Recent progress on random graph matching problems

Jian Ding
Peking University

based on joint works with

Hang Du (PKU), Shuyang Gong (PKU), Zhangsong Li (PKU)

Zongming Ma (Penn), Yihong Wu (Yale), Jiaming Xu (Duke)

Seminar on Stochastic Processes, March 2023

Application 1: Network de-anonymization

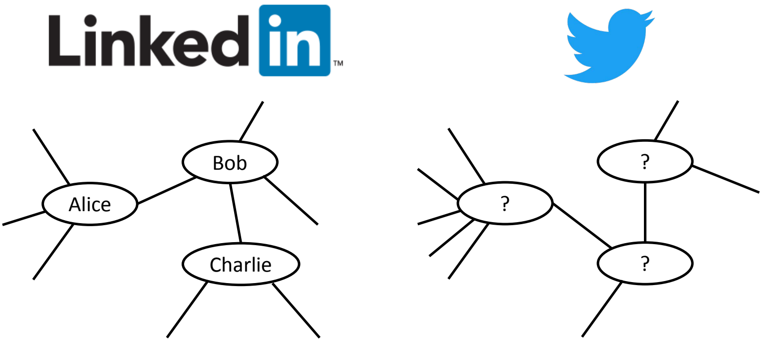


Figure: Picture courtesy of R. Srikant

Application 1: Network de-anonymization

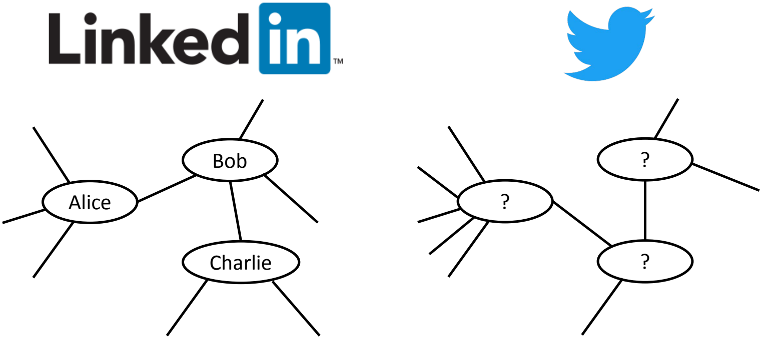


Figure: Picture courtesy of R. Srikant

- Successfully de-anonymize Netflix by matching it to IMDB [Narayanan-Shmatikov '08]
- Correctly identified 30.8% of node mappings between Twitter and Flickr [Narayanan-Shmatikov '09]

Application 2: Protein-Protein Interaction network

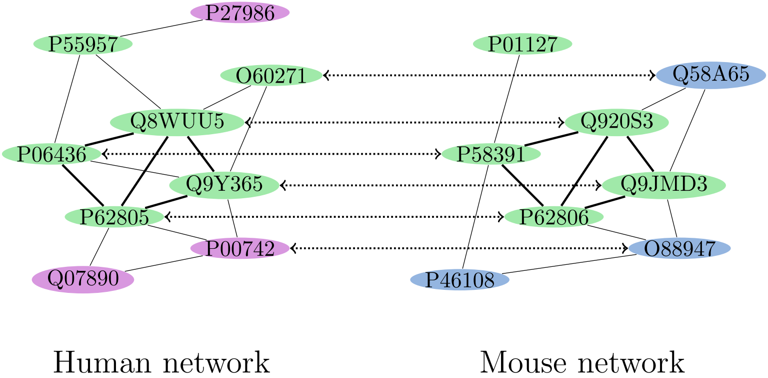


Figure: [Kazemi-Hassani-Grossglauser-Modarres '16]

Ontology: Discover proteins with similar functions across different species based on interaction network topology

Application 3: Computer vision

A fundamental problem in computer vision: Detect similar objects that undergo different deformations

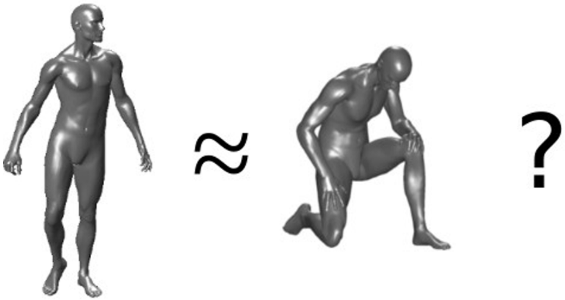


Figure: Shape REtrieval Contest (SHREC) dataset [Lähner et al '16]

Application 3: Computer vision

A fundamental problem in computer vision: Detect similar objects that undergo different deformations

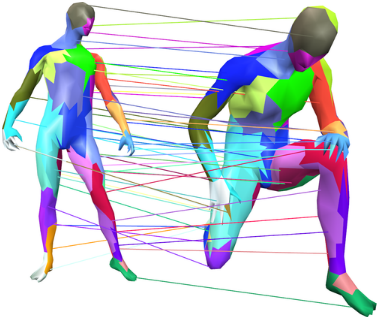


Figure: Shape REtrieval Contest (SHREC) dataset [Lähner et al '16]

3-D shapes $\rightarrow$ geometric graphs (features $\rightarrow$ nodes, distances $\rightarrow$ edges)

Determine whether two graphs are topologically similar

Application 4: Machine Translation

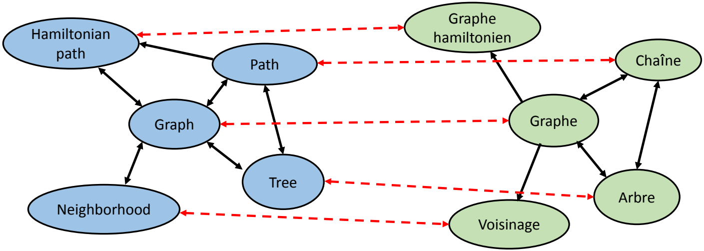
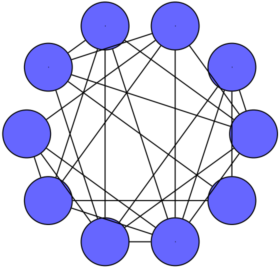
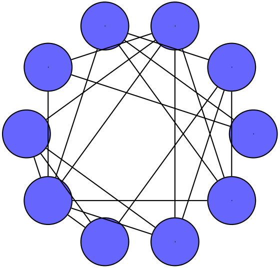


Figure: Picture courtesy of R. Srikant

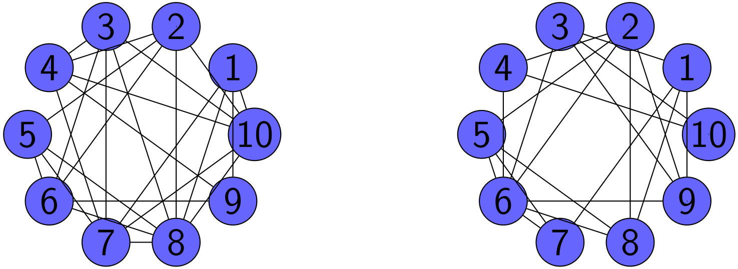
Automatically find/correct corresp. wiki articles in different languages [Fishkind-Adali-Patsolic-Meng-Lyzinski-Priebe '12]

Graph matching (network alignment)



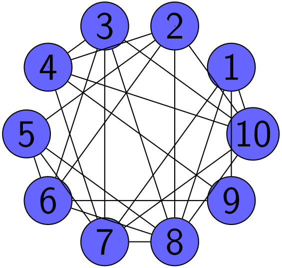


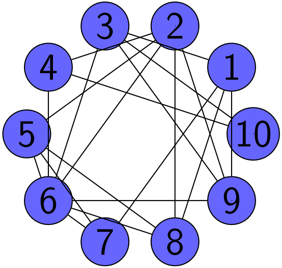
Graph matching (network alignment)



Graph matching: a fundamental mathematical question underlying these applications.

Graph matching (network alignment)

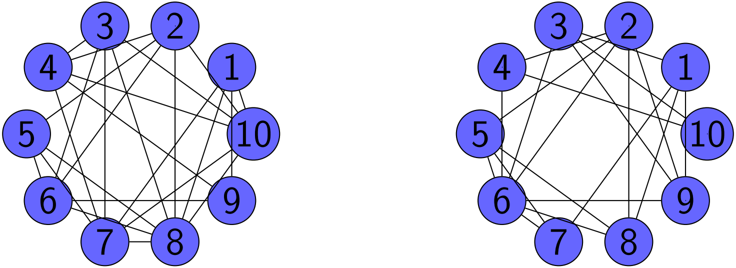




Graph matching: a fundamental mathematical question underlying these applications.

• find a bijection between two vertex sets that maximally align the edges (i.e. minimizes # of adjacency disagreements).

Graph matching (network alignment)

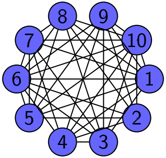


Graph matching: a fundamental mathematical question underlying these applications.

• find a bijection between two vertex sets that maximally align the edges (i.e. minimizes # of adjacency disagreements).

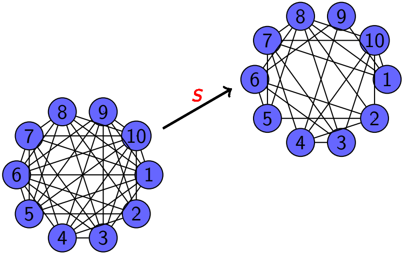
Graph matching is a hard optimization problem (NP-hard), and we seek help from randomness.

An idealistic model: Correlated Erdős-Rényi graphs model



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

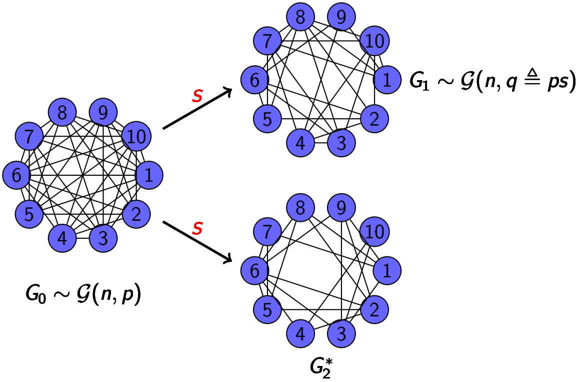
An idealistic model: Correlated Erdős-Rényi graphs model



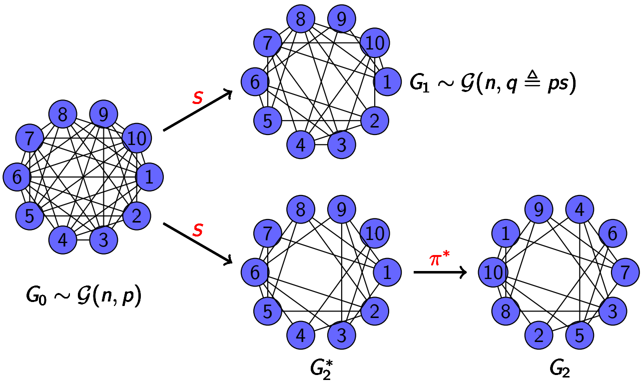
$G_1 \sim \mathcal{G}(n, q \triangleq ps)$

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

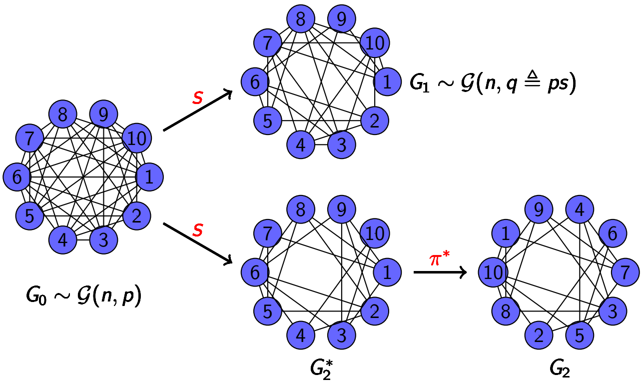
An idealistic model: Correlated Erdős-Rényi graphs model



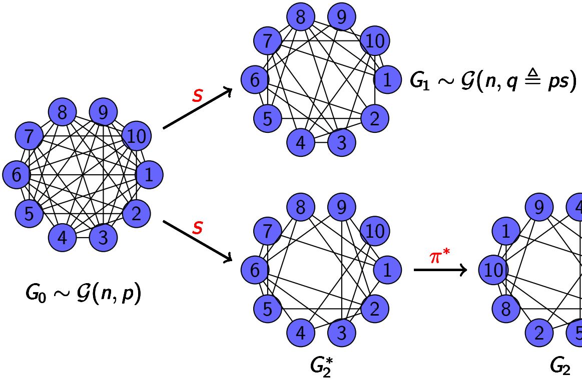
An idealistic model: Correlated Erdős-Rényi graphs model



An idealistic model: Correlated Erdős-Rényi graphs model

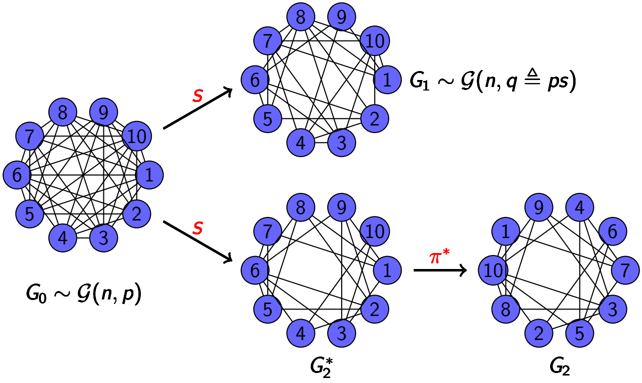


An idealistic model: Correlated Erdős-Rényi graphs model



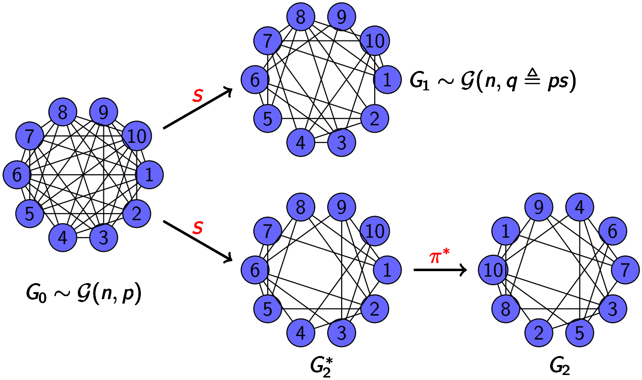
There is no structure in randomness: there is an edge between a pair of vertices with probability p independently.

An idealistic model: Correlated Erdős-Rényi graphs model



There is no structure in randomness: there is an edge between a pair of vertices with probability p independently.
Advantage: simple probabilistic model; suitable playground for developing mathematical theory.

An idealistic model: Correlated Erdős-Rényi graphs model



There is no structure in randomness: there is an edge between a pair of vertices with probability p independently.

Advantage: simple probabilistic model; suitable playground for developing mathematical theory.

Disadvantage: almost all realistic networks are not Erdős-Rényi.

Information threshold: what is the teststone?

Information threshold: what is the teststone?

Three thresholds: detection, exact recovery, partial recovery.

Information threshold: what is the teststone?

Three thresholds: detection, exact recovery, partial recovery.

- Detection: test correlation against independence.

Information threshold: what is the teststone?

Three thresholds: detection, exact recovery, partial recovery.

- Detection: test correlation against independence.
- Exact recovery: correctly match all vertices.

Information threshold: what is the teststone?

Three thresholds: detection, exact recovery, partial recovery.

- Detection: test correlation against independence.
- Exact recovery: correctly match all vertices.
- Partial recovery: correctly match a positive fraction of vertices.

Information threshold: what is the teststone?

Three thresholds: detection, exact recovery, partial recovery.

- Detection: test correlation against independence.
- Exact recovery: correctly match all vertices.
- Partial recovery: correctly match a positive fraction of vertices.

Wu-Xu-Yu'20, 21: progress based on maximal common graph (see Ganassali-Massoulié-Lelarge for $p \approx 1/n$).

Information threshold: what is the teststone?

Three thresholds: detection, exact recovery, partial recovery.

- Detection: test correlation against independence.
- Exact recovery: correctly match all vertices.
- Partial recovery: correctly match a positive fraction of vertices.

Wu-Xu-Yu'20, 21: progress based on maximal common graph (see Ganassali-Massoulié-Lelarge for $p \approx 1/n$).

- Methods: let $\hat{\pi}$ be the bijection that maximizes the number of common edges $\mathcal{E}$.

Information threshold: what is the teststone?

Three thresholds: detection, exact recovery, partial recovery.

- Detection: test correlation against independence.
- Exact recovery: correctly match all vertices.
- Partial recovery: correctly match a positive fraction of vertices.

Wu-Xu-Yu'20, 21: progress based on maximal common graph (see Ganassali-Massoulié-Lelarge for $p \approx 1/n$).

- Methods: let $\hat{\pi}$ be the bijection that maximizes the number of common edges $\mathcal{E}$.
  - ◇ Detection: $|\mathcal{E}|$ is large $\Rightarrow$ correlation.

Information threshold: what is the teststone?

Three thresholds: detection, exact recovery, partial recovery.

- Detection: test correlation against independence.
- Exact recovery: correctly match all vertices.
- Partial recovery: correctly match a positive fraction of vertices.

Wu-Xu-Yu'20, 21: progress based on maximal common graph (see Ganassali-Massoulié-Lelarge for $p \approx 1/n$).

- Methods: let $\hat{\pi}$ be the bijection that maximizes the number of common edges $\mathcal{E}$.
  - ◇ Detection: $|\mathcal{E}|$ is large $\Rightarrow$ correlation.
  - ◇ Matching: estimate π^* by $\hat{\pi}$.

Information threshold: what is the teststone?

Three thresholds: detection, exact recovery, partial recovery.

- Detection: test correlation against independence.
- Exact recovery: correctly match all vertices.
- Partial recovery: correctly match a positive fraction of vertices.

Wu-Xu-Yu'20, 21: progress based on maximal common graph (see Ganassali-Massoulié-Lelarge for $p \approx 1/n$).

- Methods: let $\hat{\pi}$ be the bijection that maximizes the number of common edges $\mathcal{E}$.
  - ⋄ Detection: $|\mathcal{E}|$ is large $\Rightarrow$ correlation.
  - ⋄ Matching: estimate π^* by $\hat{\pi}$.
- Results: determined exact recovery threshold. For detection and partial recovery thresholds, determined exactly in the dense regime ($p = n^{o(1)}$) and determined up-to-constants in the non-dense regime.

Information threshold: what is the teststone?

Three thresholds: detection, exact recovery, partial recovery.

- Detection: test correlation against independence.
- Exact recovery: correctly match all vertices.
- Partial recovery: correctly match a positive fraction of vertices.

Wu-Xu-Yu'20, 21: progress based on maximal common graph (see Ganassali-Massoulié-Lelarge for $p \approx 1/n$).

- Methods: let $\hat{\pi}$ be the bijection that maximizes the number of common edges $\mathcal{E}$.
  - ◇ Detection: $|\mathcal{E}|$ is large $\Rightarrow$ correlation.
  - ◇ Matching: estimate π^* by $\hat{\pi}$.
- Results: determined exact recovery threshold. For detection and partial recovery thresholds, determined exactly in the dense regime ($p = n^{o(1)}$) and determined up-to-constants in the non-dense regime.

- Analysis: when two graphs are correlated it is hard to analyze $\hat{\pi}$ and WXY used "$\hat{\pi} \geq \pi^*$" to lower-bound the maximal common graph.

Exact detection and partial recovery threshold

Exact detection and partial recovery threshold

D.-Du'22a+: Exact detection threshold in the non-dense regime.

Exact detection and partial recovery threshold

D.-Du'22a+: Exact detection threshold in the non-dense regime.

• There is loss in $\hat{\pi} \geq \pi^*$, due to inhomogeniety/irregularity of the common graph under π^* (which is an Erdős-Rényi).

Exact detection and partial recovery threshold

D.-Du'22a+: Exact detection threshold in the non-dense regime.

• There is loss in $\hat{\pi} \geq \pi^*$, due to inhomogeniety/irregularity of the common graph under π^* (which is an Erdős-Rényi).

• Instead of analyzing $\hat{\pi}$, we analyze the densest subgraph.

Exact detection and partial recovery threshold

D.-Du'22a+: Exact detection threshold in the non-dense regime.

• There is loss in $\hat{\pi} \geq \pi^*$, due to inhomogeniety/irregularity of the common graph under π^* (which is an Erdős-Rényi).

• Instead of analyzing $\hat{\pi}$, we analyze the densest subgraph.

• Anantharam-Salez'16: densest subgraph in Erdős-Rényi $\mathcal{G}(n, \lambda/n)$ has average degree $\varrho(\lambda) > \lambda$ (for $\lambda > 1$).

Exact detection and partial recovery threshold

D.-Du'22a+: Exact detection threshold in the non-dense regime.

• There is loss in $\hat{\pi} \geq \pi^*$, due to inhomogeniety/irregularity of the common graph under π^* (which is an Erdős-Rényi).

• Instead of analyzing $\hat{\pi}$, we analyze the densest subgraph.

• Anantharam-Salez'16: densest subgraph in Erdős-Rényi $\mathcal{G}(n, \lambda/n)$ has average degree $\varrho(\lambda) > \lambda$ (for $\lambda > 1$).

• Average degree of the maximal common graph in independent case **does not** increase by taking subgraph with linear size.

Exact detection and partial recovery threshold

D.-Du'22a+: Exact detection threshold in the non-dense regime.

• There is loss in $\hat{\pi} \geq \pi^*$, due to inhomogeniety/irregularity of the common graph under π^* (which is an Erdős-Rényi).

• Instead of analyzing $\hat{\pi}$, we analyze the densest subgraph.

• Anantharam-Salez'16: densest subgraph in Erdős-Rényi $\mathcal{G}(n, \lambda/n)$ has average degree $\varrho(\lambda) > \lambda$ (for $\lambda > 1$).

• Average degree of the maximal common graph in independent case **does not** increase by taking subgraph with linear size.

Detectable regime: *If* densest subgraph with linear size (maximized over all vertex bijections) has large edge density, *then* correlated.

Exact detection and partial recovery threshold

D.-Du'22a+: Exact detection threshold in the non-dense regime.

- There is loss in $\hat{\pi} \geq \pi^*$, due to inhomogeniety/irregularity of the common graph under π^* (which is an Erdős-Rényi).
- Instead of analyzing $\hat{\pi}$, we analyze the densest subgraph.
- Anantharam-Salez'16: densest subgraph in Erdős-Rényi $\mathcal{G}(n, \lambda/n)$ has average degree $\varrho(\lambda) > \lambda$ (for $\lambda > 1$).
- Average degree of the maximal common graph in independent case **does not** increase by taking subgraph with linear size.

Detectable regime: *If* densest subgraph with linear size (maximized over all vertex bijections) has large edge density, *then* correlated.
Undetectable regime: After truncating on the densest subgraph, the second moment of the probability density ratios (null v.s. alternative) is close to 1, so total variation distance vanishes.

Exact detection and partial recovery threshold

D.-Du'22a+: Exact detection threshold in the non-dense regime.

- There is loss in $\hat{\pi} \geq \pi^*$, due to inhomogeniety/irregularity of the common graph under π^* (which is an Erdős-Rényi).
- Instead of analyzing $\hat{\pi}$, we analyze the densest subgraph.
- Anantharam-Salez'16: densest subgraph in Erdős-Rényi $\mathcal{G}(n, \lambda/n)$ has average degree $\varrho(\lambda) > \lambda$ (for $\lambda > 1$).
- Average degree of the maximal common graph in independent case **does not** increase by taking subgraph with linear size.

Detectable regime: *If* densest subgraph with linear size (maximized over all vertex bijections) has large edge density, *then* correlated.
Undetectable regime: After truncating on the densest subgraph, the second moment of the probability density ratios (null v.s. alternative) is close to 1, so total variation distance vanishes.

D.-Du'22b+: Exact partial recovery threshold in the non-dense regime.

Poly-time approximation scheme for maximal overlap

Poly-time approximation scheme for maximal overlap

- Recall: maximal alignment of two graphs are NP-hard.

Poly-time approximation scheme for maximal overlap

- Recall: maximal alignment of two graphs are NP-hard.
- Question: maximizing the overlap between two independent Erdős-Rényi graphs with $p = n^{-\alpha}$ over all vertex matching?

Poly-time approximation scheme for maximal overlap

- Recall: maximal alignment of two graphs are NP-hard.
- Question: maximizing the overlap between two independent Erdős-Rényi graphs with $p = n^{-\alpha}$ over all vertex matching?
- D.-Du-Gong'22: Poly-time approximation scheme for $\alpha \in (1/2, 1)$.

Poly-time approximation scheme for maximal overlap

- Recall: maximal alignment of two graphs are NP-hard.
- Question: maximizing the overlap between two independent Erdős-Rényi graphs with $p = n^{-\alpha}$ over all vertex matching?
- D.-Du-Gong'22: Poly-time approximation scheme for $\alpha \in (1/2, 1)$.
  - Proves that the maximal overlap is $\approx (2\alpha - 1)^{-1} n$ (even no non-algorithmic proof previously).

Poly-time approximation scheme for maximal overlap

- Recall: maximal alignment of two graphs are NP-hard.
- Question: maximizing the overlap between two independent Erdős-Rényi graphs with $p = n^{-\alpha}$ over all vertex matching?
- D.-Du-Gong'22: Poly-time approximation scheme for $\alpha \in (1/2, 1)$.
  - Proves that the maximal overlap is $\approx (2\alpha - 1)^{-1} n$ (even no non-algorithmic proof previously).
  - Our algorithm is iterative/greedy; analysis is difficult.

Poly-time approximation scheme for maximal overlap

- Recall: maximal alignment of two graphs are NP-hard.
- Question: maximizing the overlap between two independent Erdős-Rényi graphs with $p = n^{-\alpha}$ over all vertex matching?
- D.-Du-Gong'22: Poly-time approximation scheme for $\alpha \in (1/2, 1)$.
  - Proves that the maximal overlap is $\approx (2\alpha - 1)^{-1} n$ (even no non-algorithmic proof previously).
  - Our algorithm is iterative/greedy; analysis is difficult.
  - $\alpha < 1/2$: trivial regime; $\alpha = 1/2$: delicate.

Poly-time approximation scheme for maximal overlap

- Recall: maximal alignment of two graphs are NP-hard.
- Question: maximizing the overlap between two independent Erdős-Rényi graphs with $p = n^{-\alpha}$ over all vertex matching?
- D.-Du-Gong'22: Poly-time approximation scheme for $\alpha \in (1/2, 1)$.
  - Proves that the maximal overlap is $\approx (2\alpha - 1)^{-1} n$ (even no non-algorithmic proof previously).
  - Our algorithm is iterative/greedy; analysis is difficult.
  - $\alpha < 1/2$: trivial regime; $\alpha = 1/2$: delicate.
- One of few examples where random optimization can be solved but worst-case NP hard.

Poly-time approximation scheme for maximal overlap

- Recall: maximal alignment of two graphs are NP-hard.
- Question: maximizing the overlap between two independent Erdős-Rényi graphs with $p = n^{-\alpha}$ over all vertex matching?
- D.-Du-Gong'22: Poly-time approximation scheme for $\alpha \in (1/2, 1)$.
  - Proves that the maximal overlap is $\approx (2\alpha - 1)^{-1} n$ (even no non-algorithmic proof previously).
  - Our algorithm is iterative/greedy; analysis is difficult.
  - $\alpha < 1/2$: trivial regime; $\alpha = 1/2$: delicate.
- One of few examples where random optimization can be solved but worst-case NP hard.
- Wu-Xu: maybe related to overlap gap property for correlated model.

Poly-time approximation scheme for maximal overlap

- Recall: maximal alignment of two graphs are NP-hard.
- Question: maximizing the overlap between two independent Erdős-Rényi graphs with $p = n^{-\alpha}$ over all vertex matching?
- D.-Du-Gong'22: Poly-time approximation scheme for $\alpha \in (1/2, 1)$.
  - Proves that the maximal overlap is $\approx (2\alpha - 1)^{-1} n$ (even no non-algorithmic proof previously).
  - Our algorithm is iterative/greedy; analysis is difficult.
  - $\alpha < 1/2$: trivial regime; $\alpha = 1/2$: delicate.
- One of few examples where random optimization can be solved but worst-case NP hard.
- Wu-Xu: maybe related to overlap gap property for correlated model.
- Computation for correlated model seems much more difficult.

Matching algorithms for correlated graphs (up to 2021)

Matching algorithms for correlated graphs (up to 2021)

- For each vertex, compute a "signature" and match pairs of vertices with similar signatures.

Matching algorithms for correlated graphs (up to 2021)

- For each vertex, compute a "signature" and match pairs of vertices with similar signatures. Desired properties for signature: **informative**, **comptuable**, **tractable**, generalizable.

Matching algorithms for correlated graphs (up to 2021)

- For each vertex, compute a "signature" and match pairs of vertices with similar signatures. Desired properties for signature: **informative**, **comptuable**, **tractable**, generalizable.

  ◇ By randomness, true pairs are more similar than faked ones.

Matching algorithms for correlated graphs (up to 2021)

- For each vertex, compute a "signature" and match pairs of vertices with similar signatures. Desired properties for signature: **informative**, **comptuable**, **tractable**, generalizable.
  ◇ By randomness, true pairs are more similar than faked ones.
  ◇ Challenge: true pair needs to beat **many** faked pairs.

Matching algorithms for correlated graphs (up to 2021)

- For each vertex, compute a "signature" and match pairs of vertices with similar signatures. Desired properties for signature: **informative**, **comptuable**, **tractable**, generalizable.
  - ◇ By randomness, true pairs are more similar than faked ones.
  - ◇ Challenge: true pair needs to beat **many** faked pairs.
- Dai-Cullina-Kiyavash-Grossglauser'18, Barak-Chou-Lei-Schramm-Sheng'19 D.-Ma-Wu-Xu'21

Matching algorithms for correlated graphs (up to 2021)

- For each vertex, compute a "signature" and match pairs of vertices with similar signatures. Desired properties for signature: **informative**, **comptuable**, **tractable**, generalizable.
  - ◇ By randomness, true pairs are more similar than faked ones.
  - ◇ Challenge: true pair needs to beat **many** faked pairs.
- Dai-Cullina-Kiyavash-Grossglauser'18, Barak-Chou-Lei-Schramm-Sheng'19 D.-Ma-Wu-Xu'21
- Optimization with relaxation (usually convex relaxation). (Fan-Mao-Wu-Xu'19+).

Matching algorithms for correlated graphs (up to 2021)

- For each vertex, compute a "signature" and match pairs of vertices with similar signatures. Desired properties for signature: **informative**, **comptuable**, **tractable**, generalizable.
  $\diamond$ By randomness, true pairs are more similar than faked ones.
  $\diamond$ Challenge: true pair needs to beat **many** faked pairs.
- Dai-Cullina-Kiyavash-Grossglauser'18, Barak-Chou-Lei-Schramm-Sheng'19 D.-Ma-Wu-Xu'21
- Optimization with relaxation (usually convex relaxation). (Fan-Mao-Wu-Xu'19+).
  - Original optimization problem is hard to solve, but feasible if enlarge the space of potential solutions (e.g. to a convex space).

Matching algorithms for correlated graphs (up to 2021)

- For each vertex, compute a "signature" and match pairs of vertices with similar signatures. Desired properties for signature: **informative**, **comptuable**, **tractable**, generalizable.

  ◇ By randomness, true pairs are more similar than faked ones.

  ◇ Challenge: true pair needs to beat **many** faked pairs.

- Dai-Cullina-Kiyavash-Grossglauser'18, Barak-Chou-Lei-Schramm-Sheng'19 D.-Ma-Wu-Xu'21

- Optimization with relaxation (usually convex relaxation). (Fan-Mao-Wu-Xu'19+).
  - Original optimization problem is hard to solve, but feasible if enlarge the space of potential solutions (e.g. to a convex space).
  - By randomness, the optimizer in the enlarged space recovers the original optimizer (e.g., by rounding procedure).

Matching algorithms for correlated graphs (up to 2021)

- For each vertex, compute a "signature" and match pairs of vertices with similar signatures. Desired properties for signature: **informative**, **comptuable**, **tractable**, generalizable.
  ◇ By randomness, true pairs are more similar than faked ones.
  ◇ Challenge: true pair needs to beat **many** faked pairs.
- Dai-Cullina-Kiyavash-Grossglauser'18, Barak-Chou-Lei-Schramm-Sheng'19 D.-Ma-Wu-Xu'21
- Optimization with relaxation (usually convex relaxation). (Fan-Mao-Wu-Xu'19+).
  - Original optimization problem is hard to solve, but feasible if enlarge the space of potential solutions (e.g. to a convex space).
  - By randomness, the optimizer in the enlarged space recovers the original optimizer (e.g., by rounding procedure).
  - Succeeds when noise decays in $\mathrm{polylog}(n)$.

Recent progress on matching algorithms

Recent progress on matching algorithms

- Mao-Rudelson-Tikhomirov'21$+$: poly-time algorithm based some partition trees, when correlation $\geq \mathrm{const}$ (close to 1).

Recent progress on matching algorithms

- Mao-Rudelson-Tikhomirov'21$+$: poly-time algorithm based some partition trees, when correlation $\geq \mathrm{const}$ (close to 1).
- Ganassali-Massoulié-Lelarge'20$+$,22$+$: poly-time partial matching algorithm for sparse graphs based on message passing, when correlation $> \sqrt{\text{Otter's constant}} \approx \sqrt{0.3383}$.

Recent progress on matching algorithms

- Mao-Rudelson-Tikhomirov'21+: poly-time algorithm based some partition trees, when correlation $\geq \mathrm{const}$ (close to 1).
- Ganassali-Massoulié-Lelarge'20+,22+: poly-time partial matching algorithm for sparse graphs based on message passing, when correlation $> \sqrt{\text{Otter's constant}} \approx \sqrt{0.3383}$.
- Mao-Wu-Xu-Yu'22+: poly-time algorithm when correlation $> \sqrt{\text{Otter's constant}}$, based on a carefully curated family of rooted trees called chandeliers (substantially improving MRT21+, and covers much wider parameter regime).

Recent progress on matching algorithms

- Mao-Rudelson-Tikhomirov'21+: poly-time algorithm based some partition trees, when correlation $\geq \mathrm{const}$ (close to 1).
- Ganassali-Massoulié-Lelarge'20+,22+: poly-time partial matching algorithm for sparse graphs based on message passing, when correlation $> \sqrt{\text{Otter's constant}} \approx \sqrt{0.3383}$.
- Mao-Wu-Xu-Yu'22+: poly-time algorithm when correlation $> \sqrt{\text{Otter's constant}}$, based on a carefully curated family of rooted trees called chandeliers (substantially improving MRT21+, and covers much wider parameter regime).
- D.-Li'22+: poly-time iterative algorithm for matching Gaussian matrices when correlation is non-vanishing.

Recent progress on matching algorithms

- Mao-Rudelson-Tikhomirov'21+: poly-time algorithm based some partition trees, when correlation $\geq \mathrm{const}$ (close to 1).
- Ganassali-Massoulié-Lelarge'20+,22+: poly-time partial matching algorithm for sparse graphs based on message passing, when correlation $> \sqrt{\text{Otter's constant}} \approx \sqrt{0.3383}$.
- Mao-Wu-Xu-Yu'22+: poly-time algorithm when correlation $> \sqrt{\text{Otter's constant}}$, based on a carefully curated family of rooted trees called chandeliers (substantially improving MRT21+, and covers much wider parameter regime).
- D.-Li'22+: poly-time iterative algorithm for matching Gaussian matrices when correlation is non-vanishing.
  - New feature: signal is stored in a vector where each coordinate is a pair of sets, and signal per coordinate decreases with iteration but compensated by increase on dimension.

Recent progress on matching algorithms

- Mao-Rudelson-Tikhomirov'21+: poly-time algorithm based some partition trees, when correlation $\geq \mathrm{const}$ (close to 1).
- Ganassali-Massoulié-Lelarge'20+,22+: poly-time partial matching algorithm for sparse graphs based on message passing, when correlation $> \sqrt{\text{Otter's constant}} \approx \sqrt{0.3383}$.
- Mao-Wu-Xu-Yu'22+: poly-time algorithm when correlation $> \sqrt{\text{Otter's constant}}$, based on a carefully curated family of rooted trees called chandeliers (substantially improving MRT21+, and covers much wider parameter regime).
- D.-Li'22+: poly-time iterative algorithm for matching Gaussian matrices when correlation is non-vanishing.
  - New feature: signal is stored in a vector where each coordinate is a pair of sets, and signal per coordinate decreases with iteration but compensated by increase on dimension.
  - Expected to be sharp, and **should** extend to graph matching (although with substantial challenge) assuming $np > n^{\alpha}$.

Recent progress on matching algorithms

- Mao-Rudelson-Tikhomirov'21+: poly-time algorithm based some partition trees, when correlation $\geq \mathrm{const}$ (close to 1).
- Ganassali-Massoulié-Lelarge'20+,22+: poly-time partial matching algorithm for sparse graphs based on message passing, when correlation $> \sqrt{\text{Otter's constant}} \approx \sqrt{0.3383}$.
- Mao-Wu-Xu-Yu'22+: poly-time algorithm when correlation $> \sqrt{\text{Otter's constant}}$, based on a carefully curated family of rooted trees called chandeliers (substantially improving MRT21+, and covers much wider parameter regime).
- D.-Li'22+: poly-time iterative algorithm for matching Gaussian matrices when correlation is non-vanishing.
  - New feature: signal is stored in a vector where each coordinate is a pair of sets, and signal per coordinate decreases with iteration but compensated by increase on dimension.
  - Expected to be sharp, and **should** extend to graph matching (although with substantial challenge) assuming $np > n^{\alpha}$.
  - Shed lights on many matching problems too.

Complexity theory: how to certify computational hardness?

- Traditionally, complexity theory studies hardness of computational problems for worst-case instance.

Complexity theory: how to certify computational hardness?

- Traditionally, complexity theory studies hardness of computational problems for worst-case instance.
- Usually certify hardness by reduction: if you could solve this problem, then you can solve some well-known hard problems.

Complexity theory: how to certify computational hardness?

- Traditionally, complexity theory studies hardness of computational problems for worst-case instance.
- Usually certify hardness by reduction: if you could solve this problem, then you can solve some well-known hard problems.
- For problems with random instance, we care about the hardness for a typical instance. **Evidences** of hardness include

Complexity theory: how to certify computational hardness?

- Traditionally, complexity theory studies hardness of computational problems for worst-case instance.
- Usually certify hardness by reduction: if you could solve this problem, then you can solve some well-known hard problems.
- For problems with random instance, we care about the hardness for a typical instance. **Evidences** of hardness include
  - show as hard as well-known hard problems (reduction is much more difficult on random instance than for worst-case instance);

Complexity theory: how to certify computational hardness?

- Traditionally, complexity theory studies hardness of computational problems for worst-case instance.
- Usually certify hardness by reduction: if you could solve this problem, then you can solve some well-known hard problems.
- For problems with random instance, we care about the hardness for a typical instance. **Evidences** of hardness include
  - show as hard as well-known hard problems (reduction is much more difficult on random instance than for worst-case instance);
  - show that a wide class of algorithms fail to solve the problem;

Complexity theory: how to certify computational hardness?

- Traditionally, complexity theory studies hardness of computational problems for worst-case instance.
- Usually certify hardness by reduction: if you could solve this problem, then you can solve some well-known hard problems.
- For problems with random instance, we care about the hardness for a typical instance. **Evidences** of hardness include
  - show as hard as well-known hard problems (reduction is much more difficult on random instance than for worst-case instance);
  - show that a wide class of algorithms fail to solve the problem;
  - exhibit similar structural properties as in other hard problems.

Complexity theory: how to certify computational hardness?

- Traditionally, complexity theory studies hardness of computational problems for worst-case instance.
- Usually certify hardness by reduction: if you could solve this problem, then you can solve some well-known hard problems.
- For problems with random instance, we care about the hardness for a typical instance. **Evidences** of hardness include
  - show as hard as well-known hard problems (reduction is much more difficult on random instance than for worst-case instance);
  - show that a wide class of algorithms fail to solve the problem;
  - exhibit similar structural properties as in other hard problems.
- Application in data privacy: how can we perform a minimal change on the Linkedin and Twitter network, so that it would be computationally hard to recover the matching from the this perturbed observation?

Complexity theory: how to certify computational hardness?

- Traditionally, complexity theory studies hardness of computational problems for worst-case instance.
- Usually certify hardness by reduction: if you could solve this problem, then you can solve some well-known hard problems.
- For problems with random instance, we care about the hardness for a typical instance. **Evidences** of hardness include
  - show as hard as well-known hard problems (reduction is much more difficult on random instance than for worst-case instance);
  - show that a wide class of algorithms fail to solve the problem;
  - exhibit similar structural properties as in other hard problems.
- Application in data privacy: how can we perform a minimal change on the Linkedin and Twitter network, so that it would be computationally hard to recover the matching from the this perturbed observation?
- **Information-computation gap**: a **major** challenge in many random combinatorial optimization and constraint satisfaction problems!

Perspectives and future directions

- A hub of theorists: combinatorics, probability, statistics, algorithms, complexity theory, optimization, etc.

Perspectives and future directions

- A hub of theorists: combinatorics, probability, statistics, algorithms, complexity theory, optimization, etc.
- A meeting point of theory and applications:

Perspectives and future directions

- A hub of theorists: combinatorics, probability, statistics, algorithms, complexity theory, optimization, etc.
- A meeting point of theory and applications:
  - Currently, most extensively studied models are idealistic.

Perspectives and future directions

- A hub of theorists: combinatorics, probability, statistics, algorithms, complexity theory, optimization, etc.
- A meeting point of theory and applications:
  - Currently, most extensively studied models are idealistic. Even worse, many times algorithms and analysis are based on wrong model assumptions, e.g., local tree structure for social network model.

Perspectives and future directions

- A hub of theorists: combinatorics, probability, statistics, algorithms, complexity theory, optimization, etc.
- A meeting point of theory and applications:
  - Currently, most extensively studied models are idealistic. Even worse, many times algorithms and analysis are based on wrong model assumptions, e.g., local tree structure for social network model.
  - Major challenge 1: propose models with general applicability where theorists can say something meaningful.

Perspectives and future directions

- A hub of theorists: combinatorics, probability, statistics, algorithms, complexity theory, optimization, etc.
- A meeting point of theory and applications:
  - Currently, most extensively studied models are idealistic. Even worse, many times algorithms and analysis are based on wrong model assumptions, e.g., local tree structure for social network model.
  - Major challenge 1: propose models with general applicability where theorists can say something meaningful.
  - Major challenge 2: propose models for important scientific problems worth extensive theoretic study.

Perspectives and future directions

- A hub of theorists: combinatorics, probability, statistics, algorithms, complexity theory, optimization, etc.
- A meeting point of theory and applications:
  - Currently, most extensively studied models are idealistic. Even worse, many times algorithms and analysis are based on wrong model assumptions, e.g., local tree structure for social network model.
  - Major challenge 1: propose models with general applicability where theorists can say something meaningful.
  - Major challenge 2: propose models for important scientific problems worth extensive theoretic study.
- Bridging what is wanted with what is possible.

Reference: all mentioned works available on arXiv.