Weekly Exercise 10

Dr. Csaba Domokos
Technische Universität München, Computer Vision Group
July 17th, 2017

Parameter Learning (8 points)

Exercise 1 (**Prior distribution on** $\mathbf{w}$, 2 points)**.** Let $\mathcal{D} = \{(\mathbf{x}^1, \mathbf{y}^1), (\mathbf{x}^2, \mathbf{y}^2), \dots, (\mathbf{x}^N, \mathbf{y}^N)\}$ be a set of identically and independently distributed (*i.i.d.*) training samples. Assuming $\mathbf{w}$ is a random vector with prior distribution $p(\mathbf{w})$, show that the *posterior distribution* $p(\mathbf{w}|\mathcal{D})$ can be written as

$$p(\mathbf{w}|\mathcal{D}) = p(\mathbf{w}) \prod_{n=1}^{N} \frac{p(\mathbf{y}^n|\mathbf{x}^n, \mathbf{w})}{p(\mathbf{y}^n|\mathbf{x}^n)} \ .$$

Solution. Since we are given *i.i.d.* training samples, we get

$$\begin{aligned}
p(\mathbf{w} \mid \mathcal{D}) &= \frac{p(\mathbf{y}^1, \dots, \mathbf{y}^N, \mathbf{x}^1, \dots, \mathbf{x}^N, \mathbf{w})}{p(\mathbf{y}^1, \dots, \mathbf{y}^N, \mathbf{x}^1, \dots, \mathbf{x}^N)} \\
&= \frac{p(\mathbf{y}^1, \dots, \mathbf{y}^N \mid \mathbf{x}^1, \dots, \mathbf{x}^N, \mathbf{w}) p(\mathbf{x}^1, \dots, \mathbf{x}^N) p(\mathbf{w})}{p(\mathbf{y}^1, \dots, \mathbf{y}^N \mid \mathbf{x}^1, \dots, \mathbf{x}^N)\, p(\mathbf{x}^1, \dots, \mathbf{x}^N)} \\
&= \frac{p(\mathbf{y}^1, \dots, \mathbf{y}^N \mid \mathbf{x}^1, \dots, \mathbf{x}^N, \mathbf{w})\, p(\mathbf{w})}{p(\mathbf{y}^1, \dots, \mathbf{y}^N \mid \mathbf{x}^1, \dots, \mathbf{x}^N)} \\
&= p(\mathbf{w}) \prod_{n=1}^{N} \frac{p(\mathbf{y}^n|\mathbf{x}^n, \mathbf{w})}{p(\mathbf{y}^n|\mathbf{x}^n)} \ .
\end{aligned}$$

Exercise 2 (**Negative regularized conditional log-likelihood**, 6 points)**.** Consider the objective function $L(\mathbf{w})$ corresponding to the *negative regularized conditional log-likelihood*:

$$L(\mathbf{w}) = \lambda \|\mathbf{w}\|^2 + \sum_{n=1}^{N} \langle \mathbf{w}, \varphi(\mathbf{x}^n, \mathbf{y}^n) \rangle + \sum_{n=1}^{N} \log Z(\mathbf{x}^n, \mathbf{w}) \ .$$

It has been shown in the lecture that the gradient of $\mathcal{L}(\mathbf{w})$ w.r.t. $\mathbf{w}$ is given as

$$\nabla_{\mathbf{w}} L(\mathbf{w}) = 2\lambda \mathbf{w} + \sum_{n=1}^{N} \left(\varphi(\mathbf{x}^n, \mathbf{y}^n) - \mathbb{E}_{\mathbf{y} \sim p(\mathbf{y}|\mathbf{x}^n, \mathbf{w})}[\varphi(\mathbf{x}^n, \mathbf{y})] \right) \ .$$

Show that the Hessian of $L(\mathbf{w})$ is given as

$$\begin{aligned}
\Delta_{\mathbf{w}} L(\mathbf{w}) = 2\lambda \mathbf{I} + \sum_{n=1}^{N} \Big(& \mathbb{E}_{\mathbf{y} \sim p(\mathbf{y}|\mathbf{x}^n, \mathbf{w})}[\varphi(\mathbf{x}^n, \mathbf{y}) \varphi(\mathbf{x}^n, \mathbf{y})^\mathsf{T}] \\
& - \mathbb{E}_{\mathbf{y} \sim p(\mathbf{y}|\mathbf{x}^n, \mathbf{w})}[\varphi(\mathbf{x}^n, \mathbf{y})] \mathbb{E}_{\mathbf{y} \sim p(\mathbf{y}|\mathbf{x}^n, \mathbf{w})}[\varphi(\mathbf{x}^n, \mathbf{y})]^\mathsf{T} \Big) \ .
\end{aligned}$$

Solution. Let us denote the gradient vector by $g(\mathbf{w}) = \nabla_{\mathbf{w}} L(\mathbf{w})$. Notice that we have $g(\mathbf{w}), \mathbf{w}, \varphi(\mathbf{x}^n, \mathbf{y}^n) \in \mathbb{R}^d$. The Hessian matrix is calculated, by definition, as

$$\Delta_{\mathbf{w}} L(\mathbf{w}) = \begin{pmatrix} \frac{\partial g_1}{\partial w_1} & \frac{\partial g_2}{\partial w_1} & \cdots & \frac{\partial g_d}{\partial w_1} \\ \frac{\partial g_1}{\partial w_2} & \frac{\partial g_2}{\partial w_2} & \cdots & \frac{\partial g_d}{\partial w_2} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial g_1}{\partial w_n} & \frac{\partial g_2}{\partial w_n} & \cdots & \frac{\partial g_d}{\partial w_d} \end{pmatrix} .$$

We denote each element in the Hessian matrix by h_{ij}, where $h_{ij} = \frac{\partial g_i}{\partial w_j}$. Recall that the gradient is given as

$$g(\mathbf{w}) = 2\lambda \mathbf{w} + \sum_{n=1}^{N} \varphi(\mathbf{x}^n, \mathbf{y}^n) - \sum_{n=1}^{N} \sum_{\mathbf{y}^n \in \mathcal{Y}} \frac{\exp(-\langle \mathbf{w}, \varphi(\mathbf{x}^n, \mathbf{y}^n) \rangle)}{\sum_{\mathbf{y}' \in \mathcal{Y}} \exp(-\langle \mathbf{w}, \varphi(\mathbf{x}^n, \mathbf{y}') \rangle)} \varphi(\mathbf{x}^n, \mathbf{y}^n) .$$

For $h_{ij} = \frac{\partial g_i}{\partial w_j}$, we get

$$\begin{aligned}
\frac{\partial g_i}{\partial w_j} =& 2\lambda [\![i = j]\!] - \sum_{n=1}^{N} \sum_{\mathbf{y}^n \in \mathcal{Y}} \varphi_i(\mathbf{x}^n, \mathbf{y}^n) \frac{\partial}{\partial w_j} \frac{\exp(-\langle \mathbf{w}, \varphi(\mathbf{x}^n, \mathbf{y}^n) \rangle)}{\sum_{\mathbf{y}' \in \mathcal{Y}} \exp(-\langle \mathbf{w}, \varphi(\mathbf{x}^n, \mathbf{y}') \rangle)} \\
=& 2\lambda [\![i = j]\!] + \sum_{n=1}^{N} \sum_{\mathbf{y}^n \in \mathcal{Y}} \varphi_i(\mathbf{x}^n, \mathbf{y}^n) \frac{\exp(-\langle \mathbf{w}, \varphi(\mathbf{x}^n, \mathbf{y}^n) \rangle) \varphi_j(\mathbf{x}^n, \mathbf{y}^n)}{\sum_{\mathbf{y}' \in \mathcal{Y}} \exp(-\langle \mathbf{w}, \varphi(\mathbf{x}^n, \mathbf{y}') \rangle)} \\
& - \sum_{n=1}^{N} \sum_{\mathbf{y}^n \in \mathcal{Y}} \varphi_i(\mathbf{x}^n, \mathbf{y}^n) \frac{\exp(-\langle \mathbf{w}, \varphi(\mathbf{x}^n, \mathbf{y}^n) \rangle) \sum_{\mathbf{y}' \in \mathcal{Y}} \exp(-\langle \mathbf{w}, \varphi(\mathbf{x}^n, \mathbf{y}') \rangle) \varphi_j(\mathbf{x}^n, \mathbf{y}')}{\left(\sum_{\mathbf{y}' \in \mathcal{Y}} \exp(-\langle \mathbf{w}, \varphi(\mathbf{x}^n, \mathbf{y}') \rangle) \right)^2} \\
=& 2\lambda [\![i = j]\!] + \sum_{n=1}^{N} \sum_{\mathbf{y}^n \in \mathcal{Y}} \varphi_i(\mathbf{x}^n, \mathbf{y}^n) \varphi_j(\mathbf{x}^n, \mathbf{y}^n) p(\mathbf{y}^n \mid \mathbf{x}^n, \mathbf{w}) \\
& - \sum_{n=1}^{N} \left(\sum_{\mathbf{y}^n \in \mathcal{Y}} \varphi_i(\mathbf{x}^n, \mathbf{y}^n) p(\mathbf{y}^n \mid \mathbf{x}^n, \mathbf{w}) \right) \left(\sum_{\mathbf{y}' \in \mathcal{Y}} \varphi_j(\mathbf{x}^n, \mathbf{y}') p(\mathbf{y}' \mid \mathbf{x}^n, \mathbf{w}) \right) \\
=& 2\lambda [\![i = j]\!] + \sum_{n=1}^{N} \mathbb{E}_{\mathbf{y} \sim p(\mathbf{y} \mid \mathbf{x}^n, \mathbf{w})} [\varphi_i(\mathbf{x}^n, \mathbf{y}) \varphi_j(\mathbf{x}^n, \mathbf{y})] \\
& - \sum_{n=1}^{N} \mathbb{E}_{\mathbf{y} \sim p(\mathbf{y} \mid \mathbf{x}^n, \mathbf{w})} [\varphi_i(\mathbf{x}^n, \mathbf{y})] \mathbb{E}_{\mathbf{y} \sim p(\mathbf{y} \mid \mathbf{x}^n, \mathbf{w})} [\varphi_j(\mathbf{x}^n, \mathbf{y})] .
\end{aligned}$$

Putting all h_{ij} together back into matrix representation yields the following Hessian matrix:

$$\begin{aligned}
\Delta_{\mathbf{w}} L(\mathbf{w}) = 2\lambda \mathbf{I} + \sum_{n=1}^{N} \Big(& \mathbb{E}_{\mathbf{y} \sim p(\mathbf{y} \mid \mathbf{x}^n, \mathbf{w})} [\varphi(\mathbf{x}^n, \mathbf{y}) \varphi(\mathbf{x}^n, \mathbf{y})^\top] \\
& - \mathbb{E}_{\mathbf{y} \sim p(\mathbf{y} \mid \mathbf{x}^n, \mathbf{w})} [\varphi(\mathbf{x}^n, \mathbf{y})] \mathbb{E}_{\mathbf{y} \sim p(\mathbf{y} \mid \mathbf{x}^n, \mathbf{w})} [\varphi(\mathbf{x}^n, \mathbf{y})]^\top \Big) .
\end{aligned}$$

Programming (6 points)

Exercise 3 (**Gibbs sampling**, 6 points)**.** Let us consider the problem of *binary image segmentation* and solve it by performing *probabilistic inference* via **Gibbs sampling**. In this particular exercise, we are going to design a *cow-detector* for the test images in Figure 1, which should label a pixel as foreground if it belongs to a *cow*, and background otherwise.

Figure 1: The test images for binary image segmentation to detect cows.

We define the following *energy function* for $\mathbf{y} \in \{0,1\}^{\mathcal{V}}$ such that 0 and 1 denote the background and the foreground, respectively:

$$E(\mathbf{y}) = \sum_{i \in \mathcal{V}} E_i(y_i) + w \sum_{(i,j) \in \mathcal{E}} E_{ij}(y_i, y_j) \ ,$$

where $w \in \mathbb{R}^+$ is a parameter, and $\mathcal{V}$ stands for the set of pixels, and $\mathcal{E}$ includes all pairs of 4-neighboring pixels.

To define the **unary energy functions** E_i, use the provided `*.yml` files. Each test image has its own data file, specified by the same filename. In each data file, you can read out a $H \times W$ array of float numbers. The H and W are the image height and width, and each float value p_i corresponds to the probability of that the given pixel belongs to the foreground. We provide the `cow_detector.cpp` to demonstrate how to load a data file and read out the corresponding probability values. The unary energy functions E_i for all $i \in \mathcal{V}$ are then defined as the *negative log-likelihood*:

$$E_i(y_i) = \begin{cases} -\log(1-p_i) & \text{if } y_i = 0 \\ -\log(p_i) & \text{if } y_i = 1 \ . \end{cases}$$

The **pairwise energy functions** are defined as the *contrast-sensitive Potts model* for all $(i,j) \in \mathcal{E}$,

$$E_{ij}(y_i, y_j; x_i, x_j) = \exp(-\lambda \|x_i - x_j\|^2) \cdot [\![y_i \neq y_j]\!] \ .$$

where x_i denotes the intensities of the pixel i and $\lambda = 0.5$.

Implement the *Gibbs sampling algorithm* to achieve *probabilistic inference* and calculate the *binary segmentation* as well. Choose different values for w and give the range of w that generates the best segmentation performance.