Approximate Nearest Neighbors via Point Location Among Balls

Method of Har-Peled

(improved version from notes)

- Reduce $(1+\varepsilon)$-ANN query on n points to point location in equal balls (PLEB) queries
  - Preprocessing space $O(\frac{n}{\varepsilon}\log\frac{t\,n}{\varepsilon})$
  - Preprocessing time $O(\log\frac{n}{\varepsilon})$
  - Query time $O(\log\frac{n}{\varepsilon})$

Notation

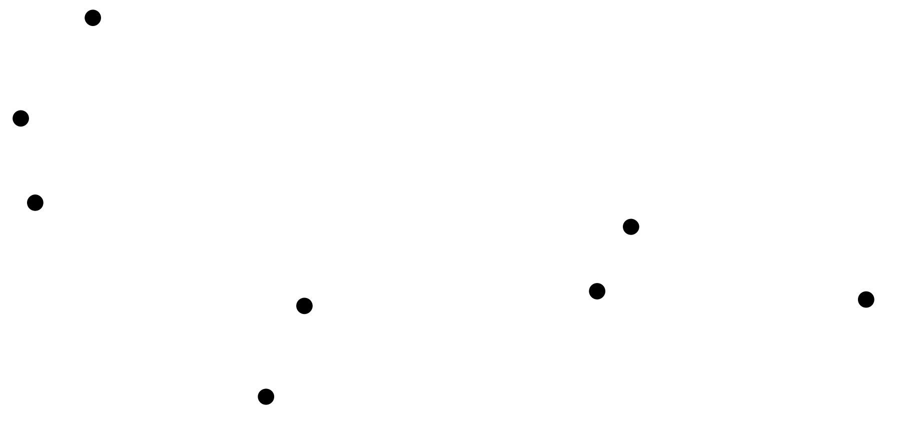
$d_P(q)$	Distance from point q to nearest neighbor point in set P
$U_{balls}(P, r)$	Union of balls of radius r about points in P
$NNbr(P, r)$	"Nearest Neighbor" data structure Returns TRUE and a witness point if query point q is in $U_{balls}(P, r)$ and FALSE otherwise
$\hat{I}(P, r, R, \varepsilon)$	"Interval Nearest Neighbor" data structure for points in set P, over range [r, R], with approximation error ε Indicates if $d_P(q)$ is outside range [r, R] or returns the ball centered at the point $(1+\varepsilon)$-ANN to q

Reduction from ANN to PLEBs

- Build a tree *D*
  - Each node v has an interval NNbr data structure $\widehat{I}_v$
  - Use $\widehat{I}_v$ to decide how to traverse the tree when search reaches node v

Constructing D

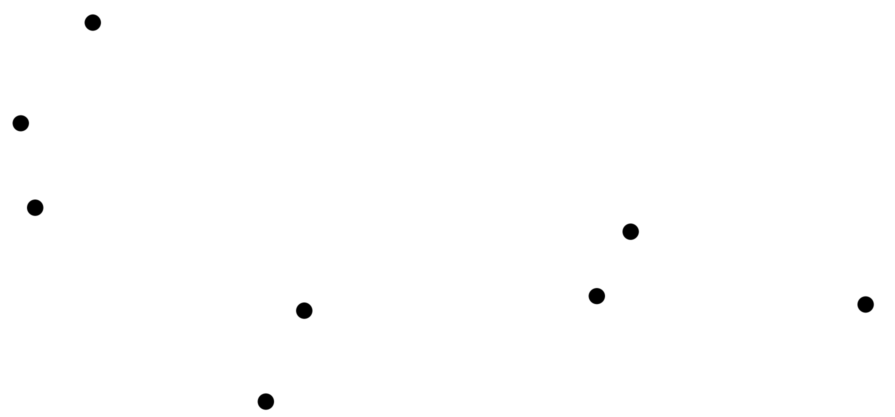
- Given set P of n points in metric space M



Constructing D

- Find the ball radius r such that $U_{balls}(P, r)$ has $\lceil n/2 \rceil$ connected components

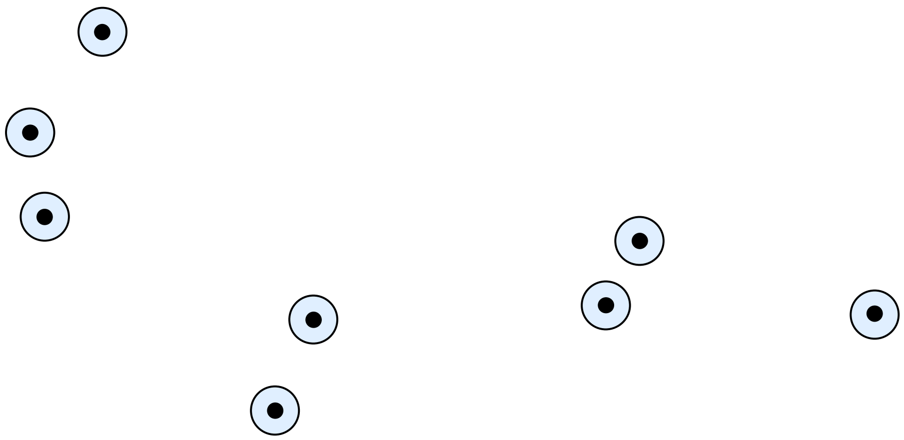
r = 0 Connected Components: 8



Constructing D

- Find the value of r such that $U_{balls}(P, r)$ has $\lceil n/2 \rceil$ connected components

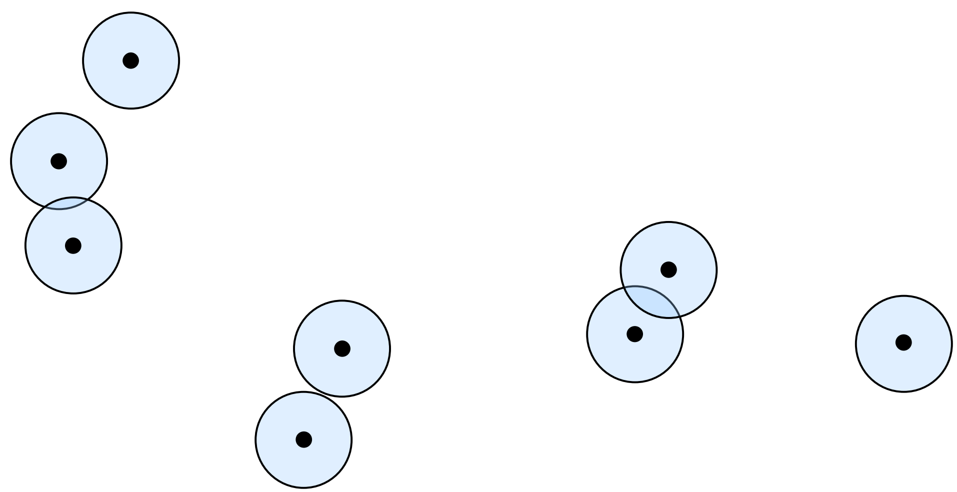
r = 0.25 Connected Components: 8



Constructing D

- Find the value of r such that $U_{balls}(P, r)$ has $\lceil n/2 \rceil$ connected components

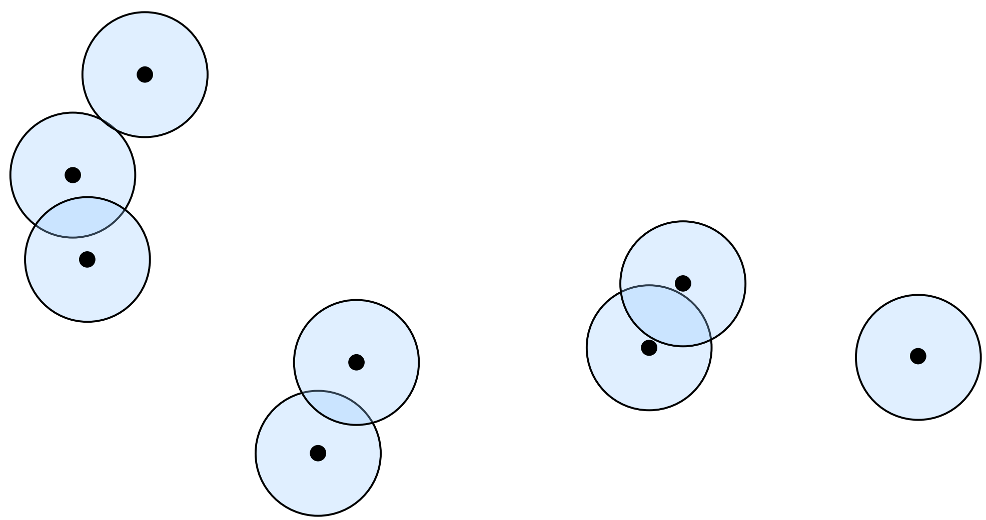
r = 0.5 Connected Components: 6



Constructing D

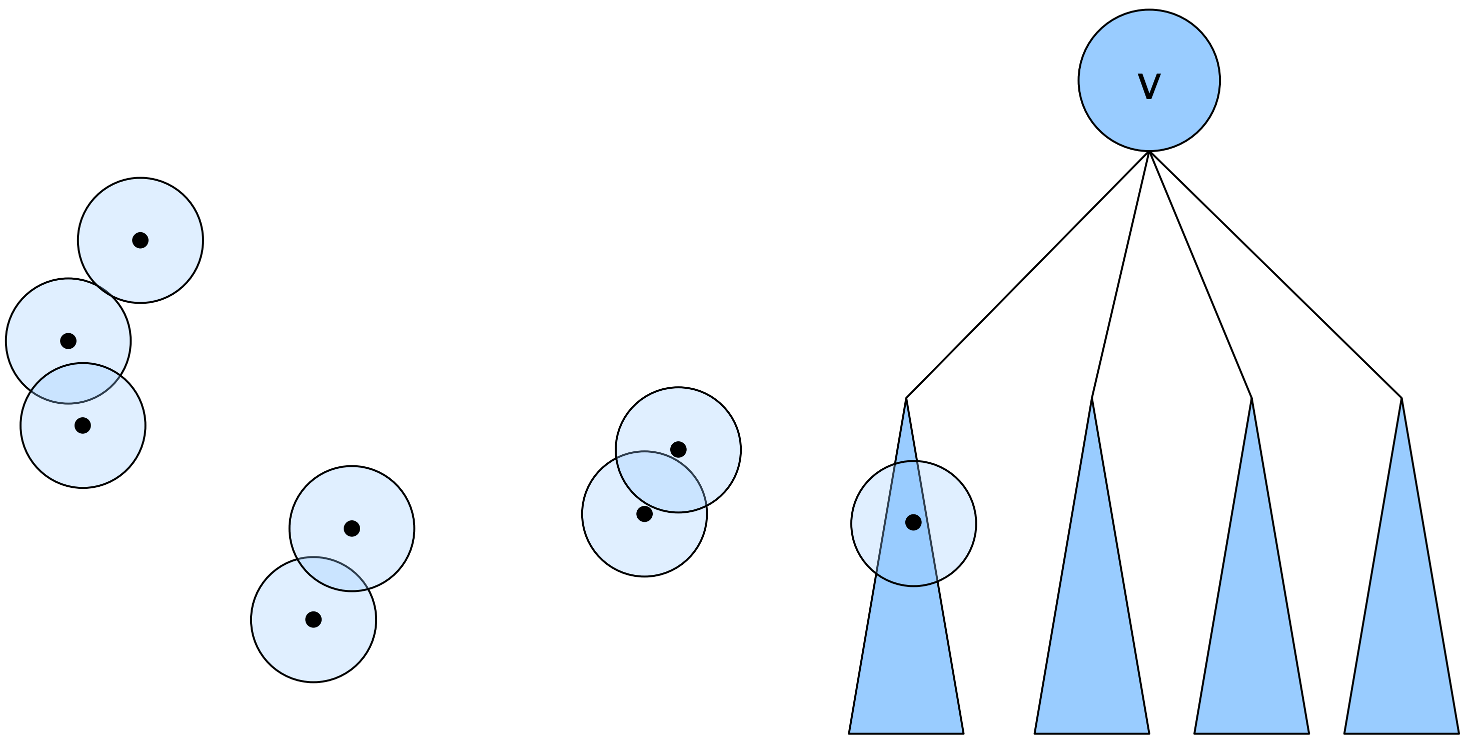
- Find the value of r such that $U_{balls}(P, r)$ has $\lceil n/2 \rceil$ connected components

r = 0.65 Connected Components: 4



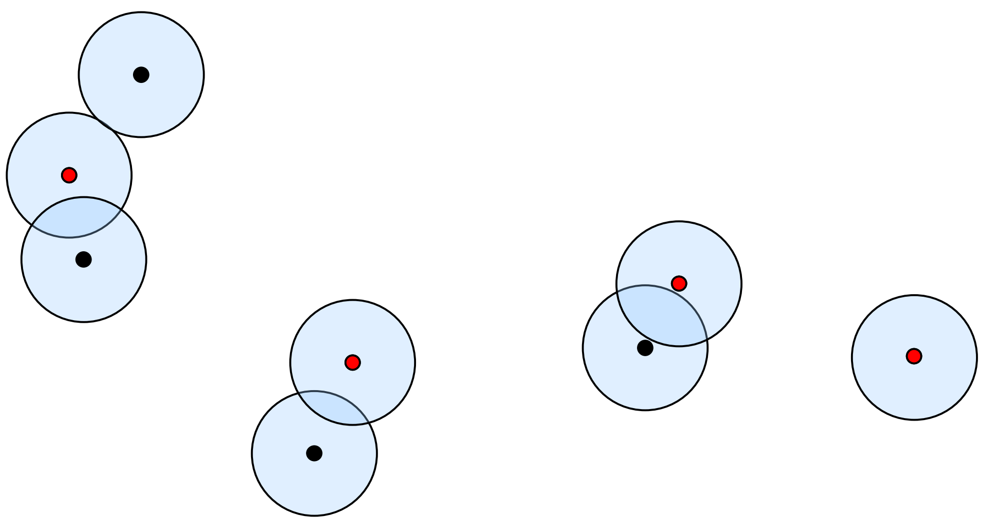
Constructing D

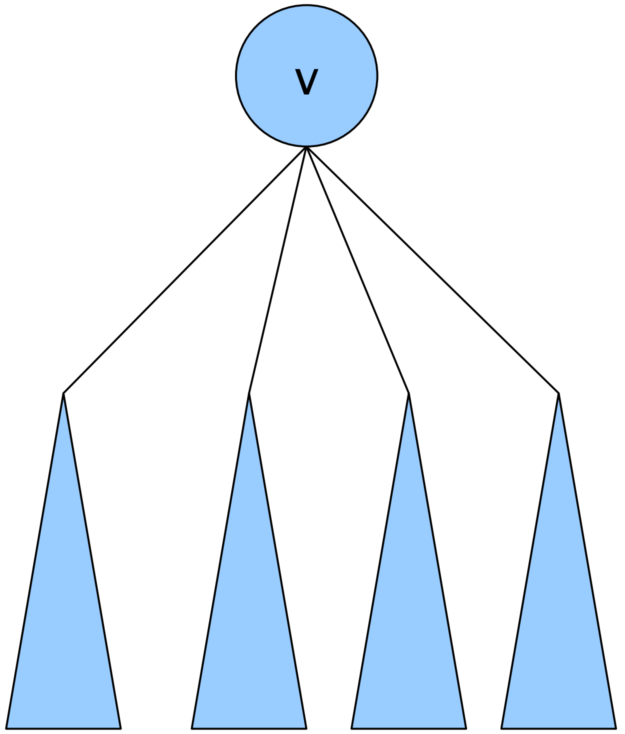
- Recursively build a sub tree for each connected component and add as child of root node v



Outer Child

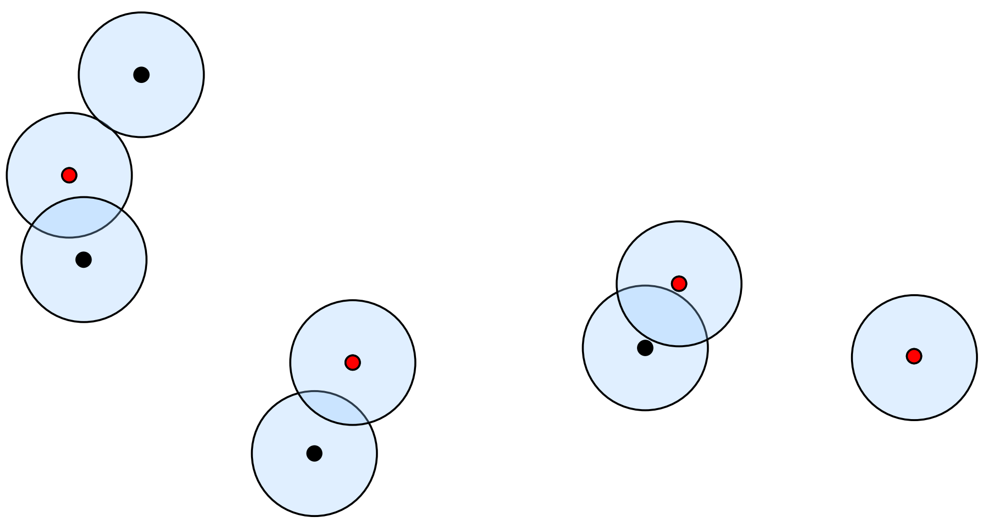
- Choose one representative from each connected component to be in set Q

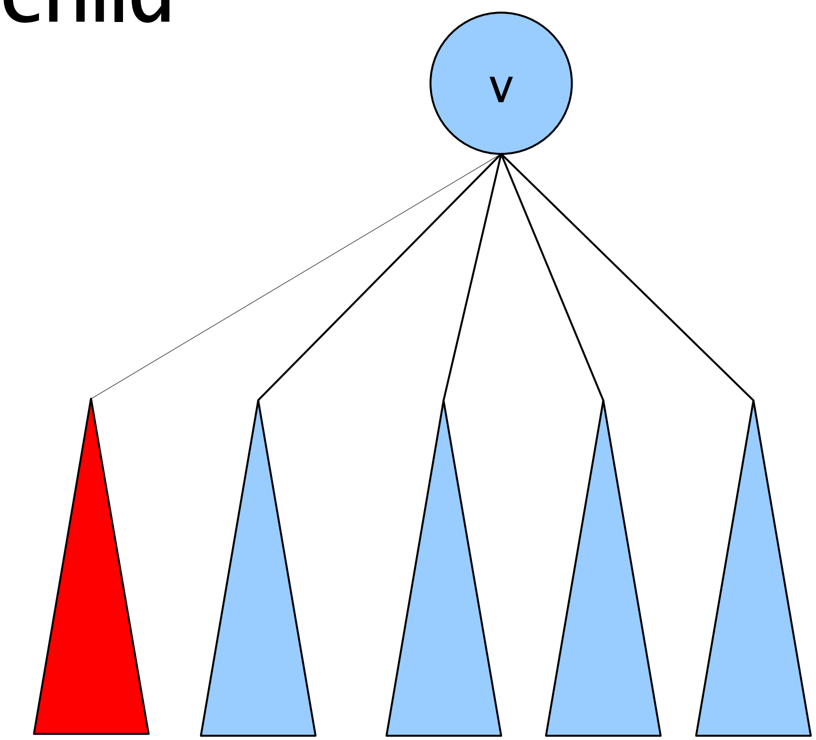




Outer Child

- Recursively build a tree over points in Q and hang it on on node v
- This child of v is the "outer child"





Constructing D

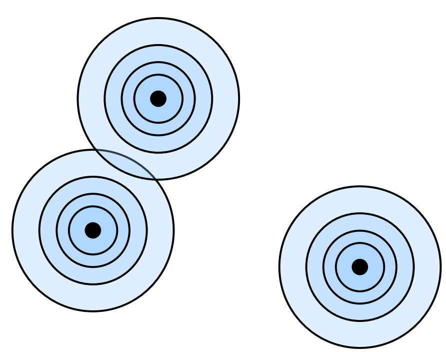
- Build the interval NNbr data structure for node v

$$\hat{I}_v = \hat{I}(P, r, R, \varepsilon/4)$$

point set — search range [r, R] — approximation error

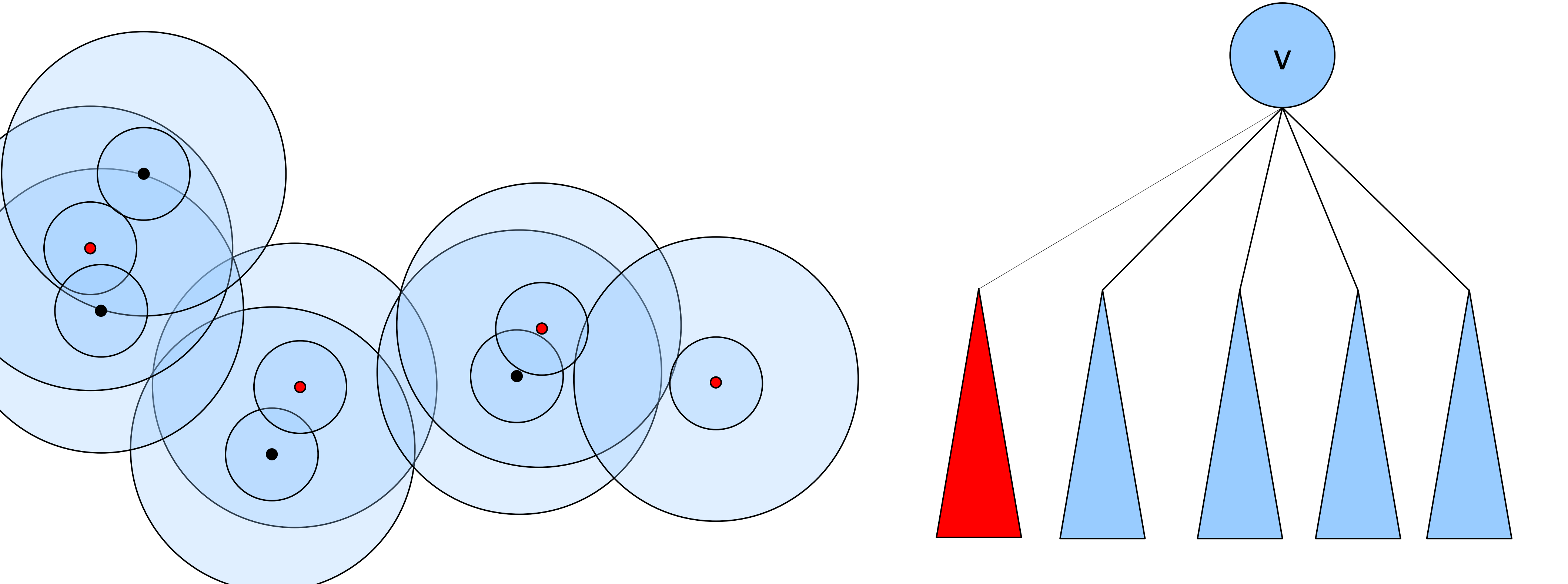
Let $R = 2\bar{c}\mu n r/\varepsilon$

Where μ & $\bar{c}$ are parameters that will be defined later...



Answering a query using D

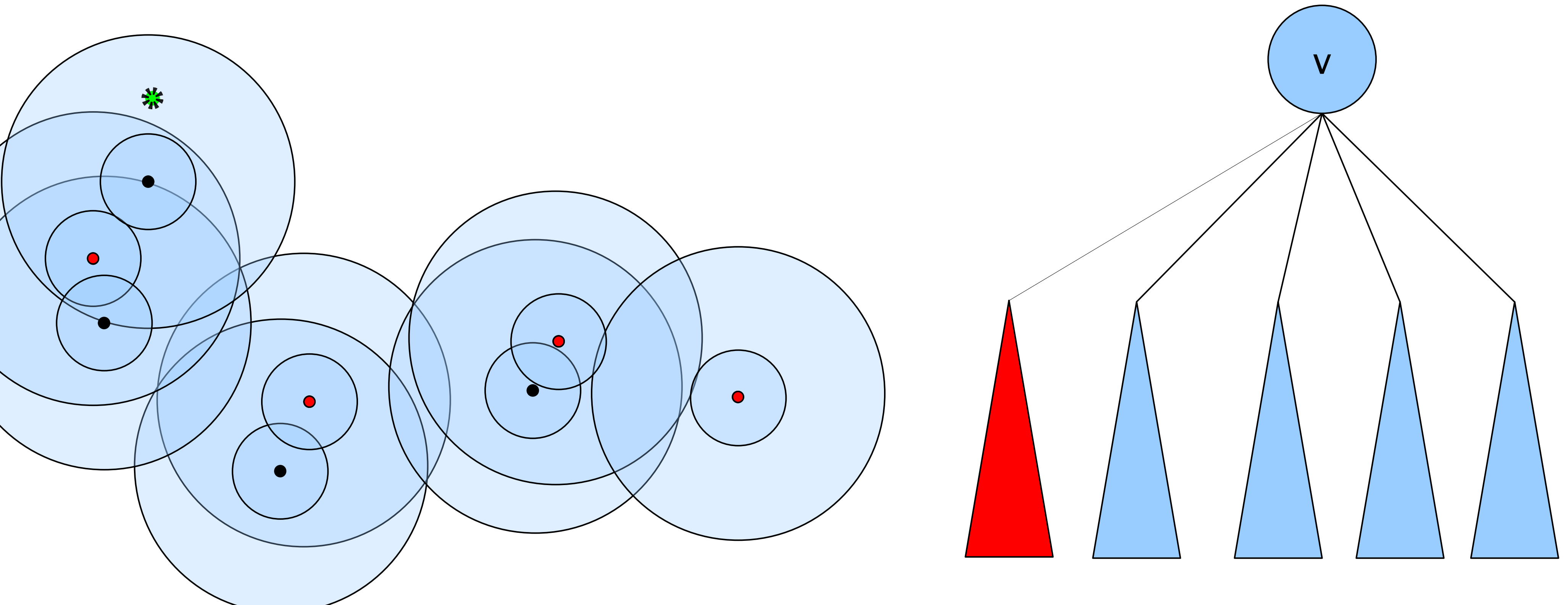
- Given query point q, use $\hat{I}_v$ to decide between three cases



Answering a query using D

Case 1:

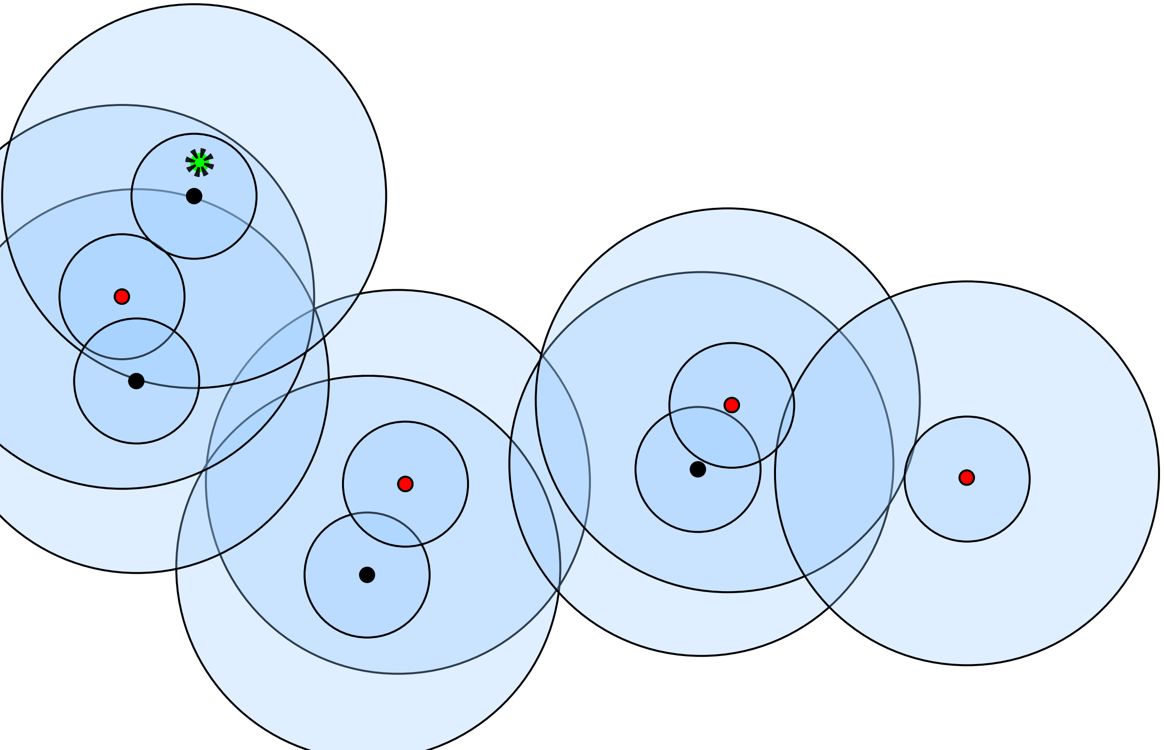
- $\hat{I}_v$ returns $(1+\varepsilon)\,ANN$ and search terminates

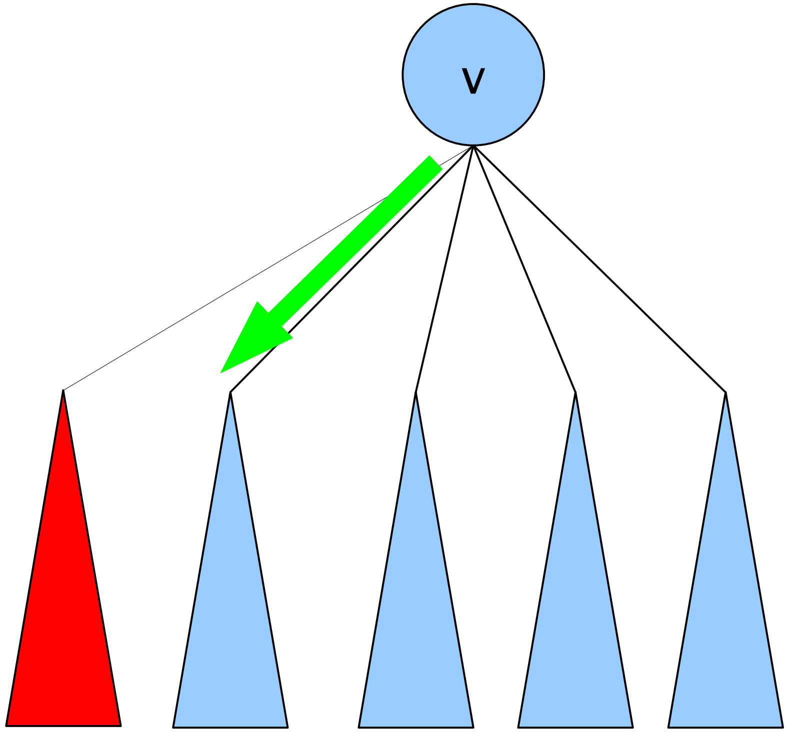


Answering a query using D

Case 2: $d_P(q) \le r_v$

- Recurse into child corresponding to connected component containing q

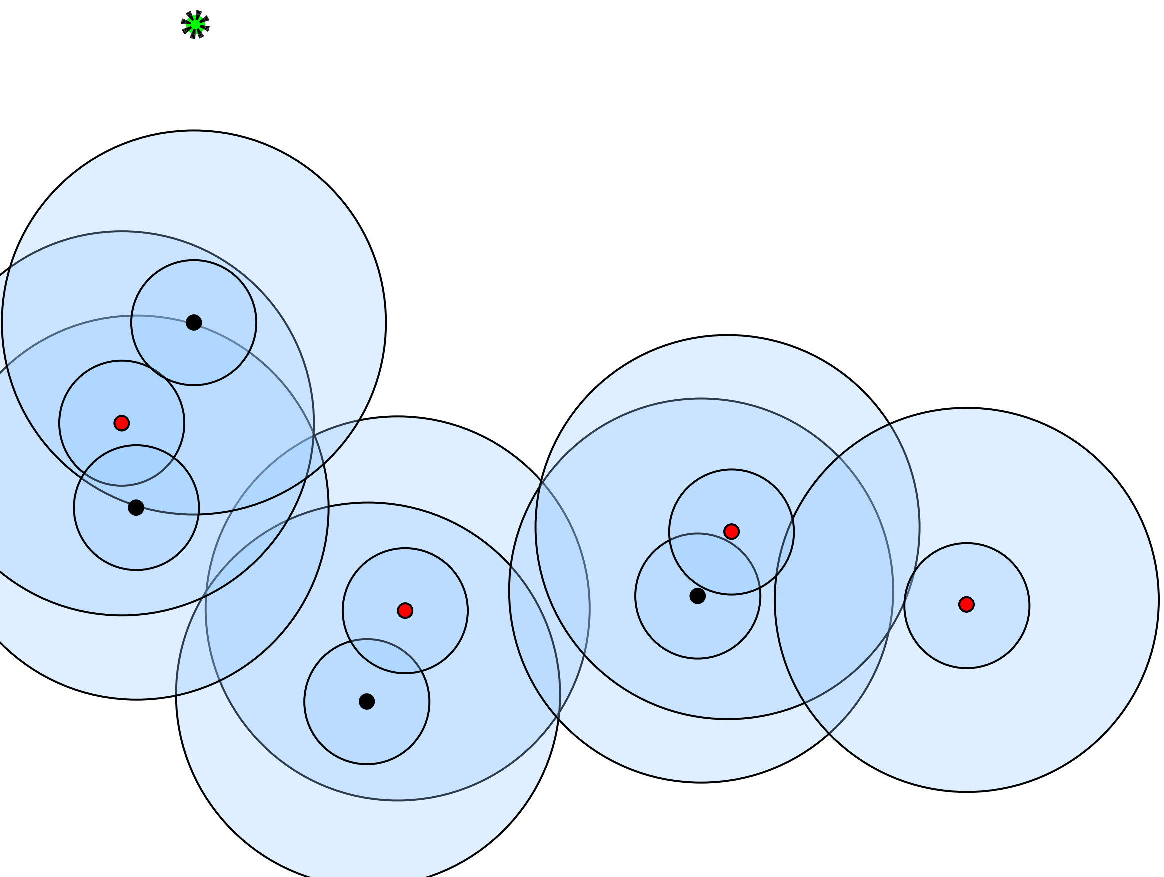


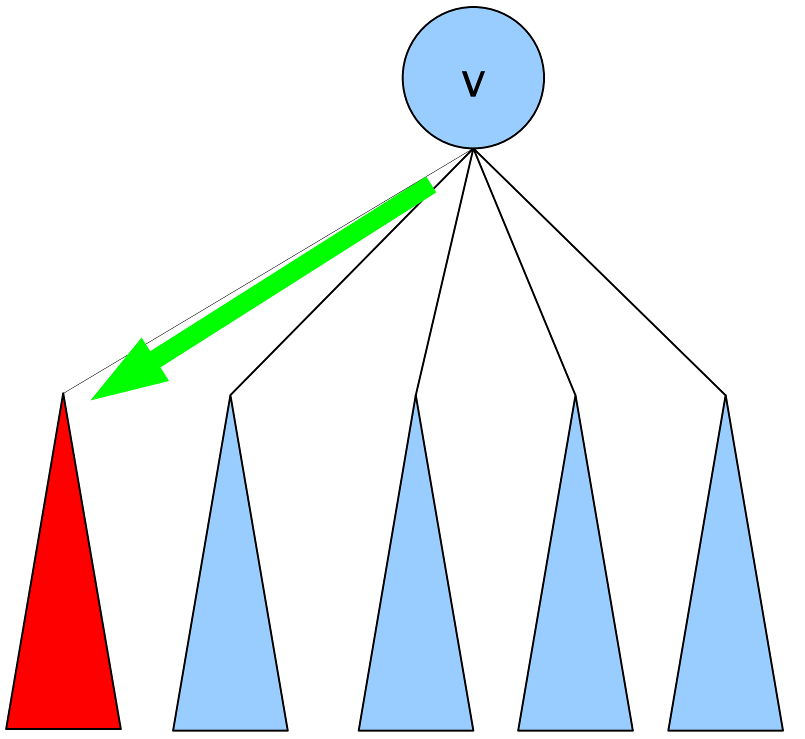


Answering a query using D

Case 3: $d_P(q) > R_v$

- Recurse into outer child





algorithm terminates

- If at step i we consider a set of size n_i then at step i+1 we consider a set of size

$$n_{i+1} \leq n_i / 2 + 1$$

- Thus search halts after number of steps

$$steps \leq \log_{3/2}(n)$$

Algorithm is correct

- Same result as target ball query on all constructed balls
- Approximation error
  - From node v to a connected component child
    - No approximation error
  - From node v to the "outer child": $1+\varepsilon/(\bar{c}\,\mu)$
  - From the interval NNbr search: $1+\varepsilon/4$

Approximation error

$$t \leq (1+\frac{\varepsilon}{4}) \prod_{i=1}^{\log_{3/2}(n)} (1+\frac{\varepsilon}{\bar{c}\,\mu})$$

$$\leq \exp(\frac{\varepsilon}{4}) \prod_{i=1}^{\log_{3/2}(n)} (\frac{c\,\varepsilon}{\bar{c}\,\mu})$$

set $\mu = \lceil \log_{3/2} n \rceil$ and $\bar{c}$ large enough so that...

$$\leq \exp(\frac{\varepsilon}{4} + \sum_{i=1}^{\log_{3/2}(n)} \frac{\varepsilon}{\bar{c}\,\mu})$$

$$\leq \exp(\frac{\varepsilon}{2})$$

$$\leq 1+\varepsilon$$

Thus result of a query on d is $(1+\varepsilon)$-ANN to query point q

Query time

- As search proceeds down tree D
  - at most two NNbr queries are performed at a node and we traverse O(log n) nodes
  - at last node the $\hat{I}_v$ data structure performs $O(\log(\log(\frac{n}{\varepsilon})/\varepsilon)) = O(\log\frac{n}{\varepsilon})$ NNbr queries
  - Query time is $O(\log\frac{n}{\varepsilon})$

Efficient Construction

- Construction space/time is currently $O(n^2)$
- Use HST of P to t-approximate metric M
- Use correspondence between subtrees in HST and connected components to find the ball radius r that gives $\lceil n/2 \rceil$ connected components
- Results in construction space/time $O(\frac{n}{\varepsilon}\log\frac{t\,n}{\varepsilon})$

• What have we done?

- Reduced an ANN query to multiple NNbr queries
- But NNbr queries seem hard to solve efficiently
  - Solution: Use deformed “approximate balls”
  - Same bounds hold for the extension to “approximate balls”

Questions