

Local Graph Clustering

Kimon Fountoulakis
Computer Science, University of Waterloo

Outline

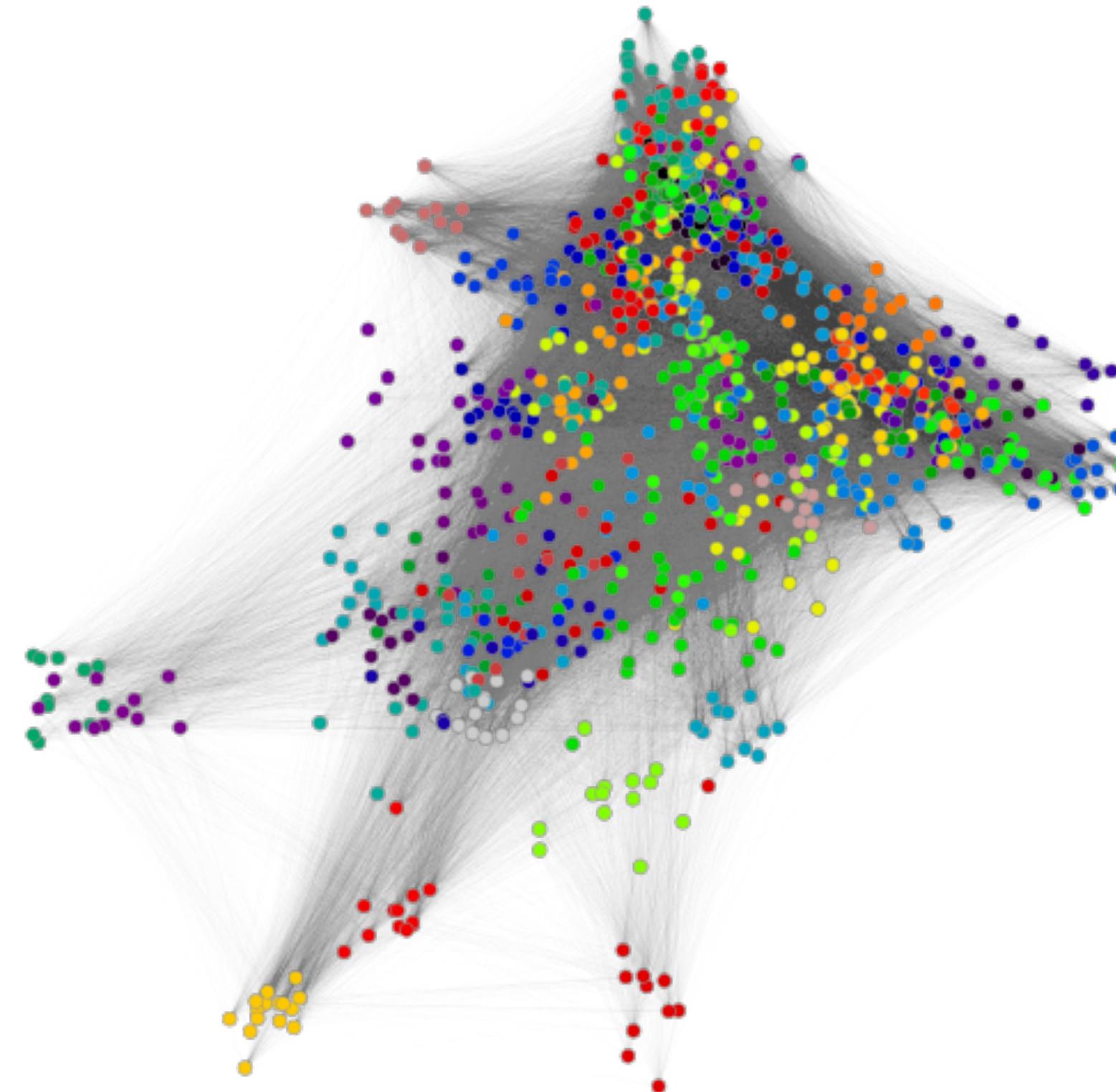
1. Local graph clustering, definition and examples
2. L1-regularized PageRank for finding clusters
3. Proximal Gradient Descent
4. Average-case recovery guarantees

What is local graph clustering and why is it useful?

- Definition: find set of nodes A given a seed node in set B
- Set A has good precision/recall w.r.t set B
- The running time depends on A instead of the whole graph
- Scalable to graphs with billions of nodes and edges
- Ideal for finding small clusters and small neighborhoods

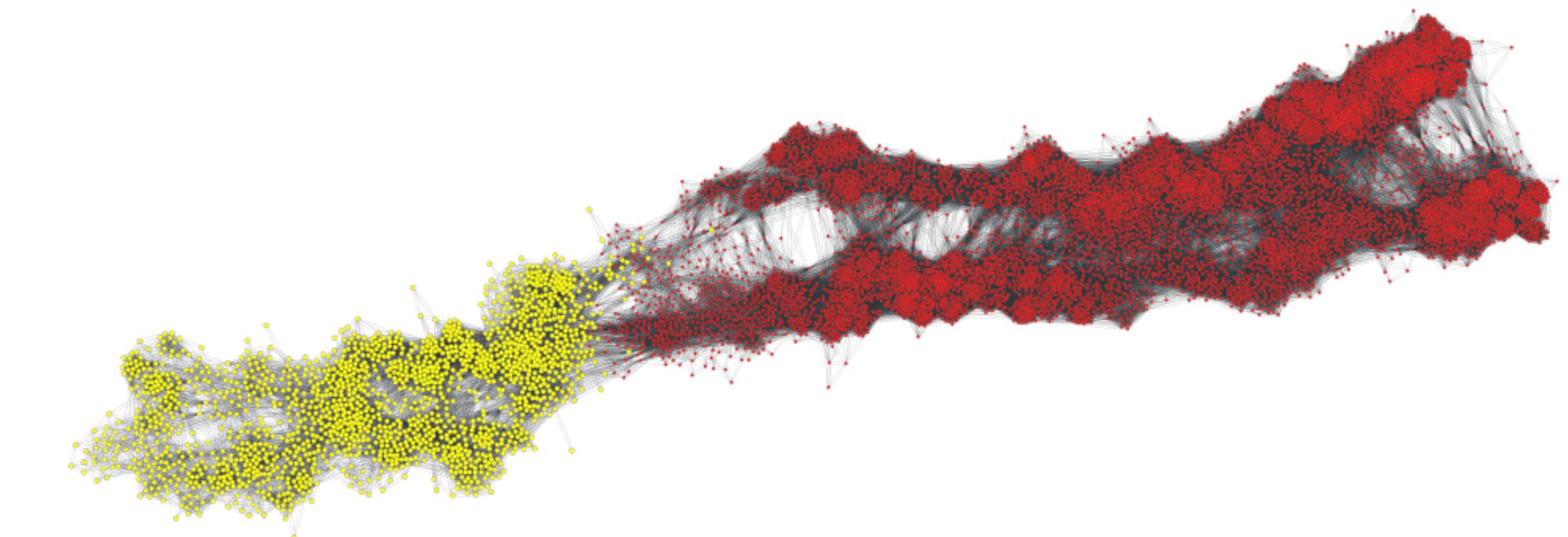
Motivation

- Real large-scale graphs have rich local structure
- We often have to deal with graphs with this structure



protein-protein interaction graph,
color denotes similar functionality

Rather than this nice Euclidean structure

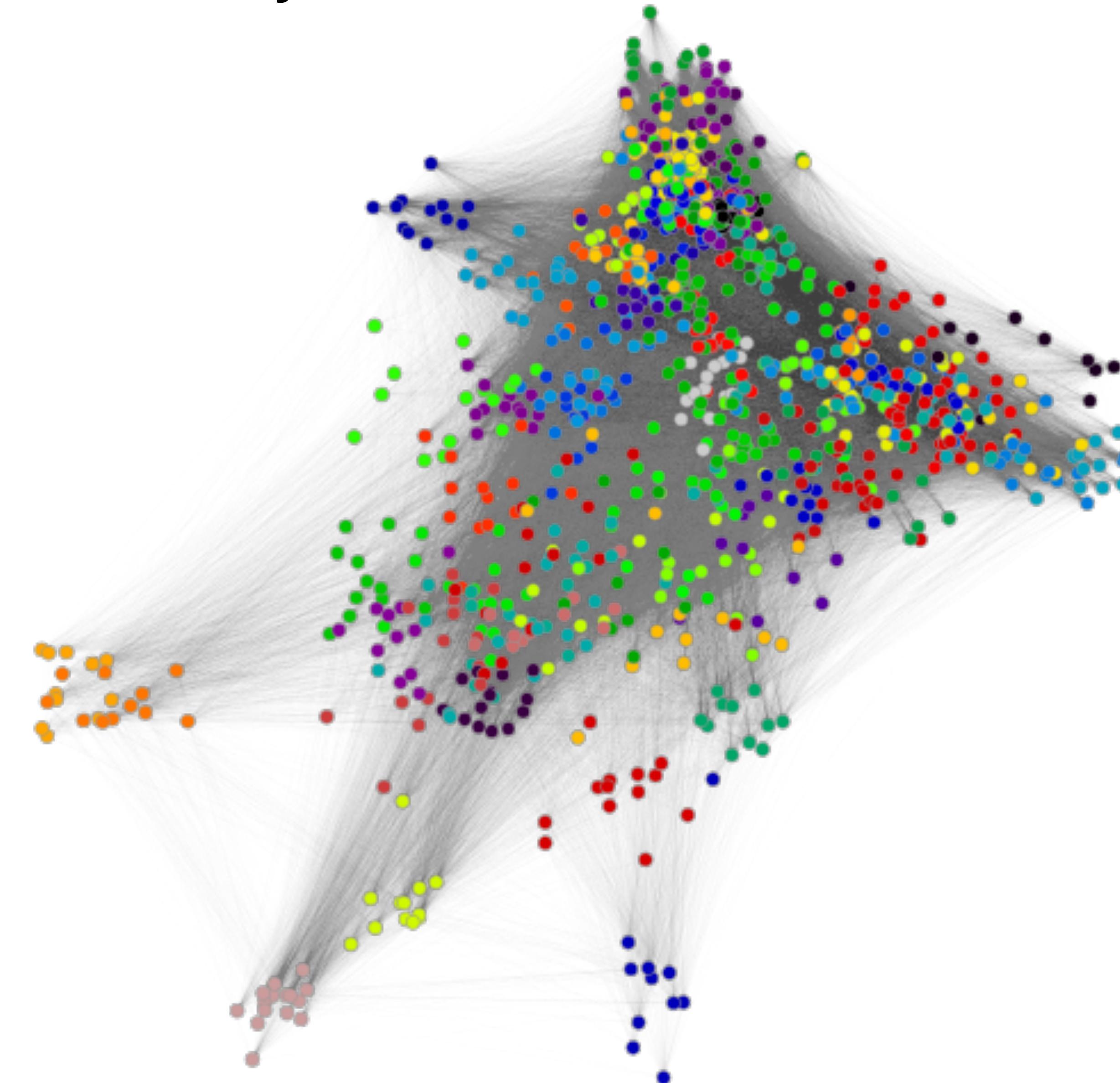


US-Senate graph,
nice bi-partition in year 1865 at the end of
the American civil ward

Our goal

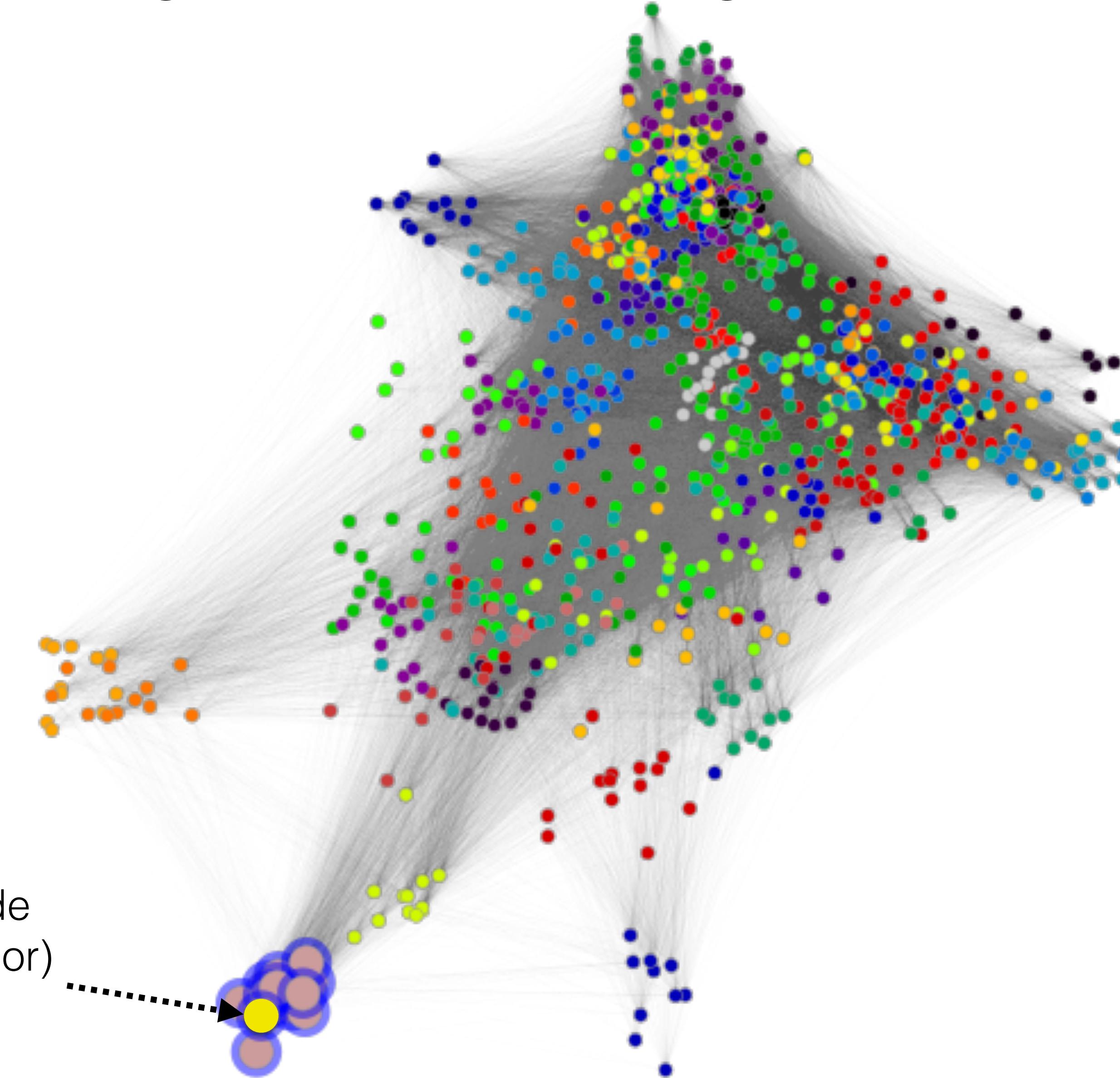
- We need methods that are able probe graphs with billions of nodes and edges.
- The running time of the new methods should depend on the size of the output instead of the size of the whole graph.
- The new methods should be supported by worst- and average-case theoretical guarantees.

Protein structure similarity: color denotes similar function



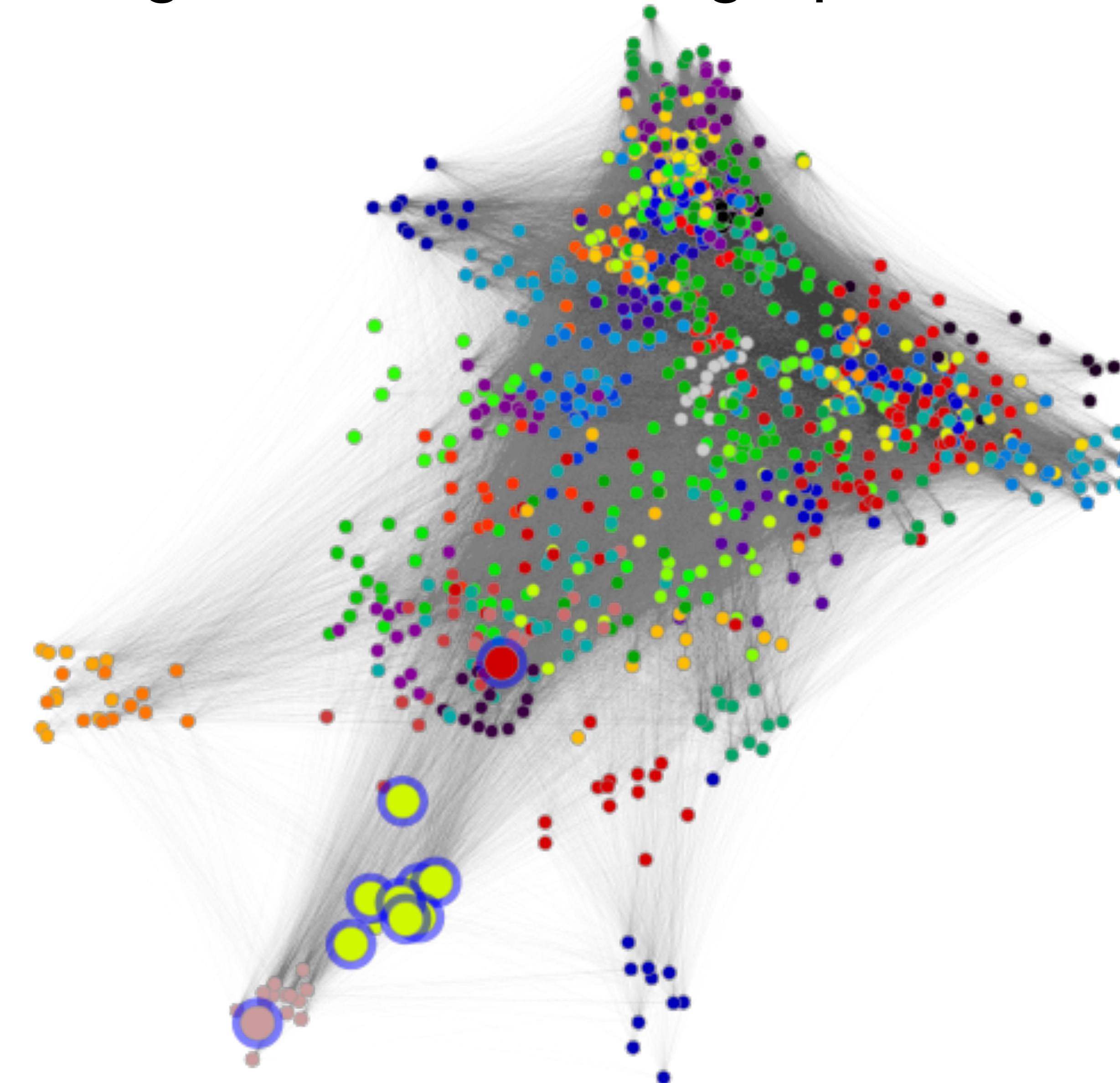
Data: The MIPS mammalian protein-protein interaction database. *Bioinformatics*, 21(6):832-834, 2005

Local graph clustering finds 2% of the graph



Data: The MIPS mammalian protein-protein interaction database. *Bioinformatics*, 21(6):832-834, 2005

Local graph clustering finds 1% of the graph



Data: The MIPS mammalian protein-protein interaction database. *Bioinformatics*, 21(6):832-834, 2005

Some definitions

- Graph: $G = (\underbrace{V}_{nodes}, \underbrace{E}_{edges})$ $|V| = n, |E| = m$
- $n \times n$ adjacency matrix: A
- An element of A is equal to 1 if two nodes are connected

Some definitions

- Degree matrix: $D = \text{diag}(A\mathbf{1}_n)$
- Each element of D shows the number of neighbors of a node
- Random walk matrix: AD^{-1}
- Lazy random walk matrix: $W = \frac{1}{2}(I + AD^{-1})$
- Graph Laplacian: $L = D - A$

Personalized random walk

- Let $\alpha \in (0, 1)$
- Consider a diffusion process where we perform lazy random walk with probability $1-\alpha$, and jump to a given seed node with probability α :
$$\alpha s 1_n^T + (1 - \alpha)W$$
- where s is an indicator vector of the seed node and alpha is the teleportation parameter.
- Simple idea:** use a random walk from a seed node. The nodes with the highest probability after k steps consist a cluster.

Good old Personalized PageRank

-Personalized PageRank: stationary distribution of:

$$p = \alpha s + (1 - \alpha)Wp$$

-Equivalently

$$p = \alpha \sum_{k=0}^{\infty} (1 - \alpha)^k W^k s$$

- That is, ranking nodes according to weighted sums of landing probabilities of different length walks from the seed node.

Let's get rid off the tail

- For the stationary personalized PageRank vector most of the probability mass is concentrated around the seed node.
- This means that the ordered personalized PageRank vector has long tail for nodes far away from the seed node.
- We can efficiently cut the tale using ℓ_1 -regularized PageRank without even having to compute the long tail.

L1-regularized PageRank

$$\text{minimize } \underbrace{\frac{1}{2}x^T Q x - \alpha x^T s}_{f(x)} + \underbrace{\rho \alpha \|Dx\|_1}_{g(x)}$$

where

$$Q = \alpha D + \frac{1-\alpha}{2} L$$

- Simple check: for $\rho=0$ the optimality conditions of the L1-regularized problem give the personalized PageRank linear system.

Properties of the ℓ_1 -regularized optimal solution

- Theorem

- If the graph is unweighted then the number of nonzero nodes in the optimal solution is bounded by $1/p$.
- If the graph is weighted then the volume of nonzero nodes in the optimal solution is bounded by $1/p$.

Sketch of proof

-**Theorem:** Let \hat{x} be the optimal solution. Then the volume of the support of the solution is bounded Bounded volume of the support

$$\text{vol}(\hat{S}) = \sum_{i \in \hat{S}} d_i \leq \frac{1}{\rho} \quad \text{where} \quad \hat{S} = \{i \mid \hat{x}_i \neq 0\}$$

Result 1: Negative partial derivatives are bounded from below

$$\rho\alpha^{1/2} \leq -\nabla_i f(\hat{x}) \quad \forall i \in \hat{S} \rightarrow \text{vol}(\hat{S})\rho\alpha \leq \|D^{1/2}\nabla f(\hat{x})\|_1$$

Result 2: Gradients are bounded from above

$$\|D^{1/2}\nabla f(\hat{x})\|_1 \leq \alpha$$

Results 1 + 2 use proof-by-algorithm + strong convexity, Results 1 + 2 give the final result.

The solution path is monotonic

-Theorem

-Let $\hat{x}(\rho)$ be the solution of the ℓ_1 -regularized problem as a function of ρ .

-Then $\hat{x}(\rho)$ is a component-wise monotone function

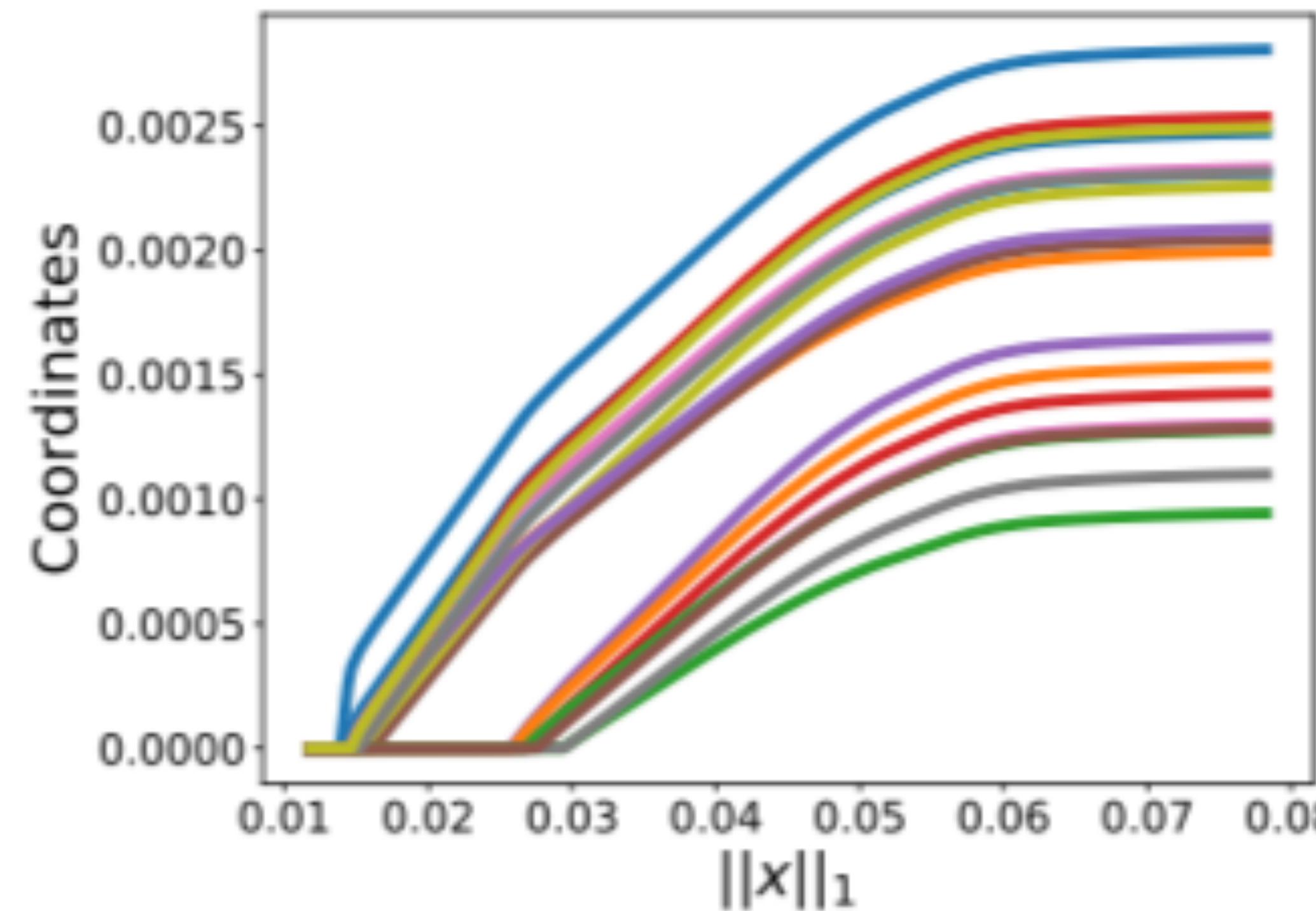
$$\hat{x}(\rho_0) \leq \hat{x}(\rho_1) \text{ for } \rho_0 > \rho_1$$

-The inequality becomes strict when a component is positive.

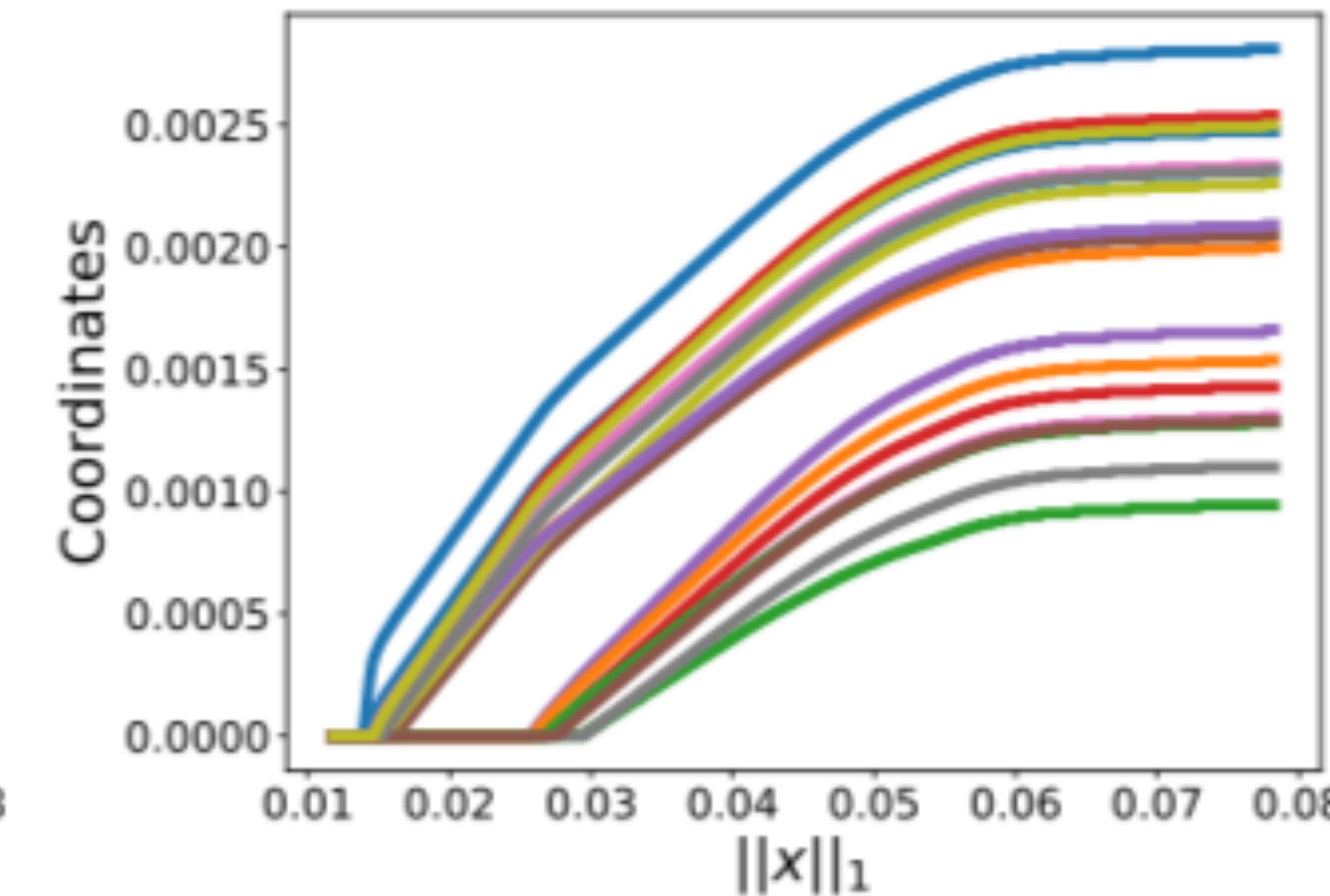
Recover the whole path with the cost of one solve (in worst-case)

-Corollary

- The stage-wise algorithm (next slide) converges to the ℓ_1 -regularized solution path if we drag the step-size of the algorithm to zero.



(a) ℓ_1 -reg. path



(b) Stage. path $\eta = 10^{-4}$

Stage-wise for recovering the whole path

-Stage-wise algorithm

- 1) Choose i such that $|d_i^{-1} \nabla_i f(x_k)|$ is the largest among $[n]$
- 2) Update $[x_{k+1}]_i = [x_k]_i + \frac{\eta}{d_i}$

-Corollary

- The stage-wise algorithm converges to the l1-regularized solution path if we drag the step-size η of the algorithm to zero.
- The running time of stage-wise depends on the nonzero nodes and its neighbors and not on the size of the whole graph.

What if we do not want to recover the whole path?

$$\text{minimize } \underbrace{\frac{1}{2}x^T Qx - \alpha x^T s}_{f(x)} + \underbrace{\rho\alpha \|Dx\|_1}_{g(x)}$$

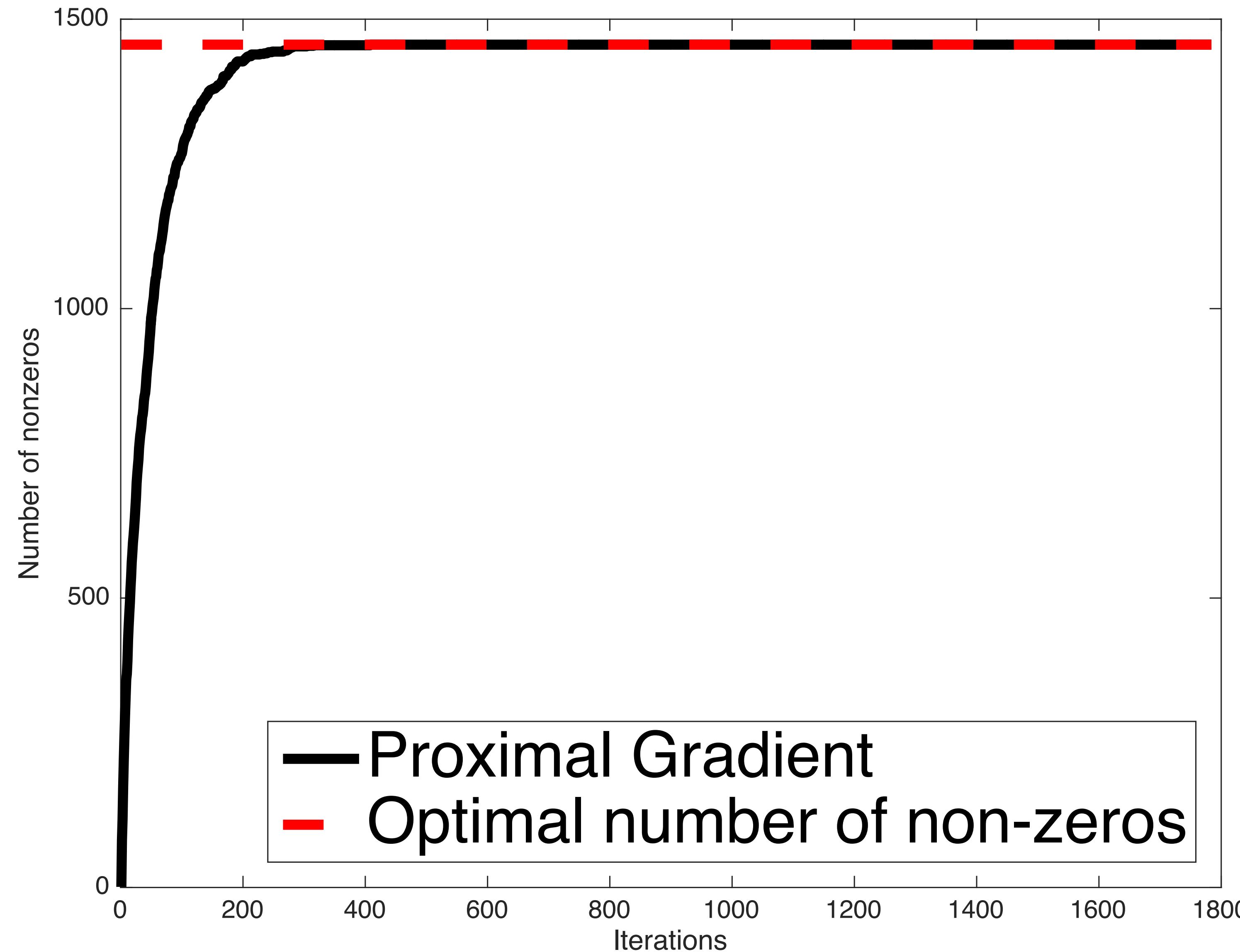
Proximal gradient descent

$$x_{k+1} := \operatorname{argmin}_x g(x) + \underbrace{f(x_k) + \langle \nabla f(x_k), x - x_k \rangle}_{\text{first-order Taylor approximation}} + \underbrace{\frac{1}{2} \|x - x_k\|_2^2}_{\text{upper bound on the approximation error}}$$

Requires careful implementation to avoid excessive running time

- Need to maintain a set of non-zero nodes
- Update x and gradient only for non-zero nodes and their neighbors at each iteration

Theorem: non-decreasing non-zero nodes



Sketch of proof

- **Theorem:** non-decreasing non-zero nodes

Result 1: Using induction we get that negative partial derivatives are bounded

$$-\nabla_i f(x_k) \geq \rho\alpha d_i^{1/2} \quad \forall i \in S_k \quad \text{and} \quad -\nabla_i f(x_k) < \rho\alpha d_i^{1/2} \quad \forall i \in [n] \setminus S_k \quad \forall k$$



Using the definition of a proximal step

Result 2: The mass of the variables is non-decreasing

$$x_k \leq x_{k+1} \quad \forall k$$

Results 1 + 2 give $S_k \subseteq S_{k+1}$

Two ways to measure performance of the ℓ_1 -regularized PageRank model

Average-case (today's talk)

- Performance under stochastic block model - recover a cluster using the output of ℓ_1 -regularized PageRank.

W. Ha, K. Fountoulakis, M. Mahoney. Statistical Guarantees of Local Graph Clustering. Under Review, 2019

Worst-case

- Use conductance to measure quality of the output. Show that the output has conductance value similar to a target cluster.

Fountoulakis et al. Variational Perspective of Local Graph Clustering, Mathematical Programming, 2017

Average-case performance

Local random model

- Given a graph G with n nodes, let K be a target cluster inside G .
- Two nodes in K are connected with probability p
- Nodes in K are connected K^c with probability q .
- The rest of edges can be drawn using **any** other model.

Expected ℓ_1 -regularized PageRank

- The optimal solution of the expected problem identifies the target cluster.

- **Theorem**

- Suppose that the seed node is selected from target cluster K . The optimal solution of

$$x^* := \operatorname{argmin} \frac{1}{2} x^T \mathbb{E}[Q] x - \alpha x^T s + \rho \alpha \|\mathbb{E}[D]x\|_1$$

- satisfies

$$\operatorname{supp}(x^*) = K$$

- as long as $\rho = \mathcal{O}\left(\frac{p}{\bar{d}^2}\right)$

- where \bar{d} is the expected degree of nodes in the target cluster.

Results for ℓ_1 -regularized PageRank for noisy data

- In practice, we do not have access to the expected graph. We are given a realization of the local random model that includes “noise”, i.e., edges from the target cluster to the rest of the graph.
- We have two results for the noisy case.
 - First result.**
 - Zero false negatives.
 - Bounded false positives.
 - Second result.**
 - With additional assumptions on the seed nodes we can show exact recovery.

Results for l1-regularized PageRank for noisy data

- Theorem (bounded false positives)

- Suppose $p^2 k \geq \mathcal{O}(\log k)$, where k is the size of the target cluster.
- and $\rho = \mathcal{O}\left(\frac{\gamma p}{\bar{d}^2}\right)$
- where $\gamma = \frac{pk}{\bar{d}}$, i.e., the probability of staying inside the target cluster in one step.
- Then with probability $1 - 6\exp(-\mathcal{O}(p^2 k))$ the optimal solution of the realized problem has **zero false negatives** and the false positives are bounded

$$\text{vol}(FP) \leq \text{vol}(K) \left(\mathcal{O}\left(\frac{1}{\gamma^2}\right) - 1 \right)$$

Noisy results for l1-regularized PageRank

- **Theorem (exact recovery)**

- Let $q = \mathcal{O}\left(\frac{1}{n}\right)$

- Then with probability at least $1 - \mathcal{O}(e^{-k})$ there exists a good seed node such that if we use that seed node we get

$$\text{supp}(\hat{x}) = K$$

- As long as

$$d_j \geq \mathcal{O}\left(\frac{1}{\gamma p}\right) \quad \forall j \in K^c$$

Noisy results for l1-regularized PageRank

- The assumption that $q = \mathcal{O}\left(\frac{1}{n}\right)$
- implies that there are constant number of edges leaving the cluster, which sounds artificial.
- but it is not, because it also covers the case were the size of the target cluster is $k = \mathcal{O}(1)$
- This is a realistic local graph clustering setting where we attempt to recover a very small target cluster of constant size with constant number of edges leaving the cluster.

Empirical performance on real and noisy data from Facebook

ground truth	conductance ground truth	score	APPR	APPR +sweep_cut	ℓ_1 -reg. PR	ℓ_1 -reg. PR +sweep_cut
2006	0.48	Precision	0.42	0.43	0.44	0.44
		Recall	0.53	0.53	0.47	0.47
		f1_score	0.47	0.47	0.46	0.46
2007	0.41	Precision	0.43	0.43	0.47	0.47
		Recall	0.72	0.72	0.65	0.65
		f1_score	0.54	0.54	0.54	0.54
2008	0.29	Precision	0.64	0.67	0.69	0.71
		Recall	0.95	0.92	0.90	0.88
		f1_score	0.77	0.78	0.78	0.79
2009	0.11	Precision	0.89	0.96	0.90	0.95
		Recall	0.99	0.98	0.98	0.98
		f1_score	0.94	0.97	0.94	0.97

-APPR (approximate personalized PageRank) is coordinate descent solver with early termination for the personalized PageRank linear system.

-APPR is one of the best local graph clustering methods

-L1-regularized PageRank is a variational version of APPR.

Software

LocalGraphClustering on [GitHub](#) 

- Written in Python with C++ routines when required
- Demonstrations on social and bioinformatics networks
- Multiple Python notebooks with numerous examples and graph visualizations
- Video presentations
- 12 methods and pipelines

Thank you!