}$ and enforce the constraints that terms equal their copy:

$$\max_{\mathbf{x}} \quad \textstyle\sum_k \langle \mathbf{w}_k, \mathbf{x}_k \rangle - \sum_p \sum_{i,j} \mathbf{c}_p(i,j) \left[\mathbf{v}_p(i) - \mathbf{u}_p(j) \right]_+$$
$$\mathbf{s.t.} \quad \forall p \in \mathcal{P} \quad \mathbf{A}_p \mathbf{x}_{p_1} = \mathbf{v}_p, \mathbf{B}_p \mathbf{x}_{p_2} = \mathbf{u}_p. \tag{12}$$

Here, the second term is not a structured linear model, but it is concave, can be handled efficiently in isolation, and has integral optima. Therefore, we can apply standard dual decomposition techniques. In Figure 2, we demonstrate how Rush *et al.* (2012) similarly use variable copying to make MAP tractable with dual decomposition. Rather than employing pairwise hinge losses as auxiliary submodels, they introduce a single tree-structured graphical model with pairwise factors that encourage agreement. In Section (10) we use this as a baseline to demonstrate the deficiencies of using variable copying to implement soft constraints.

5.2 DUAL OBJECTIVE AND BOX CONSTRAINTS

Problem (11) can be rewritten as a linear program by introducing matrices of auxiliary variables $\mathbf{z}_p \in \mathbb{R}^{s_p \times s_p}$:

$$\max_{\mathbf{x},\mathbf{z}} \quad \textstyle\sum_k \langle \mathbf{w}_k, \mathbf{x}_k \rangle - \sum_p \sum_{i,j} \mathbf{c}_p(i,j) \mathbf{z}_p(i,j) \tag{13}$$
$$\mathbf{s.t.} \quad \forall (i,j), \quad \mathbf{z}_p(i,j) \geq \mathbf{A}_p \mathbf{x}_{p_1}(i) - \mathbf{B}_p x_{p_2}(j) \tag{14}$$
$$\mathbf{z}_p \geq 0$$

This problem is well-defined only if $\mathbf{c}_p$ is non-negative in every coordinate. In this case, we have that problems (11) and (13) have the same optimal value and maximizing $\mathbf{x}$.

We defer a full derivation of the associated Lagrange dual problem for (13) to Appendix 1, since it parallels Anzaroot *et al.* (2014). The dual is similar to (10) , but imposes coordinate-wise box constraints:

$$\min_{\boldsymbol{\nu}} \quad \sum_k \max_{x_k} \left\langle \mathbf{w}_k + \sum_{p:p_2=k} \mathbf{B}_p^T \boldsymbol{\nu}_p^T \mathbf{1} - \sum_{p:p_1=k} \mathbf{A}_p^T \boldsymbol{\nu}_p \mathbf{1}, \mathbf{x}_k \right\rangle$$
$$\mathbf{s.t.} \quad 0 \leq \boldsymbol{\nu}_p \leq \mathbf{c}_p. \tag{15}$$

Unlike for hard constraints, we have a matrix-valued dual variable $\nu_p \in \mathbb{R}_+^{s_p \times s_p}$ for every $p \in \mathcal{P}$, where $\nu_p(i,j)$ corresponds to the constraint in (14) for a particular (i,j), and $\mathbb{R}_+$ denotes the non-negative real numbers. We use $\mathbf{1}$ to be a column vector of all ones, where its length is determined by the context.

These box constraints exist for the same reason that they occur in the dual problem for soft-margin SVMs (Cortes & Vapnik, 1995), since the second term in (11) is a sum of negative hinge losses. The box constraints on the dual variables $\boldsymbol{\nu}$ can be interpreted as the Lagrangian penalizing the violation of constraints, but only so much as the primal problem would penalize their violation.

The only qualitative difference between the dual problems in (15) and (4) is the box constraints. Therefore, we can employ the projected subgradient method, shown in Algorithm 2, which will converge to the global MAP optimum if $\mathcal{P}$ is feasible. At the end of Appendix 1, we derive the following complementary slackness criteria used for detecting convergence. These will hold for every $p \in \mathcal{P}$ and every coordinate pair (i,j) when maximizing over the primal variables:

Algorithm 2 Projected subgradient soft dual decomposition for general matrix-valued soft constraint penalties.

$\boldsymbol{\nu} \leftarrow \mathbf{0}$
while has not converged **do**
  for submodel k **do**
    $\tilde{\mathbf{w}}_k \leftarrow \mathbf{w}_k + \sum_{p:p_2=k} \mathbf{B}_p^T \boldsymbol{\nu}_p^T \mathbf{1} - \sum_{p:p_1=k} \mathbf{A}_p^T \boldsymbol{\nu}_p \mathbf{1}$
    $\mathbf{x}_k^* \leftarrow \max_{\mathbf{x}_k \in \mathcal{U}_k} \langle \tilde{\mathbf{w}}_k, \mathbf{x}_k \rangle$
  for soft constraint $p \in \mathcal{P}$ **do**
    $\boldsymbol{\nu}^p(i,j) \leftarrow \min(\mathbf{c}_p(i,j), \max(0, \boldsymbol{\nu}_p(i,j) - \eta^{(t)}(\mathbf{A}_p \mathbf{x}_{p_1}^*(i) - \mathbf{B}_p \mathbf{x}_{p_2}^*(j))))$

$$\begin{aligned} &\text{either } \mathbf{A}_p \mathbf{x}_{p_1}^*(i) = \mathbf{B}_p \mathbf{x}_{p_2}^*(j) \\ &\text{or } \mathbf{A}_p \mathbf{x}_{p_1}^*(i) = 1 \text{ and } \boldsymbol{\nu}_p(i,j) = 0 \\ &\text{or } \mathbf{A}_p \mathbf{x}_{p_1}^*(i) = 0 \text{ and } \boldsymbol{\nu}_p(i,j) = \mathbf{c}_p(i,j). \end{aligned} \tag{16}$$

6 MAX-MARGINALS AND MPLP

Using the subgradient method in Algorithm 2 is undesirable due to its sensitivity to step-size schedule and slow convergence in practice. In response, we now revisit hard-constraint dual objectives of the form (10) in order to explore previous use of block coordinate descent, which is parameter-free. We introduce an adaptation of the MPLP algorithm (Globerson & Jaakkola, 2007) to problems with general structured linear models as subproblems, and emphasize a primal-dual interpretation of the algorithm's updates, which we will draw on when we derive our new algorithm in the following section.

MPLP is a convergent alternative to max-product belief propagation that was shown in Sontag *et al.* (2011) to be performing block coordinate descent in a dual decomposition objective for a certain instance of (10). Specifically, there is a submodel for every node and every factor in a factor graph, and an element $p \in \mathcal{P}$ between every node and every factor that it touches. MPLP generalizes to additional cases (10) when the elements of $\mathcal{P}$ satisfy the following condition, and when the subproblems admit efficient computation of max-marginals, defined below.

Definition Let e_j denote the vector that is all zeros, except for a one in the jth coordinate. We say that the product $\mathbf{A}\mathbf{x}_k$ is a *projection variable* if it satisfies the following property:

$$\forall \mathbf{x}_k \in \mathcal{U}_k,\ \exists j\ s.t. \quad \mathbf{A}\mathbf{x}_k = e_j. \tag{17}$$

Unlike the previous subgradient algorithms, MPLP requires every element of $\mathcal{P}$ to be defined between projection variables, which can be used to represent any set of mutually-exclusive states of the structured output. This is not a strong restriction, as they can be used, for example, to zoom in on a specific graphical model node or dependency parse arc and to optionally further coarsen the values of these individual outputs. Also, the hinge loss of the previous section and 0-1 loss are equivalent for projection variables, so we are truly penalizing the event that a constraint is violated, and not imposing a linear penalty on the degree to which it is violated. Defining projection variables is necessary because MPLP requires max-marginals, and the following definition is only well-posed for projection variables:

Definition For a given projection variable $\mathbf{A}\mathbf{x}_k$ and weight vector $\mathbf{w}$, the max-marginals $\mathbf{m}_{\mathbf{w}}^A$ are a vector where the jth component is given by best possible score achievable by a valid structured output when the projection variable takes on value j, i.e.,

$$\mathbf{m}_{\mathbf{w}}^{\mathbf{A}}(j) = \max_{\mathbf{x_k} \in \mathcal{U}_k} \langle \mathbf{w}, \mathbf{x}_k \rangle \quad \text{s.t. } \mathbf{A}\mathbf{x}_k = e_j. \tag{18}$$

For a MAP assignment $\mathbf{x}^*$ with respect to $\mathbf{w}$, we have

$$\mathbf{A}\mathbf{x}^* = e_{i^*}, \text{ where } i^* = \operatorname*{argmax}_i \mathbf{m}_{\mathbf{w}}^A(i). \tag{19}$$

In other words, locally maximizing max-marginals is equivalent to finding a globally-optimal value (unless there are ties in the max-marginals).

Furthermore, max-marginals change linearly with respect to changes to $\mathbf{w}$ in the direction of their projection variable:

$$\mathbf{m}_{\mathbf{w}+\mathbf{A}^T\alpha}^A(i) = \mathbf{m}_{\mathbf{w}}^A(i) + \alpha(i). \tag{20}$$

For example, if we shift the scores for a given factor in a graphical model by a vector α, and otherwise leave the model's potentials unchanged, then the max-marginals for this factor increase by exactly α. This fact, proven in Appendix 2, applies to arbitrary projection variables, and is crucial in deriving both MPLP and our new algorithm in the next section.

In Algorithm 3, we consider a version of MPLP where block coordinate descent is performed by iteratively selecting an element $p \in \mathcal{P}$ and updating the vector-valued dual variable $\boldsymbol{\lambda}_p$. Note this differs from the algorithms in Globerson & Jaakkola (2007) and Sontag *et al.* (2011) slightly because we pass messages (i.e., dual variables) directly between submodels, rather than from submodels to primal variables and from primal variables to submodels. This results from the fact that we pose (10) via equality constraints between different parts of the structured output, not between variables and their copies (Werner, 2008).

We discuss the optimality of this choice of $\boldsymbol{\lambda}_p$ in more detail in Appendix 3, which presents a different primal-dual argument than Sontag *et al.* (2011), in order to motivate the techniques used by the new algorithm that we will introduce later. The high level idea is to invoke (20) to observe that the chosen value for $\boldsymbol{\lambda}_p$ shifts the subproblems'

Algorithm 3 An adaptation of the MPLP algorithm of Sontag *et al.* (2011) to dual decomposition with pairwise constraints between general structured linear submodels.

1: $\boldsymbol{\lambda} \leftarrow \mathbf{0}$
2: converged $\leftarrow false$
3: **while** (! converged) and (iteration < maxIterations) **do**
4: converged $\leftarrow true$
5: **for** equality constraint $p \in \mathcal{P}$ **do**
6: $\tilde{w}_{p_1} \leftarrow \mathbf{w}_{p_1} + \sum_{\substack{p': p'_1 = p_1 \\ p' \neq p}} \mathbf{A}_{p'}^T \boldsymbol{\lambda}_{p'} - \sum_{\substack{p': p'_2 = p_1 \\ p' \neq p}} \mathbf{B}_{p'}^T \boldsymbol{\lambda}_{p'}$
7: $\mathbf{m}_1 \leftarrow \text{MaxMargs}\left(\tilde{w}_{p_1}\right)$
8: $\tilde{w}_{p_2} \leftarrow \mathbf{w}_{p_2} + \sum_{\substack{p': p'_1 = p_2 \\ p' \neq p}} \mathbf{A}_{p'}^T \boldsymbol{\lambda}_{p'} - \sum_{\substack{p': p'_2 = p_2 \\ p' \neq p}} \mathbf{B}_{p'}^T \boldsymbol{\lambda}_{p'}$
9: $\mathbf{m}_2 \leftarrow \text{MaxMargs}\left(\tilde{w}_{p_2}\right)$
10: **if** $(\text{argmax}_i\, \mathbf{m}_1(i) \cap \text{argmax}_i\, \mathbf{m}_2(i) = \emptyset)$ **then**
11: converged $\leftarrow false$
12: $\boldsymbol{\lambda}_p \leftarrow \frac{1}{2}\left(\mathbf{m}_1 - \mathbf{m}_2\right)$

weights such that max-marginals for the two projection variables in p become identical in all coordinates. Therefore, with this setting of the dual variables, it is feasible to achieve the equality $\mathbf{A}_p \mathbf{x}_{p_1} = \mathbf{B}_p \mathbf{x}_{p_2}$ when maximizing over the primal variables. As a result, by strong duality, the dual of (7) is minimized with respect to $\boldsymbol{\lambda}_p$, since the primal constraints for this block are satisfied. Algorithm 3 monitors convergence by checking if all constraints are satisfied when maximizing over the primal variables. See Sontag *et al.* (2011) for a discussion of the convergence guarantees of MPLP and Meshi *et al.* (2012) for its convergence rate.

The algorithm may require multiple passes to converge, since updates for one $\boldsymbol{\lambda}_p$ may break the above optimality condition for other $p \in \mathcal{P}$. Furthermore, every time the dual variables are updated for some $p \in \mathcal{P}$, max-marginals need to be recalculated for subproblems p_1 and p_2. MPLP, and the algorithm in the next section, can not be applied for constraints between projection variables in the same submodel, since their max-marginals interact with each other. Therefore, it could not have been applied in the hard constraint experiments of Anzaroot *et al.* (2014), since they impose constraints within a chain-structured graphical model.

7 MESSAGE PASSING FOR SOFT CONSTRAINT DUAL DECOMPOSITION

We now introduce the primary contribution of the paper: a general dual block coordinate descent framework for minimizing the box-constrained dual objective (15) and Box-MPLP, a novel algorithm for solving a common special case of the problem. Naively applying the MPLP updates may violate the box constraints, and we can not simply follow them with a projection step, as this will not guarantee a decrease in the dual objective.

Analogous to Algorithm (3), our block coordinate descent steps update one vector $\boldsymbol{\nu}_p$ at a time. Since we now focus on a specific $p \in \mathcal{P}$, we define $\mathbf{y}_1 := \mathbf{A}_p \mathbf{x}_{p_1}$ $\mathbf{y}_2 := \mathbf{B}_p \mathbf{x}_{p_2}$. While MPLP is a purely dual algorithm, i.e., the update to $\boldsymbol{\lambda}_p$ in Algorithm 3 line 12 does not require reasoning about optimal settings of the corresponding primal variables, Box-MPLP requires explicitly constructing a primal-dual pair.

The algorithm has two overall steps (a) fixing all dual variables besides $\boldsymbol{\nu}_p$, define a small block-specific optimization problem, and efficiently determine what the optimal values $\mathbf{y}_1^*$ and $\mathbf{y}_2^*$ should be for it, and (b) construct a value for $\boldsymbol{\nu}_p^*$ for which maximizing over the primal variables yields the values determined in step (a) and satisfies the complementary slackness conditions (16) (a). Therefore, by construction of a primal-dual certificate, $\boldsymbol{\nu}_p^*$ minimizes the block coordinate descent objective.

In step (a), we seek primal optimizers $\mathbf{y}_1^*$ and $\mathbf{y}_2^*$. With all dual variables besides $\boldsymbol{\nu}_p$ fixed, MAP inference in the subproblems p_1 and p_2 is with respect to shifted weight vectors $\tilde{w}_{p_1}$ and $\tilde{w}_{p_2}$ as defined in Algorithm 3 lines 6 and 8 (which doesn't include $\boldsymbol{\nu}_p$ in the shift). Using (19) we can reduce the choice of $\mathbf{y}_1^*$ and $\mathbf{y}_2^*$ to a local optimization problem by obtaining max-marginals $\mathbf{m}_1$ and $\mathbf{m}_2$ for the subproblems, as in Algorithm 3 lines 7 and 9. With these, we have $(\mathbf{y}_1^*, \mathbf{y}_2^*) = (e_{i^*}, e_{j^*})$, where

$$(i^*, j^*) = \arg\max_{(i,j)} \mathbf{m}_1(i) + \mathbf{m}_2(j) - \sum_{j' \neq j} \mathbf{c}_p(i, j'). \quad (21)$$

Step (b) constructs a $\boldsymbol{\nu}_p^*$ that satisfies (16) and for which optimizing over the primal variables yields $(\mathbf{y}_1, \mathbf{y}_2) = (i^*, j^*)$. Invoking the 'linearity' of max-marginals (20), this can be expressed as the following conditions on $\boldsymbol{\nu}_p$:

$$\forall i,\ \mathbf{m}_1(i^*) - \sum_j \boldsymbol{\nu}_p(i^*, j) \quad \geq \quad \mathbf{m}_1(i) - \sum_j \boldsymbol{\nu}_p(i, j) \quad (22)$$

$$\forall j,\ \mathbf{m}_2(j^*) + \sum_i \boldsymbol{\nu}_p(i, j^*) \quad \geq \quad \mathbf{m}_2(j) + \sum_i \boldsymbol{\nu}_p(i, j). \quad (23)$$

Satisfying (16) along with (22) and (23) ensures that the independent maximizations of the reweighted problems will have the same score and same maximizing values as the joint maximization in equation (21), and thus we have a primal-dual pair for the coordinate descent subproblem.

Solving the maximization in (21) can be done, in the worst case, by enumerating all s_p^2 possible i and j. Selecting $\boldsymbol{\nu}_p$ that satisfies conditions (16), (22), and (23) requires solving a linear feasibility problem, however. While this can be done in time polynomial in s_p, we focus in the next section on an important special case where it is particularly tractable, and leave exploration of general algorithms for this feasibility problem to future work.

7.1 AGREEMENT FACTORS

Next, we focus on a particular structure of $\mathbf{c}_p$ that is both reasonable for applications and for which finding $\boldsymbol{\nu}_p$ satisfying (16), (22), and (23) can be done in time $O(s_p)$. This results in the block coordinate descent Algorithm 4.

Definition Let $\mathbf{y}_1$ and $\mathbf{y}_2$ be two projection variables with values i and j, and define vector $\alpha \in \mathbb{R}_+^{s_p}$. An *agreement factor* between $\mathbf{y}_1$ and $\mathbf{y}_2$ is a structured linear model that assigns a score of 0 if they agree and a score of $-\alpha(i)$ if they disagree. This is equivalent to a penalty matrix:

$$\mathbf{c}_p(i,j) = \begin{cases} \alpha(i) & \text{if } i = j \\ 0 & \text{otherwise.} \end{cases} \tag{24}$$

For many applications, it is sufficient to use agreement factors rather than full matrix penalties $\mathbf{c}_p(i,j)$, since they allow the model to impose a penalty if two components of the structured output are not equal. This, for example, supports the soft constraints of Rush *et al.* (2012) that we employ in our experiments. However, we show in Section 8 that matrix penalties are important to support a mapping between general graphical model factors and soft constraints.

Given the structure (24) on the penalties, there are effectively only s_p dual variables in the matrix $\boldsymbol{\nu}_p$, as the off-diagonal elements are constrained to be equal to 0 by the box constraints (15). We refer to the dual variable and costs as $\boldsymbol{\nu}_p(i)$ and $\mathbf{c}_p(i)$, and equations (22) and (23) reduce to

$$\mathbf{m}_1(i^*) - \boldsymbol{\nu}_p(i^*) \;\geq\; \mathbf{m}_1(i) - \boldsymbol{\nu}_p(i) \quad \forall i,j \tag{25}$$
$$\mathbf{m}_2(j^*) + \boldsymbol{\nu}_p(j^*) \;\geq\; \mathbf{m}_2(j) + \boldsymbol{\nu}_p(j) \quad \forall i,j \tag{26}$$

In Appendix 4 we derive an $O(s_p)$ method for choosing $\boldsymbol{\nu}_p$ that satisfies (16), (22), and (23). The overall insight is that (25) and (26) can be manipulated to yield simple upper and lower bounds on feasible values of $\boldsymbol{\nu}_p(i)$ for $i \neq i^*, j^*$, for which we choose the midpoint of the feasible interval (Algorithm 4, line 22). Also, if $i^* \neq j^*$, then $\boldsymbol{\nu}_p(i^*)$ and $\boldsymbol{\nu}_p(j^*)$ are determined by complementary slackness (line 18) and otherwise, we can set them by similarly taking the mid-point of a feasible interval obtained from (25) and (26) (line 15). If we make the further restriction that the agreement factor uniformly penalizes disagreement between values of $\mathbf{y}_1$ and $\mathbf{y}_2$, i.e. $\mathbf{c}_p$ is α in all coordinates, then we have the added benefit that Algorithm 4 line 11 can be solved in $O(s_p)$ time. See the end of Appendix 4.

Algorithm 4 Box-MPLP: block coordinate descent for soft dual decomposition with agreement factors.

1: converged $\leftarrow false$
2: **while** !converged **do**
3: converged $\leftarrow true$
4: **for** constraint $p \in \mathcal{P}$ **do**
5: $\tilde{w}_{p_1} \leftarrow \mathbf{w}_{p_1} + \sum_{\substack{p': p'_2 = p_1 \\ p' \neq p}} \mathbf{B}_{p'}^T \boldsymbol{\nu}_{p'} - \sum_{\substack{p': p_1 = p_1 \\ p' \neq p}} \mathbf{A}_{p'}^T \boldsymbol{\nu}_{p'}$
6: $\mathbf{m}_1 \leftarrow \text{MaxMargs}\left(\tilde{w}_{p_1}\right)$
7: $\tilde{w}_{p_2} \leftarrow \mathbf{w}_{p_2} + \sum_{\substack{p': p'_2 = p_2 \\ p' \neq p}} \mathbf{B}_{p'}^T \boldsymbol{\nu}_{p'} - \sum_{\substack{p': p_1 = p_2 \\ p' \neq p}} \mathbf{A}_{p'}^T \boldsymbol{\nu}_{p'}$
8: $\mathbf{m}_2 \leftarrow \text{MaxMargs}\left(\tilde{w}_{p_2}\right)$
9: **if** (16) not satisfied **then**
10: converged $\leftarrow false$
11: $i^*, j^* \leftarrow \operatorname*{argmax}_{i,j} \mathbf{m}_1(i) + \mathbf{m}(j) - \mathbf{c}_p(i)\delta(i{\neq}j)$
12: **if** $i^* = j^*$ **then**
13: $U \leftarrow \min_{i \neq i^*} \mathbf{m}_1(i^*) - \mathbf{m}_1(i)$
14: $L \leftarrow \max_{j \neq j^*} \mathbf{m}_2(j) - \mathbf{m}_2(j^*) + \mathbf{c}_p(j)$
15: $\boldsymbol{\nu}_p(i^*) \leftarrow \frac{1}{2}(U + L)$
16: **else**
17: $\boldsymbol{\nu}_p(i^*) \leftarrow 0$
18: $\boldsymbol{\nu}_p(j^*) \leftarrow \mathbf{c}_p(j^*)$
19: **for** all i such that $i \neq i^*,\ i \neq j^*$ **do**
20: $L \leftarrow -\mathbf{m}_1(i) + \mathbf{m}_1(i^*) + \boldsymbol{\nu}_p(i^*)$
21: $U \leftarrow \quad \mathbf{m}_2(j^*) - \mathbf{m}_2(j) + \boldsymbol{\nu}_p(j^*)$
22: $\boldsymbol{\nu}_p(i) \leftarrow \frac{1}{2}(U + L)$

8 SOFT CONSTRAINTS V.S. FACTORS

As identified in the introduction, a traditional way to model soft constraints is to add global factors to a graphical model. In this case, the factors contribute scores when variables are set to certain values, which differs from our use of penalties that contribute negative score when variables are *not* set to certain values. We prove in Appendix 5 that the expressivity of factors and our soft constraints are equivalent, though, as long as the soft constraints are defined between projection variables. Specifically, any table of factor scores can be mapped into a penalty matrix $\mathbf{c}_p$ by solving an associated linear system. This may require using Algorithm 2 for inference, though, since Box-MPLP only applies to diagonal $\mathbf{c}_p$.

Though the two formulations are similar, soft constraints have attractive properties compared to factors. For example our algorithms maintain primal feasibility during intermediate iterations and avoid variable copying, which fractures the evidence for variables' MAP values across submodels and requires an entire dual decomposition iteration for information to travel between output variables and their copies. Our experiments support the desirability of avoiding variable copying. In future work, we will explore solving problems that are natively expressed using factors by first mapping them to problems with soft constraints.

9 RELATED WORK

There is a precedent for constructing message passing schemes for inference problems by minimizing an associated dual problem that decomposes into local interactions (Wainwright *et al.*, 2005; Komodakis *et al.*, 2007;

Globerson & Jaakkola, 2007; Ravikumar *et al.*, 2010; Martins *et al.*, 2012; Schwing *et al.*, 2012). Many of these are based on block coordinate descent. The generalizations we make in Section 6, such as working in terms of projection variables to make MPLP apply to more general structured prediction problems than graphical models, could also be applied to a variety of these other algorithms, where the requirement that the subproblems yield max-marginals would be replaced with other requirements, such as the ability to perform MAP in the presence of additional strongly-convex terms. Our algorithm, particularly in the context of the application we consider in the next section, can also be seen as an example of special-case handling of factors that have a specific combinatorial structure (Duchi *et al.*, 2007; Martins *et al.*, 2012; Mezuman *et al.*, 2013).

Our message passing algorithm has the same optimality guarantees as those for MPLP discussed in Sontag *et al.* (2011). Unlike (projected) subgradient descent, block coordinate descent may return sub-optimal outputs because our objective is non-smooth and not strongly convex (Luo & Tseng, 1992). Analysis of the convergence rate for smoothed versions of MPLP (Meshi *et al.*, 2012) is doable, however, and we encourage exploration of (smoothed) parallel versions of Box-MPLP (Richtárik & Takáč, 2012).

10 EXPERIMENTS

We evaluate soft constraint algorithms that vary along two dimensions: whether they solve box-constrained dual decomposition objectives or unconstrained ones based on variable copying and whether they employ (projected) subgradient descent or block coordinate descent. The first dimension is captured by the distinction between Figure 1, where the consensus variable at the top is an isolated structured linear model and there are soft constraints between this and the variables in the sentences, and Figure 2, which requires variable copying and an auxiliary tree-structured submodel. While Rush *et al.* (2012) did not employ MPLP, max-marginals can be obtained for the CRF tagger and projective parser they used (Smith, 2011). Also, note that the soft constraint penalties of Rush *et al.* (2012) used in both figures take the form of agreement factors. Therefore, we can apply Box-MPLP. We compare:

- Subgradient: Algorithm 1 applied to Figure 2
- Box-Subgradient: Algorithm 2 applied to Figure 1
- MPLP: Algorithm 3 applied to Figure 2
- Box-MPLP: Algorithm. 4 applied to Figure 1

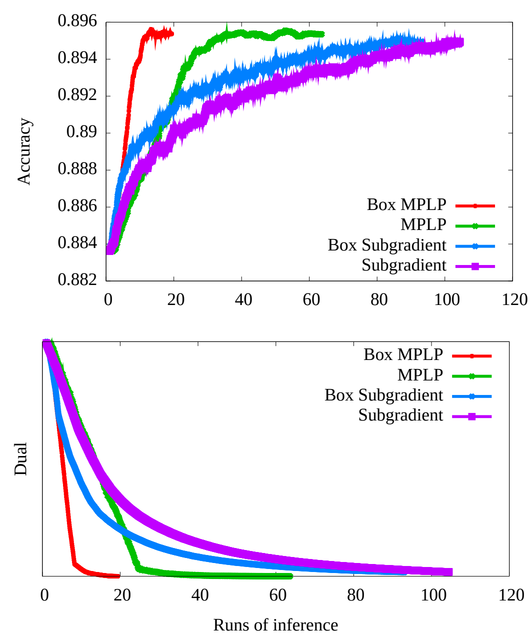
The specific problem considered by Anzaroot *et al.* (2014) problem does not admit a baseline algorithm that uses variable copying and hard-constraint dual decomposition. Therefore, besides providing experimental evidence for the effectiveness of Box-MPLP, we also seek to demonstrate the overall effectiveness of using a box-constrained objective for soft dual decomposition as an alternative to variable copying, regardless of what inference algorithm is used for minimizing the box-constrained objective. Finally, note that all algorithms provide an $O(\frac{1}{\sqrt{t}})$ convergence rate, so they can only be compared empirically.

We mirror the experimental setup of Rush *et al.* (2012) for both tagging and parsing. To measure the speed of the algorithms, we record the total number of calls to inference in sentence-level problems, which we normalize by the number of sentences in the corpus to facilitate comparison across experiments. After the first pass, we only perform inference when relevant dual variables change.

Measuring inference calls rather than wall-clock time yields a more reliable experimental setting for the following two reasons: (1) it is independent of the implementation used, and (2) it allows us to be generous to the baseline algorithms we seek to outperform. First, we ignore the cost of running MAP inference in the tree-structured auxiliary problem in Figure 2. Second, we assign a pessimistic multiplier of two for all inference calls that require max-marginals. For NLP models with millions of features, this is an exaggeration because computing the model's score vector $\mathbf{w}$ is typically the most costly step.

10.1 POS TAGGING

Figure 3: Accuracy (top) and dual objective (bottom) v.s. runs of sentence-level inference for WSJ-200 POS tagging.



Following Rush *et al.* (2012), we learn models on subsets of 50, 100, 200, and 500 sentences from the first chapter of the Penn Treebank and test on the Penn Treebank chapters test set (Marcus *et al.*, 1993). We use a bigram CRF tagger (Lafferty *et al.*, 2001). For all experiments, we report average sentence-level accuracy and the gains we obtain from corpus-wide inference in Appendix 6. Both are consistently comparable to Table 4 of Rush *et al.* (2012).

Table 1: Normalized number of inference runs for each algorithm to attain quantiles of the best dual solution in the WSJ-200 tagging experiment. If a quantile was not reached during 100 iterations, we show 'na'.

Accuracy quantile	80%	85%	90%	95%
Subgradient	70	92	na	na
MPLP	22	23	25	30
Box-Subgradient	20	35	40	54
Box-MPLP	**8**	**9**	**10**	**10**
Dual Quantile	80%	85%	90%	95%
Subgradient	24	34	56	na
MPLP	21	22	23	35
Box-Subgradient	30	35	40	54
Box-MPLP	**7**	**7**	**8**	**9**

We present results from where we train on 200 sentences, but they are representative of the others, given in Appendix 6.1. Figure 3 shows the corpus-wide tagging accuracy and dual objective as a function of the sentence-level MAP calls. Recall that we double-count all calls to max-marginal routines. Table 1 shows how much inference is necessary to reach various percentile gains in accuracy and percentile reductions in the dual objective. Box-MPLP substantially outperforms both Box-Subgradient and MPLP, and the box-constrained versions of both algorithms outperform their variable-copying-based counterparts. Compared to the baseline subgradient algorithm used by Rush *et al.* (2012), we require 10x fewer MAP calls.

10.2 DEPENDENCY PARSING

Table 2: Iteration costs for the parsing experiments.

PTB to QTB				
Accuracy quantile	80%	85%	90%	95%
Subgradient	4.1	4.3	5.2	6.1
MPLP	4.3	4.3	4.3	'na'
Box-Subgradient	**2.1**	**2.1**	**2.4**	**2.8**
Box-MPLP	2.6	2.8	3	'na'
Dual quantile	80%	85%	90%	95%
Subgradient	3.0	3.2	3.4	3.9
MPLP	4.2	4.4	4.9	4.9
Box-Subgradient	**1.6**	**1.7**	**1.8**	**2.0**
Box-MPLP	2.5	2.5	2.5	2.6
QTB to PTB				
Dual quantile	80%	85%	90%	95 %
Subgradient	15	16	18	22
MPLP	14	15	16	17
Box-Subgradient	8.1	9.2	10	12
Box-MPLP	**6.9**	**7.4**	**7.9**	**8.6**

Our corpus-wide parsing experiments present a characteristically different regime for comparing the four algorithms because the graph of connections between the subproblems is much more sparse and the overall number of necessary iterations for the algorithms to converge is much lower.

Following Rush *et al.* (2012), each set of POS tags around a token defines a context, and identical contexts are encouraged to have parents with similar POS tags by introducing various consensus structures. We mirror their domain adaptation experiments, training on the Penn Treebank (PTB) and testing on the Question Treebank (QTB), and vice-versa (Judge *et al.*, 2006). We parse with a first-order projective arc-factored parser (McDonald *et al.*, 2005) using dynamic programming for inference, which has lower accuracy than the second-order projective parser used in Rush *et al.* (2012). Table 2 summarizes our results.

In the PTB-to-QTB experiment, the box-constrained algorithms uniformly outperform their counterparts based on variable copying. Unlike our POS experiments, however, Box-MPLP does not outperform Box-Subgradient. Since all the algorithms converge so quickly, the extra computation to obtain max-marginals is too costly (in the factor-2 scheme). Box-MPLP is still about 2x faster than Subgradient, which is what Rush *et al.* (2012) used, though. For the QTB-to-PTB experiment we were unable to reproduce accuracy increases as reported in Rush *et al.* (2012); none of the optimization algorithms managed to improve the accuracy for any setting of the penalties. This is probably due to our simpler parser. However, regarding dual optimization, each coordinate descent method outperforms its corresponding subgradient method, and the boxed algorithms outperform their variable-copying alternatives. Again, Box-MPLP was about 2x faster than Subgradient. See Appendix 6.2 for accuracy and dual figures.

11 CONCLUSION AND FUTURE WORK

Soft constraints can be easily modeled by imposing box constraints on an associated dual decomposition objective. This yields fast, simple-to-implement algorithms. Box-MPLP, a block coordinate descent algorithm, provides a competitive alternative to projected subgradient descent.

Future work will explore ways to adapt the alternative message passing algorithms discussed in Section 9 to handle box constraints and consider additional combinatorial factors besides soft constraints that can be 'optimized out' by imposing constraints in an associated dual problem.

12 ACKNOWLEDGMENT

This work was supported in part by the Center for Intelligent Information Retrieval, in part by DARPA under agreement number FA8750-13-2-0020 and in part by NSF grant #CNS-0958392. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect those of the sponsor.

References

Anzaroot, Sam, Passos, Alexandre, Belanger, David, & McCallum, Andrew. 2014. Learning Soft Linear Constraints with Application to Citation Field Extraction. *In: Proceedings of the 52th Annual Meeting of the Association for Computational Linguistics*.

Chang, Ming-Wei, Ratinov, Lev, & Roth, Dan. 2012. Structured learning with constrained conditional models. *Machine learning*, **88**(3), 399–431.

Chieu, Hai Leong, & Teow, Loo-Nin. 2012. Combining local and non-local information with dual decomposition for named entity recognition from text. *Pages 231–238 of: 15th International Conference on Information Fusion*.

Cortes, Corinna, & Vapnik, Vladimir. 1995. Support-vector networks. *Machine learning*, **20**(3), 273–297.

Duchi, John, Tarlow, Daniel, Elidan, Gal, & Koller, Daphne. 2007. Using Combinatorial Optimization within Max-Product Belief Propagation. *Pages 369–376 of: Advances in Neural Information Processing Systems 19*.

Finkel, Jenny Rose, Grenager, Trond, & Manning, Christopher. 2005. Incorporating non-local information into information extraction systems by gibbs sampling. *Pages 363–370 of: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*.

Globerson, Amir, & Jaakkola, Tommi. 2007. Fixing max-product: Convergent message passing algorithms for MAP LP-relaxations. *Advances in Neural Information Processing Systems*, **21**(1.6).

Judge, John, Cahill, Aoife, & Van Genabith, Josef. 2006. Questionbank: Creating a corpus of parse-annotated questions. *Pages 497–504 of: Proceedings of the 21st International Conference on Computational Linguistics*.

Komodakis, Nikos, Paragios, Nikos, & Tziritas, Georgios. 2007. MRF optimization via dual decomposition: Message-passing revisited. *Pages 1–8 of: IEEE 11th International Conference on Computer Vision*.

Koo, Terry, Rush, Alexander M, Collins, Michael, Jaakkola, Tommi, & Sontag, David. 2010. Dual decomposition for parsing with non-projective head automata. *Pages 1288–1298 of: Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*.

Lafferty, John D, McCallum, Andrew, & Pereira, Fernando CN. 2001. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. *Pages 282–289 of: Proceedings of the Eighteenth International Conference on Machine Learning*.

Luo, Zhi-Quan, & Tseng, Paul. 1992. On the convergence of the coordinate descent method for convex differentiable minimization. *Journal of Optimization Theory and Applications*, **72**(1), 7–35.

Marcus, Mitchell P, Marcinkiewicz, Mary Ann, & Santorini, Beatrice. 1993. Building a large annotated corpus of English: The Penn Treebank. *Computational linguistics*, **19**(2), 313–330.

Martins, Andre FT, Figueiredo, Mario AT, Aguiar, Pedro MQ, Smith, Noah A, & Xing, Eric P. 2012. Alternating directions dual decomposition. *arXiv preprint arXiv:1212.6550*.

McDonald, Ryan, Crammer, Koby, & Pereira, Fernando. 2005. Online large-margin training of dependency parsers. *Pages 91–98 of: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*.

Meshi, Ofer, Jaakkola, Tommi, & Globerson, Amir. 2012. Convergence Rate Analysis of MAP Coordinate Minimization Algorithms. *Pages 3023–3031 of: Advances in Neural Information Processing Systems 25*.

Mezuman, Elad, Tarlow, Daniel, Globerson, Amir, & Weiss, Yair. 2013. Tighter Linear Program Relaxations for High Order Graphical Models. *In: Proceedings of the Twenty-Ninth Conference on Uncertainty in Artificial Intelligence (UAI-13)*. Corvallis, Oregon: AUAI Press.

Nesterov, Yurii. 2003. *Introductory lectures on convex optimization: A basic course*. Vol. 87. Springer.

Ravikumar, Pradeep, Agarwal, Alekh, & Wainwright, Martin J. 2010. Message-passing for graph-structured linear programs: Proximal methods and rounding schemes. *The Journal of Machine Learning Research*, **11**, 1043–1080.

Richtárik, Peter, & Takáč, Martin. 2012. Parallel coordinate descent methods for big data optimization. *arXiv preprint arXiv:1212.0873*.

Rush, Alexander M., & Collins, Michael. 2012. A Tutorial on Dual Decomposition and Lagrangian Relaxation for Inference in Natural Language Processing. *J. Artif. Intell. Res. (JAIR)*, **45**, 305–362.

Rush, Alexander M, Reichart, Roi, Collins, Michael, & Globerson, Amir. 2012. Improved parsing and pos tagging using intersentence consistency constraints. *Pages 1434–1444 of: Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

Schwing, Alex, Hazan, Tamir, Pollefeys, Marc, & Urtasun, Raquel. 2012. Globally Convergent Dual MAP LP Relaxation Solvers using Fenchel-Young Margins. *Pages 2393–2401 of: Advances in Neural Information Processing Systems 25*.

Smith, David A., & Eisner, Jason. 2008. Dependency Parsing by Belief Propagation. *Pages 145–156 of: Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

Smith, Noah A. 2011. Linguistic structure prediction. *Synthesis Lectures on Human Language Technologies*, **4**(2), 1–274.

Sontag, David, Globerson, Amir, & Jaakkola, Tommi. 2011. Introduction to Dual Decomposition for Inference. *In:* Sra, Suvrit, Nowozin, Sebastian, & Wright, Stephen J. (eds), *Optimization for Machine Learning*. MIT Press.

Sutton, Charles, & McCallum, Andrew. 2006. *Introduction to statistical relational learning*. MIT Press. Chap. An introduction to conditional random fields for relational learning.

Wainwright, Martin J, Jaakkola, Tommi S, & Willsky, Alan S. 2005. MAP estimation via agreement on trees: message-passing and linear programming. *Information Theory, IEEE Transactions on*, **51**, 3697–3717.

Werner, Tomás. 2008. High-arity interactions, polyhedral relaxations, and cutting plane algorithm for soft constraint optimisation (MAP-MRF). *In: CVPR 2008*.