S&DS 684: Statistical inference on graphs

Spring 2023

Administrivia

- Schedule: Tue 4-550pm, 17 Hillhouse Rm 03
- Instructor: Prof. Yihong Wu `yihongwu@yale.edu`, Rm 235 Dunham Lab (10 Hillhouse)
  - Office hours: by appointment

Administrivia

- Schedule: Tue 4-550pm, 17 Hillhouse Rm 03
- Instructor: Prof. Yihong Wu yihongwu@yale.edu, Rm 235 Dunham Lab (10 Hillhouse)
  - Office hours: by appointment
- Advanced graduate-level seminar-style class. Undergraduates need express permission from the instructor.

Administrivia

- Schedule: Tue 4-550pm, 17 Hillhouse Rm 03
- Instructor: Prof. Yihong Wu `yihongwu@yale.edu`, Rm 235 Dunham Lab (10 Hillhouse)
  - Office hours: by appointment
- Advanced graduate-level seminar-style class. Undergraduates need express permission from the instructor.
- No TF.

Administrivia

- Schedule: Tue 4-550pm, 17 Hillhouse Rm 03
- Instructor: Prof. Yihong Wu yihongwu@yale.edu, Rm 235 Dunham Lab (10 Hillhouse)
    - Office hours: by appointment
- Advanced graduate-level seminar-style class. Undergraduates need express permission from the instructor.
- No TF.
- Website: http://www.stat.yale.edu/~yw562/teaching/684/index.html

Administrivia

- Course prerequisites:

Administrivia

- Course prerequisites:
  - Maturity with probability theory and linear algebra is required

Administrivia

- Course prerequisites:
  - Maturity with probability theory and linear algebra is required
  - Familiarity with statistical theory and optimization is helpful

Administrivia

- Course prerequisites:
  - Maturity with probability theory and linear algebra is required
  - Familiarity with statistical theory and optimization is helpful
- Participation (30%):

Administrivia

- Course prerequisites:
  - Maturity with probability theory and linear algebra is required
  - Familiarity with statistical theory and optimization is helpful
- Participation (30%):
  - Attend class on time :-)

Administrivia

- Course prerequisites:
  - Maturity with probability theory and linear algebra is required
  - Familiarity with statistical theory and optimization is helpful
- Participation (30%):
  - Attend class on time :-)
  - Meaningful classroom participation

Administrivia

- Course prerequisites:
  - Maturity with probability theory and linear algebra is required
  - Familiarity with statistical theory and optimization is helpful
- Participation (30%):
  - Attend class on time :-)
  - Meaningful classroom participation
  - Proofread lecture notes and provide comments and corrections

Administrivia

- Course prerequisites:
  - Maturity with probability theory and linear algebra is required
  - Familiarity with statistical theory and optimization is helpful
- Participation (30%):
  - Attend class on time :-)
  - Meaningful classroom participation
  - Proofread lecture notes and provide comments and corrections
- Homeworks (30%): two to three problem sets

Administrivia

- Course prerequisites:
  - Maturity with probability theory and linear algebra is required
  - Familiarity with statistical theory and optimization is helpful
- Participation (30%):
  - Attend class on time :-)
  - Meaningful classroom participation
  - Proofread lecture notes and provide comments and corrections
- Homeworks (30%): two to three problem sets
- Final project (40%)

Administrivia

- Course prerequisites:
  - Maturity with probability theory and linear algebra is required
  - Familiarity with statistical theory and optimization is helpful
- Participation (30%):
  - Attend class on time :-)
  - Meaningful classroom participation
  - Proofread lecture notes and provide comments and corrections
- Homeworks (30%): two to three problem sets
- Final project (40%)
  - either presenting paper(s) or a standalone research project.

Administrivia

- Course prerequisites:
  - Maturity with probability theory and linear algebra is required
  - Familiarity with statistical theory and optimization is helpful
- Participation (30%):
  - Attend class on time :-)
  - Meaningful classroom participation
  - Proofread lecture notes and provide comments and corrections
- Homeworks (30%): two to three problem sets
- Final project (40%)
  - either presenting paper(s) or a standalone research project.
  - list of topics announced around week 6

Administrivia

- Lecture notes:
  - Yihong Wu and Jiaming Xu, "Statistical inference on graphs: Selected Topics", working draft, available at http://www.stat.yale.edu/~yw562/teaching/stats-graphs.pdf
  - Additional reading materials will be posted online.

Administrivia

- Lecture notes:
  - Yihong Wu and Jiaming Xu, "Statistical inference on graphs: Selected Topics", working draft, available at http://www.stat.yale.edu/~yw562/teaching/stats-graphs.pdf
  - Additional reading materials will be posted online.
- Highly theoretical class
  - Statistical (information-theoretical) analysis
  - Algorithms: emphasizing proof of correctness
  - No coding

Statistical problems

- Statistical tasks: using data to make informed decisions (hypotheses testing, estimation, etc)

$$\underbrace{\theta \in \Theta}_{\text{parameter}} \to \boxed{\text{Statistical model}} \to \underbrace{X}_{\text{data}} \mapsto \boxed{\text{Algorithm}} \to \underbrace{\hat{\theta}}_{\text{estimate}}$$

Includes hypothesis testing (detection) as special case.

Statistical problems

- Statistical tasks: using data to make informed decisions (hypotheses testing, estimation, etc)

$$\underbrace{\theta \in \Theta}_{\text{parameter}} \to \boxed{\text{Statistical model}} \to \underbrace{X}_{\text{data}} \mapsto \boxed{\text{Algorithm}} \to \underbrace{\hat{\theta}}_{\text{estimate}}$$

  Includes hypothesis testing (detection) as special case.
- Understanding the fundamental limits:

Statistical problems

- Statistical tasks: using data to make informed decisions (hypotheses testing, estimation, etc)

$$\underbrace{\theta \in \Theta}_{\text{parameter}} \to \boxed{\text{Statistical model}} \to \underbrace{X}_{\text{data}} \mapsto \boxed{\text{Algorithm}} \to \underbrace{\hat{\theta}}_{\text{estimate}}$$

  Includes hypothesis testing (detection) as special case.

- Understanding the fundamental limits:
  - Q1 Characterize statistical (information-theoretic) limit: What is possible/impossible?

Statistical problems

- Statistical tasks: using data to make informed decisions (hypotheses testing, estimation, etc)

$$\underbrace{\theta \in \Theta}_{\text{parameter}} \to \boxed{\text{Statistical model}} \to \underbrace{X}_{\text{data}} \mapsto \boxed{\text{Algorithm}} \to \underbrace{\hat{\theta}}_{\text{estimate}}$$

  Includes hypothesis testing (detection) as special case.

- Understanding the fundamental limits:
  - Q1 Characterize statistical (information-theoretic) limit: What is possible/impossible?
  - Q2 Can statistical limits be attained computationally efficiently, e.g., in polynomial time? If yes, how? If not, why?

Statistical problems

- Statistical tasks: using data to make informed decisions (hypotheses testing, estimation, etc)

$$\underbrace{\theta \in \Theta}_{\text{parameter}} \to \boxed{\text{Statistical model}} \to \underbrace{X}_{\text{data}} \mapsto \boxed{\text{Algorithm}} \to \underbrace{\hat{\theta}}_{\text{estimate}}$$

  Includes hypothesis testing (detection) as special case.
- Understanding the fundamental limits:
  - Q1 Characterize statistical (information-theoretic) limit: What is possible/impossible?
  - Q2 Can statistical limits be attained computationally efficiently, e.g., in polynomial time? If yes, how? If not, why?
- In this course: statistical problems of combinatorial flavor

Statistical problems

- Statistical tasks: using data to make informed decisions (hypotheses testing, estimation, etc)

$$\underbrace{\theta \in \Theta}_{\text{parameter}} \to \boxed{\text{Statistical model}} \to \underbrace{X}_{\text{data}} \mapsto \boxed{\text{Algorithm}} \to \underbrace{\hat{\theta}}_{\text{estimate}}$$

  Includes hypothesis testing (detection) as special case.
- Understanding the fundamental limits:
  - Q1 Characterize statistical (information-theoretic) limit: What is possible/impossible?
  - Q2 Can statistical limits be attained computationally efficiently, e.g., in polynomial time? If yes, how? If not, why?
- In this course: statistical problems of combinatorial flavor
  - Data = graphs (i.e. networks)

Statistical problems

- Statistical tasks: using data to make informed decisions (hypotheses testing, estimation, etc)

$$\underbrace{\theta \in \Theta}_{\text{parameter}} \to \boxed{\text{Statistical model}} \to \underbrace{X}_{\text{data}} \mapsto \boxed{\text{Algorithm}} \to \underbrace{\hat{\theta}}_{\text{estimate}}$$

  Includes hypothesis testing (detection) as special case.
- Understanding the fundamental limits:
  - Q1 Characterize statistical (information-theoretic) limit: What is possible/impossible?
  - Q2 Can statistical limits be attained computationally efficiently, e.g., in polynomial time? If yes, how? If not, why?
- In this course: statistical problems of combinatorial flavor
  - Data = graphs (i.e. networks)
  - Parameter = hidden (latent, or planted) structure, e.g., vertex labels, matching, cycle, etc

Statistical problems

- Statistical tasks: using data to make informed decisions (hypotheses testing, estimation, etc)

$$\underbrace{\theta \in \Theta}_{\text{parameter}} \to \boxed{\text{Statistical model}} \to \underbrace{X}_{\text{data}} \mapsto \boxed{\text{Algorithm}} \to \underbrace{\hat{\theta}}_{\text{estimate}}$$

  Includes hypothesis testing (detection) as special case.
- Understanding the fundamental limits:
  - Q1 Characterize statistical (information-theoretic) limit: What is possible/impossible?
  - Q2 Can statistical limits be attained computationally efficiently, e.g., in polynomial time? If yes, how? If not, why?
- In this course: statistical problems of combinatorial flavor
  - Data = graphs (i.e. networks)
  - Parameter = hidden (latent, or planted) structure, e.g., vertex labels, matching, cycle, etc
  - Focus on large-graph limit (number of vertices $\to \infty$)

Statistical problems

- Statistical tasks: using data to make informed decisions (hypotheses testing, estimation, etc)

$$\underbrace{\theta \in \Theta}_{\text{parameter}} \to \boxed{\text{Statistical model}} \to \underbrace{X}_{\text{data}} \mapsto \boxed{\text{Algorithm}} \to \underbrace{\hat{\theta}}_{\text{estimate}}$$

  Includes hypothesis testing (detection) as special case.
- Understanding the fundamental limits:
  - Q1 Characterize statistical (information-theoretic) limit: What is possible/impossible?
  - Q2 Can statistical limits be attained computationally efficiently, e.g., in polynomial time? If yes, how? If not, why?
- In this course: statistical problems of combinatorial flavor
  - Data = graphs (i.e. networks)
  - Parameter = hidden (latent, or planted) structure, e.g., vertex labels, matching, cycle, etc
  - Focus on large-graph limit (number of vertices $\to \infty$)
  - Statistical tasks: detection (null vs planted), recovery, or estimation.

Basic definitions of graphs

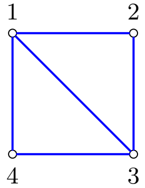
A **graph** $G = (V, E)$ consists of

- A **vertex set** $V = [n] \equiv \{1, \dots, n\}$ for some positive integer n.
- An **edge set** $E \subset \binom{V}{2}$. Each element of E is an edge $e = (i, j)$ (unordered pair). We say i and j are connected and write $i \sim j$ if $(i, j) \in E$.

We mostly focus on graphs that are undirected and simple.

Adjacency matrix representation: $A = (A_{ij})_{i,j \in [n]}$ is an $n \times n$ symmetric binary matrix with zero diagonal and

$$A_{ij} = \mathbf{1}\{i \sim j\} = \begin{cases} 1 & (i,j) \in E \\ 0 & \text{o.w.} \end{cases}.$$



$$A = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix}$$

Basic definitions of graphs

- The **neighborhood** of a given vertex $v \in V$:

$$N(v) = \{u \in V : u \sim v\}$$

- The **degree** of v:

$$d_v = |N(v)|$$

- **Induced subgraph**: For any $S \subset V$, the subgraph induced by S is $G[S] = (S, E_S)$, where

$$E_S \triangleq \{(u, v) \in E : u, v \in S\}$$

- A **clique** is a **complete** subgraph. A graph is complete iff all pairs of vertices in the graph are connected.

Basic definitions of graphs

- The **neighborhood** of a given vertex $v \in V$:

$$N(v) = \{u \in V : u \sim v\}$$

- The **degree** of v:

$$d_v = |N(v)|$$

- **Induced subgraph**: For any $S \subset V$, the subgraph induced by S is $G[S] = (S, E_S)$, where

$$E_S \triangleq \{(u, v) \in E : u, v \in S\}$$

- A **clique** is a **complete** subgraph. A graph is complete iff all pairs of vertices in the graph are connected.
- Graphs are highly useful to represent relational data, which are ubiquitous

Data represented by graphs: Social networks

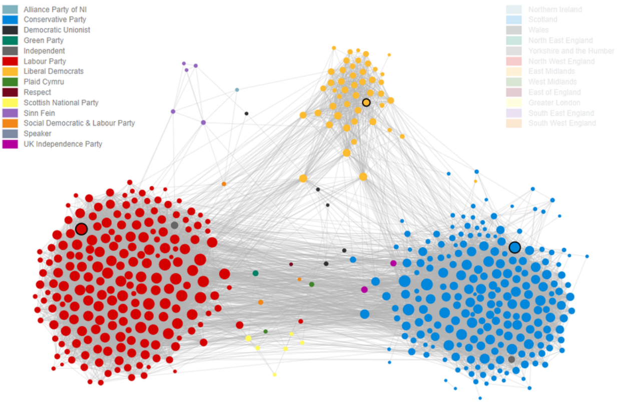


Figure: Twitter network for UK MPs circa 2015

https://www.nesta.org.uk/blog/twitter-network-uk-mps/

Data represented by graphs: Social networks

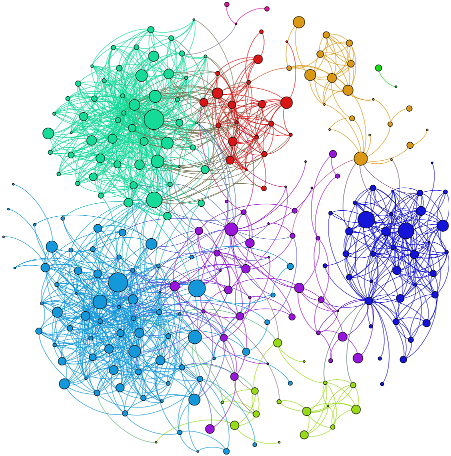


Figure: A LinkedIn network: Green=Cisco, Blue=Disney, Purple=Recruiters, etc

http://allthingsgraphed.com/2014/10/16/your-linkedin-network/

Data represented by graphs: Biological networks

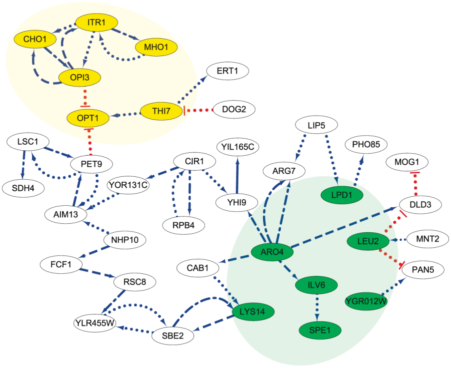


Figure: Gene regulatory network in yeast: nodes=genes, mRNA, protein, etc; edges=regulartion, expression, etc

https://www.nature.com/articles/s41598-018-37667-4

Data represented by graphs: Biological networks

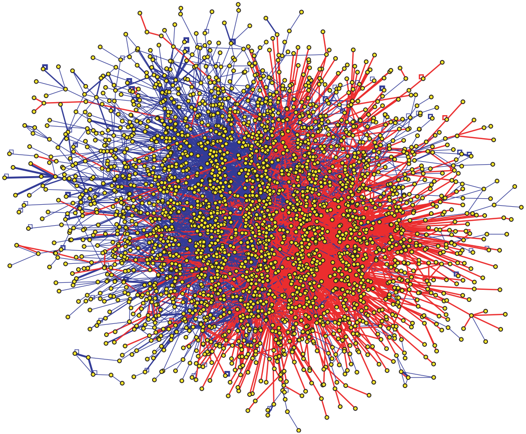
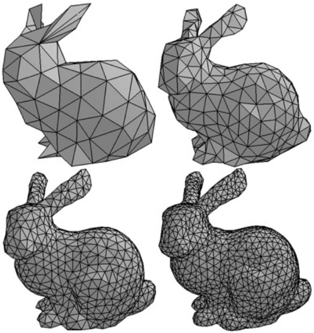


Figure: Human Protein-Protein-Interaction (PPI) network or "interactome": nodes=proteins; red edge= CCSB-HI1 interactions, blue edges= LCI interactions.

https://www.nature.com/articles/nature04209

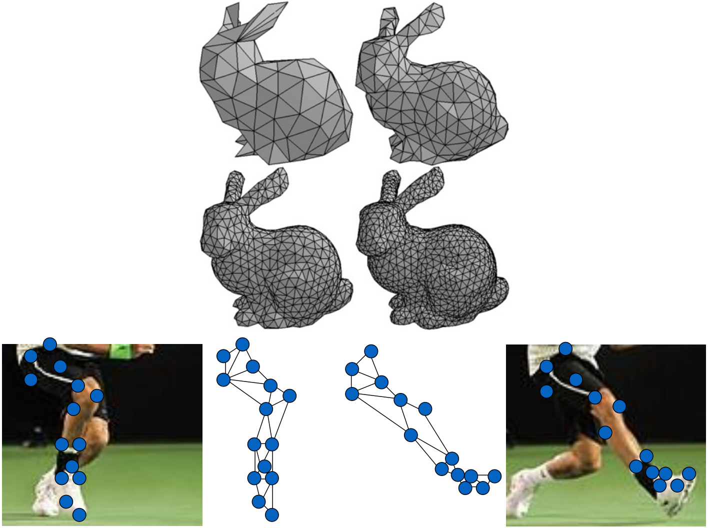
Data represented by graphs: computer vision

Graphs as discretization of geometric objects (triangulated mesh)

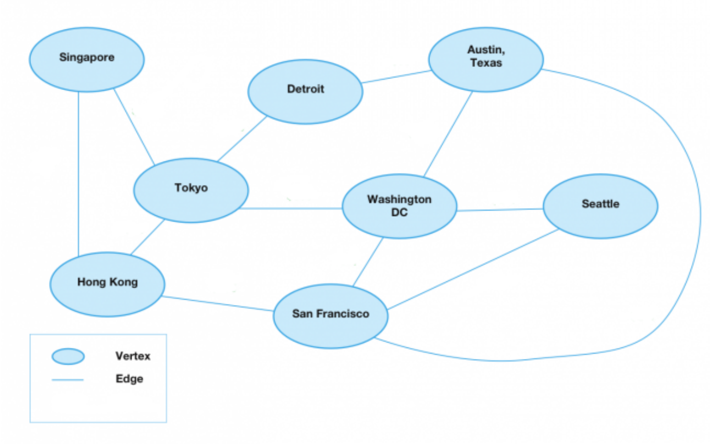


Data represented by graphs: computer vision

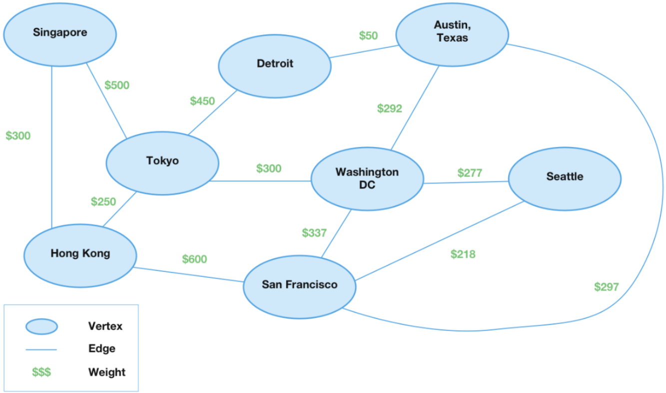
Graphs as discretization of geometric objects (triangulated mesh)



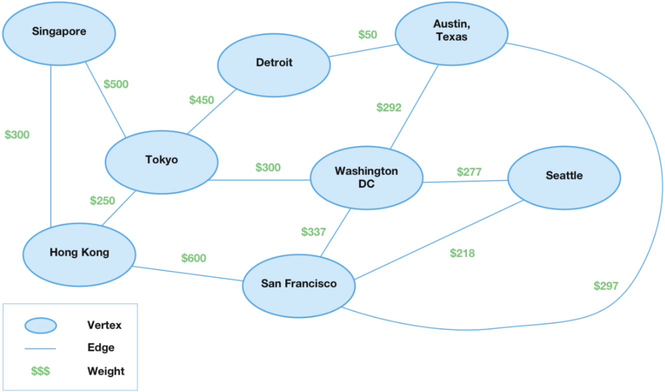
Data represented by graphs: operational research



Data represented by graphs: operational research



Data represented by graphs: operational research



We will encounter many combinatorial optimimization problems involves (weighted) graphs

Statistical model: random graphs

Common ensembles

- Erdős-Rényi graph $\mathcal{G}(n, p)$: connect each pair i and j with probability p independently

Statistical model: random graphs

Common ensembles

- Erdős-Rényi graph $\mathcal{G}(n,p)$: connect each pair i and j with probability p independently
- Erdős-Rényi graph $\mathcal{G}(n,m)$: draw $E(G)$ uniformly at random from $\binom{\binom{[n]}{2}}{m}$

Statistical model: random graphs

Common ensembles

- Erdős-Rényi graph $\mathcal{G}(n,p)$: connect each pair i and j with probability p independently
- Erdős-Rényi graph $\mathcal{G}(n,m)$: draw $E(G)$ uniformly at random from $\binom{\binom{[n]}{2}}{m}$
- Random geometric graph: draw $x_1,\ldots,x_n$ uniformly from a sphere independently; connect i and j if distance$(x_i,x_j) \le r$.

Statistical model: random graphs

Common ensembles

- Erdős-Rényi graph $\mathcal{G}(n,p)$: connect each pair i and j with probability p independently
- Erdős-Rényi graph $\mathcal{G}(n,m)$: draw $E(G)$ uniformly at random from $\binom{\binom{[n]}{2}}{m}$
- Random geometric graph: draw $x_1,\ldots,x_n$ uniformly from a sphere independently; connect i and j if distance$(x_i,x_j) \le r$.

Statistical model: random graphs

Common ensembles

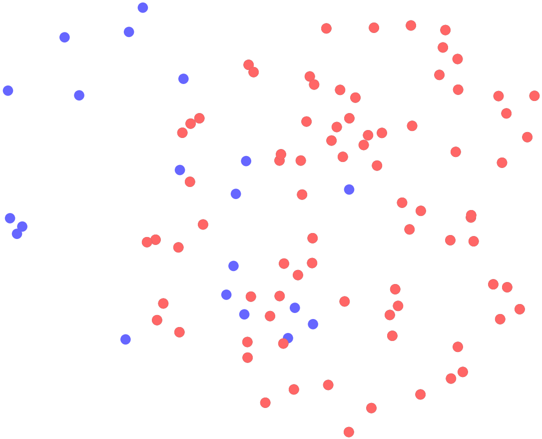
- Erdős-Rényi graph $\mathcal{G}(n, p)$: connect each pair i and j with probability p independently
- Erdős-Rényi graph $\mathcal{G}(n, m)$: draw $E(G)$ uniformly at random from $\binom{\binom{[n]}{2}}{m}$
- Random geometric graph: draw $x_1, \ldots, x_n$ uniformly from a sphere independently; connect i and j if distance$(x_i, x_j) \leq r$.
- Planted models

Vignette # 1: Planted clique

Planted clique – graph view

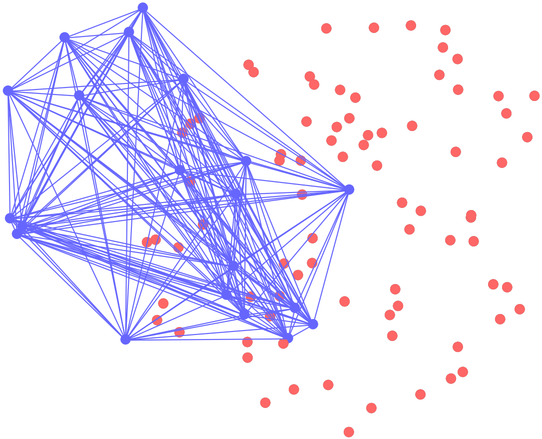
Planted clique – graph view

1. A clique of k vertices are chosen uniformly at random to form a clique



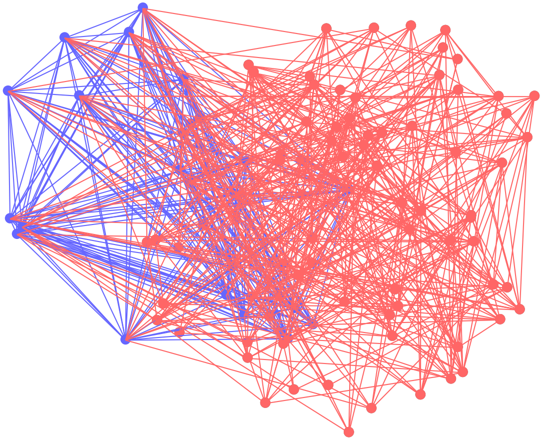
Planted clique – graph view

1. A clique of k vertices are chosen uniformly at random to form a clique



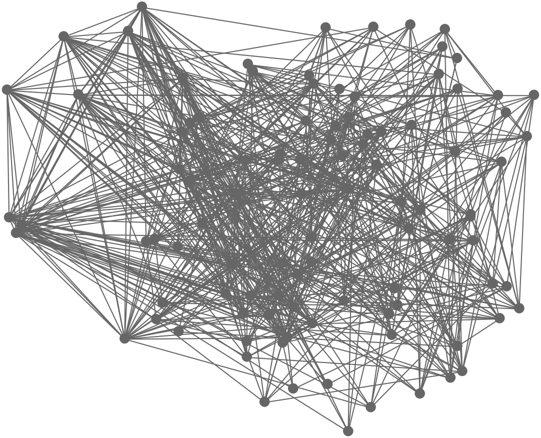
Planted clique – graph view

1. A clique of k vertices are chosen uniformly at random to form a clique
2. For every other pair of nodes, add an edge w.p. $\frac{1}{2}$



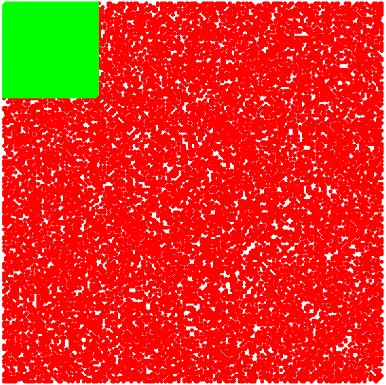
Planted clique – graph view

1. A clique of k vertices are chosen uniformly at random to form a clique
2. For every other pair of nodes, add an edge w.p. $\frac{1}{2}$

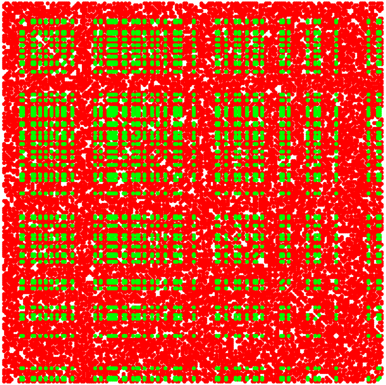


Call this $\mathcal{G}(n, k, \frac{1}{2})$, the planted clique model.

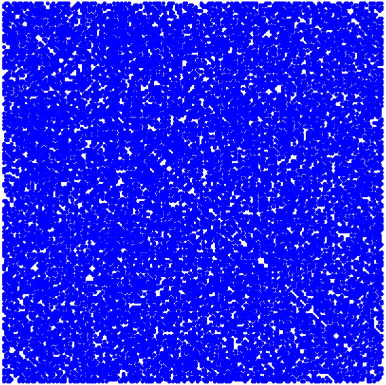
Planted clique – adjacency matrix view



Planted clique – adjacency matrix view



Planted clique – adjacency matrix view



Planted clique: detection

$$\text{Planted clique } S \longrightarrow \text{graph } G \longrightarrow \text{decision } \phi \in \{0,1\}$$

to test the following hypothesis

$$H_0 : G \sim \mathcal{G}(n, \frac{1}{2})(\text{null model}) \quad \text{vs} \quad H_1 : G \sim \mathcal{G}(n, k, \frac{1}{2})(\text{planted model})$$

- Goal:
$$\underbrace{\mathbb{P}_{G\sim\mathcal{G}(n,\frac{1}{2})}[\phi = 1]}_{\text{Type-I error}} + \underbrace{\mathbb{P}_{G\sim\mathcal{G}(n,k,\frac{1}{2})}[\phi = 0]}_{\text{Type-II error}} \to 0$$
- Question: What's the smallest clique that is detectable? How to do it fast?

Planted clique: recovery

$$\text{Planted clique } S \longrightarrow \text{graph } G \longrightarrow \text{Estimated clique } \widehat{S}$$

- Minimax framework: Find an estimator $\widehat{S} = \widehat{S}(G)$ that performs well in worst-case

$$\min_{S \in \binom{[n]}{k}} \mathbb{P}_S \left[\widehat{S}(G) = S \right] \approx 1$$

Planted clique: recovery

Planted clique $S \longrightarrow$ graph $G \longrightarrow$ Estimated clique $\widehat{S}$

- Minimax framework: Find an estimator $\widehat{S} = \widehat{S}(G)$ that performs well in worst-case

$$\min_{S \in \binom{[n]}{k}} \mathbb{P}_S \left[\widehat{S}(G) = S \right] \approx 1$$

- Bayesian framework: Find an estimator $\widehat{S} = \widehat{S}(G)$ that performs well on average

$$\mathbf{E}_{S \sim \mathsf{Unif}\left(\binom{[n]}{k}\right)} \mathbb{P}_S \left[\widehat{S}(G) = S \right] \approx 1$$

Planted clique: recovery

Planted clique $S \longrightarrow$ graph $G \longrightarrow$ Estimated clique $\widehat{S}$

- Minimax framework: Find an estimator $\widehat{S} = \widehat{S}(G)$ that performs well in worst-case

$$\min_{S \in \binom{[n]}{k}} \mathbb{P}_S\left[\widehat{S}(G) = S\right] \approx 1$$

- Bayesian framework: Find an estimator $\widehat{S} = \widehat{S}(G)$ that performs well on average

$$\mathbf{E}_{S \sim \mathsf{Unif}\left(\binom{[n]}{k}\right)} \mathbb{P}_S\left[\widehat{S}(G) = S\right] \approx 1$$

- The two formulations are equivalent by the permutaiton invariance of the model:

$$\sup_{\widehat{S}} \min_{S \in \binom{[n]}{k}} \mathbb{P}_S\left[\widehat{S}(G) = S\right] = \sup_{\widehat{S}} \mathbf{E}_{S \sim \mathsf{Unif}\left(\binom{[n]}{k}\right)} \mathbb{P}_S\left[\widehat{S}(G) = S\right].$$

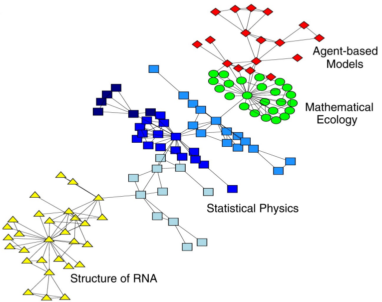
Vignette # 2: Community detection

Community detection in networks

- Networks with community structures arise in many applications
- Task: Discover underlying communities based on the network topology alone

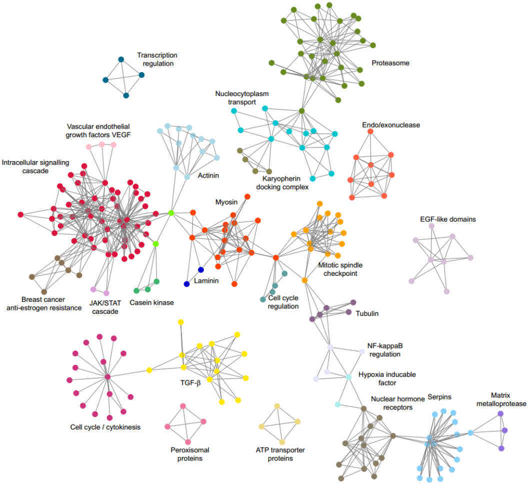
Example 1

Santa Fe Institute Collaboration network [Girvan-Newman '02]



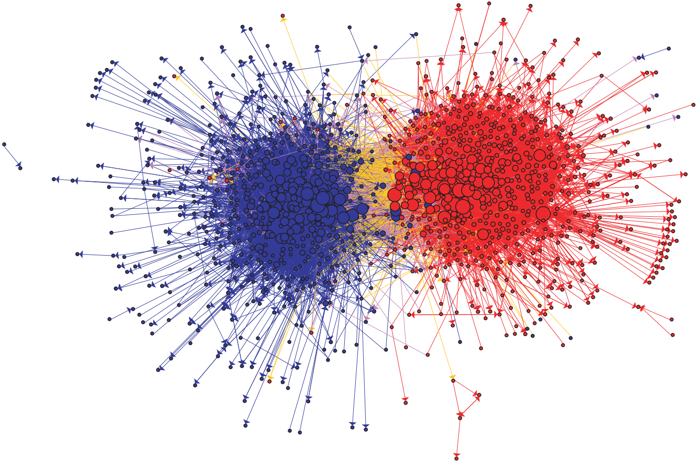
Example 2

Protein-protein interaction networks [Jonsson et al. 06']

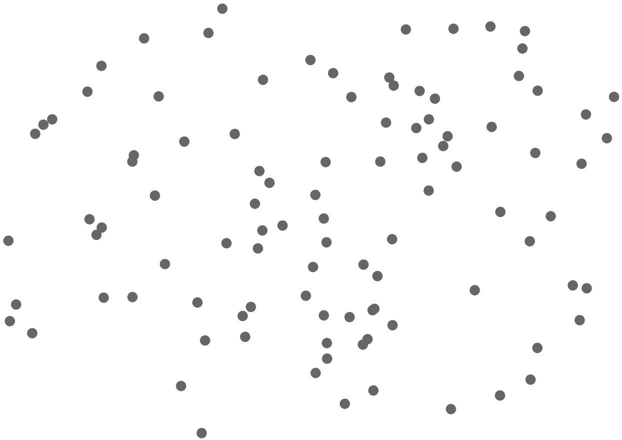


Example 3

Political blogosphere and the 2004 U.S. election [Adamic-Glance '05]

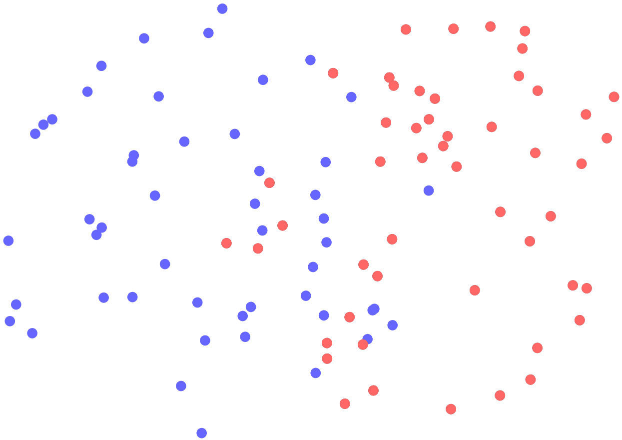


Stochastic block model – graph view



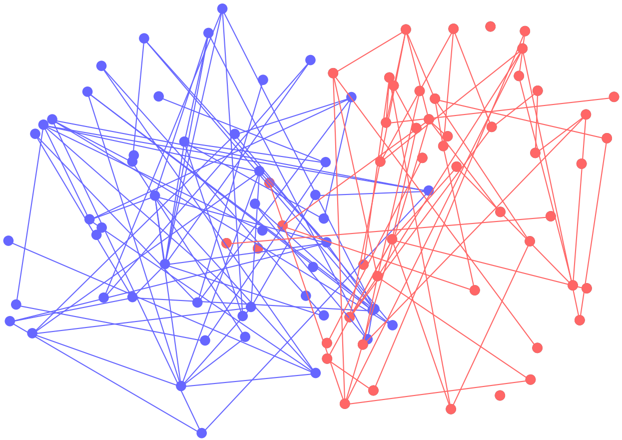
Stochastic block model – graph view

1. n nodes are randomly partitioned into 2 equal-sized communities



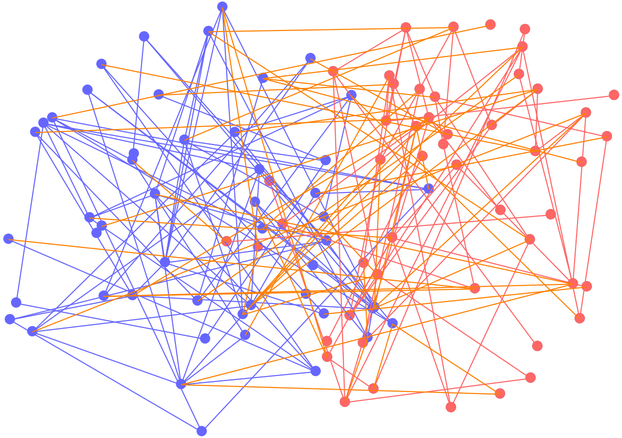
Stochastic block model – graph view

1. n nodes are randomly partitioned into 2 equal-sized communities
2. For every pair of nodes in same community, add an edge w.p. p



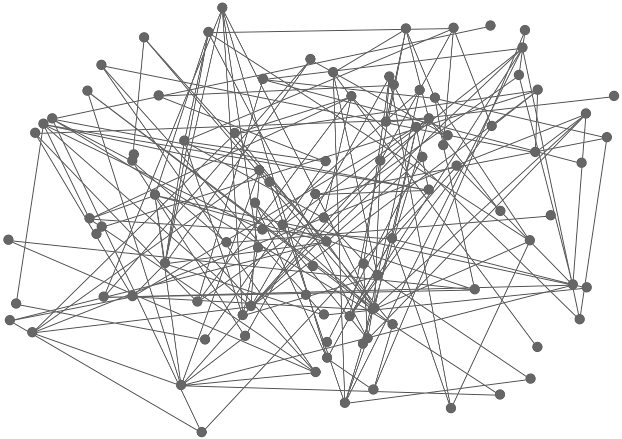
Stochastic block model – graph view

1. n nodes are randomly partitioned into 2 equal-sized communities
2. For every pair of nodes in same community, add an edge w.p. p
3. For every pair of nodes in diff. community, add an edge w.p. q



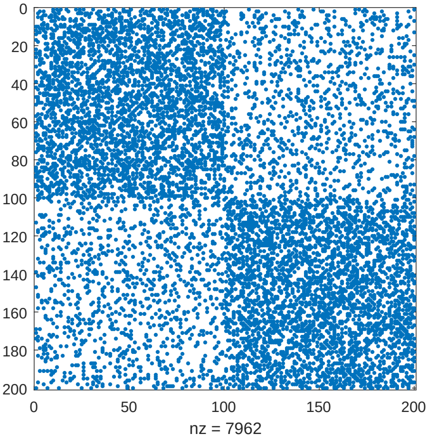
Stochastic block model – graph view

1. n nodes are randomly partitioned into 2 equal-sized communities
2. For every pair of nodes in same community, add an edge w.p. p
3. For every pair of nodes in diff. community, add an edge w.p. q

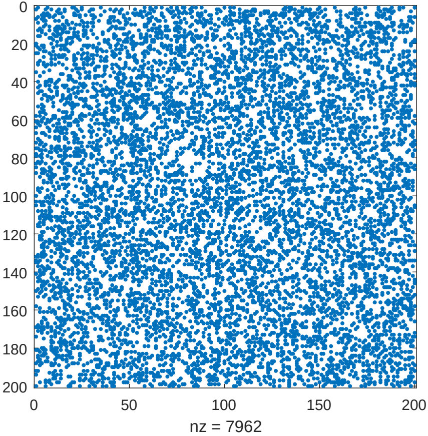


Call this SBM(n, p, q), SBM with two equal communities.

Stochastic block model – adjacency matrix view



Stochastic block model – adjacency matrix view



Stochastic block model – estimation

Planted community $\sigma \in \{\pm 1\}^n \longrightarrow$ graph $G \longrightarrow$ Estimated community $\widehat{\sigma}$

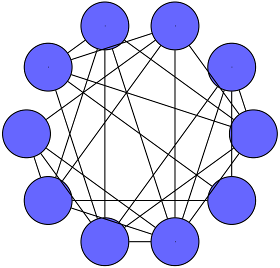
$$\mathbb{P}\big[(i,j) \in E\big] = \begin{cases} p & \sigma_i = \sigma_j \\ q & \sigma_i \neq \sigma_j \end{cases},$$

Goal:

- Detection: test $H_0 : G \sim G(n, \frac{p+q}{2})$ or $H_1 : G \sim \mathsf{SBM}(n, p, q)$
- Recovery: with high probability, $\hat{\sigma}$ and σ agree on
  - (Weak recovery) 50.0001% nodes (better than random guessing)
  - (Almost exact recovery) all but $o(n)$ nodes
  - (Exact recovery) all nodes

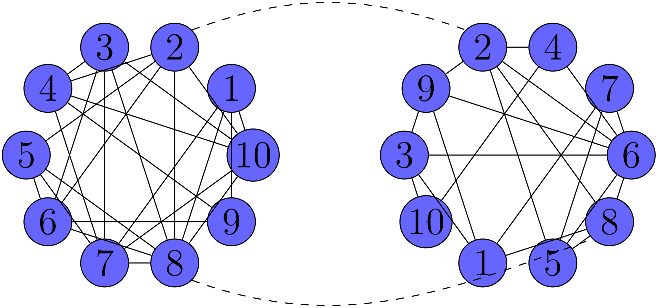
Vignette # 3: Graph matching

Graph matching (network alignment)



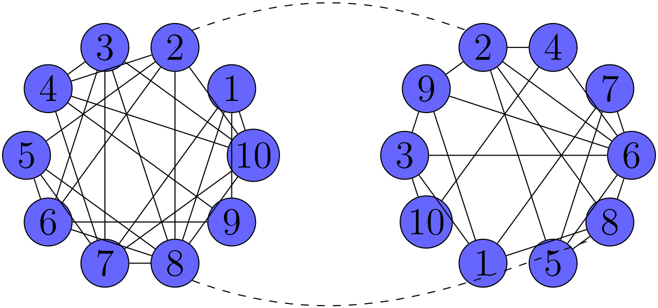
Goal: find a correspondence between two vertex sets that maximally aligns the edges

Graph matching (network alignment)



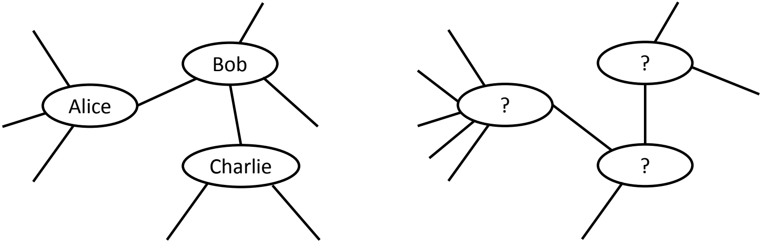
Goal: find a correspondence between two vertex sets that maximally aligns the edges

Graph matching (network alignment)



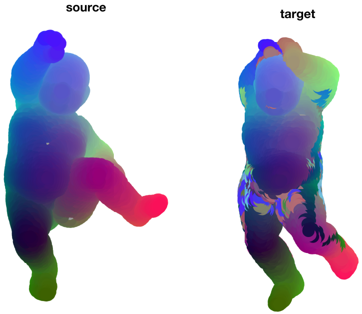
Goal: find a correspondence between two vertex sets that maximally aligns the edges Noiseless case: reduce to graph isomoprhism

Example: Network de-anonymization



Example 2: 3D shape matching

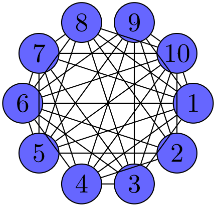
Find the correct vertex correspondance between two geometric graphs



Statistical model: Correlated Erdős-Rényi graphs

$$\text{Permutation } \pi \in S_n \longrightarrow \text{graphs } (G_1, G_2) \longrightarrow \text{Estimator } \widehat{\pi}$$

where (G_1, G_2) are Erdős-Rényi graphs correlated through the latent vertex correspondence π:

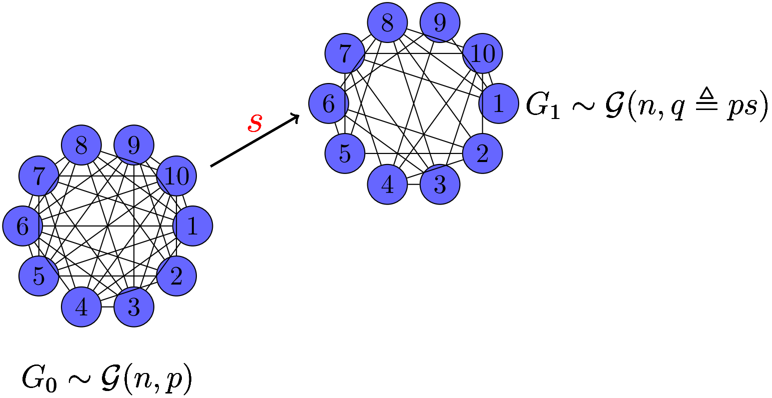


$$G_0 \sim \mathcal{G}(n, p)$$

Statistical model: Correlated Erdős-Rényi graphs

$$\text{Permutation } \pi \in S_n \longrightarrow \text{graphs } (G_1, G_2) \longrightarrow \text{Estimator } \widehat{\pi}$$

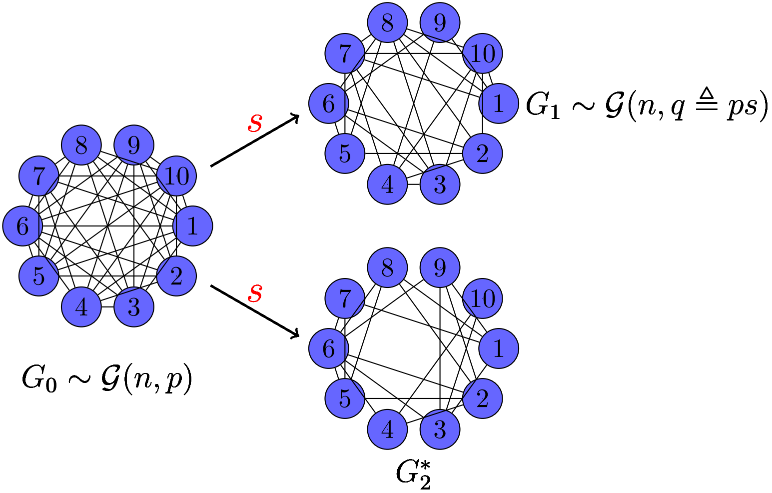
where (G_1, G_2) are Erdős-Rényi graphs correlated through the latent vertex correspondence π:



Statistical model: Correlated Erdős-Rényi graphs

Permutation $\pi \in S_n \longrightarrow$ graphs $(G_1, G_2) \longrightarrow$ Estimator $\widehat{\pi}$

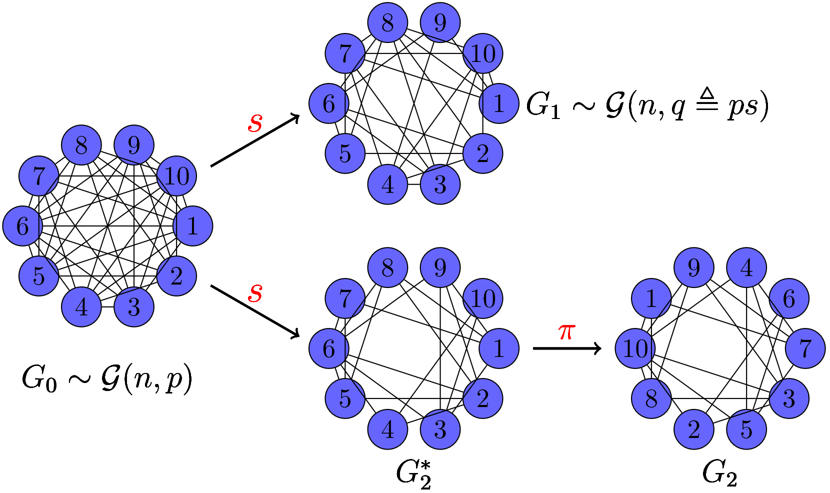
where (G_1, G_2) are Erdős-Rényi graphs correlated through the latent vertex correspondence π:



Statistical model: Correlated Erdős-Rényi graphs

Permutation $\pi \in S_n \longrightarrow$ graphs $(G_1, G_2) \longrightarrow$ Estimator $\widehat{\pi}$

where (G_1, G_2) are Erdős-Rényi graphs correlated through the latent vertex correspondence π:



Asymptotic Notation (Big-Oh)

For two sequences of positive numbers a_n and b_n,

- $a_n = O(b_n)$ or $a_n \lesssim b_n$, if there exist a constant C such that $|a_n| \leq C b_n$ for all sufficiently large n.

Asymptotic Notation (Big-Oh)

For two sequences of positive numbers a_n and b_n,

- $a_n = O(b_n)$ or $a_n \lesssim b_n$, if there exist a constant C such that $|a_n| \leq C b_n$ for all sufficiently large n.
- $a_n = \Omega(b_n)$ or $a_n \gtrsim b_n$, if $b_n = O(a_n)$
- $a_n = \Theta(b_n)$ or $a_n \asymp b_n$, if $a_n = O(b_n)$ and $a_n = \Omega(b_n)$.

Asymptotic Notation (Big-Oh)

For two sequences of positive numbers a_n and b_n,

- $a_n = O(b_n)$ or $a_n \lesssim b_n$, if there exist a constant C such that $|a_n| \leq C b_n$ for all sufficiently large n.
- $a_n = \Omega(b_n)$ or $a_n \gtrsim b_n$, if $b_n = O(a_n)$
- $a_n = \Theta(b_n)$ or $a_n \asymp b_n$, if $a_n = O(b_n)$ and $a_n = \Omega(b_n)$.
- $a_n = o(b_n)$ or $a_n \ll b_n$, if $a_n / b_n \to 0$ as $n \to \infty$.
- $a_n = \omega(b_n)$ or $a_n \gg b_n$, if $a_n / b_n \to \infty$ as $n \to \infty$.

Asymptotic Notation (Big-Oh)

For two sequences of positive numbers a_n and b_n,

- $a_n = O(b_n)$ or $a_n \lesssim b_n$, if there exist a constant C such that $|a_n| \leq C b_n$ for all sufficiently large n.
- $a_n = \Omega(b_n)$ or $a_n \gtrsim b_n$, if $b_n = O(a_n)$
- $a_n = \Theta(b_n)$ or $a_n \asymp b_n$, if $a_n = O(b_n)$ and $a_n = \Omega(b_n)$.
- $a_n = o(b_n)$ or $a_n \ll b_n$, if $a_n / b_n \to 0$ as $n \to \infty$.
- $a_n = \omega(b_n)$ or $a_n \gg b_n$, if $a_n / b_n \to \infty$ as $n \to \infty$.
- $a_n = \text{poly}(n)$ if $a_n = n^{O(1)}$.
- $a_n = \text{polylog}(n)$ if $a_n = (\log n)^{O(1)}$.

Asymptotic Notation (Big-Oh)

For two sequences of positive numbers a_n and b_n,

- $a_n = O(b_n)$ or $a_n \lesssim b_n$, if there exist a constant C such that $|a_n| \leq C b_n$ for all sufficiently large n.
- $a_n = \Omega(b_n)$ or $a_n \gtrsim b_n$, if $b_n = O(a_n)$
- $a_n = \Theta(b_n)$ or $a_n \asymp b_n$, if $a_n = O(b_n)$ and $a_n = \Omega(b_n)$.
- $a_n = o(b_n)$ or $a_n \ll b_n$, if $a_n/b_n \to 0$ as $n \to \infty$.
- $a_n = \omega(b_n)$ or $a_n \gg b_n$, if $a_n/b_n \to \infty$ as $n \to \infty$.
- $a_n = \mathrm{poly}(n)$ if $a_n = n^{O(1)}$.
- $a_n = \mathrm{polylog}(n)$ if $a_n = (\log n)^{O(1)}$.

When a_n is not positive, $O(\cdot)$ and $o(\cdot)$ also make sense by applying to $|a_n|$.

We use subscript to indicate the dependency of proportionality constants on some other quantity, for example $a_n = O_m(b_n)$ or $a_n \lesssim_m b_n$ if $a_n \leq C(m) b_n$.

We use tilde to hide logarithmic factors, for example $a_n = \tilde{O}(b_n)$ if $a_n = O(b_n \mathrm{polylog}(n))$.

Asymptotic Notation (Big-Oh)

For two sequences of positive numbers a_n and b_n,

- $a_n = O(b_n)$ or $a_n \lesssim b_n$, if there exist a constant C such that $|a_n| \leq C b_n$ for all sufficiently large n.
- $a_n = \Omega(b_n)$ or $a_n \gtrsim b_n$, if $b_n = O(a_n)$
- $a_n = \Theta(b_n)$ or $a_n \asymp b_n$, if $a_n = O(b_n)$ and $a_n = \Omega(b_n)$.
- $a_n = o(b_n)$ or $a_n \ll b_n$, if $a_n / b_n \to 0$ as $n \to \infty$.
- $a_n = \omega(b_n)$ or $a_n \gg b_n$, if $a_n / b_n \to \infty$ as $n \to \infty$.
- $a_n = \mathrm{poly}(n)$ if $a_n = n^{O(1)}$.
- $a_n = \mathrm{polylog}(n)$ if $a_n = (\log n)^{O(1)}$.

Also, we say that a sequence of events $\mathcal{E}_n$ holds with high probability (whp), if $\mathbb{P}\{\mathcal{E}_n\} \to 1$ as $n \to \infty$.

When a_n is not positive, $O(\cdot)$ and $o(\cdot)$ also make sense by applying to $|a_n|$.

We use subscript to indicate the dependency of proportionality constants on some other quantity, for example $a_n = O_m(b_n)$ or $a_n \lesssim_m b_n$ if $a_n \leq C(m) b_n$.

We use tilde to hide logarithmic factors, for example $a_n = \tilde{O}(b_n)$ if $a_n = O(b_n \mathrm{polylog}(n))$.