Práctico 2 TEOCOMP: Fundamentos de probabilidad y algoritmos probabilísticos.

Mauricio Velasco

1. (*Estadística en el problema de selección de secretarie*) Implemente el Algoritmo de generación de permutaciones uniforme mediante sorting en S_n que vimos en clase.

   a) Escriba el código de su implementación

   b) Para $n = 20$ candidatos genere una muestra de $m = 100$ permutaciones y escriba las 5 primeras y las 5 últimas.

   c) Marque los candidatos en que la firmas deberían hacer contrataciones si el ranqueo de candidatos esta dado por las diez permutaciones que generó en el punto anterior (en el problema de selección de asistente que vimos en clase).

   d) Escriba un código que reciba una permutación y calcule el número de candidatos que la firma debería contratar en el problema de selección de asistente que vimos en clase).

   e) Para muestras de $m = 100, m = 500$ y $m = 1000$ permutaciones haga un dibujo que contenga, en un solo par de ejes:

      1) Un histograma del número de contrataciones realizadas en cada permutación de su muestra (ver por ejemplo https://www.w3schools.com/python/matplotlib_histograms.asp)

      2) Una recta vertical en el valor teórico que vimos en clase.

      3) Una recta vertical en el número promedio de contrataciones realizadas.

2. (*Puntos fijos en permutaciones aleatorias*) Recuerde que los puntos fijos de una permutación $\sigma \in S_n$ son aquellos índices $i \in \{1, \ldots, n\}$ con $\sigma(i) = i$.

   a) Para un entero j cualquiera defina la variable aleatoria $Y^{(j)} : S_n \to$

$\mathbb{R}$ dada por

$$Y^{(j)}(\sigma) = \begin{cases} 1, \text{ si } \sigma(j) = j \\ 0, \text{ de lo contrario} \end{cases}$$

Si $\mathbb{P}$ es la medida de probabilidad uniforme en S_n, calcule $\mathbb{E}[Y^{(j)}]$ dando un argumento preciso para su respuesta.

b) Use la parte (*a*) para encontrar el número esperado de puntos fijos de una permutación aleatoria de S_n, elegida uniformemente.

3. (*Probabilidad de éxito de nuestra construcción de permutaciones aleatorias*) Suponga que $p_1, \ldots p_N$ son variables aleatorias independientes, cada una uniforme en $\{1, 2, \ldots, N^3\}$.

   a) Demuestre que la probabilidad de que $(p_1, \ldots, p_N)$ no tenga repeticiones es por lo menos $1 - 1/n$.

   b) Una moneda cae cara con probabilidad p y sello con probabilidad $q = 1 - p$. Calcule el número esperado de lanzamientos hasta que la moneda caiga cara por primera vez.

   c) Combinando los dos items anteriores, calcule el número esperado de ejecuciones que debe hacer nuestro **Algoritmo de permutación uniforme mediante sorting** antes de que genere una permutación (recuerde que el algoritmo se ejecuta repetidas veces hasta que las prioridades p_i salgan todas distintas asi que el problema pregunta por el número esperado de intentos antes de que esto suceda). (Sugerencia: Mezcle las partes (*a*) y (*b*) y note que su respuesta debe depender de N).

4. (**Coincidencias planetarias**) Un planeta esta habitado por k habitantes $1, 2, \ldots, k$ y dá una vuelta a su estrella cada n días (donde un día es un giro del planeta alrededor de su propio eje).

   a) Para habitantes $i, j \in \{1, \ldots, k\}$ sea

   $$X^{(ij)} = \begin{cases} 1, \text{ si los habitantes } i \text{ y } j \text{ cumplen años el mismo día} \\ 0, \text{ de lo contrario.} \end{cases}$$

   Calcule $\mathbb{E}[X_{ij}]$ asumiendo que los cumpleaños se distribuyen de manera uniforme en los distintos días. Justifique su respuesta de manera precisa.

   b) Sea X la variable aleatoria que cuenta el número de parejas de los k individuos que cumplen el mismo día. Use la parte (*a*) para calcular $\mathbb{E}[X]$ justificando matemáticamente su respuesta.

c) Use el punto anterior para demostrar que si hay por lo menos $1+\sqrt{2n}$ individuos en un cuarto entonces deberíamos esperar que al menos dos tengan el mismo cumpleaños.

d) Convierta en resultados numéricos el cálculo anterior para el planeta tierra. Cómo explicaría este resultado en una frase sencilla?

5. (*Árboles generadores conectando puntos aleatorios*) Escriba un programa que construya 20 parejas de puntos aleatorios en el cuadrado $[0,1]\times[0,1]$ con coordenadas (X_i, Y_i) independientes y uniformente distribuidas en $[0,1]$ (ver https://numpy.org/doc/stable/reference/random/generated/numpy.random.uniform.html)

   a) Escriba el código de su implementación.

   b) Implemente una rutina que dibuje los puntos (por ejemplo mediante `pyplot`) y úsela para producir las gráficas (los veinte puntos) de tres muestras distintas.

   c) Defina un grafo con vértices $\{1,\ldots,20\}$ ponga aristas entre todo par de vértices con costos $c(i,j)$ dados por las distancias entre los puntos respectivos´, es decir

   $$c(i,j)=\sqrt{(X_i-X_j)^2+(Y_i-Y_j)^2}.$$

   Implemente un código que calcule el minimum spanning tree (MST) de un grafo de estos.

   d) Dibuje los MST de las tres muestras de la primera parte.

   e) Haga un histograma de la longitud $c(T)$ de estos MST para conjuntos de $m = 100, 200, 300$ muestras de 20 puntos y dibuje una recta vertical en el valor promedio de estas longitudes.

   f) Ver http://web.mit.edu/dbertsim/www/papers/AppliedProbability/The%20minimum%20spanning%20tree%20constant%20in%20geometrical%20probability%20and%20under%20the%20independent%20model%3B%20a%20unified%20approach.pdf para algo más de contexto.