NI-VSM – 3. domácí úkol

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

Obsah

1 Úvod

Tato práce se zabývá naším řešením 3. domácího úkolu z NI-VSM na téma markovské řetězce s diskrétním časem. Používáme stejné dva texty jako v předchozích úkolech, konkrétně datové soubory *005.txt* (první text) a *004.txt* (druhý text). Cílem práce je odhadnout matici přechodu markovského řetězce pro první text a na základě této matice nalézt stacionární rozdělení π, dále na hladině významnosti 5 % otestovat hypotézu, že rozdělení znaků druhého textu se rovná rozdělení nalezenému stacionárnímu rozdělení π. Implementace řešení byla provedena v jazyce Python.

1.1 Zadání úkolu

Z obou datových souborů načtěte texty k analýze. Pro každý text zvlášť zjistěte absolutní četnosti jednotlivých znaků (symbolů včetně mezery), které se v textech vyskytují. Dále předpokládejme, že **první** text je vygenerován z homogenního markovského řetězce s diskrétním časem.

1. **(2b)** Za předpokladu výše odhadněte matici přechodu markovského řetězce pro **první text**. Pro odhad matice přechodu vizte **přednášku 17**. Odhadnuté pravděpodobnosti přechodu vhodně graficky znázorněte, např. použitím heatmapy.
2. **(2b)** Na základě matice z předchozího bodu najděte stacionární rozdělení $\boldsymbol{\pi}$ tohoto řetězce pro **první** text.
3. **(2b)** Porovnejte rozdělení znaků **druhého** textu se stacionárním rozdělením $\boldsymbol{\pi}$, tj. na hladině významnosti 5 % otestujte hypotézu, že rozdělení znaků **druhého** testu se rovná rozdělení $\boldsymbol{\pi}$ z předchozího bodu. [1]?

1.2 Parametry úlohy

Reprezentantem skupiny je **Eliáš El Frem**. Parametry jsme vypočítali dle vzorce ze zadání úkolu [1]:

$$X = ((K \cdot L \cdot 23) \bmod 20) + 1,$$
$$Y = ((X + ((K \cdot 5 + L \cdot 7) \bmod 19)) \bmod 20) + 1,$$

kde K je den narození reprezentanta skupiny a L počet písmen v příjmení reprezentanta. Pro úlohu jsme na základě výpočtu parametrů použili datové soubory *005.txt* (první text) a *004.txt* (druhý text).

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

K = 17
L = len("Frem")
fname1 = ((K*L*23) % (20)) + 1
fname2 = ((fname1 + ((K*5 + L*7) % (19))) % (20)) + 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.

2.1 Matice přechodu markovského řetězce

Z datových souborů jsme načetli oba texty a zvlášť pro každý soubor a znak spočítali četnost. Mezi znaky jsme zahrnuli i mezeru a vynechali jsme čárku a nový řádek. Pravděpodobnost výskytu znaku $P(X)$ odpovídá

$$P(X) = \frac{\text{četnost znaku}}{\text{celkový počet znaků}},$$

kde X je náhodná veličina – počet výskytů daného znaku v textu.

Náhodný proces $\{X_n \mid n \in \mathbb{N}_0\}$ s nejvýše spočetnou množinou stavů S je markovský právě tehdy když $\forall k \in \mathbb{N}, \forall n_0 < n_1 < ... < n_k \in \mathbb{N}_0$ a $\forall s_0, ..., s_k \in S$ platí

$$\mathrm{P}(X_{n_0} = s_0, ..., X_{n_k} = s_k) = \boldsymbol{p}_{s_0}(n_0) \cdot \mathbf{P}_{s_0 s_1}(n_0, n_1) \cdot \mathbf{P}_{s_1 s_2}(n_1, n_2) \cdot \cdot \cdot \cdot \cdot \mathbf{P}_{s_{k-1} s_k}(n_{k-1}, n_k),$$

kde S je nejvýše spočetná množina stavů.

Pro $i, j \in S$ označme n_{ij} počet přechodů z i do j, tj.

$$n_{ij} := |\{k \in \{0, 1, ..., n-1\} \mid s_k = \wedge s_{k+1} = j\}|.$$

Pro odhad matice přechodu markovského řetězce jsme použili metodu maximální věrohodnosti (MLE). Maximálně věrohodným odhadem matice přechodu $\mathbf{P}$ je matice $\widehat{\mathbf{P}}$ s prvky

$$\widehat{\mathbf{P}}_{ij} = \frac{n_{ij}}{n_{i\bullet}}, \ kde \ n_{i\bullet} := \sum_{j \in S} n_{ij} \ [2].$$

Jelikož se jedna o náhodný proces, musíme množinu znaků zobrazit na čísla. Zvolili jsme mapování $a \to 0, z \to 25, mezera \to 26$. Vznikla nám tedy matice s rozměry 27x27. Postupně jsme prošli pole znaků, a vždy přičetli jedničku podle posledního znaku na indexu i a aktuálního znaku na indexu j. Řádky matice jsme znormalizovali tak, že jsme vydělili hodnoty $\widehat{\mathrm{P}}_{ij}$ příslušným P_i. Matici jsme vizualizovali pomocí heatmapy.

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

#nacteni souboru
file1 = open(f'hw1-source/00{fname1}.txt','r')
file2 = open(f'hw1-source/00{fname2}.txt','r')

#odstraneni prebytecnych znaku
file1_string = file1.read().replace(',','').replace('\n','').lower()
file2_string = file2.read().replace(',','').replace('\n','').lower()

#vypocet cetnosti
def countFreqs(to_cnt):
    char_cnt = {}
    for i in to_cnt.lower():
        if not char_cnt.get(i):
            char_cnt[i] = 1
        else:
            char_cnt[i] += 1
    return char_cnt

f1_freqs = dict(sorted(countFreqs(file1_string).items()))
f2_freqs = dict(sorted(countFreqs(file2_string).items()))

#vypocet matice
matrix = np.zeros((len(f1_freqs.keys()),len(f1_freqs.keys())))

def indices(a, b):
    first =  0 if not a.isalnum() else (ord(a) - ord('a')) + 1
    second = 0 if not b.isalnum() else (ord(b) - ord('a')) + 1
    return first, second
```

```
31
32 for i in range(len(file1_string) - 1):
33     a, b = indices(file1_string[i],file1_string[i + 1])
34     matrix[a,b] += 1
35
36 for i in range(matrix.shape[0]):
37     matrix[i] /= f1_freqs[list(f1_freqs.keys())[i]]
38
39 #vytvoreni heatmapy
40 plt.figure(figsize=(10,8))
41 ax = sns.heatmap(matrix, vmin=0, vmax=matrix.max(), xticklabels=list(f1_freqs.
       keys()), yticklabels=list(f1_freqs.keys()))
```

2.2 Stacionární rozdělení

Na základě matice přechodu jsme hledali stacionární rozdělení π tohoto řetězce pro první text.

Stacionární rozdělení homogenního markovského řetězce $\{X_n \mid n \in \mathbb{N}_0\}$ s maticí přechodu $\mathbf{P}$ je vektor π (pokud existuje) takový, že

(i) $\forall i \in S : \boldsymbol{\pi}_i \geq 0$,

(ii) $\sum_{i \in S} \boldsymbol{\pi}_i = 1$,

pro který platí, že

$$\boldsymbol{\pi} \cdot \mathbf{P} = \boldsymbol{\pi} \text{ [2].}$$

Na základě matice přechodu jsme hledali stacionární rozdělení π pro první text. Poslední rovnost v definici stacionárního rozdělení můžeme upravit následovně

$$\pi \cdot \mathrm{P} = \pi \sim \pi \cdot \mathrm{P} - \pi = 0 \sim \boldsymbol{\pi} \cdot (\mathbf{P} - \mathbf{I}) = \mathbf{0},$$

kde I je identická matice (v našem případě o rozměru 27x27). Snažili jsme se tedy nalézt π takové, aby byla splněna poslední rovnice, což odpovídá hledání jádra matice $(\mathbf{P} - \mathbf{I})^{\mathrm{T}}$. Jádro matice je množina všech vektorů, které se zobrazí se zobrazí na nulový vektor pomocí této matice [3]. K nalezení jádra matice neboli nulového prostoru jsme použili knihovní funkci *linalg.null_space*.

Protože počítáme s floaty (čísla s plovoucí řádovou čárkou), v rámci výpočtu vzniká zaokrouhlovací chyba. Proto na konci provádíme kontrolu, zdali je $\pi \cdot \mathrm{P} - \pi$ menší než *machine epsilon*. Pokud je rozdíl mezi dvěma čísly menší než hodnota *machine epsilon*, považujeme je za téměř stejné [4].

```
1 #!/usr/bin/env python3
2
3 #vypocet pi
4 d = (matrix - np.eye(len(f1_freqs.keys()))).T
5 pi = linalg.null_space(d.astype("float32"),rcond= np.finfo(np.float32).eps*1000)
6 pi = (pi/sum(pi)).T
7
8 #kontrola zaokrouhlovaci chyby
9 not False in pi.dot(matrix) - pi < np.finfo(np.float32).eps
```

2.3 Hypotéza – rozdělní znaků druhého textu se rovná stacionárnímu rozdělení

Porovnávali jsme, zda stacionární rozdělení znaků prvního textu skutečně odpovídá rozdělení znaků druhého textu, věnovali jsme se tedy testování hypotéz.

Označme náhodnou veličinu X stacionární rozdělení znaků v prvním textu a náhodnou veličinu Y rozdělení znaků v druhém textu. Jako nulovou hypotézu H_0 volíme tvrzení, že se rozdělení rovnají, proti alternativě H_A, že se nerovnají. 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.

Jelikož se jedná o dvě diskrétní rozdělní, použili jsme χ^2 test dobré shody se známými parametry. Testy dobré shody obecně používáme, pokud ověřujeme hypotézy o tvaru pravděpodobnostního

rozdělení. V našem případě se jedná o ověření, zdali má zkoumaná veličina X shodné stacionární rozdělení s X.

Spočítali jsme teoretické četnosti. Některé byly menší než pět, ale jejich poměr byl zanedbatelný, proto jsme na základě Yarnoldova kritéria neslučovali.

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]

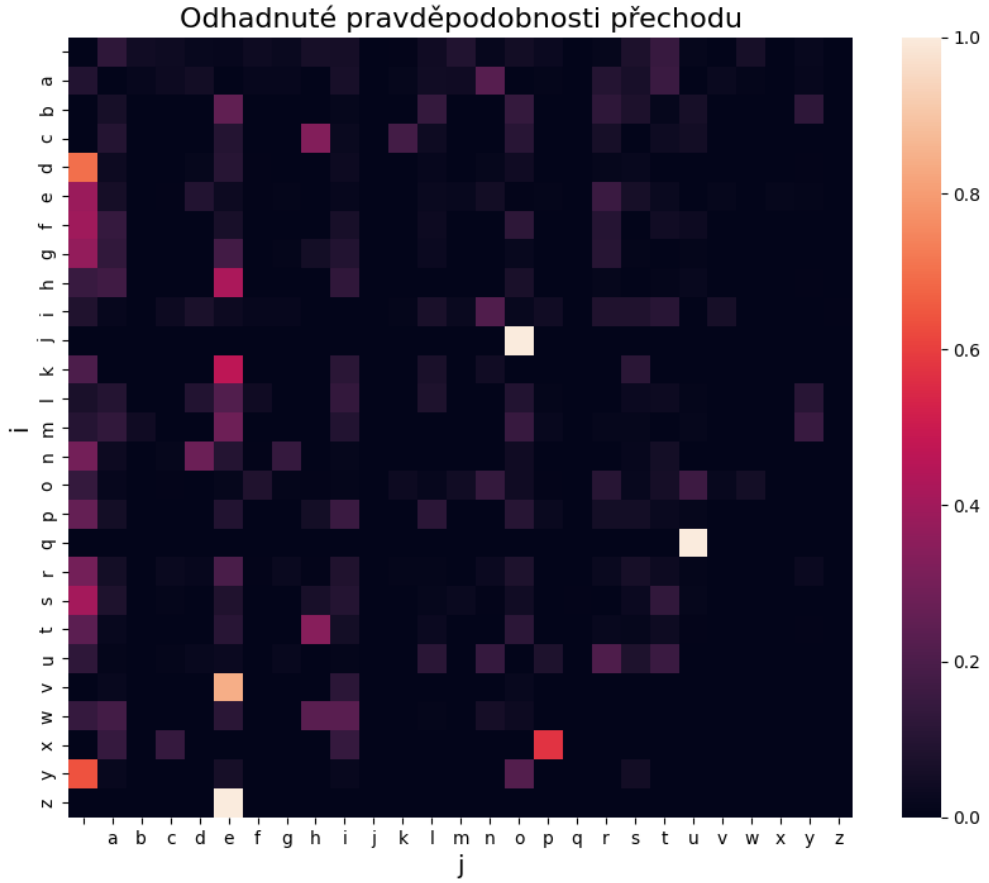
```
#!/usr/bin/env python3

#vypocet chi, kritckeho oboru, p-hodnoty
chisq = 0
n = sum(list(f2_freqs.values()))
for i in range(len(f2_freqs)):
    chisq += ((list(f2_freqs.values())[i] - n*pi[0][i])**2)/(n*pi[0][i])
chi2 = stats.chi2.isf(0.05,26)
p = stats.chi2.sf(chisq, 26)
```

3 Výsledky

V domácím úkolu jsme analyzovali dva anglické texty. Datový soubor *005.txt* obsahuje první text a *004.txt* druhý text. V tabulkách uvádíme výsledky zaokrouhleny na tři desetinná místa.

3.1 Matice přechodu markovského řetězce

Heatmapa matice přechodu je grafické zobrazení hodnot obsažených v matici přechodu. Z heatmapy tedy můžeme pozorovat nejčastější posloupnost znaků v textu. Světlejší čtverečky znamenají vyšší pravděpodobnost přechodu ze znaku i na znak j. Například je vysoká pravděpodobnost přechodu ze znaku q na znak u.



3.2 Stacionární rozdělení

Nalezené stacionární rozdělení π splňuje kontrolu s machine epsilon a vypadá následovně:

$$\pi = \begin{pmatrix} 0.199 \\ 0.070 \\ 0.012 \\ 0.016 \\ 0.040 \\ 0.100 \\ 0.017 \\ 0.018 \\ 0.051 \\ 0.058 \\ 0.001 \\ 0.008 \\ 0.033 \\ 0.028 \\ 0.050 \\ 0.055 \\ 0.014 \\ 0.001 \\ 0.050 \\ 0.046 \\ 0.071 \\ 0.018 \\ 0.008 \\ 0.017 \\ 0.001 \\ 0.019 \\ 0.001 \end{pmatrix}$$

3.3 Hypotéza – rozdělní znaků druhého textu se rovná stacionárnímu rozdělení

Protože hodnota $\chi^2 \geq \chi^2_{0.05,26}$, tedy $183.874 \geq 38.885$, 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 znaků ve druhém textu není rovno stacionárnímu rozdělení prvního textu.

$\boldsymbol{\alpha}$	$\boldsymbol{\chi^2}$	**počet stupňů volnosti**	**p-hodnota**	$\boldsymbol{\chi^2_{0.05,26}}$
0.05	183.874	26	1.032e-25	38.885

4 Závěr

Zanalyzovali jsme dva datové soubory, text **005.txt** a **004.txt**. Nalezli jsme stacionární rozdělení pro první text na základě matice přechodu markovského řetězce. Dle otestování hypotézy jsme došli k závěru, že toto rozdělení se nerovná rozdělení znaků ve druhém textu.

Reference

[1] P. Hrabák. Domácí úkol 3. https://courses.fit.cvut.cz/MI-SPI/homework/hw3/index.html.

[2] P. Hrabák, P. Novák, D.Vašata. Odhad matice přechodu a MCMC. https://courses.fit.cvut.cz/MI-SPI/lectures/files/NI-VSM-Lec-17-Lecture.pdf.

[3] D. Dombek, T. Kalvoda, L. Kleprlík, K. Klouda. Lineární algebra. https://kam.fit.cvut.cz/deploy/bi-lin//lin-text.pdf.

[4] Š. Starosta. Strojová čísla a numerická matematika. https://courses.fit.cvut.cz/MI-MPI/latex/lectures/czech/mi-mpi-prednaska-20-strojova-cisla-handout.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.