Frank-Wolfe Dual Decomposition for MAP Inference

Xu Hu
Email: huxu@onid.orst.edu

Michael Lam
Email: lamm@onid.orst.edu

***Abstract*—Many problems in computer vision are formulated as structured prediction problems, which can be addressed with conditional random fields (CRFs), a particular probabilistic graphical model (PGM). Cast as an optimization problem, the MAP (maximum a posterior) inference of CRF can be decomposed into subproblems that can be solved in parallel for efficiency. The whole idea is called dual decomposition, which is well-established in the optimization community. This paper proposes decomposing a graphical model into spanning trees, and formulates the dual problem of MAP inference as a constrained convex optimization problem, which is solved by the Frank-Wolfe algorithm [1] efficiently. Our experiments on image denoising and scene labeling show that our proposed MAP inference method performs on par with the state-of-the-art methods.**

I. Introduction

In computer vision, many problems–including image denoising and scene labeling–can be formulated as a structured prediction problem, where the goal is to minimize an energy function that maps a structured input $\mathbf{x}$ to corresponding structured outputs $\mathbf{y}$. For example, in scene labeling, the structured input $\mathbf{x}$ is an image and the structured output $\mathbf{y}$ assigns a semantic class label to every pixel in the image.

An energy function typically factorizes into a sum of local potential functions over overlapping subsets of variables. However, most MAP inference (or energy minimizations) involving discrete variables are NP-hard [2], due to high-treewidth input structures. The intractablity forces to use approximation algorithms. One common solution resorts to loopy belief propagation (LBP) [3], which is a generalization of the forward-backward algorithm for Markov chains [4]. Loopy belief propagation can often provides good approximations, but there is no guarantee to converge to a local minima. Alternatively, one could relax the MAP inference to a linear programming (LP), and use the local polytope as constraints, which is an approximation of the marginal polytope [5]. The fractional solution of LP is further rounded to produce the final MAP solution. Although LP is tractable, general LP solvers do not exploit the structure. Furthermore, the quality of the LP relaxation for MAP inference is highly related to the marginal polytope approximation [6]. Thus, LP relaxation usually provides poor results in practice.

To address this problem, a popular solution is to decompose the original NP-hard problem into several tractable subproblems that can be each solved efficiently via the dual decomposition (DD). Dual decomposition is a well-established idea in optimization, whereby an objective function that is a sum of functions over subsets of variables is decoupled into independent subproblems by introducing Lagrangian multipliers. These multipliers are then adjusted in the master problem to assure that the solutions of subproblems agree to each other. DD has been applied to MAP inference. Komodakis et al. [7] and Domke [8] use the strategy that decompose the master problem into a set of spanning trees. Sontag et al. [9] provided an alternative decomposition in terms of cliques and shows the connection between DD and the dual formulation of the LP relaxation.

In this paper, we follow the common strategy that decompose the high-treewidth graph into a set of spanning trees, thus each subproblem has treewidth one and can be solved exactly by tree structured belief propagation. We propose a different method to optimize the master problem. Unlike the projected subgradient descent used in [7], we cast the master problem as a constrained convex optimization. The well known Franke-Wolfe algorithm [1], [10] is used for solving this constrained convex optimization, in which only linear operations are used, while the projected subgradient descent requires to solve a quadratic projection in each iteration.

We evaluate the proposed MAP inference method (referred as Franke-Wolfe dual decomposition (FWDD) in the following) in two standard structured prediction tasks: image denoising and scene labeling. The results demonstrate that our FWDD performs on par with the state-of-the-art methods.

The rest of the paper is organized as follows. Sec. II reviews the basic knowledge of conditional random filed (CRF) and MAP inference. Sec. III gives the details of our FWDD inference method. Finally, Sec. V presents our experimental results.

II. Background

A probabilistic graphical model (PGM) is a probabilistic model for which a graph denotes the conditional independence structure between random variables. For tasks such as image denoising and scene labeling, structured prediction is often addressed with the conditional random field (CRF), a particular probabilistic graphical model (PGM). A CRF defines a parametric posterior distribution over the outputs (labels), y, given observed image features, x, in a factored form: $P(y|x,w) = \frac{1}{Z(x,w)} e^{w \cdot \phi(x,y)}$, where w are the model parameters, $Z(x,w)$ is the partition function, and the features, $\phi(x,y)$, decompose over the cliques in the underlying graphical model. CRF inference is typically posed as finding the joint MAP assignment that maximizes the posterior distribution: $\hat{y} = \arg\max_{y \in \mathcal{Y}} P(y|x,w)$. This MAP inference is generally intractable due to the exponential space of outputs.

In PGMs, the MAP inference is a discrete optimization defined in terms of

$$f(\mathbf{x};\boldsymbol{\theta}) = \sum_{c \in C} \boldsymbol{\theta}_c(\mathbf{x}_c), \tag{1}$$

where $\mathbf{x} = (x_1, \ldots, x_n)^\top$ are discrete variables, which are grouped into a set of cliques C in a graph $G = (V, E)$. Each clique is associated with a potential function $\boldsymbol{\theta}_c(\mathbf{x}_c)$ that characterizes the marginal distribution of the variables within the clique c [1]. To slightly abuse the notation, we will also use $c \in G$ to represent the clique c is contained in G. In computer vision, usually a 1-clique (i.e. a single node in G) represents image features at a particular pixel (or some small neighborhood around it), and usually a 2-clique (i.e. an edge in G) models the label compatibility (i.e. encourages smoothness) between two nodes.

The MAP inference is to find the most probable assignment $\mathbf{x}^*$ of f, namely,

$$\mathbf{x}^* = \arg\max_{\mathbf{x}} f(\mathbf{x}). \tag{2}$$

In general, the MAP inference is NP-hard. Exact and efficient inference is only possible when the dependency structure among features contains no loops, in other words, a tree or chain. However, in computer vision often the graphical model is loopy. To address this issue, inference is usually approximate. There are a wide range of approximate inference algorithms that employ message passing, reduce the graphical model to a network flow and so on.

A popular approximation in an optimization framework is a linear programming relaxation [6]:

$$\max_{\boldsymbol{\mu} \in \mathcal{M}_G} \langle \boldsymbol{\theta}, \boldsymbol{\mu} \rangle. \tag{3}$$

where $\langle \boldsymbol{\theta}, \boldsymbol{\mu} \rangle = \sum_c \sum_{\mathbf{x}_c} \boldsymbol{\theta}_c(\mathbf{x}_c) \boldsymbol{\mu}_c(\mathbf{x}_c)$, and $\mathcal{M}_G$ is the local polytope associated with local beliefs, *i.e.*, $\{\boldsymbol{\mu}_c\}$, formally

$$\mathcal{M}_G = \big\{ \boldsymbol{\mu} \geq 0 \colon \sum_{c \setminus a} \boldsymbol{\mu}_c(\mathbf{x}_c) = \boldsymbol{\mu}_a(\mathbf{x}_a) \quad \forall a \subseteq c, \\ \sum_a \boldsymbol{\mu}_a(\mathbf{x}_a) = 1 \quad \forall a \in G \big\}. \tag{4}$$

However, note that the linear program contains an exponential number of constraints, which makes the problem still intractable.

III. Dual Decomposition Using Franke-Wolfe

A standard solution to speed up the linear programming relaxation is to decompose the problem into several simple subproblems, solve the subproblems independently and fuse together the results of these subproblems into the final solution. Komodakis *et al*. [7] and Domke [8] proposed to decompose the intractable loopy graph into a set of spanning trees, and demonstrated promising results in several computer vision tasks. Sontag *et al*. [9] and Meshi *et al*. [11] proposed another decomposition in terms of cliques. In this way, Lagrangian multipliers can be viewed as messages passing between cliques.

In this work, we will use the spanning tree decomposition. An example of valid decomposition is shown in Fig. 1. Note that in a valid decomposition, every clique of the original problem is covered by a corresponding clique in some spanning tree subproblem. We develop Theorem 5, such that each subproblem can be solved by standard sum-product message passing, while the master problem becomes a constraint convex optimization, thus can be solved by Frank-Wolfe algorithm efficiently. Thus the optimization problem in Theorem 5 is our problem formulation.

[1]Note that the joint probability is specified as $p(\mathbf{x}) = \frac{1}{Z(\boldsymbol{\theta})} \exp(\sum_{c \in C} \boldsymbol{\theta}_c(\mathbf{x}_c))$

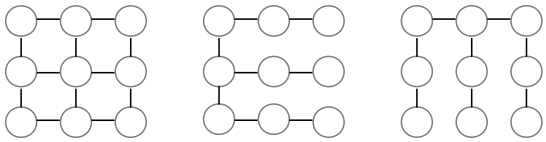

Fig. 1: **Left**: the original grid graph. **Middle and Right**: A spanning tree decomposition with two spanning trees. Note that, in a valid decomposition, each clique will be covered by some spanning trees.

Theorem 1. *Given a set of spanning trees $\mathcal{T}$ of the graph G, the linear programming relaxation in Eq. (3) can be formulated as an equivalent convex optimization*

$$\begin{aligned} \min_{\boldsymbol{\omega} \geq 0} \quad & \sum_{T \in \mathcal{T}} \max_{\boldsymbol{\nu}_T \in \mathcal{M}_T} \langle \boldsymbol{\omega}_T, \boldsymbol{\nu}_T \rangle \\ \textit{s.t.} \quad & \sum_{T \in \mathcal{T}} \boldsymbol{\omega}_T = \boldsymbol{\theta} \quad \forall T \in \mathcal{T} \end{aligned} \tag{5}$$

Here, $\boldsymbol{\omega}_T$ and $\boldsymbol{\nu}_T$ can be interpreted as potentials and beliefs of the spanning tree T. They are of the same length as $\boldsymbol{\theta}$ with zeros filled to corresponding elements where edges are not present in G.

Proof: Given a set of spanning trees $\mathcal{T}$, we can construct a potential $\boldsymbol{\theta}_T$ for each spanning tree T, such that $\sum_{T: c \in T} \boldsymbol{\theta}_{Tc} = \boldsymbol{\theta}_c$, for all $c \in C$. Then the original optimization in Eq. (3) is equivalent to

$$\max_{\boldsymbol{\mu} \in \mathcal{M}_G} \quad \sum_c \langle \sum_{T: c \in T} \boldsymbol{\theta}_{Tc}, \boldsymbol{\mu}_c \rangle \tag{6}$$

By introducing a copy of $\boldsymbol{\mu}_c$ in each spanning tree which has the clique c, and constrain them to be the same as $\boldsymbol{\mu}_c$, we decompose the original problem into subproblems in spanning trees:

$$\begin{aligned} \max_{\boldsymbol{\mu} \in \mathcal{M}_G} \max_{\boldsymbol{\nu}} \quad & \sum_c \sum_{T: c \in T} \langle \boldsymbol{\theta}_{Tc}, \boldsymbol{\nu}_{Tc} \rangle \\ \text{s.t.} \quad & \boldsymbol{\nu}_{Tc} = \boldsymbol{\mu}_c, \quad \forall T, c \\ & \boldsymbol{\nu}_T \in \mathcal{M}_T, \quad \forall T \in \mathcal{T} \end{aligned} \tag{7}$$

where $\boldsymbol{\nu} = \{\boldsymbol{\nu}_T\}_{T \in \mathcal{T}}$. [8] shows that the above decomposition is equivalent to

$$\begin{aligned} \max_{\boldsymbol{\nu}} \quad & \sum_c \sum_{T: c \in T} \langle \boldsymbol{\theta}_{Tc}, \boldsymbol{\nu}_{Tc} \rangle \\ \text{s.t.} \quad & \boldsymbol{\nu}_{Tc} = \frac{1}{N_c} \sum_{T': c \in T'} \boldsymbol{\nu}_{T'c}, \quad \forall T, c \\ & \boldsymbol{\nu}_T \in \mathcal{M}_T, \quad \forall T \in \mathcal{T} \end{aligned} \tag{8}$$

where $N_c = |T \colon c \in T|$, *i.e.*, the number of spanning trees that contain the clique c.

The Lagrangian is given by

$$L(\boldsymbol{\nu}, \boldsymbol{\lambda}) = \sum_c \sum_{T:c\in T} \langle \boldsymbol{\theta}_{Tc}, \boldsymbol{\nu}_{Tc} \rangle + \sum_c \sum_{T:c\in T} \lambda_{Tc} \left(\boldsymbol{\nu}_{Tc} - \frac{1}{N_c} \sum_{T':c\in T'} \boldsymbol{\nu}_{T'c} \right) \quad (9)$$

and the dual problem is

$$\begin{aligned} \min_{\boldsymbol{\lambda}} \max_{\boldsymbol{\nu}} \quad & L(\boldsymbol{\nu}, \boldsymbol{\lambda}) \\ \text{s.t.} \quad & \boldsymbol{\nu}_T \in \mathcal{M}_T, \quad \forall T \in \mathcal{T} \end{aligned} \quad (10)$$

Now define $\boldsymbol{\omega}_{Tc} = \boldsymbol{\theta}_{Tc} + \lambda_{Tc} - \frac{1}{N_c} \sum_{T':c\in T'} \lambda_{T'c}$. Observing that

$$\begin{aligned} \sum_{T:c\in T} \boldsymbol{\omega}_{Tc} &= \sum_{T:c\in T} \boldsymbol{\theta}_{Tc} + \lambda_{Tc} - \frac{1}{N_c} \sum_{T':c\in T'} \lambda_{T'c} \\ &= \sum_{T:c\in T} \boldsymbol{\theta}_{Tc} = \boldsymbol{\theta}_c \end{aligned} \quad (11)$$

thus we can transfer Lagrangian multipliers λ_{Tc} into $\boldsymbol{\omega}_{Tc}$, by substituting $\boldsymbol{\theta}_{Tc} = \boldsymbol{\omega}_{Tc} - \lambda_{Tc} + \frac{1}{N_c} \sum_{T':c\in T'} \lambda_{T'c}$ into Eq. (9). The dual problem can be rewriten accordingly as

$$\begin{aligned} \min_{\boldsymbol{\omega}} \max_{\boldsymbol{\nu}} \quad & \sum_c \sum_{T:c\in T} \langle \boldsymbol{\omega}_{Tc}, \boldsymbol{\nu}_{Tc} \rangle \\ \text{s.t.} \quad & \sum_{T:c\in T} \boldsymbol{\omega}_{Tc} = \boldsymbol{\theta}_c, \quad \forall c \\ & \boldsymbol{\nu}_T \in \mathcal{M}_T, \quad \forall T \in \mathcal{T} \end{aligned} \quad (12)$$

which can be simplified as the same form as Eq. (5). ■

Note that Eq. (5) is a constrained convex optimization and consists of subproblems with respect to each tree of the form

$$S(\boldsymbol{\omega}_T) = \max_{\boldsymbol{\nu}_T} \langle \boldsymbol{\omega}_T, \boldsymbol{\nu}_T \rangle, \quad (13)$$

which can be solved exactly by running the sum-product message passing on the tree T [6].

IV. Algorithm

Let $M(\boldsymbol{\omega}) = \sum_{T\in\mathcal{T}} S(\boldsymbol{\omega}_T)$ denote the master problem of Eq. (5). We will use the Frank-Wolfe algorithm [1], [10] to solve this constraint convex optimization. Briefly, the algorithm iterates the following steps:

Algorithm 1 Frank-Wolfe Algorithm

Require: Initialize: $\boldsymbol{\omega}^k$ = initial potentials
  repeat
    Direction: $\mathbf{s} \leftarrow \arg\min_{\boldsymbol{\omega}\geq 0} \langle \boldsymbol{\omega}, \nabla M(\boldsymbol{\omega}^k) \rangle$
      subject to $\sum_{T\in\mathcal{T}} \boldsymbol{\omega}_T = \boldsymbol{\theta}$.
    Step size: $\gamma \leftarrow \frac{2}{2-k}$
    Update: $\boldsymbol{\omega}^{k+1} \leftarrow \boldsymbol{\omega}^k + \gamma(\mathbf{s} - \boldsymbol{\omega}^k)$.
  until $|\langle \boldsymbol{\omega} - \mathbf{s}, \nabla M(\boldsymbol{\omega}^k) \rangle| < \epsilon$

Since $\frac{\partial M(\boldsymbol{\omega})}{\partial \boldsymbol{\omega}_T} = \frac{\partial S(\boldsymbol{\omega})}{\partial \boldsymbol{\omega}_T} = \boldsymbol{\nu}_T^*$, by Danskin's theorem, the direction step is trivial, which involves solving a linear programming with a single linear constraint. Then the step size γ is updated so that it becomes smaller with each iteration. Finally, $\boldsymbol{\omega}$ is updated where the descent direction is computed based on the computed direction and step size. This is repeated for K iterations or when the duality gap is sufficiently small: $|\langle \boldsymbol{\omega} - \mathbf{s}, \nabla M(\boldsymbol{\omega}^k) \rangle| < \epsilon$.

V. Experiments

We evaluate our approach on computer vision tasks that transform a structured input to a structured output. These computer vision tasks employ the conditional random field where the structured output consists of labels for every pixel and the structured input are image features. CRF inference is addressed with our approach.

We evaluate our approach on two tasks: image denoising and scene labeling. The image denoising task involves a toy dataset, which provides a good check that our approach is working. We then evaluate our approach on the scene labeling task. We evaluate scene labeling on the Stanford Background dataset, a standard dataset of natural scenes used in computer vision.

A. Image Denoising

Our first task is to evaluate our approach for image denoising. Given a corrupted, noisy image, the task of inference is to recover the original image; in other words, we want to denoise the image.

Setup. We model the CRF as a regular grid of pixels (i.e. every node corresponds to a pixel) with 4-connected neighbors. The unary potentials consist of pixel intensities and pairwise potentials employ the Pott's model to encourage smoothness. This graphical model is decomposed into two spanning tree graphical models that is used for dual decomposition as in Fig. 1.

Dataset. Our dataset consists of 20 binary images with various white circles placed at different locations over a black background. The dataset is corrupted with Gaussian noise (i.e. every pixel in the image is changed by a value sampled from a Gaussian distribution) for input into the denoising task. This noisy image is a grayscale image.

Metrics. Our metric is pixel accuracy. Every pixel of the inferred image is compared to the groundtruth image.

Baselines. The baseline inference algorithm is loopy belief progagation (LBP). Empirically this yields the best result among other possible baseline inference algorithms.

Quantitative Results. We present some quantitative results in Table I. Both inference methods perform well in this toy dataset, which shows that our FWDD is at least as efficient as the state-of-the-art approximation algorithm.

Approach	Pixel Accuracy
Baseline	97.71%
Ours	98.05%

TABLE I: Pixel accuracy on the image denoising task comparing our approach and the baseline approach. Pixel accuracy is the percentage of correct labels over all pixels.

Duality Gap. We demonstrate that the duality gap decreases as the number of iterations increases in Fig. 2. It shows that our FWDD is sublinear convergence.

We also demonstrate the effect of dual decomposition in 3. We see that the marginal beliefs for the tree decompositions make sense based on the "smearing" artifacts. Each tree

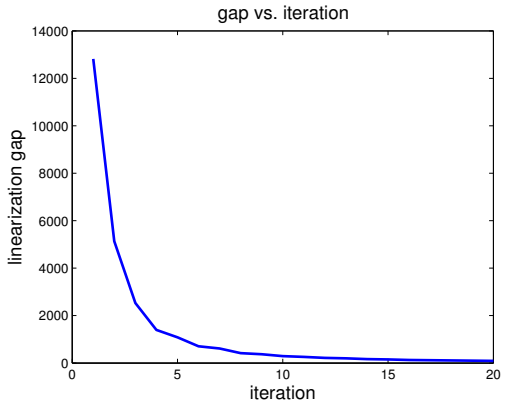

Fig. 2: The duality gap decreases as the number of iterations increases.

decomposition consists of pairwise potentials that are only present in the vertical or horizontal direction. The results of tree decomposition 1 shows "smearing" in the horizontal direction, which aligns with the spanning tree model with horizontal pairwise potentials. We also see that the final marginal beliefs "average out" these artifacts to yield a good solution.

B. Scene Labeling

We consider the task of scene labeling on a standard computer vision dataset for general images of natural scenes, the Stanford Background Dataset. We formulate the scene labeling problem as an optimization problem and use our proposed algorithm for performing inference. The task of scene labeling is to label every pixel in an image with a semantic class.

Setup. The image structure can be modeled with a CRF where the structured output consists of a semantic label for every pixel and the structured input are image features. The unary potentials for each pixel corresponds to the features around a neighborhood of that pixel. The features are color and texture. The pairwise potentials for every pair of neighboring pixels models the compatibility of the neighbor label assignments, which is used for smoothness. Similar to the image denoising task, this graphical model is also decomposed into two spanning tree graphical models that is used for dual decomposition as in Fig. 1.

Dataset. Our dataset is the Stanford Background Dataset. This is a standard dataset consisting of 715 general images of natural scenes and 8 semantic classes.

Metrics. Similar to the image denoising task, we evaluate with pixel accuracy.

Features: We follow the feature extraction process recently evaluated as suitable for semantic segmentation in [12]. Specifically, a superpixel is characterized by a feature vector consisting of the following properties: (1) texture: 64-dimensional histogram over textons; (2) color: 64-dimensional histogram of LAB colors generated by running K-means over pixels in the LAB color space; (3) layout: 12-dimensional histogram of pixel locations on a 3×4 grid. All these

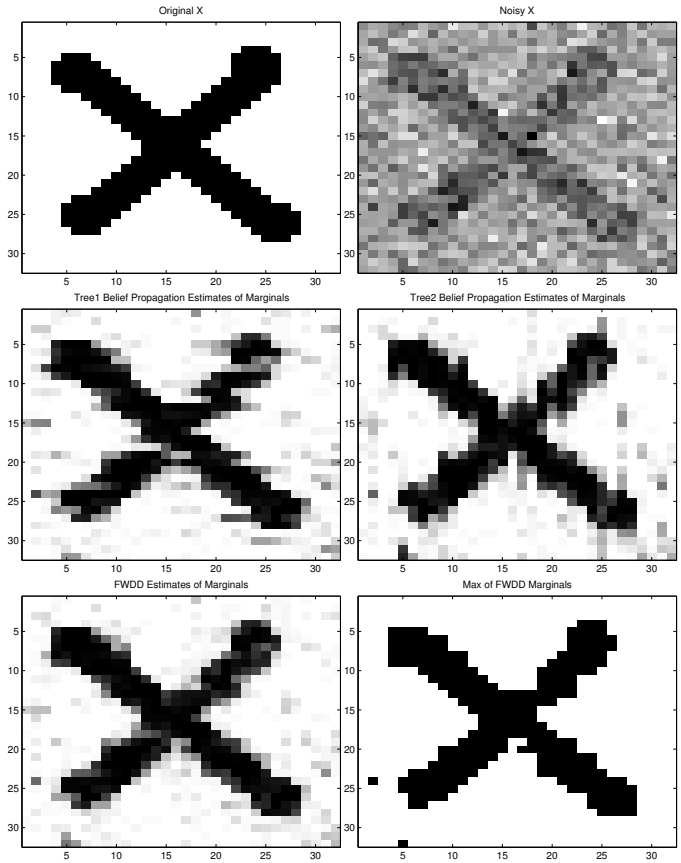

Fig. 3: The **top left** figure shows the original image and the **top right** figure shows the corrupted with Gaussian noises. The **middle left** figure shows the marginal beliefs of tree decomposition 1 and the **middle right** figure shows the marginal beliefs of tree decomposition 2. The **bottom left** figure fuses the results of the two decompositions into the final marginal beliefs. The **bottom right** figure gives the final denoised image from the final marginal beliefs.

histograms are normalized separately, and normalized again after concatenating them together to form a 140-dimensional feature vector.

Baselines. Multinomial logistic regression (LR) on individual superpixels, and Conditional Random Field (CRF) defined over superpixels with pairwise dependencies for every pair of adjacent superpixels, implemented using [13]. The superpixel features used for the unary potential are described above. The features for the pairwise potential are the difference of the superpixel features.

Quantitative Results. Tab. II shows results of related methods [14], [15], [16] and our results. The row marked "upper bound" shows the best performance obtainable using the given super-pixelization. It can be seen that our FWDD provides a 2.08% improvement over the LR baseline and outperformances all the baselines. However, our results are slightly worse than the best state-of-the-art methods. This is because they use a much better feature learning. It can be expected that using the same set of features, our FWDD can achieve similar performance.

TABLE II: The comparison of average pixel-wise accuracy on the Stanford background dataset.

Methods	Acc. %
Region Energy [17]	76.4
SHL [14]	76.9
RNN [15]	78.1
LR	76.33
CRF	77.63
FWDD	78.41

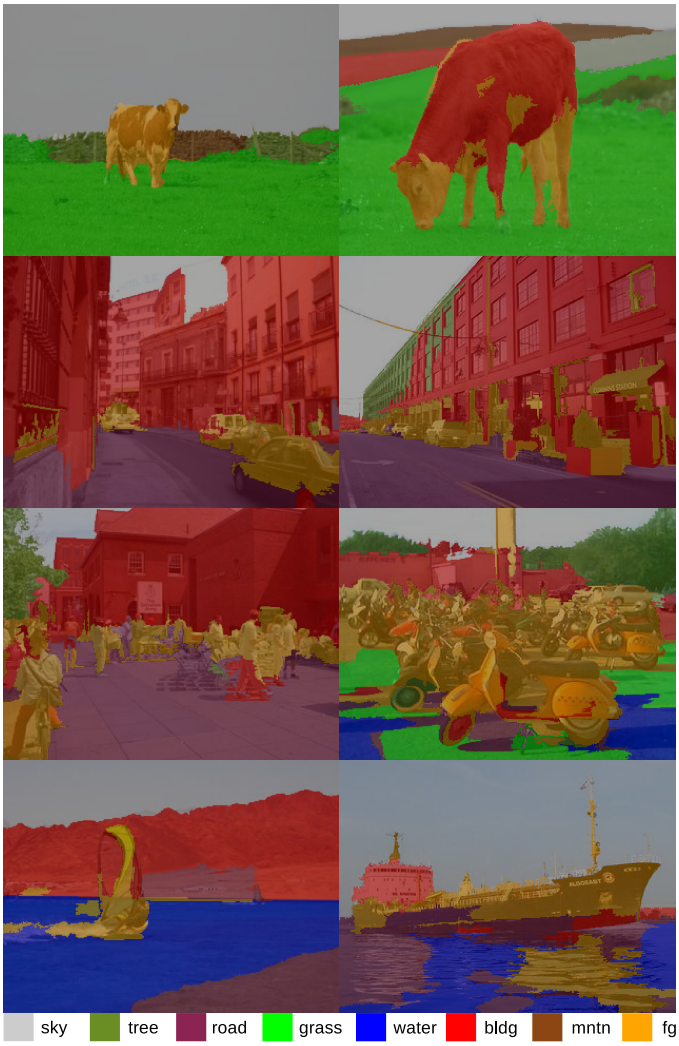

Fig. 4: Scene labeling results. The **left** shows good results. The **right** shows results with incorrect predictions.

Qualitative Results. We present some qualitative results in Fig. 4.

VI. Conclusion

We have presented an approach for solving the MAP inference problem for CRFs. Our formulation of the dual problem to MAP inference allows for efficient inference by solving subproblems. The Frank-Wolfe algorithm is able to solve our constrained convex optimization formulation. We evaluated our approach on image denoising and scene labeling and show that we are getting good results. Our formulation and algorithm is a promising framework for MAP inference of CRFs.

Contributions of Individual Team members

Hu primarily contributed to the theory and formulations while Lam primarily contributed to the codebase, although both Hu and Lam contributed across the whole project. Recognition is given to Hu for coming up with the project idea.

References

[1] M. Franke and P. Wolfe, "An algorithm for quadratic programming," *Naval Res. Logis. Quart.*, 1956.

[2] D. Koller and N. Friedman, *Probabilistic Graphical Models: Principles and Techniques*. MIT Press, 2009.

[3] J. S. Yedidia, W. T. Freeman, and Y. Weiss, "Constructing free energy approximations and generalized belief propagation algorithms," *IEEE Transactions on Information Theory, vol. 51, July 2005, pp. 2282-2313*, 2005.

[4] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proc. IEEE*, 1989.

[5] D. Sontag, T. Meltzer, A. Globerson, Y. Weiss, and T. Jaakkola, "Tightening LP relaxations for MAP using message-passing," in *24th Conference in Uncertainty in Artificial Intelligence*. AUAI Press, 2008, pp. 503–510.

[6] C. Yanover, T. Meltzer, and Y. Weiss, "Linear programming relaxations and belief propagation – an empirical study," *JMLR*, 2006.

[7] N. Komodakis, N. Paragios, and G. Tziritas, "Revisiting frank-wolfe: Projection-free sparse convex optimization," *ICCV*, 2007.

[8] J. Domke, "Dual decomposition for marginal inference," *AAAI*, 2011.

[9] D. Sontag, A. Globerson, and T. Jaakkola, "Introduction to dual decomposition for inference," *Tech. Report*, 2010.

[10] M. Jaggi, "Revisiting frank-wolfe: Projection-free sparse convex optimization," *ICML*, 2013.

[11] O. Meshi, D. Sontag, T. Jaakkola, and A. Globerson, "Learning efficiently with approximate inference via dual losses," *ICML*, 2010.

[12] A. Kae, K. Sohn, H. Lee, and E. Learned-Miller, "Augmenting CRFs with Boltzmann Machine Shape Priors for Image Labeling," *CVPR*, 2013.

[13] M. Schmidt, http://www.di.ens.fr/ mschmidt/Software/UGM.html, 2013.

[14] D. Munoz, J. A. Bagnell, and M. Hebert, "Stacked Hierarchical Labeling," in *ECCV*, 2010.

[15] R. Socher, C. C. Lin, A. Y. Ng, and C. D. Manning, "Parsing Natural Scenes and Natural Language with Recursive Neural Networks," in *ICML*, 2011.

[16] C. Farabet, C. Couprie, L. Najman, and Y. LeCun, "Learning Hierarchical Features for Scene Labeling," *IEEE Trans PAMI*, vol. 35, no. 8, pp. 1915–1929, 2013.

[17] S. Gould, R. Fulton, and D. Koller, "Decomposing a scene into geometric and semantically consistent regions," in *ICCV*, 2009.