Gaussian Conditional Random Field Network for Semantic Segmentation - Supplementary Material

Raviteja Vemulapalli[†], Oncel Tuzel[*], Ming-Yu Liu[*], and Rama Chellappa[†]
[†]Center for Automation Research, UMIACS, University of Maryland, College Park.
[*]Mitsubishi Electric Research Laboratories, Cambridge, MA.

Abstract

In Section 1 of this supplementary material, we derive the mean field update equation for the Gaussian distribution used in this paper. Section 2 provides the relevant derivative formulas for backpropagation and Section 3 presents a detailed algorithmic description of the proposed Gaussian CRF network.

Notations: We use bold face small letters to denote vectors and bold face capital letters to denote matrices. We use $\mathbf{A}^\top$, $\mathbf{A}^{-1}$, $|\mathbf{A}|$ and trace$(\mathbf{A})$ to denote the transpose, inverse, determinant and trace of a matrix $\mathbf{A}$, respectively. We use $\|\mathbf{b}\|_2^2$ to denote the squared ℓ_2 norm of a vector $\mathbf{b}$. $\mathbf{A} \succeq 0$ means $\mathbf{A}$ is symmetric and positive semidefinite. We use $\mathcal{R}$ to denote the set of real numbers and $\mathbb{E}$ to denote expectation.

1. Mean field inference

In this work, we model the conditional probability density $P(\mathbf{y}|\mathbf{X})$ as a Gaussian distribution given by

$$\begin{aligned} P(\mathbf{y}|\mathbf{X}) &\propto \exp\left\{-\frac{1}{2}\, E(\mathbf{y}|\mathbf{X})\right\}, \text{ where} \\ E\left(\mathbf{y}|\mathbf{X}\right) &= \sum_i \|\mathbf{y}_i - \mathbf{r}_i\|_2^2 + \sum_{ij} \left(\mathbf{y}_i - \mathbf{y}_j\right)^\top \mathbf{W}_{ij} \left(\mathbf{y}_i - \mathbf{y}_j\right) \\ &= \sum_i \mathbf{y}_i^\top \left(I + \sum_j \mathbf{W}_{ij}\right) \mathbf{y}_i - 2\sum_i \mathbf{r}_i^\top \mathbf{y}_i + \sum_i \mathbf{r}_i^\top \mathbf{r}_i - 2\sum_{ij} \mathbf{y}_i^\top \mathbf{W}_{ij} \mathbf{y}_j \end{aligned} \tag{1}$$

The standard mean field approach approximates the joint Gaussian distribution $P(\mathbf{y}|\mathbf{X})$ using a simpler Gaussian distribution $Q(\mathbf{y}|\mathbf{X})$ which can be written as a product of independent marginals, i.e, $Q(\mathbf{y}|\mathbf{X}) = \prod_i Q_i(\mathbf{y}_i|\mathbf{X})$ [1], where $Q(\mathbf{y}_i|\mathbf{X})$ is a Gaussian distribution with mean $\boldsymbol{\mu}_i \in \mathcal{R}^K$ and covariance $\boldsymbol{\Sigma}_i \in \mathcal{R}^{K\times K}$. The parameters $\{\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i\}$ of Q are obtained by minimizing the KL-divergence between the distributions Q and P.

[1]Note that instead of using marginals of scalar variables y_{ik}, we are using marginals of vector variables $\mathbf{y}_i$.

$$
\begin{aligned}
KL(Q||P) &= \int Q(\mathbf{y}|\mathbf{X}) \log\left[\frac{Q(\mathbf{y}|\mathbf{X})}{P(\mathbf{y}|\mathbf{X})}\right] \\
&= \int Q(\mathbf{y}|\mathbf{X}) \log\left[Q(\mathbf{y}|\mathbf{X})\right] - \int Q(\mathbf{y}|\mathbf{X}) \log\left[P(\mathbf{y}|\mathbf{X})\right] \\
&= \sum_i \int Q_i(\mathbf{y}_i|\mathbf{X}) \log\left[Q_i(\mathbf{y}_i|\mathbf{X})\right] - \int Q(\mathbf{y}|\mathbf{X}) \log\left[P(\mathbf{y}|\mathbf{X})\right] \quad \left(\text{using } Q(\mathbf{y}|\mathbf{X}) = \prod_i Q_i(\mathbf{y}_i|\mathbf{X})\right) \\
&= -\sum_i \frac{1}{2}\log\left[(2\pi e)^K |\mathbf{\Sigma}_i|\right] - \int Q(\mathbf{y}|\mathbf{X}) \log\left[P(\mathbf{y}|\mathbf{X})\right]
\end{aligned} \tag{2}
$$

$$
\begin{aligned}
\{\boldsymbol{\mu}_i^*, \mathbf{\Sigma}_i^*\} &= \underset{\{\boldsymbol{\mu}_i, \mathbf{\Sigma}_i\}}{\text{argmin}} \;\; KL(Q||P) \\
&= \underset{\{\boldsymbol{\mu}_i, \mathbf{\Sigma}_i\}}{\text{argmin}} \;\; -\sum_i \frac{1}{2}\log\left[(2\pi e)^K |\mathbf{\Sigma}_i|\right] - \int Q(\mathbf{y}|\mathbf{X}) \log\left[P(\mathbf{y}|\mathbf{X})\right] \\
&= \underset{\{\boldsymbol{\mu}_i, \mathbf{\Sigma}_i\}}{\text{argmin}} \;\; -\sum_i \log\left[|\mathbf{\Sigma}_i|\right] + \sum_i \int Q(\mathbf{y}|\mathbf{X})\, \mathbf{y}_i^\top \left(I + \sum_j \mathbf{W}_{ij}\right) \mathbf{y}_i - 2\sum_i \int Q(\mathbf{y}|\mathbf{X})\, \mathbf{r}_i^\top \mathbf{y}_i \\
&\qquad\qquad - 2\sum_{ij} \int Q(\mathbf{y}|\mathbf{X})\, \mathbf{y}_i^\top \mathbf{W}_{ij} \mathbf{y}_j \\
&= \underset{\{\boldsymbol{\mu}_i, \mathbf{\Sigma}_i\}}{\text{argmin}} \;\; -\sum_i \log\left[|\mathbf{\Sigma}_i|\right] + \sum_i \mathbb{E}\left[\mathbf{y}_i^\top \left(I + \sum_j \mathbf{W}_{ij}\right) \mathbf{y}_i\right] - 2\sum_i \mathbb{E}\left[\mathbf{r}_i^\top \mathbf{y}_i\right] \\
&\qquad\qquad - 2\sum_{ij} \mathbb{E}\left[\mathbf{y}_i^\top \mathbf{W}_{ij} \mathbf{y}_j\right] \\
&= \underset{\{\boldsymbol{\mu}_i, \mathbf{\Sigma}_i\}}{\text{argmin}} \;\; -\sum_i \log\left[|\mathbf{\Sigma}_i|\right] + \sum_i \mathbb{E}\left[\text{trace}\left(\mathbf{y}_i \mathbf{y}_i^\top \left(I + \sum_j \mathbf{W}_{ij}\right)\right)\right] - 2\sum_i \mathbb{E}\left[\mathbf{r}_i^\top \mathbf{y}_i\right] \\
&\qquad\qquad - 2\sum_{ij} \mathbb{E}\left[\text{trace}\left(\mathbf{y}_j \mathbf{y}_i^\top \mathbf{W}_{ij}\right)\right] \\
&= \underset{\{\boldsymbol{\mu}_i, \mathbf{\Sigma}_i\}}{\text{argmin}} \;\; -\sum_i \log\left[|\mathbf{\Sigma}_i|\right] + \sum_i \text{trace}\left(\mathbb{E}\left[\mathbf{y}_i \mathbf{y}_i^\top\right] \left(I + \sum_j \mathbf{W}_{ij}\right)\right) - 2\sum_i \mathbb{E}\left[\mathbf{r}_i^\top \mathbf{y}_i\right] \\
&\qquad\qquad - 2\sum_{ij} \text{trace}\left(\mathbb{E}\left[\mathbf{y}_j \mathbf{y}_i^\top\right] \mathbf{W}_{ij}\right) \\
&= \underset{\{\boldsymbol{\mu}_i, \mathbf{\Sigma}_i\}}{\text{argmin}} \;\; -\sum_i \log\left[|\mathbf{\Sigma}_i|\right] + \sum_i \text{trace}\left(\left(\mathbf{\Sigma}_i + \boldsymbol{\mu}_i \boldsymbol{\mu}_i^\top\right) \left(I + \sum_j \mathbf{W}_{ij}\right)\right) - 2\sum_i \mathbf{r}_i^\top \boldsymbol{\mu}_i \\
&\qquad\qquad - 2\sum_{ij} \text{trace}\left(\boldsymbol{\mu}_j \boldsymbol{\mu}_i^\top \mathbf{W}_{ij}\right)
\end{aligned} \tag{3}
$$

Note that in the last step, we have used the fact that $\mathbf{y}_i$ and $\mathbf{y}_j$ are independent under the distribution Q. From (3) we have,

$$
\mathbf{\Sigma}_i^* = \underset{\mathbf{\Sigma}_i}{\text{argmin}} \;\; \text{trace}\left(\mathbf{\Sigma}_i \left(I + \sum_j \mathbf{W}_{ij}\right)\right) - \log\left[|\mathbf{\Sigma}_i|\right] \tag{4}
$$

Note that (4) is a convex problem. Differentiating the cost function and setting the gradient to zero, we get $\mathbf{\Sigma}_i^* = \left(I + \sum_j \mathbf{W}_{ij}\right)^{-1}$.

From (3) we have,

$$\begin{aligned}\boldsymbol{\mu}_i^* &= \underset{\boldsymbol{\mu}_i}{\operatorname{argmin}} \ \operatorname{trace}\left(\boldsymbol{\mu}_i\boldsymbol{\mu}_i^\top\left(I+\sum_j \mathbf{W}_{ij}\right)\right) - 2\mathbf{r}_i^\top\boldsymbol{\mu}_i - 2\sum_j \operatorname{trace}\left(\boldsymbol{\mu}_j^*\boldsymbol{\mu}_i^\top\mathbf{W}_{ij}\right) \\ &= \underset{\boldsymbol{\mu}_i}{\operatorname{argmin}} \ \boldsymbol{\mu}_i^\top\left(I+\sum_j \mathbf{W}_{ij}\right)\boldsymbol{\mu}_i - 2\mathbf{r}_i^\top\boldsymbol{\mu}_i - 2\boldsymbol{\mu}_i^\top\left(\sum_j \mathbf{W}_{ij}\boldsymbol{\mu}_j^*\right)\end{aligned} \tag{5}$$

Note that (5) is a convex problem. Differentiating the cost function and setting the gradient to zero, we get

$$\boldsymbol{\mu}_i^* = \left(I+\sum_j \mathbf{W}_{ij}\right)^{-1}\left(\mathbf{r}_i+\sum_j \mathbf{w}_{ij}\boldsymbol{\mu}_j^*\right). \tag{6}$$

Hence, for the Gaussian distribution in (1), the mean field update for computing the means $\{\boldsymbol{\mu}_i\}$ is given by

$$\boldsymbol{\mu}_i \leftarrow \Big(I+\sum_j \mathbf{W}_{ij}\Big)^{-1}\Big(\mathbf{r}_i+\sum_j \mathbf{W}_{ij}\boldsymbol{\mu}_j\Big). \tag{7}$$

2. Backpropagation

Let L be the final loss function.

Backpropagating through the matrix generation layer:

Given the derivatives $dL/d\mathbf{W}_{ij}$ of the loss function with respect to the output of the matrix generation layer, we can compute the derivatives of L with respect to its input s_{ij} and parameters $\mathbf{C}$ using

$$\begin{aligned}\frac{dL}{ds_{ij}} &= \operatorname{trace}\left(\left(\frac{dL}{d\mathbf{W}_{ij}}\right)^\top\mathbf{C}\right), \\ \frac{dL}{d\mathbf{C}} &= \sum_{ij} s_{ij}\frac{dL}{d\mathbf{W}_{ij}}.\end{aligned} \tag{8}$$

Backpropagating through the similarity layer:

Given the derivatives dL/ds_{ij} of the loss function with respect to the output of the similarity layer, we can compute the derivatives of L with respect to its input $\mathbf{z}_i$ and parameters $\mathbf{f}_m$ using

$$\begin{aligned}\frac{dL}{d\mathbf{z}_i} &= 2\left(\sum_{m=1}^M \mathbf{f}_m\mathbf{f}_m^\top\right)\left(\sum_j s_{ij}\frac{dL}{ds_{ij}}\left(\mathbf{z}_j-\mathbf{z}_i\right)\right), \\ \frac{dL}{d\mathbf{f}_m} &= -2\left(\sum_{ij} s_{ij}\frac{dL}{ds_{ij}}\left(\mathbf{z}_i-\mathbf{z}_j\right)\left(\mathbf{z}_i-\mathbf{z}_j\right)^\top\right)\mathbf{f}_m.\end{aligned} \tag{9}$$

Backpropagating through the odd update layer:

Given the derivatives $dL/d\boldsymbol{\mu}_i^{out}$ of the loss function with respect to the output of an odd update layer, we can compute the derivatives of L with respect to its inputs $\mathbf{r}_i$, $\mathbf{W}_{ij}$ and $\boldsymbol{\mu}_j^{in}$ using

$$\frac{dL}{d\mathbf{r}_i} = \begin{cases} (\mathbf{I} + \sum_k \mathbf{W}_{ik})^{-1} \frac{dL}{d\boldsymbol{\mu}_i^{out}} & \text{if node } i \text{ is on an odd column} \\ 0 & \text{elsewise,} \end{cases}$$

$$\frac{dL}{d\mathbf{W}_{ij}} = \left(\mathbf{I} + \sum_k \mathbf{W}_{ik}\right)^{-1} \frac{dL}{d\boldsymbol{\mu}_i^{out}} \left(\boldsymbol{\mu}_j^{in} - \boldsymbol{\mu}_i^{out}\right)^\top, \text{ for } i \text{ in odd columns,} \tag{10}$$

$$\frac{dL}{d\boldsymbol{\mu}_j^{in}} = \begin{cases} \frac{dL}{d\boldsymbol{\mu}_j^{out}} + \sum_i \left(\mathbf{W}_{ij} (\mathbf{I} + \sum_k \mathbf{W}_{ik})^{-1} \frac{dL}{d\boldsymbol{\mu}_i^{out}}\right) & \text{if node } j \text{ is on an even column} \\ 0 & \text{elsewise.} \end{cases}$$

Backpropagating through the even update layer:

Given the derivatives $dL/d\boldsymbol{\mu}_i^{out}$ of the loss function with respect to the output of an even update layer, we can compute the derivatives of L with respect to its inputs $\mathbf{r}_i$, $\mathbf{W}_{ij}$ and $\boldsymbol{\mu}_j^{in}$ using

$$\frac{dL}{d\mathbf{r}_i} = \begin{cases} (\mathbf{I} + \sum_k \mathbf{W}_{ik})^{-1} \frac{dL}{d\boldsymbol{\mu}_i^{out}} & \text{if node } i \text{ is on an even column} \\ 0 & \text{elsewise,} \end{cases}$$

$$\frac{dL}{d\mathbf{W}_{ij}} = \left(\mathbf{I} + \sum_k \mathbf{W}_{ik}\right)^{-1} \frac{dL}{d\boldsymbol{\mu}_i^{out}} \left(\boldsymbol{\mu}_j^{in} - \boldsymbol{\mu}_i^{out}\right)^\top, \text{ for } i \text{ in even columns,} \tag{11}$$

$$\frac{dL}{d\boldsymbol{\mu}_j^{in}} = \begin{cases} \frac{dL}{d\boldsymbol{\mu}_j^{out}} + \sum_i \left(\mathbf{W}_{ij} (\mathbf{I} + \sum_k \mathbf{W}_{ik})^{-1} \frac{dL}{d\boldsymbol{\mu}_i^{out}}\right) & \text{if node } j \text{ is on an odd column} \\ 0 & \text{elsewise.} \end{cases}$$

3. Algorithmic description of the proposed Gaussian CRF network

Algorithm 1 Gaussian CRF Network

Input: Image $\mathbf{X}$

Unary Network

1: Apply the DeepLab CNN with parameters θ_u^{CNN} to image $\mathbf{X}$ to compute the unary predictions $\mathbf{r} = \{\mathbf{r}_i\}$.

$$\mathbf{r} = \text{DeepLabCNN}\left(\mathbf{X}, \theta_u^{CNN}\right).$$

Pairwise Network

2: Apply the pairwise network with paramters $\{\theta_p^{CNN}, \{\mathbf{f}_m\}, \mathbf{C} \succeq 0\}$ to image $\mathbf{X}$ to compute the pairwise matrices $\{\mathbf{W}_{ij}\}$ used in the energy function.

(a) DeepLab CNN (parameters θ_p^{CNN}): Compute per-pixel features $\mathbf{z} = \{\mathbf{z}_i\}$.

$$\mathbf{z} = \text{DeepLabCNN}\left(\mathbf{X}, \theta_p^{CNN}\right).$$

(b) Similarity layer (parameters $\{\mathbf{f}_m\}$): Compute the similarity measure $s_{ij} \in [0, 1]$ for every pair of connected pixels i and j using the features $\mathbf{z}$.

$$s_{ij} = e^{-\sum_{m=1}^{M}\left(\mathbf{f}_m^\top(\mathbf{z}_i - \mathbf{z}_j)\right)^2}.$$

(c) Matrix generation layer (parameters $\mathbf{C} \succeq 0$): Compute the matrix $\mathbf{W}_{ij} \succeq 0$ for every pair of connected pixels i and j using the similarity measure s_{ij}.

$$\mathbf{W}_{ij} = s_{ij}\mathbf{C}.$$

GMF Network

3: Initialize the GMF network input $\boldsymbol{\mu}^1 = \mathbf{r}$, and partition the nodes into even and odd columns $\boldsymbol{\mu} = \{\boldsymbol{\mu_e}, \boldsymbol{\mu_o}\}$.

4: *for* $t = 1$ *to* 5

(a) **Even update layer:** Update the even column nodes $\boldsymbol{\mu}_e^{t+1}$ using $\mathbf{r}$, $\{\mathbf{W}_{ij}\}$ and $\boldsymbol{\mu}_o^t$.

$$\boldsymbol{\mu}_i^{t+1} = \left(I + \sum_j \mathbf{W}_{ij}\right)^{-1}\left(\mathbf{r}_i + \sum_j \mathbf{W}_{ij}\boldsymbol{\mu}_j^t\right), \;\; \boldsymbol{\mu}_i \in \boldsymbol{\mu_e}, \;\; \boldsymbol{\mu}_j \in \boldsymbol{\mu_o}.$$

(b) **Odd update layer:** Update the odd column nodes $\boldsymbol{\mu}_o^{t+1}$ using $\mathbf{r}$, $\{\mathbf{W}_{ij}\}$ and $\boldsymbol{\mu}_e^{t+1}$.

$$\boldsymbol{\mu}_i^{t+1} = \left(I + \sum_j \mathbf{W}_{ij}\right)^{-1}\left(\mathbf{r}_i + \sum_j \mathbf{W}_{ij}\boldsymbol{\mu}_j^{t+1}\right), \;\; \boldsymbol{\mu}_i \in \boldsymbol{\mu_o}, \;\; \boldsymbol{\mu}_j \in \boldsymbol{\mu_e}.$$

5: Upsample $\boldsymbol{\mu}^6$ to the input image resolution using bilinear interpolation to obtain the class prediction scores at each pixel.

6: For each pixel, select the class label corresponding to the highest score.

Output: Class label at each pixel.