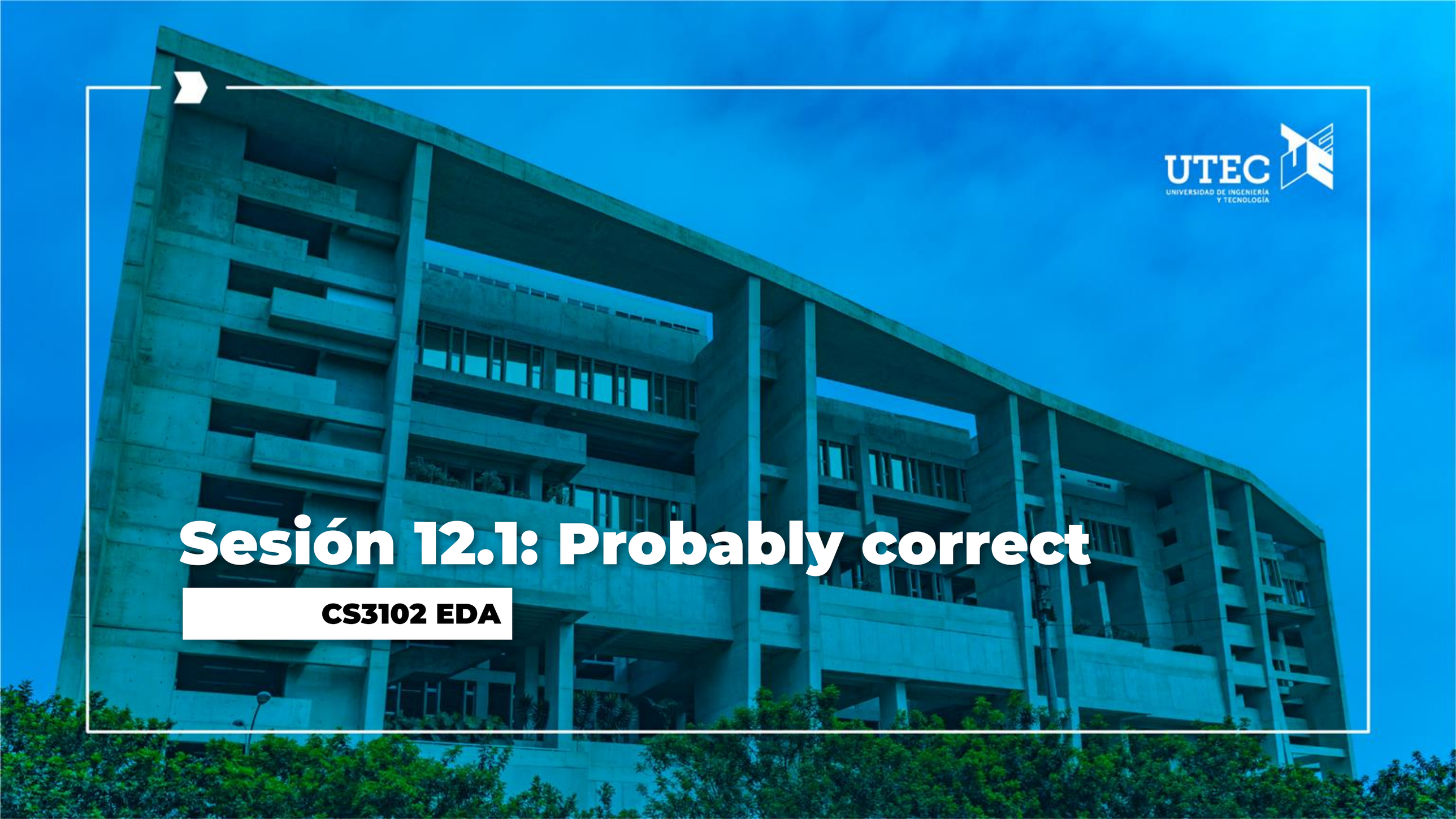


UTEC
UNIVERSIDAD DE INGENIERÍA Y TECNOLOGÍA

Sesión 12.1: Probably correct

CS3102 EDA

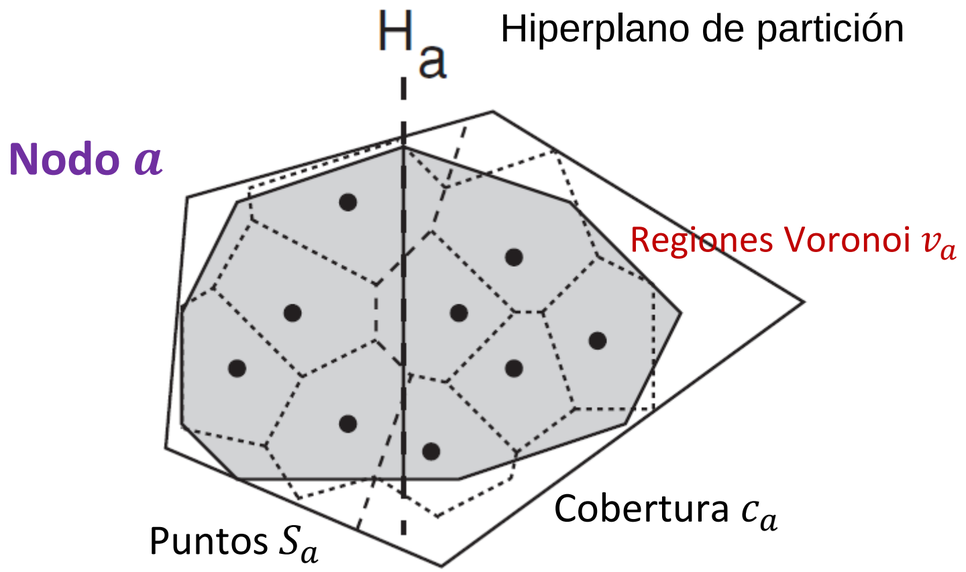
Índice

1. OS-Tree
2. AVD
3. Distance-Based Indexing Methods

1. os-Tree

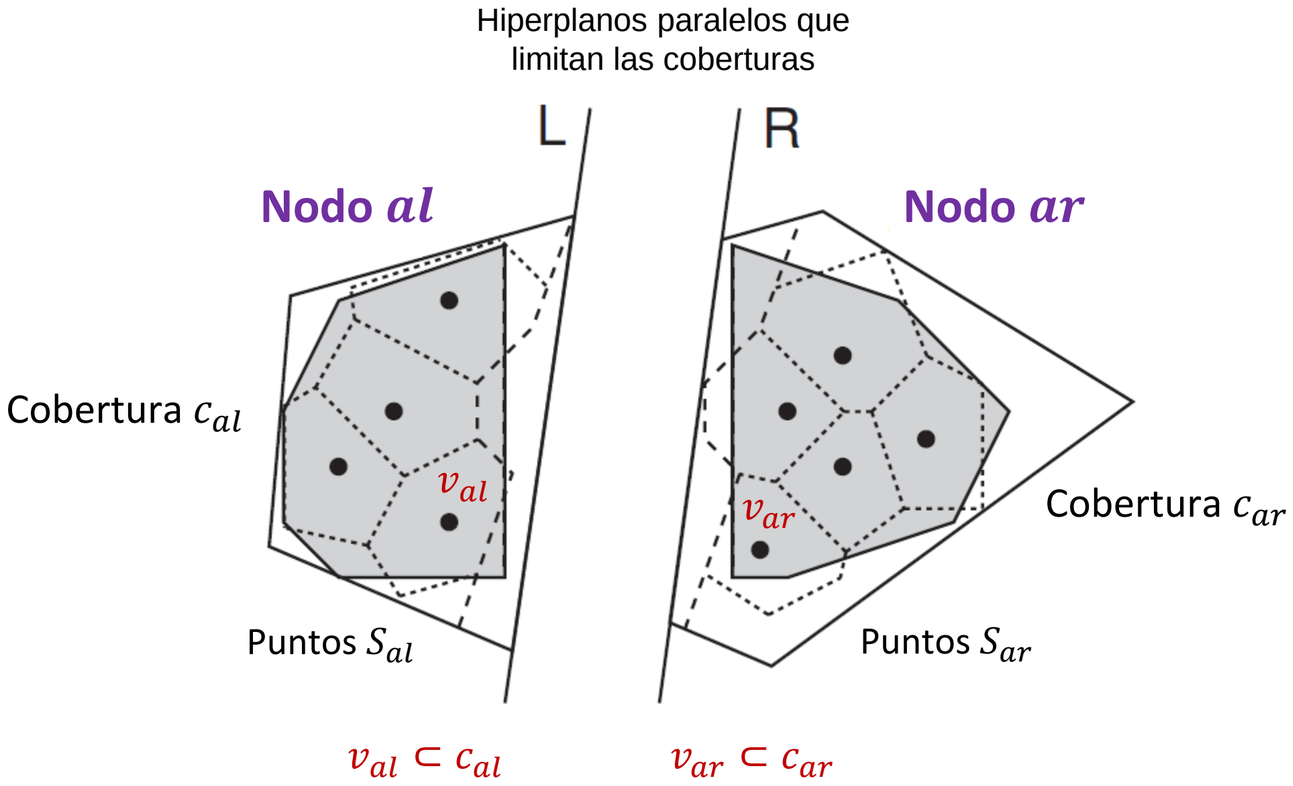
OS-*Tree*

(Overlapped-Split tree)



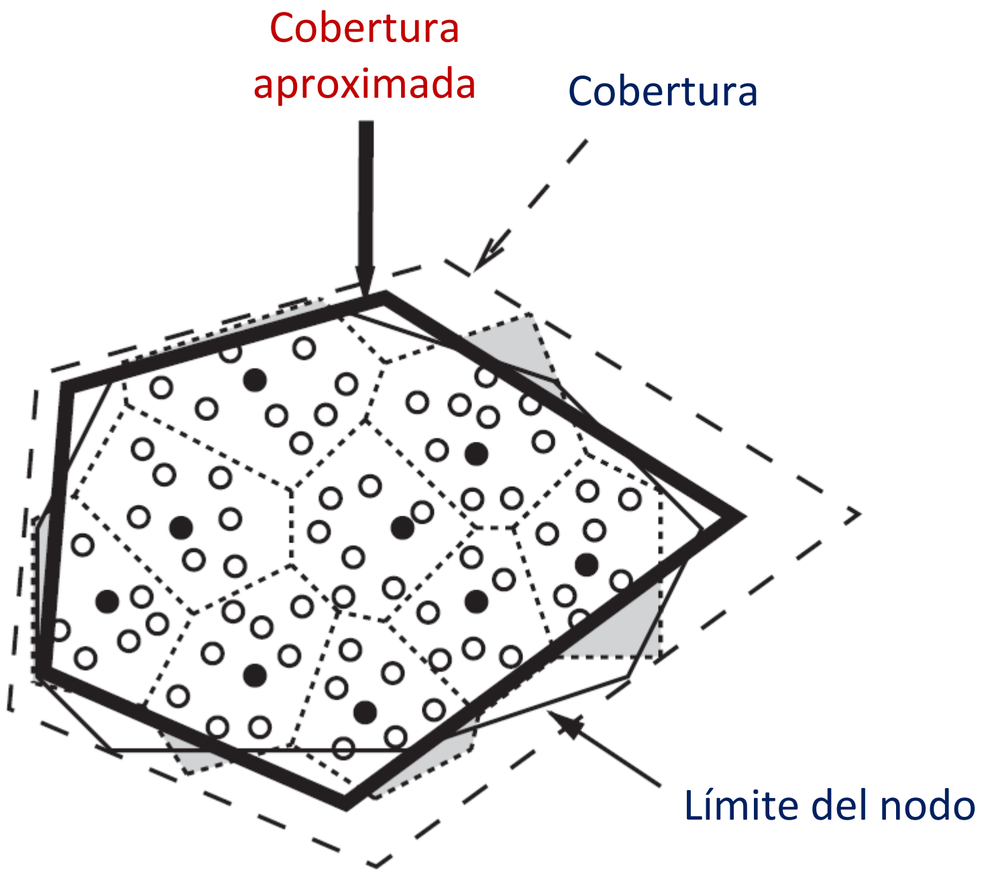
Songrit Maneewongvatana and David M. Mount (2001) "The analysis of a probabilistic approach to nearest neighbor searching". Algorithms and Data Structures: 7th International Workshop, WADS 2001 Providence, RI, USA, August 8–10, 2001 Proceedings (pp. 276-286). Berlin, Heidelberg: Springer Berlin Heidelberg.

OS-*Tree*



Songrit Maneewongvatana and David M. Mount (2001) "The analysis of a probabilistic approach to nearest neighbor searching". Algorithms and Data Structures: 7th International Workshop, WADS 2001 Providence, RI, USA, August 8–10, 2001 Proceedings (pp. 276-286). Berlin, Heidelberg: Springer Berlin Heidelberg.

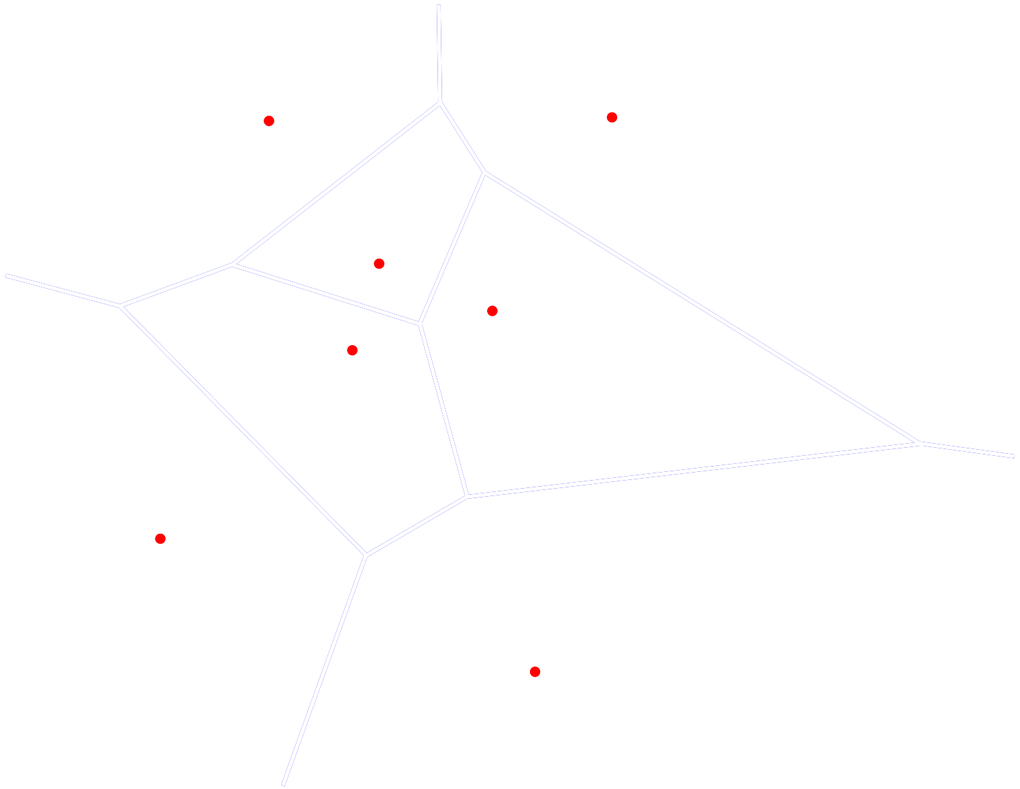
Probably correct *OS-Tree*



- Puntos de datos (S)
- Puntos de entrenamiento (T)

Songrit Maneewongvatana and David M. Mount (2001) "The analysis of a probabilistic approach to nearest neighbor searching". Algorithms and Data Structures: 7th International Workshop, WADS 2001 Providence, RI, USA, August 8–10, 2001 Proceedings (pp. 276-286). Berlin, Heidelberg: Springer Berlin Heidelberg.

Probably correct *OS-Tree*



1) Elegir los puntos que componen los subconjuntos S_{al} y S_{ar}

Buscamos el plano ortogonal a la dirección de la mayor variación de los datos.

PCA!

2) Elegir un plano de partición I_a para la celda correspondiente a a.

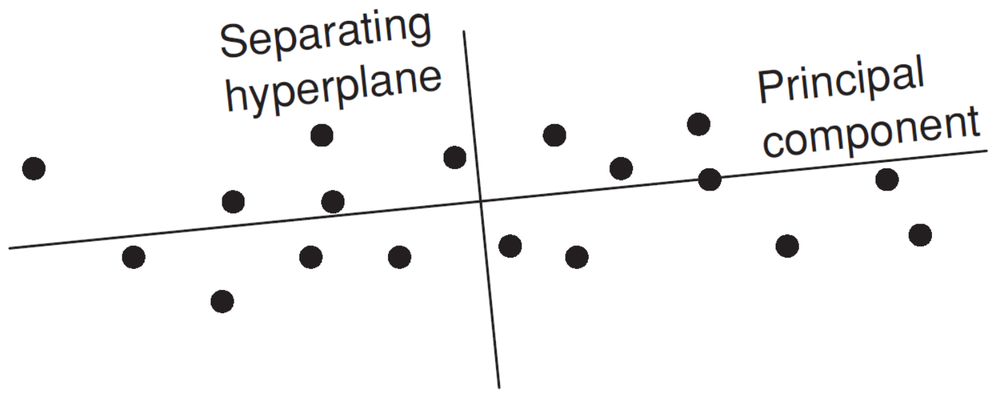
Minimizar el grado en que las regiones de Voronoi se superponen al lado contrario

SVM!

Songrit Maneewongvatana and David M. Mount (2001) "The analysis of a probabilistic approach to nearest neighbor searching". Algorithms and Data Structures: 7th International Workshop, WADS 2001 Providence, RI, USA, August 8–10, 2001 Proceedings (pp. 276-286). Berlin, Heidelberg: Springer Berlin Heidelberg.

Probably correct *OS-Tree*

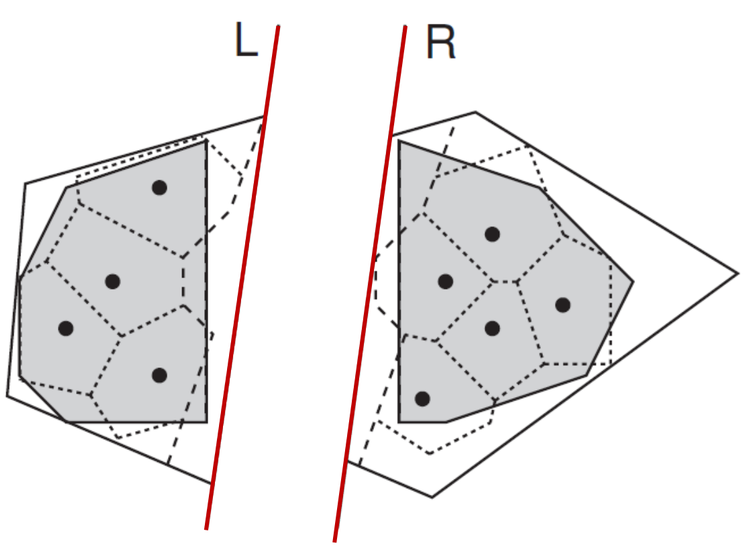
1) Elegir los puntos que componen los subconjuntos S_{al} y S_{ar}

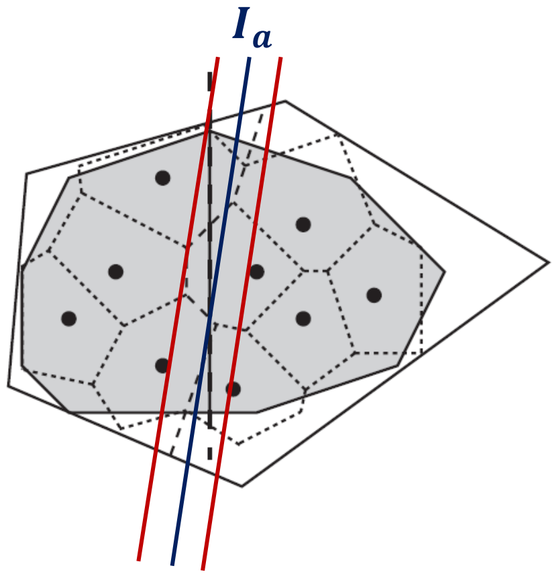


Songrit Maneewongvatana and David M. Mount (2001) "The analysis of a probabilistic approach to nearest neighbor searching".
Algorithms and Data Structures: 7th International Workshop, WADS 2001 Providence, RI, USA, August 8–10, 2001 Proceedings (pp. 276-286). Berlin, Heidelberg: Springer Berlin Heidelberg.

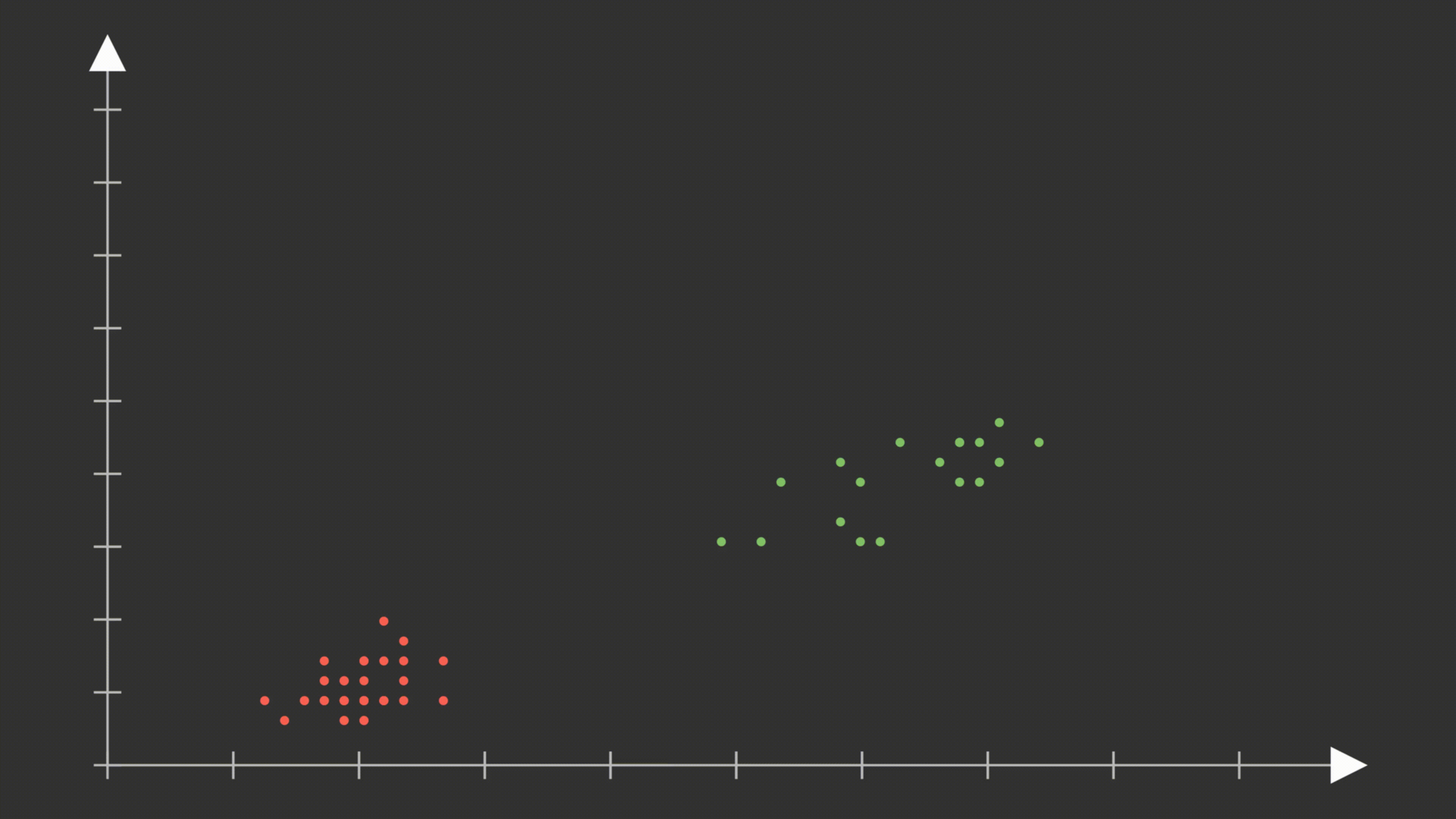
Probably correct *OS-Tree*

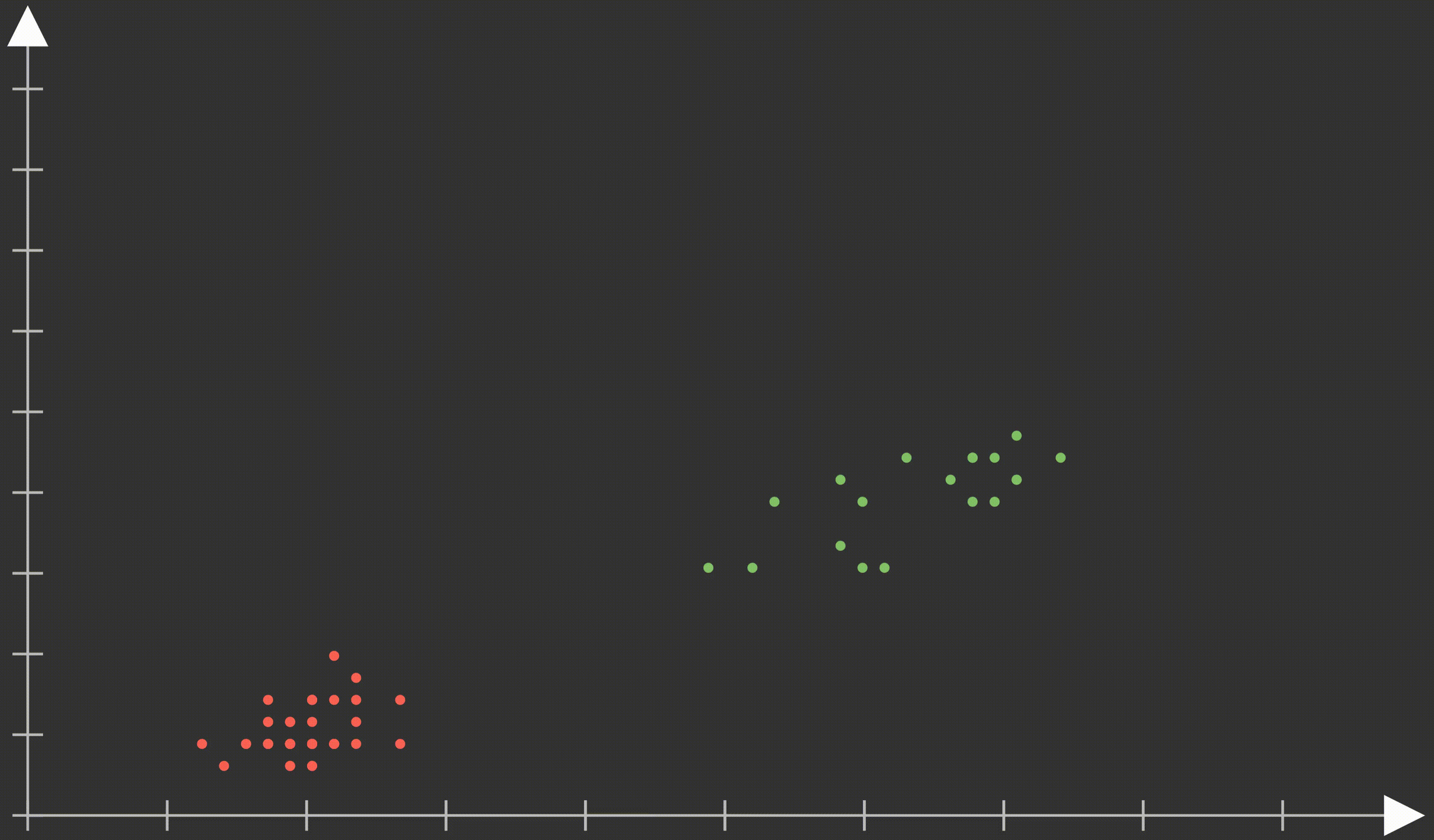
2) Elegir un plano de partición I_a para la celda correspondiente a a

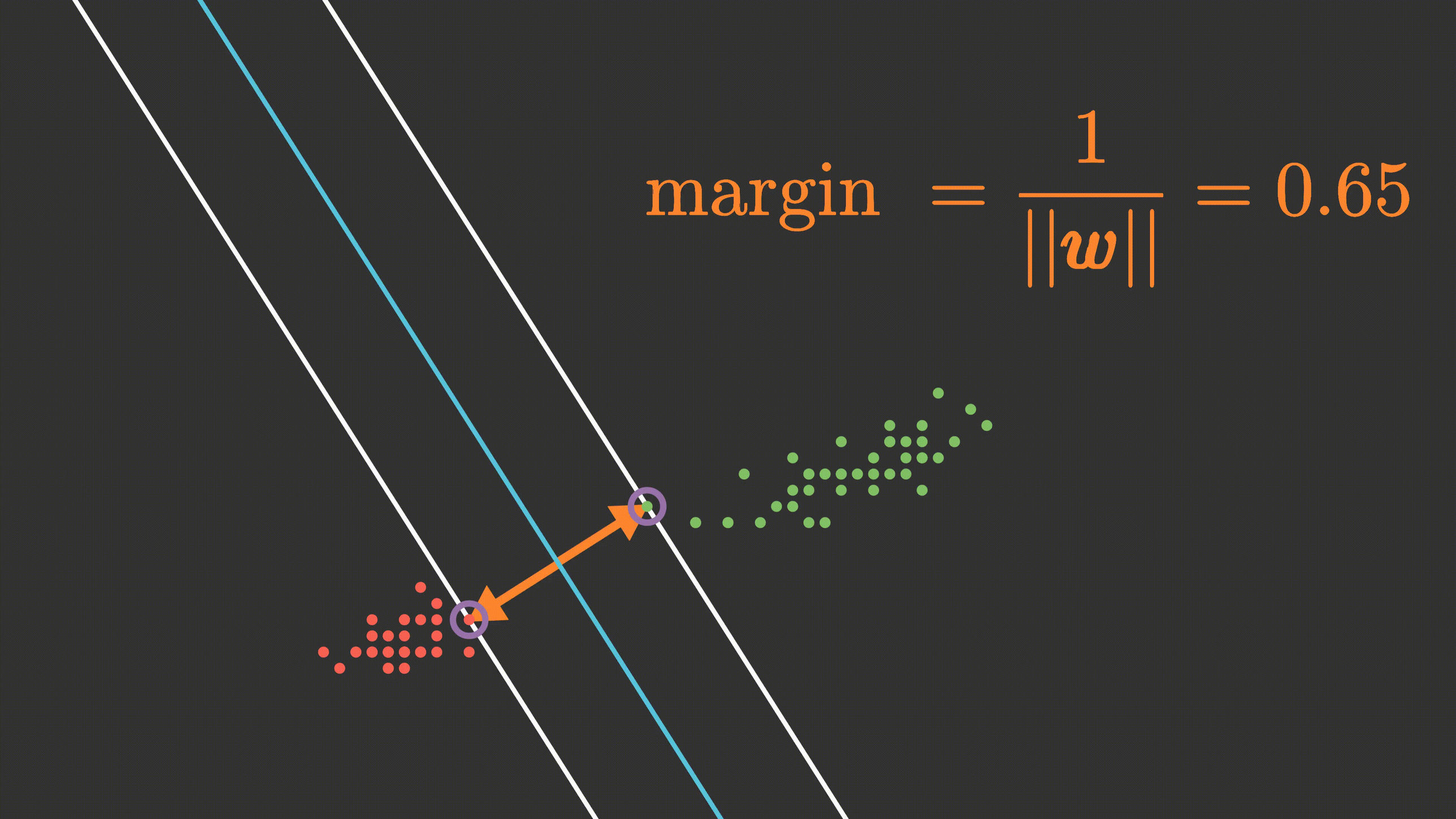




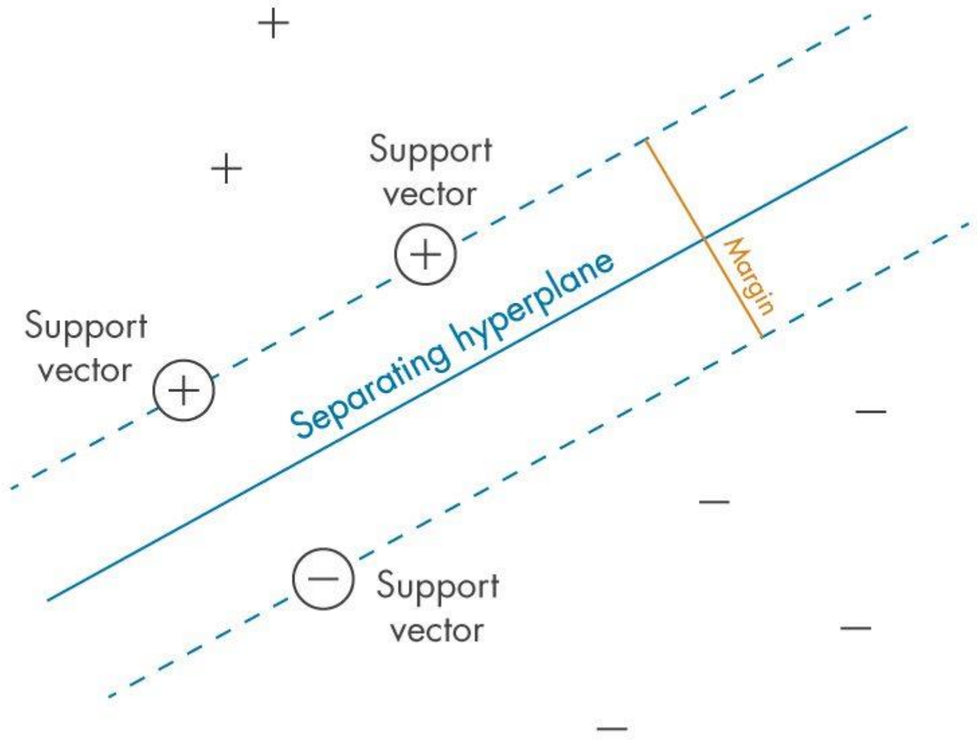
Songrit Maneewongvatana and David M. Mount (2001) "The analysis of a probabilistic approach to nearest neighbor searching". Algorithms and Data Structures: 7th International Workshop, WADS 2001 Providence, RI, USA, August 8–10, 2001 Proceedings (pp. 276-286). Berlin, Heidelberg: Springer Berlin Heidelberg.





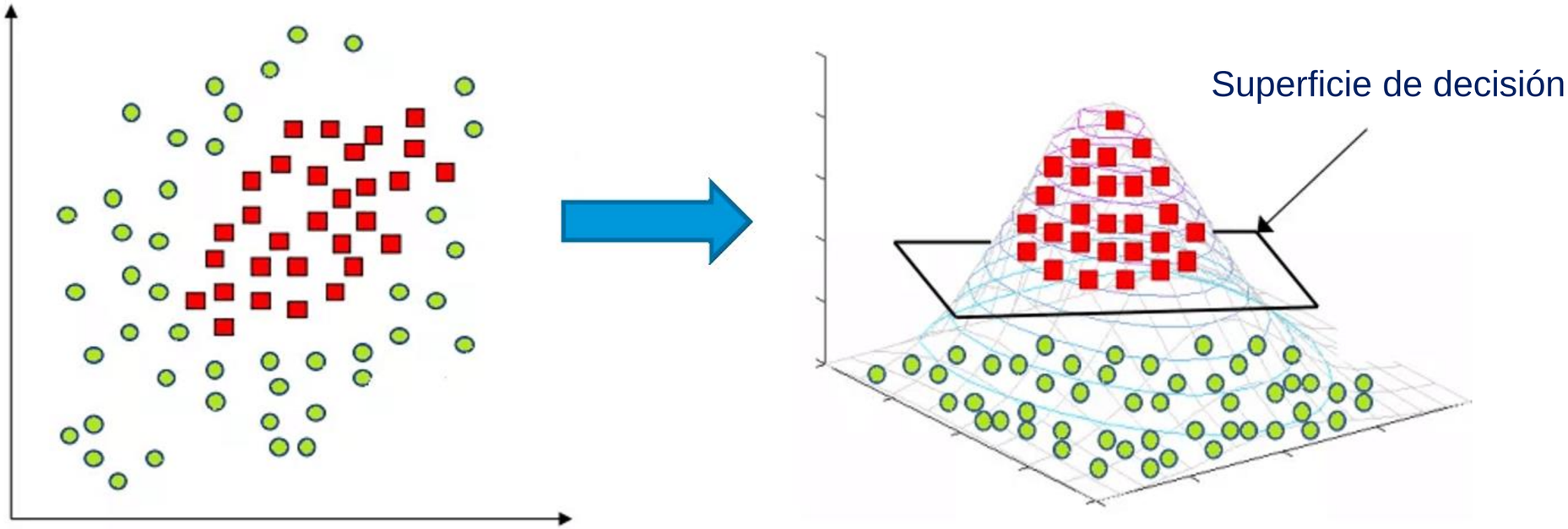

$\text{margin} = \frac{1}{||\boldsymbol{w}||} = 0.65$

Support Vector *Machine (SVM)*



Corinna Cortes and Vladimir Vapnik (1995) "Support-vector networks. Machine learning".
Machine Learning, 20, 273-297.

Support Vector *Machine (SVM)*



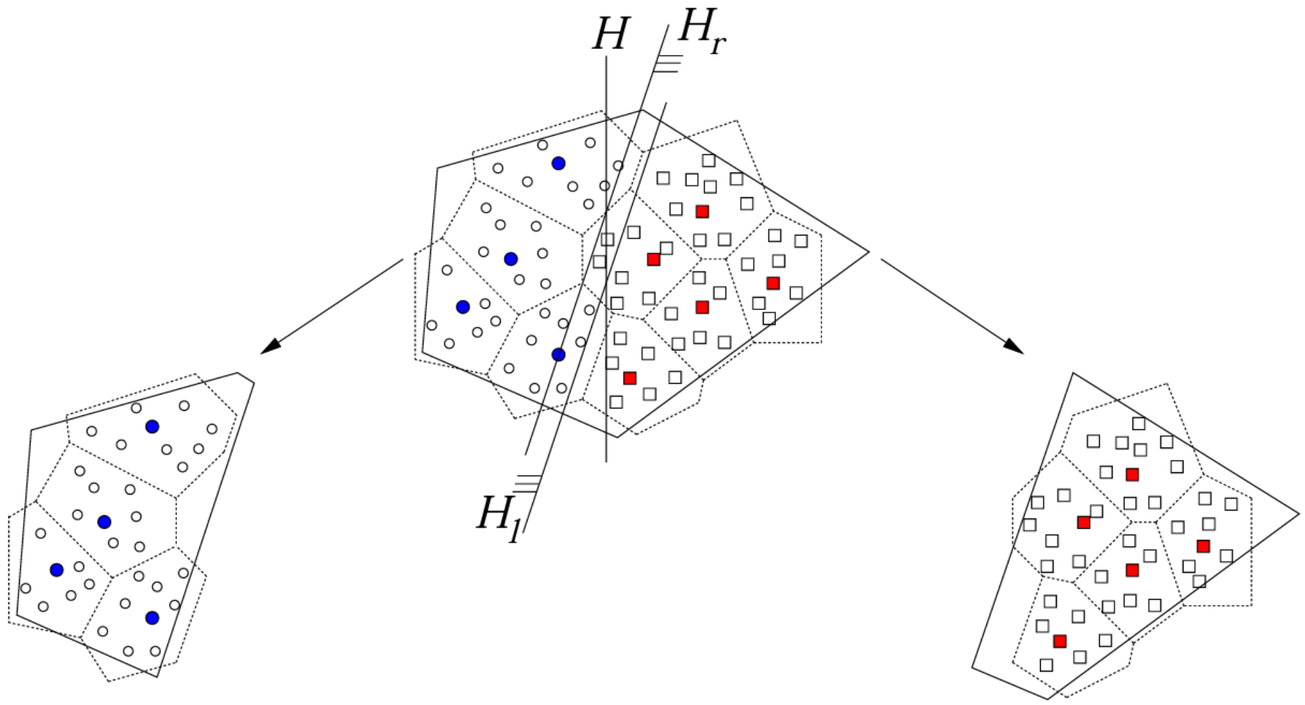
Corinna Cortes and Vladimir Vapnik (1995) "Support-vector networks. Machine learning". Machine Learning, 20, 273-297.

Support Vector *Machine (SVM)*

Tipo de SVM	Kernel de Mercer	Descripción
Función de base radial (RBF) o gaussiana	$K(x_1, x_2) = \exp\left(-\frac{\|x_1 - x_2\|^2}{2\sigma^2}\right)$	Aprendizaje de una clase. σ representa la anchura del kernel.
Lineal	$K(x_1, x_2) = x_1^\mathsf{T} x_2$	Aprendizaje de dos clases.
Polinómica	$K(x_1, x_2) = \left(x_1^\mathsf{T} x_2 + 1\right)^\rho$	ρ representa el orden del polinomio.
Sigmoide	$K(x_1, x_2) = \tanh\left(\beta_0 x_1^\mathsf{T} x_2 + \beta_1\right)$	Representa un kernel de Mercer solo para determinados valores β_0 y β_1.

Corinna Cortes and Vladimir Vapnik (1995) "Support-vector networks. Machine learning".
Machine Learning, 20, 273-297.

Probably correct *OS-Tree*



Songrit Maneewongvatana and David M. Mount (2001) "The analysis of a probabilistic approach to nearest neighbor searching". Algorithms and Data Structures: 7th International Workshop, WADS 2001 Providence, RI, USA, August 8–10, 2001 Proceedings (pp. 276-286). Berlin, Heidelberg: Springer Berlin Heidelberg.

Probably correct *OS-Tree*

Probabilidad de Fallo

$$\frac{|S|}{|T|}$$ Cuando $|T|$ es grande

Songrit Maneewongvatana and David M. Mount (2001) "The analysis of a probabilistic approach to nearest neighbor searching". Algorithms and Data Structures: 7th International Workshop, WADS 2001 Providence, RI, USA, August 8–10, 2001 Proceedings (pp. 276-286). Berlin, Heidelberg: Springer Berlin Heidelberg.

Probably correct *OS-Tree*

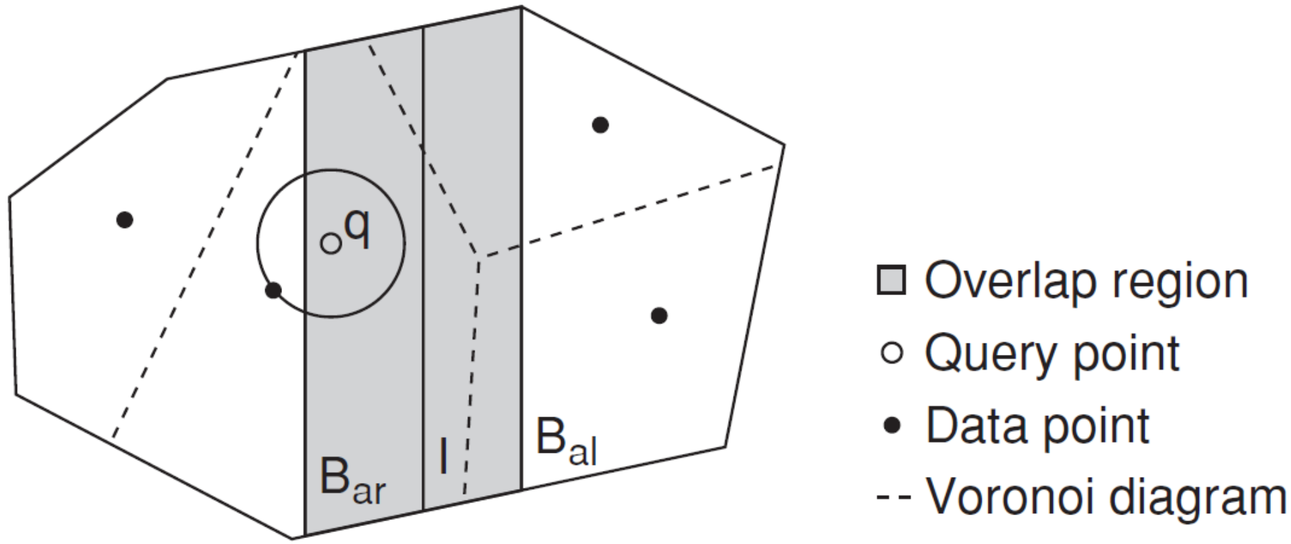
Este algoritmo es útil una cantidad media de dimensiones $(10 - 30)$

En estas dimensiones hay una buena aproximación

Songrit Maneewongvatana and David M. Mount (2001) "The analysis of a probabilistic approach to nearest neighbor searching".
Algorithms and Data Structures: 7th International Workshop, WADS 2001 Providence, RI, USA, August 8–10, 2001 Proceedings (pp. 276-286). Berlin, Heidelberg: Springer Berlin Heidelberg.

Probably correct *OS-Tree*

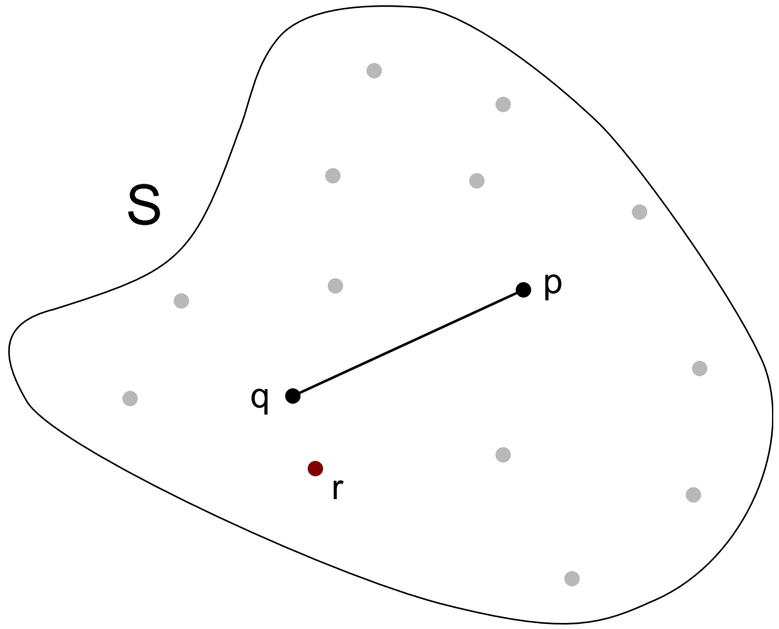
k vecinos más cercanos



Songrit Maneewongvatana and David M. Mount (2001) "The analysis of a probabilistic approach to nearest neighbor searching". Algorithms and Data Structures: 7th International Workshop, WADS 2001 Providence, RI, USA, August 8–10, 2001 Proceedings (pp. 276-286). Berlin, Heidelberg: Springer Berlin Heidelberg.

2. Approximate Voronoi Diagram (AVD)

ϵ-nearest neighbor *(ϵ-NN)*



Candidato a vecino más cercano o'

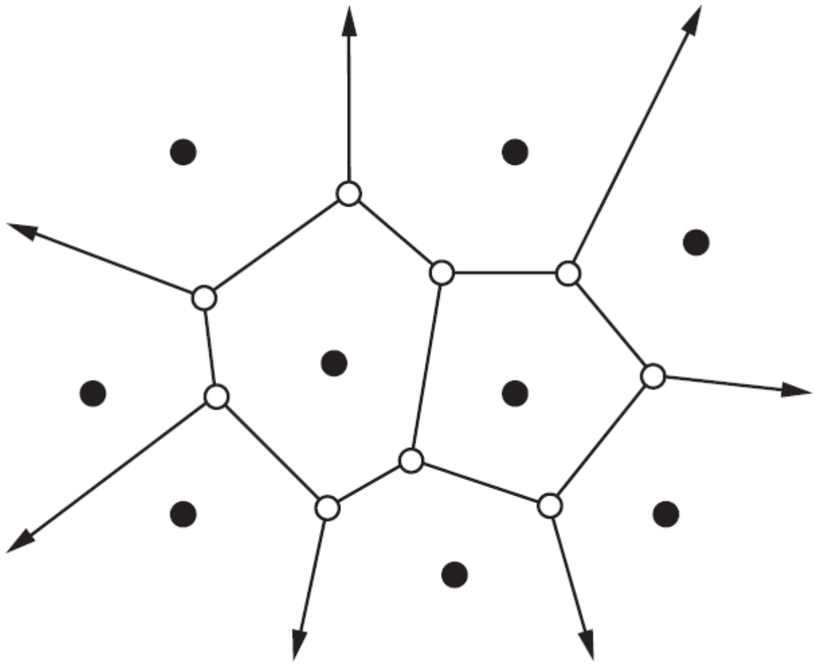
Vecino más cercano real: o

El punto p es ϵ-NN de q si:

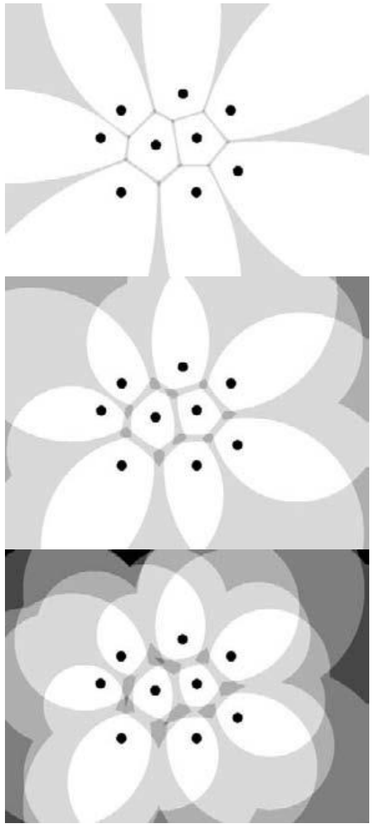
$$d(q, o') \leq (1 + \epsilon) \cdot d(q, o)$$

AVD

(Approximate Voronoi Diagram)



Voronoi Diagram



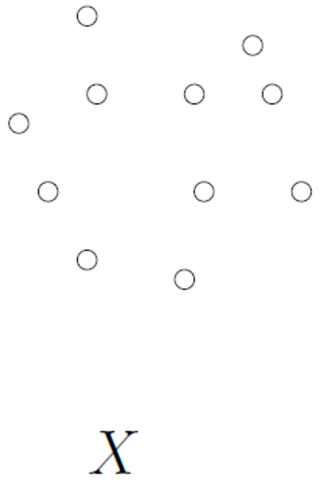
ϵ=0.1

ϵ=0.3

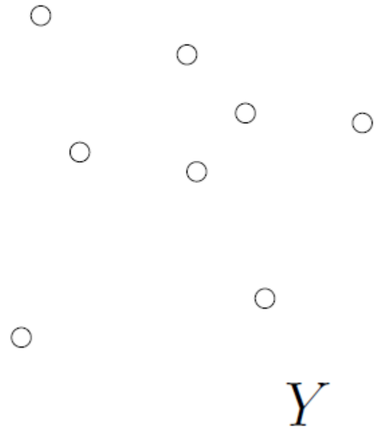
ϵ=0.5

Sariel Har-Peled (2001) "A practical approach for computing the diameter of a point set".
Proceedings of the seventeenth annual symposium on Computational geometry (pp. 177-186).

Well-Separated Pair *Decomposition (WSPD)*

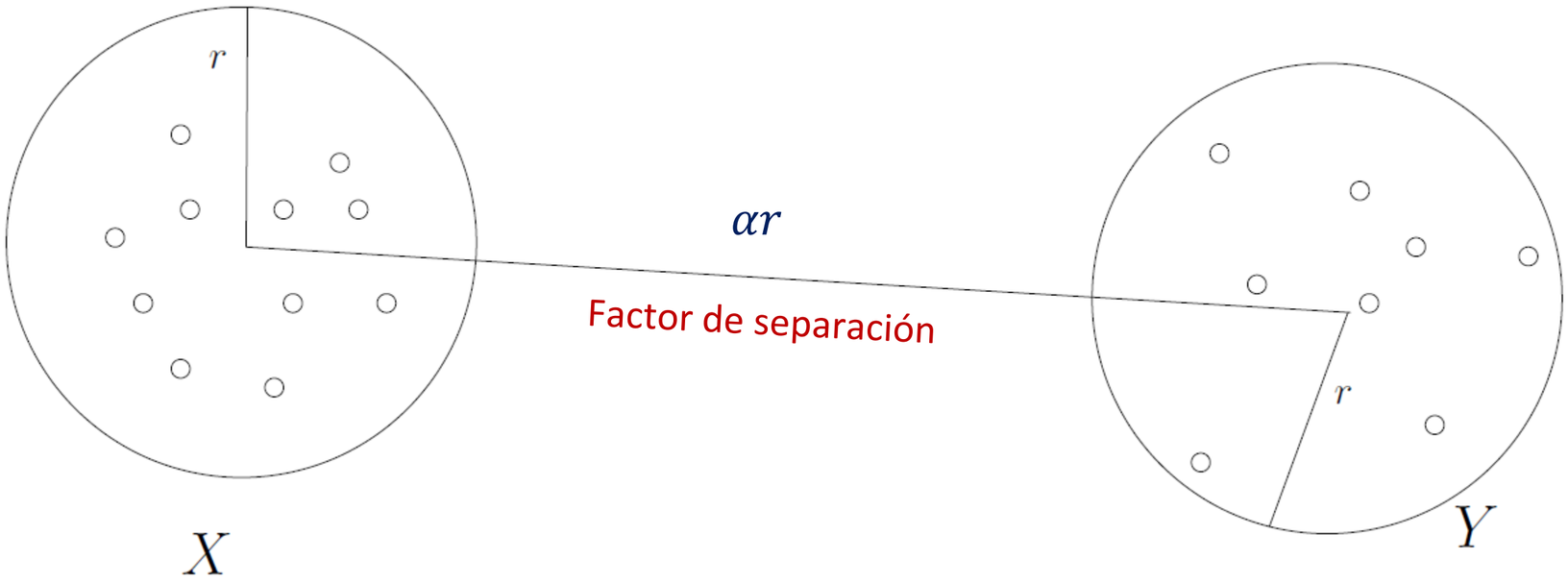


X

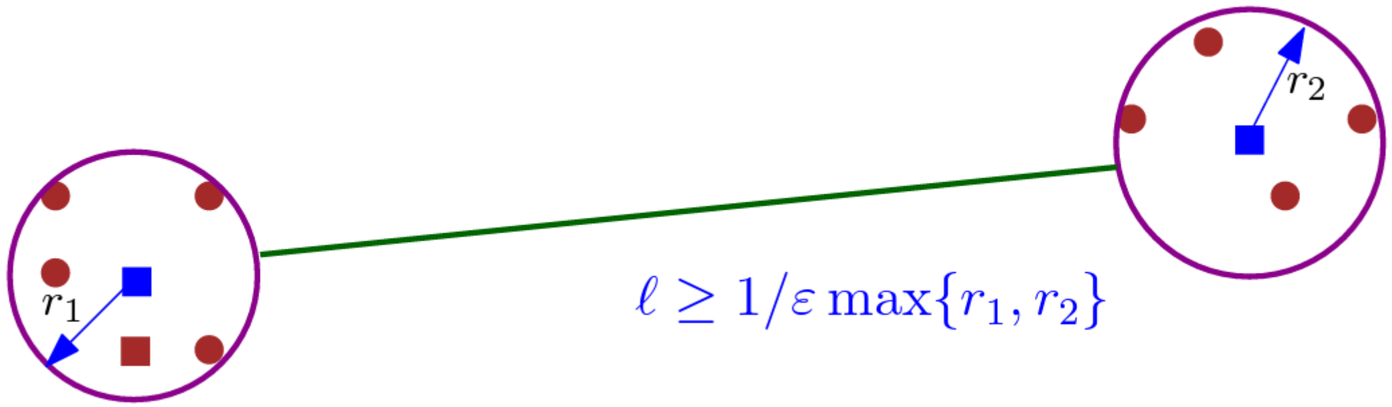


Y

Well-Separated Pair *Decomposition (WSPD)*

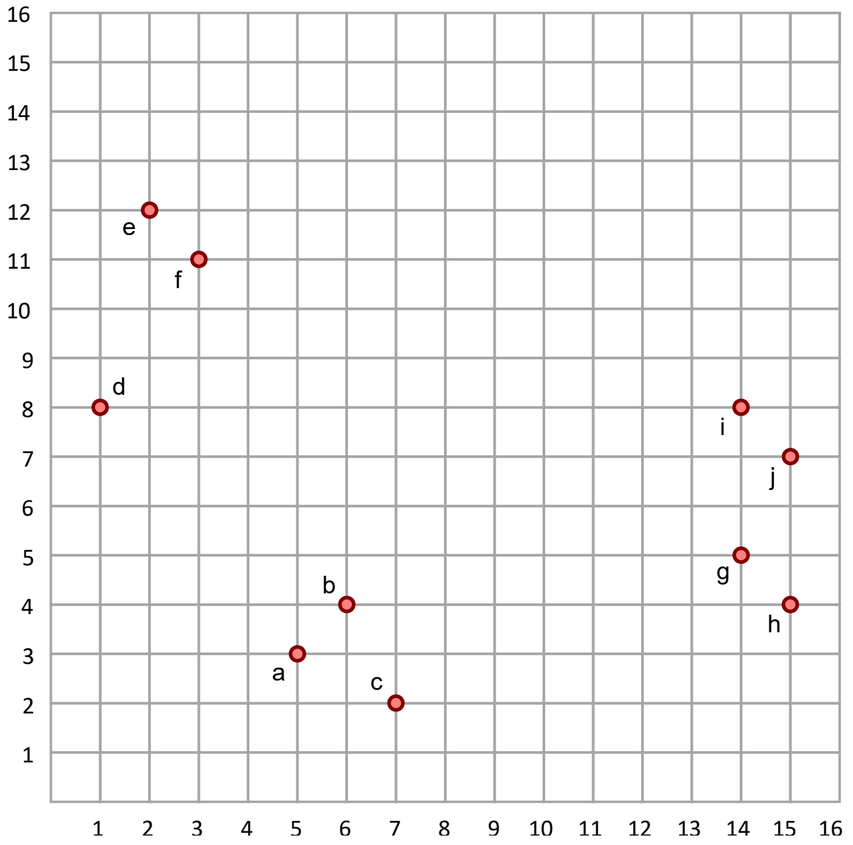


Well-Separated Pair *Decomposition (WSPD)*



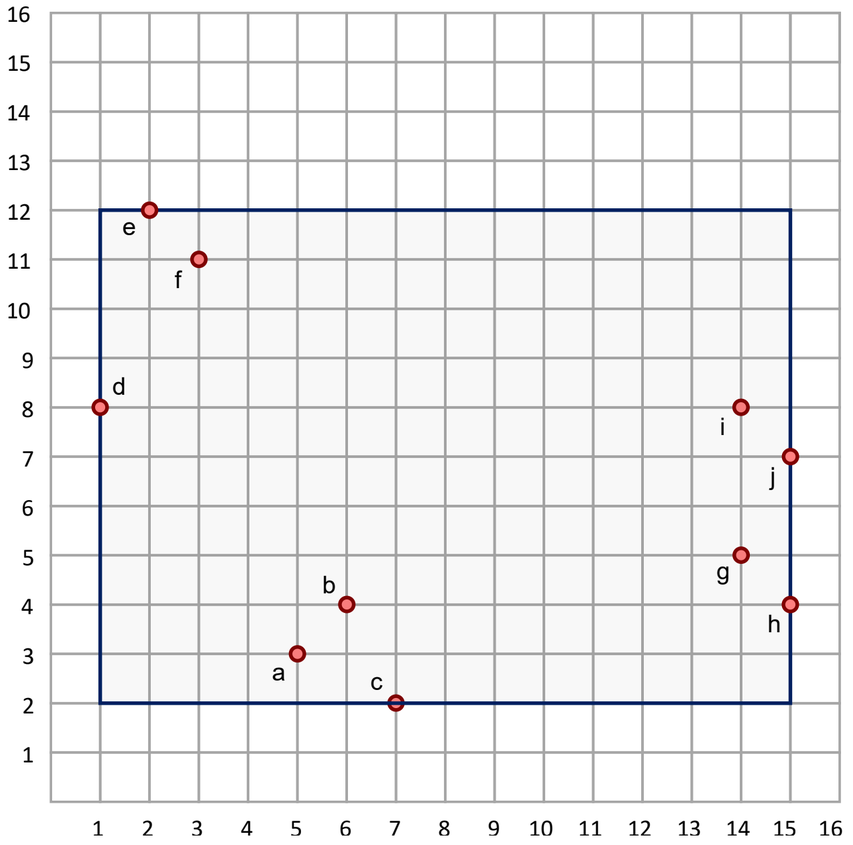
Fair split *tree*

Dividir el espacio recursivamente por la dimensión más grande del MBB.
Construir un árbol binario a partir de las particiones.



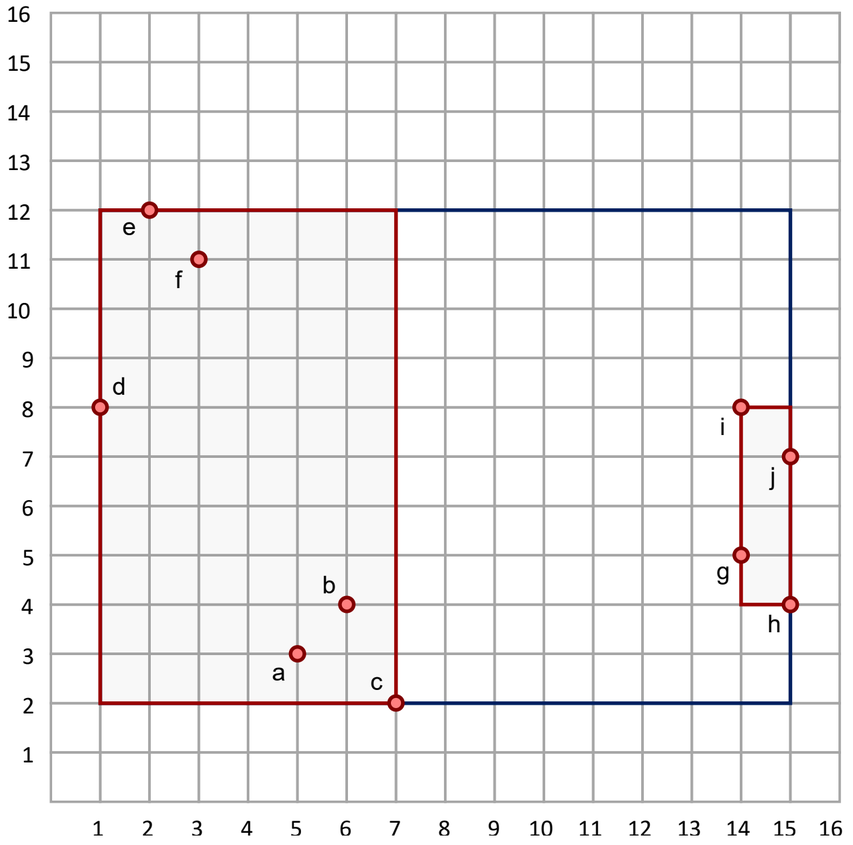
Fair split *tree*

Dividir el espacio recursivamente por la dimensión más grande del MBB.
Construir un árbol binario a partir de las particiones.



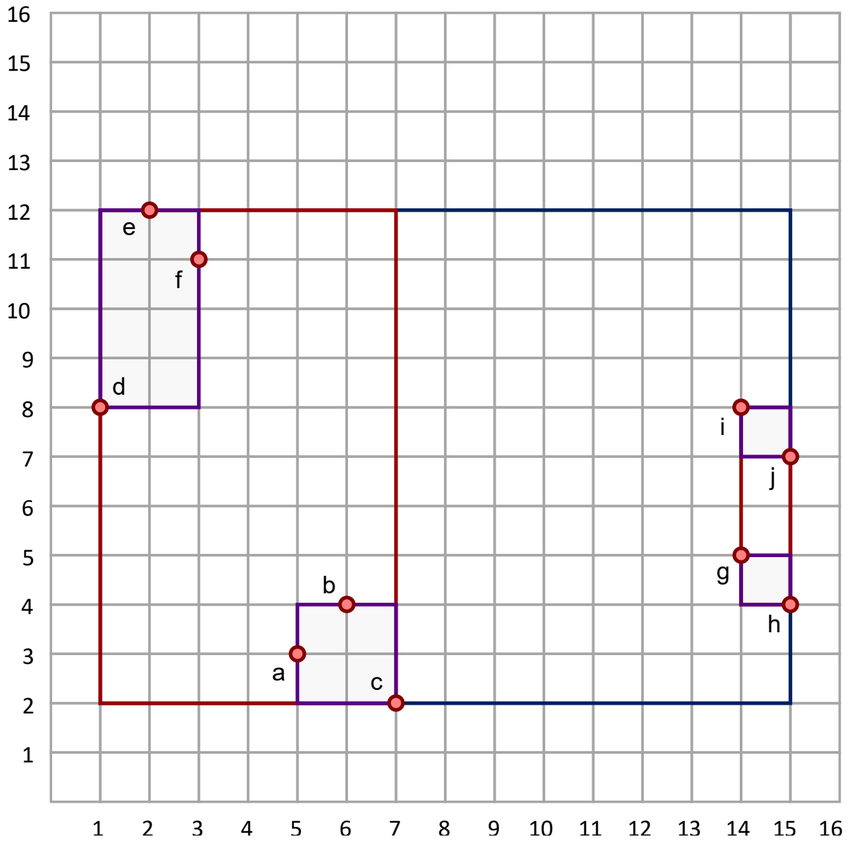
Fair split *tree*

Dividir el espacio recursivamente por la dimensión más grande del MBB.
Construir un árbol binario a partir de las particiones.



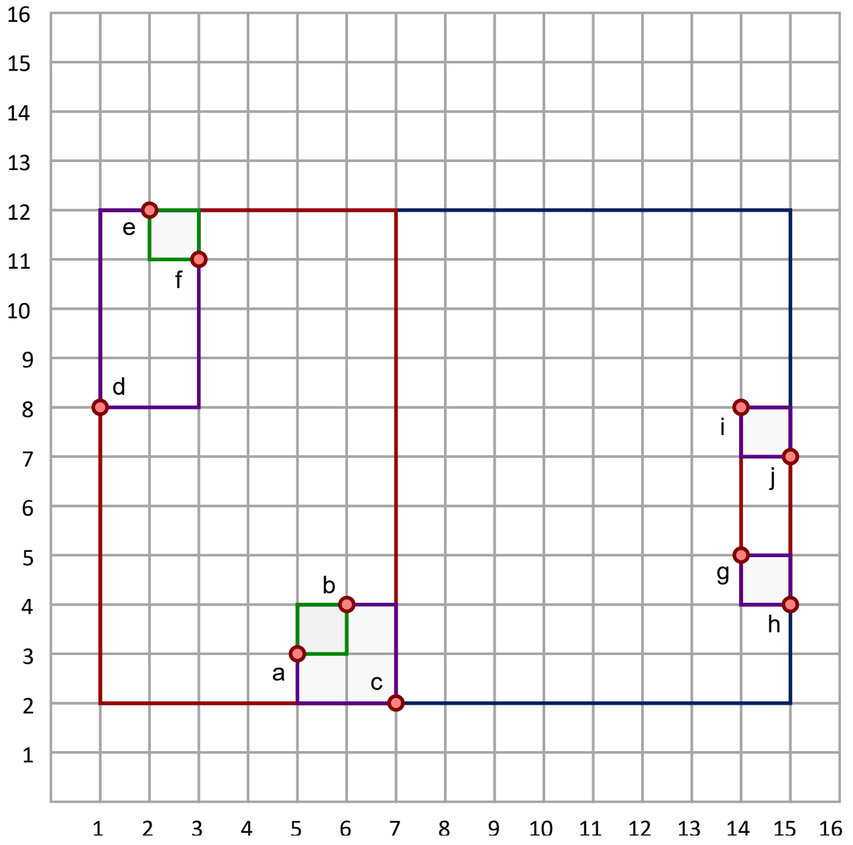
Fair split *tree*

Dividir el espacio recursivamente por la dimensión más grande del MBB.
Construir un árbol binario a partir de las particiones.

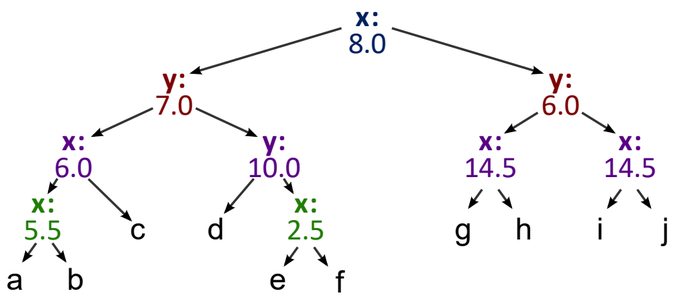


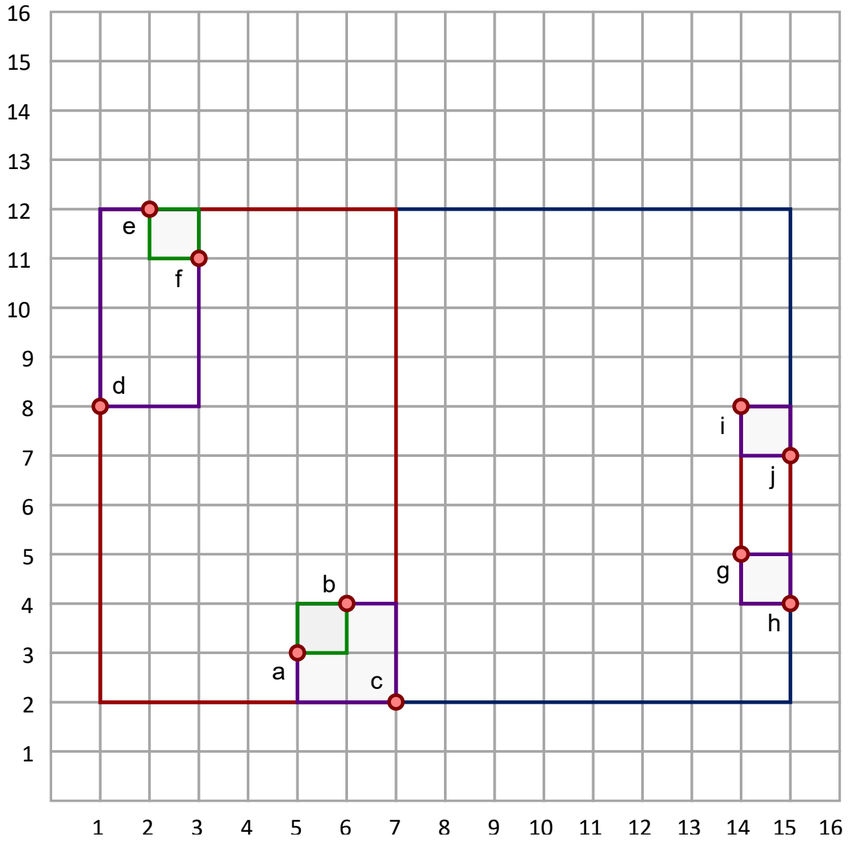
Fair split *tree*

Dividir el espacio recursivamente por la dimensión más grande del MBB.
Construir un árbol binario a partir de las particiones.

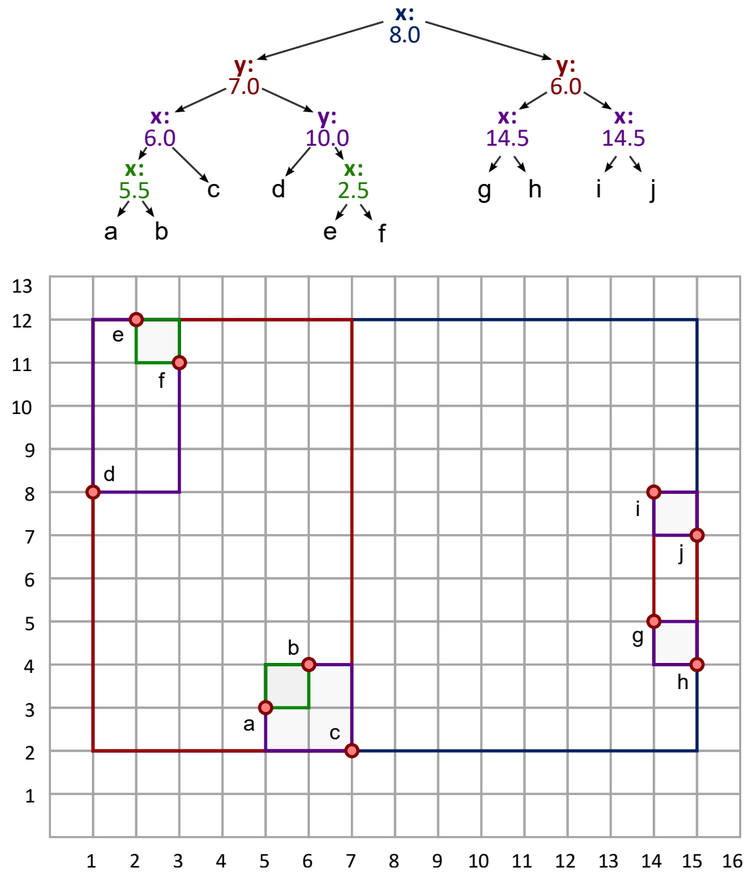


Fair split *tree*





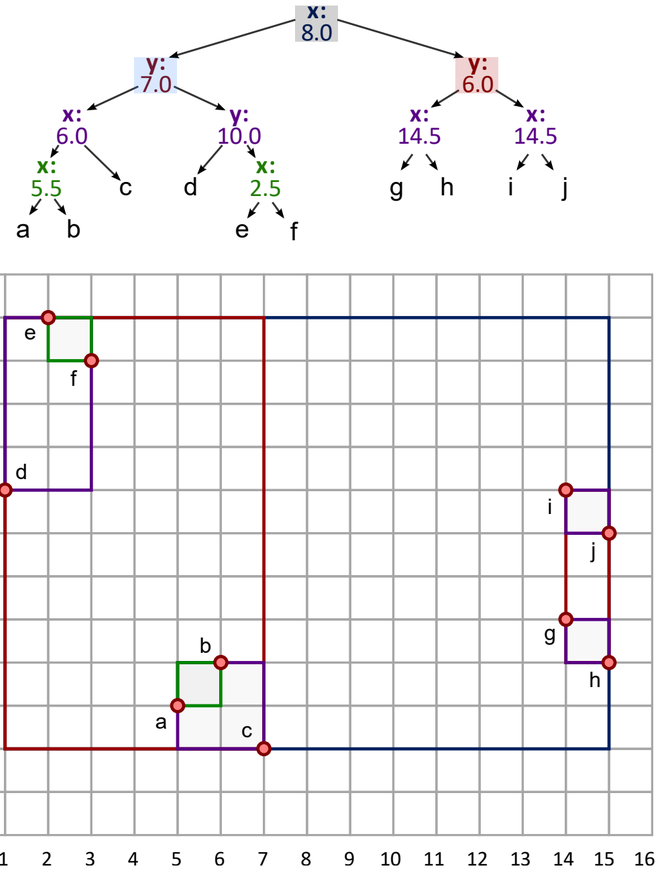
Well-Separated Pair *Decomposition (WSPD)*



Revisar los hijos de todos los nodos internos

Verificar si son WSP

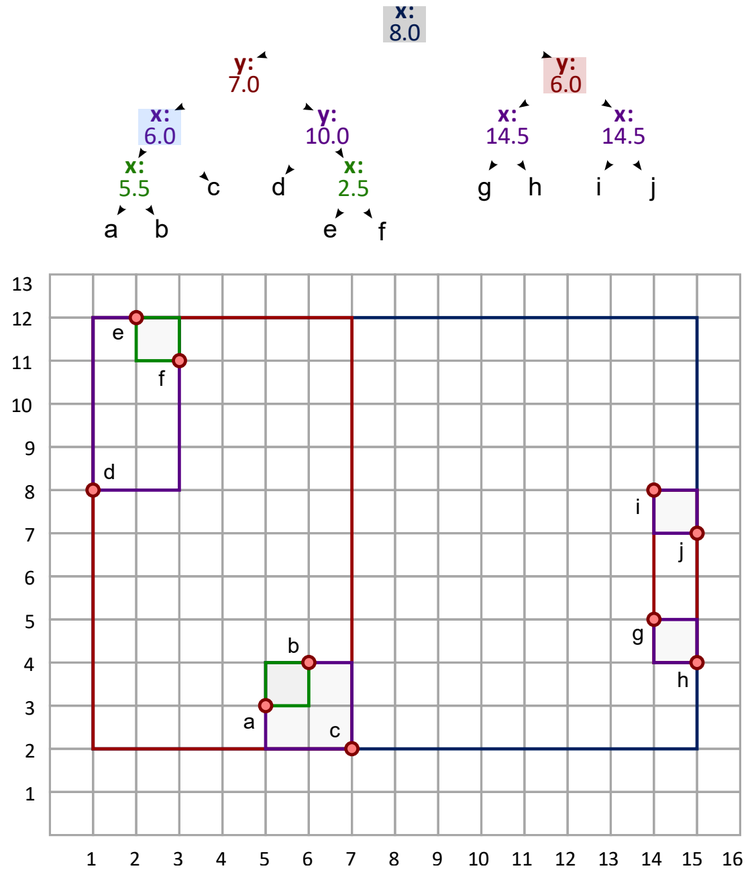
Well-Separated Pair *Decomposition (WSPD)*



Revisar los hijos de todos los nodos internos

Verificar si son WSP

Well-Separated Pair *Decomposition (WSPD)*

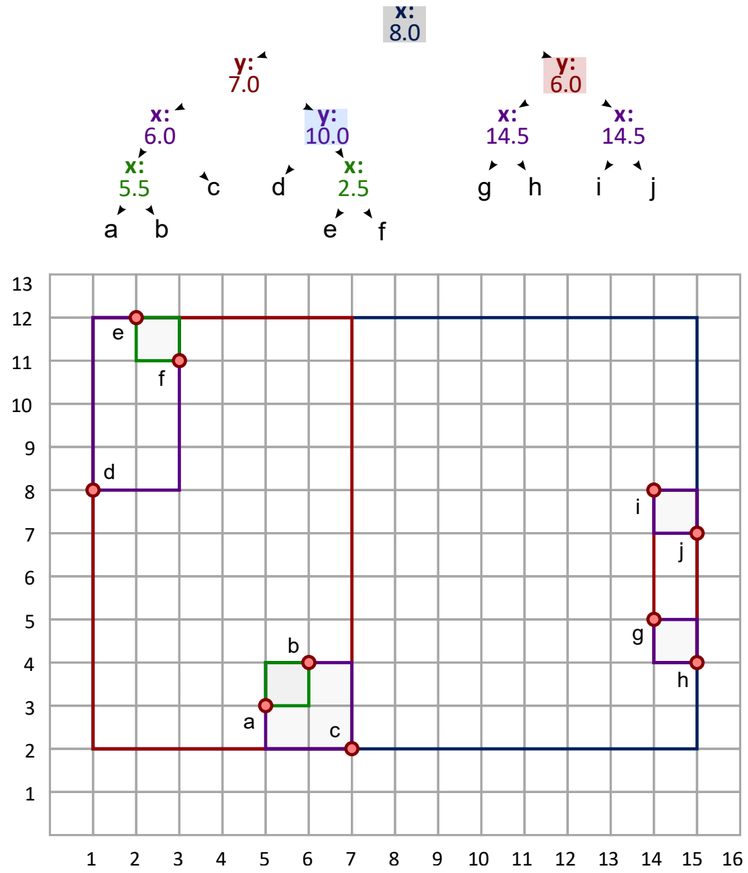


Revisar los hijos de todos los nodos internos

Verificar si son WSP

$$\{a, b, c\}, \{g, h, i, j\}$$

Well-Separated Pair *Decomposition (WSPD)*

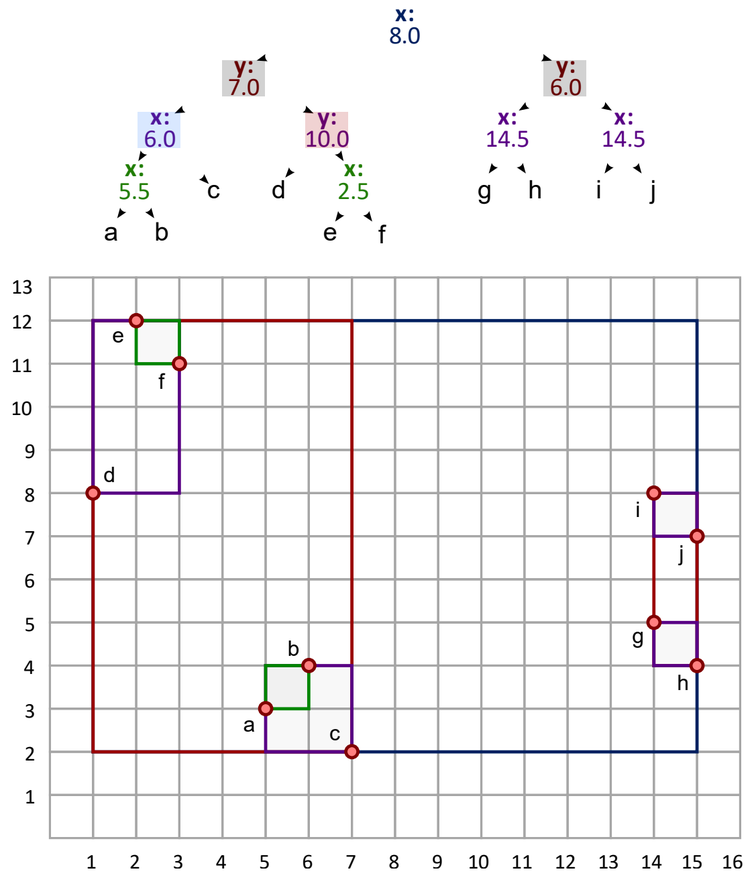


Revisar los hijos de todos los nodos internos

Verificar si son WSP

$$\{a, b, c\}, \{g, h, i, j\}$$
$$\{d, e, f\}, \{g, h, i, j\}$$

Well-Separated Pair *Decomposition (WSPD)*



Revisar los hijos de todos los nodos internos

Verificar si son WSP

$$\{a, b, c\}, \{g, h, i, j\}$$
$$\{d, e, f\}, \{g, h, i, j\}$$
$$\{a, b, c\}, \{d, e, f\}$$

UTEC

UNIVERSIDAD DE INGENIERÍA
Y TECNOLOGÍA