Conditional Random Fields as Recurrent Neural Networks

Shevchenko Alexander

Higher School of Economics

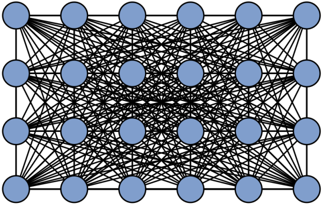
22 January, 2018

План

1. Background
2. Message Passing
3. Contribution
4. Experiments

The Fully-Connected CRF Model

- Random field $\mathcal{X} = \{x_1, \dots, x_N\}$
- Domain set $\mathcal{L} = \{l_1, \dots, l_k\}$
- $\mathcal{I}$ denotes an input image



Fully-Connected CRF energy function:

$$E(\mathcal{X}|\mathcal{I}) = \sum_i \psi_u(x_i) + \sum_{i<j} \psi_p(x_i, x_j)$$

Posterior:

$$P(\mathcal{X}|\mathcal{I}) = \frac{1}{Z(I)} \exp\{-E(\mathcal{X}|\mathcal{I})\}$$ **Intractable!**

Mean-Field Approximation

Variational posterior is searched in form:

$$Q(\mathcal{X}) = \prod_i Q_i(x_i)$$

Mean-Field Update:

$$\log Q_i(x_i) = \mathbb{E}_{Q_j \neq Q_i} \log P(\mathcal{X}|\mathcal{I}) + \text{const}$$

Which leads with current energy to:

$$Q_i(x_i) = \frac{1}{Z_i} \exp\left\{ -\psi_u(x_i) - \sum_{i \neq j} \mathbb{E}_{U_j \sim Q_j} \psi_p(x_i, U_j) \right\}$$

Still bad - big images would be a problem.

Gaussian Edge Potentials

Let pairwise potentials be in form:

$$\psi_p(x_i, x_j) = \mu(x_i, x_j) \underbrace{\sum_{m=1}^{K} w^{(m)} k^{(m)}(f_i, f_j)}_{k(f_i, f_j)}$$

where each $k^{(m)}(f_i, f_j) = \exp\left\{-0.5(f_i - f_j)^T \Lambda^{(m)} (f_i - f_j)\right\}$

We focus on particular kernel type:

$$w^{(1)} \exp \underbrace{\left(-\frac{\|p_i - p_j\|_2^2}{2\theta_\alpha^2} - \frac{\|I_i - I_j\|_2^2}{2\theta_\beta^2}\right)}_{\text{appearance kernel}} + w^{(2)} \exp \underbrace{\left(-\frac{\|p_i - p_j\|_2^2}{2\theta_\gamma^2}\right)}_{\text{smoothness kernel}}$$

Message Passing

$$Q_i(l') = \frac{1}{Z_i} \exp\left\{ -\psi_u(l') - \sum_{l \in \mathcal{L}} \mu(l', l) \sum_{m=1}^{K} w^{(m)} \sum_{i \neq j} k^{(m)}(f_i, f_j) Q_j(l) \right\}$$

Algorithm 1 Mean field in fully connected CRFs

Initialize Q ▷ $Q_i(x_i) \leftarrow \frac{1}{Z_i} \exp\{-\phi_u(x_i)\}$

while not converged **do** ▷ See Section 6 for convergence analysis

$\tilde{Q}_i^{(m)}(l) \leftarrow \sum_{j \neq i} k^{(m)}(\mathbf{f}_i, \mathbf{f}_j) Q_j(l)$ for all m ▷ **Message passing** from all X_j to all X_i

$\hat{Q}_i(x_i) \leftarrow \sum_{l \in \mathcal{L}} \mu^{(m)}(x_i, l) \sum_m w^{(m)} \tilde{Q}_i^{(m)}(l)$ ▷ **Compatibility transform**

$Q_i(x_i) \leftarrow \exp\{-\psi_u(x_i) - \hat{Q}_i(x_i)\}$ ▷ **Local update**

normalize $Q_i(x_i)$

end while

Efficient way (Optinal)

$$\widetilde{Q}_i^{(m)}(l) = \underbrace{\sum_{i \in \mathcal{V}} k^{(m)}(f_i, f_j) Q_j(l) - Q_i(l)}_{\text{message passing}} = \underbrace{[G_{\Lambda^{(m)}} * Q(l)](f_i)}_{\bar{Q}_i^{(m)}(l)} - Q_i(l)$$

Algorithm 2 Efficient message passing: $\overline{Q}_i^{(m)}(l) = \sum_{j \in \mathcal{V}} k^{(m)}(\mathbf{f}_i, \mathbf{f}_j) Q_j(l)$

$Q_\downarrow(l) \leftarrow \text{downsample}(Q(l))$	▷ **Downsample**
$\forall_{i \in \mathcal{V}_\downarrow} \overline{Q}_{\downarrow i}^{(m)}(l) \leftarrow \sum_{j \in \mathcal{V}_\downarrow} k^{(m)}(\mathbf{f}_{\downarrow i}, \mathbf{f}_{\downarrow j}) Q_{\downarrow j}(l)$	▷ **Convolution** on samples $\mathbf{f}_\downarrow$
$\overline{Q}^{(m)}(l) \leftarrow \text{upsample}(\overline{Q}_\downarrow^{(m)}(l))$	▷ **Upsample**

Futher improvement: Fast High-Dimensional Filtering Using the Permutohedral Lattice.

A Mean-field Iteration as a Stack of CNN Layers

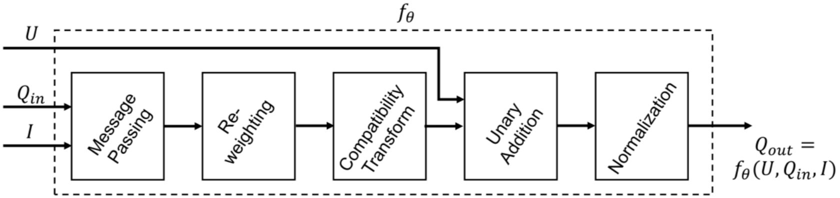
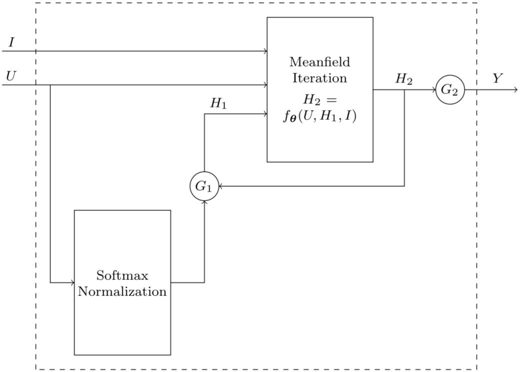


Figure 1. **A mean-field iteration as a CNN.** A single iteration of the mean-field algorithm can be modelled as a stack of common CNN layers.

A Mean-field Iteration as a Stack of CNN Layers

- **Initialization:** aka softmax.
- **Message passing:** as described above.
- **Weighting Filter Outputs:** 1×1 convolution with K input channels and 1 output channel.
- **Compatibility transform:** 1×1 convolution with k input channels and k output channels. (since accounting on each class interaction we get a matrix).
- **Adding Unary term:** Just element-wise subtraction.
- **Normalization:** aka softmax.

CRF as RNN



$$H_1(t) = \begin{cases} \text{softmax}(U), & t = 0 \\ H_2(t-1), & 0 < t \leq T, \end{cases}$$

$$H_2(t) = f_{\boldsymbol{\theta}}(U, H_1(t), I), \quad 0 \leq t \leq T,$$

$$Y(t) = \begin{cases} 0, & 0 \leq t < T \\ H_2(t), & t = T. \end{cases}$$

Experiments

Experiments

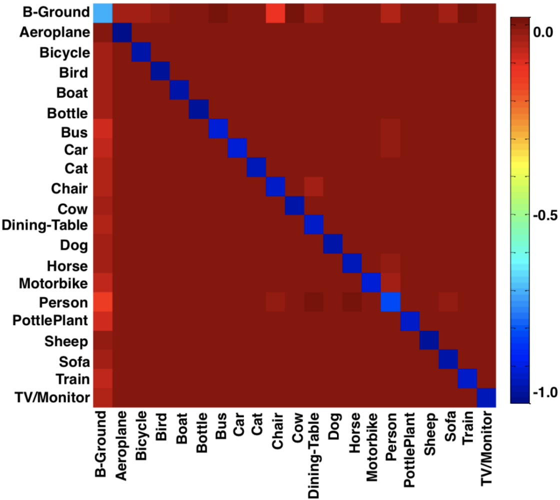
Method	Without COCO	With COCO
Plain FCN-8s	61.3	68.3
FCN-8s and CRF disconnected	63.7	69.5
End-to-end training of CRF-RNN	69.6	72.9

Table 1. Mean IU accuracy of our approach, CRF-RNN, compared with similar methods, evaluated on the reduced VOC 2012 validation set.

Method	VOC 2010 test	VOC 2011 test	VOC 2012 test
BerkeleyRC [3]	n/a	39.1	n/a
O2PCPMC [8]	49.6	48.8	47.8
Divmbest [44]	n/a	n/a	48.1
NUS-UDS [16]	n/a	n/a	50.0
SDS [23]	n/a	n/a	51.6
MSRA-CFM [13]	n/a	n/a	61.8
FCN-8s [37]	n/a	62.7	62.2
Hypercolumn [24]	n/a	n/a	62.6
Zoomout [38]	64.4	64.1	64.4
Context-Deep-CNN-CRF [35]	n/a	n/a	70.7
DeepLab-MSc [10]	n/a	n/a	71.6
Our method w/o COCO	73.6	72.4	72.0
BoxSup [12]	n/a	n/a	71.0
DeepLab [10, 41]	n/a	n/a	72.7
Our method with COCO	**75.7**	**75.0**	**74.7**

Table 2. Mean IU accuracy of our approach, CRF-RNN, compared to the other approaches on the Pascal VOC 2010-2012 test datasets. Methods from the first group do not use MS COCO data for training. The methods from the second group use both COCO and VOC datasets for training.

Experiments



Источники

Shuai Zheng, Sadeep Jayasumana, Bernardino Romera-Paredes, Vibhav Vineet, Zhizhong Su, Dalong Du, Chang Huang, and Philip H. S. Torr (2015).
Conditional Random Fields as Recurrent Neural Networks.

Vladlen Koltun et.al. (2014).
Efficient Inference in Fully Connected CRFs with Gaussian Edge Potentials.
arXiv:1210.5644.