Non-linear Attributed Graph Clustering by Symmetric NMF with PU Learning

Sana Oubenyahya   Chaimae Hilal   Kenza Ouazzani Chahdi   Mouad Et-tali

Université de Paris

Context of our problem

We consider the problem of **clustering attributed graphs**. The challenge is how to design an effective and efficient clustering method that precisely captures the hidden relationship between the topology and the attributes in real-world graphs.

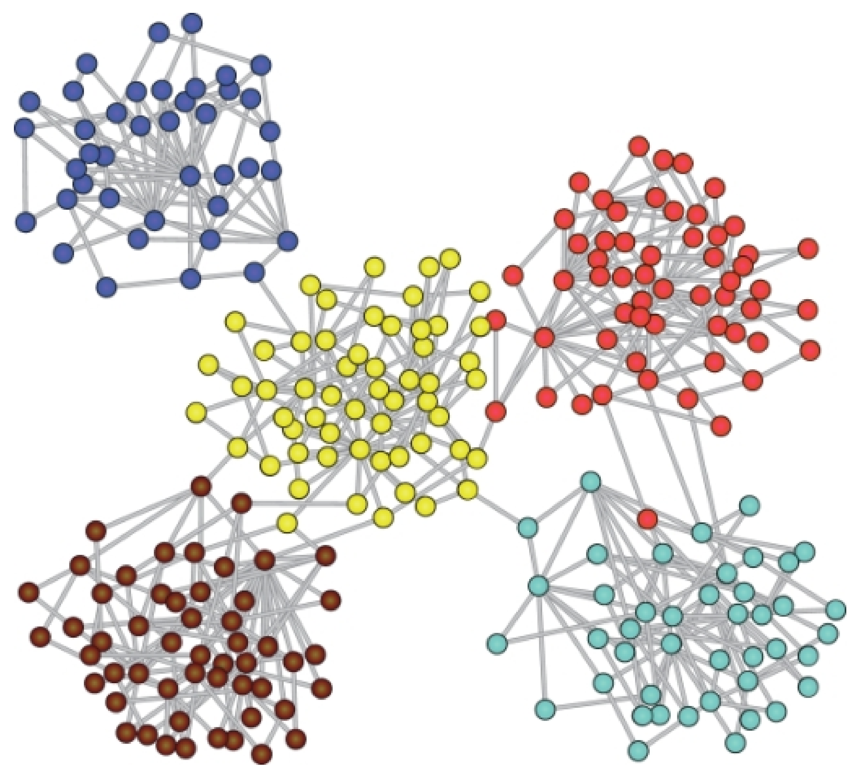

Fig 1: Example of community detection, a specific problem of Graph clustering

Although **NMF** has been shown to be effective to perform clustering, the goals of clustering and dimensionnality reduction are different. Especially in Real world graphs, this is why this article proposes Symmetric Non-negative Matrix Factorization with Positive Unlabeled Learning

Symmetric Non-negative Matrix Factorization

SymNMF inherits the advantages of NMF by enforcing nonnegativity on the clustering assignment matrix. Most of NMF's clustering successes have been around document clustering. One reason is that each basis vector represents a topic's word distribution, and documents with similar word distributions should be classified in the same group. This property is not valid in all data types. This is why SymNMF was introduced by Kuang, Ding and Park in 2012 [2].

SymNMF is based on a similarity measure between data points, and factorizes a symmetric matrix containing pairwise similarity values (not necessarily nonnegative) which is why it estimates a cluster assignment matrix U by minimizing a non-convex loss function that uses S as input :

$$\min_{\boldsymbol{U}\geq 0} \|\boldsymbol{S} - \boldsymbol{U}\boldsymbol{U}^\top\|_{\mathcal{F}}^2 \qquad (1)$$

The main goal of graph clustering is to find a partition of vertices in a graph where the similarity between vertices is high within the same cluster and low across different clusters.

PU learning

Positive unlabeled learning : Given a set of examples of a particular class P (positive class) and a set of unlabeled examples U, which contains both class P and non-class P (negative class) instances, the goal is to build a binary classifier to classify the test set T into two classes, positive and negative, where T can be U.

(1) Train classifier to predict the probability that a sample x is labled P(s=1| x)
(2) Use the classifier to predict the probability that the positive samples are labeled P(s=1|y=1|x)
(3) Use the classifier to predict the probability that sample K is labeled P(s=1|k)
(4) Estimate the probability that K is positive by calculating P(s=1|k) / P(s=1|y=1)

Non-linear Attribute Graph Clustering using SymNMF

Using SymNMF as variant of NMF we can find a reasonable cluster assignment by considering the complex relationship between the **topology** and the **attributes**.
The method in this article jointly decomposes the adjacency matrix S and the attribute matrix X into factor matrices with learning a non-linear projection function (for example sigmoid). This function can transfer a cluster assignment extracted from the adjacency matrix to that from the attribute matrix.
The article defines the method as a minimization problem of a non-convex loss and it's noted as follows:

$$\min_{\boldsymbol{U},\boldsymbol{V},\boldsymbol{H}\geq 0} \mathcal{L}_\rho(\boldsymbol{S} - \boldsymbol{U}\boldsymbol{U}^\top) + \frac{\lambda}{2}\|\boldsymbol{X} - f(\boldsymbol{U}\boldsymbol{H})\boldsymbol{V}^\top\|_{\mathcal{F}}^2$$

Where Lρ(Z) denotes an approximation error of the adjacency matrix S with the ρ-weighted loss.

$$\mathcal{L}_\rho(\boldsymbol{Z}) = \sum_{(i,j)\in E} \rho(z_{i,j}-1)^2 + (1-\rho)\sum_{(i,j)\notin E} z_{i,j}^2$$

Next we'll define some notions to illustrate the algorithm and the decompostisions:

Table 1: Definition of main symbols.

Variable	Explanation
$\boldsymbol{S} \in \mathbb{R}_+^{n\times n}$	adjacency matrix
$\boldsymbol{X} \in \mathbb{R}_+^{n\times m}$	attribute matrix
$\boldsymbol{U} \in \mathbb{R}_+^{n\times k_1}$	cluster assignment matrix
$\boldsymbol{V} \in \mathbb{R}_+^{m\times k_2}$	attribute factor matrix
$\boldsymbol{H} \in \mathbb{R}_+^{k_1\times k_2}$	cluster assignment tansfer matrix
$\boldsymbol{W} \in \mathbb{R}_+^{n\times n}$	mask matrix of $\boldsymbol{S}$
$k_1 \in \mathbb{N}$	number of clusters
$k_2 \in \mathbb{N}$	number of clusters for attributes
$\lambda \geq 0$	balancing parameter between the topology and the attributes
$\rho = [0,1]$	bias weight for $\boldsymbol{S}$
$t \in \mathbb{N}$	number of iterations

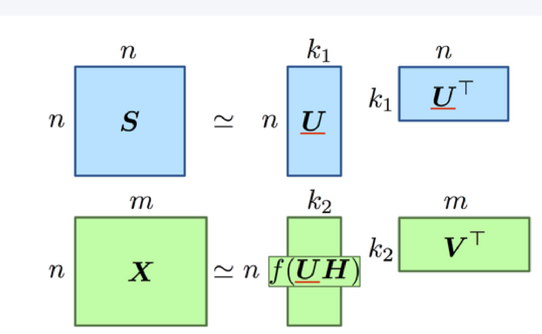

Figure 1: Illustration of NAGC. $\boldsymbol{S}$ and $\boldsymbol{X}$ are an adjacency matrix and an attributed matrix, respectively. $\boldsymbol{U}$, $\boldsymbol{V}$, and $\boldsymbol{H}$ denote a cluster assignment, an attribute factor, and a cluster assignment transfer matrices, respectively. f is a non-linear activation function. NAGC merges different cluster structures among $\boldsymbol{S}$ and $\boldsymbol{X}$ by supposing a shared parameter $\boldsymbol{U}$.

Algorithm 1 NAGC algorithm

Input: $\boldsymbol{S}, \boldsymbol{X}, k_1, k_2, \lambda, t$
Output: clustering result C

```
Preprocess: S, X
Initialize: U, V, H
while t' < t do
    # alternatively update parameters
    U^(t'+1) ← update (U^(t')) by Eq. (14)
    V^(t'+1) ← update (V^(t')) by Eq. (15)
    H^(t'+1) ← update (H^(t')) by Eq. (16)
end while
while n' < n do
    # assign each vertex to the clusters
    c_n' ← argmax_l {u_n',l | l = (1, ..., k)}
end while
```

update rules for each parameter.

$$\boldsymbol{U} \leftarrow \boldsymbol{U} \odot [2\rho\boldsymbol{S}\boldsymbol{U} + \lambda\{(\boldsymbol{X}\boldsymbol{V}) \odot f'(\boldsymbol{U}\boldsymbol{H})\}\boldsymbol{H}^\top] \oslash [2\rho(\boldsymbol{U}\boldsymbol{U}^\top \odot \boldsymbol{W})\boldsymbol{U} + 2(1-\rho)(\boldsymbol{U}\boldsymbol{U}^\top \odot \boldsymbol{W}')\boldsymbol{U} + \lambda\{(f(\boldsymbol{U}\boldsymbol{H})\boldsymbol{V}^\top\boldsymbol{V}) \odot f'(\boldsymbol{U}\boldsymbol{H})\}\boldsymbol{H}^\top] \qquad (14)$$

$$\boldsymbol{V} \leftarrow \boldsymbol{V} \odot \{\boldsymbol{X}^\top f(\boldsymbol{U}\boldsymbol{H})\} \oslash \{\boldsymbol{V} f(\boldsymbol{U}\boldsymbol{H})^\top f(\boldsymbol{U}\boldsymbol{H})\} \qquad (15)$$

$$\boldsymbol{H} \leftarrow \boldsymbol{H} \odot [\boldsymbol{U}^\top\{f'(\boldsymbol{U}\boldsymbol{H}) \odot (\boldsymbol{X}\boldsymbol{V})\}] \oslash [\boldsymbol{U}^T\{f'(\boldsymbol{U}\boldsymbol{H}) \odot f(\boldsymbol{U}\boldsymbol{H})\}\boldsymbol{V}^\top\boldsymbol{V}] \qquad (16)$$

Experimental analysis

We used four graph datasets to evaluate the clustering quality and efficiency of our different methods. The following table summerizes the characteristics of each dataset.

Table 1 : Summary of the datasets

Dataset	Vertex n	Edge $\lvert E\rvert$	Attribute m	Label k_1	Density $\lvert E\rvert/n^2$
WebKB	877	1480	1703	4	0.18%
Citeseer	3312	4660	3703	6	0.04%
Cora	2708	5278	1433	7	0.07%
polblog	1490	16630	7	2	0.75%

Results on graph clusterings

The results on the following table are an evaluation of the clustering quality by using the average and standard deviation of ARI.
The methods annotated with * indicate the parameters are initialized by random values.

Table: The average and standard deviation (in parenthesis) of ARI.

Method	Input Type	WebKB	Citeseer	Cora	polblog
Prop.	Topology, Attribute	**0.995** (±0.002)	**0.280** (±0.027)	**0.348** (±0.022)	**0.626** (±0.037)
Prop. (w/o PU)	Topology, Attribute	0.990 (±0.005)	0.221 (±0.010)	0.270 (±0.024)	0.621 (±0.000)
Prop. *	Topology, Attribute	0.982 (±0.003)	0.126 (±0.023)	0.244 (±0.038)	0.603 (±0.011)
JWNMF	Topology, Attribute	0.906 (±0.000)	0.127 (±0.000)	0.230 (±0.000)	0.517 (±0.000)
JWNMF*	Topology, Attribute	0.909 (±0.002)	0.082 (±0.009)	0.227 (±0.011)	0.504 (±0.011)
BAGC	Topology, Attribute	0.204 (±0.000)	0.000 (±0.000)	0.016 (±0.000)	0.000 (±0.000)
METIS	Topology	0.851 (±0.000)	0.156 (±0.000)	0.283 (±0.000)	0.545 (±0.000)
SNMF	Topology	0.840 (±0.100)	0.067 (±0.020)	0.211 (±0.023)	0.498 (±0.059)
NMF	Attribute	0.327 (±0.004)	0.193 (±0.023)	0.115 (±0.001)	0.000 (±0.000)
k-means	Attribute	0.260 (±0.131)	0.190 (±0.044)	0.093 (±0.034)	0.000 (±0.000)

The article's method (Prop) initialized by k-means results outperforms all the other methods for all datasets. This result confirms the potency of the non-linear projection and PU learning to the clustering quality. The results of this method without PU learning (Prop. (w/o PU)) shows the advantage of the non-linear approach. In all datasets, it performs better than the competing methods except for Cora dataset on which the METIS method took the first place.
Furthermore, the initialization by k-means result always improves the performance (Prop) compared to the one initialized by random values (Prop*).
The method JWNMF took the second place for WebKB dataset but resulted in poor performance on other datasets, whereas METIS, that is a graph clustering method, achieved the second/third places for WebKB, Cora, and Polblog datasets. However, for Citeseer dataset, NMF and k-means, those are attribute based clustering methods, took the second and third places, respectively. These results demonstrate that either the topology or attributes of the graphs affect remarkably the clustering quality, but it is more effective to combine both of them.

References

[1] Seiji Maekawa , Koh Takeuchi , Makoto Onizuka .Non-linear Attributed Graph Clustering by Symmetric NMF with PU Learning.
[2] Da Kuang, Chris Ding, Haesun Park. Symmetric Nonnegative Matrix Factorization for Graph Clustering.