Exercise 1: Majorization Minimization (Due 25/4/2017)*

Statistical Methods in Image Processing 048926

Analytic Exercises (30 points)

1. Let $B \in \mathbb{R}^{N\times N}$ be a symmetric matrix and let $x \in \mathbb{R}^N$ be a vector. Show that the following identity holds
$$x^T Bx = \sum_i a_i x_i^2 + \frac{1}{2}\sum_i \sum_j b_{ij}|x_i - x_j|^2$$
and express $\{a_i\}$ and $\{b_{ij}\}$ as functions of the entries of B.

2. Let $Q(x;x')$ be a surrogate function for minimizing $f(x)$. Show that the following properties hold.

   (a) *Gradient property:* If $f(x)$ and $Q(x;x')$ are continuously differentiable w.r.t. x then $\nabla_x f(x)|_{x=x'} = \nabla_x Q(x;x')|_{x=x'}$. Hint: use the relation between gradient and directional derivative.

   (b) *Composition:* $Q(g(x);g(x'))$ is a surrogate for minimizing $f(g(x))$ for any function g.

3. Let $\{f_k(x)\}$ be a sequence of functions and assume that $Q_k(x;x')$ is a surrogate function for minimizing $f_k(x)$ for every k. Show that $\sum_k Q_k(x;x')$ is a surrogate for minimizing $\sum_k f_k(x)$.

4. Find a_1 and a_2 such that the following function $\tilde{\rho}(\Delta,\Delta')$ is a surrogate function for minimizing $\rho(\Delta)$.
$$\tilde{\rho}(\Delta;\Delta') = a_1\Delta + \frac{a_2}{2}(\Delta - \Delta')^2. \tag{1}$$
Are there situations where (1) cannot be a surrogate function with any a_1, a_2?

5. We saw in class that for a symmetric quadratic function $\tilde{\rho}(\Delta;\Delta') = \frac{1}{2}a\Delta^2$ to be a surrogate function, we need to take $a = \frac{\rho'(\Delta')}{\Delta'}$. This is valid, of course, only for $\Delta' \neq 0$. For $\Delta' = 0$, we can use the limiting value $a = \rho''(\Delta')$ (assuming that $\rho'(0) = 0$). This leads to the following candidate surrogate function
$$\tilde{\rho}(\Delta;\Delta') = \begin{cases} \frac{\rho'(\Delta')}{2\Delta'}\Delta^2 & \Delta' \neq 0 \\ \\ \frac{\rho''(\Delta')}{2}\Delta^2 & \Delta' = 0 \end{cases} \tag{2}$$
Prove that if the potential function $\rho(\Delta)$ is such that $\rho'(\Delta)$ is concave for $\Delta > 0$ and convex for $\Delta < 0$, then (2) is a surrogate function for $\rho(\Delta)$.

*Please send your solutions to Tamar

Matlab Exercises (70 points)

1. Given an image x drawn from the prior model

$$p(x) = \frac{1}{Z} \exp \left\{ - \sum_{\{i,j\} \in C} \rho \left(\frac{x_i - x_j}{\sigma_x} \right) \right\},$$

we would like to estimate the parameter σ_x by using the Maximum Likelihood (ML) criterion

$$\hat{\sigma}_x = \arg \max_{\sigma_x} p(x)$$

(a) For the log-cosh potential $\rho(\Delta) = \log \cosh(\Delta)$, write a function that accepts an image x as input, and performs an exhaustive search to find $\hat{\sigma}_x$.

(b) For the Generalized Gaussian (GG) potential $\rho(\Delta) = \frac{|\Delta|^p}{p}$, write a function that accepts an image x and an order p as inputs, and uses the closed-form solution we saw in class to compute $\hat{\sigma}_x$.

2. **Majorization Minimization**

Given a degraded version y of an image x, such that $y = Hx + n$, the image x can be estimated by solving the MAP problem

$$\arg \min_x \frac{1}{2\sigma_n^2} \|y - Hx\|^2 + \sum_{\{i,j\} \in C} \rho(x_i - x_j).$$

Note that the scaling parameter σ_x was absorbed into ρ. Using the symmetric bound surrogate function (2), the majorization minimization algorithm works by solving a sequence of optimization problems of the form

$$\arg \min_x \frac{1}{2\sigma_n^2} \|y - Hx\|^2 + \sum_{\{i,j\} \in C} \tilde{b}_{ij} (x_i - x_j)^2,$$

where $\tilde{b}_{ij} = \frac{\rho'(x'_i - x'_j)}{2(x'_i - x'_j)}$ and x' is the image from the previous iteration. As we saw in class, this problem can be written using matrix notations as

$$\arg \min_x \frac{1}{2\sigma_n^2} \|y - Hx\|^2 + \|WDx\|^2. \tag{3}$$

(a) Implement a function that constructs the weight matrix W according to the GG potential $\rho(\Delta) = \frac{|\Delta|^p}{\sigma_x^p p}$. The function should accept the image x', the scaling parameter σ_x and the potential's order p.

(b) Implement a function that constructs the weight matrix W according to the log-cosh potential $\rho(\Delta) = \log \cosh \left(\frac{\Delta}{\sigma_x} \right)$. The function should accept the image x' and the scaling parameter σ_x.

(c) Write the gradient of (3) w.r.t x and implement a function that calculates the x at which the gradient vanishes. This can be done using the Matlab function `x = bicg(A,b)` which solves a linear system $Ax = b$. This function has two operation modes:

   i. Matrix Input. In this option the function accepts A as a matrix. Therefore, you are required to construct the matrices D (you may use the Matlab function `toeplitz`) and H (given the blur kernel h).

   ii. Function Handle (**recommended**). In this option A is given as a function handle `@Afun` that operates on an image x. You can implement the multiplications by the convolution matrices D, D^T, H, H^T in the Fourier domain (what does transposition of a convolution matrix do in the Frequency domain?). To avoid boundary effects, pad the image with repetition of its boundaries. When calling `@Afun`, it is required to specify the unknown parameter x: `x = bicg(@(x,tflag) Afun(x,W,tflag),b)`. Use Matlab help for more details. In this option **there is no need to explicitly construct any large matrix**.

(d) Implement the full MM scheme (5-10 iterations should be enough). In each iteration, update W according to x' (at the first iteration, initialize x' to be y) and then update x according to (c).

(e) Show the denoising and debluring results of the MM optimization scheme using the TV potential ($p = 1$), the GG potential with $p = \frac{2}{3}$, and the log-cosh potential. Use the two images in the supplementary mat file. Compare the results in terms of PSNR

$$\mathrm{PSNR}(x, y) = 10 \log_{10} \left(\frac{255^2}{\sum_i (x_i - y_i)^2} \right).$$

Use the estimation for $\hat{\sigma}_x$ from the clean images using the function from (1).

   i. Denoising: show the results for each of the two images, with noise variance of $\sigma = 15, 25, 35$.

   ii. Debluring: show the results for each of the two images, with each of the three different blur kernels in the mat file, and with a noise variance of $\sigma_n = 2.5$. Estimate $\hat{\sigma}_x$ from the clean images using the function from (1).

Running times may be long, so you can use small images for debugging (but submit the final results with the supplied images).