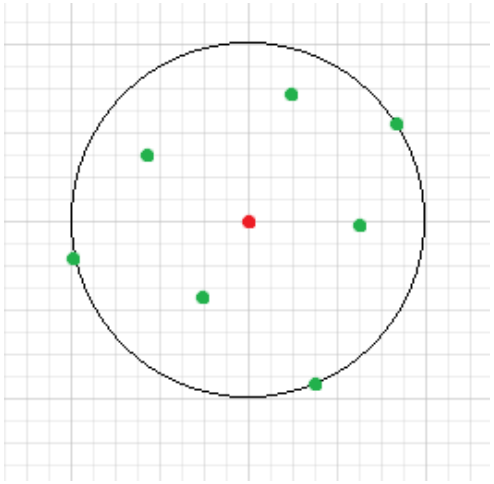
Net & Prune

الگوریتم خطی برای مسائل فاصله اقلیدسی

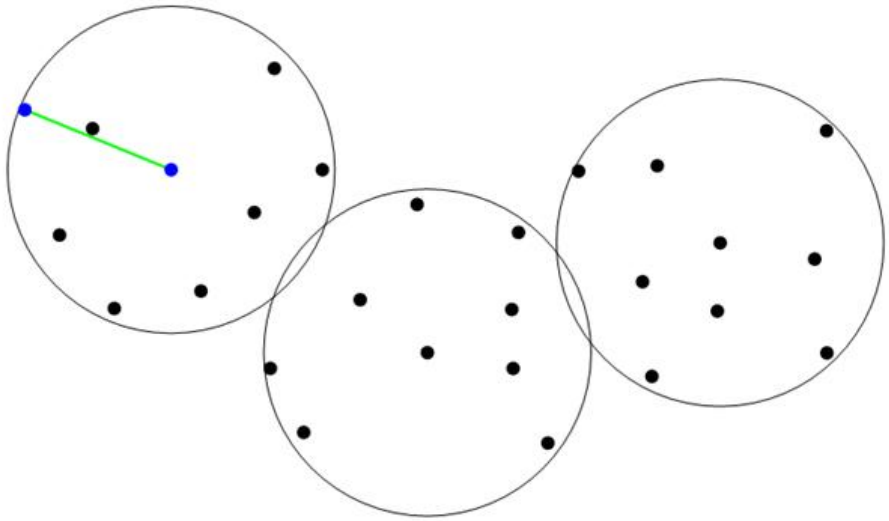
Sariel Har-Peled, Benjamin Raichel

سپیده آقاملائی

مسائل فاصله اقلیدسی

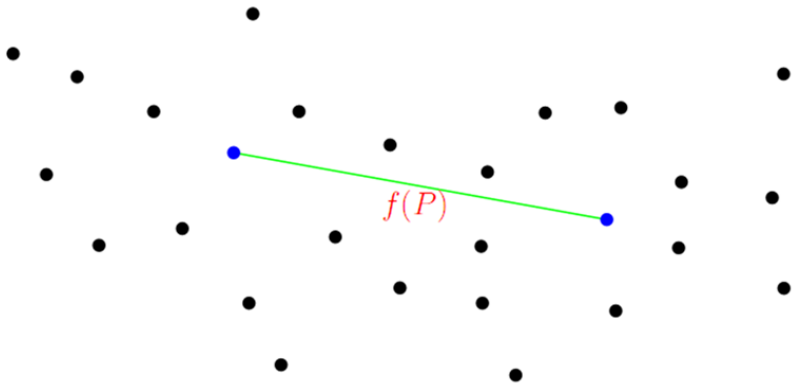


- قطر: دورترین نقطه نسبت به یک نقطه
  - $\max_{p,q \in P} ||p - q||$
  - الگوریتم تقریبی: دورترین نقطه از یک نقطه خاص
- k-مرکز
  - $\max_{p \in P} \min_{c} ||p - c||$



مسائل بهینه‌سازی فاصله

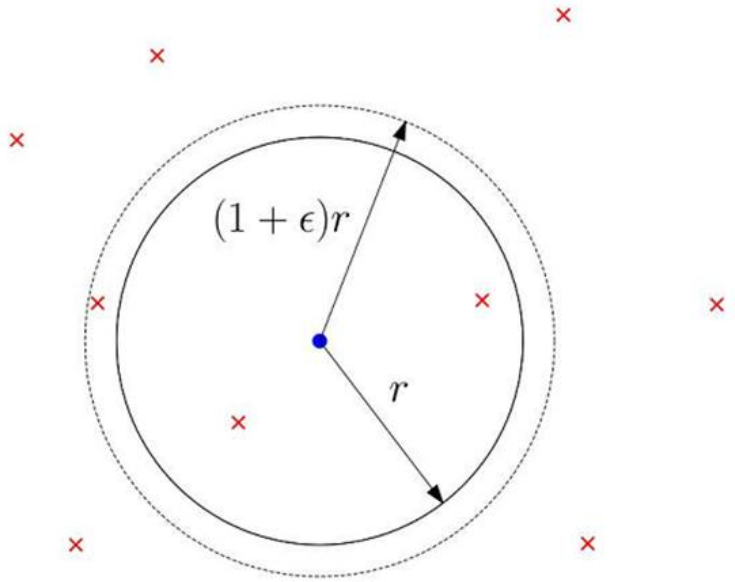
- تابعی از نقاط به فاصله‌ی نقاط
- $f: P \rightarrow \binom{P}{2}$
- تصمیم‌گیری در مورد مینیمم آن تابع
- آیا به ازای r داده شده و مجموعه نقاط P، فاصله هر دو نقطه f(P) از r کمتر یا بیشتر یا مساوی است؟



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) $\mathbf{r} > \mathbf{f(P)}$

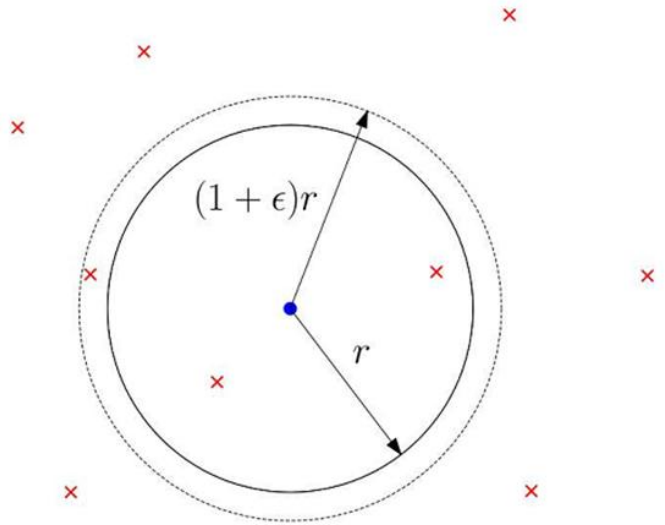
روش ۱: جستجوی دودویی روی بازه مقادیر

- مثلاً r در نزدیک‌ترین همسایه
  - بازه: صفر(شعاع بهینه) تا تقریب شعاع
- ایراد:
  - اپسیلون کم = زمان زیاد
- بهبود: گسسته بودن بازه‌ی واقعی
  - فاصله هر نقطه تا مرکز



روش ۲: جستجوی پارامتری

- گام اول: محاسبه غیرمستقیم کاندیداها (حدس پارامتر)
- مجموعه کاندیدا: فاصله‌ی همه‌ی نقاط از یک نقطه
  - مرتب‌سازی بر اساس فاصله از مرکز و جستجوی دودویی
    - شاید تبدیل تقریبی به دقیق زمان کمتری ببرد.
  - پیدا کردن عنصر k-ام بدون مرتب‌سازی



مثالی از روش prune & search (یک بعدی)

- پیدا کردن عنصر k-ام یک آرایه

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

Randomly select a value:

$$\boxed{x_1, x_2, \ldots, x_i, \ldots, x_n}$$

$$\Uparrow$$

Pivot on x_i:

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

$$\Uparrow$$

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

جواب مثال قبل!

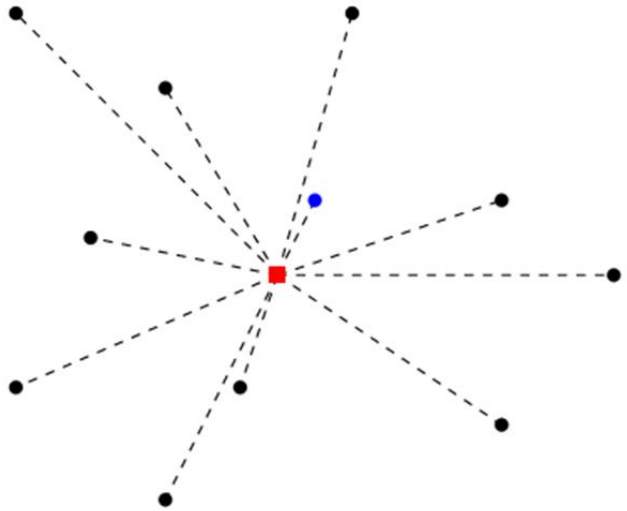
Cases:
(1) If rank of $x_i = k$: Your done!
(2) If rank of $x_i > k$: Recurse on values $< x_i$
(3) If rank of $x_i < k$: Recurse on values $> x_i$

Linear in expectation

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

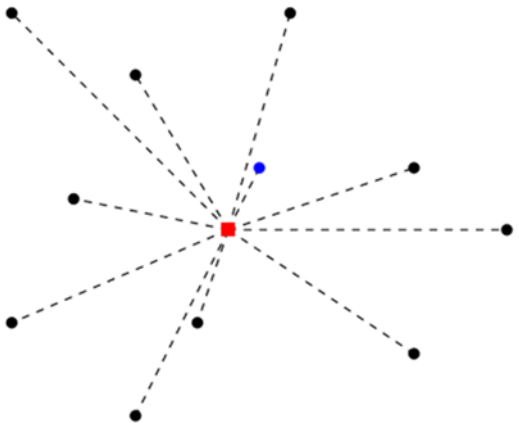
محاسبه کاندیداها: نمونه‌گیری یک فاصله (۲بعدی)

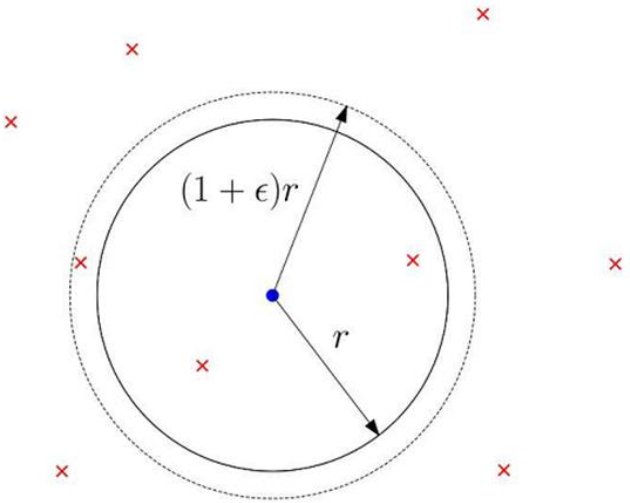
- به ازای یک نقطه p، نقطه دیگری به صورت تصادفی از بین نقاط دیگر را انتخاب می‌کنیم و فاصله‌ی آن تا p را r می‌نامیم.
- نمونه گیری **یکنواخت** است.
- قابل محاسبه در زمان O(n) است.
- کاندیداهای مینیمم:
  - زوج نقاط با فاصله کمتر از r



جستجوی پارامتری: حدس پارامتر r

- 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...



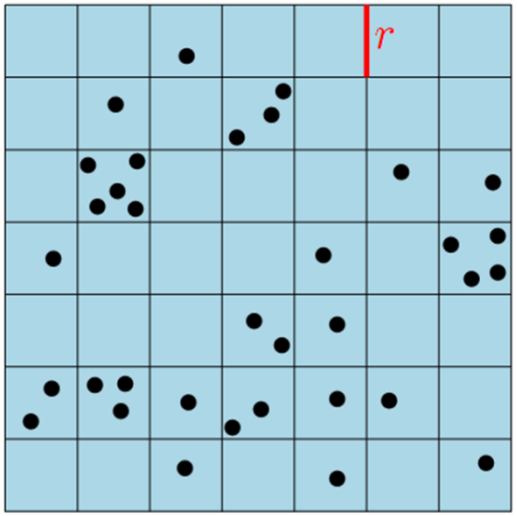


جستجوی پارامتری

- 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 نقاط کم نسبت به خانه‌های با نقطه زیاد
    - ایراد: اندازه‌ی خانه‌های توری چقدر باشد؟



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، ...

Guess, Check, Zoom!

- r همان حدس تصادفی 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!

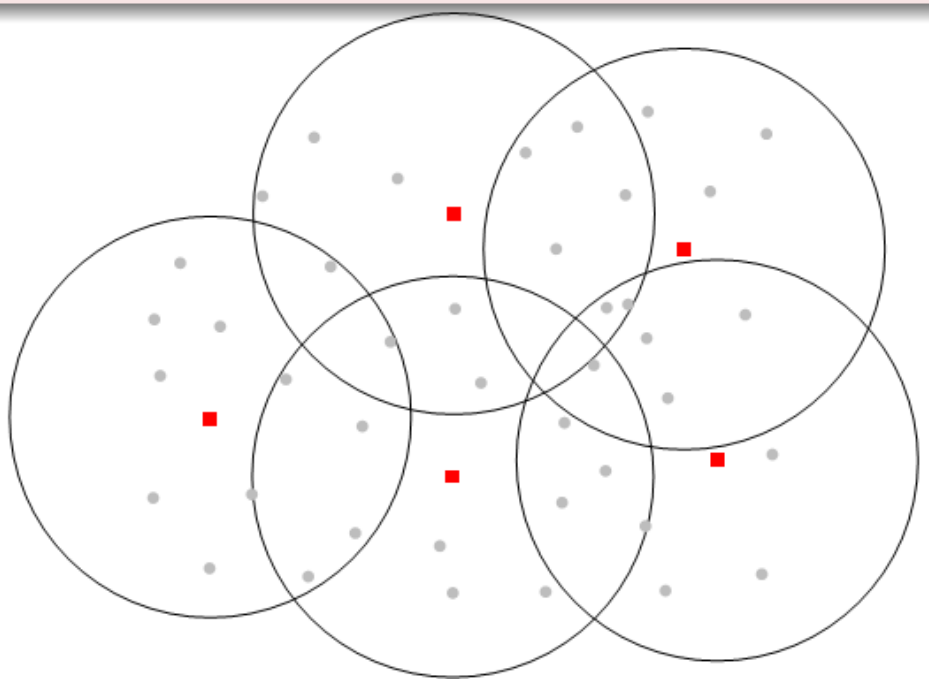
- Zoom: نقاط را به r-netها تقسیم می‌کنیم. چطور؟

r-net

r-net

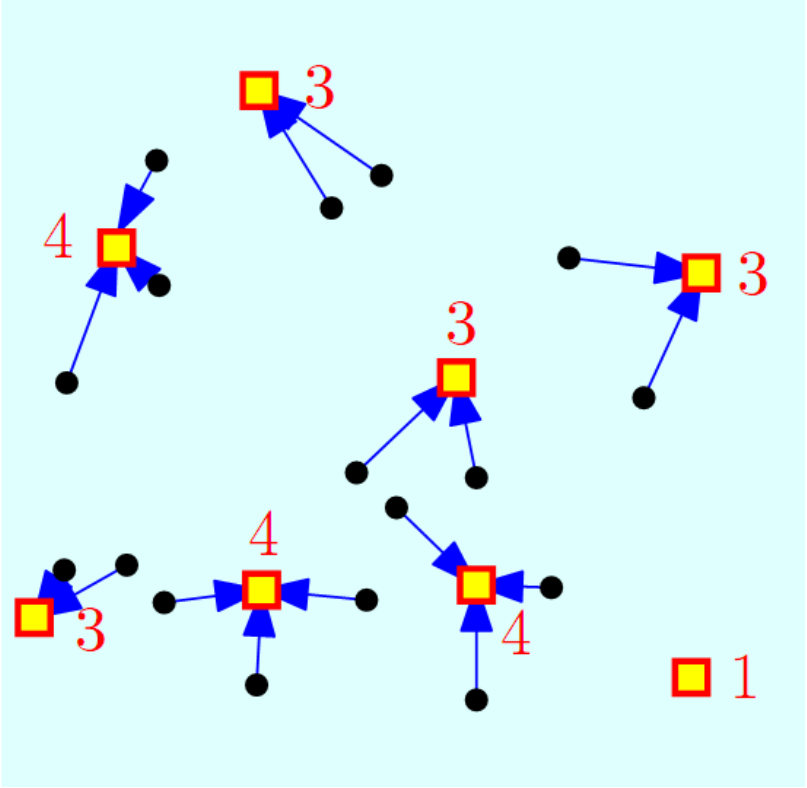
$N \subseteq P$ is an **r**-net if

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



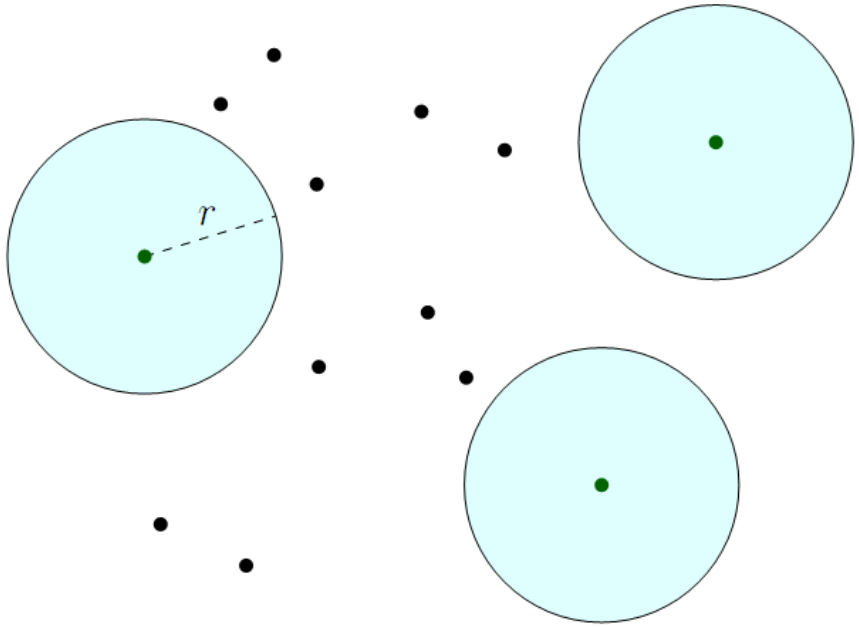
حذف نقاط با r-net (Zoom Out!)

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



نزدیک شدن (zoom in)

- همه‌ی نقاط تنها را حذف می‌کنیم.

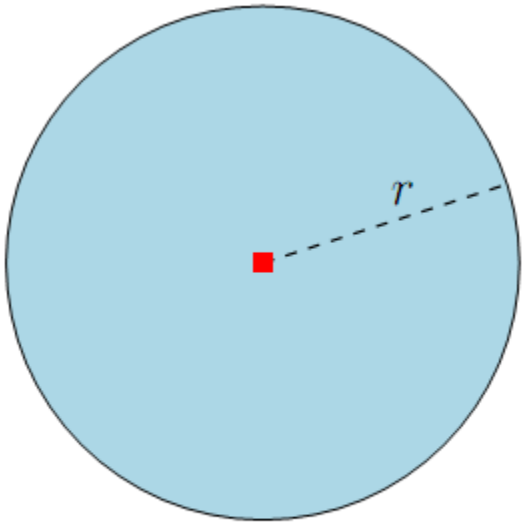


Lonely facts

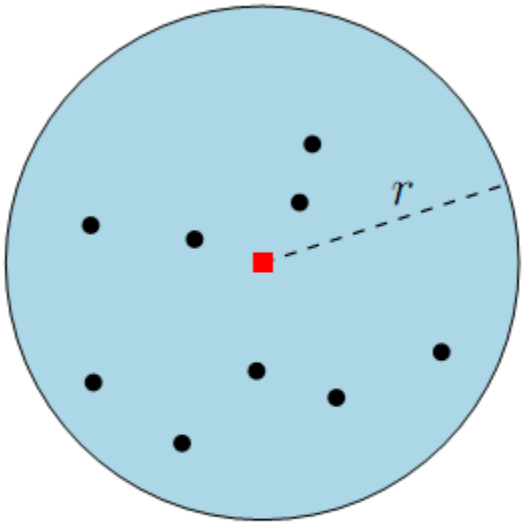
- $\mathbf{E}[|\mathbf{Lonely}|] = |P|/2$
- Can remove **Lonely** in $O(|P|)$ time.

نقاط تنها و نقاط محبوب

- به طور متوسط نصف نقاط تنها و نصف دیگر محبوب هستند.
  - تنها: همسایه‌ای در فاصله‌ی r از خودشان ندارند.
  - محبوب: نزدیک‌ترین همسایه‌شان فاصله‌ی کمتر از r دارد.



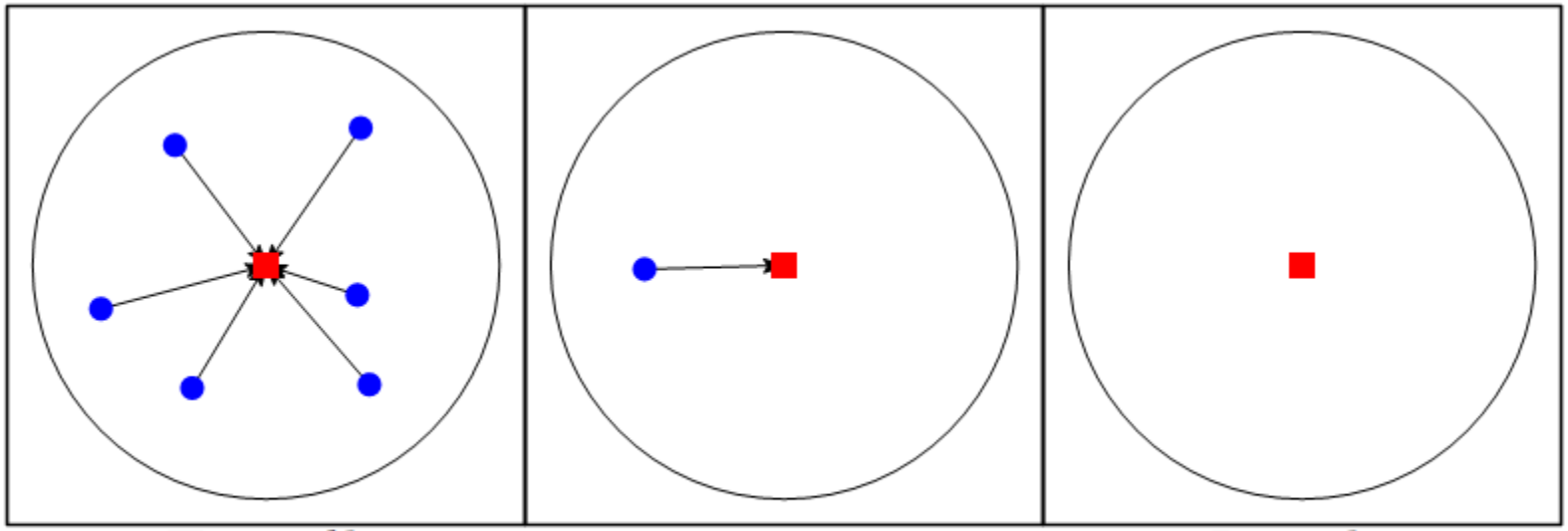
Lonely



Friendly

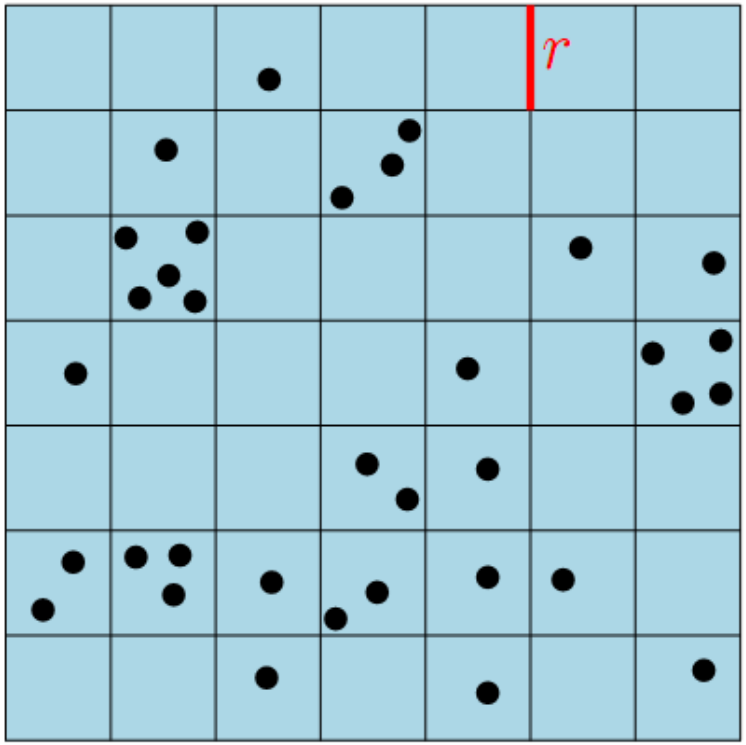
تحلیل

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



توری

- استفاده از توری به عنوان تصمیم‌گیر: hashing
  - خانه‌های با یک نقطه بدون همسایه: نقاط تنها
  - محاسبه r-net :k-center



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

- جستجو و هرس (prune & search)
  - پیدا کردن تصادفی میانه
  - LP در ابعاد پایین (Megiddo ‘84)
  - جستجوی ماتریس‌های مرتب شده (Frederickson, Jonson, ‘84)
  - …
- توری و netها:
  - نزدیک‌ترین نقاط (توری، تصادفی) (Rabin ‘76)
  - k–مرکز (توری) (Har-Peled 2004)
  - netها (Krauthgamer,Lee 2004), (Har-Peled, Mendel 2006)
  - …

نتیجه‌گیری

- مزیت‌ها:
  - برای نقاط وزن‌دار جواب می‌دهد.
  - زمان را از $O(n \log n)$ که با استفاده از تصمیم‌گیرهای بازه پیوسته به دست می‌آید به خطی کاهش می‌دهد.

سوال؟

- Net & Prune!