A Algorithm Description of ADMM

We use Alternating Direction Methods of Multipliers (ADMM) to solve convex optimization problem in RPCA and TensorRPCA , and the details are shown in Algorithm1 and Algorithm2 which are named Matrix ADMM and Tensor ADMM, respectively.

Algorithm 1: Matrix ADMM

Input: matrix data X, parameter λ.
Initialization: $H_0 = S_0 = Y_0 = 0$, $\rho = 1.1, \mu_0 = 1e-3, \epsilon = 1e-8$.
while not converged **do**

% Singular Value Thresholding %;
1. Update H_{k+1} by $H_{k+1} = \arg\min_{H} \|H\|_* + \frac{\mu_k}{2}\|H + S_k - X + \frac{Y_k}{\mu_k}\|_F^2$;

% Soft-Thresholding %;
2. Update S_{k+1} by $S_{k+1} = \arg\min_{S} \lambda\|S\|_1 + \frac{\mu_k}{2}\|H_{k+1} + S - X + \frac{Y_k}{\mu_k}\|_F^2$;
3. $Y_{k+1} = H_k + \mu_k(H_{k+1} - S_{k+1} - X)$;
4. Update μ_{k+1} by $\mu_{k+1} = \rho\mu_k$;
5. Check the convergence conditions
$\|H_{k+1} - H_k\|_\infty \leq \epsilon, \|S_{k+1} - S_k\|_\infty \leq \epsilon, \|H_{k+1} - S_{k+1} - X\|_\infty \leq \epsilon$;

end

Algorithm 2: Tensor ADMM

Input: tensor data $\mathcal{X}$, parameter λ.
Initialization: $\mathcal{H}_0 = \mathcal{S}_0 = \mathcal{Y}_0 = 0$, $\rho = 1.1, \mu_0 = 1e-2, \epsilon = 1e-5$.
while not converged **do**

% Singular Value Thresholding %;
1. Update $\mathcal{H}_{k+1}$ by $\mathcal{H}_{k+1} = \arg\min_{\mathcal{H}} \|\mathcal{H}\|_* + \frac{\mu_k}{2}\|\mathcal{H} + \mathcal{S}_k - \mathcal{X} + \frac{\mathcal{Y}_k}{\mu_k}\|_F^2$;

% Soft-Thresholding %;
2. Update $\mathcal{S}_{k+1}$ by $\mathcal{S}_{k+1} = \arg\min_{\mathcal{S}} \lambda\|\mathcal{S}\|_1 + \frac{\mu_k}{2}\|\mathcal{H}_{k+1} + \mathcal{S} - \mathcal{X} + \frac{\mathcal{Y}_k}{\mu_k}\|_F^2$;
3. $\mathcal{Y}_{k+1} = \mathcal{H}_k + \mu_k(\mathcal{H}_{k+1} - \mathcal{S}_{k+1} - \mathcal{X})$;
4. Update μ_{k+1} by $\mu_{k+1} = \min(\rho\mu_k, \mu_{\max})$;
5. Check the convergence conditions
$\|\mathcal{H}_{k+1} - \mathcal{H}_k\|_\infty \leq \epsilon, \|\mathcal{S}_{k+1} - \mathcal{S}_k\|_\infty \leq \epsilon, \|\mathcal{H}_{k+1} - \mathcal{S}_{k+1} - \mathcal{X}\|_\infty \leq \epsilon$;

end

B Select Similar Ego-networks

In section 3.3, we enhance LRD-GNN-Matrix to tensor version LRD-GNN-Tensor. The ego-networks around nodes which are similar with target node are selected to construct tensor. We evaluate similarity from two perspectives. (1) We select nodes which have similar attributes with target node. For instance, the Cosine Similarity is used to measure the similarity of attributes between nodes. (2) We select nodes which have similar local structures with target node. We use the Shannon entropy value of ego-network to measure local structure similarity. For the ego-network $\mathcal{G}_i = (\mathcal{V}_i, \mathcal{E}_i)$ around v_i, the Shannon entropy value $H(\mathcal{G}_i)$ is defined as

$$H(\mathcal{G}_i) = -\sum_{v\in\mathcal{V}_i} P(v)\log P(v) \tag{1}$$

where $P(v)$ is the probability of the random walking visiting v in ego-network. Then, we choose several nodes' ego-networks that are close to the Shannon entropy value of the central node to construct tensor.

Besides, to combine information from different orders, we also utilize the selected nodes' ego-networks after propagation(i.e., AX, A^2X) as part of the tensor. The specific methods for constructing tenosr and the performance have been presented in section 4.1.3 Ablation Study.