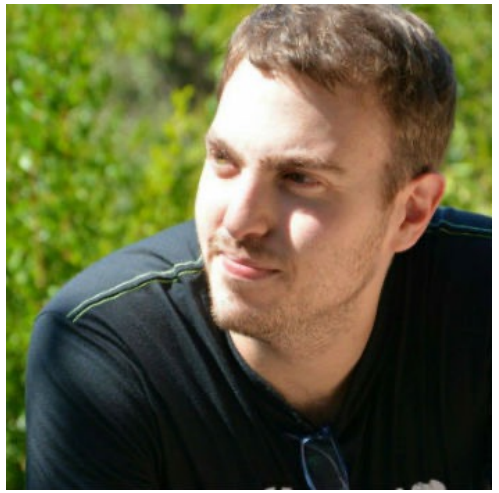
New Streaming Algorithms for High-Dimensional EMD and MST

Xi Chen
Columbia

Rajesh Jayaram
Google

Amit Levi
Waterloo



Erik Waingarten
Stanford

Earth Mover's Distance

Metric space: (X, d_x)

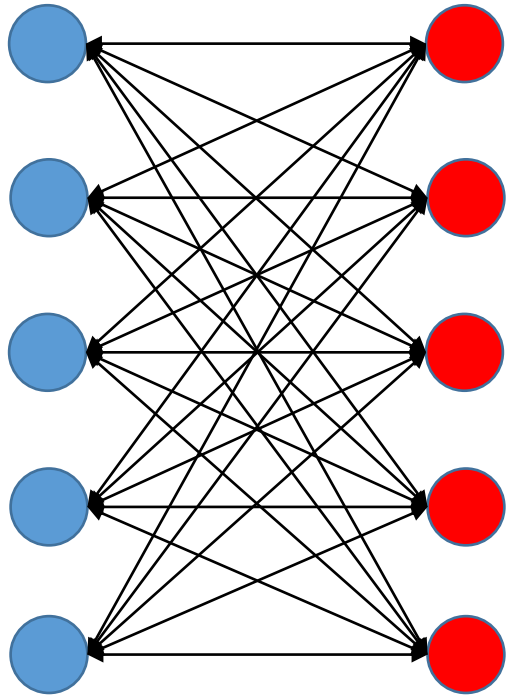
Multisets: $A = \{a_1, \dots, a_n\}, B = \{b_1, \dots, b_n\} \subset X$

Earth Mover's Distance

Metric space: (X, d_x)

Multisets: $A = \{a_1, \dots, a_n\}, B = \{b_1, \dots, b_n\} \subset X$

$$\mathrm{EMD}_X(A, B) \stackrel{\text{def}}{=} \min_{\substack{M:[n]\to[n] \\ \text{bijection}}} \sum_{i=1}^{n} d_X(a_i, b_{M(i)})$$



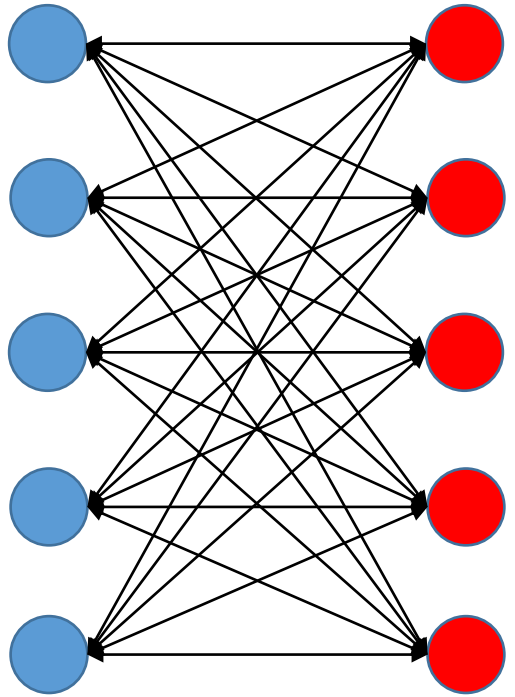
(Min cost bipartite matching with triangle inequality)

Earth Mover's Distance

Metric space: (X, d_x)

Multisets: $A = \{a_1, \dots, a_n\}, B = \{b_1, \dots, b_n\} \subset X$

$$\mathrm{EMD}_X(A, B) \stackrel{\text{def}}{=} \min_{\substack{M:[n]\to[n] \\ \text{bijection}}} \sum_{i=1}^{n} d_X(a_i, b_{M(i)})$$



(Min cost bipartite matching with triangle inequality)

Focus: output α-approximation R to $\mathrm{EMD}_X(A, B)$

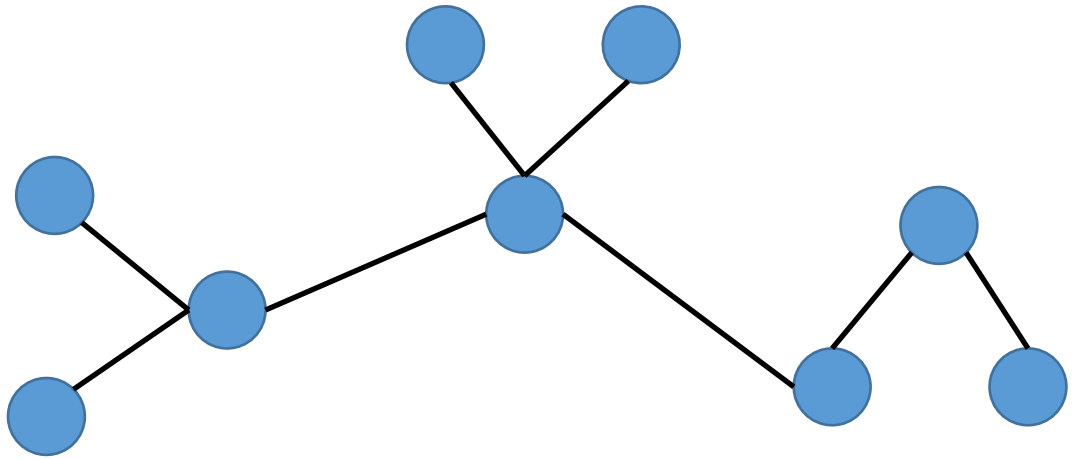
$$\mathrm{EMD}_X(A, B) \leq R \leq \alpha \cdot \mathrm{EMD}_X(A, B)$$

Metric Minimum Spanning Tree (MST)

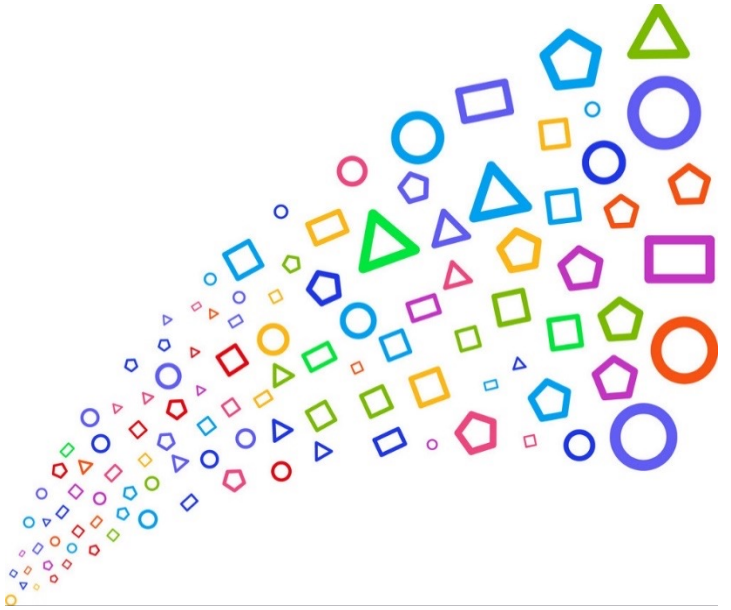
Metric space: (X, d_x)

Subset: $A = \{a_1, \ldots, a_n\} \subset X$

$$\mathrm{MST}_X(A) \stackrel{\text{def}}{=} \min_{\substack{\text{spanning} \\ \text{tree } T \text{ of } A}} \sum_{e=(u,v)\in T} d_X(u, v)$$



Geometric Streams



Geometric Streaming Model [Indyk '04]

Work in ℓ_1 metric: $(X, d_x) = ([\Delta]^d, \ell_1)$

- $[\Delta]^d = \{0,1, \dots, \Delta\}^d$

Points in $A, B \subset [\Delta]^d$ arrive in arbitrary order stream.

- Points can be inserted or deleted
- Want $\text{poly}(d, \log n\Delta)$ space

Our Contributions (EMD)

Low-Dimension

Approximation	Space	Paper
$O(d \log \Delta)$	$\text{poly} \log(n, d, \Delta)$	[Indyk Thaper '04]
$O\left(\frac{1}{\epsilon}\right)$, for $\mathbb{R}^2$	$O(n^{\epsilon})$	[Andoni, Ba, Indyk, Woodruff FOCS '09]

High-Dimension

Approximation	Space	Paper
$O(\log n \log(\text{d}\Delta))$	$\text{poly} \log(n, d, \Delta)$	[Andoni, Indyk, Krauthgamer SODA '08]

Our Contributions (EMD)

Low-Dimension

Approximation	Space	Paper
$O(d \log \Delta)$	$\text{poly}\log(n, d, \Delta)$	[Indyk Thaper '04]
$O\left(\frac{1}{\epsilon}\right)$, for $\mathbb{R}^2$	$O(n^{\epsilon})$	[Andoni, Ba, Indyk, Woodruff FOCS '09]

High-Dimension

Approximation	Space	Paper
$O(\log n \log(d\Delta))$	$\text{poly}\log(n, d, \Delta)$	[Andoni, Indyk, Krauthgamer SODA '08]
$\boldsymbol{\tilde{O}(\log n)}$	$\boldsymbol{\text{polylog}(n, d, \Delta)}$	**[This work]**

Two-pass

Our Contributions (EMD)

Low-Dimension

Approximation	Space	Paper
$O(d \log \Delta)$	$\text{poly}\log(n, d, \Delta)$	[Indyk Thaper '04]
$O\left(\frac{1}{\epsilon}\right)$, for $\mathbb{R}^2$	$O(n^{\epsilon})$	[Andoni, Ba, Indyk, Woodruff FOCS '09]

High-Dimension

Approximation	Space	Paper
$O(\log n \log(d\Delta))$	$\text{poly}\log(n, d, \Delta)$	[Andoni, Indyk, Krauthgamer SODA '08]
$\boldsymbol{\tilde{O}(\log n)}$	$\boldsymbol{\text{polylog}(n, d, \Delta)}$ $\boldsymbol{d\Delta \cdot \text{polylog}(n, d)}$	[This work]

***Bounded Overlap**

$$\frac{|A \cap B|}{|A \cup B|} = 1 - \Omega(1)$$

Two-pass

One-Pass*

Our Contributions (MST)

Low-Dimension

	Approximation	Space	Paper
	$O(d \log \Delta)$	$\text{poly}\log(n, d, \Delta)$	[Indyk Thaper '04]
	$O(\log n \log(d\Delta))$	$\text{poly}\log(n, d, \Delta)$	Follows from[Andoni, Indyk, Krauthgamer SODA '08]
One-Pass	$\boldsymbol{\tilde{O}(\log n)}$	$\boldsymbol{\text{poly}\log(n, d, \Delta)}$	**[This work]**

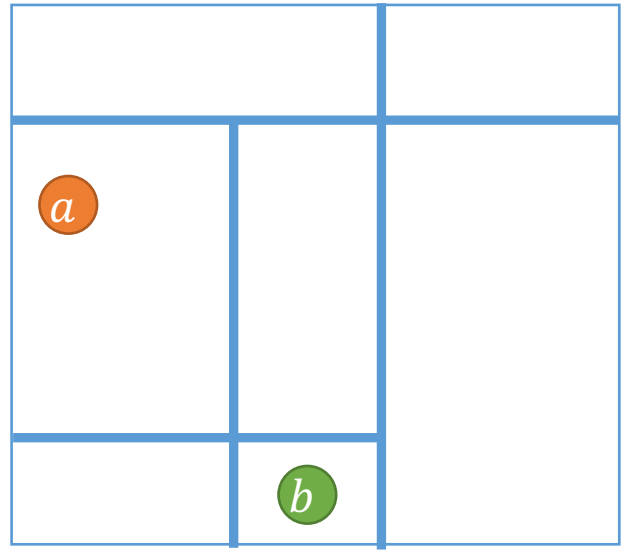
Talk Outline

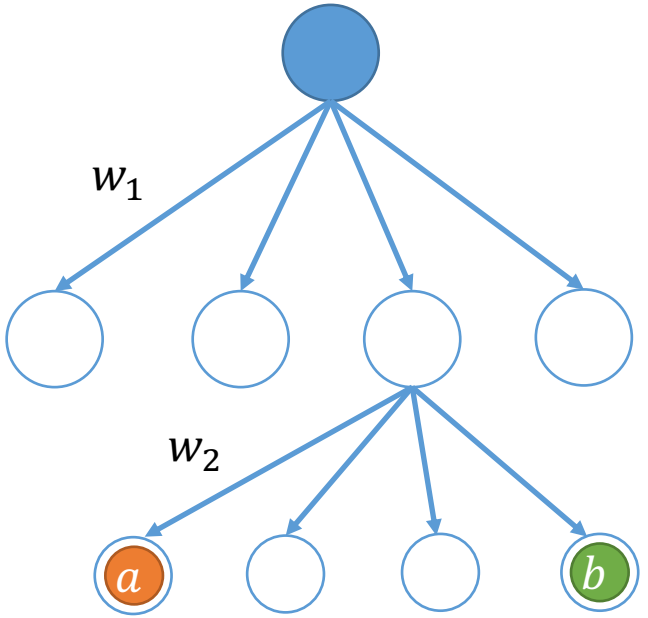
1. Quadtree Algorithm
2. New Data-Dependent Tree Embedding
3. From Data-Dependent Quadtrees to Streaming Algorithms

Quadtree Algorithm

How to sketch EMD/MST:

1. Recursively subdivide $\mathbb{R}^d$, creating tree
2. Map points to edge-weighted tree (tree metric)
3. Estimate cost of EMD/MST in tree metric

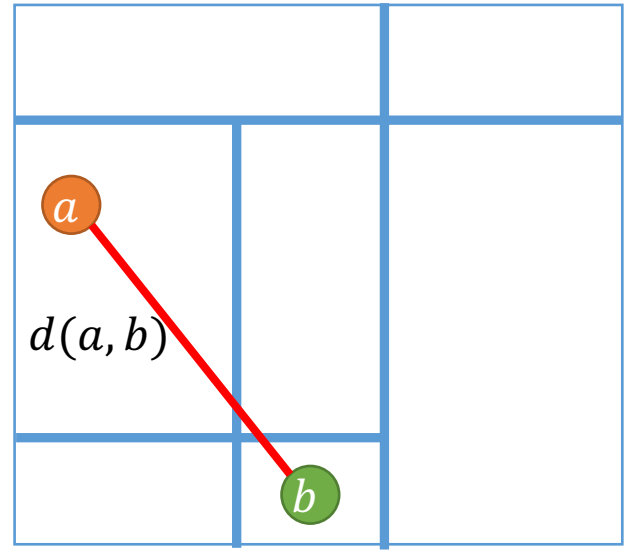


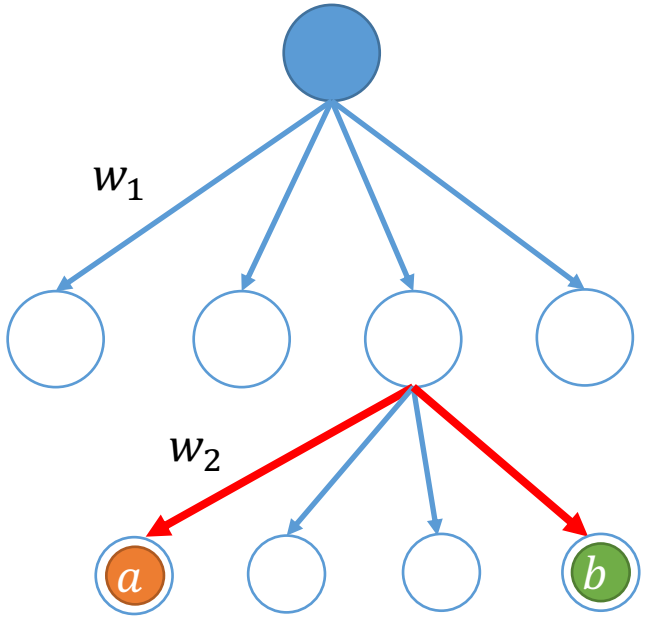


Quadtree Algorithm

How to sketch EMD/MST:

1. Recursively subdivide $\mathbb{R}^d$, creating tree
2. Map points to edge-weighted tree (tree metric)
3. Estimate cost of EMD/MST in tree metric



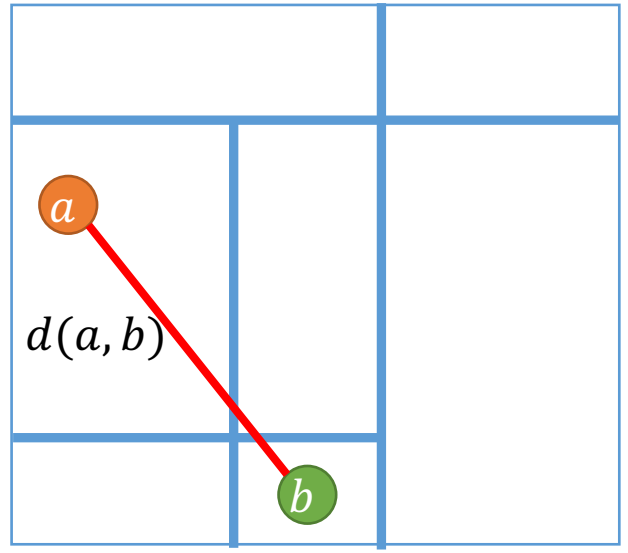


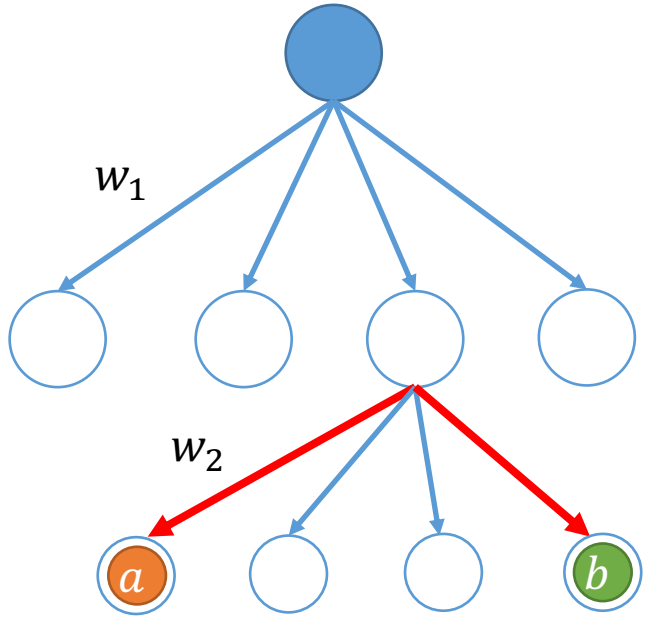
➢Want distances in tree to approximate original distances

Quadtree Algorithm

How to sketch EMD/MST:

1. Recursively subdivide $\mathbb{R}^d$, creating tree
2. Map points to edge-weighted tree (tree metric)
3. Estimate cost of EMD/MST in tree metric





- Want distances in tree to approximate original distances

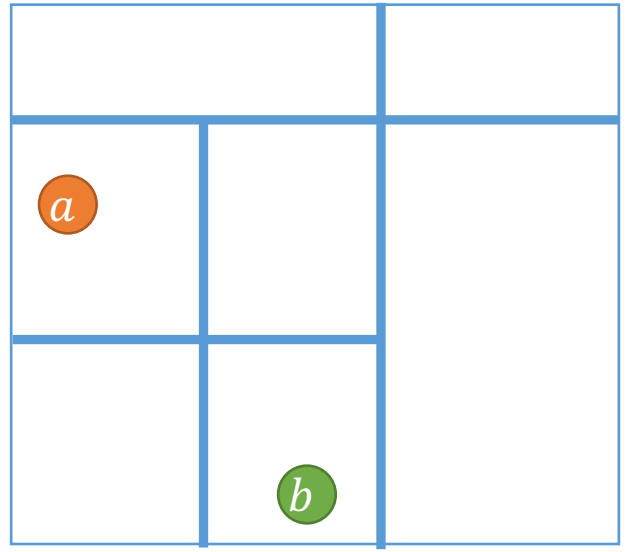
Why tree embeddings?

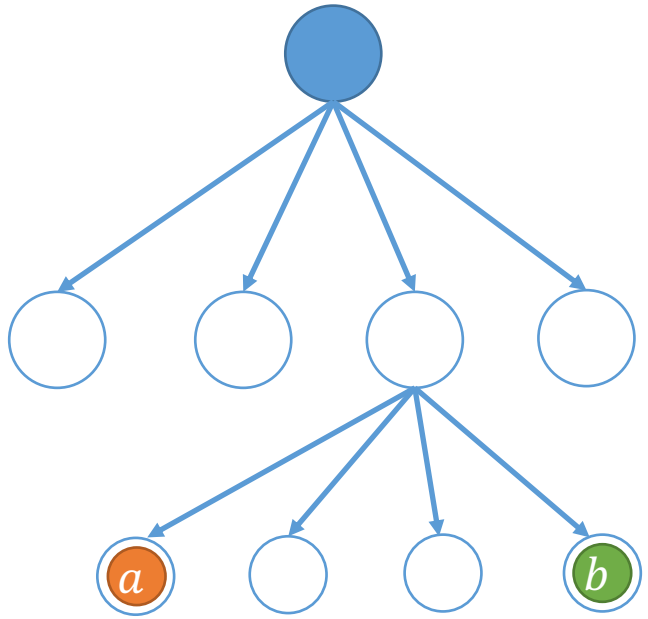
- EMD/MST in trees can be embedded into ℓ_1/ℓ_0
- ℓ_1/ℓ_0 can be estimated in a stream
  - [Indyk '04], [Flajolet-Martin '83], [Kane-Nelson-Woodruff '10]

Quadtree Algorithm

[Indyk Thaper '04], [Andoni, Indyk, Krauthgamer '08]

- Impose randomly shifted grid at $\log d\Delta$-scales
- $T :=$ recursion tree, $v \in T$ contain set $S_v \subset [\Delta]^d$

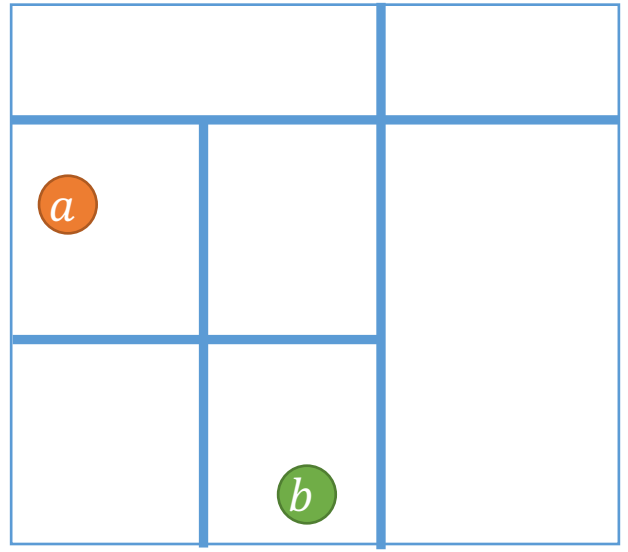


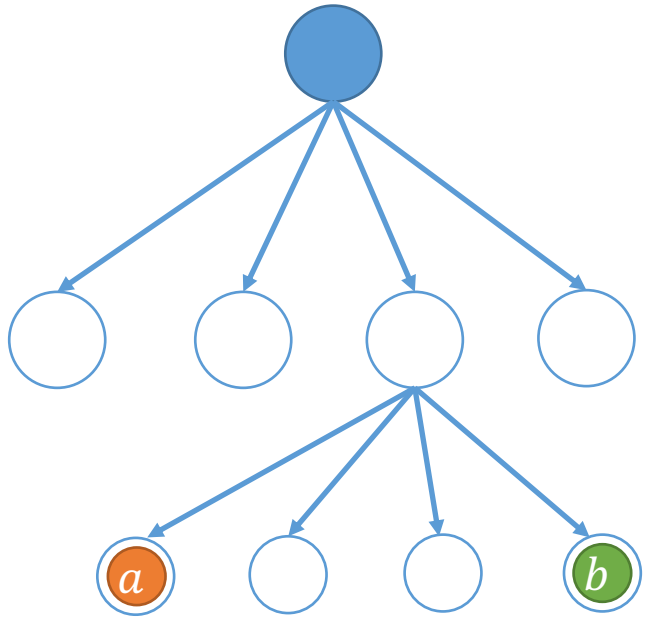


Quadtree Algorithm

[Indyk Thaper '04], [Andoni, Indyk, Krauthgamer '08]

- Impose randomly shifted grid at $\log d\Delta$-scales
- $T :=$ recursion tree, $v \in T$ contain set $S_v \subset [\Delta]^d$
- $w(u, v) = \frac{d\Delta}{2^i}$ if edge (u, v) at depth i.
  - If $\|a - b\|_1 < \frac{d\Delta}{2^i}$, want a, b together at depth i

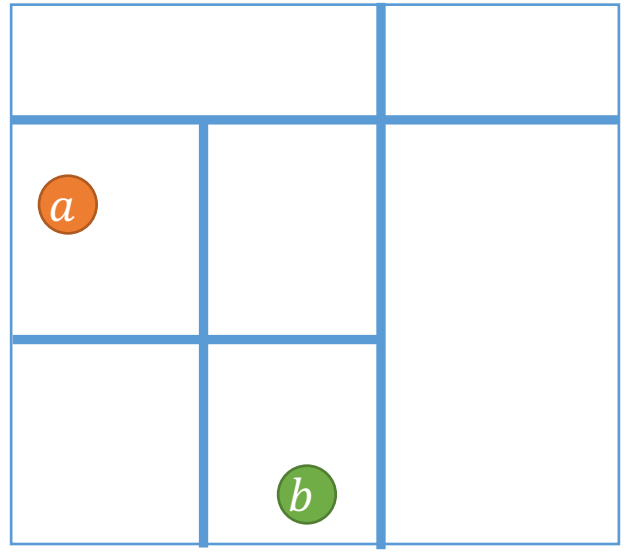


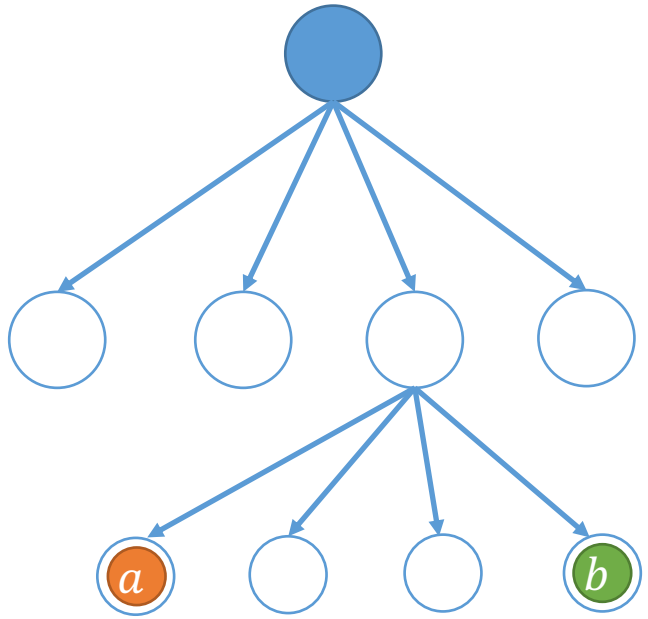


Quadtree Algorithm

[Indyk Thaper '04], [Andoni, Indyk, Krauthgamer '08]

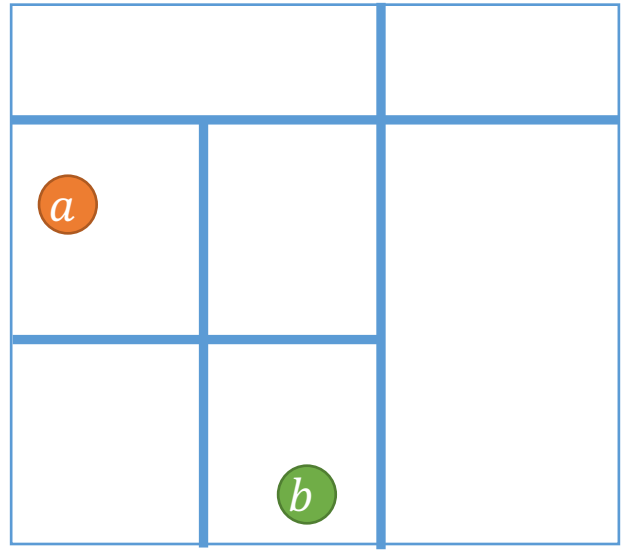
- Impose randomly shifted grid at $\log d\Delta$-scales
- $T :=$ recursion tree, $v \in T$ contain set $S_v \subset [\Delta]^d$
- $w(u, v) = \frac{d\Delta}{2^i}$ if edge (u, v) at depth i.
  - If $\|a - b\|_1 < \frac{d\Delta}{2^i}$, want a, b together at depth i
- Map $f(a) = v$, where $v \in T$ is the leaf with $S_v = \{a\}$
  - $f: [\Delta]^d \to T$



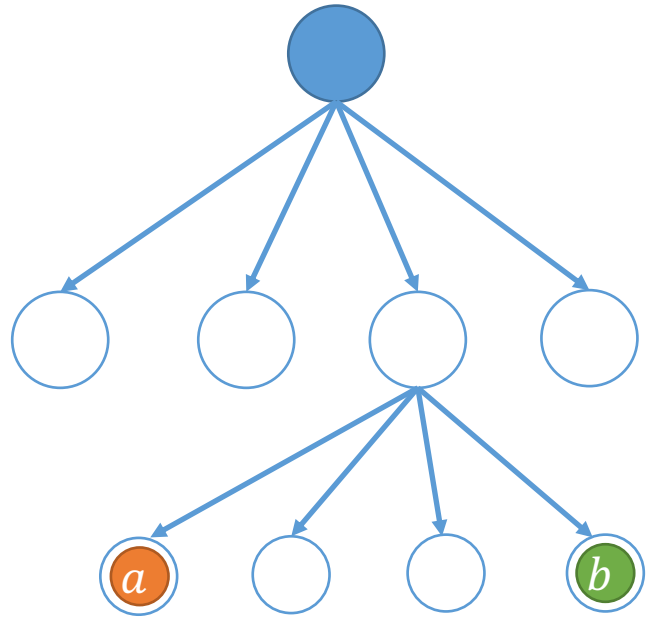


Quadtree Algorithm

[Indyk Thaper '04], [Andoni, Indyk, Krauthgamer '08]

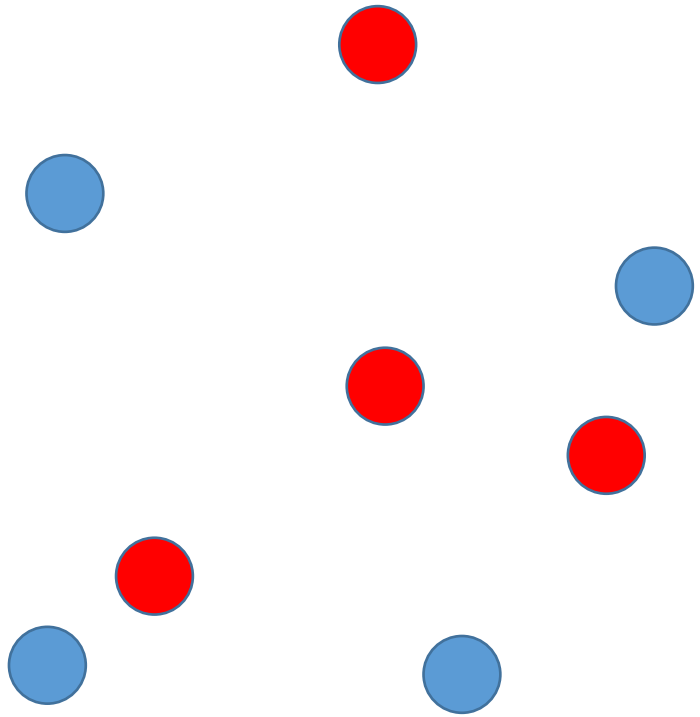


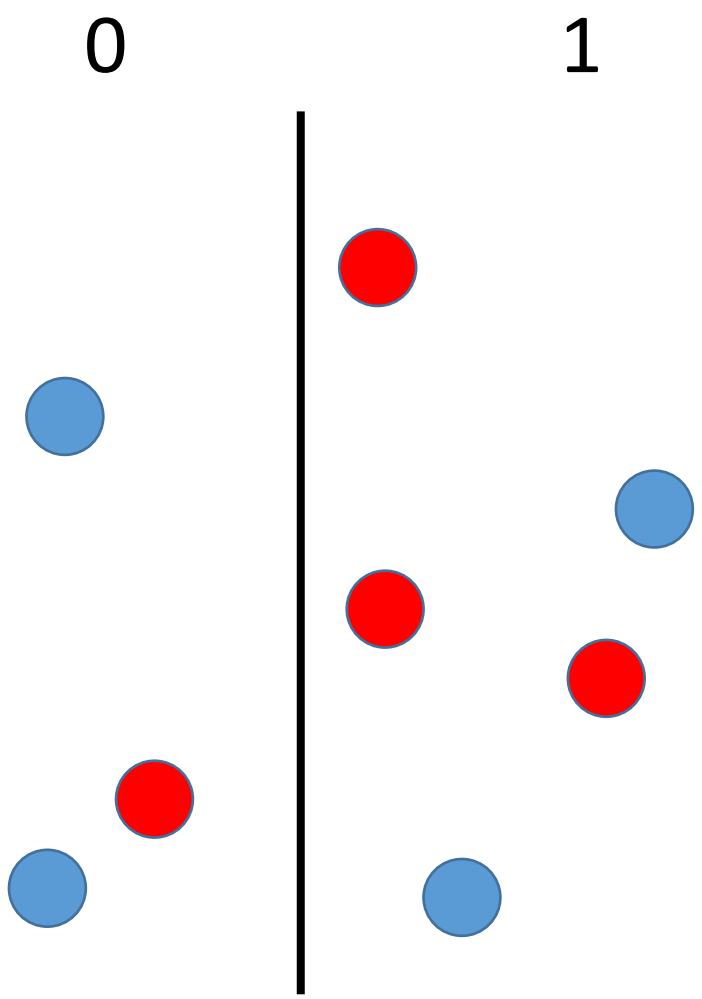
- Impose randomly shifted grid at $\log d\Delta$-scales
- $T :=$ recursion tree, $v \in T$ contain set $S_v \subset [\Delta]^d$
- $w(u, v) = \frac{d\Delta}{2^i}$ if edge (u, v) at depth i.
  - If $\|a - b\|_1 < \frac{d\Delta}{2^i}$, want a, b together at depth i
- Map $f(a) = v$, where $v \in T$ is the leaf with $S_v = \{a\}$
  - $f: [\Delta]^d \to T$

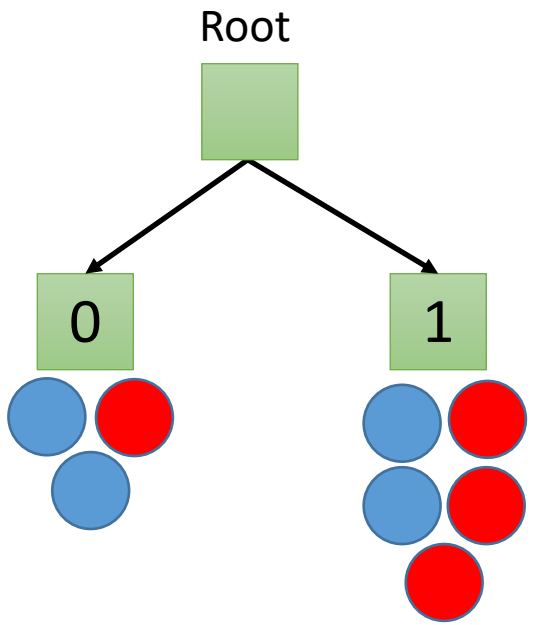


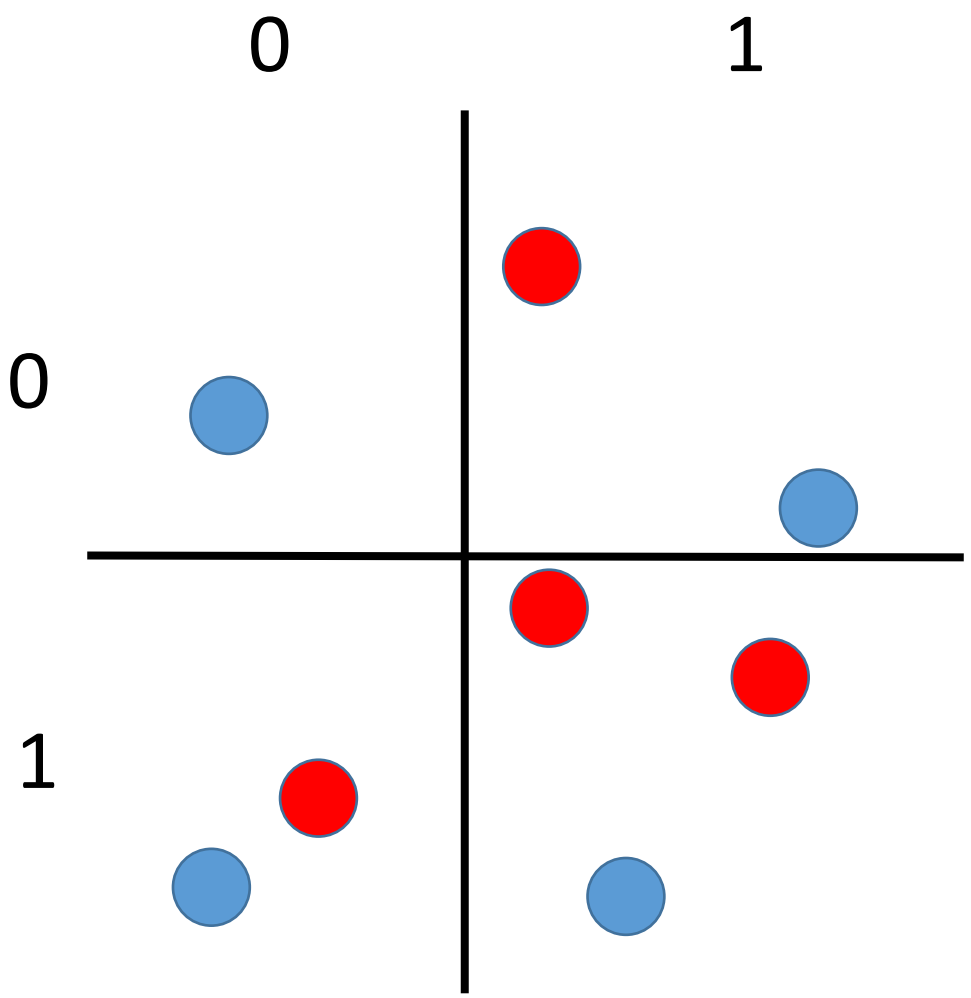
Example: to approximate EMD:

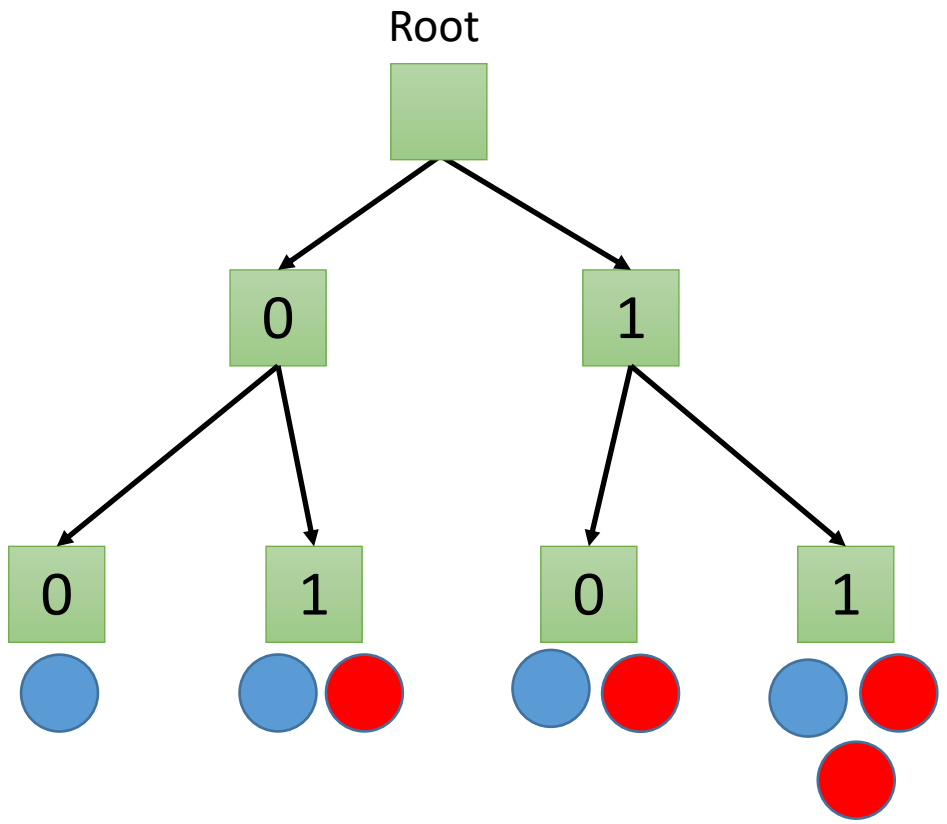
- Perform greedy-bottom up matching between $f(A), f(B)$

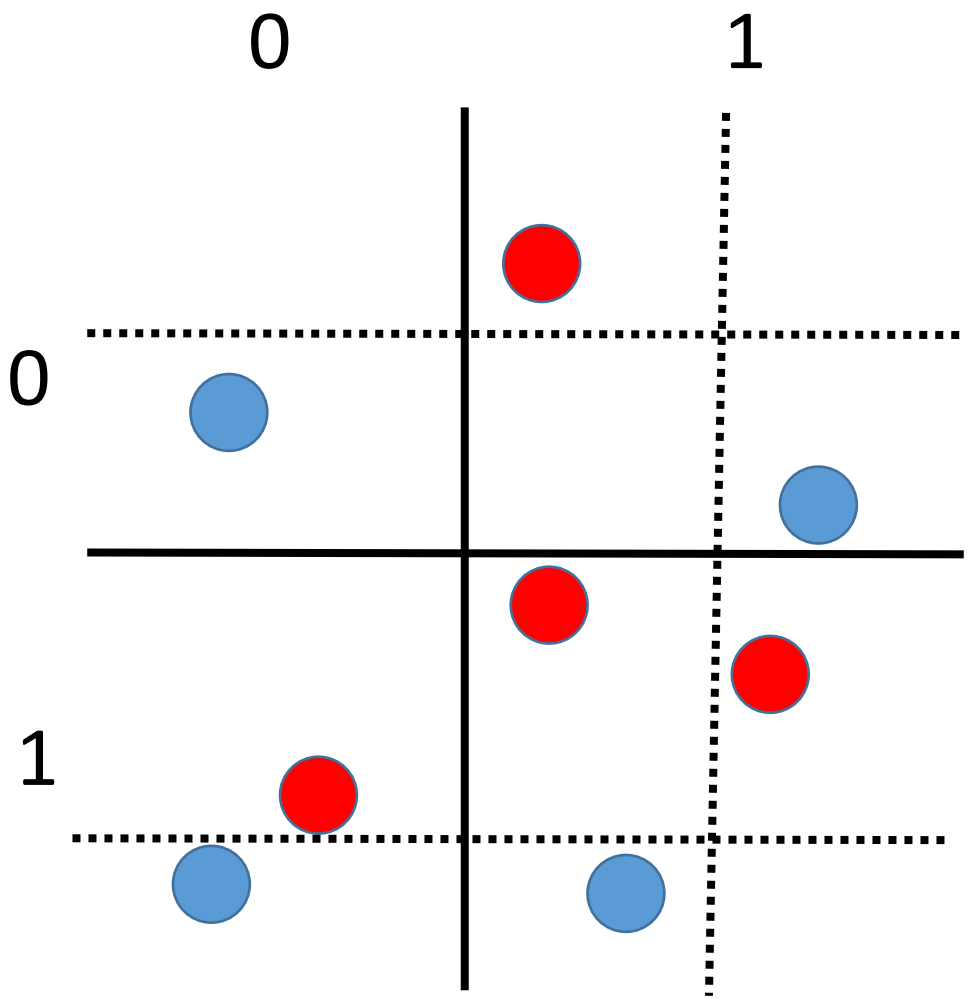

Root

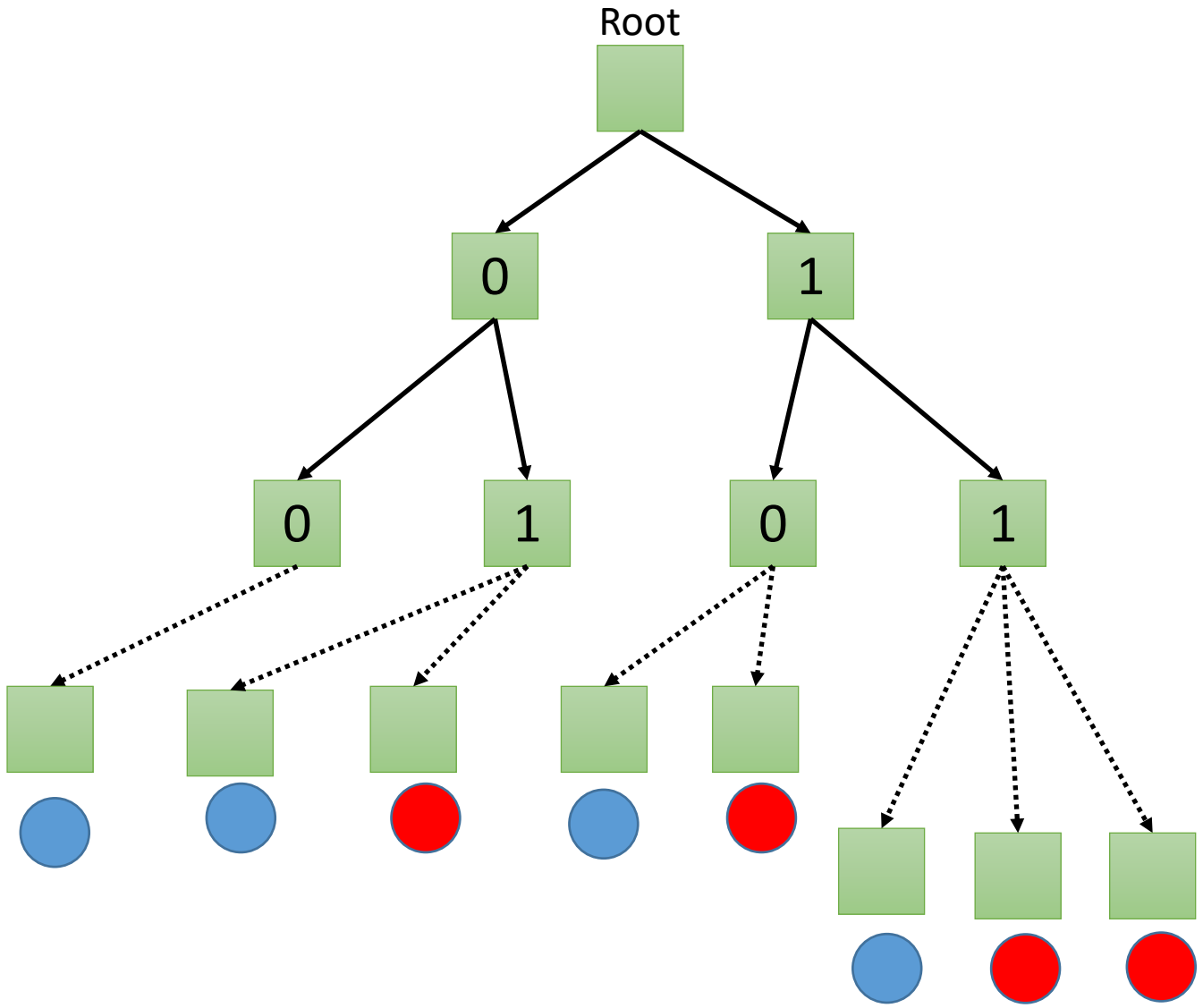

0
1

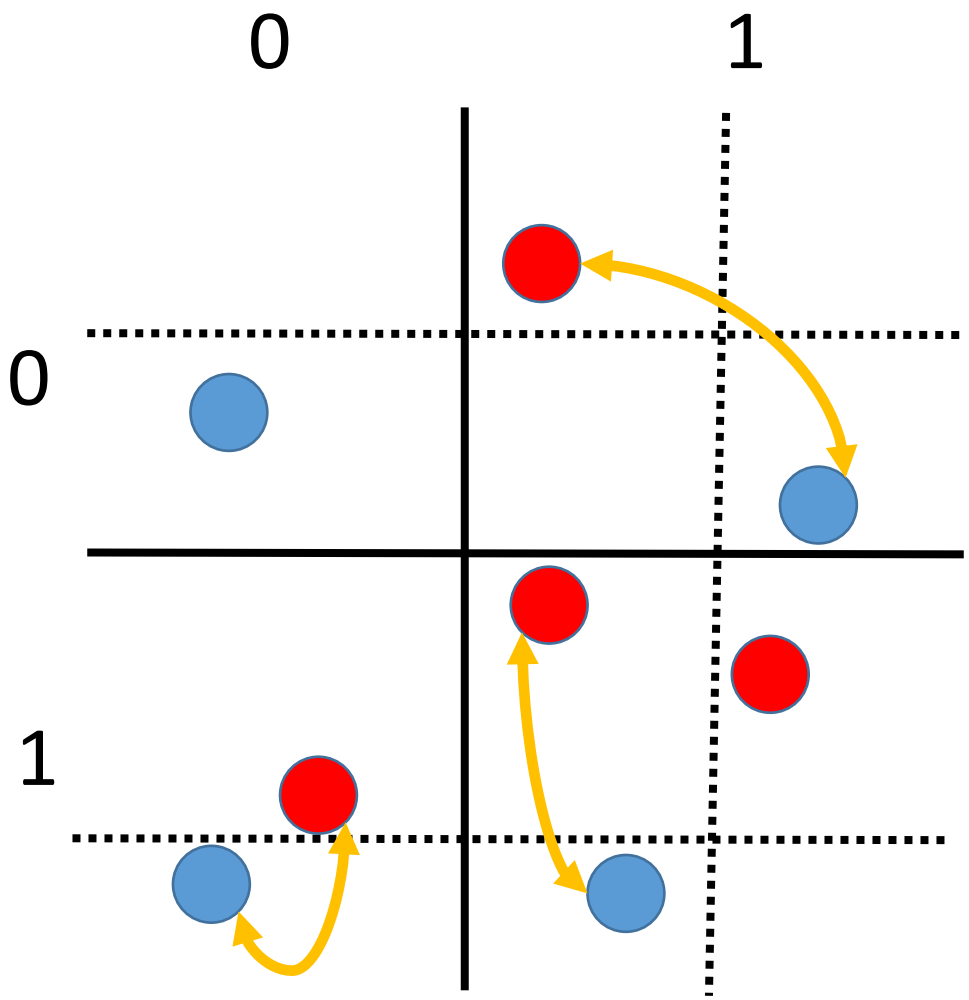

Root
0
1

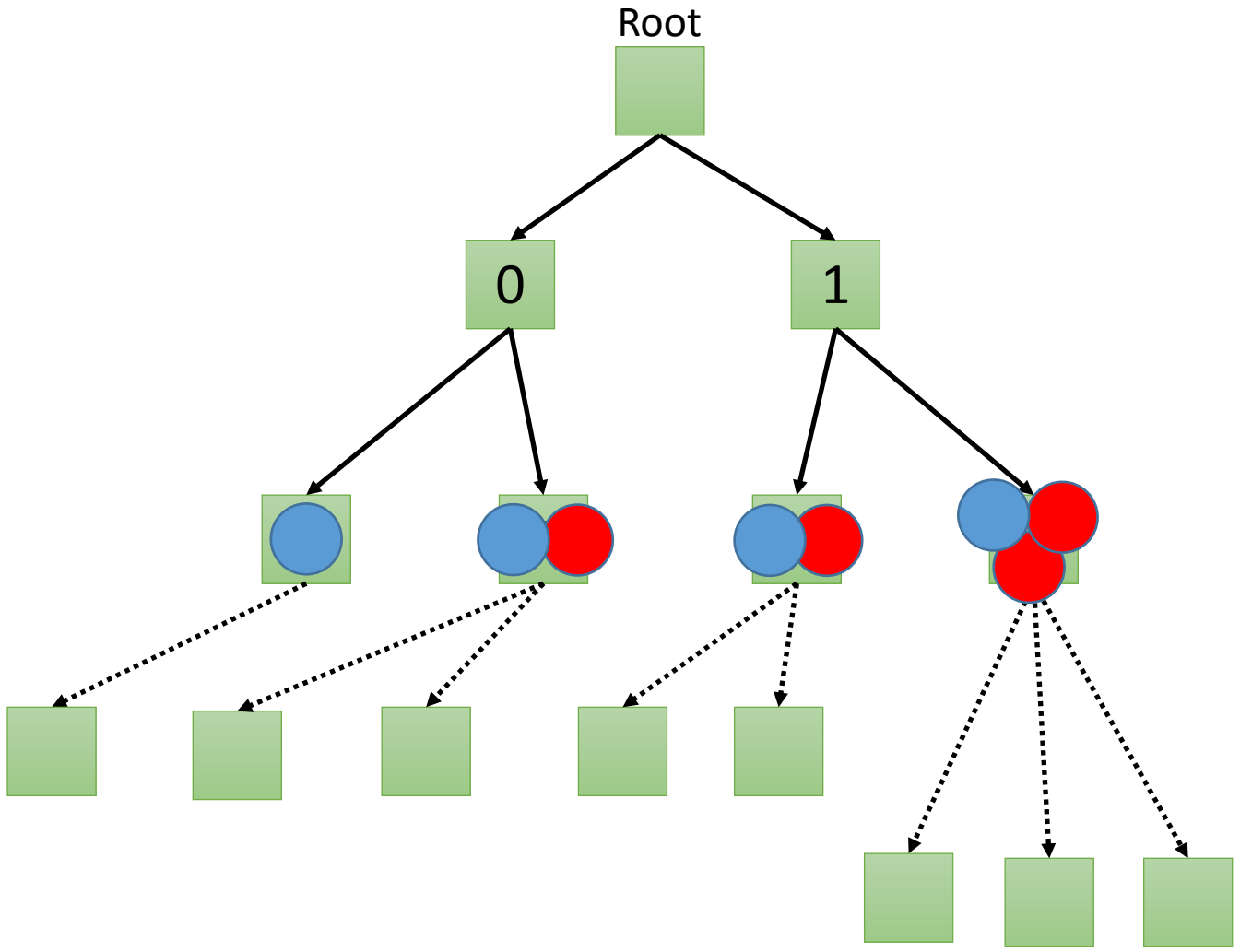

0
1
0
1

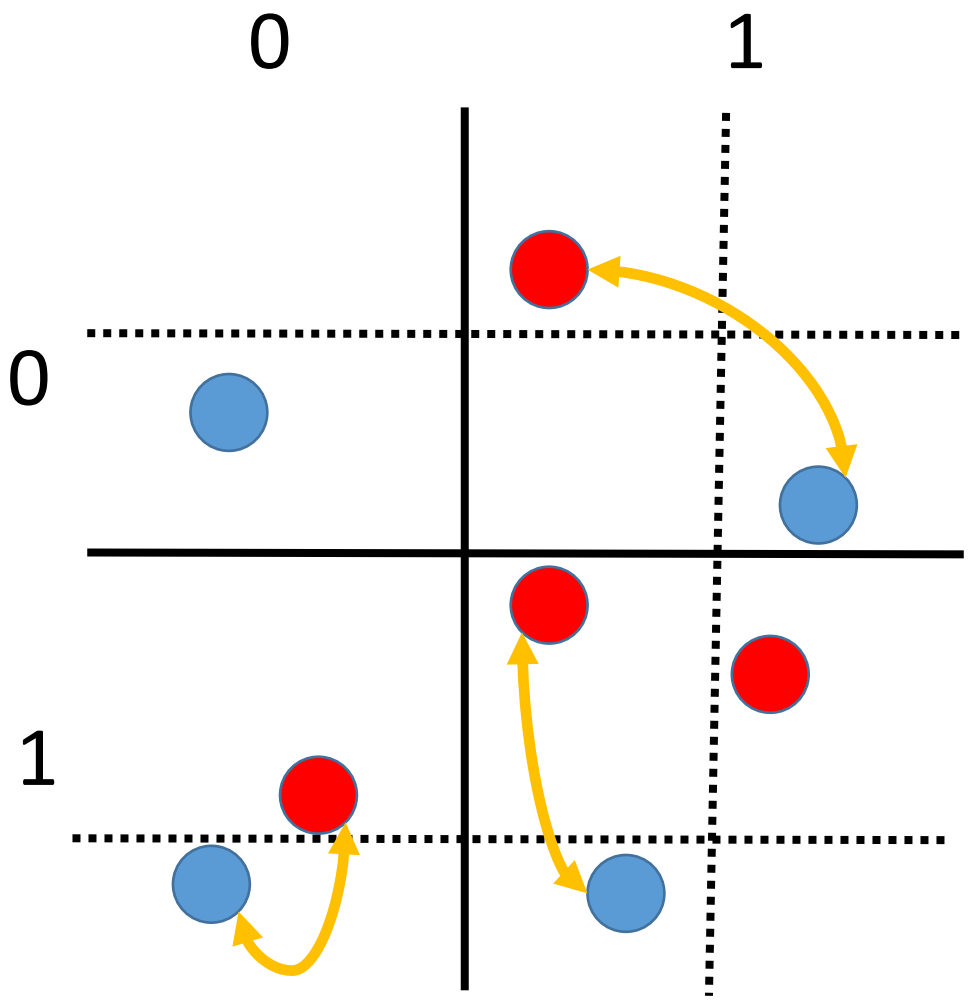

Root
0
1
0
1
0
1

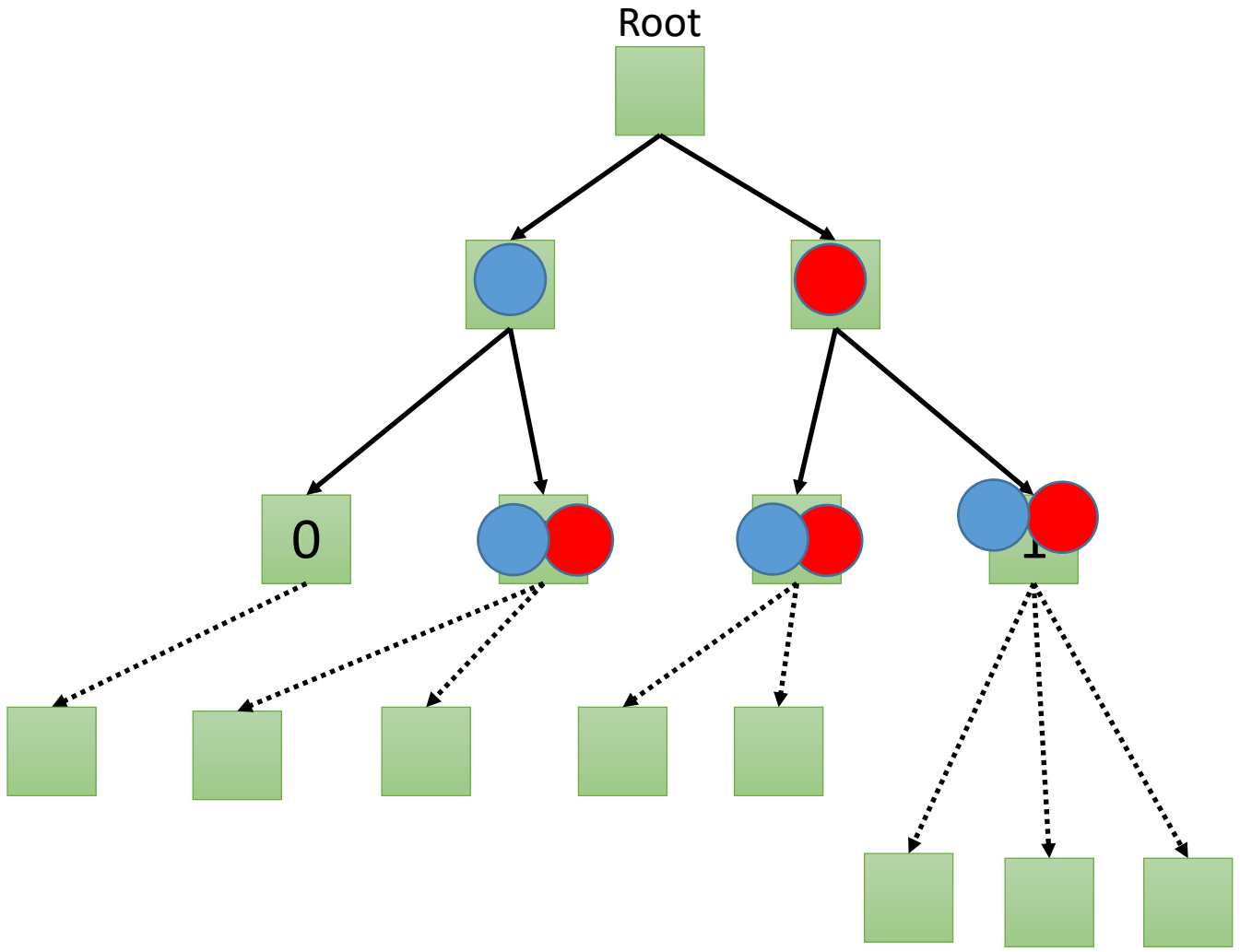

0
1
0
1

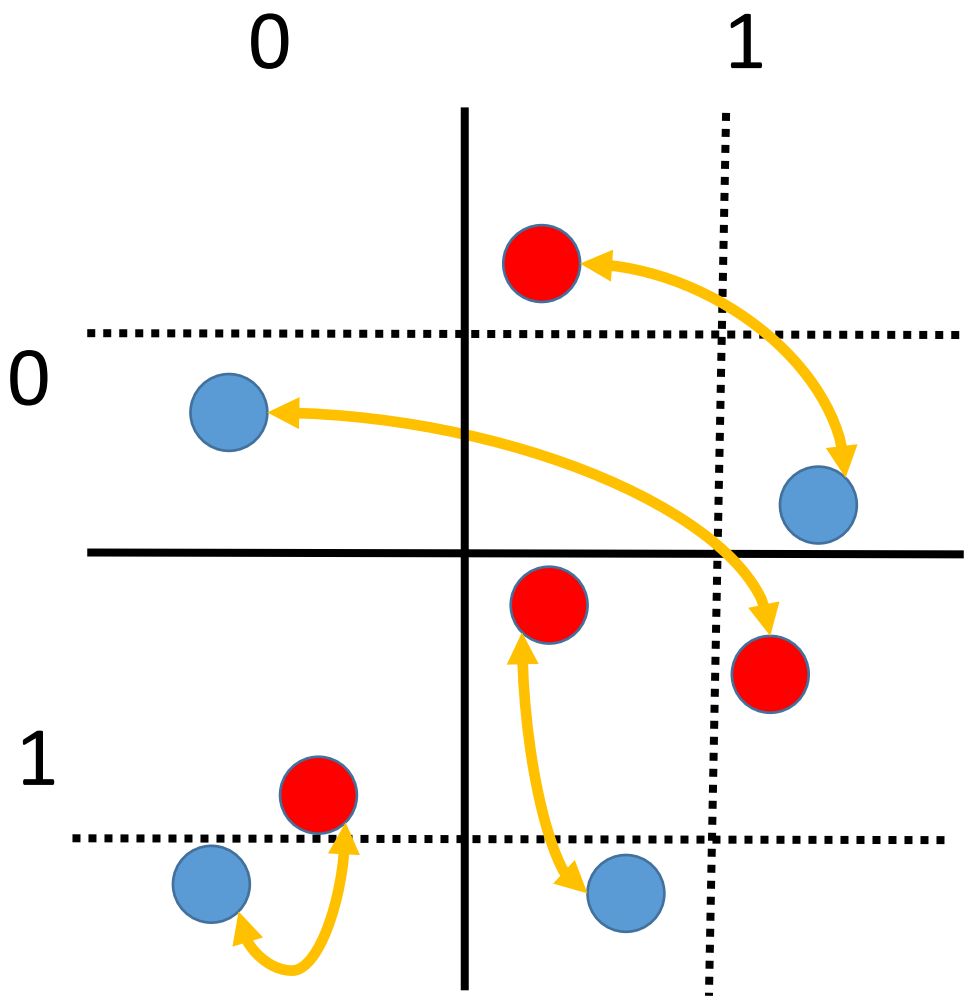

Root
0
1
0
1
0
1
0
1

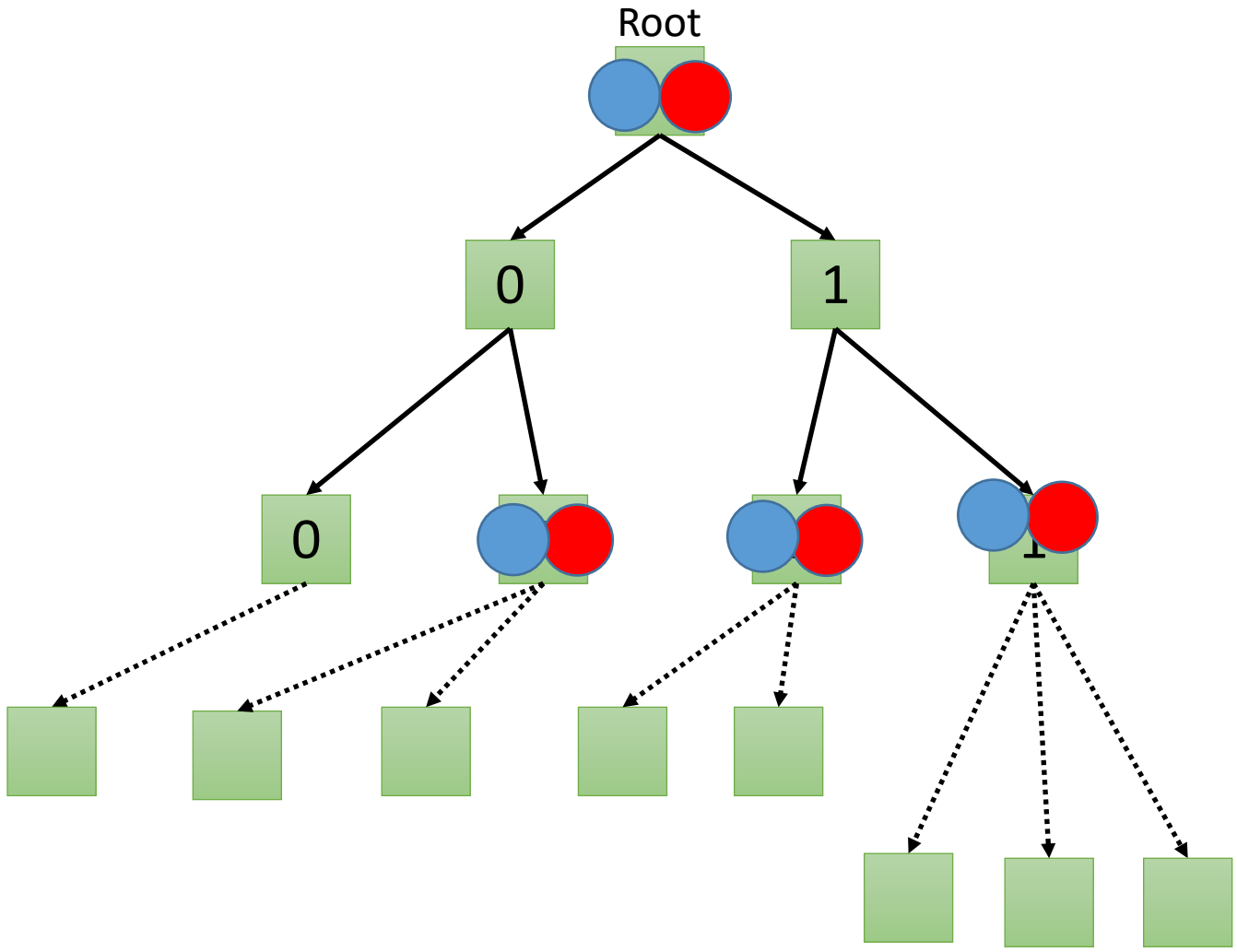

0
1
0
1

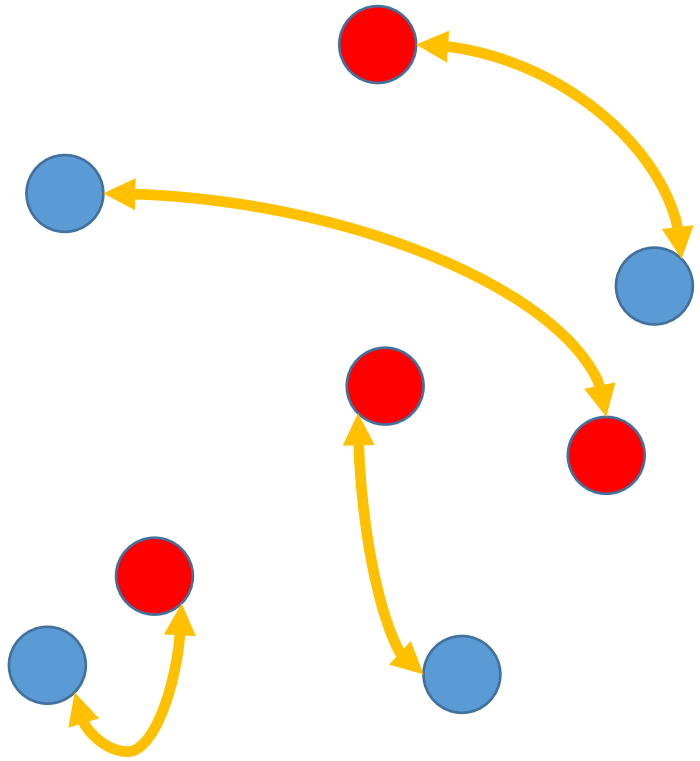

Root
0
1

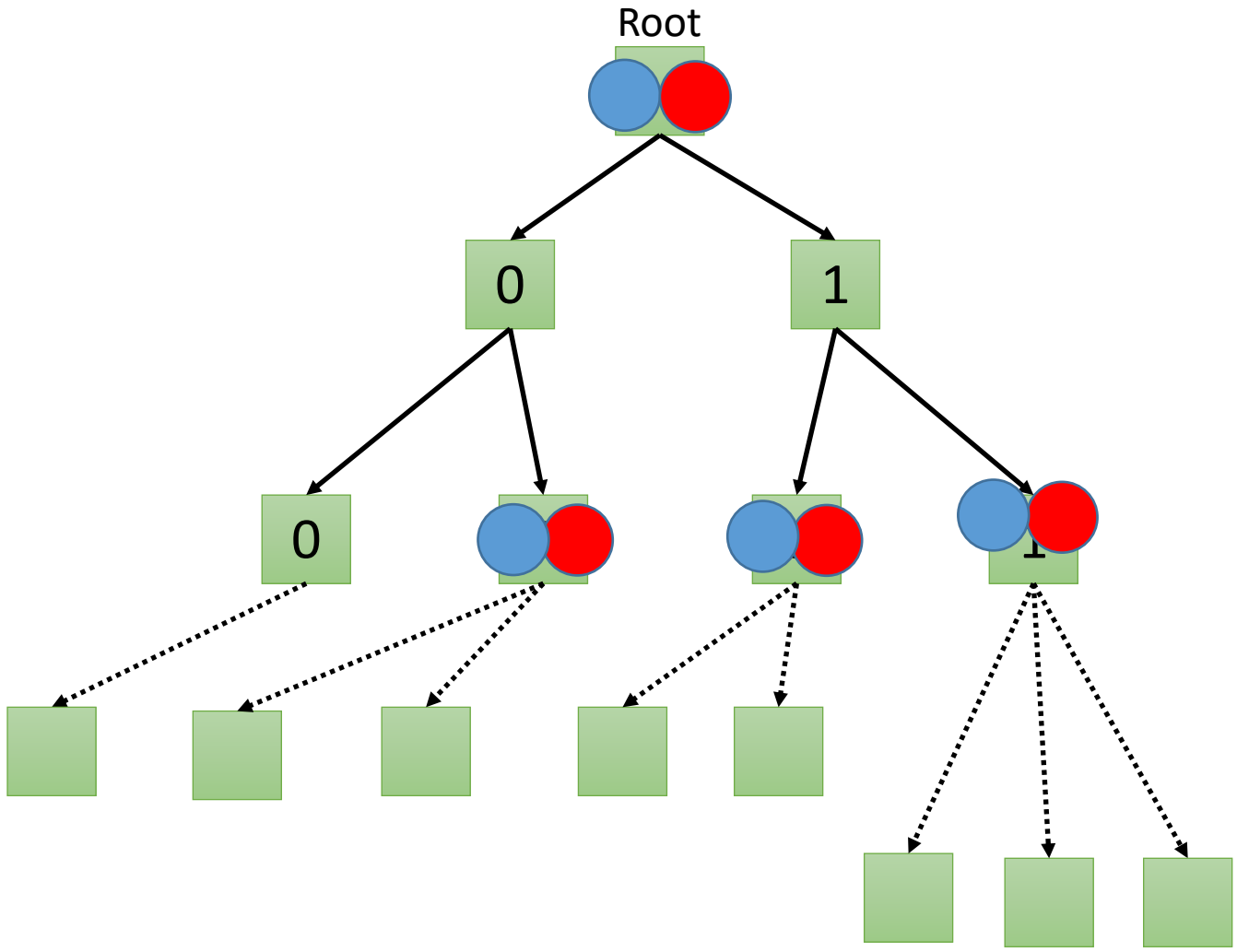

0
1
0
1


Root
0
1


0
1
0
1


Root
0
1
0
1

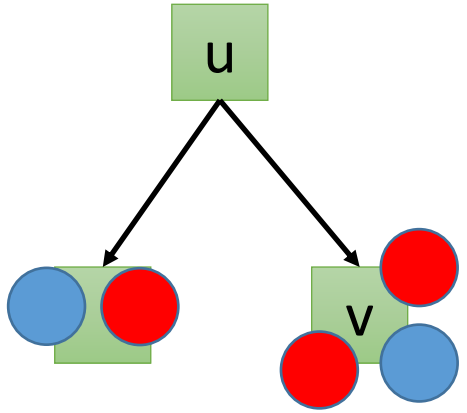



Root
0
1
0
1

Embedding EMD into ℓ_1

- Greedy matching is optimal for tree-metrics. Embed this into ℓ_1
- For $v \in T$, let $A_v := \{a \in A : a \in S_v\}$, similarly B_v
- Use Quadtree edge-weights $w(u, v)$

$$\mathrm{EMD}_T(A, B) = \sum_{(u,v)\in T} w(u, v) \cdot ||A_v| - |B_v||$$

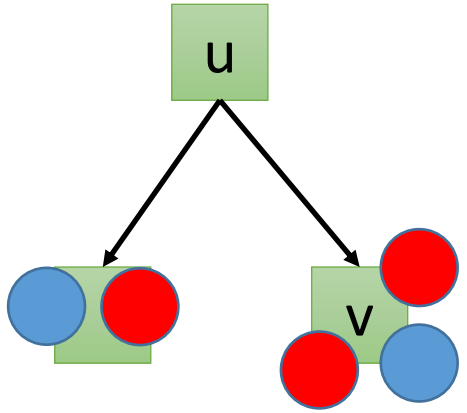


Embedding EMD into ℓ_1

- Greedy matching is optimal for tree-metrics. Embed this into ℓ_1
- For $v \in T$, let $A_v := \{a \in A : a \in S_v\}$, similarly B_v
- Use Quadtree edge-weights $w(u, v)$

$$\mathrm{EMD}_T(A, B) = \sum_{(u,v)\in T} w(u, v) \cdot ||A_v| - |B_v||$$

- Define z via $z_v = w(u, v) \cdot (A_v - B_v)$
- Estimate $\|z\|_1$ to $(1 \pm \epsilon)$ in a stream
  - Cauchy Sketch [Indyk '04]



What was known: There is a randomized tree embedding $f: [\Delta]^d \to T$ such that for any (A, B), the EMD/MST in T is a

$$O(\log n \cdot \min\{\log n, \log d\Delta\})$$

approximation of EMD/MST in $\mathbb{R}^d$.

- [Andoni, Indyk, Krauthgamer 08], [Backurs, Dong, Indyk, Razenstheyn, Wagner 20].

Talk Outline

1. Quadtree Algorithm
2. New Data-Dependent Tree Embedding
3. From Data-Dependent Quadtrees to Streaming Algorithms

Data-Dependent Tree Embeddings

➢Instead of using fixed edge weights $w(u,v)$, use data dependent weights

$$\widetilde{w}(u,v) = \mathbb{E}_{\substack{x \sim u \\ y \sim v}} [\|x - y\|_1]$$

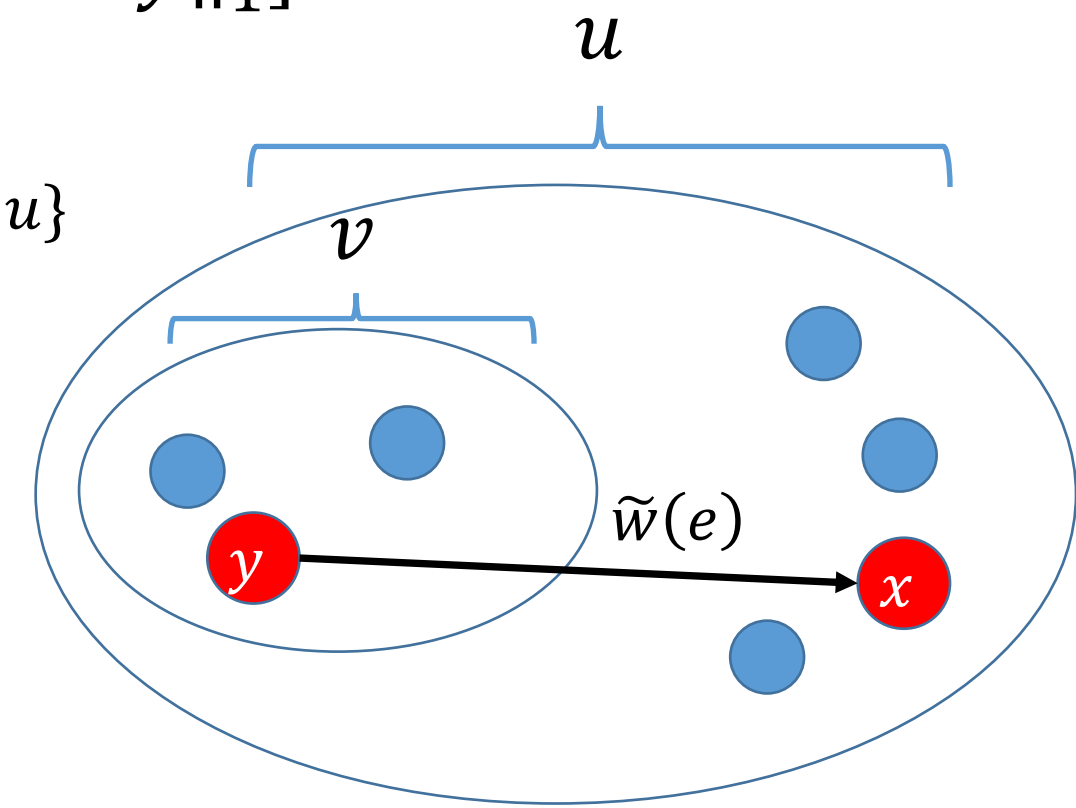
- Where $u \coloneqq \{\text{set of all points landing in vertex } u\}$
- Same vertices, different weights

Data-Dependent Tree Embeddings

➢Instead of using fixed edge weights $w(u,v)$, use data dependent weights

$$\widetilde{w}(u,v) = \mathbb{E}_{\substack{x\sim u \\ y\sim v}}[\|x-y\|_1]$$

- Where $u \coloneqq \{\text{set of all points landing in vertex } u\}$
- Same vertices, different weights



Data-Dependent Tree Embeddings

➢Instead of using fixed edge weights $w(u,v)$, use data dependent weights

$$\widetilde{w}(u,v) = \mathbb{E}_{\substack{x\sim u \\ y\sim v}}[\|x-y\|_1]$$

- Where $u \coloneqq \{\text{set of all points landing in vertex } u\}$
- Same vertices, different weights

Data-dependent weights give improved approximation!

What was known: There is a randomized tree embedding $f: [\Delta]^d \to T$ such that for any (A, B), the EMD/MST in T is a

$$O(\log n \cdot \min\{\log n, \log d\Delta\})$$

approximation of EMD/MST in $\mathbb{R}^d$.

- [Andoni, Indyk, Krauthgamer 08], [Backurs, Dong, Indyk, Razenstheyn, Wagner 20].

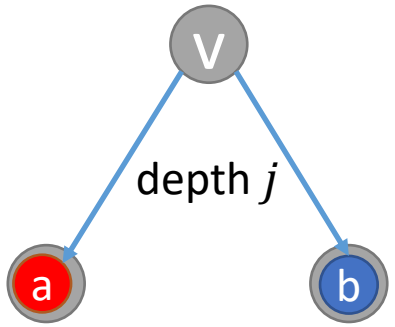
Theorem: [Chen-J-Levi-Waingarten] For any (A, B), there is a randomized tree embedding $\tilde{f}: \mathbb{R}^d \to \tilde{T}$ such that the EMD/MST in $\tilde{T}$ is a $\tilde{O}(\log n)$ approximation.

Proof: High Level

$$\widetilde{w}(u,v) = \mathbb{E}_{\substack{x\sim u \\ y\sim v}}[\|x-y\|_1]$$

$\widetilde{d_{\tilde{T}}}(a,b) :=$ tree metric using $\widetilde{w}$.

- Not hard to show $\|a-b\|_1 \leq \widetilde{d_{\tilde{T}}}(a,b)$.
- Need to show $\widetilde{d_{\tilde{T}}}(a,b) \leq \tilde{O}(\log n)\|a-b\|_1$
- $\widetilde{d_{\tilde{T}}}(a,b) \approx \widetilde{w}(u,v)$ if a,b split at vertex v



Proof: High Level

$$\widetilde{w}(u,v) = \mathbb{E}_{\substack{x\sim u \\ y\sim v}}[\|x-y\|_1]$$

$\widetilde{d_{\tilde{T}}}(a,b) :=$ tree metric using $\widetilde{w}$.

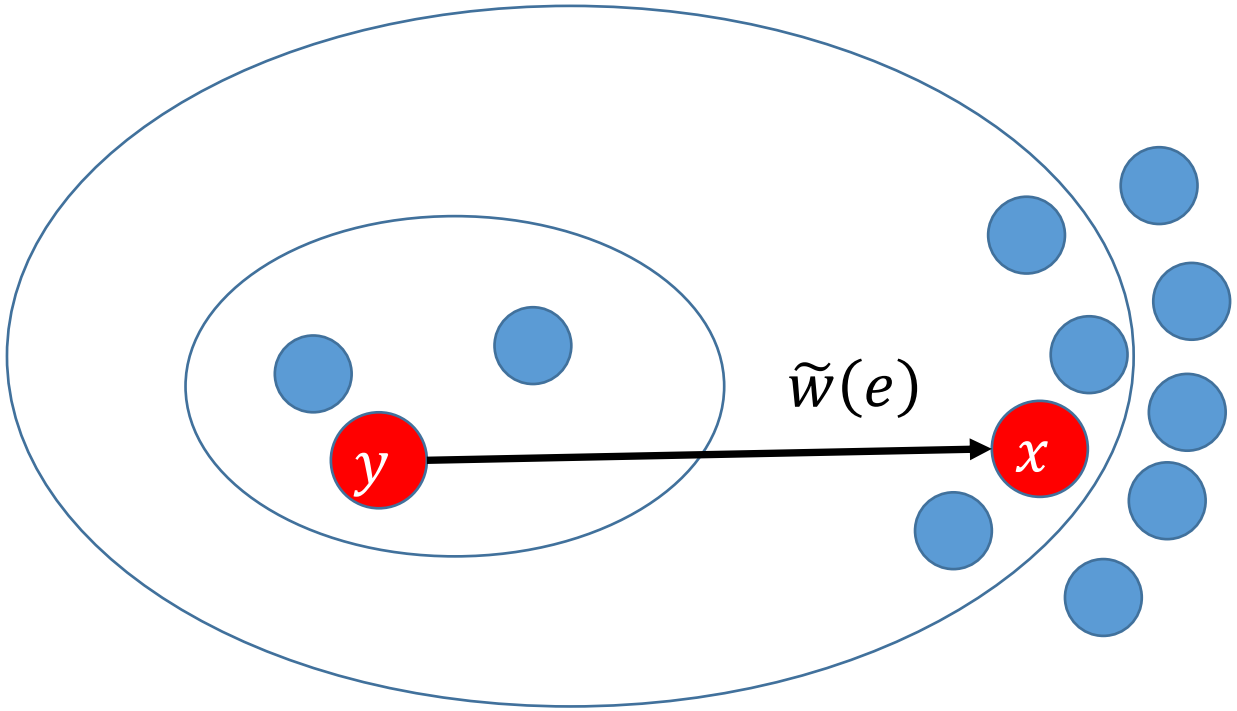
- Not hard to show $\|a-b\|_1 \leq \widetilde{d_{\tilde{T}}}(a,b)$.
- Need to show $\widetilde{d_{\tilde{T}}}(a,b) \leq \tilde{O}(\log n)\|a-b\|_1$
- $\widetilde{d_{\tilde{T}}}(a,b) \approx \widetilde{w}(u,v)$ if a,b split at vertex v
- Ideally: $\widetilde{w}(e) \approx \frac{d\Delta}{2^i}$ at depth i.
- Worst-case: $\widetilde{w}(e) = \log n \cdot \frac{d\Delta}{2^i}$

Proof: High Level

$\widetilde{d_{\tilde{T}}}(a,b) :=$ tree metric using $\tilde{w}$.

- Not hard to show $\|a-b\|_1 \leq \widetilde{d_{\tilde{T}}}(a,b)$.
- Need to show $\widetilde{d_{\tilde{T}}}(a,b) \leq \tilde{O}(\log n)\|a-b\|_1$
- $\widetilde{d_{\tilde{T}}}(a,b) \approx \tilde{w}(u,v)$ if a,b split at vertex v
- Ideally: $\tilde{w}(e) \approx \frac{d\Delta}{2^i}$ at depth i.
- Worst-case: $\tilde{w}(e) = \log n \cdot \frac{d\Delta}{2^i}$

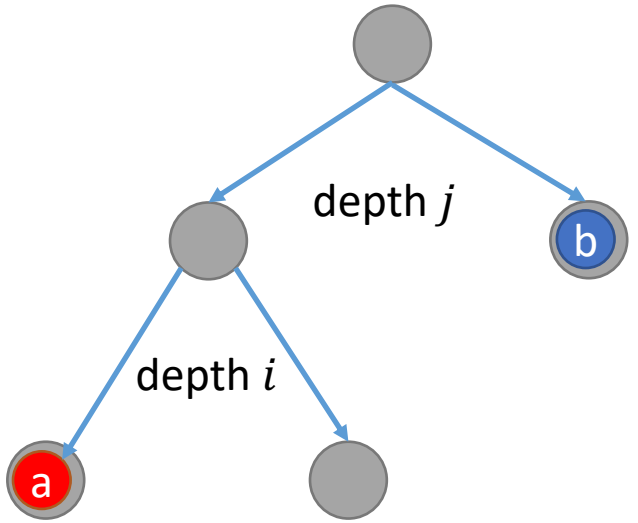
$$\tilde{w}(u,v) = \mathbb{E}_{\substack{x\sim u \\ y\sim v}}[\|x-y\|_1]$$



Worst-Case Bound

- Suppose $\|a - b\|_1 = \frac{d\Delta}{2^i}$
- a, b split early at depth $j < i$ with probability α_j. When this happens:

$$\widetilde{d_T}(a, b) \leq \frac{1}{\alpha_j} \cdot \log n \, \|a - b\|_1$$

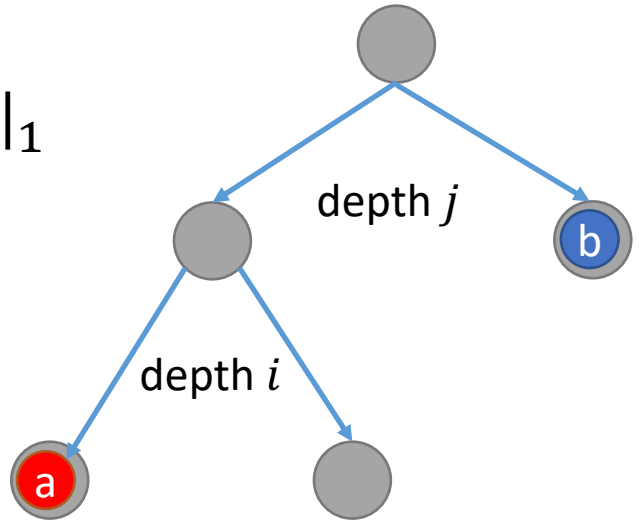


Worst-Case Bound

- Suppose $\|a-b\|_1 = \frac{d\Delta}{2^i}$
- a, b split early at depth $j < i$ with probability α_j. When this happens:

$$\widetilde{d_T}(a,b) \le \frac{1}{\alpha_j} \cdot \log n \, \|a-b\|_1$$

$$\mathbb{E}\left[\widetilde{d_T}(a,b)\right] \le \sum_{j=0}^{i} \alpha_j \cdot \frac{1}{\alpha_j} (\log n \, \|a-b\|_1) \le \log n \log d\Delta \cdot \|a-b\|_1$$



Data-Dependent Bound

Key idea: Show that $\widetilde{w}(e_j)$ cannot grow many times up the tree

Lemma: if $e_1, \dots, e_t$ be any path down the tree. Then

- $\widetilde{w}(e_j) \geq \log n \frac{d\Delta}{2^j}$ happens at most once
- $\widetilde{w}(e_j) \geq \frac{1}{2} \log n \frac{d\Delta}{2^j}$ at most twice

$\vdots$

- $\widetilde{w}(e_j) \geq \frac{1}{2^t} \log n \frac{d\Delta}{2^j}$ at most 2^t times

➢ Only $\log \log n$ levels!

Data-Dependent Bound

Key idea: Show that $\widetilde{w}(e_j)$ cannot grow many times up the tree

Lemma: if $e_1, \dots, e_t$ be any path down the tree. Then

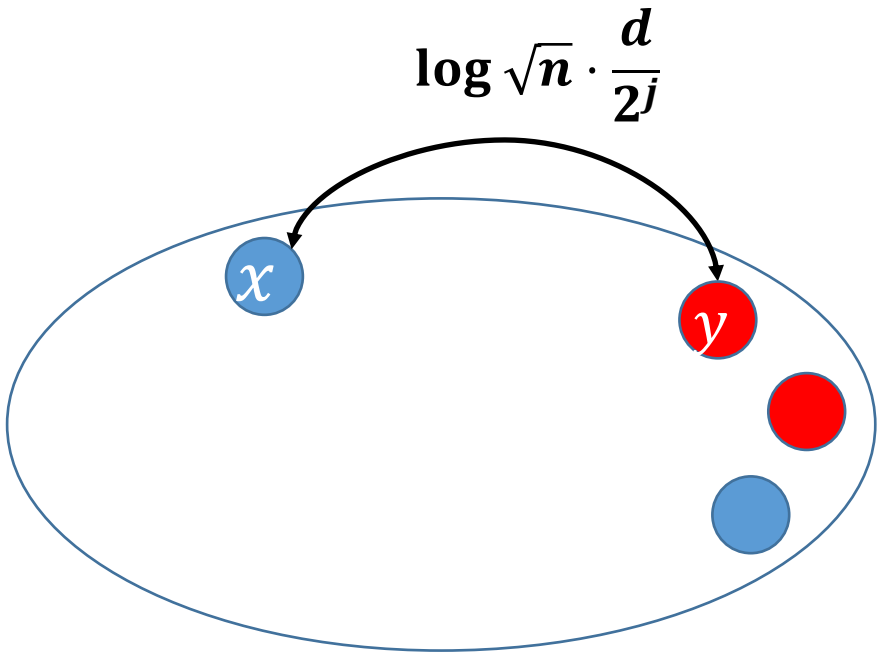
- $\widetilde{w}(e_j) \geq \log n \frac{d\Delta}{2^j}$ happens at most once
- $\widetilde{w}(e_j) \geq \frac{1}{2} \log n \frac{d\Delta}{2^j}$ at most twice

$\vdots$

- $\widetilde{w}(e_j) \geq \frac{1}{2^t} \log n \frac{d\Delta}{2^j}$ at most 2^t times

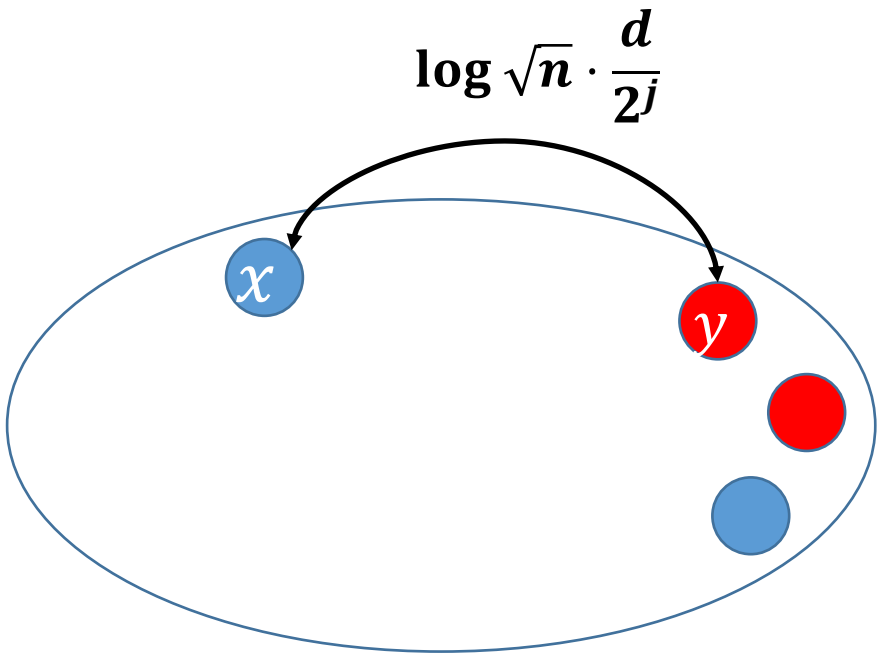
- Only $\log \log n$ levels!
- Implies expected distance blow-up of $\tilde{O}(\log n)$.

v a vertex with $\text{depth}(v) = j$



v a vertex with $\mathrm{depth}(v) = j$

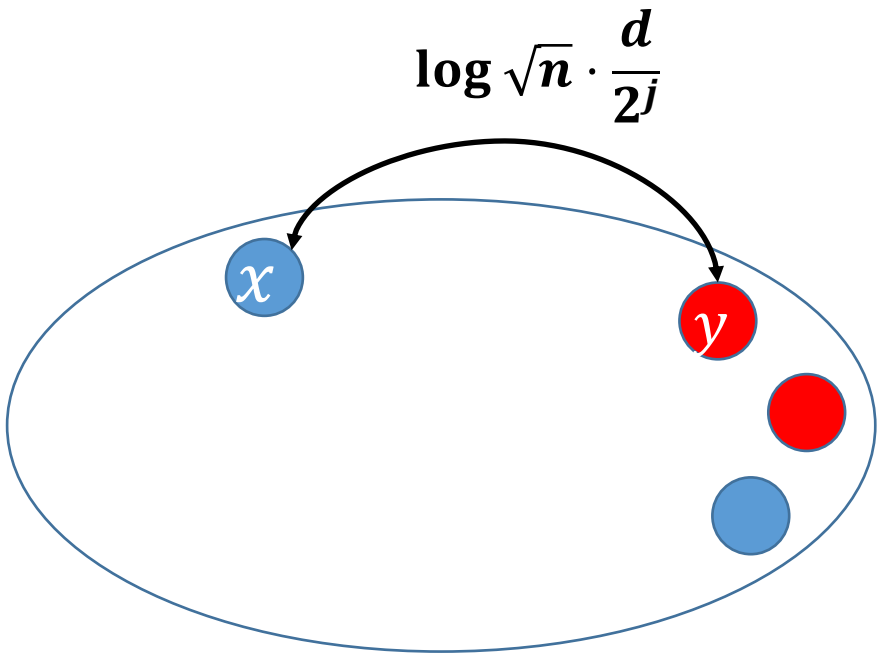
Fact: $\|x - y\|_1 = \alpha \frac{d}{2^j}$ ⟹ x, y collide with prob. $\frac{1}{2^\alpha}$



v a vertex with $\mathrm{depth}(v) = j$

Fact: $\|x - y\|_1 = \alpha \frac{d}{2^j}$ ⟹ x, y collide with prob. $\frac{1}{2^\alpha}$

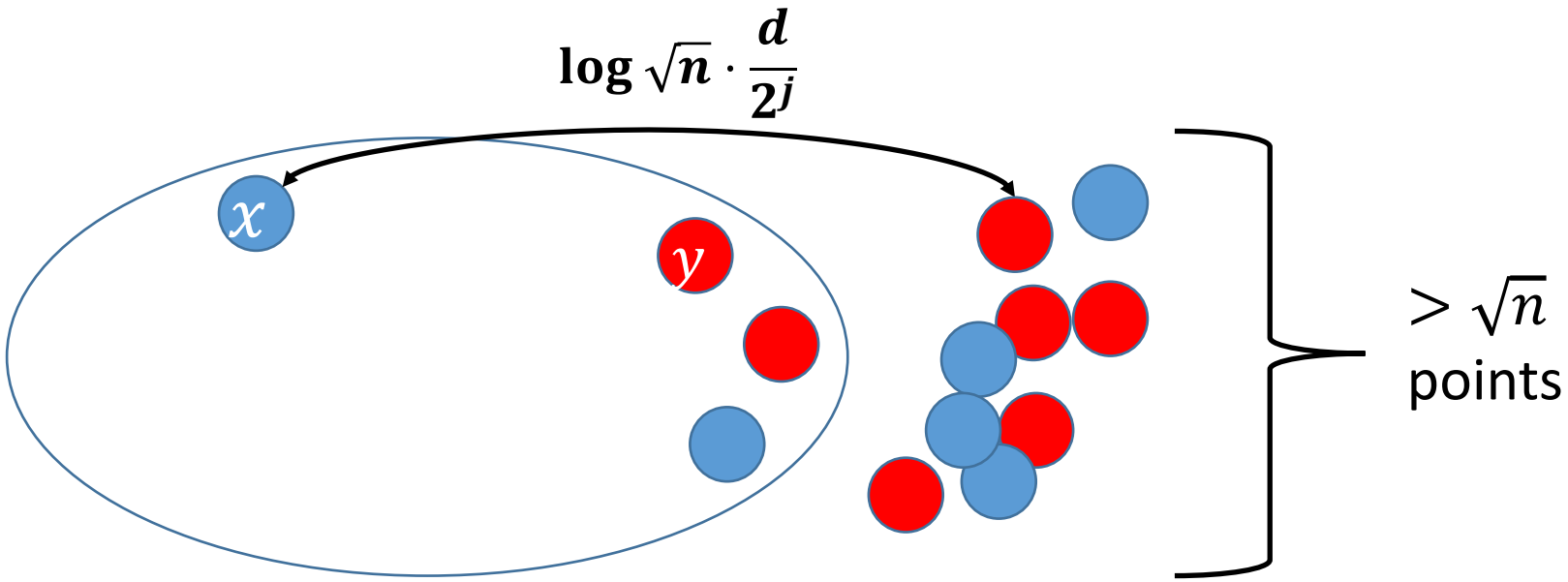
⟹ x, y collide with prob. $\frac{1}{\sqrt{n}}$



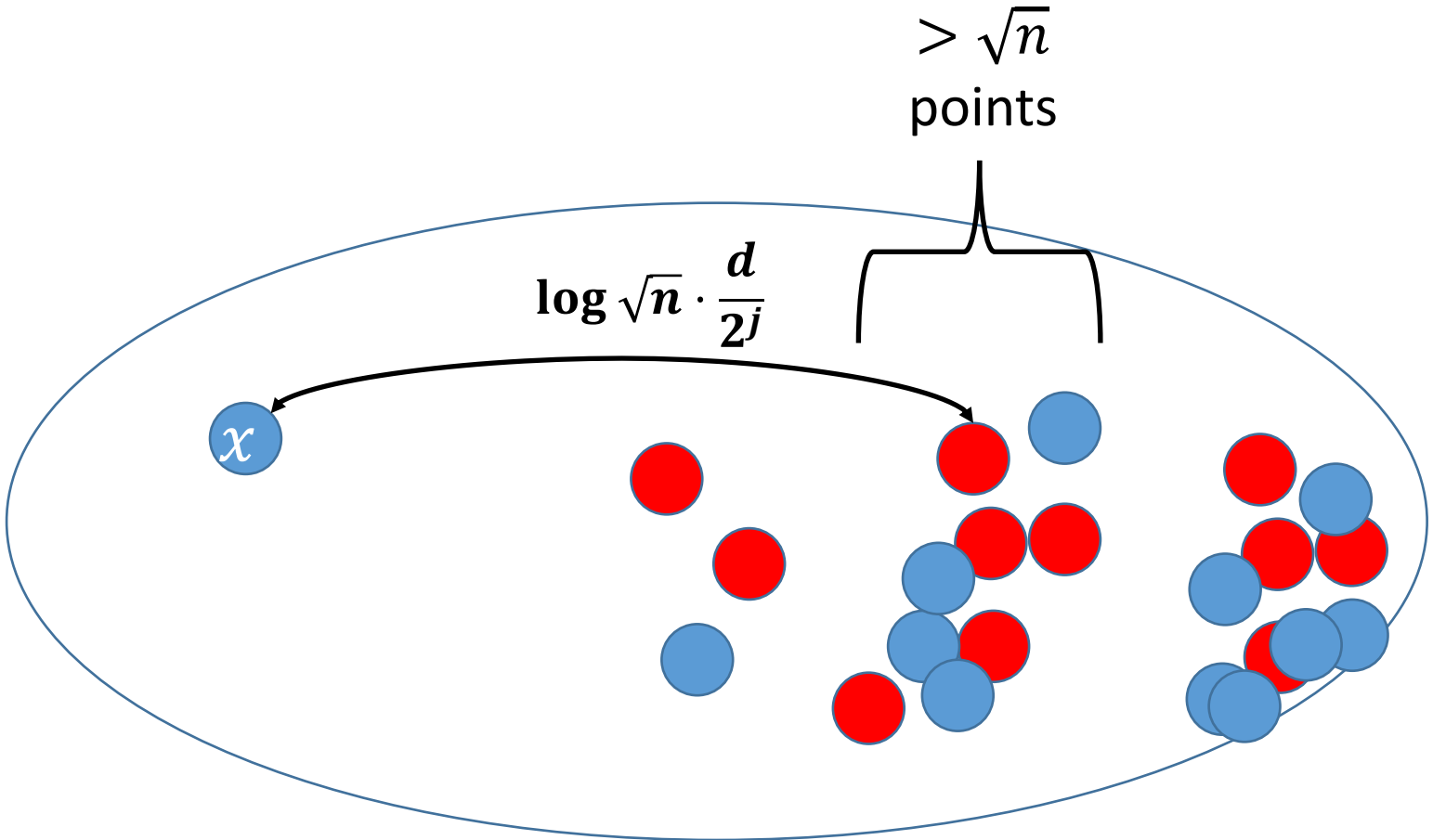
v a vertex with $\text{depth}(v) = j$

Fact: $\|x - y\|_1 = \alpha \frac{d}{2^j}$ ⟹ x, y collide with prob. $\frac{1}{2^\alpha}$

⟹ x, y collide with prob. $\frac{1}{\sqrt{n}}$

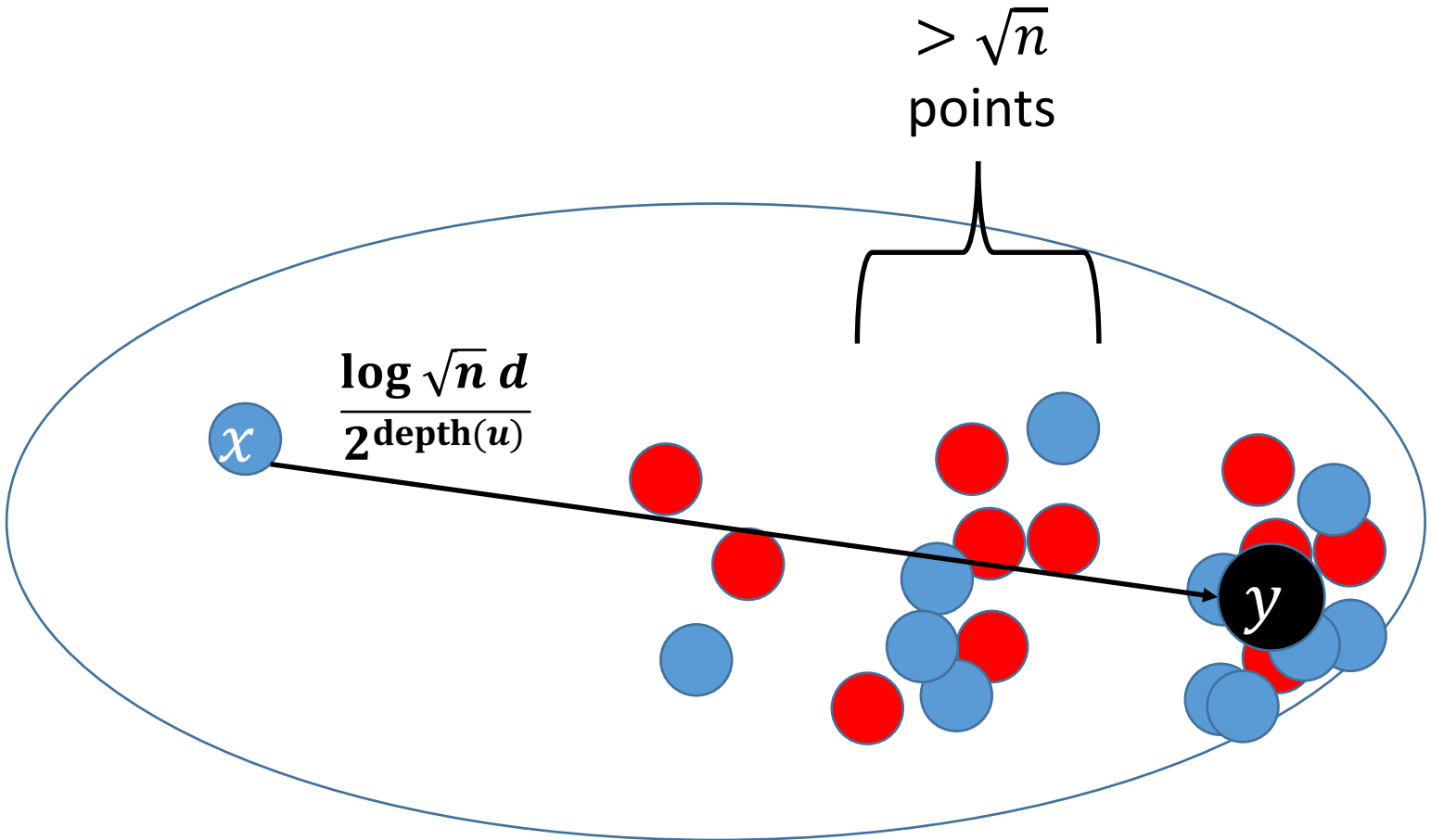


u a vertex with $\text{depth}(u) = j - \log\log n$



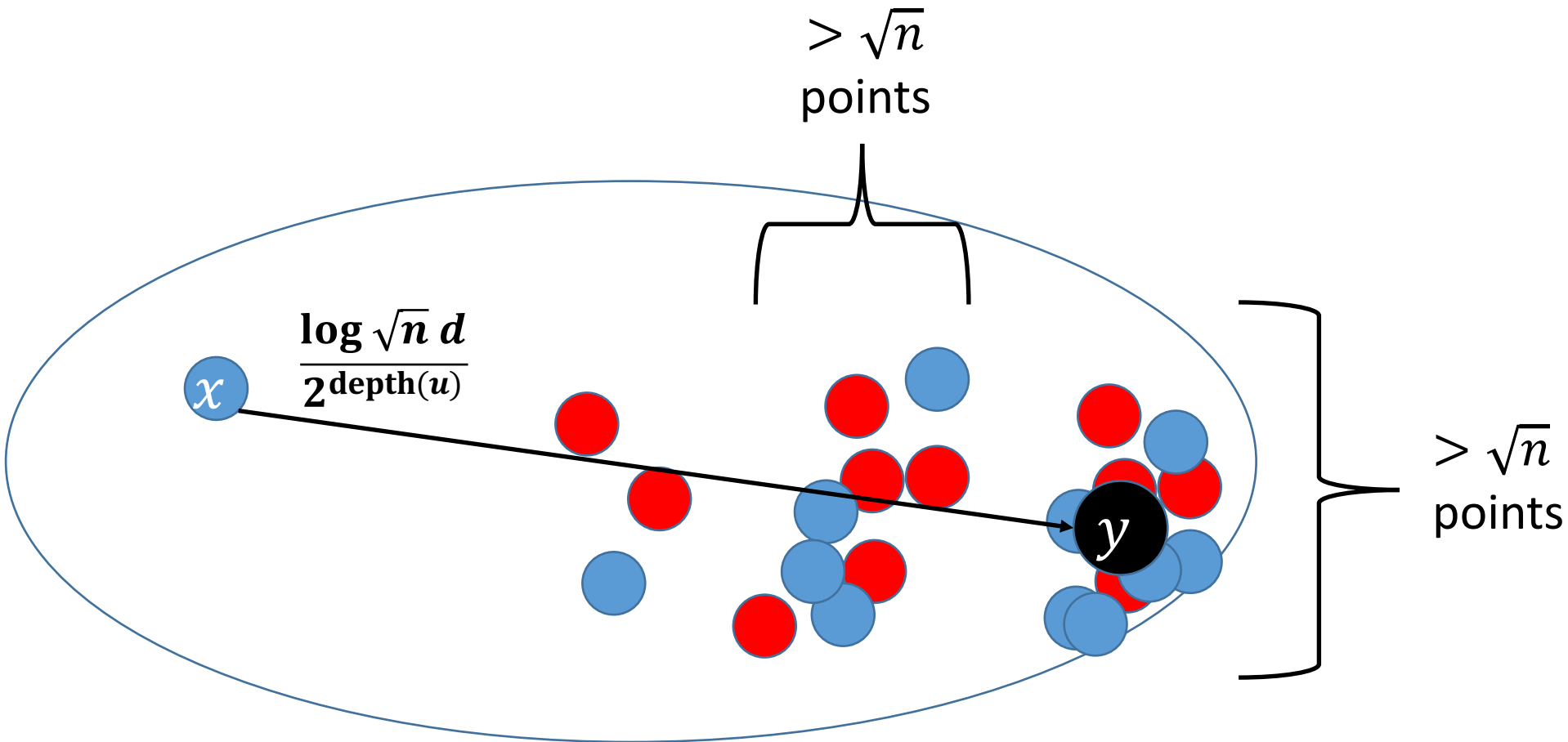
u a vertex with $\text{depth}(u) = j - \log\log n$

- If random point still far, must be $\sqrt{n}$ far points in u !



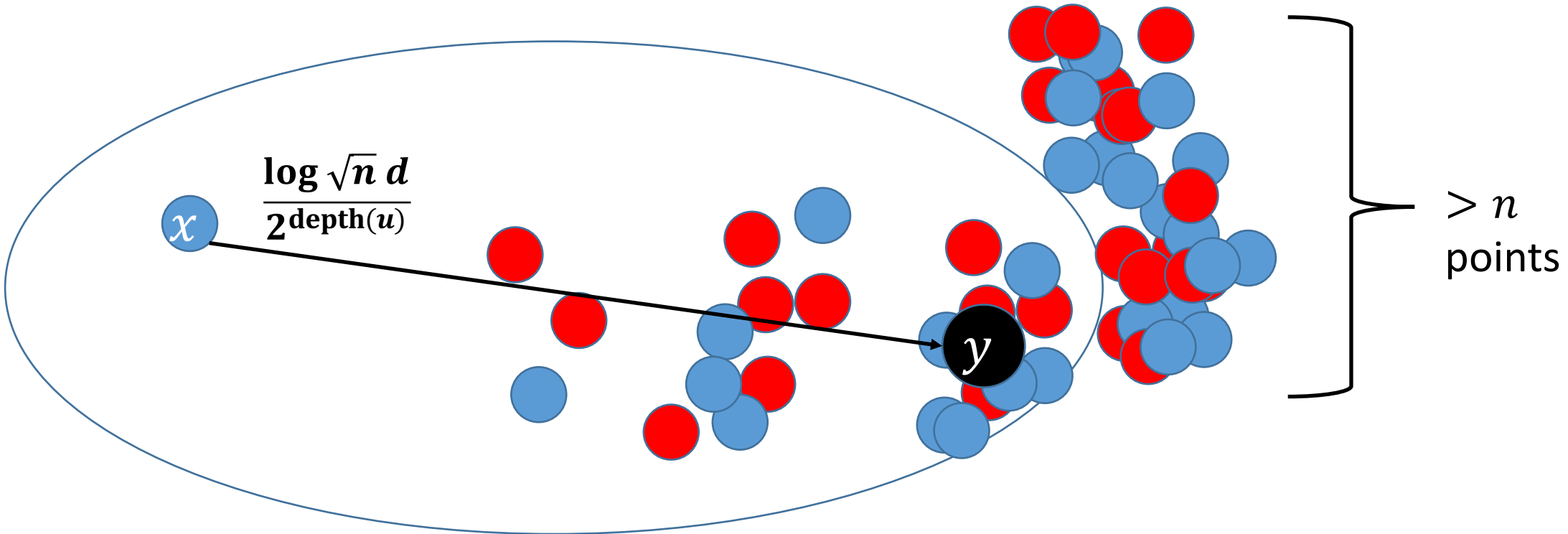
u a vertex with $\mathrm{depth}(u) = j - \log\log n$

- If random point still far, must be $\sqrt{n}$ far points in u !

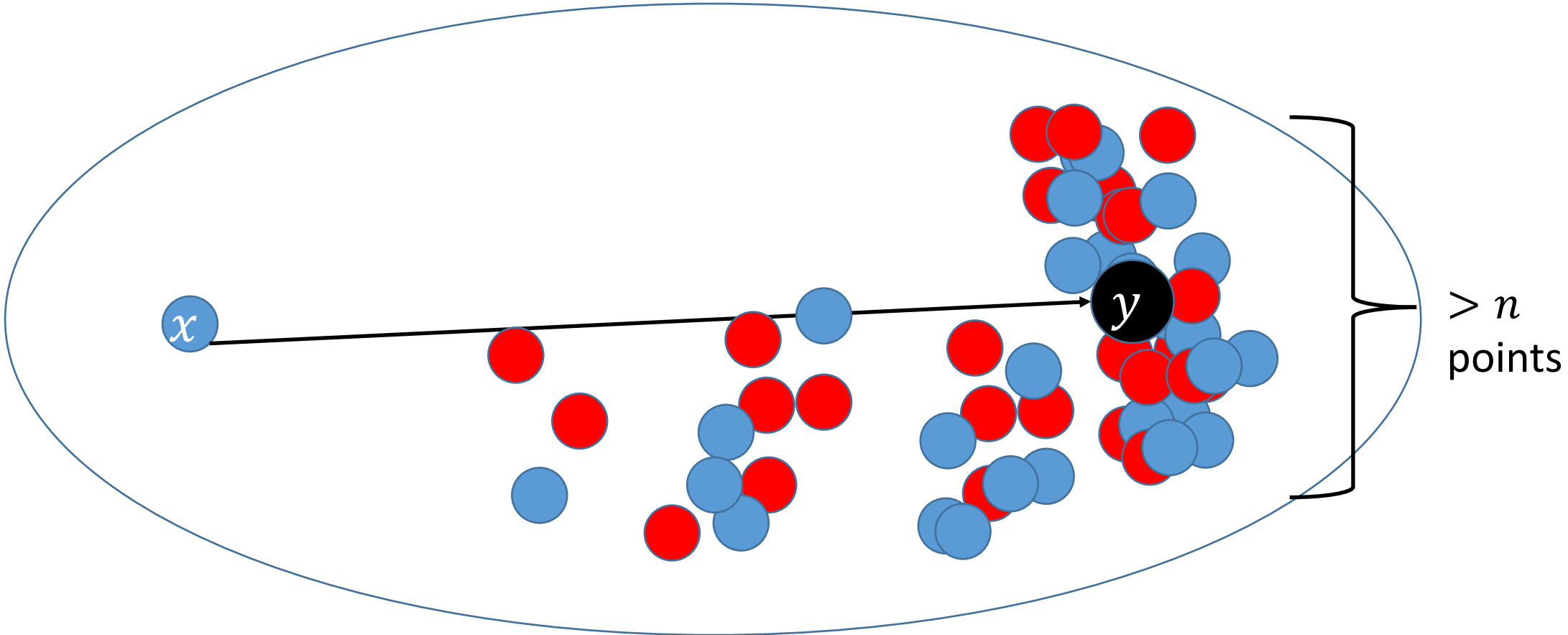


u a vertex with $\mathrm{depth}(u) = j - \log\log n$

- If random point still far, must be $\sqrt{n}$ far points in u !
- Each survives with probability $\frac{1}{\sqrt{n}}$



$$\text{depth}(w) = j - 2 \cdot \log\log n$$



Talk Outline

1. Quadtree Algorithm

2. New Data-Dependent Tree Embeddings

3. From Data-Dependent Quadtrees to Streaming Algorithms

Streaming Approach

Key Challenge: EMD over $\tilde{T}$ does not embed into ℓ_1

- Need to first look at data A, B to compute $\tilde{w}(e)$

Streaming Approach

Key Challenge: EMD over $\tilde{T}$ does not embed into ℓ_1

➢Need to first look at data A, B to compute $\tilde{w}(e)$

Two-Pass Solution: Importance Sampling

$$EMD_{\tilde{T}}(A,B) = \sum_{(u,v)\in\tilde{T}} \tilde{w}(u,v) \cdot ||A_v| - |B_v||$$

$$= \left(\sum_{(u,v)\in\tilde{T}} ||A_v| - |B_v|| \right) \cdot \mathbb{E}_{(u,v)\propto ||A_v|-|B_v||}[\tilde{w}(u,v)]$$

Streaming Approach

Key Challenge: EMD over $\tilde{T}$ does not embed into ℓ_1

➢Need to first look at data A, B to compute $\widetilde{w}(e)$

Two-Pass Solution: Importance Sampling

$$EMD_{\tilde{T}}(A,B) = \sum_{(u,v)\in\tilde{T}} \widetilde{w}(u,v) \cdot \big||A_v| - |B_v|\big|$$

$$= \left(\sum_{(u,v)\in\tilde{T}} \big||A_v| - |B_v|\big| \right) \cdot \mathbb{E}_{(u,v)\propto \big||A_v|-|B_v|\big|}[\widetilde{w}(u,v)]$$

- First Pass: sample (u,v)
- Second Pass: compute $\widetilde{w}(u,v)$

Lower Bounds + Open Problems

Lower Bounds	Paper
EMD (Space)*(Approx)$= \Omega(\log n)$	[Andoni, Indyk, Krauthgamer SODA '08]
MST (Space)*(Approx)$= \Omega(\log n)$	[This work]

Lower Bounds + Open Problems

Lower Bounds	Paper
EMD (Space)*(Approx)$= \Omega(\log n)$	[Andoni, Indyk, Krauthgamer SODA '08]
MST (Space)*(Approx)$= \Omega(\log n)$	[This work]

Do there exist (one/any-pass) $O(1)$-approximation algorithms for Geometric EMD/MST/Matching?

- Recently, [Czumaj, Jiang, Krauthgamer, Vesely, Yang] gave two-pass $O(1)$-approximation for facility location in $\mathrm{poly}(d \log \Delta)$ space.
- Lower bounds do not even rule out $O(1)$-approx in one-pass and $O(\log n)$ space.

Thank You!