Supplementary file 1

A subspace learning aided matrix factorization for drug repurposing

Amir Mahdi Zhalefar, Zahra Narimani*

Department of Computer Science and Information Technology, Institute for Advanced Studies in Basic Sciences (IASBS), Zanjan 45137-66731, Iran

Detailed Pseudocode Implementation of the Proposed Method

Input:

Data matrices $X^{(h)} \in \mathbb{R}^{m^{(h)} \times n^{(h)}}$ for $h=1,2$; neighbor sizes $K^{(h)}$; balance parameters $\alpha^{(h)}, \beta^{(h)}, \lambda^{(h)}$ for $h=1,2$; maximum iterations N_{Iter}; Gaussian scale parameters $\sigma^{(h)}$ for $h=1,2$; number of selected features $l^{(h)}$ for $h=1,2$.

Output:

Selected feature indices for $h=1,2$; new data matrices $X_{new}^{(h)} \in \mathbb{R}^{l^{(h)} \times n^{(h)}}$ for $h=1,2$.

Main():

1. Construct KNN graphs $G_0^{(h)}$ and $G_1^{(h)}$ for $h=1,2$ using **ConstructKNNGraphs(** $X^{(h)}, K^{(h)}$ **)**;
2. Compute similarity matrices and graph Laplacians using **ComputeSimilarityMatrices(** $X^{(h)}, G_0^{(h)}$ **)** for $h=1,2$;
3. Update variables until convergence using **UpdateVariables(** $X^{(h)}, W_v^{(h)}, W_s^{(h)}, L_V^{(h)}, L_S^{(h)}, \alpha^{(h)}, \beta^{(h)}, \lambda^{(h)}, \sigma^{(h)}, N_{Iter}$ **)** for $h=1,2$;
4. Select features using **EvaluateFeatures(**$X^{(h)}, S^{(h)}, l^{(h)}$**)** for $h = 1,2$;

Function ConstructKNNGraphs($X^{(h)}, K^{(h)}$**):**

1. **for** $i \leftarrow 1$ **to** $m^{(h)}$ **do**
   - Find $K^{(h)}$ nearest neighbors for each point in $X^{(h)}$;
   - Create undirected graph $G_0^{(h)} = (V_0^{(h)}, E_0^{(h)})$;
2. **return** $G_0^{(h)}$;

Function ComputeSimilarityMatrices($X^{(h)}, G_0^{(h)}$**):**

1. Compute similarity matrices $W_v^{(h)}$ and $W_s^{(h)}$;

2. Compute graph Laplacian matrices $L_V^{(h)}$ and $L_S^{(h)}$;
3. **return** $W_v^{(h)}, W_s^{(h)}, L_V^{(h)}, L_S^{(h)}$;

Function UpdateVariables($X^{(h)}, W_v^{(h)}, W_s^{(h)}, L_V^{(h)}, L_S^{(h)}, \alpha^{(h)}, \beta^{(h)}, \lambda^{(h)}, \sigma^{(h)}, N_{Iter}$**):**

1. Initialize $U^{(h)}, S^{(h)}, V^{(h)}$;
2. **for** $t \leftarrow 1$ **to** N_{Iter} **do**
    - Update $U^{(h)}, S^{(h)}, V^{(h)}$ using current values;
    - **for** $i \leftarrow 1$ **to** $m^{(h)}$ **do**
        - **for** $j \leftarrow 1$ **to** $n^{(h)}$ **do**
            - ♣ $\text{term } A' \leftarrow [X^{(h)} U^{(h)} X^{(h)\top} V^{(h)\top} + \alpha^{(h)} X^{(h)} W_S^{(h)} X^{(h)\top} + (\beta^{(h)} + \lambda^{(h)}) I_m^{(h)}]_{ij}$
            - ♣ $\text{term } B' \leftarrow [X^{(h)} U^{(h)} X^{(h)\top} S^{(h)} V^{(h)} V^{(h)\top} + \alpha^{(h)} X^{(h)} D_S^{(h)} X^{(h)\top} + \beta^{(h)} 1_{m\times m} + \lambda^{(h)} S^{(h)} S^{(h)\top}] S_{ij}^{(h)}$
            - ♣ $S_{ij}^{(h)} \leftarrow S_{ij}^{(h)} \times \dfrac{\text{term } A'}{\text{term } B'}$;
    - **for** $i \leftarrow 1$ **to** $m^{(h)}$ **do**
        - **for** $j \leftarrow 1$ **to** $n^{(h)}$ **do**
            - ♣ $\text{term } A'' \leftarrow [S^{(h)\top} X^{(h)} U^{(h)} X^{(h)\top} + \alpha^{(h)} V^{(h)\top} w^{(h)} V^{(h)}]_{ij}$
            - ♣ $\text{term } B'' \leftarrow [S^{(h)\top} X^{(h)} U^{(h)} X^{(h)\top} S^{(h)} V^{(h)\top} + \alpha^{(h)} V^{(h)\top} D^{(h)} V^{(h)}]_{ij}$
            - ♣ $V_{ij}^{(h)} \leftarrow V_{ij}^{(h)} \times \dfrac{\text{term } A''}{\text{term } B''}$;
3. **return** $U^{(h)}, S^{(h)}, V^{(h)}$;

Function EvaluateFeatures($X^{(h)}$, $S^{(h)}$, $\boldsymbol{l}^{(h)}$**):**

1. Calculate evaluation values for each feature based on $\|\boldsymbol{S}_i^{(h)}\|_2$;
2. Sort evaluation values in descending order;
3. Select top $\boldsymbol{l}^{(h)}$ features;
4. **return** selected feature indices and new data matrix $\mathbf{X}_{\mathbf{new}}^{(h)}$;