Learning the Right Model: Efficient Max-Margin Learning in Laplacian CRFs

Dhruv Batra  
TTI-Chicago  
dbatra@ttic.edu

Ashutosh Saxena  
Cornell University  
asaxena@cs.cornell.edu

Abstract

An important modeling decision made while designing Conditional Random Fields (CRFs) is the choice of the potential functions over the cliques of variables. Laplacian potentials are useful because they are robust potentials and match image statistics better than Gaussians. Moreover, energies with Laplacian terms remain convex, which simplifies inference. This makes Laplacian potentials an ideal modeling choice for some applications.

In this paper, we study max-margin parameter learning in CRFs with Laplacian potentials (LCRFs). We first show that structured hinge-loss [35] is non-convex for LCRFs and thus techniques used by previous works are not applicable. We then present the first approximate max-margin algorithm for LCRFs. Finally, we make our learning algorithm scalable in the number of training images by using dual-decomposition techniques. Our experiments on single-image depth estimation show that even with simple features, our approach achieves comparable to state-of-art results.

1. Introduction

Undirected graphical models such as Markov Random Fields (MRFs) and Conditional Random Fields (CRFs) have been successfully applied to a number of vision problems, such as image denoising, optical flow and single-image depth estimation. While designing an MRF/CRF for an application, especially one with continuous random variables, an important modeling decision is the choice of the family of potential functions over the cliques of variables.

In the context of natural images, this question has been studied as the search for suitable natural image priors [36, 38]. Some of the earliest works [12] used quadratic disagreement pairwise potentials, corresponding to Gaussian priors on images. Since then however, a large body of work [21, 34, 36, 38] has found that histograms of filter responses for natural images tend to be highly “non-Gaussian”, in that they have sharp peaks at zero and heavy tails. Consequently, recent works have focused on non-convex priors [2, 22, 23, 32, 36].

A similar situation holds for range images, i.e. images captured by laser range-scanners as opposed to traditional cameras. Huang et al. [13] presented the first analysis of range images and found that log-gradient-histograms of range images of natural scenes were also heavy-tailed and peaked at zero. More recently, Saxena et al. [24, 25] made similar observations in the context of monocular depth estimation, and found that relative depths are better modeled by a Laplacian distribution than a Gaussian.

Model. The model we consider is a CRF with Laplacian potentials, which we refer to as Laplacian CRF (LCRF) for ease of notation. Although non-convex models like Fields of Experts (FOE) [22, 36] or hyper-Laplacian priors [14] may be a better fit to natural statistics than LCRFs, there are a number of good reasons for using LCRFs.

Laplacian potentials represent a sweet spot in the trade-off between the conflicting goals of modeling and optimization. Gaussian potentials lead to easy (inference and learning) optimization problems, but are a poor match to image statistics. Non-convex models (e.g., FOE) match image statistics well but result in difficult (non-convex) optimization problems. Laplacian potentials are robust potentials and match image statistics better than Gaussians, yet energies with Laplacian terms remain convex, which simplifies inference. Moreover, in recent work, Schmidt et al. [29] found that Laplacian models actually outperformed hyper-Laplacian models on the task of image restoration, when
used with MAP inference.

Goal. In this paper, we study discriminative parameter learning in LCRFs. This is a challenging problem because LCRFs involve $\ell_1$-norm terms and thus the energy function (negative log probability) is a non-linear function of the parameters. Thus, well understood techniques like Structured SVMs (SSVMs) [35] and Max-Margin Markov Nets (M^3Ns) [5] are not directly applicable.

Contributions. We first show that the key object in max-margin learning, i.e. the structured hinge-loss [35] is non-convex for LCRFs. Thus, an exact max-margin learning algorithm is unlikely to exist. We then present an approximate max-margin algorithm for LCRFs by linearizing the non-convex $\ell_1$-norm constraints. This broadens the class of energy functions that may be learnt via SSVMs, albeit approximately. To the best of our knowledge, this is the first max-margin discriminative training algorithm for CRFs with Laplacian potentials.

In addition, we use ideas from the dual-decomposition [3, 7] literature to decompose the problem of learning parameters from a dataset of images into smaller learning problems over individual training images. We present an efficient dual-decomposition-based algorithm that scales linearly with the number of training images and is very efficient in practice. This makes our approach highly parallelizable and scalable to a large number of training images.

We apply LCRFs to the problem of single-image depth estimation, which is a difficult mathematically-ill-posed problem due to the ambiguities introduced by the projection of the 3D world onto a 2D image. Interestingly, for this problem, Saxena et al. [24] originally proposed an LCRF to model depth as a function of the image features. However, in the absence of a parameter learning algorithm, they resorted to a heuristic approach that neglected the partition function. In this work, we show that by using a principled approximate learning algorithm, we obtain improvements in depth estimates, not only over their heuristic approach but also other techniques. Specifically, we achieve state-of-art performance on one common error metric and are competitive with the state of the art on another metric.

2. Related Work

Most relevant to our work are algorithms for parameter learning in continuous random field models, max-margin methods and techniques for single-image depth estimation.

Parameter Learning in Continuous Random Fields. In the FOE model, Roth and Black [22] used contrastive divergence [10] to approximate the maximum likelihood estimation of the parameters. Weiss and Freeman [36] proposed a basis rotation algorithm for approximating the same. Note that these are generative training methods while we are interested in a discriminative training algorithm.

Tappen et al. [33] presented a Gaussian CRF model, and showed that discriminative learning in GCGRFs boils down to linear algebra operations, and is thus tractable and efficient. Scharstein and Pal [28] used MAP estimates to approximate the gradient for max-conditional-likelihood learning of parameters. In [32], Tappen trained FOE parameters by minimizing a loss function with stochastic gradient descent. Samuel and Tappen [23] presented an improved version based on implicit-differentiation. Li and Huttenlocher [18] presented a discriminative learning algorithm based on simultaneous perturbation stochastic approximation. Barbu [2] used marginal space learning to learn the parameters. Note that for LCRFs, any reasonable loss function will be non-differentiable due to $\ell_1$-norm terms, and thus gradient-based methods are not directly applicable. Instead of exploring smooth approximations (which are often slow to converge), we formulate our problem as a Structured SVM (solved via a cutting-plane algorithm).

Structured Max-Margin Learning. Taskar et al. [5] proposed a max-margin method for training Markov networks and Tschantaridis et al. [35] proposed a Structured Support Vector Machine (SSVM) framework for learning structured-output models. Both techniques have been widely used since their introduction. Li and Huttenlocher [19] proposed an SSVM-based algorithm in the context of stereo. Szummer et al. [31] used graph-cuts within an SSVM learning algorithm in the context of segmentation. We note that in both cases, the energies of the models were linear in the parameters. This is not true for LCRFs, and thus max-margin methods are not directly applicable. Overcoming this restriction is the main focus of this paper.

Single Image Depth Estimation. Saxena et al. [24, 27] considered the problem of depth estimation from a single image using a CRF. They found that CRFs with Laplacian potentials significantly outperform those with Gaussian potentials, even with their heuristic learning approach, in which they ignored the partition function. Sudderth et al. [30] used hierarchical Dirichlet Processes in order to model the depth of objects. Liu et al. [20] proposed a semantic-category based depth-estimation model that is the current state-of-art (in terms of one error metric) on the dataset of Saxena et al. [24, 27]. More recently, Li et al. [16] proposed a feedback-enabled cascaded classification model that achieved state-of-art performance (in terms of another error metric) on this dataset.

Laplacian terms. We note that Laplacian terms have been explored in several contexts, including Lasso shrinkage [8] and sparse coding [6]. These methods focus on inference techniques, i.e. how to efficiently minimize objective functions with $\ell_1$-norm terms or constraints. This paper, on the other hand, is concerned with the problem of parameter learning in CRFs that contain these potentials.
Finally, we should also point out that although the work of Zhu et al. [37] uses a similar name as us, they use Laplacian priors for a sparse structural bias, while we use Laplacian potentials on the variables of the model.

3. Laplacian CRF

We now describe our model in detail before presenting our proposed learning algorithm in Section 4.

**Notation.** Let \([n]\) be shorthand for the set \([1, 2, \ldots, n]\). Consider a collection of continuous random variables \(Y = \{y_i : i \in [n], y_i \in \mathbb{R}\}\), and a graph \(G = (V, E)\) defined over these variables, i.e. \(V = [n], E \subseteq \binom{[n]}{2}\). For a vector \(y \in \mathbb{R}^n\), we use \(P(y)\) as a shorthand for \(P(Y = y)\). Our goal is to jointly predict \(Y\) from a collection of local features \(\{x_i \in \mathbb{R}^k | i \in [n]\}\) extracted at these labeling sites. Let \(X = [x_1^T; \ldots; x_n^T]\) be the matrix holding these features as rows. Finally, let \(Q = [f_1^T; \ldots; f_m^T]\) be a matrix of linear filters operating on \(Y\).

We define a Laplacian CRF as:

\[
P(y \mid X, \theta) = \frac{1}{Z} \exp (-E(y \mid X, \theta)), \tag{1}
\]

\[
E(y \mid X, \theta) = ||y - X\theta||_1 + ||Qy||_1, \tag{2}
\]

and where \(||a||_1 = \sum_{i=1}^{n} |a_i|\) for \(a \in \mathbb{R}^n\), \(Z\) is the partition function, and \(\theta \in \mathbb{R}^k\) is the vector of model parameters (to be learnt). We can see that this model penalizes for deviations from linear predictions and for having large responses to the filters \(Q\). Comparing our model with popular models like [32], we note that they penalize filter responses via a non-convex Lorentzian penalty function, while we use a convex \(\ell_1\)-norm penalty.

Although the algorithms we develop are valid for arbitrary filters \(Q\), in this paper we only focus on gradient filters, i.e. the case when \(Q\) is the (weighted) incidence matrix of \(G\), such that rows of \(Qy\) give the (weighted) differences of neighboring labels. Thus:

\[
E(y \mid X, \theta) = ||y - X\theta||_1 + \sum_{(i,j) \in E} |w_{ij}| (y_i - y_j), \tag{3}
\]

where \(w_{ij}\) are edge-weights. These edge-weights may themselves be functions of edge-features, i.e. \(w_{ij} = x_{ij}^T \beta\), where \(x_{ij}\) is a feature extracted at edge \((i,j)\), and \(\beta\) is the (shared) edge parameter vector. For ease of explanation and to match our current implementation, in this paper we only describe learning techniques for \(\theta\) and assume \(w_{ij}\) to be known constants. However, the algorithm for learning \(\beta\) is a straightforward generalization, and is presented in the supplementary material [1] (Section 1). Finally, note that we place no restrictions on the size or type of the graph-neighborhood and there could be arbitrary “long-range” links between the variables.

4. Learning and Inference in LCRFs

Before we go into details about parameter learning we need to describe inference in LCRFs.

4.1. Inference

We focus on maximum a posteriori inference in this model, which can be written as:

\[
g(\theta) = \arg \max_{y \in \mathbb{R}^n} P(y \mid X, \theta) \tag{4a}
\]

\[
= \arg \min_{y \in \mathbb{R}^n} \{||y - X\theta||_1 + ||Qy||_1\} \tag{4b}
\]

This is an \(\ell_1\)-norm minimization problem. It is well-known [4] that such problems may be formulated as a linear program. Let \(q \equiv X\theta, A \equiv [I_{n \times n} : Q]\) (where \(I_{n \times n}\) is the \(n \times n\) identity matrix) and \(b \equiv [q : 0_{m \times 1}]\). Now:

\[
g(\theta) = \arg \min_{y \in \mathbb{R}^n} \{||y - q||_1 + ||Qy - b||_1\} \tag{5a}
\]

\[
= \arg \min_{y \in \mathbb{R}^n} ||Ay - b||_1, \tag{5b}
\]

\[
= \arg \min_{\nu \in \mathbb{R}^n, \nu \in \mathbb{R}^{(n+m)}} \nu \cdot 1 \tag{5c}
\]

\[
s.t. \quad \nu \geq Ay - b, \tag{5d}
\]

\[
\nu \geq -(Ay - b) \tag{5e}
\]

where \(\nu\) is an auxiliary variable. Notice that (5c) is now a Linear Program (LP). The trick above is to notice that absolute value minimization can be replaced by two linear lower bounds. As we will see next, this trick helps us more than once.

4.2. Parameter Learning

Parameter learning involves finding the optimal values of parameter \(\theta\) from labeled training data. Let us first consider a single training sample \((X, y^*)\), where \(y^*\) is the ground-truth labeling. We start with the margin-rescaled Structured SVM formulation of Tsochantaridis et al. [35], which minimizes the following problem:

\[
\min_{\theta, \xi} \frac{1}{2} ||\theta||_2^2 + C \xi \tag{6a}
\]

\[
s.t. \quad E(y^*|X, \theta) - E(y^*|X, \theta) \geq \Delta(y^*, y^*) - \xi \quad \forall y^* \tag{6b}
\]

\[
\xi \geq 0, \tag{6c}
\]

where \(\Delta(y^*, y^*)\) is a user-specified risk function measuring the separation between labelings, and \(C\) is a positive multiplier. Intuitively, we can see that SSVM minimizes a quadratic objective subject to constraints that enforce a soft margin between the energy of ground-truth and any other labeling, such that the margin is scaled by the risk function. It is known [35] that all of the above constraints (6b),(6c) may be expressed compactly via the *structured-hinge-loss:*

\[
HLoss(\theta) = \max \left\{ 0, E(y^*|X, \theta) - \min_{y^*} \left( E(y^*|X, \theta) - \Delta(y^*, y^*) \right) \right\}. \tag{7}
\]

It can be shown [35] that hinge-loss is a convex upper-bound on risk incurred by the MAP solution \(\hat{y}\), i.e. \(HLoss(\theta) \geq \Delta(\hat{y}(\theta), y^*)\). Thus, SSVMs can be understood as minimizing a regularized structured hinge loss:

\[
\min_{\theta} \frac{1}{2} ||\theta||_2^2 + C HLoss(\theta) \tag{8}
\]

Note that for a continuous-valued CRF, the set of all possible other labelings is an infinitely large set, and thus the above program cannot even be written down. Following the
work of Tschantaridis et al. [35], we address this problem
by using a cutting-plane approach. Specifically, we ini-
tialize this program with a small set of “bad” labelings \(\hat{\mathcal{I}}\), then
learn \(\hat{\theta}\), and if the optimal labeling corresponding to this
learnt \(\hat{x}\), is not already in the set \(\hat{\mathcal{I}}\)
(within some tolerance factor), we add it to the set and
repeat. Formally, we repeatedly solve:

\[
(\mathcal{M}: \hat{\mathcal{I}}) \quad \min_{\theta, \xi} \frac{1}{2} ||\theta||_2^2 + C \xi
\]

s.t.
\[
||y^i - X\theta||_1 + ||Qy^i||_1 - ||y^*||_1 \geq 1 - \xi, \quad \forall i \in \hat{I} \quad (9b)
\]

\[
\xi \geq 0. \quad (9c)
\]

Nonconvexity of Hinge-Loss. Recall that in a typical
SSVM or M³N, the energy function is linear in \(\theta\).
In our case, the energy function contains \(\ell_1\)-norm terms, and
thus constraints (9b) are not linear in \(\theta\). Unfortunately, this
makes the corresponding hinge-loss non-convex. Formally,
we can state the following:

**Theorem 1** Hinge Loss for the LCRF model, i.e.
\(HLoss(\theta) = \max \left\{ 0, ||y^* - X\theta||_1 + ||Qy^*||_1 -
\min_{y'} \left( ||y' - X\theta||_1 + ||Qy'||_1 - \Delta(y', y^*) \right) \right\},
\)
is non-convex in \(\theta\).

**Proof.** See Supplementary Material [1], Section 2.

Due to this non-convexity, standard techniques like sub-
gradient descent and cutting-plane methods cannot be used to
minimize the LCRF hinge-loss. Furthermore, note that the
variables \(\theta\) multiply with \(X\) and therefore every abso-
olute value term contains all the components of \(\theta\). This does
not allows use of search algorithms such as in [15].

Approximate Max-Margin Learning. We now show how
the non-convex program \((\mathcal{M}: \hat{\mathcal{I}})\) can be approximated
by a convex QP. The following exposition is described with
a 0-1 risk, however any risk-function (e.g. hamming) may
be used as long as the risk-augmented energy minimization
problem [35] is tractable. We use the same trick as we did in
(5c) to convert an \(\ell_1\)-norm minimization into an LP, using
auxiliary variables: \(\nu^*, \{d^i\} \in \mathbb{R}^n\):

\[
(\mathcal{M}\mathcal{Q}\mathcal{P}: \hat{\mathcal{I}}) \quad \min_{\theta, \xi, \{d^i\}, \{\nu^i\}} \frac{1}{2} ||\theta||_2^2 + C \xi + C_1 \sum_{j=1}^n d_j^i + C_2 \sum_{j=1}^n \sum_{i=1}^n d_j^i \quad (10a)
\]

s.t.
\[
\sum_{j=1}^n d_j^i - \sum_{j=1}^n d_j^i \geq 1 + ||Qy^*||_1 - ||Qy'||_1 - \xi \quad \forall i \in \hat{I} \quad (10b)
\]

\[
\xi \geq 0 \quad \forall i \in \hat{I} \quad (10c)
\]

\[
d^* \geq (y^* - X\theta), \quad d^* \geq -(y^* - X\theta) \quad (10d)
\]

where \(C_1, C_2\) are positive weights (see [1] for how to set
them). All constraints in the above program \((\mathcal{M}\mathcal{Q}\mathcal{P}: \hat{\mathcal{I}})\)
are linear in \(\theta, \xi, \{d^i\}, \{\nu^i\}\), and this program is a convex
quadratic program, solvable by standard techniques.
Formally, we can state the following about this approximation:

**Theorem 2** If \(\{\hat{\theta}, \hat{\xi}, \hat{d}^*, \hat{\nu}^*\}\) is the optimum solution of
\(\mathcal{M}\mathcal{Q}\mathcal{P}: \hat{\mathcal{I}}\), then \(\hat{\xi}\) is equal to the LCRF hinge-loss
\(HLoss(\hat{\theta})\), and thus an upper-bound on the loss incurred
by the MAP solution, i.e. \(\hat{\xi} = HLoss(\hat{\theta}) \geq \Delta(\hat{\nu}(\hat{\theta}), y^*)\).

**Proof.** See Supplementary Material [1], Section 3.

Thus, the constraints of the two programs – \(\mathcal{M}\mathcal{M}: \hat{\mathcal{I}}\) and
\(\mathcal{M}\mathcal{Q}\mathcal{P}: \hat{\mathcal{I}}\) – represent exactly the same object, i.e.
structured hinge-loss. The approximation comes from the extra
terms in the objective function (10a), which are necessary for
linearizing the \(\ell_1\)-norm terms.

From a computational perspective, it is important to point
out one drawback of this linearization approach. Program
\(\mathcal{M}\mathcal{Q}\mathcal{P}: \hat{\mathcal{I}}\) includes vector constraints (10d), (10e) of
dimension equal to the number of random variables \(n\).
While constraint (10d) does not grow with iterations of the
cutting-plane algorithm, constraint (10e) does. Thus,
each additional “bad” labeling added to the list \(\hat{\mathcal{I}}\) adds \(n\)
more constraints to the QP, which may become impractical.
However, as we see next, we use ideas from the dual-
decomposition [3, 7] literature to restrict this QP to a
manageable size.

4.3. Extension to Multiple Training Images
via Dual-Decomposition

Let us now extend our algorithm to learn from multi-
ple images. Consider a training dataset indexed by \(\mathcal{T} = \{1, 2, \ldots, T\}\). Let \(\xi^{(t)}\) denote the slack variable and vector
\(D^{(t)} = \{d^{(t)}_i, d^{(t)}_i \mid i \in \hat{I}\}\) hold all auxiliary vari-
ables for a training image \(t\). For brevity of description, let
us denote all linear constraints in (10) with the polytope
\(\mathcal{P}^{(t)}\). Thus, \(\{\theta^{(t)}, \xi^{(t)}, D^{(t)}\} \in \mathcal{P}^{(t)}\) denotes the set of sol-
utions feasible according to image \(t\). We can now write
down a straightforward generalization of the previous pro-
gram \(\mathcal{M}\mathcal{Q}\mathcal{P}: \hat{\mathcal{I}}\) to multiple training images:

\[
(\mathcal{M}\mathcal{Q}\mathcal{P}: \hat{\mathcal{T}}) \quad \min_{\theta, \{\xi^{(t)}\}, \{D^{(t)}\}} \frac{1}{2} ||\theta||_2^2 + C_T \sum_{t \in \mathcal{T}} \xi^{(t)} + C_T' \sum_{t \in \mathcal{T}} D^{(t)} \cdot 1 \quad (11a)
\]

s.t.
\[
\{\theta^{(t)}, \xi^{(t)}, D^{(t)}\} \in \mathcal{P}^{(t)} \quad \forall t \in \mathcal{T}. \quad (11b)
\]

Clearly, as the size of the training dataset increases, this
program becomes larger, and very quickly impractical. The
key here is to notice that this large program \((\mathcal{M}\mathcal{Q}\mathcal{P}: \hat{\mathcal{T}})\)
consists of several almost independent problems over indi-
vidual images, only coupled by the parameter \(\theta\). We follow
a dual-decomposition approach, where we solve a La-
gragian relaxation of this problem which easily decom-
poses to smaller independent sub-problems for each train-
ing image. This enables solving the problem over a dist-
ibuted architecture or a cloud of machines, and thus scales
well to large datasets. Most importantly, we prove this
Lagrangian relaxation achieves zero duality gap and in fact
is a tight relaxation (see Theorem 3). Thus, the solution
to the Lagrangian relaxation converges to the solution of
\(\mathcal{M}\mathcal{Q}\mathcal{P}: \hat{\mathcal{T}}\).
We describe this relaxation next. First, we reparameterize the previous program by allocating to each training image its own copy of the parameters $\theta^{(t)}$; 

\[
(MMQP : \tilde{T}^2) \quad \min_{\delta, (\theta^{(t)} , \xi^{(t)} , D^{(t)})} \delta \left( \frac{1}{2T} \sum_{t \in T} \|\theta^{(t)}\|_2^2 + \frac{C}{T} \sum_{t \in T} \xi^{(t)} \right) + \frac{C'}{T} \sum_{t \in T} D^{(t)} \cdot 1 \quad \text{s.t.} \quad \{\theta^{(t)} , \xi^{(t)} , D^{(t)}\} \in \mathcal{P}(t) \quad (12a)
\]

The above program $(MMQP : \tilde{T}^2)$ uses a global variable $\tilde{\theta}$ to force all training images to have the same parameters, and is equivalent to the earlier program $(MMQP : \tilde{T})$. However, we can now relax constraints (12c). We consider a Lagrangian relaxation of the above program by dualizing [7] the complicating constraints (12c);

\[
(LR : \tilde{T}^2) \quad \min_{\delta, (\theta^{(t)} , \xi^{(t)} , D^{(t)})} \delta \left( \frac{1}{2T} \sum_{t \in T} \|\theta^{(t)}\|_2^2 + \frac{C}{T} \sum_{t \in T} \xi^{(t)} \right) + \frac{C'}{T} \sum_{t \in T} D^{(t)} \cdot 1 + \sum_{t \in T} \lambda^{(t)} \cdot (\theta^{(t)} - \tilde{\theta}) \quad (13a)
\]

\[
\text{s.t.} \quad \{\theta^{(t)} , \xi^{(t)} , D^{(t)}\} \in \mathcal{P}(t) \quad \forall t \in T, \quad (13b)
\]

where $\lambda^{(t)} \in \mathbb{R}^k$ are the (unconstrained) Lagrangian multipliers, which may be thought of as indicating the penalties or costs for violating their corresponding constraints. Note that now the above problem is completely separable into independent sub-problems for each training image.

\[
(LR : \tilde{T}) \quad \min_{\theta^{(t)} , \xi^{(t)} , D^{(t)}} \left( \frac{1}{2T} \sum_{t \in T} \|\theta^{(t)}\|_2^2 + \sum_{t \in T} \lambda^{(t)} \cdot (\theta^{(t)} - \tilde{\theta}) \right) + \frac{C'}{T} \sum_{t \in T} D^{(t)} \cdot 1 - \min_{\delta} \left( \sum_{t \in T} \lambda^{(t)} \right) \cdot \tilde{\delta} \quad (14a)
\]

\[
\text{s.t.} \quad \{\theta^{(t)} , \xi^{(t)} , D^{(t)}\} \in \mathcal{P}(t) \quad \forall t \in T, \quad (14b)
\]

We can see that the (unconstrained) optimization over $\tilde{\theta}$ forces a constraint on the Lagrangian variables, i.e. $\sum_{t \in T} \lambda^{(t)} = 0$; otherwise the program will not have a finite value. Let us now define these independent sub-problems as a function of the dual variables $\lambda^{(t)}$:

\[
\mathcal{F}(t)(\lambda^{(t)}) = \min_{\theta^{(t)} , \xi^{(t)} , D^{(t)}} \left( \frac{1}{2T} \sum_{t \in T} \|\theta^{(t)}\|_2^2 + \sum_{t \in T} \lambda^{(t)} \cdot (\theta^{(t)} - \tilde{\theta}) \right) + \frac{C'}{T} \sum_{t \in T} D^{(t)} \cdot 1 \quad (15a)
\]

\[
\text{s.t.} \quad \{\theta^{(t)} , \xi^{(t)} , D^{(t)}\} \in \mathcal{P}(t). \quad (15b)
\]

We can now search for the tightest relaxation by optimizing over the dual variables $\{\lambda^{(t)}\}$. Formally, this is the Lagrangian dual of $(MMQP : \tilde{T})$:

\[
(LD : \tilde{T}) \quad \max_{\lambda^{(t)}} \left( \sum_{t \in T} \mathcal{F}(t)(\lambda^{(t)}) \right) \quad (16a)
\]

\[
\text{s.t.} \quad \sum_{t \in T} \lambda^{(t)} = 0. \quad (16b)
\]

**Algorithm 1.** We solve this dual problem via projected gradient ascent. It is easy to verify that the gradient of each sub-problem with respect to the Lagrangian multiplier is simply the optimal parameter learned from that sub-problem, i.e. $\frac{\partial \mathcal{F}(t)}{\partial \lambda^{(t)}} = \hat{\theta}^{(t)}$, where $\hat{\theta}^{(t)}$ is the optimal solution to problem (15). We also note that each subproblem is strongly convex in $\theta^{(t)}$ and thus has a unique optimum $\hat{\theta}^{(t)}$. The projection step is fairly simple – it involves satisfying the zero-mean constraint of (16b), which can be enforced by subtracting the mean of the dual variables. Overall, the update rule is given by $\lambda^{(t)} \leftarrow \left[ \lambda^{(t)} + \alpha \frac{\partial \mathcal{F}(t)}{\partial \lambda^{(t)}} \right]_0$, where $\alpha$ is the step-size and $[\cdot]_0$ is the projection operator, i.e. $[\lambda^{(t)}]_0 = \lambda^{(t)} - \frac{1}{T} \sum_{t \in T} \lambda^{(t)}$. We can see that each step of gradient ascent requires learning parameters on all subproblems. Thus, we have converted a large QP into several smaller QPs that can be independently optimized in parallel, although they need to be solved several times.

Most importantly, the following theorem shows that the Lagrangian relaxation does not introduce a second level of approximation. In fact, our algorithm exactly solves $(MMQP : \tilde{T})$.

**Theorem 3** $LD : \tilde{T}$ (16) has zero duality gap and Algorithm 1 converges to the optimum of $(MMQP : \tilde{T})$ (11).

**Proof.** See Supplementary Material [1], Section 4.

5. Experiments

We apply our learning algorithm to the problem of estimating depth from a single image (see Fig. 2 for examples). This is a difficult mathematically-ill-posed problem due to the ambiguities introduced by the projection of the 3D world onto a 2D image. Local features alone are typically not enough for estimating depth (e.g., both sky and water can be blue, a gray patch could be sidewalk or a wall).

We need to use a CRF to model the relations between the depths at neighboring regions. Hand-tuning weights on features is difficult and impractical and thus a good learning algorithm is essential for this application.

For an $h \times w$ image, our model variables $\{y_i : i \in [n], n = hw\}$ indicate log-depth at each pixel, given image features $x_i \in \mathbb{R}^k$ at the corresponding pixels in the image. We follow the same parameter sharing scheme as Saxena et al. [27], i.e. assign each row of an image its own parameter vector. We use two kinds of features: 105-dim texture features computed using the publicly-available code from Make3D [24], and 8-dim semantic-category-prediction features at each pixel provided by Liu et al. [20]. When we compare to other works using the same features, the performance difference can be attributed to the choice of model and learning algorithm used. Moreover, Saxena et al. [24] used the same (texture) features and model as us (LCRF), and trained the parameters simply by minimizing the $\ell_1$ error $||y - X\hat{\theta}||_1$. Any performance difference between their
Figure 2: Results on single-image depth estimation using LCRFs trained with our learning algorithm. (Left) Laser ground-truth depths. (Middle) Original image. (Right) Predicted depths. We can see that LCRFs model depth discontinuities well. (Best viewed in color.)

work and ours is specifically attributed to the use of our max-margin learning algorithm.

We test our approach on the Make3D Range Image dataset [25, 27], which consists of 534 images (400 training, 134 testing) with ground-truth depths obtained from a laser scanner. The variety of environments (roads, buildings, trees, indoor corridors, etc.) presents situations such as sharp depth changes (due to occlusions) and thin long structures (trees, poles, etc.), making this a challenging dataset. All results are reported on the 134 test images.
We measure our performance with a commonly used error metric on this dataset called rel-error, defined as $\frac{1}{n} \sum_{j=1}^{n} (\hat{y}_j - y_j^*)/y_j^*$, where $y_j^*$ is the ground-truth depth for pixel $j$, and $\hat{y}_j$ is the predicted depth. Another error metric that is also sometimes used is log10 error metric $\frac{1}{n} \sum_{j=1}^{n} \log_{10} (\hat{y}_j - \log_{10} y_j^*)$. For the sake of completeness, we also report RMSE errors $\sqrt{\frac{1}{n} \sum_{j=1}^{n} (\hat{y}_j - y_j^*)^2}$, which a number of previous works do not report.

We experimented with decomposition into sub-problems of various sizes, and found decomposition into 85-image sub-problems to be an ideal choice. Note that the zero-duality gap statements in Theorem 3 hold for batch decomposition as well. We typically ran Algorithm 1 for 4-8 steps, and typical cardinality of set $I$ was 5-10.

**Comparison to State-of-the-Art.** We compare our results to a number of other works that have reported results on this dataset. The comparison is shown in Table 1. The results for “Chance” baseline are taken from Saxena et al. [27].

This table also lists the major improvement source over the original work of Saxena et al. [24] (SCN). We note that a direct comparison with these methods is problematic. Some methods use more sophisticated models and sometimes additional data/features, limiting the conclusions that may be drawn from the comparison. For example SCN [24] use a hierarchical MRF; Liu et al. [20] (LGK) use semantic labels and additional geometry based priors in the CRF. Heitz et al. [9] and Li et al.’s [16] cascaded classification models combine information from object detection, image segmentation and scene categorization. Overall, these works focused on using context in order to improve performance. On the other hand, our method only uses a single 4-connected grid and no additional information. Despite the simplicity, our model achieves 0.362 rel-error, which is better than current best of 0.370. We attribute this superior performance to our learning algorithm, which enables training an accurate model (LCRF) for the problem. On RMSE and log10 errors, our approach is competitive with the state of the art, but not the lowest. Hopefully, performance can be further improved by combining these orthogonal ideas – our LCRF model and learning algorithm, semantic modeling of LGK, and multi-task information of Heitz et al.

**Stability w.r.t. Small Training Set.** We also analyzed the performance of our learning algorithm with the size of training set. Fig. 3 shows that our algorithm is less prone to overfitting as compared to the pseudo-inverse training of SCN, i.e. even with small amounts of training data, it can perform reasonably well on the test set.

**Effect of Learning Algorithm.** Finally, we try to isolate the effect of the learning algorithm from other factors. To do this, we only compare baselines that all use exactly the same features and approximate the same MRF/CRF structure (although different kinds of potentials). These include SCN [24], the “pointwise CRF” from Make3D [27] and the “Pixel CRF model” from LGK [20]. For this comparison, we trained our model with texture features only.

**Qualitative Results.** In Fig. 2, we show the predicted depths for a few examples in the test set, together with the laser ground-truth depths for comparison. Our algorithm makes quite reasonable-looking predictions. In general, CRFs suffer from the problem of over-smoothing (e.g. observed by Saxena et al. [24, 27]). However, this problem seems to be less acute in our method—we believe this is because our learning method learns the parameters while taking into account the edge terms in the CRF, and thus results in sharper (see Fig. 2-E,H,I) and more accurate depths. The problem still persists in some cases, such as in Fig. 2-J, where our algorithm was confused by the texture of the leaves, and produced over-smoothed depths.

Note that the ground-truth labels were limited to a range of 80 meters, and therefore, in most of the images in Fig. 2, we see that far-away structures are measured as 80m (same as sky) by the laser. Our model reasonably predicts even far-away parts in the image (and the actual ground-truth label is wrong!). See Fig. 2-A,D,F,G,H,I. In Fig. 2-B, we see the reflections of another building, trees and sky into a glass-paned transparent wall. The laser scanner measures depths incorrectly because the pulses get scattered by the glass. Our algorithm relies on the image and estimates the depth of the reflected structures instead.
6. Conclusions

In this paper, we considered continuous-valued CRFs with heavy-tailed Laplacian potentials. Although LCRFs are the ideal modeling choice for many applications and inference in these models is convex and tractable, parameter learning could only be performed heuristically in prior work. We presented the first (approximate) max-margin parameter learning algorithm for LCRFs, by linearizing the non-convex $\ell_1$-norm constraints. We also presented a dual-decomposition-based algorithm to make learning scalable in the number of training images.

Future work involves exploring more sophisticated decomposition techniques like Augmented Lagrangian methods. In addition, $\ell_1$-norm minimization problems are often convex relaxations for solving $\ell_0$-norm minimization i.e. cardinality constraints. We believe that the learning algorithm for LCRFs presented in this paper could be useful for MRFs/CRFs with such potentials as well.

Acknowledgements. We thank Stephen Gould for providing data from [20] & helpful comments; and the anonymous reviewers for their insightful feedback on Theorem 2.

References

[1] Authors. Please see supplementary materials. 3, 4, 5
[19] Y. Li and D. P. Huttenlocher. Learning for stereo vision using the structured support vector machine. CVPR, 2008. 2
[22] S. Roth and M. Black. Fields of experts. IJCV, 82(2), April 2009. 1, 2
[26] A. Saxena, M. Sun, and A. Y. Ng. Learning 3-d scene structure from a single still image. IJCV, 2007. 1, 6
[27] A. Saxena, M. Sun, and A. Y. Ng. Learning 3-d scene structure from a single still image. In ICCV workshop on 3D Representation for Recognition (AdRR-07), 2007. 8
1. Learning Edge-Weights

In this section, we describe how the approach in Sections 4.2 and 4.3 of the main paper can be generalized to also learn edge-weights. Recall that LCRF energy is given by:

\[ E(y | X, \theta) = ||y - X\theta||_1 + \sum_{(u,v) \in \mathcal{E}} w_{uv} (y_u - y_v), \]  

where \( w_{uv} \) are edge-weights. These edge-weights are themselves functions of edge-features, i.e. \( w_{uv} = x_{uv}^T \beta \), where \( x_{uv} \) is a feature extracted at edge \((u,v)\), and \( \beta \) is the (shared) edge parameter vector.

Similar to Section 4.2 in the main paper, let us first consider a single training sample \((X, y^*)\), where \( y^* \) is the ground-truth labeling. An SSVM formulation for learning \( \{\theta, \beta\} \) can be given by:

\[ \min_{\theta, \beta, \xi} \frac{1}{2} ||\theta||^2 + \frac{1}{2} ||\beta||^2 + C \xi \]  

\[ \text{s.t.} \quad ||y^* - X\theta||_1 + \sum_{(u,v) \in \mathcal{E}} |x_{uv}^T\beta| |y_u^* - y_v^*| \]  

\[ -||y^* - X\theta||_1 - \sum_{(u,v) \in \mathcal{E}} |x_{uv}^T\beta| |y_u^* - y_v^*| \geq 1 - \xi \]  

\[ \xi \geq 0 \quad \forall i \in \tilde{I}. \]  

Using the same linearization tricks as used in Section 4.2 of the main paper, we now show how the above program can be relaxed into a QP with auxiliary variables: \( d^*, \{d^i\} \in \mathbb{R}^n, e \in \mathbb{R}^m \). Here \( n = |\mathcal{V}| \), the number of nodes, and \( m = |\mathcal{E}| \), the number of edges.

\[ \min_{\theta, \beta, \xi, d^*, \{d^i\}, e} \frac{1}{2} ||\theta||^2 + \frac{1}{2} ||\beta||^2 + C \xi \]  

\[ + C_1 \sum_{j=1}^n d_j^* + C_2 \sum_{(i,j) \in \mathcal{I}} d_j^* + C_3 \sum_{uv} e_{uv} \]  

\[ \text{s.t.} \quad \sum_{j=1}^n d_j^* - \sum_{j=1}^n d_j^* + \sum_{(u,v)} \left(|y_u^* - y_v^*| - |y_u^* - y_v^*|\right) e_{uv} \]  

\[ \geq 1 - \xi \quad \forall i \in \tilde{I} \]  

\[ e_{uv} \geq +ax_{uv}^T \beta, \quad e_{uv} \geq -ax_{uv}^T \beta \]  

\[ d^* \geq +(y^* - X\theta), \quad d^* \geq -(y^* - X\theta) \]  

\[ \xi \geq 0. \]  

Note that during parameter learning, variables \( y^* \), \( \{y^i\} \) are known constants. Thus, similar to Section 4.2 in the main paper, all constraints in the above program are linear in \( \theta, \beta, \xi, d^* \), \( \{d^i\} \), and this program is a convex quadratic program, solvable by standard techniques. The extension to multiple images via Lagrangian decomposition is analogous to Section 4.3, where polytope \( \mathcal{P}(\tilde{I}) \) now refers to the linear constraints (3b)-(3e).

2. Proof of Theorem 1

Theorem 1 Hinge Loss for the LCRF model, i.e. \( HLoss(\theta) = \max \left\{ 0, ||y^* - X\theta||_1 + ||Qy^*||_1 - \min_{y^i} \left(||y^i - X\theta||_1 + ||Qy^i||_1 - \Delta(y^i, y^*)\right) \right\} \), is non-convex in \( \theta \).

Proof. First, let us define two functions \( f(\theta), g(\theta) \):

\[ f(\theta) \triangleq ||y^* - X\theta||_1 + ||Qy^*||_1 \]  

\[ - \min_{y^i} \left(||y^i - X\theta||_1 + ||Qy^i||_1 - \Delta(y^i, y^*)\right) \]  

\[ g(\theta) \triangleq \min_{y^i} \left(||y^i - X\theta||_1 + ||Qy^i||_1 - \Delta(y^i, y^*)\right). \]  

Thus, we can express \( f(\theta) \) and \( HLoss(\theta) \) as:
\[ f(\theta) = \|y^*-X\theta\|_1 + \|Qy^*\|_1 - g(\theta) \]
\[ HLoss(\theta) = \max \left\{ 0, f(\theta) \right\} \]

Now, we will prove that \( g(\theta) \) is non-convex in \( \theta \), thus making \( f(\theta) \) and \( HLoss(\theta) \) non-convex as well. To this end, let us rewrite \( g(\theta) \) as:
\[ g(\theta) = \min_{y^*} \left( \|X\theta - y^*\|_1 + h(y^*) \right), \]
where
\[ h(y^*) = \|Qy^*\|_1 - \Delta(y^*, y^*). \]

Now clearly \( \|X\theta - y^*\|_1 \) is convex in \( \theta \) since \( \ell_1 \)-norm \( \| \cdot \|_1 \) is a convex function and composition with a linear map \( X\theta - y^* \) preserves convexity [1]. Unfortunately, pointwise \textit{min} of a collection of convex functions is not guaranteed to be convex. Specifically, here is a simple counter-example that results in \( g(\theta) \) being non-convex in \( \theta \):
\[ g(\theta) = \min \left\{ \|\theta - 1\|, \|\theta - 2\| \right\}. \]

Clearly, the function in Fig. 1b is non-convex in \( \theta \). This completes the proof.

3. Proof of Theorem 2

Let us first recall from the main manuscript the max-margin problem we are interested in solving:
\[ (MM : \hat{\theta}) \]
\[
\begin{align*}
\min_{\theta, \xi} & \quad \frac{1}{2} \|\theta\|_2^2 + C \xi \\
n & \quad \|y^* - \xi\|_1 + \|Qy^*\|_1 \\
s & \quad - \|y^* - X\theta\|_1 - \|Qy^*\|_1 \geq 1 - \xi \quad \forall i \in \hat{I} \\
& \quad \xi \geq 0.
\end{align*}
\]

In the manuscript, we showed how the non-convex program \( (MM : \hat{\theta}) \) can be \textit{approximated} by a convex QP using auxiliary variables: \( d^* \), \( \{d^i\} \in \mathbb{R}^n \).

\[ (MMQP : \hat{\theta}) \]
\[
\begin{align*}
& \quad \min_{\theta, \xi, \{d^i\}, \{d^i\}} \frac{1}{2} \|\theta\|_2^2 + C \xi + C_1 \sum_{j=1}^{n} d^*_j + C_2 \sum_{i \in \hat{I}} \sum_{j=1}^{n} d^i_j \\
& \quad \text{s.t.} \quad \sum_{j=1}^{n} d^*_j - \sum_{j=1}^{n} d^*_j \geq 1 + \|Qy^*\|_1 - \|Qy^*\|_1 - \xi \quad \forall i \in \hat{I} \\
& \quad \xi \geq 0 \\
& \quad d^*_j \geq |y^*_j - x^*_j \hat{\theta}| \quad \forall j \in [n] \\
& \quad d^*_j \geq |y^*_j - x^*_j \hat{\theta}| \quad \forall j \in [n] \\
& \quad \hat{\xi} \geq 0.
\end{align*}
\]

\textbf{Theorem 2} If \( \{\hat{\theta}, \hat{\xi}, \hat{d}^*, \hat{d}^i\} \) is the optimum solution of \( MMQP : \hat{\theta} \), then \( \hat{\xi} \) is equal to the LCRF hinge-loss \( HLoss(\hat{\theta}) \), and thus an upper-bound on the loss incurred by the MAP solution, i.e. \( \xi = HLoss(\hat{\theta}) \geq \Delta(\hat{y}(\hat{\theta}), y^*) \).

\textbf{Proof.} Since \( \{\hat{\theta}, \hat{\xi}, \hat{d}^*, \hat{d}^i\} \) is the optimum solution of \( MMQP : \hat{\theta} \), it must be a feasible solution, i.e.:
\[ \begin{align*}
\sum_{j=1}^{n} d^*_j - \sum_{j=1}^{n} d^*_j & \geq 1 + \|Qy^*\|_1 - \|Qy^*\|_1 - \hat{\xi} \\
\hat{d}^*_j & \geq |y^*_j - x^*_j \hat{\theta}| \quad \forall j \in [n] \\
\hat{d}^*_j & \geq |y^*_j - x^*_j \hat{\theta}| \quad \forall j \in [n] \\
\hat{\xi} & \geq 0.
\end{align*} \]

\textbf{Claim 1:} If \( C_2 > 0 \), (13b) must be tight, i.e. hold with equality.

\textbf{Proof of Claim 1:} Suppose the claim is false. Thus we can reduce \( \hat{d}^*_j \), i.e. \( \hat{d}^*_j \leftarrow \hat{d}^*_j - \delta_j \) for some \( \delta_j > 0 \) without violating (13b). Note that this reduced \( \hat{d}^*_j \) would also satisfy (13a), and thus is a feasible solution that reduces the objective function by \( \sum_{j} \delta_j \). This is a contradiction to the optimality of \( \{\hat{\theta}, \hat{\xi}, \hat{d}^*, \hat{d}^i\} \) and hence the claim must hold.

The above claim implies that \( \hat{d}^*_j = |y^*_j - x^*_j \hat{\theta}| \). Thus we can rewrite (13a)-(13c) as follows:
\[ \begin{align*}
\sum_{j=1}^{n} d^*_j + \hat{\xi} & \geq 1 + \|Qy^*\|_1 - \|Qy^*\|_1 + |y^*_j - x^*_j \hat{\theta}| \\
\hat{d}^*_j & \geq |y^*_j - x^*_j \hat{\theta}| \quad \forall j \in [n] \\
\hat{d}^*_j & \geq |y^*_j - x^*_j \hat{\theta}| \quad \forall j \in [n]
\end{align*} \]

\textbf{Claim 2:} If \( C_2 > C \), then (14c) must be tight, i.e. hold with equality.

\textbf{Proof of Claim 2:} This claim is a little trickier to prove. We cannot directly apply the proof of claim 1 because decreasing \( \hat{d}^*_j \) might violate (14a). However, note that at least one of (14a),(14c) must be tight because if neither are tight
them we can directly apply proof of claim 1 to show a contradiction.

Let us assume (14a) is tight, (14c) is not tight and try to show a contradiction. Since (14c) is not tight, we can decrease \( d^j, i.e. \hat{d}_j^+ \leftarrow d_j^+ - \delta_j \) for some \( \delta_j > 0 \) without violating (14c). Since (14a) was tight, this reduced \( \hat{d}_j^+ \) is not part of a feasible solution as it now violates (14a). However, we can simply increase \( \xi_i \) to fix this, i.e. \( \xi_i \leftarrow \xi_i + \sum_j \delta_j \). Clearly, the effect of the \( \{\delta_j\} \) cancels outs in (14a) and the new solution \( \{\xi_i + \sum_j \delta_j, \; \hat{d}_j^+ - \delta_j\} \) is again a feasible solution of Program \( \text{MMQP} \; : \; \tilde{T} \) (12). However, if \( C_2 > C \), this new solution actually reduces the objective function by \( (C_2 - C) \sum_j \delta_j \). This is a contradiction since we started with the optimal solution, and thus claim 2 must hold.

Combining Claim 1 and Claim 2 together, we can rewrite (13a)-(13b) as follows:

\[
\begin{align*}
&||y^t - X\hat{\theta}||_1 + ||Qy^t||_1 \\
&-||y^t - X\bar{\theta}||_1 - ||Qy^t||_1 + 1 - \bar{\xi} \\
&d^t_j = |y^t_j - x_j^T \bar{\theta}|, \quad \forall j \in [n] \\
&\hat{d}_j^t = |y^t_j - x_j^T \hat{\theta}|, \quad \forall j \in [n] \\
\end{align*}
\]

Comparing (15a) with (11), we can see that constraint (15a) is equivalent to constraint (11b), which expresses the structured hinge-loss. Thus, \( \bar{\xi} = H\text{Loss}(\bar{\theta}) \geq \Delta(\hat{y}(\hat{\theta}), y^*) \). This completes the proof.

**4. Proof of Theorem 3**

Let us first recall from the manuscript the generalization of the parameter learning problem (MMQP : \( \tilde{T} \)) to multiple training images:

\[
\begin{align*}
\text{(MMQP : } \tilde{T}^T) \\
\min_{\theta, \xi, \bar{D}(t)} & \frac{1}{2} ||\theta||^2 + \frac{C}{T} \sum_{t \in T} \xi(t) + \sum_{t \in T} \bar{D}(t) \cdot 1 \\
\text{s.t.} & \{\theta(t), \xi(t), \bar{D}(t)\} \in \mathcal{P}(t), \quad \forall t \in T.
\end{align*}
\]

Also recall that we presented a Lagrangian relaxation based dual-decomposition algorithm that first allocated to each training image its own copy of the parameters \( \theta(t) \):

\[
\begin{align*}
\bar{\theta}, (\xi(t), \bar{D}(t)) & \min \frac{1}{2T} \sum_{t \in T} ||\theta(t)||^2 + \frac{C}{T} \sum_{t \in T} \xi(t) \\
&+ \sum_{t \in T} \bar{D}(t) \cdot 1 \\
\text{s.t.} & \{\theta(t), \xi(t), \bar{D}(t)\} \in \mathcal{P}(t), \quad \forall t \in T.
\end{align*}
\]

The Lagrangian dual of (MMQP : \( \tilde{T}^T \)) was derived as:

\[
(LD : \tilde{T}^T) \max_{\lambda(t)} \sum_{t \in T} \mathcal{F}(t)(\lambda(t)) \quad \text{s.t.} \sum_{t \in T} \lambda(t) = 0,
\]

where \( \mathcal{F}(t) \) are independent sub-problems that are functions of the dual variables \( \lambda(t) \):

\[
\begin{align*}
\mathcal{F}(t)(\lambda(t)) &= \min_{\theta(t), \xi(t), \bar{D}(t)} \frac{1}{2T} ||\theta(t)||^2 + \lambda(t) \cdot \theta(t) \\
&+ \sum_{t \in T} \lambda(t) \cdot \bar{D}(t) \cdot 1 \\
\text{s.t.} & \{\theta(t), \xi(t), \bar{D}(t)\} \in \mathcal{P}(t).
\end{align*}
\]

**Algorithm 1.** We solve this dual problem via projected gradient ascent.

**Theorem 3** \( LD : \tilde{T}^T \) (18) has zero duality gap and Algorithm 1 converges to the optimum of \( \text{MMQP} : \tilde{T}^T \) (16).

**Proof.** Our proof consists of the following steps:

1. **Convexity of LD : \( \tilde{T}^T \) (18):** First, note that by construction, a Lagrangian dual is always concave in multipliers \( \{\lambda(t)\} \) since it is a point-wise minimum of concave (linear) functions of \( \{\lambda(t)\} \).

2. **Optimality of Algorithm 1 for LD : \( \tilde{T}^T \) (18):** Thus, projected gradient ascent converges to the solution of (18).

3. **Zero Duality of LD : \( \tilde{T}^T \) (18):** To show this, we note that \( \text{MMQP} : \tilde{T}^2 \) (17) is a convex problem because it has convex (linear) constraints and the Hessian of objective is positive-definite:

\[
\begin{align*}
\frac{\partial^2 f}{\partial \theta(t) \partial \theta(t)} &= \frac{1}{2T} I_{k \times k} \geq 0 \\
\frac{\partial^2 f}{\partial \xi(t) \partial \xi(t)} &= 0 \\
\frac{\partial^2 f}{\partial D(t) \partial D(t)} &= 0 \\
\end{align*}
\]

\[
\begin{align*}
\frac{\partial^2 f}{\partial \theta(t) \partial \xi(t)} &= 0 \\
\frac{\partial^2 f}{\partial \theta(t) \partial D(t)} &= 0 \\
\frac{\partial^2 f}{\partial \xi(t) \partial D(t)} &= 0 \\
\end{align*}
\]
where,

\[
f(\theta^{(t)}, \xi^{(t)}, D^{(t)}) = \frac{1}{2T} \|\theta^{(t)}\|_2^2 + \lambda^{(t)} \cdot \theta^{(t)}
+ \frac{C}{T} \xi^{(t)} + \frac{C'}{T} D^{(t)} \cdot 1 \tag{21}
\]

Moreover, \(MMQP : \tilde{T}^T 2\) (17) satisfies Slater’s condition [1], which is a sufficient condition for zero duality gap in convex problems, since it has a non-empty feasible set.

Thus, \((LD : \tilde{T}^T)\) (18) achieves the same value as \(MMQP : \tilde{T}^T 2\) (17), which in turn has the same value as \(MMQP : \tilde{T}^T\) (16). This completes the proof. □

References