Community Detection via Random and Adaptive Sampling

Se-Young Yun (MSR-INRIA)

Marc Lelarge (INRIA-ENS)

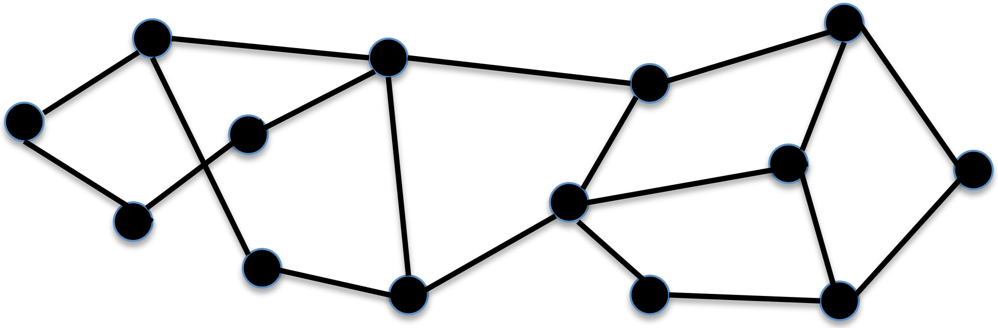
Alexandre Proutiere (KTH)

Community detection in networks

Objective: Extract *K* communities in a network of *n* nodes from random *observations.* ***K << n***. Here finite *K, n* very large

Observations

1. A graph of interactions / similarities



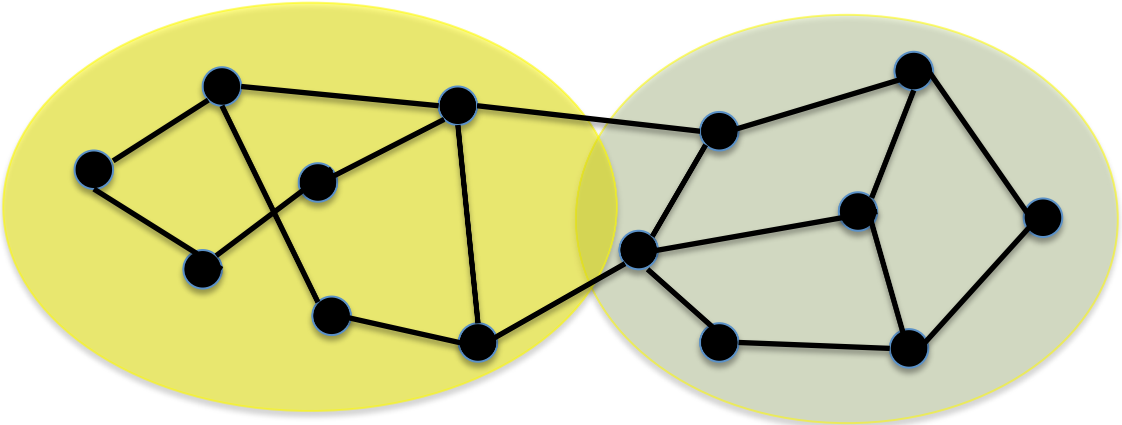
2. General sampling framework

Community detection in networks

Objective: Extract *K* communities in a network of *n* nodes from random *observations.* ***K* << *n***. Here finite *K*, *n* very large

Observations

1. A graph of interactions / similarities



2. General sampling framework

Applications

- Social networks: recommendation systems, targeted advertisement
- Biology: the role of proteins
- Distributed computing: balanced partitions
- Communication networks: caching, pro-active resource allocation (user mobility)
- …

Related work

Arora, Rao, Newman, Coja-Oghlan, Jerrum, Chen, Frieze, McSherry, Dyer, Sorkin, Kannan, Vempala, Vetta, Fortunato, Decelle, Krzakala, Karp, Condon, Reichart, Sanghavi, Nadakuditi, Girvan, Mosel, Sly, Rohe, Chatterjee, Yu, Massoulie, Lelarge, Vazirani, Karger, Feld, Fischer, Kleinberg, Gibson, Raghavan, Hopcroft, Khan, Kulis, Santo, Wellman, Hogan, Berg, White, Boorman, Kelley, Xie, Kumar, Mathieu, Schudy, Alon, Krivelevich, Sudakov, Xu, Achlioptas, Kahale, Feige, Zdeborova, Carson, Giesen, Mitshe, Shamir, Tsur, Hassibi, Oymak, Ames, Parrilo, Holland, Laskley, Pothen, Simon, Liou, Girvan, Chauhan, Leone, Ball, Karrer, …

Outline

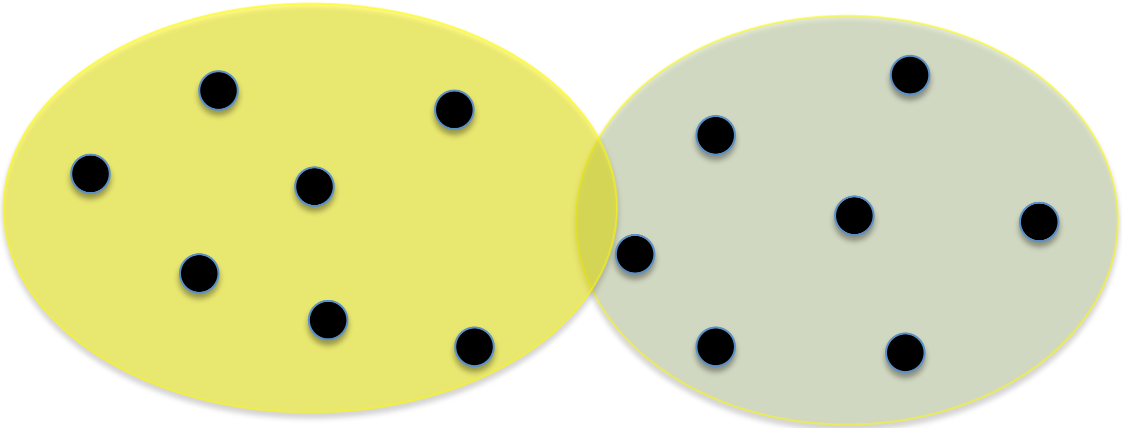
1. Community detection in the Stochastic Block Model

2. General sampling framework
    a. Fundamental limits
    b. Optimal Algorithms

3. Streaming, Memory Limited Algorithms

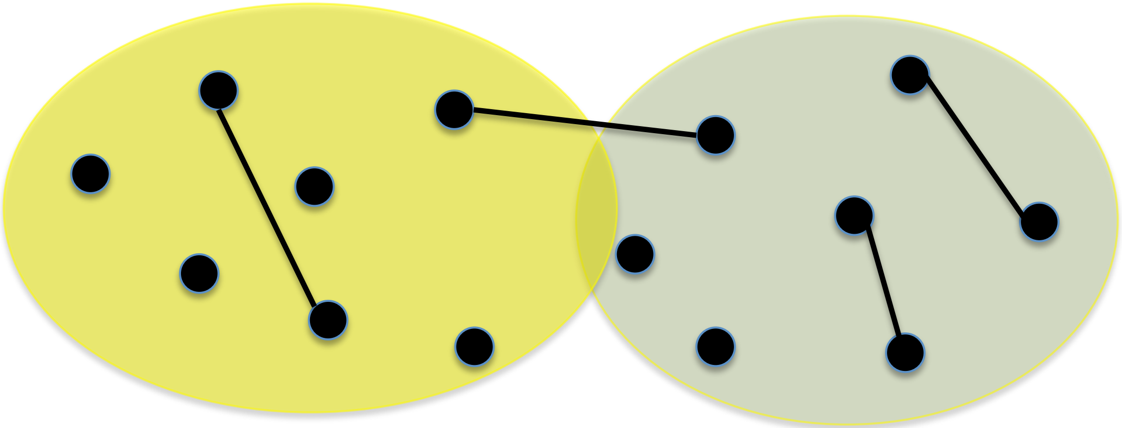
1. Community Detection in the Stochastic Block Model

Stochastic Block Model (SBM)



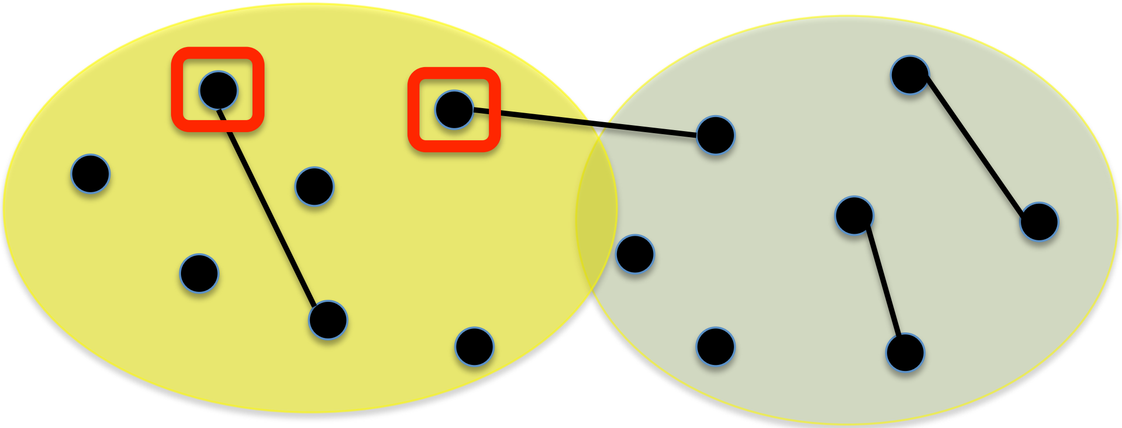
- The graph is built by considering each pair of nodes once
  - If in the same community: put an edge with probability p
  - Else: put an edge with probability $q < p$

Stochastic Block Model (SBM)



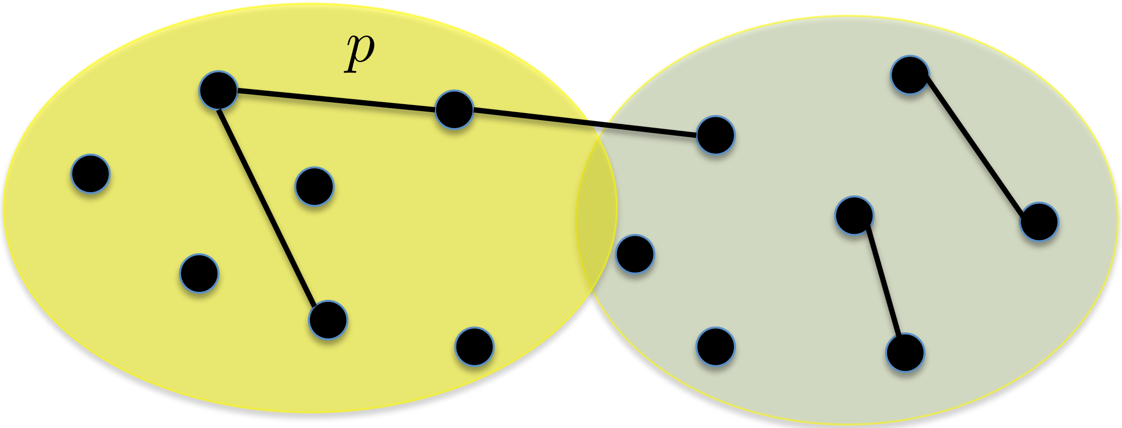
- The graph is built by considering each pair of nodes once
  - If in the same community: put an edge with probability p
  - Else: put an edge with probability $q < p$

Stochastic Block Model (SBM)



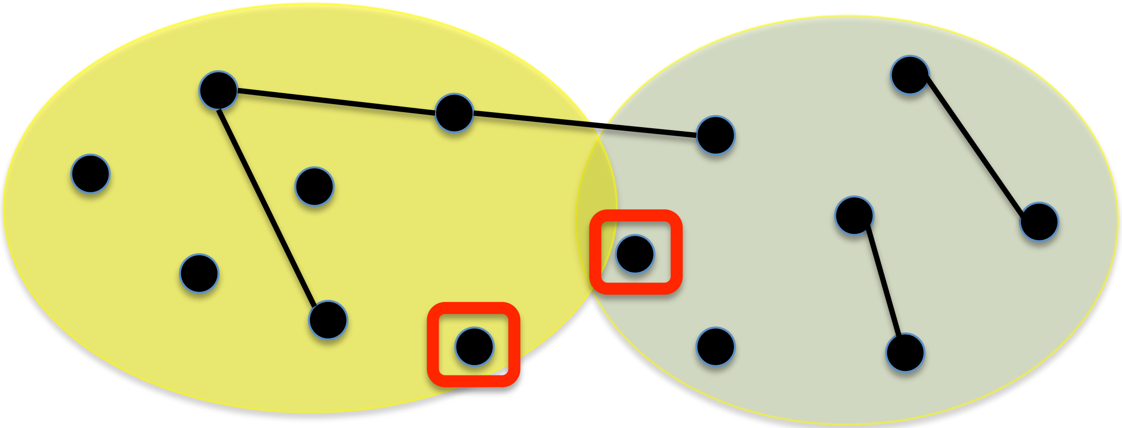
- The graph is built by considering each pair of nodes once
  - If in the same community: put an edge with probability p
  - Else: put an edge with probability $q < p$

Stochastic Block Model (SBM)



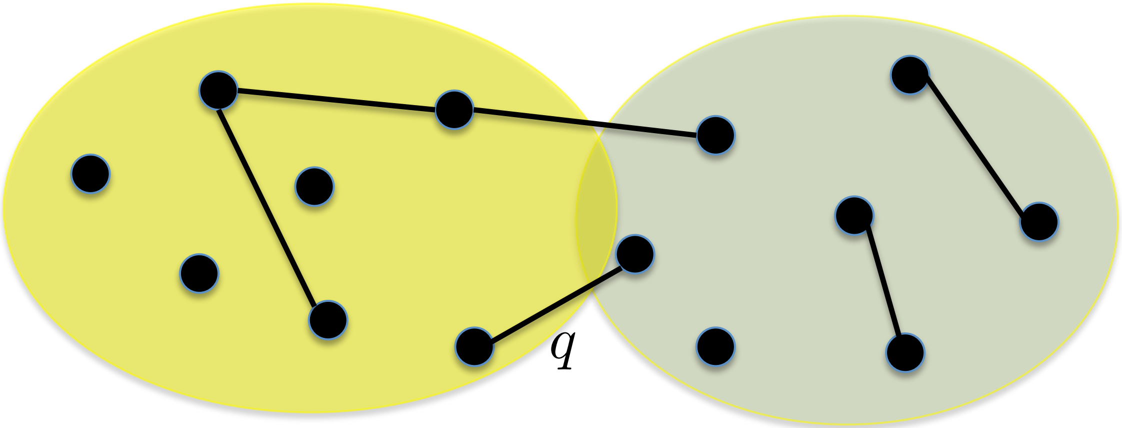
- The graph is built by considering each pair of nodes once
  - If in the same community: put an edge with probability *p*
  - Else: put an edge with probability *q* < *p*

Stochastic Block Model (SBM)



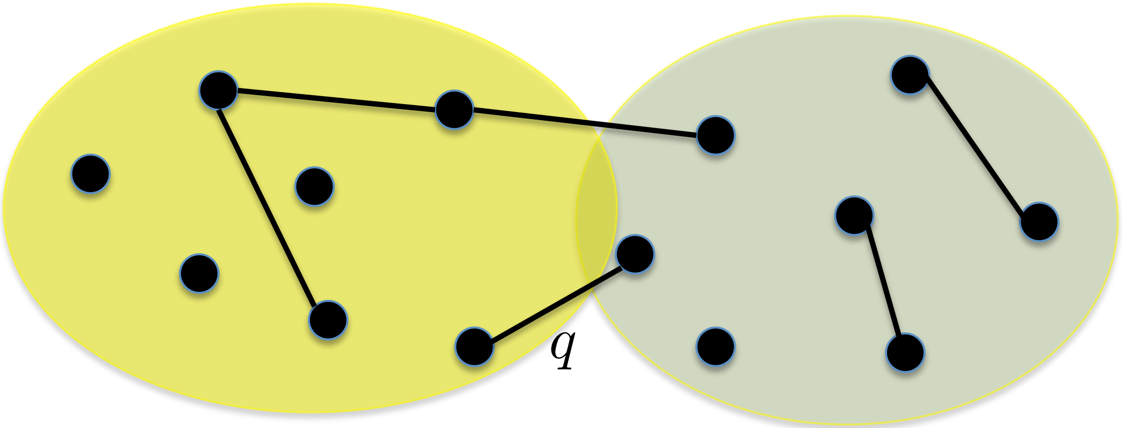
- The graph is built by considering each pair of nodes once
  - If in the same community: put an edge with probability *p*
  - Else: put an edge with probability $q < p$

Stochastic Block Model (SBM)



- The graph is built by considering each pair of nodes once
  - If in the same community: put an edge with probability *p*
  - Else: put an edge with probability *q* < *p*

Stochastic Block Model (SBM)



- Network size: *n* nodes, *n* tends to ∞
- Sparse interaction: $p, q = o(1)$
  - Very sparse $p, q \sim 1/n$ (**Massoulie**'s talk)
  - Sparse $p, q \sim f(n)/n, \; f(n) = \omega(1)$
- Dense interaction: $p, q = O(1)$

Performance metrics

Proportion of misclassified nodes under π: $\varepsilon^{\pi}(n)$

1. **Asymptotic detection**: an algorithm *detects* the clusters if it does better than the algorithm that randomly assigns nodes to clusters

2. **Accurate asymptotic detection**: an algorithm π is asymptotically accurate if $\lim_{n\to\infty} \mathbb{E}[\varepsilon^{\pi}(n)] = 0$

Asymptotic Detection in the SBM

- Two communities of equal sizes, sparse case $p = \frac{a}{n}, \quad q = \frac{b}{n}$

Theorem (Mossel-Neeman-Sly 2012)

If $a - b < \sqrt{2(a+b)}$, then asymptotic detection is impossible.

Conjectured by Decelle-Krzakala-Moore-Zdeborova 2012

Theorem (Massoulie 2013)

If $a - b > \sqrt{2(a+b)}$, then there exists an algorithm leading to clusters that are positively correlated with the true clusters.

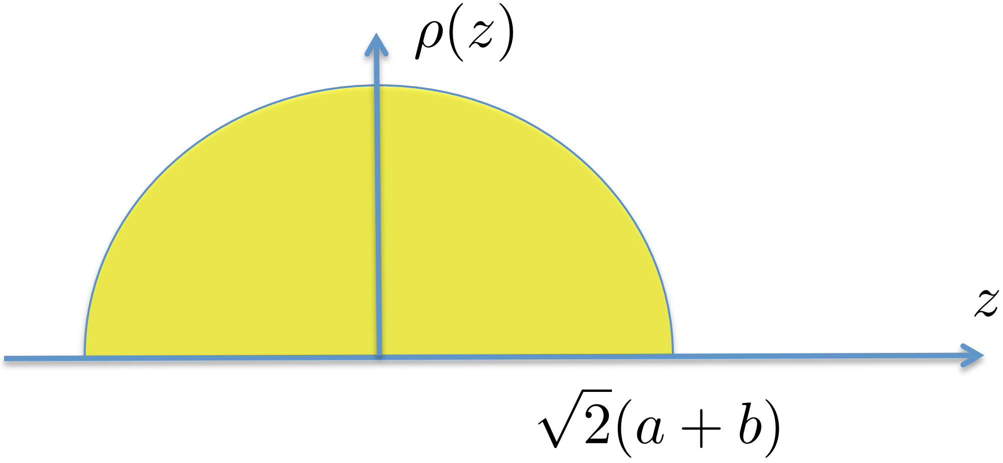
Non-rigorous Spectral Analysis

- Average adjacency matrix

$$\mathbb{E}[A] = \frac{1}{2}(a+b)11^T + \frac{1}{2}(a-b)uu^T$$

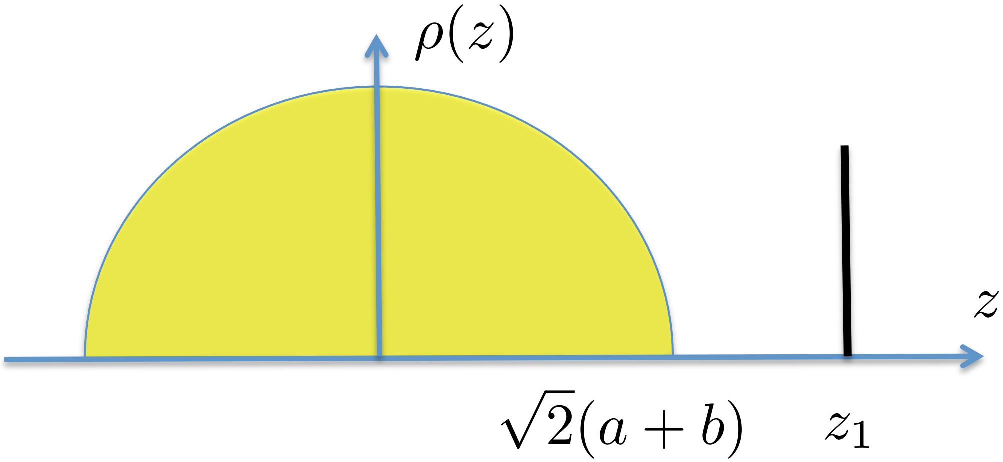
$$1 = \frac{1}{\sqrt{n}}(1,\ldots,1)^T, \quad u = \frac{1}{\sqrt{n}}(1,\ldots,1,-1,\ldots,-1)^T$$

- Noisy observation: $A = \mathbb{E}[A] + X$
- Spectral density of noise matrix X (Wigner semicircle law)



Non-rigorous Spectral Analysis

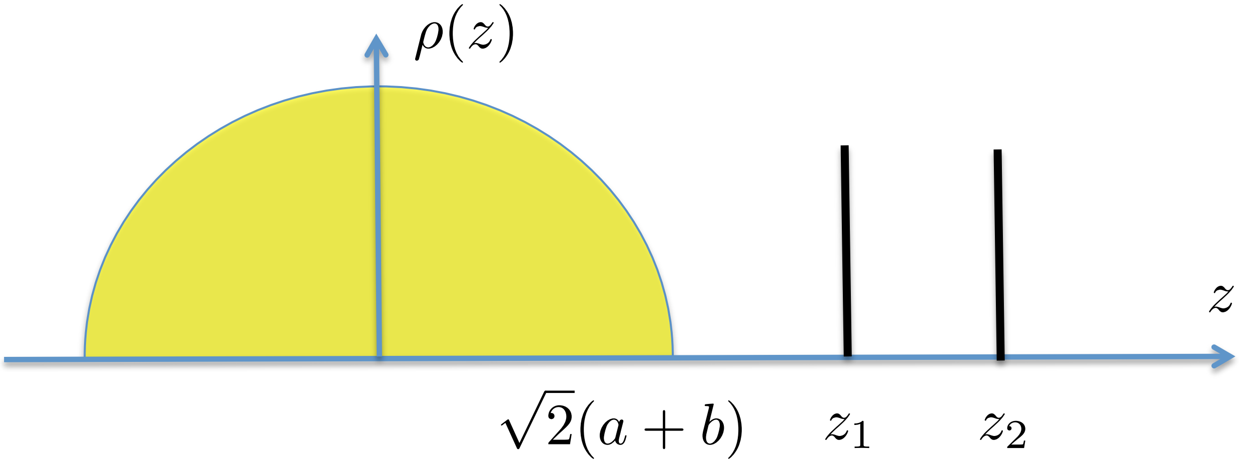
- Spectral density of the modularity matrix: $\frac{1}{2}(a-b)uu^T + X$



$$z_1 = \frac{1}{2}(a-b) + \frac{a+b}{a-b}$$

Non-rigorous Spectral Analysis

- Spectral density of the observed matrix:



- Communities are detectable if $z_1 > \sqrt{2}(a+b)$
- Method: find z_1 and the corresponding eigenvector u

Examples of algorithms

- Maximum Likelihood (NP hard problem)
  - Exact solution: Belief Propagation
  - Compressed sensing: relaxation

- Spectral method
  - Provide a rank-K approximation of the adjacency matrix

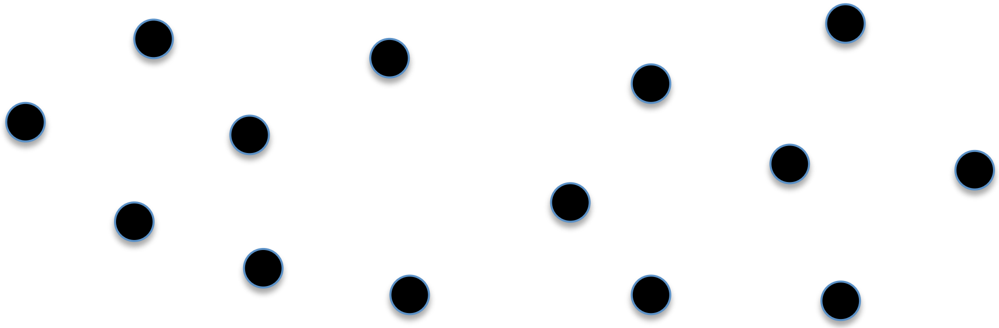
  (+ Trimming + Post-processing)

Open problems

- Very sparse graphs: condition for asymptotic detection with more than two communities
- General graphs: condition on *p, q, n* for asymptotically accurate detection? (**this talk**)
- General graphs: what is the optimal scaling of $\varepsilon^{\pi}(n)$?

2. General Sampling Framework

Sampling Framework



- Large data set available: many *samples* for the interaction of each pair of nodes
- Sample for a given node pair: Bernoulli with mean p if nodes are in the same cluster, with mean q otherwise
- Sample budget: T

Sampling Strategies

- **Non-adaptive Random Strategies**
  - The pair of nodes sampled in round *t* does not depend on past observations, and is chosen uniformly at random
  - S1: sampling with replacement
  - S2: sampling without replacement
- **Adaptive Strategies**
  - The pair of nodes sampled in round *t* depends on past observations
- Classical SBM: random sampling without replacement, and $T = n(n-1)/2$

Objectives

- Performance metric: proportion of misclassified nodes $\varepsilon(n,T)$

  Asymptotically accurate detection: $\lim_{n\to\infty} \mathbb{E}[\varepsilon(n,T)] = 0$

- Non-adaptive sampling:
  - Necessary conditions on *n, T, p, q* for the existence of asymptotically accurate algorithms
  - Asymptotically accurate clustering algorithms

- Adaptive sampling:
  - Necessary conditions *on n, T, p, q* for the existence of asymptotically accurate joint sampling and clustering algorithms
  - Asymptotically accurate sampling and clustering algorithms

Fundamental limits

- Non-adaptive sampling:

$$\kappa_1(n,T) = T\frac{2(n-2)}{n(n-1)}\min\{KL(q,p),KL(p,q)\}$$

$$+\,2\sqrt{\frac{4T(n-2)}{n(n-1)}\left[\min\{q,1-p\}\left(\log\frac{p(1-q)}{q(1-p)}\right)^2+\left(\log(\min\{\frac{p}{q},\frac{1-q}{1-p}\})\right)^2\right]}$$

Theorem Under random sampling strategy S1 or S2, for any clustering algorithm π, we have:

$$\mathbb{E}[\varepsilon^{\pi}(n,T)] \geq \frac{1}{8}\exp(-\kappa_1(n,T)),$$

Fundamental limits

- Non-adaptive sampling -- necessary conditions for asymptotically accurate detection:

$$\frac{T}{n} = \omega(1), \quad \frac{T}{n} \min(KL(q,p), KL(p,q)) = \omega(1),$$

- Dense interaction: $p, q = \Theta(1)$

$$T(p-q)^2/n = \omega(1)$$

- Sparse interaction: $p, q = o(1)$

$$T(p-q)^2/(pn) = \omega(1)$$

Fundamental limits

- Adaptive sampling:

Theorem For asymptotically accurate detection, we need:

$$\min\{p, 1-q\}\frac{T}{n} = \Omega(1) \quad \text{and} \quad \frac{T}{n}\max(KL(q,p), KL(p,q)) = \omega(1).$$

- Example: $p = \frac{\log n}{n} \qquad q = \frac{\sqrt{\log n}}{n}$
  - Non-adaptive sampling: $\frac{T}{n} = \omega(\frac{n}{\log(n)})$
  - Adaptive sampling: $\frac{T}{n} = \Omega(\frac{n}{\log(n)})$

Algorithms for non-adaptive sampling

- Spectral algorithms (extension of Coja-Oghlan's algorithm)

1. From random samples, build an observation matrix
2. Trimming (remove nodes with too many interactions)
3. Spectral decomposition (find the largest eigenvalues and corresponding eigenvectors)
4. Greedy improvement (for each node compare the number of interactions with the various clusters)

Performance

Theorem Assume that:

$$\frac{(p-q)^2}{p}\frac{\alpha T}{n} = \omega(1), \qquad \frac{(p-q)^2}{p}\frac{\alpha T}{n} \geq \log(p\frac{T}{n}).$$

Then with high probability:

$$\varepsilon^{SP}(n,T) \leq 8\exp\left(-\frac{(p-q)^2}{20p}\frac{\alpha T}{n}\right).$$

- The algorithm is asymptotically accurate under the necessary conditions for accurate detection in the case of random sampling
- The necessary conditions for accurate detection are tight!

Algorithms for adaptive sampling

- Spatial coupling idea: find reference kernels and build the clusters from these kernels

1. Kernels: select n/log(n) nodes and use T/5 samples to classify these nodes (using the previous spectral algorithm)
2. Select one of remaining nodes. Sample T/3n pairs between the selected nodes to each kernel. Classify the node.
3. Repeat 2. until no remaining node or budget

Performance

Theorem Assume that:

$$\frac{(p-q)^2}{p+q}\frac{T}{n} = \Omega(1), \qquad \frac{T}{n}\max(KL(q,p), KL(p,q)) = \omega(1).$$

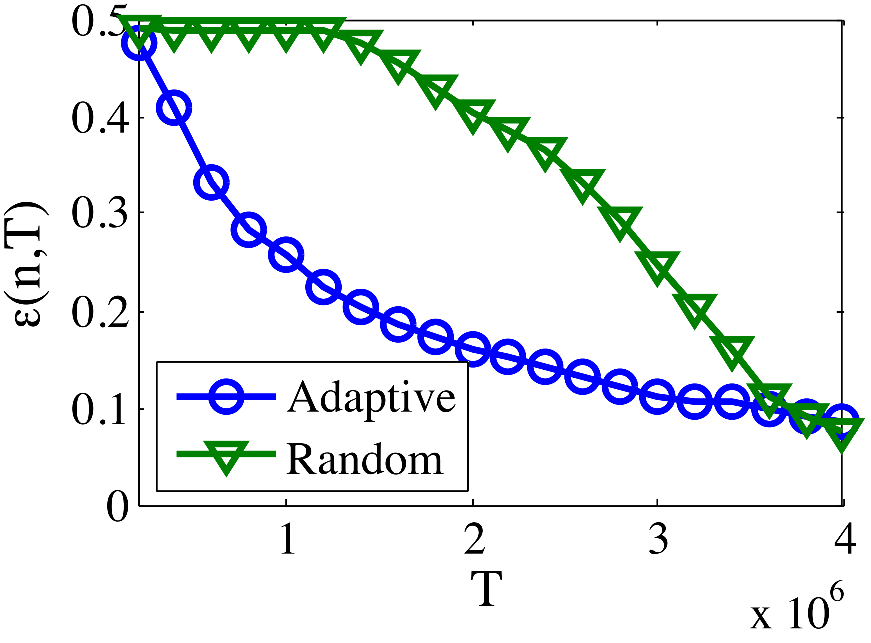
Then with high probability:

$$\varepsilon^{ASP}(n,T) \leq \exp\left(-\frac{T}{6n}(KL(q,p)+KL(p,q))\right).$$

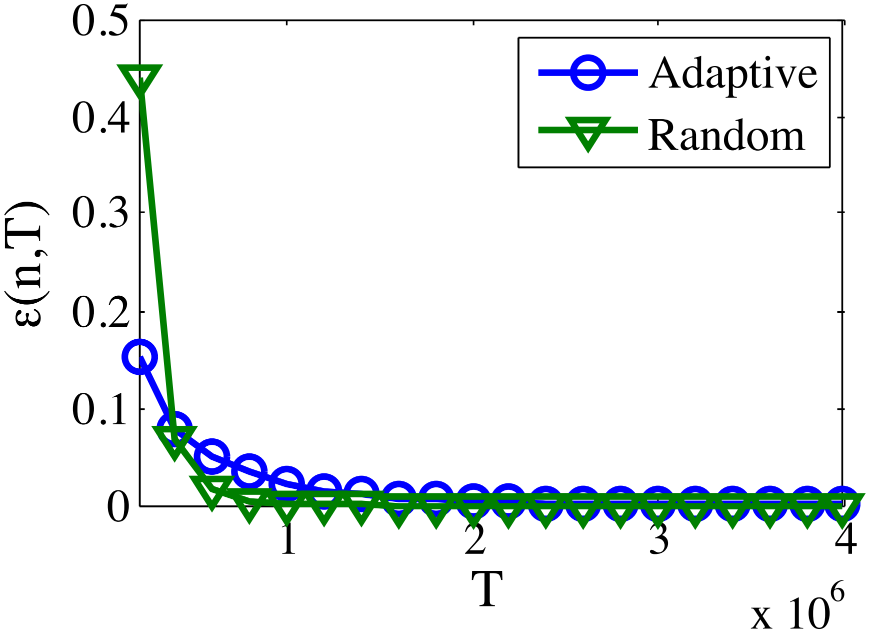
- The algorithm is asymptotically accurate under the necessary conditions for accurate detection in the case of adaptive sampling
- The necessary conditions for accurate detection are tight!

Non-adaptive vs. adaptive sampling

- n = 4000



p = 0.01, q=0.005

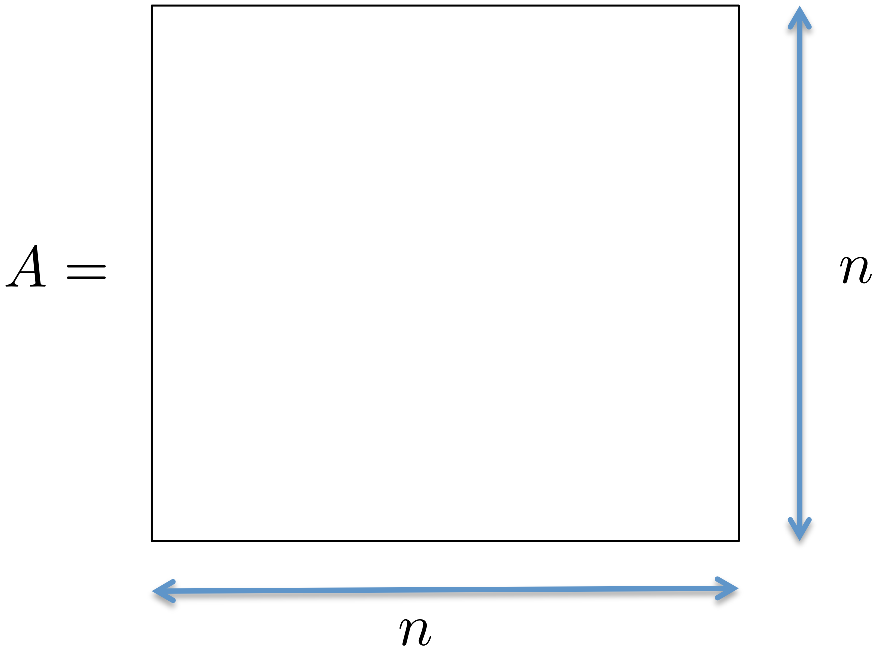


p = 0.1, q=0.05

3. Streaming, Memory-limited Algorithms

Memory and Streaming issues

- Storing and manipulating the adjacency matrix in RAM could be impossible (matlab cannot handle 5000 node networks!)
- Data about a node may arrive sequentially (one column at a time – e.g. recommender systems)



How to deal with $n = 10^7$?

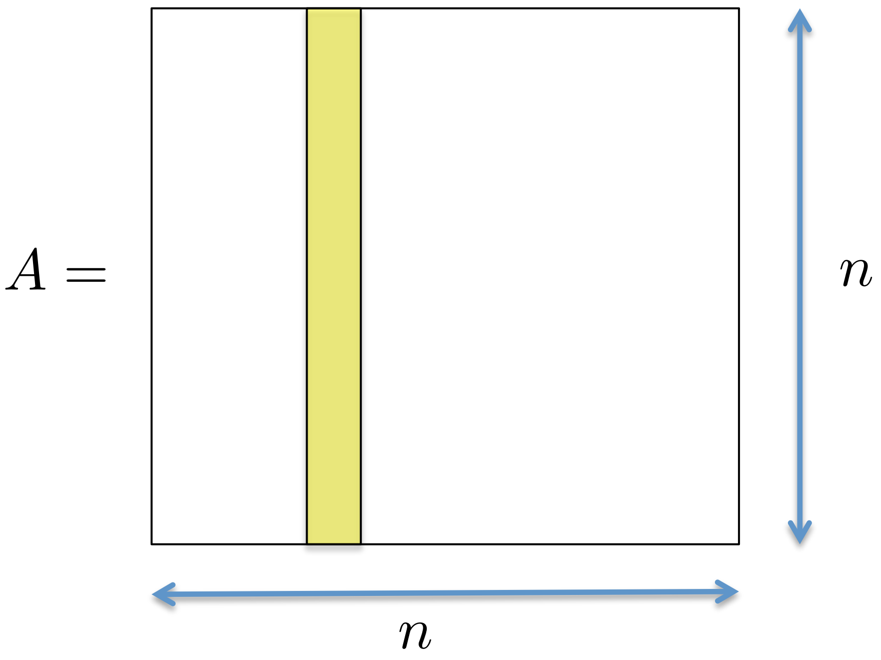
SBM with $f(n) = \omega(1)$

$$p = a\frac{f(n)}{n}$$

$$q = b\frac{f(n)}{n}$$

Memory and Streaming issues

- Storing and manipulating the adjacency matrix in RAM could be impossible (matlab cannot handle 5000 node networks!)
- Data about a node may arrive sequentially (one column at a time – e.g. recommender systems)

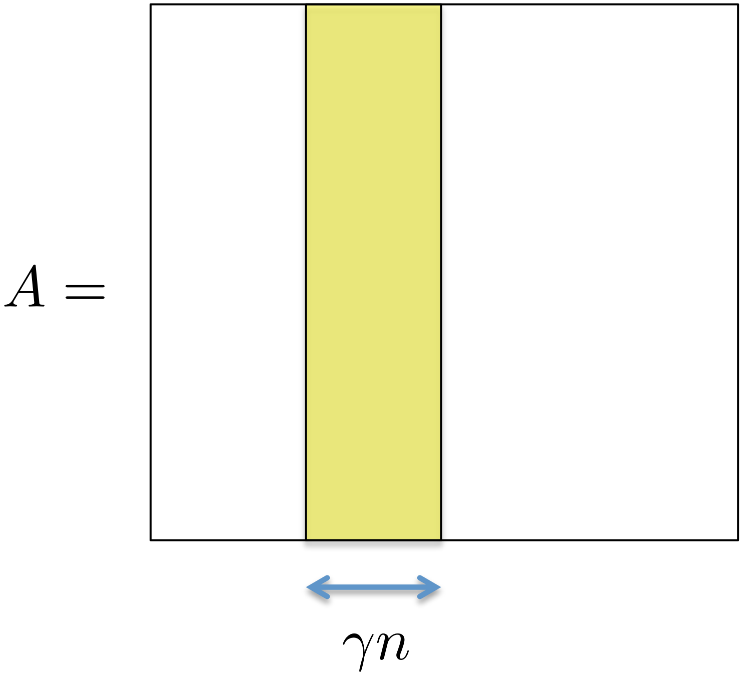


Offline algorithm: return the clusters after all the columns has been observed.

Online algorithm: classify the node immediately after its column is observed.

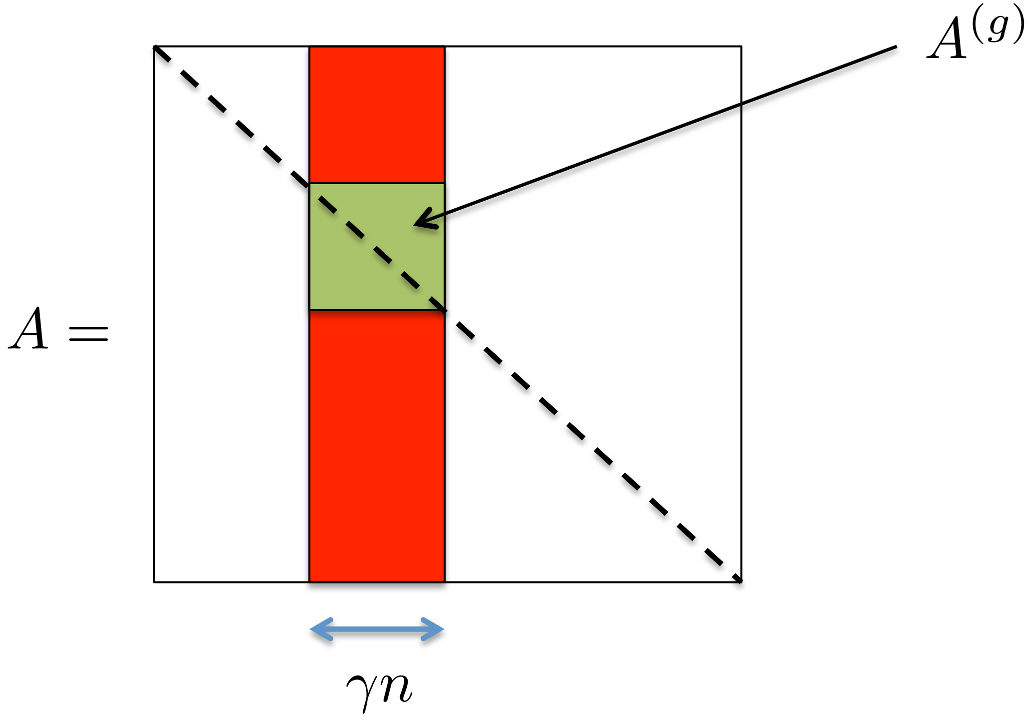
Classification with Partial Information

- Observe a proportion γ of columns
- Conditions on γ, f(n) to classify the corresponding nodes or all nodes asymptotically accurately?



Classification with Partial Information

- Observe a proportion γ of columns
- Conditions on γ, f(n) to classify the corresponding nodes or all nodes asymptotically accurately?



Fundamental limits

Theorem Assume that $\sqrt{\gamma} f(n) = o(1)$. Then asymptotic detection is impossible.

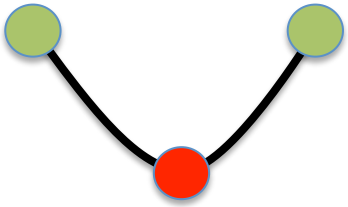
Theorem (i) If there is an algorithm classifying green nodes asymptotically accurately, then $\sqrt{\gamma} f(n) = \omega(1)$.

(ii) If there is an algorithm classifying all nodes asymptotically accurately, then $\gamma f(n) = \omega(1)$.

Remark: if one uses information about green nodes only, then it is possible to classify these nodes only if $\gamma f(n) = \omega(1)$. We have to use side information provided by red nodes.

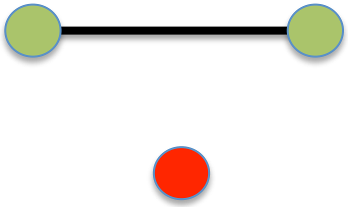
Algorithm for green nodes

- Indirect edges through red nodes



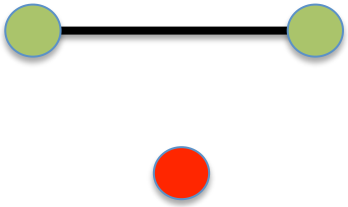
Algorithm for green nodes

- Indirect edges through red nodes



Algorithm for green nodes

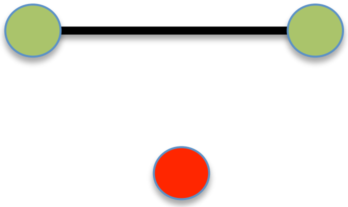
- Indirect edges through red nodes



Do it only for red nodes connected to exactly 2 green nodes
(**avoid statistical dependence**!)

Algorithm for green nodes

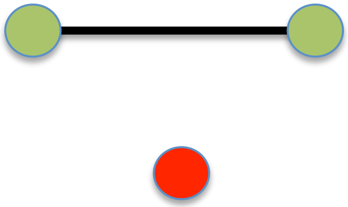
- Indirect edges through red nodes



- Result: a new adjacency matrix A'
- Algorithm: spectral method for $A^{(g)}$ and A' + keep the most informative matrix (with the highest normalized *K*-th eigenvalue)

Algorithm for green nodes

- Indirect edges through red nodes



- Result: a new adjacency matrix A'
- Algorithm: spectral method for $A^{(g)}$ and A' + keep the most informative matrix (with the highest normalized *K*-th eigenvalue)

Theorem Assume that $\sqrt{\gamma} f(n) = \omega(1)$. The above algorithm classifies the green nodes asymptotically accurately.

Algorithm for red nodes

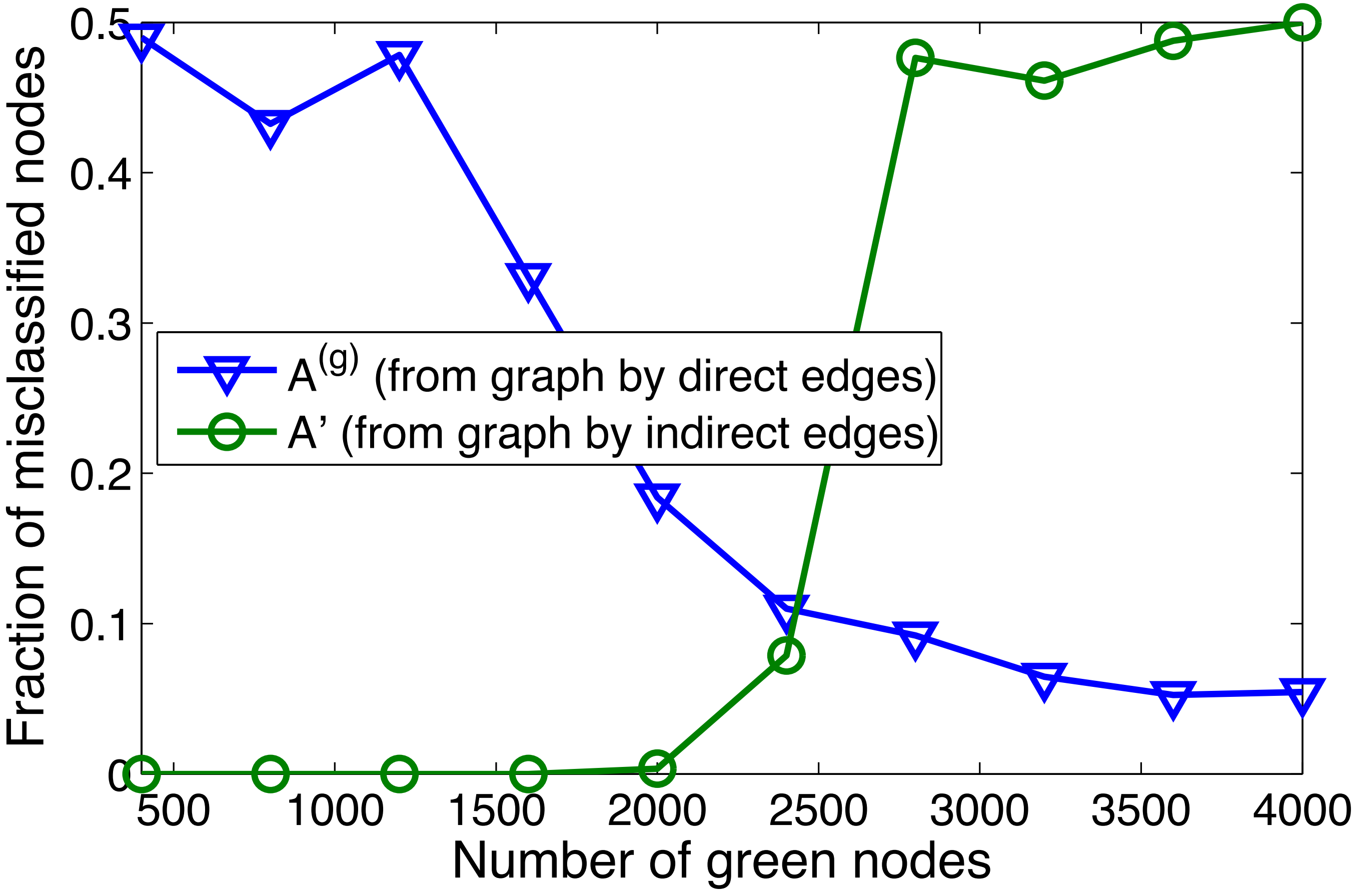
- Use the clusters of green nodes as kernels for the red nodes as in the adaptive sampling algorithm

Theorem Assume that $\gamma f(n) = \omega(1)$. The above algorithm classifies the red nodes asymptotically accurately.

The algorithms are optimal (see fundamental limits) – efficient use of side information provided by red nodes.

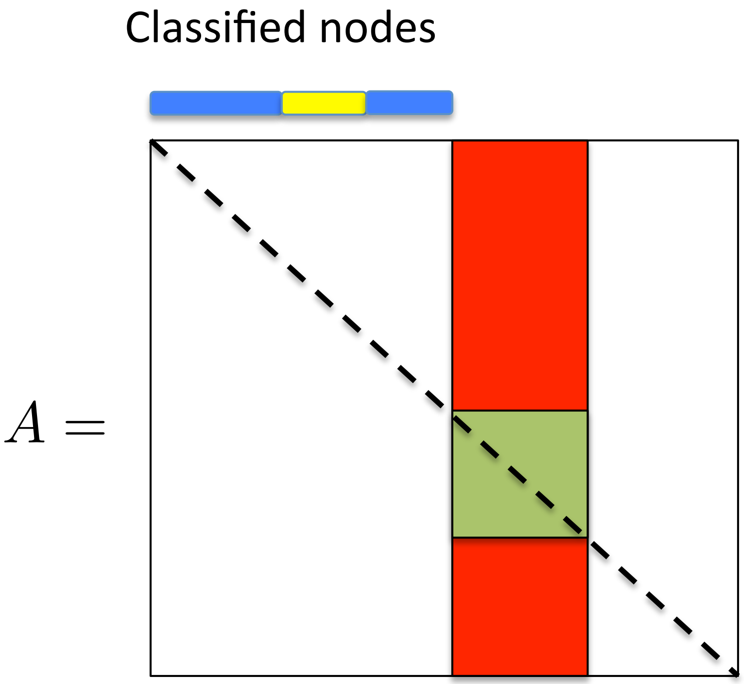
Example

n = 1000,000 ; p = 0.005, q = 0.001



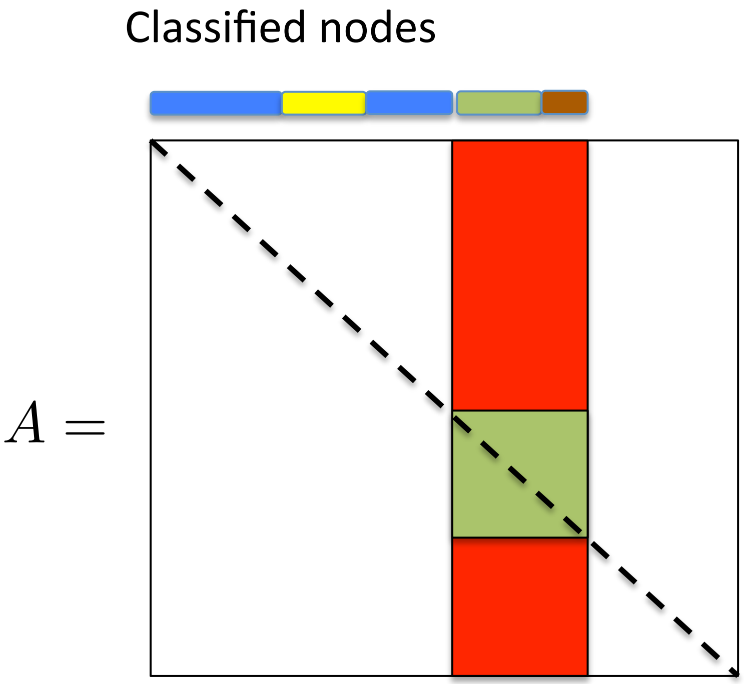
Offline Memory-efficient Algorithm

- Sequentially treat blocks of columns
  - Place the block of B columns in the memory



Offline Memory-efficient Algorithm

- Sequentially treat blocks of columns
  - Place the block of B columns in the memory
  - Classify the corresponding nodes using previous algorithm

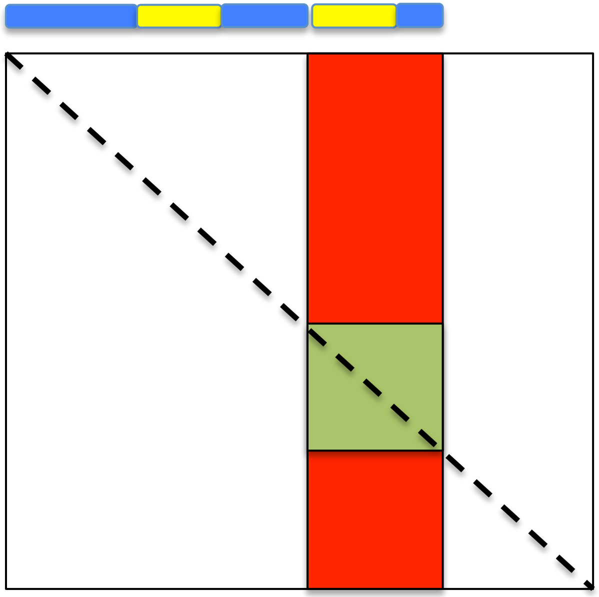


Offline Memory-efficient Algorithm

- Sequentially treat blocks of columns
  - Place the block of B columns in the memory
  - Classify the corresponding nodes using previous algorithm
  - Merge the obtained clusters with the previously observed clusters, and erase the block

Classified nodes

$A =$

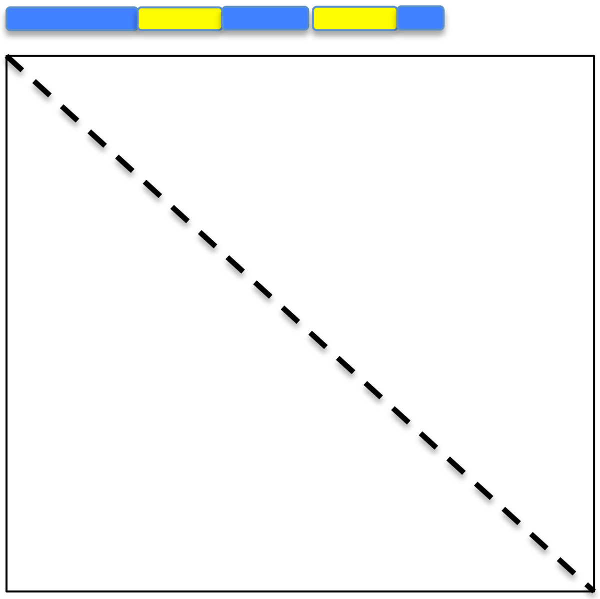


Offline Memory-efficient Algorithm

- Sequentially treat blocks of columns
  - Place the block of B columns in the memory
  - Classify the corresponding nodes using previous algorithm
  - Merge the obtained clusters with the previously observed clusters, and erase the block

Classified nodes

$A =$



Offline Memory-efficient Algorithm

- Memory-Performance trade-off

Theorem With block size $B = \frac{h(n)n}{\min(f(n), n^{1/3}) \log(n)}$, and memory $\Theta(nh(n) + n)$.

Assume that $h(n) = \omega\left(\frac{\log(n)}{\min(f(n), n^{1/3})}\right)$, then after observing

$T = \omega\left(\frac{n}{\min(f(n), n^{1/3})}\right)$ columns, we have: w.h.p.

$$\varepsilon^{\pi}(n, T) = O\left(\exp(-T\frac{\min(f(n), n^{1/3})}{n})\right)$$

Offline Memory-efficient Algorithm

- Memory-Performance trade-off
- Example: $f(n) = \log(n)^2$
  - linear memory $h(n) = 1$
  - After observing $T = n/\log(n)$, the proportion of misclassified nodes decays faster than $1/n$

Online Memory-efficient Algorithm

- Sequentially treat blocks of columns
  - Place the block of B columns in the memory
  - Classify the corresponding nodes using previous algorithm
  - Merge the obtained clusters with the clusters of the first block, and erase the block
  - Return the classification for the block, and erase it

Theorem With block size $B = \dfrac{h(n)n}{\min(f(n), n^{1/3}) \log(n)}$, and memory $\Theta(nh(n))$.
Assume that $h(n) = \omega\left(\dfrac{\log(n)}{\min(f(n), n^{1/3})}\right)$, then the above algorithm is asymptotically accurate.

Sublinear memory!

Conclusions

- A generic sampling framework extending the SBM
  - Necessary conditions for asymptotically accurate detection
  - Asymptotically optimal joint sampling and clustering algorithms
  - Our results hold in any regime! (Sparse or dense)
- Memory limited, streaming algorithm
  - Everything can be done with linear memory or even sublinear memory
  - Memory-performance trade-off
- BTW, similar results for more general models than SBM
- Cloud algorithms with communication constraints?

Thanks!

Papers

Yun-Proutiere, COLT 2014

Yun-Lelarge-Proutiere, NIPS 2014