Modely a modelování

Definice modelu:

Model je napodobenina originálu nebo jeho popis pomocí vymezených prvků a vztahů mezi nimi.

Vlastnosti dobrého modelu:

Validita – schopnost modelu vystihnout podstatné vlastnosti originálu. Je vyjádřena stupněm shody modelu s originálem v podstatných vlastnostech.

Reliabilita – míra spolehlivosti a opakovatelnosti modelu. Vysoká reliabilita znamená, že výsledek modelování je jen nepatrně ovlivněn náhodou a podstatné vlivy jsou v modelu zohledněny. Reliabilní model také umožňuje při opakovaném použití za stejných podmínek získat podobné výsledky.

Druhy modelů:

- fyzikální model: fyzicky vytvořený objekt, který napodobuje originál (model automobilu, města, stroje, atomu, …)
- abstraktní model: soubor vzorců a pravidel, které zjednodušeně popisují originál. Zjednodušení spočívá ve výběru těch typických vlastností originálu, které jsou významné pro popis jeho chování.

Druhy abstraktních modelů:

- deterministický model – prvky a vztahy mezi nimi jsou pevně dány. Chování modelu za určitých podmínek je plně předvídatelné (např. modely různých fyzikálních dějů)
- stochastický model – prvky modelu a vztahy mezi nimi mají charakter náhodných jevů či náhodných veličin nebo náhodných procesů. V modelu vystupuje jedna nebo více náhodných složek (např. regresní model vysvětlující závislost jedné veličiny na jiných veličinách).

Příklady stochastických modelů:

- modely systémů hromadné obsluhy
- markovské řetězce a procesy popisující vývoj dynamických systémů
- modely zásob (umožňují optimalizovat velikost skladu)
- modely obnovy (popisují proces náhrady těch prvků, které selhaly)
- modely přežití (využití ve zdravotnictví, pojišťovnictví, průmyslu)
- modely nabídky a poptávky

Speciální metody pro práci se stochastickými modely:

- stochastické programování
- stochastická teorie řízení
- simulační metody

Specifika simulačních metod:

- používají se při studiu složitějších dějů stochastické povahy, kde se analytické řešení získává jen obtížně
- pro experimenty jsou použita náhodná čísla. Tak lze simulovat průběhy reálných dějů při různých hodnotách parametrů a pozorovat vlastnosti modelovaného děje.

Možné problémy při simulacích:

- neadekvátní popis modelovaného děje
- nesprávná volba pravděpodobnostních rozložení náhodných veličin vyskytujících se v modelu
- chybné odhady parametrů
- nekvalitní generátory pseudonáhodných čísel.

Generátory náhodných čísel

Definice náhodné posloupnosti čísel:

Posloupnost čísel je náhodná, pokud neexistuje kratší způsob vyjádření dané posloupnosti než tato posloupnost samotná, tj. tuto posloupnost nemůžeme zhustit do kratší podoby. Proto žádná posloupnost čísel vzniklých výpočtem nemůže být náhodná, ale jen pseudonáhodná.

Náhodná čísla vznikají např. jako výsledky náhodných fyzikálních procesů. Nelze předvídat výskyt určité hodnoty a jednotlivé hodnoty jsou nezávislé.

Náhodná čísla potřebujeme např. v těchto situacích:

- generování náhodných výběrů
- simulace fyzikálních procesů
- simulované úlohy používané k řešení složitých matematických problémů, které nelze řešit analyticky
- modelování dějů, kde se vyskytují náhodné jevy nebo náhodné veličiny
- kryptografie
- počítačové hry
- atd.

Způsoby generování náhodných čísel

Mechanický způsob: losování čísel z osudí, házení kostkami, ruleta. Pro většinu aplikací je tento způsob příliš pomalý.

Fyzikální způsob: je založen na měření určitého jevu, který nastává náhodně (např. rozpad radioaktivní látky). Lze také v pravidelných intervalech měřit šum na polovodičových přechodech. Problémem je připojení generátoru šumu k počítači.

Použití tabulek náhodných čísel: K jejich tvorbě se používaly rozsáhlé soubory dat získané k jiným účelům (např. čísla v telefonním seznamu apod.). Např. z roku 1927 pocházejí Tippetovy tabulky. Pro počítačové experimenty většího rozsahu jsou ovšem nevhodné.

Softwarový způsob: na základě nějakého algoritmu vytváří počítač posloupnost čísel, která má zdánlivě vlastnosti náhodné posloupnosti, ale je pouze pseudonáhodná.

Ukázka prvních 30 čtveřic náhodných čísel z Tippetovy tabulky:

2952	6641	3992	9792	7979	5911
3170	5624	4167	9525	1545	1396
7203	5356	1300	2693	2370	7483
3408	2769	3563	6107	6913	7691
0560	5246	1112	9025	6008	8126

Generování pseudonáhodných čísel

Většina algoritmů pro generování pseudonáhodných čísel má rekurentní tvar $x_{n+1} = F(x_n, x_{n-1}, \ldots, x_0)$, kde x_0 je vhodná počáteční hodnota zvaná násada nebo semínko (seed) a F je příslušná funkce. Tyto algoritmy umožňují vytváření jen periodické posloupnosti.

Nejrozšířenějšími zdroji pro generování pseudonáhodných čísel jsou tzv. kongruenční generátory.

Příklady kongruenčních generátorů:

- aditivní generátor (Fibonacciův): $x_{n+1} = x_n + x_{n-1} \pmod M$
- multiplikativní generátor (Lehmerův): $x_{n+1} = ax_n \pmod M$
- smíšený generátor: $x_{n+1} = ax_n + b \pmod M$

Požadavky kladné na pseudonáhodná čísla používaná v simulacích:

- dostatečně dlouhé posloupnosti bez opakování (řádově aspoň 10^9 hodnot)
- rychlý výpočetní algoritmus generování pseudonáhodných čísel
- splnění důležitých vlastností (musí se řídit daným typem rozložení, musí projít testy náhodnosti a nezávislosti).

Náhodná čísla a náhodné číslice

V užším slova smyslu jsou za náhodná čísla považovány realizace nezávislých náhodných veličin z Rs(0,1), tj. hustota $\varphi(x) = \begin{cases} 1 \text{ pro } x \in (0,1) \\ 0 \text{ jinak} \end{cases}$ a za náhodné číslice (dekadické) realizace nezávislých náhodných veličin z Rd(G), kde G = {0, 1, …, 9}, tj. pravděpodobnostní funkce

$$\pi(x) = \begin{cases} \frac{1}{10} \text{ pro } x \in G \\ 0 \text{ jinak} \end{cases}.$$

Upozornění: Nezáleží na tom, zda máme k dispozici náhodná čísla nebo náhodné číslice, protože mezi nimi platí tyto vztahy:

a) Nechť $X_1, \ldots, X_n$ jsou nezávislé náhodné veličiny, $X_i \sim Rd(G)$, $G = \{0, 1, \ldots, 9\}$. Pak transformovaná náhodná veličina $Y = \sum_{i=1}^{\infty} 10^{-i} X_i \sim Rs(0,1)$.

Znamená to, že když máme k dispozici náhodné číslice s rovnoměrným diskrétním rozložením, tak jejich transformaci Y můžeme považovat za spojitou náhodnou veličinu s rovnoměrným spojitým rozložením na intervalu (0,1).

b) Nechť náhodná veličina $Y \sim Rs(0,1)$. Potom posloupnost čísel $X_1, X_2, \ldots$ desetinného rozvoje $Y = \sum_{i=1}^{\infty} 10^{-i} X_i$ tvoří posloupnost nezávislých a stejně rozložených náhodných veličin s rozložením $Rd(G)$, $G = \{0, 1, \ldots, 9\}$.

Znamená to, že rozložením náhodného čísla z intervalu (0,1) na jednotlivé číslice podle desetinného rozvoje lze tyto číslice považovat za realizace nezávislých náhodných veličin z rovnoměrného diskrétního rozložení na množině $\{0, 1, \ldots, 9\}$.

Generování náhodných čísel z jiných rozložení

Máme generovat posloupnost n nezávislých realizací náhodné veličiny s distribuční funkcí $\Phi(x)$.

Při metodě inverzní transformace se generují čísla z Rs(0,1) a vhodně se transformují.

a) Diskrétní náhodná veličina

Postup:

1. krok: Z rozložení Rs(0,1) vygenerujeme číslo u.
2. krok: Položíme $x = a_i$, kde $i = \min\{i; \Phi(a_i) \geq u\}$.

Kroky 1 a 2 se opakují n-krát.

Příklad: Opakovaně vypisovaných výběrových řízení se účastní vždy 4 firmy, označme je A, B, C, D. Pravděpodobnost, že jejich nabídky budou vybrány, jsou postupně 0,2; 0,3; 0,4 a 0,1. Pro n = 10 simulujte výsledky opakovaných výběrových řízení.

Řešení: Zavedeme náhodnou veličinu X, která nabývá hodnoty 1, když vyhraje firma A, hodnoty 2, když vyhraje B, hodnoty 3, když vyhraje C a hodnoty 4, když vyhraje D.

Najdeme distribuční funkci náhodné veličiny X.

$$\Phi(x) = \begin{cases} 0 \text{ pro } x \in (-\infty, 1) \\ 0{,}2 \text{ pro } x \in \langle 1, 2) \\ 0{,}5 \text{ pro } x \in \langle 2, 3) \\ 0{,}9 \text{ pro } x \in \langle 3, 4) \\ 1 \text{ pro } x \in \langle 4, \infty) \end{cases}$$

Vygenerujeme číslo $u \in (0,1)$ a transformujeme ho takto:

$$x = \begin{cases} 1 \text{ pro } 0 < u \leq 0{,}2 \\ 2 \text{ pro } 0{,}2 < u \leq 0{,}5 \\ 3 \text{ pro } 0{,}5 < u \leq 0{,}9 \\ 4 \text{ pro } 0{,}9 < u < 1 \end{cases}$$

Znamená to, že hodnotám veličiny X je distribuční funkcí $\Phi(x)$ přiřazena ta část intervalu (0,1), která je proporcionální pravděpodobnosti odpovídající hodnoty X.

Je-li např. u = 0,7654321, pak nejmenší index i, pro nějž $\Phi(a_i) \geq 0{,}7654321$ je 3, tedy v tomto případě vyhraje firma C.

Simulaci provedeme v MATLABu pomocí funkce simulace_DNV(n,v):

```
function [realizace]=simulace_DNV(n,v)
% autor: Petra Cabalková
% funkce generuje n realizaci diskretni nahodne veliciny
% pravdepodobnosti realizace zadava vektor rozlozeni pravdepodobnosti v
% syntaxe: [realizace]=simulace_DNV(n,v)
% vstupni parametr: n ... pocet kroku simulace
%                   v ... vektor rozlozeni pravdepodobnosti diskretni nah.veliciny
% vystupni parametr:
% realizace ... vektor realizaci diskretni nahodne veliciny
realizace=[];
m=length(v);
for i=1:n
  u=unifrnd(0,1,1,1);
  if u<v(1,1) x=1;end;
  if u>sum(v(1,1:m-1)) x=m;end;
  for j=2:(m-1)
    if (u>sum(v(1,1:j-1))) & (u<=sum(v(1,1:j))) x=j;end;
  end
realizace=[realizace x];
end
    plot(realizace,'o')
```

Funkci zavoláme s parametry n = 10, v = [0.2 0.3 0.4 0.1]. Dostaneme výstup:

realizace =

2 3 3 1 1 2 4 2 3 2

Vidíme, že při 1. konkurzu vyhraje firma B, při 2. a 3. firma C atd.

Simulaci můžeme doplnit tabulkou rozložení četností:

tabulate(realizace)

Value	Count	Percent
1	2	20.00%
2	4	40.00%
3	3	30.00%
4	1	10.00%

b) Spojitá náhodná veličina

Metoda inverzní transformace je založena na dvou následujících větách.

Věta 1.: Nechť spojitá náhodná veličina X má rostoucí distribuční funkci $\Phi(x)$ (tzn., že k ní existuje inverzní funkce $\Phi^{-1}(y)$ – tzv. kvantilová funkce). Pak transformovaná náhodná veličina $Y = \Phi(X)$ má rozložení Rs(0,1).

Důkaz: Označme $\Phi_*(y)$ distribuční funkci náhodné veličiny Y.

$$\Phi_*(y) = P(Y \le y) = P(\Phi(X) \le y) = P\left(X \le \Phi^{-1}(y)\right) = \Phi\left(\Phi^{-1}(y)\right) = y \text{ pro } y \in (0,1),$$

$$\Phi_*(y) = 0 \text{ pro } y \in \langle -\infty, 0\rangle, \quad \Phi_*(y) = 1 \text{ pro } y \in \langle 1, \infty)$$

Tedy Y ~ Rs(0,1).

Věta 2.: Nechť X ~ Rs(0,1) a nechť Φ je rostoucí spojitá distribuční funkce. Pak transformovaná náhodná veličina $Y = \Phi^{-1}(X)$ má distribuční funkci Φ.

Důkaz: Označme $\Phi_*(y)$ distribuční funkci náhodné veličiny Y.

$\Phi_*(y) = P(Y \le y) = P\left(\Phi^{-1}(X) \le y\right) = P(X \le \Phi(y)) = \Phi(y)$, protože X ~ Rs(0,1).

Postup:

1. krok: Z rozložení Rs(0,1) vygenerujeme číslo u.
2. krok: Číslo u transformujeme vztahem $x = \Phi^{-1}(u)$.

Kroky 1 a 2 se opakují n-krát.

Příklad: Generování z exponenciálního rozložení

K benzínové pumpě s jedním čerpadlem přijíždí v průměru auto každé dvě minuty.

Předpokládáme, že doba mezi příjezdy aut se řídí exponenciálním rozložením. Simulujte příjezd 10 aut k benzínové pumpě.

Řešení: Označme X dobu, která uplyne mezi dvěma po sobě následujícími příjezdy aut. Víme, že se řídí exponenciálním rozložením s parametrem $\lambda = 0{,}5$.

$\varphi(x) = \begin{cases} \lambda e^{-\lambda x} \text{ pro } x > 0 \\ 0 \text{ jinak} \end{cases}$, $\Phi(x) = \begin{cases} 1 - e^{-\lambda x} \text{ pro } x > 0 \\ 0 \text{ jinak} \end{cases}$. Odtud odvodíme:

$$u = \Phi(x) = 1 - e^{-\lambda x} \Rightarrow x = -\frac{1}{\lambda}\ln(1-u).$$

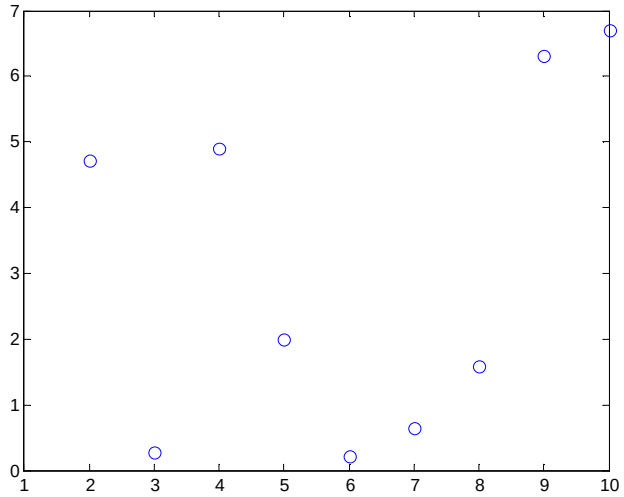
V našem případě $\lambda = 0{,}5$, tedy $x = -2\ln(1-u)$.

Samotnou simulaci provedeme v MATLABu pomocí funkce sim_expon(n,lambda):

```
function x=sim_expon(n,lambda)
% funkce generuje n realizaci spojite nahodne veliciny s exp. rozlozenim
% s exponencialnim rozlozenim
% syntaxe: x=sim_expon(n,lambda)
% vstupni parametry:
% n ... pocet realizaci
% lambda ... parametr exponencialniho rozlozeni
% vystupni parametr:
% x ... vektor realizaci nahodne veliciny s exp. rozlozenim
u=unifrnd(0,1,n,1);
x=-(1/lambda)*log(1-u);
plot(x,'o')
pocet=[1:n]';
t=cumsum(x);
figure
stairs(t,pocet)
```

x =

3.3718 4.7245 0.2716 4.8924 2.0013 0.2053 0.6528 1.5832 6.3168 6.6985



Znamená to, že při této simulaci první auto přijede 3,3718 min po začátku sledování, druhé auto přijede 4,7245 min po prvním autu atd.

Příklad: Generování z normálního rozložení

Náhodná čísla z normálního rozložení nemůžeme generovat metodou inverzní transformace, protože nelze analyticky vyjádřit inverzní funkci k distribuční funkci $\Phi(x) = \int_{-\infty}^{x} \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(t-\mu)^2}{2\sigma^2}} dt$.

a) Metoda založená na centrální limitní větě

Nechť $\{X_n\}_{n=1}^{\infty}$ je posloupnost náhodných veličin definovaných na témž pravděpodobnostním prostoru, která splňuje tyto podmínky:

- náhodné veličiny $X_1, X_2, \ldots$ jsou stochasticky nezávislé
- náhodné veličiny $X_1, X_2, \ldots$ mají všechny stejné rozložení se střední hodnotou μ a rozptylem σ^2.

Pak posloupnost standardizovaných součtů

$$U_n = \frac{\sum_{i=1}^{n} X_i - n\mu}{\sigma\sqrt{n}}, \; n = 1, 2, \ldots$$

konverguje v distribuci k náhodné veličině $U \sim N(0,1)$, tj. platí:

$\forall x \in R : \lim_{n\to\infty} P(U_n \le x) = \Phi(x)$, kde $\Phi(x)$ je distribuční funkce rozložení N(0,1).

Zvolíme náhodné veličiny s Rs(0,1):

Nechť $X_1, \ldots, X_n$ jsou stochasticky nezávislé náhodné veličiny, $X_i \sim Rs(0,1)$, tedy $E(X_i) = \frac{1}{2}, D(X_i) = \frac{1}{12}, i = 1,\ldots,n$. Pak $E\left(\sum_{i=1}^{n} X_i\right) = \sum_{i=1}^{n} E(X_i) = \frac{n}{2}$,

$D\left(\sum_{i=1}^{n} X_i\right) = \sum_{i=1}^{n} D(X_i) = \frac{n}{12}$.

Podle CLV náhodná veličina $U_n = \frac{\sum_{i=1}^{n} X_i - n\mu}{\sigma\sqrt{n}} = \frac{\sum_{i=1}^{n} X_i - \frac{n}{2}}{\sqrt{\frac{n}{12}}} \approx N(0,1)$.

Stačí volit n = 12 a pak $U_{12} = \sum_{i=1}^{12} X_i - 6 \approx N(0,1)$.

Generování provedeme v MATLABu pomocí funkce clv(mi,sigma,n):

```
function [realizace]=clv(mi,sigma,n)
% funkce generuje cisla z normalniho rozlozeni pomoci CLV
% syntaxe: realizace=clv(mi,sigma, n)
% vystupni parametr:
% realizace ... vektor realizaci nahodne veliciny s rozlozenim N(mi, sigma^2)
% vstupni parametry:
% mi ... stredni hodnota
% sigma ... smerodatna odchylka
% n ... pocet realizaci
realizace=[];
for i=1:n
u=unifrnd(0,1,12,1);
x=sum(u)-6;
realizace=[realizace;x];
end
sh=mi*ones(n,1);
realizace=sh+sigma*realizace;
hist(realizace)
```

Nevýhody:

- na jedno náhodné číslo z $N(\mu, \sigma^2)$ je zapotřebí vygenerovat 12 čísel z Rs(0,1);
- rozložení takto vygenerovaných náhodných čísel je oboustranně useknuté v bodech -6, 6;
- Aproximace není uspokojivá vně intervalu $(\mu - 3\sigma, \mu + 3\sigma)$.

Tento nedostatek lze částečně napravit tak, že vygenerovaná náhodná čísla budeme transformovat pomocí polynomu.

b) Aproximace polynomem

Realizace generované v bodě (a) upravíme takto:

$y=0,25x$

$u=a_1y + a_3y^3 + a_5y^5 + a_7y^7 + a_9y^9$,

kde koeficienty jsou:

$a_1 = 3,949846138$

$a_3 = 0,252408784$

$a_5 = 0,076542912$

$a_7 = 0,008355968$

$a_9 = 0,029899776.$

Hodnoty u lze považovat za realizace náhodné veličiny s rozložením N(0,1).

Funkci clv.m upravíme tak, aby poskytovala uspokojivější realizace normálního rozložení:

```
function [realizace]=clv_polynom(mi,sigma,n)
% funkce generuje cisla z normalniho rozlozeni pomoci CLV
% pouziva aproximaci polynomem
% syntaxe: realizace=clv_polynom(mi,sigma, n)
% vystupni parametr:
% realizace ... vektor realizaci nahodne veliciny s rozlozenim N(mi, sigma^2)
% vstupni parametry:
% mi ... stredni hodnota
% sigma ... smerodatna odchylka
% n ... pocet realizaci
a1 = 3.949846138;
a3 = 0.252408784;
a5 = 0.076542912;
a7 = 0.008355968;
a9 = 0.029899776;
realizace=[];
for i=1:n
u=unifrnd(0,1,12,1);
x=sum(u)-6;
y=0.25*x;
u=a1*y+a3*y^3+a5*y^5+a7*y^7+a9*y^9;
realizace=[realizace;u];
end
sh=mi*ones(n,1);
realizace=sh+sigma*realizace;
hist(realizace)
```

c) Metoda založená na Boxově – Müllerově transformaci

Nechť x_1, x_2 jsou dvě nezávislá čísla z Rs(0,1). Pak transformovaná čísla $z_1 = \sqrt{-2\ln x_1}\cos 2\pi x_2, z_2 = \sqrt{-2\ln x_1}\sin 2\pi x_2$ lze považovat za realizace náhodné veličiny s rozložením N(0,1).

Funkce BM_transformace(mi,sigma,n) generuje náhodná čísla z normálního rozložení právě tímto způsobem.

```
function [realizace]=BM_transformace(mi,sigma,n)
% funkce generuje cisla z normalniho rozlozeni pomoci BM transformace
% syntaxe: realizace=BM_transformace(mi,sigma, n)
% vystupni parametr:
% realizace ... vektor realizaci nahodne veliciny s rozlozenim N(mi, sigma^2)
% vstupni parametry:
% mi ... stredni hodnota
% sigma ... smerodatna odchylka
% n ... pocet realizaci (musi byt sude cislo)
realizace=[];
for i=1:n/2
u1=unifrnd(0,1,1,1);
u2=unifrnd(0,1,1,1);
x1=sqrt(-2*log(u1))*cos(2*pi*u2)
x2=sqrt(-2*log(u1))*sin(2*pi*u2)
realizace=[realizace;x1;x2];
end
sh=mi*ones(n,1);
realizace=sh+sigma*realizace;
hist(realizace)
```