UTEC
UNIVERSIDAD DE INGENIERÍA Y TECNOLOGÍA

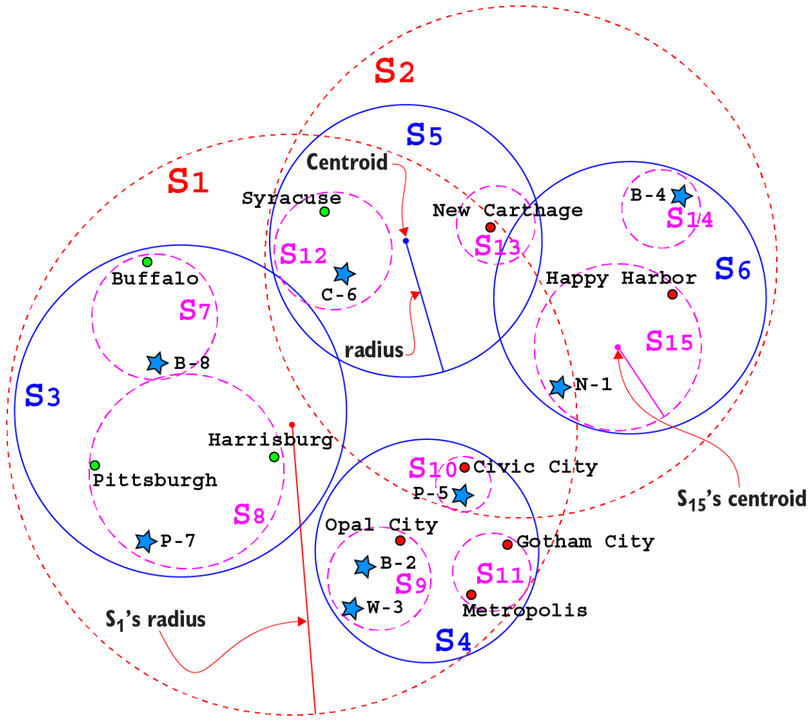
Sesión 11.1: Balls and Rectangles

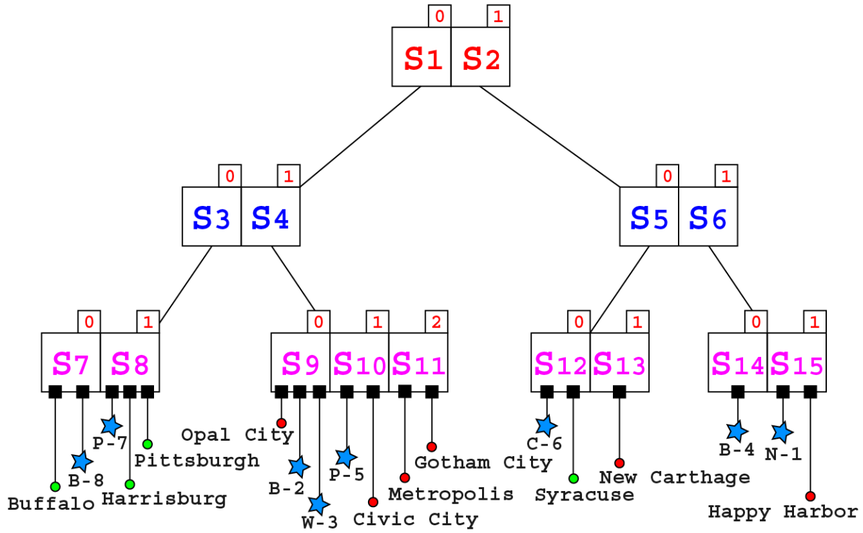
CS3102 EDA

Índice

1. SS-Tree

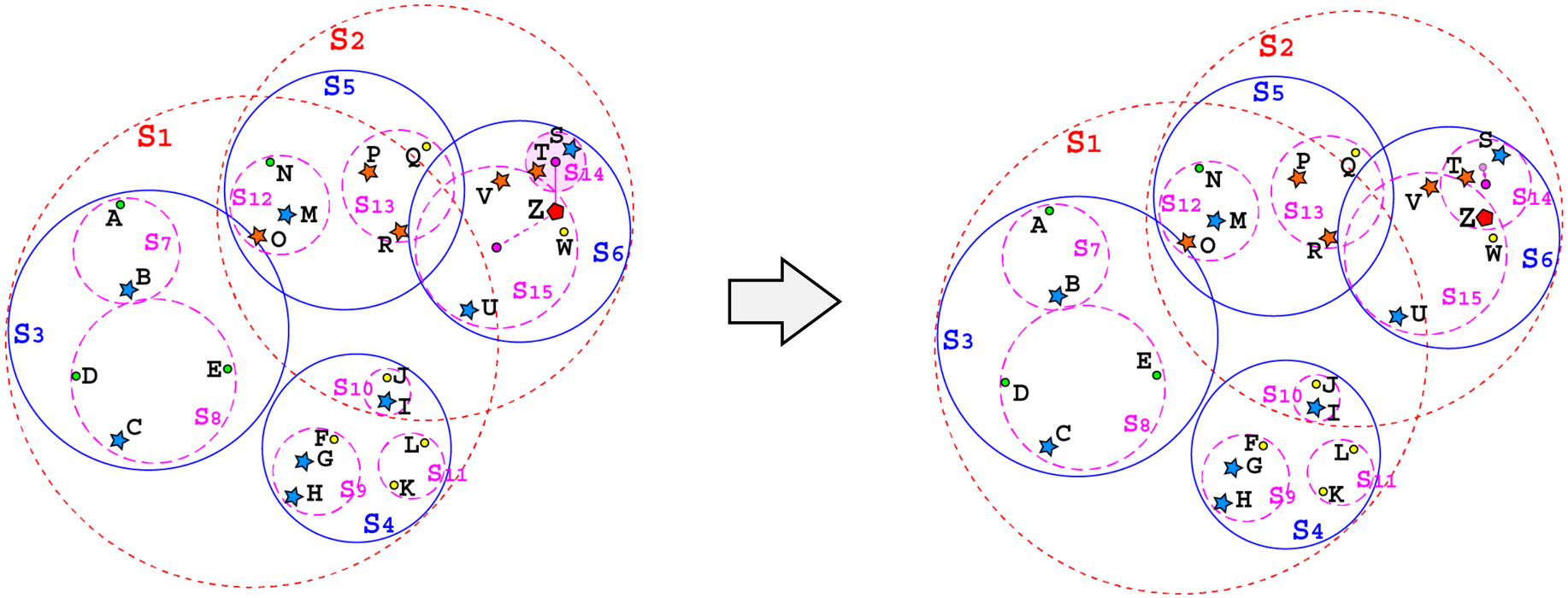
SS-*Tree*


S2
S5
S1
Centroid
Syracuse
New Carthage
B-4
S14
S12
S13
S6
C-6
Buffalo
Happy Harbor
S7
S15
radius
B-8
N-1
S3
Harrisburg
S10
Civic City
Pittsburgh
P-5
S15's centroid
S8
Opal City
Gotham City
P-7
B-2
S9
S11
W-3
Metropolis
S1's radius
S4


0 1
S1 S2
0 1
S3 S4
0 1
S5 S6
0 1
S7 S8
0 1 2
S9 S10 S11
0 1
S12 S13
0 1
S14 S15
P-7
Opal City
C-6
B-4
N-1
B-8
Pittsburgh
B-2
P-5
Gotham City
Buffalo
Harrisburg
Metropolis
Syracuse
New Carthage
W-3
Civic City
Happy Harbor

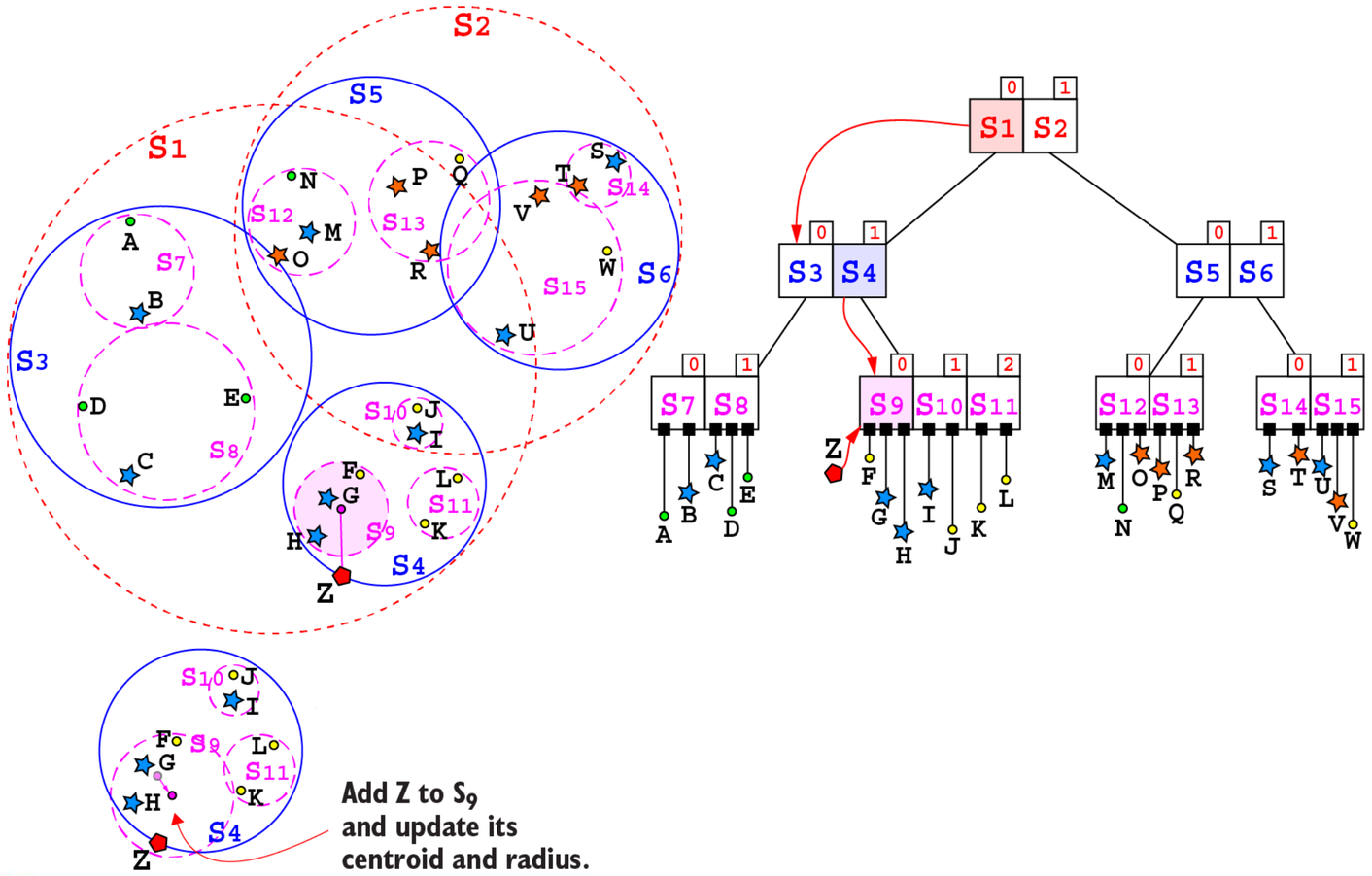
SS-*Tree*

Inserción



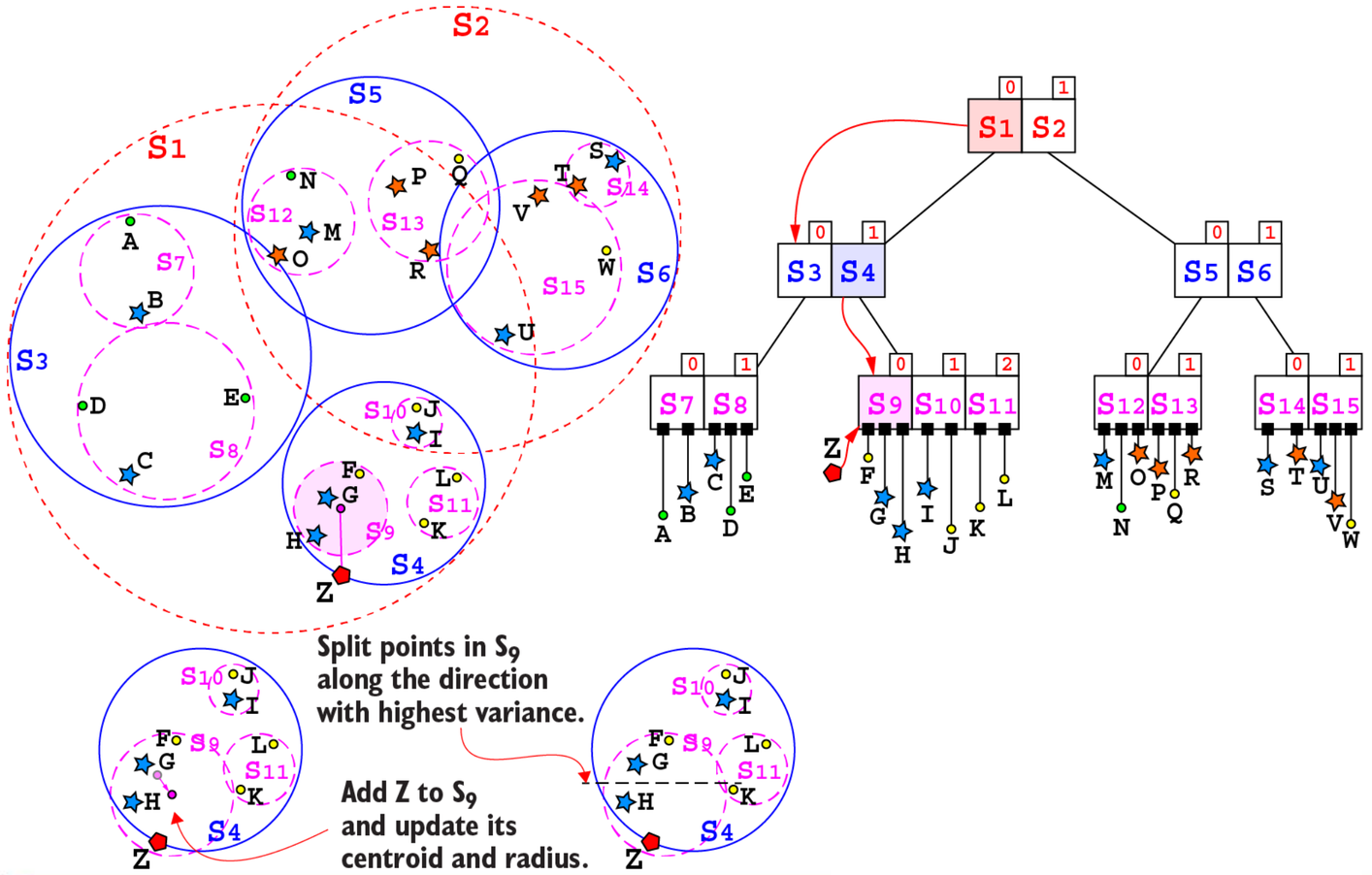
SS-*Tree*

División



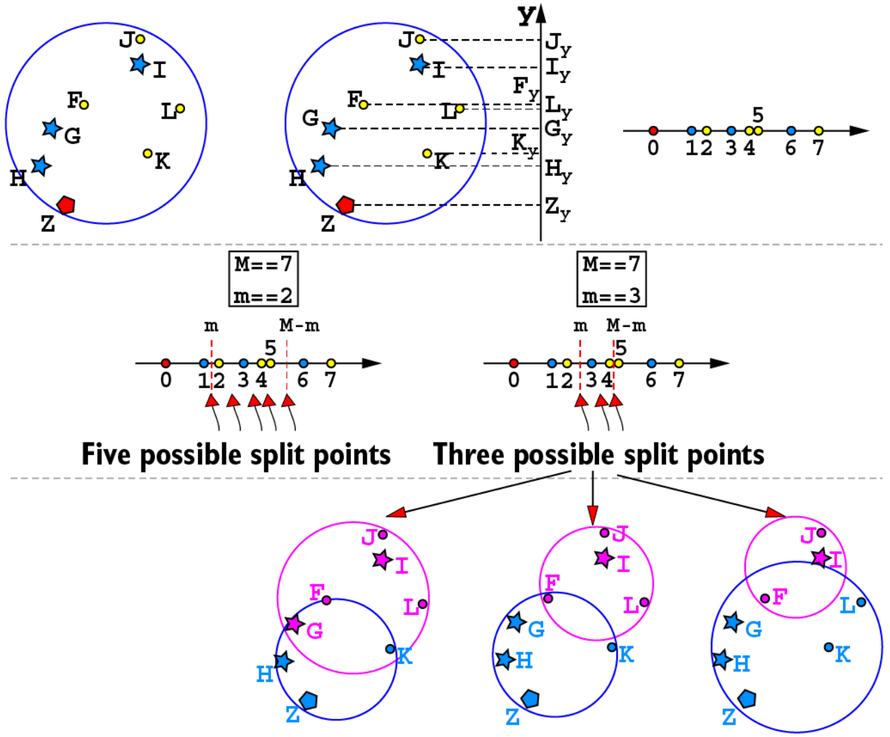
SS-*Tree*

División



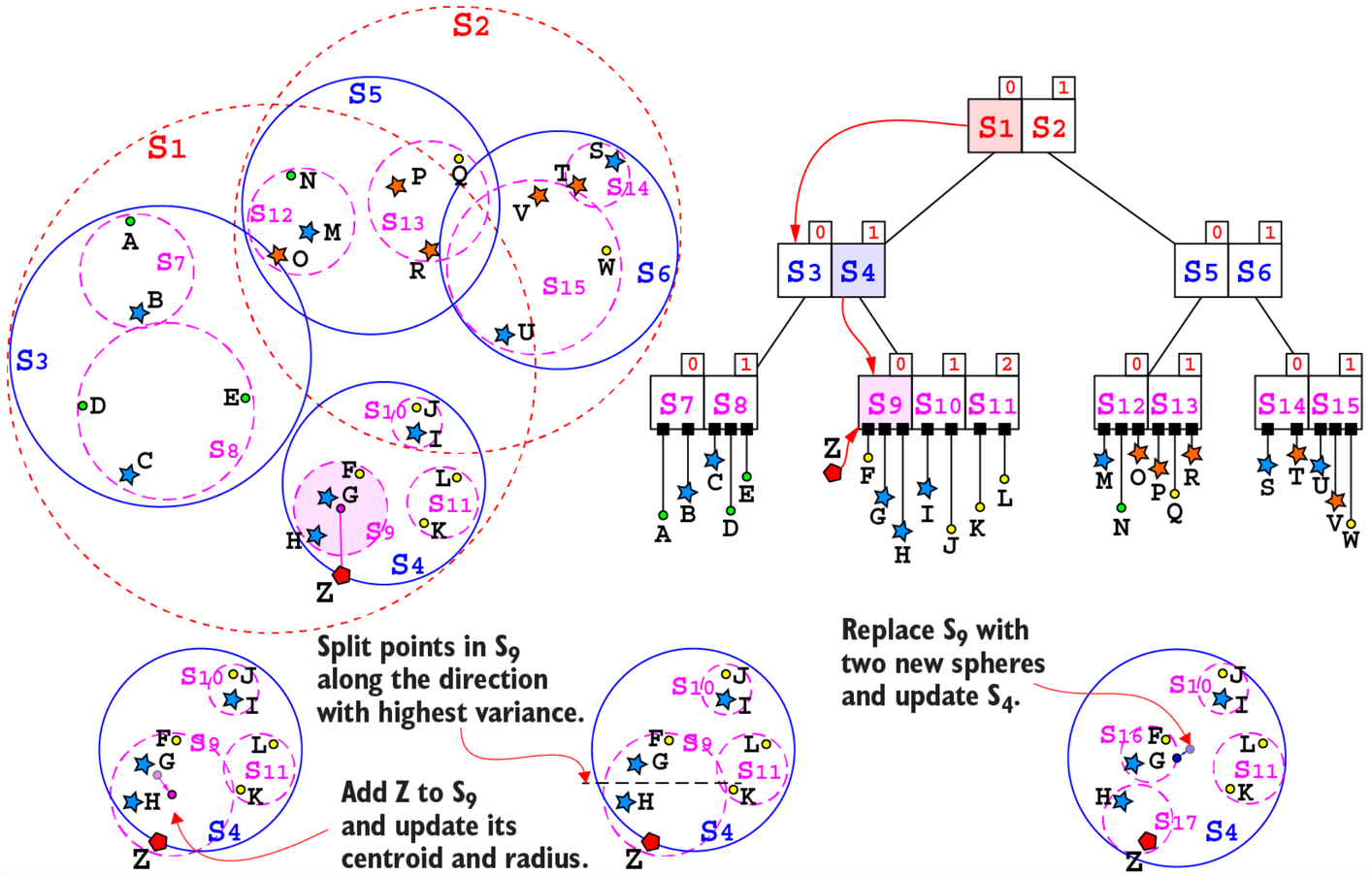
SS-*Tree*

División



SS-*Tree*

División

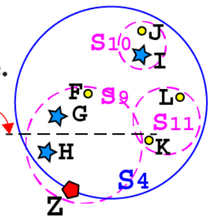


SS-*Tree*

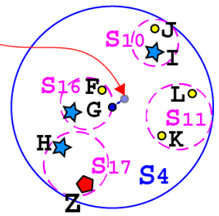
División

Varianza máxima

Split points in S_9 along the direction with highest variance.

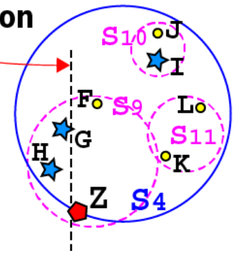


Replace S_9 with two new spheres and update S_4.

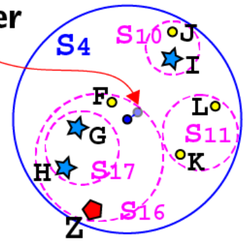


Varianza mínima

Split direction



S_4's center



SS-*Tree*

Borrado

Buscamos el nodo a eliminar, Z, en el árbol. Dentro del nodo hoja L, ejecutamos:

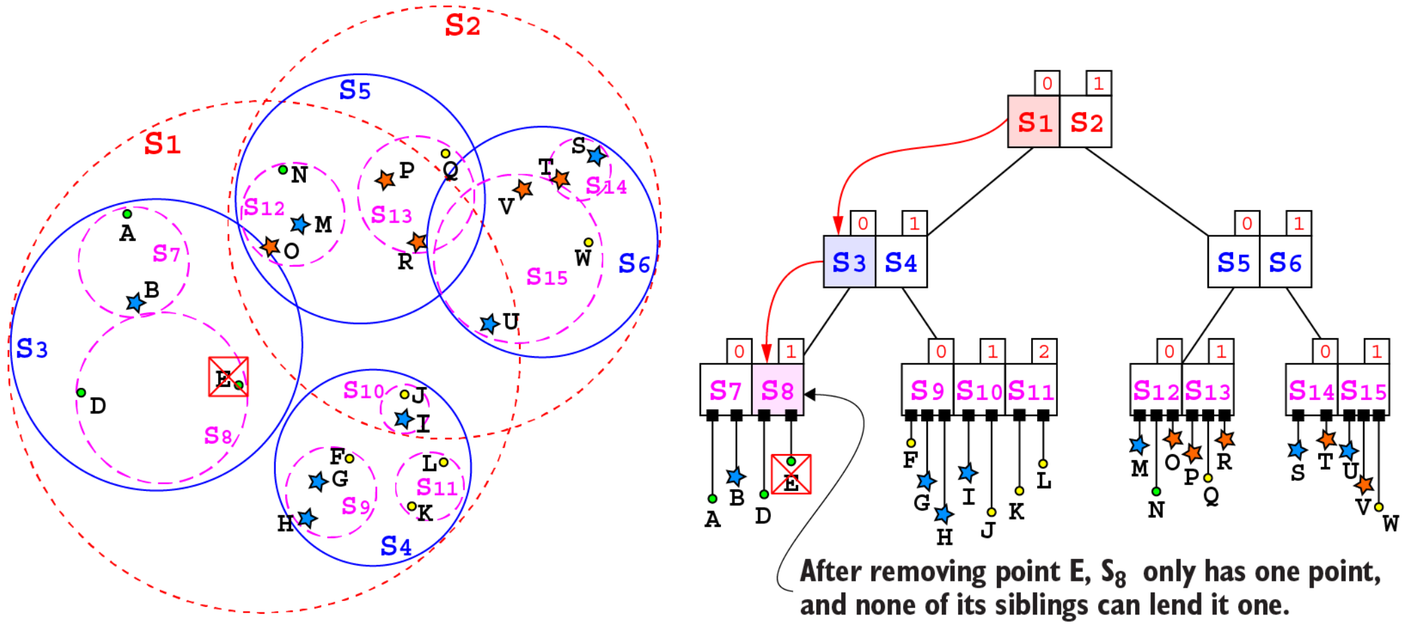
Si la hoja contiene más de m puntos, simplemente eliminamos Z de L y actualizamos su sobre delimitador.

De lo contrario:

1. Si L es la raíz, estamos bien y no tenemos que hacer nada.
2. Si L tiene al menos un hermano S con más de m puntos, podemos mover un punto de S a L.
3. Si ningún hermano de L puede prestarle un punto, entonces tendremos que fusionarnos L con uno de sus hermanos.

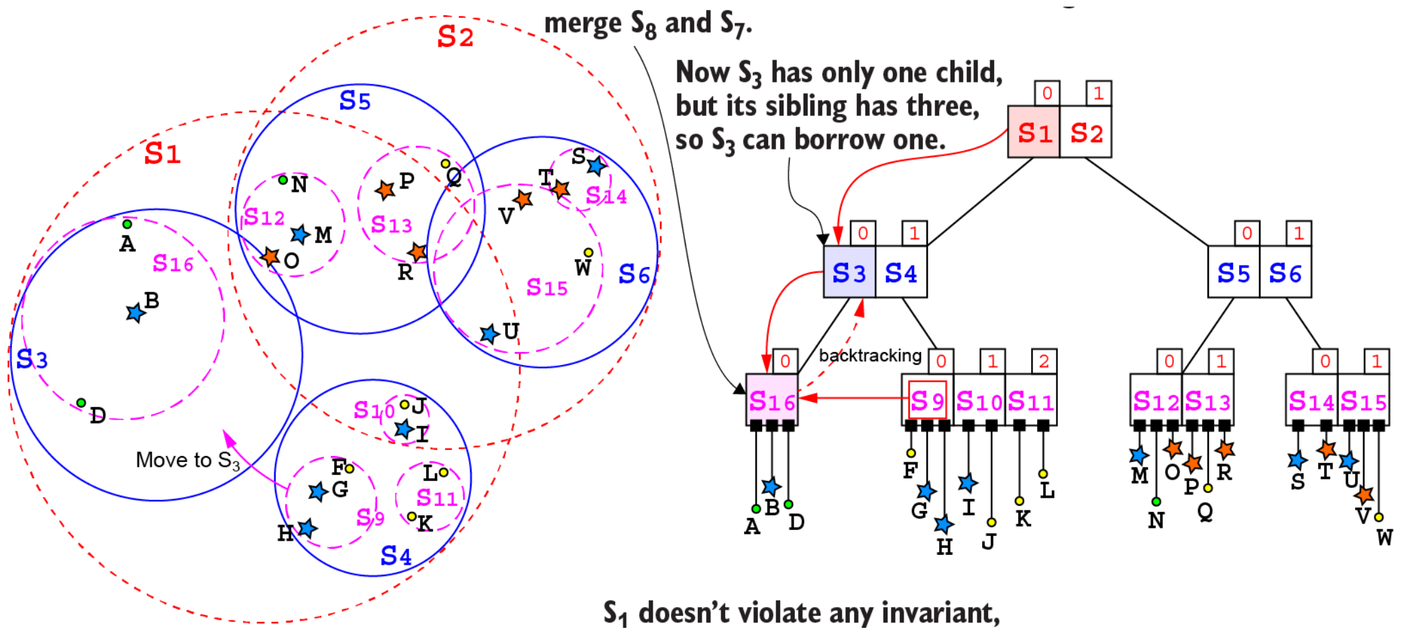
SS-*Tree*

Borrado



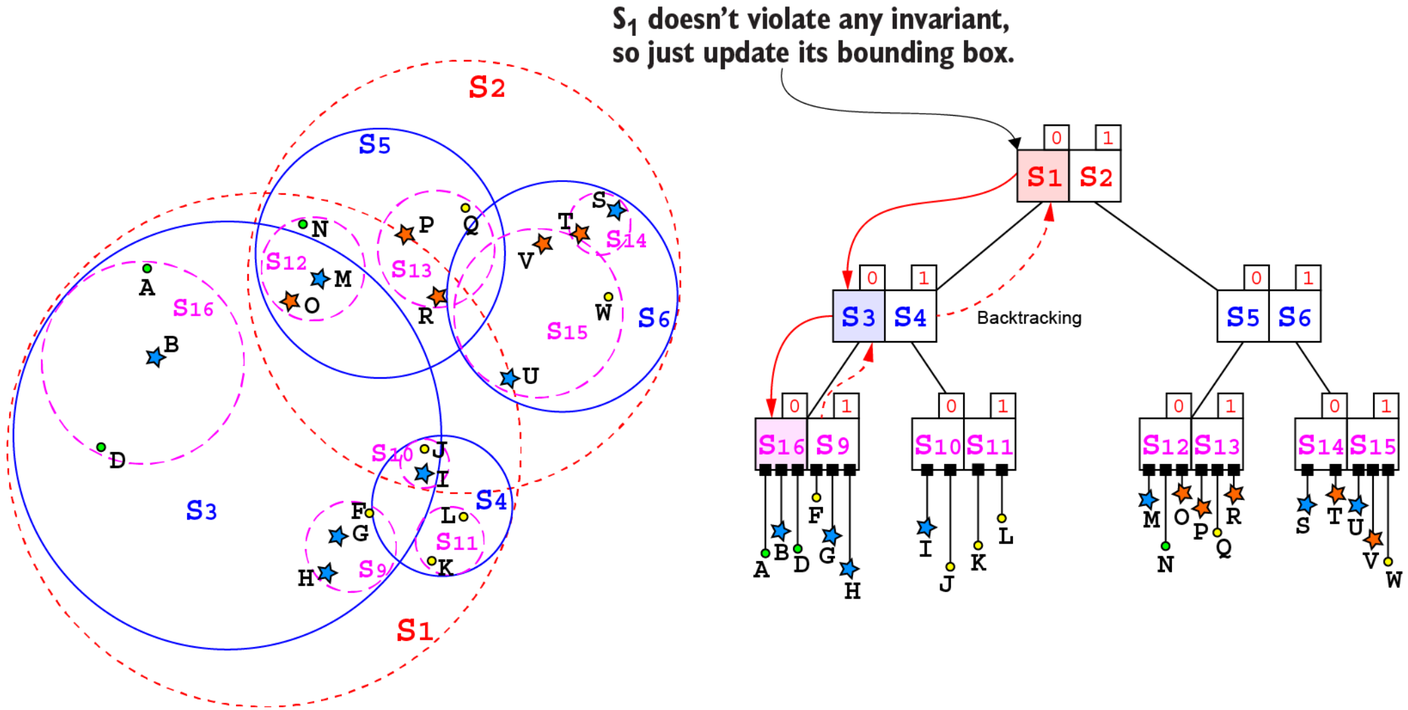
SS-*Tree*

Borrado



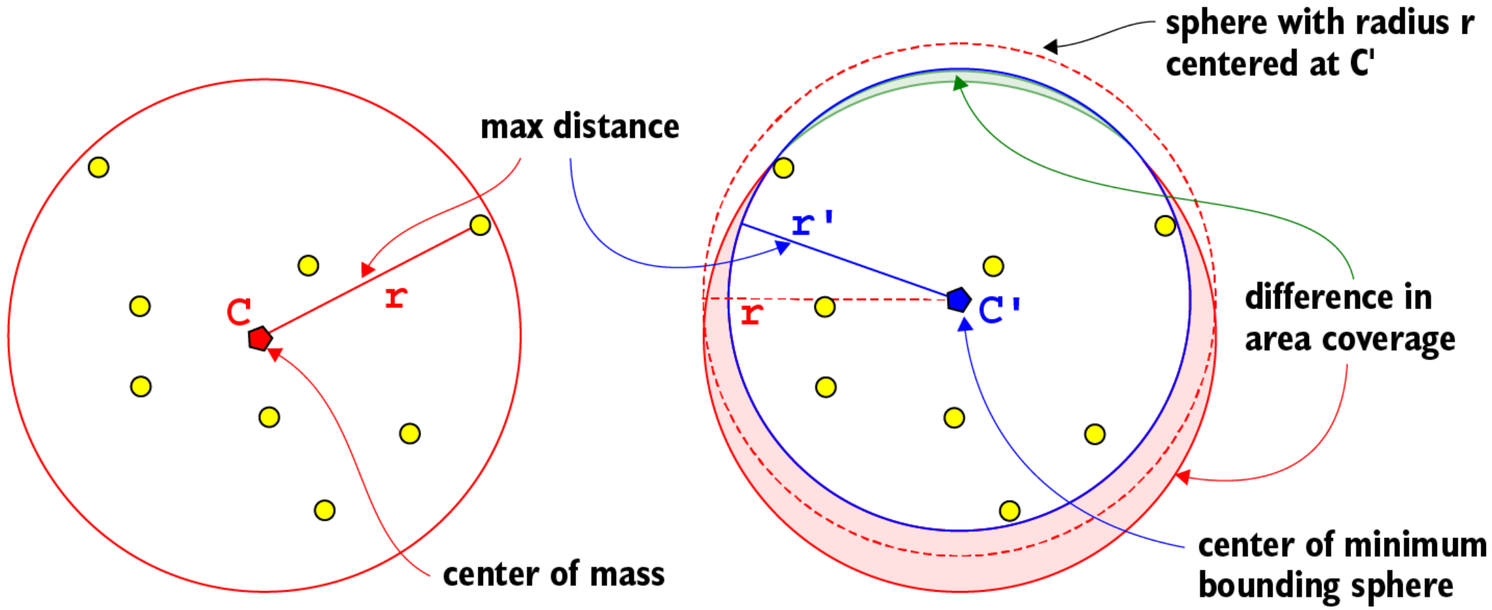
SS-*Tree*

Borrado

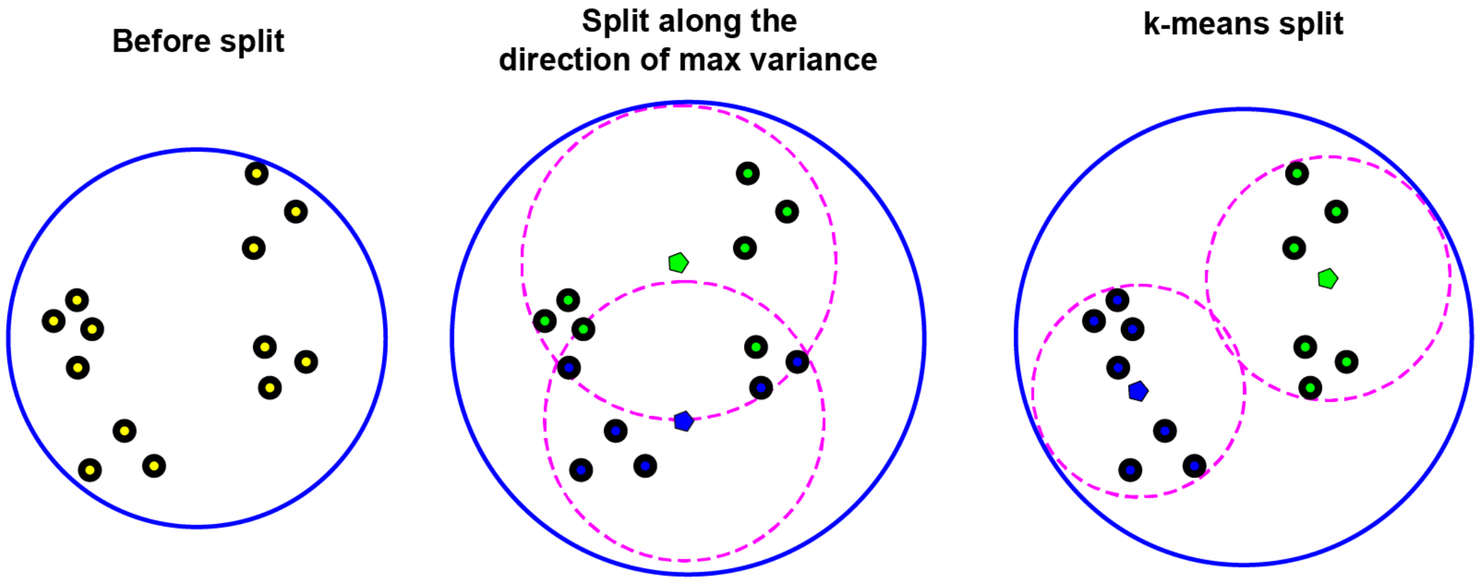


2. SS+-Tree

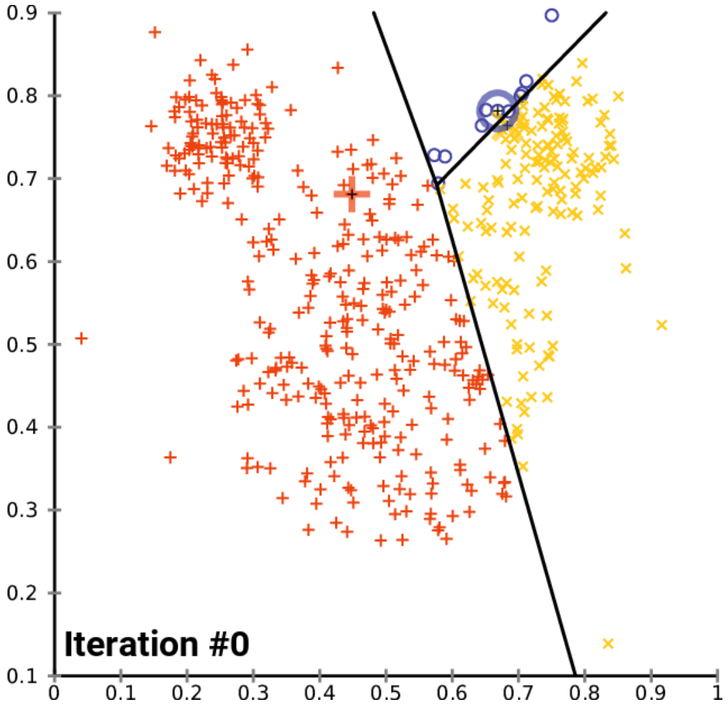
SS+-Tree


sphere with radius r
centered at C'
max distance
r'
C
r
r
C'
difference in
area coverage
center of mass
center of minimum
bounding sphere

SS+-Tree

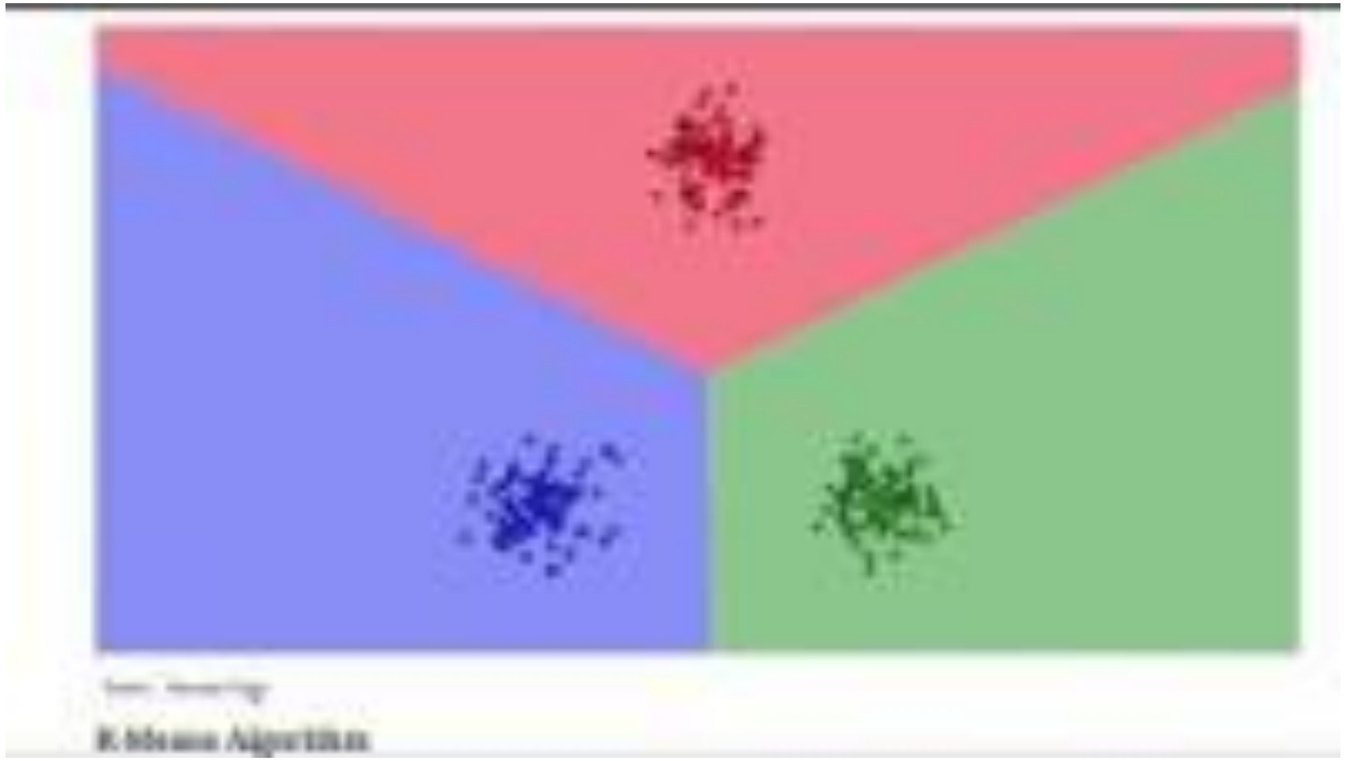

Before split
Split along the direction of max variance
k-means split

SS⁺-*Tree*



K-MEANS($\{\vec{x}_1, \ldots, \vec{x}_N\}, K$)
1 $(\vec{s}_1, \vec{s}_2, \ldots, \vec{s}_K) \leftarrow$ SELECTRANDOMSEEDS($\{\vec{x}_1, \ldots, \vec{x}_N\}, K$)
2 **for** $k \leftarrow 1$ **to** K
3 **do** $\vec{\mu}_k \leftarrow \vec{s}_k$
4 **while** stopping criterion has not been met
5 **do for** $k \leftarrow 1$ **to** K
6 **do** $\omega_k \leftarrow \{\}$
7 **for** $n \leftarrow 1$ **to** N
8 **do** $j \leftarrow \arg\min_{j'} |\vec{\mu}_{j'} - \vec{x}_n|$
9 $\omega_j \leftarrow \omega_j \cup \{\vec{x}_n\}$ *(reassignment of vectors)*
10 **for** $k \leftarrow 1$ **to** K
11 **do** $\vec{\mu}_k \leftarrow \frac{1}{|\omega_k|} \sum_{\vec{x} \in \omega_k} \vec{x}$ *(recomputation of centroids)*
12 **return** $\{\vec{\mu}_1, \ldots, \vec{\mu}_K\}$

SS+-Tree



https://www.youtube.com/watch?v=R2e3Ls9H_fc

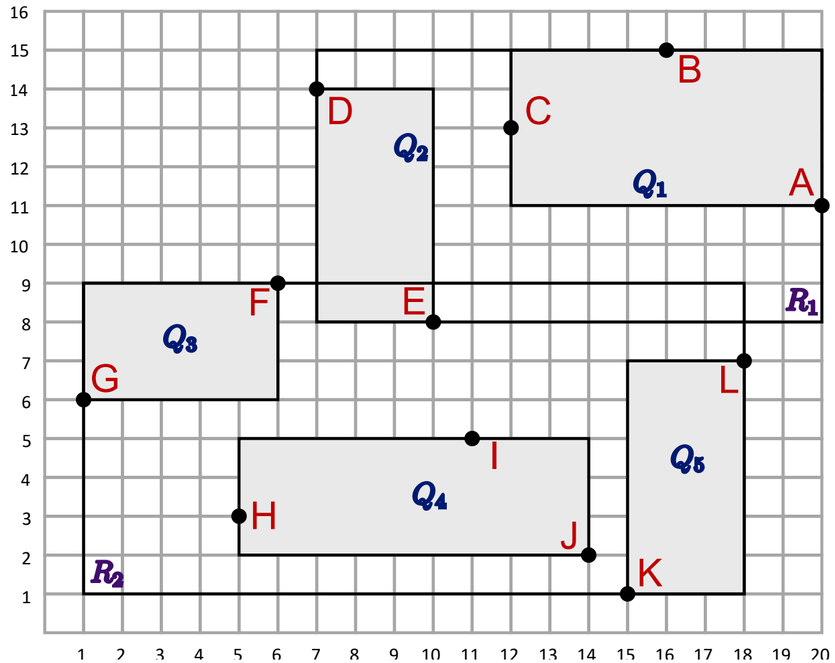
https://www.naftaliharris.com/blog/visualizing-k-means-clustering/

3. x-Tree

R*-Tree

El overlaping de R*-Tree aumenta con el número de dimensiones

R*-Tree



Overlap

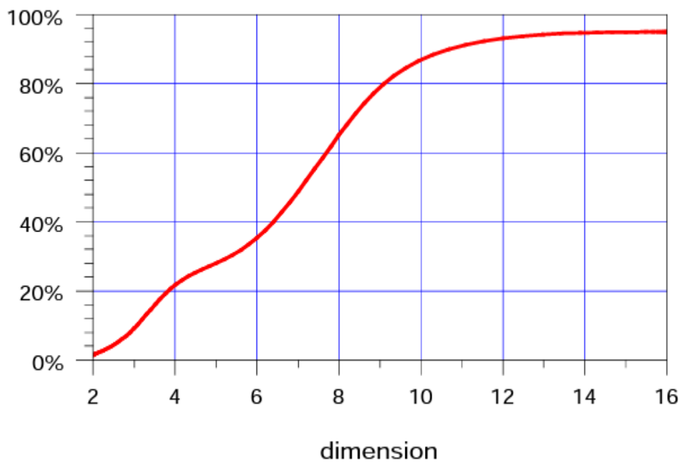
$$O = \frac{\bigcup_{i \neq j} R_i \cap R_j}{\bigcup_i R_i}$$

Weighted Overlap

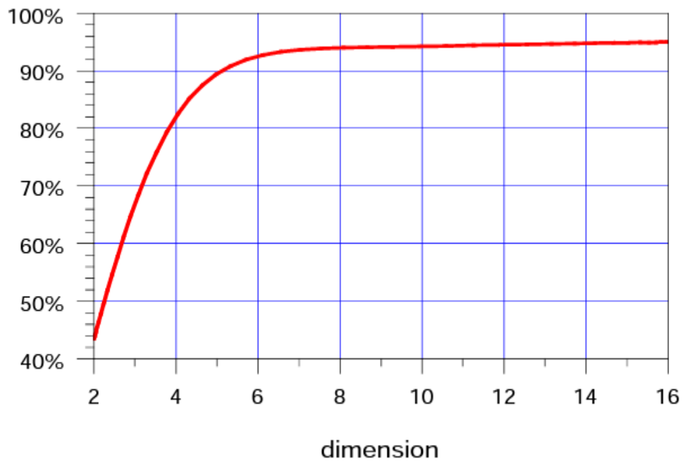
$$WO = \frac{\{p | p \in \bigcup_{i \neq j} R_i \cap R_j\}}{\{p | p \in \bigcup_i R_i\}}$$

R*-Tree

Overlap (Uniformly Distributed Data)

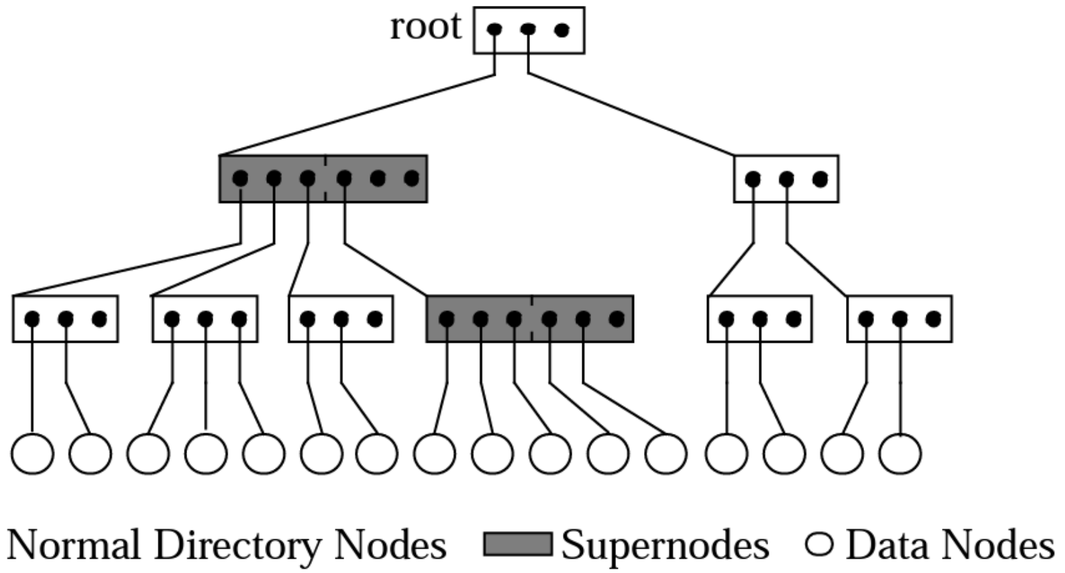


Weighted Overlap (Real Data)



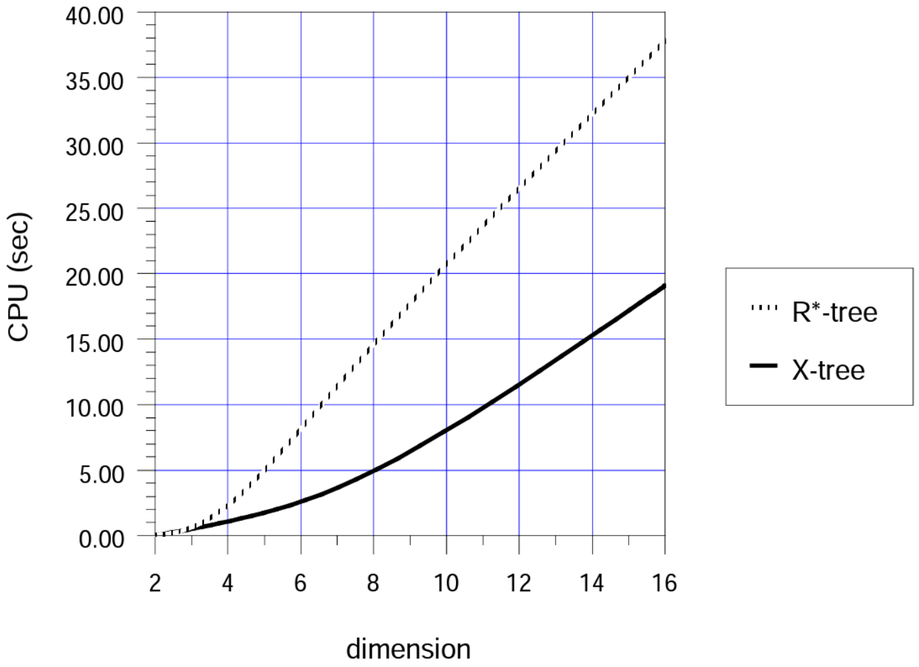
X-*Tree*

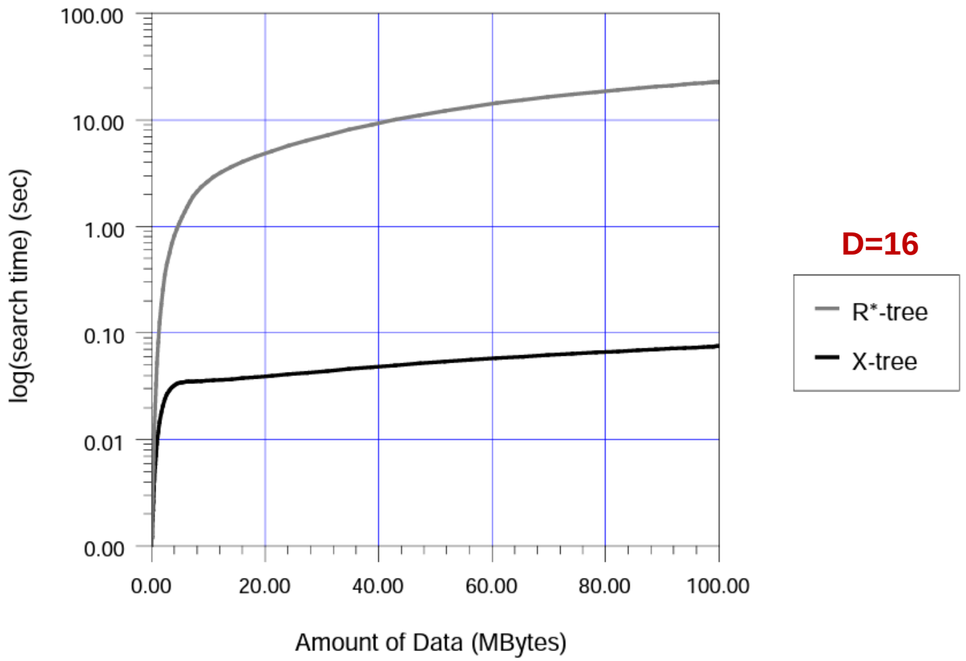
(eXtended node tree)



Berchtold, Stefan, Daniel A. Keim, and Hans-Peter Kriegel (1996) "The X-tree: An index structure for high-dimensional data". Very large data-bases (pp. 28-39).

X-*Tree*





Berchtold, Stefan, Daniel A. Keim, and Hans-Peter Kriegel (1996) "The X-tree: An index structure for high-dimensional data". Very large data-bases (pp. 28-39).

UTEC

UNIVERSIDAD DE INGENIERÍA
Y TECNOLOGÍA