

UNIVERSIDAD DE VALLADOLID

ALGORITMO DIVIDE Y VENCERÁS

Par de puntos más cercano en espacio n-dimensional

Sergio García Prado

October 18, 2015

1 Introducción

El problema que se va a analizar se basa en encontrar los dos puntos más cercanos entre un conjunto de puntos pertenecientes en un espacio n -dimensional. Es una condición obligatoria que todos los puntos pertenezca a la misma dimensión ya que de no ser así no tendría sentido comparar sus distancias.

Existen distintos enfoques para resolver este problema. El mas simple pero a la vez menos eficiente de todos se basa en comparar todos los puntos con todos e ir guardando los dos que menor distancia tienen entre sí. Esta solución tiene un coste de $O(n^2)$ algo que podemos mejorar si nos damos cuenta de algunos detalles mediante algoritmos del tipo divide y vencerás.

Motivos por los que usar divide y vencerás

1. Los puntos mas cercanos en el espacio por la propia definición de cercanía van a estar en una región próxima del espacio. Este es el motivo por el cual nos podemos ahorrar comparar dos puntos que están muy alejados en el mapa.
2. Si encontramos un mínimo en un subconjunto del espacio y este lo es también para todos los subconjuntos que contienen a este, entonces habremos encontrado el mínimo de todo el espacio.

2 Solución

La solución que se ha escogido es la de realizar particiones binarias en el espacio recursivamente hasta tener subconjuntos de pequeño tamaño (En la implementación propuesta como ejemplo se ha fijado en conjuntos de 10 puntos.) para después compararlos utilizando un algoritmo boraz que nos asegura el mínimo local de cada subconjunto. Lo siguiente es quedarse con el mínimo de las dos particiones binarias y analizar los puntos que se quedaron en la frontera de las dos particiones ya que puede darse el caso de que el par de puntos con distancia mínima contuviese el punto 1 en la particion 1 y el punto 2 en la partición 2. De no ser por la parte de combinación estos casos no se estudiarían y podría darse el caso de que el mínimo encontrado no fuera el real.

Las explicaciones se van a exponer en un espacio de 3 dimensiones pero estas son extrapolables cualquier número de dimensiones.

2.1 Divide

Lo que intentamos conseguir al dividir el espacio en particiones binarias recursivamente es agrupar los puntos que están más próximos para así intentar prescindir del mayor número de comparaciones innecesarias. Para ello se pueden tomar distintos enfoques.

2.2 Venceras

3 Pseudocódigo

Algorithm 1 Compute sum of integers in array

```
1: procedure CLOSESTPAIR( $A$ )  
2:    $sum = 0$   
3:   for each integer  $i$  in  $A$  do  
4:      $sum = sum + i$   
5:   end for  
6:   Return  $sum$   
7: end procedure
```

4 Análisis de crecimiento