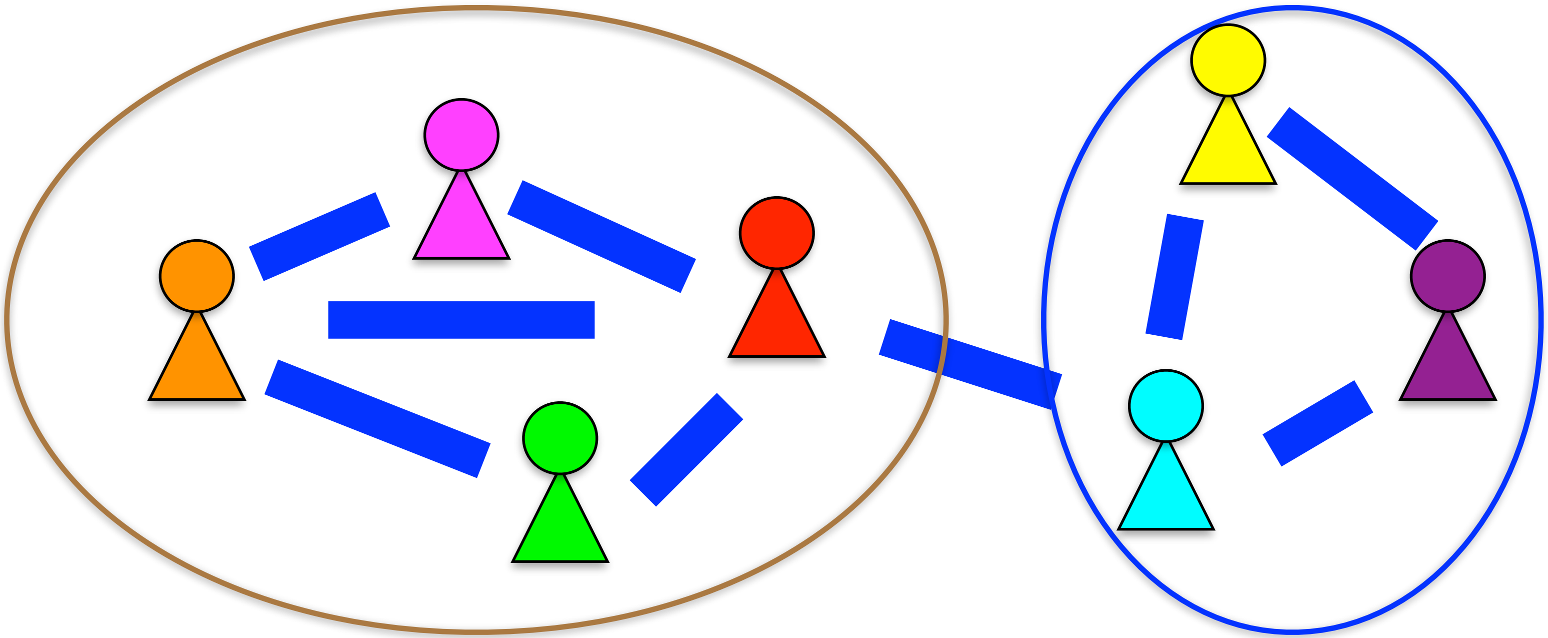
Ego-Splitting Framework: from Non-Overlapping to Overlapping Clusters.

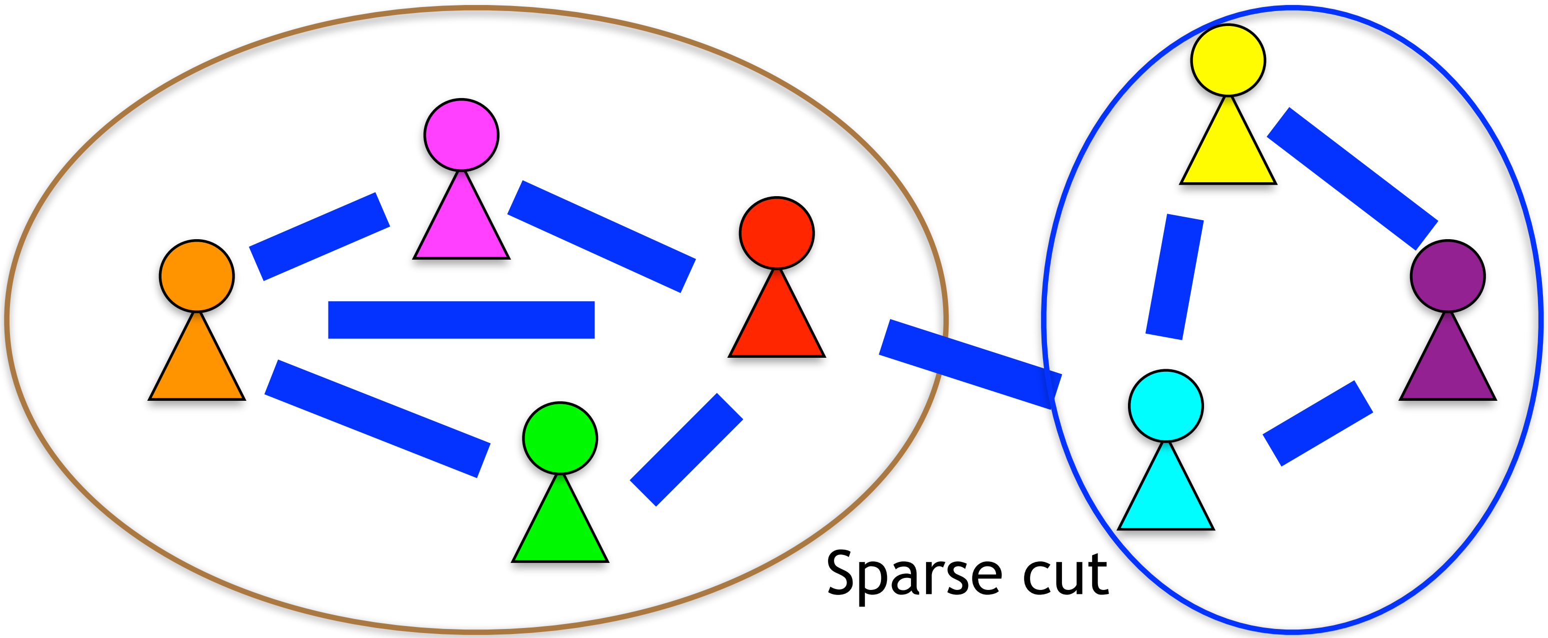
Alessandro Epasto
(Google)

Joint work with:
Silvio Lattanzi, Renato Paes Leme
(Google)

Community Detection in an Ideal World



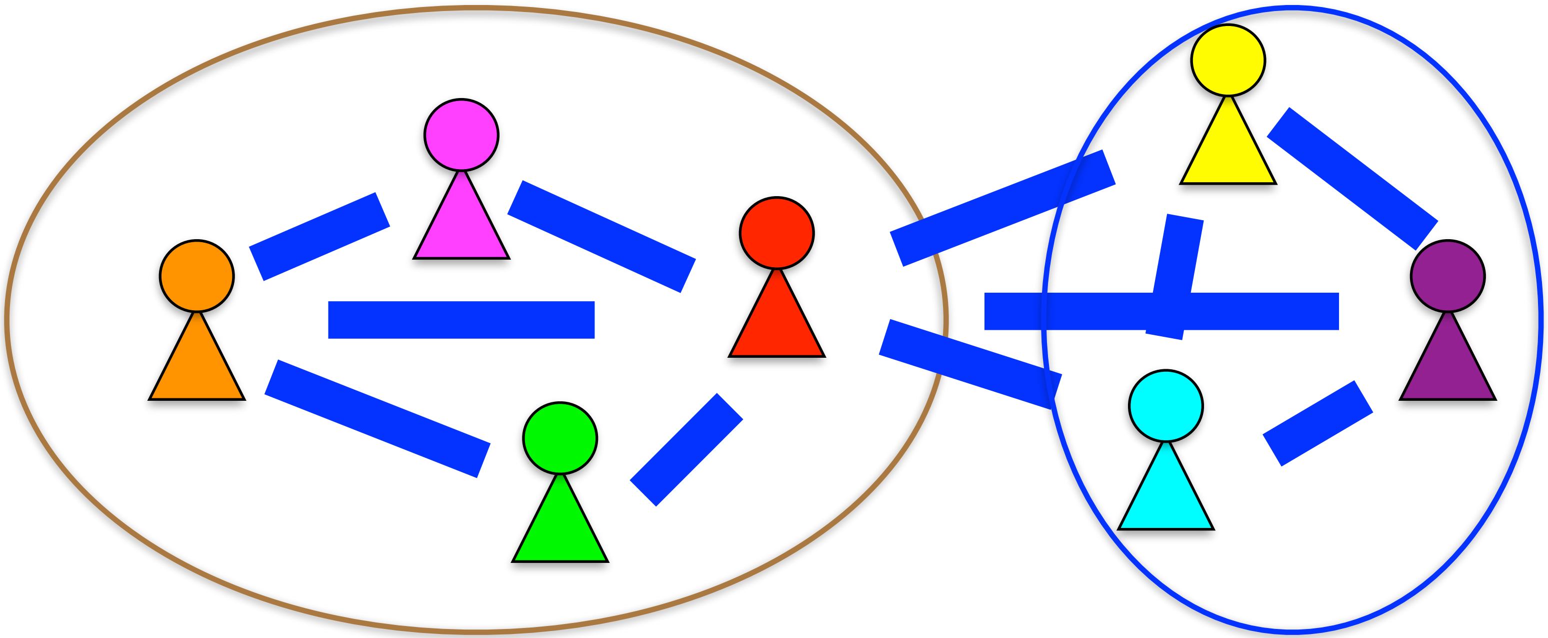
Community Detection in an Ideal World



Dense communities

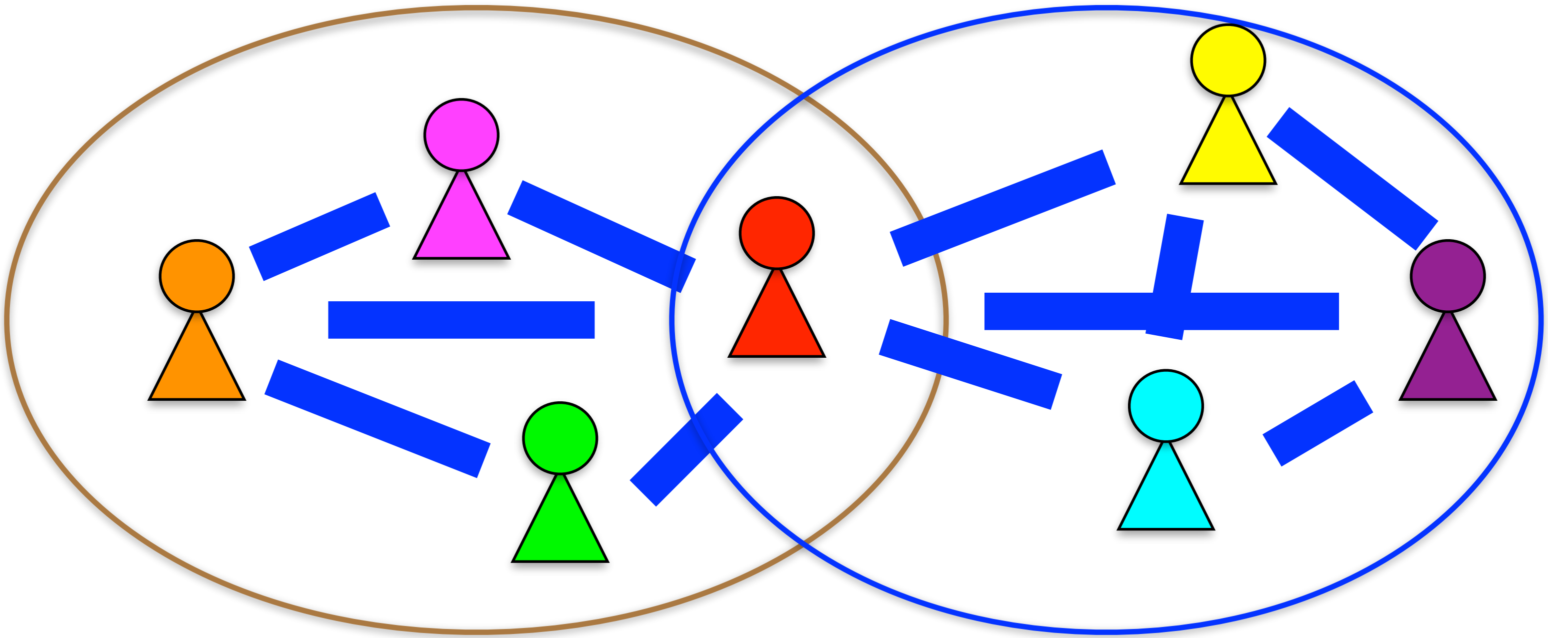
Disjoint clusters

Community Detection in the Real World



Large cut

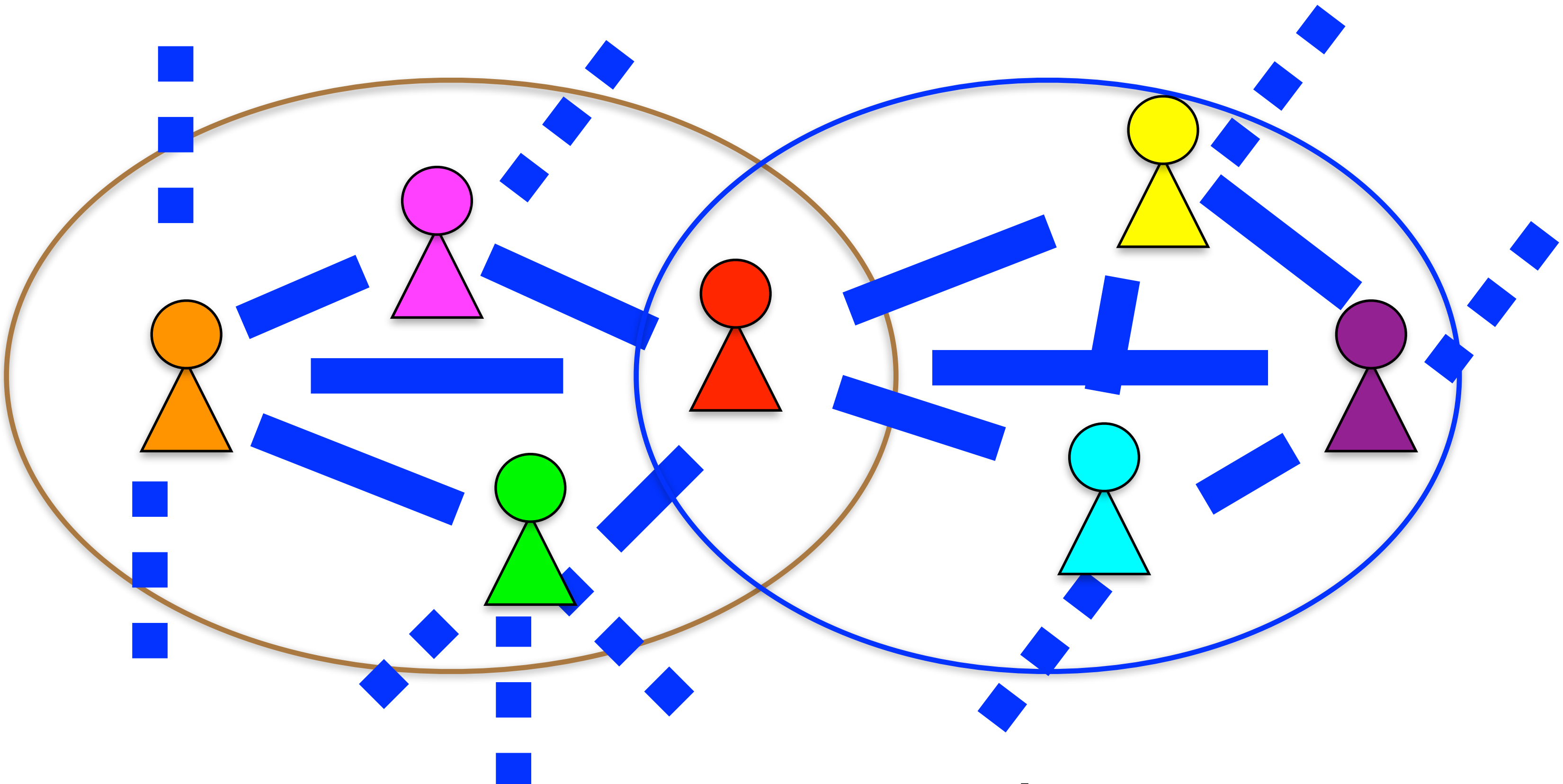
Community Detection in the Real World



Large cut

Communities **overlap** heavily.

Community Detection in the Real World



Large cut

Communities **overlap** heavily.

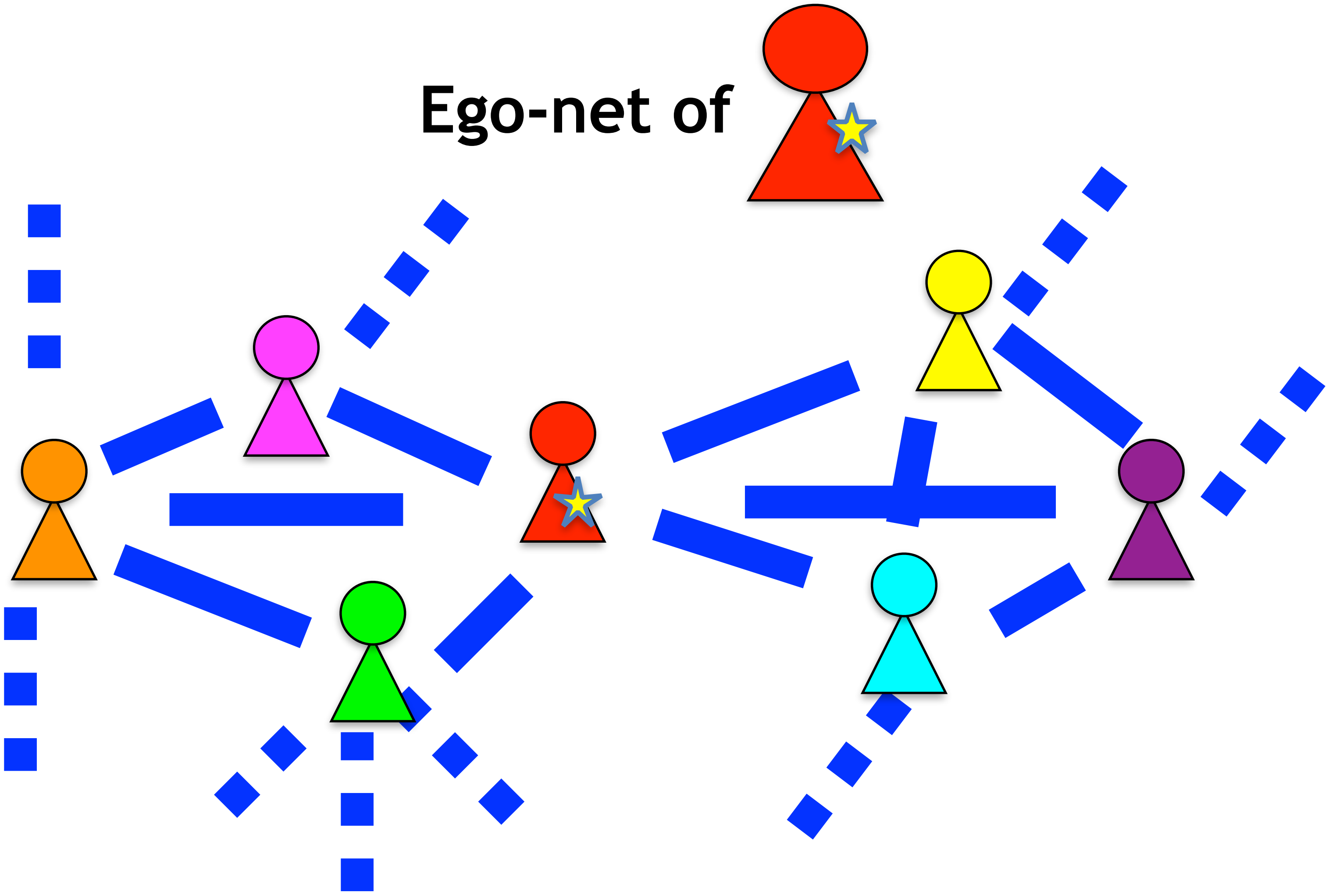
More connections with outside than with inside

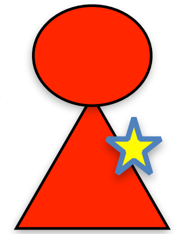
Global Community Structure

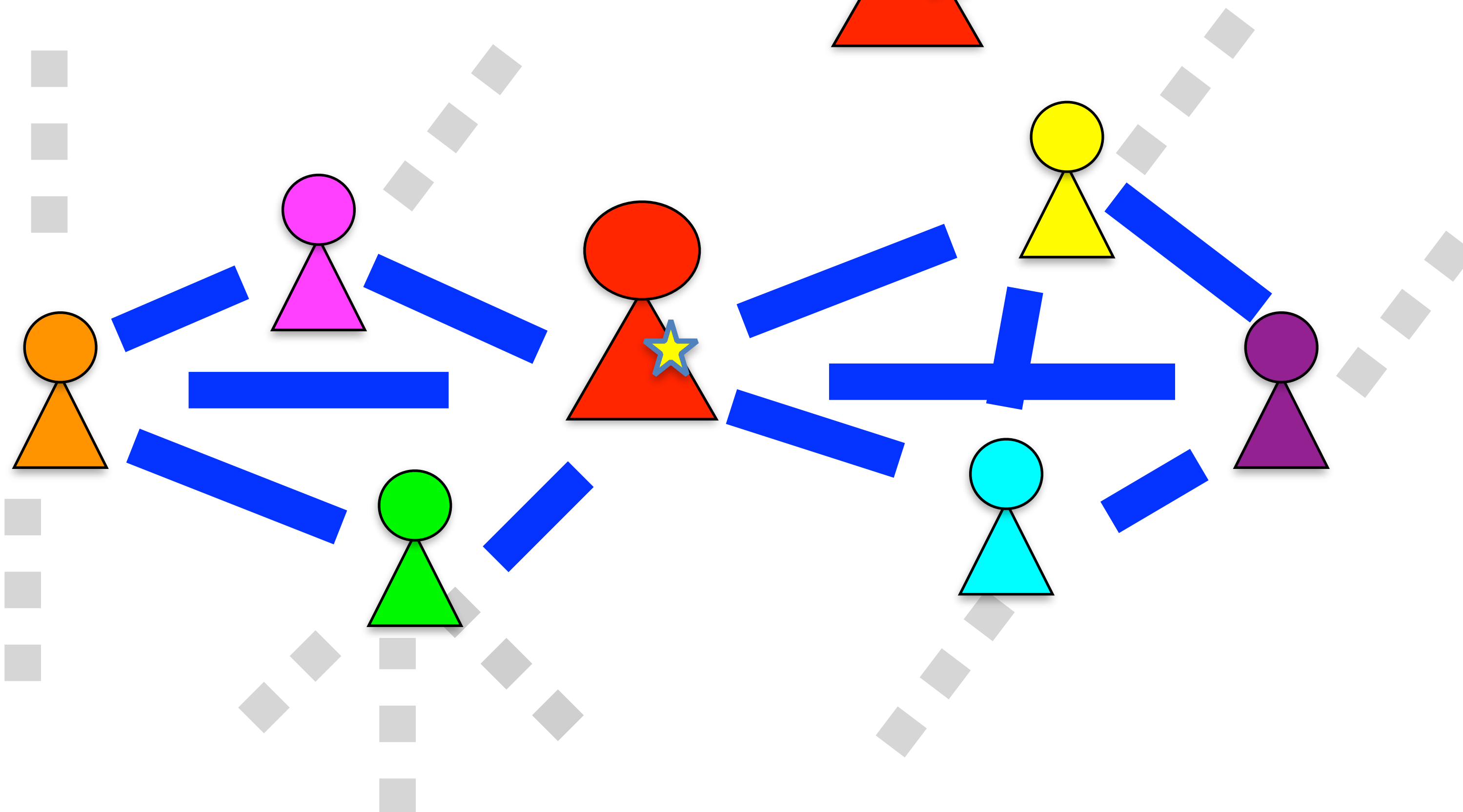
Community detection is hard at the global graph level:

- No clear **macroscopic** community structure at global graph level [Leskovec et al., 2009].
- No medium-sized low-conductance communities.
- Real-world communities do not follow the assumptions of the algorithms [Abraho et al., 2014].

Intuition: Community structure is clearer at microscopic level of node-centric structures called ego-networks.


Ego-net of

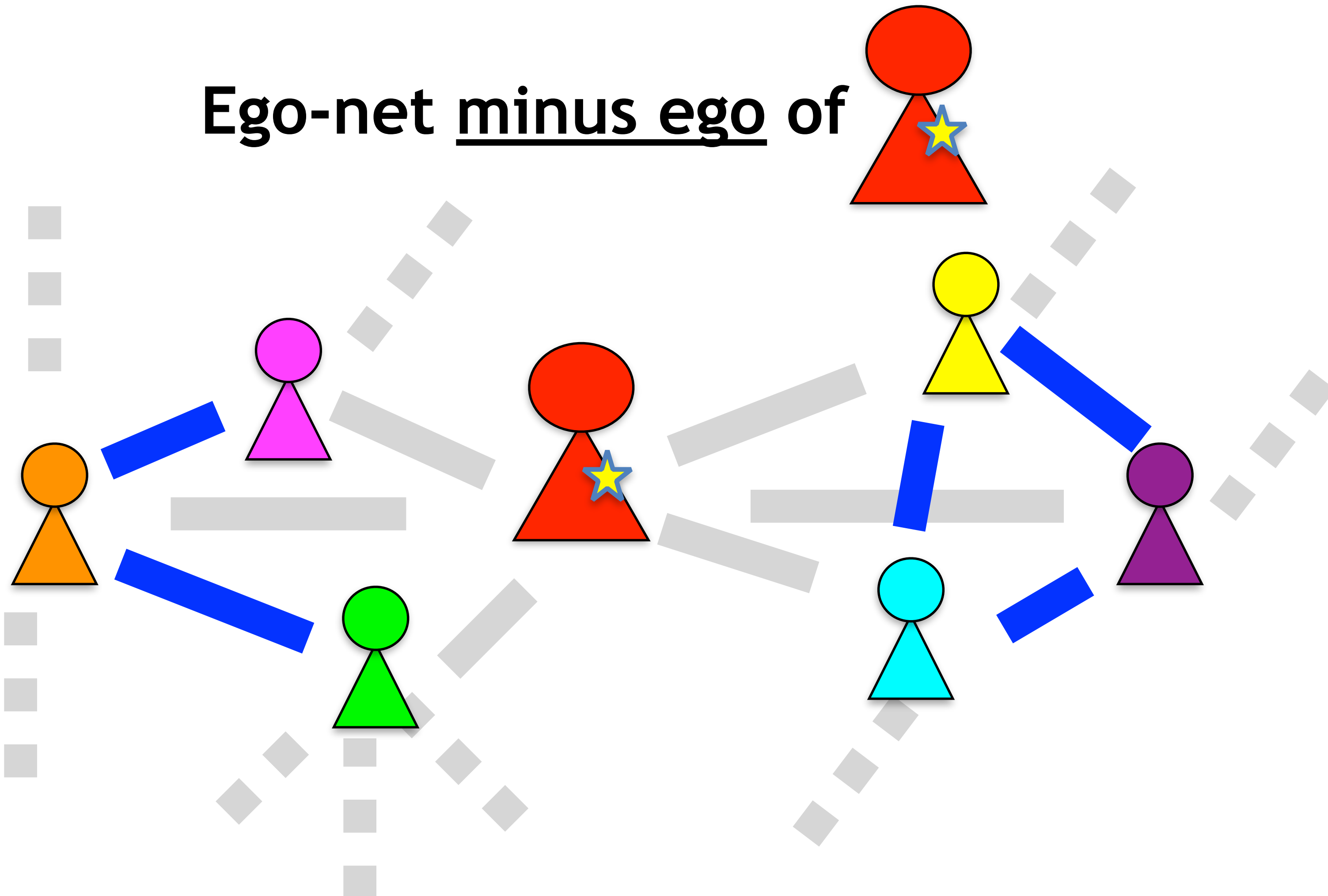
Ego-net of 



The **Ego-net** of node u (a.k.a. *ego-network*), is defined as the induced subgraph on {u, N(u)}.

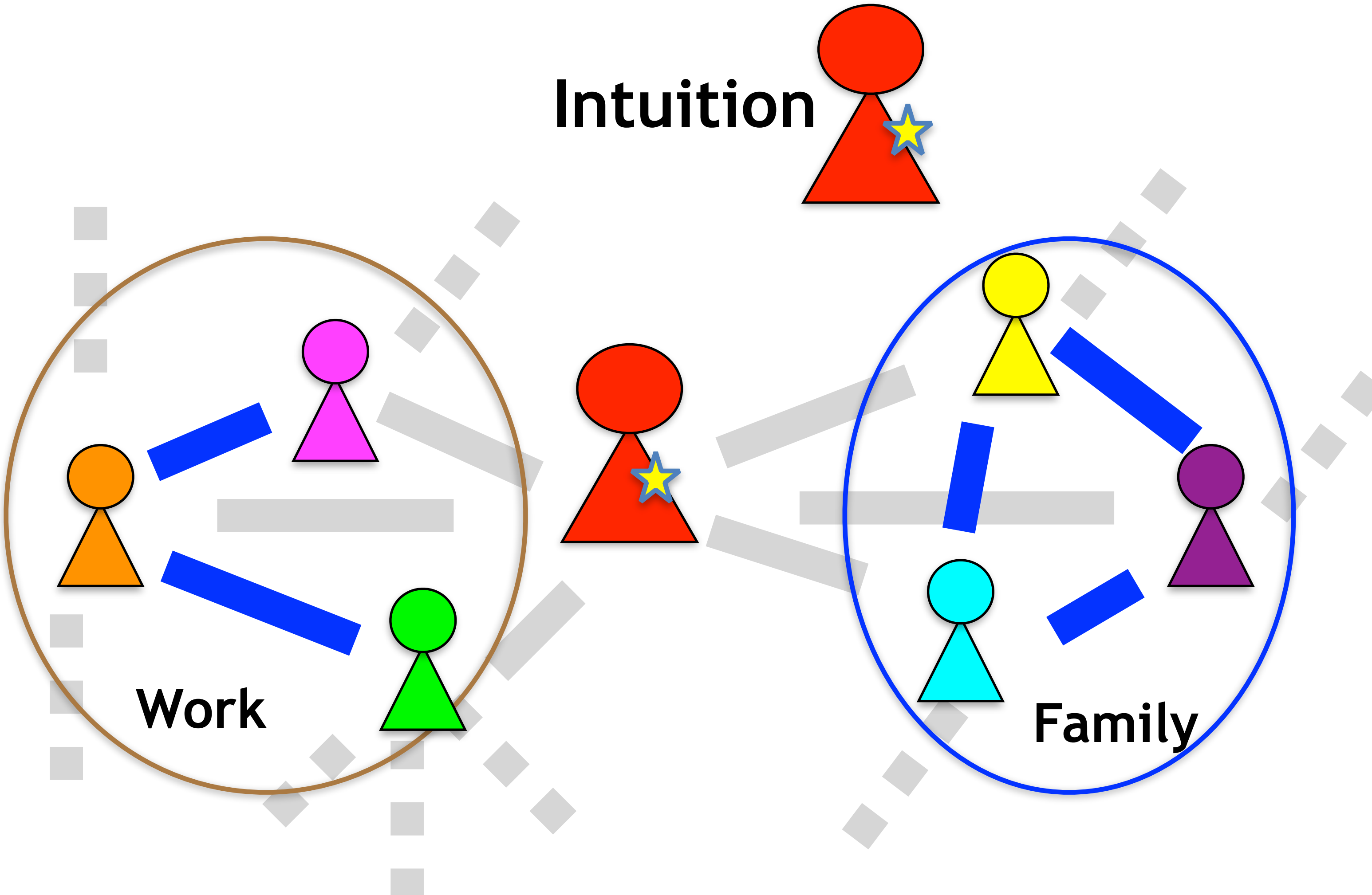
Similar definition for directed graphs.

Ego-net <u>minus ego</u> of



The **Egonet minus Ego** of node u, is defined as the induced subgraph on {N(u)}.

Similar definition for directed graphs.



Intuition: while communities overlap, usually there is a **single context** in which **two** neighbors interact.

This motivates the study of ego-networks for community detection.

Related Work

Ego-net based **community detection** has recent but rich literature:

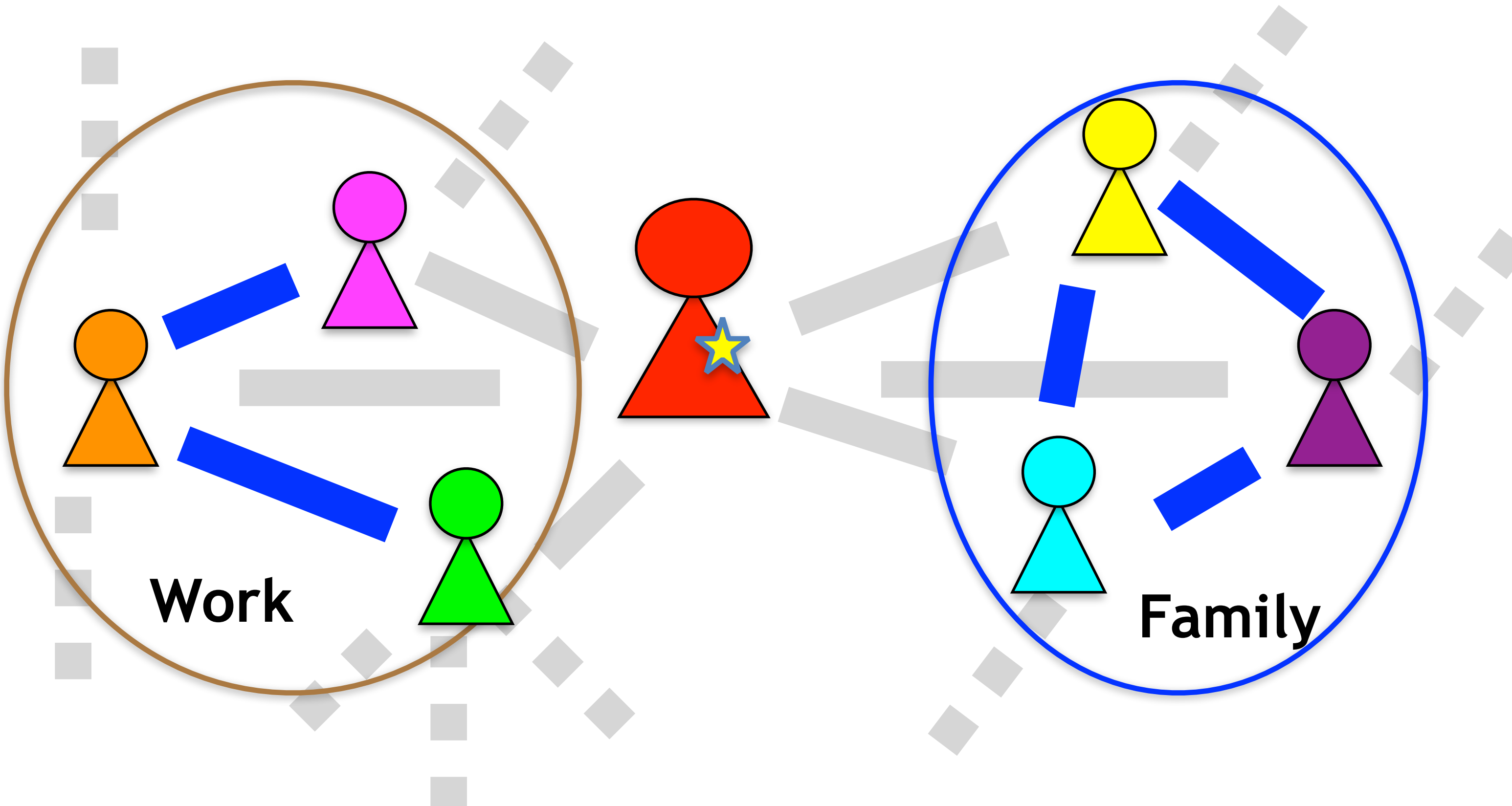
- [Freeman 1982] Definition of ego-net.
- [Rees and Gallagher, 2010]. Connected Components in Ego-Nets as communities.
- [Coscia et al. 2014], DEMON algorithm. Many follow-ups.
- Machine learning based circle detection algorithms (McAuley and Leskovec, 2012).
- [Epasto et al. 2016], Ego-net based friend suggestion.

Our Contribution

We introduce Ego-Splitting a novel distributed overlapping clustering method:

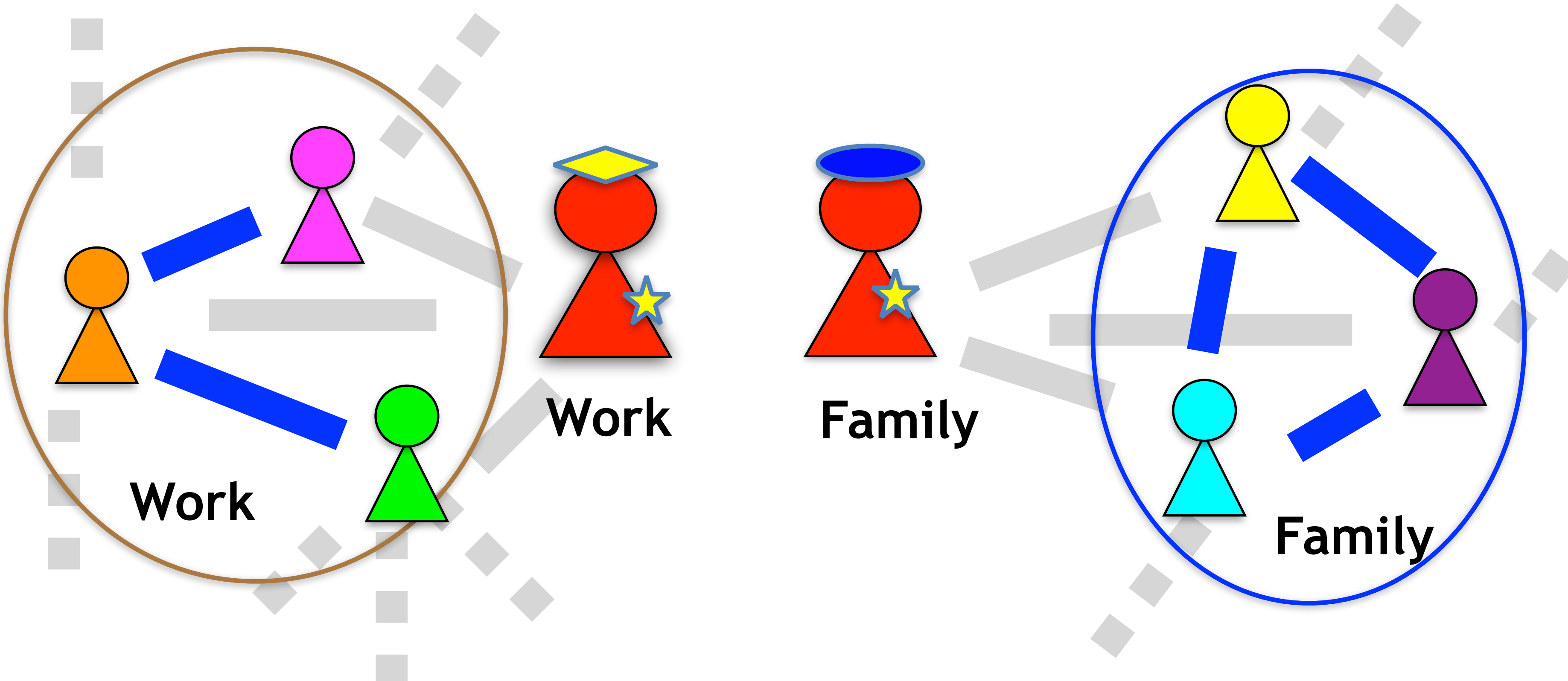
- **Highly flexible:** turns any **non-overlapping** algorithm into an **overlapping** algorithm.
- Scalable (tens of billions of nodes and edges).
- Provable theoretical guarantees.
- Based on a **novel graph-theoretic concept** of the **Persona Graph** with potential other applications.

Persona Graph Intuition



Intuition: the red node is *actually* **two** nodes which we call persona nodes.

Persona Graph Intuition



We create a *Persona Graph* where these two nodes are separated and we split the edges of the original node among the persona nodes.

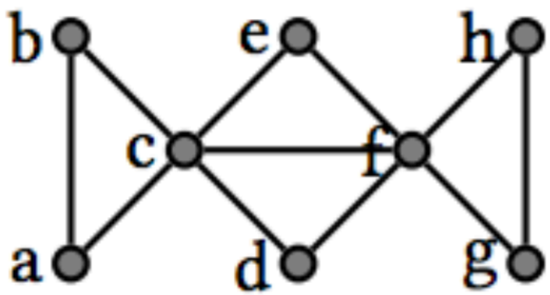
The Ego-Splitting Framework

More formally the Ego-Splitting proceeds in the following steps:

- Create the ego-net of each node.
- Partition each ego-net with a non-overlapping clustering **algorithm A1**
- **Create** the **Persona Graph**.
- **Partition** the **Persona Graph** with a non-overlapping clustering **algorithm A2**.
- Obtain the **overlapping clusters** of the original graph.

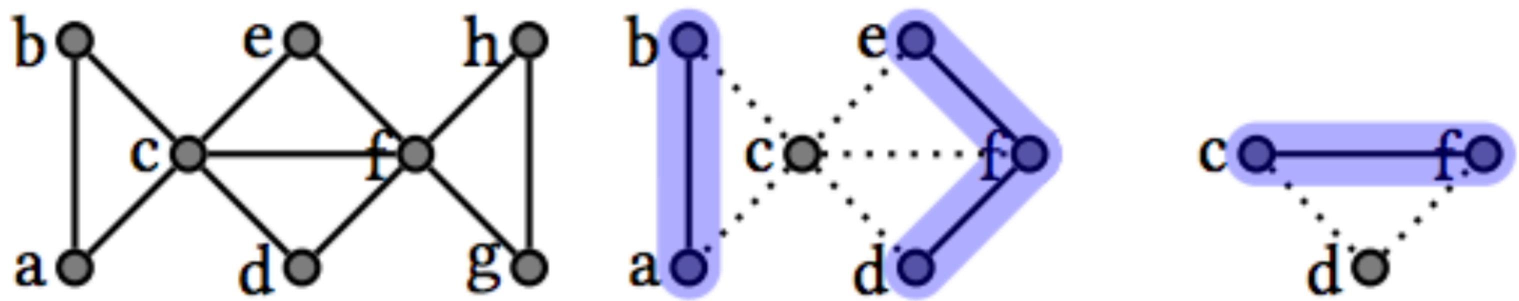
The two algorithms A1 and A2 can be arbitrary (and different).

Persona Graph - Example Construction



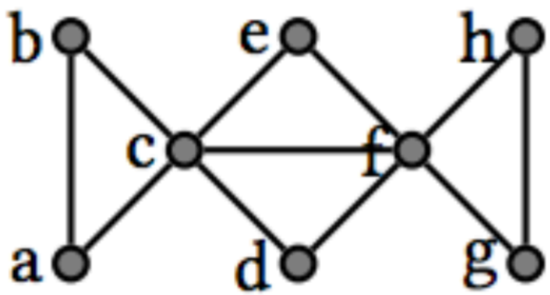
(a) original graph G

Persona Graph - Example Construction

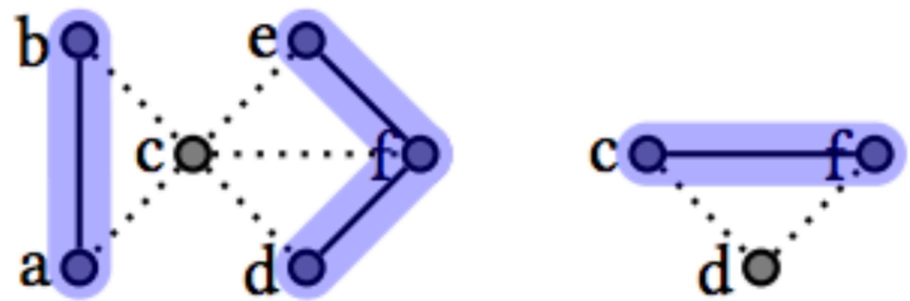


(a) original graph G (b) clustering the ego-nets

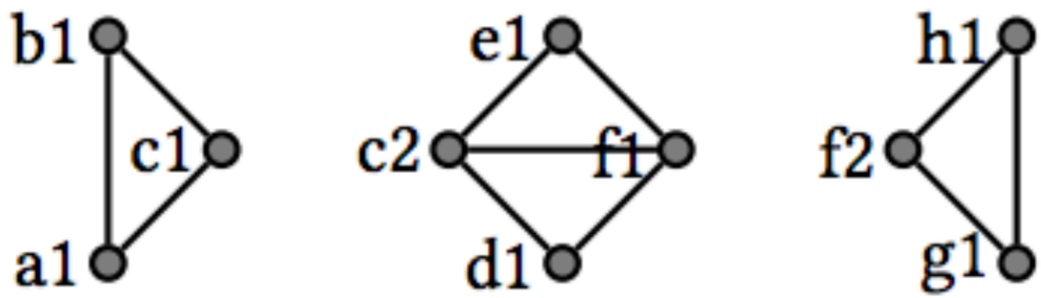
Persona Graph - Example Construction



(a) original graph G



(b) clustering the ego-nets



(c) splitting the ego we obtain the persona graph

Notice that the Persona Graph has the **same number of edges.**

Persona Graph Formal Definition

- *Step 1:* For each node u we use the local clustering algorithm to partition the ego-net of u. Let $\mathcal{A}^\ell(G[N_u]) = \{N_u^1, N_u^2, \ldots, N_u^{t_u}\}$ where $t_u = \mathrm{np}(\mathcal{A}^\ell, G[N_u])$.
- *Step 2:* Create a set V' of personas. Each node u in V will correspond to t_u personas in V' denoted by u_i for $i = 1, \ldots, t_u$.
- *Step 3:* Add edges between personas. If $(u, v) \in E$, $v \in N_u^i$ and $u \in N_v^j$ then add an edge (u_i, v_j) to E'.
- *Step 4:* Apply the global clustering algorithm $\mathcal{A}^g$ to $G' = (V', E')$ and obtain a partition $\mathcal{S}''$ of V'.
- *Step 5:* For set $C' \in \mathcal{S}''$ in the partition of V' associate a cluster $C(C') \subseteq V$ formed by the corresponding nodes of V, i.e., $C(C') = \{u \in V | \exists i \text{ s.t. } u_i \in C'\}$. Output $\mathcal{S}' = \{C(C') | C' \in \mathcal{S}''\}$.

Efficient Parallel Ego-Net Construction And Clustering

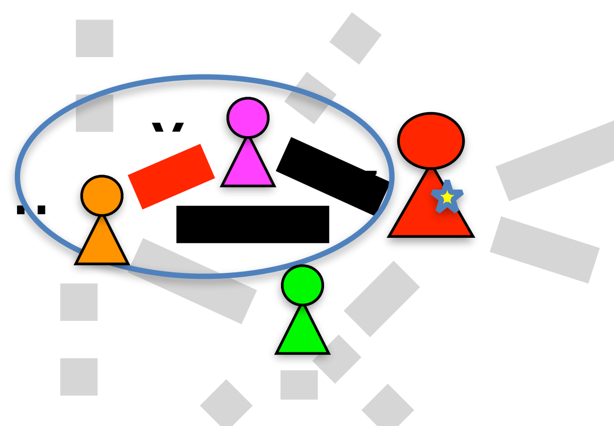
Naive approach **O(n^3) just for ego-net construction.**

[*Epasto et al. VLDB 2016]* In 2 M/R steps it is possible to construct and apply any clustering algorithm efficiently on all ego-net with small running time.

LEMMA 2. *Then the total amount of parallel work to compute the ego-nets and to run the clustering algorithm $\mathcal{A}$ on them is $O(\sqrt{m}t(m) + m^{3/2})$ and the algorithm executes only 2 MapReduce iterations.*

Intuition:

The **edge u-v** is part of ego-net of **z** iff **u-v-z** is a **triangle!**



Efficient persona graph creation and clustering

Based on similar techniques we can show that **4+R rounds** of **M/R** are sufficient to create and cluster the Person Graph with **total work** of

$$O(m^{3/2} + \sqrt{m}T_\ell(m) + T_g(m))$$

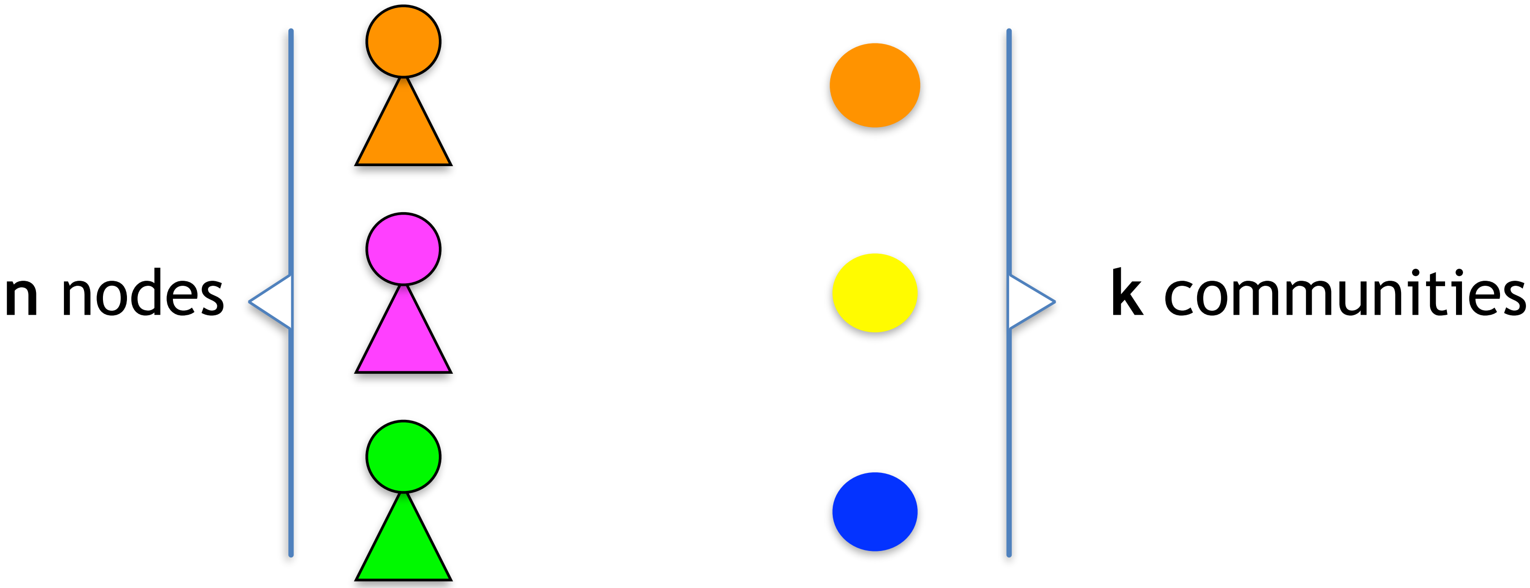
R rounds for the global clustering algorithm, Tl and Tg are the time of the local and global clustering algorithm.

Theoretical Guarantees

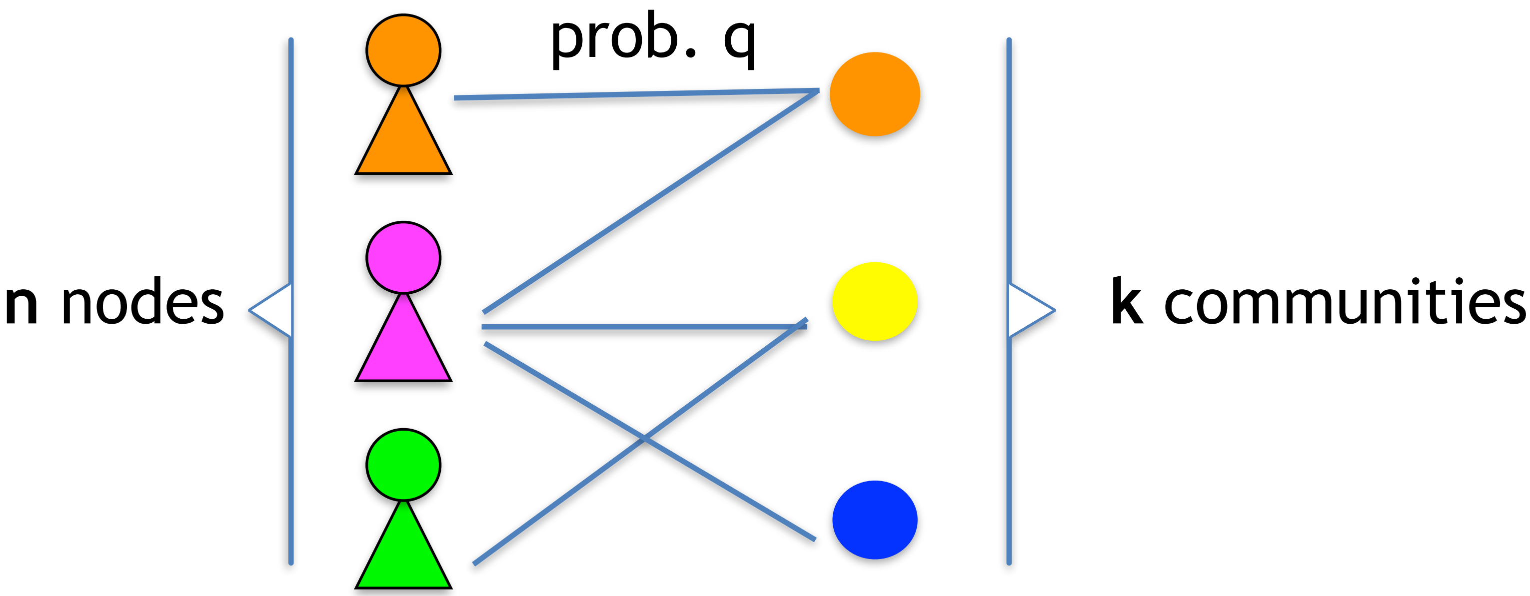
We study our Ego-Splitting framework in a simple planted overlapping clusters theoretical model.

We obtain a graph from the a probabilistic model and learn the original communities.

Probabilistic Model

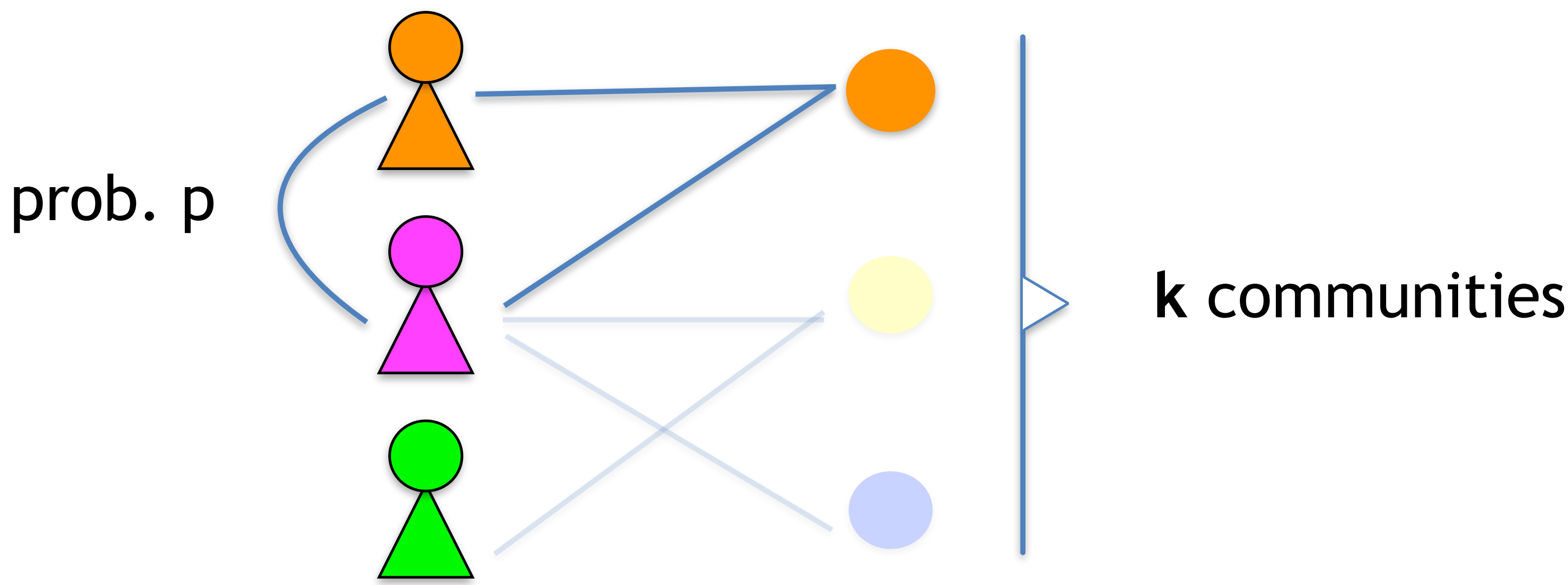


Probabilistic Model



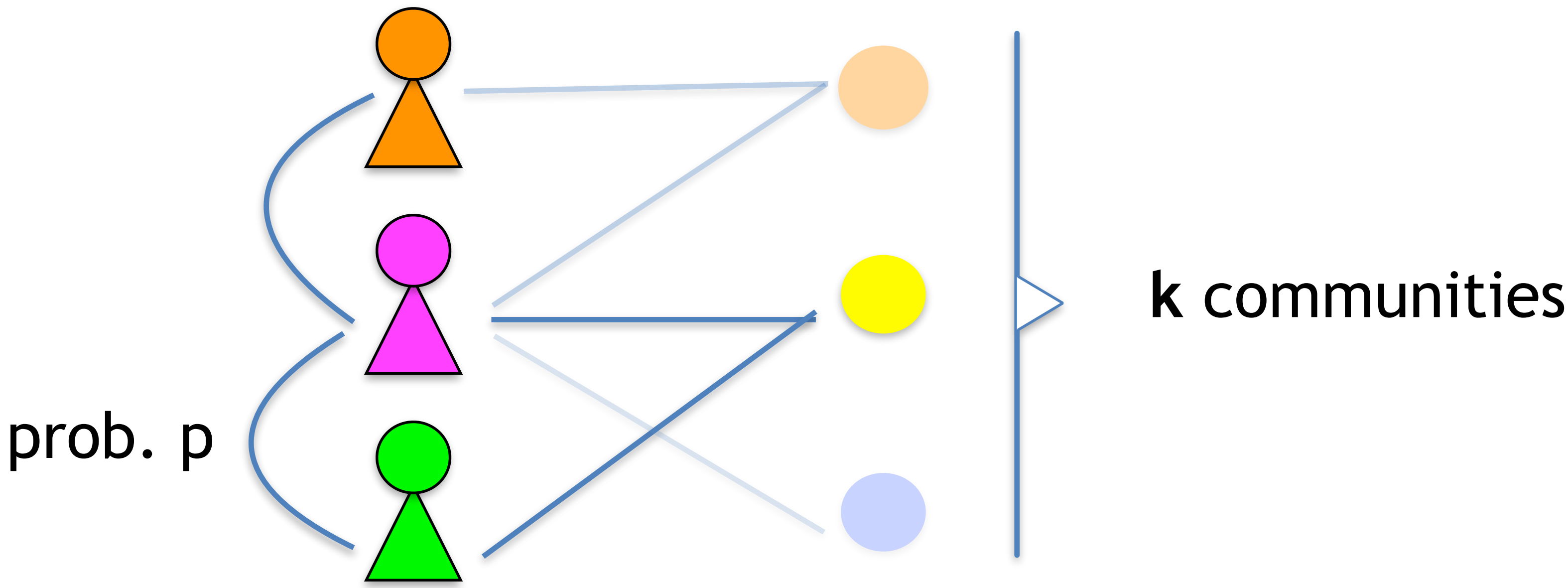
For each node-community pair draw an edges with prob. q

Probabilistic Model



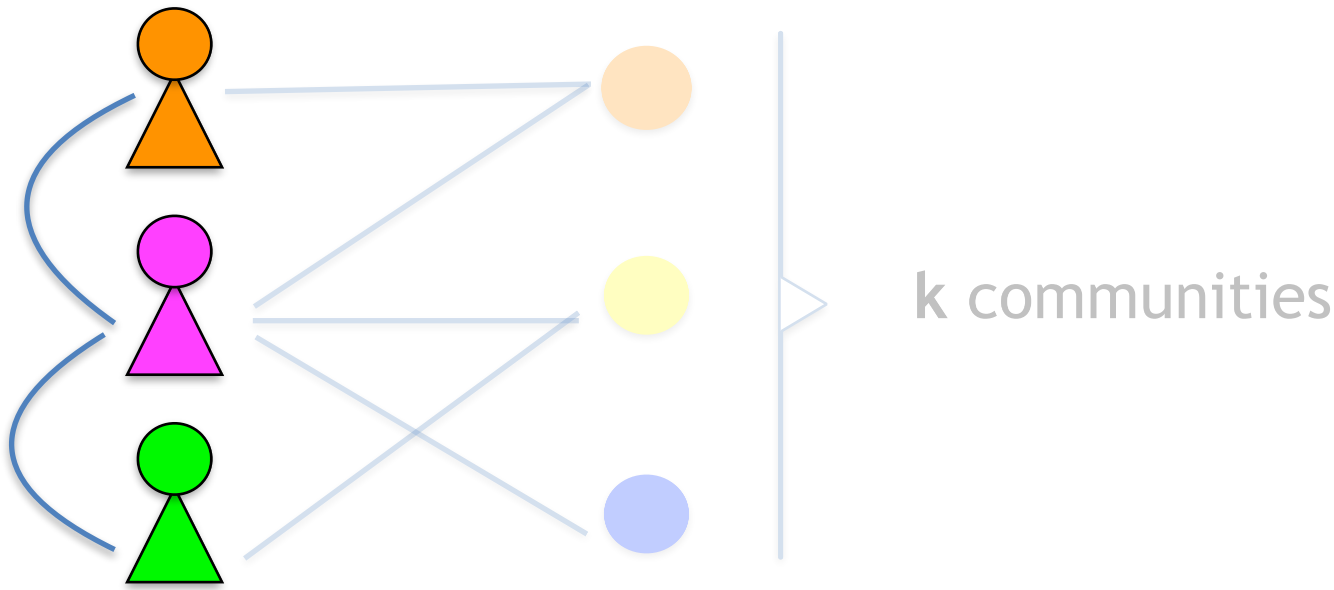
For each community c, and for each pair of nodes u,v in the community draw an edges with prob. **p** between u and v.

Probabilistic Model



This is equivalent to creating a Gn,p over each community and taking the union of the edges.

Community Reconstruction Problem



Given the graph among the nodes, reconstruct the overlapping communities.

Theoretical Guarantees

Given a P(n,k,q,p) graph we achieve perfect reconstruction (in the limit) for certain ranges of k,q and p using the simple **connected component algorithm** for the clustering.

THEOREM 5.1. *If $\mathcal{S}$ and G are sampled from a $\mathcal{P}(n,k,q,p)$ with $kq \geq 1$ and $p \geq c' \log(npq/2)/(npq/2)$, then:*

$$\mathbb{E}[J(\mathcal{S},\mathcal{S}')] \geq 1 - nk\exp(-\Omega(np^2q)) - O\left(n^3k^2p^2q^6\right)$$

Concrete settings:

- $k = n,\ q = c\log(n)/n \quad p = O(1).$

  $\epsilon < \frac{1}{6}$ constant, let $k = n,\ q = n^{\epsilon}/n$ and $p = 1/n^{\epsilon/4}$.

Proof Sketch

First we prove that each community is connected with high probability also at the level of ego-net of each member.

LEMMA 5.5. *If $p \geq 6\log(npq/2)/(npq/2)$, then with at least*

$$1 - nk\exp\left(-\Omega(np^2q)\right)$$

probability, the graph $G[C]$ is connected for all $C \in \mathcal{S}$ and $G[N_u^C]$ is connected for all $u \in C \in \mathcal{S}$.

Proof Sketch

Second we prove that if the algorithms makes no mistake at the local clustering stage the community is identified.

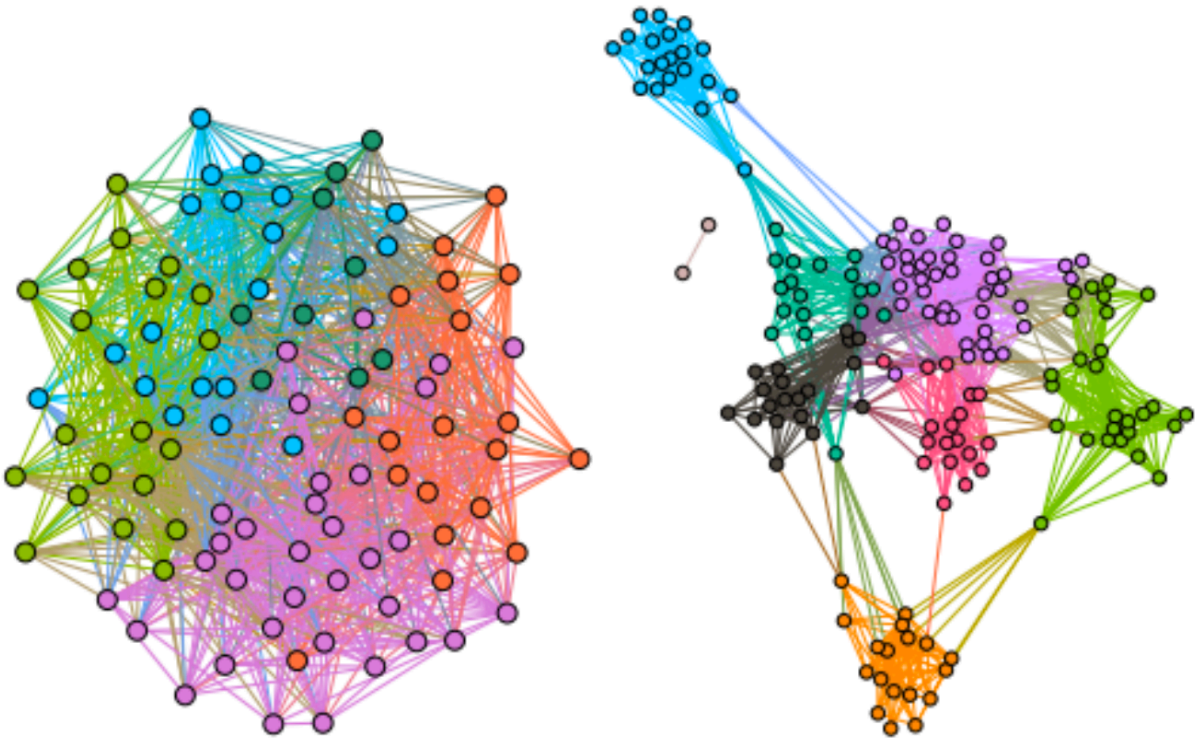
LEMMA 5.6. *Fix a cluster C, if for for all $u \in C$ the following conditions hold:*

1. *the induced graph $G[C]$ is connected.*
2. *the induced graph $G[N_u^C]$ is connected.*
3. *there are no edges between N_u^C and $N_u - N_u^C$.*

then ego-splitting with connected component reconstructs cluster C exactly.

Finally we show that the mistakes happen in limited number.

Example of Persona Graph



(a) Original Graph (b) Persona Graph

100 nodes
9 overlapping communities

The persona graph is visibly **easier** to **cluster** with non-overlapping algorithms.

Original modularity: 0.25, Persona modularity: 0.6

Empirical Evaluation

We used both real-world graphs with up to a tens of billion edges and synthetic graphs with overlapping clusters from a standard benchmark.

We evaluated our results on the ground truth clusters using the F1 score and NMI score as in previous work [Coscia et al., 2014].

We compare with the following two other approaches:

- **DEMON:** Coscia et al 2014.
- **OLP:** off-the-shelf overlapping label propagation.
- Non overlapping clustering algorithms (not reported).

Results on Synthetic Graphs

Table 2: Accuracy in synthetic benchmarks

	Ego-splitting		DEMON		OLP	
Graph	F1	NMI	F1	NMI	F1	NMI
Benchmark-0.01	**0.9368**	**0.9403**	0.4765	0.1670	0.6254	0.3149
Benchmark-0.1	**0.7878**	**0.7100**	0.1200	0.0000	0.7723	0.5571
Benchmark-0.3	**0.6714**	**0.5076**	0.1216	0.0000	0.6151	0.4405

Our method outperforms all the ones evaluated in F1 and NMI score.

Results on Real-World Graphs

Table 3: Accuracy in real-world graphs

	Ego-splitting		DEMON		OLP	
Graph	F1	NMI	F1	NMI	F1	NMI
amazon	**0.0374**	**0.0809**	0.0337	0.0310	0.0339	0.0450
dblp	**0.1662**	**0.1041**	0.1539	0.0309	0.1448	0.0645
livejournal	**0.0490**	**0.0394**	-	-	0.0115	0.0148
orkut	**0.0332**	0.0060	-	-	0.0267	**0.0129**
friendster	**0.0051**	**0.0008**	-	-	0.0010	0.0006

Our method outperforms almost all the ones evaluated in F1 and NMI score. Graphs from SNAP library.

Scalability

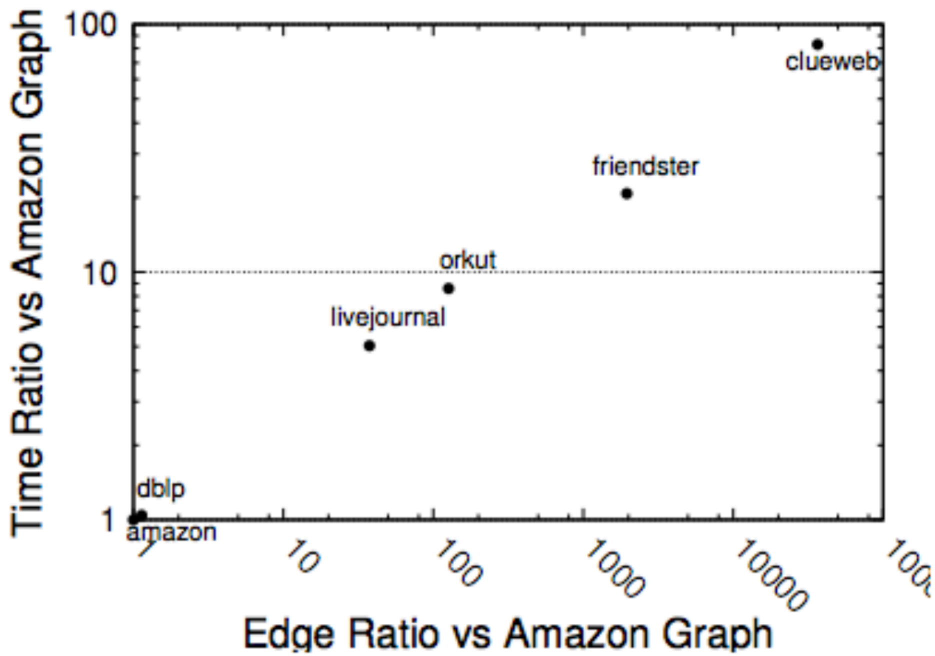


Figure 6: Running time vs size of the graph

Ratio of **wall-clock time** w.r.t smallest graph.

Our method scales to graphs with billions of nodes and edges.

Conclusions and Future Work

It is possible to construct overlapping clusters at scale with provable theoretical guarantees.

- **Future work:**
  - Other models of computation (dynamic, streaming).
  - Explore the Persona Graph.

Thank you for your attention

Contact:
aepasto@google.com
www.epasto.org

Google NYC Algorithms and Optimization team:
research.google.com/teams/nycalg/