Learning CRFs for Image Parsing with Adaptive Subgradient Descent

Honghui Zhang* Jingdong Wang† Ping Tan‡ Jinglu Wang* Long Quan*
The Hong Kong University of Science and Technology*
Microsoft Research† National University of Singapore‡

Abstract

We propose an adaptive subgradient descent method to efficiently learn the parameters of CRF models for image parsing. To balance the learning efficiency and performance of the learned CRF models, the parameter learning is iteratively carried out by solving a convex optimization problem in each iteration, which integrates a proximal term to preserve the previously learned information and the large margin preference to distinguish bad labeling and the ground truth labeling. A solution of subgradient descent updating form is derived for the convex optimization problem, with an adaptively determined updating step-size. Besides, to deal with partially labeled training data, we propose a new objective constraint modeling both the labeled and unlabeled parts in the partially labeled training data for the parameter learning of CRF models. The superior learning efficiency of the proposed method is verified by the experiment results on two public datasets. We also demonstrate the powerfulness of our method for handling partially labeled training data.

1. Introduction

The Conditional Random Field [19] (CRF) offers a powerful probabilistic formulation for image parsing problems. It has been demonstrated in previous works [18, 11, 16] that integration of different types of cues in a CRF model can significantly improve the parsing accuracy, like the smoothness preference and global consistency. However, how to properly combine multiple types of information in a CRF model to achieve excellent parsing performance still remains an open question. For this reason, the parameter learning of CRF models for image parsing tasks has received increasing attention recently.

Considerable progress on the parameter learning of CRF models has been made in the past few years. However, the parameter learning of CRF models for the image parsing tasks still remains a challenging problem for several reasons. First, as the CRF models used in many image parsing problems are of large scale and include expressive interaction, the computational challenges make the parameter learning of CRF models difficult. Given a large number of training images, the learning efficiency would become a critical issue. Second, partially labeled training data could cause the failure of some learning methods, which is common in image parsing. For example, it has been found that the learned parameters involved in the pairwise smoothness potential are forced to tend toward zeros when using partially labeled training data [25].

In this paper, we propose an adaptive subgradient descent method that iteratively learns the parameters of CRF models for image parsing. The parameter learning is iteratively carried out by solving a convex optimization problem in each iteration. The solution for the convex optimization problem gives a subgradient descent updating form with an adaptively determined updating step-size which can well balance the learning efficiency and performance of the learned CRF models. Meanwhile, to deal with partially labeled training images that are common in various image parsing tasks, a new objective constraint for the parameter learning of CRF models is proposed, which models both the labeled and unlabeled parts of partially labeled training images.

1.1. Related work

The parameter learning of CRF models is an active research topic, and investigated in many previous works [7, 27, 23, 20, 12, 2, 21, 15, 9]. Most current methods for the parameter learning of CRF models can be broadly classified into two categories: maximum likelihood-based methods [19, 17] and max-margin methods [7, 27, 23, 12]. En exhaustive review of the literature is beyond the scope of this paper, and the following review will mainly focus on the max-margin methods in which the parameter learning of CRF models is formulated as a structure learning problem based on the max-margin formulation. Naturally, the max-margin methods for general structure learning can be used for the parameter learning of CRF models, such as the 1-slack and n-slack StructSVM(structural SVM) [27, 12], M3N(max-margin markov network) [7] and Projected Subgradient [23]. The 1-slack StructSVM [12] method is an im-

proved version of the n-slack StructSVM, shown to be substantially faster than the n-slack StructSVM and the Structured SMO method proposed in the M3N [7]. The subgradient method [23] is another popular solution for structure learning problems, which is usually efficient and easy to implement.

Based on the subgradient method, recent works on the parameter learning of CRF models [15, 24] adopt different decomposition techniques. In [15], a dual decomposition approach for the random field optimization is combined with the max-margin formulation for the parameter learning of random field models, which reduces the training of a complex high order MRF (Markov Random Field) to the parallel training of a series of simple slave MRFs that are much easier to handle. In [24], a decomposed learning method which performs efficient learning by restricting the inference step to a limited part of the structured output spaces is proposed. As the updating step-sizes for the subgradient descent in these methods [23, 15, 24] are predefined and oblivious to the characteristics of the data being observed, to balance the learning efficiency and performance of the learned models, the updating step-sizes need to be carefully chosen. Inappropriate updating step-sizes could lead to bad performance of the learned CRF models or slow convergence. This motivates us to improve the subgradient method for the parameter learning of CRF models by adaptively tuning the subgradient descent, termed as *adaptive subgradient descent* in this paper. The Polyak stepsize [22] is a possible solution, if the optimal value of the objective function for the optimization problem is known or can be estimated. However, for the parameter learning problem of CRF models, the optimal value of the objective function is unknown , and how to estimate it is also unclear.

Another important but less discussed issue in the previous max-margin methods for the parameter learning of CRF models is related to partially labeled training data, which is common in image parsing tasks. To deal with the partially labeled training data, a maximum likelihood-based method which approximates the partition function with the Bethe free energy function is proposed in [28], with some limitations of the Bethe approximation discussed in [10]. It has been observed that different treatments of the partially labeled data could lead to quite different performance. To deal with the partially labeled training data, we introduce latent variables in the CRF models, inspired by the work [29, 15].

2. Learning CRF to Parse Images

Random field models are widely used to formulate various image parsing problems. These models are defined by an undirected graph $\mathcal{G} = \langle \mathcal{V}, \mathcal{E} \rangle$, with $\mathcal{V}$ and $\mathcal{E}$ denoting nodes and edges in the graph. One discrete random variable is associated with each node, which may take a value from a set of labels $\mathcal{L} = \{l_1, l_2, \cdots, l_L\}$. Given observation $\mathbf{x}$, the joint conditional distribution of the label assignment $\mathbf{y}$ and $\mathbf{x}$, $P(\mathbf{y}|\mathbf{x})$ can be expressed as:

$$P(\mathbf{y}|\mathbf{x}) = \frac{1}{Z} \exp(-\sum_{c \in \mathcal{C}} \Psi_c(\mathbf{x}_c, \mathbf{y}_c)) \tag{1}$$

The graph $\mathcal{G}$ consists of a set of cliques $\mathcal{C}$, and each clique $c \in \mathcal{C}$ is associated with a label assignment $\mathbf{y}_c$ which is a subset of $\mathbf{y}$. $Z = \sum_{\mathbf{y}'} \exp(-\sum_{c \in \mathcal{C}} \Psi_c(\mathbf{x}_c, \mathbf{y}'_c))$ is the partition function, a normalization term. The label prediction is usually obtained by solving the following MAP (max a posterior) inference problem:

$$\mathbf{y}^* = \arg\max_{\mathbf{y} \in \mathcal{L}} \log P(\mathbf{y}|\mathbf{x}) = \arg\min_{\mathbf{y} \in \mathcal{L}} E(\mathbf{x}, \mathbf{y}) \tag{2}$$

with the energy function $E(\mathbf{x}, \mathbf{y}) = \sum_{c \in \mathcal{C}} \Psi_c(\mathbf{x}_c, \mathbf{y}_c)$.

2.1. Max-margin formulation

First, we briefly review a well studied simple model to organize the interaction between the parameters and features in the energy function: the linear model that assumes the potentials can be expressed as the following linear form:

$$\Psi_c(\mathbf{x}_c, \mathbf{y}_c) = \mathbf{w}_c \cdot f(\mathbf{x}_c, \mathbf{y}_c) \tag{3}$$

where $\mathbf{w}_c$ is the parameters, and $f(\mathbf{x}_c, \mathbf{y}_c)$ is the feature vector for a clique c. This linear model has been widely used in the structure learning methods, like the StructSVM [27, 12] and Projected Subgradient [23]. Based on this linear model, the parameter learning of CRF models can be cast as a typical structure learning problem, with the corresponding energy functions for the CRF models expressed as:

$$E(\mathbf{x}, \mathbf{y}) = \mathbf{w} \cdot \Phi(\mathbf{x}, \mathbf{y}) = \sum_{c \in \mathcal{C}} \mathbf{w}_c \cdot f(\mathbf{x}_c, \mathbf{y}_c) \tag{4}$$

In the following, we review the widely used max-margin formulation for the parameter learning of CRF models:

1-slack and n-slack StructSVM Given a training set $\{(\mathbf{x}_n, \mathbf{y}_n)\}_{n=1}^N$, using the n-slack StructSVM [27], the learning problem can be formulated as [26]:

$$\arg\min_{\mathbf{w}} \frac{1}{2}\|\mathbf{w}\|^2 + \lambda \sum_{n=1}^{N} \xi_n \text{ ,s.t. } \forall n = 1, 2, \cdots, N, \forall \hat{\mathbf{y}}_\mathbf{n} \tag{5}$$
$$E(\mathbf{x}_n, \mathbf{y}_n) - E(\mathbf{x}_n, \hat{\mathbf{y}}_n) + \Delta(\mathbf{y}_n, \hat{\mathbf{y}}_n) \leq \xi_n, \xi_n \geq 0$$

$\mathbf{x}_n$ is the observation, $\mathbf{y}_n, \hat{\mathbf{y}}_n$ are the ground truth label and predicted label, $\Delta(\mathbf{y}_n, \hat{\mathbf{y}}_n)$ is the loss function. Similarly, using the 1-slack reformulation proposed in the 1-slack StructSVM [12], the learning problem can be formulated as:

$$\arg\min_{\mathbf{w}} \frac{1}{2}\|\mathbf{w}\|^2 + \lambda\xi \tag{6}$$
$$\text{s.t. } \forall \hat{\mathbf{y}}, \mathcal{H}(\mathbf{w}; \mathbf{x}^*, \mathbf{y}^*, \hat{\mathbf{y}}) \leq \xi, \xi \geq 0$$

where $\mathcal{H}(\mathbf{w};\mathbf{x}^*,\mathbf{y}^*,\hat{\mathbf{y}})$

$$= \mathbf{w}\cdot[\Phi(\mathbf{x}^*,\mathbf{y}^*)-\Phi(\mathbf{x}^*,\hat{\mathbf{y}})]+\Delta(\mathbf{y}^*,\hat{\mathbf{y}}) \quad (7)$$

$$= \sum_{n=1}^{N}\mathbf{w}\cdot[\Phi(\mathbf{x}_n,\mathbf{y}_n)-\Phi(\mathbf{x}_n,\hat{\mathbf{y}}_n)]+\sum_{n=1}^{N}\Delta(\mathbf{y}_n,\hat{\mathbf{y}}_n)$$

The objective constraint in the 1-slack StructSVM is obtained by merging the objective constraints for each sample of the training set $\{(\mathbf{x}_n,\mathbf{y}_n)\}_{n=1}^N$ in the n-slack StructSVM, with $\mathbf{x}^* = \cup\mathbf{x}_n$, $\mathbf{y}^* = \cup\mathbf{y}_n$ and $\hat{\mathbf{y}} = \cup\hat{\mathbf{y}}_n$. The 1-slack and n-slack StructSVM methods iteratively update the parameters to be learned by the cutting plane algorithm. In each iteration, they need to solve a a quadratic program (QP) problem, and the size of constraints in the QP problem linearly increases with the number of iterations.

Unconstrained max-margin formulation An unconstrained formulation of (5) is adopted in [23], which uses the projected subgradient method to minimize the following regularized objective function:

$$\rho(\mathbf{w}) = \frac{1}{2}\|\mathbf{w}\|^2+\lambda R(\mathbf{w}) \quad (8)$$

$$R(\mathbf{w}) = \max_{\hat{\mathbf{y}}}\mathcal{H}(\mathbf{w};\mathbf{x}^*,\mathbf{y}^*,\hat{\mathbf{y}}) \quad (9)$$

where $R(\mathbf{w})$ is the empirical risk. The parameters to be learned are iteratively updated by:

$$\mathbf{w}_{t+1} = \mathbf{P}[\mathbf{w}_t-\alpha_t\mathbf{g}_{\mathbf{w}}] \quad (10)$$

where $\mathbf{g}_{\mathbf{w}}$ is the subgradient of the convex function (8), $\mathbf{P}$ is the projection operator and α_t is the predefined step-size that needs to be chosen carefully. Inappropriate updating step-size could lead to bad performance of the learned CRF models or slow convergence.

3. Adaptive Subgradient Descent Learning

In this section, we propose an adaptive subgradient descent algorithm for the parameter learning of CRF models, as described in the algorithm 1. It is motivated by applying the idea proposed in the proximal bundle method [13] that uses proximal functions to control the learning rate to the subgradient methods in which the learning rate is subtly controlled by the predefined step-sizes. In each iteration, the parameter updating is carried out by solving a convex optimization problem which integrates a proximal term to preserve the previously learned information and the large margin preference to distinguish bad labeling and the ground truth labeling. The solution for the convex optimization problem gives a subgradient descent update form with an adaptively determined updating step-size for the parameter learning, which well balances the learning efficiency and performance of the learned CRF models. A typical training process of using the proposed algorithm to train CRF models for image parsing is shown in Figure 1.

3.1. Adaptive subgradient descent algorithm

In each iteration of the algorithm 1, the adaptive subgradient descent updating is carried out by solving the following convex optimization problem which has a subgradient-based solution with an adaptively determined step-size:

$$\mathbf{w}_{t+1} = \arg\min_{\mathbf{w}}\frac{1}{2}\|\mathbf{w}-\mathbf{w}_t\|^2+C\xi \quad (11)$$
$$\text{s.t. } \xi\geq 0\,,\ \mathcal{H}(\mathbf{w};\mathbf{x}^*,\mathbf{y}^*,\hat{\mathbf{y}}_t)\leq\xi \text{ and } \mathbf{w}\succeq\mathbf{0}$$

On the one hand, the updated $\mathbf{w}_{t+1}$ is expected to distinguish bad labeling and the ground truth labeling $\mathbf{y}^*$ with a sufficiently large margin and thus progress is made. Therefore, an objective constraint same as that in the 1-slack StructSVM is used in the optimization problem (11):

$$\mathcal{H}(\mathbf{w};\mathbf{x}^*,\mathbf{y}^*,\hat{\mathbf{y}}_t)\leq\xi \quad (12)$$

$\hat{\mathbf{y}}_t$ is the merged labeling configuration that most violates the constraint (12) for the current parameter $\mathbf{w}_t$. On the other hand, a proximal term that forces the learned parameter $\mathbf{w}_{t+1}$ to stay as close as possible to $\mathbf{w}_t$ is inserted into the objective function of (11), so that the previous learned information can be preserved.

Algorithm 1 Adaptive Subgradient Descent Algorithm

1: Input: training set $\{(\mathbf{x}_i,\mathbf{y}_i)\}_{i=1}^N$
2: $\mathbf{x}^* = \cup\mathbf{x}_i$, $\mathbf{y}^* = \cup\mathbf{y}_i$, initialize $\mathbf{w} = \mathbf{w}_1$
3: **for all** $t = 1, 2, \cdots, M$ **do**
4: $\quad\hat{\mathbf{y}}_t = \arg\max_{\hat{\mathbf{y}}\in\mathcal{L}}\mathcal{H}(\mathbf{w}_t;\mathbf{x}^*,\mathbf{y}^*,\hat{\mathbf{y}})$
5: $\quad C = \kappa/\sqrt{t}$
6: $\quad\mathbf{w}_{t+1} = \arg\min_{\mathbf{w}}\frac{1}{2}\|\mathbf{w}-\mathbf{w}_t\|^2+C\xi$
$\quad\text{s.t. } \xi\geq 0\,,\ \mathcal{H}(\mathbf{w};\mathbf{x}^*,\mathbf{y}^*,\hat{\mathbf{y}}_t)\leq\xi \text{ and } \mathbf{w}\succeq\mathbf{0}$
7: **end for**
8: Output: $\mathbf{w}_f = \arg\min_{\mathbf{w}_t\in\{\mathbf{w}_t\}_{t=1}^M}\mathcal{H}(\mathbf{w}_t;\mathbf{x}^*,\mathbf{y}^*,\hat{\mathbf{y}}_t)$

In addition to the objective constraint (12), we add one more constraint that the parameters to be learned are non-negative, similar to the previous work [26]. The parameters in the CRF models are required to be non-negative in many CRF-based image parsing methods, so that different potentials in the energy function can be weighted properly by the parameters, and some efficient MAP inference algorithms on CRF models can be applied, like the widely used graph cut algorithm [5]. For the inference in the fourth step of the algorithm 1, we use the α-expansion algorithm [5]. To assure the α-expansion algorithm can be applied to find the most violated labeling configuration in the fourth step, we adopt the Hamming loss for the loss function $\Delta(\mathbf{y}^*,\hat{\mathbf{y}})$ involved in (12) as [26] did. Next, we solve the optimization problem (11) by using standard tools from convex analysis [4].

3.1.1 Subgradient-based solution

Let $[d_1, d_2, \cdots, d_K] = \Phi(\mathbf{x}, \mathbf{y}^*) - \Phi(\mathbf{x}, \hat{\mathbf{y}}_t)$, $\mathbf{w}_t = [w_1^t, w_2^t, \cdots, w_K^t]$, where K is the number of parameters to be learned in the CRF models. We also assume that the entries of $[d_1, d_2, \cdots, d_K]$ are sorted with the ascending order of $\{w_i^t/d_i, i = 1, 2, \cdots, K\}$. Then, we have:

Theorem 3.1 *The subgradient-based solution for the optimization problem (11) is:*

$$w_i^{t+1} = \begin{cases} w_i^t - \alpha_t d_i & \text{if } w_i^t - \alpha_t d_i \geq 0; \\ 0 & \text{otherwise} \end{cases} \quad (13)$$

$$\alpha_t = \max_{\alpha} \mathcal{L}(\alpha), 0 \leq \alpha \leq C \quad (14)$$

where $[d_1, d_2, \cdots, d_K]$ is the subgradient of the empirical risk (9). The optimization problem (14) is a Lagrangian dual problem of the optimization problem (11), where

$$\mathcal{L}(\alpha) = -\frac{1}{2}\sum_{i=1}^{n}(\alpha d_i - w_i^t)^2 - \frac{1}{2}\mathcal{F}(\alpha) + \alpha\Delta(\mathbf{y}^*, \hat{\mathbf{y}}_t) \quad (15)$$

$$\mathcal{F}(\alpha) = \begin{cases} \sum_{i=n+1}^{K}(\alpha d_i - w_i^t)^2 & \alpha \in [0, \frac{w_{n+1}^t}{d_{n+1}}]; \\ \sum_{i=n+2}^{K}(\alpha d_i - w_i^t)^2 & \alpha \in [\frac{w_{n+1}^t}{d_{n+1}}, \frac{w_{n+2}^t}{d_{n+2}}]; \\ \cdots & \cdots \\ \sum_{i=n+j}^{K}(\alpha d_i - w_i^t)^2 & \alpha \in [\frac{w_{n+j}^t}{d_{n+j}}, C]; \end{cases} \quad (16)$$

Different from the Projected Subgradient method [23] that uses predefined updating step-sizes, the updating step-size α_t in our algorithm is adaptively determined by solving the optimization problem (14), which can well balance the learning efficiency and performance of the learned CRF models. For the limit of space, the detailed derivation and proof is presented in Appendix A of the supplementary material [1].

Next, we briefly analyze how to solve the optimization problem (14). As $\mathcal{L}(\alpha)$ is a piecewise quadratic function of α, in the kth piecewise definition domain of $\mathcal{L}(\alpha)$, $[\alpha_s, \alpha_e]$, the maximum value of $\mathcal{L}(\alpha)$ can be computed as:

$$\mathcal{L}_k^{\max} = \begin{cases} \mathcal{L}(\alpha^*) & \alpha^* \in [\alpha_s, \alpha_e]; \\ \max\{\mathcal{L}(\alpha_s), \mathcal{L}(\alpha_e)\} & \text{otherwise}; \end{cases} \quad (17)$$

Setting the partial derivatives of $\mathcal{L}(\alpha)$ with respect to α to zero gives α^*:

$$\alpha^* = \frac{\sum_{i=1}^{n} w_i^t d_i^t + \sum_{i=n+k}^{K} w_i^t d_i^t + \Delta(\mathbf{y}^*, \hat{\mathbf{y}}_t)}{\sum_{i=1}^{n} d_i^2 + \sum_{i=n+k}^{K} d_i^2} \quad (18)$$

The adaptive step-size α_t is the very one that maximizes $\mathcal{L}(\alpha)$ among all values of α. With the maximum value of $\mathcal{L}(\alpha)$ in each piecewise definition domain, α_t can be efficiently computed by searching the maximum value of $\mathcal{L}(\alpha)$, $\mathcal{L}^{\max} = \max\{\mathcal{L}_k^{\max}\}_{k=1}^{j}$.

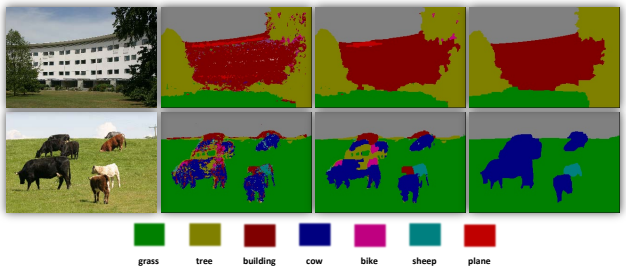


Figure 1. The training process of using the algorithm 1 to train a Robust P^N model [14] for image parsing. The first column shows the input training image; The second column is the unary classification result; The third column is the output of the Robust P^N model with the learned parameters after the first iteration; The fourth column is the output of the Robust P^N model with the final learned parameters which is obtained at the 5th iteration. These output are obtained in the fourth step of algorithm 1.

As C is an upper bound of α, to assure that appropriate progress is made in each iteration, C is initialized with a large value κ (κ= 1 in our implementation), and iteratively decreases to $\kappa/\sqrt{t}$, as stated in the fifth step of the algorithm 1. Meanwhile, to avoid the trivial solution, we set a non-zero low bound for α, $\eta/\sqrt{t}$ ($\eta = 10^{-8}$) in our implementation.

3.1.2 Convergence Analysis

Regarding the convergence of the proposed algorithm, we have the following theorem:

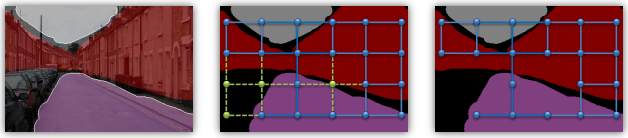
Theorem 3.2 *Suppose $\mathbf{w}^*$ is the optimal solution that minimizes (8), t is the number of iterations, $\forall\epsilon > 0$, the final solution $\mathbf{w}_f$ obtained by the algorithm 1 is bounded by:*

$$\lim_{t \to +\infty} \rho(\mathbf{w}_f) - \rho(\mathbf{w}^*) \leq \epsilon + \frac{1}{2}\|\mathbf{w}_f\|^2 - \frac{1}{2}\|\mathbf{w}^*\|^2 \quad (19)$$

The proof is given in the supplementary material [1].

4. Learning with Partially Labeled Image

Partially labeled training images are common in image parsing problems, as it is usually very time-consuming to get precise annotations by manual labeling. A typical partially labeled example is shown in Figure 2(a). The unlabeled regions in partially labeled training images are not trivial for the parameter learning of CRF models, as observed in previous works [25, 28]. As evaluating the loss on the unlabeled regions during the learning process is not feasible, discarding the unlabeled regions would be a straightforward choice, which excludes the unlabeled regions from the CRF models built for the partially labeled training images in the learning process. However, without considering the unlabeled regions, the interactions between the labeled



(a) (b) (c)

Figure 2. (a) A partially labeled training image. The unlabeled regions are shown in black; (b) and (c), the pairwise CRF models for the parameter learning with different ways to treat the unlabeled regions in the training image. (b) using the constraint (20), the nodes in the unlabeled regions and links linked to them are shown in green. (c) discarding the unlabeled regions in the parameter learning, with the nodes and links for the unlabeled regions in (b) excluded.

regions and the unlabeled regions will not be modeled in the learning process. This could affect the parameter learning of CRF models. For example, for the boundaries between the labeled regions and unlabeled regions, as these boundaries are mostly not the real boundaries between different categories, the pairwise smoothness should be preserved on these boundaries. Without the interactions between the labeled regions and the unlabeled regions, the pairwise smoothness constraint on these boundaries will not be encoded in the learning process.

To deal with partially labeled training images, we propose a new objective constraint for the parameter learning of CRF models by modifying the objective constraint (12), with the CRF models built for partially labeled training images in the learning process taking into both the labeled regions and the unlabeled regions. Let R_k and R_u denote the labeled regions and the unlabeled regions in the partially labeled training images, $\mathbf{y}_k^*$ denote the ground truth label for R_k. In each iteration of the algorithm 1, the obtained label prediction $\hat{\mathbf{y}}_t$ can be divided into two parts: the labeling configuration for R_k and the labeling configuration for R_u, and we denote them as $\hat{\mathbf{y}}_t^k$ and $\hat{\mathbf{y}}_t^u$. Then, the new objective constraint is defined as:

$$\mathcal{H}(\mathbf{w}; \mathbf{x}^*, \mathbf{y}_t^*, \hat{\mathbf{y}}_t) \leq \xi \tag{20}$$

where the ground truth label $\mathbf{y}_t^* = \mathbf{y}_k^* \cup \hat{\mathbf{y}}_t^u$ consists of the ground truth label $\mathbf{y}_k^*$ for R_k and the predicted label $\hat{\mathbf{y}}_t^u$ for R_u. Note that when there are no unlabeled regions in the training images, (12) and (20) are the same. A simple pairwise CRF model for a partially labeled training image is shown in Figure 2, with different ways to handle the unlabeled regions in the partially labeled training images illustrated.

5. Experiment

To evaluate the proposed method, we choose one typical CRF model widely used in the image parsing: the Robust P^N model [14], with its energy function defined as:

$$\begin{aligned} E(\mathbf{x}, \mathbf{y}) &= \sum_{i \in \mathcal{V}} \Psi_i(y_i) + \sum_{(i,j) \in \mathcal{E}} \Psi_{ij}(y_i, y_j) + \sum_{c \in \mathcal{S}} \Psi_c(\mathbf{y_c}) \\ &= \mathbf{w}_u \cdot f_u(\mathbf{x}, \mathbf{y}) + \mathbf{w}_p \cdot f_p(\mathbf{x}, \mathbf{y}) + \mathbf{w}_c \cdot f_c(\mathbf{x}, \mathbf{y}) \\ &= \mathbf{w}^T \cdot \Phi(\mathbf{x}, \mathbf{y}) \end{aligned} \tag{21}$$

where $f_u(\mathbf{x}, \mathbf{y})$, $f_p(\mathbf{x}, \mathbf{y})$ and $f_c(\mathbf{x}, \mathbf{y})$ are the label dependent feature vectors for the unary potential, pairwise potential and high order potential to enforce label consistency, and $\Phi(\mathbf{x}, \mathbf{y}) = [f_u(\mathbf{x}, \mathbf{y}), f_p(\mathbf{x}, \mathbf{y}), f_c(\mathbf{x}, \mathbf{y})]$, $\mathbf{w} = [\mathbf{w}_u, \mathbf{w}_p, \mathbf{w}_c]$. The parameters to be learned include $\mathbf{w}_u = [w_u]$, $\mathbf{w}_c = [w_c]$ and $\mathbf{w}_p = [w_p^1, w_p^2, \cdots, w_p^L]$, where L is the number of categories. Similar to [25], the unary potential is defined on pixel level, multiplied by the weight parameter $\mathbf{w}_u$. The Robust Higher Order Potential is defined the same as that in [14], multiplied by the weight parameter $\mathbf{w}_c$. The pairwise smoothness potential is defined as:

$$\Psi_{ij}(y_i, y_j) = \begin{cases} \lambda(i,j)(w_p^{y_i} + w_p^{y_j}), y_i \neq y_j; \\ 0, y_i = y_j \end{cases} \tag{22}$$

where $\lambda(i,j) = 1/(1 + c \parallel D_i - D_j \parallel^2)$ is the contrast sensitive between neighboring pixels. D_i and D_j are the RGB color vectors of nodes $i, j \in \mathcal{T}$ in the graph.

The evaluation of the proposed method is carried out on two public datasets: the MSRC-21 dataset [25] and CBCL StreetScenes dataset [3], with it compared with another two widely used methods: the Projected Subgradient method [23][1] and the 1-slack StructSVM method [12] which has been demonstrated to be much more efficient than the n-slack StructSVM method [27]. [26] using the n-slack StructSVM method is not included in the comparison, as it needs to solve a large scale QP problem in each iteration and will be very time-consuming, when the number of training images is large.

The procedure of the evaluation includes three successive steps: 1) unary potential training which is identical in our method, the 1-slack StructSVM method and the Projected Subgradient method; 2) parameter learning of the CRF model; 3) testing of the learned CRF model. Each dataset for the evaluation is randomly split with a partition ratio 45%/10%/45% for the above three steps respectively. As our focus is evaluating the parameter learning algorithms for CRF models, we choose simple patch level features: Texton + SIFT for the unary potential training, with the Random Forest classifier [6](50 trees, max depth = 32) chosen as the classifier model. The evaluation procedure is repeated five times with random split of the datasets for the evaluation, and the results reported in the following sections are the averaged results. The parameter learning of

[1] As the energy function for the Robust P^N model is submodular, no decomposition in [15] is necessary for the model, which makes [15] and the Projected Subgradient method [23] equivalent in this situation.

the Robust P^N model starts with the unary classification, with the parameters to be learned initialized as $\mathbf{w}_u = \mathbf{1}$, $\mathbf{w}_p = \mathbf{0}$, $\mathbf{w}_c = \mathbf{0}$ for our method, the 1-slack StructSVM method and Projected Subgradient method. For the MAP inference on the Robust P^N model, the α-expansion algorithm is used. For the performance evaluation, we use two criteria: CAA (category average accuracy, the average proportion of pixels correctly labeled in each category) and GA (global accuracy, proportion of all pixels correctly labeled) same as the previous works on the image parsing [25, 18].

Robust P^N model	Unary classification	StructSVM [12] (25 iterations)	Subgradient [23] (200 iterations)	Our method (5 iterations)
GA	0.63	0.692	0.701	**0.708**
CAA	0.421	0.476	**0.507**	0.503

(a) MSRC-21 dataset

Robust P^N model	Unary classification	StructSVM [12] (33 iterations)	Subgradient [23] (200 iterations)	Our method (5 iterations)
GA	0.666	0.687	0.692	**0.696**
CAA	0.623	0.641	0.644	**0.647**

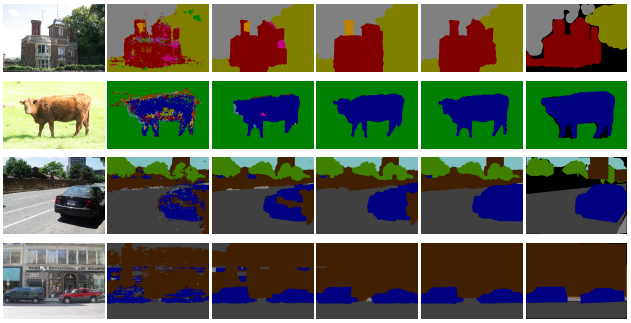
(b) CBCL StreetScenes dataset

Table 1. The segmentation accuracy obtained with unary classification, the Robust P^N models learned by the 1-slack StructSVM method, Projected Subgradient method and our method on the datasets for the evaluation. The average numbers of iterations used by different methods to achieve the reported segmentation accuracy are indicated in the parentheses.

5.1. Performance of the learned models

The segmentation accuracy achieved with the unary classification as well as the Robust P^N models learned by our method, the 1-slack StructSVM method and Projected Subgradient method on the two datasets for the evaluation is given in Table 1. As some critical parameters in the learning formulation of our method, the 1-slack StructSVM method and Projected Subgradient method could influence the performance of the learned CRF models to varying degrees, these critical parameters are carefully tuned for a fair comparison by trying different values, and the achieved best results are reported for the performance comparison in Table 1. These critical parameters include the initial upper bound of the updating step-size in our method, the updating step-sizes and the weight of constraint violation in the Projected Subgradient method, the weight of slack variable in the 1-slack StructSVM method. More details are explained in the supplementary material [1]. Several examples of the parsing results obtained by different methods are illustrated in Figure 3.

MSRC-21 dataset This dataset contains 591 images covering 21 categories. The segmentation accuracy achieved with the Robust P^N models learned by our method, 1-slack StructSVM method and Projected Subgradient method are given in Table 1 (a). From the comparison in Table 1 (a), we find that compared with the unary classification, the segmentation accuracy is improved to varying degrees by the Robust P^N models learned by all the three methods. The segmentation accuracy achieved with the Robust P^N model learned by our method outperforms that achieved with the model learned by the 1-slack StructSVM method and is comparable to that achieved with the Robust P^N model learned by the Projected Subgradient method.



(a) (b) (c) (d) (e) (f)

Figure 3. The parsing results obtained by unary classification and the Robust P^N models learned by the 1-slack StructSVM [12], Projected Subgradient [23] and our method on the MSRC-21 dataset and the CBCL street scene dataset. (a) test images; (b) unary classification; (c) 1-slack StructSVM; (d) Projected Subgradient; (e) our method, using the modified constraint (20); (f) ground truth annotation

Time Cost Learning of Robust P^N model	StructSVM [12]	Subgradient [23]	Our method
MSRC-21	114 min	935 min	26 min
CBCL	276 min	1512 min	51 min

Table 2. The average time cost of the 1-slack StructSVM method, Projected Subgradient method and our method for the parameter learning of the Robust P^N models on the MSRC-21 dataset and CBCL StreetScenes dataset. All methods are implemented with C++ and tested on the same platform (Intel i7 2.8G, 8G RAM).

CBCL StreetScenes dataset This dataset contains 3547 images of street scenes, covering nine categories: *car*, *pedestrian*, *bicycle*, *building*, *tree*, *sky*, *road*, *sidewalk*, and *store*. We exclude the three categories with frequency of occurrence under 1%: *pedestrian*, *bicycle* and *store* in the test. The segmentation accuracy achieved with the Robust P^N models learned by our method, the 1-slack StructSVM method and Projected Subgradient method is given in Table 1(b). Similar to the result on the MSRC-21 dataset, the Robust P^N models learned by all the three methods improve the segmentation accuracy to varying degrees. We

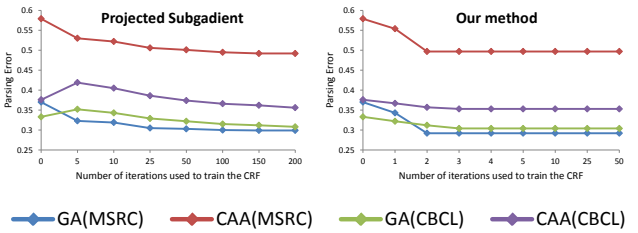


Figure 4. The number of iterations used to train the Robust P^N models in the Projected Subgradient method [23] and our method to the corresponding parsing error achieved with the learned Robust P^N models on the MSRC-21 dataset and the CBCL StreetScenes dataset. The parsing error is measured by the loss of GA (global accuracy) and CAA (category average accuracy).

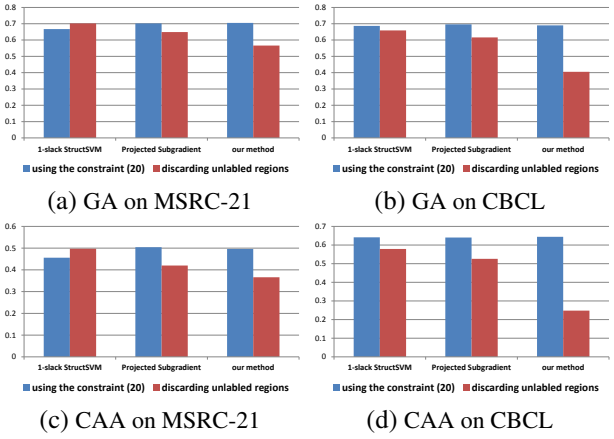
also find that the segmentation accuracy achieved with the Robust P^N model learned by our method is slightly better than that achieved with the models learned by the 1-slack StructSVM method and Projected Subgradient method.

5.2. Learning efficiency

The average time cost to train the Robust P^N models by different methods is presented in Table 2. The learning efficiency of our method and the Projected Subgradient method depends on the predefined numbers of iterations used to train the CRF models, such as M in the algorithm 1. The relationship between the numbers of iterations used to train the Robust P^N models and the corresponding performance of the trained models is plotted in Figure 4 for these two methods. In Figure 4, we find that on the test set of both datasets for the evaluation, the parsing errors of the Robust P^N model learned by our method become stable rapidly, after only five iterations. By contrast, on the test sets of both datasets for the evaluation, the parsing errors of the Robust P^N model learned by the Projected Subgradient method decreased gradually, approximately stable after 200 iterations. Therefore, in our experiment, the numbers of iterations used to train the CRF models in our method and the Projected Subgradient method are set as 5 and 200 respectively. For the 1-slack StructSVM method, it converged after about 25 and 33 iterations on the MSRC-21 dataset and CBCL StreetScenes dataset respectively, with the stop condition that the learned parameters of the CRF model keep unchanged, same as that in [26].

5.3. Learning with partially labeled training images

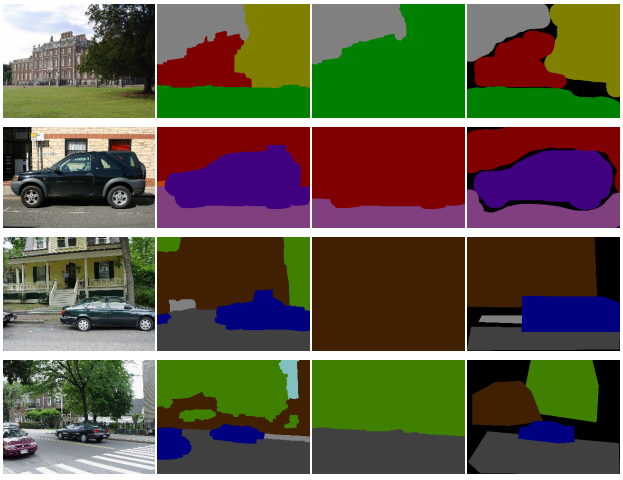
To evaluate the influence of the unlabeled regions in partially labeled training images on the parameter learning of CRF models, we test the three learning methods with two different ways to treat the unlabeled regions, using the modified objective constraint (20) and discarding the unlabeled regions. Before the evaluation, we first force the boundaries



(a) GA on MSRC-21 (b) GA on CBCL

(c) CAA on MSRC-21 (d) CAA on CBCL

Figure 5. The segmentation accuracy achieved on the MSRC-21 and CBCL dataset with the Robust P^N models learned by our method, the 1-slack StructSVM [12] and Projected Subgradient method [23], using different ways to treat the unlabeled regions. (a) and (b), the global accuracy on the MSRC-21 and the CBCL dataset; (c) and (d), the category average accuracy on the MSRC-21 and the CBCL dataset



(a) (b) (c) (d)

Figure 6. The parsing results obtained with the Robust P^N models learned by our method, with different ways to treat the unlabeled regions in partially labeled training images. (a) test images; (b) using the modified objective constraint (20); (c) discarding the unlabeled regions; (d) ground truth annotation

between different categories in the annotation masks unlabeled by clearing the labels of the boundary pixels between different categories, similar to the segmentation annotation in the VOC dataset [8].

The evaluation result is given in Figure 5, and we find that when using the modified objective constraint (20) for the parameter learning, the parsing performance of Robust P^N models learned by our method and the Projected Sub-

gradient method is significantly improved on both datasets for the evaluation. This indicates that modeling the interactions between the labeled regions and unlabeled regions of the partially labeled training images in the learning process is important for our method and the Projected Subgradient method. Please note that the results of our method and the Projected Subgradient method reported in Table 1 are also obtained by using the modified objective constraint (20), as many images in the two datasets for the evaluation are partially labeled. For the 1-slack StructSVM method, the learned models using the modified objective constraint (20) achieve better performance on the CBCL StreetScenes dataset and worse performance on the MSRC-21 dataset. Several examples of the parsing results obtained with the Robust P^N models learned by our method are illustrated in Figure 6, with different ways to treat the unlabeled regions in partially labeled training images.

6. Conclusion

We present an adaptive subgradient descent method to learn parameters of CRF models for image parsing. In each iteration of the algorithm, the adaptive subgradient descent updating is carried out by solving a simple convex optimization problem which has a subgradient-based solution with an adaptively determined step-size. The adaptively determined updating step-size can well balance the learning efficiency and performance of the learned CRF models. Meanwhile, the proposed method is capable of handling partially labeled training data robustly, with a new objective constraint modeling both the labeled and unlabeled parts in the partially labeled training images for the parameter learning.

Acknowledgements This work was partially supported by the Hong Kong RGC GRF 618510 and 618711, NSFC/RGC Joint Research Scheme N-HKUST607/11, and the National Basic Research Program of China (2012CB316300).

References

[1] https://sites.google.com/site/honghuizhanghk/supplementary.pdf. 4, 6

[2] K. Alahari, C. Russell, and P. H. S. Torr. Efficient piecewise learning for conditional random fields. *CVPR*, 2010. 1

[3] S. Bileschi. *StreetScenes: Towards Scene Understanding in Still Images*. PhD thesis, MIT, 2007. 5

[4] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004. 3, 9

[5] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *PAMI*, 23(11):1222–1239, 2001. 3

[6] L. Breiman. Random forests. *Machine Learning*, 2001. 5

[7] B.Taskar, V.Chatalbashev, and D.Koller. Learning associative markov networks. *ICML*, 2004. 1, 2

[8] M. Everingham, L. Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *IJCV*, 88:303–338, 2010. 7

[9] S. Gould. Max-margin learning for lower linear envelope potentials in binary markov random fields. *ICML*, 2011. 1

[10] U. Heinemann and A. Globerson. What cannot be learned with bethe approximations. *NIPS*, 2011. 2

[11] H.Zhang, J.Xiao, and L.Quan. Supervised label transfer for semantic segmentation of street scenes. *European Conference on Computer Vision*, 2010. 1

[12] T. Joachims, T. Finley, and C.-N. J. Yu. Cutting-plane training of structural svms. *Machine Learning*, 77:27–59, 2009. 1, 2, 5, 6, 7, 11

[13] K. C. Kiwiel. A proximal bundle method with approximate subgradient linearizations. *SIAM Journal on Optimization*, 16:1007 – 1023, 2006. 3

[14] P. Kohli, L. Ladicky, and P. Torr. Robust higher order potentials for enforcing label consistency. *IJCV*, 2009. 4, 5

[15] N. Komodakis. Learning to cluster using high order graphical models with latent variables. *ICCV*, 2011. 1, 2, 5

[16] P. Krahenbuhl and V. Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. *NIPS*, 2011. 1

[17] S. Kumar and M. Hebert. Discriminative fields for modeling spatial dependencies in natural images. *NIPS*, 2004. 1

[18] L. Ladicky, C. Russell, P. Kohli, and P. Torr. Associative hierarchical crfs for object class image segmentation. *ICCV*, 2009. 1, 6

[19] J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields : Probabilistic models for segmenting and labeling sequence data. *ICML*, 2001. 1

[20] D. Munoz, J. A. Bagnell, N. Vandapel, and M. Hebert. Contextual classification with functional max-margin markov networks. *CVPR*, 2009. 1

[21] S. Nowozin, P. V. Gehler, and C. H. Lampert. On parameter learning in crf-based approaches to object class image segmentation. *ECCV*, 2010. 1

[22] B. Polyak. A general method for solving extremum problems. *Soviet Math*, 3:593–597, 1967. 2

[23] N. D. Ratliff, J. A. Bagnell, and M. A. Zinkevich. (online) subgradient methods for structured prediction. *AISTATS*, 2006. 1, 2, 3, 4, 5, 6, 7, 11

[24] R. Samdani and D. Roth. Efficient decomposed learning for structured prediction. *ICML*, 2012. 2

[25] J. Shotton, J. Winn, C. Rother, and A. Criminisi. Textonboost for image understanding: Multi-class object recognition and segmentation by jointly modeling texture, layout, and context. *ECCV*, 2006. 1, 4, 5, 6

[26] M. Szummer, P. Kohli, and D. Hoiem. Learning crfs using graph cuts. *ECCV*, 2008. 2, 3, 5, 7

[27] I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun. Large margin methods for structured and interdependent output variables. *JMLR*, 2005. 1, 2, 5

[28] J. Verbeek and W. Triggs. Scene segmentation with crfs learned from partially labeled images. *NIPS*, 2007. 2, 4

[29] C.-N. J. Yu and T. Joachims. Learning structural svms with latent variables. *ICML*, 2009. 2