Traslaciones en $2n$ puntos[1]

M. Abellanas[2] S. Canales[3] A. García[4] G. Hernández[2] F.Martínez[2]
B. Palop[5] P. Ramos[6] V. Sacristán[7] J. Tejel[4] J. Urrutia[8]

Resumen

Se considera el siguiente problema: dado un conjunto de $2n$ puntos, averiguar si la mitad de ellos son los transformados de la otra mitad mediante una traslación. Se aborda también el problema combinatorio del número de traslaciones diferentes que pueden ser soluciones del problema.

1 Introducción

Dado un conjunto S de $2n$ puntos en $\mathbb{R}^d$, vamos a estudiar el problema de decidir si ese conjunto puede dividirse en dos conjuntos de n puntos A y B, de forma que A sea congruente con B mediante una traslación. El origen de este problema (para $d = 2$) es el siguiente: imaginemos que tenemos en una misma placa fotográfica las imágenes de n puntos (por ejemplo estrellas) y las de esos mismos puntos desplazados (imagen tomada instantes después) y queremos determinar cuales son los puntos originales.

El caso $d = 1$ es especialmente interesante. En primer lugar porque es un problema bastante natural: podemos imaginar que tenemos mezcladas medidas de n objetos realizadas con dos aparatos distintos, siendo la diferencia entre dos medidas del mismo objeto constante, y queremos determinar un valor para cada objeto. (Igualmente si la única diferencia entre los aparatos es que las unidades de medida son distintas, tomando logaritmos el problema es el mismo). En segundo lugar, porque el conjunto $S \subset \mathbb{R}^d$ puede dividirse en dos mitades A y B congruentes mediante una traslación si y sólo si las proyecciones de A y B sobre cualquier eje coordenado también son congruentes respecto de una traslación.

Creemos que este problema no ha sido estudiado previamente. Un problema relacionado es el de dados dos conjuntos A y B de n puntos en $\mathbb{R}^d$, determinar si A es congruente con B (permitiendo giros y traslaciones). Brass [1] da un algoritmo $O(n^{\lceil d/3 \rceil} \log n)$, para este problema, óptimo cuando $d \leq 3$, ya que una cota inferior es $\Omega(n \log n)$, incluso para $d = 1$. Una buena referencia incluyendo variantes y extensiones de este último problema es [3].

Por otra parte, parece natural considerar que en el problema puede haber puntos repetidos, es decir tenemos coordenadas (ó medidas) de $2n$ objetos, pero objetos distintos pueden tener las mismas coordenadas (ó medidas). Consideraremos que S es el conjunto de las coordenadas de esos $2n$ puntos y por tanto, en general, S será un multiconjunto de $\mathbb{R}^d$. Nuestro problema es decidir si S se puede dividir en dos mitades A y B (también multiconjuntos) de forma que $B = A + t$, para algún $t \in \mathbb{R}^d$. Cuando esto sea posible diremos que S admite la traslación t, y generalmente nos va a interesar hallar una o todas las traslaciones que admite S, y además para cada traslación t una partición de S en dos multiconjuntos $A = \{a_1, \ldots, a_n\}$ y $B = \{b_1, \ldots, b_n\}$, ordenados de forma que $b_i = a_i + t$. Llamaremos a esa partición una partición ordenada, o una partición ordenada admisible. Puesto que si S admite t, también admite $-t$, pero la partición es idéntica, en

[1]Trabajo desarrollado en el TGCC1 (Taller de Geometría Computacional,UPM Cercedilla 2001) Financiado parcialmente por DGES PB98-0933, CAM 07T/0014/2001, UAH2002/2001.

[2]Dpto. Matemática Aplicada, F. de Informática, UPM. {mabellanas, gregorio, fmartinez}@fi.upm.es

[3]Dpto. de Matemática Aplicada y Computación UPCO. scanales@einf.upco.es

[4]Dpto. de Métodos Estadísticos. Universidad de Zaragoza. {olaverri,jtejel}@posta.unizar.es

[5]Dpto. de Ciencias de la Computación. Universidad de Alcalá. b.palop@uah.es

[6]Dpto. de Matemáticas . Universidad de Alcalá. pedro.ramos@uah.es

[7]Dpto. de Matemática Aplicada 2. UPC. vera@ma2.upc.es

[8]Instituto de Matemáticas, Universidad Nacional Autónoma de México. urrutia@matem.unam.mx

lo que sigue supondremos que todas las traslaciones $t \in \mathbb{R}^d$ verifican $t \geq 0$, con $\geq$ el orden lexicográfico usual.

2 Sobre el número de traslaciones

Es fácil obtener un conjunto de puntos en $\mathbb{R}^d$ que admita las traslaciones $t_1, t_2, \ldots, t_h$. Tomamos un punto cualquiera $x_0 \in \mathbb{R}^d$ y formamos la familia de conjuntos de puntos $S_0 = \{x_0\}$, $S_i = S_{i-1} \cup (S_{i-1} + t_i)$, para $i = 1, \ldots h$. El conjunto S_h así obtenido tiene 2^h puntos (quizás alguno repetido) y por construcción admite las traslaciones $t_1, \ldots, t_h$.

En dirección contraria tenemos un resultado más débil:

Lema 1 *Si el conjunto S de puntos en $\mathbb{R}^d$ admite h traslaciones linealmente independientes contiene al menos 2^h puntos distintos.*

Demostración: Sean $t_1, \ldots, t_h$ las traslaciones que admite S, x_0 un punto cualquiera de S, $I_h = \{1, \ldots, h\}$ y J un subconjunto cualquiera de I_h. Veamos primero que S contiene un elemento de la forma

$$x_0 + \sum_{i=1}^{h} b_i t_i$$

donde $b_i = +1$ ó -1 si $i \in J$, y $b_i = 0$ en los demás casos. Esto se demuestra por inducción sobre el número de elementos del subconjunto J. Si $J = \emptyset$ es cierto por hipótesis. Si J contiene k índices, y es J' igual a J menos un índice cualquiera i, por inducción hay un elemento x de S que se obtiene sumando o restando a x_0 los vectores traslación de los índices de J'. Pero como S admite la traslación t_i también $x + t_i$ ó $x - t_i$ debe pertenecer a S, y es un punto de la forma buscada.

Notar que el resultado anterior es cierto para arbitrarios t_i, pero si además estos t_i son linealmente independientes, los puntos $x_0 + \sum_{i=1}^{h} b_i t_i$ obtenidos para los 2^h diferentes subconjuntos J son necesariamente distintos, de lo que se obtiene el resultado. ∎

Cuando no hay independencia lineal, muchos de los puntos de la forma $x_0 + \sum_{i=1}^{h} b_i t_i$ pueden coincidir y por tanto puede suceder que $t(n)$, el número de traslaciones que admite como máximo un conjunto S de $2n$ puntos, sea mayor que $\log_2(2n)$. De hecho, si $d = 1$, y si tomamos como S un conjunto de $2n$ números naturales consecutivos, entonces S admite la traslación de valor n. Pero si i es un divisor de n, también cada subsucesión de $2i$ puntos consecutivos admite la traslación de valor i, y por tanto esa traslación también es admitida por S. Por tanto tenemos que $t(n) \geq d(n)$, siendo $d(n)$ el número de divisores de n. Tomando $\overline{d}(n) = max_{i \leq n} d(i)$, esta función crece más rápidamente que $\log^k n$ para $k > 0$ arbitrariamente grande. De hecho es conocido que (ver [2])

$$\log \overline{d}(n) \sim \frac{(\log 2)(\log n)}{\log \log n}$$

3 Algoritmos

Conocemos que incluso para $d = 1$, cualquier algoritmo para determinar si un conjunto A de n puntos es congruente con otro B tiene una cota inferior $\Omega(n \log n)$. Pero notar, que si n es impar, si es S el conjunto formado por los n puntos de A y por los n de $B + t$, con t un valor suficientemente grande, (suponiendo que todos los valores son positivos basta tomar t mayor que el doble del máximo de los a_i), está claro que S admite una traslación si y sólo si A es congruente con B. Igualmente, si n es par, duplicando por ejemplo el dato mayor de A y el mayor de B, (para conseguir un número impar de puntos), y repitiendo la construcción anterior del conjunto S, de nuevo S admite una traslación si y sólo si A es congruente con B. Por tanto el problema de decidir si un conjunto S de $2n$ puntos admite o no una traslación tiene una complejidad $\Omega(n \log n)$.

Como hemos señalado en la introducción, un conjunto S en $\mathbb{R}^d$ admite una traslación t con partición A y B si y sólo si, las proyecciones de A y B sobre cualquier eje coordenado son congruentes mediante la traslación proyección de t. Por tanto para saber las traslaciones que admite un conjunto S de $\mathbb{R}^d$, primero, si es necesario, modificamos mediante un pequeño giro los datos iniciales de forma que las primeras componentes de puntos distintos sean distintas. Esto puede hacerse con complejidad $O(n \log n)$. A continuación calculamos las traslaciones admisibles respecto de la primera coordenada y sus correspondientes particiones ordenadas. Si la traslación t_i produce la partición ordenada A_i,B_i

en el conjunto de primeras coordenadas, como puntos distintos tienen primeras coordenadas distintas, añadiendo a cada elemento de A_i y B_i las otras $d-1$ coordenadas, obtenemos una partición ordenada A_i,B_i del conjunto S. Esa partición corresponderá a una traslación de S si y sólo si todos los valores $b_i - a_i$ (ahora de $\mathbb{R}^d$) coinciden. Así pues, comprobar que una partición ordenada admisible en una dimensión puede extenderse a dimensión d puede hacerse en tiempo lineal. Por tanto, si es $C_1(n)$ la complejidad de determinar para $2n$ puntos de $\mathbb{R}$ todas las traslaciones $t_1, \ldots, t_h$ que admite, y determinar además una partición ordenada A_i,B_i para cada traslación t_i, entonces podemos determinar las traslaciones y particiones ordenadas de $2n$ puntos en $\mathbb{R}^d$ con complejidad $O(C_1(n) + t(n)n)$, siendo $t(n)$ el número de traslaciones que como máximo podemos tener en el caso unidimensional.

Veamos ahora un algoritmo de complejidad $O(n^2)$ para resolver el problema unidimensional. Así pues $\Omega(n \log n) \leq C_1(n) \leq O(n^2)$.

Algoritmo 1.

Problema. Conocemos $2n$ valores $s_1, \ldots, s_{2n}$ y queremos calcular un conjunto maximal de valores distintos $t_1, \ldots, t_h$, todos ≥ 0, y para cada t_j una partición A_j,B_j de S, de forma que el elemento i-ésimo de B_j se obtiene sumando t_j al elemento i-ésimo de A_j.

Método. Ordenamos los $2n$ puntos y para cada valor $t = s_i - s_1$ $(i = 2, \ldots, n+1)$ comprobamos si t es una traslación admisible. Para ello usamos dos punteros $i1, i2$ que comienzan en 1 e i, y marcamos estos vértices. Tenemos que buscar si el primer vértice no marcado después del primer puntero, al sumarle t aparece después del segundo puntero. Si no aparece ese par esa traslación no es válida, en caso contrario marcamos los puntos de ese par y continuamos el proceso hasta marcar todos los puntos ó hasta encontrar un par que falla.

Complejidad. Hay que comprobar si son admisibles n posibles traslaciones y cada comprobación supone hasta $O(n)$ operaciones en el peor de los casos. De hecho, en el caso de n puntos equiespaciados se hacen del orden de n^2 operaciones. Por tanto su complejidad es $O(n^2)$.

Si el algoritmo anterior responde negativamente, es decir, no hay traslaciones admisibles, parece natural calcular los subconjuntos maximales y disjuntos A y B de S tales que $B = A + t$. El siguiente algoritmo resuelve este problema.

Algoritmo 2:

Problema. La entrada siguen siendo $2n$ números, quizás algunos repetidos, pero ahora en salida para cada distancia t entre dos de esos números calculamos subconjuntos ordenados y disjuntos A_t y B_t de forma que B_t es congruente con A_t mediante la traslación t, y además el cardinal de A_t (y B_t) es máximo entre los que verifican esa propiedad. En particular S admite la traslación t si y sólo si A_t tiene cardinal n.

Método. Ordenamos los $2n$ puntos de S y calculamos las $n(2n-1)$ distancias de cada punto con los siguientes a él. Ordenamos léxicograficamente todas las ternas (t, i, j) dónde t es la distancia entre el punto i-ésimo y el j-ésimo $(j > i)$. Supongamos que las distancias distintas que aparecen son $t_1, \ldots, t_m$ y que la distancia t_k aparece n_k veces. Notar que debe ser $\sum_{k=1}^{m} n_k = n(2n-1)$.

Supongamos que la distancia t_k aparece con las n_k parejas de índices $(i_1, j_1), \ldots, (i_{n_k}, j_{n_k})$. Queremos hallar un subconjunto P_k de estas parejas verificando que ningún índice se repite y con cardinal máximo (mirando ese conjunto de pares como un grafo, P_k es un emparejamiento de cardinal máximo). Así, los primeros índices de esas parejas de P_k definen el conjunto A, los segundos el conjunto B, y por construcción se verifica $B = A + t_k$, con A y B disjuntos y de cardinal máximo entre los que verifican lo anterior. Para formar P_k, suponemos inicialmente todos los índices sin marcar, y vamos explorando los pares (i_l, j_l), para $l = 1, \ldots, n_k$. Si los dos índices del par l-ésimo están sin marcar, los incluimos en la lista de pares P_k y los marcamos. Al finalizar, guardamos la lista P_k, y quitamos la marca a cada uno de los índices que aparecen en la lista P_k. Notar que formar esa lista P_k requiere $O(n_k)$ pasos. Hemos de realizar este proceso para cada una de las distancias distintas $t_1, \ldots, t_m$ que encontremos.

Corrección. Tenemos que demostrar que dados los pares (ordenados lexicográficamente)

$(i_1, j_1), \ldots, (i_{n_k}, j_{n_k})$, correspondientes a todos los puntos a una misma distancia t_k, el proceso anterior produce un subconjunto P_k sin índices repetidos y maximal entre los que cumplen esta condición. Para ello nos basta ver que siempre hay un subconjunto maximal incluyendo al primer par (i_1, j_1), pues de hecho nuestro algoritmo se reduce a elegir siempre el primer par en orden lexicográfico de una lista de pares admisibles (en cada paso son pares admisibles los que no tienen marcado ninguno de sus índices). Supongamos que (i_1, j_1) no aparece en un subconjunto maximal P_k de pares disjuntos. Si aparece un par (i_1, j_h) en P_k tiene que suceder que los puntos j_1 y j_h coincidan, y por tanto intercambiando j_h y j_1 en todos los pares de P_k obtenemos otro subconjunto maximal que incluye al par (i_1, j_1). Igualmente, si aparece el par (i_h, j_1), i_1 e i_h son el mismo punto y podemos intercambiar sus ocurrencias en P_k. Finalmente, si no estamos en ninguno de los dos casos anteriores, y puesto que P_k es maximal, la única posibilidad es que P_k contenga un par de la forma (j_1, i_h), y podemos cambiar ese par por el (i_1, j_1).

Complejidad. Se necesita ordenar las $O(n^2)$ distancias entre n puntos sobre una recta, lo que puede hacerse en $O(n^2 \log n)$. Es fácil comprobar que este problema es computacionalmente equivalente a la ordenación del conjunto $X+Y$, donde X e Y son listas de n enteros cada una. Si este problema de ordenación de $X+Y$ puede o no efectuarse en $O(n^2)$ operaciones es un problema abierto desde hace mucho tiempo. Se sabe al respecto que existe un árbol de decisión de profundidad $O(n^2)$ para el problema, o, equivalentemente, que $X+Y$ puede ordenarse usando sólo $O(n^2)$ comparaciones, pero esto no implica que deba existir un algoritmo de complejidad $O(n^2)$ para ordenar el conjunto. (Ver [4] [5] [6]). La ordenación de las distancias determina la complejidad del método, porque la exploración y marcado de la listas de n_k pares sólo requiere $O(n^2)$ pasos.

La complejidad del algoritmo 2 es mayor que la del algoritmo 1, pero da mucha mas información. En particular permite encontrar un subconjunto $S' \subset S$, con máximo número de elementos, tal que S' admite una traslación.

4 Cuestiones abiertas y extensiones

Hay dos destacadas cuestiones abiertas para el problema más simple de traslaciones en $\mathbb{R}$. La primera es calcular o acotar $t(n)$, el número máximo de traslaciones que admite un conjunto de $2n$ puntos. Conjeturamos que coincide con $d(n)$, el número de divisores de n, pero ni siquiera hemos demostrado que sea $o(n)$. Notar que si los $2n$ puntos los colocamos en un círculo, y consideramos como movimientos válidos giros sobre el centro, puede haber hasta $\Theta(n)$ giros admisibles.

La otra cuestión es la de encontrar un algoritmo con complejidad menor que $O(n^2)$, idealmente alcanzando la cota inferior $\Omega(n \log n)$, para el problema inicial.

Hay dos extensiones naturales para este problema. La primera es decidir si el conjunto S se puede dividir en dos mitades A y B congruentes (con giros y traslaciones). La segunda es decidir si se puede dividir S en tres partes, ó en general en k partes, congruentes mediante traslaciones. Para este último problema, una simple modificación del algoritmo 1, comprobando las $\binom{n}{k}$ posibles traslaciones del primer elemento, nos da un algoritmo de complejidad $O(\binom{n}{k}n)$.

Otro problema distinto es, dado S, buscar una traslación no nula t y un subconjunto A de cardinal máximo, de forma que el conjunto $A+t$ también aparezca en S. Parece natural exigir que cuando S sea un multiconjunto, si un elemento a_i aparece r veces en A, entonces $a_i + t$ tiene que aparecer al menos r veces en S. Es fácil modificar el algoritmo 2 para resolver este problema. Otro problema interesante es el de buscar una partición de S en dos mitades ordenadas A y B, de forma que el conjunto B sea lo mas parecido posible a $A+t$ para algún t. Si es $x_i = a_i - b_i$, podemos buscar las particiones ordenadas que hagan mínimo el rango de los valores x_i (ó alternativamente que hagan mínima su varianza). A diferencia de las variantes anteriores, este último problema parece NP-duro.

Referencias

[1] P. Brass, C. Knauer, Testing the congruence of d-dimensional point sets, *In Proc. 16th An-*

nu. ACM Symp. on Computational Geometry (2000), 310-314.

[2] S. Ramanujan, Highly composite numbers, *Proc. London Math. Soc.* (2) 14 (1915).

[3] H. Alt, L.J. Guibas, Discrete geometric shapes: Matching, interpolation, and approximation. *In J.R. Sack and J. Urrutia, editors, Handbook of Computational Geometry. Elsevier* (2000).

[4] H. Nagamochi, T. Ibaraki, Deterministic O(nm) time edge-splitting in undirected graphs, *J. Combinatorial Optimization*1 (1997), 5-46.

[5] T. Watanabe, A. Nakamura, Edge-connectivity augmentation problems, *J. Comput. System Sci.* 35 (1987), 96-144.

[6] T. Watanabe, A. Nakamura, A smallest augmentation to 3-connect a graph, *Disc. Appl. Math.* 28 (1990), 183-186.