Agulha de Buffon

Lauro Didier Lins

19 de Maio de 2004

Resumo

No século XVIII o matemático e naturalista francês Conde de Buffon propôs um interessante experimento. É sobre este experimento e sua simulação através de um programa na linguagem C que tratamos aqui.

1 Experimento

No século XVIII o matemático e naturalista francês Conde de Buffon estava interessado na probabilidade de uma agulha de comprimento ℓ lançada num plano marcado por linhas paralelas tocar numa destas linhas marcadas. Sabe-se que as linhas marcadas no plano estão distantes a ($a \geq \ell$) unidades das suas paralelas "vizinhas". Fixados a e ℓ, para sabermos se houve ou não choque da agulha com uma das linhas do plano, observamos duas quantidades: d, a distância entre o centro da agulha e a linha paralela do plano mais próxima a este centro, e θ, o ângulo que a agulha faz com uma reta passando pelo seu centro e paralela às retas do plano. É fácil ver que $d \in [0, \frac{a}{2}]$, pois qualquer ponto no plano está a uma distância máxima de $\frac{a}{2}$ para a reta mais próxima. Além disso, para qualquer agulha no plano, podemos ter uma figura (como a Figura 1) onde a reta mais próxima ao centro da agulha não aparece completamente abaixo da agulha (no caso da Figura 1, a reta mais próxima aparece completamente acima da agulha). Para isto, numa situação em que a agulha esteja completamente acima da paralela mais próxima a seu centro (e.g. a Figura 1 de cabeça pra baixo), basta girar a figura 180^o. Ou seja, toda configuração de agulha no plano tem um valor $\theta \in [0, \pi]$. Uma vez conhecidos ℓ, a, d e θ saber se a agulha toca alguma linha paralela do plano é equivalente a verificar se a inequação

$$d \leq \frac{\ell}{2} \sin(\theta)$$

é verdadeira.

Dada uma situação de agulha no plano, já sabemos definir se houve ou não choque da agulha com uma das linhas paralelas. Porém, Buffon não estava

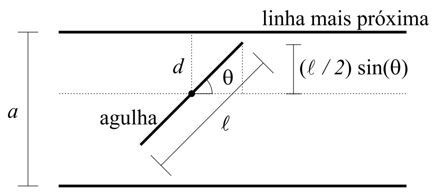

Figura 1: agulha no plano e as medidas d e θ

interessado em saber se houve choque numa situação específica, mas sim qual a probabilidade de uma agulha lançada no plano tocar uma das paralelas. Com o que já formulamos, sabemos que as configurações de uma agulha no plano podem ser vistas como os pares do conjunto $[0, \frac{a}{2}] \times [0, \pi]$. Uma boa hipótese para se admitir é que a natureza não tem preferência por nenhuma configuração de agulha no plano e que θ não interfere em d nem d interfere em θ [2]. Seguindo esta hipótese, podemos dizer que a menor distância d do centro da agulha para a sua paralela mais próxima é uma variável aleatória D com distribuição uniforme, $D \sim \mathcal{U}_{(0,\frac{a}{2})}$, e o ângulo θ é também uma variável aleatória Θ com distribuição uniforme, $\Theta \sim \mathcal{U}_{(0,\pi)}$. Além disso D e Θ são independentes. Feitas estas suposições podemos caracterizar completamente o vetor aleatório (D, Θ). A probabilidade do evento $\Omega \subseteq [0, \frac{a}{2}] \times [0, \pi]$ é dada por

$$P[\omega \in \Omega] = \int_{\substack{\omega\in\Omega \\ \omega=(x,y)}} f_D(x) f_\Theta(y)\, dx\, dy = \int_{\omega\in\Omega} \frac{2}{a}\frac{1}{\pi}\, dx\, dy = \frac{2}{a\pi}\int_{\omega\in\Omega} dx\, dy,$$

onde f_D é a densidade de D e f_Θ é a densidade de Θ. Agora podemos definir a variável aleatória H como função do vetor aleatório (D, Ω)

$$H = \begin{cases} 1, & D \leq \frac{\ell}{2}\sin(\Theta) \\ 0, & \text{caso contrário.} \end{cases}$$

Note que $H = 1$ exatamente nas situações onde há choque da agulha com uma das paralelas e que $H = 0$ quando não há choque. Portanto a probabilidade de haver choque é $P[H = 1]$. Olhando para a Figura 2, podemos notar em cinza a região exata do suporte da densidade da distribuição (D, Θ) em que $H = 1$.

Segue então que

$$P[H = 1] = P[D \leq \frac{\ell}{2}\sin(\Theta)] = \frac{2}{a\pi}\int_{y=0}^{\pi}\int_{x=0}^{\frac{\ell}{2}\sin(y)} dx\, dy = \frac{2}{a\pi}$$

$$\frac{2}{a\pi}\int_{y=0}^{\pi}\frac{\ell}{2}\sin(y) dy = \frac{2}{a\pi}\frac{\ell}{2}(\cos(0) - \cos(\pi)) = \frac{2\ell}{a\pi}.$$

Podemos também calcular a esperança de H

$$E(H) = 1P[H = 1] + 0P[H = 0] = P[H = 1] = \frac{2\ell}{a\pi}.$$

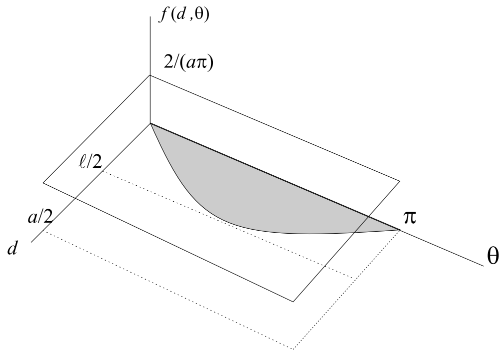

Figura 2: densidade conjunta de (D, Θ) e a região (em cinza) onde a agulha toca a linha mais próxima a ela.

Pela lei dos grandes números, se $H_1, \ldots H_n$ é uma amostra independente e identicamente distribuída à distribuição de H, então

$$\frac{H_1 + \ldots + H_n}{n} \xrightarrow{P} \frac{2\ell}{a\pi} = E(H)\,,$$

que é equivalente a

$$\frac{n}{H_1 + \ldots + H_n}\frac{2\ell}{a} \xrightarrow{P} \pi. \tag{1}$$

Na convergência em probabilidade desta última fórmula, podemos concluir o seguinte resultado: se repetirmos n vezes (para n grande) o lançamento da agulha de Buffon e anotarmos o número de vezes h em que a agulha tocou numa das paralelas do plano então

$$\frac{n}{h}\frac{2\ell}{a} \approx \pi.$$

Ou seja, Buffon descobriu um experimento bastante simples para aproximar o número π. Na época em que não havia computadores e seus geradores de números pseudo-aleatórios, lançar agulhas era uma alternativa viável para aproximar π.

2 Simulação

Para simular o experimento da Agulha de Buffon fizemos um programa em C. Este programa realiza o experimento para $a = \ell = 1$ e exibe a aproximação para π após 10^1, 10^2, 10^3, 10^4, 10^5, 10^6, 10^7, 10^8 e 10^9 lançamentos de agulha. O gerador de números pseudo-aleatórios da distribuição uniforme que utilizamos na simulação dos lançamentos é devido a George Marsaglia. O programa foi compilado e executado no Windows XP e o compilador utilizado foi o GCC [3].

Os resultados obtidos para a simulação estão na Tabela 1. Nela podemos verificar a convergência (1). A medida que n cresce a diferença absoluta entre o valor real de π e o valor aproximado $\hat{\pi}$ diminui. Entretanto, esta queda não é rápida. Por exemplo, após 5 minutos de computação e 1 bilhão de lançamentos a aproximação ainda foi de apenas 3 casas decimais.

n	h	$\hat{\pi}$	$\lvert\pi - \hat{\pi}\rvert$	tempo(seg.)
10	7	2.8571429	0.2844498	0.00
100	69	2.8985507	0.2430420	0.00
1000	646	3.0959752	0.0456175	0.00
10000	6407	3.1215858	0.0200069	0.00
100000	63941	3.1278835	0.0137092	0.03
1000000	637499	3.1372598	0.0043329	0.30
10000000	6365076	3.1421463	0.0005536	3.11
100000000	63652509	3.1420600	0.0004673	31.48
1000000000	636564070	3.1418676	0.0002749	314.84

Tabela 1: n - número de lançamentos; h - número de lançamentos que tocou numa paralela; $\hat{\pi}$ - estimativa para π.

A Código Fonte

A.1 buffon.c

```
/*********************************************************
 * PROGRAM: buffon.c
 * DATE: 15-05-2004
 * DESC.: APROXIMATE PI BY SIMULATING BUFFON'S
 *        NEEDLE EXPERIMENT.
 *********************************************************/

#include <time.h>   /* for timing the code */
#include <stdio.h>  /* for writing to files */
#include <math.h>   /* for exponentials and logs */
```

```
/* George Marsaglia´s uniform random number generator */
/* (has a very large period, > 2^60, and passes Diehard tests; */
/* uses a multiply-with-carry method) */
#define s1new (s1=(18000*(s1&0xFFFF)+(s1>>16)))
#define s2new (s2=(30903*(s2&0xFFFF)+(s2>>16)))
#define UNI (((s1new<<16)+(s2new&0xFFFF))*2.32830643708e-10)
unsigned long s1=362436069, s2=521288629;
#define setseed(seed1,seed2) {s1=seed1;s2=seed2;}

int main(void)
{
   double theta, distance;

   /* time store locations */
   clock_t startTime, endTime;

   /* set seed of UNI */
   setseed(362436069,521288629);

   /* log Pi aproximation on these iterations */
   long log_points[] = { 10, 100, 1000, 10000, 100000,
     1000000, 10000000, 100000000, 1000000000};
   int numLogPoints = 9;

   /* start time count */
   startTime = clock();

   /* simulate throws of a 1 unit needle on a sheet with one */
   /* unit distant parallel lines and acumulate the number of */
   /* hits (i.e. needle crossing/touching a line) */
   long i=1;
   long hits = 0;
   int logIndex = 0;
   while (logIndex < numLogPoints) {

      /* set next log point */
      long nextLogPoint = log_points[logIndex];
      logIndex++;

      /* iterate */
      for (;i<=nextLogPoint;i++) {

         /* get a sample from a U(0,PI) */
         theta = UNI * M_PI;

         /* get a sample from a U(0,0.5) */
         distance = UNI * 0.5;

         /* see if it falls on the hit area */
         if (distance <= 0.5 * sin(theta))
            hits++;

      }

      /* elapsed time */
```

```
      endTime = clock();

      /* log current aprox*/
      printf("NThrows: %11d Hits: %11d Pi Aprox.: %1.7f (%8.2f s.)\n",
         nextLogPoint,
         hits,
         (2.0*nextLogPoint)/hits,
         ((double)(endTime-startTime))/CLOCKS_PER_SEC);
   }
   return(0);
}
```

A.2 buffon.out

```
NThrows:         10 Hits:          7 Pi Aprox.: 2.8571429 (    0.00 s.)
NThrows:        100 Hits:         69 Pi Aprox.: 2.8985507 (    0.00 s.)
NThrows:       1000 Hits:        646 Pi Aprox.: 3.0959752 (    0.00 s.)
NThrows:      10000 Hits:       6407 Pi Aprox.: 3.1215858 (    0.00 s.)
NThrows:     100000 Hits:      63941 Pi Aprox.: 3.1278835 (    0.03 s.)
NThrows:    1000000 Hits:     637499 Pi Aprox.: 3.1372598 (    0.30 s.)
NThrows:   10000000 Hits:    6365076 Pi Aprox.: 3.1421463 (    3.11 s.)
NThrows:  100000000 Hits:   63652509 Pi Aprox.: 3.1420600 (   31.48 s.)
NThrows: 1000000000 Hits:  636564070 Pi Aprox.: 3.1418676 (  314.84 s.)
```

Referências

[1] Rao, C.R. (1997). *Statistics And Truth: Putting Chance to Work*. 2.ed. World Scientific.

[2] Buffon's Needle. URL: http://whistleralley.com/buffon/buffon.htm Consultado em 15 maio de 2004.

[3] GCC Home Page - GNU Project - Free Software Foundation. URL: http://gcc.gnu.org Consultado em 10 maio 2004.