High Order Regularization for Semi-Supervised Learning of Structured Output Problems

Yujia Li[1] YUJIALI@CS.TORONTO.EDU
Richard Zemel[1,2] ZEMEL@CS.TORONTO.EDU
[1]Department of Computer Science, University of Toronto, Toronto, ON, Canada
[2]Canadian Institute for Advanced Research, Toronto, ON, Canada

Abstract

Semi-supervised learning, which uses unlabeled data to help learn a discriminative model, is especially important for structured output problems, as considerably more effort is needed to label its multi-dimensional outputs versus standard single output problems. We propose a new max-margin framework for semi-supervised structured output learning, that allows the use of powerful discrete optimization algorithms and high order regularizers defined directly on model predictions for the unlabeled examples. We show that our framework is closely related to Posterior Regularization, and the two frameworks optimize special cases of the same objective. The new framework is instantiated on two image segmentation tasks, using both a graph regularizer and a cardinality regularizer. Experiments also demonstrate that this framework can utilize unlabeled data from a different source than the labeled data to significantly improve performance while saving labeling effort.

1. Introduction

Structured prediction is the problem of predicting a multi-dimensional output from input, where the structure of the output has to be considered when making predictions. Typical examples of structured prediction include sequence labeling problems in NLP, where the output is a 1-D chain of labels, and semantic image segmentation from computer vision, where the output is a (grid) graph of pixel labels. Due to the complexity of the outputs, obtaining labels for structured prediction problems requires considerably more effort than for standard classification or regression tasks. As a result, while large classification datasets, such as ImageNet, contain millions of labeled examples, the largest publicly available image segmentation datasets, e.g., PASCAL VOC, have only a few thousand examples with complete labels. At the same time, large amounts of unlabeled data are typically very easy to obtain.

This combination of difficulty to obtain labeled examples for structured prediction problems, with abundant unlabeled data makes semi-supervised learning (SSL) especially worth exploring. However, SSL is challenging for structured prediction because the complex high dimensional output space makes a lot of operations intractable. A dominant approach to SSL is to use unlabeled data to regularize the model by ensuring that its predictions on unlabeled data are consistent with some prior beliefs. For example, entropy regularization (Lee et al., 2006) and low density separation (Zien et al., 2007) regularize the model so that it makes confident predictions on unlabeled data. Graph-based methods (Altun et al., 2006; Subramanya et al., 2010), on the other hand, regularize the model to make smooth predictions for unlabeled data on a graph.

Recently, posterior regularization (PR) (Ganchev et al., 2010) has been introduced as a general framework to incorporate prior constraints about predictions into structured prediction models. A version of it has also been applied to graph-based SSL for sequence labeling (He et al., 2013). In PR, constraints are specified as regularizers on posterior distributions, and a decomposition technique is used to make the optimization tractable for structured outputs.

In this paper, we propose a new max-margin framework for semi-supervised structured output learning, that allows regularizers to be defined directly on the predictions of the model for unlabeled data, instead of using the posterior distribution as a proxy. This makes it possible to specify a range of regularizers that are not easy to define on distributions, including those involving loss functions and cardinality of outputs. One advantage of a max-margin framework is that at test time we typically only want to produce the most likely output, which is generally easier than marginal inference in probabilistic frameworks. For example, in image segmentation, MAP inference can be done efficiently on graphs with submodular pairwise potentials using powerful discrete optimization techniques like graph

Proceedings of the 31^{st} International Conference on Machine Learning, Beijing, China, 2014. JMLR: W&CP volume 32. Copyright 2014 by the author(s).

cuts, which is key to the success of many segmentation methods. However, marginal inference is intractable due to the extremely loopy structure of the graph. Therefore while most of the previous work on SSL studied sequences, our new framework is especially suitable for structured outputs beyond 1-D sequences.

In this paper we also explore the relationship between our method and PR. We show that the two approaches are actually very closely related: our framework and PR optimize two special cases of the same objective function for some general settings. This connection opens a range of new possibilities of designing and analyzing frameworks for incorporating prior constraints into the model.

We then demonstrate the new framework with an application to graph-based SSL for image segmentation. In graph-based SSL, an important issue is to choose a proper similarity metric in the output space. We utilize the loss function, which offers a natural similarity metric in the output space, as the metric in our formulation.

The rest of the paper is organized as follows. Section 2 briefly discusses related work. Section 3 describes the proposed framework in detail. Section 4 shows the connection between our framework and PR. Section 5 presents our experiment results on two foreground-background segmentation tasks. Section 6 concludes the paper.

2. Related Work

The earliest work on SSL dates back to the study of the wrapper method known as self-training, in the 1960s, e.g., (Scudder III, 1965). Self-training iteratively uses the predictions of the model on unlabeled data as true labels to retrain the model. Because of its heuristic nature, this method is hard to analyze and its performance gains from the unlabeled data are typically not significant.

A wide range of SSL methods have been developed for classification problems to date (Nigam et al., 1998; Joachims, 1999; Grandvalet & Bengio, 2005; Zhu et al., 2003; Zhou et al., 2004; Belkin et al., 2006; Blum & Mitchell, 1998); see (Zhu, 2005) and (Chapelle et al., 2006) for excellent surveys and additional references.

Some researchers have adapted these methods to structured output problems. These methods generally fall into one of the following categories:

(a). Co-training, which iteratively uses the predictions made by models trained on different views of the same data to label the unlabeled set and update the model using the predicted labels (Brefeld & Scheffer, 2006). The applicability of this method is limited due to the requirement of multi-view data.

(b). Generative models, which use unlabeled data to help learning a model of the joint input-output distribution $p(\mathbf{x}, \mathbf{y})$. While having some early success for classification problems (Nigam et al., 1998), generative models make strong assumptions about the data and have to date achieved limited success on structured output problems.

(c). Low density separation based methods, which encourage confident predictions on unlabeled data. This translates to low entropy of the output posterior distribution in a probabilistic modeling framework (Lee et al., 2006), and large margin for methods in a max-margin framework (Zien et al., 2007). A combined objective is optimized to minimize the sum of the task loss on the labeled data and a separation regularizer on the unlabeled data.

(d). Graph based methods, which construct a graph that connects examples that are nearby in the input space, and then encourage the predictions by the model for pairs of connected examples to be close as well. Most of the work in this category deals with sequence labeling problems. Altun et al. (2006) uses a graph on parts of $\mathbf{y}$ to derive a graph regularized kernel which is used in a max-margin framework. Unlike our framework described below, this approach is not able to incorporate other high order regularizers. Subramanya et al. (2010) proposes a semi-supervised Conditional Random Field (CRF) that infers labels for unlabeled data by propagation on a graph of parts, and then retrains the model using the inferred labels. Finally, Vezhnevets et al. (2011) proposes a graph-based method for semi-supervised image segmentation, which utilizes unlabeled examples in learning by inferring labels for them based on a graph defined on image superpixels.

Recently, other general frameworks for SSL in structured output problems have been defined that can be viewed as graph-based. Posterior regularization (PR) (Ganchev et al., 2010) is a framework to incorporate constraints on structured probabilistic models through regularizers defined on posterior distributions. He et al. (2013) applies this general PR framework to graph-based SSL also using a CRF model. PR is closely related to our framework: we show in Section 4 that the two frameworks are optimizing special cases of the same objective. Constraint Driven Learning (CODL) (Chang et al., 2007) and Generalized Expectation Criteria (Mann & McCallum, 2010) are two other notable frameworks for incorporating contraints into the model.

A separate but related line of research is the study of transfer learning or domain adaptation (Pan & Yang, 2010), where most of the labeled data comes from a source domain and task performance is evaluated in a different target domain, typically with little labeled data available. We explore some domain adaptation settings in our experiments presented in Section 5.

3. Formulation

3.1. Background: Structured Output Learning

In structured output problems, the aim is to learn a mapping from $\mathbf{x}$ in input space $\mathcal{X}$ to $\mathbf{y}$ in structured output space $\mathcal{Y}$, given a set of labeled data $D_L = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{L}$. The mapping is usually implicitly determined by a score function

$f(\mathbf{x}, \mathbf{y}, \mathbf{w})$ where $\mathbf{w}$ is the set of parameters and the prediction $\mathbf{y}^* = \operatorname{argmax}_{\mathbf{y}} f(\mathbf{x}, \mathbf{y}, \mathbf{w})$.

There are two dominant paradigms of structured output learning, based on how the score function is used. The max-margin methods (Taskar et al., 2004; Tsochantaridis et al., 2005) maximize the margin between the score for the correct output and all other outputs. The structured hinge loss is usually used in max-margin methods:

$$\mathcal{L}_h(\mathbf{x}_i, \mathbf{y}_i, \mathbf{w}) = \max_{\mathbf{y}} [f(\mathbf{x}_i, \mathbf{y}, \mathbf{w})] - f(\mathbf{x}_i, \mathbf{y}_i, \mathbf{w}) \quad (1)$$

A standard approach in max-margin learning is to incorporate the task loss into the hinge loss (Taskar et al., 2004),

$$\mathcal{L}(\mathbf{x}_i, \mathbf{y}_i, \mathbf{w}) = \max_{\mathbf{y}} [f(\mathbf{x}_i, \mathbf{y}, \mathbf{w}) + \Delta(\mathbf{y}, \mathbf{y}_i)] - f(\mathbf{x}_i, \mathbf{y}_i, \mathbf{w}) \quad (2)$$

where Δ is the task loss. The second paradigm includes probabilistic models, such as CRFs (Lafferty et al., 2001), which interpret the score function as implying a distribution over the outputs $p(\mathbf{y}|\mathbf{x}) \propto \exp(f(\mathbf{x}, \mathbf{y}, \mathbf{w}))$ and then adapt $\mathbf{w}$ to maximize the conditional likelihood.

As a concrete example, for binary segmentation problems, $\mathcal{X}$ is the space of images and $\mathcal{Y} = \{0, 1\}^P$, where P is the number of pixels in an image. $f(\mathbf{x}, \mathbf{y}, \mathbf{w})$ usually has the form of $f(\mathbf{x}, \mathbf{y}, \mathbf{w}) = \sum_{i \in V} f^u(\mathbf{x}, y_i, \mathbf{w}^u) + \sum_{(i,j) \in E} f^p(\mathbf{x}, y_i, y_j, \mathbf{w}^p)$, which is a sum of *unary* potentials defined on individual pixels and *pairwise* potentials defined on pairs of neighboring pixels. Usually $G = (V, E)$ is a grid graph. When the pairwise potentials satisfy certain properties, namely submodularity, the exact optimal $\mathbf{y}^*$ can be found using graph cuts. See (Nowozin & Lampert, 2011) for an excellent review of structured learning and prediction.

3.2. High Order Regularized SSL

In an SSL setting, we have a set of unlabeled data $D_U = \{\mathbf{x}_j\}_{j=L+1}^{L+U}$ in addition to the labeled data D_L. Our objective for learning is composed of a loss defined on labeled data, and a regularizer defined directly on predictions of the model on unlabeled data:[1]

$$\min_{\mathbf{w}} \quad \sum_{i=1}^{L} \mathcal{L}(\mathbf{x}_i, \mathbf{y}_i, \mathbf{w}) + R\left(\{\mathbf{y}_j\}_{j=L+1}^{L+U}\right) \quad (3)$$
$$\text{s.t.} \quad \mathbf{y}_j = \operatorname*{argmax}_{\mathbf{y}} f(\mathbf{x}_j, \mathbf{y}, \mathbf{w}), \quad \forall j \geq L+1$$

In this max-margin formulation, $\mathcal{L}$ is a loss function such as the scaled hinge loss defined above, R is the (high order) regularizer, and the constraints force $\{\mathbf{y}_j\}_{j=L+1}^{L+U}$ to be predictions of the model for unlabeled data.

R specifies prior constraints about the predictions on unlabeled data. A *high-order* regularizer is one that imposes constraints on sets of output elements rather than independently on each element. One example of a high-order R is the cardinality regularizer, where $R(\mathbf{Y}_U)$ is a function of $\mathbf{1}^\top \mathbf{Y}_U$, and the vector $\mathbf{Y}_U$ is defined as the concatenation of all $\mathbf{y}_j$'s for $j \geq L+1$. For example, in a part-of-speech NLP task, this could refer to the number of words labeled as verbs, while in an image segmentation task it could refer to the number of pixels labeled as foreground. This is useful to encourage the predicted labels to have similar count statistics as the labeled data. As observed in many previous papers, e.g., (Zhu et al., 2003; Wang et al., 2008), enforcing this type of constraint is important for imbalanced datasets. In Section 3.3, we describe a graph based regularizer R and its combination with cardinality regularizers. A variety of other high-order regularizers, e.g., (Vicente et al., 2008; Kohli et al., 2009; Tarlow et al., 2010; Chang et al., 2007; Carlson et al., 2010), have been defined in various structured output settings.

Minimizing the objective in Eq. 3 is difficult due to the hard constraints that make R a complicated and possibly non-continuous function of $\mathbf{w}$. To solve this difficulty, we utilize some relaxations of the hard constraints.

We observe that these constraints are equivalent to the following when the maximum is unique,

$$f(\mathbf{x}_j, \mathbf{y}_j, \mathbf{w}) = \max_{\mathbf{y}} f(\mathbf{x}_j, \mathbf{y}, \mathbf{w}), \quad \forall j \geq L+1. \quad (4)$$

Since we have $\max_{\mathbf{y}} f(\mathbf{x}_j, \mathbf{y}, \mathbf{w}) \geq f(\mathbf{x}_j, \mathbf{y}_j, \mathbf{w})$ for all $\mathbf{y}_j$, the amount of constraint violation can be measured by the difference $\max_{\mathbf{y}} f(\mathbf{x}_j, \mathbf{y}, \mathbf{w}) - f(\mathbf{x}_j, \mathbf{y}_j, \mathbf{w})$. We therefore replace the constraints by a term in the objective that penalizes constraint violation,

$$\min_{\mathbf{w}, \mathbf{Y}_U} \quad \sum_{i=1}^{L} \mathcal{L}(\mathbf{x}_i, \mathbf{y}_i, \mathbf{w}) + R(\mathbf{Y}_U)$$
$$+ \mu \sum_{j=L+1}^{L+U} \left[\max_{\mathbf{y}} f(\mathbf{x}_j, \mathbf{y}, \mathbf{w}) - f(\mathbf{x}_j, \mathbf{y}_j, \mathbf{w}) \right] \quad (5)$$

where μ measures the tolerance of constraint violation. When $\mu \to +\infty$, this is equivalent to Eq. 3; when $\mu < +\infty$, this becomes a relaxation of Eq. 3, where $\mathbf{Y}_U$ can be different from the predictions made by the model. This relaxation decouples $\mathbf{w}$ from R and makes it possible to optimize the objective by iterating two steps, alternatively fixing $\mathbf{w}$ or $\mathbf{Y}_U$ and optimize over the other, where both steps are easier to solve than Eq. 3:

Step 1. Fix $\mathbf{w}$ and optimize over $\mathbf{Y}_U$. The optimization problem becomes

$$\min_{\mathbf{Y}_U} \quad R(\mathbf{Y}_U) - \mu \sum_{j=L+1}^{L+U} f(\mathbf{x}_j, \mathbf{y}_j, \mathbf{w}) \quad (6)$$

This step infers labels for those unlabeled examples, based on both the current model and the regularizer. This is a MAP inference problem, and the hard part is to handle the high-order regularizer $R(\mathbf{Y}_U)$. A wide range of methods have been developed for computing MAP in models

[1] Here we are ignoring data independent regularizers, e.g., L1 and L2, in this formulation for simplicity, but it is straightforward to incorporate them into the model.

with high-order potentials (Vicente et al., 2008; Kohli et al., 2009; Tarlow et al., 2010; Tarlow & Zemel, 2012). We discuss the approach for our loss-based graph regularizer and cardinality regularizers in more detail in Section 3.3.

Step 2. Fix $\mathbf{Y}_U$ and optimize over $\mathbf{w}$. The optimization problem becomes

$$\min_{\mathbf{w}} \sum_{i=1}^{L} \mathcal{L}(\mathbf{x}_i, \mathbf{y}_i, \mathbf{w}) + \mu \sum_{j=L+1}^{L+U} \left[\max_{\mathbf{y}} f(\mathbf{x}_j, \mathbf{y}, \mathbf{w}) - f(\mathbf{x}_j, \mathbf{y}_j, \mathbf{w}) \right] \tag{7}$$

This step updates the model using both the labeled data and the labels inferred from Step 1 for unlabeled data. Note that the last term is just $\mathcal{L}_h$ in Eq. 1, and this optimization is no harder than optimizing a fully supervised model, which can be solved by methods such as subgradient descent.

Thus our learning algorithm proceeds by iteratively solving the optimization problems in Eq. 6 and Eq. 7.

3.3. Graph-Based SSL for Image Segmentation

In this section we describe an application of the proposed framework to graph-based SSL for binary segmentation, but we note that our method can be easily extended to multi-class segmentation. Graph-based SSL uses a graph so constructed that examples close on this graph should have similar outputs. The model is then regularized by this graph to make predictions that are smooth on it. Here we assume the graph is represented by edge weights s_{ij} which measures the similarity between example i and j, and the two examples are connected only when $s_{ij} > 0$.

Choosing a proper output similarity metric is important for graph-based SSL methods. For classification, most graph-based methods define this similarity as the squared difference of two posterior distributions (Zhu et al., 2003; Zhou et al., 2004). For structured prediction, (Subramanya et al., 2010; He et al., 2013) follow this approach but use marginal distributions over parts of output in the squared difference.

However, structured output problems have a natural similarity metric in the output space, defined by the loss function. For probabilistic models, it is not easy to incorporate the loss function into the similarity metric. But our framework allows the use of loss functions in the regularizer R.

We define the graph regularizer

$$R_G(\mathbf{Y}_U) = \lambda \sum_{i,j:s_{ij}>0} s_{ij} \Delta(\mathbf{y}_i, \mathbf{y}_j) \tag{8}$$

where the sum is over all edges in the graph, connecting both labeled and unlabeled examples, and λ is a weight factor. This regularizer requires $\mathbf{y}_i$ and $\mathbf{y}_j$ to be close when s_{ij} is large.

To use this regularizer into our framework, we need to solve the MAP inference problem in Step 1 of the algorithm:

$$\min_{\mathbf{Y}_U} \lambda \sum_{i,j:s_{ij}>0} s_{ij} \Delta(\mathbf{y}_i, \mathbf{y}_j) - \mu \sum_{j=L+1}^{L+U} f(\mathbf{x}_j, \mathbf{y}, \mathbf{w}). \tag{9}$$

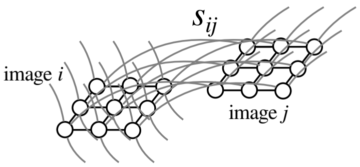


Figure 1. Graph structure with Hamming loss. Black edges represent intra image structure, and grey edges represent graph constraints.

Here each $f(\mathbf{x}_j, \mathbf{y}, \mathbf{w})$ is a sum of unary and pairwise potentials, and the graph regularizer is a high order potential. For decomposable loss functions like Hamming loss, the graph regularizer becomes a sum of submodular pairwise potentials. The MAP inference is then a standard inference problem for pairwise graphical models and can be solved via graph cuts. The structure of this graph is shown in Fig. 1. More complicated loss functions, such as the PASCAL loss, can also be handled using an iterative leave-one-out optimization method described in the supplementary material.

The graph regularizer can also be combined with other types of high order regularizers, for example the cardinality regularizers described earlier. In fact, graphs with submodular pairwise potentials have a known short-boundary bias (Kohli et al., 2013) which favors a small number of cut edges (pairs of pixels that have different labels). This bias can cause some serious problems in SSL when the number of labeled examples is not balanced across classes. In our binary segmentation problem, usually the majority of pixels belong to background and only a small portion belong to foreground. Then when we run the optimization, this bias would make the model predict much more background for the unlabeled images. In the extreme case when unary potentials are weak, all unlabeled pixels will be predicted to have the dominant label. The use of cardinality regularizers is then especially important.

We define a cardinality regularizer

$$R_C(\mathbf{Y}_U) = \gamma\, h(1^\top \mathbf{Y}_U) \tag{10}$$

where γ is a weight parameter and

$$h(x) = \max\{0, |x - x_0| - \delta\}^2 \tag{11}$$

x_0 is the expected number of foreground pixels computed according to the number of total pixels and the proportion of foreground in labeled images, and δ is the deviation from x_0 that can be tolerated without paying a cost. We use $\delta = x_0/5$ throughout all our experiments.

Then the optimization problem in Step 1 becomes

$$\min_{\mathbf{Y}_U} \lambda \sum_{i,j:s_{ij}>0} s_{ij} \Delta(\mathbf{y}_i, \mathbf{y}_j) + \gamma\, h(1^\top \mathbf{Y}_U) - \mu \sum_{j=L+1}^{L+U} f(\mathbf{x}_i, \mathbf{y}_j, \mathbf{w}) \tag{12}$$

Finding the optimum of this problem is in general not easy. However, finding the optimum for both a submodular pairwise MRF and a cardinality potential plus unary potentials can be done very efficiently. We therefore decompose the objective into two parts and use dual-decomposition (Sontag et al., 2011) for the optimization. Details about this can be found in the supplementary material.

4. Connection to Posterior Regularization

There is a surprising connection between the proposed framework and the PR based SSL method described in (He et al., 2013). We show in this section that for some general settings the two methods are optimizing special cases of the same objective. The key results are: under a zero temparature limit, (1) the KL-divergence term in PR (see below) becomes the constraint violation penalty in our framework (Eq. 5), and (2) the posterior distribution becomes the (hard) model prediction.

The idea of PR is to regularize the posterior distributions so that they are consistent with some prior knowledge. For graph-based SSL the prior knowledge is the smoothness of the posterior distribution over the graph. PR optimizes the following objective

$$\min_{\mathbf{w}} \sum_{i=1}^{L} \mathcal{L}(\mathbf{x}_i, \mathbf{y}_i, \mathbf{w}) + \lambda R\left(\{p_{\mathbf{w}}(\mathbf{y}|\mathbf{x}_j)\}_{j=L+1}^{L+U}\right) \quad (13)$$

where $\mathcal{L}(\mathbf{x}_i, \mathbf{y}_i, \mathbf{w}) = -\log p_{\mathbf{w}}(\mathbf{y}_i|\mathbf{x}_i)$ is the negative conditional log likelihood for labeled data, and R is the posterior regularizer.

In PR, auxiliary distributions $\{q_j(\mathbf{y})\}_{j=L+1}^{L+U}$ are introduced to make the optimization easier, and the following objective is used instead:

$$\begin{aligned}\min_{\mathbf{w},q} \quad & \sum_{i=1}^{L} \mathcal{L}(\mathbf{x}_i, \mathbf{y}_i, \mathbf{w}) + \lambda R(q) \\ & + \mu \sum_{j=L+1}^{L+U} \mathrm{KL}(q_j(\mathbf{y}) || p_{\mathbf{w}}(\mathbf{y}|\mathbf{x}_j)). \end{aligned} \quad (14)$$

Optimizing this objective will learn $\mathbf{w}$ and q such that the $p_{\mathbf{w}}$ distribution is consistent with labeled data, the q distribution is smooth on the graph, and the two distributions should also be close to each other in terms of KL-divergence. This objective is then optimized in an alternating approach similar to the method utilized in our model as described above.

To relate this formulation of PR to our proposed method, we introduce a temperature parameter T, and define $p_{\mathbf{w}}(\mathbf{y}|\mathbf{x}, T) = \frac{1}{Z_T^p} \exp\left(\frac{f(\mathbf{x},\mathbf{y},\mathbf{w})}{T}\right)$ and $q(\mathbf{y}, T) = \frac{1}{Z_T^q} \exp\left(\frac{g(\mathbf{y})}{T}\right)$. Here Z_T^p and Z_T^q are normalizing constants, and $g(\mathbf{y})$ is an arbitrary score function. The temparature augmented objective has the form of

$$\begin{aligned}\min_{\mathbf{w},q} \quad & \sum_{i=1}^{L} \mathcal{L}(\mathbf{x}_i, \mathbf{y}_i, \mathbf{w}, T) + \lambda R(q_T) \\ & + \mu \sum_{j=L+1}^{L+U} T\mathrm{KL}(q_j(\mathbf{y}, T) || p_{\mathbf{w}}(\mathbf{y}|\mathbf{x}_j, T)) \end{aligned} \quad (15)$$

where $\mathcal{L}(\mathbf{x}_i, \mathbf{y}_i, \mathbf{w}, T) = -\log p_{\mathbf{w}}(\mathbf{y}_i|\mathbf{x}_i, T)$ and $R(q_T)$ is the regularizer defined on $\{q_j(\mathbf{y}, T)\}_{j=L+1}^{L+U}$. This objective is the same as the PR objective when $T = 1$. Next we show that when $T \to 0$ this becomes the objective of our method in Eq. 5.

Using the definition of p and q, the KL-divergence term can be rewritten as

$$\begin{aligned} & T\mathrm{KL}(q_j(\mathbf{y}, T) || p_{\mathbf{w}}(\mathbf{y}|\mathbf{x}_j, T)) \\ = & \sum_{\mathbf{y}} q_j(\mathbf{y}, T) \left[g_j(\mathbf{y}) - f(\mathbf{x}_j, \mathbf{y}, \mathbf{w}) \right] + T Z_T^p - T Z_T^q \end{aligned} \quad (16)$$

Denote $\mathbf{y}_j = \mathrm{argmax}_{\mathbf{y}}\, q_j(\mathbf{y}, T)$, and let $T \to 0$, then

$$q_j(\mathbf{y}, T) \to \begin{cases} 1, & \mathbf{y} = \mathbf{y}_j \\ 0, & \text{otherwise} \end{cases} \quad (17)$$

and

$$\begin{aligned} T Z_T^p & \to \lim_{T \to 0} T \log \sum_{\mathbf{y}} \exp\left(\frac{f(\mathbf{x}_j, \mathbf{y}, \mathbf{w})}{T}\right) \\ & = \max_{\mathbf{y}} f(\mathbf{x}_j, \mathbf{y}, \mathbf{w}) \end{aligned} \quad (18)$$

$$T Z_T^q \to \lim_{T \to 0} T \log \sum_{\mathbf{y}} \exp\left(\frac{g_j(\mathbf{y})}{T}\right) = g_j(\mathbf{y}_j) \quad (19)$$

Substituting the above equations into Eq. 16,

$$\begin{aligned} & T\mathrm{KL}(q_j(\mathbf{y}, T) || p_{\mathbf{w}}(\mathbf{y}|\mathbf{x}_j, T)) \\ & \to \max_{\mathbf{y}} f(\mathbf{x}_j, \mathbf{y}, \mathbf{w}) - f(\mathbf{x}_j, \mathbf{y}_j, \mathbf{w}) \end{aligned} \quad (20)$$

as $T \to 0$. This is identical to the constraint violation penalty in Eq. 5.

The relation between the regularizer terms depends on the specific regularizers used in the model. For example, R can be defined as $\sum_{i,j:s_{ij}>0} s_{ij} \sum_c (p_{\mathbf{w}}(y_{ic} = 1|\mathbf{x}_i) - p_{\mathbf{w}}(y_{jc} = 1|\mathbf{x}_j))^2$, where c indexes pixels, as in (He et al., 2013). Here $p_{\mathbf{w}}(y_{ic} = 1|\mathbf{x}_i) = 1$ for labeled foreground pixels and $p_{\mathbf{w}}(y_{ic} = 1|\mathbf{x}_i) = 0$ for labeled background pixels, to only regularize the posterior distributions for the unlabeled data.

For the regularizer term in this case, according to Eq. 17, for binary segmentation we have $q_j(y_c = 1, T) \to y_{jc}$ as $T \to 0$ for each pixel c. Therefore

$$\begin{aligned} R(q_T) & \to \sum_{i,j:s_{ij}>0} s_{ij} \sum_c (y_{ic} - y_{jc})^2 \\ & = \sum_{i,j:s_{ij}>0} s_{ij} \Delta(\mathbf{y}_i, \mathbf{y}_j) \end{aligned} \quad (21)$$

where $\Delta(\mathbf{y}_i, \mathbf{y}_j)$ is the Hamming loss.

Finally, for $\mathcal{L}(\mathbf{x}_i, \mathbf{y}_i, \mathbf{w}, T)$ term, it is known, e.g., in (Hazan & Urtasun, 2010), that as $T \to 0$ this term converges to the structured hinge loss.[2]

Remark. Hazan & Urtasun (2010) proposed a framework that unifies the max-margin and probabilistic methods for structured prediction. Our result here can be thought of as an extension of this to semi-supervised learning of structured output problems. Moving to the max-margin formulation loses the uncertainty representation of the probabilistic models, but has the ability to specify high order constraints directly on model predictions and to use powerful discrete optimization algorithms, therefore overcoming some difficulties of inference in loopy probabilistic models. In addition, our generalized formulation also opens up the possibility of probabilistic models using temperatures other than 1, which can have some desirable properties, e.g., when T is close to 0 the posterior distribution will be much more concentrated.

5. Experiments

5.1. Datasets and Model Details

We explore the efficacy of the proposed framework on two semi-supervised foreground-background segmentation tasks. For the first task, we use the Weizmann Horse dataset (Borenstein & Ullman, 2002), a fully-labeled set of 328 images. For the unlabeled Horse dataset, we used images labeled “horse” in CIFAR-10 (Krizhevsky & Hinton, 2009), which are not segmented. For the second task, we constructed a labeled set of 214 “bird” images from the PASCAL VOC 2011 segmentation data (Everingham et al., 2010). The unlabeled Bird images come from the Caltech-UCSD Bird (CUB) dataset (Welinder et al., 2010). Note that this setting of SSL is especially challenging as the unlabeled data comes from a different source than the labeled data; utilizing unlabeled examples that are extremely different than the labeled ones will hamper the performance of an SSL learning algorithm. For the unlabeled sets we therefore selected images that were similar to at least one image in the labeled set, resulting in 500 unlabeled Horse images from CIFAR-10, and 600 unlabeled Bird images from CUB. For all the images in both tasks, and their corresponding segmentations, we resize them to 32×32, which is also the size of all CIFAR-10 images.

The Bird images contain considerably more variation than the Horse images, as the birds are in a diverse set of poses and are often occluded. We found that utilizing the PASCAL birds alone for training, validation and test did not leave enough training examples to attain reasonable segmentation performance. We thus created an additional labeled set of 600 bird images using the CUB dataset (a different set of 600 images than the aforementioned unlabeled set). Details on how we generated segmentations for these images are in the supplementary material; these generated segmentations are available online.

In our experiments we compare four types of models: (1) the baseline **Initial** model, which forms the basis for each of the others; (2) a **Self-Training** model that iteratively uses the current model to predict labels for unlabeled data and updates itself using these predictions as true labels; (3) **Graph**, our graph-based SSL method that uses the graph regularizer R_G; (4) **Graph-Card**, our SSL method utilizing both graph and cardinality regularizer $R_G + R_C$.

The Initial model is trained in a fully supervised way on only labeled data by subgradient decent on scaled structured hinge loss. The model's score function f is defined as in the example given in Section 3.1. We extracted a 149 dimensional descriptor for each pixel in an image by applying a filter bank. Then a multi-layer neural network is trained using these descriptors as input to predict binary labels[3]. The log probability of each class is used as the unary potential. For pairwise potentials, we used a standard 4-connected grid neighborhood and the common Potts model, where $f^p(\mathbf{x}, y_i, y_j) = -p_{ij}\mathbf{I}[y_i \neq y_j]$ and p_{ij} is a penalty for assigning different labels for neighboring pixels y_i and y_j. We define p_{ij} as the sum of a constant term that encourages smoothing and a local contrast sensitive term defined in (Boykov & Jolly, 2001) which scales down the penalty when the RGB difference between pairs of pixels is large. In our experiments, we fix the pairwise potentials and focus on learning parameters in the neural network for unary potentials only.

During learning, the gradients are back-propagated through the neural network to update parameters. Since neural networks are highly nonlinear models, it is hard to find the optimal $\mathbf{w}$ in Eq. 7 in every Step 2 of our algorithm. Instead, we only take a few gradient steps in Step 2 of each iteration. Other hyper parameters, e.g. λ, μ, γ, are tuned using the validation set, see supplementary material for more details on parameter settings.

For the graph-based models, we used the Histogram of Oriented Gradients (HOG) (Dalal & Triggs, 2005) image features to construct the graph. We set $s_{ij} = 1$ if examples i and j are one of each other's 5 nearest neighbors, and $s_{ij} = 0$ otherwise. Fig. 2 shows some nearest neighbor search results using HOG distance.

5.2. Experimental Settings

For our experiments, we examine how the performance of the SSL methods change with the number of labeled images, by randomly selecting L images from the training set to be used as labeled data and adding the remaining images to the unlabeled set. Starting from $L = 5$, we gradually in-

[2] With a loss term added to the score function f, which can be set to 0 for $T = 1$ case to prove the equivalence.

[3] We also tried a linear model initially, but neural nets significantly outperform linear models by about 10%.

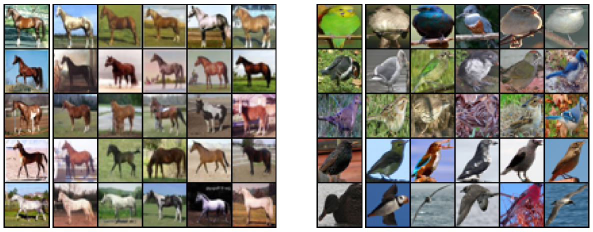


Figure 2. Left most column are query images, and the 5 columns on the right are the nearest neighbors retrieved based on HOG similarity. All query images are randomly chosen. Left: query from Weizmann dataset, retrieve CIFAR-10 horses. Right: query from PASCAL dataset, retrieve CUB birds.

crease L to the entire training set. Note that while we vary the training and unlabeled sets in this way, the validation and test sets remain constant, in order to make comparisons fair. This process is repeated 10 times, each time including randomly selected images in the training set. All models are evaluated using per-pixel prediction accuracy averaged over pixels in all images, and we report the mean and standard deviation of the results over the 10 repetitions.

We ran three types of experiments. In the first one, the training, validation and test set were all drawn from the same dataset. For the Horse task, there were up to 200 training images, 48 validation, and 80 test images, drawn from the Weizmann set, and 500 unlabeled images from CIFAR-10. For the Bird task, there were up to 200 training images, 200 validation, and 200 test images, and 600 unlabeled images, all drawn from the CUB dataset.

The second experiment explored domain adaptation. In many experimental settings, there are insufficient labeled examples to obtain good performance after splitting the dataset into training, validation, and test. This was the case with our PASCAL Bird dataset, which necessitated labeling examples from the CUB set. An interesting question is whether training on one domain, the source domain, can transfer to a different, target domain, when the unlabeled data comes from the target domain, i.e., the same dataset as the test set, and both differ from the training set. It is possible for the model to learn special features about the target domain by using unlabeled data, therefore obtaining larger performance gains. In the second experiment we explored the performance of the various models in a version of this domain adaptation setting on the Bird segmentation task.

The third experiment directly assesses the impact of drawing the validation set from the same dataset as the source, versus drawing the validation from the target domain. In our original bird experiment the validation set comes from the source domain, while in the second experiment it comes from the target domain; tuning hyperparameters on the target domain may contribute to some of the performance gains. To examine this, we compared the models in two more settings, both of which use a training set of 40 images drawn from the PASCAL dataset, and the same 200 CUB test images and 600 unlabeled CUB images. The experiments differ in that in the first setup the validation set is composed of 174 images drawn from the source domain, the PASCAL set, while in the second they are from the target CUB domain. Table 1 lists the datasets used in each experimental setting.

Experiment	train	validation	test	unlabeled
(1) Horse	W-200$^-$	W-48	W-80	R-500$^+$
(1) Bird	C-200$^-$	C-200	C-200	C-600$^+$
(2) Domain Adapt.	P-214$^-$	C-200	C-200	C-600$^+$
(3) Val: Source	P-40$^-$	P-174	C-200	C-600$^+$
(3) Val: Target	P-40$^-$	C-174	C-200	C-600$^+$

Table 1. Experimental settings and datasets. Each dataset description follows the format [dataset code]-[size]. Dataset codes: P for PASCAL VOC birds, C for CUB birds, W for Weizmann horses, R for CIFAR-10 horses. Superscript "-" means at most, and "+" means at least, see paper for more details.

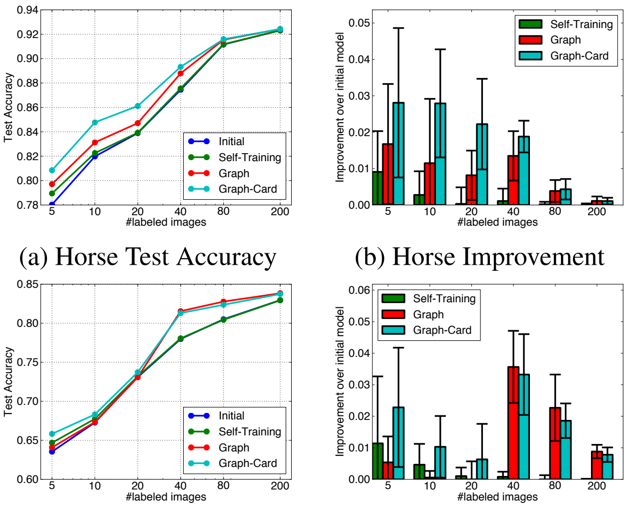
5.3. Results

Experiment 1. Results for the first basic SSL experiments are shown in Fig. 3; (a),(c) show how test set performance changes as the number of labeled images increases, while Fig. 3(b),(d) show the improvement from SSL using the three methods compared to the initial model more directly.

As can be seen, for both segmentation task self-training achieves a small improvement with very few labeled examples, but does not help too much in general, as it is mostly reinforcing the model itself. Graph-based methods work significantly better than self-training throughout. For Horse segmentation, the use of unlabeled data helps the most when the number of labeled images are small. The improvement becomes smaller as the number of images increases. The model saturates and achieves very high accuracy (more than 92%) with 200 labeled images, where using unlabeled data does not make too much difference.

For Bird segmentation, graph-based methods achieve a small improvement over self-training and the initial model when the number of labeled images is small ($L \leq 20$). This can be explained by the complexity of the bird dataset; more examples are required to achieve reasonable segmentations. There is a jump in performance from $L = 20$ to $L = 40$: as the initial model gets better, combining with the graph, inferred labels for unlabeled data become much better and therefore more helpful. From Fig. 3 we can see that when $L = 40$, using graph-based methods the test accuracy nearly matches that of a fully supervised model trained with all 200 labeled images, thus saving a lot of labeling work.

Comparing "Graph-Card" and "Graph", we can see that using a cardinality regularizer further improves performance over only using the graph regularizer, as in most horse seg-



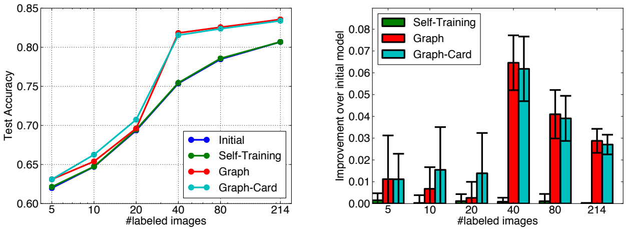
(a) Horse Test Accuracy

(b) Horse Improvement

(c) Bird Test Accuracy

(d) Bird Improvement

Figure 3. Experiment 1 (a),(c): Test performance for the initial model and the 3 SSL methods; (b),(d): improvements for the three methods over the initial model.



(a) Acc vs. #labeled

(b) Improvement

Figure 4. Experiment 2: Results for the domain adaptation Bird task, where the unlabeled and validation and test sets are from a different dataset than the training set. The curve for “Initial” is behind “Self-Training”.

mentation cases and bird segmentation with few labeled images. It is most helpful when the number of images are small, where the initial model is very weak and the short-boundary bias becomes especially significant when inferring labels for unlabeled images. For a lot of cases, the use of a cardinality potential can compensate for this bias.

Experiment 2. Fig. 4 shows the results for the domain adaptation setting, where the training data is from one dataset while the unlabeled data and the test and validation examples come from a different set. Compared to the original bird experiment, we observe that: (1) the performance jump from $L = 20$ to $L = 40$ is considerably larger; (2) the gap between SSL methods and the initial model is also more significant; and (3) the improvement from self-training is almost non-existent.

Experiment 3. We compare the “Graph-Card” method across the two settings, where the validation set is either

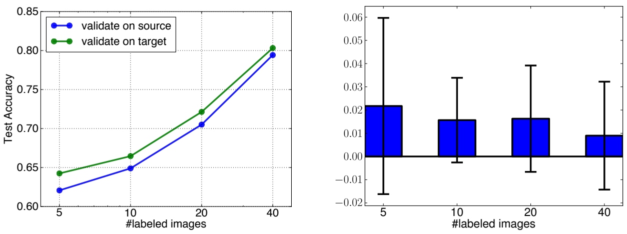


Figure 5. Experiment 3: Comparison between validation on source domain and validation on target domain. Left: test accuracy as number of labeled images increases. Right: difference between the two settings (validate on target vs. validate on source).

from the source or the target domain. Fig. 5 summarizes the results. In this comparison, the model validated on the target domain performs consistently better than the model validated on the source domain. However, the difference decreases as the number of labeled images increases, as in both settings the method is getting closer to the limit, which can be seen from other experiments on bird segmentation, where the performance levels off when $L \geq 40$.

6. Conclusion and Future Work

In this paper, we proposed a new framework for semi-supervised structured output learning that allows the use of expressive high order regularizers defined directly on model predictions for unlabeled data. We proved that this framework and PR are closely related. Experimental results on image segmentation tasks demonstrated the effectiveness of our framework, and its ability to strongly benefit from unlabeled data in a domain adaptation setting.

Looking forward, we are exploring the learning of the input similarity metric s_{ij} in our graph-based SSL example, and also incorporating other types of high order regularizers. Developing more efficient inference algorithms for these high order regularizers is important for the success of the method. On the application side, our segmentation tasks are especially relevant when combined with an object detector. SSL for a structured prediction model that performs segmentation and detection jointly is an interesting and challenging future direction.

Acknowledgments

We thank Charlie Tang and Danny Tarlow for helpful discussions.

References

Altun, Yasemin, McAllester, David, and Belkin, Mikhail. Maximum margin semi-supervised learning for structured variables. In *NIPS*, 2006.

Belkin, Mikhail, Niyogi, Partha, and Sindhwani, Vikas. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *JMLR*, 2006.

Blum, Avrim and Mitchell, Tom. Combining labeled and unlabeled data with co-training. In *COLT*, 1998.

Borenstein, Eran and Ullman, Shimon. Class-specific, top-down segmentation. In *ECCV*, 2002.

Boykov, Y.Y. and Jolly, M.P. Interactive graph cuts for optimal boundary & region segmentation of objects in nd images. In *ICCV*, 2001.

Brefeld, Ulf and Scheffer, Tobias. Semi-supervised learning for structured output variables. In *ICML*, 2006.

Carlson, Andrew, Betteridge, Justin, Wang, Richard C, Hruschka Jr, Estevam R, and Mitchell, Tom M. Coupled semi-supervised learning for information extraction. In *WSDM*, 2010.

Chang, Ming-Wei, Ratinov, Lev, and Roth, Dan. Guiding semi-supervision with constraint-driven learning. In *ACL*, 2007.

Chapelle, Olivier, Schölkopf, Bernhard, Zien, Alexander, et al. *Semi-supervised learning*. MIT press Cambridge, 2006.

Dalal, Navneet and Triggs, Bill. Histograms of oriented gradients for human detection. In *CVPR*, 2005.

Everingham, M., Gool, L. Van, Williams, C. K. I., Winn, J., and Zisserman, A. The pascal visual object classes (voc) challenge. *IJCV*, 2010.

Ganchev, Kuzman, Graça, Joao, Gillenwater, Jennifer, and Taskar, Ben. Posterior regularization for structured latent variable models. *JMLR*, 2010.

Grandvalet, Yves and Bengio, Yoshua. Semi-supervised learning by entropy minimization. In *NIPS*, 2005.

Hazan, Tamir and Urtasun, Raquel. A primal-dual message-passing algorithm for approximated large scale structured prediction. In *NIPS*, 2010.

He, Luheng, Gillenwater, Jennifer, and Taskar, Ben. Graph-based posterior regularization for semi-supervised structured prediction. In *CoNLL*, 2013.

Joachims, Thorsten. Transductive inference for text classification using support vector machines. In *ICML*, 1999.

Kohli, Pushmeet, Torr, Philip HS, et al. Robust higher order potentials for enforcing label consistency. *IJCV*, 2009.

Kohli, Pushmeet, Osokin, Anton, and Jegelka, Stefanie. A principled deep random field model for image segmentation. In *CVPR*, 2013.

Krizhevsky, Alex and Hinton, Geoffrey. Learning multiple layers of features from tiny images. Technical report, Department of Computer Science, University of Toronto, 2009.

Lafferty, John, McCallum, Andrew, and Pereira, Fernando CN. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML*, 2001.

Lee, Chi-Hoon, Wang, Shaojun, Jiao, Feng, Schuurmans, Dale, and Greiner, Russell. Learning to model spatial dependency: Semi-supervised discriminative random fields. In *NIPS*, 2006.

Mann, Gideon S and McCallum, Andrew. Generalized expectation criteria for semi-supervised learning with weakly labeled data. *JMLR*, 2010.

Nigam, Kamal, McCallum, Andrew, Thrun, Sebastian, and Mitchell, Tom. Learning to classify text from labeled and unlabeled documents. *AAAI*, 1998.

Nowozin, Sebastian and Lampert, Christoph H. *Structured learning and prediction in computer vision*. Now publishers Inc, 2011.

Pan, Sinno Jialin and Yang, Qiang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 2010.

Scudder III, H. Probability of error of some adaptive pattern-recognition machines. *IEEE Transactions on Information Theory*, 1965.

Sontag, David, Globerson, Amir, and Jaakkola, Tommi. Introduction to dual decomposition for inference. *Optimization for Machine Learning*, 2011.

Subramanya, Amarnag, Petrov, Slav, and Pereira, Fernando. Efficient graph-based semi-supervised learning of structured tagging models. In *EMNLP*, 2010.

Tarlow, Daniel and Zemel, Richard S. Structured output learning with high order loss functions. In *AISTATS*, 2012.

Tarlow, Daniel, Givoni, Inmar E, and Zemel, Richard S. Hop-map: Efficient message passing with high order potentials. In *AISTATS*, 2010.

Taskar, Ben, Guestrin, Carlos, and Koller, Daphne. Max-margin markov networks. In *NIPS*, 2004.

Tsochantaridis, Ioannis, Joachims, Thorsten, Hofmann, Thomas, and Altun, Yasemin. Large margin methods for structured and interdependent output variables. In *JMLR*, 2005.

Vezhnevets, Alexander, Ferrari, Vittorio, and Buhmann, Joachim M. Weakly supervised semantic segmentation with a multi-image model. In *ICCV*, 2011.

Vicente, Sara, Kolmogorov, Vladimir, and Rother, Carsten. Graph cut based image segmentation with connectivity priors. In *CVPR*, 2008.

Wang, Jun, Jebara, Tony, and Chang, Shih-Fu. Graph transduction via alternating minimization. In *ICML*, 2008.

Welinder, P., Branson, S., Mita, T., Wah, C., Schroff, F., Belongie, S., and Perona, P. Caltech-UCSD Birds 200. Technical report, California Institute of Technology, 2010.

Zhou, Dengyong, Bousquet, Olivier, Lal, Thomas Navin, Weston, Jason, and Schölkopf, Bernhard. Learning with local and global consistency. In *NIPS*, 2004.

Zhu, Xiaojin. Semi-supervised learning literature survey. Technical report, Department of Computer Science, University of Wisconsin-Madison, 2005.

Zhu, Xiaojin, Ghahramani, Zoubin, Lafferty, John, et al. Semi-supervised learning using gaussian fields and harmonic functions. In *ICML*, 2003.

Zien, Alexander, Brefeld, Ulf, and Scheffer, Tobias. Transductive support vector machines for structured variables. In *ICML*, 2007.

Supplementary Material for Paper "High Order Regularization for Semi-Supervised Learning of Structured Output Problems"

Yujia Li[1] YUJIALI@CS.TORONTO.EDU
Richard Zemel[1,2] ZEMEL@CS.TORONTO.EDU
[1]Department of Computer Science, University of Toronto, Toronto, ON, Canada
[2]Canadian Institute for Advanced Research, Toronto, ON, Canada

In this supplementary material we present:

- Dual decomposition inference for optimizing Equation 12 in the main paper.
- A leave-one-out algorithm for optimizing Equation 9 in the main paper for PASCAL loss.
- Remark on the connection between our method and Constraint Driven Learning (CODL).
- More details about the bird datasets.
- More details about hyper parameter settings.
- More experiment results.

1. Optimizing Equation 12 using Dual Decomposition

Eq.12 is a special case of the following more general optimization problem

$$\min_{\mathbf{y}} f^u(\mathbf{y}) + f^p(\mathbf{y}) + h(1^\top \mathbf{y})$$

where $\mathbf{y} \in \{0,1\}^P$, f^u, f^p and h are unary, pairwise and cardinality potentials respectively. To see this, note that in Eq.12 f is a sum of unary and pairwise potentials and $\Delta(\mathbf{y}_i, \mathbf{y}_j)$ is a sum of pairwise terms. This is hard to optimize due to the interaction between the pairwise potential and high order cardinality potential.

In dual decomposition, we decompose the original problem into two subproblems that are more tractable. We define $A(\mathbf{y}) = \mu f^u(\mathbf{y}) + f^p(\mathbf{y})$ and $B(\mathbf{y}) = (1-\mu) f^u(\mathbf{y}) + h(\mathbf{y})$, where μ is a fixed constant, e.g. 0.5, then the original objective is $A(\mathbf{y}) + B(\mathbf{y})$.

For any $\boldsymbol{\lambda} \in \mathbb{R}^P$, we have a lower bound on the original objective,

$$\mathcal{L}(\lambda) = \min_{\mathbf{y}} \left\{ A(\mathbf{y}) + \boldsymbol{\lambda}^\top \mathbf{y} \right\} + \min_{\mathbf{y}} \left\{ B(\mathbf{y}) - \boldsymbol{\lambda}^\top \mathbf{y} \right\}$$

As $\boldsymbol{\lambda}^\top \mathbf{y}$ is just a sum of very simple unary potentials, each of the subproblems here are easy to solve. For the first one, graph cuts can be used to find exact optimum, and for the second one we can use methods described in (Gupta et al., 2007).

We then maximize this lower bound over $\boldsymbol{\lambda}$, to make it as tight as possible and hence approach the optimum of the original problem. We can compute the subgradient of the lower bound with respect to $\boldsymbol{\lambda}$,

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{\lambda}} = \hat{\mathbf{y}}^A - \hat{\mathbf{y}}^B$$

where $\hat{\mathbf{y}}^A$ is the optimal $\mathbf{y}$ for the first subproblem and $\hat{\mathbf{y}}^B$ is the optimal $\mathbf{y}$ for the second subproblem. In our experiments we follow this subgradient to optimize the lower-bound, but a wide range of other optimization techniques can be applied here as well.

Once the optimization terminates, we have to decode the final $\mathbf{y}^*$ as the solution to the two subproblems may not agree. For this we can calculate the original objective for all $\hat{\mathbf{y}}^A$ and $\hat{\mathbf{y}}^B$'s encountered during the optimization and choose one that has the smallest objective value. Other heuristics can be applied here as well.

2. Optimizing Equation 9 for PASCAL Loss

For a single class, let $\mathbf{y} \in \{0,1\}^P$ denote the prediction for each pixel whether it belongs to that class, and let $\mathbf{y}^*$ be the ground truth. The PASCAL loss is defined as

$$\Delta(\mathbf{y}, \mathbf{y}^*) = 1 - \frac{\sum_i \mathbf{I}[y_i = 1 \text{ and } y_i^* = 1]}{\sum_i \mathbf{I}[y_i = 1 \text{ or } y_i^* = 1]}.$$

(Tarlow & Zemel, 2012) describes an efficient method to compute MAP for high order factors such as the PASCAL loss, with the true label fixed. Here we use this as a subroutine in our optimization.

Since the graph term in Eq. 9 is a sum of many high order factors, direct optimization is very hard, and even messages are hard to compute. We therefore use a leave-one-out algorithm instead. This algorithm iterates through all j's one by one. For each j, all $\mathbf{y}_{j'}$ for $j' \neq j$ are fixed and we optimize over $\mathbf{y}_j$ only. For a single j, the corresponding optimization problem has the form of

$$\min_{\mathbf{y}_j} \quad \lambda \sum_{i: s_{ij} > 0} s_{ij} \Delta(\mathbf{y}_i, \mathbf{y}_j) - \mu f(\mathbf{x}_j, \mathbf{y}_j, \mathbf{w})$$

This is a sum of unary and pairwise potentials in f plus a set of PASCAL loss high order potentials.

We can again use dual decomposition to do the optimization. There are two types of subproblems: (1) unary potentials + pairwise potentials, which can be optimized using graph cuts; (2) unary potentials + one PASCAL loss potential, which can be optimized by invoking the optimization subroutine.

This optimization for a single j can also be done by message passing, as messages for the PASCAL loss can be efficiently computed as described in (Tarlow & Zemel, 2012).

If for each j this optimization can decrease the objective, then Step 1 of our proposed algorithm in Section 3.2 will monotonically decrease the objective in Eq.5, therefore our algorithm is still guaranteed to converge.

3. Remark on the Connection between Our Method and Constraint Driven Learning (CODL)

The Constraint Driven Learning (CODL) (Chang et al., 2007) is similar to our algorithm described in Section 3.2 in the main paper, which is also an alternating optimization method, with some notable and important differences:

1. Few people have explored the use of high order regularizers in CODL, and the optimization in Step 1 is usually done by heuristic search rather than using efficient discrete optimization algorithms.

2. In Step 2, CODL uses the inferred labels as true labels while we use them in the constraint relaxation penalty.

3. The CODL learning algorithm does not correspond to the optimization of a unified objective function. As we derive the algorithm from a joint optimization problem, it is possible to develop variants other than the coordinate descent currently used.

4. More Details about the Bird Datasets

We obtained the PASCAL VOC "bird" dataset by first restricting the image based on the bounding box containing the bird, and then labeling all bird pixels as 1 and all other pixels as 0, resulting in 214 bird images with segmentations.

We selected images from the pool of unlabeled images by utilizing the Histogram of Oriented Gradients (HOG)(Dalal & Triggs, 2005) image features as the distance measure and choosing the set closest to the labeled images. We choose 500 images from CIFAR-10 for the horse segmentation task, and 600 images from CUB for the bird segmentation task according to this criterion.

We obtained labels for CUB dataset, from the rough segmentations provided in CUB. The rouph segmentations provides a localization of the object but are not very precise around the boundary. Usually the rouph segmentations will include a significant amount of background in the foreground mask. To refine this, we fixed the pixel labels in the interior of the foreground area and the background area, and try to relabel the boundary pixels. More specifically, we used GrabCut (Rother et al., 2004), by alternating appearance model fitting and segmentation label updates. The fixed foreground and background areas are used to train the initial appearance model. After that, an extra hole filling operation from mathematical morphology is used to post process the results. These segmentations generated in this way capture most of the details of bird silhouettes. Fig. 1 shows some sample images and generated segmentation masks for the CUB dataset.

5. More Details about Hyper Parameter Settings

We formulate μ as μ'/U where U is the number of unlabeled examples. We found that μ' is not very sensitive to different datasets, and we used the same $\mu' = 100$ for all splits of all datasets. More tuning of this parameter for different datasets may result in even better performance.

The parameter γ for cardinality potentials is fixed to 1 for all experiments.

Parameter λ and learning rate, momentum, etc. for the neural networks are tuned using the validation set. We found that λ is more sensitive to datasets than μ' and γ, but a wide range of λ works quite well.

6. More Experiment Results

Some segmentation results for the intial model, and the models trained with self-training, our graph based method and our graph based method + cardinality regularizer are shown in Fig. 2. These examples are randomly chosen with models trained with 40 labeled images on one split of the corresponding datasets.

The effect of using cardinality regularizers is most obvious

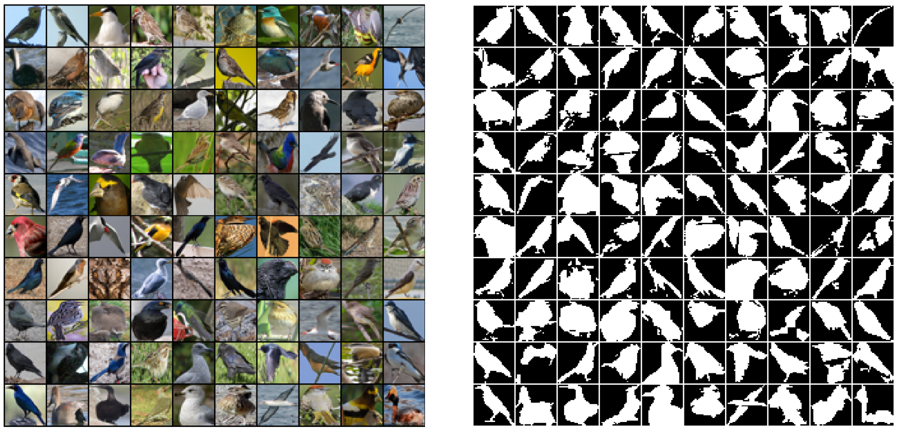


Figure 1. Samples from the CUB dataset and generated segmentations.

from the horse segmentation results. On the bird dataset, there is a significant difference between methods that use graph sturcture ("Graph" and "Graph-Card") and those do not ("Initial" and "Self-Train").

References

Chang, Ming-Wei, Ratinov, Lev, and Roth, Dan. Guiding semi-supervision with constraint-driven learning. In *ACL*, 2007.

Dalal, Navneet and Triggs, Bill. Histograms of oriented gradients for human detection. In *CVPR*, 2005.

Gupta, Rahul, Diwan, Ajit A, and Sarawagi, Sunita. Efficient inference with cardinality-based clique potentials. In *Proceedings of the 24th International Conference on Machine Learning*, 2007.

Rother, Carsten, Kolmogorov, Vladimir, and Blake, Andrew. Grabcut: Interactive foreground extraction using iterated graph cuts. *ACM Transactions on Graphics (TOG)*, 2004.

Tarlow, Daniel and Zemel, Richard S. Structured output learning with high order loss functions. In *Proceedings of the 15th Conference on Artificial Intelligence and Statistics*, 2012.

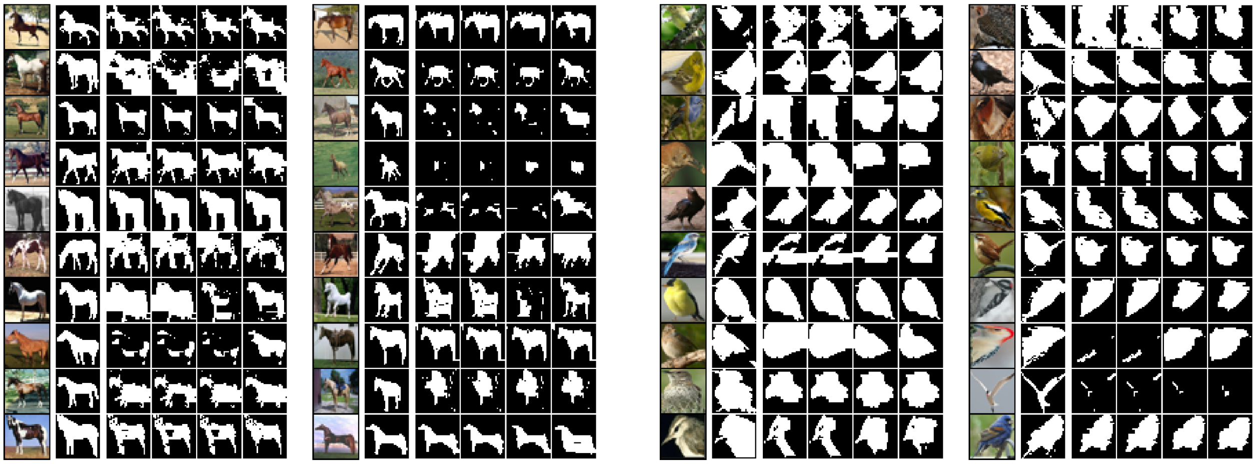


Figure 2. Example segmentation results for horse and bird images. In each row, the first two columns contain the original image and the ground truth segmentation. The next four columns are results obtained using “Initial”, “Self-Training”, “Graph”, “Graph-Card”.