# COMP 9102 Assignment 3

Wang Cheng

## 1 Implementation

### 1.1 Building the matrix

Since we treat retweet data as an undirected graph, the matrix should be symmetric.

```
for row in data:
  matrix[row[0] - 1, row[1] - 1] = 1
  matrix[row[1] - 1, row[0] - 1] = 1
```

### 1.2 Personal page-rank

PPR-based proximity vector for node u is defined as follows: $p_u = (1-\alpha)Ap_u + \alpha e_u$. By setting $p_u^{(0)} = \mathbf{0}$, $p_u$ is computed iteratively.

### 1.3 Evaluation of clustering

We evaluated the quality of clustering using the label file by following criteria:

**entropy** For each cluster, the class distribution of the data is calculated first, i.e., for cluster $j$ we compute $p_{ij}$, the 'probability' that a member of cluster $j$ belongs to class $i$ as follows: $p_{ij} = m_{ij}/m_j$, where $m_j$ is the number of values in cluster $j$ and $m_{ij}$ is the number of values of class $i$ in cluster $j$. Then using this class distribution, the **entropy of each cluster** $j$ is calculated using the standard formula $e_j = \sum_{i=1}^{L} p_{ij} \log_2 p_{ij}$, where $L$ is the number of classes. The **total entropy for a set of clusters** is calculated as the sum of the entropies of each cluster weighted by the size of each cluster, i.e., $e = \sum_{i=1}^{K} \frac{m_i}{m} e_j$, where $m_j$ is the size of cluster $j$, $K$ is the number of clusters, and $m$ is the total number of data points.

**purity** Using the terminology derived for entropy, the purity of cluster j, is given by $purity_j = \max p_{ij}$ and the overall purity of a clustering by $purity = \sum_{i=1}^{K} \frac{m_i}{m}$

**NMI** Normalized Mutual Information $NMI(Y,C) = \frac{I(Y;C)}{[H(Y)+H(C)]/2}$ is calculated by the following steps (All logs are base-2):

1. Calculate entropy of class labels

2. Calculate entropy of cluster labels
3. Calculate mutual information by $I(Y, C) = H(Y) - H(Y|C)$, where $H(Y|C)$ is the conditional entropy of class labels for clustering
4. Calculate NMI

## 2 Experiment

To run the code: `python assignment3.py -k 5`. Here, `-k` is the number of clusters to form.