On Parameter Learning in CRF-based Approaches to Object Class Image Segmentation

Sebastian Nowozin[1], Peter V. Gehler[2], and Christoph H. Lampert[3]

[1] Microsoft Research Cambridge, UK,
[2] ETH Zurich, Switzerland,
[3] Institute of Science and Technology, Austria.

Abstract. Recent progress in per-pixel object class labeling of natural images can be attributed to the use of multiple types of image features and sound statistical learning approaches. Within the latter, Conditional Random Fields (CRF) are prominently used for their ability to represent interactions between random variables. Despite their popularity in computer vision, *parameter learning* for CRFs has remained difficult, popular approaches being cross-validation and *piecewise training*.
In this work, we propose a simple yet expressive tree-structured CRF based on a recent hierarchical image segmentation method. Our model combines and weights multiple image features within a hierarchical representation and allows simple and efficient globally-optimal learning of $\approx 10^5$ parameters. The tractability of our model allows us to pose and answer some of the open questions regarding parameter learning applying to CRF-based approaches. The key findings for learning CRF models are, from the obvious to the surprising, i) multiple image features always help, ii) the limiting dimension with respect to current models is the amount of training data, iii) piecewise training is competitive, iv) current methods for max-margin training fail for models with many parameters.

1 Introduction

Computer vision increasingly addresses high-level vision tasks such as scene understanding, object class image segmentation, and class-level object recognition. Two drivers of this development have been the abundance of digital images and the use of statistical machine learning models. Yet, it remains unclear what classes of models are suited best to these tasks. *Random field models* [1, 2] have found many applications due to their ability to concisely express dependencies between multiple random variables, making them attractive for many high-level vision tasks. *Parameter learning* in these rich models is essential to find from a large set of possible candidates the model instance that best explains the observed data and generalizes to unseen data. Despite the importance of parameter learning, current applications of random fields in computer vision sidestep many issues, making assumptions that are intuitive, but largely heuristic. The reason for this gap between principled modeling and use of heuristics is the intractability of many random field models, which makes it necessary use approximations.

To shed light on the currently used practices we take the task of object class image segmentation and propose a simple, yet expressive hierarchical multi-scale CRF model in which parameter learning can be analyzed in isolation.

In our model, parameter learning *is* tractable, allowing us to experimentally address the following open questions regarding conditional random fields for object class image segmentation: 1. What is the effect of combining multiple image features on the resulting model performance? 2. How does the size of the training set and the accuracy of optimizing the training objective influence the resulting performance? 3. Is it better to learn the models part-by-part (*piecewise*) or jointly? 4. Does maximum margin training offer an advantage over maximum likelihood estimation?

Outline. We first describe random fields in Section 2. In Section 3 we discuss the current computer vision literature on parameter learning in CRFs. Our novel model is introduced in Section 4 and we report experiments in Section 5.

2 Learning Random Fields

In this section we review basic results about random field models, factor design and define the problems that need to be solved to perform prediction and parameter learning.

2.1 Random Field Models and Factor Graphs

Discrete random field models, also known as Markov networks, are a popular model to describe interacting variables [2]. In particular we will focus on *conditional random fields* (CRF) [3, 4]. For a set $Y = \{Y_1, \dots, Y_V\}$ of random variables, each taking values in a label set $\mathcal{Y} = \{1, \dots, C\}$, a set of observation variables $X = \{X_1, \dots, X_W\}$, and a parameter vector $\boldsymbol{w} \in \mathbb{R}^D$, a conditional random field specifies a probability distribution as

$$p(Y = \boldsymbol{y} | X = \boldsymbol{x}, \boldsymbol{w}) = \frac{1}{Z(\boldsymbol{x}, \boldsymbol{w})} \exp(-E(\boldsymbol{y}; \boldsymbol{x}, \boldsymbol{w})), \tag{1}$$

where $E(\boldsymbol{y}; \boldsymbol{x}, \boldsymbol{w})$ is an *energy function* and $Z(\boldsymbol{x}, \boldsymbol{w}) = \sum_{\boldsymbol{y} \in \mathcal{Y}^V} \exp(-E(\boldsymbol{y}; \boldsymbol{x}, \boldsymbol{w}))$ is a normalizing constant known as *partition function* [1]. The energy function is specified in terms of *log-potential functions*, also known as *log-factors*. Let $\mathcal{F} \subseteq 2^V \times 2^W$ be a set of subsets of the variables. Then $\mathcal{F}$ specifies a factorization of (1), or equivalently an additive decomposition of the energy function as

$$E(\boldsymbol{y}; \boldsymbol{x}, \boldsymbol{w}) = \sum_{F \in \mathcal{F}} E_F(\boldsymbol{y}_F; \boldsymbol{x}_F, \boldsymbol{w}), \tag{2}$$

where $\boldsymbol{y}_F$ and $\boldsymbol{x}_F$ denote the restrictions of Y and X to the elements appearing in F, respectively. The energy function E_F operates only on the variables appearing in the set F.

The factorization is often given implicitly by means of an undirected graphical model [1]. For all practical purposes, it is more convenient to directly specify $\mathcal{F}$ used in (2) in terms of a *factor graph* [5]. For each element $F \in \mathcal{F}$, a factor graph contains a *factor node* (drawn as ■), which is connected to all *variable nodes* (drawn as ◯) that are members of F. The factor graph compactly defines $\mathcal{F}$ in (2). An example is shown in Figure 2 (page 6).

In order to fully specify the random field model, the form of the individual terms $E_F(\boldsymbol{y}_F; \boldsymbol{x}_F, \boldsymbol{w})$ in the summation (2) has to be defined. Each term corresponds to one factor F in the factor graph and specifies the local interactions between a small set of random variables. In practice the different factors have one of a few different roles such as incorporating observations into the model or enforcing a consistent labeling of the variables. Therefore, *clique templates* [4] (also known as *parameter tying*) are used, replicating parameters across groups of factors with the same purpose. We let $T = \{1, \dots, |T|\}$ denote a small set of different factor purposes and split the parameter vector as $\boldsymbol{w} = (\boldsymbol{w}_1^\top, \dots, \boldsymbol{w}_{|T|}^\top)^\top$, then the energy of each factor can be written as $E_F^{t(F)}(\boldsymbol{y}_F; \boldsymbol{x}_F, \boldsymbol{w}_{t(F)})$, where $t(F)$ is the type of the factor. As an additional notation, let $\boldsymbol{\mu}_F \in \{0,1\}^{\mathcal{Y}^F}$ be a set of binary indicator variables indexed by $\boldsymbol{y}_F \in \mathcal{Y}^F$ and let $\mu_F(\boldsymbol{y}_F) \in \{0,1\}$ be one if $Y_F = \boldsymbol{y}_F$, zero otherwise. Let the scalar $\theta_{F,\boldsymbol{y}_F}(\boldsymbol{x}_F, \boldsymbol{w}_{t(F)}) = E_F^{t(F)}(\boldsymbol{y}_F; \boldsymbol{x}_F, \boldsymbol{w}_{t(F)})$ be the evaluated energy when $Y_F = \boldsymbol{y}_F$. By suitably concatenating all μ_F, θ_F we can rewrite the energy (2) as the inner product $\langle \boldsymbol{\theta}(\boldsymbol{x}, \boldsymbol{w}), \boldsymbol{\mu} \rangle$. Because this form is linear, the distribution (1) is an *exponential family distribution* [1] with *sufficient statistics* $\boldsymbol{\mu}$ and so called *canonical parameters* $\boldsymbol{\theta}(\boldsymbol{x}, \boldsymbol{w})$.

What is left to do is to give the form of the *feature function* $\boldsymbol{\theta}_F(\boldsymbol{x}_F, \boldsymbol{w}_{t(F)})$ for all factor types $t(F) \in T$. As we will see below an important requirement for efficient parameter learning is that the energy function is *linear* in $\boldsymbol{w}$. The energy function $E_F^{t(f)}$ related to one factor F is already a linear function in the output of the feature function $\boldsymbol{\theta}_F : \mathcal{X}^F \times \mathbb{R}^{D_{t(F)}} \to \mathbb{R}^{\mathcal{Y}^F}$. Therefore, the energy will only be linear in $\boldsymbol{w}$ if we make the feature function also a linear function in its second argument $\boldsymbol{w}$. To this end, we will write $\boldsymbol{\theta}_F(\boldsymbol{x}_F, \boldsymbol{w}_{t(F)}) = H_F^{t(F)}(\boldsymbol{x}_F)\boldsymbol{w}_{t(F)}$, where $H_F^{t(F)}(\boldsymbol{x}_F)$ is a linear map from $\mathbb{R}^{D_{t(F)}}$ onto $\mathbb{R}^{\mathcal{Y}^F}$, mapping the parameters $\boldsymbol{w}_{t(F)}$ to energies. Due to the identity $E_F^{t(F)}(\boldsymbol{y}_F; \boldsymbol{x}_F, \boldsymbol{w}_{t(F)}) = \langle \boldsymbol{\theta}_F(\boldsymbol{x}_F, \boldsymbol{w}_{t(F)}), \mu_F(\boldsymbol{y}_F) \rangle = \langle H_F^{t(F)}(\boldsymbol{x}_F)\boldsymbol{w}_{t(F)}, \mu_F(\boldsymbol{y}_F) \rangle = \langle \boldsymbol{w}_{t(F)}, \phi(\boldsymbol{x}_F, \boldsymbol{y}_F) \rangle$ we can make explicit the linearity in *both* $\boldsymbol{w}_{t(F)}$ and $\mu_F(\boldsymbol{y}_F)$, where $\phi(\boldsymbol{x}_F, \boldsymbol{y}_F) = \mu_F(\boldsymbol{y}_F) H_F^{t(F)}(\boldsymbol{x}_F)$ is also known as joint feature map in the CRF literature. *Why is this important?* Linearity in $\boldsymbol{w}$ leads to convex learning problems (so that local optimality implies global optimality); linearity in μ leads to an exponential family distribution.

2.2 Inference Problems

The random field model is now fully specified and we can consider inference and learning tasks. The two tasks of our interest are the test-time prediction task, labeling an image with a likely segmentation, and the parameter learning task in which we have fully annotated training data and want to estimate a good parameter vector $\boldsymbol{w}$. In computer vision, predictions are most often made by solving an energy minimization problem as follows.

Problem 1 (MAP-MRF Labeling Problem). Given an observation $\boldsymbol{x}$ and a parameter vector $\boldsymbol{w}$, find the $\boldsymbol{y} \in \mathcal{Y}^V$ that maximizes the aposteriori probability $p(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{w})$, that is, solve

$$\boldsymbol{y}^* = \operatorname*{argmax}_{\boldsymbol{y}\in\mathcal{Y}^V} p(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{w}) = \operatorname*{argmin}_{\boldsymbol{y}\in\mathcal{Y}^V} E(\boldsymbol{y};\boldsymbol{x},\boldsymbol{w}).$$

For general factor graphs this problem is NP-hard [2].

To address the parameter learning problem we use the principle of *maximum likelihood* to find a point estimate for $\boldsymbol{w}$. We now define the estimation problem but in Section 5.4 make connections to maximum-margin procedures.

Problem 2 (Regularized CML Estimation (CMLE)). Given a set of N fully observed independent and identically distributed (iid) instances $\{(\boldsymbol{x}_n,\boldsymbol{y}_n)\}_{n=1,\dots,N}$ and given a prior $p(\boldsymbol{w})$ over $\mathbb{R}^D$, find $\boldsymbol{w}^* \in \mathbb{R}^D$ with maximum regularized conditional likelihood, that is, solve

$$\begin{aligned}\boldsymbol{w}^* &= \operatorname*{argmax}_{\boldsymbol{w}\in\mathbb{R}^D}\; p(\boldsymbol{w}) \prod_{n=1}^{N} p(\boldsymbol{y}_n|\boldsymbol{x}_n,\boldsymbol{w}) \\ &= \operatorname*{argmax}_{\boldsymbol{w}\in\mathbb{R}^D} \left[\frac{1}{N}\log p(\boldsymbol{w}) - \frac{1}{N}\sum_{n=1}^{N}\left(E(\boldsymbol{y}_n;\boldsymbol{x}_n,\boldsymbol{w}) + \log Z(\boldsymbol{x}_n,\boldsymbol{w})\right)\right]. \qquad (3)\end{aligned}$$

From the fact that $E(\boldsymbol{y}_n;\boldsymbol{x}_n,\boldsymbol{w})$ is a linear function in $\boldsymbol{w}$ it follows [2, section 20.3.2] that the log-likelihood (3) is a concave differentiable function in $\boldsymbol{w}$ and therefore $\boldsymbol{w}^*$ can be found using gradient descent. In the case that $\log p(\boldsymbol{w})$ is strictly concave in $\boldsymbol{w}$, (3) has a unique maximizer. Despite this, it is hard to solve Problem 2 for general factor graphs. The reason is that evaluating (3) for a given $\boldsymbol{w}$ requires computing the partition function $Z(\boldsymbol{x}_n,\boldsymbol{w})$ for each sample, a task involving summation of an exponential number of terms.

In our model presented in Section 4 we therefore consider *tree-structured* factor graphs. These are by definition acyclic and the partition function can be computed efficiently by rearranging the exponential number of terms as a recursion along the tree. This algorithm for computing $\log Z(\boldsymbol{x}_n,\boldsymbol{w})$ and $\nabla_{\boldsymbol{w}} \log Z(\boldsymbol{x}_n,\boldsymbol{w})$ is known as *sum-product algorithm* [5]. Likewise, for tree-structured factor graphs we can efficiently solve the MAP-MRF problem by the *max-product* algorithm.

3 Literature Review

Literature on CRF-based object class segmentation. CRF-based approaches to object class image segmentation can be distinguished by what kind of factors they use (unary, pairwise, higher-order factors), the model capacity, that is, how many free parameters they have, how the model structure is defined (pixel grid, superpixels, etc.) and how the parameter learning is performed.

Regarding the *representation*, the main lines are pixel- or pixel-blocks based approaches [6–11], superpixel-based representations [12–14], superpixel hierarchies [15, 16], and hybrid (both pixels and superpixels) representations [17–19].

For parameter learning, most works cited before use a form of piecewise training or cross validation on one to five hand-chosen parameters. Models in which joint parameter learning is performed are rare and often use an approximation, such as loopy BP in [14, 11], pseudolikelihood in [9], and contrastive divergence in [10]. Pincipled max-margin learning is performed in [6, 12, 19].

Literature on comparing learning methods for CRFs. Because we address the effect of different parameter learning methods, let us summarize existing comparisons of parameter learning methods. Kumar et al. [20] compare a large number of approximate CRF learning methods on a synthetic binary low-level vision task with four parameters. Similar experiments on the same dataset have been done by Korc and Förstner [21]. The excellent study of Parise and Welling [22] compares learning methods for generative binary non-vision MRF models with fixed, non-replicated structure. Finley and Joachims [23] compare learning methods for intractable MRF models advocating max-margin learning on relaxations.

Importance of tree-based models. Many early models for low-level vision were based on tree-structured generative MRFs (for an extensive survey see [24]), where the structure of the tree is fixed and simple, such as a quad-tree on a 2D grid. The use of tree-structured models for high-level vision tasks is much less common. One reason is that we now have efficient algorithms for MAP inference for certain potential functions for graphs of arbitrary structure. This offers more modeling freedom on the graph structure while restricting the potential function class. But recently there seems to be reconsideration of tree-based hierarchical models for high-level vision tasks where the tree structure is *adapted to the image content* [15, 16, 25]. Infact, even the more complex hybrid models listed above [17–19] base their multi-scale structure on a hierarchical tree of superpixels. Whereas obviously tree-based models are a restricted model class, the ability to learn arbitrary potential functions and the adapted nature of the tree structure to the image content offer drastic improvements over the early tree-based models considered before [24].

Lim et al. [25] is closest to our approach: a segmentation hierarchy is used as a multi-scale model for object class image segmentation. For each image region a linear classifier is learned, using features derived from the hierarchy. The main drawbacks of the otherwise sensible approach are the lack of pairwise interactions between image regions and the use of an adhoc test-time prediction function.

4 Model

We now define a tree-structured model for object class image segmentation. The model is naturally multi-scale and adapted to the image content. Due to

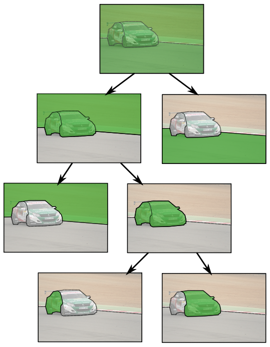

Fig. 1. Illustration of a hierarchical UCM segmentation. The hierarchy ranges from a superpixel partitioning at the leaf level to the entire image at the root. Each node's image region is shaded in green. (Figure best viewed in color.)

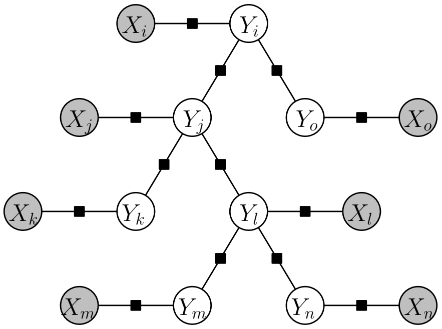

Fig. 2. Tree-structured factor graph CRF induced by the hierarchical segmentation. Each shaded segment r in Figure 1 has an observation variable X_r (drawn shaded) and a class variable Y_r. Factors (drawn as ■) encode interactions.

its tree structure, test-time image labeling as well as joint parameter learning are tractable. The tractability allows us to answer for the first time important questions regarding modeling choices, such as: What is the required image granularity for object class image segmentation? How to parametrize and learn the factors? What limits the current model performance? Is joint parameter learning superior to piecewise training?

The model is based on the recent *ultrametric contour maps* (UCM) hierarchical segmentation method of Arbeláez [26]. We use the UCM segmentation to define a tree structured factor graph. The factors are then suitably parametrized such that parameter estimation from training data can be performed. This idea is illustrated in Figures 1 and 2. In Figure 1 we illustrate the output of the UCM method: a segmentation tree that recursively partitions the image into regions. The leaves of the segmentation tree form a superpixel segmentation of the image, whereas interior nodes represent larger image regions. Ideally object instances – such as the car in the Figure 1 – are eventually represented by a single interior node. We use the structure of the segmentation tree to define a factor graph as shown in Figure 2. The shaded nodes correspond to image information observed for each image region, whereas the white nodes represent the class variables to be predicted, one for each region. The factor nodes (drawn as ■) link both observation and class variables, as well as pairs of class variables.

Because the hierarchical model structure is based on the UCM segmentation, it is naturally adapted to the image content. Moreover, it is a multi-scale representation of the image [26]. Our factor-graph can concisely represent a probability distribution over all possible labelings.

In next three subsections we discuss the choice of superpixel granularity, how to parametrize factors and how to perform training and prediction.

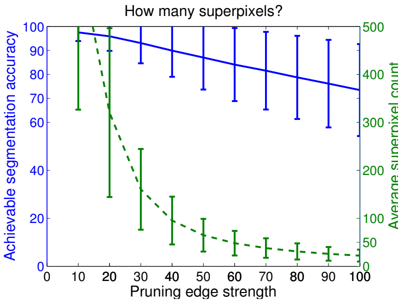

Fig. 3. Upper bound on the achievable VOC 2009 segmentation accuracy as a function of the preserved UCM edge strength. The left axis (solid, blue) shows the accuracy, the right axis (dashed, green) shows the mean number of superpixels per image. For each curve one unit standard deviations over the 749 training images is shown.

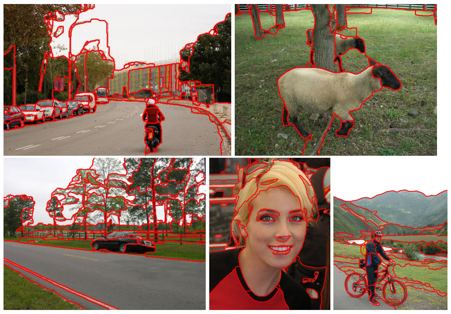

Fig. 4. Visualization of the superpixels of the hierarchical segmentation. Shown are examples from the VOC 2009 segmentation set, with the chosen edge pruning parameter of 40, leading to an average of ≈ 100 superpixels and ≈ 200 tree nodes per image.

4.1 Experiment: How many Superpixels?

When using a fixed precomputed representation of the image such as superpixels, it is fair to ask how much representational power is lost in the process: because we associate one discrete random variable with each superpixel, an error on this representational level cannot be corrected later.

To determine this trade-off, we produce UCM segmentations using the code of Arbeláez [26] for the 749 images in the PASCAL VOC 2009 segmentation challenge [27]. By thresholding the obtained UCM maps at increasing values we obtain a set of successively coarser hierarchical segmentations. For each threshold we evaluate the maximum achievable accuracy if we could label all leaves of the segmentation tree knowing the ground truth pixel labeling.

The results are shown in Figure 3. Even with a relatively small average number of superpixels the segmentation accuracy is above 70%. While this number appears to be low, it can be put into perspective by recognizing that the currently best state-of-the-art segmentation models applied to the VOC 2009 data set – including non-CRF approaches and methods trained on substantially more training data – achieve $25 - 36\%$ using the same evaluation measure [27]. Gould et al. [13] carried out a similar experiment on the MSRC and Sowerby data sets, and their results agree with our observations. For the following experiments we choose a pruning edge strength of 40, yielding an average of ≈ 100 superpixels per image and a maximum achievable accuracy of ≈ 90%. For this choice, Figure 4 shows typical example segmentations for the VOC 2009 images. For each image shown, the achievable accuracy is between 89.7% and 90.3%.

4.2 Features and Factors

We now describe how to parametrize the factors used in our model, starting with the unary observation factors.

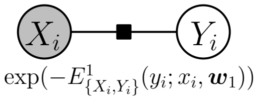

Fig. 5. Unary energy $E^1_{\{X_i,Y_i\}}(y_i;x_i,\boldsymbol{w}_1)$.

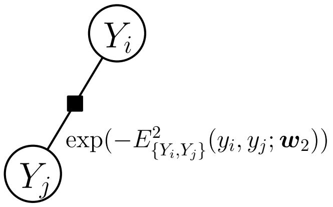

Fig. 6. Pairwise data-independent energy $E^2_{\{Y_i,Y_j\}}(y_i,y_j;\boldsymbol{w}_2)$.

Unary observation factors. The most important factors, the unary observation factors, describe the interaction between the image content and the variables of interest. We use multiple image features representing appearance statistics based on shape, color and texture to span a rich feature space describing an image region. As shown in Figure 5 and described in Section 2.1, the unary energy takes the following general form

$$E^1_{\{X_i,Y_i\}}(y_i;x_i,\boldsymbol{w}_1) = \langle \theta^1_{\{X_i\}}(x_i,\boldsymbol{w}_1), \boldsymbol{\mu}_{\{Y_i\}} \rangle = \langle H^1_{\{X_i\}}(x_i)\boldsymbol{w}_1, \boldsymbol{\mu}_{\{Y_i\}} \rangle.$$

Within this form, we define $H^1_{\{X_i\}}(x_i)$ as the concatenation of multiple image features. In particular, we define $H^1_{\{X_i\}}(x_i) = (f_{\text{SIFT}}(x_i), f_{\text{QHOG}}(x_i), f_{\text{QPHOG}}(x_i), f_{\text{STF}}(x_i))^\top$, where each f_a is an image feature related to the image region associated with X_i. As image features $f_a : \mathcal{X} \to \mathbb{R}^{D_a}$ we use the following: $a \in A = \{\text{SIFT}, \text{QHOG}, \text{QPHOG}, \text{STF}\}$, where SIFT are normalized bag-of-words histograms of quantized scale-invariant feature points ($D_{\text{SIFT}} = 512$). The QHOG features are soft-quantized histogram of oriented gradient vectors of the image content within a bounding box of the image region X_i ($D_{\text{QHOG}} = 512$). Similarly, the QPHOG features are soft-quantized pyramid of histogram of oriented gradient features of the black-and-white mask describing the image region X_i ($D_{\text{QPHOG}} = 512$). The STF features are normalized histograms of semantic texton forest responses within the image regions [28] ($D_{\text{STF}} = 2024$). For the above features $\boldsymbol{w}_1 \in \mathbb{R}^{D \times \mathcal{Y}}$, where $D = \sum_{a \in A} D_a = 3560$, such that $\boldsymbol{w}_1$ in total has $C \cdot D$ elements. The SIFT and STF features model general image statistics in the region X_i, whereas the QHOG and QPHOG features are responses to a template of shapes and appearances obtained by clustering the training data. If the hierarchical segmentation contains a region that describes an object instance, we hope to obtain a high response in these features. More details regarding the features used are available in the supplementary materials.

Data-independent pairwise factor. The pairwise factor shown in Figure 6 models the interaction of labels (Y_i, Y_j), where i and j form a children-parent pair in the hierarchical segmentation. If for example, y_i is labeled with a class, then y_j is likely to be labeled with the same class. We consider two possible energies of the form shown in Figure 6, the first one having the commonly used form

$$E^{2,P}_{\{Y_i,Y_i\}}(y_i, y_j; \boldsymbol{w}_{2,P}) = \langle \boldsymbol{w}_{2,P}, \boldsymbol{\mu}_{\{Y_i,Y_j\}} \rangle,$$

where we set $H^{2,P}_{\emptyset}$ to the identity operator, such that $\boldsymbol{w}_{2,P} \in \mathbb{R}^{\mathcal{Y} \times \mathcal{Y}}$ is a simple table of energy values for each possible configuration (y_i, y_j). This setting contains the *generalized Potts* model for pairwise interactions as a special case. Note that unlike in random fields defined on a pixel grid we do not assume regular/submodular/attractive energies and also do not require symmetry of the matrix $\boldsymbol{w}_{2,P}$. This is important because the role of child and parent variable is known; for instance, a children-parent region labeling of (“*car*″, “*background*″) is more likely to occur than (“*background*″, “*car*″). We consider a second type of energy as a baseline: the constant energy, making all variables $Y_i \in Y$ independent. We define it as parameter-less energy $E^{2,\text{constant}}_{\{Y_i,Y_i\}}(y_i, y_j) = 0$.

4.3 Training and Testing

Training. For solving Problem 2 we use the LBFGS method from the minFunc package of Mark Schmidt[4] and for the inference we use libDAI [29]. In the experiments we state the number of LBFGS iterations used.

For each instance in the training set, we set as ground truth label $\boldsymbol{y} \in \mathcal{Y}^V$ not the discrete labeling vector but the actual distribution $\boldsymbol{\mu}_V \in [0,1]^{\mathcal{Y}^V}$ of pixel labels within each image region. This faithfully represents the actual ground truth information and reduces to the discrete label case if all pixels within a region have the same label.

For the prior distribution over the parameters $\boldsymbol{w}_1$, $\boldsymbol{w}_2$, and $\boldsymbol{w}_3$ we choose a multivariate Normal distribution, such that $p(\boldsymbol{w}_1; \sigma) = \mathcal{N}(\boldsymbol{0}, \sigma^2 I)$, $p(\boldsymbol{w}_2; \tau) = \mathcal{N}(\boldsymbol{0}, \tau^2 I)$, and $p(\boldsymbol{w}_3; \tau) = \mathcal{N}(\boldsymbol{0}, \tau^2 I)$. This leads to two hyper-parameters (σ, τ) to be selected by model selection.

Test-time prediction. For a given test image $\boldsymbol{x}$ and trained model $\boldsymbol{w}^*$ we find the MAP labeling $\boldsymbol{y}^* = \operatorname{argmin}_{\boldsymbol{y} \in \mathcal{Y}^V} E(\boldsymbol{y}; \boldsymbol{x}, \boldsymbol{w}^*)$. In $\boldsymbol{y}^*$ we have one label per hierarchical image region, whereas the original task is to label each pixel with a unique label. It could therefore be the case that two region labels contradict each other in their pixel assignments. We could enforce consistency by assigning infinite energies to children-parent labelings of the form (y_c, y_p) where $y_c \neq y_p$ and $y_p \neq$ “background”. However, inconsistent labelings are absent in the training data and hence the model parameters are already chosen such that inconsistent labelings are unlikely. Experiments confirm this: on holdout data less than 0.7% of all children-parent links are inconsistently labeled. Therefore,

[4] `http://people.cs.ubc.ca/~schmidtm/Software/minFunc.html`

Unary features	seg-val	Train time	D
SIFT	6.13%	22h01m	11,193
QHOG	8.40%	19h30m	11,193
QPHOG	7.35%	36h03m	11,193
STF	6.76%	39h36m	42,945
QHOG,QPHOG	10.92%	24h35m	21,945
SIFT,QHOG,QPHOG	14.54%	26h17m	32,697
all features	15.04%	41h39m	75,201

Fig. 7. The result of feature combination at the unary factors.

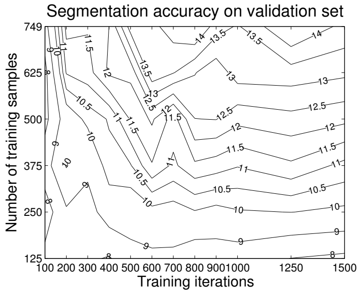

Fig. 8. VOC 2009 validation accuracy as the training set size and number of LBFGS iterations vary.

for making test-time predictions we label each pixel with the label of its largest region that is not assigned a background label. In case no such region exist, the background label is assigned.

5 Experiments

Throughout the experiments section we use the PASCAL VOC 2009 dataset [27]. The segmentation challenge contains 1499 annotated images (749 training, 750 validation), labeling each pixel with either "background" or one of 20 object classes, such as car, person, bottle, etc. The dataset is widely accepted to be difficult and realistic. We report the official PASCAL VOC2009 segmentation challenge performance measure [27] which is the average over 20 object classes of the intersection/union metric. Except for the final challenge evaluation, all models are trained on the segmentation `train` set (749 images) and we report the performance on the segmentation `val` set (750 images).

5.1 Quantifying the Effect of Feature Combination

For high level vision tasks such as object recognition, image classification and segmentation it is now well accepted that the combination of multiple image features is essential for obtaining good performance [30]. On the other hand, the use of multiple image features leads to models with many parameters and thus a possibly higher estimation error or overfitting.

We verify our model by evaluating the performance of individual features versus their combination. We do not perform model selection and fix $\sigma = 1000$, $\tau = 1000$. We train using 700 LBFGS iterations on the segmentation `train` set and report the performance on the segmentation `val` set.

Table 7 reports the results. As expected, combining multiple features is essential to obtain reasonable performance levels. Combining the three SIFT, QHOG, and QPHOG features doubles the performance of each individual one.

Moreover, we find that adding any reasonable image feature never decreased the performance. This shows that our model can combine multiple image features in a robust way, and a high dimensionality D of the parameter space does not lead to overfitting. We submitted a model trained using all features on the segmentation `trainval` dataset to the VOC2009 challenge. Some good and erroneous segmentations of this model are shown in Figure 9. A discussion of the challenge results and how other CRF-based approaches fared can be found in the supplementary materials.

5.2 Training Set Size and Learning Tradeoff

For any machine learning model, there exists a tradeoff between the expressivity of the model, the scalability to large training sets and the feasibility of optimization [31]. This experiment determines what the limiting dimension of our model is: the model class, the training set size or the training procedure. We train using the SIFT, QHOG and QPHOG features as we vary the training set size and the LBFGS iterations.[5] We evaluate each model on the validation set.

The results are shown in Figure 8. Up to about 600 LBFGS iterations the performance increases with more iterations. This is true for all training set sizes, but eventually the performance levels off when enough iterations are used. Uniformly the performance increases when more training samples are used. This indicates that the model has enough expressive power to achieve high accuracy but is currently limited by the small amount of annotated training data.

5.3 Piecewise versus Joint Parameter Learning

Piecewise training [32] is a two-step procedure where in the first step the factor graph is decomposed into disjoint subgraphs and each subgraph is trained individually. In the second step the learned weights are fixed and the factors joining the subgraphs are jointly trained. Piecewise training is an effective approximation and has been extensively used. Despite this, it has so far not been studied how much is lost compared to joint training of the model.

To quantify what is lost we use CMLE training with 700 iterations on the SIFT, QHOG and QPHOG features. We first produce a model without pairwise potentials (Unary only) by selecting $\sigma \in \{10, 100, 1000\}$ for best performance on the validation set. The learned parameters are fixed and the pairwise energy $E^{2,P}$ is used to retrain, selecting $\tau \in \{10, 100, 1000\}$ for best performance on the validation set (Piecewise, Potts). The canonical competitor to this piecewise-trained model is a jointly trained model (Joint), with $\sigma, \tau = 1000$ fixed.

The results are shown in Table 1. The training time is reduced, but it is surprising that the loss due to piecewise training of the unary energies is negligible.

[5] The training set size is within $\{125, 250, 375, 500, 625, 749\}$, the training iterations within $\{100, 200, \ldots, 1000, 1250, 1500\}$.

Model	seg-val	Training time
Unary only,	9.98%	2h15m
Piecewise, Potts	14.50%	(2h15)+10h28m
Joint	14.54%	26h17m

Table 1. VOC 2009 segmentation accuracy on validation set for the best performing unary-only model, the best piecewise-trained model, and the jointly-trained model.

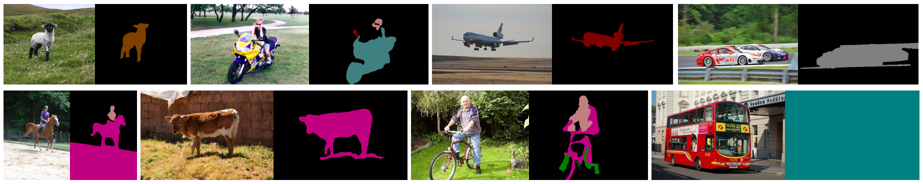

Fig. 9. VOC test predictions. Top: success, bottom row: typical failures (background labeled, wrong label, clutter, entire image labeled).

5.4 Maximum Likelihood versus Max-Margin

So far we have estimated the parameters of our models using the principle of maximum likelihood. An alternative method to estimate $\boldsymbol{w}$ from training data is the *maximum margin principle* [33], recently applied to learn structured prediction models in computer vision [34, 6, 12, 19] using the structured SVM formulation.

We use the standard margin-rescaling structured SVM formulation [33], which we describe in the supplementary materials. The use of the structured SVM entails the choice of a semi-metric $\Delta(\boldsymbol{y}_n, \boldsymbol{y})$ and the parameter C_{svm}. For $\Delta : \mathcal{Y}^V \times \mathcal{Y}^V \rightarrow \mathbb{R}_+$ we choose the same function as [12], weighting the regions by their relative sizes, something that is not possible in standard CMLE training.

We evaluate the structured SVM against CMLE with 500 LBFGS iterations. For the structured SVM we use the popular cutting plane training procedure [33], solved using the Mosek QP solver. We evaluate $C_{\text{svm}} \in \{10^{-5}, 10^{-4}, \ldots, 1\}$ for the structured SVM model and $(\sigma, \tau) \in \{100, 1000\} \times \{1, 10, 100, 1000\}$ for CMLE and report the best achieved performance on the validation set using the SIFT,QHOG,QPHOG features and the data-independent pairwise Potts factor. For larger values of C_{svm} the cutting-plane training procedure failed; we describe this in detail in the supplementary materials.

	Accuracy CMLE	Training time CMLE	Accuracy SVM	Training time SVM
Potts	13.65%	24h11m	13.21%	165h10m

Table 2. Results of maximum likelihood training and structured support vector machine training. See main text for details.

The results shown in Table 2 show that the CMLE training procedure requires less time and outperforms the structured SVM model consistently. It is unclear and remains to be examined whether this is due to the failure of the structured SVM optimization procedure for large values of C_{svm} or because of an inferior estimator.

6 Conclusions and Future Work

We draw the following conclusions for the class of tree-structured/hierarchical CRF based approaches to object class image segmentation:

- Current CRF models are limited by the amount of training data and available image features; more of both consistently leads to better performance,
- Piecewise training of unary observation factors is competitive with joint training and reduces the required training time considerably,
- Max-margin training is not well-tested within computer vision; current methods are slow and unstable in case of many parameters.

This work provides recommendations for the tractable, tree-structured case on a popular high-level vision task. In the future we plan to provide a larger study examining whether our conclusions extend to general intractable CRF models learned using approximate inference. Additionally, we would like to examine other high-level data-driven vision tasks.

References

1. Lauritzen, S.L.: Graphical Models. Oxford University Press (1996)
2. Koller, D., Friedman, N.: Probabilistic Graphical Models: Principles and Techniques. MIT Press (2009)
3. Lafferty, J., McCallum, A., Pereira, F.: Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In: ICML. (2001)
4. Sutton, C., McCallum, A.: An introduction to conditional random fields for relational learning. In: Introduction to Statistical Relational Learning. MIT Press (2007)
5. Kschischang, F.R., Frey, B.J., Loeliger, H.A.: Factor graphs and the sum-product algorithm. IEEE Transactions on Information Theory **47** (2001) 498–519
6. Szummer, M., Kohli, P., Hoiem, D.: Learning CRFs using graph cuts. In: ECCV. (2008)
7. Winn, J.M., Shotton, J.: The layout consistent random field for recognizing and segmenting partially occluded objects. In: CVPR. (2006)
8. Shotton, J., Winn, J., Rother, C., Criminisi, A.: Textonboost for image understanding: Multi-class object recognition and segmentation by jointly modeling texture, layout, and context. IJCV **81** (2007)
9. Kumar, S., Hebert, M.: Discriminative random fields: A discriminative framework for contextual interaction in classification. In: ICCV. (2003)
10. He, X., Zemel, R.S., Carreira-Perpiñán, M.Á.: Multiscale conditional random fields for image labeling. In: CVPR. (2004)

11. Schnitzspan, P., Fritz, M., Schiele, B.: Hierarchical support vector random fields: Joint training to combine local and global features. In: ECCV. (2008)
12. Nowozin, S., Lampert, C.H.: Global connectivity potentials for random field models. In: CVPR. (2009)
13. Gould, S., Rodgers, J., Cohen, D., Elidan, G., Koller, D.: Multi-class segmentation with relative location prior. IJCV **80** (2008) 300–316
14. Batra, D., Sukthankar, R., Chen, T.: Learning class-specific affinities for image labelling. In: CVPR. (2008)
15. Reynolds, J., Murphy, K.: Figure-ground segmentation using a hierarchical conditional random field. In: CRV. (2007)
16. Plath, N., Toussaint, M., Nakajima, S.: Multi-class image segmentation using conditional random fields and global classification. In: ICML. (2009)
17. Kohli, P., Ladický, L., Torr, P.H.S.: Robust higher order potentials for enforcing label consistency. In: CVPR. (2008)
18. Ladický, L., Russell, C., Kohli, P.: Associative hierarchical crfs for object class image segmentation. In: ICCV. (2009)
19. Munoz, D., Bagnell, J.A., Vandapel, N., Hebert, M.: Contextual classification with functional max-margin markov networks. In: CVPR. (2009)
20. Kumar, S., August, J., Hebert, M.: Exploiting inference for approximate parameter learning in discriminative fields: An empirical study. In: EMMCVPR. (2005)
21. Korc, F., Förstner, W.: Approximate parameter learning in conditional random fields: An empirical investigation. In: DAGM, Springer (2008)
22. Parise, S., Welling, M.: Learning in Markov random fields: An empirical study. In: Joint Statistical Meeting JSM2005. (2005)
23. Finley, T., Joachims, T.: Training structural SVMs when exact inference is intractable. In: ICML. (2008)
24. Willsky, A.S.: Multiresolution markov models for signal and image processing. Proceedings of the IEEE (2002)
25. Lim, J.J., Gu, C., Arbeláez, P., Malik, J.: Context by region ancestry. In: ICCV. (2009)
26. Arbeláez, P.: Boundary extraction in natural images using ultrametric contour maps. In: Workshop on Perceptual Organization in Computer Vision. (2006)
27. Everingham, M., Gool, L.V., Williams, C.K., Winn, J., Zisserman, A.: The PASCAL Visual Object Classes Challenge 2009 (VOC2009) Results. http://www.pascal-network.org/challenges/VOC/voc2009/workshop/ (2009)
28. Shotton, J., Johnson, M., Cipolla, R.: Semantic texton forests for image categorization and segmentation. In: CVPR. (2008)
29. Mooij, J.M.: libDAI: A free/open source C++ library for discrete approximate inference methods. http://www.libdai.org/ (2008)
30. Gehler, P., Nowozin, S.: On feature combination for multiclass object classification. In: ICCV. (2009)
31. Bottou, L., Bousquet, O.: The tradeoffs of large scale learning. In: NIPS. (2007)
32. Sutton, C.A., McCallum, A.: Piecewise training for undirected models. In: UAI. (2005)
33. Tsochantaridis, I., Joachims, T., Hofmann, T., Altun, Y.: Large margin methods for structured and interdependent output variables. JMLR **6** (2005) 1453–1484
34. Blaschko, M.B., Lampert, C.H.: Learning to localize objects with structured output regression. In: ECCV. (2008)