Algoritmos aleatorizados

IIC2283

Algoritmos aleatorizados

Vamos a permitir a los algoritmos tener una componente aleatoria

- En general esto significa que un algoritmo toma algunas decisiones dependiendo de valores escogidos al azar (según una distribución de probabilidades)

Algoritmos aleatorizados

Vamos a permitir a los algoritmos tener una componente aleatoria

- En general esto significa que un algoritmo toma algunas decisiones dependiendo de valores escogidos al azar (según una distribución de probabilidades)

Hablamos entonces de algoritmos aleatorizados

Algoritmos aleatorizados

La ejecución de un algoritmo aleatorizado depende entonces de valores escogidos al azar

- Distintas ejecuciones pueden dar resultados distintos

Algoritmos aleatorizados

La ejecución de un algoritmo aleatorizado depende entonces de valores escogidos al azar

- Distintas ejecuciones pueden dar resultados distintos

Vamos a considerar dos tipos de algoritmos aleatorizados:

Algoritmos aleatorizados

La ejecución de un algoritmo aleatorizado depende entonces de valores escogidos al azar

- Distintas ejecuciones pueden dar resultados distintos

Vamos a considerar dos tipos de algoritmos aleatorizados:

- **Monte Carlo:** el algoritmo siempre entrega un resultado, pero hay una probabilidad de que sea incorrecto

Algoritmos aleatorizados

La ejecución de un algoritmo aleatorizado depende entonces de valores escogidos al azar

- Distintas ejecuciones pueden dar resultados distintos

Vamos a considerar dos tipos de algoritmos aleatorizados:

- **Monte Carlo:** el algoritmo siempre entrega un resultado, pero hay una probabilidad de que sea incorrecto
- **Las Vegas:** si el algoritmo entrega un resultado es correcto, pero hay una probabilidad de que no entregue resultado

¿Cuáles son las ventajas de los algoritmos aleatorizados?

¿Cuáles son las ventajas de los algoritmos aleatorizados?

Existen problemas para los cuales los algoritmos aleatorizados son más eficientes que los algoritmos usuales (sin una componente aleatoria)

- Por ejemplo, el problema de verificar si un número es primo

¿Cuáles son las ventajas de los algoritmos aleatorizados?

Existen problemas para los cuales los algoritmos aleatorizados son más eficientes que los algoritmos usuales (sin una componente aleatoria)

- Por ejemplo, el problema de verificar si un número es primo

Existen problemas para los cuales los únicos algoritmos eficientes conocidos son aleatorizados

- Por ejemplo, el problema de verificar si dos polinomios en varias variables son equivalentes

¿Cuáles son las ventajas de los algoritmos aleatorizados?

Existen problemas para los cuales los algoritmos aleatorizados son más eficientes que los algoritmos usuales (sin una componente aleatoria)

- Por ejemplo, el problema de verificar si un número es primo

Existen problemas para los cuales los únicos algoritmos eficientes conocidos son aleatorizados

- Por ejemplo, el problema de verificar si dos polinomios en varias variables son equivalentes

Vamos a ver en detalle estos ejemplos ...

Algoritmos de Monte Carlo: equivalencia de polinomios

Consideramos polinomios en $\mathbb{Q}$

Suponemos inicialmente que un polinomio es una expresión de la forma:

$$p(x) \quad = \quad \sum_{i=1}^{k} \prod_{j=1}^{\ell_i} (a_{i,j} x + b_{i,j})$$

donde cada $a_{i,j}, b_{i,j} \in \mathbb{Q}$

La forma canónica de $p(x)$ es una expresión de la forma:

$$p(x) \quad = \quad \sum_{i=0}^{\ell} c_i x^i$$

donde cada $c_i \in \mathbb{Q}$ y $\ell \leq \text{máx}\{\ell_1, \ldots, \ell_k\}$

Si $c_\ell \neq 0$, entonces $p(x)$ no es el polinomio nulo y su grado es ℓ

Algoritmos de Monte Carlo: equivalencia de polinomios

Dados dos polinomios $p(x)$ y $q(x)$, queremos verificar si son idénticos.

- Para cada $a \in \mathbb{Q}$, se tiene que $p(a) = q(a)$

Algoritmos de Monte Carlo: equivalencia de polinomios

Dados dos polinomios $p(x)$ y $q(x)$, queremos verificar si son idénticos.

- Para cada $a \in \mathbb{Q}$, se tiene que $p(a) = q(a)$

¿Cómo podemos resolver este problema?

- La operación básica a contar es la suma y multiplicación de números racionales

Un algoritmo para la equivalencia de polinomios

EquivPol$(p(x), q(x))$
transforme $p(x)$ es su forma canónica $\sum_{i=0}^{k} c_i x^i$
transforme $q(x)$ es su forma canónica $\sum_{i=0}^{\ell} d_i x^i$
if $k \neq \ell$ **then return** no
else
for $i := 0$ **to** k **do**
if $c_i \neq d_i$ **then return** no
return sí

Un algoritmo para la equivalencia de polinomios

Ejercicio

Muestre que el algoritmo anterior en el peor caso es $O(n^2)$, donde $n = |p(x)| + |q(x)|$

Un algoritmo para la equivalencia de polinomios

Ejercicio

Muestre que el algoritmo anterior en el peor caso es $O(n^2)$, donde $n = |p(x)| + |q(x)|$

¿Es posible resolver este problema utilizando un menor número de operaciones?

Un algoritmo aleatorizado para la equivalencia de polinomios

Suponga que:

$$\begin{aligned} p(x) &= \sum_{i=1}^{k} \prod_{j=1}^{r_i} (a_{i,j} x + b_{i,j}) \\ q(x) &= \sum_{i=1}^{\ell} \prod_{j=1}^{s_i} (c_{i,j} x + d_{i,j}) \end{aligned}$$

Un algoritmo aleatorizado para la equivalencia de polinomios

Suponga que:

$$
\begin{aligned}
p(x) &= \sum_{i=1}^{k} \prod_{j=1}^{r_i} (a_{i,j} x + b_{i,j}) \\
q(x) &= \sum_{i=1}^{\ell} \prod_{j=1}^{s_i} (c_{i,j} x + d_{i,j})
\end{aligned}
$$

Utilizamos el siguiente algoritmo aleatorizado:

EquivPolAleatorizado($p(x)$, $q(x)$)
    $K := 1 + \max\{r_1, \ldots, r_k, s_1, \ldots, s_\ell\}$
    escoja al azar y con distribución uniforme un elemento a
        del conjunto de números naturales $\{1, \ldots, 100 \cdot K\}$
    if $p(a) = q(a)$ **then return** sí
    else return no

Un algoritmo aleatorizado para la equivalencia de polinomios

El algoritmo sólo necesita realizar $O(n)$ operaciones, donde $n = |p(x)| + |q(x)|$

- Ya que necesita calcular $p(a)$ y $q(a)$

Un algoritmo aleatorizado para la equivalencia de polinomios

El algoritmo sólo necesita realizar $O(n)$ operaciones, donde $n = |p(x)| + |q(x)|$

- Ya que necesita calcular $p(a)$ y $q(a)$

Pero el algoritmo puede dar una respuesta equivocada

- ¿Cuál es la probabilidad de error?

Calculando la probabilidad de error

Sean $p(x)$ y $q(x)$ dos polinomios dados como entrada a **EquivPolAleatorizado**

- Si los polinomios $p(x)$ y $q(x)$ son equivalentes, entonces el algoritmo responde **sí** sin cometer error

- Si los polinomios $p(x)$ y $q(x)$ no son equivalentes, el algoritmo puede responder **sí** al sacar al azar un elemento $a \in \{1, \ldots, 100 \cdot K\}$ tal que $p(a) = q(a)$

  Esto significa que a es una raíz del polinomio $r(x) = p(x) - q(x)$

Calculando la probabilidad de error

$r(x)$ no es el polinomio nulo y es de grado a lo más K

- Por lo tanto $r(x)$ tiene a lo más K raíces en $\mathbb{Q}$

Calculando la probabilidad de error

$r(x)$ no es el polinomio nulo y es de grado a lo más K

- Por lo tanto $r(x)$ tiene a lo más K raíces en $\mathbb{Q}$

Concluimos que:

$$\begin{aligned}\Pr(a \text{ sea una raíz de } r(x)) &\leq \frac{K}{100 \cdot K} \\ &= \frac{1}{100}\end{aligned}$$

Un mejor algoritmo aleatorizado

La probabilidad de error del algoritmo está acotada por $\frac{1}{100}$

- ¿Es aceptable esta probabilidad?

Un mejor algoritmo aleatorizado

La probabilidad de error del algoritmo está acotada por $\frac{1}{100}$

- ¿Es aceptable esta probabilidad?

Ejercicio

De un algoritmo que resuelva el problema de equivalencia de polinomios, que en el peor caso sea $O(n)$ y que tenga una probabilidad de error acotada por $\frac{1}{100^{10}}$

Un mejor algoritmo aleatorizado

La probabilidad de error del algoritmo está acotada por $\frac{1}{100}$

- ¿Es aceptable esta probabilidad?

Ejercicio

De un algoritmo que resuelva el problema de equivalencia de polinomios, que en el peor caso sea $O(n)$ y que tenga una probabilidad de error acotada por $\frac{1}{100^{10}}$

¿Confiaría en este algoritmo lineal?

Un mejor algoritmo aleatorizado

La probabilidad de error del algoritmo está acotada por $\frac{1}{100}$

- ¿Es aceptable esta probabilidad?

Ejercicio

De un algoritmo que resuelva el problema de equivalencia de polinomios, que en el peor caso sea $O(n)$ y que tenga una probabilidad de error acotada por $\frac{1}{100^{10}}$

¿Confiaría en este algoritmo lineal?

- ¿Para qué probabilidad estaría dispuesto a confiar?

Una solución para el ejercicio

Suponga que $p(x)$ y $q(x)$ son de la forma definida en la versión anterior de **EquivPolAleatorizado**.

EquivPolAleatorizado$(p(x),\ q(x))$

$K := 1 + \text{máx}\{r_1, \ldots, r_k, s_1, \ldots, s_\ell\}$
$A := \{1, ..., 100 \cdot K\}$
$total := 0$
for $i := 1$ **to** 10 **do**
  escoja al azar y con distribución uniforme un elemento a en A
  if $p(a) = q(a)$ **then** $total = total + 1$
if $total = 10$ **then return** sí
return no

Un segundo ejemplo: una definición general de polinomios

Consideramos polinomios en varias variables en $\mathbb{Q}$

Un monomio es una expresión de la forma $cx_1^{\ell_1} \cdots x_n^{\ell_n}$, donde $c \in \mathbb{Q}$ y cada $\ell_i \in \mathbb{N}$

Un monomio $cx_1^{\ell_1} \cdots x_n^{\ell_n}$ es nulo si $c = 0$

- No es nulo si $c \neq 0$

El grado de un monomio $cx_1^{\ell_1} \cdots x_n^{\ell_n}$ no nulo es $\ell_1 + \cdots + \ell_n$

Un segundo ejemplo: una definición general de polinomios

Un polinomio es una expresión de la forma:

$$p(x_1, \dots, x_n) \quad = \quad \sum_{i=1}^{\ell} \prod_{j=1}^{m_i} \left(\sum_{k=1}^{n} a_{i,j,k} x_k + b_{i,j} \right)$$

donde cada $a_{i,j,k} \in \mathbb{Q}$ y cada $b_{i,j} \in \mathbb{Q}$

Un segundo ejemplo: una definición general de polinomios

La forma canónica de un polinomio $p(x_1, \ldots, x_n)$ es única, y es igual a 0 o a una suma de monomios que satisface las siguiente propiedades:

- cada monomio en la forma canónica es de la forma $cx_1^{\ell_1} \cdots x_n^{\ell_n}$ con $c \neq 0$
- si $cx_1^{\ell_1} \cdots x_n^{\ell_n}$ y $dx_1^{m_1} \cdots x_n^{m_n}$ son dos monomios distintos en la forma canónica, entonces $\ell_i \neq m_i$ para algún $i \in \{1, \ldots, n\}$

Un polinomio $p(x_1, \ldots, x_n)$ es nulo si su forma canónica es 0

El grado de un polinomio $p(x_1, \ldots, x_n)$ no nulo es el mayor grado de los monomios en su forma canónica.

Equivalencia de polinomios en varias variables

Dos polinomios $p(x_1, \ldots, x_n)$ y $q(x_1, \ldots, x_n)$ son idénticos si para cada secuencia $a_1, \ldots, a_n \in \mathbb{Q}$ se tiene que:

$$p(a_1, \ldots, a_n) \quad = \quad q(a_1, \ldots, a_n)$$

Equivalencia de polinomios en varias variables

Dos polinomios $p(x_1, \ldots, x_n)$ y $q(x_1, \ldots, x_n)$ son idénticos si para cada secuencia $a_1, \ldots, a_n \in \mathbb{Q}$ se tiene que:

$$p(a_1, \ldots, a_n) \quad = \quad q(a_1, \ldots, a_n)$$

Nuevamente queremos verificar si dos polinomios son idénticos.

Equivalencia de polinomios en varias variables

¿Podemos verificar en tiempo polinomial si dos polinomios en varias variables son equivalentes?

Equivalencia de polinomios en varias variables

¿Podemos verificar en tiempo polinomial si dos polinomios en varias variables son equivalentes?

Tenemos un problema: calcular la forma canónica de un polinomio toma tiempo exponencial

Equivalencia de polinomios en varias variables

¿Podemos verificar en tiempo polinomial si dos polinomios en varias variables son equivalentes?

Tenemos un problema: calcular la forma canónica de un polinomio toma tiempo exponencial

Pero existe un algoritmo aleatorizado eficiente para este problema.

Equivalencia de polinomios en varias variables

¿Podemos verificar en tiempo polinomial si dos polinomios en varias variables son equivalentes?

Tenemos un problema: calcular la forma canónica de un polinomio toma tiempo exponencial

Pero existe un algoritmo aleatorizado eficiente para este problema.

- Esto no es trivial ya que un polinomio $p(x_1, \ldots, x_n)$ puede tener una cantidad infinita de raíces

Equivalencia de polinomios en varias variables

¿Podemos verificar en tiempo polinomial si dos polinomios en varias variables son equivalentes?

Tenemos un problema: calcular la forma canónica de un polinomio toma tiempo exponencial

Pero existe un algoritmo aleatorizado eficiente para este problema.

- Esto no es trivial ya que un polinomio $p(x_1, \ldots, x_n)$ puede tener una cantidad infinita de raíces
  - Por ejemplo: $p(x_1, x_2) = (x_1 - 1)(x_2 - 3)$

Equivalencia de polinomios en varias variables

¿Podemos verificar en tiempo polinomial si dos polinomios en varias variables son equivalentes?

Tenemos un problema: calcular la forma canónica de un polinomio toma tiempo exponencial

Pero existe un algoritmo aleatorizado eficiente para este problema.

- Esto no es trivial ya que un polinomio $p(x_1, \ldots, x_n)$ puede tener una cantidad infinita de raíces
  - Por ejemplo: $p(x_1, x_2) = (x_1 - 1)(x_2 - 3)$
- El ingrediente esencial es el lema de Schwartz-Zippel

El ingrediente principal

Lema de Schwartz-Zippel

Sea $p(x_1, \ldots, x_n)$ un polinomio no nulo de grado k, y sea A un subconjunto finito y no vacío de $\mathbb{Q}$. Si $a_1, \ldots, a_n$ son elegidos de manera uniforme e independiente desde A, entonces

$$\Pr(p(a_1, \ldots, a_n) = 0) \quad \leq \quad \frac{k}{|A|}$$

Un algoritmo aleatorizado para la equivalencia de polinomios en varias variables

Vamos a dar un algoritmo aleatorizado eficiente para el problema de verificar si dos polinomios en varias variables son equivalentes

Un algoritmo aleatorizado para la equivalencia de polinomios en varias variables

Vamos a dar un algoritmo aleatorizado eficiente para el problema de verificar si dos polinomios en varias variables son equivalentes

Suponga que la entrada del algoritmo está dada por los siguientes polinomios:

$$\begin{aligned} p(x_1, \ldots, x_n) &= \sum_{i=1}^{\ell} \prod_{j=1}^{r_i} \left(\sum_{k=1}^{n} a_{i,j,k} x_k + b_{i,j} \right) \\ q(x_1, \ldots, x_n) &= \sum_{i=1}^{m} \prod_{j=1}^{s_i} \left(\sum_{k=1}^{n} c_{i,j,k} x_k + d_{i,j} \right) \end{aligned}$$

Un algoritmo aleatorizado para la equivalencia de polinomios en varias variables

EquivPolAleatorizado$(p(x_1, \ldots, x_n),\ q(x_1, \ldots, x_n))$
$K := 1 + \text{máx}\,\{r_1, \ldots, r_\ell, s_1, \ldots, s_m\}$
$A := \{1, ..., 100 \cdot K\}$
sea $a_1, \ldots, a_n$ una secuencia de números elegidos de manera uniforme e independiente desde A
if $p(a_1, \ldots, a_n) = q(a_1, \ldots, a_n)$ **then return** sí
else return no

Utilizando el lema de Schwartz-Zippel

Vamos a calcular la probabilidad de error del algoritmo:

Utilizando el lema de Schwartz-Zippel

Vamos a calcular la probabilidad de error del algoritmo:

- Si los polinomios $p(x_1, \ldots, x_n)$ y $q(x_1, \ldots, x_n)$ son equivalentes, entonces el algoritmo responde **sí** sin cometer error

Utilizando el lema de Schwartz-Zippel

Vamos a calcular la probabilidad de error del algoritmo:

- Si los polinomios $p(x_1, \ldots, x_n)$ y $q(x_1, \ldots, x_n)$ son equivalentes, entonces el algoritmo responde **sí** sin cometer error
- Si los polinomios $p(x_1, \ldots, x_n)$ y $q(x_1, \ldots, x_n)$ no son equivalentes, el algoritmo puede responder **sí** al escoger una secuencia de números $a_1, \ldots, a_n$ desde A tales que $p(a_1, \ldots, a_n) = q(a_1, \ldots, a_n)$
  - Donde $A = \{1, \ldots, 100 \cdot K\}$

Utilizando el lema de Schwartz-Zippel

Vamos a calcular la probabilidad de error del algoritmo:

- Si los polinomios $p(x_1, \ldots, x_n)$ y $q(x_1, \ldots, x_n)$ son equivalentes, entonces el algoritmo responde **sí** sin cometer error

- Si los polinomios $p(x_1, \ldots, x_n)$ y $q(x_1, \ldots, x_n)$ no son equivalentes, el algoritmo puede responder **sí** al escoger una secuencia de números $a_1, \ldots, a_n$ desde A tales que $p(a_1, \ldots, a_n) = q(a_1, \ldots, a_n)$
  - Donde $A = \{1, \ldots, 100 \cdot K\}$

Esto significa que $(a_1, \ldots, a_n)$ es una raíz del polinomio $r(x_1, \ldots, x_n) = p(x_1, \ldots, x_n) - q(x_1, \ldots, x_n)$

Utilizando el lema de Schwartz-Zippel

$r(x_1, \ldots, x_n)$ no es el polinomio nulo y es de grado t con $t < K$

- Dado que $K = 1 + \text{máx}\{r_1, \ldots, r_\ell, s_1, \ldots, s_m\}$

Utilizando el lema de Schwartz-Zippel

$r(x_1, \ldots, x_n)$ no es el polinomio nulo y es de grado t con $t < K$

- Dado que $K = 1 + \max\{r_1, \ldots, r_\ell, s_1, \ldots, s_m\}$

Utilizando el lema de Schwartz-Zippel obtenemos:

$$\Pr(r(a_1, \ldots, a_n) = 0) \ \leq \ \frac{t}{|A|} \ < \ \frac{K}{|A|} \ = \ \frac{K}{100 \cdot K} \ = \ \frac{1}{100}$$

Utilizando el lema de Schwartz-Zippel

$r(x_1, \ldots, x_n)$ no es el polinomio nulo y es de grado t con $t < K$

- Dado que $K = 1 + \max\{r_1, \ldots, r_\ell, s_1, \ldots, s_m\}$

Utilizando el lema de Schwartz-Zippel obtenemos:

$$\Pr(r(a_1, \ldots, a_n) = 0) \;\leq\; \frac{t}{|A|} \;<\; \frac{K}{|A|} \;=\; \frac{K}{100 \cdot K} \;=\; \frac{1}{100}$$

La probabilidad de error del algoritmo está entonces acotada por $\frac{1}{100}$

Un mejor algoritmo aleatorizado para el problema general

Ejercicio

De un algoritmo aleatorizado que resuelva el problema de equivalencia de polinomios en varias variables.

- La probabilidad de error del algoritmo debe estar acotada por $\frac{1}{100^{10}}$
- Debe existir una constante c tal que el algoritmo en el peor caso es $O(m^c)$, donde m es el tamaño de la entrada
  - Si consideramos $p(x_1, \ldots, x_n)$ y $q(x_1, \ldots, x_n)$ como palabras sobre un cierto alfabeto, entonces $m = |p(x_1, \ldots, x_n)| + |q(x_1, \ldots, x_n)|$
  - Recuerdo que la operación básica a contar es la suma y multiplicación de números racionales.

Una aplicación: polinomios como circuitos