Análisis Probabilístico y Algoritmo 'Hecho Azaroso'

M. Andrea Rodríguez-Tastets
Ayudante: Erick Elejalde

Universidad de Concepción,Chile
www.inf.udec.cl\ ∼andrea
andrea@udec.cl

I Semestre - 2014

Repaso: Probabilidad

- Cada argumento probabilístico se refiere en última instancia a algún espacio de muestra, el cual es un conjunto de eventos elementales. Ejemplo, el lanzamiento de dos monedas tiene como espacio de muestra $\{CC, CS, SC, SS\}$.
- Un evento es un subconjunto del espacio de muestra. Por ejemplo, el evento "dos monedas que se lanzan y dan lo mismo" es $\{CC, SS\}$

Distribución de Probabilidad Específicas

- Distribución de probabilidad discreta: el espacio de muestra es finito o contablemente infinito. Ejemplo, lanzamiento de dos monedas a la vez; lanzar una moneda infinitamente.
- Distribución de probabilidad uniforme: el espacio de muestreo S es finito y cada evento elemental tiene la misma probabilidad, $1/|S|$. Ejemplo, lanzamiento de dos monedas a la vez.

Repaso: Distribución de Probabilidad

Una distribución de probabilidad Pr sobre un espacio de muestra S es una función desde eventos de S a un número real tal que :

$$\begin{array}{rcl} Pr\{A\} & \geq & 0 \\ Pr\{S\} & = & 1 \\ Pr\{A \cup B\} & = & Pr\{A\} + Pr\{B\}; \text{ para eventos } A \text{ y } B \text{ independientes} \\ Pr\{A\} & & ; \text{ probabilidad de evento } A \subseteq S \end{array}$$

Además:

$$\begin{array}{rcl} Pr\{\emptyset\} & = & 0 \\ \text{si } A \subseteq B & \text{entonces} & Pr\{A\} \leq Pr\{B\} \\ Pr\{S - A\} & = & 1 - Pr\{A\} \\ Pr\{A \cup B\} & = & Pr\{A\} + Pr\{B\} - Pr\{A \cap B\} \leq Pr\{A\} + Pr\{B\} \end{array}$$

Ejemplo: Asuma $Pr\{\{CC\}\} = Pr\{\{CS\}\} = Pr\{\{SC\}\} = Pr\{\{SS\}\} = 1/4$.

$Pr\{$"al menos una cara"$] = Pr\{\{CC \cup CS \cup SC\}\} = Pr\{\{CC\}\} + Pr\{\{CS\}\} + Pr\{\{SC\}\} = 3/4$ y $Pr\{$ "menos que una cara"$\} = 1 - Pr\{$"al menos una cara"$\} = 1/4$.

Problema del lanzamiento de monedas

- Asuma que se lanza una moneda justa n veces. Cada evento elemental en el espacio de muestra es una secuencia de n caras o sellos, describiendo la salida de un experimento o intento. El tamaño del espacio de muestra es de 2^n.
- Sea A el evento "k caras y $n-k$ sellos". Entonces $Pr\{A\} = C(n,k)/2^n$, hay $C(n,k) = \left(\begin{array}{c} n \\ k \end{array}\right) = \frac{n!}{k!(n-k)!}$ secuencias de largo n para las cuales k son caras y $n-k$ son sellos, y cada uno tiene probabilidad de $1/2^n$.
- Ejemplo: se $n = 5$ y $k = 3$. El evento es: $\{CCCSS, CCSSC, CSSCC, SSCCC, CCSCS,$ $CSCSC, SCSCC, CSCCS, SCCSC, SCCCS\}$. Entonces $Pr\{$"3 caras y 2 sellos"$\} = C(5,3)/2^5 = 10/32$.

Independencia de eventos

- Dos eventos A y B son independientes si $Pr\{A \cap B\} = Pr\{A\}Pr\{B\}$, es decir, la probabilidad de que los dos eventos ocurran simultaneamente es el producto de las probabilidades de que ocurran separadamente A y B.

Variables Aleatorias Discretas

- Un indicador de una variable aleatoria $I\{A\}$ asociado con un evento A se define como:

$$I\{A\} = \begin{cases} 1 & \text{Si } A \text{ ocurre} \\ 0 & \text{Si } A \text{ no ocurre} \end{cases}$$

- Una variable aleatoria discreta Y retorna posible valores de un espacio de mestreo.
- Defina el indicador de variable aletaoria X_A como el indicador de una variable aletaria Y que denota eventos elementales en S que satisfacen A, expresado como $Y = A$. Así , $Pr\{X_A\}$ es la suma de las probabilidad de eventos en S tal que $Y = A$.
- Ejemplo, Sea X_A el indicador de variable aleatoria en el espacio S que es 1 si una de las monedas es cara, entonces el subconjunto de eventos en S que satisface A es $\{CC, CS, SC\}$. Si consideramos que todos los eventos son igualmente probables, entonces cada uno tiene una probabilidad de $1/4$, lo que nos da que $Pr\{X_A\} = Pr\{I\{A\}\} = (3 \times 1/4) = 3/4$.

Variables Aleatorias Independientes

- Es común definir más de una variable aleatoria sobre el mismo espacio de muestra. Por ejemplo, X_A es el valor máximo de un dado tirado, o X_B el valor mínimo de un dado tirado.
- Dos variables aleatoria X e Y son independientes si para todo v y w, los eventos $X = v$ y $X = w$ son independientes.

Valor Esperado: Propiedades

El valor esperado de un indicador de variable aleatoria discreta es:

Definición: En general $E[X] = \sum_{x=1}^{n} xPr\{X = x\}$. Para un indicador de variable aleatoria A que se hace 1 si se cumple y 0 si no, $E[X_A] = E[I\{A\}] = Pr\{A\}$.

Propiedades:

$$
\begin{aligned}
E[X+Y] &= E[X] + E[Y]; \text{ para cualquier par de variables } X \text{ e } Y, \\
&\quad \text{incluso si no son independientes} \\
E[aX] &= aE[X]; \text{ para cualquier variable } X \text{ y cualquier constante } a \\
E[XY] &= E[X]E[Y]; \text{ para cualquier par independiente de variables aletaoria} \\
&\quad X \text{ e } Y
\end{aligned}
$$

Análisis probabilístico: Introducción

- Análisis probabilístico es el uso de probabilidad en el análisis de problemas.
- Se usa comúnmente para analizar el tiempo de ejecución, pero puede ser usado para otras definiciones de costo.
- Requiere conocer o hacer algunas presunciones acerca de la distribución de las entradas.
- Se determina entonces el tiempo esperado, promediando el tiempo sobre todo las posibles entradas.
- Cuando no es posible definir una distribución razonable de las entradas, análisis probabilístico no se puede aplicar.

Ejemplo 1: contratación

Asuma que usted quiere contratar a un asistente y para eso contrata una agencia que le envía candidatos que usted entrevista. Suponga además que la agencia le enviará n candidatos, numerados de 1 a n. El procedimiento considera que usted puede, después de entrevistar a un candidato i, determinar si el candidato i es mejor a cualquier candidato que haya seleccionado hasta el momento de manera que, si es mejor que el asistente actual, está dispuesto a despedir al asistente actual y contratar a i. El procedimiento considera un candidato ficticio 0, quien siempre será peor calificado que cualquiera de los candidatos de 1 a n.

Ejemplo 1: algoritmo

Contratacion − Asistente(n)

```
begin
  best ← 0
  for i ← 1 to n do
    entreviste a candidato i
    if candidato i es mejor que best then
      best ← i
      contrate a i
    endif
  endfor
end
```

Ejemplo 1: análisis

- El peor caso es cuando la agencia te envía a los postulantes en orden creciente de sus cualidades. Entonces se contrata y despide en turnos, es decir, n contrataciones y despidos.
- El mejor caso es cuando la agencia te envía el mejor postulante la primera vez. En ese caso se contrata y despide solo una vez.

Ejemplo 1: análisis (cont.)

- ¿Qué se puede decir del caso promedio?
- Primero hay que decir qué significa el promedio.
- Una entrada al problema es un orden de llegada de los postulantes. Hay entonces $n!$ entradas diferentes.
- Asuma que hay alguna distribución de las entradas, por ejemplo, cada orden de entrada es igualmente probable pero otras distribuciones son también posibles.
- El costo promedio es el valor esperado.

Valor Esperado: Problema de Contratación

- Queremos conocer el costo esperado del algoritmo de contratación, en términos de cuántas veces se contrata un postulante.
- Evento elementario es una secuencia de los n postulantes. El espacio de muestreo son la $n!$ posibles secuencias de postulates.
- Asuma una distribución uniforme, así cada secuencia es igualmente probable, entonces tiene probabilidad $1/n!$.
- Variable aleatoria X es el número de postulantes que son contratados, dado la secuencia de entrada s. ¿Cuál es $E[X]$?

Valor Esperado: Problema de Contratación (cont.)

- El problema se puede presentar de una forma distinta. En vez de tener una variable aleatoria de cuántos postulantes son contratados, se puede considerar n variables, cada una indicando si un postulante es contratado o no. Sea X_i el indicador de variable aleatoria definida como 1 si postulante i es contratado y 0 si no.
- $E[Xi] = Pr\{$ "postulante i es contratado" $\}$, la probabilidad de contratación de i es la probabilidad de que i sea mejor que los otros previos $i-1$ postulantes.

Valor Esperado: Problema de Contratación (cont.)

- Asuma $n = 4$ y $i = 3$. La pregunta es ¿En qué fracción de todas las entradas el 3er postulante es mejor que los otros 2 postulantes previos? En general, ya que todas las permutaciones son igualmente probables, si consideramos solo los primeros i postulantes, el mejor de ellos es igualmente probable de ocurrir en cualquiera de las i posiciones. Así $Pr\{X_i = 1\} = 1/i$.
- Sea $X = \sum_i X_i$ la variable que define el número de contratos totales. Entonces,

$$
\begin{aligned}
E[X] &= E[\sum_i X_i] \\
&= \sum_i E[X_i]; \text{ por propiedad de } E \\
&= \sum_i Pr\{X_i = 1\}; \text{ por propiedad de } X_i \\
&= \sum_i 1/i; \text{ por argumento descrito anteriormente} \\
&\leq ln\ n + 1; \text{ por fórmula para número armónico}
\end{aligned}
$$

Problema de Contratación: discusión

- El número promedio de contrataciones es *ln n*, el cual es mucho mejor que el peor caso de *n*. Esto se basa en que la agencia envía los postulantes en forma aleatoria.
- ¿Qué sucede si no se puede confiar en eso? Quizás la agencia, por ejemplo, envía siempre los mejores postulantes primero.
- Si uno puede obtener la lista de postulante con anterioridad, uno puede crear su propia aleatoridad al aplicar permutaciones aleatorias a la lista de postulantes. Esto lleva de un análisis probabilístico (pasivo) a un algoritmo hecho azaroso (activo), poniendo la aleatoridad bajo nuestro control.

Ejercicios

- En el problema de contratación, asumiendo que los candidatos son presentados en orden aleatorio, ¿Qué probabilidad existe de contratar solo a uno? ¿Qué probabilidad de contratar exactamente n veces?
- Analice el siguiente caso y determine el valor esperado. Cada uno de n clientes le da un sombrero al encargado de un restaurant. El encargado luego entrega un sombrero de vuelta a cada uno de los clientes en forma aleatoria. ¿Cuál es el valor esperado del número de clientes que recibe de vuelta su propio sombrero?
- Sea $A[1 \dots n]$ un arreglo de n números distintos. Si $i < j$ y $A[i] > A[j]$ el par (i, j) se dice que es una inversión. Asuma que los n elementos en A son una permutación aleatoria de (1, 2, . . . , n). Indique el valor esperado del número de inversiones.

Ejercicios (cont.)

- Considere que la profesora de Análisis de Algoritmos realiza interrogaciones a n grupos de alumnos. El tiempo que demora en cada interrogación depende de si las respuestas a los problemas de las tareas están correctas (1) o incorrectas (3). Cada grupo tiene 2 preguntas y tiene la misma probabilidad de tener una pregunta correcta o incorrecta. La profesora comienza a llamar a interrogación, supuestamente en forma aleatoria. ¿cuál es el mayor tiempo, el menor tiempo y el tiempo esperado de revisión de las tareas.

Randomized Algorithms

- En vez de asumir (quizás en forma incorrecta) que las entradas exhiben alguna distribución, haga su propia distribución de la entrada.
- Sobre una misma entrada, un algoritmo tiene múltiples posible ejecuciones.

Randomized Algorithms: Problema de contratación

- Asuma que tenemos acceso a la lista entera de postulantes antes.
- Altere aleatoriamente el orden de la lista de postulantes.
- Entonces entreviste a los candidatos de la secuencia aleatoria
- Número esperado de contrataciones y despidos es $O(ln\ n)$ no importa la entrada original.

Análisis Probabilístico versus Randomized Algorithms?

- Análisis probabilístico de un algoritmo determinístico: asuma una distribución de probabilidad en la entrada.
- Algoritmos hechos azarosos: use una elección aleatoria de entrada para el algoritmo.

Proceso de crear un algoritmo hecho azaroso

Input: array $A[1 \dots n]$
Para $i := 1$ a n hacer
  $j \leftarrow$ *valor* entre i y n con probabilidad uniforme
  intercambiar A[i] por A[j]

¿ Por qué funciona esto?

- Hay que mostrar que después de la $i-$ésima iteración del loop: por cada posible i-permutación, el arreglo $A[1 \dots i]$ *contiene esta i-permutación con probabilidad de* $(n-i)!/n!$
- Base: Después de la primera iteración, $A[1]$ contiene cada permutación de un elemento desde $\{1, \dots, n\}$ con probabilidad de $(n-1)!/n! = 1/n$ ya que $A[1]$ se intercambia con un elemento tomado del arreglo completo uniformemente en forma aleatoria.
- Inducción: Asuma que después $(i-1)$-primeras iteraciones de iteración de loop, $A[1 \dots i-1]$ iguala cada permutación de $i-1$ elementos desde $\{1, \dots, n\}$ con probabilidad $(n-(i-1))!/n!$.
- La probabilidad que $A[1 \dots i]$ contiene la permutación $x1, x2, \dots, xi$ es la probabilidad que $A[1 \dots i-1]$ contiene $x1, x2, , xi-1$ después de $(i-1)$ iteraciones y que la $i-$ésima iteración pone xi en $A[i]$.

Proceso de crear un algoritmo azaroso (cont.)

- Sea $e1$ el evento de que $A[1 \ldots i-1]$ contiene $x1, x2, , xi-1$ después de la $(i-1)$ iteración.
- Sea $e2$ el evento que la $i-$ésima iteración pone xi en $A[i]$.
- Queremos demostrar que $Pr\{e1 \cap e2\} = (n-i)!/n!$
- Desafortunadamente $e1$ y $e2$ no son independientes: si alguno aparece en $A[1 \ldots i-1]$, entonces no está disponible para aparecer en $A[i]$. Se necesita el concepto de probabilidad condicional: $Pr\{A|B\} = Pr\{A \cap B\}/Pr\{B\}$.
- Entonces: $Pr\{e1 \cap e2\} = Pr\{e2|e1\}Pr\{e1\}$ y $Pr\{e2|e1\} = 1/(n-i+1)$, ya que xi está disponible en $A[i \ldots n]$ para ser escogido porque que $e1$ ya ocurrió y no incluyó a xi y cada elemento en $A[i \ldots n]$ es igualmente probable de ser escogido.
- $Pr\{e1\} = (n-(i-1))!/n!$ por hipótesis de inducción.
  Así $Pr\{e1 \cap e2\} = (1/(n-i+1)) \times ((n-(i-1))!/n!) = (n-i)!/n!$
- Después de la última iteración, la hipótesis de inducción nos dice que $A[1 \ldots n]$ iguala cada permutación de los n elementos de $\{1, \ldots, n\}$ con probabilidad de $(n-n)!/n! = 1/n!$. Así el algoritmos nos da una permutación aleatoria uniforme.

Proceso de crear un algoritmo azaroso (2)

Permute-By-Sorting
Input: array $A[1 \dots n]$
$n \leftarrow length[A]$
Para $i := 1$ a n hacer
$P[i] \leftarrow RANDOM(1, n^3)$
Ordene A usando P como clave de ordenamiento

¿ Por qué funciona esto?

- La idea es demostrar que la probabilidad de cualquier permutación es de $1/n!$.
- Sea X_i el evento donde un elemento $A[i]$ recibe la $i-$ésima menor prioridad. Entonces queremos obtener :

$$Pr\{X_1 \cap X_2 \cap \cdots \cap X_n\} = \left(\frac{1}{n}\right)\left(\frac{1}{n-1}\right)\dots\left(\frac{1}{1}\right) = \frac{1}{n!}. \tag{1}$$

Quicksort

- Quicksort determinístico: $\Theta(n^2)$ tiempo peor, $\Theta(nlogn)$ tiempo promedio, asumiendo que cada permutación de la entrada es igualmente probable.
- Quicksort hecho azaroso: no se basa en una distribución de la entrada

Quicksort hecho azaroso

Dos enfoques posibles son:

- Uno es que aleatoriamente se permute el arreglo de entrada y entonces hacer el Quicksort determinítico.
- El otro es aleatoriamente escoger un elemento pivote en cada llamada recursiva, lo que es llamado "muestra aleatoria", y que corre en $\Theta(nlogn)$. En este enfoque, dado un arreglo $A[1 \ldots n]$, se llama a una rutina $RandQuickSort(A, 1, n)$

RandQuickSort(A, p, r)
 if $p < r$ then
  $q \leftarrow RandPartition(A, p, r)$
  $RandQuickSort(A, p, q - 1)$
  $RandQuickSort(A, q + 1, r)$
 endif
end

Quicksort hecho azaroso (cont.)

RandPartition(A, p, r)
$i \leftarrow$ índice elegido aleatoriamente entre p y r
intercambiar $A[r]$ y $A[i]$
return *Partition*(A, p, r)

Partition(A, p, r)
$x \leftarrow A[r]$ (pivote)
$i \leftarrow p1$
for $j := p$ **to** $r - 1$ **do**
if $A[j] \leq x$ **then**
$i \leftarrow i + 1$
intercambiar $A[i]$ y $A[j]$
endif
endfor
intercambiar $A[i + 1]$ y $A[r]$
return $i + 1$

Quicksort hecho azaroso: caso peor

Sea $T(n)$ el peor caso para el procedimiento con una entrada de tamaño n. Se tiene entonces la recurrencia:

$$T(n) = max_{0 \leq q \leq n-1}(T(q) + T(n-q-1)) + \Theta(n),$$

donde el parámetro q toma valores entre 0 y $n-1$, ya que *RandPartition* produce dos subproblema que en total suman $n-1$. Supongamos que $T(n) \leq cn^2$ para alguna constante c. Por sustitución:

$$\begin{aligned} T(n) &\leq max_{0 \leq q \leq n-1}(cq^2 + c(n-q-1)^2) + \Theta(n), \\ &= max_{0 \leq q \leq n-1} c(q^2 + (n-q-1)^2) + \Theta(n). \end{aligned}$$

La expresión $q^2 + (n-q-1)^2$ toma su máximo valor sobre $0 \leq q \leq n-1$ en los puntos extremos. De esta forma $max_{0 \leq q \leq n-1}(q^2 + (n-q-1)^2) \leq (n-1)^2 = n^2 - 2n + 1$. Luego:

$$T(n) \leq c^2 - c(2n-1) + \Theta(n) \leq cn^2.$$

Quicksort hecho azaroso: tiempo esperado

El tiempo esperado del quicksort es dominado por el tiempo en el procedimiento *RandPartition*. Cada vez que se llama a *RandPartition*, se tiene un elemento pivote seleccionado en forma aleatoria, el cual no será nunca más incluido en futuras llamadas al *RandQuickSort* y *RandPartition*. Por esta razón deben haber a lo más n llamadas al *RandPartition*. Una llamada al *RandPartition* toma O(1) más el tiempo que es proporcional al número de iteraciones que se hacen en *Partition*. Entonces si se puede acotar el número de iteraciones, se puede acotar el tiempo total de la ejecución de *RandQuickSort*.

Lema: Sea X el número de comparaciones que se realizan en *Partition* en la ejecución completa del *RandQuickSort* sobre n elementos. Entonces el tiempo de ejecución de *RandQuickSort* es $O(n + X)$.

Quicksort hecho azaroso: tiempo esperado (cont)

En lo que sigue, nos centraremos en calcular el valor esperado de X, determinando una cota total en el número total de comparaciones hechas.

Sea $z_1, \ldots, z_n$ los valores en el arreglo de entrada A, donde el elemento $i-$ésimo es el $i-$ésimo más pequeño y $Z_{ij} = \{z_i, \ldots, z_j\}$. Cada par de elementos es comparado a lo más una vez ya que las comparaciones se realizan respecto al pivote y el pivote se usa solo en una llamada a *Partition*.

Se define: X_{ij} como el indicador de que z_i es comparado a z_j,
donde se está considerando la comparación en cualquier momento durante la ejecución del algoritmo. Ya que cada par es comparado solo una vez, se puede caracterizar el total número de comparaciones X como:

$$X = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} X_{ij}$$

Quicksort hecho azaroso: tiempo esperado (cont)

$$
\begin{aligned}
E[X] &= E\left[\sum_{i=1}^{n-1} \sum_{j=i+1}^{n} X_{ij}\right] \\
&= \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} E[X_{ij}] \\
&= \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} Pr\{z_i \text{ es comparado a } z_j\}, . \qquad (2)
\end{aligned}
$$

Quicksort hecho azaroso: tiempo esperado (cont)

- Interesa saber cuándo dos elementos no serán comparados ya que permite determinar las posible comparaciones.
- Una vez que un pivote x, tal que $z_i < x < z_j$ es escogido, entonces z_i y z_j no serán nunca comparados en el futuro.
- Si z_i es escogido antes que cualquier otro elemento en $Z_{i,j}$, entonces z_i es comparado con todos los elementos en $Z_{i,j} - \{z_i\}$. Inversamente, si z_j es escogido antes que cualquier otro elemento en $Z_{i,j}$, entonces z_j es comparado con todos los elementos en $Z_{i,j} - \{z_j\}$.
- Entonces z_i será comparado a z_j si y solo si el primer elemento a ser escogido como pivote es z_i o z_j.

Quicksort hecho azaroso: tiempo esperado (cont)

Asumiendo que cada elemento de un subconjunto $Z_{i,j}$ es igualmente probable de ser escogido como primer pivote, entonces:

$$\begin{aligned}
Pr\{z_i \text{ es comparado a } z_j\} &= Pr\{z_i \text{ or } z_j \text{ es el primer pivote escogido de } Z_{ij}\} \\
&= Pr\{z_i \text{ es el primer pivote escogido de} Z_{ij}\} + \\
&\quad Pr\{z_j \text{ es el primer pivote escogido de} Z_{ij}\} \\
&= \frac{1}{j-i+1} + \frac{1}{j-i+1} = \frac{2}{j-i+1}
\end{aligned}$$

Luego:

$$\begin{aligned}
E[X] &= \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \frac{2}{j-i+1} \\
&= \sum_{i=1}^{n-1} \sum_{k=1}^{n-i} \frac{2}{k+1} < \sum_{i=1}^{n-1} \sum_{k=1}^{n} \frac{2}{k} \\
&= \sum_{i=1}^{n-1} O(logn) = O(nlogn).
\end{aligned}$$

Ejercicio

- Considere la siguiente estrategia de búsqueda de un elemento x en un arreglo desordenado $A[1 \dots n]$. Escoja un índice i aleatorio en A. Si $A[i] = x$, entonces termina, sino continue la búsqueda escogiendo un nuevo elemento. Notar que se puede examinar un elemento del arreglo más de una vez.
- Escriba el pseudocódigo. Asegúrese que la estrategia termina cuando todos los índices hay sido escogidos.
- Suponga que hay exactamente un índice tal que $A[i] = x$. ¿Cuál es el número esperado de índices que deben ser escogidos antes de encontrar x?
- Generaliza para $k > 1$ veces en que x está en A.
- ¿Qué sucede si no hay un índice?

Ejercicio (cont.)

- Considere el problema de generar grafos en forma aleatoria, dado un conjunto de vértices v y un *número* x de arcos. Indique el número de grafos posible y la probabilidad que cada grafo debiera tener para que sean creados en forma aleatoria.