Experimental study of approximation algorithms for a graph clustering problem

Morshinin A.V.

Sobolev Institute of Mathematics SB RAS

Omsk-2023

The research was supported by RSF grant 22-71-10015

Clustering problems

In the *clustering problem* we must split a set of objects into several subsets based on the similarity of the objects to each other. A set of objects can be represented as vertices of a graph, and the similarity of objects can be specified as edges of this graph.

A graph is clustered if each of its components is a clique (*cluster*). Additionally, restrictions on the number of components, the size of components can be introduced. Clustering problems are related to *unsupervised learning*. However, *semi-supervised* methods and algorithms are also applicable. In this approach, we have a supervisor who can partition some objects across clusters.

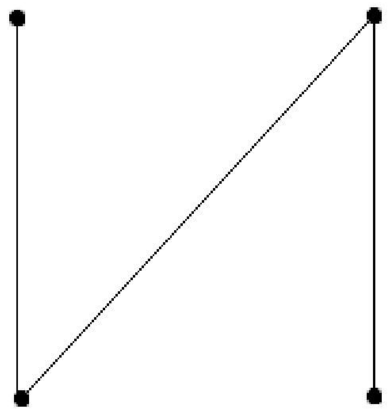
Basic definitions

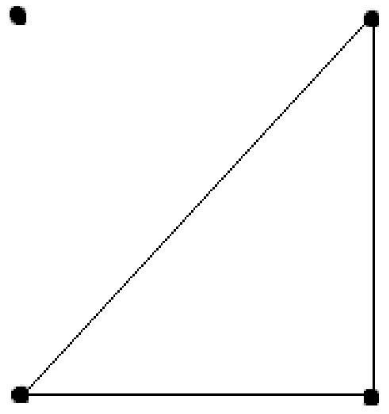
A *cluster graph* is a graph, each component of which is a complete graph.

The *distance* $\rho(G_1, G_2)$ between two labeled graphs $G_1 = (V, E_1)$ and $G_2 = (V, E_2)$ is the cardinality of the symmetric difference $E_1 \Delta E_2$ (the elements of this set are called *disagreements*).

For the vertex v of the graph $G = (V, E)$ we call a *neighborhood* $N_G(v)$ the set of $u \in V$ joined with v.

Cluster graph example





We have to remove one edge and add one edge.

Semi supervised graph clustering

k-SEMI-SUPERVISED GRAPH CLUSTERING. The input is a graph $G = (V, E)$, an integer $2 \leq k \leq |V|$, a set $Z = \{z_1, \ldots, z_k\} \subset V$. The aim is to find cluster graph C with k clusters minimizing the number of disagreements. Additionally, vertices of Z must belong to different clusters of C.

The problem is *NP*-hard.

3-approximation algorithm for 2-ss-clustering

The Neighborhood Semi-Supervised Algorithm (NS).

Construct the set F of feasible solutions according to the rules.

a) For each vertex $v \in V \setminus \{z_1, z_2\}$ build two cluster graphs C_v^1 and C_v^2 by the following. The 1st cluster of the 1st graph is $\{v\} \cup (N_G(v) \setminus \{z_1\}) \cup \{z_2\}$, the 1st cluster of the 2nd graph is $\{v\} \cup (N_G(v) \setminus \{z_2\}) \cup \{z_1\}$. Both 2nd clusters contain not yet clustered vertices.

b) For each vertex $v \in \{z_1, z_2\}$ build cluster graphs C_v by the following. The 1st cluster is $(\{v\} \cup N_G(v)) \setminus \{x\}$. 2nd cluster contains not yet clustered vertices ($x = z_1$ if $v = z_2$, $x = z_2$ if $v = z_1$).

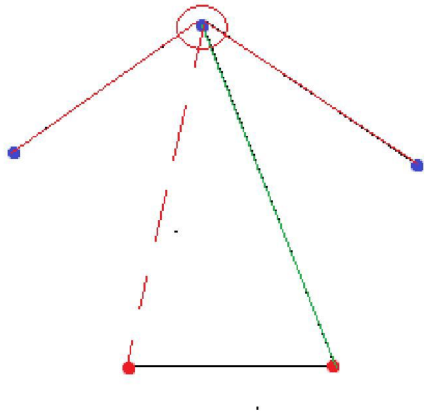
Return $C_{NS} \in F$ with minimum $\rho(G, C_{NS})$.

Local search for 2-ss-clustering

LocalSearch.

Let C be a cluster graph for a graph $G = (V, E)$. For each $v \in V \setminus \{z_1, z_2\}$ let C_v be the same cluster graph as C, except with v in the opposite cluster. We then define $\lambda_v = \rho(G, C) - \rho(G, C_v)$, the improvement caused by the change. Let u be the vertex with maximum λ_v. If $\lambda_v \leq 0$, stop, otherwise let $C \leftarrow C_w$ and repeat.

Local search for 2-ss-clustering



When we move a blue vertex to a red cluster, we don't count the green edge, but we count the red ones. Therefore, the value of the objective function will increase by 2.

Approximation algorithms for 2-ss-clustering

The 2-approximation **Neighborhood Semi-Supervised with LocalSearch Algorithm (NSLS)** from is the application of local search to each cluster graph $C \in F$. Let C_{NSLS} be a solution built by this algorithm.

We also want to research **Pre-Clustered Neighborhood Semi-Supervised with LocalSearch Algorithm (PNSLS)** which applies local search only for C_{z1} and C_{z2} from F. Let C_{PNSLS} be a solution built by this algorithm.

Background of experimental study

Complexity of **NS** - $O(n^2)$, **NSLS** - $O(n^4)$, **PNSLS** - $O(n^3)$.

It's easy to see that $\rho(G, C_{NS}) \leq \rho(G, C_{NSLS})$ and $\rho(G, C_{PNSLS}) \leq \rho(G, C_{NSLS})$ for any $G = (V, E)$. Let's define $E_{NS}(G) = \rho(G, C_{NS}) / \rho(G, C_{NSLS})$ and $E_{PNSLS}(G) = \rho(G, C_{PNSLS}) / \rho(G, C_{NSLS})$ as errors of **NS** and **PNSLS** relatively to **NSLS**.

Then define $\boldsymbol{E}_{NS}(n)$ and $\boldsymbol{E}_{PNSLS}(n)$ as expected values of the E_{NS} and E_{PNSLS} for all graphs with n vertices. Let's formulate the main assumption to be investigated.

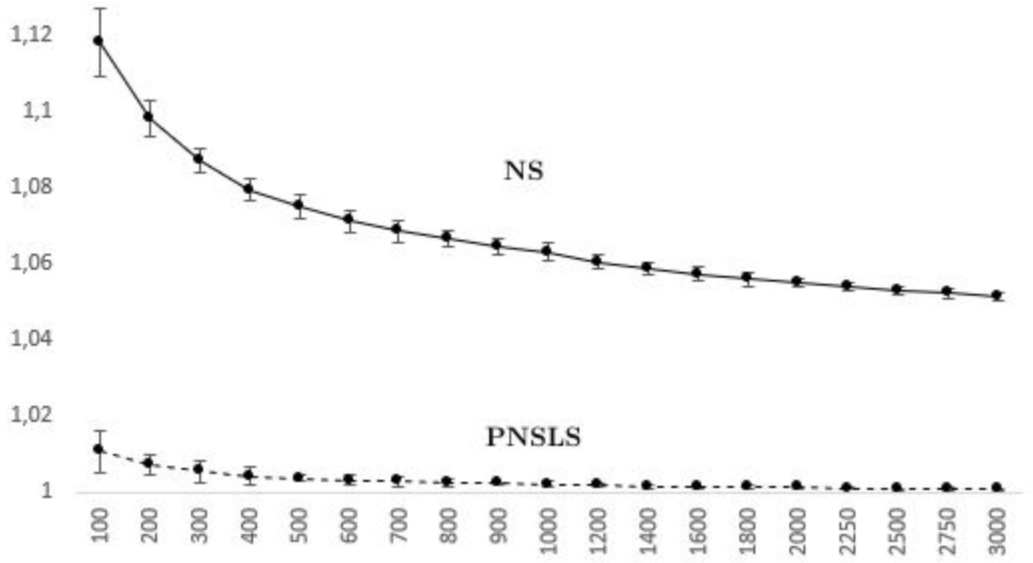
Assumption. As the number of vertices n increases, $\boldsymbol{E}_{PNSLS}(n)$ tends to 1 and $\boldsymbol{E}_{NS}(n)$ doesn't tend to 1.

Experimental study

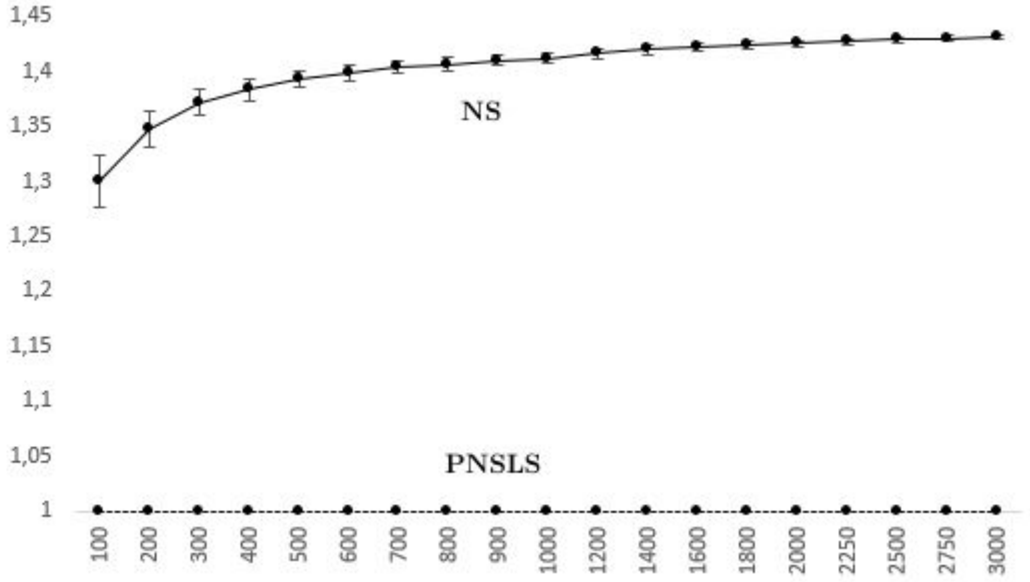
The experimental study was done on random graphs $G(n, p)$ generated with *Erdős-Rényi model*. The number of vertices $n \in \{100, 200, \ldots, 3000\}$, graph density $p \in \{0.33, 0.5, 0.67\}$. 100 examples were solved for each n and p. Based on the sample data, the sample mean of $\boldsymbol{E}_{NS}(n)$ and $\boldsymbol{E}_{PNSLS}(n)$ was calculated. Further, with significance level $\alpha = 0.05$, we calculated the confidence interval. We used the quantile of the normal distribution to calculate the confidence interval.

Statistical validity was obtained by the *Kolmogorov-Smirnov test*. For each n and p, the statistic was less than the critical value 1.36.

Error plot of $\boldsymbol{E}_{NS}$ and $\boldsymbol{E}_{PNSLS}$ for p = 0.33



Error plot of $\boldsymbol{E}_{NS}$ and $\boldsymbol{E}_{PNSLS}$ for $p = 0.67$



Average working time of **NS**, **NSLS** and **PNSLS**

n	500	600	700	800	900	1000	1200	1400	1600	1800	2000	2500	3000
NS	0	0	0	0	0	1	1	2.21	4.01	5.99	8.04	15.49	27.65
NSLS	0.07	1	1.04	2	3	4	6.65	10.73	16.45	25.32	32.84	66.45	119.39
PNSLS	0	0	0	0	0	1	1	2.32	4.02	6	8.04	15.78	27.78

Thank you for your attention.