NI-VSM – 4. domácí úkol

Eliška Krátká (kratkeli), Ondřej Wrzecionko (wrzecond), Eliáš El Frem (elfreeli)

Obsah

1 Úvod

Tato práce se zabývá řešením 4. domácího úkolu z NI-VSM na téma Poissonův proces a systémy hromadné obsluhy. Cílem práce je simulovat trajektorie procesu hromadné obsluhy dle modelu $M|G|\infty$ a porovnat odhadnuté rozdělení s rozdělením systému $M|G|\infty$. Řešení je implementováno v programovacím jazyce Python. Reprezentantem skupiny je **Eliáš El Frem**.

1.1 Popis problému

Uvažujte model hromadné obsluhy $M|G|\infty$.

- Požadavky přichází podle Poissonova procesu s intenzitou $\lambda = 10\ \mathrm{s}^{-1}$
- Doba obsluhy jednoho požadavku (v sekundách) má rozdělení $S \sim \mathrm{Ga}(4, 2)$, tj. Gamma s parametry $a = 4,\ p = 2$.
- Časy mezi příchody a časy obsluhy jsou nezávislé.
- Systém má (teoreticky) nekonečně paralelních obslužných míst (každý příchozí je rovnou obsluhován).

Označme N_t počet zákazníků v systému v čase t. Předpokládejme, že na začátku je systém prázdný, tj. $N_0 = 0$.

1.2 Zadání úkolu

1. **(2b)** Simulujte jednu trajektorii $\{N_t(\omega) \mid t \in (0, 10\ \mathrm{s})\}$. Průběh trajektorie graficky znázorněte.
2. **(2b)** Simulujte $n = 500$ nezávislých trajektorií pro $t \in (0, 100)$. Na základě těchto simulací odhadněte rozdělení náhodné veličiny N_{100}.
3. **(2b)** Diskutujte, jaké je limitní rozdělení tohoto systému pro $t \to +\infty$ (vizte přednášku 23). Pomocí vhodného testu otestujte na hladině významnosti 5 %, zda výsledky simulace N_{100} odpovídají tomuto rozdělení.

Upozornění: *Při simulování z Gamma rozdělení si řádně v dokumentaci prostudujte, jaké parametry vámi zvolený nástroj používá. Často je používaná dvojice:* shape parameter $k = p$ a scale parameter $\theta = 1/a$ [1].

2 Postup řešení

V této části popisujeme postup řešení domácího úkolu společně s důležitými částmi zdrojového kódu, které jsme implementovali v programovacím jazyce Python.

Procesem hromadné obsluhy $X = \{X_t \mid t \geq 0\}$ rozumíme proces, který zaznamenává počet zákazníků v systému hromadné obsluhy (tj. na serveru a ve frontě) v čase t [2]. Dle zadání úkolu uvažujeme model hromadné obsluhy $M|G|\infty$. Příchody se řídí Poissonovým processem s intenzitou $\lambda = 10$ s^{-1}. Doba obsluhy jednoho požadavku (v sekundách) má rozdělení $S \sim \mathrm{Ga}(4, 2)$, tj. Gamma s parametry $a = 4$, $p = 2$. Zákazník je obsluhován okamžitě po příchodu, nikdo nečeká ve frontě.

Pro rozdělení Gamma jsme použili funkci z knihovny *numpy*. Tato funkce Gamma rozdělení pracuje s parametrem škálování θ. Parametr škálování jsme nastavili na hodnotu $\theta = 1/a$, abychom byli konzistentní s definicí Gamma rozdělení v NI–VSM (viz přednáška 2).

2.1 Simulace jedné trajektorie

Simulujeme jednu trajektorii procesu hromadné obsluhy N_t zákazníků v časovém rozmezí deseti sekund, tedy $\{N_t(\omega) \mid t \in (0, 10\,\mathrm{s})\}$. Předpokládáme, že na začátku je systém prázdný, tj. $N_0 = 0$.

Při simulování trajektorie jsme pracovali s náhodně generovanou hodnotou *numpy.random*. Jelikož jsme během řešení spouštěli program opakovaně, zafixovali jsme si *numpy.random* na pevně danou hodnotu.

Nejprve jsme museli zjistit, kolik požadavků přijde na náš systém v prvních deseti sekundách. Jelikož se model řídí Poissonovým rozdělením, vybírali jsme n náhodných vzorků z Poissonova rozdělení.

Dále jsme simulovali časy příchodů. Využili jsme věty 19.6 z přednášky 23 a časy příchodů jsme vygenerovali pomocí rovnoměrného rozdělení Unif(0, t). Časy obsluhy jsme vygenerovali pomocí již zmíněného Gamma rozdělení Ga(4, 2). Koncové časy jsme vypočítali přičtením časů počátečních příchodů k dobám obsluhy a omezili jsme se pouze na ty požadavky, které byly obslouženy do deseti sekund.

Abychom mohli graficky znázornit průběh simulované trajektorie, museli jsme zjistit, kolik je v každé sekundě aktuálně zpracovávaných požadavků.

```python
#!/usr/bin/env python3

def getTrajectory(time):
    #zadefinovani parametru
    a = 4
    p = 2
    lamb = 10

    #odhad poctu pozadavku - vyber n nahodnych vzorku z Poissonova rozdeleni
    cnt = np.random.poisson(lamb*time)

    #simulovani casu prichodu
    starting = np.random.uniform(0,time,cnt)

    #casy obsluhy
    ending = starting+np.random.gamma(p,1/a,cnt)

    #omezeni pouze na pozadavky do deseti sekund
    ending = ending[np.where(ending < time)]

    #vypocet aktualne zpracovavanych pozadavku
    eventlist = []

    for i in starting:
        eventlist.append((i,'S'))
    for i in ending:
        eventlist.append((i,'E'))

    eventlist.sort()

    trajectory = {}
    currcount = 0

    for i in eventlist:
```

```python
        if i[1] == 'S':
            currcount += 1
        else:
            currcount -= 1
        trajectory[i[0]] = currcount
    return trajectory
```

2.2 Simulace 500 trajektorií, odhad rozdělení

Simulujeme $n = 500$ nezávislých trajektorií pro $t \in (0, 100)$. K simulaci jsme využili funkci *getTrajectory* z předchozího úkolu. Získali jsme pole 500 hodnot, které reprezentují počet požadavků v čase N_{100}. Hodnoty jsem graficky znázornili pomocí histogramu. Pomocí výběrového průměru a výběrového rozptylu jsme určili parametry a odhadli rozdělení náhodné veličiny N_{100}.

```python
#!/usr/bin/env python3

#vypocet poctu pozadavku v case N_100
customerCounts = []
for i in range(500):
    customerCounts.append(list(getTrajectory(100).items()).pop()[1])

#vykresleni histogramu
plt.hist(customerCounts)

#vypocet vyberoveho prumeru a rozptylu
countArr = np.array(customerCounts)
print("Mean:",countArr.mean())
print("Var:",countArr.var(ddof=1))
```

2.3 Testování shody rozdělení

Dle výsledků druhého úkolu jsme došli k závěru, že rozdělení náhodné veličiny N_{100} připomíná svým tvarem Poissonovo rozdělení. Tento odhad podporuje i teorie z přednášky 23, slide 21. Z dlouhodobého hlediska $t \to +\infty$ má počet zákazníků v systému Poissonovo rozdělení s intenzitou λ/μ [3]. V systému hromadné obsluhy $M|G|\infty$ je G obecné rozdělení se střední hodnotu $1/\mu$. Dle zadání úkolu pracujeme s Gamma rozdělením, jehož střední hodnota je rovna p/a [4]. Z tohoto vztahu dostáváme hledanou intenzitu Poissonova rozdělení $\lambda = 5$.

$$p/a = 1/\mu$$
$$\mu = a/p$$

$$\lambda/\mu = \lambda/(a/p) = 10/(4/2) = 5$$

Chceme testovat shodnost rozdělení náhodné veličiny N_{100} s Poissonovým rozdělením s intezitou $\lambda = 5$. Použijeme metodiku testování hypotéz. Jako nulovou hypotézu H_0 volíme tvrzení, že odhadnuté rozdělení odpovídá Poissonovu rozdělení Poisson(5), proti alternativě H_A, že tomu tak není. Pokud bychom tedy na základě testu na hladině α zamítli H_0 ve prospěch H_A, znamená to, že si jsme na $1 - \alpha$ procent jisti, že se rozdělení nerovnají.

Vygenerovali jsme 500 hodnot Poissonova rozdělení s intenzitou 5. U odhadnutého rozdělení náhodné veličiny N_{100} a u vygenerovaného Poissonova rozdělení jsme spočetli četnosti jednotlivých hodnot. Zkontrolovali jsme, že teoretické četnosti Poissonova rozdělení splňují Pearsonovu statistiku. Některé četnosti byly menší než pět, ale jejich poměr byl zanedbatelný, proto jsme je na základě Yarnoldova kritéria neslučovali. Použili jsme χ^2 test dobré shody při známých parametrech s počtem stupňů volnosti $k-1$, kde k je počet binů.

Zda H_0 zamítneme nebo nezamítneme, jsme vyhodnotili na základě kritického oboru

$$\chi^2 \geq \chi^2_{\alpha,k-1},$$

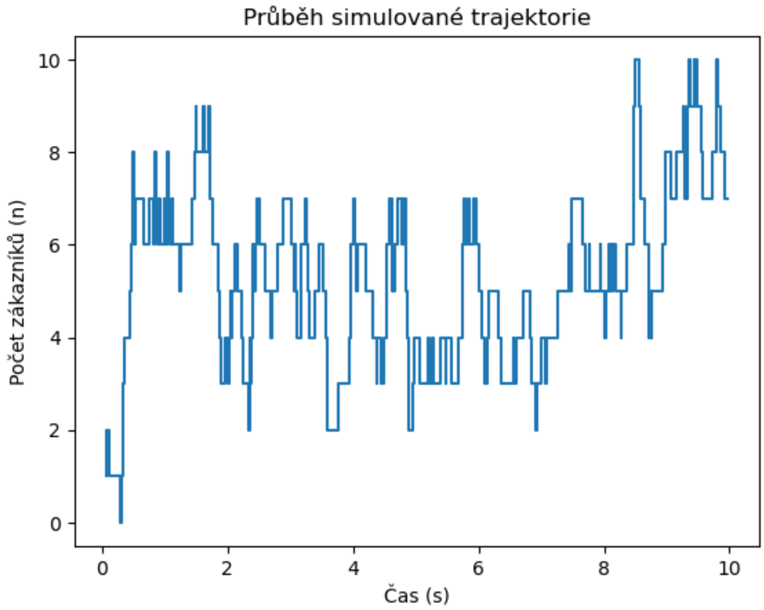
kde χ^2 je Pearsonova statistika s $k-1$ stupni volnosti [5]

```
#!/usr/bin/env python3

#vygenerovani 500 Poissonova rozdeleni s intenzitou 5
poisson_items = np.random.poisson(5, 500)

#vypocet cetnosti
poisson_freqs = np.unique(poisson_items,return_counts=True)[1]
poisson_freqs = poisson_freqs/poisson_freqs.sum()
countArr_freqs = np.unique(countArr, return_counts=True)[1]
countArr_freqs =  countArr_freqs/countArr_freqs.sum()

#vypocet chi^2
allCusts = countArr.sum()
print(scipy.stats.chisquare(countArr_freqs*allCusts, poisson_freqs*allCusts))
bins =  len(countArr_freqs)
print(f'Bins: {bins}\nchi2: {scipy.stats.chi2.ppf(1-0.05,bins-1)}')
```

3 Výsledky

V tabulkách uvádíme výsledky zaokrouhleny na tři desetinná místa.

3.1 Simulace jedné trajektorie

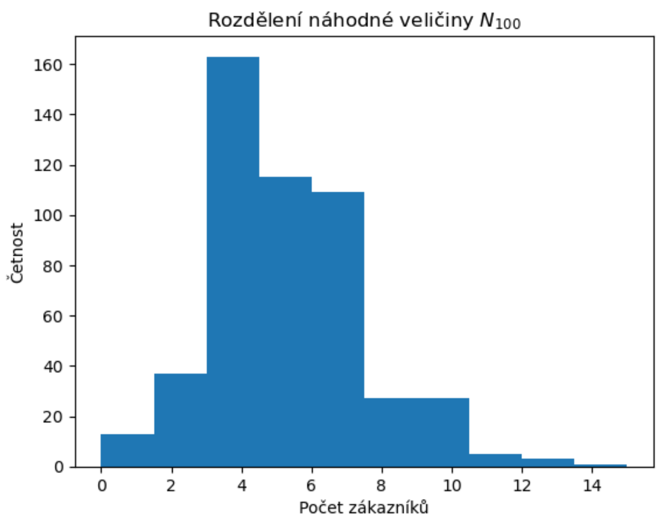


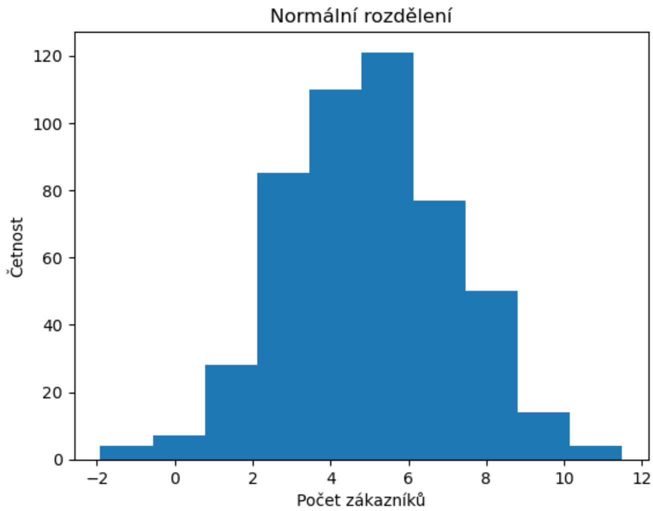
3.2 Simulace 500 trajektorií, odhad rozdělení

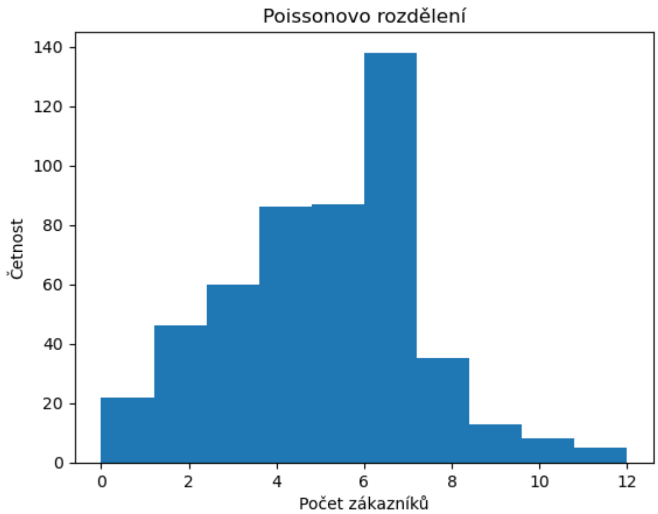
Pro odhadnuté rozdělení náhodné veličiny N_{100} jsme vypočetli výběrový průměr a rozptyl.

Výběrový průměr	**Výběrový rozptyl**
5.012	4.721

Rozdělení náhodné veličiny N_{100} jsem porovnali s Normálnímu rozdělením N(5.012, 4.721) a Poissonovým rozdělením Poisson(5.012). Pozorujeme, že se svým tvarem podobá spíše Poissonovu rozdělení.







3.3 Testování shody rozdělení

Protože hodnota $\chi^2 \geq \chi^2_{0.05,14}$, tedy $165.498 \geq 23.685$, zamítáme nulovou hypotézu H_0 ve prospěch alternativní hypotézy H_A. P-hodnota je velmi nízká, což nám umožňuje s téměř jistotou tvrdit, že rozdělení náhodné veličiny N_{100} není rovno Poissonovu rozdělení s intenzitou $\lambda = 5$.

$\boldsymbol{\alpha}$	$\boldsymbol{\chi^2}$	**počet stupňů volnosti**	**p-hodnota**	$\boldsymbol{\chi^2_{0.05,14}}$
0.05	165.498	14	5.5464e-28	23.685

4 Závěr

Simulovali jsme trajektorii procesu hromadné obsluhy dle modelu $M|G|\infty$. Odhadli jsme rozdělení náhodné veličiny N_{100} na základě simulací 500 nezávislých trajektorií. Následně jsme provedli test dobré shody na hladině významnosti 5 % a porovnali odhadnuté rozdělení s Poissonovým rozdělením. Došli jsme k závěru, že odhadnuté rozdělení se Poissonovu rozdělení nerovná.

Reference

[1] P. Hrabák. Domácí úkol 4. https://courses.fit.cvut.cz/MI-SPI/homework/hw4/index.html.

[2] P. Hrabák, P. Novák, D.Vašata. Systémy hromadné obsluhy. https://courses.fit.cvut.cz/MI-SPI/lectures/files/NI-VSM-Lec-22-Slides.pdf.

[3] P. Hrabák, P. Novák, D.Vašata. Nehomogenní poissonův proces. https://courses.fit.cvut.cz/MI-SPI/lectures/files/NI-VSM-Lec-23-Slides.pdf.

[4] P. Hrabák, P. Novák, D.Vašata. Náhodné veličiny. https://courses.fit.cvut.cz/MI-SPI/lectures/files/NI-VSM-Lec-02-Slides.pdf.

[5] P. Hrabák, P. Novák, D.Vašata. Markovské řetězce se spojitým časem. https://courses.fit.cvut.cz/MI-SPI/lectures/files/NI-VSM-Lec-18-Handout.pdf.