Anomaly Detection in Social Networks

Course Project of CS 591 C1 Compressive Sensing

Zuoqi Zhang, Yuefeng Li, Siyuan Tang

Social Networks

- Over 1.5 billion Facebook users worldwide; over 300 million pictures uploaded to Facebook daily
- Over 215 million Twitter users; over 500 million Tweets sent daily
- Over 200 million LinkedIn members in over 200 countries

Graph

Undirected graph $G = (V, E)$

- Vertices (V) → users
- Edges (E) → social connections, e.g. friendship

Community

Groups of nodes that have

- Dense connections within each group (community)
- Less connections crossing communities

Graph Embedding

Graph embedding is better than using adjacency matrix because it is more compressive.

Graph embedding is trying to find a mapping from a graph to a lower-dimensional vector space, while preserving relevant graph properties.

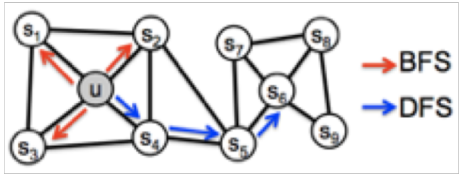
Given a graph $G(V, E)$ and a field F, an exact dot product representation (DPR) X of G, of dimension d ($d \ll |V|$) over F, is a mapping X:

$V \to F_d$ such that for $i \neq j, X(vi) \cdot X(v_j) = 1$ if $(vi, v_j) \in E$, and $X(vi) \cdot X(v_j) = 0$ otherwise.

It is a minimization problem with constraints:

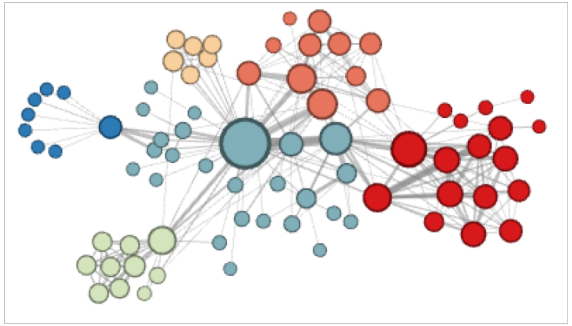
$$\min_{\{\boldsymbol{x}_1,\ldots,\boldsymbol{x}_N\}} \sum_{ij\in\mathbb{E}} (a_{ij} - < \boldsymbol{x}_i, \boldsymbol{x}_j >)^2 + Sparsity\{< \boldsymbol{x}_i, \boldsymbol{x}_j >\}$$

node2vec



The *node2vec* framework learns low-dimensional representations for nodes in a graph by optimizing a neighborhood preserving objective.

The objective is flexible, and the algorithm accommodates for various definitions of network neighborhoods by simulating biased random walks.

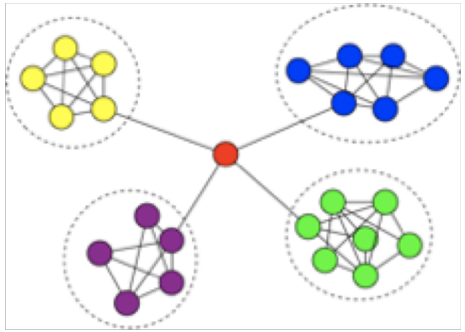


For example, the graph visualization above depicts the color-coded communities exhibiting homophily discovered by *node2vec* in the Les Misérables Network.

Membership Vector

Given a graph $G(V, E)$ which has k communities, for each node v_i, the membership vector of v_i is defined as $\boldsymbol{y_i} = \left(y_i^{(1)}, y_i^{(2)}, \dots, y_i^{(k)}\right)$, where $0 \leq y_i^{(j)} \leq 1$ and $\sum_{j=1}^{k} y_i^{(j)} = 1, 1 \leq j \leq k$.

$y_i^{(j)}$ indicates the probability that v_i belongs to the community c_j (degree of membership).



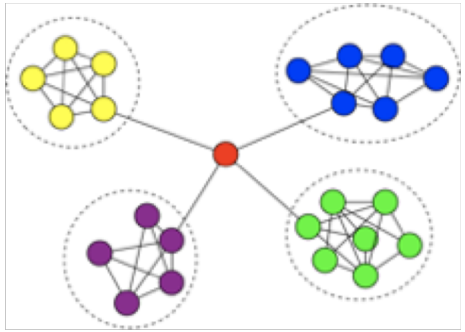
$k = 4$

$$\begin{cases} (1, 0, 0, 0) \\ (0, 1, 0, 0) \\ (0, 0, 1, 0) \\ (0, 0, 0, 1) \end{cases}$$

$(0.25, 0.25, 0.25, 0.25)$

What is an Anomaly?

Some node may have edges with multiple communities that are not correlated, which makes it more difficult to do community analysis in a network.



Definition

Given a node v_i and its membership vector $\boldsymbol{y_i}$, calculate the anomaly score of this node by the following formula:

$$AScore(v_i) = \frac{Avg(\boldsymbol{y_i})}{Max(\boldsymbol{y_i})} \cdot Cnt\left(y_i^{(j)} \geq Avg(\boldsymbol{y_i})\right)$$

If $AScore(v_i) \geq thre$, then v_i is labeled as an anomaly.

$$k = 4 \qquad \begin{cases} (1,0,0,0) \\ (0,1,0,0) \\ (0,0,1,0) \\ (0,0,0,1) \end{cases} \qquad \Longrightarrow AScore = \frac{1}{1} \cdot 1 = 1$$

$$(0.25, 0.25, 0.25, 0.25) \qquad \Longrightarrow AScore = \frac{0.25}{0.25} \cdot 4 = 4$$

Objective Function

$l_2 - norm$

$$min\ O = \sum_{i=1}^{n}\sum_{j=1}^{k} y_i^{(j)} \cdot \left|\left|\boldsymbol{x_i} - \boldsymbol{m_j}\right|\right|_2^2 + \alpha \sum_{i=1}^{n} \left|\left|\boldsymbol{y_i}\right|\right|_2^2 \quad s.t. 0 \le y_i^{(j)} \le 1, \sum_{j=1}^{k} y_i^{(j)} = 1$$

$\boldsymbol{y_i} = (1, 0, 0, \dots, 0)$

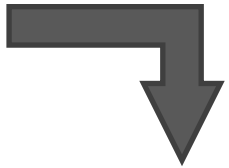
$\boldsymbol{y_i} = \left(\frac{1}{k}, \frac{1}{k}, \frac{1}{k}, \dots, \frac{1}{k}\right)$

- $\boldsymbol{x_i} \rightarrow graph\ embedding\ vector\ of\ v_i$
- $\boldsymbol{y_i} \rightarrow membership\ vector\ of\ v_i$
- $\boldsymbol{m_j} \rightarrow center\ of\ community\ c_j$
- $\alpha \rightarrow trade - off$

Membership Vector Learning

$$min\ O = \sum_{i=1}^{n}\sum_{j=1}^{k} y_i^{(j)} \cdot \left|\left|\boldsymbol{x_i} - \boldsymbol{m_j}\right|\right|_2^2 + \alpha \sum_{i=1}^{n} \left|\left|\boldsymbol{y_i}\right|\right|_2^2 \qquad s.t. 0 \le y_i^{(j)} \le 1, \sum_{j=1}^{k} y_i^{(j)} = 1$$

Set partial derivatives to 0



Algorithm Probability Learning

Input: feature vectors $X = \{\boldsymbol{x_1}, \boldsymbol{x_2}, \dots, \boldsymbol{x_n}\}$

number of clusters k

parameter α

Output: $y_i^{(j)}, 1 \le i \le n, 1 \le j \le k$ (possibility of node i belonging to cluster j)

1: **Initialization:** $y_i^{(j)}$, $\boldsymbol{m_j}$ (by K-Means Algorithm)

2: t = 0

3: **Repeat**

4: **Update** $\boldsymbol{m_j}$ (by equation (4.12))

5: **Update** $y_i^{(j)}$ (by equation (4.18))

6: t = t + 1

7: **Until** convergence or $t > t_{max}$

$$\boldsymbol{m_j} = \frac{\sum_{i=1}^{n} y_i^{(j)} \cdot \boldsymbol{x_i}}{\sum_{i=1}^{n} y_i^{(j)}}$$

$$y_i^{(j)} = \frac{2\alpha + \sum_{j=1}^{k} \left|\left|\boldsymbol{x_i} - \boldsymbol{m_j}\right|\right|_2^2 - k\left|\left|\boldsymbol{x_i} - \boldsymbol{m_j}\right|\right|_2^2}{2k\alpha}$$

Anomaly Removal

1. Calculate $AScore$ for each node using its final membership vector
2. Remove nodes who have an $AScore$ higher than the threshold we set
3. Remove the associated edges

Experiment

- Dataset of Facebook from SNAP (Stanford Network Analysis Project)
- Undirected graph
- 36,692 nodes
- 183,831 edges

$$Q = \sum_{c \in C} \left(\frac{I_c}{E} - \left(\frac{2I_c + O_c}{2E} \right)^2 \right)$$

- **Modularity** of a network
- More edges within a community
- Less edges crossing two communities

Future Work

- Use directed graph instead of un-directed graph to model networks like Twitter, Instagram, etc.
- Improve the accuracy of anomaly detection metrics

Thanks!

Contact us:

Zuoqi Zhang (zqzhang@bu.edu)

Yuefeng Li (ferrisyl@bu.edu)

Siyuan Tang (sytang7@bu.edu)