دانشکده مهندسی برق

گزارش کد پایتون زنجیره و فرآیند مارکوف

نام نگارنده: ساناز مطیع

استاد راهنما: دکتر سعید عبادالهی

شماره دانشجویی: ۹۹۴۱۳۰۸۲

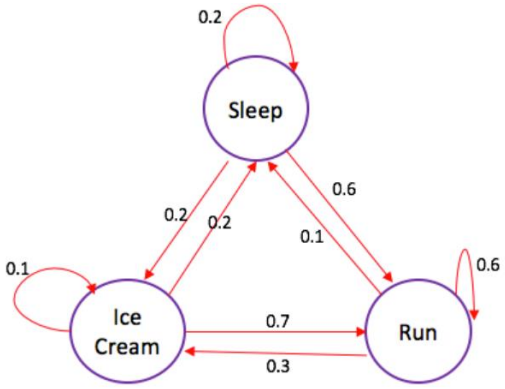
زمستان ۱۴۰۱


بسم الله الرحمن الرحيم

محاسبه احتمال در زنجیره مارکوف

مسئله: به عنوان مثال در نظر می‌گیریم اگر سارا ناراحت باشد در روز آینده بستنی می‌خرد، می‌دود و یا می‌خوابد. حال محاسبه میکنیم چقدر احتمال دارد سارا در روز سوم بدود.

حل: در این کد ابتدا با استفاده از گراف زیر فضای حالت و ماتریس احتمال انتقال را تعریف کردیم.




```
states = ["Sleep","Icecream","Run"]
transitionName = [["SS","SR","SI"],["RS","RR","RI"],["IS","IR","II"]]
transitionMatrix = [[0.2,0.6,0.2],[0.1,0.6,0.3],[0.2,0.7,0.1]]
```

چک می‌کنیم مجموع هر سطر ماتریس احتمال انتقال برابر با یک باشد.

```
if sum(transitionMatrix[0])+sum(transitionMatrix[1])+sum(transitionMatrix[2]) != 3:
    print("Somewhere, something went wrong. Transition matrix, perhaps?")
else: print("All is gonna be okay, you should move on!! ;)")
```

حالت شروع و پایان را مشخص می‌کنیم.

```
starting_state = str(input("please enter starting state\n"))
last_state = str(input("please enter last state\n"))
```

محاسبه می‌کنیم با توجه به حالت شروع و درصد های احتمال بین هر دو حالت activity_forecastدر تابع سارا در روز سوم چه کاری انجام میدهد.

ذخیره می‌کنیم.list_activityتابع را ده هزار بار تکرار می‌کنیم و خروجی تابع را در

```
for iterations in range:(۱،۱۰۰۰۰)
    list_activity.append(activity_forecast(2))
```

سپس تعداد دفعاتی که سارا دویده را محاسبه می‌کنیم و تقسیم بر ده هزار می‌کینم تا درصد آن به‌دست آید.

```
for smaller_list in list_activity:
    if(smaller_list[2] == last_state):
        count += 1
percentage = (count/10000) * 100
```

طبق نتایج به دست آمده مشاهده می‌کنیم احتمال اینکه سارا در روز سوم بدود به حالت شروع آن بستگی ندارد و فقط به ماتریس احتمال انتقال بستگی دارد.

```
All is gonna be okay, you should move on!! ;)
please enter starting state
Sleep
please enter last state
Run
The probability of starting at state:Sleep and ending at state: Run 61.480000000000004%
All is gonna be okay, you should move on!! ;)
please enter starting state
Run
please enter last state
Run
The probability of starting at state:Run and ending at state: Run 62.74999999999999%
All is gonna be okay, you should move on!! ;)
please enter starting state
Icecream
please enter last state
Run
The probability of starting at state:Icecream and ending at state: Run 62.13999999999999%
```

محاسبه توزیع مانا در زنجیره مارکوف

تعریف: توزیع π، توزیع مانا زنجیره مارکوف P نامیده می‌شود اگر $\pi P = \pi$. توزیع مانا زنجیره مارکوف برای تمام زنجیره توزیع مانا است یعنی اگر توزیع Xt توزیع مانا π باشد، آنگاه توزیع مانا $Xt+1$ نیز π است

مسئله: به‌دست آوردن توزیع مانا در یک زنجیره مارکوف دلخواه

حل: ابتدا ماتریس احتمال انتقال از کاربر گرفته می‌شود

```
R = int(input("Enter the number of rows:"))
C = int(input("Enter the number of columns:"))
transition_matrix =[]
```

```
print("Enter the entries in a single line (separated by space): ")
entries = list(map(float, input().split()))
transition_matrix = np.array(entries).reshape(R, C)
```

سپس چک می‌کنیم مجموع هر سطر ماتریس احتمال انتقال برابر با یک شود.

```
sumMatrix = np.sum(transition_matrix,axis=1, dtype='float')
for i in range (0,C-1):
   if sumMatrix[i] =! [١]:
      print("Somewhere, something went wrong. Transition matrix, perhaps?")
      quit()
```

سپس بردارهای ویژه چپ ماتریس احتمال انتقال را به‌دست می‌آوریم و بردار ویژه نظیر مقدار ویژه یک را جدا می‌کنیم.

```
eigenvals, eigenvects = np.linalg.eig(transition_matrix.T)
close_to_1_idx = np.isclose(eigenvals,1)
target_eigenvect = eigenvects[:,close_to_1_idx]
target_eigenvect = target_eigenvect[٠,:]
```

بردار ویژه به‌دست آمده را نرمالیزه میکنیم. این بردار ویژه همان توزیع ماناست.

```
stationary_distrib = target_eigenvect / sum(target_eigenvect)
```

با تشکر از توجه شما