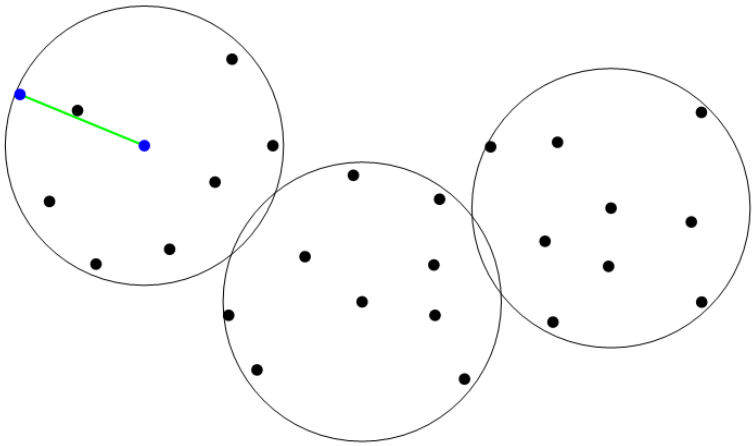
Net and Prune: A Linear Time Algorithm for Euclidean Distance Problems

Sariel Har-Peled, Benjamin Raichel

سپیده آقاملائی

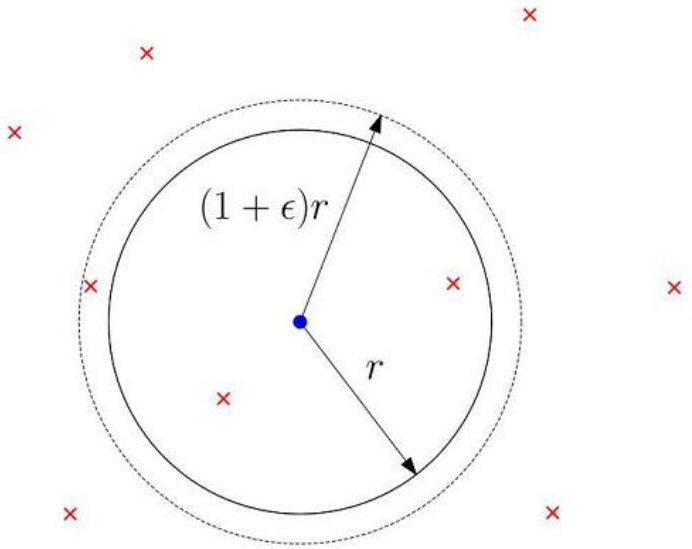
مقدمه

- بهینه‌سازی هندسی:
  - یک تابع بر حسب فاصله نقاط
  - تصمیم‌گیری در مورد مینیمم آن تابع
  - مثال:
    - نزدیک‌ترین زوج نقاط: مینیمم فاصله زوج نقاط
    - قطر: مینیمم ۱/ فاصله زوج نقاط
    - شعاع خوشه‌بندی: k نقطه وجود داشته باشند که فاصله هر نقطه به نزدیک‌ترین آنها حداکثر r باشد.



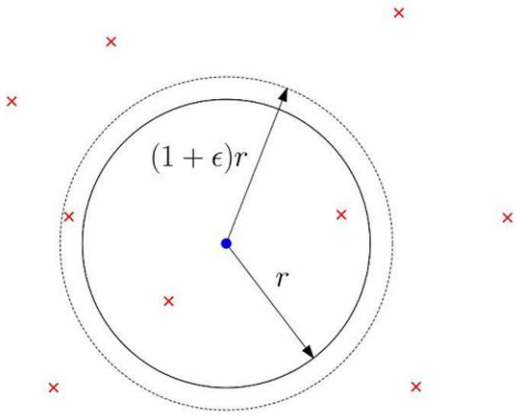
مسأله تصمیم‌گیری مینیمم تابع

- به ازای مقدار داده شده، آیا این مینیمم تابع است؟
  - جستجوی دودویی روی بازه مقادیر
    - مثلاً r در نزدیک‌ترین همسایه
      - بازه: صفر(شعاع بهینه) تا تقریب شعاع
  - ایراد:
    - دقت خوب = اپسیلون کم = زمان زیاد



جستجوی پارامتری

- گام اول: محاسبه غیرمستقیم کاندیداها
  - در جستجوی دودویی از داده‌ها استفاده کمی می‌کنیم.
    - بازه‌ی پیوسته مقادیر آنها



- داده‌های ما یک سری نقطه هستند (گسسته)
  - نمی‌توانیم آنها را مرتب کنیم که جستجوی دودویی روی نقاط انجام بدهیم...
    - (تابع: فاصله بقیه نقاط از یک نقطه)
  - اما بلدیم عنصر k-ام یک مجموعه را بدون مرتب کردن در زمان کمی به دست بیاوریم.

مثالی از روش prune & search (یک بعدی)

- پیدا کردن عنصر k-ام یک آرایه

Task: Return **k**th smallest number in $X = \{x_1, \ldots, x_n\}$.

Randomly select a value:

$$x_1, x_2, \ldots, x_i, \ldots, x_n$$

$\Uparrow$

Pivot on x_i:

$$\text{All Values} < x_i \quad | \quad x_i \quad | \quad \text{All Values} > x_i$$

$\Uparrow$

rank of x_i = size of this set $+1$

جواب مثال قبل!

Cases:

(1) If rank of $x_i = k$: Your done!

(2) If rank of $x_i > k$: Recurse on values $< x_i$

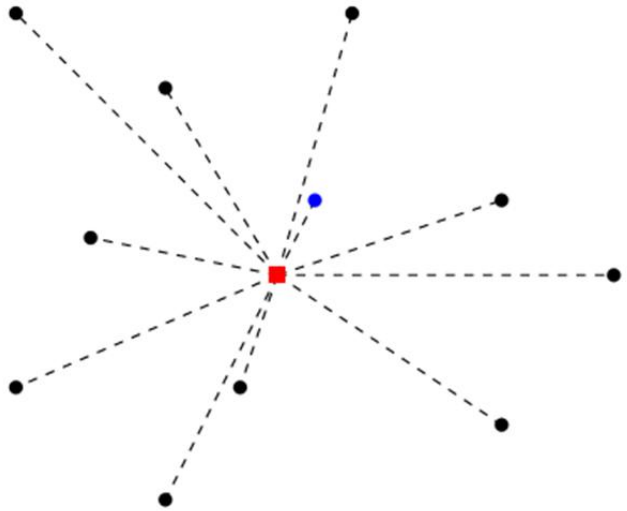
(3) If rank of $x_i < k$: Recurse on values $> x_i$

Linear in expectation

- x_i randomly sampled $\Rightarrow |< x_i| = |> x_i| = |X|/2$.
- $|X|$ decreases geometrically:
  $\Rightarrow$ Run time $= \sum_i |X|/2^i = O(|X|)$

محاسبه کاندیداها: نمونه‌گیری یک فاصله

- به ازای یک نقطه p، نقطه دیگری به صورت تصادفی از بین نقاط دیگر را انتخاب می‌کنیم و فاصله‌ی آن تا p را r می‌نامیم.
- نمونه گیری یکنواخت است.
- قابل محاسبه در زمان O(n) است.
- کاندیداهای مینیمم:
  - زوج نقاط با فاصله کمتر از r

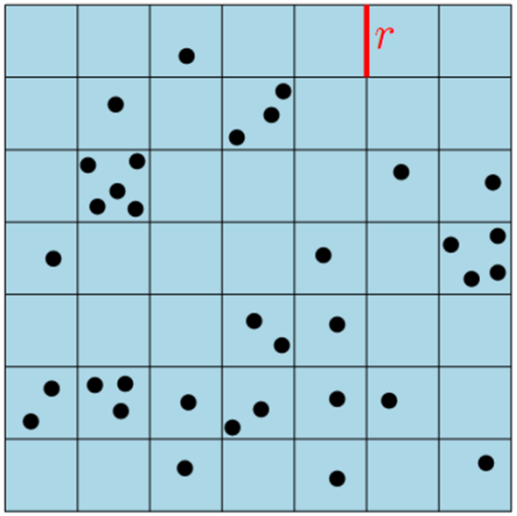


جستجوی پارامتری

- گام دوم: هرس پارامتری
  - حالت‌هایی که بیشتر/کمتر از پارامتر هستند، هرس می‌شوند.
  - مثال: در k-center یالها را به ترتیب وزن اضافه می‌کردیم...
- زمان چندجمله‌ای
- چیزی مثل quadtree خوب نیست چون نقاط با فاصله کمتر/بیشتر را نگه می‌دارد.
- اما ایده‌ی برداشتن نقاط به صورت سلسله مراتبی نقاط خوب است!
- اما ایده‌ی توری خوب است!

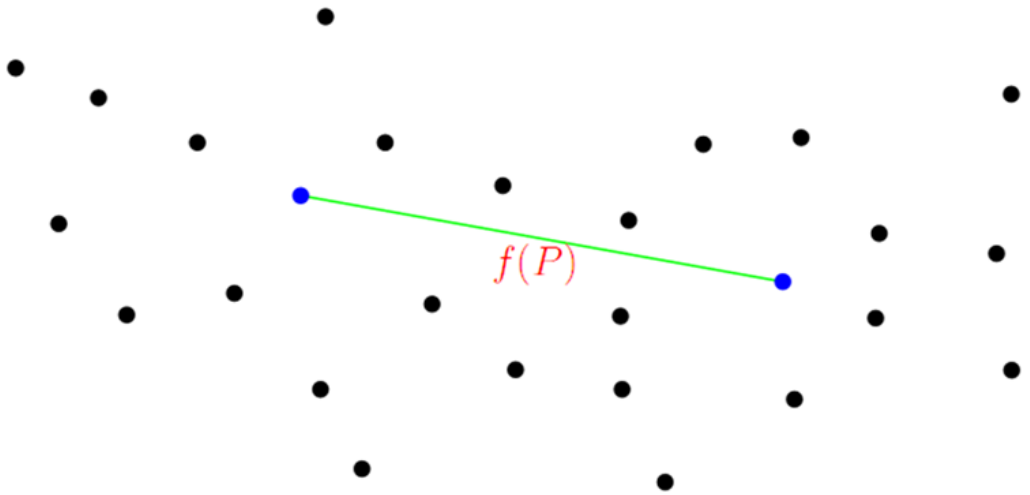
جستجوی پارامتری

- گام دوم: هرس پارامتری با زمان چندجمله‌ای
  - توری (grid)
    - خانه‌های با یک نقطه قطعاً مینیمم فاصله نیستند و حذف می‌شوند.
    - ایراد: اندازه‌ی خانه‌های توری چقدر باشد؟



تصمیم گیری

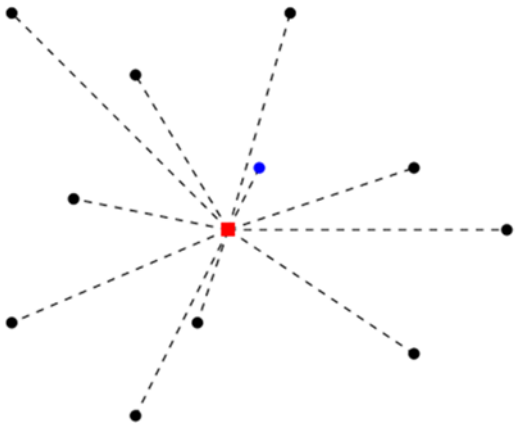
- تصمیم گیری: آیا به ازای r داده شده و مجموعه نقاط P، فاصله هر دو نقطه f(P) از r کمتر یا بیشتر یا مساوی است؟

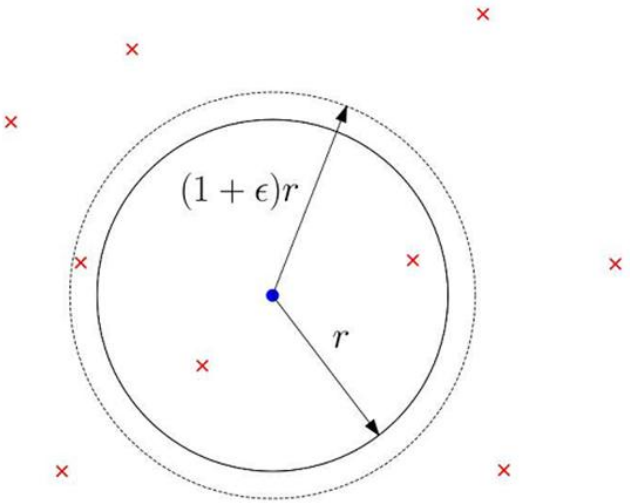


Linear time decider **D(r)** s.t. for $\mathbf{r} \in \mathbb{R}^+$, **D(r)** returns:
(1) $\mathbf{r} \approx \mathbf{f(P)}$ (2) $\mathbf{r} < \mathbf{f(P)}$ (3) $\mathbf{r} > \mathbf{f(P)}$

حدس هوشمندانه!

- دو نقطه را برداریم و فاصله‌ی بین آنها را r قرار بدهیم و تصمیم‌گیری را حل کنیم.
- اگر تصادفی برداریم می‌شود به طور متوسط در O(|P|) به جواب رسید.
- راه بهتری هم وجود دارد؟
  - قبلاً ما از جواب تقریبی استفاده می‌کردیم...





Guess, Check, Zoom!

- r همان حدس روش قبلی و f(P) مجموعه‌ی همه‌ی فواصل دو به دوی نقاط است.

Case 1: $r \approx f(P)$

Bullseye! Congratulations your done!

Case 2: $r < f(P)$

Can't see forest for the trees.

ZOOM OUT!

Case 3: $r > f(P)$

Head is in Magellenic clouds.

ZOOM IN!

- Zoom: نقاط را به r-netها تقسیم می‌کنیم. چطور؟

پیدا کردن نقاط

- در پیدا کردن عنصر k–ام ما فقط مقدار داشتیم. اینجا نقاط هم هستند.
- راه حل: نقاطی که فاصله کمتر/بیشتر دارند دور بریزیم.

The algorithm we want

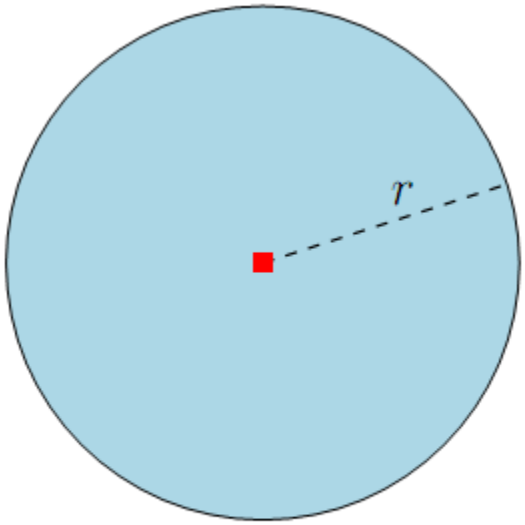
1) Sample a distance $\mathbf{r}$.
2) If $\mathbf{r} \approx \mathbf{f(P)} \rightarrow$ done.
3) If $\mathbf{r} < \mathbf{f(P)} \rightarrow$ throw out $\mathbf{1/2}$ of points by zooming out.
4) If $\mathbf{r} > \mathbf{f(P)} \rightarrow$ throw out $\mathbf{1/2}$ of points by zooming in.

Running time:

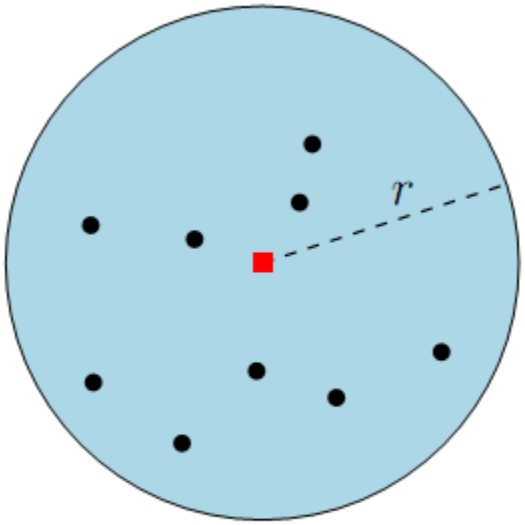
- In ith round $|P_i| \leq |P| / 2^i$
- Total run time: $\sum_i |P_i| \leq \sum_i |P| / 2^i = O(|P|)$

نقاط تنها و نقاط محبوب

- تنها: همسایه‌ای در فاصله‌ی r از خودشان ندارند.
- محبوب: نزدیک‌ترین همسایه‌شان فاصله‌ی کمتر از r دارد.



Lonely



Friendly

مرحله zoom

In Expectation

r randomly sampled $\Rightarrow$ $\mathbf{E[|Lonely|] = E[|Friendly|] = |P|/2}$.

Zooming out:

- Throw out most of **Friendly**.

Zooming in:

- Throw out most of **Lonely**.

Recall

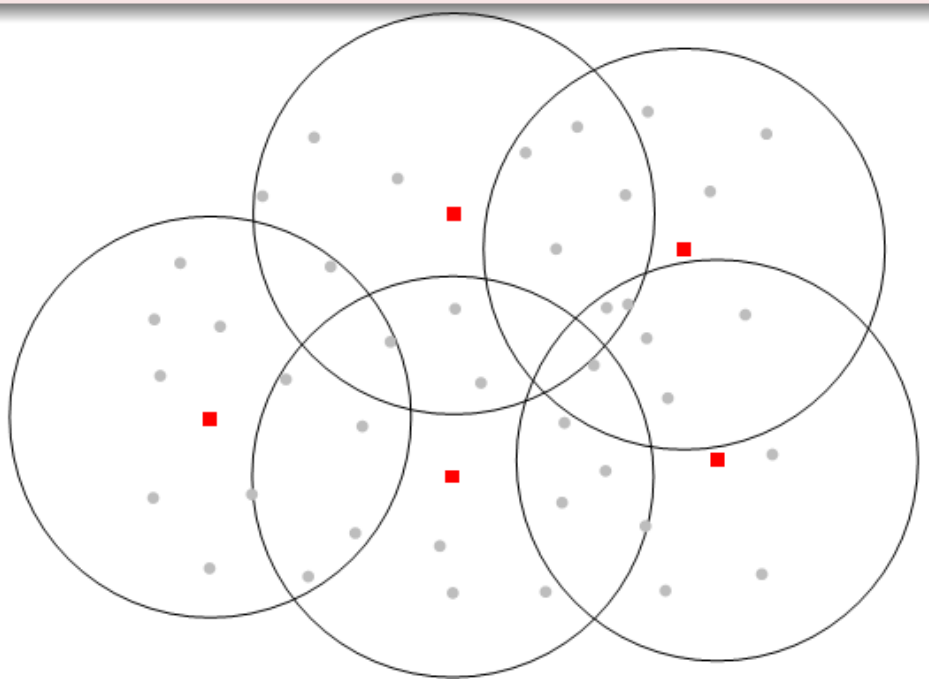
Throw out constant fraction in $\mathbf{O(|P|)}$ time
$\Rightarrow$ geometric series $\Rightarrow$ linear run time.

نقاط کاندیدای مرحله بعد

r-net

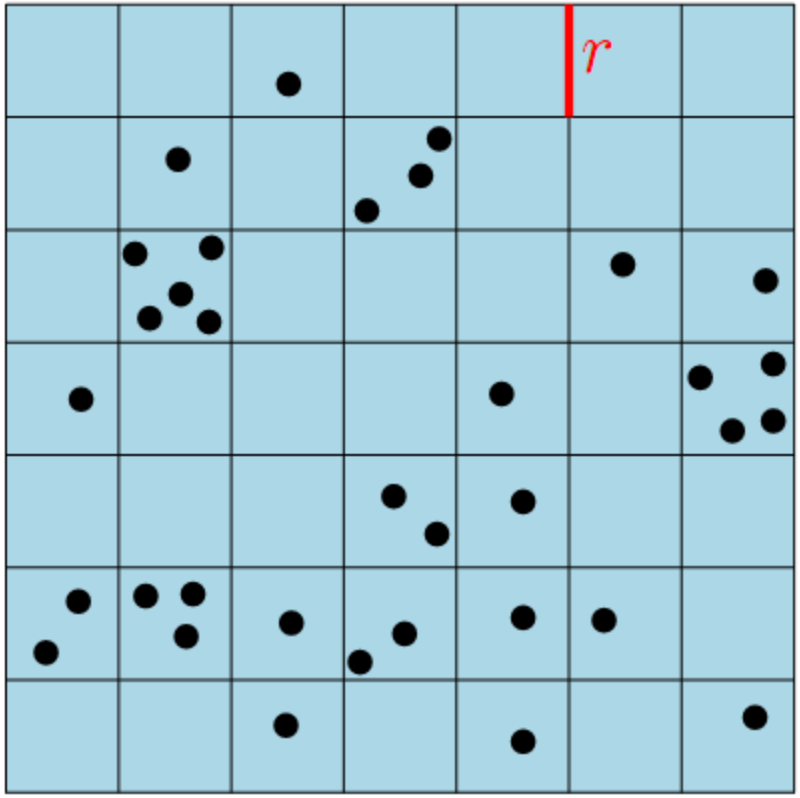
$N \subseteq P$ is an **r**-net if

- Balls of radius **r** centered at **N** cover the point set.
- For any two $p, q \in N$, we have $d(p, q) \geq r$.



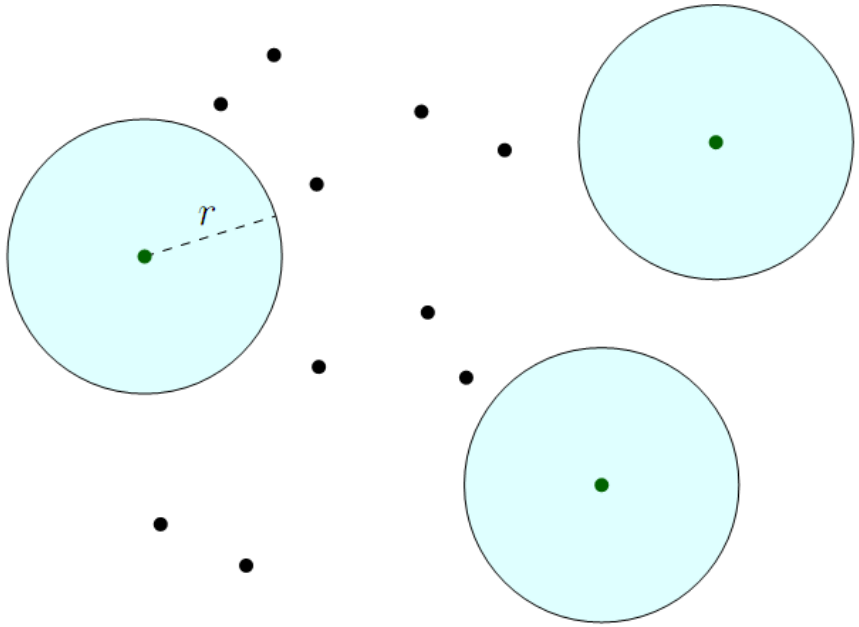
توری

- استفاده از توری به عنوان تصمیم‌گیر
  - خانه‌های خالی
  - خانه‌های با یک نقطه: نقاط تنها
  - محاسبه r-net



نزدیک شدن (zoom in)

- همه‌ی نقاط تنها را حذف می‌کنیم.

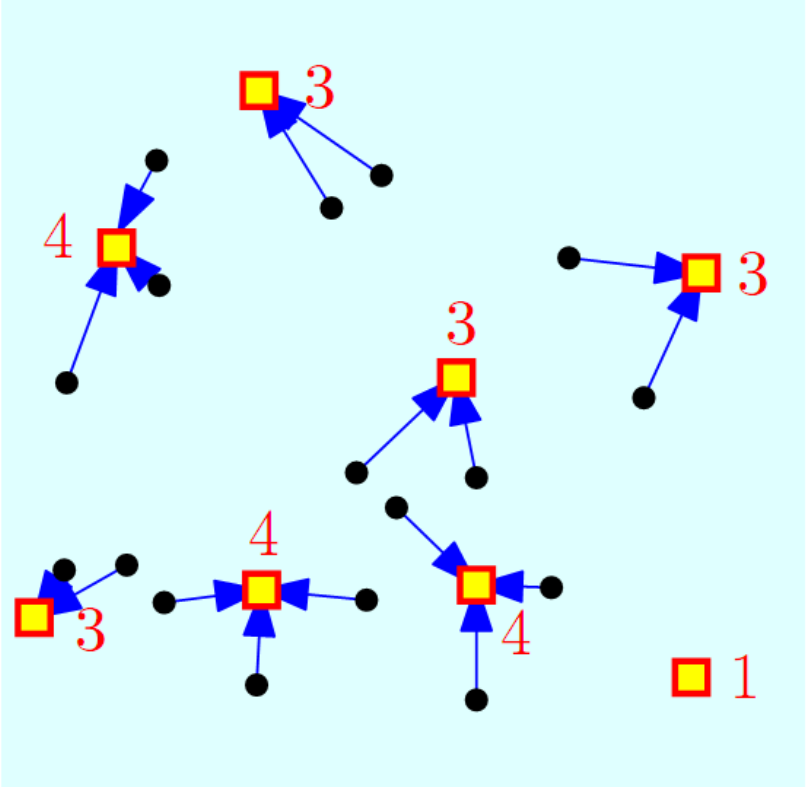


Lonely facts

- $\mathbf{E}[|\mathbf{Lonely}|] = |\mathbf{P}|/2$
- Can remove **Lonely** in $\mathbf{O}(|\mathbf{P}|)$ time.

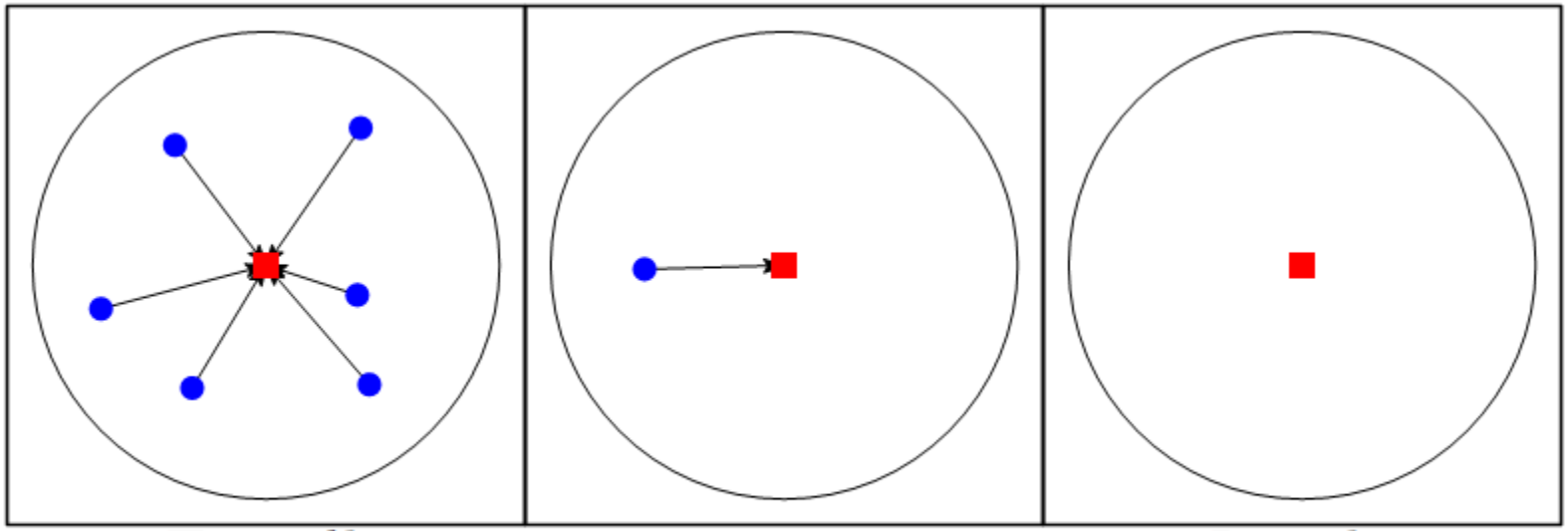
حذف نقاط با r-net

- فقط مرکزها را نگه می‌داریم. (مرکز = مربع زرد)



تحلیل

- حداقل نصف نقاط محبوب توسط r-net حذف می‌شوند.



اثبات

How much do r-nets throw away?

- Half of **Friendly** removed by **r**-net
- **r** randomly sampled $\Rightarrow$ $\mathbf{E[|Friendly|] = |P|/2}$
- $\Rightarrow$ $\mathbf{1/4}$ of **P** removed by **r**-net.

r-nets: Running Time

- **r**-net easily computable in $\mathbf{O(|P|)}$ time.
- $\Rightarrow$ $\mathbf{O(|P|)}$ time to throw out constant fraction.

تحلیل الگوریتم

The Technique in 3 steps

(1) Call **D(r)** on randomly sampled value from **NN(P)**.
(2) **r** too small: net points together.
(3) **r** too large: prune away isolated points.

Trash removal

How much do we remove:
(i) Net removes $|P|/4$
(ii) Prune removes $|P|/2$

Running time:

- In **i**th round $|P_i| \leq (3/4)^i |P|$
- Total run time: $\sum_i |P_i| \leq \sum_i (3/4)^i |P| = O(|P|)$

کارهای قبلی

Prune and Search:

- Randomized median selection
- Low dim LP: **[Megiddo, 1984]**
- Searching sorted matrices: **[Frederickson and Johnson, 1984]**
- many others (see paper)

Grids and nets:

- $\mathcal{CP}$, Grids, randomization: **[Rabin, 1976]**
- **k**-center, grids: **[Har-Peled, 2004]**
- nets: **[Krauthgamer and Lee, 2004]** **[Har-Peled and Mendel, 2006]**
- many others (see paper)

روش ترکیبی (Net & Prune)

- روش هرس پارامتری و توری روی حالت‌های محدود شده‌ی مسأله جواب می‌دهند.
  - تصمیم‌گیری: توری
  - K-center: هرس پارامتری
- روش گفته شده الگوریتم تقریبی می‌دهد که با احتمال بالا جواب درست می‌دهد.
- کاربردها: k-مرکز، k-امین سنگین‌ترین یال MST، k-امین نزدیک‌ترین همسایه، کوچکترین توپ شامل k نقطه، ...

سوال؟

- Net & Prune!