From tree matching to sparse graph alignment.

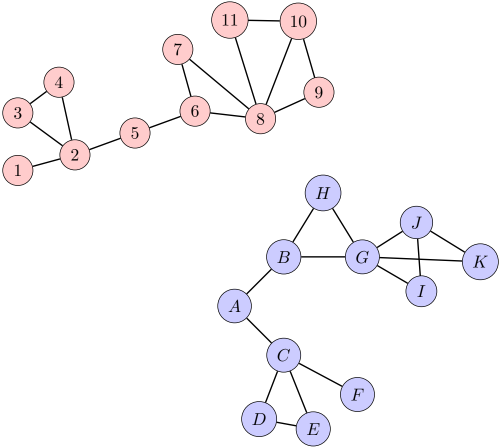
Luca Ganassali and Laurent Massoulié

INRIA, Paris

Dyogene Seminar, June 25, 2020

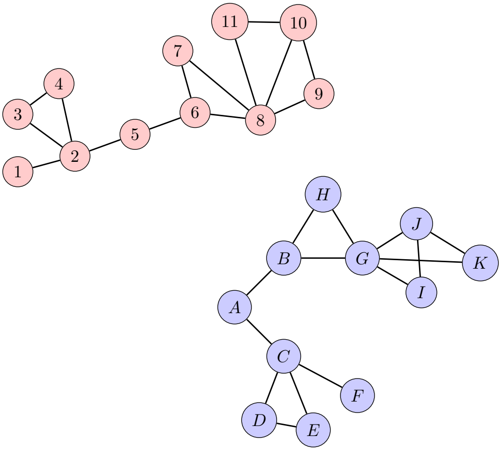
ÉCOLE NORMALE SUPÉRIEURE
PSL RESEARCH UNIVERSITY PARIS

Introduction: the graph isomorphism problem



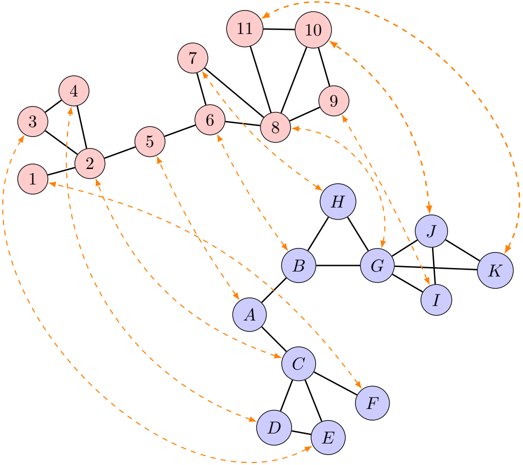
Introduction: the graph isomorphism problem

Question: Given two graphs $G = (V, E)$ and $G' = (V', E')$, is there a *graph isomorphism*, i.e. a bijection $f : V \to V'$ such that $(i, j) \in E \iff (f(i), f(j)) \in E'$?



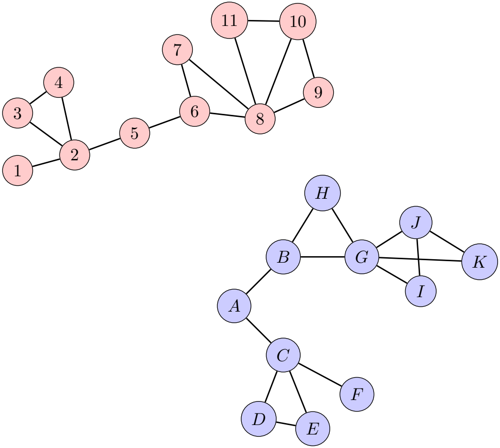
Introduction: the graph isomorphism problem

Question: Given two graphs $G = (V, E)$ and $G' = (V', E')$, is there a *graph isomorphism*, i.e. a bijection $f : V \to V'$ such that $(i, j) \in E \iff (f(i), f(j)) \in E'$?

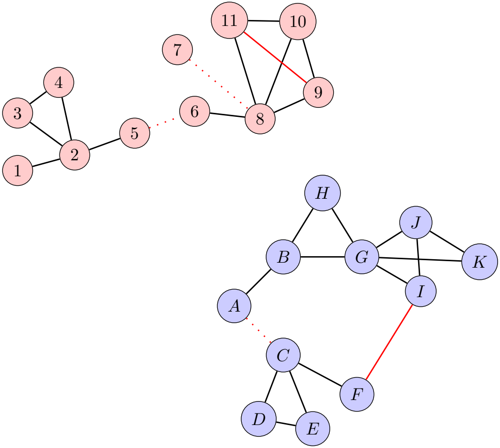


Problem in NP, thought to be neither in P nor NP-complete.

Introduction: graph alignment

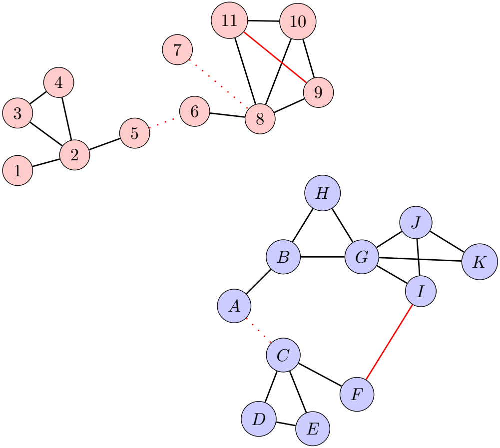


Introduction: graph alignment



Introduction: graph alignment

Relaxed version: Is there a bijection $f : V \to V'$ that *preserves most edges*?



Introduction: applications

Introduction: applications

- De-anonimization of networks.

Introduction: applications

- De-anonimization of networks.
- Align protein interaction networks for cells of species $\rightarrow$ infer function of proteins in biology of species A from knowledge of protein function in biology of species B

Introduction: applications

- De-anonimization of networks.
- Align protein interaction networks for cells of species $\rightarrow$ infer function of proteins in biology of species A from knowledge of protein function in biology of species B
- Align graphs between words in languages A and B to construct dictionary between the two languages

Introduction: applications

- De-anonimization of networks.
- Align protein interaction networks for cells of species $\rightarrow$ infer function of proteins in biology of species A from knowledge of protein function in biology of species B
- Align graphs between words in languages A and B to construct dictionary between the two languages
- ...

Introduction: graph alignment

Formally: f minimizes

$$\sum_{i=1}^{n} \left(\mathbf{1}_{(i,j)\in E} - \mathbf{1}_{(f(i),f(j))\in E'} \right) .$$

Introduction: graph alignment

Formally: f minimizes

$$\sum_{i=1}^{n} \left(\mathbf{1}_{(i,j)\in E} - \mathbf{1}_{(f(i),f(j))\in E'} \right).$$

$\longrightarrow$ instance of the NP-hard quadratic assigment problem (QAP):

$$\max_{\Pi} \mathrm{Tr} \left(G \Pi G' \Pi^\top \right),$$

where Π runs over all permutation matrices.

Generative random graphs models

Generative random graphs models

Erdős-Rényi random graph $\mathcal{G}(n, p)$: n vertices, each edge present with probability p, independently from other edges.

Generative random graphs models

Erdős-Rényi random graph $\mathcal{G}(n, p)$: n vertices, each edge present with probability p, independently from other edges.

Correlated Erdős-Rényi random graphs $\mathcal{G}(n, p, s)$, with planted permutation:

Generative random graphs models

Erdős-Rényi random graph $\mathcal{G}(n,p)$: n vertices, each edge present with probability p, independently from other edges.

Correlated Erdős-Rényi random graphs $\mathcal{G}(n,p,s)$, with planted permutation:

- Start with a 'parent graph' $G_0 \sim \mathcal{G}(n, p/s)$,

Generative random graphs models

Erdős-Rényi random graph $\mathcal{G}(n, p)$: n vertices, each edge present with probability p, independently from other edges.

Correlated Erdős-Rényi random graphs $\mathcal{G}(n, p, s)$, with planted permutation:

- Start with a 'parent graph' $G_0 \sim \mathcal{G}(n, p/s)$,
- Form G_1 as a $s-$sub-sampling of G_0: independently, keep each edge with probability s.

Generative random graphs models

Erdős-Rényi random graph $\mathcal{G}(n, p)$: n vertices, each edge present with probability p, independently from other edges.

Correlated Erdős-Rényi random graphs $\mathcal{G}(n, p, s)$, with planted permutation:

- Start with a 'parent graph' $G_0 \sim \mathcal{G}(n, p/s)$,
- Form G_1 as a $s-$sub-sampling of G_0: independently, keep each edge with probability s.
- Form G_2' as an other independent $s-$sub-sampling of G_0.

Generative random graphs models

Erdős-Rényi random graph $\mathcal{G}(n, p)$: n vertices, each edge present with probability p, independently from other edges.

Correlated Erdős-Rényi random graphs $\mathcal{G}(n, p, s)$, with planted permutation:

- Start with a 'parent graph' $G_0 \sim \mathcal{G}(n, p/s)$,
- Form G_1 as a $s-$sub-sampling of G_0: independently, keep each edge with probability s.
- Form G_2' as an other independent $s-$sub-sampling of G_0.
- Shuffle labels of G_2' uniformly at random to form G_2. Formally, $G_2 = \Pi^\top G_2' \Pi$, where $\Pi = \Pi_\sigma$ is the matrix of a uniform permutation σ.

Informational and computational thresholds

Informational and computational thresholds

Exact recovery of σ:

- Information-theoretically feasible iff $nps = \log n + \omega(1)$ [Cullina-Kiyavash'16].
- Polynomial time feasible if $np \geq (\log n)^{\alpha}$ and $1 - s \leq (\log n)^{-\beta}$[Ding et al.'18].
- Exact recovery requires at least a mean degree of order $\log n$.

Informational and computational thresholds

Exact recovery of σ:

- Information-theoretically feasible iff $nps = \log n + \omega(1)$ [Cullina-Kiyavash'16].
- Polynomial time feasible if $np \geq (\log n)^{\alpha}$ and $1 - s \leq (\log n)^{-\beta}$[Ding et al.'18].
- Exact recovery requires at least a mean degree of order $\log n$.

This work: partial recovery of σ

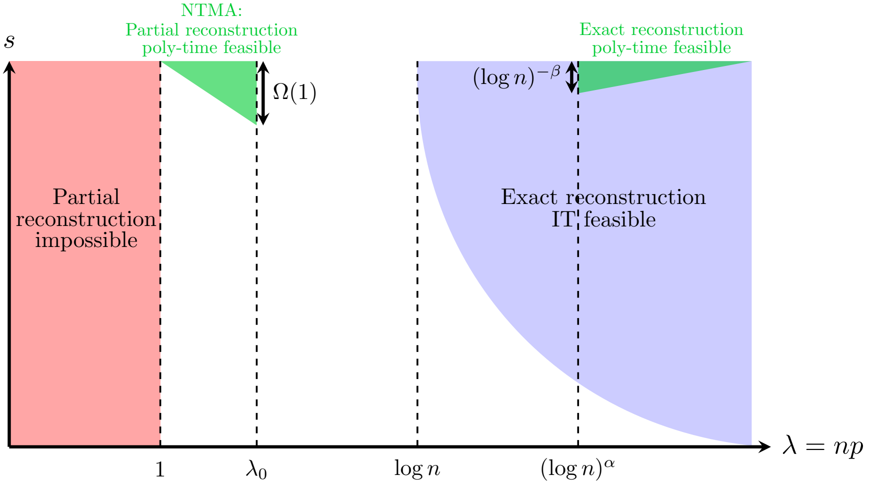
- Polynomial-time recovery, in sparse regime $np = O(1)$.
- Relaxed objective: find a one-to-one $\hat{\sigma}$ from G_1, G_2, such that

$$\mathrm{overlap}(\hat{\sigma}) := \frac{1}{n} \sum_{i=1}^{n} \mathbf{1}_{\hat{\sigma}(i)=\sigma(i)} = \Omega(1),$$

  and

$$\frac{1}{n} \sum_{i=1}^{n} \mathbf{1}_{\hat{\sigma}(i)\neq\sigma(i)} = o(1).$$

Informational and computational thresholds



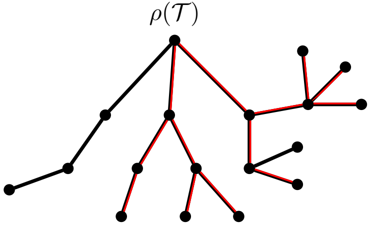
Matching weight of two trees

Matching weight of two trees

Given two rooted trees $\mathcal{T}, \mathcal{T}'$, their **matching weight at depth** d $\mathcal{W}_d(\mathcal{T}, \mathcal{T}')$ is the largest number of leaves at depth d of a common rooted sub-tree $\mathcal{T}''$.

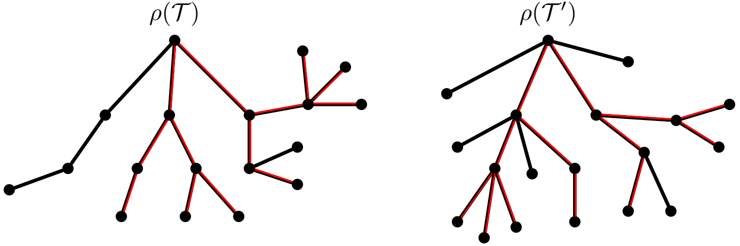
Matching weight of two trees

Given two rooted trees $\mathcal{T}, \mathcal{T}'$, their **matching weight at depth** d $\mathcal{W}_d(\mathcal{T}, \mathcal{T}')$ is the largest number of leaves at depth d of a common rooted sub-tree $\mathcal{T}''$.



Matching weight of two trees

Given two rooted trees $\mathcal{T}, \mathcal{T}'$, their **matching weight at depth** d $\mathcal{W}_d(\mathcal{T}, \mathcal{T}')$ is the largest number of leaves at depth d of a common rooted sub-tree $\mathcal{T}''$.



Recursive computation:

$$\mathcal{W}_d(\mathcal{T}, \mathcal{T}') = \sup_{\mathfrak{m}} \sum_{(i,u)\in\mathfrak{m}} \mathcal{W}_{d-1}(\mathcal{T}_{i\leftarrow\rho(\mathcal{T})}, \mathcal{T}'_{u\leftarrow\rho(\mathcal{T}')}).$$

Intuition for main result

$(G_1, G_2) \sim \mathcal{G}(n, p = \lambda/n, s)$ with planted permutation σ.

Intuition for main result

$(G_1, G_2) \sim \mathcal{G}(n, p = \lambda/n, s)$ with planted permutation σ.

- if $u = \sigma(i)$, the neighborhoods $\mathcal{N}_i$ of i in G_1 and $\mathcal{N}_u$ in $G_2 \simeq$ GW trees of offspring $\mathcal{P}(\lambda)$, with intersection of offspring $\mathcal{P}(\lambda s)$. Thus

$$\mathcal{W}_d(\mathcal{N}_i, \mathcal{N}_u) \geq \sharp\text{leaves at depth } d \text{ in the intersection} \simeq (\lambda s)^d$$

Intuition for main result

$(G_1, G_2) \sim \mathcal{G}(n, p = \lambda/n, s)$ with planted permutation σ.

- if $u = \sigma(i)$, the neighborhoods $\mathcal{N}_i$ of i in G_1 and $\mathcal{N}_u$ in $G_2 \simeq$ GW trees of offspring $\mathcal{P}(\lambda)$, with intersection of offspring $\mathcal{P}(\lambda s)$. Thus

$$\mathcal{W}_d(\mathcal{N}_i, \mathcal{N}_u) \geq \sharp\text{leaves at depth } d \text{ in the intersection} \simeq (\lambda s)^d$$

- if $u \neq \sigma(i)$, $(\mathcal{N}_i, \mathcal{N}_u) \simeq$ independent GW trees of offspring $\mathcal{P}(\lambda)$.

Intuition for main result

$(G_1, G_2) \sim \mathcal{G}(n, p = \lambda/n, s)$ with planted permutation σ.

- if $u = \sigma(i)$, the neighborhoods $\mathcal{N}_i$ of i in G_1 and $\mathcal{N}_u$ in $G_2 \simeq$ GW trees of offspring $\mathcal{P}(\lambda)$, with intersection of offspring $\mathcal{P}(\lambda s)$. Thus

$$\mathcal{W}_d(\mathcal{N}_i, \mathcal{N}_u) \geq \sharp\text{leaves at depth } d \text{ in the intersection} \simeq (\lambda s)^d$$

- if $u \neq \sigma(i)$, $(\mathcal{N}_i, \mathcal{N}_u) \simeq$ independent GW trees of offspring $\mathcal{P}(\lambda)$.

Theorem

For $\lambda \in (1, \lambda_0]$ and $s \in (s^*(\lambda), 1]$, then there exists $\gamma < \lambda s$ such that

$$\mathcal{W}_d(\mathcal{T}, \mathcal{T}') \ll \gamma^d \text{ as } d \to \infty.$$

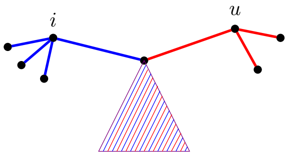
Intuition for main result

Compare $\mathcal{W}_d(\mathcal{N}_i, \mathcal{N}_u)$ to $(\lambda s)^d$?

Intuition for main result

Compare $\mathcal{W}_d(\mathcal{N}_i, \mathcal{N}_u)$ to $(\lambda s)^d$?

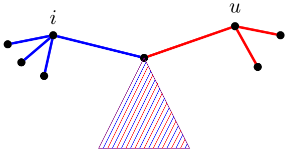
NO!



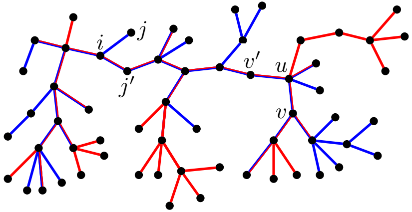
Intuition for main result

Compare $\mathcal{W}_d(\mathcal{N}_i, \mathcal{N}_u)$ to $(\lambda s)^d$?

NO!



'Dangling trees trick': look at both $\mathcal{W}_{d-1}(j \leftarrow i, v \leftarrow u)$ and $\mathcal{W}_{d-1}(j' \leftarrow i, v' \leftarrow u)$.



Neighborhood tree matching algorithm, main result

NTMA algorithm: $\mathcal{S} = \varnothing$. For all pairs $(i, u) \in V(G_1) \times V(G_2)$ whose $d-$neighborhoods $\mathcal{N}_i$ and $\mathcal{N}_u$ are trees:
If there exists $j \neq j' \overset{G_1}{\sim} i$, $v \neq v' \overset{G_2}{\sim} u$ such that $\mathcal{W}_{d-1}(j \leftarrow i, v \leftarrow u) > \tau$ and $\mathcal{W}_{d-1}(j' \leftarrow i, v' \leftarrow u) > \tau$, then add (i, u) to $\mathcal{S}$.

Neighborhood tree matching algorithm, main result

NTMA algorithm: $\mathcal{S} = \varnothing$. For all pairs $(i, u) \in V(G_1) \times V(G_2)$ whose $d-$neighborhoods $\mathcal{N}_i$ and $\mathcal{N}_u$ are trees:
If there exists $j \neq j' \overset{G_1}{\sim} i$, $v \neq v' \overset{G_2}{\sim} u$ such that $\mathcal{W}_{d-1}(j \leftarrow i, v \leftarrow u) > \tau$ and $\mathcal{W}_{d-1}(j' \leftarrow i, v' \leftarrow u) > \tau$, then add (i, u) to $\mathcal{S}$.

Theorem

Assume $\lambda s > 1$, $\lambda \in (1, \lambda_0]$ and $s \in (s^*(\lambda), 1]$. Then for $d = \Theta(\log n)$ and $\tau = \Theta(\gamma^{d-1})$, with high probability:

$$\frac{1}{n} \sum_{i=1}^{n} \mathbf{1}_{(i,\sigma(i)) \in \mathcal{S}} = \Omega(1) \quad \textit{and} \quad \frac{1}{n} \sum_{i=1}^{n} \mathbf{1}_{\exists u \neq \sigma(i), (i,u) \in \mathcal{S}} = o(1).$$

In practice...

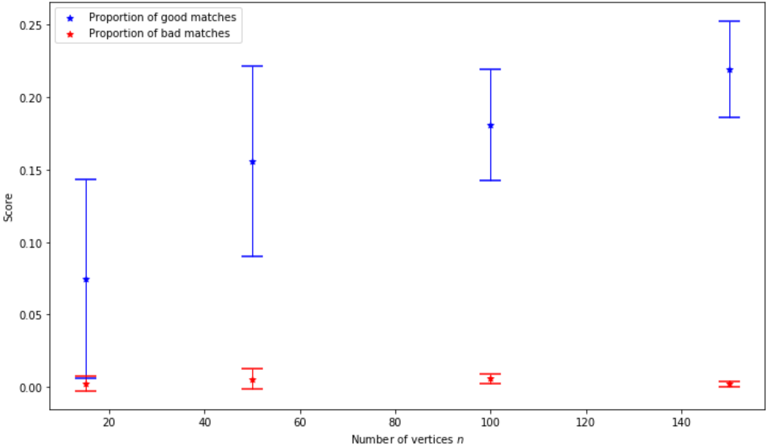


Figure: Mean score of NTMA-2 for $\lambda = 2.1$, $d = 5$, and $s = 1$ (isomorphism case).

In practice...

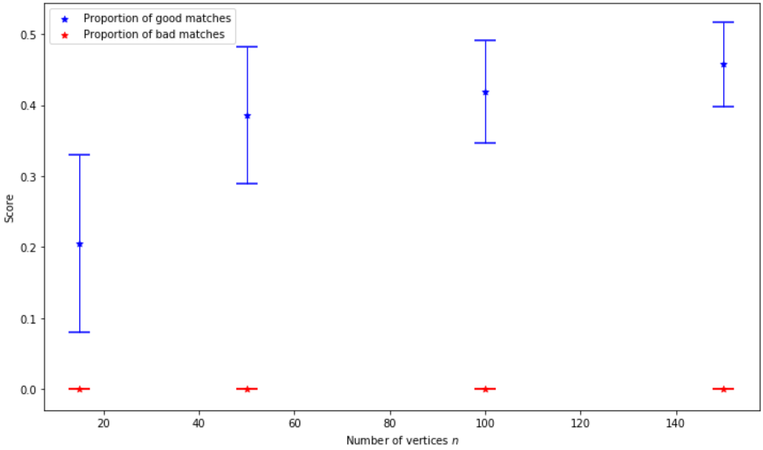


Figure: Mean score of NTMA-2 for $\lambda = 2.1$, $d = 5$, and $s = 0.95$.

In practice...

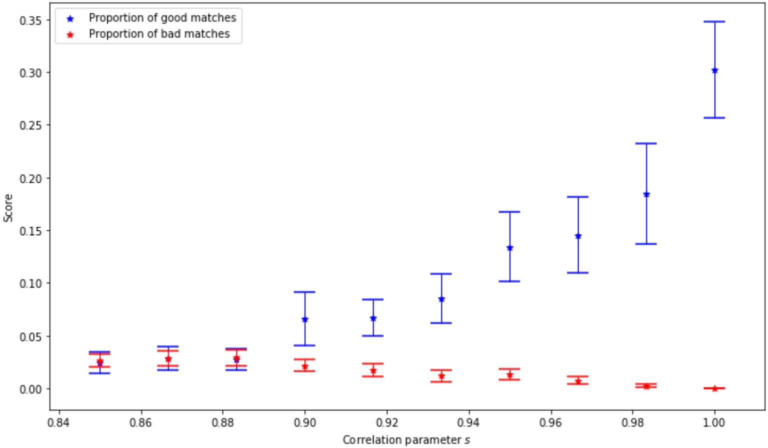


Figure: Mean score of NTMA-2 for $n = 150$, $\lambda = 1.4$, $d = 5$, and varying s.

Take-home messages

- Graph alignment, important unsupervised learning problem with many applications.

Take-home messages

- Graph alignment, important unsupervised learning problem with many applications.
- NTMA: first method proven to succeed at partial alignment in relevant regime of sparse graphs.

Take-home messages

- Graph alignment, important unsupervised learning problem with many applications.
- NTMA: first method proven to succeed at partial alignment in relevant regime of sparse graphs.
- Outlook: boundaries of phases in (λ, s) diagram, in particular IT-feasibility and poly-time feasibility of partial alignment, and correlation detection in trees.

Thank you!