Simultaneous Graph Learning and Matrix Recovery

Seyyid Emre Sofuoglu, Selin Aviyente

March 7, 2022

1 Methods

Our aim is to learn the underlying manifold of a data on a Cartesian Product Graph and recovery missing or grossly corrupted entries. The objective can be written in convex terms as:

$$\begin{aligned} &\text{minimize}_{G,S,\Phi_1,\Phi_2}\text{tr}(G^\top \Phi_1 G) + \text{tr}(G\Phi_2 G^\top) + \|S\|_1, \\ &\text{s.t.}\mathcal{P}_\Omega[Y] = \mathcal{P}_\Omega[G+S],\ \Phi_1,\Phi_2 \in \mathcal{S}, \end{aligned} \tag{1}$$

where Φ_1, Φ_2 are Graph Laplacians for the row graph and column graph, respectively, and $\mathcal{S}$ is the space of undirected graph Laplacians. When the graph constraints are explicitely written, (1) becomes:

$$\begin{aligned} &\text{minimize}_{G,S,\Phi_1,\Phi_2}\text{tr}(G^\top \Phi_1 G) + \text{tr}(G\Phi_2 G^\top) + \|S\|_1 + \|\Phi_1\|_F^2 + \|\Phi_2\|_F^2, \\ &\text{s.t.}\mathcal{P}_\Omega[Y] = \mathcal{P}_\Omega[G+S],\ \Phi_i = \Phi_i^\top,\ \Phi_i\mathbf{1} = 0,\ \Phi_i \succeq 0,\ \text{tr}(\Phi_i) = 2I_i \end{aligned} \tag{2}$$

1.1 Optimization

For (2) to be optimized, the graph variable G can be separated into two variables to avoid large inverses.