

## Weekly Exercise 12

Dr. Csaba Domokos  
Technische Universität München, Computer Vision Group  
July 31st, 2017

### Parameter Learning

(5 Points)

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

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

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

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

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

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

- a)  $f(x) = \langle c, x \rangle$ , where  $c \in \mathbb{R}^n$  is a constant
- b)  $f(x) = \|x\|_2 = \sqrt{\sum_{i=1}^n x_i^2}$
- c)  $f(x) = \|x\|_1 = \sum_{i=1}^n |x_i|$

### 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 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,

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

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

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

where  $\lambda = 0.5$ .

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.