Community Detection in Social Network with Pairwisely Constrained Symmetric Non-Negative Matrix Factorization

Amish Garg
Department of Computer Science
Indian Institute of Technology, Roorkee
Roorkee, India
agarg@cs.iitr.ac.in

Shubhang Tripathi
Department of Computer Science
Indian Institute of Technology, Roorkee
Roorkee, India
stripathi1@cs.iitr.ac.in

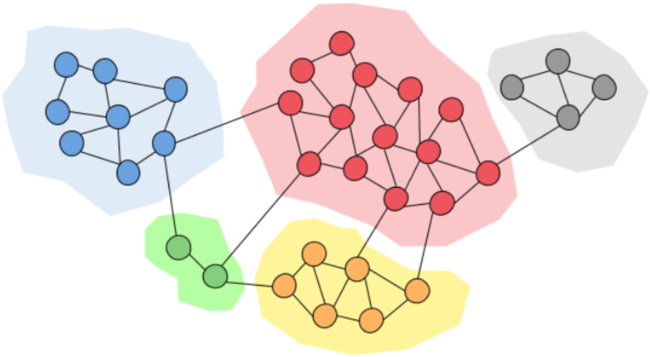

Fig. 1. Community Detection

I. Introduction

We desire to detect the most inter-related community in community detection studies of a social network or collaborative network. There are a couple of standard method for community detection in social network analysis. These methods are optimizing an objective function, statistical inference, network partitioning, detection of overlapping sub-networks.

We study two types of community detection, unsupervised and semi-supervised. Research suggests that to achieve better results it is a good idea to combine some amount labeled data with the unlabeled data. For the supervised approach we have to select data.

In this paper, we study non-negative objects. Some examples of non-negative objects are image (where the image is represented as a hexadecimal value denoting the color of a pixel). The text vectors can also be represented as word vectors. To represent a network we can use a adjacency matrix **A**. A new method called Non-negative matrix factorization was introduced by [2] reduce the dimension. We use NMF to increase productivity by introducing non-negative matrix constraints.

NMF falls under the category of unsupervised methods. It can be represented as $X{\approx}UV$ with $U \geq 0$ and $V \geq 0$.

In this report, we have discuss a method called pairwisely constrained nonnegative symmetric matrix factorization (PC-SNMF). The speciality of this algotihm this that it avoids adding the community label information into objective function directy, rather it iteratively factorizes while keeping pairwise constraints intact.

II. Theory

A. Non-Negative Matrix Factorisation

The goal of NMF is to factorize X into UV, with an approximation that:

$$X \approx UV^T \tag{1}$$

where,

$$X = [x_1, x_2, ..., x_n] \in R^{mxn} \tag{2}$$

where x_j denotes j-th data point and is an m-dimensional vector.

We use the given objective function, and the goal is to minimize.

$$J = ||X - UV^T||^2 \tag{3}$$

B. Community-detection task

Adjacency matrix A is used to represent a network. The value corresponding to two nodes in the adjacency matrix is 1 if there is a connection, else the value corresponding to the nodes is 0. We denote the count of communities in the network using the variable k. U has dimensions $n \times k$ and each column is regarded as the center of one community. In order to identify the community of j-th node we take,

$$c_j = arg \max_c v_{jc} \tag{4}$$

C. Pairwise Constraints guided Non-negative Matrix Factorization

This algorithm was proposed by [3]. It is abbreviated as PCNMF. Objective function is defined as:

$$J = ||X - UV^T||^2 + \lambda tr(V^T LV) \tag{5}$$

where $L = D - A$ is laplacion graph.
D is defined as $d_{ii} = \sum_{j=1}^{n} A_{ij}$.

$$A_{ij} = \begin{cases} \alpha, & \text{if } x_i, x_j \text{ have the same class label} \\ -(1-\alpha), & \text{if } x_i, x_j \text{ have the different class label} \\ 0, & \text{otherwise} \end{cases}$$

D. Constrained Non-negative Matrix Factorization

This algorithm was proposed by [4]. It is abbreviated as CNMF. Objective function is defined as:

$$J = ||X - UZA||^2 \tag{6}$$

A is an auxiliary matrix.

$$\overline{A_{ij}} = \begin{cases} \alpha, & \text{if } x_i, x_j \text{ have the same class label} \\ 0, & \text{if } x_i, x_j \text{ have the different class label} \\ A_{ij} + I_{ij}, & \text{otherwise} \end{cases}$$

where, α is a positive constant, and I is the Identity matrix.

III. Pairwisely Constrained Symmetric NMF model

Given an undirected and unweighed network **G** consisting of n nodes a_1, a_2, ... , a_n, the network is represented as a matrix $X_{n\times n}$ derived from the adjacency matrix, but with all diagonal elements set to **1** instead of **0**.

$$x_{ij} = \begin{cases} 1, & \text{if } a_i, a_j \text{ have a connection or i = j} \\ 0, & \text{otherwise} \end{cases}$$

Since this is a semi-supervised learning algorithm, the labels for s nodes are available and for the rest $r = n - s$ nodes is not available. This data is used to generate the pairwise constraints. If two labelled nodes have the same label, then a must-link constraint is generated for them, otherwise a cannot-link constraint is generated.

These constraints are constructed in the form of a **must-link pairwisely constrainted symmetric matrix** $M_{n\times n}$ and a **cannotlink pairwisely constrainted symmetric matrix** $C_{n\times n}$ respectively.

$$m_{pj} = \begin{cases} 1, & \text{if } a_p, a_j (p \neq j) \text{ have same label} \\ 0, & \text{otherwise} \end{cases}$$

$$c_{pj} = \begin{cases} 1, & \text{if } a_p, a_j (p \neq j) \text{ have different label} \\ 0, & \text{otherwise} \end{cases}$$

With these constraints, the minimization function of the **PCSNMF** model comes out to be:

$$\begin{aligned} J = &\sum_{i=1}^{n}\sum_{j=1}^{n}(x_{ij} - \sum_{c=1}^{k} s_{ic}s_{jc})^2 \\ &+ \alpha\sum_{j=1}^{n}(\sum_{p:m_{pj}=1} \sum_{c=1}^{k} \sum_{h=1,h\neq c}^{k} s_{jc}s_{ph} + \sum_{p:c_{pj}=1} \sum_{c=1}^{k} s_{jc}s_{pc}) \end{aligned} \tag{7}$$

Here s_{ij} comes from the matrix $S_{n\times k}$, which is the factorized matrix, needed for community detection. The above equation can be written in matrix form using an auxiliary matrix $B_{k\times k}$, which satisfies:

$$b_{ij} = \begin{cases} 0, & \text{if i = j} \\ 1, & \text{otherwise} \end{cases}$$

The matrix equation then is:

$$J = \parallel X - SS^T \parallel^2 + \alpha[tr(S^T MSB) + tr(S^T CS)] \tag{8}$$

In this equation, the first term ($\parallel X - SS^T \parallel^2$), corresponds to the cost function for the non-negative matrix factorization. The second term consists of two parts: the first part corresponds to the penalty for violating the must-link conditions, and the second part corresponds to the penalty for violating the cannot-link conditions.

Using the fact that $\parallel A \parallel^2$:

$$\begin{aligned} J &= tr((X - SS^T)^T(X - SS^T)) \\ &+ \alpha[tr(S^T MSB) + tr(S^T CS)] \\ &= tr((X^T - (SS^T)^T)(X - SS^T)) \\ &+ \alpha[tr(S^T MSB) + tr(S^T CS)] \end{aligned} \tag{9}$$

Using $(AB)^T = B^T A^T$ and $(A^T)^T = A$:

$$\begin{aligned} J &= tr((X^T - SS^T)(X - SS^T)) \\ &+ \alpha[tr(S^T MSB) + tr(S^T CS)] \\ &= tr(X^T X - X^T SS^T - SS^T X + SS^T SS^T \\ &+ \alpha[tr(S^T MSB) + tr(S^T CS)] \\ &= tr(X^T X) - tr(X^T SS^T) - tr(SS^T X) + tr(SS^T SS^T) \\ &+ \alpha[tr(S^T MSB) + tr(S^T CS)] \end{aligned} \tag{10}$$

Now, using $tr(A) = tr(A^T)$:

$$\begin{aligned} J &= tr(X^T X) - 2tr(X^T SS^T) + tr(SS^T SS^T) \\ &+ \alpha[tr(S^T MSB) + tr(S^T CS)] \end{aligned} \tag{11}$$

In order to minimize this cost function, under the constraints that $s_{ij} \geqslant 0$, we use the concept of Lagrange multipliers, with ϕ_{ij} being the lagrange multiplier for the inequality $s_{ij} \geqslant 0$ and $\phi_{n\times k}$ being the matrix [ϕ_{ij}]. Now, the lagrange function is:

$$L = J + tr(\phi S^T) \tag{12}$$

To minimize, the derivative of L, with respect to S must be 0:

$$\frac{\partial L}{\partial S} = -4X^T S + 4SS^T S + 2\alpha(MSB + CS) + \phi = 0 \tag{13}$$

Now, using the multiplicative update rules for **NMF** [5] and the KKT conditions ($\phi_{ic} s_{ic}$ = 0):

$$s_{ic} \leftarrow s_{ic} \otimes \frac{2(X^T S)_{ic}}{2(SS^T S)_{ic} + \alpha(MSB + CS)_{ic}} \tag{14}$$

This equation is the basis for the iterative calculation of the factorized matrix **S**

IV. Algorithm

In the PCSNMF algorithm, we first create the required $\mathbf{X}_{n \times n}$ matrix which is derived from the adjacency matrix, but with all diagonal entries set to 1. Then we randomly sample some nodes whose labels will be considered for the creation of the **must-link** and **cannot-link** conditions. This is shown with the help of the **sample** function in the algorithm shown. Once this is done, we create these conditions in the form of the $\mathbf{M}_{n \times n}$ and $\mathbf{C}_{n \times n}$ matrices, which are as described previously. Finally, we create the $\mathbf{B}_{k \times k}$ matrix as well.

Once these matrices have been computed, we begin computation of the factorized matrix $\mathbf{S}_{n \times k}$. It's entries are initialized to random values and then it is updated iteratively using the multiplicative update equation described previously.

Once we have the factorized matrix, we can extract the community labels for each node x_j by taking the index corresponding to highest value in the row S_j, i.e.

$$label(x_j) = arg \max_c S_{jc} \tag{15}$$

V. Results

We write the code for PCSNMF and run experiments over it using different datasets. We have discussed the results of our finding the this section.

A. Data sets

We use the basic dataset Karate Club for primary testing and finding basic results. The dataset contains $N = 34$ nodes and $k = 2$ communities.

Algorithm 1 The PCSNMF Algorithm

```
Require: G: the input Graph
Require: k: The expected number of communities
Require: s: The number of labelled nodes
  # generate adjacency matrix from the input graph
  $X \leftarrow get\_adjacency\_matrix(G)$

  # Make the diagonal entries 1 as needed
  $n \leftarrow length(G)$
  for i = 1 to $n$ do
    $X_{ii} \leftarrow 1$
  end for

  # Randomly sample s nodes from the Graph
  $labelled\_nodes \leftarrow sample(nodes(G), s)$

  # create the M and C matrices
  $M \leftarrow [0]_{n \times n}$
  $C \leftarrow [0]_{n \times n}$
  for $node_i, node_j \in labelled\_nodes, i \neq j$ do
    if $label(node_i) = label(node_j)$ then
      $M_{ij} \leftarrow 1$
    else
      $C_{ij} \leftarrow 1$
    end if
  end for

  # Create the B matrix
  $B \leftarrow [1]_{k \times k}$
  for i = 1 to $k$ do
    $B_{ii} \leftarrow 0$
  end for

  # Randomly Initialize the $S$ matrix
  $S \leftarrow random\_array(n, k)$

  # Perform the required updation steps
  for iter = 1 to NUM_ITERATIONS do
    $XTS \leftarrow X^T S$
    $SSTS \leftarrow SS^T S$
    $MSBCS \leftarrow MSB + CS$
    for i = 1 to n do
      for c = 1 to k do
        $S_{ic} \leftarrow S_{ic} \times \frac{2(XTS)_{ic}}{2(SSTS)_{ic}+\alpha MSBCS_{ic}}$
      end for
    end for
  end for
```

Algorithm 2 Extract community information from S matrix

```
  $communities \leftarrow [0]_n$
  for i = 1 to n do
    # The community will be the one which has highest
    projection value
    $communities_i \leftarrow argmax(S_i)$
  end for
```

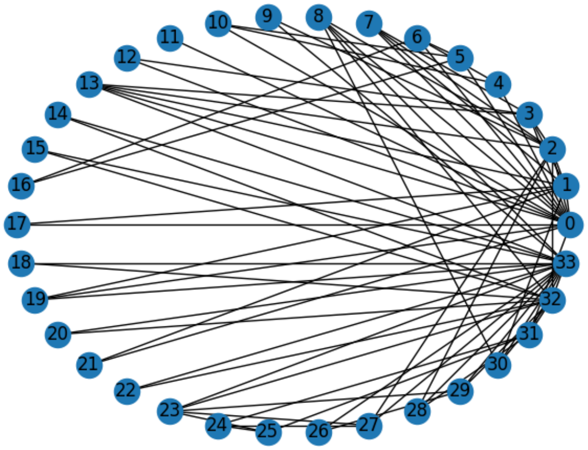

Fig. 2. Karate Club Network

B. Evaluation metrics

.

We have used three different kinds of metrics to evaluate the results of community detection and commenting on the performance of our method. [6] [7] [8]. We basically compare the detected community label to its ground truth value in our metrics. These metrics are define below.

1) Accuracy:

$$AC = \frac{\sum_{i=1}^{n} \delta(\gamma_i, map(l_i))}{n} \tag{16}$$

where, $n =$ number of nodes in the network, $l_i =$ community label, $\gamma_i =$ label detected by the method.

$$\delta(x,y) = \begin{cases} 1, & \text{x = y} \\ 0, & \text{otherwise} \end{cases}$$

$map(l_i)$ converts l_i to the equivalent label from the network. Kuhn-Munkres algorithm [9] can be used to find the best mapping.

2) NMI:

$$MI(C,C') = \sum_{c_i \in C, c'_j \in C'} p(c_i, c'_j) \cdot log \frac{p(c_i, c'_j)}{p(c_i, c'_j) \cdot p(c'_j)} \tag{17}$$

where, C and C' are two sets of given network communities, and $MI(C,C')$ is their mutual information metric.

$MI(C,C')$ ranges from 0 and $max(H(C), H(C))$, $H(C) =$ entropy of C and $H(C) =$ entropy of C'. If two communities are identical $MI(C,C') = max(H(C), H(C'))$, else if two communities are independent then $MI(C,C') = 0$. For all different kinds of permutations the value of MI remains unchanged.

NMI is normalized value of MI as follows:

$$NMI(C,C') = \frac{MI(C,C')}{max(H(C), H(C'))} \tag{18}$$

3) Modularity (Q): It is a way to measure the structure of network. It also measures community detection strength. It is defined as follows:

$$Q = \frac{1}{4m} s^T B s \tag{19}$$

where, $s =$ column vector, $B =$ real symmetric matrix.

$$B_{ij} = A_{ij} - \frac{k_i k_j}{2m} \tag{20}$$

C. Experimental Setup

We take 200-300 iterations of around 10 experiments and take average while reporting the results. $\alpha = 0.0005$ for our experiments. We write the code using Python 3.9 and import the karate club graph from the networkx package. There are $k = 2$ communities with text labels, we convert them to numeric 0 and 1 for easier comparision purposes.

D. Results

We observe that across all the experimental only 1 out of 34 labels is detected incorrectly.

Metric	Value
AC	0.970588
NMI	0.837169
Q(actual)	0.383711
Q(from predicted)	0.39423

References

[1] X. Shi, H. Lu, Y. He and S. He, ”Community detection in social network with pairwisely constrained symmetric non-negative matrix factorization,” 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM), 2015, pp. 541-546, doi: 10.1145/2808797.2809383.

[2] Lee DD, Seung HS. Learning the parts of objects by non-negative matrix factorization. Nature. 1999 Oct 21;401(6755):788-91. doi: 10.1038/44565. PMID: 10548103.

[3] Y. Yang and B. Hu, ”Pairwise Constraints-Guided Non-negative Matrix Factorization for Document Clustering,” IEEE/WIC/ACM International Conference on Web Intelligence (WI’07), 2007, pp. 250-256, doi: 10.1109/WI.2007.66.

[4] Haifeng Liu, Zhaohui Wu, Xuelong Li, Deng Cai and T. S. Huang, ”Constrained Nonnegative Matrix Factorization for Image Representation,” in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 34, no. 7, pp. 1299-1311, July 2012, doi: 10.1109/TPAMI.2011.217.

[5] Burred, Juan José. “Detailed derivation of multiplicative update rules for NMF.” (2017).

[6] Wei Xu, Xin Liu, and Yihong Gong. 2003. Document clustering based on non-negative matrix factorization. In Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval (SIGIR ’03). Association for Computing Machinery, New York, NY, USA, 267–273. DOI:https://doi.org/10.1145/860435.860485

[7] Newman ME. Modularity and community structure in networks. Proc Natl Acad Sci U S A. 2006 Jun 6;103(23):8577-82. doi: 10.1073/pnas.0601602103. Epub 2006 May 24. PMID: 16723398; PMCID: PMC1482622.

[8] D. Cai, X. He, J. Han and T. S. Huang, ”Graph Regularized Nonnegative Matrix Factorization for Data Representation,” in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 33, no. 8, pp. 1548-1560, Aug. 2011, doi: 10.1109/TPAMI.2010.231.

[9] L. Lovasz and M. Plummer, Matching theory. Elsevier Science Ltd, 1986, no. 121.