HW 6: Clustering, LSTM, Transformers & Reinforcement Learning CPE 490/590 ST

Instructor: Rahul Bhadani

Due: April 25, 2025, 11:59 PM
100 points

You are allowed to use a generative model-based AI tool for your assignment. However, you must submit an accompanying reflection report detailing how you used the AI tool, the specific query you made, and how it improved your understanding of the subject. You are also required to submit screenshots of your conversation with any large language model (LLM) or equivalent conversational AI, clearly showing the prompts and your login avatar. Some conversational AIs provide a way to share a conversation link, and such a link is desirable for authenticity. Failure to do so may result in actions taken in compliance with the plagiarism policy.

Additionally, you must include your thoughts on how you would approach the assignment if such a tool were not available. Failure to provide a reflection report for every assignment where an AI tool is used may result in a penalty, and subsequent actions will be taken in line with the plagiarism policy.

Submission instruction:

Submission instruction for this homework supersedes one mentioned in the Syllabus.

This homework requires all answers recorded in a single `.ipynb` Python notebook. You may upload hand-written PDF for the theory portion. You can use a combination of text cell (i.e. markdown formatted cell) and code cell to provide your answer. To add equations you should be able to use Latex syntax in the text cells of your Python notebook. As a part of your submission, you must provide executed notebook with code, text, and outputs. Alternatively, you can also provide a url (whose permission you must change to 'anyone with link can view') of your Python notebook from Google Colab. The naming convention for your notebook should follow the format {firstname_lastname}_CPE 490/590 ST_hw06.ipynb. For example, if your name is Sam Wells, and you are enrolled in CPE 490 your file name should be sam_wells_CPE490_hw06.ipynb.

Please refer to https://github.com/rahulbhadani/CPE490_590_Sp2025/blob/master/Code/ for hands-on.

Please refer to https://github.com/rahulbhadani/iris_classifier_app for an example web app to make an inference on IRIS dataset.

Theory

1 K-means (5 Points)

Given $\{(1,2),(2,1),(4,5),(5,4)\}$, perform one iteration of K-mens with initial centroids at $(1,1)$ and $(5,5)$.

Answer

Let's call the centroid centers as $C1(1,1)$ and $C2(5,5)$.
Let's first calculate the Euclidean distance of each point from the centroid $C1$:

$$\begin{aligned} A1 - \mathrm{C}1 &= \sqrt{(1-1)^2 + (2-1)^2} = 1 \\ A2 - \mathrm{C}1 &= \sqrt{(2-1)^2 + (1-1)^2} = 1 \\ A3 - \mathrm{C}1 &= \sqrt{(4-1)^2 + (5-1)^2} = 5 \\ A4 - \mathrm{C}1 &= \sqrt{(5-1)^2 + (4-1)^2} = 5 \end{aligned} \tag{1}$$

Similarly, the Euclidean distance of each point from the centroid $C2$ is:

$$\begin{aligned} A1 - \mathrm{C}2 &= \sqrt{(1-5)^2 + (2-5)^2} = 5 \\ A2 - \mathrm{C}2 &= \sqrt{(2-5)^2 + (1-5)^2} = 5 \\ A3 - \mathrm{C}2 &= \sqrt{(4-5)^2 + (5-5)^2} = 1 \\ A4 - \mathrm{C}2 &= \sqrt{(5-5)^2 + (4-5)^2} = 1 \end{aligned} \tag{2}$$

Then A1 and A2 belongs to the first cluster and A3 and A4 belongs to the second cluster.
New Centeroids

$$\mathrm{C}1 = \frac{1+2}{2}, \frac{2+1}{2} = (1.5, 1.5) \tag{3}$$

and

$$C2 = \frac{4+5}{2}, \frac{4+5}{2} = (4.5, 4.5) \tag{4}$$

2 DBSCAN (5 Points)

Explain how increasing ϵ (epsilon) affects the number of clusters in DBSCAN.

Answer

ϵ is the maximum distance between two points for them to be in the same neighborhood. Small epsilon leads to tighter clusters. However, small ϵ leads to high number of noisy points. With large ϵ value, we can get larger clusters. A larger radius allows more points to be connected to a cluster, reducing the number of noise points. However, it can lead to over-generalization.

3 Anomaly Detection in Clustering (5 Points)

How would you use clustering techniques to detect anomaly?

Answer

Clustering tecnique can group together data-points corresponding to normal behavior in same or similar clusters. The best clustering algorithm in this situation might be Gaussian Mixture Model or DBSCAN.

4 Forward Pass in RNN (5 Points)

Given the input $x_t = [0.5]$, weight matrices $W_{xh} = [0.8]$, $W_{hh} = [0.2]$, bias $b_h = [0.1]$, and previous hidden state $h_{t-1} = [0.3]$. Use tanh activation, calculate h_t.

Answer

$$
\begin{aligned}
h_t &= \tanh(W_{xh} \cdot x_t + W_{hh} h_{t-1} + b_h) \\
&= \tanh(0.8 \times 0.5 + 0.2 \times 0.3 + 0.1) \\
&= \tanh(0.4 + 0.06 + 0.1) \\
&= \tanh(0.56) \\
&\approx 0.508
\end{aligned} \tag{5}
$$

5 RNN Cell (15 Points)

Consider a single recurrent neural network (RNN) cell. The cell has an input x_t at time step t, a hidden state h_{t-1} from the previous time step, and computes the new hidden state h_t and output y_t. The cell uses the following formulas to update its state and compute the output:

$$h_t = \sigma(W_h \cdot h_{t-1} + W_x \cdot x_t + b_h)$$

$$y_t = W_y \cdot h_t + b_y$$

where W_h and W_x are weight matrices for the hidden state and input, respectively, b_h is the bias for the hidden state, W_y is the weight matrix for the output, b_y is the bias for the output. h_t formula contains recurrent connection as we saw in the class. σ is the sigmoid activation function.

Given:

1.
$$W_h = \begin{bmatrix} 0.5 & -0.6 \\ 0.8 & -0.1 \end{bmatrix} \tag{6}$$

2.
$$W_x = \begin{bmatrix} -0.2 \\ 0.4 \end{bmatrix} \tag{7}$$

3.
$$b_h = \begin{bmatrix} -0.1 \\ 0.2 \end{bmatrix} \tag{8}$$

4.
$$W_y = \begin{bmatrix} 0.3 & -0.7 \end{bmatrix} \tag{9}$$

5.
$$b_y = \begin{bmatrix} 0.1 \end{bmatrix} \tag{10}$$

6.
$$h_{t-1} = \begin{bmatrix} 0.7 \\ -0.8 \end{bmatrix} \tag{11}$$

7.
$$x_t = \begin{bmatrix} 0.5 \end{bmatrix} \tag{12}$$

Calculate h_t and y_t for time step t.

Answer

$$\begin{aligned} W_h \cdot h_{t-1} + W_x \cdot x_t + b_h &= \begin{bmatrix} 0.5 & -0.6 \\ 0.8 & -0.1 \end{bmatrix} \times \begin{bmatrix} 0.7 \\ -0.8 \end{bmatrix} + \begin{bmatrix} -0.2 \\ 0.4 \end{bmatrix} \times \begin{bmatrix} 0.5 \end{bmatrix} + \begin{bmatrix} -0.1 \\ 0.2 \end{bmatrix} + \begin{bmatrix} -0.1 \\ 0.2 \end{bmatrix} \\ &= \\ &= \begin{bmatrix} 0.83 \\ 0.64 \end{bmatrix} + \begin{bmatrix} -0.1 \\ 0.2 \end{bmatrix} + \begin{bmatrix} -0.1 \\ 0.2 \end{bmatrix} \\ &= \begin{bmatrix} 0.83 \\ 0.64 \end{bmatrix} + \begin{bmatrix} -0.20 \\ 0.40 \end{bmatrix} = \begin{bmatrix} 0.63 \\ 1.04 \end{bmatrix} \end{aligned} \tag{13}$$

$$h_t = \sigma(W_h \cdot h_{t-1} + W_x \cdot x_t + b_h) = \begin{bmatrix} \sigma(0.63) \\ \sigma(1.04) \end{bmatrix} = \begin{bmatrix} \dfrac{1}{1 + \exp(-0.63)} \\ \dfrac{1}{1 + \exp(-1.04)} \end{bmatrix} = \begin{bmatrix} 0.652 \\ 0.739 \end{bmatrix} \tag{14}$$

$$\begin{aligned} y_t &= W_y \cdot h_t + b_y = \begin{bmatrix} 0.3 & -0.7 \end{bmatrix} \times \begin{bmatrix} 0.652 \\ 0.739 \end{bmatrix} + \begin{bmatrix} 0.1 \end{bmatrix} \\ &= 0.1956 - 0.5173 + 0.1 = -0.2217 \end{aligned} \tag{15}$$

6 Scaled Dot-Product Attention (5 Points)

Consider Query, Key, Value:

$$\begin{aligned} Q &= \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \\ K &= \begin{bmatrix} 2 & 1 \\ 4 & 3 \end{bmatrix} \\ V &= \begin{bmatrix} 0.5 & 1.0 \\ 1.5 & 2.0 \end{bmatrix} \end{aligned} \tag{16}$$

Compute the output of the scaled dot-product attention (without masking) given the above query, key, and value.

Answer

Scaled dot product attention is given By

$$\begin{aligned} \text{Attention}(Q, K, V) &= \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V \\ QK^\top &= \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \times \begin{bmatrix} 2 & 4 \\ 1 & 3 \end{bmatrix} = \begin{bmatrix} 1 \cdot 2 + 2 \cdot 1 & 1 \cdot 4 + 2 \cdot 3 \\ 3 \cdot 2 + 4 \cdot 1 & 3 \cdot 4 + 4 \cdot 3 \end{bmatrix} = \begin{bmatrix} 4 & 10 \\ 10 & 24 \end{bmatrix} \\ \frac{QK^T}{\sqrt{2}} &= \begin{bmatrix} 2.828 & 7.071 \\ 7.071 & 16.970 \end{bmatrix} \\ \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right) &= \begin{bmatrix} 0.014166 & 0.98583 \\ 5.0197 \times 10^{-5} & 0.99995 \end{bmatrix} \\ \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V &= \begin{bmatrix} 1.4858 & 1.9858 \\ 1.4999 & 1.9999 \end{bmatrix} \end{aligned} \tag{17}$$

Practice

See notebook for Solution

7 Markov Decision Process (45 Points)

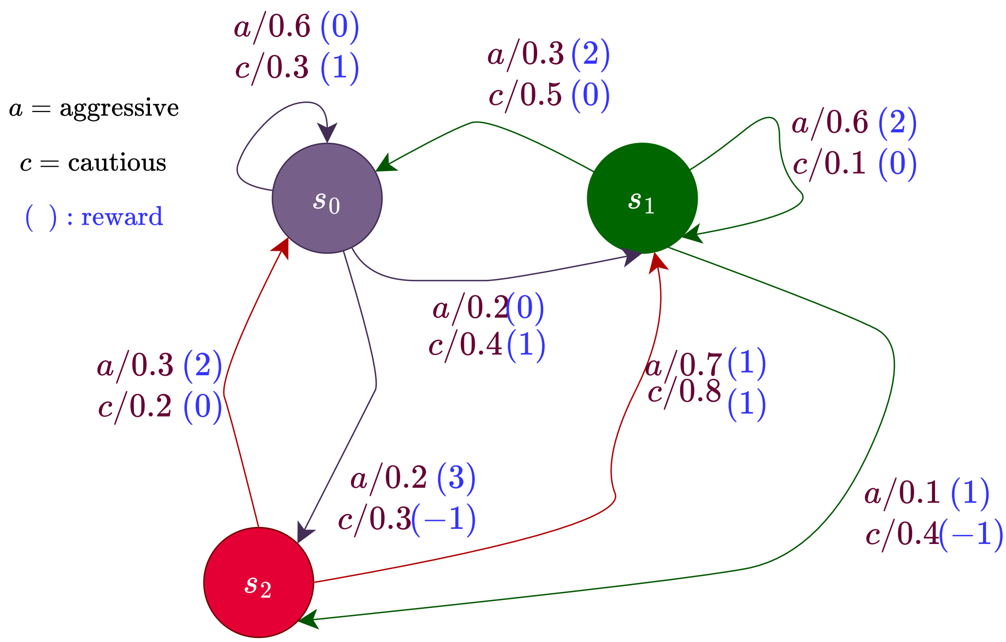

Figure 1: Q7

Write Python Code for the following

1. Consider Markov Decision Process given in Figure 1. Compute the steady-state distribution under action a as well as under action c by solving Eigenvalue problem. (10 Points).
2. By randomly transitioning states, simulate the state transition, and compute the simulated steady-state distribution for each actions. (15 Points).
3. Now, consider actions and reward, compute the optimal policy using policy iteration and value iteration. (20 Points).

8 K-means (15 points)

Generate a half-moon dataset containing 500 samples using Python. Use `noise=0.2`. Implement a K-mean clustering on the half-moon dataset. Evaluate the cluster quality on four metrics: (i) Adjusted Rand Index, (ii) V-measure, (iii) Silhouette Score, and (iv) Calinski-Harabasz Index.