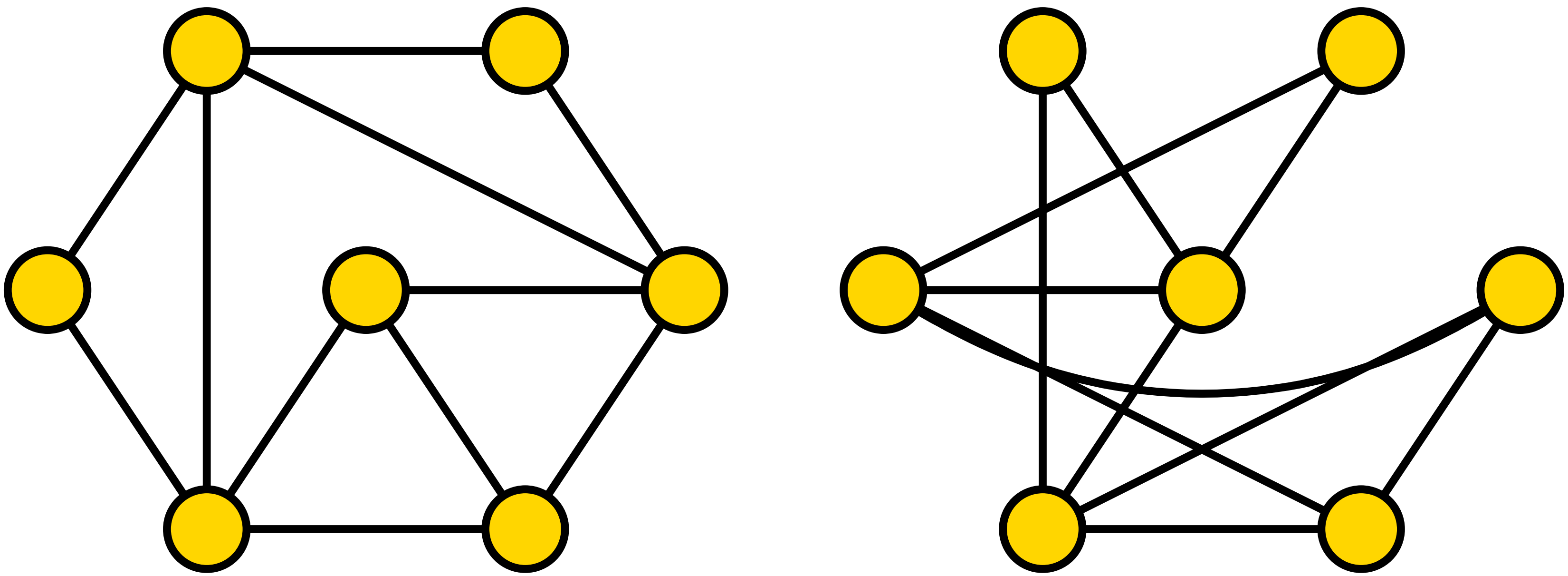
A Near-Optimal Algorithm for Testing Isomorphism of Two Unknown Graphs

Krzysztof Onak
IBM T.J. Watson Research Center

Joint work with **Xiaorui Sun** (Columbia)

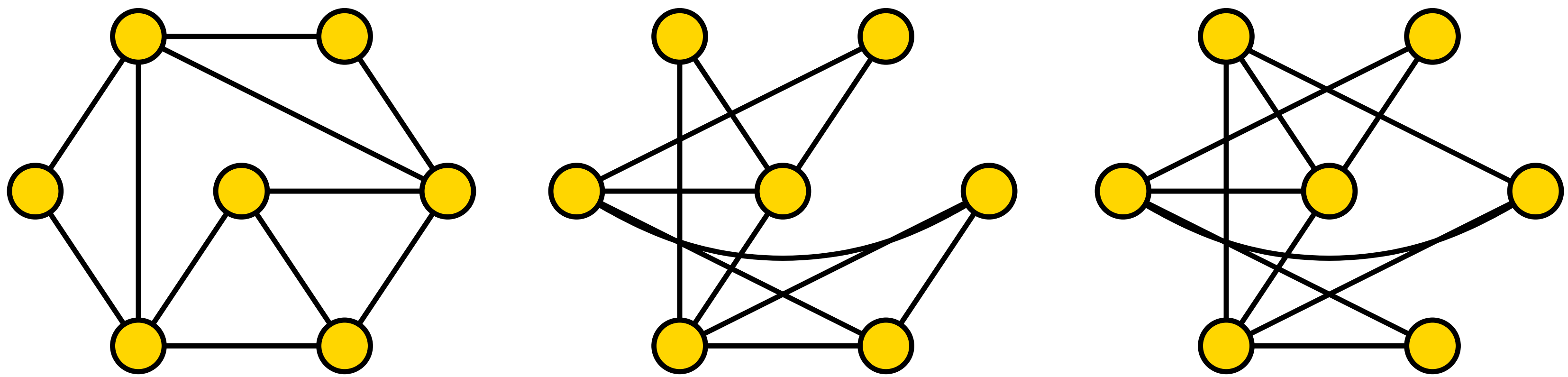
Graph Isomorphism

- Problem: Are two graphs identical?



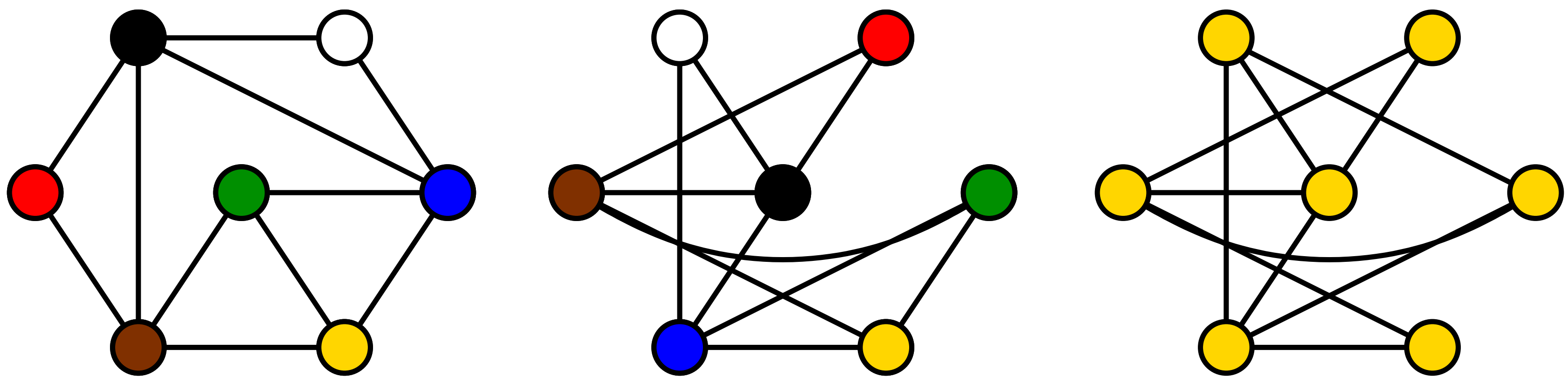
Graph Isomorphism

- Problem: Are two graphs identical?



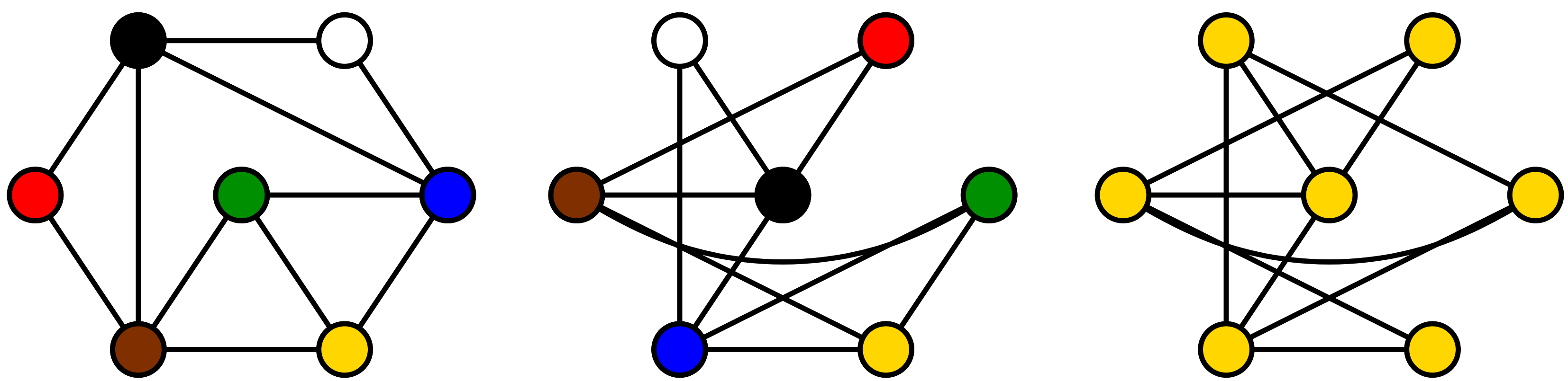
Graph Isomorphism

- Problem: Are two graphs identical?



Graph Isomorphism

- Problem: Are two graphs identical?

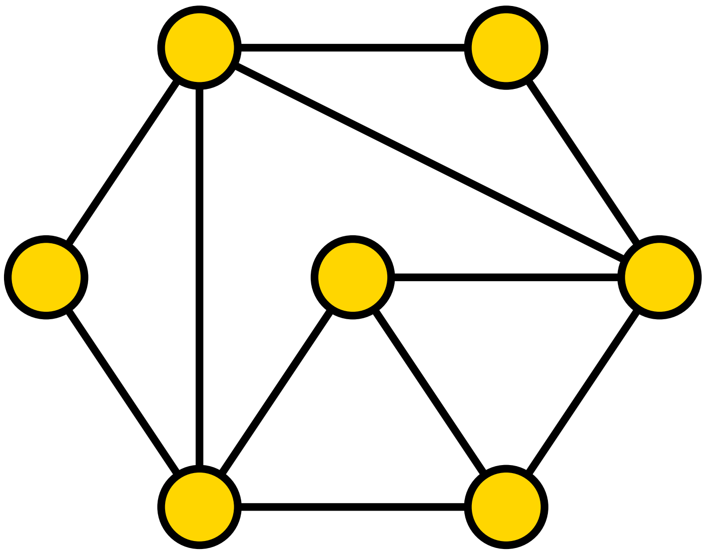


Status:

- Not known to be in P or NP-hard
- Best known algorithm: $2^{\tilde{O}(\sqrt{n})}$ time (early 1980's)

Property Testing Version

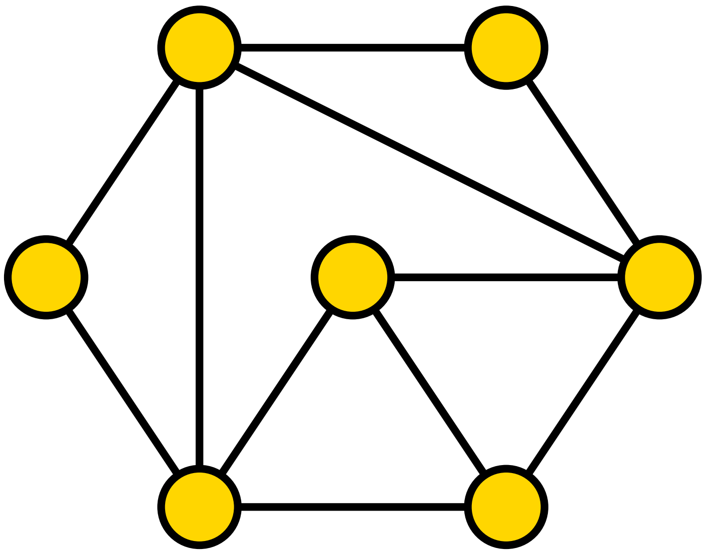
- This Talk: Dense graph model



	1	1	0	1	1	0
1		0	0	1	0	0
1	0		0	0	1	0
0	0	0		1	1	1
1	1	0	1		0	1
1	0	1	1	0		1
0	0	0	1	1	1	

Property Testing Version

- This Talk: Dense graph model

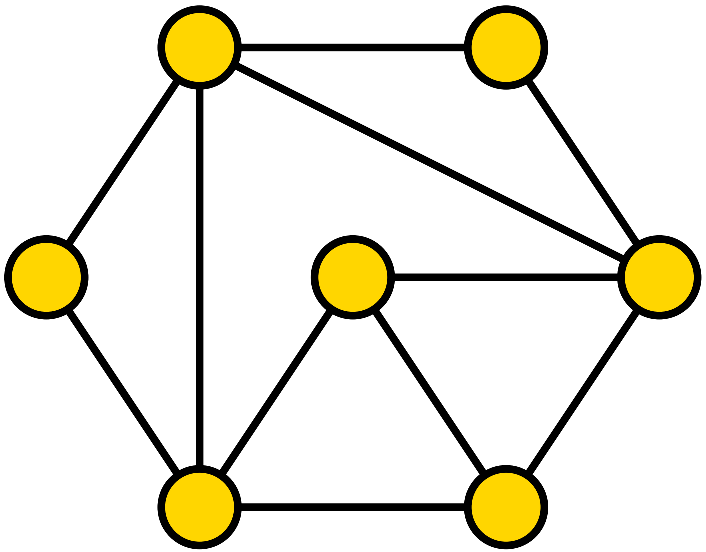


	1	1	0	1	1	0
1		0	0	1	0	0
1	0		0	0	1	0
0	0	0		1	1	1
1	1	0	1		0	1
1	0	1	1	0		1
0	0	0	1	1	1	

- Requirements:
  - accept with probability 9/10 if graphs isomorphic
  - reject with probability 9/10 if for any matching of vertices at least ϵn^2 edges disagree

Property Testing Version

- This Talk: Dense graph model



	1	1	0	1	1	0
1		0	0	1	0	0
1	0		0	0	1	0
0	0	0		1	1	1
1	1	0	1		0	1
1	0	1	1	0		1
0	0	0	1	1	1	

- Requirements:
  - accept with probability 9/10 if graphs isomorphic
  - reject with probability 9/10 if for any matching of vertices at least ϵn^2 edges disagree
- This talk: focus on the complexity as a function of n, assume that ϵ is a small constant (say, $\epsilon = 10^{-9}$)

Query Complexity

- Fischer, Matsliah (2006):

	Upper bound	Lower bound
One sided error, one graph known	$\widetilde{O}(n)$	$\Omega(n)$
One sided error, both graphs unknown	$\widetilde{O}(n^{3/2})$	$\Omega(n^{3/2})$
Two sided error, one graph known	$\widetilde{O}(n^{1/2})$	$\Omega(n^{1/2})$
Two sided error, both graphs unknown	$\widetilde{O}(n^{5/4})$	$\Omega(n)$

(one sided error = never reject if isomorphic)

Query Complexity

- Fischer, Matsliah (2006):

	Upper bound	Lower bound
One sided error, one graph known	$\widetilde{O}(n)$	$\Omega(n)$
One sided error, both graphs unknown	$\widetilde{O}(n^{3/2})$	$\Omega(n^{3/2})$
Two sided error, one graph known	$\widetilde{O}(n^{1/2})$	$\Omega(n^{1/2})$
Two sided error, both graphs unknown	$\widetilde{O}(n^{5/4})$	$\Omega(n)$

(one sided error = never reject if isomorphic)

- Remaining open case:

neither graph known, two-sided testing

Query Complexity

- Fischer, Matsliah (2006):

	Upper bound	Lower bound
One sided error, one graph known	$\widetilde{O}(n)$	$\Omega(n)$
One sided error, both graphs unknown	$\widetilde{O}(n^{3/2})$	$\Omega(n^{3/2})$
Two sided error, one graph known	$\widetilde{O}(n^{1/2})$	$\Omega(n^{1/2})$
Two sided error, both graphs unknown	$\widetilde{O}(n^{5/4})$	$\Omega(n)$

(one sided error = never reject if isomorphic)

- Remaining open case:

  neither graph known, two-sided testing

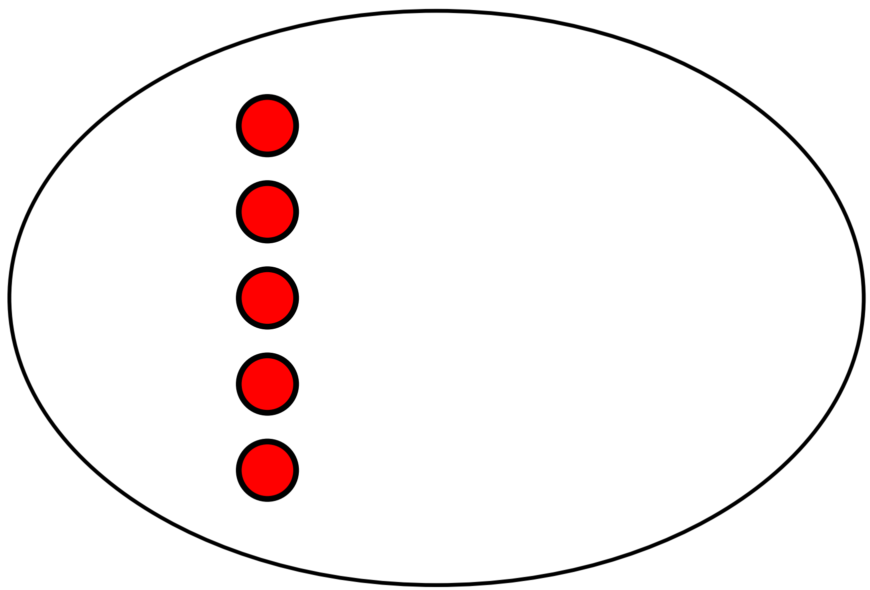
- Our result:

  Algorithm that makes $n \cdot 2^{O(\sqrt{\log n})}$ queries

Review of Fischer-Matsliah Techniques

Core sets

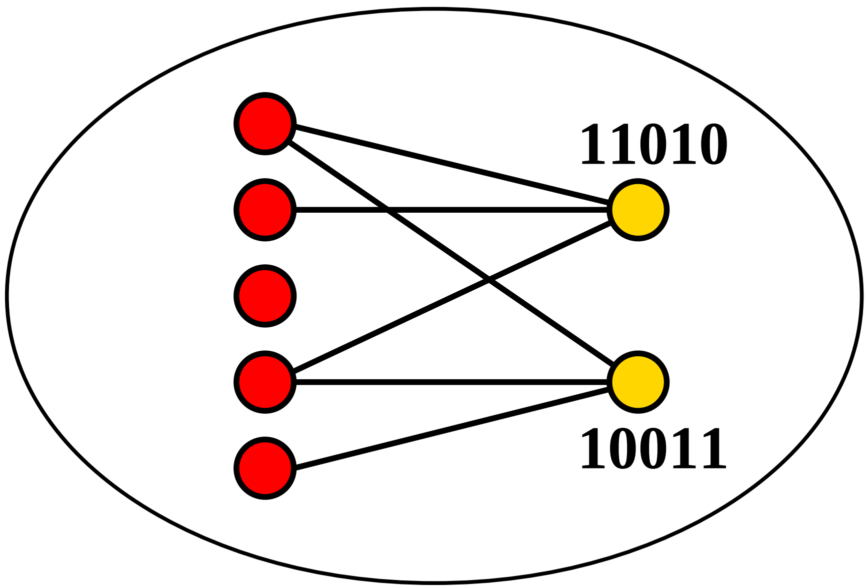
- Core set = list of $\mathrm{polylog}\, n$ vertices $(v_1, v_2, \ldots, v_k)$



Core sets

- Core set = list of $\mathrm{polylog}\, n$ vertices $(v_1, v_2, \ldots, v_k)$
- Every vertex u has a label $l \in \{0,1\}^k$ with respect to core set:

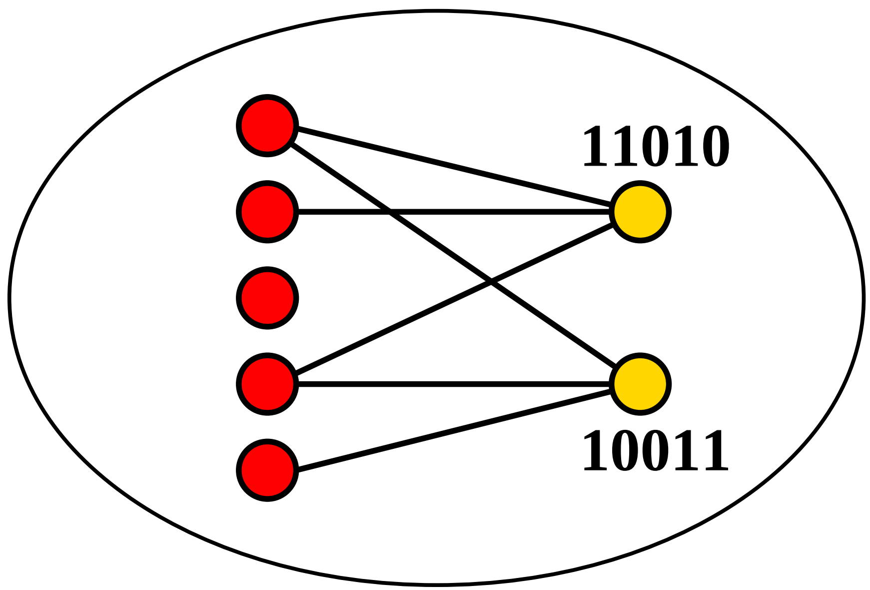
$$l_i = \begin{cases} 1 & \text{if } v_u \text{ and } u \text{ connected} \\ 0 & \text{otherwise} \end{cases}$$



Core sets

- Core set = list of $\text{polylog}\, n$ vertices $(v_1, v_2, \ldots, v_k)$
- Every vertex u has a label $l \in \{0,1\}^k$ with respect to core set:

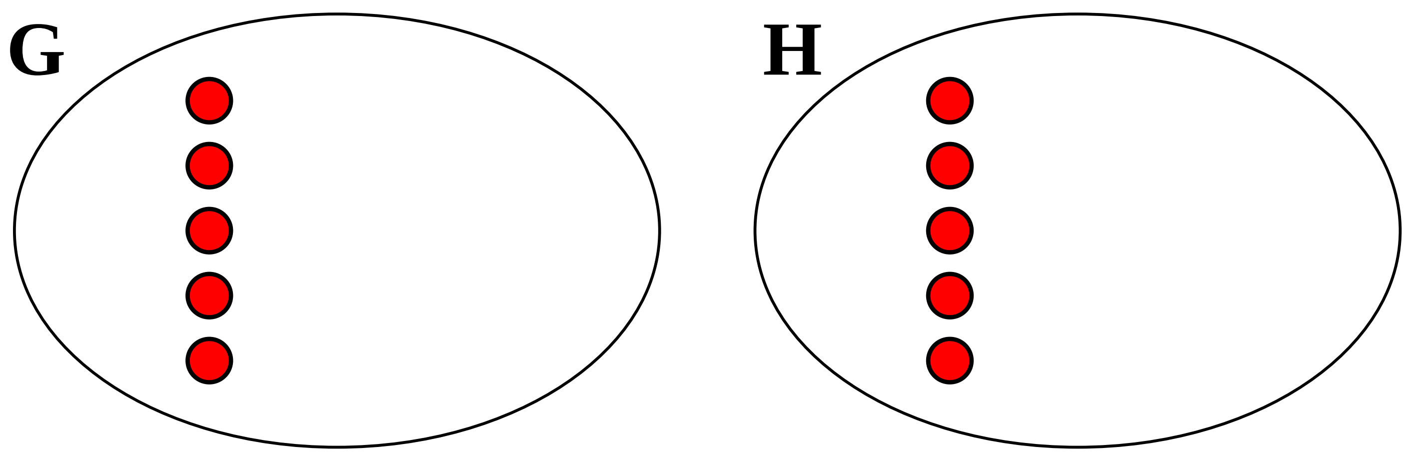
$$l_i = \begin{cases} 1 & \text{if } v_u \text{ and } u \text{ connected} \\ 0 & \text{otherwise} \end{cases}$$



- Intuition: a large random core set partitions vertices into sets of similar vertices

Important Theorem

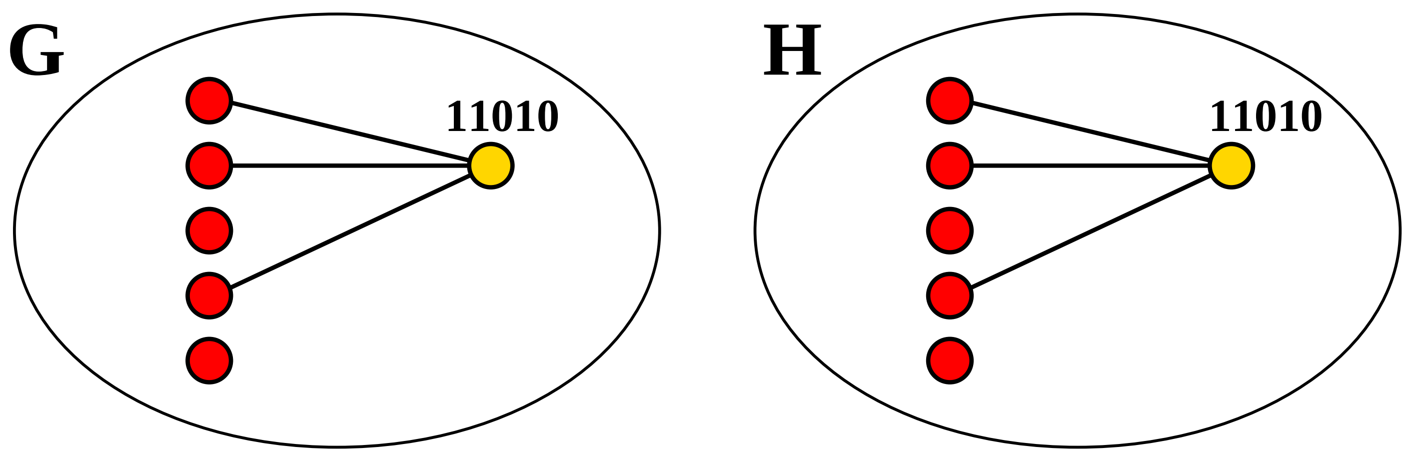
If G and H have core sets $(v_1, \ldots, v_k)$ and $(w_1, \ldots, w_k)$, respectively, such that:



Important Theorem

If G and H have core sets $(v_1, \ldots, v_k)$ and $(w_1, \ldots, w_k)$, respectively, such that:

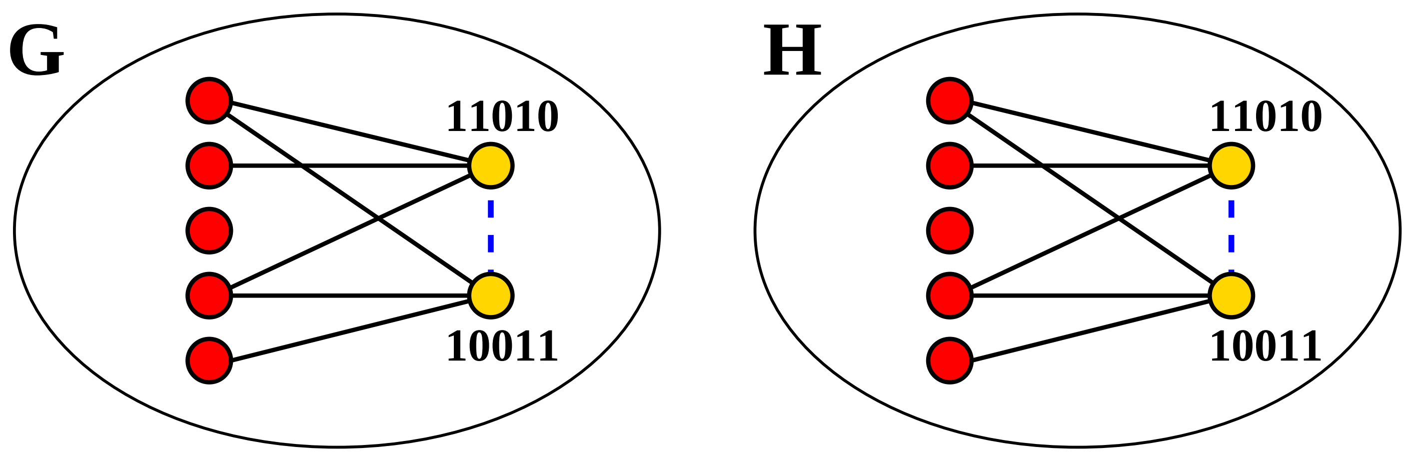
- the distributions on labels are ϵ_1-close



Important Theorem

If G and H have core sets $(v_1, \ldots, v_k)$ and $(w_1, \ldots, w_k)$, respectively, such that:

- the distributions on labels are ϵ_1-close
- if you take two random vertices in G and two random vertices in H with the same labels, then w.p. $1 - \epsilon_2$, the connectivity is the same

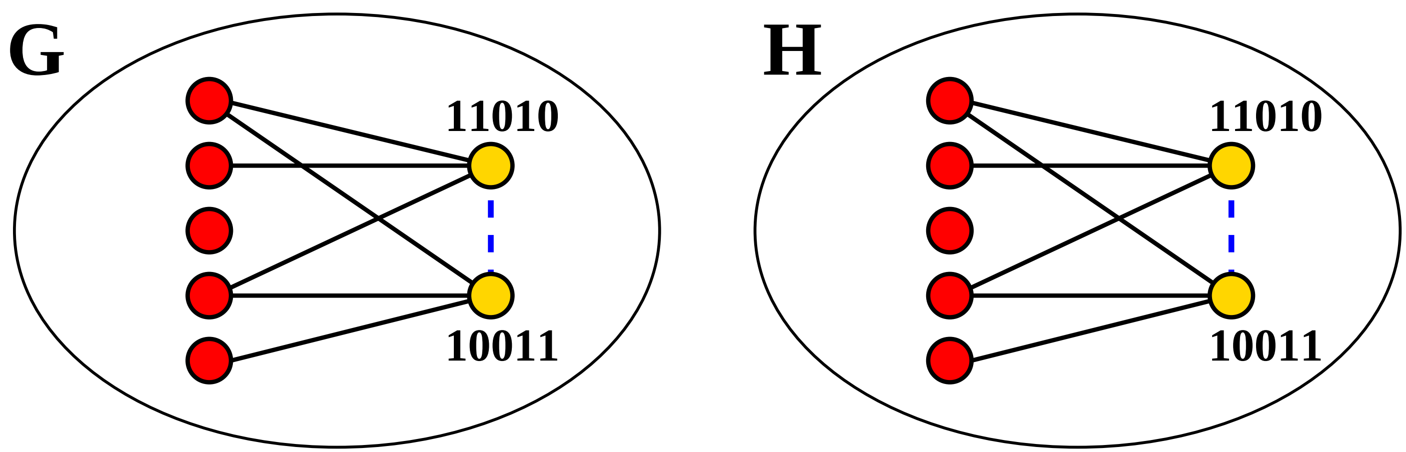


Important Theorem

If G and H have core sets $(v_1, \ldots, v_k)$ and $(w_1, \ldots, w_k)$, respectively, such that:

- the distributions on labels are ϵ_1-close
- if you take two random vertices in G and two random vertices in H with the same labels, then w.p. $1 - \epsilon_2$, the connectivity is the same

then G and H are ϵ_3-close to being isomorphic

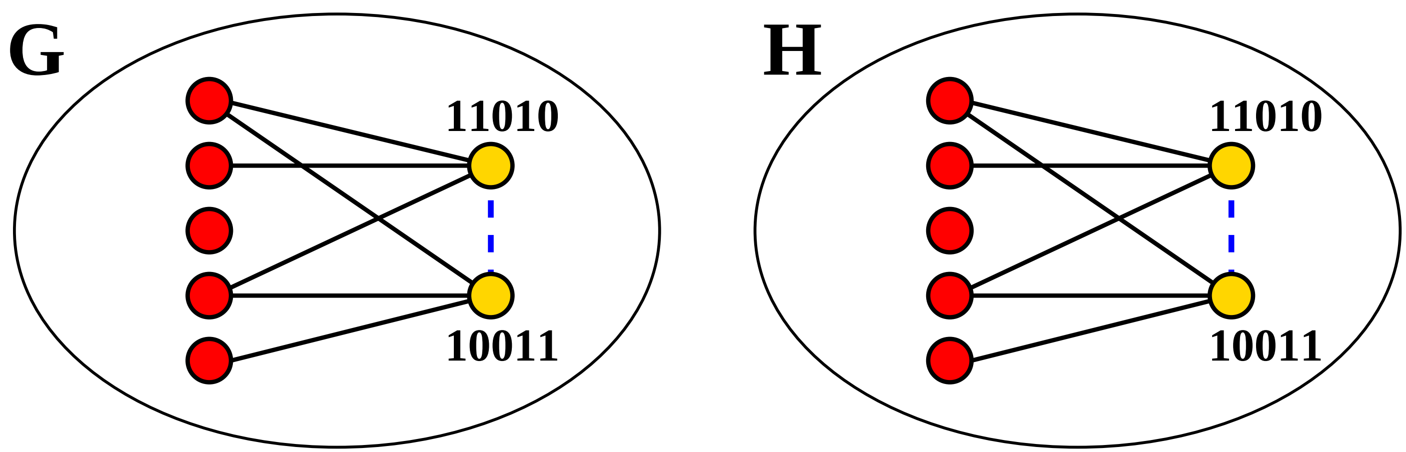


Important Theorem

If G and H have core sets $(v_1, \ldots, v_k)$ and $(w_1, \ldots, w_k)$, respectively, such that:

- the distributions on labels are ϵ_1-close
- if you take two random vertices in G and two random vertices in H with the same labels, then w.p. $1 - \epsilon_2$, the connectivity is the same

then G and H are ϵ_3-close to being isomorphic

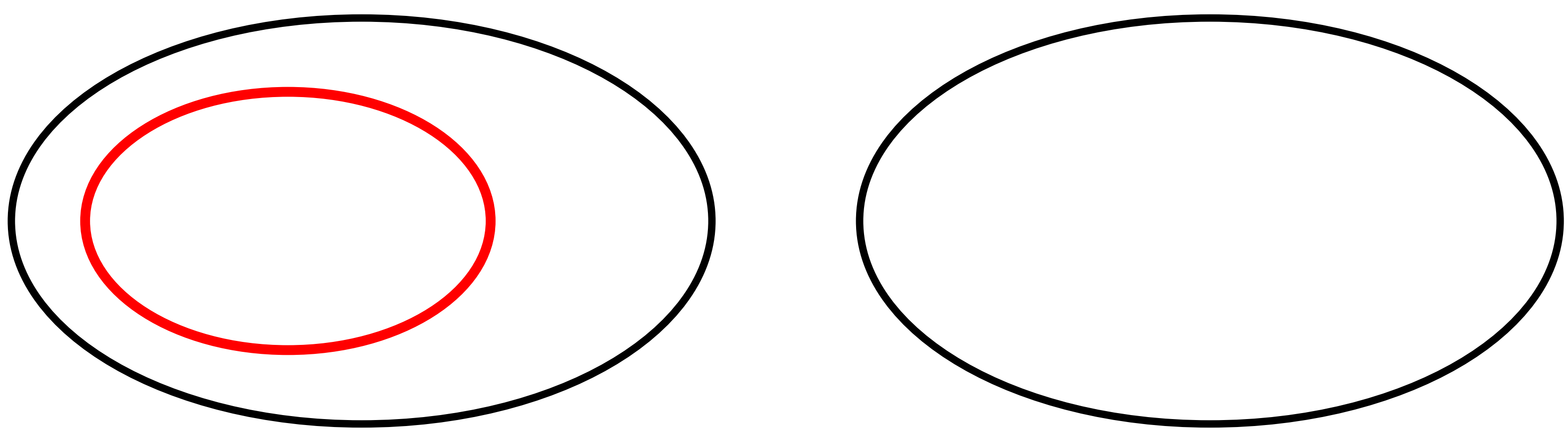


Generic tester:

Search for such a pair of core sets and accept if found

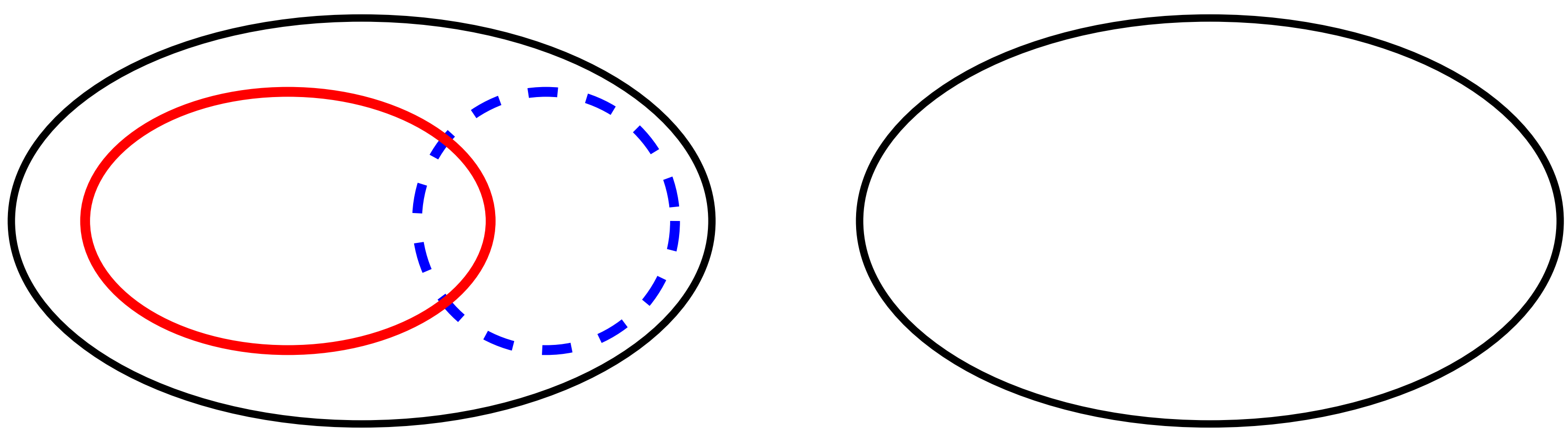
The Fischer-Matsliah Algorithm

- Select $\tilde{O}(n^{3/4})$ random vertices $V_G^\star$ from G



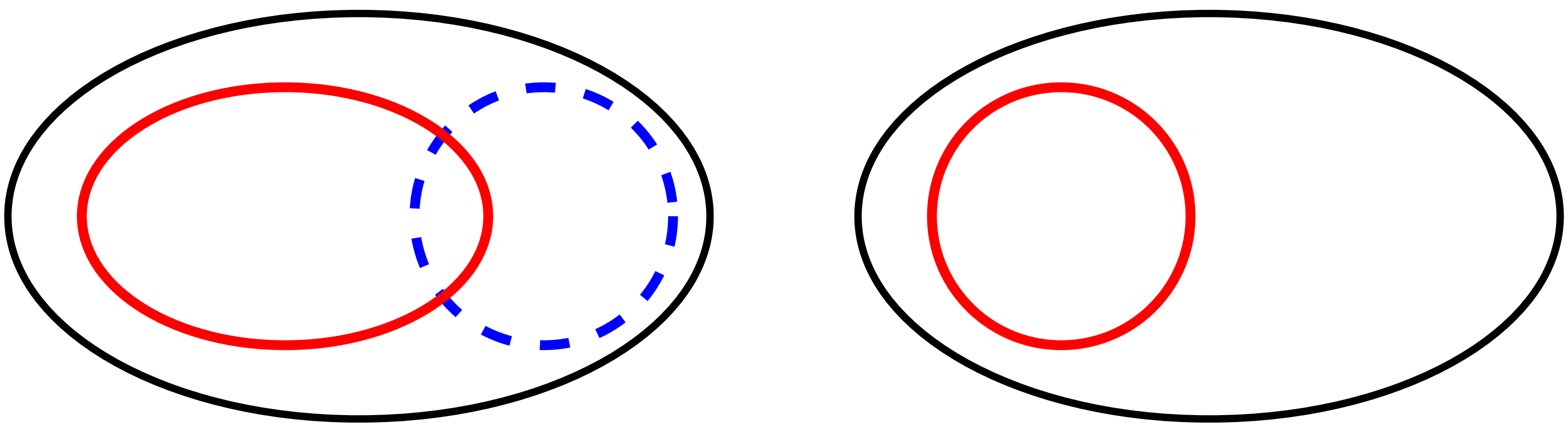
The Fischer-Matsliah Algorithm

- Select $\tilde{O}(n^{3/4})$ random vertices $V_G^\star$ from G and query their connections to $\tilde{O}(n^{1/2})$ random vertices



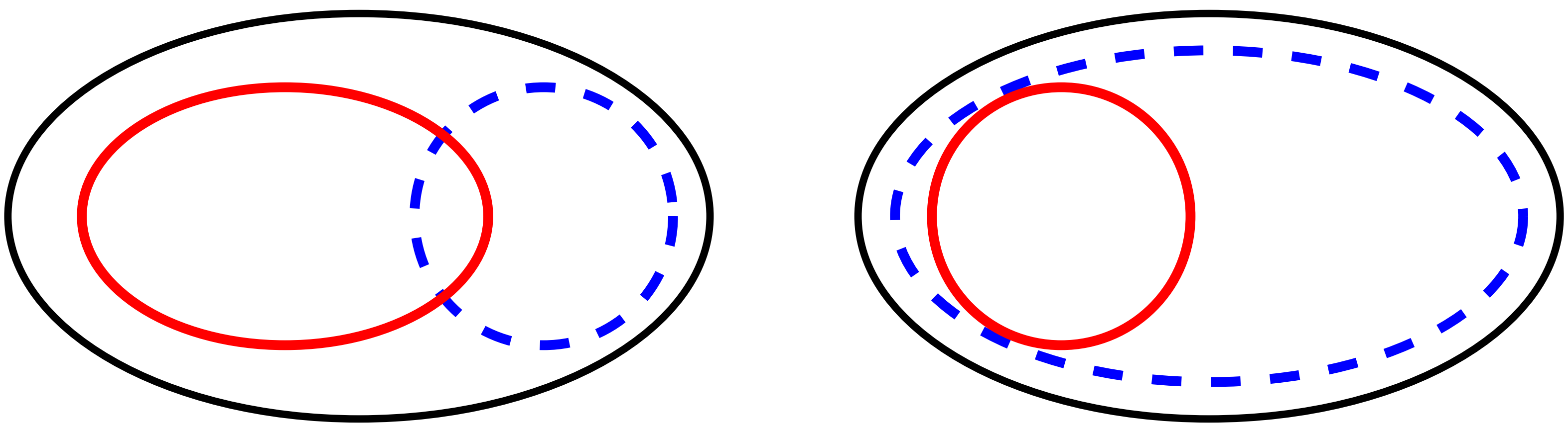
The Fischer-Matsliah Algorithm

- Select $\tilde{O}(n^{3/4})$ random vertices $V_G^\star$ from G and query their connections to $\tilde{O}(n^{1/2})$ random vertices
- Select $\tilde{O}(n^{1/4})$ random vertices $V_H^\star$ from H



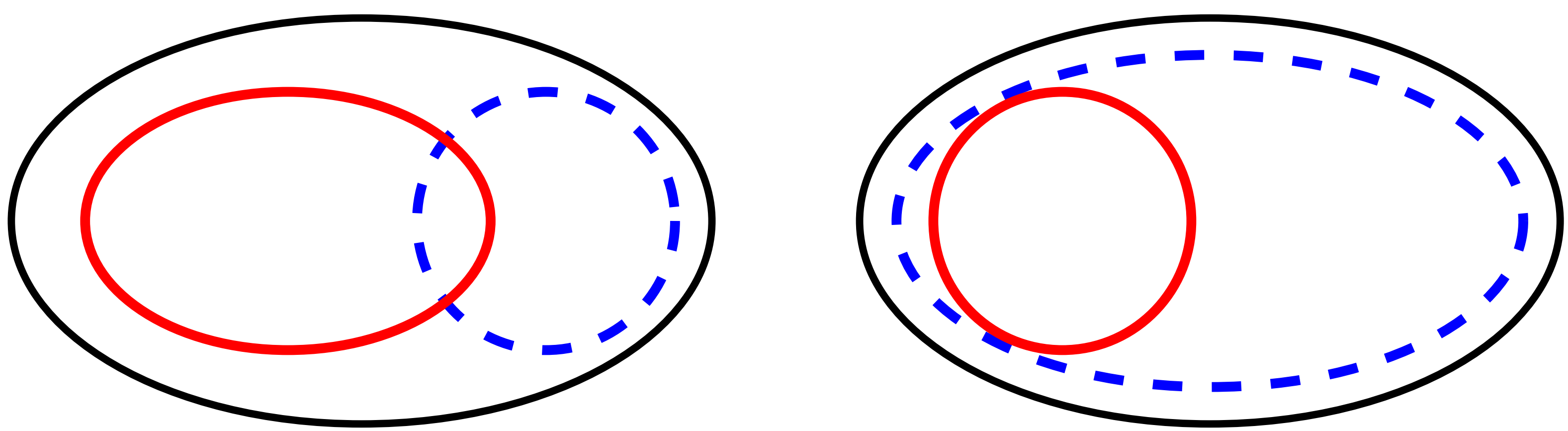
The Fischer-Matsliah Algorithm

- Select $\tilde{O}(n^{3/4})$ random vertices $V_G^\star$ from G and query their connections to $\tilde{O}(n^{1/2})$ random vertices
- Select $\tilde{O}(n^{1/4})$ random vertices $V_H^\star$ from H and query their connections to all other vertices



The Fischer-Matsliah Algorithm

- Select $\tilde{O}(n^{3/4})$ random vertices $V_G^\star$ from G and query their connections to $\tilde{O}(n^{1/2})$ random vertices
- Select $\tilde{O}(n^{1/4})$ random vertices $V_H^\star$ from H and query their connections to all other vertices
- Enumerate over all possible core sets from $V_G^\star$ and $V_H^\star$ ($\mathrm{quasipoly}(n)$ options)



The Fischer-Matsliah Algorithm

- Select $\tilde{O}(n^{3/4})$ random vertices $V_G^\star$ from G and query their connections to $\tilde{O}(n^{1/2})$ random vertices
- Select $\tilde{O}(n^{1/4})$ random vertices $V_H^\star$ from H and query their connections to all other vertices
- Enumerate over all possible core sets from $V_G^\star$ and $V_H^\star$ ($\mathrm{quasipoly}(n)$ options)
- For each pair of candidates:
  - Step 1: See if distributions of labels similar (Use the $\tilde{O}(\sqrt{n})$-sample tester of Batu et al. (2001))

The Fischer-Matsliah Algorithm

- Select $\tilde{O}(n^{3/4})$ random vertices $V_G^\star$ from G and query their connections to $\tilde{O}(n^{1/2})$ random vertices
- Select $\tilde{O}(n^{1/4})$ random vertices $V_H^\star$ from H and query their connections to all other vertices
- Enumerate over all possible core sets from $V_G^\star$ and $V_H^\star$ ($\mathrm{quasipoly}(n)$ options)
- For each pair of candidates:
  - Step 1: See if distributions of labels similar (Use the $\tilde{O}(\sqrt{n})$-sample tester of Batu et al. (2001))
  - Step 2: Select $\mathrm{polylog}\, n$ pairs of random vertices in G and see if the connectivity of their labels almost the same in H

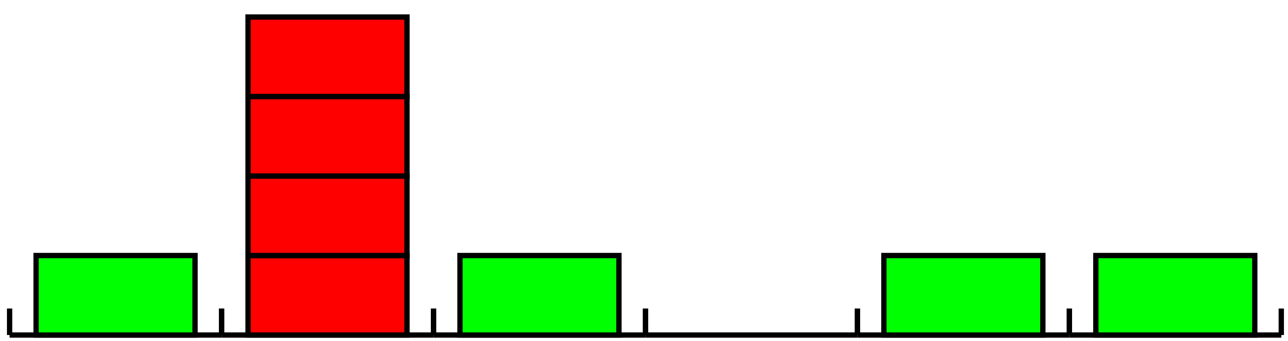
First Attempts

Better Label Distribution Testing?

- Idea: Use a different distribution tester
  - Select $\tilde{O}(\sqrt{n})$ vertices from G and H
  - Query connections to random subset of size $\tilde{O}(n^{2/3})$
  - Use the distribution tester of Batu et al. (2000)
  - Query Complexity: $\tilde{O}(n^{7/6})$

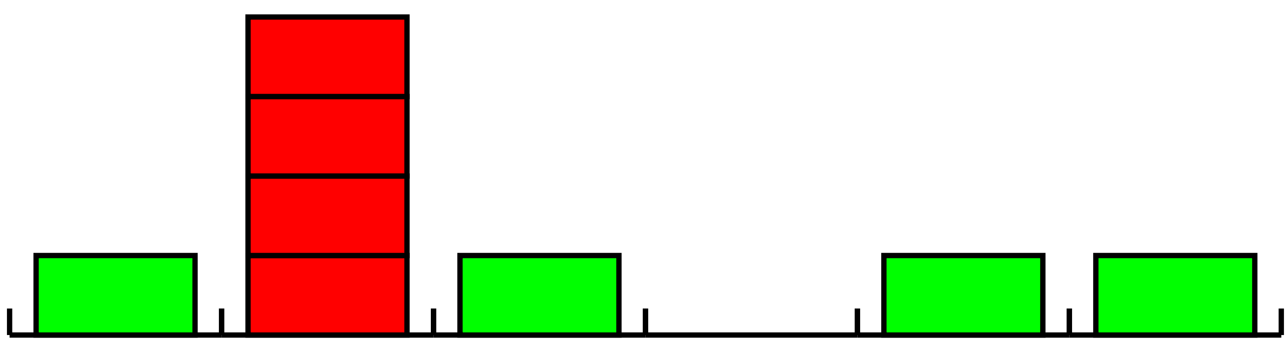
Better Label Distribution Testing?

- Idea: Use a different distribution tester
  - Select $\tilde{O}(\sqrt{n})$ vertices from G and H
  - Query connections to random subset of size $\tilde{O}(n^{2/3})$
  - Use the distribution tester of Batu et al. (2000)
  - Query Complexity: $\tilde{O}(n^{7/6})$
- Problem: Unclear how to do Step 2
  - Only a subset of labels known
  - Sampling subsets and comparing discovered identical labels introduces a bias



Better Label Distribution Testing?

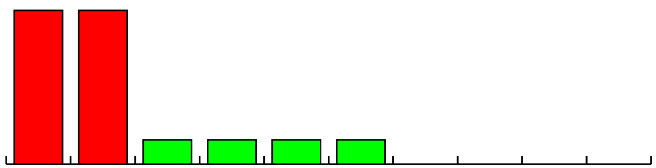
- Idea: Use a different distribution tester
  - Select $\tilde{O}(\sqrt{n})$ vertices from G and H
  - Query connections to random subset of size $\tilde{O}(n^{2/3})$
  - Use the distribution tester of Batu et al. (2000)
  - Query Complexity: $\tilde{O}(n^{7/6})$
- Problem: Unclear how to do Step 2
  - Only a subset of labels known
  - Sampling subsets and comparing discovered identical labels introduces a bias



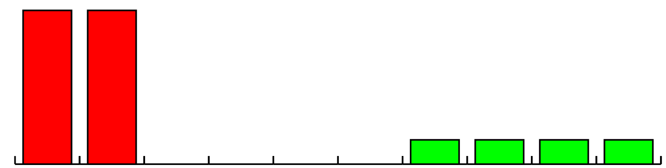
- Ignore this issue for now

Exponent Less than $7/6$?

- Could the $\Omega(n^{2/3})$ distribution testing lower bound [Valiant 2008] imply a $\Omega(n^{7/6})$ lower bound?



vs.



Exponent Less than $7/6$?

- Could the $\Omega(n^{2/3})$ distribution testing lower bound [Valiant 2008] imply a $\Omega(n^{7/6})$ lower bound?

- We have additional information

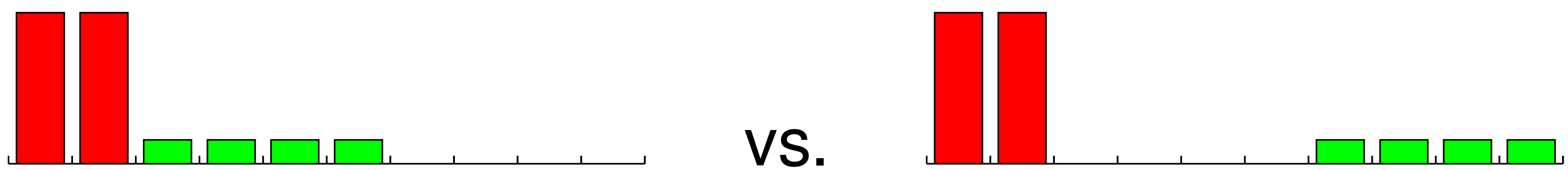
Exponent Less than $7/6$?

- Could the $\Omega(n^{2/3})$ distribution testing lower bound [Valiant 2008] imply a $\Omega(n^{7/6})$ lower bound?

- We have additional information:
  - Estimate in advance distances between adjacency vectors in both graphs
  - We care about random core sets, where labels preserve distances between adjacency vectors
  - We should be able to distinguish heavy and light elements

Exponent Less than $7/6$?

- Could the $\Omega(n^{2/3})$ distribution testing lower bound [Valiant 2008] imply a $\Omega(n^{7/6})$ lower bound?



- We have additional information:
  - Estimate in advance distances between adjacency vectors in both graphs
  - We care about random core sets, where labels preserve distances between adjacency vectors
  - We should be able to distinguish heavy and light elements
- Nice consistent clustering of slightly different adjacency vectors still difficult

Our Algorithm

Overview

$(O^\star(f(n)) \equiv O(f(n)) \cdot 2^{O(\sqrt{\log n})})$

Overview

$(O^\star(f(n)) \equiv O(f(n)) \cdot 2^{O(\sqrt{\log n})})$

- Compute good approximations of distances between adjacency vectors with $O^\star(n)$ queries

Overview

$(O^\star(f(n)) \equiv O(f(n)) \cdot 2^{O(\sqrt{\log n})})$

- Compute good approximations of distances between adjacency vectors with $O^\star(n)$ queries
- Sample $O^\star(\sqrt{n})$ vertices from both graphs

Overview

$(O^\star(f(n)) \equiv O(f(n)) \cdot 2^{O(\sqrt{\log n})})$

- Compute good approximations of distances between adjacency vectors with $O^\star(n)$ queries
- Sample $O^\star(\sqrt{n})$ vertices from both graphs
- Consider all pairs of corsets of size $O^\star(1)$

Overview

$(O^\star(f(n)) \equiv O(f(n)) \cdot 2^{O(\sqrt{\log n})})$

- Compute good approximations of distances between adjacency vectors with $O^\star(n)$ queries
- Sample $O^\star(\sqrt{n})$ vertices from both graphs
- Consider all pairs of corsets of size $O^\star(1)$
  - Step 0: sample $O^\star(\sqrt{n})$ vertices to see if the core sets preserve distances well

Overview

($O^\star(f(n)) \equiv O(f(n)) \cdot 2^{O(\sqrt{\log n})}$)

- Compute good approximations of distances between adjacency vectors with $O^\star(n)$ queries
- Sample $O^\star(\sqrt{n})$ vertices from both graphs
- Consider all pairs of corsets of size $O^\star(1)$
  - Step 0: sample $O^\star(\sqrt{n})$ vertices to see if the core sets preserve distances well
  - Step 1: relaxed test required to reject only sets of labels that are far in Earth-Mover Distance

Overview

$(O^\star(f(n)) \equiv O(f(n)) \cdot 2^{O(\sqrt{\log n})})$

- Compute good approximations of distances between adjacency vectors with $O^\star(n)$ queries
- Sample $O^\star(\sqrt{n})$ vertices from both graphs
- Consider all pairs of corsets of size $O^\star(1)$
  - Step 0: sample $O^\star(\sqrt{n})$ vertices to see if the core sets preserve distances well
  - Step 1: relaxed test required to reject only sets of labels that are far in Earth-Mover Distance
  - Step 2: approximate matching of labels, rejection sampling to avoid biases

Overview

$(O^\star(f(n)) \equiv O(f(n)) \cdot 2^{O(\sqrt{\log n})})$

- Compute good approximations of distances between adjacency vectors with $O^\star(n)$ queries
- Sample $O^\star(\sqrt{n})$ vertices from both graphs
- Consider all pairs of corsets of size $O^\star(1)$
  - Step 0: sample $O^\star(\sqrt{n})$ vertices to see if the core sets preserve distances well
  - Step 1: relaxed test required to reject only sets of labels that are far in Earth-Mover Distance
  - Step 2: approximate matching of labels, rejection sampling to avoid biases

Will show simplified versions

Overview

$(O^\star(f(n)) \equiv O(f(n)) \cdot 2^{O(\sqrt{\log n})})$

- Compute good approximations of distances between adjacency vectors with $O^\star(n)$ queries
- Sample $O^\star(\sqrt{n})$ vertices from both graphs
- Consider all pairs of corsets of size $O^\star(1)$
  - Step 0: sample $O^\star(\sqrt{n})$ vertices to see if the core sets preserve distances well
  - Step 1: relaxed test required to reject only sets of labels that are far in Earth-Mover Distance
  - Step 2: approximate matching of labels, rejection sampling to avoid biases

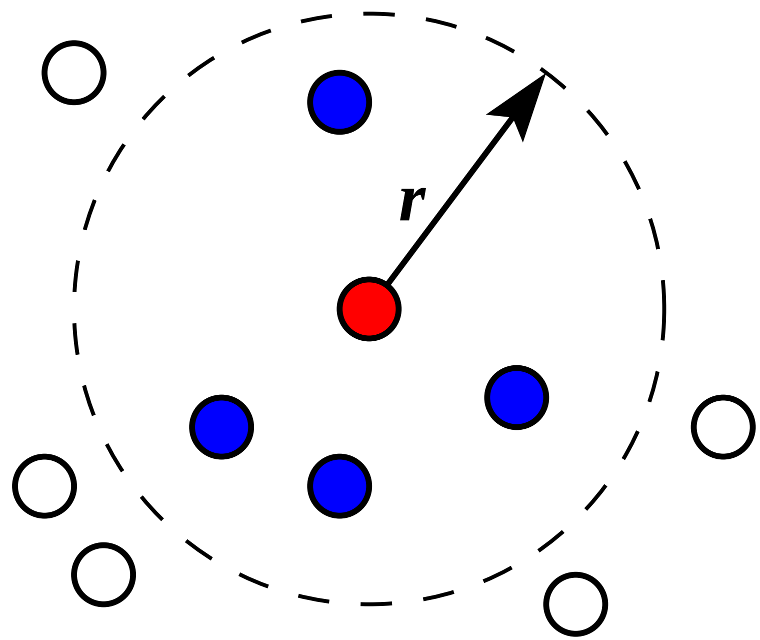
Will show simplified versions
Assume labels preserve all distances

Collisions

- Standard distribution testing collisions: identical labels

Collisions

- Standard distribution testing collisions: identical labels
- Here:
  - pick small threshold r
  - labels at distance r collide



Collisions

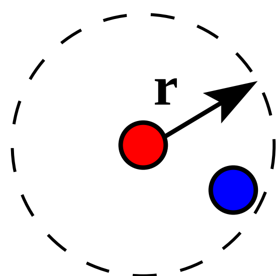
- Standard distribution testing collisions: identical labels
- Here:
  - pick small threshold r
  - labels at distance r collide
- Can apply the standard distribution testing?
  - Not in a trivial way
  - Colliding vertices can have very different degrees
  - Hard to partition them into classes of similar degrees

Collisions

- Standard distribution testing collisions: identical labels
- Here:
  - pick small threshold r
  - labels at distance r collide
- Can apply the standard distribution testing?
  - Not in a trivial way
  - Colliding vertices can have very different degrees
  - Hard to partition them into classes of similar degrees
- Our solution:
  - Randomize the threshold r
  - Design a tool for estimating the number of collisions
  - Reject only if Earth-Mover Distance large

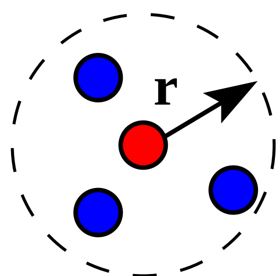
Estimating Number of Collisions

- Found colliding labels for vertices $v \in V[G]$ and $w \in V[H]$



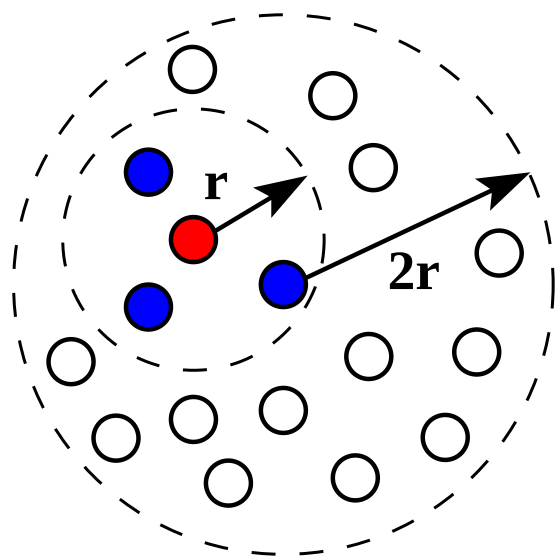
Estimating Number of Collisions

- Found colliding labels for vertices $v \in V[G]$ and $w \in V[H]$
- Question: How many vertices in H have labels colliding with v?



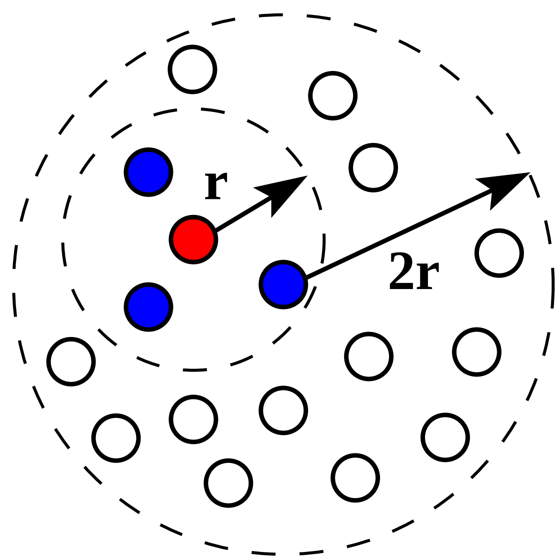
Estimating Number of Collisions

- Found colliding labels for vertices $v \in V[G]$ and $w \in V[H]$
- Question: How many vertices in H have labels colliding with v?
- Solution: Sample similar vertices in H up to distance $2r + (\text{error-term})$ and see their labels



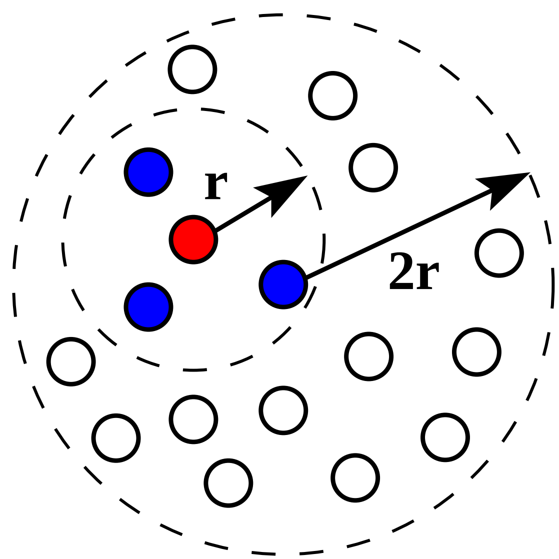
Estimating Number of Collisions

- Found colliding labels for vertices $v \in V[G]$ and $w \in V[H]$
- Question: How many vertices in H have labels colliding with v?
- Solution: Sample similar vertices in H up to distance $2r + (\text{error-term})$ and see their labels
- Requirement: good multiplicative approximation



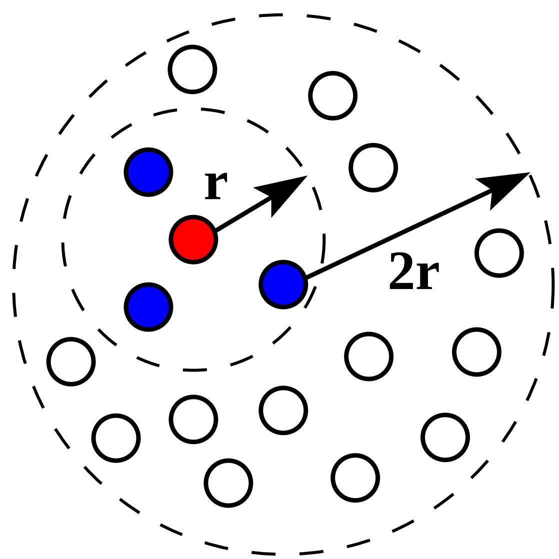
Estimating Number of Collisions

- Found colliding labels for vertices $v \in V[G]$ and $w \in V[H]$
- Question: How many vertices in H have labels colliding with v?
- Solution: Sample similar vertices in H up to distance $2r + (\text{error-term})$ and see their labels
- Requirement: good multiplicative approximation
- Problem: What if few of them do?



Estimating Number of Collisions

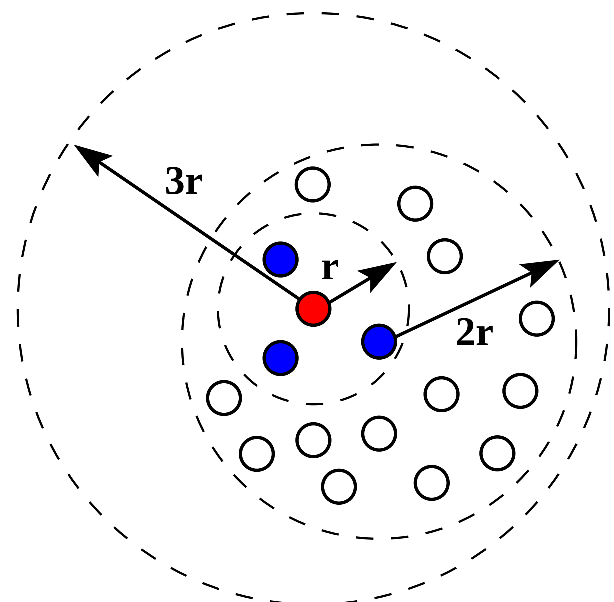
- Found colliding labels for vertices $v \in V[G]$ and $w \in V[H]$
- Question: How many vertices in H have labels colliding with v?
- Solution: Sample similar vertices in H up to distance $2r +$ (error-term) and see their labels
- Requirement: good multiplicative approximation
- Problem: What if few of them do?
- Pick a threshold $r = 2^{-t}/\operatorname{polylog} n$, where $t \in [0, O(\sqrt{\log n})]$



Estimating Number of Collisions

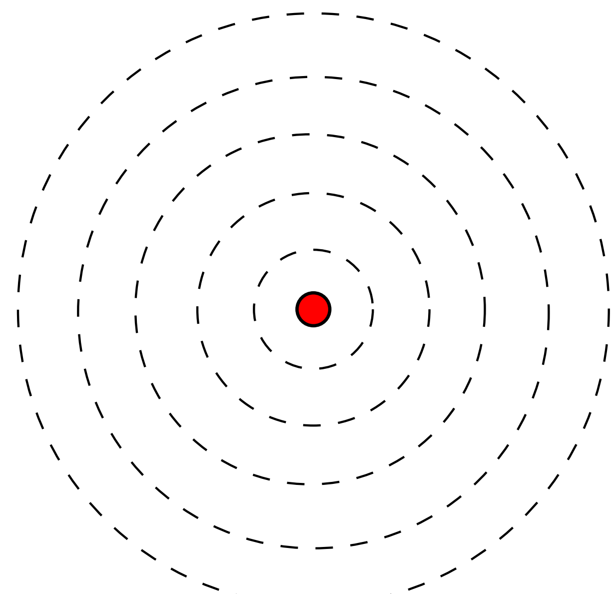
- Found colliding labels for vertices $v \in V[G]$ and $w \in V[H]$
- Question: How many vertices in H have labels colliding with v?
- Solution: Sample similar vertices in H up to distance $2r +$ (error-term) and see their labels
- Requirement: good multiplicative approximation
- Problem: What if few of them do?
- Pick a threshold $r = 2^{-t} / \operatorname{polylog} n$, where $t \in [0, O(\sqrt{\log n})]$

- For most thresholds most points will require $O^{\star}(1)$ samples



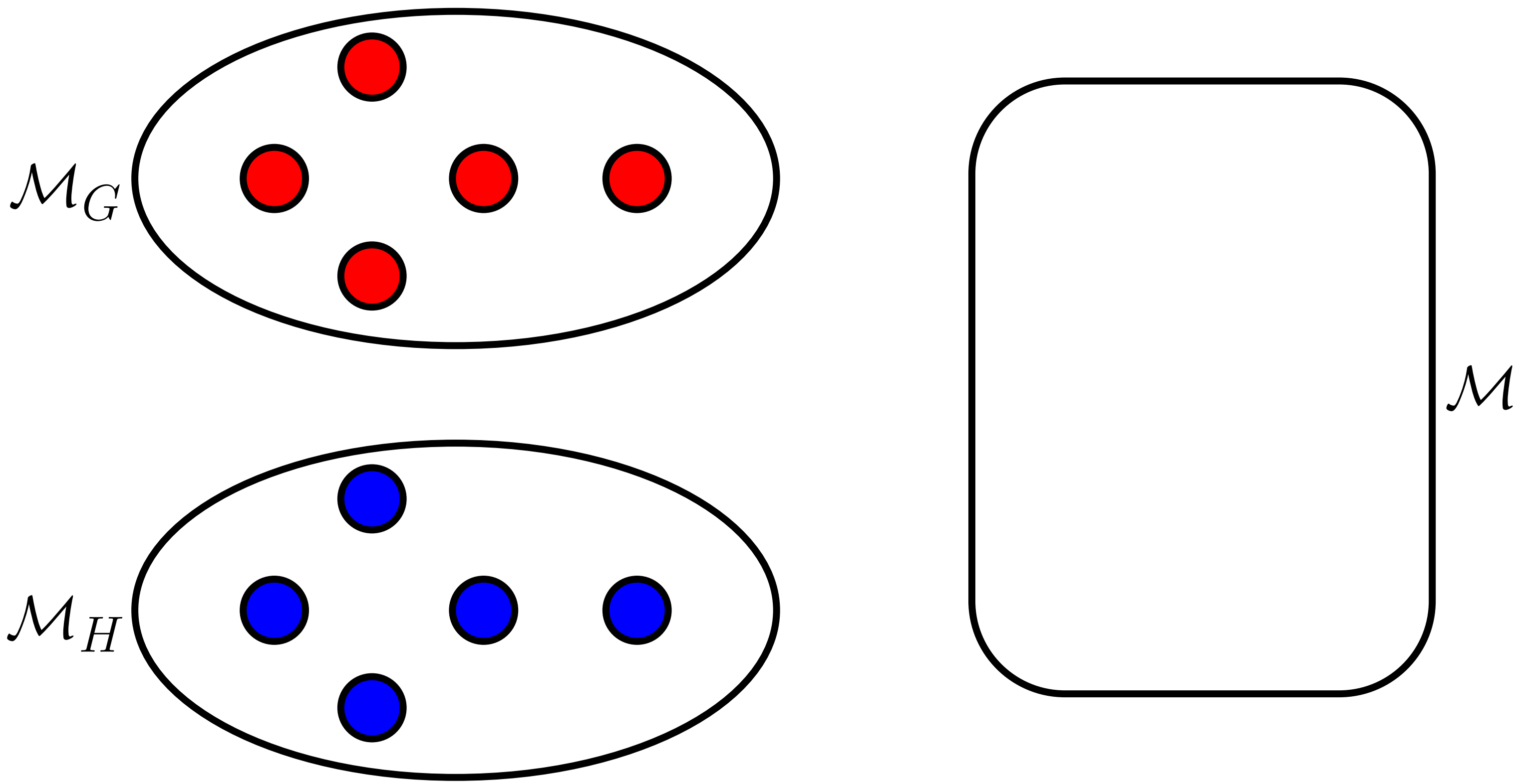
Estimating Number of Collisions

- Found colliding labels for vertices $v \in V[G]$ and $w \in V[H]$
- Question: How many vertices in H have labels colliding with v?
- Solution: Sample similar vertices in H up to distance $2r + (\text{error-term})$ and see their labels
- Requirement: good multiplicative approximation
- Problem: What if few of them do?
- Pick a threshold $r = 2^{-t}/\operatorname{polylog} n$, where $t \in [0, O(\sqrt{\log n})]$
- For most thresholds most points will require $O^{\star}(1)$ samples



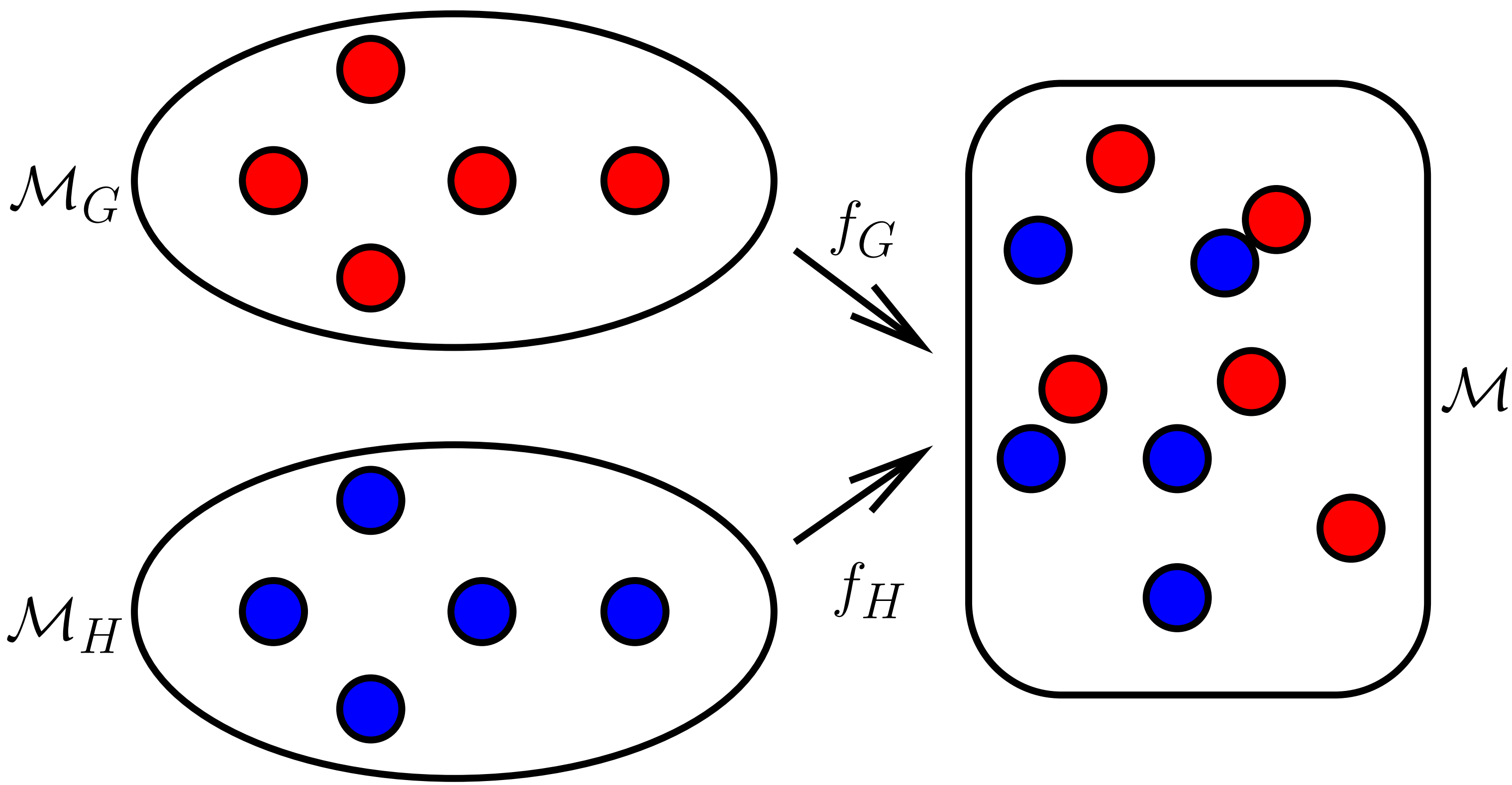
Step 1: Earth-Mover Distance Testing

- Three metric spaces: $\mathcal{M}_G$, $\mathcal{M}_H$, $\mathcal{M}$
  (diameter bounded by 1, $|\mathcal{M}_G| = |\mathcal{M}_H| = n$)



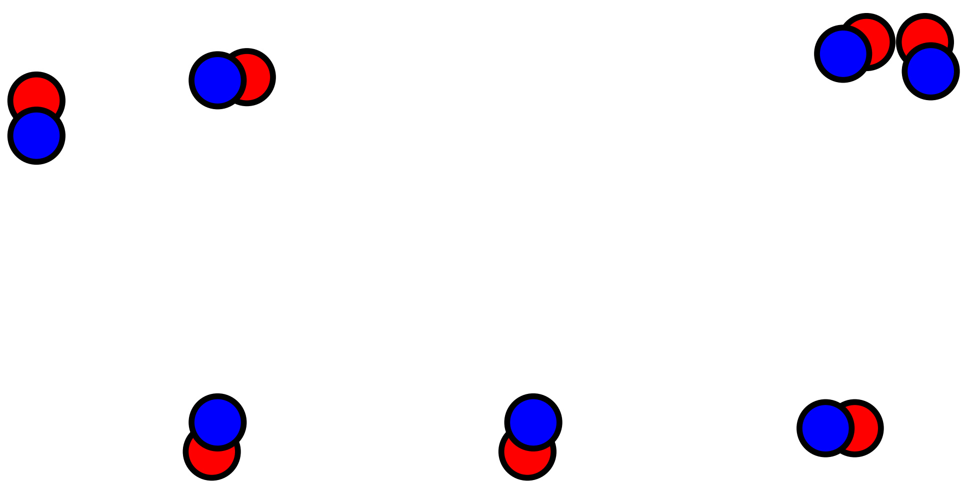
Step 1: Earth-Mover Distance Testing

- Three metric spaces: $\mathcal{M}_G$, $\mathcal{M}_H$, $\mathcal{M}$
  (diameter bounded by 1, $|\mathcal{M}_G| = |\mathcal{M}_H| = n$)
- $f_G : \mathcal{M}_G \to \mathcal{M}$ and $f_H : \mathcal{M}_H \to \mathcal{M}$ preserve distances up to a small additive term



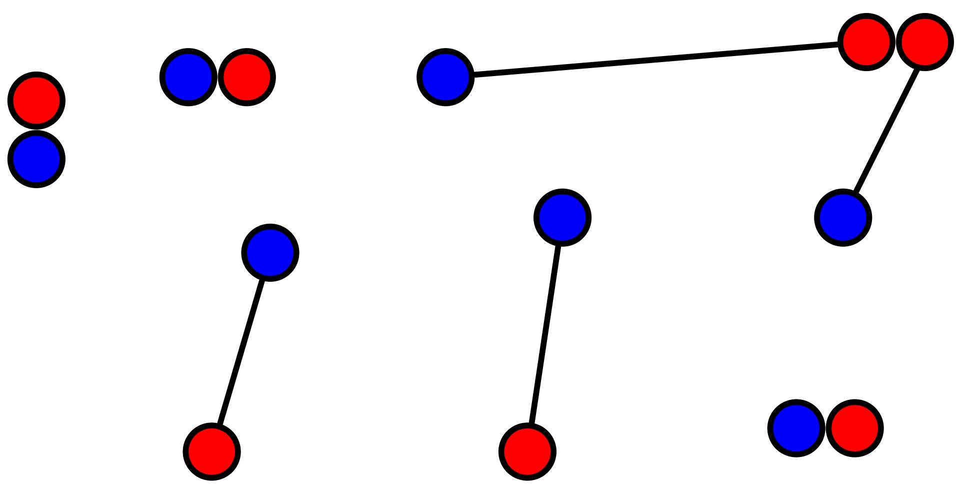
Step 1: Earth-Mover Distance Testing

- Three metric spaces: $\mathcal{M}_G$, $\mathcal{M}_H$, $\mathcal{M}$
  (diameter bounded by 1, $|\mathcal{M}_G| = |\mathcal{M}_H| = n$)
- $f_G : \mathcal{M}_G \to \mathcal{M}$ and $f_H : \mathcal{M}_H \to \mathcal{M}$ preserve distances up to a small additive term
- Tester required to
  - Accept if $f_G(\mathcal{M}_G) = f_H(\mathcal{M}_H)$



Step 1: Earth-Mover Distance Testing

- Three metric spaces: $\mathcal{M}_G$, $\mathcal{M}_H$, $\mathcal{M}$
  (diameter bounded by 1, $|\mathcal{M}_G| = |\mathcal{M}_H| = n$)
- $f_G : \mathcal{M}_G \to \mathcal{M}$ and $f_H : \mathcal{M}_H \to \mathcal{M}$ preserve distances up to a small additive term
- Tester required to
  - Accept if $f_G(\mathcal{M}_G) = f_H(\mathcal{M}_H)$
  - Reject if $\mathrm{EMD}(f_G(\mathcal{M}_G), f_H(\mathcal{M}_H)) \geq \epsilon n$



Step 1: Earth-Mover Distance Testing

- Three metric spaces: $\mathcal{M}_G$, $\mathcal{M}_H$, $\mathcal{M}$
  (diameter bounded by 1, $|\mathcal{M}_G| = |\mathcal{M}_H| = n$)
- $f_G : \mathcal{M}_G \to \mathcal{M}$ and $f_H : \mathcal{M}_H \to \mathcal{M}$ preserve distances up to a small additive term
- Tester required to
  - Accept if $f_G(\mathcal{M}_G) = f_H(\mathcal{M}_H)$
  - Reject if $\mathrm{EMD}(f_G(\mathcal{M}_G), f_H(\mathcal{M}_H)) \geq \epsilon n$
- Our solution:
  - Step 1a: make sure almost all points in $f_G(\mathcal{M}_G)$
    collide with $f_H(\mathcal{M}_H)$, and vice versa

Step 1: Earth-Mover Distance Testing

- Three metric spaces: $\mathcal{M}_G$, $\mathcal{M}_H$, $\mathcal{M}$ (diameter bounded by 1, $|\mathcal{M}_G| = |\mathcal{M}_H| = n$)
- $f_G : \mathcal{M}_G \to \mathcal{M}$ and $f_H : \mathcal{M}_H \to \mathcal{M}$ preserve distances up to a small additive term
- Tester required to
  - Accept if $f_G(\mathcal{M}_G) = f_H(\mathcal{M}_H)$
  - Reject if $\mathrm{EMD}(f_G(\mathcal{M}_G), f_H(\mathcal{M}_H)) \geq \epsilon n$
- Our solution:
  - Step 1a: make sure almost all points in $f_G(\mathcal{M}_G)$ collide with $f_H(\mathcal{M}_H)$, and vice versa
  - Step 1b: if two points collide make sure their metrics can be matched

More Details

Step 1a: make sure almost all points in $f_G(\mathcal{M}_G)$ collide with $f_H(\mathcal{M}_H)$

More Details

Step 1a: make sure almost all points in $f_G(\mathcal{M}_G)$ collide with $f_H(\mathcal{M}_H)$

- Take $O^\star(\sqrt{n})$ samples

More Details

Step 1a: make sure almost all points in $f_G(\mathcal{M}_G)$ collide with $f_H(\mathcal{M}_H)$

- Take $O^\star(\sqrt{n})$ samples
- Estimate weighted number of collisions

More Details

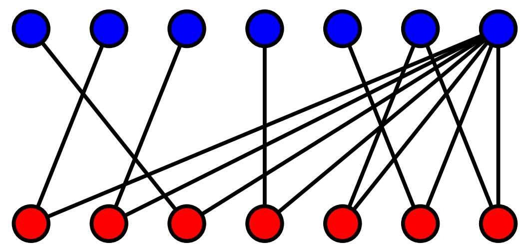
Step 1a: make sure almost all points in $f_G(\mathcal{M}_G)$ collide with $f_H(\mathcal{M}_H)$

- Take $O^\star(\sqrt{n})$ samples
- Estimate weighted number of collisions
- Divide each collision by the estimated number of collisions

More Details

Step 1a: make sure almost all points in $f_G(\mathcal{M}_G)$ collide with $f_H(\mathcal{M}_H)$

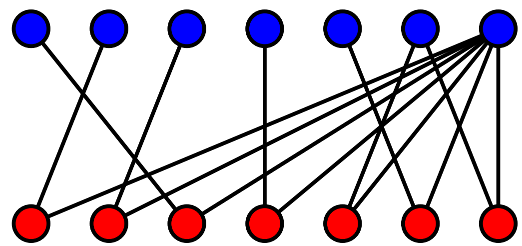
- Take $O^\star(\sqrt{n})$ samples
- Estimate weighted number of collisions
- Divide each collision by the estimated number of collisions
- Bad case:



More Details

Step 1a: make sure almost all points in $f_G(\mathcal{M}_G)$ collide with $f_H(\mathcal{M}_H)$

- Take $O^\star(\sqrt{n})$ samples
- Estimate weighted number of collisions
- Divide each collision by the estimated number of collisions
- Bad case:

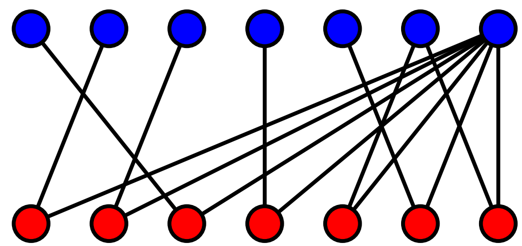


- Randomized threshold makes it unlikely to happen for a large fraction of vertices

More Details

Step 1a: make sure almost all points in $f_G(\mathcal{M}_G)$ collide with $f_H(\mathcal{M}_H)$

- Take $O^\star(\sqrt{n})$ samples
- Estimate weighted number of collisions
- Divide each collision by the estimated number of collisions
- Bad case:



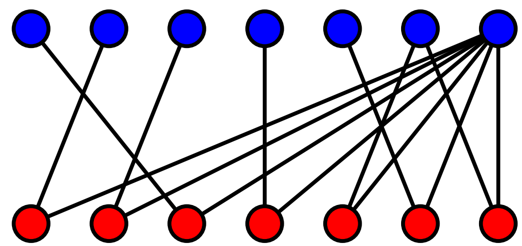
- Randomized threshold makes it unlikely to happen for a large fraction of vertices

Step 1b: if two points p_G and p_H collide make sure their metrics can be matched

More Details

Step 1a: make sure almost all points in $f_G(\mathcal{M}_G)$ collide with $f_H(\mathcal{M}_H)$

- Take $O^\star(\sqrt{n})$ samples
- Estimate weighted number of collisions
- Divide each collision by the estimated number of collisions
- Bad case:



- Randomized threshold makes it unlikely to happen for a large fraction of vertices

Step 1b: if two points p_G and p_H collide make sure their metrics can be matched

- Is there a mapping $\mathcal{F} : \mathcal{M}_G \to \mathcal{M}_H$, where $\mathcal{F}(p_G) = p_H$ and all distances preserved up to a small additive term?

Why This Works

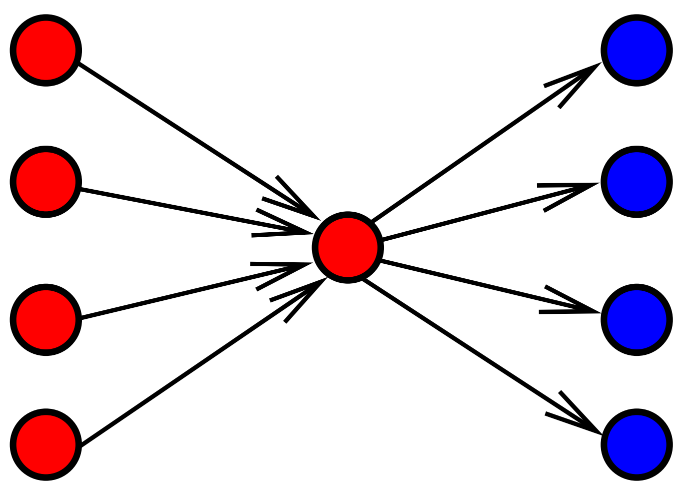
- Need to show that $\mathrm{EMD}(f_G(\mathcal{M}_G), f_H(\mathcal{M}_H)) \leq \epsilon n$ if passes the test

Why This Works

- Need to show that $\mathrm{EMD}(f_G(\mathcal{M}_G), f_H(\mathcal{M}_H)) \leq \epsilon n$ if passes the test
- We show a large fractional matching using short edges

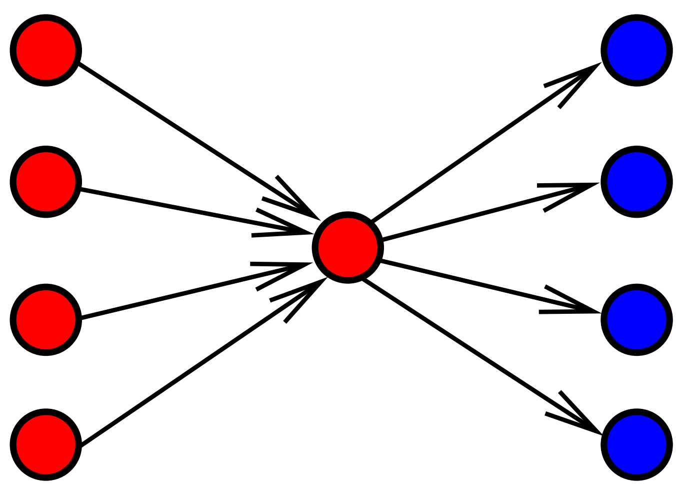
Why This Works

- Need to show that $\mathrm{EMD}(f_G(\mathcal{M}_G), f_H(\mathcal{M}_H)) \leq \epsilon n$ if passes the test
- We show a large fractional matching using short edges
- Two step process:
  - Each point in $f_G(\mathcal{M}_G)$ sends flow to hubs in $f_G(\mathcal{M}_G)$
  - Each point in $f_H(\mathcal{M}_H)$ receives flow from hubs in $f_G(\mathcal{M}_G)$



Why This Works

- Need to show that $\mathrm{EMD}(f_G(\mathcal{M}_G), f_H(\mathcal{M}_H)) \leq \epsilon n$ if passes the test
- We show a large fractional matching using short edges
- Two step process:
  - Each point in $f_G(\mathcal{M}_G)$ sends flow to hubs in $f_G(\mathcal{M}_G)$
  - Each point in $f_H(\mathcal{M}_H)$ receives flow from hubs in $f_G(\mathcal{M}_G)$
- Each point sends/receives one unit, but hubs handle arbitrary amount of flow



Why This Works

- Need to show that $\mathrm{EMD}(f_G(\mathcal{M}_G), f_H(\mathcal{M}_H)) \leq \epsilon n$ if passes the test
- We show a large fractional matching using short edges
- Two step process:
  - Each point in $f_G(\mathcal{M}_G)$ sends flow to hubs in $f_G(\mathcal{M}_G)$
  - Each point in $f_H(\mathcal{M}_H)$ receives flow from hubs in $f_G(\mathcal{M}_G)$
- Each point sends/receives one unit, but hubs handle arbitrary amount of flow
- Amount sent to/received from point x proportional to

$$\exp(-\delta(x, y) \cdot \operatorname{polylog} n)$$

Why This Works

- Need to show that $\mathrm{EMD}(f_G(\mathcal{M}_G), f_H(\mathcal{M}_H)) \leq \epsilon n$ if passes the test
- We show a large fractional matching using short edges
- Two step process:
  - Each point in $f_G(\mathcal{M}_G)$ sends flow to hubs in $f_G(\mathcal{M}_G)$
  - Each point in $f_H(\mathcal{M}_H)$ receives flow from hubs in $f_G(\mathcal{M}_G)$
- Each point sends/receives one unit, but hubs handle arbitrary amount of flow
- Amount sent to/received from point x proportional to
$$\exp(-\delta(x, y) \cdot \operatorname{polylog} n)$$
- For almost matching metrics: each hub receives and sends almost the same amount

Why This Works

- Need to show that $\mathrm{EMD}(f_G(\mathcal{M}_G), f_H(\mathcal{M}_H)) \le \epsilon n$ if passes the test
- We show a large fractional matching using short edges
- Two step process:
  - Each point in $f_G(\mathcal{M}_G)$ sends flow to hubs in $f_G(\mathcal{M}_G)$
  - Each point in $f_H(\mathcal{M}_H)$ receives flow from hubs in $f_G(\mathcal{M}_G)$
- Each point sends/receives one unit, but hubs handle arbitrary amount of flow
- Amount sent to/received from point x proportional to

$$\exp(-\delta(x, y) \cdot \operatorname{polylog} n)$$

- For almost matching metrics: each hub receives and sends almost the same amount
- There is a close point: edges longer than some small threshold carry almost no flow

Step 2

- Question: Do labels describe connectivity?

Step 2

- Question: Do labels describe connectivity?
- Problem: Sampling pairs of similar labels more likely
  (Reason why Fischer & Matsliah didn't get $\tilde{O}(n^{7/6})$?)

Step 2

- Question: Do labels describe connectivity?
- Problem: Sampling pairs of similar labels more likely (Reason why Fischer & Matsliah didn't get $\tilde{O}(n^{7/6})$?)
- Use rejection sampling to uniformly sample a label for $V[G]$ with a colliding label for $V[H]$

Step 2

- Question: Do labels describe connectivity?
- Problem: Sampling pairs of similar labels more likely (Reason why Fischer & Matsliah didn't get $\tilde{O}(n^{7/6})$?)
- Use rejection sampling to uniformly sample a label for $V[G]$ with a colliding label for $V[H]$
- Keep a sample with probability

  $1/($estimated-number-of-collisions$)$

Step 2

- Question: Do labels describe connectivity?
- Problem: Sampling pairs of similar labels more likely (Reason why Fischer & Matsliah didn't get $\tilde{O}(n^{7/6})$?)
- Use rejection sampling to uniformly sample a label for $V[G]$ with a colliding label for $V[H]$
- Keep a sample with probability
  $$1/(\text{estimated-number-of-collisions})$$
- For independent endpoints, slight difference in labels shouldn't introduce too much error

Step 2

- Question: Do labels describe connectivity?
- Problem: Sampling pairs of similar labels more likely (Reason why Fischer & Matsliah didn't get $\tilde{O}(n^{7/6})$?)
- Use rejection sampling to uniformly sample a label for $V[G]$ with a colliding label for $V[H]$
- Keep a sample with probability
  $1/($estimated-number-of-collisions$)$
- For independent endpoints, slight difference in labels shouldn't introduce too much error
- Accept if similar labels imply same connectivity most of the time

Final Remarks

Open Questions

- Is the $2^{O(\sqrt{\log n})}$ factor necessary?

Open Questions

- Is the $2^{O(\sqrt{\log n})}$ factor necessary?
- Can approximate the distance efficiently?
  Improve the running time?

Open Questions

- Is the $2^{O(\sqrt{\log n})}$ factor necessary?
- Can approximate the distance efficiently?
  Improve the running time?
- Sparse graphs?
  - hyperfinite bounded-degree [Newman Sohler 2011]
    $O(1)$ queries
  - arbitrary degree forests [Kusumoto Yoshida 2014]
    $\mathrm{polylog}(n)$ queries

Open Questions

- Is the $2^{O(\sqrt{\log n})}$ factor necessary?
- Can approximate the distance efficiently?
  Improve the running time?
- Sparse graphs?
  - hyperfinite bounded-degree [Newman Sohler 2011]
    $O(1)$ queries
  - arbitrary degree forests [Kusumoto Yoshida 2014]
    $\mathrm{polylog}(n)$ queries
- Unified framework for various kinds of isomorphisms?
  (graphs, functions, etc.)

Questions?