

Special Homework 1

15-359/659 PnC II

2026 Fall

Basic part: Due 12:50 pm on October 30, 2026
Advanced part: Due 12:50 pm on November 20, 2026

1 Background

We consider the problem of graph matching, also known as graph alignment or noisy graph isomorphism. This problem has been used to model social network de-anonymization, domain adaptation in machine learning, knowledge graph completion, etc.

Graph generations. In this problem, we start with a baseline graph G drawn from an Erdős–Rényi random graph model with parameters (n, p) , denoted by $G \sim \text{ER}(n, p)$, where $0 < p < 1$. Specifically, the graph G has n vertices, labeled by the integers in $[n] \triangleq \{1, 2, \dots, n\}$. Each pair of distinct vertices (i, j) is connected by an edge independently with probability p .

We now generate two *subsampled* versions of G , denoted as G'_1 and G_2 , respectively. In particular, let G'_1 be a graph on the same vertex set $[n]$. For each distinct vertex pair (i, j) , if graph G has an edge between i and j , graph G'_1 keeps this edge with probability s for $0 < s < 1$ independently; otherwise, if graph G does not have an edge between i and j , graph G'_1 does not have an edge either. The other graph, G_2 , is generated in the same way, with independent randomness for the subsampling.

We then generate an anonymized version of graph G'_1 , denoted as G_1 , by applying a permutation σ^* to the vertices of G'_1 . Here the permutation σ^* is chosen uniformly at random from S_n , the set of all possible permutations on $[n]$. We view a permutation as a mapping from each $i \in [n]$ to where it is permuted to in $[n]$; e.g., $\sigma^*(i)$ is the new label that vertex i of G'_1 gets in G_1 .

De-anonymization question. Our goal is to estimate the permutation σ^* , given observations of graph instances G_1 and G_2 . In particular, we are interested in the maximum a posteriori (MAP) estimator, i.e., we seek to find the permutation σ that maximizes $\mathbb{P}(\sigma^* = \sigma \mid G_1, G_2)$.

Deriving the posterior. We know that $\mathbb{P}(\sigma^* = \sigma \mid G_1, G_2) \propto \mathbb{P}(G_1, G_2 \mid \sigma^* = \sigma)$ since the prior distribution of σ^* is uniform. We now derive an explicit form of $\mathbb{P}(G_1, G_2 \mid \sigma^* = \sigma)$.

Let's first define the operation of applying a permutation σ to graph G_2 . By this we mean that we relabel each vertex i of G_2 to $\sigma(i)$. Let $\sigma(G_2)$ denote the resultant graph after the relabeling.

Now define

$$\begin{aligned} m_{11}(\sigma) &= |\{(i, j) \in [n] \times [n] : i \neq j, \text{there is an edge between } i \text{ and } j \text{ in both } G_1 \text{ and } \sigma(G_2)\}|, \\ m_{10}(\sigma) &= |\{(i, j) \in [n] \times [n] : i \neq j, \text{there is an edge between } i \text{ and } j \text{ in } G_1 \text{ but not } \sigma(G_2)\}|, \\ m_{01}(\sigma) &= |\{(i, j) \in [n] \times [n] : i \neq j, \text{there is an edge between } i \text{ and } j \text{ in } \sigma(G_2) \text{ but not } G_1\}|, \\ m_{00}(\sigma) &= |\{(i, j) \in [n] \times [n] : i \neq j, \text{there is no edge between } i \text{ and } j \text{ in either } G_1 \text{ or } \sigma(G_2)\}|. \end{aligned}$$

It is not hard to see

$$\mathbb{P}(G_1, G_2 \mid \sigma^* = \sigma) = (ps^2)^{m_{11}(\sigma)} (ps(1-s))^{m_{10}(\sigma) + m_{01}(\sigma)} (1-2ps+ps^2)^{m_{00}(\sigma)}.$$

We can further simplify this as follows. Let

$$k(\sigma) = m_{10}(\sigma) + m_{01}(\sigma).$$

Note that

$$m_{11}(\sigma) = \frac{1}{2} (|E(G_1)| + |E(G_2)| - k(\sigma)),$$

where $|E(G_1)|$ and $|E(G_2)|$ are the number of edges in G_1 and G_2 , respectively. Then

$$m_{00}(\sigma) = \binom{n}{2} - m_{11}(\sigma) - m_{10}(\sigma) - m_{01}(\sigma) = \binom{n}{2} - \frac{1}{2}(|E(G_1)| + |E(G_2)| - k(\sigma)) - k(\sigma).$$

Combining these, and noticing that $|E(G_1)|$ and $|E(G_2)|$ do not depend on σ , we have

$$\begin{aligned} & \mathbb{P}(G_1, G_2 \mid \sigma^* = \sigma) \\ &= (ps^2)^{\frac{1}{2}(|E(G_1)| + |E(G_2)| - k(\sigma))} (ps(1-s))^{k(\sigma)} (1-2ps+ps^2)^{\binom{n}{2} - \frac{1}{2}(|E(G_1)| + |E(G_2)| - k(\sigma)) - k(\sigma)} \\ &\propto \left(\frac{\sqrt{ps^2}\sqrt{1-2ps+ps^2}}{ps(1-s)} \right)^{-k(\sigma)}. \end{aligned}$$

Let

$$\theta = \frac{\sqrt{ps^2}\sqrt{1-2ps+ps^2}}{ps(1-s)}. \quad (1)$$

It is not hard to verify that $\theta > 1$. Then $\mathbb{P}(G_1, G_2 \mid \sigma^* = \sigma) \propto \theta^{-k(\sigma)}$. Therefore, the posterior distribution satisfies

$$\mathbb{P}(\sigma^* = \sigma \mid G_1, G_2) \propto \theta^{-k(\sigma)}.$$

2 Problem Statement

Basic part. You are given graph instances G_1 and G_2 . You are also given the values of the parameters n, p, s , and thus you can compute θ using Equation (1). Design an MCMC algorithm that samples from the posterior distribution. That is, your algorithm should output a permutation σ on $[n]$ that follows the distribution

$$\mathbb{P}(\sigma^* = \sigma \mid G_1, G_2) \propto \theta^{-k(\sigma)},$$

where $k(\sigma)$ is the total number of (i, j) pairs such that there is an edge between (i, j) in G_1 but not in $\sigma(G_2)$ or in $\sigma(G_2)$ but not in G_1 .

[Hint] One possible route is to consider the operation where you swap two distinct vertices i and j in your permutation.

Advanced part. Provide an upper bound on the mixing time, $t_{\text{mix}}(\varepsilon)$, for the Markov chain formed by the states of your MCMC algorithm.

Your answer will be ranked based on two criteria: (1) the order of the mixing time; and (2) the conditions imposed on n, p, s by your algorithm. The smaller the mixing time and the wider the range of parameters for which your analysis works, the higher your ranking.

For this problem, you can think of p and s as functions of n , and derive the order of the mixing time in terms of n . For conditions on n, p, s , the easiest range is when $nps^2 \leq 1$. It gets harder when the value of nps^2 becomes larger. If you can design an algorithm with a mixing time polynomial in n for some cases under $nps^2 = \log n + \omega(n)$, that will likely be publishable results. If your result is better than an existing result on the scoreboard in some aspect (from another student or your previous submission), then you need to include a proof for it.