The More the Merrier: Parameter Learning for Graphical Models with Multiple MAPs

Franziska Meier FMEIER@USC.EDU
U. of Southern California, 941 Bloom Walk, Los Angeles, CA 90089 USA

Amir Globerson AMIR.GLOBERSON@MAIL.HUJI.AC.IL
Hebrew U. of Jerusalem, Jerusalem, Israel

Fei Sha FEISHA@USC.EDU
U. of Southern California, 941 Bloom Walk, Los Angeles, CA 90089 USA

Abstract

Conditional random field (CRFs) is a popular and effective approach to structured prediction. When the underlying structure does not have a small tree-width, maximum likelihood estimation (MLE) is in general computationally hard. Discriminative methods such as Perceptron or Max-Margin Markov Networks circumvent this problem by requiring the MAP assignment only, which is often more tractable, either exactly or approximately with linear programming (LP) relaxations. In this paper, we propose an approximate learning method for MLE of CRFs. We leverage LP relaxations to find multiple diverse MAP solutions and use them to approximate the intractable partition function. The proposed approach is easy to parallelize, and yields competitive performance in test accuracies on several structured prediction tasks.

1. Introduction

We study the problem of parameter estimation for structured prediction models. In this setting, we want to predict a complex label $\boldsymbol{x}$ given an observation $\boldsymbol{o}$. For example, in semantic parsing of images, pixels values are observations and pixels' visual categories (SKY, WATER, or BOAT) are labels. We construct a graphical model on $\boldsymbol{x}$ for each (instantiated) value of $\boldsymbol{o}$ to encode the interdependencies of these labels. The maximum a posteriori (MAP) assignment of the conditional model $\boldsymbol{x}^* = \arg\max p(\boldsymbol{x}|\boldsymbol{o})$ is then used as the prediction. For training such models, two popular frameworks have been developed: conditional random fields (CRFs) which maximizes the conditional likelihood $p(\boldsymbol{x}|\boldsymbol{o})$ of the training data, and perceptron/max-margin Markov networks (M^3N) which minimizes the discriminative loss when the MAP assignment $\boldsymbol{x}^*$ is different from the ground-truth labeling of $\boldsymbol{o}$ (Taskar et al., 2003; Collins, 2002).

Both these learning problems are hard for models defined on general graphs (Koller & Friedman, 2009). However, training methods such as perceptron and M^3N requires computing the MAP assignment only, whereas CRF training also requires calculating marginals and partition functions. For MAP inference, recent works (e.g., Globerson & Jaakkola, 2007; Sontag et al., 2010) have provided efficient approximation procedures that yield optimality certificates and often work well in practice. Plugging these into the learning scheme results in tractable and scalable learning algorithms that work well in practice. However, in the CRF case, it is not immediately clear how to leverage these algorithms for learning.

The above motivated us to develop a learning algorithm for CRFs that employs approximate MAP inference methods. Our key idea is to use them to find the main modes of $p(\boldsymbol{x}|\boldsymbol{o})$ and then use those modes to approximate the marginals of the distribution. Concretely, we seek to identify a set of K assignments that are representative of the distribution, in particular, have higher probabilities "locally".

While in theory such assignments can be found through sampling the distribution, we leverage approximate MAP inference to provide a deterministic procedure in identifying them. Our key observation is that those assignments not only should have high probabilities but also should be sufficiently different from each other — a set of very similar assignments would result

in a poor qualitative description of $p(\boldsymbol{x}|\boldsymbol{o})$ (unless the distribution happens to be unimodal). This intuition is strongly supported by our empirical studies, where we show that either using the MAP (corresponding to $K = 1$) or using the assignments without encouraging being distinct result in worse prediction accuracies on models trained for structured prediction tasks such as OCR and image segmentation.

The proposed deterministic procedure for identifying such diverse assignments, which are termed as DIV P-MAP , has several appealing properties. It is computationally tractable and easily parallelizable. This is in contrast to other approaches for inferring multiple MAPs which compute them serially (Fromer & Globerson, 2009; Batra et al., 2012).

2. Background

We focus on parameter learning for structure prediction models. We start by describing the problem setting and introducing the necessary notations, followed by a brief discussion on two common techniques for estimating parameters for such models.

Conditional random fields (CRFs) A conditional random field (CRF) defines the distribution of a set of random variables $\boldsymbol{x}$, conditioned on an observation $\boldsymbol{o}$

$$p(\boldsymbol{x}|\boldsymbol{o}, \boldsymbol{\lambda}) = \frac{1}{Z(\boldsymbol{o})} \exp\left(\boldsymbol{\lambda}^T F(\boldsymbol{x}, \boldsymbol{o})\right) \tag{1}$$

where $F(\boldsymbol{x}, \boldsymbol{o})$ is a feature vector, computed on the underlying graph $\mathcal{G}(V, E)$ of the CRF. In the graph, the vertices V correspond to the random variables in $\boldsymbol{x}$ (and the observation $\boldsymbol{o}$). The edges in E encode the interdependencies among those variables. We consider the case where $\boldsymbol{x}$ are discrete. $\boldsymbol{\lambda}$ denotes the parameters of the model. They are to be learned from a training data set $\mathcal{D} = \{(\boldsymbol{x}_n, \boldsymbol{o}_n)\}_{n=1}^N$.

The optimal $\boldsymbol{\lambda}$ can be estimated via gradient ascend on the conditional log-likelihood of the training data. Concretely, the gradient is the sum of gradients of each observed training example,

$$\begin{aligned} \frac{\partial \log \mathcal{D}}{\partial \boldsymbol{\lambda}} &= \sum_n \frac{\partial \log p(\boldsymbol{x}_n|\boldsymbol{o}_n, \boldsymbol{\lambda})}{\partial \boldsymbol{\lambda}} \\ &= \sum_n F(\boldsymbol{x}_n, \boldsymbol{o}_n) - \mathbb{E}_{p(\boldsymbol{x}|\boldsymbol{o}_n, \boldsymbol{\lambda})}[F(\boldsymbol{x}, \boldsymbol{o}_n)] \end{aligned} \tag{2}$$

where the first term is the empirical evaluation of the feature vector on the training data and the second term the feature vector's expectation with respect to the model's distribution.

At the t-th iteration, the gradient ascent improves the log-likelihood by updating the parameter according to

$$\boldsymbol{\lambda}^t = \boldsymbol{\lambda}^{t-1} + \eta \frac{\partial \log \mathcal{D}}{\partial \boldsymbol{\lambda}} \tag{3}$$

where η is the learning rate (ie., step size). The batch update eq. (3) requires aggregating all training samples' contribution to the gradient. Alternatively, another popular approach is to use stochastic gradient ascend to maximize the likelihood. Specifically, let $(\boldsymbol{x}^t, \boldsymbol{o}^t)$ be a random sample selected from the training data at the t-th iteration. The stochastic gradient update gives rise to

$$\boldsymbol{\lambda}^t = \boldsymbol{\lambda}^{t-1} + \eta \frac{\partial \log p(\boldsymbol{x}^t|\boldsymbol{o}^t, \boldsymbol{\lambda})}{\partial \boldsymbol{\lambda}} \tag{4}$$

Compared to the batch version, stochastic gradient updates often result in significant improvement in a smaller number of iterations as they permit exploring the parameter space faster.

Perceptron The perceptron algorithm introduced in (Collins, 2002) can be viewed as a form of stochastic gradient update except that the "gradients" used by the perceptron are further approximated.

At the t-th iteration, the perceptron algorithm computes the MAP assignment $\boldsymbol{x}^*$ corresponding to a randomly chosen training instance $(\boldsymbol{x}^t, \boldsymbol{o}^t)$,

$$\boldsymbol{x}^* = \arg\max_{\boldsymbol{x}} \log p(\boldsymbol{x}|\boldsymbol{o}^t, \boldsymbol{\lambda})$$

The update to the parameter $\boldsymbol{\lambda}$ is given by the difference between the empirical evaluation of the feature vector and an approximate to its expectation,

$$\boldsymbol{\lambda}^t = \boldsymbol{\lambda}^{t-1} + F(\boldsymbol{x}^t, \boldsymbol{o}^t) - F(\boldsymbol{x}^*, \boldsymbol{o}^t) \tag{5}$$

Note that, when the optimal "decoding" $\boldsymbol{x}^*$ is the same as the ground-truth $\boldsymbol{x}^t$, the parameter is not updated.

Contrasting the perceptron update eq. (5) to the stochastic gradient update eq. (4), it is easy to see that the perceptron learning approximates the conditional distribution $p(\boldsymbol{x}|\boldsymbol{o}^t, \boldsymbol{\lambda})$ by putting all probability mass on the MAP assignment $\boldsymbol{x}^*$. Our approach exploits this observation further, aiming to approximate the distribution better by using several assignments.

Note that, while the perceptron update minimizes the following discriminative loss

$$\min_{\boldsymbol{\lambda}} \sum_n \max \boldsymbol{\lambda}^T F(\boldsymbol{x}, \boldsymbol{o}_n) - \boldsymbol{\lambda}^T F(\boldsymbol{x}_n, \boldsymbol{o}_n) \tag{6}$$

the Max-margin Markov Nework ($\mathrm{M}^3\mathrm{N}$) approach analogously minimizes the structured hinge loss

$$\min_{\boldsymbol{\lambda}} \sum_n [\max \boldsymbol{\lambda}^T F(\boldsymbol{x}, \boldsymbol{o}_n) + \ell(\boldsymbol{x}, \boldsymbol{x}_n) - \boldsymbol{\lambda}^T F(\boldsymbol{x}_n, \boldsymbol{o}_n)]_+ \tag{7}$$

where the loss $\ell(\boldsymbol{x}, \boldsymbol{x}_n)$ counts the difference between an assignment and the ground-truth. Since the decomposable loss function depends on the (loss-augmented) MAP assignment only too, M^{3}N requires only the MAP inference be tractable.

3. Learning with multiple MAPs

Recent works (e.g., Globerson & Jaakkola, 2007; Sontag et al., 2010) have provided efficient approximation procedures for computing the MAP assignment, which are well suited for learning parameters with perceptron and M^{3}N algorithms. However, for maximum likelihood estimation used in CRFs, those procedures are not immediately applicable as a single MAP assignment is inadequate in capturing the whole shape of the conditional distribution $p(\boldsymbol{x}|\boldsymbol{o})$, which is needed for computing expectations of the feature vector.

3.1. Main idea

Our idea is to explore *multiple* assignments to represent the conditional distribution, instead of putting all the probability mass on a single configuration $\boldsymbol{x}^*$. Suppose we have a set of such assignments $\mathcal{X} = (\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, \cdots, \boldsymbol{x}^{(K)})$, where we have used $\boldsymbol{x}^{(1)}$ to denote the MAP configuration $\boldsymbol{x}^*$. We will approximate the expectation in eq. (2) with a weighted average,

$$\frac{\partial \log p(\boldsymbol{x}|\boldsymbol{o}, \boldsymbol{\lambda})}{\partial \boldsymbol{\lambda}} \approx F(\boldsymbol{x}, \boldsymbol{o}) - \sum_{k=1}^{K} w_k F(\boldsymbol{x}^{(k)}, \boldsymbol{o}) \quad (8)$$

where the nonnegative weight w_k is proportional to the likelihood of the k-th configuration $\boldsymbol{x}^{(k)}$

$$w_k = \frac{e^{\boldsymbol{\lambda}^T F(\boldsymbol{x}^{(k)}, \boldsymbol{o})}}{\sum_{j=1}^{K} e^{\boldsymbol{\lambda}^T F(\boldsymbol{x}^{(j)}, \boldsymbol{o})}} \quad (9)$$

Note that since w_k is always less than 1, the approximation is less biased towards the MAP configuration $\boldsymbol{x}^{(1)}$ than the perceptron algorithm. Our empirical studies will show that the approximation leads to models with very much improved performance on the testing data.

How to choose the K assignments in $\mathcal{X}$? We address this question in the following section.

3.2. Multiple MAPs: best and diverse

To approximate $p(\boldsymbol{x}|\boldsymbol{o})$ well with K discrete items, we would want those assignments which correspond to the modes of the distribution. We discuss several options.

K-BEST MAP Intuitively, $p(\boldsymbol{x}^{(k)}|\boldsymbol{o})$ would be "local maxima" at $\boldsymbol{x}^{(k)}$. Thus, we choose them in serial

$$\boldsymbol{x}^{(k)} = \underset{\boldsymbol{x} \neq x^j, j=1,2,\ldots,k-1}{\arg\max} p(\boldsymbol{x}|\boldsymbol{o}) \quad (10)$$

Such assignments, which are termed K-BEST MAP, can be found efficiently for tree structured graphs, and approximated for general graphs (Yanover & Weiss, 2004; Fromer & Globerson, 2009; Batra, 2012). However, these assignments tend to be very similar to each other, tightly clustering around the most dominant mode $\boldsymbol{x}^{(1)}$. For example, any assignment $\boldsymbol{x}$ that is different from $\boldsymbol{x}^{(1)}$ by just one vertex's value could be a viable candidate for $\boldsymbol{x}^{(2)}$. Unless the conditional distribution $p(\boldsymbol{x}|\boldsymbol{o})$ happens to be unimodal, $\mathcal{X}$ would not be a good approximation to the distribution.

K-DIV MAP To avoid collecting very similar assignments in $\mathcal{X}$, Batra et al. (2012) proposed to identify K assignments that are distinct from each other — they are at least D vertices different

$$\boldsymbol{x}^{(k)} = \underset{\sum_{v=1}^{|V|} \mathbb{I}(\boldsymbol{x}_v \neq \boldsymbol{x}_v^j) \geq D, j=1,2,\ldots,k-1}{\arg\max} p(\boldsymbol{x}|\boldsymbol{o}) \quad (11)$$

where $\mathbb{I}(\cdot)$ is the indicator function, comparing the v-th vertex of the two assignments — one is to be found and the other is a assignment from the previously computed $(k-1)$ assignments.

Batra et al. (2012) showed how the inference in eq. (11) can be solved efficiently. The key observation there is that the constraints enforcing distinctiveness among assignments are decomposable on the vertices and thus are easily incorporated into the framework of dual decomposition for the MAP inference problem eq. (2) (Sontag et al., 2010).

Both ways of finding multiple MAPs are serial in nature — one has to know all $(k-1)$ MAPs before computing the k-th MAP assignment. Thus, computing them on large problems like image segmentation would be quite time-consuming.

To this end, we introduce a different formulation of computing diverse MAP assignments that enables us to *parallelize* the estimation of the other $(K-1)$ MAP assignments after the first MAP $\boldsymbol{x}^{(1)}$ has been found.

Diverse Parallel MAP (DIV P-MAP) The K best or diverse assignments as defined above need to be calculated in serial. Here we explore an alternative definition that is more amenable to parallelization. Given the MAP $\boldsymbol{x}^{(1)}$, we define $\boldsymbol{x}^{(k)}$ to be the best assignment that is between $(k-1)D$ and kD away from $\boldsymbol{x}(1)$ (for some parameter D). By considering the top assignments of this kind, we are effectively representing the best assignments at different distances from the MAP.

Formally our k-th MAP requires solving

$$\begin{aligned} \boldsymbol{x}^{(k)} = \quad & \arg\max p(\boldsymbol{x}|\boldsymbol{o}) \\ \text{s.t} \quad & (k-1)D \leq \sum_{v=1}^{|V|} \mathbb{I}(\boldsymbol{x}_v \neq \boldsymbol{x}_v^{(1)}) \leq kD \end{aligned} \tag{12}$$

where the lower and upper bound constraints are expressed in terms of k for $k = 2, \ldots, K$ and parameter D. Thus, the computation of the k-th MAP only depends on $x^{(1)}$ and the value of k and is entirely independent of all other diverse MAPs. As a result each of the optimization problems will be optimizing for the k-th MAP in its own interval, allowing for a distributed computation of all $x^{(k)}$. The parameter D defines the size of the interval in which each $x^{(k)}$ is optimized.

We derive the algorithm in detail in the appendix.

4. Experiments

We conduct extensive experiments on two structure prediction tasks, handwritten digits recognition (OCR) and multi-class image segmentation, to validate the effectiveness of our approach of using diverse MAP solutions to learn parameters of CRFs.

There are two parameters to set in our approach DIV P-MAP : D defines the distances from the first MAP assignments eq (12), and η which is the learning rate of the stochastic gradient ascent eq. (4). In all our experiments, we fix $\eta = 0.1$ and D to $0.1 * |V|$ for the OCR task and to $0.2 * |V|$ for the segmentation tasks, where $|V|$ is the number of vertices in the CRF.

4.1. OCR Task

We use the data provided in (Taskar et al., 2003). We model a word as a sequential CRF, where the vertices represent the letters that can take 26 different states. The observations are binary images in the size of 16×8. The unary potentials are defined as weighted sums of the pixel values in the binary images, resulting a total $16 \times 8 \times 26$ number of parameters. The pairwise potentials are the 26×26 transition matrix between the states. All these parameters are learned from the data. We used randomly selected 600 words as training samples and 100 words as test samples. Our evaluation metric is the classification accuracy measured in the number of correctly labeled letters in the test words.

Figure 1 contrasts the performance of several ways of learning the model parameters: (i) EXACT: the forward-backward procedure for computing the gradients exactly; (ii) PERCEPTRON: the single MAP assignment $\boldsymbol{x}^{(1)}$) is used to approximate the gradient; (iii) K-BEST MAP: use multiple MAP assignments that are not required to be significantly distinct from each other to approximate the gradient; (iv) DIV P-MAP : our proposed approach that uses multiple but distinct MAP assignments to approximate the gradient. The approximate gradient is then used in the stochastic gradient ascent eq. (4) to learn parameters. The figure displays the training curves — how accuracies on training and testing words are changed following each sweep through the training samples.

The results clearly demonstrate the advantage of using multiple MAP assignments over the single one (as in PERCEPTRON) in learning the models. Specifically, both training and test accuracies are improved.

The results also support strongly our hypothesis that approximating gradients with multiple but distinct MAP assignments is superior to methods that do not require distinctiveness among those assignments. Note that while the training accuracies are not improved, the testing accuracies are significantly improved by requiring diversity in MAP assignments.

Finally, note that a larger K leads to better test accuracies in general and also to a bigger improvement between diverse MAPs and regular MAPs.

4.2. Image Segmentation

For this task, we use a pairwise grid-structured CRF. We use the MSRC-21 image dataset (Shotton et al., 2007) and extracted 822-dimensional texton, color, HOG and location feature vectors on every grid point (pixel), following (Ladicky et al., 2009). We define the unary potential to be the weighted sum of the extracted feature vector, one for each state (i.e., the number of classes). For foreground/background segmentation, the number of states is 2 and for multi-class segmentation, the number of states is 21, corresponding to the 21 object classes. The pairwise potential is used for smoothing segmentation, ie., to prefer neighboring pixels to have the same class. To this end, we use Potts like potentials which are defined by either 2×2 or 21×21 numbers. Our goal is to learn all the parameters defining both types of potentials.

For computing K-BEST MAP assignments, neither (Fromer & Globerson, 2009) nor (Batra, 2012) is scalable to large problems. Thus, we reduced the computational cost by subsampling the images and study binary foreground/background segmentation, allowing extensive experimentation on K-BEST MAP.

Forground/background segmentation We define object classes WATER,GRASS,SKY and ROAD as background and all other classes to be foreground. We randomly select 20 images for training and 10 images

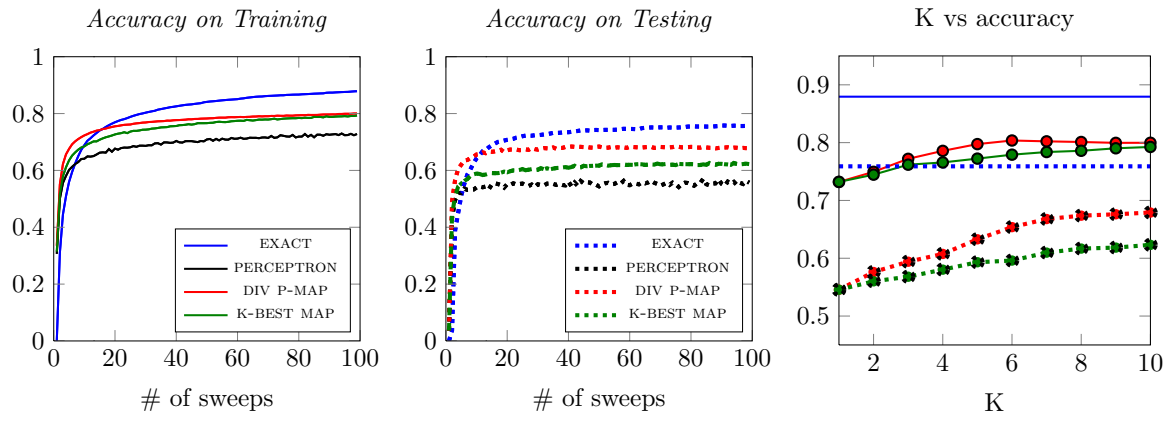

Figure 1. OCR Task. (left) Learning curves on the training data (left) and on the testing data (middle). On the rightmost, accuracies after 100 sweeps for different values for K by different methods. Our proposed approach DIV P-MAP clearly outperforms PERCEPTRON and K-BEST MAPwhich does not consider diversity in multiple MAP assignments

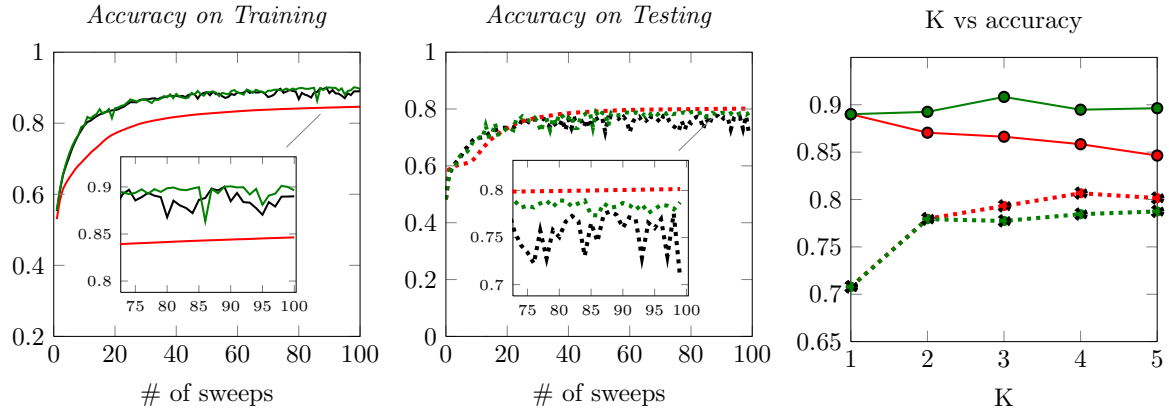

Figure 2. Forground/background segmentation. (left) Learning curves on the training data (left) and on the testing data (middle). Inlets show the details in the last 25 sweeps of training. On the rightmost, accuracies after 100 sweeps for different values for K by different methods. Our approach DIV P-MAP improves slightly over other methods.

for testing. Initial parameters are randomly selected.

Figure 2 displays the learning curves of accuracies — percentage of correctly labeled pixels, averaged over 3 random trials. Using multiple MAPs (either K-BEST MAPor DIV P-MAP) leads to a slightly improved accuracy on the testing data than the PERCEPTRON algorithm. Varying the number of multiple MAPs assignments does not seem to have a major effect.

Multi-class segmentation Fig. 3 displays the results on the full 21-class segmentation. In this setting, our approach is clearly advantageous to other competing ones. Note that the training accuracy by our method is often lower, suggesting that our method is effective in controlling overfitting.

5. Conclusion

We have presented an approximate learning algorithm for CRFs which utilizes multiple diverse MAP assignments to approximate the gradient. We evaluate the method on several tasks in structured prediction and demonstrate its advantage over competing methods.

References

Batra, D. An Efficient Message-Passing Algorithm for the M-Best MAP Problem. In *UAI*, 2012.

Batra, D, Yadollahpour, Payman, and Guzman-Rivera, A. Diverse M-Best Solutions in Markov Random Fields. *ECCV*, 2012.

Collins, Michael. Discriminative training methods for hidden Markov models. In *ACL*, 2002.

Fromer, M and Globerson, A. An LP View of the M-best MAP problem. *NIPS*, 2009.

Globerson, A and Jaakkola, T. Fixing max-product: Convergent message passing algorithms for MAP LP-relaxations. *NIPS*, 21(1.6), 2007.

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

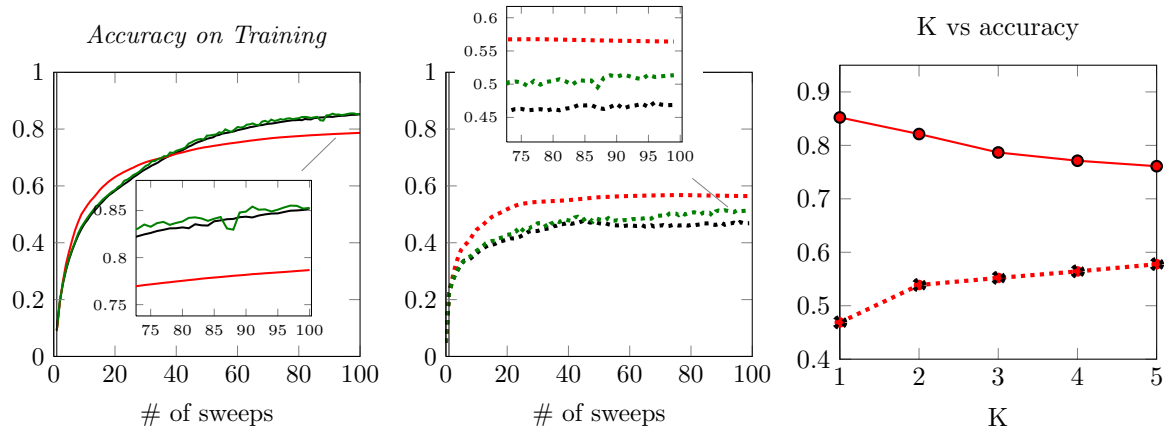

Figure 3. Multi-class segmentation results. Solids curves for training and dashed for testing. Red-colored is for DIV P-MAP , while black-colored and green-colored are for PERCEPTRON and K-BEST MAP, respectively. Our method clearly dominates.

Ladicky, L., Russell, C., Kohli, P., and Torr, P. H S. Associative hierarchical CRFs for object class image segmentation. In *ICCV*, pp. 739–746, 2009.

Shotton, Jamie, Winn, John, Rother, Carsten, and Criminisi, Antonio. TextonBoost for Image Understanding: Multi-Class Object Recognition and Segmentation by Jointly Modeling Texture, Layout, and Context. *IJCV*, 81(1):2–23, December 2007.

Sontag, D., Globerson, A, and Jaakkola, T. Introduction to dual decomposition for inference. *Optimization for Machine Learning*, 2010.

Taskar, B, Guestrin, Carlos, and Koller, D. Max-Margin Markov Networks. In *NIPS*, 2003.

Yanover, Chen and Weiss, Yair. Finding the M Most Probable Configurations Using Loopy Belief Propagation. *NIPS*, 16:289, 2004.

A. Derivation of DIV P-MAP

We illustrate the key steps using pairwise CRFs. The log-linear model is given by the sum of the unary and pairwise potentials, defined on vertices and edges respectively,

$$\log p(\boldsymbol{x}|\boldsymbol{o},\boldsymbol{\lambda}) \propto \sum_{i\in V}\theta_i(x_i) + \sum_{(i,j)\in E}\theta_{ij}(x_i,x_j)$$

Our approach works as follows. We compute the first MAP solution $\boldsymbol{x}^{(1)}$ with existing approaches (Sontag et al., 2010). For other assignments defined in eq. (12), we combine the dual decomposition and additional subgradient descent steps to update the dual variables corresponding to the lower and upper bound constraints. For pairwise CRFs, we compute $x^{(k)}$ as

$$\begin{aligned} x^{(k)} = \quad & \arg\max_{\boldsymbol{x}} \sum_{i\in V}\theta_i(x_i) + \sum_{(i,j)\in E}\theta_{i,j}(x_i,x_j) \\ \text{s.t.} \quad & L \le \sum_{i\in V}\mathbb{I}(x_i = x_i^{(1)}) \le U \end{aligned} \tag{13}$$

where $L = (k-1)D$ and $U = kD$. The Lagrangian of this optimization problem is given by,

$$\begin{aligned} \mathcal{L}(\alpha,\beta,\boldsymbol{x}) = & \sum_{i\in V}(\theta_i(x_i) + (\alpha-\beta)\mathbb{I}(x_i \neq x_i^{(1)})) \\ & + \sum_{(i,j)\in E}\theta_{i,j}(x_i,x_j) - \alpha L + \beta\, U \end{aligned}$$

where α and β are the dual variables for the constraints. The framework of dual decomposition gives rise to the following dual problem

$$\min_{\alpha,\beta\ge 0,\boldsymbol{\delta}} J(\boldsymbol{\delta},\alpha,\beta) = \min_{\boldsymbol{\delta},\alpha,\beta\ge 0}\max_{\boldsymbol{x},\boldsymbol{x}^f}\mathcal{L}(\boldsymbol{\delta},\alpha,\beta,\boldsymbol{x},\boldsymbol{x}^f)$$

where we have created a duplicate for each variable x_i in each of the pairwise factors, collectively referred to them as $\boldsymbol{x}^f$, and $\boldsymbol{\delta}$ is the vector of dual variables, one for each consistency constraint between a duplicate and its original. For details, see (Sontag et al., 2010).

Note that the difference $\mathbb{I}(x_i \neq x_i^{(1)})$ can be directly absorbed in $\theta_i(x_i)$. Thus, the maximization of the Lagrangian is no different from the standard dual decomposition approach. Then we use max product linear programming (MPLP) to minimize $J(\boldsymbol{\delta},\alpha,\beta)$ with respect to $\boldsymbol{\delta}$, while holding α and β fixed. Then we minimize over α and β while holding $\boldsymbol{\delta}$ fixed.

Specifically, let $\hat{x}_i$ be the solution to the maximization. We use the subgradient descent to update α and β

$$\begin{aligned} \alpha^{t+1} &= [\alpha^t - \eta(\sum_{i\in V}\mathbb{I}(\hat{x}_i \neq x_i^{(1)}) - L)]_+ \\ \beta^{t+1} &= [\beta^t - \eta(U - \sum_{i\in V}\mathbb{I}(\hat{x}_i \neq x_i^{(1)}))]_+ \end{aligned}$$

where η is the step size and $[]_+$ is the Heaviside step function.