Non-Parametric CRFs for Image Labeling

Jeremy Jancsary **Sebastian Nowozin** **Carsten Rother**

Microsoft Research Cambridge

Abstract

We introduce a powerful non-parametric image labeling framework, Regression Tree Fields (RTFs), and discuss its application to image restoration. The conditional structure and the parameters of our model are estimated from training data so as to directly optimize for popular performance measures, resulting in excellent predictive performance at low computational cost.

1 Introduction

Probabilistic graphical models have emerged as a standard tool for building computer vision models [1, 2]. They allow us to make predictions given noisy image observations by relating the observed image to the variables of interest in a coherent way.

There are three *key challenges* that need to be overcome in order to use graphical models to solve computer vision tasks: parameterization, inference, and learning. *Parameterization* is the specification of the model structure and its parameters that need to be estimated from training data. *Inference* refers to the test-time task of reasoning about the state of the variables that interest us, given the observation. *Learning* means to estimate model parameters from training data so as to make good predictions at test-time. All these tasks are related to each other and even for simple models turn out to be intractable, necessitating approximations in model specification, inference, and estimation [2].

Our work addresses all three challenges and provides an efficient, yet highly effective framework for structured prediction tasks that allows us to surpass the state of the art in important applications, as illustrated via the real-world problem of image restoration in this paper.

2 Model

Our model is a Gaussian conditional random field, the factors of which are determined by means of non-parametric regression trees in a manner that will be made precise shortly.

2.1 Gaussian Conditional Random Fields

Gaussian CRFs were first introduced in [3]. Consider an observed input image $\mathbf{x}$, and a corresponding labeling, the output image $\mathbf{y}$. Gaussian CRFs model the probability of each output given an input image, $p(\mathbf{y} \mid \mathbf{x}; \mathbf{w}) \propto \exp[-E(\mathbf{y} \mid \mathbf{x}; \mathbf{w})]$, via a quadratic energy

$$E(\mathbf{y} \mid \mathbf{x}; \mathbf{w}) \stackrel{\text{def}}{=} \tfrac{1}{2}\mathbf{y}^\mathsf{T}\mathbf{Q}(\mathbf{x}; \mathbf{w})\mathbf{y} - \mathbf{y}^\mathsf{T}\mathbf{l}(\mathbf{x}; \mathbf{w}). \tag{1}$$

Together with the input $\mathbf{x}$, the model parameters $\mathbf{w}$ determine the coefficients $\mathbf{Q}(\mathbf{x}; \mathbf{w}) \succ 0$ and $\mathbf{l}(\mathbf{x}; \mathbf{w})$ of the energy. For a given input $\mathbf{x}$, the prediction $\hat{\mathbf{y}}(\mathbf{x}; \mathbf{w})$ under this model is given by

$$\hat{\mathbf{y}}(\mathbf{x}; \mathbf{w}) \stackrel{\text{def}}{=} \operatorname*{argmax}_{\mathbf{y}} p(\mathbf{y} \mid \mathbf{x}; \mathbf{w}) = [\mathbf{Q}(\mathbf{x}; \mathbf{w})]^{-1}\mathbf{l}(\mathbf{x}; \mathbf{w}), \tag{2}$$

which is typically found by solving the sparse linear system $\mathbf{Q}(\mathbf{x}; \mathbf{w})\mathbf{y} = \mathbf{l}(\mathbf{x}; \mathbf{w})$. The solution is both the mean and the mode of the Gaussian density $p(\mathbf{y} \mid \mathbf{x}; \mathbf{w})$.

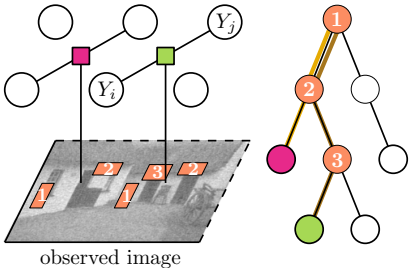

Figure 1: Illustration of how regression trees and random fields are combined in an RTF: A pairwise factor type is instantiated on a grid of random variables. At each instantiation, a tree is evaluated on the surrounding image content, performing a sequence of tests (1, 2, 3) until a leaf node is reached. The selected leaf node determines the effective interaction used for the factor. The conditional model now becomes a Gaussian random field, enabling efficient inference as a solution to a linear system.

2.2 Parameter Estimation

Given i.i.d. training data $\mathcal{D} = \{(\mathbf{x}^{(i)}, \mathbf{y}^{(i)})\}_{i=1}^{N}$, one may wish to use the maximum likelihood estimate (MLE) of the model parameters, due to its favorable asymptotic properties [4]:

$$\hat{\mathbf{w}}_{\text{MLE}} = \operatorname{argmin}_{\mathbf{w}\in\Omega} - \textstyle\sum_i \log p(\mathbf{y}^{(i)} \mid \mathbf{x}^{(i)}; \mathbf{w}), \tag{3}$$

where constraint set Ω enforces positive-definiteness of the precision matrix $\mathbf{Q}(\mathbf{x}; \mathbf{w})$. However, to optimize (3), one must repeatedly compute the *mean parameters*

$$\boldsymbol{\mu} \stackrel{\text{def}}{=} \mathbb{E}_{\mathbf{y}\sim p(\mathbf{y}|\mathbf{x};\mathbf{w})}[\mathbf{y}] \text{ and } \boldsymbol{\Sigma} \stackrel{\text{def}}{=} \mathbb{E}_{\mathbf{y}\sim p(\mathbf{y}|\mathbf{x};\mathbf{w})}[\mathbf{y}\mathbf{y}^{\mathbf{T}}]. \tag{4}$$

The complexity of the latter computation is cubic in the number of pixels and hence prohibitive even for instances of modest size.

2.2.1 Maximum Pseudolikelihood Estimation

In [5], we overcome the computational limitations of MLE by maximizing the *pseudolikelihood* [6] instead (MPLE), where the model parameters are estimated as

$$\hat{\mathbf{w}}_{\text{MPLE}} = \operatorname{argmin}_{\mathbf{w}\in\Omega} - \textstyle\sum_i \sum_{j\in\mathcal{V}} \log p(\mathbf{y}_j^{(i)} \mid \mathbf{y}_{\mathcal{V}\setminus j}^{(i)}, \mathbf{x}^{(i)}; \mathbf{w}). \tag{5}$$

Hence, the objective decomposes into likelihoods of single pixels $j \in \mathcal{V}$ conditioned on the observed labels of the other pixels. For $\mathbf{y}_j \in \mathbb{R}^D$, these conditioned subgraphs are just D-dimensional Gaussians, so the mean parameters $\boldsymbol{\mu}_{j|\mathcal{V}\setminus j} \in \mathbb{R}^D$ and $\boldsymbol{\Sigma}_{j|\mathcal{V}\setminus j} \in \mathbb{R}^{D\times D} \succ 0$ are low-dimensional, which renders the approach efficient.

2.2.2 Empirical Risk Minimization

However, MPLE (just like MLE) fails to take into account the loss function that is ultimately applied to the predictions of a system at test time. Hence, in [7], we estimate the model parameters from training data by minimizing the empirical risk

$$R_\ell(\mathcal{D}, \mathbf{w}) \stackrel{\text{def}}{=} \frac{1}{N}\sum_i^N \ell\left(\hat{\mathbf{y}}(\mathbf{x}^{(i)}; \mathbf{w}), \mathbf{y}^{(i)}\right) \approx \mathbb{E}_{p(\mathbf{y},\mathbf{x})}\left[\ell\left(\hat{\mathbf{y}}(\mathbf{x}; \mathbf{w}), \mathbf{y}\right)\right]. \tag{6}$$

The loss function $\ell(\hat{\mathbf{y}}, \mathbf{y}) : \mathcal{Y} \times \mathcal{Y} \mapsto \mathbb{R}_+$ measures the error present in the prediction $\hat{\mathbf{y}}$ relative to the ground truth $\mathbf{y}$. By choosing $\hat{\mathbf{w}}_{\text{ERM}} = \operatorname{argmin}_{\mathbf{w}} R(\mathcal{D}, \mathbf{w})$, the model is determined such that its predictions incur the least possible loss on the training data. As we will demonstrate empirically, this approach has benefits beyond the computational perspective.

2.3 Regression Tree Fields

Regression tree fields [5] extend the original Gaussian CRF model in two ways. First, the energy of a labeling is specified in terms of local models over subsets of pixels. The parameters of these local models include a linear term and an inverse covariance matrix, both estimated from data. Second, the local model that is in effect at a given position of the image is determined via a regression tree (cf. Fig. 1), rendering the approach non-parametric.

2.3.1 Parameterization

Each local energy term working on a subset of pixels is called a *factor* and denoted by F. The components of $\mathbf{y}$ corresponding to the pixels covered by factor F will be denoted by column vector $\mathbf{y}_F$. Factors sharing the same parameters are grouped into *types*. The set of all factors F of type t is denoted by $\mathcal{F}_t$. Each factor type t defines a regression tree that stores at its leaves $l \in \mathcal{L}_t$ a set of parameters $\mathbf{w}_t = \{\mathbf{L}_t^{(l)}, \mathbf{Q}_t^{(l)}\}_{l \in \mathcal{L}_t}$. We define $\mathbf{L}_t(\mathbf{x}_F)$ and $\mathbf{Q}_t(\mathbf{x}_F)$ as maps to the parameters $\mathbf{Q}_t^{(l_*)}$ and $\mathbf{L}_t^{(l_*)}$ of the leaf l_* that was selected for factor F of type t given the observed input image $\mathbf{x}$.

The energy of a particular factor F of type t then assumes the form

$$E_t(\mathbf{y}_F \mid \mathbf{x}_F; \mathbf{w}_t) \stackrel{\text{def}}{=} \tfrac{1}{2}\mathbf{y}_F^\mathsf{T}\mathbf{Q}_t(\mathbf{x}_F)\mathbf{y}_F - \mathbf{y}_F^\mathsf{T}\mathbf{L}_t(\mathbf{x}_F)\mathbf{b}_t(\mathbf{x}_F), \tag{7}$$

where $\mathbf{b}_t(\mathbf{x}_F) \in \mathbb{R}^{B_t}$ is a linear basis vector whose dimensionality depends on the factor type. In the simplest case, this term is constant, $\mathbf{b}_t(\cdot) = 1 \in \mathbb{R}$, but in addition, we use more general image features in our experiments. For the parameters, if $\mathbf{y}_F \in \mathbb{R}^{D_t}$, we have $\mathbf{L}_t(\mathbf{x}_F) \in \mathbb{R}^{D_t \times B_t}$ and $\mathbf{Q}_t(\mathbf{x}_F) \in \mathbb{R}^{D_t \times D_t} \succ 0$. The positive-definiteness constraint on the latter implies that all $\mathbf{Q}_t^{(l)}$ parameters must also be positive-definite and ensures that

$$E(\mathbf{y} \mid \mathbf{x}; \mathbf{w}) \stackrel{\text{def}}{=} \sum_t \sum_{F \in \mathcal{F}_t} E_t(\mathbf{y}_F \mid \mathbf{x}_F; \mathbf{w}_t) \tag{8}$$

leads to a Gaussian density $p(\mathbf{y} \mid \mathbf{x}; \mathbf{w}) \propto \exp[-E(\mathbf{y} \mid \mathbf{x}; \mathbf{w})]$. Predictions are obtained as in (2).

3 Loss-Specific Training of Regression Tree Fields

The RTF parameterization is powerful, but how can we find trees and leaf parameters to minimize the empirical risk? Ideally both the structure of the regression trees as well as their parameters are jointly chosen to minimize this objective. But because the RTF model is a random field, all parts of the model interact with each other and this makes joint minimization challenging. Nonetheless, joint loss-based training is indeed possible at reasonable computational cost.

The intuition behind this is as follows: In [5], we already showed that for the pseudo-likelihood objective, it is possible to efficiently split tree nodes based on the largest increase in gradient norm. An increase in the gradient norm indicates that further decrease in the objective function is possible and hence indicates a good split. This approach avoids the mistake of learning trees separately, but still does not account for the loss function at test time.

Consequently, in [7], we extended this approach to *any* differentiable loss function ℓ, which indeed results in considerable gains in practice.

4 Application to Image Denoising and Deblocking

The image restoration problem maps into our framework as follows. The observed input image $\mathbf{x}$ denotes the corrupted image, which is generated from ground truth $\mathbf{y}$ via some perturbation process. In the classical image denoising setting, an additive white Gaussian noise assumption is made, that is, $\mathbf{x} = \mathbf{y} + \mathbf{z}$ for $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I})$. This is the setting we will consider here; however, we emphasize that our model is also well-suited for structured noise, such as JPEG blocking artefacts [7].

In any case, the restored image $\hat{\mathbf{y}}$ can be obtained as the prediction of our model given the corrupted input, i.e. $\hat{\mathbf{y}} \stackrel{\text{def}}{=} \hat{\mathbf{y}}(\mathbf{x}; \mathbf{w}) = [\mathbf{Q}(\mathbf{x}; \mathbf{w})]^{-1}\mathbf{l}(\mathbf{x}; \mathbf{w})$.

5 Experiments

We adhere to a strict experimental protocol, using the disjoint training, validation and test splits from the BSDS500 database [9] (images scaled by a factor of 0.5).

The images are perturbed with additive white Gaussian noise (AWGN), for noise levels $\sigma \in \{20, 30, 40, 50\}$. The results achieved by our system configurations, as well as the strongest competitors, are shown in Table 1.

Ground truth **Input** ($\sigma = 40$) **Out.** PSNRRTF$_{\text{ALL}}$ **Output** EPLL

Figure 2: Visual improvement in denoising quality: Our PSNRRTF$_{\text{ALL}}$-system clearly produces more natural restorations than EPLL [8], the strongest competitor.

Method	**PSNR** (↑ better)				**MAE** (↓ better)				**SSIM** (↑ better)			
σ	20	30	40	50	20	30	40	50	20	30	40	50
Input	22.11	18.59	16.09	14.15	15.96	23.93	31.91	39.89	0.541	0.401	0.307	0.242
FoE [10]	28.87	26.81	25.45	24.47	6.79	8.56	10.03	11.24	0.848	0.776	0.712	0.660
BM3D [11]	29.25	27.32	25.98	25.09	6.40	7.95	9.25	10.22	0.855	0.793	0.741	0.699
LSSC [12]	29.40	27.39	26.08	25.09	6.39	7.96	9.23	10.33	0.861	0.799	0.745	0.700
EPLL [8]	29.38	27.44	26.17	25.22	6.37	7.90	9.12	10.17	0.864	0.800	0.747	0.703
UNIFORMAVG	29.47	27.50	26.21	25.25	6.30	7.84	9.08	10.12	0.863	0.802	0.749	0.705
PSNRRTF$_{\text{PLAIN}}$	28.95	26.97	25.71	24.76	6.78	8.44	9.72	10.85	0.840	0.771	0.716	0.666
PSNRRTF$_{\text{ALL}}$	29.67	27.72	26.43	25.51	6.14	7.62	8.80	9.78	0.868	0.809	0.758	0.717
MAERTF$_{\text{PLAIN}}$	28.92	26.94	25.69	24.75	6.78	8.43	9.71	10.81	0.840	0.771	0.715	0.669
MAERTF$_{\text{ALL}}$	29.67	27.72	26.43	25.50	6.12	7.59	8.77	9.74	0.867	0.808	0.758	0.717
SSIMRTF$_{\text{PLAIN}}$	28.49	26.55	25.31	24.41	7.17	8.92	10.23	11.34	0.844	0.778	0.721	0.676
SSIMRTF$_{\text{ALL}}$	29.23	27.14	25.67	24.75	6.60	8.39	9.96	11.06	0.872	0.815	0.766	0.726
NLPLRTF$_{\text{PLAIN}}$	28.61	26.66	25.32	24.42	7.09	8.80	10.28	11.37	0.828	0.758	0.694	0.653
NLPLRTF$_{\text{ALL}}$	29.60	27.64	26.34	25.40	6.20	7.71	8.92	9.93	0.866	0.806	0.755	0.714

Table 1: Denoising test set results. We compare state-of-the-art competitors to configurations of our method (RTF). For each measure, the result of the strongest competitor is printed in blue, and the best RTF result is printed in green. The gain of our method is statistically significant as per Wilcoxon signed-ranks test ($p < 10^{-5}$ for each blue-green pair in each column).

We consider eight configurations of our method, based on the combinations of loss functions we optimize (PSNRRTF, MAERTF, SSIMRTF, NLPLRTF) and two different feature sets: using only a generic filterbank (RTF$_{\text{PLAIN}}$), as well as the filterbank *and* predictions by the competing methods FoE, BM3D, LSSC and EPLL (RTF$_{\text{ALL}}$). Note that the NLPLRTF-systems are trained to minimize the negative log-pseudolikelihood, so NLPLRTF$_{\text{PLAIN}}$ corresponds to the system in [5].

In all cases, an RTF$_{\text{ALL}}$-system trained for the specific loss achieves the best result. In terms of PSNR, the gains over the best published method range from 0.26dB to 0.29dB across the different noise levels. This is a substantial improvement and is clearly visible, as shown in Fig. 2. The gains are even more pronounced in terms of MAE and SSIM. Note that it is not at all apparent how the other systems could be made to take into account these measures.

Observe that RTFs trained for a specific loss perform much better than the NLPLRTF of [5]. The impressive difference between PSNRRTF$_{\text{PLAIN}}$ and NLPLRTF$_{\text{PLAIN}}$ ranges from 0.31db to 0.39db. This gap narrows as more powerful features are added to the models, but remains statistically significant.

6 Conclusion

We proposed a novel framework for non-parametric conditional random fields, based on two ideas. First, non-parametric regression trees as a flexible representation. Second, loss-specific training, selecting all aspects of the model so as to optimize a task-specific loss. Jointly, these ideas result in significant gains in the important real-world problem of image denoising.

References

[1] Blake, A., Kohli, P., Rother, C.: Markov random fields for vision and image processing. MIT Press (2011)

[2] Koller, D., Friedman, N.: Probabilistic Graphical Models: Principles and Techniques. MIT Press (2009)

[3] Tappen, M.F., Liu, C., Adelson, E.H., Freeman, W.T.: Learning Gaussian conditional random fields for low-level vision. In: CVPR. (2007)

[4] Hogg, R.V., McKean, J.W., Craig, A.T.: Introduction to Mathematical Statistics. Pearson Education (2005)

[5] Jancsary, J., Nowozin, S., Sharp, T., Rother, C.: Regression tree fields – An efficient, non-parametric approach to image labeling problems. In: CVPR. (2012)

[6] Besag, J.: Efficiency of pseudolikelihood estimation for simple Gaussian fields. Biometrica (64) (1977) 616–618

[7] Jancsary, J., Nowozin, S., Rother, C.: Loss-specific training of non-parametric image restoration models: A new state of the art. In: ECCV. (2012)

[8] Zoran, D., Weiss, Y.: From learning models of natural image patches to whole image restoration. In: ICCV. (2011)

[9] Arbelaez, P., Maire, M., Fowlkes, C., Malik, J.: Contour detection and hierarchical image segmentation. IEEE Trans. Pattern Anal. Mach. Intell. **33**(5) (2011) 898–916

[10] Schmidt, U., Gao, Q., Roth, S.: A generative perspective on MRFs in low-level vision. In: CVPR. (2010)

[11] Dabov, K., Foi, A., Katkovnik, V., Egiazarian, K.O.: Image denoising by sparse 3-D transform-domain collaborative filtering. IEEE Trans. Image Process. **16**(8) (2007) 2080–2095

[12] Mairal, J., Bach, F., Ponce, J., Shapiro, G., Zisserman, A.: Non-local sparse models for image restoration. In: ICCV. (2009)