

Metode Simulasi Monte Carlo

Review Algoritma

Probabilitas pelemparan coin Tunggal

Probabilitas pelemparan coin Ganda

Nilai π

Nilai Integral

Kasus nilai $f(x) = x\sin(x)$

Oleh:

Tri Budi Santoso

Achmad Basuki

Miftahul Huda

EEPIS-ITS

Speech Signal Processing



Review Algoritma Monte Carlo

- Merupakan dasar dari semua algoritma untuk metode simulasi
- Didasari pada pemikiran penyelesaian suatu masalah untuk mendapatkan hasil lebih baik dengan cara memberi alternatif nilai sebanyak-banyaknya (nilai terbangkit) untuk mendapatkan tingkat ketelitian yang lebih tinggi
- Misal untuk memperoleh tingkat ketelitian sampai 0,01 maka diperlukan pembangkitan nilai sebanyak 10000, dsb.
- Teknik pembuatan program bersifat bebas hampir tidak ada rule yang terlalu mengikat

Setiap masalah simulasi dapat didekati dengan metode Monte Carlo ?

Dengan catatan kunci:

- Mampu memformulasikan masalah
- Membuat Overview Sistem
- Penyederhanaan sistem menuju algoritma
- Menyusun algoritma
- Pembuatan Program

Probabilitas pelemparan coin Tunggal

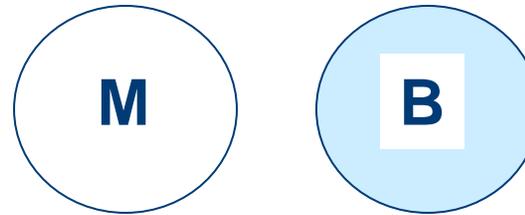
Sebuah pelemparan coin sebanyak 100 kali diperoleh hasil 45 kali keluar muka dan 55 kali keluar belakang. Dari data ini didapat nilai probabilitas untuk keluarnya

- muka sebagai $p(M) = N(M)/N_{total}$
- belakang sebagai $p(B) = N(B)/N_{total}$

Dari model diatas susun algoritma dan program untuk kasus pelemparan sebanyak 1000 kali 2000 kali dsb... sampai dengan 10000 kali.

Apa yang terjadi?

Algoritma



1. Bangkitkan nilai 0/1 sebanyak 1000 kali ($N=1000$) dengan cara:
 $n = (\text{int})\text{rand}()\%2$
2. Klasifikasi
Jika $n=0$, maka $M=M+1$
Jika $n=1$, maka $B=B+1$
3. Hitung probabilitas M dengan cara M/N dan probabilitas B dengan cara B/N

```

#include<stdio.h>
#include<stdlib.h>

#define N 1001
int nn=N-1;
int x[N];

void bangkit()
{
for (int i=0;i<nn; i++)
{
x[i]=(int)rand()%2;
//printf("\n x[%d]: %d",i,x[i]);
}
}

```

```

void klasfikasi()
{ double M=0.0,B=0.0;

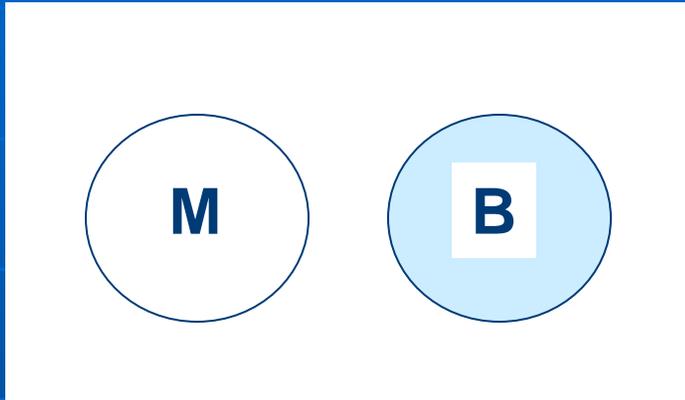
for (int i=0;i<nn; i++)
{
if(x[i]==0)
M=M+1;
else
B=B+1;
}
printf("\n Nilai Terbangkit");
printf("\n M=%f \t B=%f",M,B);
printf("\n\n Probabilitas");
printf("\n p(M)=%f \t p(B)=%f",M/nn,B/nn);
}

void main()
{
bangkit();
klasfikasi();
}

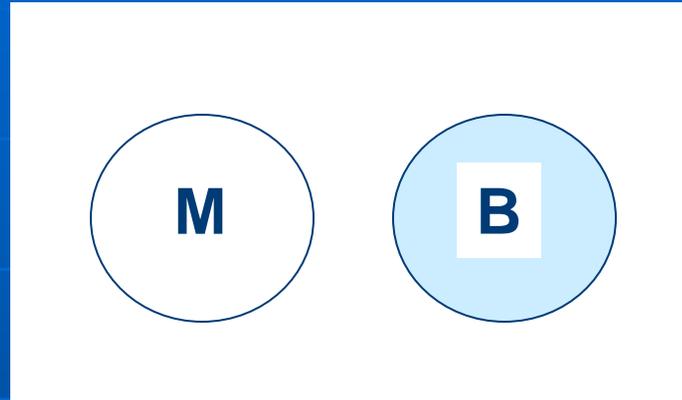
```

Probabilitas pelemparan coin Ganda

Coin 1



Coin 2



Dari teori peluang akan muncul:

MM $\rightarrow \frac{1}{4}$

MB atau BM $\rightarrow \frac{1}{2}$

BB $\rightarrow \frac{1}{4}$

Dengan metode Monte Carlo
dapatkan tingkat ketelitian sampai
0.01 untuk menyelesaikan kasus
tersebut....

Model Sistem menuju Algoritma

Untuk mendapatkan ketelitian sampai 0,01 maka harus dilakukan pelemparan sebanyak 1000 (N_{total}) kali.
Dari hasil pelemparan catat keluaranya angka-angka:

$$N(MM) = \sum (MM) = \dots\dots kali$$

$$N(MB) = \sum (MB) = \dots\dots kali$$

$$N(BB) = \sum (BB) = \dots\dots kali$$

Dari hasil diatas hitung peluang dengan cara:

$$P(MM) = N(MM)/N_{total}$$

$$P(MB) = N(MB)/N_{total}$$

$$P(BB) = N(BB)/N_{total}$$

Algorithma

1. Bangkitkan nilai 0/1 sebanyak 1000 kali ($N=1000$) dengan cara:

$n1 = (\text{int})\text{rand}()\%2$ dan $n2 = (\text{int})\text{rand}()\%2$

2. Klasifikasi

Jika $n1=0$ dan $n2=0$, maka $MM=MM+1$

Jika $n1=0$ dan $n2=1$ atau $n1=1$ dan $n2=0$
maka $MB=MB+1$

Jika $n1=1$ dan $n2=1$, maka $BB=BB+1$

3. Hitung probabilitas MM dengan cara $N(MM)/N$
dan probabilitas untuk nilai MB serta BB