Álgebra Lineal Avanzada

Optimización imagen

Supongamos que $X_\mathrm{c} \in \mathbb{R}^{n\times m}$ contiene información, contaminada por ruido, sobre una imagen. Deseamos obtener una imagen suave, $X \in \mathbb{R}^{n\times m}$, minimizando las funciones de costo

$$f_1(X) = \sum_j^m \sum_i^{n-1} (x_{(i+1),j} - x_{i,j})^2$$

y

$$f_2(X) = \sum_i^n \sum_j^{m-1} (x_{i,(j+1)} - x_{i,j})^2 \ ,$$

las cuales cuantifican la diferencia total entre píxeles adyacentes verticales y horizontales, respectivamente. Al mismo tiempo, se desea que X se mantenga cerca de X_c, lo cual expresamos con la función de costo

$$f_3(X) = \sum_i^n \sum_j^m (x_{i,j} - x_{\mathrm{c}_{i,j}})^2 \ .$$

Las funciones de costo pueden escribirse de manera compacta como

$$f_1(X) = \mathrm{trace}(X^\top D_1^\top D_1 X) \ , \quad f_2(X) = \mathrm{trace}(D_2^\top X^\top X D_2)$$

y

$$f_3(X) = \mathrm{trace}\Big((X - X_\mathrm{c})^\top (X - X_\mathrm{c})\Big) \ ,$$

donde

$$D_1 := \begin{pmatrix} -1 & 1 & 0 & \cdots & 0 & 0 & 0 \\ 0 & -1 & 1 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & -1 & 1 & 0 \\ 0 & 0 & 0 & \cdots & 0 & -1 & 1 \end{pmatrix}$$

y

$$D_2 := \begin{pmatrix} -1 & 0 & \cdots & 0 & 0 \\ 1 & -1 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & -1 & 0 \\ 0 & 0 & \cdots & 1 & -1 \\ 0 & 0 & \cdots & 0 & 1 \end{pmatrix} .$$

Una versión suave de X_c puede obtenerse resolviendo el problema de optimización

$$\min_{X \in \mathbb{R}^{n\times m}} f(X) \ , \quad f(X) := \delta(f_1(X) + f_2(X)) + f_3(X) \ , \quad \delta > 0 \ .$$

La función f es cuadrática y convexa en X, por lo que le valor óptimo puede obtenerse haciendo la primera derivada de f igual a cero. Esto nos lleva a la ecuación de Lyapunov

$$\left(I + \delta D_1^\top D_1\right) X + X\delta(D_2 D_2^\top) = X_\mathrm{c} \ . \tag{1}$$

Descargue el archivo *stones_c.jpg* y cargue la imagen en la matriz X_c. Suavice X_c resolviendo la ecuación (1). Las funciones de Matlab *imread*, *imshow* y *lyap* pueden ser útiles.