Sparse Random Graphs-II

Souvik Dhara

Research Fellow, Simons Institute

Graph Limits and Processes on Networks:
From Epidemics to Misinformation Boot Camp

Recap: Erdős-Rényi subcritical phase

➢ Considered $\mathrm{ER}_n(\frac{\lambda}{n})$: Erdős-Renyi random graph with n vertices and edge probability $\frac{\lambda}{n}$

➢ Studied relation between exploration and branching processes, and showed that exploration can be dominated by a Poisson(λ) branching process

➢ For $\lambda < 1$: Showed $\mathbb{E}[C(v)] = O(1)$

Theorem: Subcritical $\mathrm{ER}_n(\frac{\lambda}{n})$

If $\lambda < 1$, then

$$\frac{\max_v C(v)}{\log n} \xrightarrow{\mathbb{P}} \frac{1}{I_\lambda}, \qquad \text{where } I_\lambda = \lambda - 1 - \log \lambda$$

Recap: Erdős-Rényi supercritical phase

We proved

Theorem: Supercritical $\mathrm{ER}_n(\frac{\lambda}{n})$

Let $C_{(i)}$:= i-th largest component of $\mathrm{ER}_n(\frac{\lambda}{n})$. If $\lambda > 1$, then

$$\frac{C_{(1)}}{n} \xrightarrow{\mathbb{P}} \zeta_\lambda > 0 \quad \text{and} \quad \frac{C_{(2)}}{n} \xrightarrow{\mathbb{P}} 0$$

Recap: Erdős-Rényi supercritical phase

We proved

Theorem: Supercritical $\mathrm{ER}_n(\frac{\lambda}{n})$

Let $C_{(i)}$:= i-th largest component of $\mathrm{ER}_n(\frac{\lambda}{n})$. If $\lambda > 1$, then

$$\frac{C_{(1)}}{n} \xrightarrow{\mathbb{P}} \zeta_\lambda > 0 \quad \text{and} \quad \frac{C_{(2)}}{n} \xrightarrow{\mathbb{P}} 0$$

The two main ingredients to prove this were...

Recap: Erdős-Rényi supercritical phase

We proved

Theorem: Supercritical $\mathrm{ER}_n(\frac{\lambda}{n})$

Let $C_{(i)} :=$ i-th largest component of $\mathrm{ER}_n(\frac{\lambda}{n})$. If $\lambda > 1$, then

$$\frac{C_{(1)}}{n} \xrightarrow{\mathbb{P}} \zeta_\lambda > 0 \quad \text{and} \quad \frac{C_{(2)}}{n} \xrightarrow{\mathbb{P}} 0$$

The two main ingredients to prove this were...

1. **Local neighborhood approximation:** Exploration from u (uniform vertex) is approximately BP whp and when BP survives, $C(u)$ is large

Recap: Erdős-Rényi supercritical phase

We proved

Theorem: Supercritical $\mathrm{ER}_n(\frac{\lambda}{n})$

Let $C_{(i)} :=$ i-th largest component of $\mathrm{ER}_n(\frac{\lambda}{n})$. If $\lambda > 1$, then

$$\frac{C_{(1)}}{n} \xrightarrow{\mathbb{P}} \zeta_\lambda > 0 \quad \text{and} \quad \frac{C_{(2)}}{n} \xrightarrow{\mathbb{P}} 0$$

The two main ingredients to prove this were...

1. **Local neighborhood approximation:** Exploration from u (uniform vertex) is approximately BP whp and when BP survives, $C(u)$ is large
2. **Two large components intersect:** u_1, u_2 uniform vertices

$$\lim_{L\to\infty} \lim_{n\to\infty} \mathbb{P}\left(C(u_1) \geqslant L, C(u_2) \geqslant L, u_1 \not\leftrightarrow u_2\right) = 0$$

Recap: Erdős-Rényi supercritical phase

We proved

Theorem: Supercritical $\mathrm{ER}_n(\frac{\lambda}{n})$

Let $C_{(i)}$:= i-th largest component of $\mathrm{ER}_n(\frac{\lambda}{n})$. If $\lambda > 1$, then

$$\frac{C_{(1)}}{n} \xrightarrow{\mathbb{P}} \zeta_\lambda > 0 \quad \text{and} \quad \frac{C_{(2)}}{n} \xrightarrow{\mathbb{P}} 0$$

The two main ingredients to prove this were...

1. **Local neighborhood approximation:** Exploration from u (uniform vertex) is approximately BP whp and when BP survives, $C(u)$ is large
2. **Two large components intersect:** u_1, u_2 uniform vertices

$$\lim_{L\to\infty} \lim_{n\to\infty} \mathbb{P}\left(C(u_1) \geqslant L, C(u_2) \geqslant L, u_1 \not\leftrightarrow u_2\right) = 0$$

➡ Was shown by growing two neighborhoods, and they must intersect when neighborhoods become large enough $\Omega(\sqrt{n})$

Plan today

- Consider other models with more realistic features, summarize results, and give heuristics for applying BP approximation technique

- Percolation, Epidemics: Use Path counting to prove results on general graphs and see whether we can apply these results to sparse graphs

- Using Stochastic Process Convergence in to find limits of component sizes of Random Graphs

Lets start by looking at a few Random Graph models with more 'realistic' features

Lets start by looking at a few Random Graph models with more 'realistic' features

- Global communities:
- Heterogeneous degrees:
- Dynamically evolving graphs:

Lets start by looking at a few Random Graph models with more 'realistic' features

- Global communities: **Stochastic Block Model**
- Heterogeneous degrees:
- Dynamically evolving graphs:

Lets start by looking at a few Random Graph models with more 'realistic' features

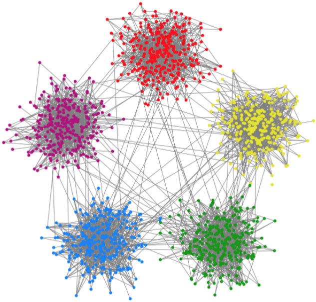
- Global communities: **Stochastic Block Model**
- Heterogeneous degrees: **Configuration Model**
- Dynamically evolving graphs:

Lets start by looking at a few Random Graph models with more 'realistic' features

- Global communities: **Stochastic Block Model**
- Heterogeneous degrees: **Configuration Model**
- Dynamically evolving graphs: **Preferential Attachment Model**

Stochastic Block Model

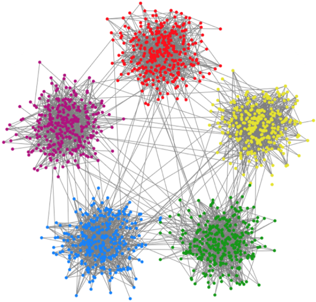
➢ Model with global community structure – popular model in CS/ML for *community detection problem*



Pic source: Abbe (2018)

Stochastic Block Model

➢ Model with global community structure – popular model in CS/ML for *community detection problem*



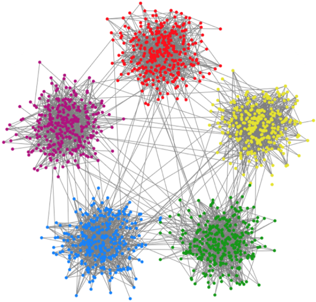
Model description:

1. $K \geqslant 2$ communities, size of community $i = n_i$, where $\frac{n_i}{n} \to \rho_i$, $\rho_i > 0$

Pic source: Abbe (2018)

Stochastic Block Model

➢ Model with global community structure – popular model in CS/ML for *community detection problem*



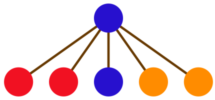
Model description:

1. $K \geqslant 2$ communities, size of community $i = n_i$, where $\frac{n_i}{n} \to \rho_i$, $\rho_i > 0$
2. Edge between community i, j w.p. $\frac{P_{ij}}{n}$ ($P_{ij} \in (0, 1)$), independently

Pic source: Abbe (2018)

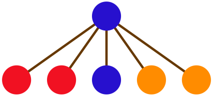
Local neighborhoods of Stochastic Block Model

Local neighborhoods of Stochastic Block Model



Local neighborhood approximated by this BP

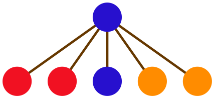
Local neighborhoods of Stochastic Block Model



Local neighborhood approximated by this BP

➢ Uniform vertex u in community i w.p. ρ_i

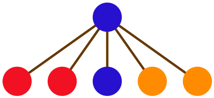
Local neighborhoods of Stochastic Block Model



Local neighborhood approximated by this BP

- ➢ Uniform vertex u in community i w.p. ρ_i
- ➢ Poisson$(\rho_j P_{ij})$ neighbors from community j

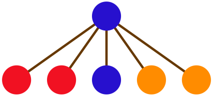
Local neighborhoods of Stochastic Block Model



Local neighborhood approximated by this BP

- ➢ Uniform vertex u in community i w.p. ρ_i
- ➢ Poisson($\rho_j P_{ij}$) neighbors from community j
- ➢ Gives rise to Multi-type Branching Process

Local neighborhoods of Stochastic Block Model

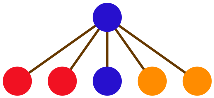


Local neighborhood approximated by this BP

- ➢ Uniform vertex u in community i w.p. ρ_i
- ➢ Poisson$(\rho_j P_{ij})$ neighbors from community j
- ➢ Gives rise to Multi-type Branching Process

Let $P^\star_{ij} = \rho_j P_{ij}$ and $\lambda_1(P^\star)$ be largest eigenvalue of $P^\star$.

Local neighborhoods of Stochastic Block Model



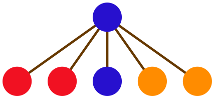
Local neighborhood approximated by this BP

- Uniform vertex u in community i w.p. ρ_i
- Poisson$(\rho_j P_{ij})$ neighbors from community j
- Gives rise to Multi-type Branching Process

Let $P^\star_{ij} = \rho_j P_{ij}$ and $\lambda_1(P^\star)$ be largest eigenvalue of $P^\star$. Then

Fact: $\mathbb{P}(\text{BP survives}) = \zeta > 0 \quad$ when $\lambda_1(P^\star) > 1$

Local neighborhoods of Stochastic Block Model



Local neighborhood approximated by this BP

- ➢ Uniform vertex u in community i w.p. ρ_i
- ➢ Poisson$(\rho_j P_{ij})$ neighbors from community j
- ➢ Gives rise to Multi-type Branching Process

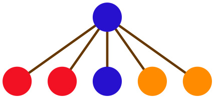
Let $P^\star_{ij} = \rho_j P_{ij}$ and $\lambda_1(P^\star)$ be largest eigenvalue of $P^\star$. Then

Fact: $\mathbb{P}(\text{BP survives}) = \zeta > 0$ when $\lambda_1(P^\star) > 1$

Theorem: Giant for SBM

1. For $\lambda_1(P^\star) < 1$: $\frac{C_{(1)}}{n} \xrightarrow{\mathbb{P}} 0$
2. For $\lambda_1(P^\star) > 1$: $\frac{C_{(1)}}{n} \xrightarrow{\mathbb{P}} \zeta > 0$ and $\frac{C_{(2)}}{n} \xrightarrow{\mathbb{P}} 0$ whp

Local neighborhoods of Stochastic Block Model



Local neighborhood approximated by this BP

➢ Uniform vertex u in community i w.p. ρ_i

➢ Poisson$(\rho_j P_{ij})$ neighbors from community j

➢ Gives rise to Multi-type Branching Process

Let $P^\star_{ij} = \rho_j P_{ij}$ and $\lambda_1(P^\star)$ be largest eigenvalue of $P^\star$. Then

Fact: $\mathbb{P}(\text{BP survives}) = \zeta > 0$ when $\lambda_1(P^\star) > 1$

Theorem: Giant for SBM

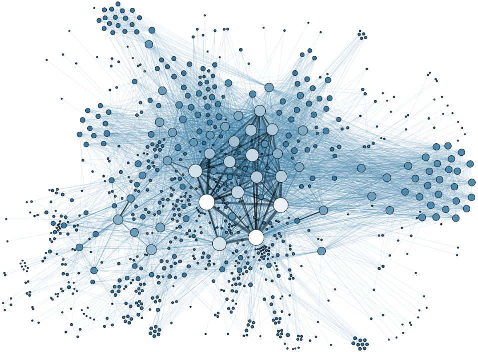
1. For $\lambda_1(P^\star) < 1$: $\frac{C_{(1)}}{n} \xrightarrow{\mathbb{P}} 0$
2. For $\lambda_1(P^\star) > 1$: $\frac{C_{(1)}}{n} \xrightarrow{\mathbb{P}} \zeta > 0$ and $\frac{C_{(2)}}{n} \xrightarrow{\mathbb{P}} 0$ whp

➢ There is a more challenging and general models with continuum of colors
➡ See foundational work of Bollobás, Janson, Riordan (2007) on general inhomogeneous random graphs

Up next: Model for degree-heterogeneous networks

Pic source: Wikimedia Commons

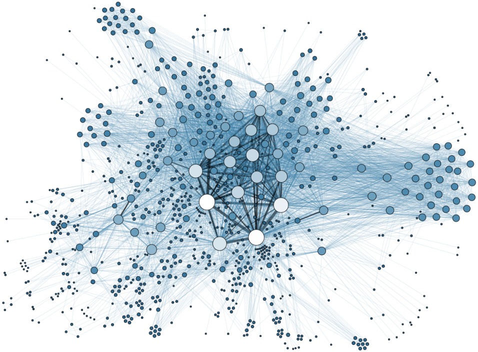
Up next: Model for degree-heterogeneous networks



- ➢ Such degree-heterogenous networks with hubs are common occurrences
  - ➡ The degree distribution can be power-law, truncated power-law etc., but it is definitely quite far from Poisson

Pic source: Wikimedia Commons

Up next: Model for degree-heterogeneous networks



- Such degree-heterogenous networks with hubs are common occurrences
  - The degree distribution can be power-law, truncated power-law etc., but it is definitely quite far from Poisson
  - Need a simple, analytically tractable model – **Configuration Model**

Pic source: Wikimedia Commons

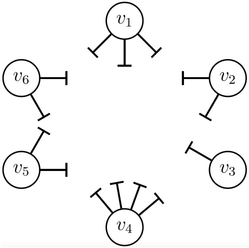
Configuration Model

Canonical model to generate graphs with given degrees $\mathbf{d} = (d_1, \ldots, d_n)$

Pic source: van der Hofstad (2017)

Configuration Model

Canonical model to generate graphs with given degrees $\mathbf{d} = (d_1, \ldots, d_n)$

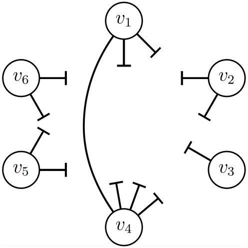


➢ Start with d_i *half-edges* to vertex i

Pic source: van der Hofstad (2017)

Configuration Model

Canonical model to generate graphs with given degrees $\mathbf{d} = (d_1, \ldots, d_n)$

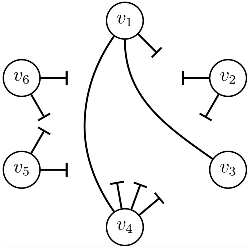


- ➢ Start with d_i *half-edges* to vertex i
- ➢ Pair half-edges uniformly

Pic source: van der Hofstad (2017)

Configuration Model

Canonical model to generate graphs with given degrees $\mathbf{d} = (d_1, \ldots, d_n)$

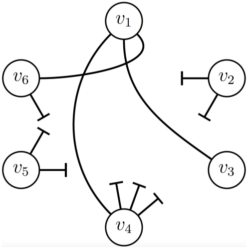


- ➢ Start with d_i *half-edges* to vertex i
- ➢ Pair half-edges uniformly

Pic source: van der Hofstad (2017)

Configuration Model

Canonical model to generate graphs with given degrees $\mathbf{d} = (d_1, \ldots, d_n)$



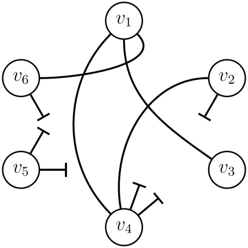
➢ Start with d_i *half-edges* to vertex i

➢ Pair half-edges uniformly

Pic source: van der Hofstad (2017)

Configuration Model

Canonical model to generate graphs with given degrees $\mathbf{d} = (d_1, \ldots, d_n)$

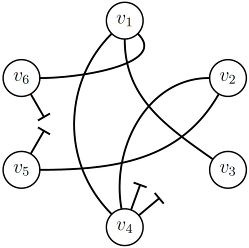


- Start with d_i *half-edges* to vertex i
- Pair half-edges uniformly

Pic source: van der Hofstad (2017)

Configuration Model

Canonical model to generate graphs with given degrees $\mathbf{d} = (d_1, \ldots, d_n)$

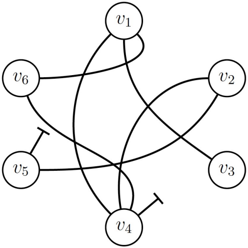


- Start with d_i *half-edges* to vertex i
- Pair half-edges uniformly

Pic source: van der Hofstad (2017)

Configuration Model

Canonical model to generate graphs with given degrees $\mathbf{d} = (d_1, \ldots, d_n)$

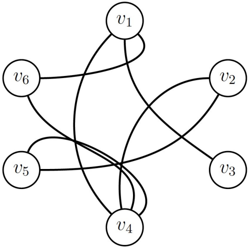


- ➢ Start with d_i *half-edges* to vertex i
- ➢ Pair half-edges uniformly

Pic source: van der Hofstad (2017)

Configuration Model

Canonical model to generate graphs with given degrees $\mathbf{d} = (d_1, \ldots, d_n)$

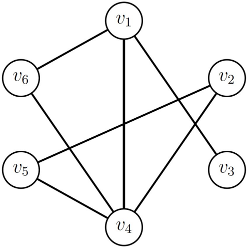


- ➢ Start with d_i *half-edges* to vertex i
- ➢ Pair half-edges uniformly

Pic source: van der Hofstad (2017)

Configuration Model

Canonical model to generate graphs with given degrees $\mathbf{d} = (d_1, \ldots, d_n)$

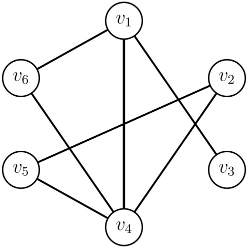


- ➢ Start with d_i *half-edges* to vertex i
- ➢ Pair half-edges uniformly

Pic source: van der Hofstad (2017)

Configuration Model

Canonical model to generate graphs with given degrees $\mathbf{d} = (d_1, \ldots, d_n)$

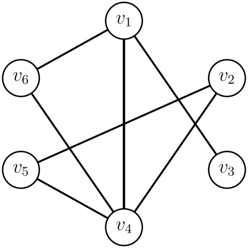


- ➢ Start with d_i *half-edges* to vertex i
- ➢ Pair half-edges uniformly
- ➢ Self-loops/multi-edges may occur

Pic source: van der Hofstad (2017)

Configuration Model

Canonical model to generate graphs with given degrees $\mathbf{d} = (d_1, \ldots, d_n)$

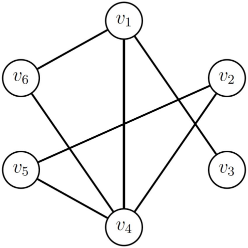


- Start with d_i *half-edges* to vertex i
- Pair half-edges uniformly
- Self-loops/multi-edges may occur
- Denote resulting (multi)-graph by $\mathrm{CM}_n(\mathbf{d})$

Pic source: van der Hofstad (2017)

Configuration Model

Canonical model to generate graphs with given degrees $\mathbf{d} = (d_1, \ldots, d_n)$



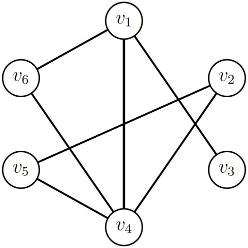
- Start with d_i *half-edges* to vertex i
- Pair half-edges uniformly
- Self-loops/multi-edges may occur
- Denote resulting (multi)-graph by $\mathrm{CM}_n(\mathbf{d})$

Interesting Fact: Law of $\mathrm{CM}_n(\mathbf{d})$ given no loops/multi-edges produced is same as Uniform distribution over all possible simple graphs with degree $\mathbf{d}$

Pic source: van der Hofstad (2017)

Configuration Model

Canonical model to generate graphs with given degrees $\mathbf{d} = (d_1, \ldots, d_n)$



- ➢ Start with d_i *half-edges* to vertex i
- ➢ Pair half-edges uniformly
- ➢ Self-loops/multi-edges may occur
- ➢ Denote resulting (multi)-graph by $\mathrm{CM}_n(\mathbf{d})$

Interesting Fact: Law of $\mathrm{CM}_n(\mathbf{d})$ given no loops/multi-edges produced is same as Uniform distribution over all possible simple graphs with degree $\mathbf{d}$

Brief History:

➢ Introduced by Bender and Canfield (1978), Bollobás (1980) to study uniform random regular graphs

➢ Giant emergence studied by Molloy & Reed (1995, 1998)

Pic source: van der Hofstad (2017)

Regularity conditions on the degree

Regularity conditions on the degree

$D_n :=$ degree of uniform vertex, so the distribution of D_n is the empirical degree distribution $\frac{1}{n}\sum_i \delta_{d_i}$

Regularity conditions on the degree

$D_n :=$ degree of uniform vertex, so the distribution of D_n is the empirical degree distribution $\frac{1}{n}\sum_i \delta_{d_i}$

Regularity conditions.

1. Convergence of degree distribution. $D_n \xrightarrow{d} D$

Regularity conditions on the degree

$D_n :=$ degree of uniform vertex, so the distribution of D_n is the empirical degree distribution $\frac{1}{n}\sum_i \delta_{d_i}$

Regularity conditions.

1. Convergence of degree distribution. $D_n \xrightarrow{d} D$
2. Convergence of moment. $\mathbb{E}[D_n] \to \mathbb{E}[D] < \infty$

Regularity conditions on the degree

D_n := degree of uniform vertex, so the distribution of D_n is the empirical degree distribution $\frac{1}{n}\sum_i \delta_{d_i}$

Regularity conditions.

1. Convergence of degree distribution. $D_n \xrightarrow{d} D$
2. Convergence of moment. $\mathbb{E}[D_n] \to \mathbb{E}[D] < \infty$

Ensures sparsity: $\mathbb{E}[D_n] = \frac{1}{n}\sum_i d_i \to \text{constant}$

Regularity conditions on the degree

$D_n :=$ degree of uniform vertex, so the distribution of D_n is the empirical degree distribution $\frac{1}{n}\sum_i \delta_{d_i}$

Regularity conditions.

1. Convergence of degree distribution. $D_n \xrightarrow{d} D$
2. Convergence of moment. $\mathbb{E}[D_n] \to \mathbb{E}[D] < \infty$

   Ensures sparsity: $\mathbb{E}[D_n] = \frac{1}{n}\sum_i d_i \to \text{constant}$

3. $\mathbb{P}(D = 2) < 1$, otherwise generated graph is union of cycles

Regularity conditions on the degree

$D_n :=$ degree of uniform vertex, so the distribution of D_n is the empirical degree distribution $\frac{1}{n}\sum_i \delta_{d_i}$

Regularity conditions.

❶ Convergence of degree distribution. $D_n \xrightarrow{d} D$

❷ Convergence of moment. $\mathbb{E}[D_n] \to \mathbb{E}[D] < \infty$

$$\text{Ensures sparsity:} \quad \mathbb{E}[D_n] = \frac{1}{n}\sum_i d_i \to \text{constant}$$

❸ $\mathbb{P}(D = 2) < 1$, otherwise generated graph is union of cycles

➢ If the degrees are iid samples from a power-law with finite mean, then these conditions are satisfied

Regularity conditions on the degree

$D_n :=$ degree of uniform vertex, so the distribution of D_n is the empirical degree distribution $\frac{1}{n}\sum_i \delta_{d_i}$

Regularity conditions.

❶ Convergence of degree distribution. $D_n \xrightarrow{d} D$

❷ Convergence of moment. $\mathbb{E}[D_n] \to \mathbb{E}[D] < \infty$

Ensures sparsity: $$\mathbb{E}[D_n] = \frac{1}{n}\sum_i d_i \to \text{constant}$$

❸ $\mathbb{P}(D = 2) < 1$, otherwise generated graph is union of cycles

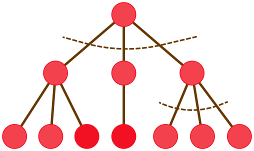
➢ If the degrees are iid samples from a power-law with finite mean, then these conditions are satisfied

➢ Most often, one also assumes $\mathbb{E}[D_n^2] \to \mathbb{E}[D^2] < \infty$, which ensures

$$\liminf_{n\to\infty} \mathbb{P}(\mathrm{CM}_n(\mathbf{d}) \text{ is simple}) > 0 \qquad \text{Janson (2009)}$$

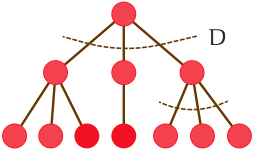
so that the results carry over to uniform graphs

Local neighborhoods of $\mathrm{CM}_n(\mathbf{d})$



BP approximation:

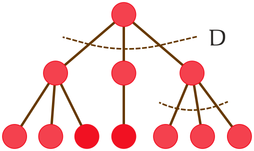
Local neighborhoods of $\mathrm{CM}_n(\mathbf{d})$



BP approximation:

❶ Starts D many progeny

Local neighborhoods of $\mathrm{CM}_n(\mathbf{d})$

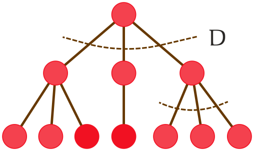


BP approximation:

❶ Starts D many progeny

Let n_l = # vertices of degree l and $\frac{n_l}{n} \to p_l$

Local neighborhoods of $\mathrm{CM}_n(\mathbf{d})$



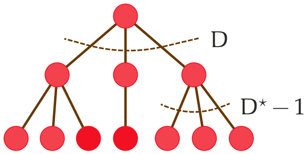
BP approximation:

❶ Starts D many progeny

Let n_l = # vertices of degree l and $\frac{n_l}{n} \to p_l$

Next progeny is $D^\star - 1$ with $\mathbb{P}(D^\star = k) = \dfrac{kp_k}{\sum_l lp_l}$ **Size-biased distribution**

Local neighborhoods of $\mathrm{CM}_n(\mathbf{d})$



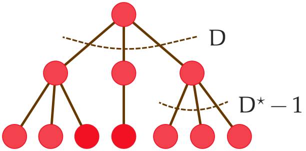
BP approximation:

❶ Starts D many progeny

❷ Produces $D^\star - 1$ in next step

Let $n_l = \#$ vertices of degree l and $\frac{n_l}{n} \to p_l$

Next progeny is $D^\star - 1$ with $\mathbb{P}(D^\star = k) = \dfrac{kp_k}{\sum_l lp_l}$ **Size-biased distribution**



BP approximation:

❶ Starts D many progeny

❷ Produces $D^\star - 1$ in next step

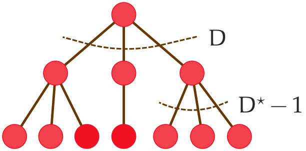
Let n_l = # vertices of degree l and $\frac{n_l}{n} \to p_l$

Next progeny is $D^\star - 1$ with $\mathbb{P}(D^\star = k) = \dfrac{kp_k}{\sum_l lp_l}$ **Size-biased distribution**

Now,

$$\mathbb{E}[D^\star - 1] = \frac{\sum_k (k-1)kp_k}{\sum_k kp_k}$$

Local neighborhoods of $CM_n(\mathbf{d})$



BP approximation:

❶ Starts D many progeny

❷ Produces $D^\star - 1$ in next step

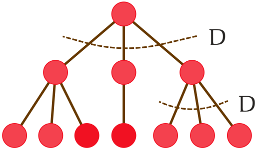
Let $n_l = \#$ vertices of degree l and $\frac{n_l}{n} \to p_l$

Next progeny is $D^\star - 1$ with $\mathbb{P}(D^\star = k) = \frac{kp_k}{\sum_l lp_l}$ **Size-biased distribution**

Now,

$$\mathbb{E}[D^\star - 1] = \frac{\sum_k (k-1)kp_k}{\sum_k kp_k} = \frac{\mathbb{E}[D(D-1)]}{\mathbb{E}[D]}$$

Local neighborhoods of $CM_n(\mathbf{d})$



BP approximation:

❶ Starts D many progeny

❷ Produces $D^\star - 1$ in next step

Let n_l = # vertices of degree l and $\frac{n_l}{n} \to p_l$

Next progeny is $D^\star - 1$ with $\mathbb{P}(D^\star = k) = \frac{kp_k}{\sum_l lp_l}$ **Size-biased distribution**

Now,

$$\mathbb{E}[D^\star - 1] = \frac{\sum_k (k-1)kp_k}{\sum_k kp_k} = \frac{\mathbb{E}[D(D-1)]}{\mathbb{E}[D]}$$

Therefore,

$$\mathbb{P}(\text{BP survives}) = \zeta > 0 \quad \text{when } \nu := \frac{\mathbb{E}[D(D-1)]}{\mathbb{E}[D]} > 1$$

Emergence of Giant for $\mathrm{CM}_n(\mathbf{d})$

Let $\nu := \frac{\mathbb{E}[D(D-1)]}{\mathbb{E}[D]}$

Theorem: Giant for $\mathrm{CM}_n(\mathbf{d})$

❶ For $\nu < 1$: $\frac{C_{(1)}}{n} \xrightarrow{\mathbb{P}} 0$ whp

❷ For $\nu > 1$: $\frac{C_{(1)}}{n} \xrightarrow{\mathbb{P}} \zeta > 0$ and $\frac{C_{(2)}}{n} \xrightarrow{\mathbb{P}} 0$ whp

Emergence of Giant for $\mathrm{CM}_n(\mathbf{d})$

Let $\nu := \frac{\mathbb{E}[D(D-1)]}{\mathbb{E}[D]}$

Theorem: Giant for $\mathrm{CM}_n(\mathbf{d})$

❶ For $\nu < 1$: $\frac{C_{(1)}}{n} \xrightarrow{\mathbb{P}} 0$ whp

❷ For $\nu > 1$: $\frac{C_{(1)}}{n} \xrightarrow{\mathbb{P}} \zeta > 0$ and $\frac{C_{(2)}}{n} \xrightarrow{\mathbb{P}} 0$ whp

For the proof, there were two ingredients:

Emergence of Giant for $\mathrm{CM}_n(\mathbf{d})$

Let $\nu := \frac{\mathbb{E}[D(D-1)]}{\mathbb{E}[D]}$

Theorem: Giant for $\mathrm{CM}_n(\mathbf{d})$

❶ For $\nu < 1$: $\frac{C_{(1)}}{n} \xrightarrow{\mathbb{P}} 0$ whp

❷ For $\nu > 1$: $\frac{C_{(1)}}{n} \xrightarrow{\mathbb{P}} \zeta > 0$ and $\frac{C_{(2)}}{n} \xrightarrow{\mathbb{P}} 0$ whp

For the proof, there were two ingredients:

➡ Local neighborhood approximation – Just discussed

Emergence of Giant for $\mathrm{CM}_n(\boldsymbol{d})$

Let $\nu := \frac{\mathbb{E}[D(D-1)]}{\mathbb{E}[D]}$

Theorem: Giant for $\mathrm{CM}_n(\boldsymbol{d})$

❶ For $\nu < 1$: $\frac{C_{(1)}}{n} \xrightarrow{\mathbb{P}} 0$ whp

❷ For $\nu > 1$: $\frac{C_{(1)}}{n} \xrightarrow{\mathbb{P}} \zeta > 0$ and $\frac{C_{(2)}}{n} \xrightarrow{\mathbb{P}} 0$ whp

For the proof, there were two ingredients:

- ➡ Local neighborhood approximation – Just discussed
- ➡ Will skip Two large components intersect

$$\mathbb{P}\left(C(u_1) \geqslant L, C(u_2) \geqslant L, u_1 \not\leftrightarrow u_2\right) \approx 0$$

Can be proved using similar ideas as ER, but is more complicated [a]

[a] see van der Hofstad (2021): *The giant in random graphs is almost local*

Emergence of Giant for $\mathrm{CM}_n(\boldsymbol{d})$

Let $\nu := \frac{\mathbb{E}[D(D-1)]}{\mathbb{E}[D]}$

Theorem: Giant for $\mathrm{CM}_n(\boldsymbol{d})$

❶ For $\nu < 1 :$ $\frac{C_{(1)}}{n} \xrightarrow{\mathbb{P}} 0$ whp

❷ For $\nu > 1 :$ $\frac{C_{(1)}}{n} \xrightarrow{\mathbb{P}} \zeta > 0$ and $\frac{C_{(2)}}{n} \xrightarrow{\mathbb{P}} 0$ whp

Two cases with different qualitative behavior than Erdős-Rényi:

Emergence of Giant for $\mathrm{CM}_n(\mathbf{d})$

Let $\nu := \frac{\mathbb{E}[D(D-1)]}{\mathbb{E}[D]}$

Theorem: Giant for $\mathrm{CM}_n(\mathbf{d})$

❶ For $\nu < 1$: $\frac{C_{(1)}}{n} \xrightarrow{\mathbb{P}} 0$ whp

❷ For $\nu > 1$: $\frac{C_{(1)}}{n} \xrightarrow{\mathbb{P}} \zeta > 0$ and $\frac{C_{(2)}}{n} \xrightarrow{\mathbb{P}} 0$ whp

Two cases with different qualitative behavior than Erdős-Rényi:

➢ It may be that $\nu = \infty$, e.g., for power-law degree distribution with infinite variance, and giant always exists in such networks

Emergence of Giant for $\mathrm{CM}_n(\mathbf{d})$

Let $\nu := \frac{\mathbb{E}[D(D-1)]}{\mathbb{E}[D]}$

Theorem: Giant for $\mathrm{CM}_n(\mathbf{d})$

❶ For $\nu < 1$: $\frac{C_{(1)}}{n} \xrightarrow{\mathbb{P}} 0$ whp

❷ For $\nu > 1$: $\frac{C_{(1)}}{n} \xrightarrow{\mathbb{P}} \zeta > 0$ and $\frac{C_{(2)}}{n} \xrightarrow{\mathbb{P}} 0$ whp

Two cases with different qualitative behavior than Erdős-Rényi:

➢ It may be that $\nu = \infty$, e.g., for power-law degree distribution with infinite variance, and giant always exists in such networks

➢ It may be that $\zeta = 1$, e.g., if $\mathbb{P}(D \geqslant 3) = 1$, then BP survives w.p. 1

Model for dynamically growing networks

➢ Around '90s, huge interest for dynamically growing networks that produce heterogeneous degree distribution

Model for dynamically growing networks

➢ Around '90s, huge interest for dynamically growing networks that produce heterogeneous degree distribution

➢ To model this, Barabási and Albert (1999) proposed the *Preferential attachment model*. Idea goes back to Yule (1925)

Model for dynamically growing networks

➢ Around '90s, huge interest for dynamically growing networks that produce heterogeneous degree distribution

➢ To model this, Barabási and Albert (1999) proposed the *Preferential attachment model*. Idea goes back to Yule (1925)

➡ Rich-get-richer principle: New vertices connect to high-degree vertices

Model for dynamically growing networks

➢ Around '90s, huge interest for dynamically growing networks that produce heterogeneous degree distribution

➢ To model this, Barabási and Albert (1999) proposed the *Preferential attachment model*. Idea goes back to Yule (1925)

➡ Rich-get-richer principle: New vertices connect to high-degree vertices

➢ Bollobás, Riordan, Spencer and Tusnády (2001) were to first study this model rigorously

Preferential attachment model

- New vertices come with m connections

Preferential attachment model

➢ New vertices come with m connections

1. Start with a single vertex v_1 with m self-loops

Preferential attachment model

➢ New vertices come with m connections

1. Start with a single vertex v_1 with m self-loops
2. At time t, v_t arrives with m potential connections. Let $\deg(v_i, t, e) :=$ degree of v_i at time t after e-th edge is paired

Preferential attachment model

- New vertices come with m connections

1. Start with a single vertex v_1 with m self-loops
2. At time t, v_t arrives with m potential connections. Let $\deg(v_i, t, e) :=$ degree of v_i at time t after e-th edge is paired
   - The e edge connects with $v_i \neq v_t$ w.p. $\propto \deg(v_i, t, e-1)$

Preferential attachment model

➢ New vertices come with m connections

1. Start with a single vertex v_1 with m self-loops
2. At time t, v_t arrives with m potential connections. Let $\deg(v_i, t, e) :=$ degree of v_i at time t after e-th edge is paired
   - The e edge connects with $v_i \neq v_t$ w.p. $\propto \deg(v_i, t, e-1)$
   - Connects to itself w.p. $\propto \deg(v_i, t, e-1) + 1$

Preferential attachment model

➢ New vertices come with m connections

1. Start with a single vertex v_1 with m self-loops
2. At time t, v_t arrives with m potential connections. Let $\deg(v_i, t, e) :=$ degree of v_i at time t after e-th edge is paired
   - The e edge connects with $v_i \neq v_t$ w.p. $\propto \deg(v_i, t, e-1)$
   - Connects to itself w.p. $\propto \deg(v_i, t, e-1) + 1$
3. After n steps, we get a graph with n vertices and nm edges

Preferential attachment model

➢ New vertices come with m connections

1. Start with a single vertex v_1 with m self-loops
2. At time t, v_t arrives with m potential connections. Let $\deg(v_i, t, e) :=$ degree of v_i at time t after e-th edge is paired
   - The e edge connects with $v_i \neq v_t$ w.p. $\propto \deg(v_i, t, e-1)$
   - Connects to itself w.p. $\propto \deg(v_i, t, e-1) + 1$
3. After n steps, we get a graph with n vertices and nm edges

➢ If $m = 1$, this process produces a tree called preferential attachment tree

Preferential attachment model properties

- ➢ Graph is always connected, so no question of giant emergence here

Bollobás, Riordan, Spencer and Tusnády (2001)

Preferential attachment model properties

- Graph is always connected, so no question of giant emergence here
- **Interesting questions:** Degree distribution, local neighborhood structure

Bollobás, Riordan, Spencer and Tusnády (2001)

Preferential attachment model properties

- Graph is always connected, so no question of giant emergence here
- **Interesting questions:** Degree distribution, local neighborhood structure

Theorem: Degree distribution of PAM

Let $P_k(n) = \frac{\#\text{vertices of degree } k}{n}$.

Bollobás, Riordan, Spencer and Tusnády (2001)

Preferential attachment model properties

- Graph is always connected, so no question of giant emergence here
- **Interesting questions:** Degree distribution, local neighborhood structure

Theorem: Degree distribution of PAM

Let $P_k(n) = \frac{\#\text{vertices of degree } k}{n}$. Fix $m \geqslant 1$.

Bollobás, Riordan, Spencer and Tusnády (2001)

Preferential attachment model properties

- Graph is always connected, so no question of giant emergence here
- **Interesting questions:** Degree distribution, local neighborhood structure

Theorem: Degree distribution of PAM

Let $P_k(n) = \frac{\#\text{vertices of degree } k}{n}$. Fix $m \geqslant 1$. Then

$$\mathbb{P}\left(\max_k |P_k(n) - p_k| \geqslant C\sqrt{\frac{\log n}{n}}\right) \to 0, \quad \text{as } n \to \infty,$$

Bollobás, Riordan, Spencer and Tusnády (2001)

Preferential attachment model properties

- Graph is always connected, so no question of giant emergence here
- **Interesting questions:** Degree distribution, local neighborhood structure

Theorem: Degree distribution of PAM

Let $P_k(n) = \frac{\#\text{vertices of degree } k}{n}$. Fix $m \geqslant 1$. Then

$$\mathbb{P}\left(\max_k |P_k(n) - p_k| \geqslant C\sqrt{\frac{\log n}{n}}\right) \to 0, \quad \text{as } n \to \infty,$$

where

$$p_k = ck^{-3}(1 + O(1/k))$$

Bollobás, Riordan, Spencer and Tusnády (2001)

Preferential attachment model properties

- Graph is always connected, so no question of giant emergence here
- **Interesting questions:** Degree distribution, local neighborhood structure

Theorem: Degree distribution of PAM

Let $P_k(n) = \frac{\#\text{vertices of degree } k}{n}$. Fix $m \geqslant 1$. Then

$$\mathbb{P}\left(\max_k |P_k(n) - p_k| \geqslant C\sqrt{\frac{\log n}{n}}\right) \to 0, \quad \text{as } n \to \infty,$$

where

$$p_k = ck^{-3}(1 + O(1/k))$$

Preferential Attachment produces networks with Power-law degrees

---

Bollobás, Riordan, Spencer and Tusnády (2001)

Preferential attachment model properties

- Graph is always connected, so no question of giant emergence here
- **Interesting questions:** Degree distribution, local neighborhood structure

Theorem: Degree distribution of PAM

Let $P_k(n) = \frac{\#\text{vertices of degree } k}{n}$. Fix $m \geqslant 1$. Then

$$\mathbb{P}\left(\max_k |P_k(n) - p_k| \geqslant C\sqrt{\frac{\log n}{n}}\right) \to 0, \quad \text{as } n \to \infty,$$

where

$$p_k = ck^{-3}(1 + O(1/k))$$

Preferential Attachment produces networks with Power-law degrees

- Proof relies on Martingale arguments and Azuma-Hoeffding's inequality

---

Bollobás, Riordan, Spencer and Tusnády (2001)

Summary: Random graph models with realistic features

Global communities: Stochastic Block Model

- Model with k communities of sizes $n\rho_i$, and edge probabilities depend on communities
- Local neighborhoods are mixed Poisson Branching Processes

Summary: Random graph models with realistic features

Global communities: Stochastic Block Model

- Model with k communities of sizes $n\rho_i$, and edge probabilities depend on communities
- Local neighborhoods are mixed Poisson Branching Processes

Degree-heterogeneity: Configuration Model

- Given degrees, pair half-edges uniformly
- Local neighborhoods are explored with a size-biased distribution

Summary: Random graph models with realistic features

Global communities: Stochastic Block Model

- Model with k communities of sizes $n\rho_i$, and edge probabilities depend on communities
- Local neighborhoods are mixed Poisson Branching Processes

Degree-heterogeneity: Configuration Model

- Given degrees, pair half-edges uniformly
- Local neighborhoods are explored with a size-biased distribution

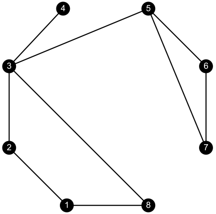
Dynamically evolving graphs: Preferential Attachment Model

- Vertices arrive sequentially and connects to vertices depending on degrees
- Leads to power-law degree distribution

Next lets study Percolation problem and its relation to Epidemic threshold

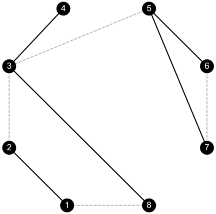
Percolation on finite networks

Percolation: Given graph G, keep each edge w.p. p independently



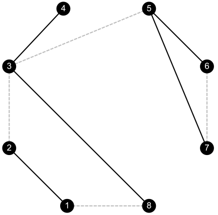
Percolation on finite networks

Percolation: Given graph G, keep each edge w.p. p independently



Percolation on finite networks

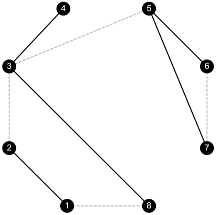
Percolation: Given graph G, keep each edge w.p. p independently



➢ If G is complete graph then the percolated graph is Erdős-Rényi

Percolation on finite networks

Percolation: Given graph G, keep each edge w.p. p independently



- If G is complete graph then the percolated graph is Erdős-Rényi

Def: Percolation threshold

Let u be a uniform vertex. p_c called percolation threshold on $(G_n)_{n\geqslant 1}$ if for any $\varepsilon > 0$

- For $p < p_c(1-\varepsilon)$: $\frac{C(u)}{n} \xrightarrow{\mathbb{P}} 0$
- For $p > p_c(1+\varepsilon)$: $\frac{C(u)}{n} = \Theta(1)$ whp

Percolation and Epidemics

SIR infection model:

➢ An infected node spreads infection to its neighbor w.p. p

➢ Infected nodes are removed after one round

Percolation and Epidemics

SIR infection model:

➢ An infected node spreads infection to its neighbor w.p. p

➢ Infected nodes are removed after one round

➢ Same mechanism applies to spread of information in social networks or spread of self-replicating malware in computer systems

Percolation and Epidemics

SIR infection model:

➢ An infected node spreads infection to its neighbor w.p. p

➢ Infected nodes are removed after one round

➢ Same mechanism applies to spread of information in social networks or spread of self-replicating malware in computer systems

Coupling between SIR model and Percolation:

Percolation and Epidemics

SIR infection model:

➢ An infected node spreads infection to its neighbor w.p. p

➢ Infected nodes are removed after one round

➢ Same mechanism applies to spread of information in social networks or spread of self-replicating malware in computer systems

Coupling between SIR model and Percolation:

➡ Suppose infection starts at vertex v

Percolation and Epidemics

SIR infection model:

➢ An infected node spreads infection to its neighbor w.p. p

➢ Infected nodes are removed after one round

➢ Same mechanism applies to spread of information in social networks or spread of self-replicating malware in computer systems

Coupling between SIR model and Percolation:

- ➡ Suppose infection starts at vertex v
- ➡ Can spread through an edge with probability p

Percolation and Epidemics

SIR infection model:

➢ An infected node spreads infection to its neighbor w.p. p

➢ Infected nodes are removed after one round

➢ Same mechanism applies to spread of information in social networks or spread of self-replicating malware in computer systems

Coupling between SIR model and Percolation:

- ➡ Suppose infection starts at vertex v
- ➡ Can spread through an edge with probability p
- ➡ Infection spread is same as exploration on percolated graph so that $C(v)$ in percolated graph equals the size of finally infected vertices

Percolation and Epidemics

SIR infection model:

➢ An infected node spreads infection to its neighbor w.p. p

➢ Infected nodes are removed after one round

➢ Same mechanism applies to spread of information in social networks or spread of self-replicating malware in computer systems

Coupling between SIR model and Percolation:

- Suppose infection starts at vertex v
- Can spread through an edge with probability p
- Infection spread is same as exploration on percolated graph so that $C(v)$ in percolated graph equals the size of finally infected vertices
- $C(v) \approx \zeta n$ whp $\iff$ Infection from v spreads to $\approx \zeta n$ population whp

Percolation and Epidemics

SIR infection model:

➢ An infected node spreads infection to its neighbor w.p. p

➢ Infected nodes are removed after one round

➢ Same mechanism applies to spread of information in social networks or spread of self-replicating malware in computer systems

Coupling between SIR model and Percolation:

- ➡ Suppose infection starts at vertex v
- ➡ Can spread through an edge with probability p
- ➡ Infection spread is same as exploration on percolated graph so that $C(v)$ in percolated graph equals the size of finally infected vertices
- ➡ $C(v) \approx \zeta n$ whp $\iff$ Infection from v spreads to $\approx \zeta n$ population whp

Finding epidemic threshold is same as finding percolation threshold...

Percolation threshold on general graphs

➢ For general graphs: Draief, Ganesh, Massoulié (2006)

Theorem

Suppose G is a connected graph. Let $\lambda_1(A)$ denote largest eigenvalue of adjacency matrix A.

Percolation threshold on general graphs

➢ For general graphs: Draief, Ganesh, Massoulié (2006)

Theorem

Suppose G is a connected graph. Let $\lambda_1(A)$ denote largest eigenvalue of adjacency matrix A. For $p < \frac{1}{\lambda_1(A)}$:

$$\mathbb{E}[C(v)] \leqslant \frac{\sqrt{n}}{1 - p\lambda_1(A)} \quad \text{for any vertex } v \text{ and any graph } G$$

Percolation threshold on general graphs

➢ For general graphs: Draief, Ganesh, Massoulié (2006)

Theorem

Suppose G is a connected graph. Let $\lambda_1(A)$ denote largest eigenvalue of adjacency matrix A. For $p < \frac{1}{\lambda_1(A)}$:

$$\mathbb{E}[C(v)] \leqslant \frac{\sqrt{n}}{1 - p\lambda_1(A)} \quad \text{for any vertex } v \text{ and any graph } G$$

$$\implies p_c \geqslant \frac{1}{\lambda_1(A)}$$

Percolation threshold on general graphs

➢ For general graphs: Draief, Ganesh, Massoulié (2006)

Theorem

Suppose G is a connected graph. Let $\lambda_1(A)$ denote largest eigenvalue of adjacency matrix A. For $p < \frac{1}{\lambda_1(A)}$:

$$\mathbb{E}[C(v)] \leqslant \frac{\sqrt{n}}{1 - p\lambda_1(A)} \quad \text{for any vertex } v \text{ and any graph } G$$

$$\implies p_c \geqslant \frac{1}{\lambda_1(A)}$$

Proof: Using Path counting. On Board

Percolation threshold on general dense graphs

➢ Is $p_c = \frac{1}{\lambda_1(A)}$?

Percolation threshold on general dense graphs

➢ Is $p_c = \frac{1}{\lambda_1(A)}$? *Yes, for dense graphs.*

See: Bollobás, Borgs, Chayes, Riordan (2010)

Percolation threshold on general dense graphs

- Is $p_c = \frac{1}{\lambda_1(A)}$? *Yes, for dense graphs.*
- Suppose $(G_n)_{n \geqslant 1}$ is a dense sequence of graphs ($\liminf \frac{\#\text{edges}}{n^2} > 0$)

See: Bollobás, Borgs, Chayes, Riordan (2010)

Percolation threshold on general dense graphs

- Is $p_c = \frac{1}{\lambda_1(A)}$? *Yes, for dense graphs*.

- Suppose $(G_n)_{n \geqslant 1}$ is a dense sequence of graphs ($\liminf \frac{\#\text{edges}}{n^2} > 0$)

Theorem

Consider percolation on G_n w.p. $p = \min\{\frac{c}{\lambda_1(A)}, 1\}$.

See: Bollobás, Borgs, Chayes, Riordan (2010)

Percolation threshold on general dense graphs

➢ Is $p_c = \frac{1}{\lambda_1(A)}$? *Yes, for dense graphs.*

➢ Suppose $(G_n)_{n \geqslant 1}$ is a dense sequence of graphs ($\liminf \frac{\#\text{edges}}{n^2} > 0$)

Theorem

Consider percolation on G_n w.p. $p = \min\{\frac{c}{\lambda_1(A)}, 1\}$. Then

1. If $c \leqslant 1$, then $\frac{C_{(1)}}{n} \xrightarrow{\mathbb{P}} 0$
2. If $c > 1$, then $\frac{C_{(1)}}{n} = \Theta(1)$ whp

See: Bollobás, Borgs, Chayes, Riordan (2010)

Percolation threshold on general dense graphs

➢ Is $p_c = \frac{1}{\lambda_1(A)}$? *Yes, for dense graphs.*

➢ Suppose $(G_n)_{n\geqslant 1}$ is a dense sequence of graphs ($\liminf \frac{\#\text{edges}}{n^2} > 0$)

Theorem

Consider percolation on G_n w.p. $p = \min\{\frac{c}{\lambda_1(A)}, 1\}$. Then

1. If $c \leqslant 1$, then $\frac{C_{(1)}}{n} \xrightarrow{\mathbb{P}} 0$
2. If $c > 1$, then $\frac{C_{(1)}}{n} = \Theta(1)$ whp

However, for sparse graphs, $\frac{1}{\lambda_1(A)}$ is not the right threshold...

See: Bollobás, Borgs, Chayes, Riordan (2010)

Percolation threshold on sparse random graphs

Fact: For any connected graph G

$$\max\left\{\frac{1}{n}\sum_i d_i, \sqrt{d_{\max}}\right\} \leqslant \lambda_1(A) \leqslant d_{\max}$$

Percolation threshold on sparse random graphs

Fact: For any connected graph G

$$\max\left\{\frac{1}{n}\sum_i d_i, \sqrt{d_{max}}\right\} \leqslant \lambda_1(A) \leqslant d_{max}$$

$\implies \frac{1}{\lambda_1(A)} \to 0$ for sparse graphs (often like n^{-c} for power-law networks), but 'often' *percolation threshold is* $\Theta(1)$

Percolation threshold on sparse random graphs

Fact: For any connected graph G

$$\max\left\{\frac{1}{n}\sum_i d_i, \sqrt{d_{\max}}\right\} \leqslant \lambda_1(A) \leqslant d_{\max}$$

$\Longrightarrow \frac{1}{\lambda_1(A)} \to 0$ for sparse graphs (often like n^{-c} for power-law networks), but 'often' *percolation threshold is* $\Theta(1)$

Example:

1. For $\mathrm{ER}_n(\frac{\lambda}{n})$: $p_c = \frac{1}{\lambda}$
2. For $\mathrm{CM}_n(\mathbf{d})$: $p_c = \frac{\mathbb{E}[D]}{\mathbb{E}[D(D-1)]}$ (under regularity conditions on $\mathbf{d}$)

Percolation threshold on sparse random graphs

Fact: For any connected graph G

$$\max\left\{\frac{1}{n}\sum_i d_i, \sqrt{d_{\max}}\right\} \leqslant \lambda_1(A) \leqslant d_{\max}$$

$\implies \frac{1}{\lambda_1(A)} \to 0$ for sparse graphs (often like n^{-c} for power-law networks), but 'often' *percolation threshold is* $\Theta(1)$

Example:

1. For $\mathrm{ER}_n(\frac{\lambda}{n})$: $p_c = \frac{1}{\lambda}$
2. For $\mathrm{CM}_n(\mathbf{d})$: $p_c = \frac{\mathbb{E}[D]}{\mathbb{E}[D(D-1)]}$ (under regularity conditions on $\mathbf{d}$)

➢ For general sparse graphs, percolation on G is always viewed as a random graph.

Percolation threshold on sparse random graphs

Fact: For any connected graph G

$$\max\left\{\frac{1}{n}\sum_i d_i, \sqrt{d_{max}}\right\} \leqslant \lambda_1(A) \leqslant d_{max}$$

$\Longrightarrow$ $\frac{1}{\lambda_1(A)} \to 0$ for sparse graphs (often like n^{-c} for power-law networks), but 'often' *percolation threshold is* $\Theta(1)$

Example:

1. For $\mathrm{ER}_n(\frac{\lambda}{n})$: $p_c = \frac{1}{\lambda}$
2. For $\mathrm{CM}_n(\mathbf{d})$: $p_c = \frac{\mathbb{E}[D]}{\mathbb{E}[D(D-1)]}$ (under regularity conditions on $\mathbf{d}$)

➢ For general sparse graphs, percolation on G is always viewed as a random graph. So, percolation threshold can be obtained by verifying

1. *The percolated graph converges locally weakly*
2. *Two large components intersect*

Finally, lets conclude with a fascinating technique that combines Random Graph theory and convergence of Stochastic Process

Back to Erdős-Rényi

Erdős-Rényi (1960) showed for $\mathrm{ER}_n(\frac{\lambda}{n})$:

- For $\lambda < 1$: $C_{(1)} = O(\log n)$ whp
- For $\lambda > 1$: $\frac{C_{(1)}}{n} \xrightarrow{\mathbb{P}} \zeta_\lambda$

Back to Erdős-Rényi

Erdős-Rényi (1960) showed for $\mathrm{ER}_n(\frac{\lambda}{n})$:

- For $\lambda < 1$: $C_{(1)} = O(\log n)$ whp
- For $\lambda > 1$: $\frac{C_{(1)}}{n} \xrightarrow{\mathbb{P}} \zeta_\lambda$
- For $\lambda = 1$: $C_{(1)} \sim n^{2/3}$, but also something very different happens...

Aldous (1997)

Back to Erdős-Rényi

Erdős-Rényi (1960) showed for $\mathrm{ER}_n(\frac{\lambda}{n})$:

- For $\lambda < 1$: $\mathcal{C}_{(1)} = O(\log n)$ whp
- For $\lambda > 1$: $\frac{\mathcal{C}_{(1)}}{n} \xrightarrow{\mathbb{P}} \zeta_\lambda$
- For $\lambda = 1$: $\mathcal{C}_{(1)} \sim n^{2/3}$, but also something very different happens...
    - ➡ All components has $\mathcal{C}_{(1)} \sim n^{2/3}$, $\mathcal{C}_{(2)} \sim n^{2/3}$, $\mathcal{C}_{(3)} \sim n^{2/3}$ ...

Aldous (1997)

Back to Erdős-Rényi

Erdős-Rényi (1960) showed for $\mathrm{ER}_n(\frac{\lambda}{n})$:

- For $\lambda < 1$: $C_{(1)} = O(\log n)$ whp
- For $\lambda > 1$: $\frac{C_{(1)}}{n} \xrightarrow{\mathbb{P}} \zeta_\lambda$
- For $\lambda = 1$: $C_{(1)} \sim n^{2/3}$, but also something very different happens...
  - ➡ All components has $C_{(1)} \sim n^{2/3}$, $C_{(2)} \sim n^{2/3}$, $C_{(3)} \sim n^{2/3} \dots$
  - ➡ Limit of component sizes are non-degenerate random variable

Aldous (1997)

Back to Erdős-Rényi

Erdős-Rényi (1960) showed for $\mathrm{ER}_n(\frac{\lambda}{n})$:

- For $\lambda < 1$: $C_{(1)} = O(\log n)$ whp
- For $\lambda > 1$: $\frac{C_{(1)}}{n} \xrightarrow{\mathbb{P}} \zeta_\lambda$
- For $\lambda = 1$: $C_{(1)} \sim n^{2/3}$, but also something very different happens...
  - All components has $C_{(1)} \sim n^{2/3}$, $C_{(2)} \sim n^{2/3}$, $C_{(3)} \sim n^{2/3}$ ...
  - Limit of component sizes are non-degenerate random variable

Theorem: Critical $\mathrm{ER}_n(\frac{\lambda}{n})$

For $\lambda = 1$:

$$n^{-2/3}(C_{(i)})_{i \geqslant 1} \xrightarrow{d} X \quad \text{in } \ell^2$$

Description of X will be clear soon...

Aldous (1997)

Exploration process method

Key Idea:

➡ Explore graph and encode component sizes in terms of a walk

Exploration process method

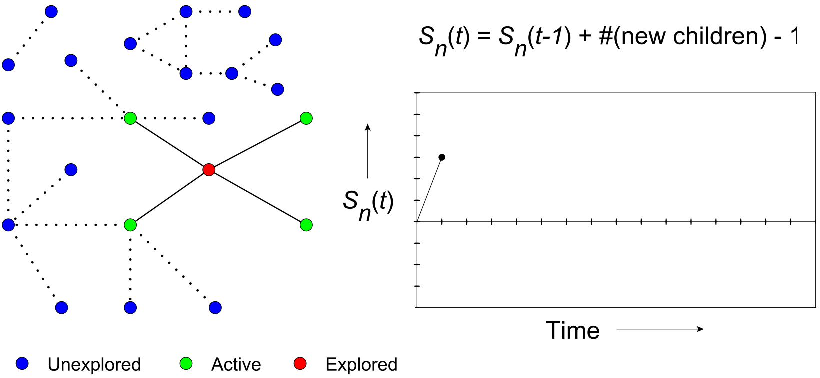
Key Idea:

- ➡ Explore graph and encode component sizes in terms of a walk
- ➡ Take scaling limits of the walk
- ➡ Recover limits of component sizes from the limiting process

Exploration process method

Key Idea:

- ➡ Explore graph and encode component sizes in terms of a walk
- ➡ Take scaling limits of the walk
- ➡ Recover limits of component sizes from the limiting process

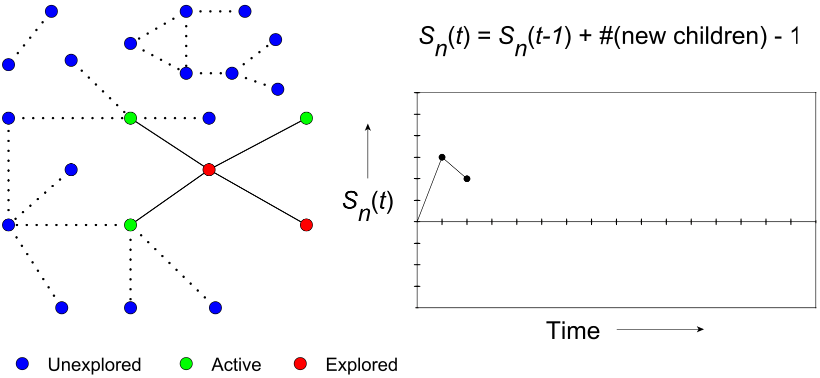


➢ Component sizes are *excursion lengths* of S_n

Exploration process method

Key Idea:

- ➡ Explore graph and encode component sizes in terms of a walk
- ➡ Take scaling limits of the walk
- ➡ Recover limits of component sizes from the limiting process

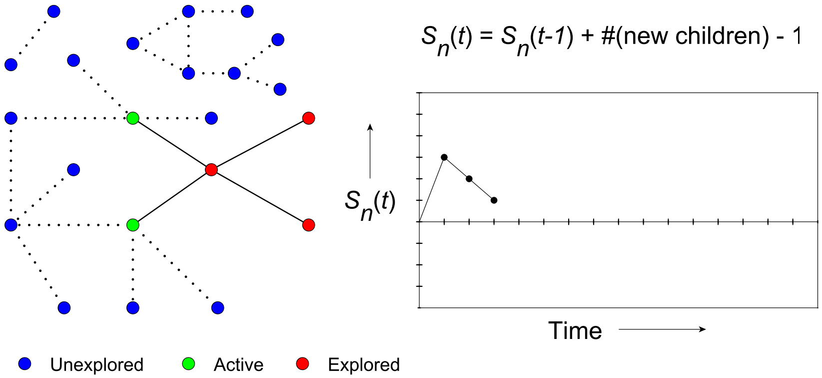


➢ Component sizes are *excursion lengths* of S_n

Exploration process method

Key Idea:

- ➡ Explore graph and encode component sizes in terms of a walk
- ➡ Take scaling limits of the walk
- ➡ Recover limits of component sizes from the limiting process

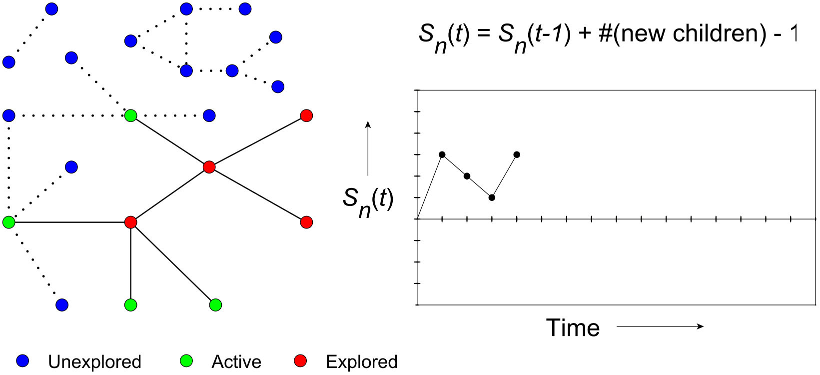


➢ Component sizes are *excursion lengths* of S_n

Exploration process method

Key Idea:

- ➡ Explore graph and encode component sizes in terms of a walk
- ➡ Take scaling limits of the walk
- ➡ Recover limits of component sizes from the limiting process

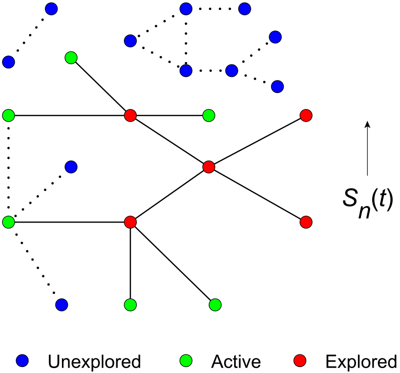


➢ Component sizes are *excursion lengths* of S_n

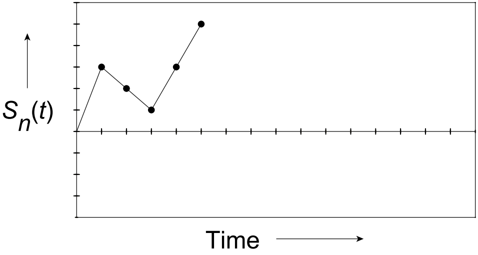
Exploration process method

Key Idea:

- ➡ Explore graph and encode component sizes in terms of a walk
- ➡ Take scaling limits of the walk
- ➡ Recover limits of component sizes from the limiting process



$S_n(t) = S_n(t\text{-}1)$ + #(new children) - 1



➢ Component sizes are *excursion lengths* of S_n

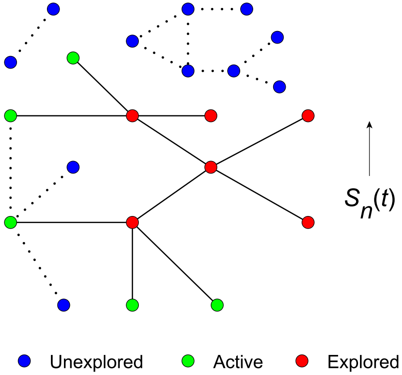
Exploration process method

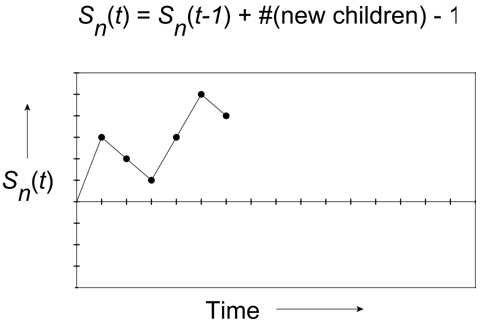
Key Idea:

➡ Explore graph and encode component sizes in terms of a walk

➡ Take scaling limits of the walk

➡ Recover limits of component sizes from the limiting process





➢ Component sizes are *excursion lengths* of S_n

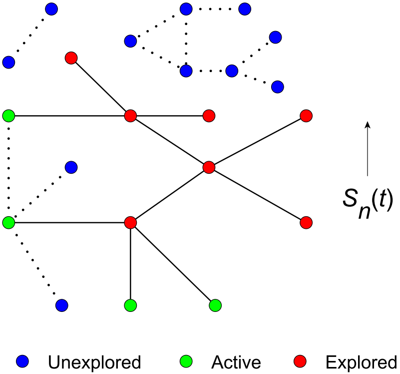
Exploration process method

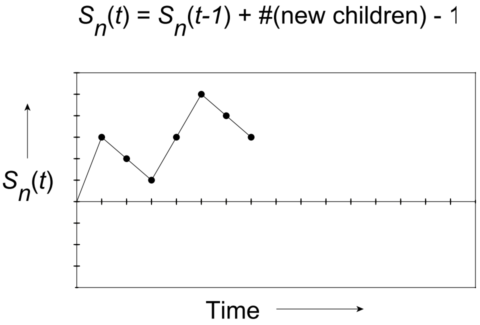
Key Idea:

➡ Explore graph and encode component sizes in terms of a walk

➡ Take scaling limits of the walk

➡ Recover limits of component sizes from the limiting process



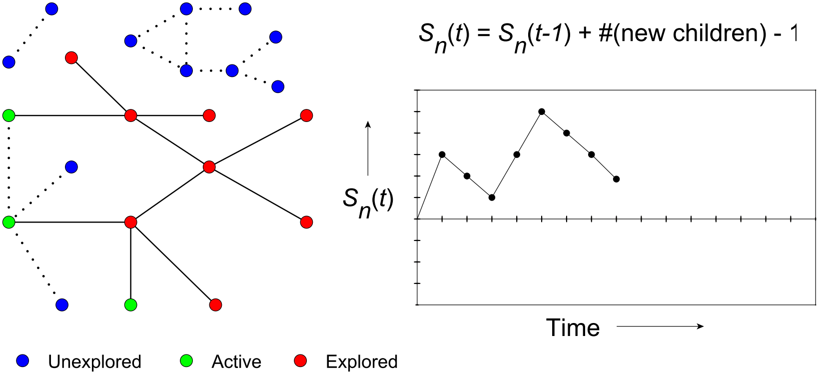


➢ Component sizes are *excursion lengths* of S_n

Exploration process method

Key Idea:

- ➡ Explore graph and encode component sizes in terms of a walk
- ➡ Take scaling limits of the walk
- ➡ Recover limits of component sizes from the limiting process

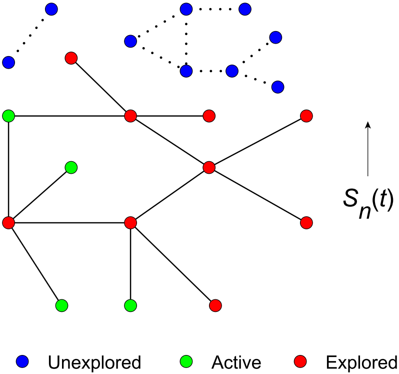


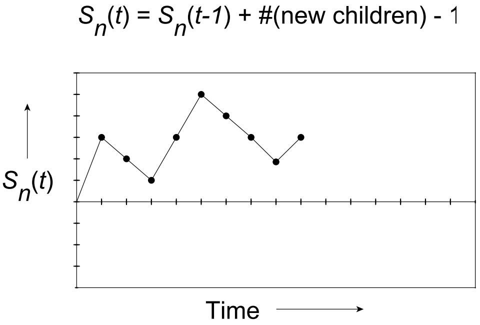
➢ Component sizes are *excursion lengths* of S_n

Exploration process method

Key Idea:

- ➡ Explore graph and encode component sizes in terms of a walk
- ➡ Take scaling limits of the walk
- ➡ Recover limits of component sizes from the limiting process



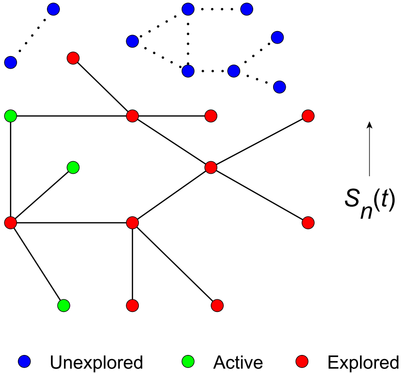


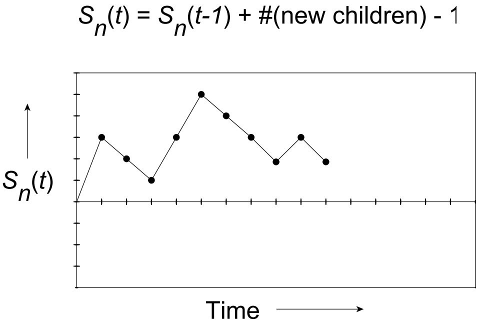
➢ Component sizes are *excursion lengths* of S_n

Exploration process method

Key Idea:

- ➡ Explore graph and encode component sizes in terms of a walk
- ➡ Take scaling limits of the walk
- ➡ Recover limits of component sizes from the limiting process



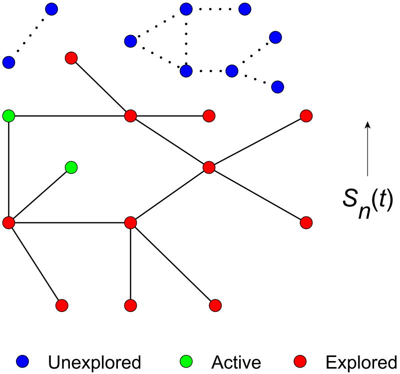


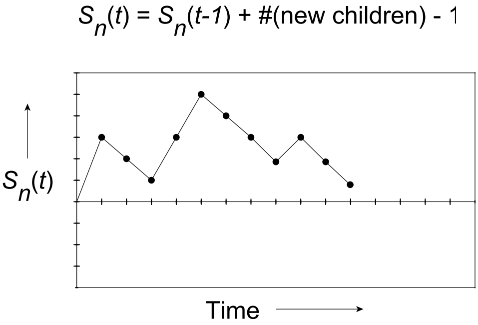
➢ Component sizes are *excursion lengths* of S_n

Exploration process method

Key Idea:

- ➡ Explore graph and encode component sizes in terms of a walk
- ➡ Take scaling limits of the walk
- ➡ Recover limits of component sizes from the limiting process



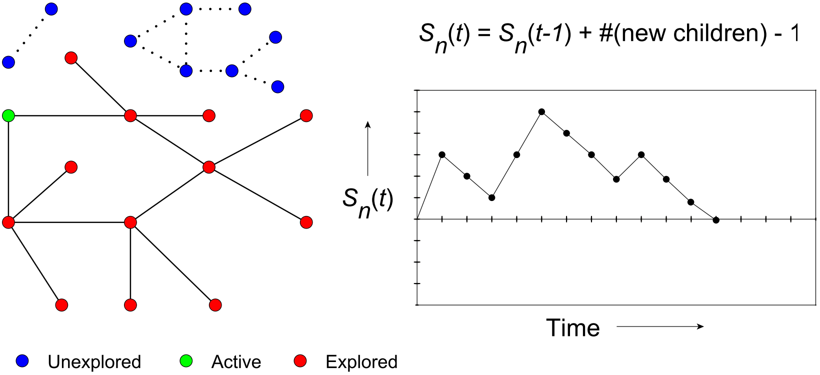


➢ Component sizes are *excursion lengths* of S_n

Exploration process method

Key Idea:

- ➡ Explore graph and encode component sizes in terms of a walk
- ➡ Take scaling limits of the walk
- ➡ Recover limits of component sizes from the limiting process

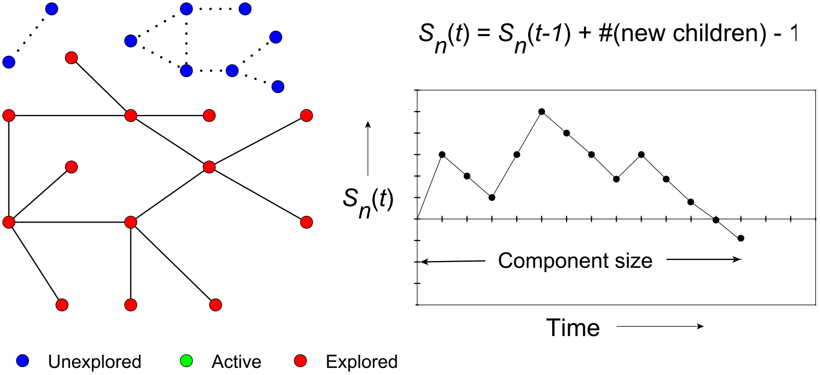


➢ Component sizes are *excursion lengths* of S_n

Exploration process method

Key Idea:

- ➡ Explore graph and encode component sizes in terms of a walk
- ➡ Take scaling limits of the walk
- ➡ Recover limits of component sizes from the limiting process

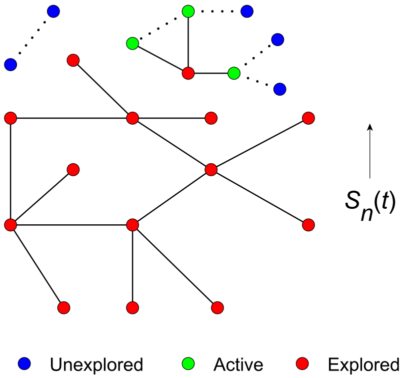


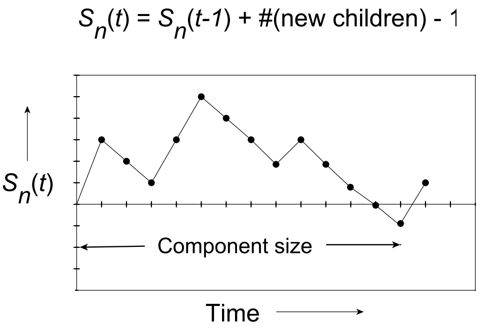
➢ Component sizes are *excursion lengths* of S_n

Exploration process method

Key Idea:

- ➡ Explore graph and encode component sizes in terms of a walk
- ➡ Take scaling limits of the walk
- ➡ Recover limits of component sizes from the limiting process





➢ Component sizes are *excursion lengths* of S_n

Limit of the exploration process

Proposition: $(n^{-1/3} S_n(tn^{2/3}) : t \geqslant 0) \xrightarrow{d} \left(B(t) - \frac{t^2}{2} : t \geqslant 0\right)$

Limit of the exploration process

Proposition: $(n^{-1/3} S_n(tn^{2/3}) : t \geqslant 0) \xrightarrow{d} \left(B(t) - \frac{t^2}{2} : t \geqslant 0\right)$

➢ Proof uses Martingale Functional CLT.

Limit of the exploration process

Proposition: $(n^{-1/3} S_n(tn^{2/3}) : t \geqslant 0) \xrightarrow{d} \left(B(t) - \frac{t^2}{2} : t \geqslant 0\right)$

➢ Proof uses Martingale Functional CLT.

Heuristic: component sizes are excursion lengths of S_n,

Limit of the exploration process

Proposition: $(n^{-1/3} S_n(tn^{2/3}) : t \geqslant 0) \xrightarrow{d} \left(B(t) - \frac{t^2}{2} : t \geqslant 0\right)$

➢ Proof uses Martingale Functional CLT.

Heuristic: component sizes are excursion lengths of S_n, so excursion lengths of $n^{-1/3} S_n(tn^{2/3})$ gives us $n^{-2/3} \times$ comp. size

Limit of the exploration process

Proposition: $(n^{-1/3}S_n(tn^{2/3}) : t \geqslant 0) \xrightarrow{d} \left(B(t) - \frac{t^2}{2} : t \geqslant 0\right)$

➢ Proof uses Martingale Functional CLT.

Heuristic: component sizes are excursion lengths of S_n, so excursion lengths of $n^{-1/3}S_n(tn^{2/3})$ gives us $n^{-2/3}\times$ comp. size

$$\implies n^{-2/3}(C_{(i)})_{i\geqslant 1} \xrightarrow{d} (\gamma_i)_{i\geqslant 1},$$

where γ_i is the i-th largest excursion of $\left(B(t) - \frac{t^2}{2} : t \geqslant 0\right)$

Limit of the exploration process

Proposition: $(n^{-1/3}S_n(tn^{2/3}) : t \geqslant 0) \xrightarrow{d} \left(B(t) - \frac{t^2}{2} : t \geqslant 0\right)$

➢ Proof uses Martingale Functional CLT.

Heuristic: component sizes are excursion lengths of S_n, so excursion lengths of $n^{-1/3}S_n(tn^{2/3})$ gives us $n^{-2/3}\times$ comp. size

$$\implies n^{-2/3}(C_{(i)})_{i\geqslant 1} \xrightarrow{d} (\gamma_i)_{i\geqslant 1},$$

where γ_i is the i-th largest excursion of $\left(B(t) - \frac{t^2}{2} : t \geqslant 0\right)$

Limit of exploration process gives limit of comp. sizes

Exploration process method contd.

Revisiting the method:

Exploration process method contd.

Revisiting the method:

- ➡ Explore graph and encode component sizes in terms of a walk

Exploration process method contd.

Revisiting the method:

- ➡ Explore graph and encode component sizes in terms of a walk
- ➡ Take scaling limits of the walk

Exploration process method contd.

Revisiting the method:

- Explore graph and encode component sizes in terms of a walk
- Take scaling limits of the walk
- Recover limits of component sizes from the limiting process

Exploration process method contd.

Revisiting the method:

- ➡ Explore graph and encode component sizes in terms of a walk
- ➡ Take scaling limits of the walk
- ➡ Recover limits of component sizes from the limiting process

➢ Method also works for supercritical case. In that case the limit is deterministic. See Janson & Luczak (2007)

Summary

Percolation and Epidemics

Summary

Percolation and Epidemics

- Percolation can be coupled to SIR epidemics

Summary

Percolation and Epidemics

- Percolation can be coupled to SIR epidemics
- Showed $\mathbb{E}[C(u)] \leqslant \frac{\sqrt{n}}{1-p\lambda_1(A)}$ for $p < \frac{1}{\lambda_1(A)}$ for any connected graph

Summary

Percolation and Epidemics

- Percolation can be coupled to SIR epidemics
- Showed $\mathbb{E}[C(u)] \leqslant \frac{\sqrt{n}}{1-p\lambda_1(A)}$ for $p < \frac{1}{\lambda_1(A)}$ for any connected graph
- $p_c = \frac{1}{\lambda_1(A)}$ for dense graph but not for sparse graphs

Summary

Percolation and Epidemics

- Percolation can be coupled to SIR epidemics
- Showed $\mathbb{E}[C(u)] \leqslant \frac{\sqrt{n}}{1-p\lambda_1(A)}$ for $p < \frac{1}{\lambda_1(A)}$ for any connected graph
- $p_c = \frac{1}{\lambda_1(A)}$ for dense graph but not for sparse graphs

Exploration process convergence

Summary

Percolation and Epidemics

- Percolation can be coupled to SIR epidemics
- Showed $\mathbb{E}[C(u)] \leqslant \frac{\sqrt{n}}{1-p\lambda_1(A)}$ for $p < \frac{1}{\lambda_1(A)}$ for any connected graph
- $p_c = \frac{1}{\lambda_1(A)}$ for dense graph but not for sparse graphs

Exploration process convergence

- Used it to find non-degenerate limits of component sizes for $ER_n(\frac{\lambda}{n})$ with $\lambda = 1$

Further reading

Emergence of Giant and Random Graph Models

1. van der Hofstad: *Random graphs and complex networks Vol 1, Vol 2*
2. van der Hofstad: *The giant in random graphs is almost local*

Percolation and Epidemics

1. Draief, Ganesh, Massoulié: *Thresholds for virus spread on networks*

Critical behavior

1. Aldous: *Brownian excursions, critical random graphs and the multiplicative coalescent*
2. Dhara: *Doctoral thesis, Critical percolation on random networks with prescribed degrees* (Chapter 1 contains survey on Critical behavior)

Further reading

Emergence of Giant and Random Graph Models

1. van der Hofstad: *Random graphs and complex networks Vol 1, Vol 2*
2. van der Hofstad: *The giant in random graphs is almost local*

Percolation and Epidemics

1. Draief, Ganesh, Massoulié: *Thresholds for virus spread on networks*

Critical behavior

1. Aldous: *Brownian excursions, critical random graphs and the multiplicative coalescent*
2. Dhara: *Doctoral thesis, Critical percolation on random networks with prescribed degrees* (Chapter 1 contains survey on Critical behavior)

Thank You!