fast label propagation algorithm for community detection

Vincent A. Traag
Leiden University
Centre for Science and Technology Studies

Lovro Šubelj
University of Ljubljana
Faculty of Computer and Information Science

NetSci '23

community detection

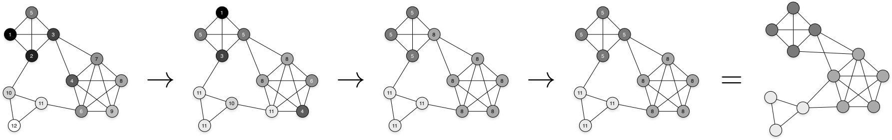
many approaches with different interpretations & algorithms

graph partitioning, objective functions, networks dynamics, generative models etc.

label propagation one of fastest algorithms

but not most robust or preferable for small networks

even faster variant today



label propagation

heuristic algorithm for community detection

1. create singleton partition $\{c_i\}_{i=1}^{n} \leftarrow \{1, 2, 3 \ldots n\}$
2. update each label c_i to most common label in neighborhood Γ_i

$$\{c\} \leftarrow \mathrm{argsmax}_c \sum_j A_{ij}\delta(c_j, c)$$

$$c_i \leftarrow \mathrm{random}_c\{c\}$$

3. repeat 2. until convergence or stopping criteria

community structure $\approx$ local optima of $\sum_{ij} A_{ij}\delta(c_i, c_j)$

global optima $c_i = c$ is trivial and uninteresting

algorithm guarantees $\rightarrow$ labels c_i are locally optimal

each node i has most neighbors with label c_i

algorithm variants

LPA

original label propagation algorithm

retention

label propagation algorithm with retention strategy

1. update order

update c_i in parallel (oscillations)

update c_i in random order (✓)

2. breaking ties

always sample $c_i \leftarrow \mathrm{random}_c\{c\}$ (LPA)

only when $c_i \notin \{c\}$ (retention)

3. stopping criteria

all c_i are locally optimal (LPA)

no c_i has changed (retention)

fast variant

queue

maintain **queue of nodes** that could update their label

after updating c_i **push neighbors** $j \in \Gamma_i$ **to queue** if $c_j \neq c_i$

FLPA

fast variant of label propagation algorithm

FLPA has **same guarantees** as LPA

1. update order

...

queue of nodes **to update** (FLPA)

2. breaking ties

always sample $c_i \leftarrow \mathrm{random}_c\{c\}$ (LPA)

...

3. stopping criteria

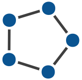
...

queue is **empty** (FLPA)

theoretical graphs

analysis for three theoretical graphs

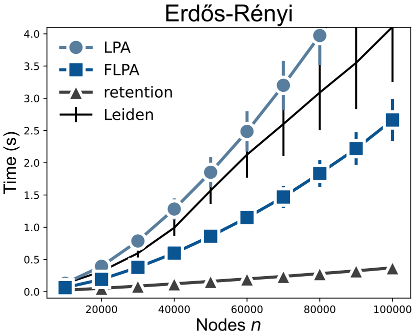
LPA, FLPA & retention strategy find same partition but FLPA has lower complexity

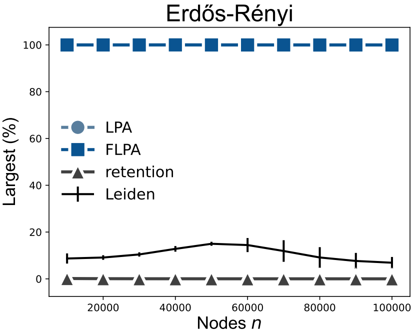
Graph	LPA	retention	FLPA
	$\Omega(2m)$	$\Theta(2m)$	$\Theta\left(m + \frac{1}{n-2}\right)$
	$\Theta\left(2m + \frac{2n(n-2)}{(n-1)^2}\right)$	$\Theta(2m)$	$\Theta\left(m + \frac{n-2}{(n-1)^2}\right)$
	depends on n	$\approx$ 2.7 nodes	$\approx$ 4.1 nodes

random graphs

Erdős-Rényi graphs without structure

LPA & FLPA find no community structure different from retention strategy

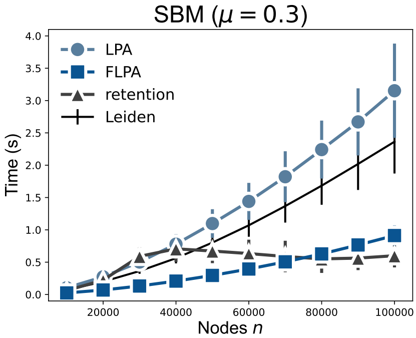


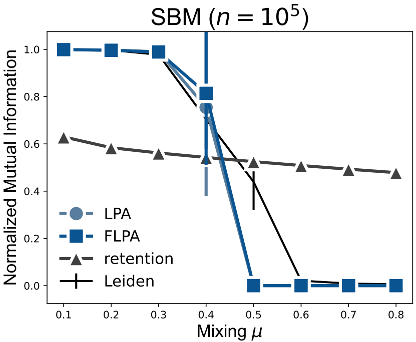


community structure

SBM graphs with 100 planted communities

LPA & FLPA find planted community structure different from retention strategy

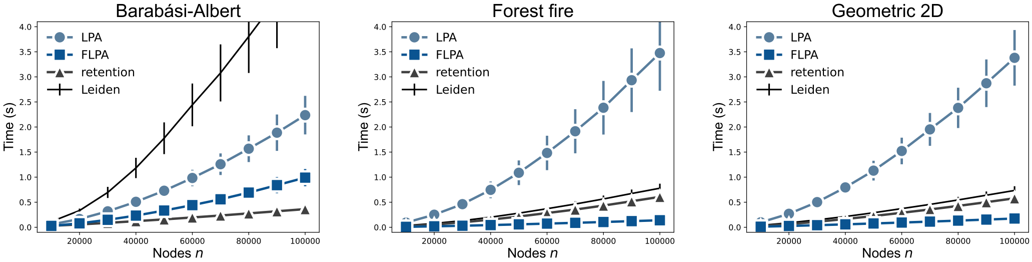




other structure

synthetic graphs with different structure

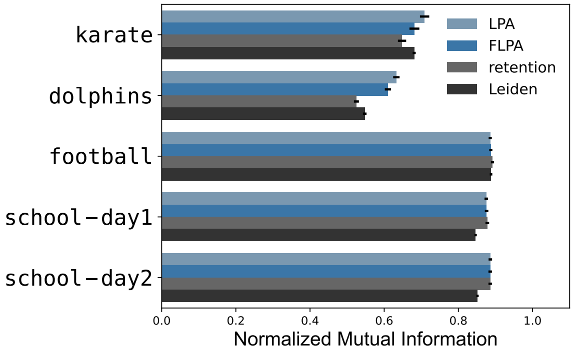
FLPA is consistently faster than LPA & Leiden algorithm



social networks

small social networks with sociological partitioning of nodes

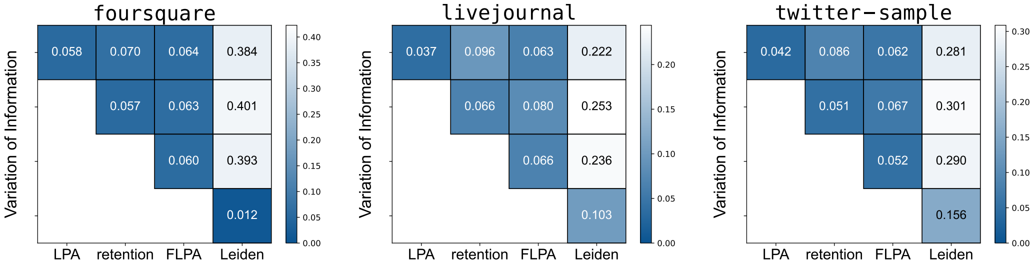
LPA & FLPA find similar partitions better than retention strategy



large networks (1/2)

comparison of partitions of large empirical networks

LPA & FLPA **find similar partitions** different from Leiden algorithm



large networks (2/2)

runtime for large empirical networks

FLPA up to 700× faster than LPA & 15× faster than retention strategy

Network	Nodes	Edges	Algorithm	Time (s)	Speedup
`com-dblp`	317 080	1 049 866	LPA	185.0	188.5×
			retention	10.6	10.8×
			FLPA	1.0	
`roadnet-ca`	1 965 206	5 533 214	LPA	940.4	161.8×
			retention	23.2	4.0×
			FLPA	5.8	
`us-patents`	3 774 768	16 522 438	LPA	26 704.4	705.2×
			retention	601.8	15.9×
			FLPA	37.9	
`foursquare`	3 935 215	22 809 624	LPA	977.3	63.8×
			retention	117.6	7.7×
			FLPA	15.3	
`livejournal`	4 847 571	68 993 773	LPA	2 248.1	30.2×
			retention	959.6	12.9×
			FLPA	74.4	
`twitter-sample`	5 384 162	16 011 444	LPA	1 343.5	93.0×
			retention	92.8	6.4×
			FLPA	14.5	

conclusions

LPA is useful for **fast first look** at network

other slower methods are arguably more robust and preferable

FLPA is **consistently faster** than LPA with **same guarantees**

FLPA seems to bring benefits to LPA at no additional costs

FLPA is **generally preferable** to LPA

`graph_info(G)`, network compression, FLPA followed by Leiden etc.

thank you!

arXiv:2209.13338v2

http://github.com/vtraag/igraph/tree/flpa

Traag & Šubelj (2023). Large network community detection by fast label propagation. *Scientific Reports* **13**, 2701.

Vincent A. Traag
Leiden University
v.a.traag@cwts.leidenuniv.nl
http://www.traag.net

Lovro Šubelj
University of Ljubljana
lovro.subelj@fri.uni-lj.si
http://lovro.fri.uni-lj.si