Revista Digital
Scientia Omnibus Portus
ISSN 2792-6885 Volumen 1 - Número 2 (Noviembre 2021)

ALGORITMO DE LAS VEGAS. FACTORIZACIÓN DE ENTEROS

Martínez-López, Francisco J. IES Aguadulce
Ferrón-de-Haro, José R. IES Virgen del Carmen
Gallegos-Ruiz, Amalia IES Sol de Portocarrero
Martínez-Puertas, Sergio Universidad de Almería

Correo para correspondencia: fml199@ual.es

Recibido el 12/11/2021
Aceptado el 19/11/2021

INTRODUCCIÓN

Un **algoritmo probabilista** (o probabilístico) es un algoritmo que basa su resultado en la toma de algunas decisiones al azar, de tal forma que, en promedio, obtiene una buena solución al problema planteado para cualquier distribución de los datos de entrada. Es decir, al contrario que un algoritmo determinista, que, a partir de unos mismos datos, se pueden obtener distintas soluciones y, en algunos casos, soluciones erróneas.

CLASIFICACIÓN ALGORITMOS PROBABILÍSTICOS

- ✓ Algoritmos que no garantizan la corrección de la solución:
  - Algoritmos numéricos: Dan una solución aproximada (intervalo de confianza del 90%).
  - Algoritmos Monte Carlo: Dan la respuesta exacta con una alta probabilidad. En ocasiones dan respuestas incorrectas.
- ✓ Algoritmos que nunca dan una solución incorrecta:
  - Algoritmos de Las Vegas: Toman decisiones al azar. Si no encuentran la solución correcta lo admiten.

ALGORITMO DE LAS VEGAS

Un algoritmo de Las Vegas nunca da una solución falsa. Si no encuentra solución lo admite. Toma decisiones al azar para encontrar una solución antes que un algoritmo determinista.

Hay dos tipos de algoritmos de Las Vegas:

a) Los que siempre encuentran una solución correcta, aunque las decisiones al azar no sean afortunadas y la eficiencia disminuya (Algoritmos de Sherwood).

```
algoritmo LV(ent x:tpx; sal s:tpsolución; sal éxito:booleano)
{éxito devuelve verdad si LV encuentra la solución
y en ese caso s devuelve la solución encontrada}
función repe_LV(x:tpx) devuelve tpsolución
variables s:tpsolución; éxito:booleano
principio
       repetir
              LV(x,s,éxito) hastaQue éxito;
devuelve s
fin
```

b) Los que a veces, debido a decisiones desafortunadas, no encuentran una solución. Son aceptables siempre y cuando la probabilidad de que fallen sea baja. Resuelven problemas para los que no se conocen algoritmos deterministas eficientes (p.ej: factorización de enteros grandes).

El número de ejecuciones del bucle es $1/p(x)$.

Sea *v(x)* el tiempo esperado de ejecución de LV si éxito=verdad y *f*(*x*) el tiempo esperado si éxito=falso.

Entonces, el tiempo esperado *t*(*x*) de repe_LV es:

$$t(x)=p(x)v(x)+(1-p(x))(f(x)+t(x))$$

$$\rightarrow t(x)=v(x)+\frac{1-p(x)}{p(x)}f(x)$$

ALGORITMO DE LAS VEGAS. FACTORIZACIÓN DE ENTEROS

Problema: Descomponer un número en sus factores primos.

1.- *Problema más sencillo*: *partición.*– Dado un entero $n>1$, encontrar un divisor no trivial de n, suponiendo que n no es primo.
2.- *Factorización = test de primalidad + partición.*- Para factorizar n, hemos determinado si n es primo, si no, encontramos un divisor m de n y recursivamente factorizamos m y n/m.

Solución para el problema de la partición:

```
función partición(ent n:entero; sal n:entero)
variables m:entero; éxito:booleano
principio
       m:=2; éxito:=falso;
       mq m<=[sqrt(n)] and not éxito hacer
              si m divide a n
                     entonces éxito:=verdad
                     sino m:=m+1
              fsi
       fmq;
       si éxito
              entonces devuelve m
              sino devuelve n
       finsi
fin
```

El coste de la solución es demasiado alto. Por ejemplo, partir un número "duro" de unas 40 cifras, si cada ejecución del bucle tarda 1 nanosegundo, el algoritmo puede tardar miles de años.

Existen varios algoritmos de Las Vegas para factorizar números grandes. Con un pequeño ejemplo intentaremos explicar cómo sería el algoritmo eficiente para la factorización de números enteros grandes. Estos algoritmos eficientes se basan en el siguiente teorema:

"Sea n un numero entero compuesto. Sean a y b dos enteros distintos entre 1 y n-1 tales que a+b <> n . Si a^2 mod n = b^2 mod n, entonces mcd(a+b, n) es un divisor no trivial de n".

Por ejemplo n=2537. Sean a=2012 y b=1127. Observe que a^2= 1595n +1629 y b^2= 500n + 1629, tanto a^2 como b^2 son iguales a 1629 mod n. Dado que a <> b y que a+b <> n, el teorema dice que mcd(a+b, n)= mcd(3139,2537)= 43 es un divisor no trivial de n, lo cual es cierto. Esto sugiere un enfoque para descomponer n: buscar dos números distintos entre 1 y n-1 que tengan el mismo cuadrado mod n pero cuya suma no sea n, y utilizar el algoritmo de Euclides para calcular el mcd de su suma con n. Esto está muy bien siempre y cuando existan tales números, cuando n es compuesto, y siempre y cuando podamos hallarlos eficientemente.

La primera cuestión se resuelve rápidamente. Supuesto que n tenga al menos dos divisores primos diferentes, a^2 mod n admite al menos cuatro raíces cuadradas distintas en aritmética mod n. Prosiguiendo con nuestro ejemplo, 1629 admite exactamente cuatro raíces cuadradas mod 2537, a saber, 525,1127,1410 y 2012. Estas raíces vienen en parejas: 525+2012=1127+1410=2537. Dos cualquiera de estas raíces servirán, siempre y cuando no vengan de la misma pareja.

¿Y entonces, como podemos hallar a y b con la propiedad deseada? Este es el momento en que entra en juego lo aleatorio. Un entero se llama k-uniforme si todos sus divisores primos se encuentran entre los k-numeros más pequeños. Por ejemplo 120= 2^3*3*5 es 3-suave pero 35=5*7 no lo es. Cuando k es pequeño, los enteros k-suaves se pueden factorizar mediante la prueba de la división. En su primera fase el algoritmo de descomposición de las Vegas selecciona aleatoriamente un entero x entre 1 y n-1 y calcula $y = x^2$ mod n. Si y es k-suave entonces tanto x como la factorización de y se almacenan en una tabla. En caso contrario seleccionamos aleatoriamente otro x. El proceso se repite hasta que se hayan encontrado k+1 enteros diferentes para los cuales se conozcan la factorización de sus cuadrados mod n.

Prosiguiendo con este ejemplo en que n=2537, tomemos k=7. Entonces nos conciernen solamente los primos 2, 3, 5, 7, 11, 13 y 17. Seleccionamos aleatoriamente un primer entero x=1769.Calculamos su cuadrado módulo n: x^2=1233n+1240 y, por tanto, y=1240. No tiene éxito el

intento de factorizar $1240=2^3*5*31$, puesto que 31 no es divisible por ninguno de los primos admisibles. Un segundo intento con x=2455 es más afortunado: su cuadrado módulo n es $1650=2*3*5^2*11$. Prosiguiendo hasta conseguir 8 éxitos, obtenemos la tabla siguiente:

$X_1 = 2455 \rightarrow Y_1 = 1650 = 2*3*5^2*11$

$X_2 = 970 \rightarrow Y_2 = 2210 = 2*5*13*17$

$X_3 = 1105 \rightarrow Y_3 = 728 = 2^3*7*13$

$X_4 = 1458 \rightarrow Y_4 = 2295 = 3^3*5*17$

Esta tabla se utiliza para formar una matriz M de dimensiones k+1 * k sobre {0,1}. Cada fila corresponde a un éxito; cada columna corresponde a cada uno de los primos admisibles. La entrada M_{ij} recibe como valor 0 si el j-ésimo primo aparece como potencia par (incluyendo el 0), en la factorización de Y_i, en caso contrario $M_{ij}=1$. Por ejemplo, $M_{31}=1$, porque el primer primo, 2, aparece elevado a 3 en Y_3, y $M_{32}=0$ porque el segundo primo, 3, aparece con la potencia par 0. Para seguir con nuestro ejemplo, obtenemos la matriz siguiente:

$$M \rightarrow \begin{pmatrix} 1&1&0&0&1&0&0 \\ 1&0&1&0&0&1&1 \\ 1&0&0&1&0&1&0 \\ 0&1&1&0&0&0&1 \\ 1&0&1&0&1&0&0 \\ 1&1&0&0&0&1&1 \\ 0&1&0&1&0&0&0 \\ 0&0&0&0&1&1&0 \end{pmatrix}$$

Dado que esta matriz contiene más filas que columnas, las filas no pueden ser linealmente independientes. Este conjunto se puede hallar mediante eliminación de Gauss-Jordan. En este ejemplo, hay 7 soluciones distintas tales como las filas, 1, 2, 4 y 8 o las filas 1, 3, 4, 5, 6 y 7. Veamos ahora lo que sucede cuando se multiplican las Y_i correspondientes a las filas seleccionadas. Nuestros dos ejemplos producen respectivamente.

$$Y_1 * Y_2 * Y_4 * Y_8 = 2^6 * 3^4 * 5^4 * 11^2 * 13^2 * 17^2$$

$$Y_1 * Y_3 * Y_4 * Y_5 * Y_6 * Y_7 = 2^8 * 3^{10} * 5^4 * 7^2 * 11^2 * 13^2 * 17^2$$

Los exponentes de estos productos son necesariamente pares por construcción. Por tanto, se pueden obtener una raíz cuadrada de estos productos dividiendo por 2 los exponentes.

En aritmética módulo n se puede obtener una raíz cuadrada del mismo producto multiplicando los X_i correspondientes, puesto que cada $Y_i=X_i^2$ mod n. En nuestro ejemplo, los dos enfoques para calcular una raíz cuadrada módulo n de $Y_1 * Y_2 * Y_4 * Y_8$ dan lugar a

$$a = 2^3 * 3^2 * 5^2 * 11 * 13 * 17 \bmod 2537 = 2012$$

Tal como vimos anteriormente basta con calcular el máximo común divisor de a+b y n para obtener un divisor no trivial de n. En general, esta técnica produce dos enteros a y b entre 1 y n-1 tales que a^2 mod n= b^2 mod n. Sin embargo no hay garantías, de que a<>b ni de que a+b<>n. De hecho, la utilización de $Y_1 * Y_3 * Y_4 * Y_5 * Y_6 * Y_7$ en lugar de $Y_1 * Y_2 * Y_4 * Y_8$ se obtiene

$$a' = 2^4 * 3^5 * 5^2 * 7 * 11 * 13 * 17 \bmod 2537 = 1973$$

$$b' = 2455 * 1105 * 1458 * 216 * 80 * 1844 \bmod 2537 = 564$$

Que resulta inútil porque a' + b' = n. Sin embargo, se puede demostrar que todo este proceso tiene éxito tiene una probabilidad mínima del 50% a no ser que mcd (a,n) sea un divisor no trivial de Y, lo cual nos viene igualmente bien a efectos de la descomposición. Si tenemos el caso en que a' + b' = n buscaremos otros conjuntos de filas de M que sumen 0 en aritmética módulo 2. Si eso tampoco funciona, buscaremos unos pocos pares más X_iY_i y volveremos a probar con la matriz aumentada resultante.

Queda por determinar el valor de k para optimizar el rendimiento. Cuanto mayor sea mayor sera la probabilidad de que x^2 mod n sea k-suave y por otra parte cuanto más pequeño sea, más deprisa se efectuara la prueba de k-suavidad y más deprisa factorizaremos los valores k-suaves que encontremos y menos valores necesitaremos. El valor apropiado de k es muy complicado de saber, un valor que podemos considerar aceptable es el de 7, como el utilizado en nuestro ejemplo.

BIBLIOGRAFÍA

Allen Weiss, M. (2000). *Estructuras de datos en java.* Ed. Addison Wesley.

Brassard, G. y Bratley, P. (1997). *Fundamentos de Algoritmia*. Ed. Prentice Hall.

O´Rourke, J. (1994). *Computacional Geometry in C.* Cambridge University Press.

BIOGRAFÍAS

Francisco Javier Martínez es Ingeniero en Informática y Máster en Dirección de Empresas, por la Universidad de Almería. En 2015 se doctora con una tesis doctoral titulada "*Optimización de Algoritmos de Clasificación Basados en Redes Neuronales y Máquinas de Soporte Vectorial",* habiendo publicado, para ello, artículos en revistas como Soft Computing, Applied Soft Computing y Journal of Computational Science. Ha desempeñado, desde 2017 hasta 2021, el puesto de Inspector de Educación, y, actualmente, es Profesor de Enseñanza Secundaria en el IES Aguadulce, Profesor-Tutor de la UNED, en el centro asociado de Almería, y preparador de oposiciones para los cuerpos de profesor de secundaria, con la especialidad de informática, y profesor técnico de formación profesional, con la especialidad de sistemas y aplicaciones informáticas. Otras publicaciones relevantes son los artículos titulados *"El cómic como uso didáctico en el aprendizaje de las Matemáticas"*, en el Boletín de la Titulación de Matemáticas de la UAL, *"El método de MonteCarlo para el estudio de la primalidad de números",* en la revista Epsilon, así como diversos libros, destacando los publicados por la editorial Ra-Ma, "Teoría, diseño e implementación de Compiladores de Lenguajes" y "Programación de bases de datos relacionales".

José Ramón Ferrón de Haro es Ingeniero en Informática por la Universidad de Almería. Después de trabajar como programador informático y profesor de informática de cursos de formación permanente de la Junta de Andalucía decidió dedicarse a la docencia en centros educativos comenzando como profesor de informática en ciclos formativos de Informática para posteriormente realizar las oposiciones docentes y obtener plaza por la especialidad de Matemáticas. Actualmente docente de Matemáticas en el IES Virgen del Carmen (Jaén) donde imparte Matemáticas en bilingüe de francés.

Amalia Gallegos es graduada en Ingeniería Informática y Máster en Dirección de Empresas, por la Universidad de Almería. Profesora de Enseñanza Secundaria en el IES Sol de Portocarrero. Otras publicaciones relevantes son el artículo titulado *"El método de MonteCarlo para el estudio de la primalidad de números",* en la revista Epsilon, así como diversos libros, destacando los publicados por la editorial Ra-Ma, "Teoría, diseño e implementación de Compiladores de Lenguajes" y "Programación de bases de datos relacionales" o los publicados por el Instituto Europeo de Desarrollo y Tecnología Multimedia, SL, bajo los títulos de "Manual de Word 2007 Iniciación", "Manual de Word 2007 Avanzado", "Manual de Excel 2007 Iniciación" y "Manual de Excel 2007 Avanzado".

Sergio Martínez es licenciado en Matemáticas y Doctor en Matemáticas por la Universidad de Almería y actualmente es Profesor Titular de Universidad en el Departamento de Matemáticas de la Universidad de Almería. Su investigación actual se centra en la estimación de parámetros en poblaciones finitas, como la media, la función de distribución y los cuantiles mediante técnicas de calibración. Recientemente, ha escrito varios artículos relacionados con el análisis de grandes volúmenes de datos a través de estructuras de redes neuronales mixtas y artículos relacionados con métodos econométricos para la fijación de precios en empresas hoteleras. Sus trabajos han sido publicados en revistas como Sociological Methods & Research, International Journal of Hospitality Management, Applied Soft Computing, Journal of Computational and Applied Mathematics.