Closed-Form Approximate CRF Training for Scalable Image Segmentation

Alexander Kolesnikov[1], Matthieu Guillaumin[2], Vittorio Ferrari[3], and Christoph H. Lampert[1]

[1] IST Austria
[2] ETH Zürich
[3] University of Edinburgh

Abstract. We present LS-CRF, a new method for training cyclic Conditional Random Fields (CRFs) from large datasets that is inspired by classical closed-form expressions for the maximum likelihood parameters of a generative graphical model with tree topology. Training a CRF with LS-CRF requires only solving a set of independent regression problems, each of which can be solved efficiently in closed form or by an iterative solver. This makes LS-CRF orders of magnitude faster than classical CRF training based on probabilistic inference, and at the same time more flexible and easier to implement than other approximate techniques, such as pseudolikelihood or piecewise training. We apply LS-CRF to the task of semantic image segmentation, showing that it achieves on par accuracy to other training techniques at higher speed, thereby allowing efficient CRF training from very large training sets. For example, training a linearly parameterized pairwise CRF on 150,000 images requires less than one hour on a modern workstation.

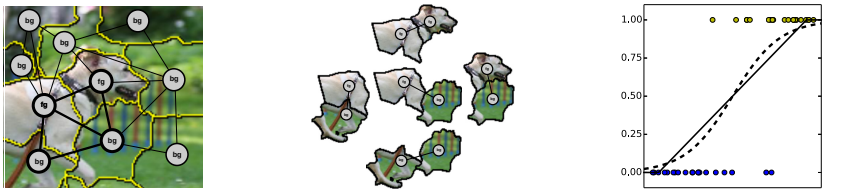
1 Introduction

Many areas of computer vision research have recently made a transition from datasets with a few thousand images to much larger datasets with millions of images. One area that is relatively untouched by this trend is *semantic image segmentation*, i.e. the task of assigning a semantic label, such as *grass* or *building* to every pixel in an image. There are few aspects that hold back the field: a lack of large-scale image segmentation datasets, because manually creating their annotation is tedious and costly, and a lack of learning techniques for structured prediction models that scale to truly large amounts of training data. In this work, we make two contributions that address both problems:

- a method for CRFs training that is *scalable* (training sets can be 100,000 images or larger), *flexible* (allowing for linear or nonlinear predictors), and *easy to implement* (only a few lines of code are required for the linear variant),
- two new *large scale datasets* [1] (over 180,000 training images) for image segmentation, assembled from ImageNet [8] and PASCAL VOC [11] datasets and augmented with semi-automatically created figure-ground annotations.

[1] Available online at `http://pub.ist.ac.at/~akolesnikov/HDSeg`

D. Fleet et al. (Eds.): ECCV 2014, Part III, LNCS 8691, pp. 550–565, 2014.



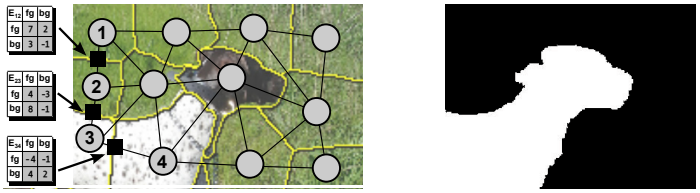
(a) superpixel graph (b) pairwise decomposition (c) parameter regression

Fig. 1. Schematic illustration of LS-CRF for image segmentation – training phase: (a) we are given images with predefined graph structure (here based on superpixels) and per-node annotation (here: **fg/bg**), (b) we form training subproblems from all edges in the graph (shown for bold subgraph), (c) for each label combination, we train a linear (solid line) or nonlinear (dashed line) regressor to predict the label combination's conditional probability

We call the new method LS-CRF, where the LS stands both for *least squares* and for *large scale*. It can in principle be used for any pairwise discrete-valued CRFs, but in particular it is suitable for the classes of CRFs that occur in computer vision applications, such as image segmentation, where 1) the CRF (i.e. its underlying graph) is cyclic, 2) all variables are of the same "type", e.g. pixels or superpixels, 3) each variable takes values in a rather small label set, and 4) many training examples are available.

Figures 1 and 2 illustrate LS-CRF for semantic image segmentation. A formal definition and justification are given in Section 2. Our contribution lies in the steps 1(b), 1(c) and 2(a), which we explain in detail in Section 2. The main idea is to decompose the joint learning problem into smaller, tractable subproblems and learn their parameters by independent least-squares regression tasks. For a new image, the output of the regressors provide values for the energy tables for all pairwise terms. This induces a (conditional) probability distribution over the label set, from which we infer a segmentation, for example by maximum-a-posteriori (MAP) prediction.

Compared to existing methods, LS-CRF has several advantages: in contrast to generic maximum likelihood training for CRFs, no step of joint probabilistic inference (i.e. computing marginals over the training data) is required during training. Instead, the training is formulated as solving a collection of subproblems, which can be solved efficiently and even in parallel. This makes training scalable even to large datasets. In our experiments, training with over 150,000 images takes less than one hour on a modern workstation. In contrast to related techniques, such as pseudolikelihood or piecewise training, a closed-form solution for the optimal parameters is available for the case of a linear parameterization and squared loss, which can be solved in just a few lines of code. However, LS-CRF is not limited to linear parameterizations. It also allows efficient learning of nonlinear, even non-parametric, energy functions. In our experiments, we demonstrate this by learning a CRF with an energy function based on gradient boosted decision trees.



(a) predicted energy (b) segmentation result

Fig. 2. Schematic illustration of LS-CRF for image segmentation – prediction phase: (a) for a new image we use the regressors to predict an energy table for each pairwise term (visualized on three edges between four nodes), (b) The energy function defines a probability distribution which yields a segmentation (here: by MAP prediction)

The two new datasets that we introduce, *HorseSeg* and *DogSeg*, are described in Section 5. They are meant to facilitate experiments in two setups that reflect recent trends in computer vision research: large-scale learning, and learning with weakly annotated data. In combination the datasets consist of over 180,000 images that were taken from the PASCAL VOC and the ImageNet dataset. For the test images we provide manually created segmentation masks, thereby allowing for an unbiased evaluation. The training images come with three different levels of annotation: all training images have a class label, some training images have object bounding boxes, few training images have manually created per-pixel segmentations. In this form, the datasets can be used to evaluate segmentation algorithms based on semi-supervised or weakly supervised learning.

In addition, for each training image we provide a segmentation mask created automatically using the *segmentation transfer* method [18]. With this annotation, the datasets can serve as a benchmark for the efficiency of large scale learning methods, or to analyze the stability of learning algorithms to annotation noise.

2 Closed-Form Training of Conditional Random Fields

In this section we formally introduce LS-CRF training and justify its construction from classical results about probabilistic inference and estimation theory. We highlight the similarities and differences of LS-CRF to previous methods for CRF learning and discuss possible extensions.

2.1 Conditional Random Fields

We follow the standard notation for the probabilistic learning of discrete conditional random fields (CRFs). See, e.g., the tutorial [24] for an introduction in the context of computer vision. To facilitate the discussion, we refer to the objects of interest in the context of image segmentation, but note that all steps are also applicable for training CRFs for other tasks.

We denote input images by $x \in \mathcal{X}$, and outputs (segmentations) by $y \in \mathcal{Y}$. Any y is a collection of interacting parts, $y = (y_1, \ldots, y_m)$, where each part y_s for $s = 1, \ldots, m$ takes a value in a finite label set, $\mathcal{L} = \{1, \ldots, r\}$. Every y has a graph, $G = (\mathcal{V}, \mathcal{E})$, associated with it. The vertex set, $\mathcal{V} = \{1, \ldots, m\}$, reflects the parts, and the edge set, $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$, contains an edge for every directly interacting pair of parts. For example, in image segmentation the edge $\mathcal{E}$ typically contains all pairs of adjacent pixels or superpixels.

A CRF models a conditional distribution of outputs given inputs by

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

where $E(x, y; w)$ is an *energy function* with parameter w, and the normalizing constant $Z(x; w) = \sum_{y \in \mathcal{Y}} \exp(-E(x, y; w))$ is called the *partition function*. Most popular for computer vision applications are energy functions consisting of *unary* and *pairwise* terms.

$$E(x, y; w) = \sum_{s \in \mathcal{V}} \sum_{j \in \mathcal{L}} \theta_{s;j} [\![y_s = j]\!] \; + \sum_{(s,t) \in \mathcal{E}} \; \sum_{(j,k) \in \mathcal{L} \times \mathcal{L}} \theta_{st;jk} [\![y_s = j \wedge y_t = k]\!], \tag{2}$$

where $[\![P]\!] = 1$ if the predicate P is true, and 0 otherwise. The coefficients are functions of a subset of the input x and parameters w. Writing lower indices to indicate which parts of the input and parameter vector to use, we set $\theta_{s;j} = \log g(x_s, w_j)$ and $\theta_{st;jk} = \log f(x_{st}, w_{jk})$. Different choices of g and f result in model of different complexity and expressive power.

Given a training set, $\{(x^1, y^1), \ldots, (x^n, y^n)\}$, the goal of *CRF learning* is to identify the parameters maximizing the *conditional likelihood* (CL) of the data, or equivalently, that minimizes the negative logarithm of this quantity, i.e.

$$w = \operatorname{argmin}_{\bar{w}} \ell_{CL}(\bar{w}), \quad \text{with} \quad \ell_{CL}(\bar{w}) = \sum\nolimits_{i=1}^{n} E(x^i, y^i; \bar{w}) + \log Z(x^i; \bar{w}). \tag{3}$$

Unfortunately, computing expression (3) exactly is computationally infeasible (more exactly, $\#P$-hard [6]), even for small n, unless the underlying graphs are cycle-free. Consequently, all currently used methods for CRF training on cyclic graphs, as they occur in image segmentation tasks, aim only for approximate solutions. Ultimately, their performance on real data determines their usefulness. We discuss successful previous techniques in Section 3.

2.2 Least Squares CRF Training (LS-CRF)

To introduce LS-CRF, we assume an energy function in *canonical form*, i.e. the energy function contains exactly one term for each maximal clique of the underlying graph and no terms for smaller cliques. For pairwise models in which each part occurs in at least one edge this means that the energy has only pairwise terms. Note that we do not lose any expressive power by this: for every energy function with unary and pairwise terms there is an energy function with only pairwise terms that has identical values for all labelings.

Algorithm 1. LS-CRF – Training

input training data: $(x^i, y^i, G^i)_{i=1,\dots,n}$,
x^i: images, y^i: ground truth annotation, $G^i = (\mathcal{V}^i, \mathcal{E}^i)$: graphs,
input regularization parameter: $\lambda \geq 0$,
1: set $\phi^i_{st} \leftarrow \phi(x^i_{st}) \in \mathbb{R}^D$ // edge feature vector for all $i = 1, \dots, n$, and $(s,t) \in \mathcal{E}^i$
2: **for** $j, k \in \mathcal{L} \times \mathcal{L}$ **do**
3: set $\mu^i_{st;jk} \leftarrow [\![y^i_s = j \wedge y^i_t = k]\!]$ for all $i = 1, \dots, n$, and $(s,t) \in \mathcal{E}^i$
4: form training set $\mathcal{S}_{jk} = \bigcup_{i \in I} \bigcup_{(s,t) \in \mathcal{E}^i} \{(\phi^i_{st}, \mu^i_{st;jk})\}$
5: learn w_{jk} from $\mathcal{S}_{jk}$ by regularized least squares regression
$w_{jk} \leftarrow \operatorname{argmin}_{\bar{w}_{jk}} \sum_{(\phi,\mu) \in \mathcal{S}_{jk}} \|f(\phi, \bar{w}_{jk}) - \mu\|^2 + \lambda \|\bar{w}_{jk}\|^2$
6: **end for**
output parameter vector $w = (w_{jk})_{(j,k) \in \mathcal{L} \times \mathcal{L}}$.

Algorithm 2. LS-CRF – Prediction

input image x, graph $G = (V, \mathcal{E})$,
input weight vector $w = (w_{jk})_{(j,k) \in \mathcal{L} \times \mathcal{L}}$,
1: $\phi_{st} \leftarrow \phi(x_{st}, w_{jk}) \in \mathbb{R}^D$ // edge feature vector for all $(s,t) \in \mathcal{E}$
2: **for** $j, k \in \mathcal{L} \times \mathcal{L}$ **do**
3: $\theta_{st;jk} = \log f_{jk}(\phi_{st})$ for all $(s,t) \in \mathcal{E}$
4: **end for**
5: $E(x, y; w) = \sum_{(s,t) \in \mathcal{E}, (j,k) \in \mathcal{L} \times \mathcal{L}} \theta_{st;jk} [\![y_s = j \wedge y_t = k]\!]$
output energy function $E(x, y; w)$

Algorithm 1 shows the training phase of our method in pseudocode. The main step is solving multiple independent regression problems in line 5. For each label combination, $(j,k) \in \mathcal{L} \times \mathcal{L}$, we form a training set, $S_{jk} = \{(\phi^1, \mu^1), \dots, (\phi^N, \mu^N)\}$, by merging of all pairs $(\phi^i_{st}, \mu^i_{st;jk})$, where ϕ^i_{st} is a feature vector reflecting the visual information in the parts x_s and x_t, and $\mu^i_{st;jk}$ is an indicator whether the edge (s,t) in the training example x^i was labeled with the current label combination (j,k) or not. Note that the resulting number of samples N will be much larger than the number of training images n, since every training image contributes many edges.

We obtain w_{jk} by solving the regularized least squares regression problem

$$w_{jk} = \operatorname{argmin}_{\bar{w}_{jk}} \sum\nolimits_{(\phi,\mu) \in \mathcal{S}_{jk}} \|f_{jk}(\phi, \bar{w}_{jk}) - \mu\|^2 + \lambda \|\bar{w}_{jk}\|^2, \tag{4}$$

where $\lambda \geq 0$ is a regularization parameter. In case of a linear parameterization, $f(\phi, w) = \langle w, \phi \rangle$, Equation (4) has a closed-form solution,

$$w_{jk} = (\Phi\Phi^\top + \lambda \, \mathrm{I})^{-1} \Phi \boldsymbol{\mu}, \tag{5}$$

where $\Phi = (\phi^1 | \dots | \phi^N)$ is the feature matrix and $\boldsymbol{\mu} = (\mu^1, \dots, \mu^N)^\top$ is the vector of outputs. Computing w_{jk} requires solving a linear system of size $D \times D$ for D-dimensional feature representation, which is possible efficiently even for D in the order of thousands. Since only matrix operations are required, Equation (5) can also be solved easily on a GPU.

To fully train a CRF, one solves one instance of Equation (5) for each label combination (j, k). Only the target vector $\boldsymbol{\mu}$ differs between these, so compared to a naive loop we obtain a substantial speedup by precomputing an LU-factorization and then solving the individual problems by backwards substitution.

When the number of training examples is too large for Φ to fit into memory, the computation of $\Phi\Phi^\top$ can become the computational bottleneck. In this case, the use of iterative least-squares solvers instead of the closed-form expression is advisable, for example LBFGS [7], or plain stochastic gradient descent [5].

Algorithm 2 shows how the result of LS-CRF training is used to predict labelings for new images. We use the trained regression functions, f_{jk}, to predict the values of the pairwise terms of an energy function $E(x, y; w)$. As for any CRF, the resulting energy induces a conditional probability distribution over all possible labelings, from which we can predict an output labeling. In this work we rely on MAP prediction, i.e. $y^* = \operatorname{argmax}_{y \in \mathcal{Y}} p(y|x; w)$.

2.3 Extensions

Several extensions of the above procedure are possible. For example, pairwise terms of different types can be learned (e.g. vertical versus horizontal edges), by forming individual learning problems for each of them. A particular advantage of LS-CRF is that the main learning step, Equation (4), is not limited to the situation of linearly parameterized energies. This allows increasing the expressive power of the model by training nonlinear predictors when necessary, e.g. for low-dimensional feature spaces. It is also possible to use different loss functions than the squared loss. To illustrate this we also report on experiments using gradient boosted decision trees [13] and logistic loss in Section 5.

2.4 Motivation and Analysis

Like all tractable techniques for the training of cyclic CRFs, LS-CRF does not solve the exact CRF training problem but an approximation to it. In this section, we justify this approximation based on two observation: the fact that one can overcome the intractability of CRF training by constructing bounds to the partition function that decompose into smaller parts, and a classical result how to estimate the optimal parameter for tree-shaped generative probability distributions from samples.

The first aspect LS-CRF shares with other approximate training methods: for any CRFs with an energy function consisting only of pairwise terms the following inequality holds

$$\log Z(x; \theta) = \log \sum_{y \in \mathcal{Y}} \exp(-E(x, y; w) \leq \sum_{(s,t) \in \mathcal{E}} \log \sum_{(y_s, y_t) \in \mathcal{L} \times \mathcal{L}} \exp(-E_{st}(x, y; w)), \quad (6)$$

where $E_{st}(x, y; \theta) = \sum_{(j,k) \in \mathcal{L} \times \mathcal{L}} \theta_{st;jk} [\![y_s = j \wedge y_t = k]\!]$ is the term of the energy function corresponding to the edge (s, t). This relation has been observed, for

example, in [29], where it is used to motivate *piecewise training*: one replaces the $\log Z$ term in the original log-likelihood function (Equation (3)) by its upper bound and performs gradient descent optimization on the resulting approximate objective. This allows for faster training, since the right hand side of (6) decomposes additively over the edge set.

LS-CRF goes one step further, as it estimates the parameters of the decomposed model not by gradient descent on the decomposed approximate conditional likelihood, but by a direct regression between the input features and suitably chosen output values. To show the justification of this procedure, we rely on the classical, but rarely used, result that there exists a closed-form expression for estimating the parameter of a probability distribution (not conditional) with tree-shaped graph structure from samples (see [33, page 150]).

We assume the same situation as Equation 2, except that there is no dependence on x, so $\theta_{s;j}$ and $\theta_{st;jk}$ are constants. Given samples $y^1, \ldots, y^n$ from $p(y;\theta)$, an elementary calculation shows that $\hat{\theta}_{s;j} = \log \hat{\mu}_{s;j}$ and $\hat{\theta}_{st;jk} = \log \frac{\hat{\mu}_{st;jk}}{\hat{\mu}_{s;j}\hat{\mu}_{t;k}}$ are consistent estimators of the optimal $\theta_{s;j}$ and $\theta_{st;jk}$, where $\hat{\mu}_{s;j} = \frac{1}{T}\sum_{i=1}^{n} [\![y_s^i = j]\!]$ and $\hat{\mu}_{st;jk} = \frac{1}{T}\sum_{i=1}^{n} [\![y_s^i = j \wedge y_t^i = k]\!]$ are the empirical unary and pairwise marginals over the training set.

For LS-CRF we generalize the above closed-form expression to a technique for estimating the parameters of a conditional distribution, i.e. a CRF. The samples we observe, $\{(x^1, y^1), (x^2, y^2), \ldots, (x^n, y^n)\}$, are now pairs of inputs and outputs. In particular, no input repeats and for each input only one output is available. Therefore, the empirical marginals are just indicators which labels and label pairs occurred: $\hat{\mu}_{s;j}(x^i) = [\![y_s^i = j]\!]$, and $\hat{\mu}_{st;jk}(x^i) = [\![y_s^i = j \wedge y_t^i = k]\!]$. Nevertheless, we know that these values are i.i.d. samples from the conditional distributions $p(y_s = j|x)$ and $p(y_s = j, y_t = k|x)$, respectively. We can learn predictors $f_{s;j}(x) \approx p(y_s = j|x)$ and $f_{st;jk}(x) \approx p(y_s = j, y_t = k|x)$, and obtain parameter estimates $\hat{\theta}_{s;j}(x) = \log f_{s;j}$ and $\hat{\theta}_{st;jk} = \log \frac{f_{st;jk}}{f_{s;j}(x) f_{t;k}}$ that generalize from the observed samples to unseen data. Algorithm 1 implements the step of learning the pairwise predictor, $f_{st;jk}$, which we'll see below is sufficient for us.

For tree-shaped models, the learned predictors provide a consistent estimator of the conditional distribution, as long as the underlying regression technique is consistent. For cyclic models, we do not have this guarantees. However, we can apply the above construction for each edge (which is loop-free) of the decomposed representation. This yields a partial energy

$$\begin{aligned} E_{st}(x;\hat{\theta}_{st}) &= \sum\nolimits_j \hat{\theta}_{s;j} [\![y_s = j]\!] + \sum\nolimits_{j,k} \hat{\theta}_{st;jk} [\![y_s = j \wedge y_t = k]\!] + \sum\nolimits_k \hat{\theta}_{t;k} [\![y_t = k]\!] \\ &= \sum\nolimits_{j,k} \big(\hat{\theta}_{s;j} + \hat{\theta}_{st;jk} + \hat{\theta}_{t;k}\big) [\![y_s = j \wedge y_t = k]\!] \\ &= \sum\nolimits_{j,k} (\log f_{st;jk}) [\![y_s = j \wedge y_t = k]\!], \end{aligned} \tag{7}$$

which reflects how Algorithm 2 constructs the energy function. It also shows that only estimates for the pairwise terms are required, not for the unary ones.

3 Related Work

Probabilistic CRF training aims at maximizing the conditional likelihood of the training data. However, doing so requires probabilistic inference as a subroutine, and this is provably intractable for cyclic models [6]. Consequently, all practical methods rely on approximations. The simplest idea is to use approximate inference methods during the optimization. This can lead to unstable behavior of the optimization process or even divergence [9]. This is also not scalable, since even approximate probabilistic inference is computationally demanding.

The empirically most successful techniques form a tractable approximation to the conditional likelihood and then solve for its optimum, in particular pseudolikelihood (PL) [4] and piecewise (PW) training [29]. LS-CRF is related to PW training in that it also relies on a per-edge decomposition, but it differs in how it estimates the parameters. In particular, it supports non-linear and even non-parametric estimates of the energy function. To our knowledge this feature is only shared by *decision tree fields (DTFs)* [25], which differ, however, in other aspects. In particular, they are trained by a PL approximation.

Structured support vector machines (SSVMs) [32] have been proposed as a alternative to CRFs, based not on probabilistic reasoning but on the maximum margin principle. Like CRFs, exact SSVMs training is intractable for almost all cyclic models, since it requires repeated runs of MAP prediction during training. Similarly to CRFs, it has been observed that training with approximate MAP can lead to a drop in prediction quality [12]. In contrast to CRFs, so far few decomposition-based techniques exists for SSVMs. Prominent examples are [10,21,23], which introduce a scheme of alternating between two steps: solving independent subproblems and updating Lagrange multipliers to enforce consistency between the solutions of the subproblems. However many iterations may be required until convergence, and experience in a large-scale setting is so far missing. Another example is pseudo-max [28], which replaces the SSVM objective with a tractable approximation inspired by the PL method.

In computer vision it is common to use hybrid techniques (e.g. [14,26,27]): first, ordinary classifiers, e.g. SVMs or random forests, are trained to predict unary features. Afterwards, a CRF or SSVM is run using the outputs of the classifiers as features or drop-in replacements for the unary terms. This does not avoid training a CRF altogether, though, it only reduces the number of parameters that need to be learned.

4 Implementation

Implementing LS-CRF is straight-forward, since it only requires solving multiple regression problems. Several efficient software packages are readily available for this task. In our experiments with linear regression, we use the *Vowpal Wabbit* package[2] (LBFGS optimization, learning rate 0.5, no explicit regularization), clamping the predictions to the interval $[10^{-9}, 1]$. Vowpal Wabbit is particularly

[2] `http://hunch.net/~vw/`

suitable for large scale learning, since it supports a variety of infrastructures, from single CPUs to compute clusters. As nonlinear regressors we train gradient boosted regression trees using the *MatrixNet* package [31] with default parameters (500 oblivious trees, depth 6, learning rate 0.1).

We also implement several baselines: to study the impact of pairwise terms in the energy function, we train CRFs with only unary terms using the same regression methods as LS-CRF. We also implemented CRF training by piecewise training using Vowpal Wabbit, and by pseudolikelihood using the *grante* library[3]. In the unary-only case, both of these techniques are equivalent to logistic regression, for which we again use Vowpal Wabbit.

In addition, we implement two baselines that require global inference during training: a standard maximum-likelihood CRF trained by gradient descent with approximate gradients obtained by TRW belief propagation, and a structured SVM with approximate subgradients based on MAP prediction using MQPBO+TRWS (see below). Note that convergence for the latter two methods cannot be ensured, see our discussion in Section 5.1. Implementation and model selection is also more involved for these methods, so we suspect that improvements in terms of speed and convergence would be possible.

At test time, we create segmentations of the test images by MAP prediction. For this, we apply MQPBO [19] followed by TRWS [20], both from the *OpenGM2* library [2]. We observed that this combination is very efficient and usually yield close to optimal results.

5 Experimental Evaluation

Our focus in this work lies on large-scale semantic image segmentation, where a semantic label should be assigned to each pixel or superpixel in an image. In the case of *multi-class semantic segmentation*, these labels correspond to semantic classes, such as *road*, *car* or *person*. The number of labels, r, is typically between 5 and 20 in this case. A case of special interest is *figure–ground segmentation*, where $r = 2$ and the labels indicate *foreground (fg)* and *background (bg)*.

In Section 5.1 we report on experiments on the relatively small *Stanford background* dataset. The reason is that many techniques are applicable in this setup, so we can compare LS-CRF to previous techniques. In Section 5.2 we report on experiments on the *DogSeg* and *HorseSeg* datasets. These two datasets that we introduce in this work are by far larger than existing ones and show how LS-CRF allows scaling CRF training to very large datasets, and how the quality of semi-automatically generated annotation influences the segmentation quality.

5.1 Small Scale Experiments – *Stanford Background* Dataset

The *Stanford background* dataset [16], consists of 715 natural images with manually created ground truth annotation. We represent each image by a graph of

[3] `http://www.nowozin.net/sebastian/grante/`

[4] Note that *grante* is not multi-threaded, so PL runtimes are higher than necessary.

Table 1. Result of LS-CRF (top table), baselines (middle table; LR=logistic regression, PL=pseudolikelihood, PW=piecewise, ML=maximum likelihood, MM=maximum margin) and CRF-like approaches from the literature (bottom table) on the *Stanford background* dataset. Standard deviations are computed over 5 splits of the dataset. Since absolute results are hardly comparable between different publications due to different features used, we also report the absolute and relative improvement from pairwise terms (u→p) where possible.

	Accuracy				**Training Time**	
	unary	pairwise	u→p (abs)	u→p (rel)	unary	pairwise
LS-CRF linear	70.3 ± 1.0	73.4 ± 1.1	**3.1**	**4.4%**	66s	216s
LS-CRF nonlinear	70.9 ± 1.1	73.3 ± 1.1	**2.4**	**3.4%**	15m	150m
LR	70.4 ± 0.8	–	–	–	45s	—
PL	—	71.0 ± 1.1	**0.6**	**0.9%**	—	2356s[4]
PW	—	72.9 ± 0.9	**2.5**	**3.6%**	—	340s
ML	—	(see text)	–	–	—	>3 hours
MM	–	(see text)	–	–	—	>3 hours
Gould *et al.* [16]	–	74.3	–	–		
Gould, Zhang [17]	–	73.4-73.9	–	–		
Lempitsky *et al.* [22]	–	75.0-81.1	–	–		
Gould [15]	73.0	78.6	**5.6**	**7.7%**		
Tighe, Lazebnik [30]	76.9	77.5	**0.6**	**0.8%**		

SLIC superpixels [1] with an edge for any touching pair of superpixels. For each superpixel, s, we compute of feature vector, ϕ_s, by concatenating of the following base features: average RGB color in s (3-dimensional), center of s in relative image coordinates (2-dim.), rootSIFT descriptor [3] at the center of s (128-dim.), predicted per-label probabilities (8-dim.), where the last vector consists of the outputs of per-label training boosted tree classifiers on a subset of the training data. For each edge (s, t) we define a feature representation by concatenating the features of the contributing superpixels, $\phi_{st} = [\phi_s, \phi_t]$.

Table 1 summarizes the results on the Stanford Background dataset in numeric form. For segmentation examples, please see the supplemental material. We compare following setups: LS-CRF (unary-only or pairwise energies) with linear or nonlinear parameterization, pseudolikelihood (PL), piecewise training (PW), approximate maximum likelihood (ML) training. We also include an SSVM with approximate maximum margin (MM) training. Note, that PL, PW and ML all reduce to logistic regression (LR) when used with only unary terms.

We also report results for related methods (pairwise cyclic CRF) from the literature. In general, it is problematic to compare between results from different segmentation papers on an absolute scale, as the choice of representation and image features has a large impact. This is visible also in Table 1 by large differences already between models with only unary terms. A more robust measure is the increase a method achieves between unary and pairwise energies, which we also report for all methods where data for both cases is available.

Overall, the results show that for learning a segmentation model with only unary terms, the squared loss objective of LS-CRF achieves comparable result to the usual probabilistic loss (i.e. logistic regression). Including pairwise terms into the model improves the segmentation accuracy in all cases. The increase from LS-CRF is comparable to what other methods achieve, including results from

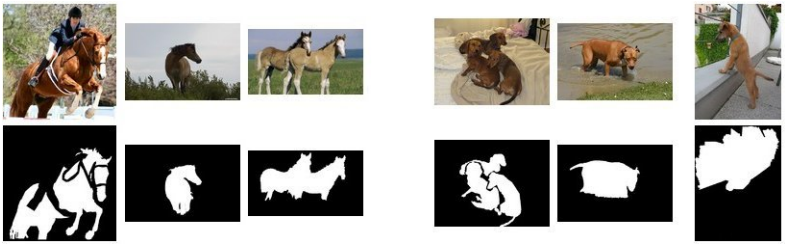


Fig. 3. Example images and segmentation masks from the *HorseSeg* (left) and *DogSeg* (right) datasets. For each dataset we illustrate: manually created annotation (left), annotation from bound boxes (middle) and annotation from per-image labels (right). Annotation from bounding boxes is typically rather accurate, the quality of annotation from per-image class labels varies substantially.

the literature, with the exception of pseudolikelihood training, which was less beneficial in this case. In terms of runtime, LS-CRF was the fastest method for training models with pairwise terms. For unary-only models, logistic regression training is slightly faster.

The nonlinear and the linear versions of LS-CRF achieve comparable performance on the Stanford background dataset, likely because the amount of training data per label or label pair is small, so the additional flexibility of a nonlinear decision function is not required.

ML and MM training are two special cases, as they require joint probabilistic inference or MAP prediction during training. We used TRW for approximate marginal inference, and MQPBO+TRWS for approximate MAP and stopped training after three hours. The results obtained were 64.8 ± 1.4 (lower than unary-only) and 72.5 ± 0.8 (lower than LS-CRF) for ML and MM, respectively. Since there is no fundamental reason why exact training should perform worse than approximate one, we do not believe that these numbers are representative for either of the methods. Better performance might be achievable with better approximate inference techniques and at the expense of longer training times.

5.2 Large Scale Experiments – *HorseSeg* and *DogSeg* Datasets

Our main interest lies on the scalable learning regime, where one hundred thousand or more images are available for training. To analyze this situation we make a second contribution of this paper besides LS-CRF: a new benchmark for large-scale CRF training in the form of two figure–ground image segmentation datasets, *HorseSeg* with 25,078 images and *DogSeg* with 156,368 images, which are available online at `http://pub.ist.ac.at/~akolesnikov/HDSeg`. The images for both datasets were collected from the ImageNet dataset [8] and the *trainval* part of PASCAL VOC2012 [11]. As test set, we use 241 horse images and 306 dog images, which were annotated manually with figure-ground segmentations. Note, that part of these images were annotated via Amazon Mechanical Turks and can contain a small amount of noise.

All images from the PASCAL dataset have manually created ground truth annotation as well. The remaining images from ImageNet have manual annotation of two different detail levels: each image has a class label, and approximately one quarter of all images also have annotation in form of a manually specified object bounding box. As such, the dataset provides a natural benchmark for weakly supervised or semi-supervised image segmentation in a large scale regime.

To facilitate experiments on large-scale supervised training, we also provide semi-automatically created figure-ground annotation for all images of the two datasets, based on the following procedure. For the images with bounding box annotation (6,044 of the *horse* images and 42,763 of the *dog* images), we apply the *segmentation transfer* method of [18] to the bounding box region. A visual inspection of the output (see Figure 3) shows that the resulting segmentations are of high accuracy, so using them as a proxy for manual annotation seems justified. For the remaining images only a per-image class label is known. To these images we apply the unconstrained segmentation transfer, which also yields figure–ground segmentation mask, but of mixed quality (see Figure 3). One aspect we study in our experimental evaluation is in how far the quality of image segmentation methods can be improved by adding such semi-automatically generated annotation to the training data.

Note that even though part of the training annotation is generated algorithmically, the evaluation is performed with respect to manually created ground truth, so the evaluation procedure is not biased towards the label transfer method.

For our experiments on *HorseSeg* and *DogSeg* we use the same feature representation as for the Stanford dataset, except that we add the output of per-pixel classifiers for foreground and background as two additional dimensions. The questions we would like to answer are: *1) can we scale CRF training to truly large datasets?* and *2) what is the effect of relying on semi-automatically generated annotation for this task?*

We study these questions in three sets of experiments using different subsets of the training data: (a) only images with manually created annotation, (b) images with annotation created manually or by segmentation transfer using bounding box information, (c) all training images. In each case we train CRFs with pairwise terms using the linear and nonlinear variants of LS-CRF, and we compare their segmentation performance to models with only unary terms (which is always computationally feasible). In situation (a), we use the feature vectors of all available superpixels to train the unary-only models, and all neighboring superpixel pairs to train LS-CRF with pairwise terms. In the larger setup, (b) and (c), we reduce the redundancy in the data by using only 25% of all superpixels for the unary-only models, sampled in a class-balanced way. For pairwise models, we record the ratio of pairs with *same label* versus with *different label*. Preserving this ratio, we sample 10% of all superpixels pairs, in a way that combinations with both *foreground* and both *background* are equally likely, and also *foreground/background* and *background/foreground* transitions are equally likely. The percentages are chosen such that the training problems in both situations are of comparable size. On the *DogSeg* dataset, they consists of approximately

Table 2. Results of LS-CRF for figure-ground segmentation on different subsets of the *HorseSeg* and *DogSeg* datasets. Top table: training time (in minutes), bottom row: segmentation accuracy (per-class average in %). Row indicate which subset of the training data was used; manual: only images with manually created annotation, bbox: additionally all images with annotation created automatically from bounding boxes, all: additionally all images with annotation created from per-image class labels.

a) **Training time**	linear/unary	linear/pairwise	nonlinear/unary	nonlinear/pairwise
HorseSeg - manual (147 images)	<1m	<1m	<1m	1.5m
HorseSeg - bbox (6K images)	<1m	<1m	9m	32m
HorseSeg - all (25K images)	1m	2m	88m	354m
DogSeg - manual (249 images)	2.5m	2.5m	<1m	2m
DogSeg - bbox (43K images)	7m	14m	110m	348m
DogSeg - all (156K images)	20m	46m	519m	668m

b) **Accuracy**	linear/unary	linear/pairwise	nonlinear/unary	nonlinear/pairwise
HorseSeg - manual (147 images)	81.4	82.2	81.6	83.8
HorseSeg - bbox (6K images)	82.0	83.6	83.6	86.4
HorseSeg - all (25K images)	81.4	82.5	83.3	84.5
DogSeg - manual (249 images)	77.8	78.5	79.1	80.8
DogSeg - bbox (43K images)	78.5	80.9	81.2	83.8
DogSeg - all (156K images)	78.1	80.0	80.2	82.2

90K data points for situation (a), 3.5M data points for (b), and 13M data points for (c), except the pairwise/nonlinear case, where we use only 6.5M data points for memory reasons. On the *HorseSeg*, the number are roughly half as big for (a), and one sixth for (b) and (c).

The first result is that we can answer question 1) in the positive. Table 2a) lists the approximate training time on a 12 core workstation with enabled hyper-threading, not including the overhead from feature extraction. The table shows that training linear LS-CRF is highly efficient, requiring less than one hour even for the largest dataset. The use of pairwise terms instead of just unaries only incurs a modest slowdown. For comparison, even if probabilistic inference were possible within less than one second per image, training a CRF by ordinary maximum likelihood learning on this dataset would take days or weeks. Training nonlinear predictors is computationally more expensive, but still feasible.

Figure 4 illustrates some segmentation result. More images are provided in the supplemental material. Table 2b) shows a quantitative evaluation of the segmentation accuracy of the different setups. The results allow us to make several observations that we believe will be relevant to other researchers in the area. First, it has been reported previously that pairwise terms have only a minor beneficial effect for image segmentation tasks (e.g. [14]). Our experiments confirm this observation when a linear representation is used and the number of training examples is small. However, when a large training set was used, the difference between unary-only and pairwise models increases.

Second, the use of nonlinear predictors consistently improved the segmentation quality, even though this comes at a significant cost in training time. This can be a useful insight also for other CRF training methods, which rely predominantly on linearly parameterized energy functions.

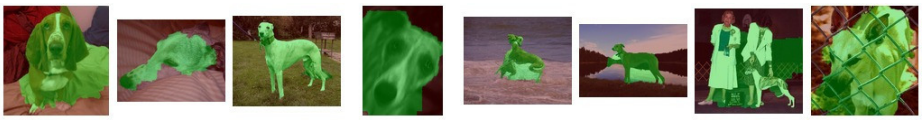


Fig. 4. Example segmentations by LS-CRF for the *DogSeg* dataset

Third, we see a substantial improvement of the segmentation quality when increasing the number of training images by adding images that had their annotation created automatically from bounding boxes information. This indicates that segmentation transfer followed by large-scale CRF learning could be a promising way for leveraging large amounts of images for training even without manually labelling them with pixelwise segmentations. Adding also images with annotation created just from per-image class labels still improves results over not using additional images, but decreases the quality compared to the situation when only images with bounding box information were used. This observation suggests that a tradeoff between the number of training examples (i.e. data collection) and the quality of annotations (i.e. labeling effort) is necessary. We plan to study this aspect in future work.

As a first step in this direction, we performed additional experiments in which we measure how useful the annotated annotation is by itself. For the *HorseSeg* dataset, we trained using exclusively the images with annotation from bounding boxes, or on images with annotation from per-image labels, i.e. we did not use the images with manually created annotation during training. This resulting per-class accuracies are 82.0 (unary) and 83.9 (pairwise) for the bounding box case, and 81.1 (unary) and 82.8 (pairwise) for the per-image case. Comparing this to the values 81.4 (unary) and 82.2 (pairwise) from Table 2b), we see that the generated segmentations do contain useful information. Even training only on images with segmentations inferred from their class label yields comparable results to training on the (much smaller) set of manually annotated images.

6 Summary

In this work we make two main contributions: 1) we present LS-CRF a new technique for inference-free CRF training that scales to very large training sets. Because of its simplicity and flexibility, we believe it has the potential to become a standard tool for training cyclic CRFs. Moreover, in future work we plan to consider approximations other then separable bound (6), for example, the Bethe approximation [33]. 2) we introduce two new benchmark datasets consisting of over 180,000 images and segmentation masks. It is meant to facilitate research on large-scale CRF training and in particular on image segmentation, which so far is held back by the lack of suitably large-scale datasets.

From our experimental evaluation we obtain first insights in this direction: we saw that the positive effect of pairwise terms increased with the size of the training set, that training CRFs with nonlinear energies is feasible and results in

better segmentation models, and that learning with semi-automatic annotation is practical and useful for image segmentation.

Acknowledgements. We would like to thank Yandex LLC for providing the *MatrixNet* package. This work was in parts funded by the European Research Council under the European Unions Seventh Framework Programme (FP7/2007-2013)/ERC grant agreement no 308036.

References

1. Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P., Süsstrunk, S.: SLIC superpixels compared to state-of-the-art superpixel methods. PAMI 34(11) (2012)
2. Andres, B., Beier, T., Kappes, J.H.: OpenGM: A C++ library for discrete graphical models. ArXiv e-prints 1206.0111 (2012), `http://arxiv.org/abs/1206.0111`
3. Arandjelović, R., Zisserman, A.: Three things everyone should know to improve object retrieval. In: CVPR (2012)
4. Besag, J.: Statistical analysis of non-lattice data. The Statistician (1975)
5. Bottou, L., Bousquet, O.: The tradeoffs of large scale learning. In: NIPS (2007)
6. Bulatov, A., Grohe, M.: The complexity of partition functions. Theoretical Computer Science 348(2) (2005)
7. Byrd, R.H., Lu, P., Nocedal, J., Zhu, C.: A limited memory algorithm for bound constrained optimization. SIAM Journal on Scientific Computing (SISC) 16(5), 1190–1208 (1995)
8. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: ImageNet: A large-scale hierarchical image database. In: CVPR (2009)
9. Domke, J.: Learning graphical model parameters with approximate marginals inference. PAMI (2013)
10. Domke, J.: Structured learning via logistic regression. In: NIPS (2013)
11. Everingham, M., van Gool, L., Williams, C., Winn, J., Zisserman, A.: The Pascal visual object classes (VOC) challenge. IJCV 88(2) (2010)
12. Finley, T., Joachims, T.: Training structural SVMs when exact inference is intractable. In: ICML (2008)
13. Friedman, J.H.: Greedy function approximation: a gradient boosting machine. Annals of Statistics (2001)
14. Fulkerson, B., Vedaldi, A., Soatto, S.: Class segmentation and object localization with superpixel neighborhoods. In: ICCV (2009)
15. Gould, S.: Multiclass pixel labeling with non-local matching constraints. In: CVPR, pp. 2783–2790 (2012)
16. Gould, S., Fulton, R., Koller, D.: Decomposing a scene into geometric and semantically consistent regions. In: ICCV (2009)
17. Gould, S., Zhang, Y.: PATCHMATCHGRAPH: Building a graph of dense patch correspondences for label transfer. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012, Part V. LNCS, vol. 7576, pp. 439–452. Springer, Heidelberg (2012)
18. Guillaumin, M., Kuettel, D., Ferrari, V.: ImageNet Auto-annotation with Segmentation Propagation. IJCV (2014)
19. Kohli, P., Shekhovtsov, A., Rother, C., Kolmogorov, V., Torr, P.: On partial optimality in multi-label MRFs. In: ICML (2008)

20. Kolmogorov, V.: Convergent tree-reweighted message passing for energy minimization. PAMI 28(10) (2006)
21. Komodakis, N.: Efficient training for pairwise or higher order CRFs via dual decomposition. In: CVPR, pp. 1841–1848 (2011)
22. Lempitsky, V.S., Vedaldi, A., Zisserman, A.: A pylon model for semantic segmentation. NIPS 24, 1485–1493 (2011)
23. Meshi, O., Sontag, D., Jaakkola, T., Globerson, A.: Learning efficiently with approximate inference via dual losses (2010)
24. Nowozin, S., Lampert, C.H.: Structured learning and prediction in computer vision. Foundations and Trends in Computer Graphics and Vision 6 (2011)
25. Nowozin, S., Rother, C., Bagon, S., Sharp, T., Yao, B., Kohli, P.: Decision tree fields. In: ICCV (2011)
26. Schroff, F., Criminisi, A., Zisserman, A.: Object class segmentation using random forests. In: BMVC (2008)
27. 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(1) (2009)
28. Sontag, D., Meshi, O., Globerson, A., Jaakkola, T.S.: More data means less inference: A pseudo-max approach to structured learning. In: NIPS (2010)
29. Sutton, C., McCallum, A.: Piecewise training of undirected models. In: UAI (2005)
30. Tighe, J., Lazebnik, S.: SuperParsing: Scalable nonparametric image parsing with superpixels. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part V. LNCS, vol. 6315, pp. 352–365. Springer, Heidelberg (2010)
31. Trofimov, I., Kornetova, A., Topinskiy, V.: Using boosted trees for click-through rate prediction for sponsored search. In: International Workshop on Data Mining for Online Advertising and Internet Economy (2012)
32. Tsochantaridis, I., Joachims, T., Hofmann, T., Altun, Y.: Large margin methods for structured and interdependent output variables. JMLR 6 (2005)
33. Wainwright, M.J., Jordan, M.I.: Graphical models, exponential families, and variational inference. Foundations and Trends in Machine Learning (2008)