Weekly Exercise 12

Dr. Csaba Domokos and Lingni Ma
Technische Universität München, Computer Vision Group
July 05, 2016 (submission deadline: July 12, 2016)

Parameter Learning (5 Points)

Exercise 1 (**loss minimizing parameter learning**, 2 Points). Calculate the expected loss $\mathbb{E}_{(\mathbf{x},\mathbf{y})\sim d(\mathbf{y}|\mathbf{x})}\left[\Delta_H(\mathbf{y}, f(\mathbf{x}))\right]$ of the Hamming loss:

$$\Delta_H(\mathbf{y}, \mathbf{y}') = \frac{1}{|\mathcal{V}|}\sum_{i\in\mathcal{V}} [\![\mathbf{y}_i \neq \mathbf{y}'_i]\!],$$

where $d(\mathbf{y}|\mathbf{x})$ denotes the true data distribution and $f : \mathcal{X} \to \mathcal{Y}$ is a prediction function.

Solution.

$$\begin{aligned}
\mathbb{E}_{\mathbf{y}\sim d(\mathbf{y}|\mathbf{x})}\left[\Delta_H(\mathbf{y}, f(\mathbf{x}))\right] &= \sum_{\mathbf{y}\in\mathcal{Y}} d(\mathbf{y}|\mathbf{x})\Delta_H(\mathbf{y}, f(\mathbf{x})) \approx \sum_{\mathbf{y}\in\mathcal{Y}} p(\mathbf{y}|\mathbf{x},\mathbf{w})\Delta_H(\mathbf{y}, f(\mathbf{x})) \\
&= \sum_{\mathbf{y}\in\mathcal{Y}} p(\mathbf{y}|\mathbf{x},\mathbf{w})\frac{1}{|\mathcal{V}|}\sum_{i\in\mathcal{V}}[\![\mathbf{y}_i \neq f(\mathbf{x}_i)]\!] \\
&= \sum_{\mathbf{y}\in\mathcal{Y}} p(\mathbf{y}|\mathbf{x},\mathbf{w})\frac{1}{|\mathcal{V}|}\left(\sum_{i\in\mathcal{V}} 1 - [\![\mathbf{y}_i = f(\mathbf{x}_i)]\!]\right) \\
&= \sum_{\mathbf{y}\in\mathcal{Y}} p(\mathbf{y}|\mathbf{x},\mathbf{w}) - \sum_{\mathbf{y}\in\mathcal{Y}} p(\mathbf{y}|\mathbf{x},\mathbf{w})\frac{1}{|\mathcal{V}|}\left(\sum_{i\in\mathcal{V}}[\![\mathbf{y}_i = f(\mathbf{x}_i)]\!]\right) \\
&= 1 - \frac{1}{|\mathcal{V}|}\sum_{i\in\mathcal{V}} p\big(\mathbf{y}_i = f(\mathbf{x}_i)|\mathbf{x},\mathbf{w}\big).
\end{aligned}$$

Exercise 2 (**parameter learning** 3 Points). Compute the *sub-differential* at a point $\mathbf{x} \in \mathbb{R}^n$

$$\partial f(\mathbf{x}) = \{\mathbf{w} \in \mathbb{R}^n \mid f(\mathbf{x}) + \langle \mathbf{w}, \mathbf{y} - \mathbf{x}\rangle \leq f(\mathbf{y}), \forall \mathbf{y} \in \mathbb{R}^n\},$$

of the following convex functions $f : \mathbb{R}^n \to \mathbb{R}$:

a) $f(\mathbf{x}) = \langle \mathbf{c}, \mathbf{x}\rangle$, where $\mathbf{c} \in \mathbb{R}^n$ is a constant

b) $f(\mathbf{x}) = \|\mathbf{x}\|_2 = \sqrt{\sum_{i=1}^n x_i^2}$

c) $f(\mathbf{x}) = \|\mathbf{x}\|_1 = \sum_{i=1}^n |x_i|$

Solution.

a) By definition of $\partial f(\mathbf{x})$:

$$\begin{aligned}&\langle c, \mathbf{x}\rangle + \langle \mathbf{w}, \mathbf{y}-\mathbf{x}\rangle \le \langle \mathbf{c}, \mathbf{y}\rangle, \quad \forall \mathbf{y} \in \mathbb{R}^n\\ &\Leftrightarrow \langle \mathbf{c}-\mathbf{w}, \mathbf{x}-\mathbf{y}\rangle \le 0, \quad \forall \mathbf{y} \in \mathbb{R}^n\\ &\Leftrightarrow \mathbf{w} = \mathbf{c}.\end{aligned}$$

Hence $\partial f(\mathbf{x}) = \{\mathbf{c}\}$. Note that if $f(\mathbf{x})$ is differentiable at $\mathbf{x} \in \mathbb{R}^n$ we have that $\partial f(\mathbf{x}) = \{\nabla f(\mathbf{x})\}$.

b) If $\mathbf{x} = \mathbf{0}$:

$$\begin{aligned}&\langle \mathbf{w}, \mathbf{y}\rangle - \|\mathbf{y}\|_2 \le 0, \quad \forall \mathbf{y} \in \mathbb{R}^n\\ &\langle \mathbf{w}, \mathbf{y}\rangle - \|\mathbf{y}\|_2 \le \|\mathbf{w}\|_2 \|\mathbf{y}\|_2 - \|\mathbf{y}\|_2 = (\|\mathbf{w}\|_2 - 1)\|\mathbf{y}\|_2 \le 0.\end{aligned}$$

Hence $\partial f(\mathbf{0}) = \{\mathbf{w} \in \mathbb{R}^n \mid \|\mathbf{w}\|_2 \le 1\}$.
For $\mathbf{x} \neq 0$, $f(\mathbf{x})$ is differentiable, hence:

$$\partial f(\mathbf{x}) = \left\{\frac{\mathbf{x}}{\|\mathbf{x}\|_2}\right\}.$$

c) Before we calculate the subgradient of the ℓ_1-norm, lets first notice that the subgradient of the maximum of convex subdifferetiable functions, *i.e.*,

$$f(x) = \max_{i=1,2,\dots,n} f_i(x)\ ,$$

is the convex hull of all $f_i(x)$ that the maximum at x. For example, consider the absolution function $f(x) = |x|$. This function can be written as $f(x) = \max(x, -x)$. For the later $\max$ representation, at $x = 0$, both $f(x) = x$, and $f(x) = -x$ are maximum. The subgraident at $x = 0$ is the convex hull of subgradient of the two functions, hence $[-1, 1]$. Now we can construct the ℓ_1-norm as a maximum of 2^n linear functions,

$$\|x\|_1 = \max\left\{\mathbf{s}^\mathsf{T}\mathbf{x} \mid s_i \in \{-1, 1\}\right\}.$$

The function $\mathbf{s}^\mathsf{T}\mathbf{x}$ is differentiable and has a unique subgradient, where

$$g_i = \begin{cases}1 & x_i > 0\\ -1 & x_i < 0\\ 1 \text{ or } -1 & x_i = 0\end{cases}$$

The subgradient of ℓ_1-norm is the convex hull of all the subgradients, therefore,

$$\partial f(\mathbf{x}) = \left\{\mathbf{g} \mid \|\mathbf{g}\|_\infty \le 1\ , \mathbf{g}^\mathsf{T}\mathbf{x} = \|\mathbf{x}\|_1\right\}$$

Programming (12 Points)

Exercise 3 (**Probabilistic and loss minimizing parameter learning**, 12 Points)**.** In `Exercise 11`, we have considered the problem of binary image segmentation and have solved it by performing probabilistic inference via Gibbs sampling. In particular, we have developed a cow detector for the images in Figure 1.For this sake, we have defined 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. In `Exercise 11`, we chose the parameter w arbitrarily. In this exercise, we are going to learn the optimal w by applying both *probabilistic parameter learning* and *loss minimizing parameter learning*.

Again, the unary energy functions E_i is provided in `*.yml` files. Each 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,

$$E_i(y_i = 0) = -\log(1 - p_i)$$
$$E_i(y_i = 1) = -\log(p_i) \ .$$

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

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

where $\lambda = 0.5$.

To learn the parameters, we provide 42 images (under sub-folder `rgb-training`). You should use all of them in training. Once you learned the parameters, you can use the learned w to test on images in Figure 1 (see sub-folder `rgb-test`). Do not use any images from the test set during training.

Use your previous implementations to get MAP inference by applying `graph cuts` (see `Exercise 6`) and to get probabilistic inference by applying `Gibbs sampling` algorithm (see `Exercise 11`). Implement **both** *probabilistic parameter learning* and *loss minimizing parameter learning* to estimate the optimal value for parameter w. Compare the optimal w you have learned.

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