$notebook_name$

$number_of_pages$

new_page

$notebook_created$

$notebook_modified$

Adaptive Graph Embedding

① smooth filter

② adaptive encoder

① Laplacian smoothing Filter

Rayleigh quotient: $R(L,x) = \frac{x^T L x}{x^T x} = \frac{\sum_{(i,j)\in\mathcal{E}}(x_i - x_j)^2}{\sum_{i\in V} x_i^2}$ (1)

对L进行特征值分解 $L = U\Lambda U^{-1}$，其中$U \in R^{n\times n}$包含特征向量

于是 $R(L,u_i) = \frac{u_i^T L u_i}{u_i^T u_i} = \lambda_i$ (2)

令 $x = Up = \sum_{i=1}^{n} p_i u_i$?，p_i为特征向量u_i的系数

$R(L,x) = \frac{x^T L x}{x^T x} = \frac{\sum_{i=1}^{n} p_i^2 \lambda_i}{\sum_{i=1}^{n} p_i^2}$ (4)

去除高频成分.

Generalized Laplacian Smoothing Filter

$H = I - kL$ (5)

$\tilde{x} = Hx = U(I - k\Lambda)U^{-1}Up = \sum_{i=1}^{n}(1-k\lambda_i)p_i u_i = \sum_{i=1}^{n} p_i' u_i$ (6)

因此，频率响应函数 $1-k\lambda$ 应为非负递减函数从得到低通滤波器.

$\tilde{X} = H^t X$ (7). 将t个filter stacking up

The choice of k

renormalization trick

$$I_N + D^{-1/2} A D^{-\frac{1}{2}} \longrightarrow \tilde{D}^{-\frac{1}{2}} \hat{A} \tilde{D}^{-\frac{1}{2}}$$

where $\tilde{A} = I_N + A$ 使特征值由 $[0,2] \rightarrow [-1,1]$

同时避免数值不稳定和梯度爆炸.

$\tilde{D}_{ij} = \sum_j \tilde{A}_{ij}$

$$\tilde{L}_{sym} = \tilde{D}^{-\frac{1}{2}} \tilde{L} \tilde{D}^{-\frac{1}{2}} \quad (8)$$

$$H = I - k\tilde{L}_{sym} \quad (9).$$ (若 $k=1$, 则为GCN的 filter)

$$R(L, \tilde{x}) = \frac{\tilde{x}^T L \tilde{x}}{\tilde{x}^T \tilde{x}} = \frac{\sum_{i=1}^{n} p_i'^2 \lambda_i}{\sum_{i=1}^{n} p_i'^2} \quad (10).$$

$p_i' = (1-k\lambda_i)p_i$

$p_i'^2$ 随着 λ_i 的↑而↓ k 的最佳取值为 $\frac{1}{\lambda_{max}}$

Adaptive Graph Embedding

① smooth filter

② adaptive encoder

Adaptive Encoder

$$Z = f(\tilde{X}; W) = \tilde{X}W \quad (11)$$

使用 pairwise node similarity 方法, high similarity记为正类, low记为负样本.

Similarity matrix $$S = \frac{ZZ^T}{\|Z\|_2^2} \quad (12)$$

Training sample Selection

令 r_{ij} 表示 the rank of node pair (v_i, v_j)

$$l_{ij} = \begin{cases} 1 & , r_{ij} \le r_{pos} \\ 0 & , r_{ij} > r_{neg} \\ \text{none} & , \text{others} \end{cases} \quad (13)$$

(generated label of node pair (v_i, v_j))

初始S值.

$$S = \frac{\tilde{X}\tilde{X}^T}{\|\tilde{X}\|_2^2} \quad (14)$$

Loss

$$\mathcal{L} = \sum_{(v_i, v_j) \in O} -l_{ij}\log(S_{ij}) - (1 - l_{ij})\log(1 - S_{ij}) \quad (15)$$

Thresholds Update

$$r'_{pos} = r_{pos} + \boxed{\frac{r_{pos}^{ed} - r_{pos}^{st}}{T}}\ (一) \quad (16)$$

$$r'_{neg} = r_{neg} + \boxed{\frac{r_{neg}^{ed} - r_{neg}^{st}}{T}}\ (十) \quad (17)$$

应用

对于 node clustering 问题，使用 spectral clustering 方法；

使用 Davies-Bouldin index (DBI) 衡量聚类效果.

Similarity matrix

$$S = \frac{ZZ^T}{\|Z\|^2} \quad (12).$$

r_{pos} r_{neg}

代码理解

Update_similarity ( ):

$f\_adj = \frac{ZZ^T}{\|Z\|^2}$

cosin = f_adj. reshape([-1,]) (size = 2333264)

pos_num = round (r_{pos}^{st} (超参数) × len(cosin))

neg_num = round((1 − r_{neg}^{st}) × len(cosin))

pos_inds = np.argparition(−cosin, pos_num)[: pos_num]

rank the similarity sequence in the descending order

Sim = (zx * zy). sum(1)

Sim = self. act(sim) [self.act = torch. sigmoid]

return sim.

torch. sum(1) 对行求和.

Adaptive Graph Embedding.

① Laplacian smooth.

② Encoder MLP. Deep autoencoder.

similarity 每10代更新一次.

loss

Embedding

(nodes x hidden_units)

nodes x nodes

adj

Spectual Clustering [sklearn.cluster.SpectralClustering]