Dept. Computer Science
University of Helsinki
Technical Report C-2007-87

An Efficient Method for Large Margin Parameter Optimization in Structured Prediction Problems

Huizhen Yu[*]
janey.yu@cs.helsinki.fi

Juho Rousu[†]
juho.rousu@cs.helsinki.fi

March 2007, revised December 2007

Abstract

We consider structured prediction problems with a parametrized linear prediction function, and the associated parameter optimization problems in large margin type of discriminative training. We propose a dual optimization approach which uses the restricted simplicial decomposition method to optimize a reparametrized dual problem. Our reparametrization reduces the dimension of the space of the dual function to one that is linear in the number of parameters and training examples, and hence independent of the dimensionality of the prediction outputs. This in conjunction with simplicial decomposition makes our approach efficient. We discuss the connections of our approach with related earlier works, and we show its advantages.

[*] Huizhen Yu is with the Helsinki Institute for Information Technology (HIIT), University of Helsinki, Finland.

[†] Juho Rousu is with the Department of Computer Science, University of Helsinki, Finland.

Contents

1 Introduction

The large margin framework of discriminative training and parameter optimization for prediction with structured outputs emerges from a line of recent researches starting with Collins [Col02], Altun et al. [ATH03] and Taskar et al. [TGK04]. In contrast to earlier discriminative training frameworks in which one often encounters difficult non-convex optimization problems, in the large margin framework one has convex quadratic programming problems with linear constraints. Even though the number of these constraints is exponential in the dimension of the prediction outputs, the aforementioned works, as well as the later ones e.g., [TLJJ05, TJHA05, RSSS06], show that solving the associated optimization problem is tractable for a broad class of prediction problems by utilizing their structures. This and other reasons (such as the capability of using kernels) make the large margin framework attractive for discriminative training.

Since the work of [Col02, ATH03, TGK04], more efficient and scalable methods have been researched for solving the associated optimization problems or their variants. Besides their large scale character, the main difficulty of these problems is the large number of constraints as mentioned earlier. Several approaches have been proposed: in Taskar et al. [TLJJ05], a minmax solution method of the gradient type; in Tsochantaridis et al. [TJHA05] (as well as [ATH03]), dual optimization using column generation; and in Rousu et al. [RSSS06], dual optimization with the conditional gradient method.

In this paper we propose a new dual optimization approach for solving the primal problems (not with kernels) under the formulations of [TGK04, TLJJ05] and [TJHA05]. Our approach is motivated by the use of reduction of dual variables as in [TGK04, RSSS06], and by the use of the conditional gradient method as in [RSSS06], and particularly by their combined use as in [RSSS06]. However, both our reduction and optimization methods are different from and more effective than the counterparts in these earlier works when solving the primal problems (see discussions in Section 3.4). In this respect our work extends the earlier ones.

In our approach we first reduce the number of dual variables by reparametrization of the dual function. We then solve the reduced dual problem by applying the restricted simplicial decomposition method (RSD) of Hearn et al. [HLV87]. Our reparametrization utilizes the linear structure of the prediction function and is new to our knowledge. It reduces the dimension of the space of the dual function to one that is linear in the number of parameters and training examples, and hence independent of the dimensionality of the prediction outputs. The reparametrization is thus useful, when applied in conjunction with simplicial decomposition, for cases where the number of parameters is less than the dimensionality of the prediction outputs, and more generally, for cases where on a single input, the number of parameters involved in the prediction is typically less than the dimensionality of the output space.

The restricted simplicial decomposition method is one of the methods for large-scale convex optimization problems with linear constraints. (It also has a counterpart for problems with non-linear constraints, which we do not use in the present paper.) It belongs to the family of feasible direction methods, which includes the conditional gradient (known by other names such as the Frank-Wolfe method), and the simplicial decomposition method (Holloway [Hol74], Hohenbalken [Hoh77]; see also Bertsekas [Ber99], Chapter 2). These methods are based on similar ideas of making successive inner approximation of the feasible region by convex hulls of feasible points particularly chosen and solving the original problem iteratively by solving the so-called master problems on the inner approximation. The methods are suitable for problems, such as ours, where the feasible region is described by an enormous number of constraints, but an extreme point of the feasible region along a descent direction can be found relatively efficiently. Two simplicial decomposition type of methods have been in fact applied to our problems: [RSSS06] and [TJHA05] (see discussions in Section 3.4).

The restricted simplicial decomposition method sets a limit on the number of extreme points of the convex hulls used for inner approximation, and by doing so it keeps control of the size of master problems. It is a method intermediate between the simplicial decomposition method which does not have control of the size of master problems due to the continually growing number of extreme points that form inner approximation, and the conditional gradient method which can exhibit inefficient zigzag behavior (see e.g., [HLV87], and [Ber99], Chapter 2) due to its use of a line segment for inner approximation. As demonstrated experimentally in [HLV87], a moderate increase of the dimension of the approximating convex hulls in the RSD can alleviate significantly the zigzagging problem.

The paper is organized as follows. In Section 2, we first give a short description of the restricted simplicial decomposition method (Section 2.1), and we then give a brief review of the formulation of the large margin structured learning problem (Section 2.2). In Section 3 we present our reparametrization and the reduced dual problem, we describe the application of the restricted simplicial decomposition method, and we show that the direction-finding subproblems still have the favorable form of the so-called loss-augmented inference problems. In this section we also discuss the connections of our method with related earlier works, and show its advantages. In Section 4 we demonstrate how to apply our approach to several problems considered by the earlier works [TLJJ05, TJHA05], namely, binary segmentation of images and problems formulated with loss-scaled slack constraints, which have slightly different formulations than that being considered in Section 3. For the image segmentation application, we show that in our approach the subproblems have a favorable structure and can be solved by regular flow/min-cut algorithms, while in the earlier approach [TLJJ05] the subproblems created have a more complex form and have to be solved by convex-cost flow/min-cut algorithms which increase the overall computation overhead. The same is true also for the sentence alignment application considered in [TLJJ05]. We will not give the details because this application can be formulated as in Section 3 and is thus covered by the general discussion there.

2 Preliminaries

2.1 Restricted Simplicial Decomposition Method (Hearn et al. [HLV87])

Consider the problem of minimizing a convex differentiable function $f(x)$ on $\Re^n$ subject to linear constraints $Ax \leq b$:

$$\min_{x:\, Ax\leq b} \quad f(x)$$

The simplicial decomposition method solves this problem by iterating between direction-finding subproblems and master problems until it finds an optimal solution. The method is initialized with a feasible point x_0, and at iteration k, it solves the subproblem

$$\begin{aligned} z_k = \operatorname*{arg\,min}_{z \in \Re^n} \quad & \nabla f(x_k)' \, z \\ \text{subj.} \quad & Az \leq b, \end{aligned}$$

in order to find a new feasible extreme point z_k along a descent direction. If $\nabla f(x_k)'(z_k - x_k) = 0$, then x_k is evidently an optimal solution. Otherwise, the method proceed to solve the master problem of minimizing $f(x)$ on an inner approximation of the feasible region by the convex hull of the points $\{x_0, z_0, z_1, \ldots, z_k\}$:

$$x_{k+1} = \operatorname*{arg\,min}_{x \in \text{conv}(S_{k+1})} \quad f(x), \qquad \text{where} \;\; S_{k+1} = \{z_0, z_1, \ldots, z_k\} \cup \{x_0\}.$$

The master problem is identical to optimizing $f(\sum_{i=1}^m q_i y_i)$ over $q = (q_1, \ldots q_m)$ subject to a simplex constraint on q, where $m = |S_{k+1}|$ and $y_i \in S_{k+1}$:

$$\begin{aligned} \min_q \quad & f\Big(\sum_{i=1}^m q_i y_i\Big) \\ \text{subj.} \quad & q \geq 0, \;\; \sum_{i=1}^m q_i = 1. \end{aligned}$$

The restricted simplicial decomposition method solves the same subproblems and master problems, except that in the latter it sets a limit on the dimension of the simplexes, i.e., the number of points in S_k, for the purpose of having control over the size of master problems. When the number of points in S_k reaches the limit, we simply replace one point in the set by x_k.[1] After solving the master problem, we may decrease the size of S_{k+1} by removing points that share zero weight in the expression of x_{k+1}. But we do not have to do so particularly if the dimension of the convex hulls is strictly less than the dimension of the feasible region. These details are evident and hence omitted here.

When we set the dimension limit in RSD to be 1, we obtain the conditional gradient algorithm; when we set the limit to be no less than the total number of extreme points of the feasible region, we obtain the simplicial decomposition method. The convergence of RSD is evident; moreover, the method has finite convergence when we set the dimension limit to be no less than the dimension of the face of the feasible region that contains the optimal solution [HLV87].

For solving the master problems on the simplexes, a projected Newton method (Bertsekas [Ber82]) can be applied and its efficiency is demonstrated in [HLV87].

In our problems, the feasible region is a Cartesian product of sets. Utilizing this feature, the application of the simplicial decomposition methods can take a slightly different form, as we will discuss later.

2.2 Large Margin Structured Learning Problem

Formulation

Let (x, y) be an input datum and label pair, where x in practice can correspond to raw measurements, and y corresponds to labels that we want to infer after observing x. The space of x can be arbitrary, and the space of y, denoted by $\mathcal{Y}^x$, is determined by x and can be different for different values of

[1] More precisely, this is done in [HLV87] as follows. Denote by r the dimension limit, and denote the current set of extreme points by S_k^s. We also keep a set denoted by S_k^x, which is either an empty set or a singleton set consisting of x_j for some $j \leq k$. The union of the two sets is denoted by S_k. When $|S_k^s| < r$, we add z_k to it to form S_{k+1}^s, and keep $S_{k+1}^x = S_k^x$. When $|S_k^s| = r$, we delete from the set the point that shares the smallest weight in the expression of x_k as a convex combination of points in S_k, and add the point z_k to form S_{k+1}^s; we also set $S_{k+1}^x = \{x_k\}$. We solve the master problem on the convex hull of $S_{k+1} = S_{k+1}^s \cup S_{k+1}^x$.

x. For example, when x are images or sentences, y can be tags associated with each pixel or word, whose spaces are thus different for images of different sizes or sentences of different lengths.

We have a score function $F(x, \cdot; w) : \mathcal{Y}^x \to \Re$ with parameters $w \in \Re^d$, using which we predict the y with the maximal score to be the label for a given x. We assume that F is *linear* in w for fixed x and y. Furthermore, we assume that F is a sum of functions of components of y of the form

$$F(x, y; w) = \sum_{c \in \mathcal{C}^x} \theta_c(w)' \, \Phi_c(x, y_c). \tag{1}$$

Here each Φ_c is a vector-valued function of a (usually) small subset of components of y, denoted by y_c; $\mathcal{C}^x$ denotes the index set of all such functions; and $\theta_c(w)$ denotes a vector that has as entries a certain subset of components of w. We call Φ_c feature functions, or features.

The decomposition in (1) is a model assumption, and it usually corresponds to a graph that encodes believed direct interactions between components of x and y. For this reason c is sometimes also referred to as a "hyper-edge." There are also associated probabilistic interpretations based on e.g., Markov random fields and conditional Markov random fields, which are outside our present scope. As an illustration, in the case of images for instance, the simplest choice of the graph can be a grid; $\mathcal{C}^x$ are thus indices of the edges and nodes of a grid and each component of y is associated with a node; and for each node or edge c, $\Phi_c(x, y_c)$ can indicate the compatibility of the label y_c with local measurements of the image at sites associated with c.

We call the problem of finding a maximal scoring label for a given x and w,

$$\max_{y \in \mathcal{Y}^x} \sum_{c \in \mathcal{C}^x} \theta_c(w)' \, \Phi_c(x, y_c), \tag{2}$$

the inference problem. Depending on the decomposition structure, this problem can sometimes be solved efficiently by special algorithms, e.g., dynamic programming, shortest path, min-cut algorithms, linear programming.

Large margin parameter optimization and primal problem

Consider the problem of determining the parameter w of the prediction function given a set of training examples $\{(x^i, y^i), i \in I\}$, where (x^i, y^i) corresponds to the input datum and true label pair for the i-th example. We denote the label space of x^i by $\mathcal{Y}_i$.

The discriminative training formulated by Taskar et al. [TGK04] takes the large margin approach of SVM and chooses the parameter to be the solution of the following convex quadratic programming problem:

$$\begin{aligned} &\min_{w,\xi} \ \tfrac{1}{2}\|w\|^2 + C \sum_{i \in I} \xi_i \\ &\text{subj.} \quad F(x^i, y; w) - F(x^i, y^i; w) + l(x^i, y) \le \xi_i, \quad \xi_i \ge 0, \quad \forall\, y \in \mathcal{Y}_i, \quad i \in I. \end{aligned}$$

Here $l(x, \cdot) : \mathcal{Y}^x \to \Re$ is a so-called loss function; it is non-negative and depends on the true label y^x of x, with $l(x, y)$ being the "loss" of predicting y and with $l(x, y^x) = 0$ naturally. The constraints imply that the desired parameter w should not only make the true label the highest scoring label for every training example, if this is possible, but also keep the score of a false label y lower by a gap that is larger than the loss $l(x, y)$. We note that there is also a probabilistic interpretation of this parameter training criterion, for which we refer readers to [TGK04].

To simplify notation, we now rewrite the primal problem, expressing the constraints explicitly in a linear form of w. We write

$$F(x^i, y; w) - F(x^i, y^i; w) = w'\big(\mathbf{f}(x^i, y) - \mathbf{f}(x^i, y^i)\big) = w'\Delta\mathbf{f}(x^i, y)$$

where $\mathbf{f}(x^i, y)$ denotes the vector properly defined by collecting associated terms in the right hand side of the equation

$$w'\mathbf{f}(x^i, y) = \sum_{c \in \mathcal{C}^{x^i}} \theta_c(w)' \, \Phi_c(x^i, y_c),$$

and $\Delta\mathbf{f}(x^i, y)$ denotes the difference of $\mathbf{f}(x^i, y)$ relative to the features of x^i with true label y^i:

$$\Delta\mathbf{f}(x^i, y) = \mathbf{f}(x^i, y) - \mathbf{f}(x^i, y^i).$$

With this notation, the primal problem is equivalently written as

$$\begin{aligned} \min_{w,\xi} \quad & \tfrac{1}{2}\|w\|^2 + C\sum_{i \in I} \xi_i \\ \text{subj.} \quad & w'\Delta\mathbf{f}(x^i, y) + l(x^i, y) \leq \xi_i, \quad \xi_i \geq 0, \quad \forall\, y \in \mathcal{Y}_i, \quad i \in I. \end{aligned} \tag{3}$$

Loss-augmented inference problems

It has been noticed that in various approaches for solving the above optimization problem, (as well as in our approach as we will discuss later), we encounter as subproblems the following variant of the inference problem (2) with an additional loss term in the objective:

$$\max_{y \in \mathcal{Y}^x} \sum_{c \in \mathcal{C}^x} \theta_c(w)'\, \Phi_c(x, y_c) + l(x, y). \tag{4}$$

This is referred to as the loss-augmented inference problem. Depending on the structure of $\mathcal{C}^x$ and the decomposition structure of the loss function, the loss-augmented inference problems can sometimes be solved by the same algorithm for inference problems with little or slightly more complexity.

3 Our Approach

We solve the primal problem (3) by dual optimization. Our approach is motivated by the following observation on minimizing a function $Q(\alpha)$ on certain feasible set $E \subset \Re^n$ using simplicial decomposition type of methods. Suppose that Q can be written as a function $\bar{Q}$ on a lower dimensional space $\Re^k$ after a linear transformation A: $Q(\alpha) = \bar{Q}(A\alpha)$. Then it can be more efficient to carry out the optimization in the new coordinates after the transformation $\bar{\alpha} = A\alpha$, no matter how this transformation renders simple constraints of α into complex ones of $\bar{\alpha}$. To see this, let us denote by $A(E)$ the image of the set E under the linear transformation. The direction-finding subproblems at feasible points $\bar{\alpha}$ in the new coordinates

$$\begin{aligned} \min_{\bar{d} \in \Re^k} \quad & \nabla\bar{Q}(\bar{\alpha})'\, \bar{d} \\ \text{subj.} \quad & \bar{\alpha} + \bar{d} \in A(E) \end{aligned}$$

are equivalent to those in the old coordinates at feasible points α such that $\bar{\alpha} = A\alpha$:

$$\begin{aligned} \min_{d \in \Re^n} \quad & \nabla Q(\alpha)'\, d \\ \text{subj.} \quad & \alpha + d \in E. \end{aligned}$$

Hence the computation complexity of solving direction-finding subproblems will not increase because of the transformation. The complexity can indeed decrease in computing gradients and objective functions of master problems due to working in a lower dimensional space. Consequently, the overall complexity of minimization $\bar{Q}$ can be much less than minimizing Q directly, particularly when an optimal solution $\bar{\alpha}^*$ is sufficient for the purpose and a corresponding optimal solution α^* is not necessary, as in our case of obtaining the primal optimal solution through dual optimization.

3.1 A Reduced Dual Formulation via Reparametrization

We now derive the dual function with a suitable reparametrization that reduces the number of dual variables. Instead of working with the dual function directly, it is more convenient to work with the

Lagrangian function and utilize the linear structure therein. Introducing non-negative multipliers $\alpha_i(y)$ for each i and $y \in \mathcal{Y}_i$, the Lagrangian function is

$$\tfrac{1}{2}\|w\|^2 + C\sum_{i\in I}\xi_i + \sum_{i\in I}\sum_{y\in\mathcal{Y}_i}\alpha_i(y)\big(w'\mathbf{\Delta f}(x^i,y) + l(x^i,y) - \xi_i\big).$$

Consider the sum in the third term for a fixed i, which can be written as

$$w'\sum_{y\in\mathcal{Y}_i}\mathbf{\Delta f}(x^i,y)\alpha_i(y) + \sum_{y\in\mathcal{Y}_i} l(x^i,y)\alpha_i(y) - \xi_i\sum_{y\in\mathcal{Y}_i}\alpha_i(y).$$

Corresponding to the first two terms in the right hand side, we introduce variables $\beta_i \in \Re^d$, $\gamma_i \in \Re$ defined by

$$\beta_i = \sum_{y\in\mathcal{Y}_i}\mathbf{\Delta f}(x^i,y)\,\alpha_i(y), \tag{5}$$

$$\gamma_i = \sum_{y\in\mathcal{Y}_i} l(x^i,y)\,\alpha_i(y). \tag{6}$$

Or, in matrix notation, for some matrix A_i defined through Eqs. (5) and (6), we can write

$$\begin{bmatrix}\beta_i\\ \gamma_i\end{bmatrix} = A_i\alpha_i,$$

where α_i denotes the vector with components $\alpha_i(y)$. Note that β_i, γ_i are linear transformations of α_i with the transformations *depending* on i.

Correspondingly, the minimax problem for the primal problem can be equivalently written as

$$\min_{\substack{w\\ \xi\geq 0}}\ \max_{\substack{\beta,\gamma,\alpha\\ s.t.\ \forall i\in I,\ \begin{bmatrix}\beta_i\\ \gamma_i\end{bmatrix}=A_i\alpha_i\\ \alpha_i\geq 0}}\ \tfrac{1}{2}\ \|w\|^2 + \sum_{i\in I}\xi_i\Big(C - \sum_{y\in\mathcal{Y}_i}\alpha_i(y)\Big) + \sum_{i\in I}\left(w'\beta_i + \gamma_i\right).$$

Exchanging the order of min and max, the equivalent dual problem is,

$$\begin{aligned}
&\max_{\beta,\gamma}\quad -\tfrac{1}{2}\Big\|\sum_{i\in I}\beta_i\Big\|^2 + \sum_{i\in I}\gamma_i\\
&\text{subj.}\quad \begin{bmatrix}\beta_i\\ \gamma_i\end{bmatrix} = A_i\alpha_i,\quad \sum_{y\in\mathcal{Y}_i}\alpha_i(y)\leq C,\quad \alpha_i\geq 0,\ \ \forall i\in I.
\end{aligned}$$

Note that the feasible set of (β,γ) is the Cartesian product of feasible sets of $(\beta_i,\gamma_i), \forall i \in I$, and each of the latter is the image of a higher dimensional simplex under a linear transformation. We can write the dual problem also as

$$\begin{aligned}
&\max_{\beta,\gamma}\quad -\tfrac{1}{2}\Big\|\sum_{i\in I}\beta_i\Big\|^2 + \sum_{i\in I}\gamma_i \qquad (7)\\
&\text{subj.}\quad \begin{bmatrix}\beta_i\\ \gamma_i\end{bmatrix} \in \mathcal{D}_i = A_i\left(\Big\{\alpha_i\Big|\sum_{y\in\mathcal{Y}_i}\alpha_i(y)\leq C,\quad \alpha_i\geq 0\Big\}\right),\quad \forall i\in I.
\end{aligned}$$

A dual optimal solution (β^*,γ^*) defines the primal optimal solution w^* by

$$w^* = -\sum_{i\in I}\beta_i^*.$$

3.2 Dual Optimization

We solve the dual problem (7) using the restricted simplicial decomposition method. An initial feasible point is easy to find: the origin is trivially a feasible point; corresponding to any label $y \in \mathcal{Y}_i$ one can define a feasible (β_i, γ_i) by $\beta_i = \bar{C}\Delta\mathbf{f}(x^i, y)$, $\gamma_i = \bar{C}l(x^i, y)$ with $\bar{C} \in [0, C]$, for instance. The direction-finding subproblems have the same form as the loss-augmented inference problems, and can thus be solved by efficient special algorithms if available. Choices of simplexes in master problems can be made combining with coordinate ascent. We give the details in what follows.

Direction-finding subproblems

Due to the product form of the feasible region $\prod_{i \in I} \mathcal{D}_i$, the direction-finding subproblem at an iteration is decomposed into direction-finding on each individual $\mathcal{D}_i$. Consider any feasible point $(\bar{\beta}, \bar{\gamma}) \in \mathcal{D}$. For training example i, let $g_{i,\beta}, g_{i,\gamma}$ be the components of the gradient of the dual function (7) with respect to β_i, γ_i, respectively:

$$g_{i,\beta} = -\sum_{j \in I} \bar{\beta}_j, \qquad g_{i,\gamma} = 1.$$

They are the same across all examples, as can be seen easily from the form of the dual function. The direction-finding sub-subproblem for training example i is

$$\max_{(\tilde{\beta}_i, \tilde{\gamma}_i) \in \mathcal{D}_i} \; g_{i,\beta}' \, \tilde{\beta}_i + \tilde{\gamma}_i. \tag{8}$$

It follows from the definition of $\mathcal{D}_i$ and Eqs. (5)-(6), the defining equations of the transformation A_i, that a maximum of the above problem can be found among the points $(\tilde{\beta}_i = 0, \tilde{\gamma}_i = 0)$ and $(\tilde{\beta}_i, \tilde{\gamma}_i)$ that correspond to $\alpha_i(y) = C$ for some y. Since the latter include the zero vector (choose the corresponding y to be y^i), we can write the direction-finding sub-subproblem equivalently as

$$\max_{y \in \mathcal{Y}_i} \; g_{i,\beta}' \, \Delta\mathbf{f}(x^i, y) + l(x^i, y). \tag{9}$$

Problem (9) is further equivalent to, neglecting the term $g_{i,\beta}' \, \mathbf{f}(x^i, y^i)$ that is constant with respect to y,

$$\max_{y \in \mathcal{Y}_i} \; g_{i,\beta}' \, \mathbf{f}(x^i, y) + l(x^i, y) = \max_{y \in \mathcal{Y}_i} \sum_{c \in \mathcal{C}^{x^i}} \theta_c(g_{i,\beta})' \, \Phi_c(x^i, y) + l(x^i, y).$$

This is the loss-augmented inference problem with parameters $g_{i,\beta}$. Its optimal solution y^* defines an optimal solution $(\tilde{\beta}_i^*, \tilde{\gamma}_i^*)$ of the direction-finding sub-subproblem (8) through Eqs. (5)-(6) by

$$\tilde{\beta}_i^* = C\Delta\mathbf{f}(x^i, y^*), \qquad \tilde{\gamma}_i^* = C\, l(x^i, y^*).$$

Notice that $(\tilde{\beta}_i^* = 0, \tilde{\gamma}_i^* = 0)$, i.e., $y^* = y^i$, implies that the parameter vector w corresponding to the current dual variables $\bar{\beta}$ by $w = -\sum_{j \in I} \bar{\beta}_j$ satisfies the margin constraints in the primal problem (3) for the i-th training example with $\xi_i = 0$.

Application of RSD

There are a variety of ways of applying RSD:

- Applying RSD with one simplex. We can do this with or without coordinate ascent.

  (i) Without coordinate ascent, we introduce a further reparametrization

$$z = \Big(\sum_{i \in I} \beta_i, \sum_{i \in I} \gamma_i \Big)$$

and optimize the dual function on the space of z. In the direction-finding subproblem of an iteration we solve for all training examples the loss-augmented inference problems (9), the solutions of which define an optimum $\tilde{z}^*$ of the direction-finding subproblem by

$$\tilde{z}^* = \Big(\sum_{i \in I} \tilde{\beta}_i^*, \sum_{i \in I} \tilde{\gamma}_i^* \Big).$$

Since the dimension of z is the dimension of w plus 1, for the master problems we can choose the dimension limit of RSD to be that number, if it is not too large. (Note that in such a case if the solution of a master problem lies in the relative interior of the simplex, then it is an optimal dual solution; and that points sharing zero weight in the expression of the solution of a master problem have no use in the future and can be discarded.)

(ii) With coordinate ascent, at each iteration we select a working set of training examples $\hat{I}$, and we solve the subproblem and master problem on the space of $\{(\beta_i, \gamma_i), i \in \hat{I}\}$ and with $(\beta_i, \gamma_i), i \notin \hat{I}$ fixed at their current values.

- Applying RSD with a product of simplexes. Utilizing the product form of the feasible region $\prod_{i \in \hat{I}} \mathcal{D}_i$, we approximate it by a product of convex hulls

$$\prod_{i \in \hat{I}} \mathrm{conv}(S^i),$$

where S^i consist of extreme or feasible points of the set $\mathcal{D}^i$. In the direction-finding subproblem of an iteration we solve the inference problem (9) for all training examples in $\hat{I}$. We update the set S^i for each individual example separately and in a way similar to that in the RSD method described earlier (see Footnote 1). The master problems take the form of optimizing the dual function on $\prod_{i \in I} \mathrm{conv}(S^i)$ and have $|\hat{I}|$ simplex constraints instead of one.

The product of convex hulls covers a much larger portion of the feasible region than the single convex hull used in the RSD with one simplex. Thus the RSD with a product of simplexes can make more progress per iteration than its one simplex counterpart, at the expense of the size of master problems that it needs to solve.

We discuss a number of complexity issues of the above dual optimization method. Let r be the dimension limit of the RSD, and let k the the size of the working set of training examples. The master problems are quadratic programming problems with at most $r+1$ variables and one simplex constraint (if we apply RSD with one simplex), or with at most $(r+1) \times k$ variables and k simplex constraints (if we apply RSD with a product of simplexes). The complexity of solving these master problems do not depend on the size and structure of the original problem, once we have computed the extreme points forming the simplexes and the associated matrices in the quadratic objective functions of the master problems. The complexity of computing these latter matrices increases linearly in the working set size, but it does not depend on the structure of the prediction problem.

Finding the extreme points by solving the direction-finding subproblems requires solving the associated loss-augmented inference problems for all training examples in the working set. This computation overhead increases as the size of the working set grows, and furthermore, it implicitly, through the inference algorithm, depends on the dimensionality as well as the structure of the prediction outputs. Thus for most structured prediction tasks finding the extreme points forms the main computation overhead. Consequently, when doing inference is computationally intensive, one may want to save in memory the extreme points so far found, raise the dimension limit in RSD, and correspondingly solve larger master problems, since the latter have fixed complexity independent of that of the inference problem.

Coordinate ascent can be inefficient, thus if computationally affordable, one may want to choose a large working set of examples, or even avoid its use. At later stages of the optimization, one may pick "key" examples into the working set.

3.3 Further Remarks on Reparametrization

Our reparametrization applies to the case where the prediction function depends explicitly on features (instead of kernels) as formulated in Section 2.2. Besides a number of advantages that we already mentioned, we emphasize here a few more of its attractive features:

- In computing the gradients, no multiplication is needed, and it is also not necessary to evaluate the kernels. (The evaluation of kernels is done implicitly and in an efficient way, in the inference problem associated with direction-finding subproblems.) Furthermore, for all examples the gradient components are the same and thus need no separate computation.
- It is not necessary to evaluate kernels for obtaining the matrix associated with the quadratic objective function in the master problems, as that matrix is simply the identity matrix due to the reparametrization.
- To record an extreme point, it is not necessary to record the corresponding label y, while one would have to do so if one works with the dual variables α directly as in [TJHA05].

We also note that our reparametrization can be useful even when the dimension of w is large, so long as for one input, there are typically a relatively small number of components of w involved in the prediction task. Such a case can be one where given an input x, only a relatively small number of features are non-zero for all possible outputs associated with x. (Think of the case where the inputs are sentences, the features and corresponding weight parameters are associated with words. Then only a few words can occur in a given sentence, even though the dictionary of words, thus the number of parameters, is large.)

3.4 Comparison with Related Works

We compare our approach to three other approaches in the context of a linear prediction function as formulated in Section 2.2. We refer readers to the respective references for details of these earlier works. Besides the use of the RSD method, our work differs from the others mostly in our reparametrization approach.

Tsochantaridis et al. [TJHA05] uses the column generation method for dual optimization. Their algorithm successively discovers active primal constraints and includes their corresponding dual variables in the dual optimization. This approach is very close to ours, as it can be equivalently viewed as the simplicial decomposition method applied on the space of the dual variables α directly, without reparametrization. The feasible region of α in one of their cases is approximated by a product of feasible sets of α_i for all examples. The extreme points of these latter feasible sets include the origin and the points of the form $\alpha_i(y) = C$ for ys obtained in solving the direction-finding subproblems. The master problems are to minimize the dual function on the product of the convex hulls of these extreme points. It is argued in [TJHA05] that the size of the master problems would not grow too big, based on a keen observation: at the optimal primal solution, only few constraints can be active, thus few dual variables $\alpha_i(y)$ can be non-zero, due to complementary slackness. Nevertheless, we see that the size of master problems can still grow beyond the range that can be efficiently handled by the current optimization algorithms, thus restricted simplicial decomposition methods need to be employed in some form. Besides this size issue of the master problems, working on the space of α, when the output space is large, can have a number of inefficiencies, such as the storing of the corresponding ys for the extreme points, the unnecessary evaluation of kernels and kernel matrices, as mentioned in Section 3.3.

The reparametrization used in Taskar et al. [TGK04] and Rousu et al. [RSSS06] has the same linear transformation for all examples and can be viewed analogously as the simple operation of marginalization. This reparametrization, however, does not fully simplify the dual problem to the extent as ours has done. (Recall that complex reparametrization does not necessarily complicate the optimization problem, as addressed at the beginning of this section.) The number of dual variables after the reduction in their approach depends on the dimensionality of the prediction outputs y

as well as the structure of the prediction problem. Furthermore, a kernel on the space of the reparametrized variables is introduced, and has to be evaluated in the optimization, both of which are unnecessary in our context.

Taskar et al. [TLJJ05] formulates the primal problem as a minmax problem and solves it by using general algorithms for variational inequalities with monotone operators. Because there are projection operations involved in these algorithms, more complicated variants of the inference problems have to be solved in the optimization, (even though their method, like ours, can exploit both the lower dimensionality of the parameter space and the structure in the prediction outputs). For example, while regular flow/min-cut algorithms can be used for direction-finding in the dual optimization approach (see Section 4), convex-cost flow algorithms are needed for similar tasks in their approach. This is a drawback of their method, as we see it, since solving these subproblems is in all three approaches the main part of the computation overhead.

4 Applications

In this section we demonstrate the application of our approach to several special problems that have been considered in Taskar et al. [TLJJ05] and Tsochantaridis et al. [TJHA05]. Some of these problems have different formulations than the one in Sections 2.2 and 3.

First we consider parameter optimization for the problem of binary segmentation of images, as considered in [TLJJ05]. In this case there are additional sign constraints on w that do not depend on training examples. As these constraints are of the simple type, we analytically optimize away their corresponding dual variables, while we reparametrize the rest of dual variables as before to obtain a reduced dual problem. For this problem as well as the sentence alignment problem, also considered in [TLJJ05], applying dual optimization with simplicial decomposition type of methods, we only need regular min-cut algorithms for solving direction-finding subproblems. By contrast, convex-cost min-cut algorithms are needed in the minmax approach of [TLJJ05].

Next we consider two "slack re-scaling" formulations as considered in [TJHA05]. The constraints of these problems have a different form; also, in one of the formulation – the quadratic penalty for margin violation, the feasible region is unbounded. We demonstrate the application of our reparametrization idea, as well as the form and interpretation of the restricted simplicial decomposition method for the unbounded feasible region case.

4.1 Binary Segmentation of Images

The setting is as in [TLJJ05]. Consider a relatively simple mathematical model for segmenting an object from the background given a 2-dimensional or 3-dimensional image x. Each component of y takes binary values which can represent object or non-object pixel, for instance. The components of y correspond to nodes on a 2-dimensional or 3-dimensional grid with pair-wise edges between adjacent nodes.

The weights w consist of those associated with node features, denoted by w_1, and those associated with edge features, denoted by w_2. The score function is of the form

$$F(x, y; w) = \sum_{n \in N} w_1' \Phi_n(x, y_n) - \sum_{e \in E} w_2' \Phi_e(x) \, \delta(y_{e1} \neq y_{e2}).$$

Here N and E denote the sets of nodes and edges, respectively; Φ_n and Φ_e are node and edge feature vectors indexed by node n and edge e, respectively; y_{e1} and y_{e2} denote the two nodes adjacent to edge e; and $\delta(\cdot)$ denotes the indicator function. The edge feature Φ_e is "turned on" only when the two nodes adjacent to e disagree, i.e., $y_{e1} \neq y_{e2}$, which means that according to y there is an object-background boundary at the location e. Correspondingly Φ_e measures the differences in the image near the location of e, and the scoring term penalizes y if local evidences from x do not support the labels of y_e.

There are several assumptions on the signs of features and weights, as well as on the structure of the loss function. They are for the purpose of maintaining an equivalence relation between some min-cut problems and the inference or loss-augmented inference problems encountered in optimizing the weights w, so that there are efficient algorithms for solving the latter. The details of these assumptions are as follows.

- The edge feature $\Phi_e(x)$ is *non-negative.*
- The edge weight vector w_2 is constrained to be *non-negative.*
- The loss function can be *decomposed into node losses*: $l(x,y) = \sum_{n \in N} d_n(y^x_n, y_n)$, where d_n is a non-negative function and y^x denotes the true label of x.

Under these conditions, the inference problem, with the loss term, is of the form:

$$\max_{y \in \mathcal{Y}_i} \sum_{n \in N} \big(w_1' \Phi_n(x, y_n) + d_n(y^x_n, y_n) \big) - \sum_{e \in E} w_2' \Phi_e(x)\, \delta(y_{e1} \neq y_{e2}),$$

which can be translated into a min-cut problem and thus solved efficiently (Greig et al. [GPS89], Section 2; see also [TLJJ05]).

As before, for notational simplicity, we define $\mathbf{f}_1$, $\mathbf{f}_2$ by collecting terms of the right hand side of the respective equations

$$w_1' \mathbf{f}_1(x^i, y) = \sum_{n \in N} w_1' \Phi_n(x, y_n), \qquad w_2' \mathbf{f}_2 = - \sum_{e \in E} w_2' \Phi_e(x)\, \delta(y_{e1} \neq y_{e2}).$$

And we define their differences relative to the features of x^i with the true label y^i by

$$\Delta \mathbf{f}_1(x^i, y) = \mathbf{f}_1(x^i, y) - \mathbf{f}_1(x^i, y^i), \qquad \Delta \mathbf{f}_2 = \mathbf{f}_2(x^i, y) - \mathbf{f}_2(x^i, y^i).$$

Primal problem and reduced dual formulation

The primal problem is of the same form as before except the non-negative constraints on w_2:

$$\min_{w, \xi} \ \tfrac{1}{2} \|w_1\|^2 + \tfrac{1}{2} \|w_2\|^2 + C \sum_{i \in I} \xi_i \tag{10}$$

$$\text{subj.} \quad w_1' \Delta \mathbf{f}_1(x^i, y) + w_2' \Delta \mathbf{f}_2(x^i, y) + l(x^i, y) \leq \xi_i, \quad \forall\, y \in \mathcal{Y}_i, \quad i \in I, \tag{11}$$

$$w_2 \geq 0, \qquad \xi \geq 0.$$

We reparametrize the dual variables α_i associated with constraints (11) by introducing $\beta_i = (\beta_{i,1}, \beta_{i,2})$ and γ_i defined as

$$\beta_{i,1} = \sum_{y \in \mathcal{Y}_i} \Delta \mathbf{f}_1(x^i, y)\, \alpha_i(y), \tag{12}$$

$$\beta_{i,2} = \sum_{y \in \mathcal{Y}_i} \Delta \mathbf{f}_2(x^i, y)\, \alpha_i(y), \tag{13}$$

$$\gamma_i = \sum_{y \in \mathcal{Y}_i} l(x^i, y)\, \alpha_i(y). \tag{14}$$

Writing the above equations in matrix notation as

$$\begin{bmatrix} \beta_i \\ \gamma_i \end{bmatrix} = A_i \alpha_i,$$

the dual problem of (10) can be written as

$$\begin{aligned} \max_{\beta,\gamma} \quad & -\tfrac{1}{2}\Big\| \sum_{i\in I} \beta_{i,1} \Big\|^2 - \tfrac{1}{2}\Big\| \Big[-\sum_{i\in I} \beta_{i,2} \Big]_+ \Big\|^2 + \sum_{i\in I} \gamma_i \qquad (15) \\ \text{subj.} \quad & \begin{bmatrix} \beta_i \\ \gamma_i \end{bmatrix} \in \mathcal{D}_i = A_i \left(\Big\{ \alpha_i \Big| \sum_{y\in\mathcal{Y}_i} \alpha_i(y) \le C, \quad \alpha_i \ge 0 \Big\} \right), \quad \forall i \in I, \end{aligned}$$

where $[\cdot]_+$ denotes the mapping $\max\{0,\cdot\}$ applied component-wise, and its presence in the dual function is due to the non-negativity constraints on w_2. The dual function (15) is differentiable. An optimal dual solution (β^*, γ^*) relates to the optimal primal solution w^* by

$$w_1^* = -\sum_{i\in I} \beta_{i,1}^*, \qquad w_2^* = \Big[-\sum_{i\in I} \beta_{i,2}^* \Big]_+.$$

Direction-finding subproblems and application of RSD

The direction-finding subproblems can be solved by regular min-cut algorithms. For each i, we denote the gradient of the dual function (15) with respect to (β_i, γ_i) component-wisely as $(g_{i,\beta_1}, g_{i,\beta_2}, g_{i,\gamma})$. At any feasible point $(\bar\beta, \bar\gamma) \in \mathcal{D}$, we have

$$g_{i,\beta_1} = -\sum_{j\in I} \bar\beta_{j,1}, \qquad g_{i,\beta_2} = \Big[-\sum_{j\in I} \bar\beta_{j,2} \Big]_+, \qquad g_{i,\gamma} = 1.$$

Notice that g_{i,β_2} is *non-negative.* The sub-subproblem of direction-finding on $\mathcal{D}_i$ is equivalent to

$$\max_{y\in\mathcal{Y}_i} \quad g'_{i,\beta_1} \, \Delta\mathbf{f}_1(x^i, y) + g'_{i,\beta_2} \, \Delta\mathbf{f}_2(x^i, y) + l(x^i, y),$$

and is further equivalent to, omitting constant terms and using the definition of loss $l(x, y)$,

$$\max_{y\in\mathcal{Y}_i} \quad \sum_{n\in N} \Big(g'_{i,\beta_1} \, \Phi_n(x^i, y_n) + d_n(y^i, y_n) \Big) - \sum_{e\in E} g'_{i,\beta_2} \, \Phi_e(x^i) \, \delta(y_{e1} \neq y_{e2}). \qquad (16)$$

Because of the non-negativity of g_{i,β_2} and features Φ_e, the above problem can be solved efficiently by min-cut algorithms. Notice that we do not need the quadratic-cost min-cut algorithms as needed in the approach of [TLJJ05].

The dual function in this case is non-quadratic, differentiable but not twice differentiable everywhere. Nevertheless, restricted simplicial decomposition methods can be applied in various ways as described in Section 3.2.

4.2 Problems with Loss-Scaled Slack Formulations

Tsochantaridis et al. [TJHA05] considers problem formulations in which the penalty terms for margin violation are quadratic and/or scaled by the loss, and which, as they report, can be sounder and can give more favorable results than using the linear, unscaled penalty terms of Section 2.2. We show that our approach of reparametrization and dual optimization applies to their formulations, which, from the point of view of optimization, differ only slightly from the one in Section 2.2.

As we addressed at the beginning of Section 3, reparametrization by a linear transformation of the dual variables does not change essentially the direction-finding subproblems, so if the structure of these subproblems admits efficient algorithms on the space of α, then it remains to be so in the new coordinates under the linear transformation. Thus we will only show suitable reparametrization for the primal problems considered in [TJHA05], and omit most of the details of applying RSD except for the second variant which has an unbounded feasible region.

Variant I

In a "slack re-scaling" formulation considered by [TJHA05], the primal problem is, in our notation,

$$\min_{w,\xi} \ \tfrac{1}{2}\|w\|^2 + \tfrac{C}{n}\sum_{i\in I}\xi_i \tag{17}$$

$$\text{subj.} \quad w'\Delta\mathbf{f}(x^i,y) \le \frac{\xi_i}{l(x^i,y)} - 1, \quad \xi_i \ge 0, \quad \forall\, y \in \mathcal{Y}_i\backslash y^i, \ \ i \in I.$$

where n is the number of training examples. Let $\alpha_i(y)$ be the non-negative multiplies associated with the constraints for respective $y \in \mathcal{Y}_i\backslash y^i$. Similar to the derivation in Section 3.1, utilizing the linear structure in the Lagrangian function, we introduce variables β_i, γ_i defined by

$$\beta_i = \sum_{y\in\mathcal{Y}_i\backslash y^i} \Delta\mathbf{f}(x^i,y)\,\alpha_i(y), \tag{18}$$

$$\gamma_i = \sum_{y\in\mathcal{Y}_i\backslash y^i} \alpha_i(y). \tag{19}$$

Or, in matrix notation, for some matrix A_i defined through Eqs. (18) and (19), we can write

$$\begin{bmatrix} \beta_i \\ \gamma_i \end{bmatrix} = A_i\alpha_i.$$

The reduced dual problem can be seen as

$$\max_{\beta,\gamma} \ -\tfrac{1}{2}\Big\|\sum_{i\in I}\beta_i\Big\|^2 + \sum_{i\in I}\gamma_i \tag{20}$$

$$\text{subj.} \quad \begin{bmatrix} \beta_i \\ \gamma_i \end{bmatrix} \in \mathcal{D}_i = A_i\left(\Big\{\alpha_i \,\Big|\, \sum_{y\in\mathcal{Y}_i\backslash y^i}\frac{\alpha_i(y)}{l(x^i,y)} \le \tfrac{C}{n}, \ \ \alpha_i \ge 0\Big\}\right), \quad \forall i \in I.$$

Notice that the set $\mathcal{D}_i$ is different from the one in Section 3.1. A dual optimal β^* defines the optimal primal solution w^* by $w^* = -\sum_{i\in I}\beta_i^*$.

There are a number of ways of applying RSD to solve the reduced dual problem, as described in Section 3.2 and will not be repeated here. The direction-finding subproblem can be decomposed into sub-subproblems of direction-finding on $\mathcal{D}_i$ for each individual training example. Let $g_{i,\beta}$ denote the gradient of the dual function with respect to β_i at a dual feasible point. It can be seen, from Eqs. (18) and (19) and the definition of $\mathcal{D}_i$, that the direction finding sub-subproblem on $\mathcal{D}_i$ is equivalent to

$$\max_{\alpha_i\ge 0} \ \sum_{y\in\mathcal{Y}_i\backslash y^i}\alpha_i(y)\,g_{i,\beta}'\,\Delta\mathbf{f}(x^i,y) + \sum_{y\in\mathcal{Y}_i\backslash y^i}\alpha_i(y)$$

$$\text{subj.} \quad \sum_{y\in\mathcal{Y}_i\backslash y^i}\frac{\alpha_i(y)}{l(x^i,y)} \le \tfrac{C}{n},$$

which we rewrite as

$$\max_{\alpha_i\ge 0} \ \sum_{y\in\mathcal{Y}_i\backslash y^i}\frac{\alpha_i(y)}{l(x^i,y)}\,l(x^i,y)\,g_{i,\beta}'\,\Delta\mathbf{f}(x^i,y) + \sum_{y\in\mathcal{Y}_i\backslash y^i}\frac{\alpha_i(y)}{l(x^i,y)}\,l(x^i,y)$$

$$\text{subj.} \quad \sum_{y\in\mathcal{Y}_i\backslash y^i}\frac{\alpha_i(y)}{l(x^i,y)} \le \tfrac{C}{n}.$$

It can be seen, (since $l(x^i,y)$ is positive), a maximum of the above problem is attained either at $\alpha_i = 0$ with the optimal value 0, or at a vector associated with a maximum of the following problem

$$\max_{y\in\mathcal{Y}_i\backslash y^i} \ l(x^i,y)\,g_{i,\beta}'\,\Delta\mathbf{f}(x^i,y) + l(x^i,y).$$

It is not necessary to consider the two cases separately, because $l(x^i, y^i) = 0$ and $\Delta\mathbf{f}(x^i, y^i) = 0$. As can be verified, we can include y^i in the above maximization and solve the loss-augmented inference problem

$$\max_{y \in \mathcal{Y}_i} \; l(x^i, y)\, g'_{i,\beta}\, \Delta\mathbf{f}(x^i, y) + l(x^i, y),$$

then its optimal solution y^* defines the optimal solution $(\tilde{\beta}_i^*, \tilde{\gamma}_i^*)$ of the direction-finding subsubproblem by

$$\tilde{\beta}_i^* = \tfrac{C}{n}\, l(x^i, y^*)\, \Delta\mathbf{f}(x^i, y^*), \qquad \tilde{\gamma}_i^* = \tfrac{C}{n}\, l(x^i, y^*). \tag{21}$$

Variant II

Another formulation in [TJHA05] has a quadratic penalty for margin violation, and its primal problem is, in our notation,[2]

$$\min_{w,\xi} \; \tfrac{1}{2}\|w\|^2 + \tfrac{C}{2n} \sum_{i \in I} \xi_i^2 \tag{22}$$

$$\text{subj.} \quad w' \Delta\mathbf{f}(x^i, y) \leq \frac{\xi_i}{l(x^i, y)} - 1, \quad \xi_i \geq 0, \quad \forall\, y \in \mathcal{Y}_i \backslash y^i, \quad i \in I.$$

Similarly we can introduce variables $\beta_i, \zeta_i, \gamma_i$ defined by

$$\beta_i = \sum_{y \in \mathcal{Y}_i \backslash y^i} \Delta\mathbf{f}(x^i, y)\, \alpha_i(y), \tag{23}$$

$$\zeta_i = \sum_{y \in \mathcal{Y}_i \backslash y^i} \frac{1}{l(x^i, y)}\, \alpha_i(y), \tag{24}$$

$$\gamma_i = \sum_{y \in \mathcal{Y}_i \backslash y^i} \alpha_i(y). \tag{25}$$

Or, in matrix notation, for some matrix A_i defined through Eqs. (23)-(25), we can write

$$\begin{bmatrix} \beta_i \\ \zeta_i \\ \gamma_i \end{bmatrix} = A_i \alpha_i.$$

The reduced dual problem can be seen as

$$\max_{\beta,\zeta,\gamma} \; -\tfrac{1}{2}\Big\| \sum_{i \in I} \beta_i \Big\|^2 - \tfrac{n}{2C} \sum_{i \in I} \zeta_i^2 + \sum_{i \in I} \gamma_i \tag{26}$$

$$\text{subj.} \quad \begin{bmatrix} \beta_i \\ \zeta_i \\ \gamma_i \end{bmatrix} \in \mathcal{D}_i = A_i\left(\{\alpha_i \,|\, \alpha_i \geq 0\}\right), \quad \forall i \in I.$$

The feasible region is unbounded, but the idea of the simplicial decomposition methods still applies, (indeed the column generation algorithm in [TJHA05] can be viewed as the simplicial decomposition method applied on the space of α), and the application of RSD is also straightforward. However, since there are still a few details worth to mention, we give an account of the application of RSD in what follows.

In this case the feasible region $\prod_{i \in I} \mathcal{D}_i$ is a closed convex cone[3] and is also a product of closed convex cones $\mathcal{D}_i$. We successively approximate it from inside by a polyhedral cone, or a product of polyhedral cones, generated by a finite number of direction points (rays) that we find by

[2] They use the square root of the loss in the constraints, but this difference is immaterial for the purpose of describing the forms of the primal and dual problems here.

[3] A cone is a set that is scale-invariant.

solving direction-finding subproblems. Here we remind the readers that the cone generated by the points $\{d_1, \ldots, d_r\}$, which we denote by cone $\big(\{d_1, \ldots d_r\}\big)$, is the set of points that are non-negative combinations of d_i, i.e.,

$$\text{cone}\,\big(\{d_1, \ldots d_r\}\big) = \Big\{ z \,\Big|\, z = \sum_{j=1}^{r} q_j d_j,\ q \geq 0 \Big\}.$$

In the master problems, we optimize the dual function on the approximating polyhedral cones. If we approximate the feasible region by one cone, (instead of a product of cones), the master problems have the form of

$$\max_{q \geq 0} \quad Q\Big(\sum_{j=1}^{r} q_j d_j \Big),$$

where $\{d_1, \ldots, d_r\}$ is the set of direction points. The case of approximation by a product of cones is similar. The projected Newton method [Ber82] is well suited for solving such problems with simple constraints.

We now give the details as well as the interpretation of the direction-finding subproblems. Aiming to find a feasible point along an ascent direction, we consider a bounded subset of the feasible cone $\mathcal{D}_i$, one choice of which can be,[4] for instance,

$$\tilde{\mathcal{D}}_i(K) = A_i \left(\Big\{ \alpha_i \,\Big|\, \alpha_i \geq 0,\ \sum_{y \in \mathcal{Y}_i \backslash y^i} \frac{\alpha_i(y)}{l(x^i, y)} \leq K \Big\} \right),$$

where K is some positive scalar sufficiently large so that the current point lies in $\prod_{i \in I} \tilde{\mathcal{D}}_i(K)$, however, the exact value of K will not be needed anywhere in the algorithm. We can now proceed as before to find extreme points of these bounded sets that are along ascent directions. For each training example i, denote by $g_{i,\beta}$ and $g_{i,\zeta}$ the components of the gradient of the dual funciton (26) with respect to β_i and ζ_i, respectively. We solve the sub-subproblem of direction-finding on $\tilde{\mathcal{D}}_i(K)$:

$$\begin{aligned} \max_{\tilde{\beta}_i, \tilde{\zeta}_i, \tilde{\gamma}_i} \quad & g_{i,\beta}' \, \tilde{\beta}_i + g_{i,\zeta} \, \tilde{\zeta}_i + \tilde{\gamma}_i \\ \text{subj.} \quad & \begin{bmatrix} \tilde{\beta}_i \\ \tilde{\zeta}_i \\ \tilde{\gamma}_i \end{bmatrix} \in \tilde{\mathcal{D}}_i(K). \end{aligned} \tag{27}$$

Similar to the derivation for the first variant, by Eqs. (23)-(25), the problem (27) is equivalent to

$$\begin{aligned} \max_{\alpha_i \geq 0} \quad & \sum_{y \in \mathcal{Y}_i \backslash y^i} \frac{\alpha_i(y)}{l(x^i, y)} \, l(x^i, y) \, g_{i,\beta}' \, \Delta\mathbf{f}(x^i, y) + \sum_{y \in \mathcal{Y}_i \backslash y^i} \frac{\alpha_i(y)}{l(x^i, y)} g_{i,\zeta} + \sum_{y \in \mathcal{Y}_i \backslash y^i} \frac{\alpha_i(y)}{l(x^i, y)} \, l(x^i, y) \\ \text{subj.} \quad & \sum_{y \in \mathcal{Y}_i \backslash y^i} \frac{\alpha_i(y)}{l(x^i, y)} \leq K, \end{aligned}$$

whose maximum is attained either at $\alpha_i = 0$ with the optimal value 0, or at a vector associated with a maximum of the following problem:

$$\max_{y \in \mathcal{Y}_i \backslash y^i} \quad l(x^i, y) \, g_{i,\beta}' \, \Delta\mathbf{f}(x^i, y) + g_{i,\zeta} + l(x^i, y),$$

which is equivalent to, by neglecting the constant term $g_{i,\zeta}$,

$$\max_{y \in \mathcal{Y}_i \backslash y^i} \quad l(x^i, y) \, g_{i,\beta}' \, \Delta\mathbf{f}(x^i, y) + l(x^i, y).$$

[4] A different choice of this bounded subset results a different form of loss-augmented inference problems that need to be solved for direction-finding.

Let y^* be a maximum of the above problem. Then it can be seen that an optimal solution $(\tilde{\beta}_i^*, \tilde{\zeta}_i^*, \tilde{\gamma}_i^*)$ of problem (27) is attained either at the zero vector or at the vector $K(d_{i,\beta}^*, d_{i,\zeta}^*, d_{i,\gamma}^*)$, where $(d_{i,\beta}^*, d_{i,\zeta}^*, d_{i,\gamma}^*)$ can be viewed as a direction and is defined by

$$d_{i,\beta}^* = l(x^i, y^*)\,\Delta\mathbf{f}(x^i, y^*), \qquad d_{i,\zeta}^* = 1, \qquad d_{i,\gamma}^* = l(x^i, y^*). \tag{28}$$

Furthermore, like in the first variant, we can do the following instead: We solve the loss-augmented inference problem

$$\max_{y \in \mathcal{Y}_i} \; l(x^i, y)\, g'_{i,\beta}\, \Delta\mathbf{f}(x^i, y) + l(x^i, y), \tag{29}$$

and with y^* being a maximum and q^* being the optimal value, we let

$$(d_{i,\beta}^*, d_{i,\zeta}^*, d_{i,\gamma}^*) = \begin{cases} \text{defined by Eq. (28)}, & \text{if } y^* \neq y^i \text{ and } q^* \geq -g_{i,\zeta}, \\ (0,0,0) & \text{otherwise.} \end{cases} \tag{30}$$

Then the point $K(d_{i,\beta}^*, d_{i,\zeta}^*, d_{i,\gamma}^*)$ is an optimal solution of problem (27), as can be verified.

To check if the current point (β, ζ, γ) is an optimal dual solution, we check for each i, if the direction from the current point $(\beta_i, \zeta_i, \gamma_i)$ to the point $K(d_{i,\beta}^*, d_{i,\zeta}^*, d_{i,\gamma}^*)$, for *some* K, is an ascent direction (with all other variables fixed at their current values). In other words, we check if there exists $K > 0$ such that

$$g'_{i,\beta}\,(K d_{i,\beta}^* - \beta_i) + g_{i,\zeta}\,(K d_{i,\zeta}^* - \zeta_i) + (K d_{i,\gamma}^* - \gamma_i) > 0,$$

which is equivalent to for some $K > 0$,

$$g'_{i,\beta}\, d_{i,\beta}^* + g_{i,\zeta}\, d_{i,\zeta}^* + d_{i,\gamma}^* > \tfrac{1}{K}\big(g'_{i,\beta}\,\beta_i + g_{i,\zeta}\,\zeta_i + \gamma_i\big).$$

This can be verified by checking the signs of both sides – once more, the knowledge of K is not needed.

If the above relation holds for at least one i, then the current point (β, ζ, γ) is not optimal. In the case of applying RSD with one polyhedral cone approximating the feasible region, we add the direction point $d^* = (d_\beta^*, d_\zeta^*, d_\gamma^*)$ to the set of direction points that generate the approximating cone on which the master problem will be formed. In the case of applying RSD with a product of cones approximating the respective sets $\mathcal{D}_i$, we add the direction points $d_i^* = (d_{i,\beta}^*, d_{i,\zeta}^*, d_{i,\gamma}^*)$ to their respective sets for each individual training example. Applying RSD with coordinate ascent and working sets is similar. When the size of the set of direction points reaches the preset limit, we can replace one direction point with the current point (β, ζ, γ) or $(\beta_i, \zeta_i, \gamma_i)$ before adding in d^* or d_i^*, similar to the procedure in RSD for the bounded case (see Footnote 1).

5 Conclusions

In this paper we have presented a new dual optimization approach for solving the parameter optimization problems that arise from recent large margin discriminative training formulations for structured prediction problems. Our approach is to reparametrize the dual problem and to apply the restricted simplicial decomposition method. We have shown that our reparametrization brings efficiency. While our reparametrization applies to the case where the prediction function depends explicitly on the features, instead of kernels, we note that the restricted simplicial decomposition method remains to be well suited for solving the optimization problems arising from a kernelized formulation.

Acknowledgments

We thank Prof. Dimitri Bertsekas and Sandor Szedmak for helpful discussions. This work is in part supported by the SMART project and by the IST Programme of the European Community, under the PASCAL Network of Excellence, IST-2002-506778.

References

[ATH03] Y. Altun, I. Tsochantaridis, and T. Hofmann, *Hidden Markov support vector machines*, Proc. 20th Int. Conf. Machine Learning, 2003.

[Ber82] D. P. Bertsekas, *Projected Newton methods for optimization problems with simple constraints*, SIAM J. Control and Optimization **20** (1982), no. 2, 221–246.

[Ber99] ______, *Nonlinear programming*, 2nd ed., Athena Scientific, Belmont, MA, 1999.

[Col02] M. Collins, *Discriminative training methods for hidden Markov models*, Proc. Conf. Empirical Methods in Natural Language Processing, 2002.

[GPS89] D. M. Greig, B. T. Porteous, and A. H. Seheult, *Exact maximum a posteriori estimation for binary images*, J. Royal Statistical Society Series B **51** (1989), no. 2, 271–279.

[HLV87] D. W. Hearn, S. Lawphongpanich, and J. A. Ventura, *Restricted simplicial decomposition: Computation and extensions*, Mathematical Programming Study **31** (1987), 99–118.

[Hoh77] B. Von Hohenbalken, *Simplicial decomposition in nonlinear programming algorithms*, Mathematical Programming **13** (1977), 49–68.

[Hol74] C. A. Holloway, *An extension of the Frank-Wolfe method of feasible directions*, Mathematical Programming **6** (1974), 14–27.

[RSSS06] J. Rousu, C. Saunders, S. Szedmak, and J. Shawe-Taylor, *Kernel-based learning of hierarchical multilabel classification models*, J. Machine Learning Research **7** (2006), 1601–1626.

[TGK04] B. Taskar, C. Guestrin, and D. Koller, *Max-margin Markov networks*, Proc. Advances in Neural Information Processing Systems 16, 2004.

[TJHA05] I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun, *Large margin methods for structured and interdependent output variables*, J. Machine Learning Research **6** (2005), 1453–1484.

[TLJJ05] B. Taskar, S. Lacoste-Julien, and M. I. Jordan, *Structured prediction, dual extragradient and Bregman projections*, Technical Report 697, Department of Statistics, University of California, Berkeley, 2005.