Structured Prediction via the Extragradient Method

Ben Taskar
Computer Science
UC Berkeley, Berkeley, CA 94720
taskar@cs.berkeley.edu

Simon Lacoste-Julien
Computer Science
UC Berkeley, Berkeley, CA 94720
slacoste@cs.berkeley.edu

Michael I. Jordan
Computer Science and Statistics
UC Berkeley, Berkeley, CA 94720
jordan@cs.berkeley.edu

Abstract

We present a simple and scalable algorithm for large-margin estimation of structured models, including an important class of Markov networks and combinatorial models. The estimation problem can be formulated as a quadratic program (QP) that exploits the problem structure to achieve polynomial number of variables and constraints. However, off-the-shelf QP solvers scale poorly with problem and training sample size. We recast the formulation as a convex-concave saddle point problem that allows us to use simple projection methods. We show the projection step can be solved using combinatorial algorithms for min-cost convex flow. We provide linear convergence guarantees for our method and present experiments on two very different structured prediction tasks: 3D image segmentation and word alignment, illustrating the favorable scaling properties of our algorithm.

1 Introduction

The scope of discriminative learning methods has been expanding to encompass prediction tasks with increasingly complex structure. Much of this recent development builds upon graphical models to capture sequential, spatial, recursive or relational structure, but as we will discuss in this paper, the structured prediction problem is broader still. For graphical models, two major approaches to discriminative estimation have been explored: (1) maximum conditional likelihood [16, 17] and (2) maximum margin [7, 1, 27]. For the broader class of models that we consider here, the conditional likelihood approach is intractable, but the large margin formulation yields a tractable convex program.

We interpret the term *structured model* very broadly, as a compact scoring scheme over a (possibly very large) set of combinatorial structures and a method for finding the highest scoring structure. In graphical models, the scoring scheme is embodied in a probability distribution over possible assignments of the prediction variables as a function of input variables. In models based on combinatorial problems, the scoring scheme is usually a

simple sum of weights associated with vertices, edges, or other components of a structure; these weights are often represented as parametric functions of a set of features. Given training data consisting of instances labeled by desired structured outputs (e.g., matchings) and a set of features that parameterize the scoring function, the learning problem is to find parameters such that the highest scoring outputs are as close as possible to the desired outputs.

Example of prediction tasks solved via combinatorial optimization problems include bipartite and non-bipartite matching in alignment of 2D shapes [5], word alignment in natural language translation [19] and disulfide connectivity prediction for proteins [3]. All of these problems can be formulated in terms of a tractable optimization problem.

It is also possible to find interesting subfamilies of graphical models for which large-margin methods are tractable whereas likelihood-based methods are not; an example is the class of Markov random fields used for object segmentation in vision [15, 2].

Tractability is not necessarily sufficient to obtain algorithms that work effectively in practice. In particular, although the problem of large margin estimation can be formulated as a quadratic program (QP) in several cases of interest [25, 26], and although this formulation exploits enough of the problem structure so as to achieve a polynomial representation in terms of the number of variables and constraints, off-the-shelf QP solvers scale poorly with problem and training sample size for these models.

In this paper, we present a solution methodology for structured prediction problems that does not require a general-purpose QP solver. Rather, we show that the key computational bottleneck for these problems can be formulated as a min-cost convex flow problem, a problem for which specialized efficient algorithms are available. We illustrate the effectiveness of this approach on two very different large-scale structured prediction tasks: 3D image segmentation and word alignment in translation.

2 Structured models

We begin by discussing two special cases of the general framework that we present subsequently: (1) a class of Markov networks used for segmentation, and (2) a bipartite matching model for word alignment. Despite significant differences in the setup for these models, they share the property that in both cases the problem of finding the highest-scoring output can be formulated as a linear program (LP).

Markov networks. We consider a special class of Markov networks, common in vision applications, in which inference reduces to a tractable min-cut problem [9]. Focusing on binary variables, $\mathbf{y} = \{y_1, \ldots, y_N\}$, and pairwise potentials, we define a joint distribution over $\{0,1\}^N$ via $P(\mathbf{y}) \propto \prod_{j \in \mathcal{V}} \phi_j(y_j) \prod_{jk \in \mathcal{E}} \phi_{jk}(y_j, y_k)$, where $(\mathcal{V}, \mathcal{E})$ is an undirected graph, and where $\{\phi_j(y_j); j \in \mathcal{V}\}$ are the node potentials and $\{\phi_{jk}(y_j, y_k), jk \in \mathcal{E}\}$ are the edge potentials.

In image segmentation (see Fig. 1a), the node potentials capture local evidence about the label of a pixel or laser scan point. Edges usually connect nearby pixels in an image, and serve to correlate their labels. Assuming that such correlations tend to be *positive* (connected nodes tend to have the same label) leads us to restrict the form of edge potentials to be of the form $\phi_{jk}(y_j, y_k) = \exp\{-s_{jk}\mathbb{1}(y_j \neq y_k)\}$, where s_{jk} is a non-negative penalty for assigning y_j and y_k different labels. Expressing node potentials as $\phi_j(y_j) = \exp\{s_j y_j\}$, we have $P(\mathbf{y}) \propto \exp\left\{\sum_{j \in \mathcal{V}} s_j y_j - \sum_{jk \in \mathcal{E}} s_{jk}\mathbb{1}(y_j \neq y_k)\right\}$. Under this restriction of the potentials, it is known that the problem of computing the maximizing assignment, $\mathbf{y}^* = \arg\max P(\mathbf{y} \mid \mathbf{x})$, has a tractable formulation as a min-cut problem [9]. In par-

(a) (b)

Figure 1: Structured prediction applications: (a) Articulated object detection; (b) Word alignment in machine translation.

ticular, we obtain the following LP:

$$\begin{aligned} \max_{z} \quad & \sum_j s_j z_j - \sum_{jk} s_{jk} z_{jk} \\ \text{s.t.} \quad & 0 \le z_j \le 1, \ \forall j \in \mathcal{V}; \\ & z_j - z_k \le z_{jk}, \ \ z_k - z_j \le z_{jk}, \ \forall jk \in \mathcal{E}. \end{aligned} \tag{1}$$

In the above LP, continuous variables z_j correspond to the relaxation of the binary variables y_j. Note that the last line is equivalent to $|z_j - z_k| \le z_{jk}$. Because s_{jk} is positive, $z_{jk} = |z_k - z_j|$ at the maximum, which is equivalent to $\mathbb{1}(z_j \neq z_k)$ if the z_j, z_k variables are binary. An integral optimal solution always exists, since the constraint matrix is totally unimodular [23].

We can parametrize the node and edge weights s_j and s_{jk} in terms of user-provided features $\mathbf{x}_j$ and $\mathbf{x}_{jk}$ associated with the nodes and edges. In particular, in 3D range data, $\mathbf{x}_j$ might be spin image features or spatial occupancy histograms of a point j, while $\mathbf{x}_{jk}$ might include the distance between points j and k, the dot-product of their normals, etc. The simplest model of dependence is a linear combination of features: $s_j = \mathbf{w}_n^\top \mathbf{f}_n(\mathbf{x}_j)$ and $s_{jk} = \mathbf{w}_e^\top \mathbf{f}_e(\mathbf{x}_{jk})$, where $\mathbf{w}_n$ and $\mathbf{w}_e$ are node and edge parameters, and $\mathbf{f}_n$ and $\mathbf{f}_e$ are node and edge feature mappings, of dimension d_n and d_e, respectively.[1] We assume that the feature mappings $\mathbf{f}$ are provided by the user and our goal is to estimate parameters $\mathbf{w}$ from labeled data. We abbreviate the score assigned to a labeling $\mathbf{y}$ for an input $\mathbf{x}$ as $\mathbf{w}^\top \mathbf{f}(\mathbf{x}, \mathbf{y}) = \sum_j y_j \mathbf{w}_n^\top \mathbf{f}_n(\mathbf{x}_j) - \sum_{jk} y_{jk} \mathbf{w}_e^\top \mathbf{f}_e(\mathbf{x}_{jk})$, where $y_{jk} = \mathbb{1}(y_j \neq y_k)$.

Matchings. Consider modeling the task of word alignment of parallel bilingual sentences (Fig. 1b) as a maximum weight bipartite matching problem, where nodes correspond to the words in the two sentences. For simplicity, assume that each word aligns to one or zero words in the other sentence. The edge weight s_{jk} represents the degree to which word j in one sentence can translate into the word k in the other sentence. Our objective is to find an alignment that maximizes the sum of edge scores. We represent a matching using a set of binary variables y_{jk} that are set to 1 if word j is assigned to word k in the other sentence, and 0 otherwise. The score of an assignment is the sum of edge scores:

[1] To ensure non-negativity of s_{jk}, we assume the edge features $\mathbf{f}_e$ to be non-negative and restrict $\mathbf{w}_e \ge 0$. This constraint is easily incorporated into the formulation we present below, but for brevity, we do not include it explicitly.

$s(\mathbf{y}) = \sum_{jk} s_{jk} y_{jk}$. The maximum weight bipartite matching problem, $\arg\max_{\mathbf{y} \in \mathcal{Y}} s(\mathbf{y})$, can be found by solving the following LP:

$$\max_{z} \quad \sum_{jk} s_{jk} z_{jk} \;\; \text{s.t.} \;\; \sum_{j} z_{jk} \leq 1, \quad \sum_{k} z_{jk} \leq 1, \quad 0 \leq z_{jk} \leq 1, \tag{2}$$

where again the continuous variables z_{jk} correspond to the relaxation of the binary variables y_{jk}. As in the min-cut problem, this LP is guaranteed to have integral solutions for any scoring function $s(\mathbf{y})$ [23].

For word alignment, the scores s_{jk} can be defined in terms of the word pair jk and input features associated with $\mathbf{x}_{jk}$. We can include the identity of the two words, relative position in the respective sentences, part-of-speech tags, string similarity (for detecting cognates), etc. We let $s_{jk} = \mathbf{w}^\top \mathbf{f}(\mathbf{x}_{jk})$ for some user-provided feature mapping $\mathbf{f}$ and abbreviate $\mathbf{w}^\top \mathbf{f}(\mathbf{x}, \mathbf{y}) = \sum_{jk} y_{jk} \mathbf{w}^\top \mathbf{f}(\mathbf{x}_{jk})$.

General structure. More generally, we consider prediction problems in which the input $\mathbf{x} \in \mathcal{X}$ is an arbitrary structured object and the output is a vector of values $\mathbf{y} = (y_1, \ldots, y_{L_\mathbf{x}})$, for example, a matching or a cut in the graph. We assume that the length $L_\mathbf{x}$ and the structure of $\mathbf{y}$ depend deterministically on the input $\mathbf{x}$. In our word alignment example, the output space is defined by the length of the two sentences. Denote the output space for a given input $\mathbf{x}$ as $\mathcal{Y}(\mathbf{x})$ and the entire output space as $\mathcal{Y} = \bigcup_{\mathbf{x} \in \mathcal{X}} \mathcal{Y}(\mathbf{x})$.

Consider the class of structured prediction models $\mathcal{H}$ defined by the linear family:

$$h_\mathbf{w}(\mathbf{x}) = \arg\max_{\mathbf{y} \in \mathcal{Y}(\mathbf{x})} \mathbf{w}^\top \mathbf{f}(\mathbf{x}, \mathbf{y}), \tag{3}$$

where $\mathbf{f}(\mathbf{x}, \mathbf{y})$ is a vector of functions $\mathbf{f} : \mathcal{X} \times \mathcal{Y} \mapsto \mathbb{R}^n$. This formulation is very general. Indeed, it is too general for our purposes—for many $\mathbf{f}, \mathcal{Y}$ pairs, finding the optimal $\mathbf{y}$ is intractable. We specialize to the class of models in which the optimization problem in Eq. (3) can be solved in polynomial time; this is still a very large class of models.

3 Max-margin estimation

We assume a set of training instances $S = \{(\mathbf{x}^{(i)}, \mathbf{y}^{(i)})\}_{i=1}^m$, where each instance consists of a structured object $\mathbf{x}^{(i)}$ (such as a graph) and a target solution $\mathbf{y}^{(i)}$ (such as a matching). Consider learning the parameters $\mathbf{w}$ in the conditional likelihood setting. We can define $P_\mathbf{w}(\mathbf{y} \mid \mathbf{x}) = \frac{1}{Z_\mathbf{w}(\mathbf{x})} \exp\{\mathbf{w}^\top \mathbf{f}(\mathbf{x}, \mathbf{y})\}$, where $Z_\mathbf{w}(\mathbf{x}) = \sum_{\mathbf{y}' \in \mathcal{Y}(\mathbf{x})} \exp\{\mathbf{w}^\top \mathbf{f}(\mathbf{x}^{(i)}, \mathbf{y}')\}$, and maximize the conditional log-likelihood $\sum_i \log P_\mathbf{w}(\mathbf{y}^{(i)} \mid \mathbf{x}^{(i)})$, perhaps with additional regularization of the parameters $\mathbf{w}$. However, computing the partition function $Z_\mathbf{w}(\mathbf{x})$ is #P-complete [29, 12] for the two structured prediction problems we presented above, matchings and min-cuts.

Instead, we adopt the max-margin formulation of [27], which directly seeks to find parameters $\mathbf{w}$ such that:

$$\arg\max_{\mathbf{y}_i \in \mathcal{Y}^{(i)}} \mathbf{w}^\top \mathbf{f}(\mathbf{x}^{(i)}, \mathbf{y}) \;\approx\; \mathbf{y}^{(i)}, \quad \forall i,$$

where $\mathcal{Y}^{(i)} = \mathcal{Y}(\mathbf{x}^{(i)})$ and $\mathbf{y}_i$ denotes the appropriate vector of variables for example i. The solution space $\mathcal{Y}^{(i)}$ depends on the structured object $\mathbf{x}^{(i)}$; for example, the space of possible matchings depends on the precise set of nodes and edges in the graph.

As in univariate prediction, we measure the error of prediction using a loss function $\ell(\mathbf{y}^{(i)}, \mathbf{y}_i)$. To obtain a convex formulation, we upper bound the loss $\ell(\mathbf{y}^{(i)}, h_\mathbf{w}(\mathbf{x}^{(i)}))$ using the hinge function: $\max_{\mathbf{y}_i \in \mathcal{Y}^{(i)}} [\mathbf{w}^\top \mathbf{f}_i(\mathbf{y}_i) + \ell_i(\mathbf{y}_i) - \mathbf{w}^\top \mathbf{f}_i(\mathbf{y}^{(i)})]$, where $\ell_i(\mathbf{y}_i) =$

$\ell(\mathbf{y}^{(i)}, \mathbf{y}_i)$, and $\mathbf{f}_i(\mathbf{y}_i) = \mathbf{f}(\mathbf{x}^{(i)}, \mathbf{y}_i)$. Minimizing this upper bound will force the true structure $\mathbf{y}^{(i)}$ to be optimal with respect to $\mathbf{w}$ for each instance i:

$$\min_{||\mathbf{w}|| \leq \gamma} \quad \sum_i \max_{\mathbf{y}_i \in \mathcal{Y}^{(i)}} [\mathbf{w}^\top \mathbf{f}_i(\mathbf{y}_i) + \ell_i(\mathbf{y}_i)] - \mathbf{w}^\top \mathbf{f}_i(\mathbf{y}^{(i)}), \tag{4}$$

where γ is a regularization parameter. Note that this formulation is equivalent to the standard formulation using slack variables ξ and slack penalty C presented in [27][2]

The key to solving Eq. (4) efficiently is the *loss-augmented inference problem*, $\max_{\mathbf{y}_i \in \mathcal{Y}^{(i)}} [\mathbf{w}^\top \mathbf{f}_i(\mathbf{y}_i) + \ell_i(\mathbf{y}_i)]$. This optimization problem has precisely the same form as the prediction problem whose parameters we are trying to learn—$\max_{\mathbf{y}_i \in \mathcal{Y}^{(i)}} \mathbf{w}^\top \mathbf{f}_i(\mathbf{y}_i)$—but with an additional term corresponding to the loss function. Tractability of the loss-augmented inference thus depends not only on the tractability of $\max_{\mathbf{y} \in \mathcal{Y}^{(i)}} \mathbf{w}^\top \mathbf{f}_i(\mathbf{y}_i)$, but also on the form of the loss term $\ell_i(\mathbf{y}_i)$. A natural choice in this regard is the Hamming distance, which simply counts the number of variables in which a candidate solution $\mathbf{y}_i$ differs from the target output $\mathbf{y}^{(i)}$. In general, we need only assume that the loss function decomposes over the variables in $\mathbf{y}^{(i)}$.

For example, in the case of bipartite matchings the Hamming loss counts the number of different edges in the matchings $\mathbf{y}_i$ and $\mathbf{y}^{(i)}$ and can be written as: $\ell_i^H(\mathbf{y}_i) = \sum_{jk} y_{jk}^{(i)} + \sum_{jk}(1 - 2y_{jk}^{(i)})y_{i,jk}$. Thus the loss-augmented matching problem for example i can be written as an LP similar to Eq. (2) (without the constant term $\sum_{jk} y_{jk}^{(i)}$):

$$\begin{aligned} \max_z \quad & \sum_{jk} z_{i,jk}[\mathbf{w}^\top \mathbf{f}(\mathbf{x}_{jk}^{(i)}) + 1 - 2y_{jk}^{(i)}] \\ \text{s.t.} \quad & \sum_j z_{i,jk} \leq 1, \quad \sum_k z_{i,jk} \leq 1, \quad 0 \leq z_{i,jk} \leq 1. \end{aligned}$$

Generally, when we can express $\max_{\mathbf{y}_i \in \mathcal{Y}^{(i)}} \mathbf{w}^\top \mathbf{f}(\mathbf{x}^{(i)}, \mathbf{y}_i)$ as an LP, $\max_{\mathbf{z}_i \in \mathcal{Z}_i} \mathbf{w}^\top \mathbf{F}_i \mathbf{z}_i$, where $\mathcal{Z}_i = \{\mathbf{z}_i : \mathbf{A}_i \mathbf{z}_i \leq \mathbf{b}_i,\ \mathbf{z}_i \geq 0\}$, for appropriately defined constraints $\mathbf{A}_i, \mathbf{b}_i$ and feature matrix $\mathbf{F}_i$, we have a similar LP for the loss-augmented inference for each example i: $d_i + \max_{\mathbf{z}_i \in \mathcal{Z}_i} (\mathbf{w}^\top \mathbf{F}_i + \mathbf{c}_i)^\top \mathbf{z}_i$ for appropriately defined $d_i, \mathbf{F}_i, \mathbf{c}_i, \mathbf{A}_i, \mathbf{b}_i$. Let $\mathbf{z} = \{\mathbf{z}_1, \ldots, \mathbf{z}_m\}$, $\mathcal{Z} = \mathcal{Z}_1 \times \ldots \times \mathcal{Z}_m$.

We could proceed by making use of Lagrangian duality, which yields a joint convex optimization problem; this is the approach explored in [25, 26]. Previous problem-specific algorithms for solving the resulting programs include structured SMO [27, 24] and structured exponentiated gradient [4]. Both algorithms, however, only work for decomposable models (e.g., sequences, trees, triangulated graphs), since they essentially exploit dynamic programming decompositions of such problems. Unfortunately, such decompositions do not hold for matchings and min-cuts (this makes computing the partition function and conditional likelihood estimation intractable).

Instead we take a different tack here, posing the problem in its natural saddle-point form:

$$\min_{||\mathbf{w}|| \leq \gamma} \max_{\mathbf{z} \in \mathcal{Z}} \quad \sum_i \mathbf{w}^\top \mathbf{F}_i \mathbf{z}_i + \mathbf{c}_i^\top \mathbf{z}_i - \mathbf{w}^\top \mathbf{F}_i \mathbf{y}^{(i)}. \tag{5}$$

[2]The correspondence can be seen as follows: let $\mathbf{w}^*(C)$ be a solution to the optimization problem with slack penalty C and define $\gamma(C) = ||\mathbf{w}^*(C)||$. Then $\mathbf{w}^*$ is also a solution to Eq. (4). Conversely, one can invert the mapping $\gamma(\cdot)$ to obtain the set of values of C that give rise to the same solution as Eq. (4) for a specific γ.

4 Extragradient method

Eq. (5) can be written as $\min_{||\mathbf{w}||\leq\gamma} \max_{\mathbf{z}\in\mathcal{Z}} L(\mathbf{w},\mathbf{z})$, where

$$L(\mathbf{w},\mathbf{z}) = \sum_i \mathbf{w}^\top \mathbf{F}_i \mathbf{z}_i + \mathbf{c}_i^\top \mathbf{z}_i - \mathbf{w}^\top \mathbf{F}_i \mathbf{y}^{(i)}.$$

$L(\mathbf{w},\mathbf{z})$ is bilinear in $\mathbf{w}$ and $\mathbf{z}$, with gradient given by: $\nabla_{\mathbf{w}} L(\mathbf{w},\mathbf{z}) = \sum_i \mathbf{F}_i(\mathbf{z}_i - \mathbf{y}^{(i)})$ and $\nabla_{\mathbf{z}_i} L(\mathbf{w},\mathbf{z}) = \mathbf{F}_i^\top \mathbf{w} + \mathbf{c}_i$.

We denote the Euclidean projection of a vector onto $\mathcal{Z}_i$ as $P_{\mathcal{Z}_i}(\mathbf{v}) = \arg\min_{\mathbf{u}\in\mathcal{Z}_i} ||\mathbf{v}-\mathbf{u}||$ and projection onto the ball $||\mathbf{w}|| \leq \gamma$ as $P_\gamma(\mathbf{w}) = \gamma\mathbf{w}/\max(\gamma, ||\mathbf{w}||)$.

We need to solve a saddle-point problem. Consider a simple iterative method based on gradient projections:

$$\mathbf{w}^{k+1} = P_\gamma(\mathbf{w}^k - \beta_k \nabla_{\mathbf{w}} L(\mathbf{w}^k,\mathbf{z}^k)); \qquad \mathbf{z}_i^{k+1} = P_{\mathcal{Z}_i}(\mathbf{z}_i^k + \beta_k \nabla_{\mathbf{z}_i} L(\mathbf{w}^k,\mathbf{z}^k)),$$

where β_k is a step size. This method is generally not guaranteed to converge for bilinear objectives [14, 13, 11].

A well-known solution strategy for saddle-point optimization is provided by the *extragradient method* [14]. An iteration of the extragradient method consists of two very simple steps, prediction and correction:

$$\text{(Prediction)} \quad \left\{ \begin{array}{rcl} \bar{\mathbf{w}}^{k+1} &=& P_\gamma(\mathbf{w}^k - \beta_k \nabla_{\mathbf{w}} L(\mathbf{w}^k,\mathbf{z}^k)); \\ \bar{\mathbf{z}}_i^{k+1} &=& P_{\mathcal{Z}_i}(\mathbf{z}_i^k + \beta_k \nabla_{\mathbf{z}_i} L(\mathbf{w}^k,\mathbf{z}^k)); \end{array} \right. \tag{6}$$

$$\text{(Correction)} \quad \left\{ \begin{array}{rcl} \mathbf{w}^{k+1} &=& P_\gamma(\mathbf{w}^k - \beta_k \nabla_{\mathbf{w}} L(\bar{\mathbf{w}}^{k+1},\bar{\mathbf{z}}^{k+1})); \\ \mathbf{z}_i^{k+1} &=& P_{\mathcal{Z}_i}(\mathbf{z}_i^k + \beta_k \nabla_{\mathbf{z}_i} L(\bar{\mathbf{w}}^{k+1},\bar{\mathbf{z}}^{k+1})). \end{array} \right. \tag{7}$$

which translates into:

$$(P)\ \bar{\mathbf{w}}^{k+1} = P_\gamma(\mathbf{w}^k + \beta_k \sum_i \mathbf{F}_i(\mathbf{y}^{(i)} - \mathbf{z}_i^k)); \qquad \bar{\mathbf{z}}_i^{k+1} = P_{\mathcal{Z}_i}(\mathbf{z}_i^k + \beta_k(\mathbf{F}_i^\top \mathbf{w}^k + \mathbf{c}_i));$$

$$(C)\ \mathbf{w}^{k+1} = P_\gamma(\mathbf{w}^k + \beta_k \sum_i \mathbf{F}_i(\mathbf{y}^{(i)} - \bar{\mathbf{z}}_i^{k+1})); \quad \mathbf{z}_i^{k+1} = P_{\mathcal{Z}_i}(\mathbf{z}_i^k + \beta_k(\mathbf{F}_i^\top \bar{\mathbf{w}}^{k+1} + \mathbf{c}_i)),$$

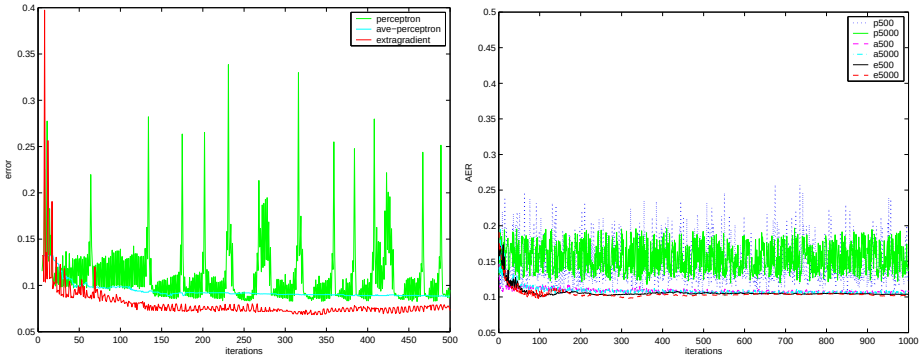
where β_k is an appropriately chosen step size. The algorithm starts with a feasible point $\mathbf{w}^0 = 0, \mathbf{z}_i^0 = \mathbf{y}^{(i)}$ and step size $\beta_0 = 1$. After each prediction step, it computes

$$r_k = \beta_k \frac{||\nabla L(\mathbf{w}^k,\mathbf{z}^k) - \nabla L(\bar{\mathbf{w}}^{k+1},\bar{\mathbf{z}}^{k+1})||}{\sqrt{||\mathbf{w}^k - \bar{\mathbf{w}}^{k+1}||^2 + ||\mathbf{z}^k - \bar{\mathbf{z}}^{k+1}||^2)}}. \tag{8}$$

If r_k is greater than a threshold ν, the step size is decreased using an Armijo type rule: $\beta_k = (2/3)\beta_k \min(1, 1/r_k)$, and a new prediction step is computed until $r_k \leq \nu$, where $\nu \in (0,1)$ is a parameter of the algorithm. Under mild conditions, the method is guaranteed to converge linearly to a solution $\mathbf{w}^*, \mathbf{z}^*$ [14, 11]. See Appendix A for details.

The key step influencing the efficiency of the algorithm is the Euclidean projection onto the feasible sets $\mathcal{Z}_i$. In case of word alignment, $\mathcal{Z}_i$ is the convex hull of bipartite matchings and the problem reduces to the much-studied minimum cost quadratic flow problem [6]. The projection problem $P_{\mathcal{Z}_i}(\mathbf{z}_i')$ is given by

$$\begin{array}{ll} \min\limits_z & \sum\limits_{jk} \frac{1}{2}(z'_{i,jk} - z_{i,jk})^2 \\ \text{s.t.} & \sum\limits_j z_{i,jk} \leq 1, \quad \sum\limits_k z_{i,jk} \leq 1, \quad 0 \leq z_{i,jk} \leq 1. \end{array}$$



(a) (b)

Figure 2: (a) Object segmentation test error rate for the perceptron, the averaged perceptron and the extragradient. (b) Alignment error rate (on gold dataset) for the perceptron, the averaged perceptron and the extragradient, named p, a and e respectively (from up to bottom on image). 500 and 5000 refer to the number of sentences in the training data. This legend is used for the remaining plots.

We can now use a standard reduction of bipartite matching to min cost flow by introducing a source node connected to all the words in one sentence and a sink node connected to all the words in the other sentence, using edges of capacity 1 and cost 0. The original edges jk have a quadratic cost $\frac{1}{2}(z'_{i,jk} - z_{i,jk})^2$ and capacity 1. Now the minimum cost flow from the source to the sink computes projection of $\mathbf{z}'_i$ onto $\mathcal{Z}_i$.

The reduction of the min-cut polytope projection to a convex network flow problem can also be done and is shown in Appendix B. Algorithms for solving this problem are nearly as efficient as those for solving regular min-cost flow problems. We use standard, publicly-available code for solving this problem [10][3].

5 Experiments

Object segmentation. We tested our algorithm for 3D scan segmentation using the restricted class of MRFs we described above. The dataset is a challenging collection of cluttered scenes containing articulated wooden puppets. It contains eleven different single-view scans of three puppets of different sizes and in different positions. It also contains clutter and occluding objects such as rope, sticks and rings. Each scan has around $225,000$ points, which we subsampled to around $7,000$ points with standard software. Our goal was to segment the scenes into two classes—puppet and background. Five of the scenes comprise our training set, and the rest are kept for testing. Sample scans from the training and test set can be seen at `http://www.cs.berkeley.edu/~taskar/3DSegment/`. We used features described in [2] for node potentials, and for edge potentials we used the surface links that are output by the scanner. Figure 2 shows test set error as a function of (batch) iterations the extragradient algorithm and the two versions of the perceptron algorithm of [7] (standard perceptron and the averaged perceptron). We used five scenes as the training data, containing appropriately $46,000$ nodes and $107,000$ edges. Training time took about an hour on a 2.80GHz Pentium 4 machine. Extragradient is much more stable than regular perceptron and has a consistently lower error rate (about 1% absolute difference) than the averaged perceptron.

[3] Available from http://www.math.washington.edu/∼tseng/netflowg_nl/.

Word alignment. We also tested our matching algorithm on word-level alignment using a data set from the 2003 NAACL set [20], the English-French Hansards task. This corpus consists of 1.1M automatically aligned sentences, and comes with a validation set of 39 sentence pairs and a test set of 447 sentences. The validation and test sentences have been hand-aligned and are marked with both *sure* and *possible* alignments. Using these alignments, *alignment error rate* (AER) is calculated as:

$$AER(A, S, P) = 1 - \frac{|A \cap S| + |A \cap P|}{|A| + |S|}$$

Here, A is a set of proposed index pairs, S is the set of sure gold pairs, and P is the set of possible gold pairs (where $S \subseteq P$).

Since our method is a supervised algorithm, we need labeled examples. We used GIZA++ [21] to produce intersected bidirectional model 4 alignments for the unlabeled sentence pairs. We then took the first 5K sentence pairs from these 1.1M model 4 alignments. This gave us more training data, but of course the labels were noisier. However, since most of our features were fairly general, this seemed to work just as well (results below).

For the feature design, first, and most importantly, we want to include information about word association; translation pairs are likely to co-occur in a bilingual text. This feature can be captured using the Dice coefficient:

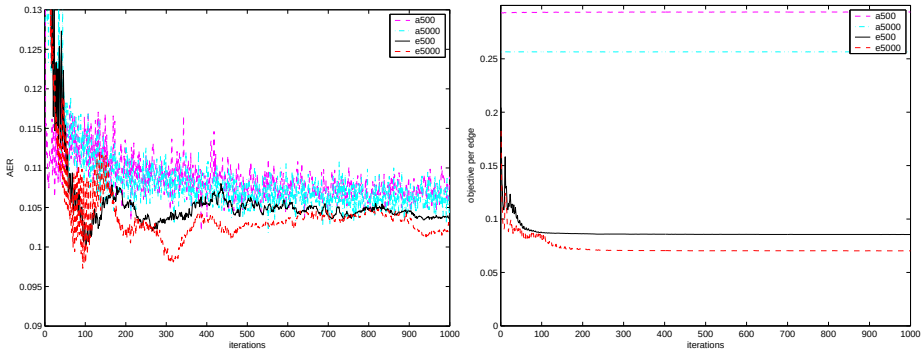
$$dice(e, f) = \frac{2C_{EF}(e, f)}{C_E(e)C_F(f)}$$

Here, C_E and C_F are counts of word occurrences in each language, while C_{EF} is the number of co-occurrences of the two words. With just this feature on a pair of word tokens (which depends only on their types), we can already make a stab at word alignment, aligning, say, each English word with the French word (or null) with the highest Dice value, simply as a matching-free heuristic model. In addition to these features, we included other kinds of information, including difference between word positions, word-similarity features designed to capture cognate (and exact match) information, and identity of the top five most frequently occurring words. In Fig. 2b, we compare test word alignment error rates using the extragradient method and the two versions of the perceptron algorithm (standard perceptron and the averaged perceptron). We plot error for two different training set sizes for both algorithms, with 500 and 5000 sentences, which correspond to 37,000 and 555,000 edges respectively. It took roughly 10 minutes and 3 hours respectively to do 500 training iterations on those training sets using a 2.8GHz Pentium 4 machine, showing the linear scaling of the algorithm (linear in the number of edges). This graph shows the significant oscillation in the performance of the perceptron, as can be expected from non-separable data. A zoomed version of the graph is shown in Fig. 3a in order to compare the averaged perceptron with the extragradient more closely. Their performance is similar, though the extragradient does slightly better (3% better in average). The slightly larger oscillation for the averaged perceptron could be due to the fact that it was implemented as an online algorithm whereas we implemented the extragradient in a batch fashion. Fig. 3b shows the evolution of the objective:

$$\sum_{i,jk} y_{jk}^{(i)} + \max_{\mathbf{z} \in \mathcal{Z}} L(\mathbf{w}, \mathbf{z}) \tag{9}$$

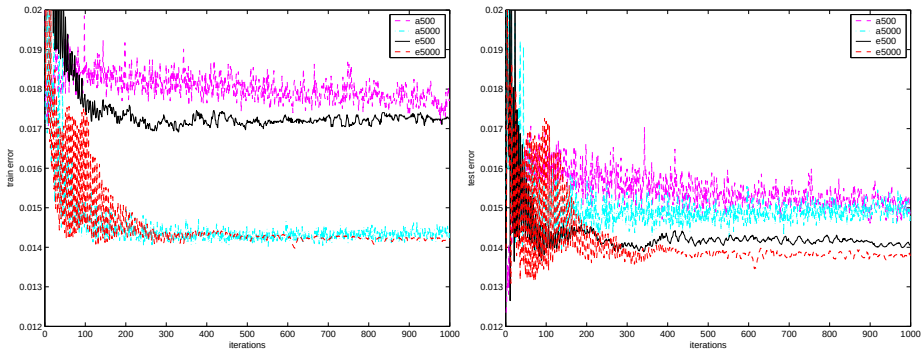
which is the hinge upper bound of the loss function that we are minimizing in Eq. (5). From it, we can see that the extragradient seemed to have converged within the first 500 iterations.

Implementations details and other observations. We plot a measure which is better related to the objective we are trying to optimize in Fig. 4: the error rate is defined as



(a) (b)

Figure 3: (a) Alignment error rate of the averaged perceptron compared to the extragradient on the gold word alignment data (zoomed-in version of Fig. 2). (b) Objective of Eq. (9) divided by the number of edges for the averaged perceptron and the extragadient on the word alignment training data.



(a) (b)

Figure 4: (a) **Training error.** The error rate for the word alignment data is defined as the number of correctly labelled edges divided by the total number of edges. (b) **Test error.** Same as a) but for the gold dataset (considering the 'possible' edges as labelled with 0).

the number of correctly labelled edges over the total number of edges. This is an overly optimistic measure as most edges are 0 by the constraint of our formulation (a word aligns to zero or one word in the other sentence), but its evolution over time can be interesting. The training error of the averaged perceptron and the extragradient are similar, but the extragradient performs consistently better after convergence on the test data (8% better in average), probably because it is maximizing the margin. Note that the training error is greater on the 500 sentences than on the 5000 sentences because the average length was longer for the 5000 sentences and so the proportion of 0 edges was greater for the 5000 sentences.

In our experiments, we used a very large value of γ (1000-10000), the limit on the norm for the parameter vector. The particular value chosen did not seem to have a significant effect on performance.

6 Conclusion

We have presented a general solution strategy for large-scale structured prediction problems. We have shown that these problems can be formulated as saddle-point optimization problems, problems that are amenable to solution by the extragradient algorithm. Key to our approach is the recognition that the projection step in the extragradient algorithm can be solved by network flow algorithms. Network flow algorithms are among the most well-developed in the field of combinatorial optimization, and yield stable, efficient algorithmic platforms. We have exhibited the favorable scaling of this overall approach in two concrete, large-scale learning problems. It is also important to note that the general approach extends to a much broader class of problems.

Acknowledgements

We would like to thank Paul Tseng for kindly answering our questions about the minimum cost quadratic flow implementation.

References

[1] Y. Altun, I. Tsochantaridis, and T. Hofmann. Hidden Markov support vector machines. In *Proc. ICML*, 2003.

[2] D. Anguelov, B. Taskar, V. Chatalbashev, D. Koller, D. Gupta, G. Heitz, and A. Ng. Discriminative learning of Markov random fields for segmentation of 3d scan data. In *Proc. CVPR*, 2005.

[3] P. Baldi, J. Cheng, and A. Vullo. Large-scale prediction of disulphide bond connectivity. In *Proc. NIPS*, 2004.

[4] P. Bartlett, M. Collins, B. Taskar, and D. McAllester. Exponentiated gradient algorithms for large-margin structured classification. In *NIPS*, 2004.

[5] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24, 2002.

[6] D. P. Bertsekas and P. Tseng L. C. Polymenakos. An e-relaxation method for separable convex cost network flow problems. *SIAM J. Optim.*, 7(3):853–870, 1997.

[7] M. Collins. Discriminative training methods for hidden Markov models: Theory and experiments with perceptron algorithms. In *Proc. EMNLP*, 2002.

[8] R. Glowinski. *Numerical Methods for Nonlinear Variational Problems*. Springer, 1984.

[9] D. M. Greig, B. T. Porteous, and A. H. Seheult. Exact maximum a posteriori estimation for binary images. *J. R. Statist. Soc. B*, 51, 1989.

[10] F. Guerriero and P. Tseng. Implementation and test of auction methods for solving generalized network flow problems with separable convex cost. *Journal of Optimization Theory and Applications*, 115(1):113–144, October 2002.

[11] B.S. He and L. Z. Liao. Improvements of some projection methods for monotone nonlinear variational inequalities. *JOTA*, 112:111:128, 2002.

[12] M. Jerrum and A. Sinclair. Polynomial-time approximation algorithms for the Ising model. *SIAM J. Comput.*, 22, 1993.

[13] E. Khobotov. Modification of the extra-gradient method for solving variational inequalities and certain optimization problems. *USSR Comput. Math. & Math. Phys.*, 27(5):120–127, 1989.

[14] G. M. Korpelevich. The extragradient method for finding saddle points and other problems. *Ekonomika i Matematicheskie Metody*, 12:747:756, 1976.

[15] S. Kumar and M. Hebert. Discriminative fields for modeling spatial dependencies in natural images. In *NIPS*, 2003.

[16] J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML*, 2001.

[17] J. Lafferty, X. Zhu, and Y. Liu. Kernel conditional random fields: Representation and clique selection. In *ICML*, 2004.

[18] Z.-Q. Luo and P. Tseng. Error bound and convergence analysis of matrix splitting algorithms for the affine variational inequality problem. *SIAM J. Optimization*, 2, 1992.

[19] E. Matusov, R. Zens, and H. Ney. Symmetric word alignments for statistical machine translation. In *Proc. COLING*, 2004.

[20] R. Mihalcea and T. Pedersen. An evaluation exercise for word alignment. In *Proceedings of the HLT-NAACL 2003 Workshop, Building and Using parallel Texts: Data Driven Machine Translation and Beyond*, pages 1–6, Edmonton, Alberta, Canada, 2003.

[21] Franz Och and Hermann Ney. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–52, 2003.

[22] S.M. Robinson. Bounds for errors in the solution set of a perturbed linear program. *Linear Algebra and its Applications*, 6, 1973.

[23] A. Schrijver. *Combinatorial Optimization: Polyhedra and Efficiency*. Springer, 2003.

[24] B. Taskar. *Learning Structured Prediction Models: A Large Margin Approach*. PhD thesis, Stanford University, 2004.

[25] B. Taskar, V. Chatalbashev, and D. Koller. Learning associative Markov networks. In *Proc. ICML*, 2004.

[26] B. Taskar, V. Chatalbashev, D. Koller, and C. Guestrin. Learning structured prediction models: a large margin approach. In *ICML*, 2005.

[27] B. Taskar, C. Guestrin, and D. Koller. Max margin Markov networks. In *NIPS*, 2003.

[28] P. Tseng. On linear convergence of iterative methods for the variational inequality problem. *Computational and Applied Mathematics*, 60, 1995.

[29] L. G. Valiant. The complexity of computing the permanent. *Theoretical Computer Science*, 8:189–201, 1979.

A Convergence analysis

the convergence analysis of the extragradient method is based on a more general framework of variational inequalites [8]. We begin by establishing some notation and definitions. Because the optimization is over a convex set, a feasible point $(\mathbf{w}^*, \mathbf{z}^*)$, with $||\mathbf{w}^*|| \leq \gamma, \mathbf{z}^* \in \mathcal{Z}$, is a solution of Eq. (5) if and only if

$$\begin{aligned} \nabla_{\mathbf{w}} L(\mathbf{w}^*, \mathbf{z}^*)^\top (\mathbf{w} - \mathbf{w}^*) \geq 0, \qquad & \forall \mathbf{w} : ||\mathbf{w}|| \leq \gamma; \\ \nabla_{\mathbf{z}_i} L(\mathbf{w}^*, \mathbf{z}^*)^\top (\mathbf{z}_i - \mathbf{z}_i^*) \leq 0, \qquad & \forall i, \forall \mathbf{z}_i \in \mathcal{Z}_i. \end{aligned}$$

Let us combine $\mathbf{w}$ and $\mathbf{z}$ into one vector:

$$\mathbf{u} = \begin{pmatrix} \mathbf{w} \\ \mathbf{z}_1 \\ \vdots \\ \mathbf{z}_m \end{pmatrix}; \qquad \mathcal{U} = \left\{ \begin{pmatrix} \mathbf{w} \\ \mathbf{z}_1 \\ \vdots \\ \mathbf{z}_m \end{pmatrix} : ||\mathbf{w}|| \leq \gamma, \mathbf{z}_i \in \mathcal{Z}_i, \quad \forall i \right\},$$

and denote the projection operator onto $\mathcal{U}$ as $P_{\mathcal{U}}(\mathbf{v}) = \arg\min_{\mathbf{u} \in \mathcal{U}} ||\mathbf{u} - \mathbf{v}||$. Note that $\mathcal{U}$ is convex since it is a direct product of convex sets. We denote the (affine) gradient operator on this joint space as

$$T(\mathbf{u}) = \begin{pmatrix} \nabla_{\mathbf{w}} L(\mathbf{w}, \mathbf{z}) \\ -\nabla_{\mathbf{z}_1} L(\mathbf{w}, \mathbf{z}) \\ \vdots \\ -\nabla_{\mathbf{z}_m} L(\mathbf{w}, \mathbf{z}) \end{pmatrix} = \begin{pmatrix} 0 & \mathbf{F}_1 \cdots \mathbf{F}_m \\ -\mathbf{F}_1^\top & \\ \vdots & 0 \\ -\mathbf{F}_m^\top & \end{pmatrix} \begin{pmatrix} \mathbf{w} \\ \mathbf{z}_1 \\ \vdots \\ \mathbf{z}_m \end{pmatrix} - \begin{pmatrix} \sum \mathbf{F}_i \mathbf{y}^{(i)} \\ \mathbf{c}_1 \\ \vdots \\ \mathbf{c}_m \end{pmatrix} = \mathbf{F}\mathbf{u} - \mathbf{a}.$$

With these definitions, the optimality condition for a point $\mathbf{u}^* \in \mathcal{U}$ becomes:

$$T(\mathbf{u}^*)^\top (\mathbf{u}' - \mathbf{u}^*) \geq 0, \qquad \forall \mathbf{u}' \in \mathcal{U}. \tag{10}$$

We denote the set of solutions satisfying these conditions as $\mathcal{U}^*$. Problems of this form are called variational inequalities and arise in many important applications such as optimization, solving system of equations and market equilibrium [8].

Let us define a residual error function for Eq. (10):

$$e(\mathbf{u}, \beta) = \mathbf{u} - P_{\mathcal{U}}(\mathbf{u} - \beta T(\mathbf{u})).$$

Note that for $\beta > 0$, the zeros of $e(\mathbf{u}, \beta)$ are precisely the solutions of Eq. (10):

$$e(\mathbf{u}, \beta) = 0 \qquad \Leftrightarrow \qquad \mathbf{u} \in \mathcal{U}^*. \tag{11}$$

This is a consequence of a characterization of the projection of an arbitrary point $\mathbf{v}$ onto a convex set $\mathcal{U}$; let $\mathbf{u} \in \mathcal{U}$, then we have:

$$(\mathbf{v} - \mathbf{u})^\top (\mathbf{u}' - \mathbf{u}) \leq 0, \ \forall \mathbf{u}' \in \mathcal{U} \qquad \Leftrightarrow \qquad \mathbf{u} = P_{\mathcal{U}}(\mathbf{v}). \tag{12}$$

Assume $e(\mathbf{u}, \beta) = 0, \beta > 0$ and let $\mathbf{v} = \mathbf{u} - \beta T(\mathbf{u})$. Since $e(\mathbf{u}, \beta) = 0$, $\mathbf{u} = P_{\mathcal{U}}(\mathbf{v}) = P_{\mathcal{U}}(\mathbf{u} - \beta T(\mathbf{u}))$. Plugging $\mathbf{v}$ in Eq. (12), shows that $\mathbf{u}$ satisfies the optimality condition in Eq. (10). Conversely, assume $\mathbf{u}^*$ satisfies Eq. (10) and let $\mathbf{u}^+ = P_{\mathcal{U}}(\mathbf{u}^* - \beta T(\mathbf{u}^*))$. By Eq. (12), we have

$$(\mathbf{u}^* - \beta T(\mathbf{u}^*) - \mathbf{u}^+)^\top (\mathbf{u}' - \mathbf{u}^+) = (\mathbf{u}^* - \mathbf{u}^+)^\top (\mathbf{u}' - \mathbf{u}^+) - \beta T(\mathbf{u}^*)^\top (\mathbf{u}' - \mathbf{u}^+) \leq 0, \ \forall \mathbf{u}' \in \mathcal{U}.$$

In particular, for $\mathbf{u}' = \mathbf{u}^*$, we have $||\mathbf{u}^* - \mathbf{u}^+||^2 \leq \beta T(\mathbf{u}^*)^\top (\mathbf{u}^* - \mathbf{u}^+)$. Since the right-hand-side, $\beta T(\mathbf{u}^*)^\top (\mathbf{u}^* - \mathbf{u}^+)$, is non-positive by Eq. (10), and the left-hand-side is non-negative, we have $\mathbf{u} = \mathbf{u}^+$ and hence $e(\mathbf{u}, \beta) = 0$.

We will make use of the following theorem (which holds for more general operators $T(\mathbf{u})$) to prove the convergence of the extragradient method for structured prediction problems.

Theorem A.1 (Theorem 2.1 in [11]) *Let $\{\beta_k\}$ be a sequence such that $\inf_k\{\beta_k\} = \beta_{min} > 0$ and let $c_0 > 0$ be a constant. If the sequence $\{\mathbf{u}^k\}$ satisfies:*

$$||\mathbf{u}^{k+1} - \mathbf{u}^*||^2 \leq ||\mathbf{u}^k - \mathbf{u}^*||^2 - c_0||e(\mathbf{u}^k, \beta_k)||^2, \qquad \forall \mathbf{u}^* \in \mathcal{U}^*, \tag{13}$$

then $\{\mathbf{u}^k\}$ converges to a solution point of Eq. (10).

The extragradient algorithm can be written as:

$$(P) \qquad \bar{\mathbf{u}}^{k+1} = P_{\mathcal{U}}(\mathbf{u}^k - \beta_k T(\mathbf{u}^k));$$
$$(C) \qquad \mathbf{u}^{k+1} = P_{\mathcal{U}}(\mathbf{u}^k - \beta_k T(\bar{\mathbf{u}}^{k+1})),$$

with $r_k = \beta_k||T(\mathbf{u}^k) - T(\bar{\mathbf{u}}^{k+1})||/||\mathbf{u}^k - \bar{\mathbf{u}}^{k+1}||$. Under the assumption that $r_k < \nu < 1$, Korpelevich [14] showed that for the extragradient, $c_0 = (1-\nu)^2$ in Eq. (13):

$$||\mathbf{u}^{k+1} - \mathbf{u}^*||^2 \leq ||\mathbf{u}^k - \mathbf{u}^*||^2 - (1-\nu)^2||e(\mathbf{u}^k, \beta_k)||^2, \qquad \forall \mathbf{u}^* \in \mathcal{U}^*,$$

It remains to show that we can lower bound β_k away from zero for the Armijo type rule $\beta_k = (2/3)\beta_k \min(1, 1/r_k)$. Note that

$$\max_{\mathbf{u},\mathbf{v}} \frac{||T(\mathbf{u}) - T(\mathbf{v})||}{||\mathbf{u} - \mathbf{v}||} = \max_{||\mathbf{u}||=1} ||\mathbf{F}\mathbf{u}|| = ||\mathbf{F}||,$$

where $||\mathbf{F}||$ is the standard matrix norm induced by the Euclidean vector norm $||\cdot||$. Hence, $\beta_k < \nu/||\mathbf{F}||$ ensures $r_k < \nu$, which leaves β_k constant and away from zero. The norm of $\mathbf{F}$ ultimately depends on the problem (mincut or matching), the number of examples and the feature map $\mathbf{f}(\mathbf{x}, \mathbf{y})$.

The local linear convergence of the extragradient method is well established for affine $T(\mathbf{u})$ and polyhedral $\mathcal{U}$ [18, 28]. If the norm constraint $||\mathbf{w}|| \leq \gamma$ is not active near the limit point of the sequence, this is effectively our setting[4]. If the norm constraint is active, the situation is more complicated.

Let $d(\mathbf{u}, \mathcal{U}^*)$ be the shortest distance from $\mathbf{u}$ to a solution of Eq. (10): $d(\mathbf{u}, \mathcal{U}^*) = \inf_{\mathbf{u}^* \in \mathcal{U}^*} ||\mathbf{u} - \mathbf{u}^*||$. Note that Eq. (13) implies that

$$d(\mathbf{u}^{k+1}, \mathcal{U}^*)^2 \leq d(\mathbf{u}^k, \mathcal{U}^*)^2 - c_0||e(\mathbf{u}^k, \beta_k)||^2. \tag{14}$$

To prove local linear convergence, it suffices to show that for some constants $\epsilon > 0, c_1 > 0$,

$$d(\mathbf{u}, \mathcal{U}^*) \leq c_1||e(\mathbf{u}, \beta_k)||, \quad \forall \mathbf{u} \in \mathcal{U}, \qquad \text{whenever} \qquad ||e(\mathbf{u}, \beta_k)|| \leq \epsilon,$$

which implies that for sufficiently large k,

$$\frac{d(\mathbf{u}^{k+1}, \mathcal{U}^*)}{d(\mathbf{u}^k, \mathcal{U}^*)} \leq \sqrt{1 - \frac{c_0}{c_1}} \tag{15}$$

For polyhedral spaces $\mathcal{U} = \{\mathbf{u} : \mathbf{u} \geq 0;\ \mathbf{A}\mathbf{u} \leq \mathbf{b}\}$, the constant is given by

$$c_1 = ||\mathbf{A}^\top||/\lambda(\mathbf{B}\mathbf{B}^\top),$$

where $\lambda(\mathbf{B}\mathbf{B}^\top)$ is the smallest eigenvalue of the matrix $\mathbf{B}\mathbf{B}^\top$, with $\mathbf{B}$ formed by linearly independent rows of $\mathbf{A}$ corresponding to constraints active at the solution [22, 18].

For the case of active norm constraint $||\mathbf{w}|| \leq \gamma$ near the limit point of the sequence, the feasible region $\mathcal{U}$ is not polyhedral, and no linear convergence proofs are known to us. However, this setting is equivalent to the penalized version of the problem, where the constraint is moved into the objective:

$$\min_{\mathbf{w}} \max_{\mathbf{z} \in \bar{\mathcal{Z}}} \quad \frac{1}{2C}||\mathbf{w}||^2 + \sum_i \mathbf{w}^\top \mathbf{F}_i \mathbf{z}_i + \mathbf{c}_i^\top \mathbf{z}_i - \mathbf{w}^\top \mathbf{F}_i \mathbf{y}^{(i)}. \tag{16}$$

[4]Which was the case in our experiments as we have used a large γ.

In this case, $L(\mathbf{w},\mathbf{z}) = \frac{1}{2C}||\mathbf{w}||^2 + \sum_i \mathbf{w}^\top \mathbf{F}_i \mathbf{z}_i + \mathbf{c}_i^\top \mathbf{z}_i - \mathbf{w}^\top \mathbf{F}_i \mathbf{y}^{(i)}$, with gradient given by: $\nabla_{\mathbf{w}} L(\mathbf{w},\mathbf{z}) = \frac{\mathbf{w}}{C} + \sum_i \mathbf{F}_i(\mathbf{z}_i - \mathbf{y}^{(i)})$ and $\nabla_{\mathbf{z}_i} L(\mathbf{w},\mathbf{z}) = \mathbf{F}_i^\top \mathbf{w} + \mathbf{c}_i$. Letting $\mathbf{u}$ and $\mathcal{U}$ be as before, but omitting the norm constraint on $\mathbf{w}$, we define the gradient operator which differs from the previous definition only in the upper left corner of $\mathbf{F}$:

$$\tilde{T}(\mathbf{u}) = \begin{pmatrix} \nabla_{\mathbf{w}} L(\mathbf{w},\mathbf{z}) \\ -\nabla_{\mathbf{z}_1} L(\mathbf{w},\mathbf{z}) \\ \vdots \\ -\nabla_{\mathbf{z}_m} L(\mathbf{w},\mathbf{z}) \end{pmatrix} = \begin{pmatrix} \frac{1}{C}\mathbf{F}_1 \cdots \mathbf{F}_m \\ -\mathbf{F}_1^\top \\ \vdots \quad 0 \\ -\mathbf{F}_m^\top \end{pmatrix} \begin{pmatrix} \mathbf{w} \\ \mathbf{z}_1 \\ \vdots \\ \mathbf{z}_m \end{pmatrix} - \begin{pmatrix} \sum \mathbf{F}_i \mathbf{y}^{(i)} \\ \mathbf{c}_1 \\ \vdots \\ \mathbf{c}_m \end{pmatrix} = \tilde{\mathbf{F}}\mathbf{u} - \mathbf{a}.$$

In this form, $\tilde{T}(\mathbf{u})$ is affine and $\mathcal{U}$ is polyhedral, so local linear convergence is guaranteed. We experimented with both forms of reguralization (with γ and C) and found their convergence behavior very similar.

B Min-cut polytope projections

Consider projection for a singe example i:

$$\begin{aligned} \min_z \quad & \sum_{j\in\mathcal{V}} \frac{1}{2}(z'_j - z_j)^2 + \sum_{jk\in\mathcal{E}} \frac{1}{2}(z'_{jk} - z_{jk})^2 \\ \text{s.t.} \quad & 0 \le z_j \le 1, \quad \forall j; \quad z_j - z_k \le z_{jk}, \quad z_k - z_j \le z_{jk}, \quad \forall jk. \end{aligned} \tag{17}$$

Let $h_j^+(z_j) = \frac{1}{2}(z'_j - z_j)^2$ if $0 \le z_j$, else ∞. We introduce non-negative Lagrangian variables $\lambda_{jk}, \lambda_{kj}$ for the two constraints for each edge jk and λ_{j0} for the constraint $z_j \le 1$ each node j.

The Lagrangian is given by:

$$\begin{aligned} L(\mathbf{z},\lambda) \quad = \quad & \sum_j h_j^+(z_j) + \sum_{jk} \frac{1}{2}(z'_{jk} - z_{jk})^2 - \sum_j (1 - z_j)\lambda_{j0} \\ & - \sum_{jk} (z_{jk} - z_j + z_k)\lambda_{jk} \; - \sum_{jk} (z_{jk} - z_k + z_j)\lambda_{kj} \end{aligned}$$

Letting $\lambda_{0j} = \lambda_{j0} + \sum_{k:jk\in\mathcal{E}} (\lambda_{jk} - \lambda_{kj})$, note that

$$\sum_j z_j \lambda_{0j} = \sum_j z_j \lambda_{j0} + \sum_{jk} (z_j - z_k)\lambda_{jk} + \sum_{jk} (z_k - z_j)\lambda_{kj}.$$

So the Lagrangian becomes:

$$L(\mathbf{z},\lambda) = \sum_j h_j^+(z_j) + z_j\lambda_{0j} - \lambda_{j0} + \sum_{jk} \frac{1}{2}(z'_{jk} - z_{jk})^2 - z_{jk}(\lambda_{jk} + \lambda_{kj}).$$

Now, minimizing $L(\mathbf{z},\lambda)$ with respect to $\mathbf{z}$, we have

$$\inf_{\mathbf{z}} L(\mathbf{z},\lambda) = \sum_{jk} q_{jk}(\lambda_{jk} + \lambda_{kj}) + \sum_j q_{0j}(\lambda_{0j}) - \lambda_{j0},$$

where $q_{jk}(\lambda_{jk} + \lambda_{kj}) = \inf_{z_{jk}} \left[\frac{1}{2}(z'_{jk} - z_{jk})^2 - z_{jk}(\lambda_{jk} + \lambda_{kj})\right]$ and $q_{0j}(\lambda_{0j}) = \inf_{z_j}[h_j^+(z_j) + z_j\lambda_{0j}]$. The minimizing values of $\mathbf{z}$ are:

$$\begin{aligned} z_j^* \quad &= \quad \arg\inf_{z_j} \left[h_j^+(z_j) + z_j\lambda_{0j}\right] = \begin{cases} 0 & \lambda_{0j} \ge z'_j; \\ z'_j - \lambda_{0j} & \lambda_{0j} \le z'_j; \end{cases} \\ z_{jk}^* \quad &= \quad \arg\inf_{z_{jk}} \left[\frac{1}{2}(z'_{jk} - z_{jk})^2 - z_{jk}(\lambda_{jk} + \lambda_{kj})\right] = z'_{jk} + \lambda_{jk} + \lambda_{kj}. \end{aligned}$$

Hence, we have:

$$\begin{aligned} q_{jk}(\lambda_{jk}+\lambda_{kj}) &= -z'_{jk}(\lambda_{jk}+\lambda_{kj}) - \frac{1}{2}(\lambda_{jk}+\lambda_{kj})^2 \\ q_{0j}(\lambda_{0j}) &= \begin{cases} \frac{1}{2}{z'_j}^2 & \lambda_{0j} \geq z'_j; \\ z'_j\lambda_{0j} - \frac{1}{2}\lambda_{0j}^2 & \lambda_{0j} \leq z'_j. \end{cases} \end{aligned}$$

The dual of the projection problem is thus:

$$\begin{aligned} \max_{\lambda} \quad & \sum_j q_{0j}(\lambda_{0j}) - \lambda_{j0} + \sum_{jk} -z'_{jk}(\lambda_{jk}+\lambda_{kj}) - \frac{1}{2}(\lambda_{jk}+\lambda_{kj})^2 \qquad (18) \\ \text{s.t.} \quad & \lambda_{j0} - \lambda_{0j} + \sum_{jk}(\lambda_{jk}-\lambda_{kj}) = 0, \ \ \forall j; \\ & \lambda_{jk}, \lambda_{kj} \geq 0, \ \ \forall jk; \quad \lambda_{j0} \geq 0, \ \ \forall j \ . \end{aligned}$$

Interpreting λ_{jk} as flow from node j to node k, and λ_{kj} as flow from k to j and $\lambda_{j0}, \lambda_{0j}$ as flow from and to a special node 0, we can identify the constraints of Eq. (18) as conservation of flow constraints. The last transformation we need is to address the presence of cross-terms $\lambda_{jk}\lambda_{kj}$ in the objective. Note that in the flow conservation constraints, λ_{jk}, λ_{kj} always appear together as $\lambda_{jk} - \lambda_{kj}$. Since we are minimizing $(\lambda_{jk}+\lambda_{kj})^2$ subject to constraints on $\lambda_{jk} - \lambda_{kj}$, at least one of λ_{jk}, λ_{kj} will be zero at the optimum and the cross-terms can be ignored. Note that all λ variables are non-negative except for λ_{0j}'s. Many standard flow packages support this problem form, but we can also transform the problem to have all non-negative flows by introducing extra variables. The final form has a convex cost for each edge:

$$\begin{aligned} \min_{\lambda} \quad & \sum_j -q_{0j}(\lambda_{0j}) + \lambda_{j0} + \sum_{jk} z'_{jk}\lambda_{jk} + \frac{1}{2}\lambda_{jk}^2 + \sum_{jk} z'_{jk}\lambda_{kj} + \frac{1}{2}\lambda_{kj}^2 \qquad (19) \\ \text{s.t.} \quad & \lambda_{j0} - \lambda_{0j} + \sum_{jk} \lambda_{jk} - \lambda_{kj} = 0, \ \ \forall j; \\ & \lambda_{jk}, \lambda_{kj} \geq 0, \ \ \forall jk; \quad \lambda_{j0} \geq 0, \ \ \forall j \ . \end{aligned}$$