Linear Algebra for Electrical Engineers

Homework VI

Due 11/03 23:59

Please checkout the eTL homework announcement for the submission format. Skeleton code is available at this link: https://github.com/3dvision-snu/linear-algebra-2020-fall

1. The 2-vectors $\mathbf{p}_1, \cdots, \mathbf{p}_N$ represent the locations or positions of N objects, for example, factories, warehouses, and stores. The last K of these locations are fixed and given; the goal in a *placement problem* is to choose the locations of the first $N-K$ objects. Our choice of the locations is guided by an undirected graph; an edge between two objects means we would like them to be close to each other. In *least squares placement*, we choose the locations $\mathbf{p}_1, \cdots, \mathbf{p}_{N-K}$ so as to minimize the sum of the squares of the distances between objects connected by an edge,

$$\|\mathbf{p}_{i_1} - \mathbf{p}_{j_1}\|^2 + \cdots + \|\mathbf{p}_{i_L} - \mathbf{p}_{j_L}\|^2,$$

where the L edges of the graph are given by $(i_1, j_1), \cdots, (i_L, j_L)$.

   (a) Let $\mathcal{D}$ be the Dirichlet energy of the graph, as defined on page 135 of the textbook. Show that the sum of squared distance between the N objects can be expressed as $\mathcal{D}(\mathbf{u}) + \mathcal{D}(\mathbf{v})$, where $\mathbf{u} = ((\mathbf{p}_1)_1, \cdots, (\mathbf{p}_N)_1)$ and $\mathbf{v} = ((\mathbf{p}_1)_2), \cdots, (\mathbf{p}_N)_2)$ are N-vectors containing the first and second coordinates of the objects, repectively. (0.1 points)

   (b) Express the least squares placement problem as a least squares problem, with variable $\mathbf{x} = (\mathbf{u}_{1:(N-K)}, \mathbf{v}_{1:(N-K)})$. In other words, express the objective above (the sum of squares of the distances across edges) as $\|A\mathbf{x} - \mathbf{b}\|$, for and appropriate $m \times n$ matrix A and m-vector $\mathbf{b}$. You will find that $m = 2L$. *Hint.* Recall that $\mathcal{D}(\mathbf{y}) = \|B^\top \mathbf{y}\|^2$, where B is the incidence matrix of the graph. (0.1 points)

   (c) Solve the least squares placement problem with numpy for the specific problem with $N = 10, K = 4, L = 13$, fixed locations

$$\mathbf{p}_7 = (0, 0), \mathbf{p}_8 = (0, 1), \mathbf{p}_9 = (1, 1), \mathbf{p}_{10} = (1, 0)$$

   and edges (1, 3), (1, 4), (1, 7), (2, 3), (2, 5), (2, 8), (2, 9), (3, 4), (3, 5), (4, 6), (5, 6), (6, 9), (6, 10). Write the final results to the report. (0.5 points)

2. Suppose that A has linearly independent columns, with $\hat{\mathbf{x}} = A^{\dagger}\mathbf{b}$ minimizing $\|A\mathbf{x}-\mathbf{b}\|^2$. In this exercise, you are to implement an iterative method, named *Richardson iteration* to compute least squares solution $\hat{\mathbf{x}}$. We randomly initialize $\mathbf{x}^{(0)}$ and for $k = 0, 1, \cdots,$ and iterate

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - \frac{1}{\|A\|} A^{\top}(A\mathbf{x}^{(k)} - \mathbf{b}),$$

This iteration is proven to converge to least square solution $\hat{\mathbf{x}}$. Generate a random 20×10 matrix A and 20-vector $\mathbf{b}$ and compute $\hat{\mathbf{x}} = A^{\dagger}\mathbf{b}$. Run the algorithm for 300 iterations and plot $\|\mathbf{x}^{(k)} - \hat{\mathbf{x}}\|$ to verify the convergence. You may use `np.linalg.norm` to compute the norm of matrices and vectors. Attach the plot to the report. (0.3 points)