HMM

DSC 180B - Group 3

2023-01-27

HMM: Overview

```
states = ('Rainy', 'Sunny') → Hidden variables (X)

observations = ('walk', 'shop', 'clean') → Actions (Y)

start_probability = {'Rainy': 0.6, 'Sunny': 0.4}

transition_probability = {
   'Rainy' : {'Rainy': 0.7, 'Sunny': 0.3},
   'Sunny' : {'Rainy': 0.4, 'Sunny': 0.6},
   }

emission_probability = {
   'Rainy' : {'walk': 0.1, 'shop': 0.4, 'clean': 0.5},
   'Sunny' : {'walk': 0.6, 'shop': 0.3, 'clean': 0.1},
   }
```

states = (“exe1”, exe2”, “exe3”, ...)
observations = (“app/tab1”, “app/tab2”, “app/tab3”, …)
start_probability = {
“chrome.exe”: P(“chrome.exe” appears first in the sequence),
“cmd.exe”: P(“cmd.exe” appears first in the sequence), ….}
transition_probability = {
“chrome.exe”: {
“cmd.exe” : P(“cmd.exe” | “chrome.exe”)
“explorer.exe”: P(“explorer.exe” | “chrome.exe”),
…},
…}
emission_probabiblity = {
“chrome.exe”: {
“google doc”: P(“google doc” | “chrome.exe”),
“google drive”: P(“google drive” | “chrome.exe”),
…},
….}

Transition Probability

From chrome.exe → cmd.exe,

$$P(cmd.exe \mid chrome.exe)$$
$$= \frac{P(chrome.exe, cmd.exe)}{P(chrome.exe)}$$
$$= \frac{\#\ pair\ occurences\ of\ chrome.exe\ and\ cmd.exe}{\#\ all\ occurrences\ of\ chrome.exe}$$

Transition Probability

```
def get_transition_probability(pair_freq, X):
    transition_prob = defaultdict(int)
    for pair in pair_freq:
        total_occ = sum([x == pair[0] for x in X])
        transition_prob[pair] += pair_freq[pair] / total_occ
    return transition_prob
```

From **chrome.exe** → **cmd.exe**,

$$P(cmd.exe \mid chrome.exe)$$
$$= \frac{P(chrome.exe, cmd.exe)}{P(chrome.exe)}$$
$$= \frac{\#\ pair\ occurences\ of\ chrome.exe\ and\ cmd.exe}{\#\ all\ occurrences\ of\ chrome.exe}$$

```
def get_pair_frequency(X, y):
    """Get the frequency of the pairs
    pair_freq = defaultdict(int)
    for index in range(len(X)):
        pair = (X[index], y[index])
        pair_freq[pair] += 1
    return pair_freq
```

```
sum([x == pair[0] for x in X])
```

HMM:
Model + Accuracy

```
def predict_HMM(df, n, rand_state):
    """Put everything together for the HMM model"""
    df = get_clean_data(df)
    all_pairs = get_all_pairs(df)

    X_tr, y_tr, X_test, y_test = split_train_test(all_pairs, rand_state)
    pair_freq = get_pair_frequency(X_tr, y_tr)
    transition_prob = get_transition_probability(pair_freq, X_tr)
    transition_matrix = get_transition_matrix(transition_prob, X_tr)

    accuracy = get_accuracy(X_test, y_test, transition_matrix, n)
    return [transition_matrix, accuracy]
```

- **Train/Test**: 80/20
  - **train_test_split** function from sklearn
- **Transition MT**:
  - Converted from the dictionary of **transition probabilities**
- **Accuracy**:
  - User 2
  - On Jan 27, 2023

Parameters	Accuracy
predict_HMM(df2, n=1, rand_state=20)	37.11 %
predict_HMM(df2, n=2, rand_state=20)	57.22%
predict_HMM(df2, n=5, rand_state=20)	80.41%
predict_HMM(df2, n=10, rand_state=20)	88.66%
predict_HMM(df2, n=15, rand_state=20)	94.33%

HMM:
Model Accuracy

- User 1 on Jan 28, 2023

Parameters	Accuracy
predict_HMM(df1, n=1, rand_state=20)	49.21 %
predict_HMM(df1, n=2, rand_state=20)	67.43%
predict_HMM(df1, n=5, rand_state=20)	87.74%
predict_HMM(df1, n=10, rand_state=20)	96.32%
predict_HMM(df1, n=15, rand_state=20)	98.07%

HMM:
Model Accuracy

- User 2 on Jan 28, 2023

Parameters	Accuracy
predict_HMM(df2, n=1, rand_state=18)	36.42 %
predict_HMM(df2, n=2, rand_state=18)	57.62%
predict_HMM(df2, n=5, rand_state=18)	80.46%
predict_HMM(df2, n=10, rand_state=18)	92.38%
predict_HMM(df2, n=15, rand_state=18)	95.36%

Emission Probability

From **chrome.exe → Google Doc**

$$P(gg\ doc \mid chrome.exe)$$
$$= \frac{P(chrome.exe, gg\ doc)}{P(chrome.exe)}$$
$$= \frac{\#\ pair\ occurences\ of\ chrome.exe\ and\ gg\ doc}{\#\ all\ occurrences\ of\ chrome.exe}$$

Emission Probability

```python
def find_emission_prob(executables, apps, from_exe, to_app):
    """Find the emission probability
    P(to_app | from_exe) = P(from_exe, to_app) / P(from_exe)"""
    emission_numer = find_joint_prob(executables, apps, from_exe, to_app)
    emission_denom = find_exe_prob(executables, from_exe)
    return emission_numer / emission_denom
```

From **chrome.exe** → **Google Doc**,

$$P(gg\ doc \mid chrome.exe) = \frac{P(chrome.exe, gg\ doc)}{P(chrome.exe)}$$

```python
def find_joint_prob(executables, apps, from_exe, to_app):
    fromExe_indices = np.where(executables == from_exe)[0]
    toApp_indices = np.where(apps == to_app)[0]
    co_appear = len(set(fromExe_indices + 1) & set(toApp_indices))
    return co_appear / len(executables)
```

```python
def find_exe_prob(executables, exe_name):
    numerator = sum(exe_name == executables)
    denominator = len(executables)
    return numerator / denominator
```

Emission MT: Results

```
def emission_mt(executables, apps):
    """Find the emission matrix"""
    emission_prob = emission_dict(executables, apps)
    emission_matrix = pd.DataFrame.from_dict(emission_prob)
    return (emission_prob, emission_matrix.T)
```

- User 1
- On Jan 28, 2023

	esrv.exe	Downloads	DSC180A_HW_Week4 - Microsoft Visual Studio	Movies & TV	File Explorer	2023 (CCG DCA UCSD-HDSI Capstone) | Microsoft Teams - Google Chrome	Volume Control	Search	Task Scheduler	Create Task	...	patch-1 - Google Chrome	How to Fix Git Error: You need to resolve your current index first - Google Chrome	• HMM.ipynb - System-Usage-Analysis - Visual Studio Code
VsDebugConsole.exe	0.164634	0.012195	0.036585	0.012195	0.182927	0.000000	0.006098	0.000000	0.012195	0.000000	...	0.00	0.000000	0.000000
explorer.exe	0.018135	0.000000	0.023316	0.020725	0.049223	0.002591	0.067358	0.012953	0.002591	0.000000	...	0.00	0.000000	0.000000
devenv.exe	0.050360	0.000000	0.050360	0.007194	0.100719	0.000000	0.000000	0.007194	0.021583	0.007194	...	0.00	0.000000	0.000000
ApplicationFrameHost.exe	0.000000	0.071429	0.023810	0.000000	0.095238	0.023810	0.023810	0.095238	0.071429	0.166667	...	0.00	0.000000	0.000000
chrome.exe	0.088207	0.013423	0.024928	0.000959	0.212848	0.000000	0.062320	0.016299	0.000959	0.000000	...	0.00	0.000000	0.005753
ShellExperienceHost.exe	0.025862	0.000000	0.000000	0.008621	0.025862	0.000000	0.000000	0.008621	0.000000	0.000000	...	0.00	0.000000	0.000000
SearchApp.exe	0.000000	0.000000	0.000000	0.058824	0.411765	0.000000	0.000000	0.000000	0.000000	0.000000	...	0.00	0.000000	0.000000
Unable To Open Process	0.166667	0.000000	0.083333	0.500000	0.083333	0.000000	0.000000	0.041667	0.000000	0.000000	...	0.00	0.000000	0.000000
SnippingTool.exe	0.000000	0.000000	0.013514	0.006757	0.121622	0.006757	0.000000	0.000000	0.000000	0.000000	...	0.00	0.000000	0.000000
DB Browser for SQLite.exe	0.009091	0.000000	0.018182	0.000000	0.054545	0.000000	0.009091	0.000000	0.000000	0.000000	...	0.00	0.000000	0.000000
Code.exe	0.026882	0.005376	0.010753	0.000000	0.037634	0.000000	0.021505	0.005376	0.000000	0.000000	...	0.00	0.002688	0.000000

Emission MT: Results

```python
def emission_mt(executables, apps):
    """Find the emission matrix"""
    emission_prob = emission_dict(executables, apps)
    emission_matrix = pd.DataFrame.from_dict(emission_prob)
    return (emission_prob, emission_matrix.T)
```

- User 2
- On Jan 28, 2023

	esrv.exe	Foreground - Microsoft Visual Studio	Google Docs - Google Chrome	Messenger	sdk	Public -- 2022-2023 (CCG DCA UCSD-HDSI Capstone) | Microsoft Teams	Search	Administrator: Command Prompt	pip documentation v22.3.1 - Google Chrome	File Explorer	...	output - Notepad	output - Excel	Output Dataframe as txt - Google Chrome
VsDebugConsole.exe	0.220339	0.254237	0.025424	0.016949	0.008475	0.016949	0.042373	0.008475	0.000000	0.084746	...	0.000000	0.000000	0.000000
devenv.exe	0.132353	0.000000	0.000000	0.014706	0.000000	0.029412	0.014706	0.000000	0.000000	0.220588	...	0.000000	0.000000	0.000000
chrome.exe	0.057500	0.042500	0.000000	0.155000	0.000000	0.000000	0.037500	0.045000	0.000000	0.152500	...	0.000000	0.000000	0.000000
Messenger.exe	0.017391	0.008696	0.008696	0.000000	0.000000	0.000000	0.000000	0.008696	0.000000	0.156522	...	0.000000	0.000000	0.000000
explorer.exe	0.134831	0.011236	0.000000	0.082397	0.000000	0.003745	0.029963	0.003745	0.000000	0.011236	...	0.007491	0.011236	0.003745
Teams.exe	0.075758	0.166667	0.015152	0.166667	0.000000	0.000000	0.000000	0.000000	0.000000	0.181818	...	0.000000	0.000000	0.000000
SearchHost.exe	0.060606	0.030303	0.000000	0.000000	0.000000	0.000000	0.000000	0.060606	0.000000	0.212121	...	0.000000	0.000000	0.000000

- Proofread code and add a small fix after receiving Sruti's feedback
- Regenerated the results based on the updated code

Thank you!

- Source Code:
  - https://github.com/miloncl/System-Usage-Analysis/blob/main/HMM.ipynb