Efficient piecewise learning for conditional random fields

Karteek Alahari, Chris Russell, Philip H. S. Torr

Efficient Piecewise Learning for Conditional Random Fields

Karteek Alahari Chris Russell Philip H. S. Torr
Oxford Brookes University
Oxford, UK
http://cms.brookes.ac.uk/research/visiongroup

Abstract

Conditional Random Field models have proved effective for several low-level computer vision problems. Inference in these models involves solving a combinatorial optimization problem, with methods such as graph cuts, belief propagation. Although several methods have been proposed to learn the model parameters from training data, they suffer from various drawbacks. Learning these parameters involves computing the partition function, which is intractable. To overcome this, state-of-the-art structured learning methods frame the problem as one of large margin estimation. Iterative solutions have been proposed to solve the resulting convex optimization problem. Each iteration involves solving an inference problem over all the labels, which limits the efficiency of these structured methods. In this paper we present an efficient large margin piecewise learning method which is widely applicable. We show how the resulting optimization problem can be reduced to an equivalent convex problem with a small number of constraints, and solve it using an efficient scheme. Our method is both memory and computationally efficient. We show results on publicly available standard datasets.

1. Introduction

Conditional random fields (CRFs) offer a powerful tool for obtaining a probabilistic formulation for many applications in Computer Vision and related areas [14, 15, 26]. A CRF is defined over a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ denotes a set of vertices and $\mathcal{E}$ is the set of edges, which specifies a pairwise relationship between the vertices[1]. The vertices represent discrete random variables $\mathbf{Y} = \{Y_1, \cdots, Y_N\}$. A labelling of a CRF corresponds to a classification of the vertices by assigning a label to each vertex (variable) from a set of labels $\mathcal{L} = \{1, \cdots, K\}$. In other words, a labelling is specified by a binary vector $\mathbf{y} = \{y_{1:1}, \cdots, y_{1:K}, y_{2:1} \cdots, y_{N:K}\}$, where N is the number of vertices, *i.e.* $|\mathcal{V}| = N$. Each binary indicator variable $y_{i:k} = 1$, if the corresponding random variable Y_i takes the label $k \in \mathcal{L}$, and $y_{i:k} = 0$ otherwise. Also, $\sum_k y_{i:k} = 1, \forall i$. Given some observed data (denoted by $\mathbf{x}$), a CRF models the conditional probability of a labelling $\mathbf{y}$ as follows[2]:

$$\Pr(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta}) = \frac{1}{Z(\boldsymbol{\theta})} \prod_{\substack{i \in \mathcal{V} \\ k \in \mathcal{L}}} \exp(y_{i:k} \boldsymbol{\theta}_k^\top h_i(\mathbf{x})) \prod_{\substack{(i,j) \in \mathcal{E} \\ k,l \in \mathcal{L}}} \exp(y_{i:k} y_{j:l} \boldsymbol{\theta}_{kl}^\top \nu_{ij}(\mathbf{x})), \quad (1)$$

where $\boldsymbol{\theta} = (\boldsymbol{\theta}_k, \boldsymbol{\theta}_{kl}) \in \mathcal{R}^{d \times 1}$ are the parameters of the CRF. The vectors $h_i(\mathbf{x})$ and $\nu_{ij}(\mathbf{x})$ represent features for the vertex $i \in \mathcal{V}$ and the edge $(i,j) \in \mathcal{E}$ respectively. The unary potential $\exp(y_{i:k} \boldsymbol{\theta}_k^\top h_i(\mathbf{x}))$ denotes the cost of the assignment $Y_i = k$, while the pairwise potential $\exp(y_{i:k} y_{j:l} \boldsymbol{\theta}_{kl}^\top \nu_{ij}(\mathbf{x}))$ denotes the cost of the assignment: $Y_i = k$ and $Y_j = l$. The normalizing factor $Z(\boldsymbol{\theta})$ given by:

$$Z(\boldsymbol{\theta}) = \sum_{\mathbf{y}' \in \mathcal{L}^N} \prod_{\substack{i \in \mathcal{V} \\ k \in \mathcal{L}}} \exp(y'_{i:k} \boldsymbol{\theta}_k^\top h_i(\mathbf{x})) \prod_{\substack{(i,j) \in \mathcal{E} \\ k,l \in \mathcal{L}}} \exp(y'_{i:k} y'_{j:l} \boldsymbol{\theta}_{kl}^\top \nu_{ij}(\mathbf{x})), \quad (2)$$

is the partition function. When using the CRF model, there are two main issues that need to be addressed: (i) How to set the value of the parameters $\boldsymbol{\theta}$; and (ii) How to perform inference in order to obtain the optimal labelling, *i.e.* the labelling with the maximum conditional probability $\Pr(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta})$. The latter issue has received great attention and several inference algorithms have been proposed in the literature (for an overview, see [26]). However, parameter estimation of a CRF still remains a challenging problem, with considerable progress being made in the past few years.

Recent methods for parameter estimation of a CRF can be broadly classified into three categories – maximum likelihood-based methods [12, 21, 25], large margin

[1] Note that we have assumed a pairwise CRF. However, this assumption is not restrictive since any CRF can be converted to an equivalent pairwise CRF, *e.g.* using a method similar to the one described in [31], and efficient inference algorithms are available for many such CRFs [13].

[2] Using the notation of [1].

based approaches [16, 27, 28], and other iterative methods [23, 32]. Owing to the intractability of computing the partition function $Z(\boldsymbol{\theta})$ for computer vision applications, maximum likelihood-based methods resort to using approximations, such as pseudo-likelihood [12], some form of local training [25], mode of the model distribution [21], or sampling [19]. While these likelihood approximation methods have shown encouraging results, they can lead to poor accuracy due to noisy estimates, as noted in [21, 25]. Methods using a large margin approach pose parameter estimation as a convex optimization problem. The convex problem is solved iteratively, and each iteration involves performing inference for each training image, which can be computationally expensive. By restricting themselves to a subset of random field models, Taskar *et al*. [28] and Szummer *et al*. [27] provide efficient solutions. Other large margin approaches [16] use the structured output regression formulation proposed by [29]. The algorithm employs a cutting-plane method to solve the quadratic optimization algorithm. The model parameters are updated using the most violated constraint (in this case, the labelling with the smallest cost value) in every iteration. Finding the exact most violated constraint is not tractable for random fields commonly encountered in computer vision, and hence approximation algorithms are used. Other iterative based methods are either limited to CRFs with a few hundred nodes, thus impractical for the labelling problems we consider [23], or require an initial model with pre-set parameters [32].

In summary, previous methods can lead to poor accuracy due to approximations, or are restricted to a subset of random field models. We aim to address these issues in this paper. To obtain an efficient and accurate learning scheme, we decompose the random field into tree-structured graphs, where each graph comprises of variable Y_i and its corresponding Markov blanket, which is set of its neighbours. This decomposition results in a large optimization problem in terms of the number of constraints. We reduce this problem to an equivalent convex problem with a small number of constraints, similar to the approach of [10]. An efficient method to solve it using stochastic gradient descent is then proposed. One of the main advantages of our method is the ease of training, as demonstrated in the experimental results.

Outline of the paper. In Section 2 we present the problem formulation, and briefly review two methods related to our work. Section 3 explains our piecewise large margin approach for parameter learning. Details of the optimization problem and the gradient descent approach are also given here. Implementation details, datasets and experimental results are shown in Section 4. A comparison with other parameter learning methods is also discussed. Section 4.3 presents a few generalizations of our model. Concluding remarks are provided in Section 5.

2. Preliminaries

We begin by formulating our approach for the CRF parameter estimation problem. The unary and pairwise potentials are as defined in (1). For example, in case of the image segmentation problem, the feature vectors for a vertex can be functions of the intensity, colour and texture, and those of an edge can be a difference of the feature vectors of the two vertices the edge connects.

Given a set of training data $\mathbf{X} = \{\mathbf{x}^m, m = 1, \cdots, M\}$, along with their ground truth labels $\mathbf{Y} = \{\mathbf{y}^m, m = 1, \cdots, M\}$, the problem of parameter estimation is to obtain a value for the parameter $\boldsymbol{\theta}$, such that the model assigns a high probability to the correct labelling and a low one to all possible incorrect labellings. In the context of foreground-background segmentation problem, an element of the training set, $\mathbf{x}^m$, corresponds to an image, and the ground truth labels contain binary values representing foreground or background at each pixel. The model is learnt such that we obtain a high probability to the correct segmentation and a low one to all other possible segmentations.

2.1. Pseudo-likelihood

The maximum likelihood estimate of the parameters $\hat{\boldsymbol{\theta}}$ (using equation (1)) is given by:

$$\begin{aligned}\hat{\boldsymbol{\theta}} &= \arg\max_{\boldsymbol{\theta}} \sum_{m=1}^{M} \sum_{\substack{i\in\mathcal{V}\\ k\in\mathcal{L}}} y^m_{i:k} \boldsymbol{\theta}_k^\top h_i(\mathbf{x}^m) \\ &\quad + \sum_{\substack{(i,j)\in\mathcal{E}\\ k,l\in\mathcal{L}}} y^m_{i:k} y^m_{j:l} \boldsymbol{\theta}_{kl}^\top \nu_{ij}(\mathbf{x}^m) - \log Z^m(\boldsymbol{\theta}), \quad (3)\end{aligned}$$

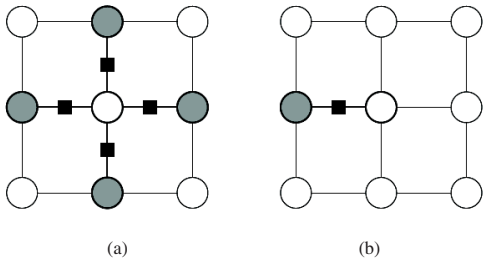
where m indexes over the training images and M denotes the number of training images. Solving this estimation problem for loopy random fields encountered in computer vision is intractable.

A common approach is to use pseudo-likelihood [2] to approximate the likelihood to overcome this issue [3, 12]. The estimation problem now becomes:

$$\hat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} \sum_{m=1}^{M} \sum_{\substack{i\in\mathcal{V}\\ k\in\mathcal{L}}} PL(i,k,\mathbf{x}^m). \quad (4)$$

Here $PL(\cdot)$ is the pseudo-likelihood, and is given by:

$$\begin{aligned}PL(i,k,\mathbf{x}^m) &= \sum_{k\in\mathcal{L}} y^m_{i:k} \boldsymbol{\theta}_k^\top h_i(\mathbf{x}^m) \\ &\quad + \sum_{\substack{j\in\mathcal{N}_i\\ k,l\in\mathcal{L}}} y^m_{i:k} y^m_{j:l} \boldsymbol{\theta}_{kl}^\top \nu_{ij}(\mathbf{x}^m) - z_i^m + b, \quad (5)\end{aligned}$$



(a) (b)

Figure 1. *Difference between pseudo-likelihood (*PL*), in (a), and piecewise pseudo-likelihood [25] (*PWPL*), in (b) is shown here. In* PL*, a variable is conditioned on all its neighbours in the Markov blanket, while in* PWPL *it is conditioned only on the neighbours within a single factor. (Figure taken from [25]).*

where

$$z_i^m = \log \sum_{y_i^m \in \mathcal{L}} \prod_{k \in \mathcal{L}} \exp(y_{i:k}^m \boldsymbol{\theta}_k^\top h_i(\mathbf{x}^m)) \prod_{\substack{j \in \mathcal{N}_i \\ k,l \in \mathcal{L}}} y_{i:k}^m y_{j:l}^m \boldsymbol{\theta}_{kl}^\top \nu_{ij}(\mathbf{x}^m), \quad (6)$$

is the local partition function, b is a constant, and $\mathcal{N}_i$ is the Markov blanket at vertex i, *i.e*. the set of its neighbours in the random field model. For example, in the 4-neighbourhood case used for CRF based image segmentation, the Markov blanket is the set of 4 pixels—above, below, left of, and right of a pixel i. This problem can be solved by gradient descent like approaches [12] or auto-regression [3]. One of the main advantages of using pseudo-likelihood is the asymptotic guarantee (*i.e*. as the size of the data tends to infinity) that its maximum matches that of the original likelihood. However, parameter learning methods using pseudo-likelihood are computationally expensive.

Another approach to approximate the likelihood estimation in (3) is to use the piecewise pseudo-likelihood (PWPL) model proposed by Sutton and McCallum [25]. Here, the likelihood is conditioned on all the variables in the factor graph associated with the variable. Figure 1 illustrates the difference between PWPL and pseudo-likelihood models. They show interesting results on linear-chain CRFs. However, it is not clear if this method generalizes to large random field problems, involving millions of variables, commonly occurring in computer vision.

2.2. Max-Margin Learning

Taskar *et al*. [28] proposed an alternative approach to learn the parameters of a random field discriminatively. Consider the logarithm of the probability in equation (1). It can be re-written according to the notation in [1] as:

$$\log \Pr(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta}) = \boldsymbol{\theta}\mathbf{F}\mathbf{y} - \log Z(\boldsymbol{\theta}), \quad (7)$$

where $\boldsymbol{\theta} = (\boldsymbol{\theta}_k; \boldsymbol{\theta}_{kl})$ with the operator (;) denoting vector concatenation. The vector $\mathbf{y}$ contains the labels of all the variables in the random field, and the matrix $\mathbf{F}$ is composed of unary and pairwise features, *i.e*. $h_i(\mathbf{x})$ and $\nu_{ij}(\mathbf{x})$. Given a training image[3] $(\mathbf{x}, \hat{\mathbf{y}})$, they maximize the margin of confidence in the true label assignment $\hat{\mathbf{y}}$ over all other assignments $\mathbf{y} \neq \hat{\mathbf{y}}$. In other words, the goal is to maximize the gain of the true labels $\hat{\mathbf{y}}$ over all the other labels $\mathbf{y}$ by solving the following quadratic program:

$$\max \ \gamma \quad \text{s.t.} \quad \boldsymbol{\theta}\mathbf{F}(\hat{\mathbf{y}} - \mathbf{y}) \geq \gamma \ell(\hat{\mathbf{y}}, \mathbf{y}); \ \ \|\boldsymbol{\theta}\|^2 \leq 1, \quad (8)$$

where the gain depends on the number of misclassified variables, $\ell(\hat{\mathbf{y}}, \mathbf{y})$. This formulation has two advantages: (i) It eliminates the partition function; and (ii) It is similar to the well-known soft-margin optimization problem. However, their iterative optimization involves performing inference to find $\mathbf{y}$ at every step of the algorithm, which is computationally intensive.

In our work we combine the benefits of using pseudo-likelihood and max-margin learning. We propose a discriminative max-margin piecewise learning using the pseudo-likelihood framework. We decompose the random field into distinct *pieces* (according to pseudo-likelihood), and treat each *piece* as an individual training sample. We then perform a discriminative learning over these samples and solve it efficiently, as described in the following section.

3. The Piecewise Model

The pseudo-likelihood term, given by $PL(\cdot)$ in equation (5), for each vertex i represents a tree-structured graph. We consider each of these graphs as an independent training example to learn the parameters. We denote the energy function on this tree-structured graph for a vertex i in vector form as[4]:

$$E^{\mathbf{i}}(\mathbf{y}) = \boldsymbol{\theta}^\top \mathbf{f}(\mathbf{i}, \mathbf{j}, \mathbf{x}, \mathbf{y}) + b, \quad (9)$$

where $\mathbf{i}$ is the set of all nodes in the tree-structured graph, $\boldsymbol{\theta} = (\boldsymbol{\theta}_k; \boldsymbol{\theta}_{kl}), \forall k, l \in \mathcal{L}$, and $\mathbf{j} = \{j | j \in \mathcal{N}_i\}$, the neighbours of the vertex i. The operator (;) denotes vector concatenation. Also, $\mathbf{f}(\mathbf{i}, \mathbf{j}, \mathbf{x}, \mathbf{y})$ is a vector formed by concatenating the unary and the pairwise features. The number of possible labellings for each pseudo-likelihood tree structure is $|\mathcal{L}|^{N_p}$, where N_p is the number of vertices in the tree (5 when using the 4-neighbourhood random field). Among these labellings, one of them is the ground truth labelling, which is referred to as the positive training example. All the other labellings form the negative example set, which is exponentially large. Let M_+ and M_- denote the number of positive and negative training examples respectively. The feature vectors corresponding to the m^{th} positive and the n^{th} negative training example are denoted by $\mathbf{f}_+^m$ and $\mathbf{f}_-^n$ respectively.

[3] For simplicity we describe this approach using one training image. It can be easily extended to multiple images easily, *e.g*. by concatenation.

[4] For brevity, we have dropped the index m over training images.

3.1. Parameter Learning

The parameter vector $\boldsymbol{\theta}$ and the bias b should ideally satisfy the following margin constraints:

$$\begin{aligned} \boldsymbol{\theta}^\top \mathbf{f}_+^m(i,\mathbf{j},\mathbf{x},\mathbf{y}) + b &\geq 1, \forall m \in \{1,\cdots,M_+\}, \\ \boldsymbol{\theta}^\top \mathbf{f}_-^n(i,\mathbf{j},\mathbf{x},\mathbf{y}) + b &\leq -1, \forall n \in \{1,\cdots,M_-\}. \end{aligned} \quad (10)$$

This ensures that the parameters discriminate between the positive and negative examples with respect to the quality function $E^{\mathbf{i}}(\cdot)$ in equation (9). The most discriminative parameter vector is obtained by maximizing the margin (which is equivalent to minimizing $\|\boldsymbol{\theta}\|^2$, the L^2 norm of the parameter vector). However, it is not always possible to separate the data by solving this hard-margin optimization problem. It is common to introduce slack variables in such cases [30]. The optimal parameter vector is now learnt by solving the following soft-margin optimization problem:

$$(\boldsymbol{\theta}^*, b^*) = \arg\min_{\boldsymbol{\theta},b} \frac{1}{2}\|\boldsymbol{\theta}\|^2 + C(\sum_m \xi_+^m + \sum_m \xi_-^n), \quad (11)$$

$$\text{subject to} \quad \boldsymbol{\theta}^\top \mathbf{f}_+^m(i,\mathbf{j},\mathbf{x},\mathbf{y}) + b \geq 1 - \xi_+^m, \forall m, \quad (12)$$

$$\boldsymbol{\theta}^\top \mathbf{f}_-^n(i,\mathbf{j},\mathbf{x},\mathbf{y}) + b \leq -1 + \xi_-^n, \forall n, \quad (13)$$

$$\xi_+^m \geq 0, \forall m \in \{1,\cdots,M_+\}, \quad (14)$$

$$\xi_-^n \geq 0, \forall n \in \{1,\cdots,M_-\}. \quad (15)$$

The tradeoff between the accuracy and regularization of the parameter vector is controlled by the user-defined constant $C \geq 0$. The slack variables ξ_+^m and ξ_-^n denote the hinge loss for positive and negative examples respectively.

The above convex problem is seemingly easy to solve. However, it cannot be solved efficiently because the inequality (13) specifies $|\mathcal{L}|^{N_p} - 1$ constraints for each tree structured training example. An iterative method proposed for the supervised case in [8] approximates this large problem using a small subset of constraints. The algorithm alternates between two steps: (i) Given a current estimate of the parameters, a subset of labellings that maximize $\boldsymbol{\theta}^\top \mathbf{f}_-^n$ for each negative example is found using max-sum belief propagation (BP) [17]; and (ii) Using the subset of labellings obtained in step (i), a new parameter vector and bias are computed. As noted in [10], this method is susceptible to local minima and is heavily dependent on obtaining a good initial estimate of the parameters. Recently, Kumar *et al*. [10] proposed an efficient algorithm to obtain the globally optimal solution to this problem. The key step in their approach is reducing the original large problem to an equivalent one with a polynomial number of constraints. We briefly review their main ideas relevant to our work in the next section.

3.2. Constraint Reformulation

The main bottleneck in solving problem (11) is the inequality (13), which specifies an exponential number of constraints (in terms of the number of nodes in the pseudo-likelihood graph). For example, using 4-neighbourhood in the stereo matching problem, where it is common to have a label set $\mathcal{L}$ composed of 25 disparity labels, each negative image leads to nearly 10 million constraints. The inequality (13) can be reduced to an equivalent set of $O(N_p|\mathcal{L}|^2)$ constraints, where N_p is the number of nodes in the pseudo-likelihood graph, and $|\mathcal{L}|$ is the number of labels [10]. Let t^n be an upper bound on the set of values $\boldsymbol{\theta}^\top \mathbf{f}_-^n$. We now reformulate inequality (13) as:

$$t^n + b \leq -1 + \xi_-^n, \;\; t^n \geq \boldsymbol{\theta}^\top \mathbf{f}_-^n(i,\mathbf{j},\mathbf{x},\mathbf{y}), \forall n. \quad (16)$$

The upper bound t^n on the values $\boldsymbol{\theta}^\top \mathbf{f}_-^n(i,\mathbf{j},\mathbf{x}), \forall n$, can be specified by a polynomial number of constraints. We define variables $S_{ij;y_{i:k}}^n$ using $|\mathcal{L}|$ constraints such that,

$$S_{ji;y_{i:k}}^n \geq \boldsymbol{\theta}_l^\top y_{j:l} h_j(\mathbf{x}) + \boldsymbol{\theta}_{kl}^\top y_{i:k} y_{j:l} \nu_{ij}(\mathbf{x}), \forall y_{j:l}, l \in \mathcal{L}, \quad (17)$$

where $j \in \mathcal{V}_p - \{i\}$ and $k \in \mathcal{L}$. The set of vertices in the pseudo-likelihood graph are denoted by $\mathcal{V}_p$. The smallest value of $S_{ji;y_{i:k}}^n$ which satisfies the above inequality is the message that j passes to i when performing max-sum BP on the pseudo-likelihood graph with potentials given in (1). Thus, the upper bound is given by:

$$t^n \geq \boldsymbol{\theta}^\top y_{i:k} h_i(\mathbf{x}) + \sum_{j \in \mathcal{V}_p - \{i\}} S_{ji;y_{i:k}}^n, \forall y_{i:k}, k \in \mathcal{L}, \quad (18)$$

The inequality (13) can be replaced by inequalities (16), (17), and (18) in the soft-margin optimization problem (11). The original optimization problem is now reformulated as:

$$(\boldsymbol{\theta}^*, b^*) = \arg\min_{\boldsymbol{\theta},b} \frac{1}{2}\|\boldsymbol{\theta}\|^2 + C(\sum_m \xi_+^m + \sum_m \xi_-^n), \quad (19)$$

$$\begin{aligned} \text{s.t.} \quad & \boldsymbol{\theta}^\top \mathbf{f}_+^m(i,\mathbf{j},\mathbf{x},\mathbf{y}) + b \geq 1 - \xi_+^m, \;\; \xi_+^m \geq 0, \forall m, \\ & t^n + b \leq -1 + \xi_-^n, \;\; \xi_-^n \geq 0, \forall n, \\ & t^n \geq \boldsymbol{\theta}^\top y_{i:k} h_i(\mathbf{x}) + \sum_{j \in \mathcal{V}_p - \{i\}} S_{ji;y_{i:k}}^n, \forall y_{i:k}, n, \\ & S_{ji;y_{i:k}}^n \geq \boldsymbol{\theta}_l^\top y_{j:l} h_j(\mathbf{x}) + \boldsymbol{\theta}_{kl}^\top y_{i:k} y_{j:l} \nu_{ij}(\mathbf{x}), \forall y_{j:l}, n. \end{aligned}$$

The number of constraints can be further reduced if the pairwise features $\nu_{ij}(\mathbf{x})$ are restricted to form a Potts model, as shown in [10]. In fact, this is applicable to other commonly used pairwise features such as truncated linear, and truncated quadratic models using the distance transform technique of [7]. The above problem is solved using the dual decomposition method in [10]. We follow an alternative method and solve the problem in the primal itself, as described below.

Stochastic Gradient Descent. The form of the problem (19) is very similar to the Support Vector Machine (SVM) learning problem. Many methods exist in literature to solve the SVM learning problem. We use a Stochastic Gradient Descent algorithm because of its efficiency [4]. It is an iterative algorithm to solve linear SVMs, where every iteration consists of choosing a random training sample, and updating the weight vector. The iterative updates are chosen according to the method described in [4]. The gradient at every step is computed by performing max-sum BP on the chosen training sample. Note that any other efficient online SVM solver can be used instead of this method. We chose to use this gradient based algorithm owing to its theoretical and empirical advantages when solving max-margin problems similar to (19). The reader is referred to [18] for a discussion on these advantages.

4. Experimental Results

We evaluated the proposed learning framework on two publicly available datasets, namely man-made structure database [12] and the Middlebury-2005 stereo vision data in [21]. We compare our results with those reported in these papers.

4.1. Man-made Structure Database

This dataset contains images of man-made structures, such as houses, cathedrals, buildings. The task is to detect these structures in natural scenes, assign *structured* or *non-structured* labels. The training and the test set contain 108 and 129 images respectively. The images are selected from the Corel image database, and are of size 256×384 pixels. Each image is divided into non-overlapping 16×16 pixel-blocks, and each such block is assigned a ground truth annotation (*structured* or *non-structured*) manually. In all, the training set contains $3,004$ structured and $36,269$ non-structured blocks. We represent each pixel-block as a node in the CRF, thus resulting in a 16×24 grid structure.

Feature computation. We use the feature set described in [12]. The unary feature at a node (pixel-block) i (*i.e.* $h_i(\mathbf{x})$) is computed using pixel windows over the block. Three scales (16×16, 32×32, and 64×64 pixel windows) are used, and in each scale, three moment and two orientation based features are computed. Two features are additionally chosen from these multiscale features using highest peaks from the feature histograms. A 14-dimensional vector is composed by taking the first two moment and orientation based features at each scale, and the two additional 'peak' features. The unary feature vector contains the 14 moment and orientation features, their squares and all their pairwise products. Thus, $h_i(\mathbf{x})$ is a 119-dimensional unary feature vector. The pairwise feature vector $\nu_{ij}(\mathbf{x})$ is a difference of unary feature vectors $h_i(\mathbf{x})$ and $h_j(\mathbf{x})$. We refer the reader to [12] for more details on the feature computation.

(a) (b) (c)

Figure 2. *Qualitative results on the man-made structure database. We show (a) the original image; (b) result of [12]; and (c) result of our method on three sample images from this database. The white squares overlaid on the image denote the presence of a structure. Our results correspond to an average false positive per image of 1.40. It can be observed that our performance is comparable to, if not better than, [12].*

Results. The weight vectors corresponding to the unary and pairwise features have 119-dimensions each. These were learnt using our piecewise model (§3). The algorithm was run until convergence (on average 120 iterations, depending on the initialization). The unary parameters are used as is, but the pairwise terms are truncated using a common approximation [20] such that graph cut inference is possible [5, 9]. The qualitative results are shown in Figure 2. It can be observed that our performance is comparable to the state-of-the-art results [12] on this dataset. Table 1 shows a quantitative evaluation of our method in terms of false positive and detection rates. We obtain a similar false positive rate and better detection percentage compared to [12]. However, our approach is computationally efficient. Our training procedure takes 409 seconds to converge compared to 627 seconds of their method. Furthermore, our approach can be easily generalized to multi-class problems, as shown in the following section.

4.2. Middlebury-2005 Dataset

This dataset contains 9 stereo pairs (left and right images constitute a pair) in all. The problem is to compute the disparity (or correspondence) between the left and the right image. Since ground truth disparities are not available for three of the pairs (Computer, Drumsticks, Dwarves), they were discarded for this performance evaluation. We used the other images, namely, Art, Books, Dolls, Laun-

Method	Art	Books	Dolls	Laundry	Moebius	Reindeer	*Average*
Grid structure in [16]	14.66	19.12	12.70	19.16	10.88	11.72	14.71
Long-range in [16]	**12.11**	**15.68**	**12.14**	**15.82**	**10.80**	15.26	13.64
Our method (*without long-range edges*)	12.94	16.24	12.21	16.72	10.82	**11.10**	**13.34**

Table 2. *Quantitative results showing the error rates measured as the percentage of bad pixels in the non-occluded regions on the Middlebury-2005 database. We compare our results with the models using the standard loss function (*i.e. *ignore the pixels in the occluded region when comparing with ground truth result) in [16]. 'Grid structure' refers to the model without long-range edges, and 'Long-range' is the one with these edges. Average denotes the average error rate over all the images. Bold fonts indicate the best performance (or lowest error rate). Note that our method shows better results than 'Grid structure' on all the images, and shows comparable performance to 'Long-range' on most of the images.*

Method	FP per image	DR %
MRF shown in [12]	2.36	57.20
DRF [12]	**1.37**	70.50
DRF [11]	1.76	72.54
Our method	1.40	**72.60**

Table 1. *Quantitative results on the man-made structure database. We show the average False Positive (*FP*) and Detection Rates (*DR*) on the test set containing 129 images. A comparison with both the Discriminative Random Field (*DRF*) methods proposed in [12] and [11] is also shown. Bold fonts indicate the lowest false positive error rate or the highest detection rate. Note that our performance is comparable to these methods. In fact, we provide a better false positive (per image) measure and similar detection rate accuracy. However, our method is computationally efficient and scales well to multi-class problems.*

dry, Moebius, Reindeer, in a leave-one-out training framework (*i.e*. for each stereo pair problem, we train the model on all the other pairs). As noted in [16], these scenes are more challenging than the previous ones on the Middlebury Stereo Evaluation page [22]. We use the feature set described in [16]. The unary terms are composed of Birchfield-Tomasi matching costs for each disparity label, and the pairwise term is a difference of disparity labels. The number of disparity levels for each image pair is identical to that used in [16]. Inference is performed on the learnt energy function using the α-expansion move making algorithm [6].

Results. A quantitative evaluation of our method is shown in Table 2. We perform better than the 'Grid structure' model proposed in [16]. As we do not use long-range edges in our approach, we perform slightly worse than the 'Long-range' model. We believe including these edges in our energy function will significantly improve the results.

4.3. Discussion

In the formulation discussed so far, we restricted ourselves to decomposing a CRF into sub-graphs corresponding to the Markov blanket of a single pixel. This is not an inherent limitation of the framework. Any other tree structured sub-graph, including scan-lines, can be solved in the same way. In these case, our approach will efficiently find the most violating constraint using the trick proposed by [10].

Under our formulation, each Markov blanket (or sub-graph) is an individual training exemplar, and a unique slack variable corresponds to each sub-graph, while existing max-margin approaches treat the entire image as a single exemplar [27]. Of the two approaches, ours should be more robust to errors in data annotation — for example consider the problem of learning models for image segmentation. In these problems [12, 24], annotation of the training and test set must be done by hand, and it is common to find inaccurate ground truth labelling in large regions of the image, particularly near object boundaries. Such data is often inseparable in these regions, and global approaches such as [27] can only learn a limited amount from these images. By way of contrast, our decomposition of the image into sub-graphs allows us to disregard some of these mislabelled exemplars while learning from the remainder of the image.

5. Summary

This paper presents a novel method to compute the parameters of a Conditional Random Field model. Our method is: (i) applicable for large random fields commonly used in computer vision; (ii) efficient in terms of memory and computation; and (iii) easily applicable for multi-label problems. We show the effectiveness of our approach on two publicly available datasets. Future work includes extending this model to learn the structure of random fields.

Acknowledgements. We thank S. Kumar and Y. Li for help with datasets used in this paper. This work was supported by the EPSRC research grant EP/C006631/1(P), the IST Programme of the European Community, under the PASCAL2 Network of Excellence, IST-2007-216886. P. H. S. Torr is in receipt of Royal Society Wolfson Research Merit Award.

References

[1] D. Anguelov, B. Taskar, V. Chatalbashev, D. Koller, D. Gupta, G. Heitz, and A. Ng. Discriminative learn-

ing of markov random fields for segmentation of 3d scan data. In *CVPR*, pages 169–176, 2005.

[2] J. Besag. Statistical analysis of non-lattice data. *Journal of the Royal Statistical Society. Series D*, 24(3):179–195, 1975.

[3] A. Blake, C. Rother, M. Brown, P. Perez, and P. H. S. Torr. Interactive image segmentation using an adaptive GMMRF model. In *ECCV*, volume 1, pages 428–441, 2004.

[4] A. Bordes, L. Bottou, and P. Gallinari. SGDQN: Careful quasi-newton stochastic gradient descent. *JMLR*, 10:1737–1754, 2009.

[5] Y. Boykov and V. Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *PAMI*, 26(9):1124–1137, 2004.

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

[7] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient belief propagation for early vision. In *CVPR*, volume 1, pages 261–268, 2004.

[8] P. F. Felzenszwalb, D. McAllester, and D. Ramanan. A discriminatively trained, multiscale, deformable part model. In *CVPR*, 2008.

[9] V. Kolmogorov and R. Zabih. What energy functions can be minimized via graph cuts? *PAMI*, 26(2):147–159, 2004.

[10] M. P. Kumar, P. H. S. Torr, and A. Zisserman. Efficient discriminative learning of parts-based models. In *ICCV*, 2009.

[11] S. Kumar and M. Herbert. Discrimative fields for modelling spatial dependencies in natural images. In *NIPS*, 2003.

[12] S. Kumar and M. Herbert. Discriminative random fields: A discriminative framework for contextual interaction in classification. In *ICCV*, 2003.

[13] L. Ladicky, C. Russell, P. Kohli, and P. H. S. Torr. Associative hierarchical crfs for object class image segmentation. In *ICCV*, 2009.

[14] J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labelling sequence data. In *ICML*, pages 282–289, 2001.

[15] S. Z. Li. *Markov Random Field Modeling in Computer Vision*. Springer-Verlag, 2000.

[16] Y. Li and D. P. Huttenlocher. Learning for stereo vision using the structured support vector machine. In *CVPR*, 2008.

[17] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, 1998.

[18] N. Ratliff, J. A. Bagnell, and M. Zinkevich. (online) subgradient methods for structured prediction. In *AISTATS*, 2007.

[19] S. Roth and M. J. Black. On the spatial statistics of optical flow. *IJCV*, 74(1):33–50, 2007.

[20] C. Rother, S. Kumar, V. Kolmogorov, and A. Blake. Digital tapestry. In *CVPR*, volume 1, pages 589–596, 2005.

[21] D. Scharstein and C. Pal. Learning conditional random fields for stereo. In *CVPR*, 2007.

[22] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithm. *IJCV*, 47:7–42, 2002.

[23] M. Schmidt, K. Murphy, G. Fung, and R. Rosales. Structure learning in random fields for heart motion abnormality detection. In *CVPR*, 2008.

[24] J. Shotton, J. M. Winn, C. Rother, and A. Criminisi. TextonBoost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation. In *ECCV*, volume 1, pages 1–15, 2006.

[25] C. Sutton and A. McCallum. Piecewise pseudolikelihood for efficient training of conditional random fields. In *ICML*, pages 863–870, 2007.

[26] R. Szeliski, R. Zabih, D. Scharstein, O. Veksler, V. Kolmogorov, A. Agarwala, M. F. Tappen, and C. Rother. A comparative study of energy minimization methods for Markov random fields. In *ECCV*, volume 2, pages 16–29, 2006.

[27] M. Szummer, P. Kohli, and D. Hoiem. Learning crfs using graph cuts. In *ECCV*, pages 582–595, 2008.

[28] B. Taskar, C. Guestrin, and D. Koller. Max-margin Markov networks. In *NIPS*, 2003.

[29] I. Tsochantaridis, T. Hofmann, T. Joachims, and Y. Altun. Support vector machine learning for interdependent and structured output spaces. In *ICML*, pages 823–830, 2004.

[30] V. Vapnik. *The Nature of Statistical Learning Theory*. Springer, 1995.

[31] J. S. Yedidia, W. T. Freeman, and Y. Weiss. Bethe free energy, kikuchi approximations, and belief propagation algorithms. Technical Report TR2001-16, MERL, 2001.

[32] L. Zhang and S. Seitz. Estimating optimal parameters for mrf stereo from a single image pair. *PAMI*, 29(2):331–342, 2007.