Speech Recognition: Assignment #2

Professor Ying Shen

2252709
Xuanhe Yang

October, 2024

Problem 1

Teacher Mood Model Report

Introduction

This report utilizes the Viterbi algorithm to analyze the mood fluctuations of a teacher over a week based on given homework assignments. We aim to infer the teacher's emotional state from the assignments and provide a detailed computation process along with the results.

Task Description

The homework assignments given by the teacher during the week are as follows:

- Monday: A
- Tuesday: C
- Wednesday: B
- Thursday: A
- Friday: C

We represent the observation sequence as `['A', 'C', 'B', 'A', 'C']` and define the states as `['good', 'neutral', 'bad']`, representing good mood, neutral mood, and bad mood, respectively.

Model Definition

States

- good
- neutral
- bad

Observations

- A (Homework A)
- B (Homework B)
- C (Homework C)

Probability Matrices

Initial Probabilities: Assuming equal initial probabilities for all emotional states:

$$\text{start\_prob} = \left[\frac{1}{3}, \frac{1}{3}, \frac{1}{3}\right]$$

Transition Probability Matrix:

$$\text{transition\_prob} = \begin{bmatrix} 0.2 & 0.3 & 0.5 \\ 0.2 & 0.2 & 0.6 \\ 0.0 & 0.2 & 0.8 \end{bmatrix}$$

Emission Probability Matrix:

$$\text{emit\_prob} = \begin{bmatrix} 0.7 & 0.2 & 0.1 \\ 0.3 & 0.4 & 0.3 \\ 0.0 & 0.1 & 0.9 \end{bmatrix}$$

Viterbi Algorithm Implementation

Algorithm Steps

1. **Initialization:** Calculate the probabilities of each state for the first observation.
2. **Dynamic Programming:** For each observation, compute the probabilities for the current state based on the previous time step's state probabilities, and record the path.
3. **Backtracking:** Identify the path with the highest probability and trace back to determine the entire emotional sequence.

Computation Process

Initial Viterbi Matrix

Initially, the Viterbi matrix at time step zero is calculated as follows:

$$P(\text{good}) = P(A|\text{good}) \times P(\text{initial probability of good}) = 0.7 \times \frac{1}{3} = 0.2333$$
$$P(\text{neutral}) = P(A|\text{neutral}) \times P(\text{initial probability of neutral}) = 0.3 \times \frac{1}{3} = 0.1$$
$$P(\text{bad}) = P(A|\text{bad}) \times P(\text{initial probability of bad}) = 0.0 \times \frac{1}{3} = 0.0$$

Thus, the initial Viterbi matrix at time step zero is:

$$\begin{bmatrix} 0.2333 & 0 & 0 & 0 & 0 \\ 0.1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Time Step 1

$$\begin{aligned}\text{Calculating } P(\text{state=good, time=1}) :&\\ P(\text{previous state=good}) &\times P(\text{good} \to \text{good}) \times P(2|\text{good})\\ &= 0.2333333 \times 0.2000000 \times 0.1000000 = 0.0046667\end{aligned}$$

$$\begin{aligned}\text{Calculating } P(\text{state=good, time=1}) :&\\ P(\text{previous state=neutral}) &\times P(\text{neutral} \to \text{good}) \times P(2|\text{good})\\ &= 0.1000000 \times 0.2000000 \times 0.1000000 = 0.0020000\end{aligned}$$

$$\begin{aligned}\text{Calculating } P(\text{state=good, time=1}) :&\\ P(\text{previous state=bad}) &\times P(\text{bad} \to \text{good}) \times P(2|\text{good})\\ &= 0.0000000 \times 0.0000000 \times 0.1000000 = 0.0000000\end{aligned}$$

$$\begin{aligned}\text{Calculating } P(\text{state=neutral, time=1}) :&\\ P(\text{previous state=good}) &\times P(\text{good} \to \text{neutral}) \times P(2|\text{neutral})\\ &= 0.2333333 \times 0.3000000 \times 0.3000000 = 0.0210000\end{aligned}$$

$$\begin{aligned}\text{Calculating } P(\text{state=neutral, time=1}) :&\\ P(\text{previous state=neutral}) &\times P(\text{neutral} \to \text{neutral}) \times P(2|\text{neutral})\\ &= 0.1000000 \times 0.2000000 \times 0.3000000 = 0.0060000\end{aligned}$$

$$\begin{aligned}\text{Calculating } P(\text{state=bad, time=1}) :&\\ P(\text{previous state=good}) &\times P(\text{good} \to \text{bad}) \times P(2|\text{bad})\\ &= 0.2333333 \times 0.5000000 \times 0.9000000 = 0.1050000\end{aligned}$$

$$\begin{aligned}\text{Calculating } P(\text{state=bad, time=1}) :&\\ P(\text{previous state=neutral}) &\times P(\text{neutral} \to \text{bad}) \times P(2|\text{bad})\\ &= 0.1000000 \times 0.6000000 \times 0.9000000 = 0.0540000\end{aligned}$$

After column 1, Viterbi matrix:

$$\begin{bmatrix} 0.23333333 & 0.00466667 & 0 & 0 & 0 \\ 0.1 & 0.021 & 0 & 0 & 0 \\ 0 & 0.105 & 0 & 0 & 0 \end{bmatrix} \tag{1}$$

Time Step 2

$$\begin{aligned}\text{Calculating } P(\text{state=good, time=2}):&\\ P(\text{previous state=good}) &\times P(\text{good} \to \text{good}) \times P(1|\text{good})\\ &= 0.0046667 \times 0.2000000 \times 0.2000000 = 0.0001867\end{aligned}$$

$$\begin{aligned}\text{Calculating } P(\text{state=good, time=2}):&\\ P(\text{previous state=neutral}) &\times P(\text{neutral} \to \text{good}) \times P(1|\text{good})\\ &= 0.0210000 \times 0.2000000 \times 0.2000000 = 0.0008400\end{aligned}$$

$$\begin{aligned}\text{Calculating } P(\text{state=good, time=2}):&\\ P(\text{previous state=bad}) &\times P(\text{bad} \to \text{good}) \times P(1|\text{good})\\ &= 0.1050000 \times 0.0000000 \times 0.2000000 = 0.0000000\end{aligned}$$

$$\begin{aligned}\text{Calculating } P(\text{state=neutral, time=2}):&\\ P(\text{previous state=good}) &\times P(\text{good} \to \text{neutral}) \times P(1|\text{neutral})\\ &= 0.0046667 \times 0.3000000 \times 0.4000000 = 0.0005600\end{aligned}$$

$$\begin{aligned}\text{Calculating } P(\text{state=neutral, time=2}):&\\ P(\text{previous state=neutral}) &\times P(\text{neutral} \to \text{neutral}) \times P(1|\text{neutral})\\ &= 0.0210000 \times 0.2000000 \times 0.4000000 = 0.0016800\end{aligned}$$

$$\begin{aligned}\text{Calculating } P(\text{state=bad, time=2}):&\\ P(\text{previous state=good}) &\times P(\text{good} \to \text{bad}) \times P(1|\text{bad})\\ &= 0.0046667 \times 0.5000000 \times 0.1000000 = 0.0002333\end{aligned}$$

$$\begin{aligned}\text{Calculating } P(\text{state=bad, time=2}):&\\ P(\text{previous state=neutral}) &\times P(\text{neutral} \to \text{bad}) \times P(1|\text{bad})\\ &= 0.0210000 \times 0.6000000 \times 0.1000000 = 0.0012600\end{aligned}$$

After column 2, Viterbi matrix:

$$\begin{bmatrix} 0.23333333 & 0.00466667 & 0.0008400 & 0 & 0 \\ 0.1 & 0.021 & 0.0084 & 0 & 0 \\ 0 & 0.105 & 0.0084 & 0 & 0 \end{bmatrix} \tag{2}$$

Time Step 3

$$\begin{aligned}
\text{Calculating } P(\text{state=good, time=3}) :& \\
P(\text{previous state=good}) \times& \ P(\text{good} \to \text{good}) \times P(0|\text{good}) \\
&= 0.0008400 \times 0.2000000 \times 0.7000000 = 0.0001176
\end{aligned}$$

$$\begin{aligned}
\text{Calculating } P(\text{state=good, time=3}) :& \\
P(\text{previous state=neutral}) \times& \ P(\text{neutral} \to \text{good}) \times P(0|\text{good}) \\
&= 0.0084000 \times 0.2000000 \times 0.7000000 = 0.0011760
\end{aligned}$$

$$\begin{aligned}
\text{Calculating } P(\text{state=neutral, time=3}) :& \\
P(\text{previous state=good}) \times& \ P(\text{good} \to \text{neutral}) \times P(0|\text{neutral}) \\
&= 0.0008400 \times 0.3000000 \times 0.3000000 = 0.0000756
\end{aligned}$$

$$\begin{aligned}
\text{Calculating } P(\text{state=neutral, time=3}) :& \\
P(\text{previous state=neutral}) \times& \ P(\text{neutral} \to \text{neutral}) \times P(0|\text{neutral}) \\
&= 0.0084000 \times 0.2000000 \times 0.3000000 = 0.0005040
\end{aligned}$$

$$\begin{aligned}
\text{Calculating } P(\text{state=bad, time=3}) :& \\
P(\text{previous state=good}) \times& \ P(\text{good} \to \text{bad}) \times P(0|\text{bad}) \\
&= 0.0008400 \times 0.5000000 \times 0.0000000 = 0.0000000
\end{aligned}$$

After column 3, Viterbi matrix:

$$\begin{bmatrix} 0.23333333 & 0.00466667 & 0.0008400 & 0.001176 & 0 \\ 0.1 & 0.021 & 0.0084000 & 0.000504 & 0 \\ 0 & 0.105 & 0.0084000 & 0 & 0 \end{bmatrix} \tag{3}$$

Problem 1 continued on next page. . .

Time Step 4

$$\begin{aligned}
\text{Calculating } P(\text{state=good, time=4}) &: \\
P(\text{previous state=good}) &\times P(\text{good} \to \text{good}) \times P(2|\text{good}) \\
&= 0.0011760 \times 0.2000000 \times 0.1000000 = 0.00002352
\end{aligned}$$

$$\begin{aligned}
\text{Calculating } P(\text{state=good, time=4}) &: \\
P(\text{previous state=neutral}) &\times P(\text{neutral} \to \text{good}) \times P(2|\text{good}) \\
&= 0.0005040 \times 0.2000000 \times 0.1000000 = 0.00001008
\end{aligned}$$

$$\begin{aligned}
\text{Calculating } P(\text{state=bad, time=4}) &: \\
P(\text{previous state=good}) &\times P(\text{good} \to \text{bad}) \times P(2|\text{bad}) \\
&= 0.0011760 \times 0.5000000 \times 0.9000000 = 0.00052920
\end{aligned}$$

$$\begin{aligned}
\text{Calculating } P(\text{state=bad, time=4}) &: \\
P(\text{previous state=neutral}) &\times P(\text{neutral} \to \text{bad}) \times P(2|\text{bad}) \\
&= 0.0005040 \times 0.6000000 \times 0.9000000 = 0.00027216
\end{aligned}$$

After column 4, Viterbi matrix:

$$\begin{bmatrix} 0.23333333 & 0.00466667 & 0.00084 & 0.001176 & 0.00002352 \\ 0.1 & 0.021 & 0.0084 & 0.000504 & 0.00010584 \\ 0 & 0.105 & 0.0084 & 0 & 0.0005292 \end{bmatrix} \tag{4}$$

Final state with max probability: state=bad, probability=0.0005292

Backtracking

$$\begin{aligned}
&\text{Backtracking step at time 4:} \quad \text{state=good} \\
&\text{Backtracking step at time 3:} \quad \text{state=neutral} \\
&\text{Backtracking step at time 2:} \quad \text{state=bad} \\
&\text{Backtracking step at time 1:} \quad \text{state=good}
\end{aligned}$$

Best path: ['good', 'bad', 'neutral', 'good', 'bad']
Best path probability: 0.0005292

Final State and Path

At the final time step (Friday), the state with the highest probability is determined. After backtracking through the path, we obtain:

$$\text{Best Path: } [{}'good', {}'bad', {}'neutral', {}'good', {}'bad']$$

with a corresponding maximum probability.

Conclusion

By employing the Viterbi algorithm, we successfully inferred the emotional fluctuations of the teacher over the week. The results demonstrate notable variations in the teacher's mood, which may relate to the workload and assignment challenges. Future research could explore additional observation variables and emotional states to further enhance the accuracy and applicability of the model.

Code Implementation

```
import numpy as np

# Define states and observations
states = ['good', 'neutral', 'bad']
observations = ['A', 'C', 'B', 'A', 'C']

# Set NumPy print options to suppress scientific notation
np.set_printoptions(suppress=True, floatmode='fixed')

# Transition probability matrix
transition_prob = np.array([
[0.2, 0.3, 0.5],  # From good
[0.2, 0.2, 0.6],  # From neutral
[0.0, 0.2, 0.8]   # From bad
])

# Emission probability matrix
emit_prob = np.array([
[0.7, 0.2, 0.1],  # P(A|good), P(B|good), P(C|good)
[0.3, 0.4, 0.3],  # P(A|neutral), P(B|neutral), P(C|neutral)
[0.0, 0.1, 0.9]   # P(A|bad), P(B|bad), P(C|bad)
])

def viterbi(_obs, _states, _start_prob, _trans_prob, _emit_prob):
n_states = len(_states)  # Number of states
n_obs = len(_obs)        # Number of observations

# Initialize the probability matrix
viterbi_matrix = np.zeros((n_states, n_obs))
path = np.zeros((n_states, n_obs), dtype=int)

# Initialize the first column of the Viterbi matrix
for i in range(n_states):
viterbi_matrix[i][0] = _start_prob[i] * _emit_prob[i][_obs[0]]
path[i][0] = 0  # Starting point for paths

print("Initial Viterbi matrix:")
print(viterbi_matrix)

# Dynamic programming to compute subsequent columns
for t in range(1, n_obs):
for j in range(n_states):
max_prob = -1
max_state = -1
for i in range(n_states):
# Calculate the probability for state j at time t
prob = (viterbi_matrix[i][t - 1] *
_trans_prob[i][j] *
_emit_prob[j][_obs[t]])
```

```
print(f"Calculating P(state={_states[j]}, time={t}):")
print(f"  P(previous state={_states[i]}) * P({_states[i]} -> {_states[j]}) * P({_obs[t
    ]}|{_states[j]})")
print(f"  = {viterbi_matrix[i][t - 1]:.7f} * {_trans_prob[i][j]:.7f} * {_emit_prob[j][
    _obs[t]]:.7f} = {prob:.7f}")

if prob > max_prob:
max_prob = prob
max_state = i  # Store the state that gives the maximum probability
viterbi_matrix[j][t] = max_prob
path[j][t] = max_state  # Store the best previous state

print(f"After column {t}, Viterbi matrix:")
print(viterbi_matrix)

# Backtrack to find the best path
result_path = []
max_final_prob = -1
max_final_state = -1

# Find the state with the maximum probability at the final time step
for i in range(n_states):
if viterbi_matrix[i][n_obs - 1] > max_final_prob:
max_final_prob = viterbi_matrix[i][n_obs - 1]
max_final_state = i

result_path.append(max_final_state)  # Add the best final state to the path
print(f"Final state with max probability: state={_states[max_final_state]}, probability={
    max_final_prob:.7f}")

# Backtracking through the path
for t in range(n_obs - 1, 0, -1):
prev_state = path[result_path[0]][t]
result_path.insert(0, prev_state)  # Insert the best previous state
print(f"Backtracking step at time {t}: state={_states[prev_state]}")

return result_path, max_final_prob  # Return the best path and its probability

# Initial probabilities, assuming equal initial probabilities for all states
start_prob = np.array([1/3, 1/3, 1/3])

# Convert observation sequence to indices
obs_indices = [0, 2, 1, 0, 2]  # A=0, B=1, C=2

# Execute the Viterbi algorithm
best_path, best_prob = viterbi(obs_indices, states, start_prob, transition_prob,
    emit_prob)

# Convert state indices back to state names
best_path_states = [states[i] for i in best_path]

print("Best path:", best_path_states)
print(f"Best path probability: {best_prob:.7f}")
```