Data mining package report

Kandati Vishnu Sai, Midhilesh E., Sanjay Seetharaman

November 2020

1 Introduction

Hi! This is the section for introduction. Create your own tex file and include the corresponding reference in main.tex.

2 Graph Mining

Hi! This is a new section.

3 DeepInf: Social influence prediction

3.1 Introduction

Deepinf focus on the prediction of user-level social influence. It aims to **predict the action status of a user given the action**. It is a deep learning based framework to represent both influence dynamics and network structures into a latent space and tries to minimize the negative likelihood that was defined in the section 1.

3.2 Model framework

3.2.1 Sampling Near Neighbour

Give a user v, a r-ego network $\mathcal{G}_v^r$ is extracted using breadth-first search (BFS) starting from user v. However, $\mathcal{G}_v^r$ may have different size due to the small world property in the social network. Since most deep learning models expects fixed size data, the graph $\mathcal{G}_v^r$ can be sampled to fixed size.

For sampling a fixed size graph **random walk the restart** (RWR) was used. RWR algorithm is defined as following steps.

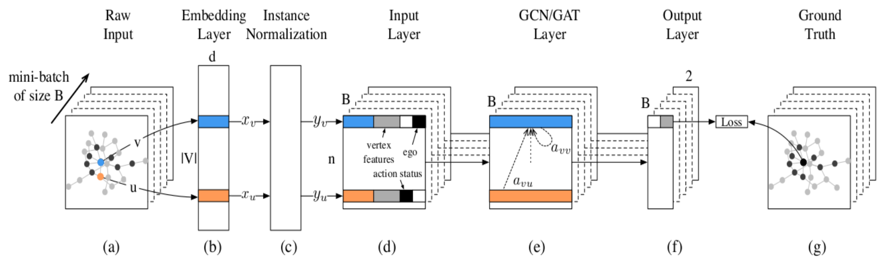
- Start random walks from either the ego user v or one of her active neighbors randomly.
- The random walk iteratively travels to its neighborhood with the probability that is proportional to the weight of each edge.
- At each step, the walk is assigned a probability to return to the starting node, that is, either the ego user v or one of v's active neighbors.
- Run the algorithm until a fixed number of vertices denoted by $\hat{\Gamma}_v^r$ with $|\hat{\Gamma}_v^r|$=n.

After running this algorithm, a sub-graph $\hat{\mathcal{G}}_v^r$ and denote $\hat{S}_v^t$= $\{ s_u^t : u \in \hat{\Gamma}_v^r \}$ be the action statuses of v's sampled neighbours. Therefore we re-define the optimization objective in section 1 as:

$$\mathcal{L}(\theta) = -\sum_{i=1}^{N} \log(P_\theta(s_{v_i}^{t_i+\Delta t} | \hat{\mathcal{G}}_{v_i}^t, s_{v_i}^{t_i})) \quad (1)$$

3.2.2 Neural Network

With the retreived $\hat{\mathcal{G}}_{v_i}^t$ and $\hat{S}_v^t$ for each user, an effective neural network model to incorporate both the structural properties in $\hat{\mathcal{G}}_{v_i}^t$ and action statuses in $\hat{S}_v^t$.



Deepinf neural network model consist of network embedding layer, instance normalization layer, input layer and GCN layer as described in the above diagram.

3.2.3 Embedded layer

Network embedding technique encode network structural properties into low dimensional matrix $\mathbf{X} \in \mathcal{R}^{\mathcal{D} \times |V|}$. Deepinf uses Deepwalk algorithm for mapping each users into $\mathcal{R}^{\mathcal{D}}$ space.

3.2.4 Instance Normalization

Instance normalization can remove instance-specific mean and variance, which encourages the downstream model to focus on users' relative positions in latent embedding space rather than their absolute positions. It also prevents overfitting.

Let x_u be the low dimension representation for the user $u \in \hat{\Gamma_v^r}$, the instance normalized vector y_u is obtained by

$$y_{ud} = \frac{x_{ud} - \mu_{ud}}{\sqrt{\sigma_d^2 + \epsilon}} \tag{2}$$

for each embedding dimension d = 1...$\mathcal{D}$, where

$$\mu_d = \frac{1}{n} \sum_{u \in \hat{\Gamma_v^r}} x_{ud}, \;\; \sigma_d^2 = \frac{1}{n} \sum_{u \in \hat{\Gamma_v^r}} (x_{ud} - \mu_{ud})^2 \tag{3}$$

3.2.5 Input Layer

3.2.6 GCN

3.3 Evaluation Metrics