OSNOVE PROGRAMIRANJA U PAJTONU

PREDAVANJE 10:
ALGORITMI ZASNOVANI NA SLUČAJNIM BROJEVIMA

Miloš Kovačević

Đorđe Nedeljković

Marija Petronijević

Dušan Isailović

SADRŽAJ PREDAVANJA

- Pseudoslučajni brojevi
- Funkcije za slučajni odabir i mešanje
- Monte Karlo simulacija

DETERMINISTIČKI I NEDETERMINISTIČKI ALGORITMI

Algoritam koji daje uvek iste izlaze, za iste ulaze,
naziva se deterministički (svi dosadašnji algoritmi).

Realni sistemi su često prilično neodređeni – ne mogu se modelovati determinističkim (potpuno određenim) algoritmom.

Primer: teniski turnir (bolji igrač obično pobeđuje lošijeg).

Da bi se modelovali neodređeni sistemi,
u algoritmu mora da postoji izvor slučajnosti!

Algoritam koji, u ponvljenom izvršavanju, daje različite izlaze za isti ulaz je nedeterministički.

Primer: bolji teniser A pobeđuje lošijeg tenisera B, sa verovatnoćom 0.7

VEROVATNOĆA DOGAĐAJA

Izvesnost dešavanja nekog događaja A u matematici se kvantifikuje brojem P(A):
$P(A) \in [0, 1]$, P(A) je verovatnoća dešavanja za A.

Verovatnoća se neformalno definiše kao odnos broja povoljnih
i broja svih mogućih ishoda za događaj.

Primer: baca se fer kocka za jamb. Koja je verovatnoća da padne broj veći od 4?

Događaj A – pao je broj > 4. Povoljni ishodi $\{5, 6\}$. Svi ishodi $\{1, 2, 3, 4, 5, 6\}$.
$P(A) = 2/6 = 1/3 = 0.333\ldots$

Što je događaj izvesniji njegova verovatnoća je bliža 1.
Ako je $P(A) = 1$, događaj je siguran. Ako je $P(A) = 0$, događaj je nemoguć.

PRESEK DVA DOGAĐAJA. NEZAVISNOST

Dati su događaji A i B. Događaj AB realizuje se samo ako su se desili i A i B.
P(AB) – broj povoljnih ishoda i za A i za B, podeljen sa ukupnim brojem ishoda.

Primer: baca se fer kocka. A – pao broj > 3. B – pao paran broj.
Povoljni za A: {4, 5, 6}; povoljni za B {2, 4, 6}; povoljni za AB: {4, 6}
P(AB) = 2/6 = 1/3

Dva događaja su nezavisna ako dešavanje jednog ne utiče na dešavanje drugog i obrnuto (nema uzročno posledične veze). Primer: ishodi u dva uzastopna bacanja.

Matematička formulacija nezavisnosti:
A i B nezavisni ako i samo ako P(AB) = P(A) P(B)

Da li su A i B iz prethodnog primera nezavisni?

SLUČAJNA PROMENLJIVA

Slučajna promenljiva – veličina koja karakteriše ishod nekog slučajnog procesa, a koja uzima brojne vrednost iz nekog skupa S.

Primer: broj pisama posle 10 bacanja novčića. Broj pisama $X \in [0, 10]$

Ako je S prebrojiv onda se radi o diskretnoj slučajnoj promenljivoj.
Ako je S neprebrojiv, onda se radi o kontinualnoj slučajnoj promenljivoj.

Primer za broj pisama - X je diskretna slučajna promenljiva

Neprekidne slučajne prom. imaju beskonačno mnogo mogućih vrednosti – na primer, visina slučajno odabranog studenat GRF (između 0 i 250cm)

Stanje realnog sistema se često opisuje većim brojem različitih, neprekidnih slučajnih promenljivih (promenljive stanja).

DISKRETNA RASPODELA

Slučajne promenljive opisuju se raspodelom verovatnoće.

Diskretna raspodela opisuje verovatnoću da slučajna promenljiva uzme vrednost $X = x_i$: p_i = P(X= x_i) i zadata je tabelarno

Primer: X je broj dečaka u porodici sa troje dece:

X	**0**	**1**	**2**	**3**
$P(X)$	1/8	3/8	3/8	1/8

Zbir svih verovatnoća mora bit jednak 1!

Tabela 6.2: Diskretna raspodela verovatnoća: broj dečaka u tročlanoj porodici. Na primer, ako je $X = 2$, moglo je biti $110, 101$ ili 011 (1 na i-toj poziciji označava da je i-to rođeno dete dečak). Kako su verovatnoće rođenja dečaka i devojčice jednake, te kako su ishodi na svakoj poziciji međusobno nezavisni, važi $P(X = 2) = 3 \cdot \frac{1}{2}\frac{1}{2}\frac{1}{2} = \frac{3}{8}$.

NEPREKIDNA RASPODELA

Kolika je verovatnoća da neprekidna X uzme tačnu vrednost x_i - $P(X = x_i)$?

Primer: X visina studenta, P(X = 186cm) = ?

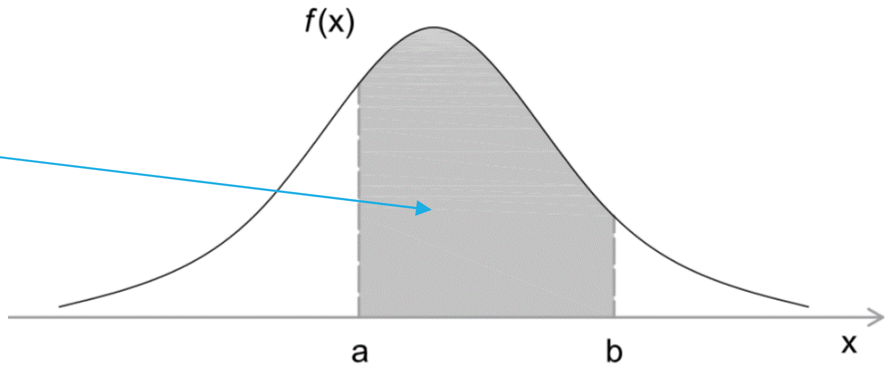
Verovatnoća ovde može dobiti geometrijsku interpretaciju:
bira se tačno jedna tačka na intervalu od 0 do 250cm. Broj povoljnih ishoda odgovara dužini intervala koju čini tačno jedna tačka, na primer 186cm (0), a broj svih mogućih ishoda odgovara dužini intervala [0, 250].
Sledi P(X = 186cm) = 0/250 = 0;

Verovatnoća za svaki pojedinačan ishod je 0!

Verovatnoća da X uzme vrednost iz [a, b] defineše se preko funkcije gustine raspodele:

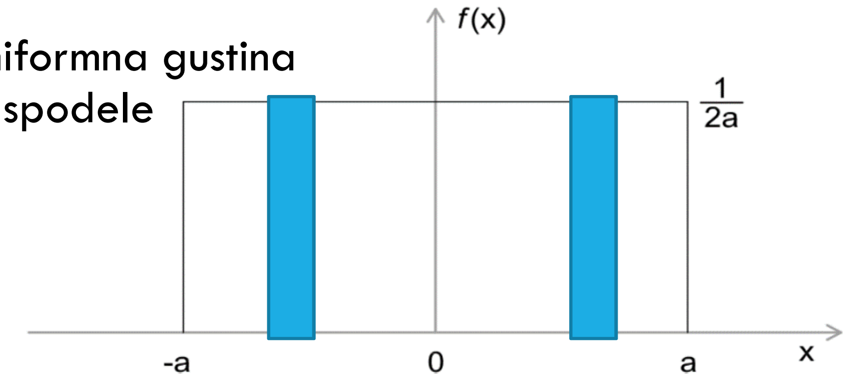
$$P(a \leq X \leq b) = \int_a^b f(x)dx$$

$$P(-\infty \leq X \leq \infty) = \int_{-\infty}^{\infty} f(x)dx = 1$$



SLUČAJNI BROJEVI

uniformna gustina raspodele



Sekvenca brojeva *S* je slučajna ako važi:

1) brojevi iz *S* su uniformno raspoređeni u odnosu na *S* (jednake verovatnoće izbora)

2) Za svako n važi da, za poznatih prvih *n-1* brojeva, nije moguće odrediti sledeći

Računar ne može generisati slučajnu sekvencu u determinističkom postupku, sem ako se, na ulaz algoritma, ne dovede slučajna veličina iz fizičkog sveta.

U programiranju se koriste pseudoslučajni brojevi koji ne ispunjavaju uslov (2).
Za početni broj (seme), pseudoslučajna sekvenca je potpuno poznata i ponavlja se posle konačno mnogo brojeva!

Pseudoslučajna sekvenca je, na izgled, dovoljno slučajna za najveći broj primena!

PSEUDOSLUČAJNI BROJEVI U PAJTONU – MODUL `random`

Modul pruža različite funkcije koje rade sa generatorom pseudoslučajnih brojeva

funkcija	**opis**
`seed(s)`	inicijalizuje generator semenom s. Ako se seme izostavi, uzima se trenutno sistemsko vreme
`random()`	vraća sledeći uniformno raspoređeni realni broj na intervalu $[0,1)$
`uniform(a,b)`	vraća sledeći uniformno raspoređeni realni broj na intervalu $[a,b]$
`gauss(mu, sigma)`	vraća sledeći normalno raspoređeni realni broj iz $N(mu,sigma)$
`randrange(n)`	vraća sledeći uniformno raspoređeni ceo broj na intervalu $[0,n)$
`randint(a,b)`	vraća sledeći uniformno raspoređeni ceo broj na intervalu $[a,b]$

PSEUDOSLUČAJNI BROJEVI U PAJTONU – MODUL `random`

```
>>> import random as r
>>> [r.random() for i in range(3)]
[0.033646681621365304, 0.2320163346900348, 0.8893950894419019]
>>> r.seed(123)
>>> [r.random() for i in range(3)]
[0.052363598850944326, 0.08718667752263232, 0.4072417636703983]
>>> r.seed(123)
>>> [r.random() for i in range(3)]
[0.052363598850944326, 0.08718667752263232, 0.4072417636703983]
>>> [r.uniform(5, 9) for i in range(3)]
[5.15261466440929, 7.144808160135708, 6.328790794038719]
>>> [r.gauss(0, 1) for i in range(3)]
[0.3529214268998316, -0.47259875675165125, -0.4695527782004237]
>>> [r.randrange(10) for i in range(20)]
[0, 6, 8, 6, 8, 9, 7, 8, 6, 5, 8, 0, 2, 1, 7, 4, 2, 5, 5, 8]
>>> [r.randint(2,4) for i in range(20)]
[2, 2, 2, 3, 3, 4, 3, 3, 3, 2, 3, 3, 4, 3, 2, 4, 4, 4, 4, 3]
```

isto seme
ista sekvenca

Problem 6.1 — Pogodi lutajući broj. Računar "zamišlja" broj od 1 do n, a igrač se trudi da ga pogodi. Ako je igrač neuspešan u pogađanju, računar, posle svakih $m \leq n$ pokušaja, bira novi broj iz istog intervala. Realizovati igru pogađanja, uz ispis broja pokušaja i mogućnost ponovnog igranja. ■

```python
import random as r
def partija(n, m):

    pokusaji = 0
    x = r.randint(1, n)

    while True:

        p = int(input('Vaš broj? '))
        pokusaji += 1

        if p < 1 or p > n:
            return -1
        elif p == x:
            return pokusaji
        elif pokusaji % m == 0:
            x = r.randint(1, n)
            print('Menjam broj!')
```

generiše uniformni ceo broj iz [1, n]

kada želi da prekine, igrač unese broj van intervala

FUNKCIJE ZA SLUČAJNI ODABIR I MEŠANJE

Uz pomoć funkcija iz modula `itertools`, biraju se kombinatorne sturkture objekata iz kolekcije.

Za biranje slučajnih uzoraka objekata iz kolekcije,
koriste se funkcije modula `random`

Funkcija `choice()` bira tačno jedan,
a `sample()` više objekata iz sekvence ili skupa (biranje bez vraćanja).

```
>>> import random as r
>>> kockica = [1, 2, 3, 4, 5, 6]
>>> r.choice(kockica)   # jedno bacanje
5
>>> r.sample(kockica, 3)  # slucajni uzorak od 3 elementa
[1, 4, 3]
```

FUNKCIJE ZA SLUČAJNI ODABIR I MEŠANJE

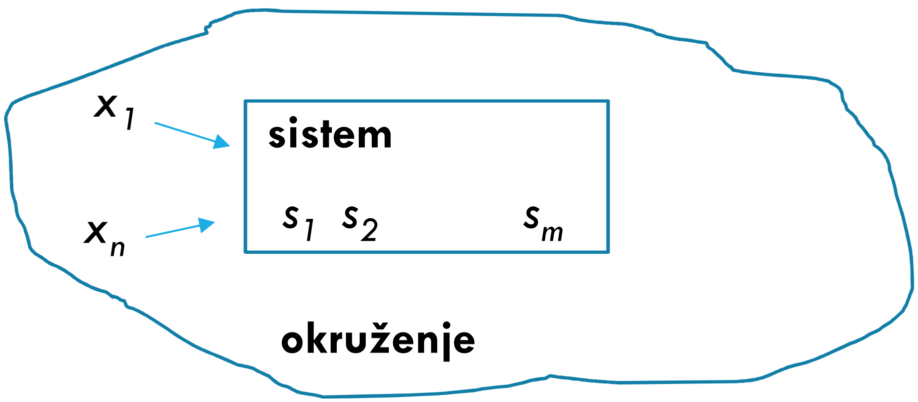
Funkcija `shuffle()` meša promenljive sekvence poput liste.

```
>>> import random as r
>>> samoglasnici = ['a', 'e', 'i', 'o', 'u']
>>> r.shuffle(samoglasnici)
>>> samoglasnici
['o', 'e', 'a', 'u', 'i']
```

PONAŠANJE REALNIH SISTEMA

Podsećanje: programiranje predstavlja vid modelovanja realnih sistema.

Ponašanje sistema zavisi od stanja u kome se on nalazi.
Stanje sistema zavisi od ulaznih fiz. veličina koje na sistem deluju iz okruženja.



s_i – promenljive stanja

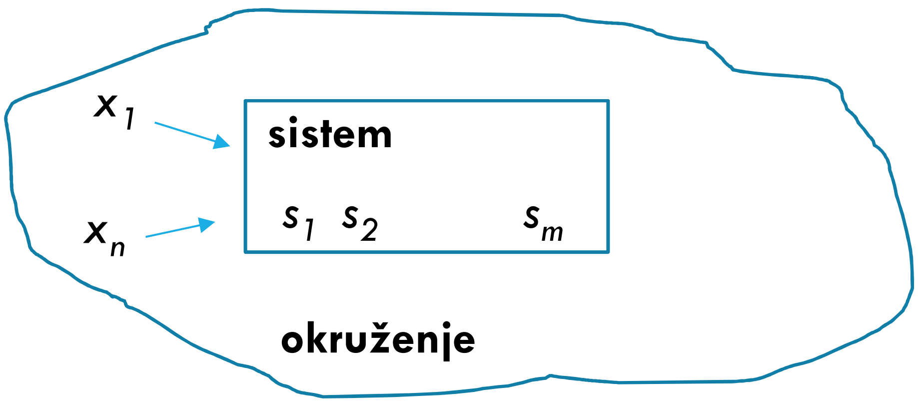
$$S_i = f_i(X_1, X_2, \ldots, X_n), i = 1, \ldots, m$$

f_i – determinističke funkcije
X_i – slučajne promenljive

Analiza ponašanja sistema na različite ulaze često je teška, dugotrajna, skupa ili neizvodljiva – pravi se računarski model i vrši simulacija.

PROBABILISTIČKI ALGORITMI: MONTE KARLO SIMULACIJA

Nedeterministički postupci za približno rešavanje problema – probabilistički algoritmi (koriste generatore pseudoslučajnih brojeva).



s_i – promenljive stanja

$$S_i = f_i(X_1, X_2, \ldots, X_n), i = 1, \ldots, m$$

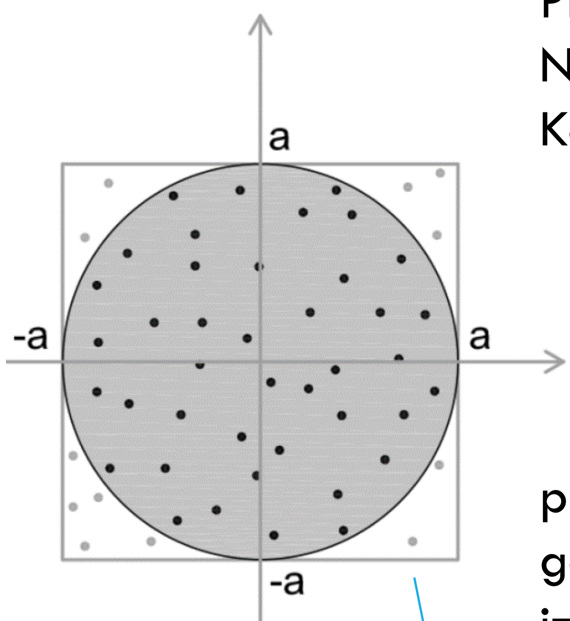
f_i – determinističke funkcije

X_i – slučajne promenljive

Monte Karlo simulacija: podrazumeva da su poznate raspodele za X_i pa u velikom broju opita generiše različite X_i i posmatra ponašanje sistema.

Problem 6.4 — Površina kruga i zaboravljeno π. Dat je krug K, poluprečnika a, sa centrom u $(0,0)$. Izračunati površinu kruga K, bez upotrebe broja π!

Pretpostavimo da je oko kruga opisan kvadrat stranice $2a$.
Neka se iz kvadrata na slučajan način bira tačka.
Koja je verovatnoća p da tačka pripada i krugu?

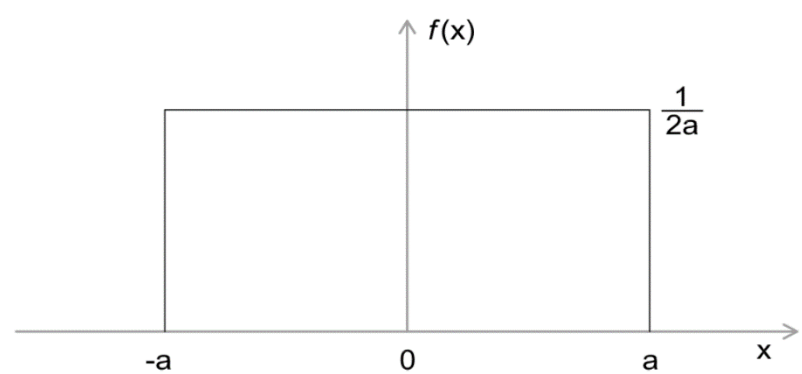


Verovatnoća p je P_{kruga} / $P_{kvadrata}$.
Ako je poznata, onda se P_{kruga} lako računa bez pi !

p se može proceniti u Monte Karlo simulaciji:
generiše se N parova koordinata tačaka iz intervala [-a, a]
iz uniformne raspodele, pa se prebroje tačke u krugu (n).
Sada je p = n / N

uslov pripadnosti tačke krugu

$$x_i^2 + y_i^2 \leq a^2 \implies (x_i, y_i) \in K$$



POVRŠINA KRUGA – MONTE KARLO

```
import random as r
import math

def površina(a, n):
    ''' Monte Karlo metoda: površina kruga poluprečnika a,
        n slučajnih tačaka u [-a, a]'''
    unutra = 0
    for i in range(n):
        x, y = r.uniform(-a, a), r.uniform(-a, a)
        if x**2 + y**2 <= a**2:
            unutra += 1
    return 4 * a**2 * unutra / n
```

jedan opit

POVRŠINA KRUGA – MONTE KARLO

```
# test primer
a = float(input('poluprečnik? '))
p_stvarno = a**2 * math.pi

print('{:<9}\t{:<9}\t{:<9}'.format('n', 'Pk', 'rel. greška'))
for i in range(7):
    n = 10**i
    p = površina(a, n)
    rel = abs(p - p_stvarno) / p_stvarno # relativna greška
    print('{:<9}\t{:<9}\t{:<9}'.format(n, p, rel))
```

Kad broj opita raste
rel. greška se smanjuje

```
poluprečnik? 10
n          Pk         rel. greška
1          400.0      0.27323954473516265
10         360.0      0.1459155902616464
100        316.0      0.00585924034078516
1000       308.4      0.018332311009189646
10000      314.56     0.0012755779797319375
100000     314.692    0.001695747029494557
1000000    314.212    0.00016785957581230386
```